

Inhaltsverzeichnis

1 Spannung messen und Grenzwertüberwachung.....	3
1.1 Simulierten Sensorwert anzeigen und Grenzwerte überwachen.....	3
1.2 Schieberegler für Grenzwerte, Standardwerte einstellen.....	4
1.3 Verlaufsdigramm hinzufügen.....	5
1.4 Hardware hinzufügen: Potentiometer als Füllstandssensor.....	6
2 FOR-Schleife, Graph und Diagramm.....	8
2.1 FOR-Schleifen-Test.....	8
2.2 FOR-Schleifen-Test mit Array.....	8
2.3 Quadrat-Zahlen im Array darstellen.....	8
2.4 Quadrat-Zahlen im Array darstellen. Array zu Beginn löschen. Eigenschaftsknoten erstellen.	9
2.5 Quadratzahlen mit Signalverlaufsgraph außerhalb der Schleife.....	10
2.6 Quadratzahlen mit Signalverlaufsgraph außerhalb der Schleife, Array-Werte und Graph bei Programmstart löschen.....	10
2.7 Quadratzahlen mit Signalverlaufsdigramm innerhalb der Schleife.....	11
2.8 Quadratzahlen mit Signalverlaufsdigramm innerhalb der Schleife Diagramm zu Beginn löschen.....	11
3 Übung Polynomzeichner mit For-Schleife, Formelknoten, Array.....	12
3.1 Datentypen.....	12
3.2 Schleifen.....	12
3.3 Formelknoten.....	12
3.4 Array.....	12
3.5 Schleifentunnel.....	13
3.6 Grafische Darstellung.....	13
3.7 Polynomzeichner mit ungeradzahligen Werten von X.....	13
4 Case-Struktur.....	15
4.1 Case mit 2 Alternativen: True und False.....	15
4.2 Case mit mehreren Alternativen und Textring.....	15
4.3 Case mit mehreren Alternativen und Optionsfeld.....	15
5 Kennlinie automatisch aufnehmen.....	16
5.1 Spannung ausgeben und Spannung messen.....	16
5.2 Manuelle Kennlinienaufnahme.....	17
5.3 Automatische Kennlinienaufnahme mit 10 Messwerten.....	18
5.4 Automatische Kennlinienaufnahme 100 Messwerte.....	20
5.5 Kennlinienaufnahme mit Rückkopplungsknoten und „Array erstellen“.....	22
5.6 Diagramm jedes Mal löschen.....	24
5.7 Diagramm löschen wählbar / warten auf neue Messwertaufnahme.....	25
6 Widerstand und Temperatur messen.....	27
6.1 Einfache Widerstandsmessung.....	27
6.2 Widerstandsmessung und Temperaturbestimmung mit PT1000.....	27

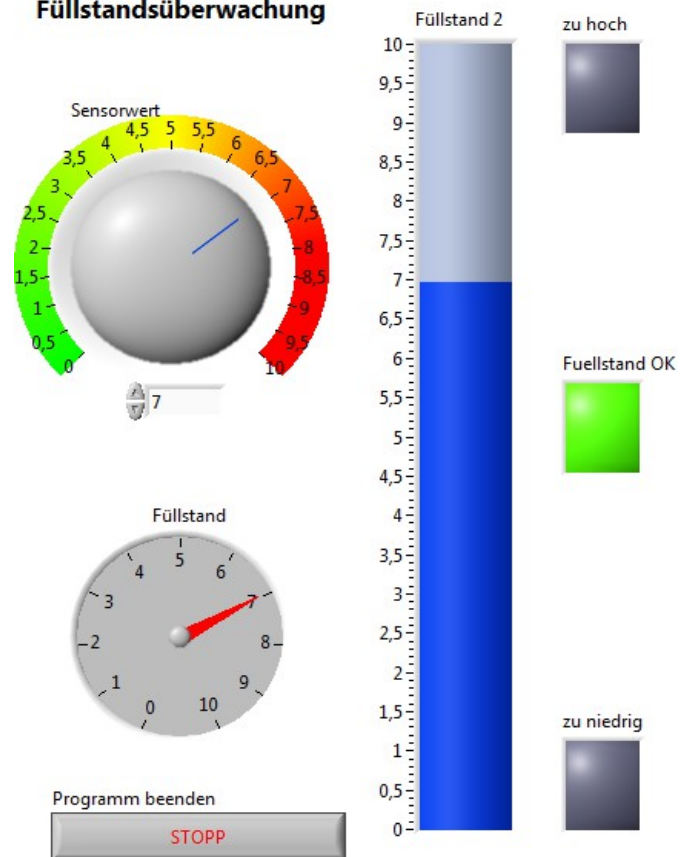
7 Kennlinienaufnahme Solarzellen.....	28
7.1 MPP-Suche mit 100 U-I-Messwerten eines Solarpanels.....	30
7.2 Theorieteil.....	34
8 Datenaustausch PC – Controller über die COM-Schnittstelle.....	35
8.1 Datenübertragung COM-Schnittstelle Controller.....	35
8.2 COM-Schnittstelle mit Umwandlung der Variablenformate.....	36
8.3 Fernsteuerung einer RGB-LED-Lichterkette am Controller über Labview am PC.....	38

1 Spannung messen und Grenzwertüberwachung

1.1 Simulierten Sensorwert anzeigen und Grenzwerte überwachen

- While-Schleife: Programm wird dauernd wiederholt bis Stopp gedrückt.
- Drehregler gibt einen Wert zwischen 0 und 10 vor.
- Zeiger-Anzeigeinstrument zeigt den vom Drehregler vorgegebenen Wert an.
- Tankanzeige ist parallel geschaltet, zeigt also das Gleiche an
- LEDs neben der Tankanzeige zeigen an, in welchem Bereich die Spannung ist.
- Die Vergleiche und die logische Verknüpfung UND ermitteln die Bereiche.

Füllstandsüberwachung



Handling-Anleitung

Labview-Startfenster → Datei → Neu
 Menüleiste → Fenster → Nebeneinander
 Menüleiste → Ansicht → Werkzeugpalette
 • in die Mitte zwischen beide Fenster

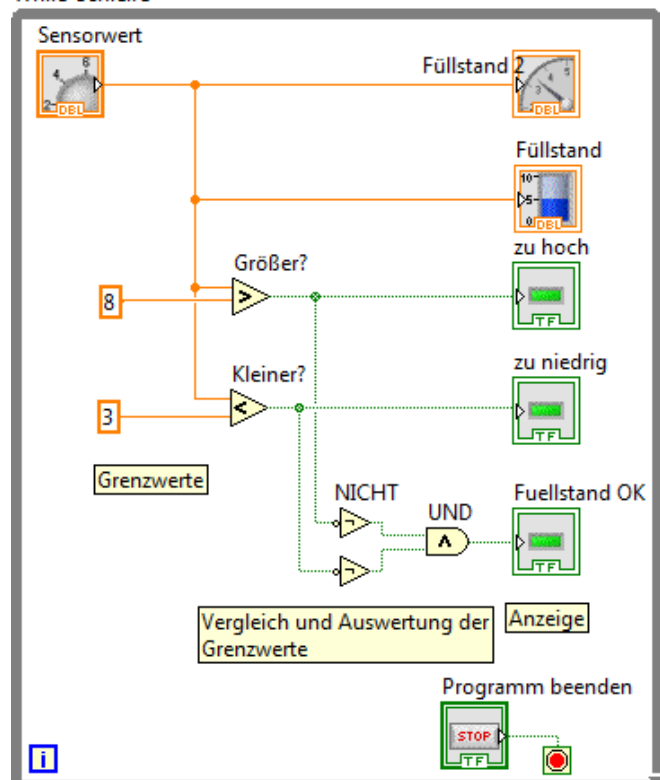
Frontpanel → Modern → Numerisch → Drehregler
 Frontpanel → Modern → Numerisch → Rundinstrument
 Komponenten auf Blockdiagramm verbinden
 Blockdiagramm → Rechtsklick → Programmierung → Strukturen → while-Schleife über den beiden Komponenten aufziehen
 Rechtsklick auf den roten Punkt der Schleife (Schleifenbedingung) → Erstellen → Bedienelement → Stopp-Button entsteht
 Programm mit dickem weißen Pfeil unter der Menüleiste starten

LEDs im Menü des Frontpanels unter Boolesch

Größer, Kleiner im Menü des Blockdiagramms unter Vergleich

Logisches UND und NICHT im Menü des Blockdiagramms unter Boolesch

While-Schleife



1.2 Schieberegler für Grenzwerte, Standardwerte einstellen

Handling-Anleitung

Feste zahlen der Grenzwerte durch Schieberegler ersetzen

- Frontpanel → Rechtsklick → Modern → Numerisch → Schieber mit Zeiger
- Auf gleiche Größe wie den Füllstandstank ziehen

Zahlenanzeige sichtbar bei Schieberegler:

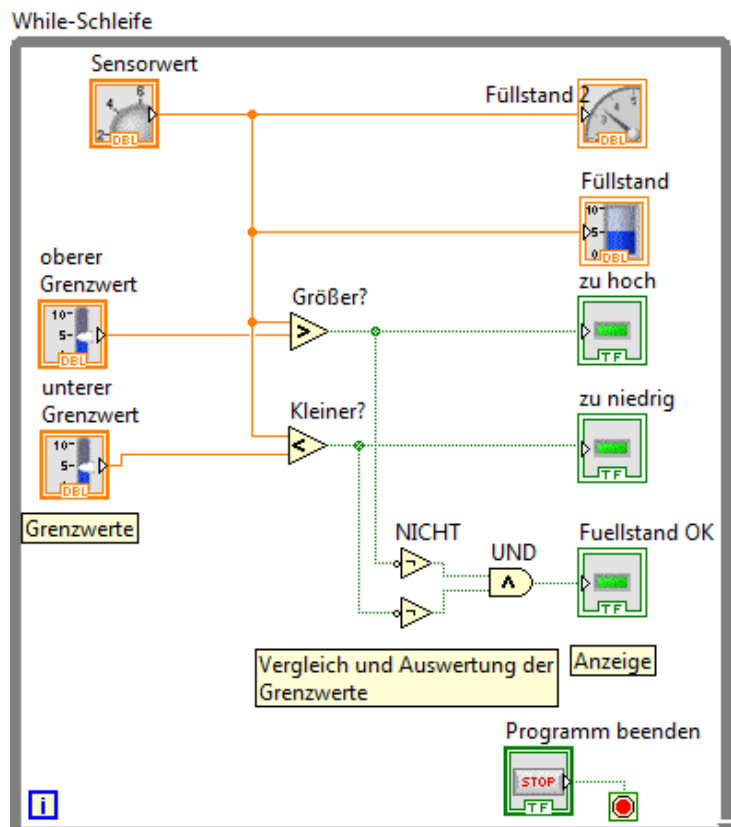
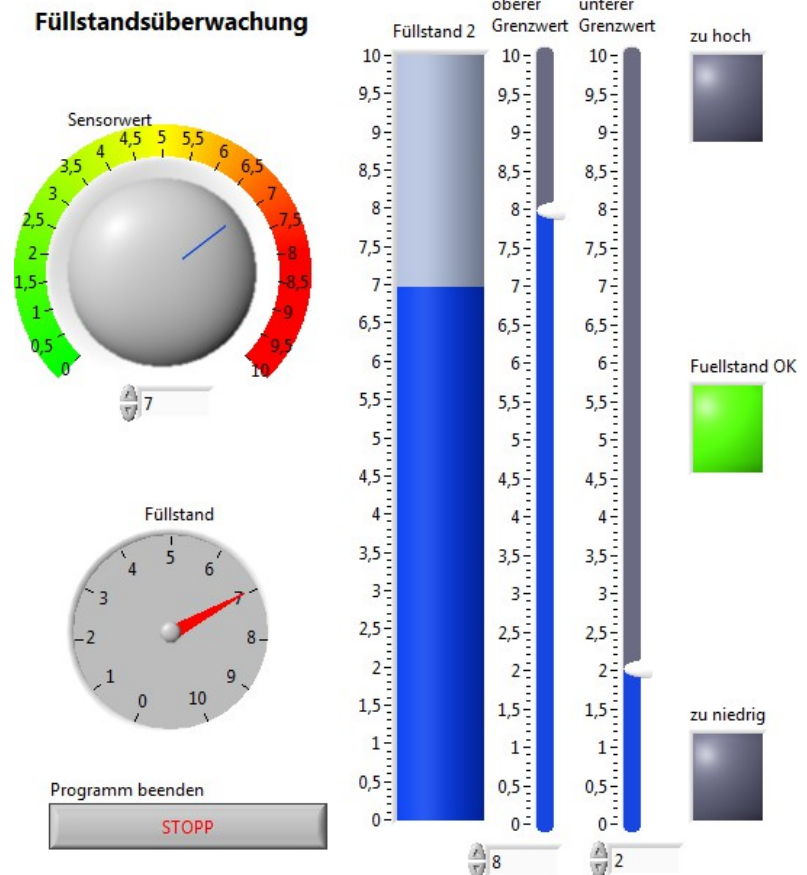
- Rechtsklick auf Schieber → Eigenschaften → Reiter Darstellung → Zahlenanzeige einblenden

Nur eine Nachkommastelle bei Anzeige:

- Reiter Anzeigeformat → nacheinander Numerisch und Zahlenanzeige0 wählen und Anzahl der Stellen auf 1 reduzieren

Standardwerte einstellen: am Schieberegler gewünschten Wert einstellen, dann:

- Rechte Maus auf Schieberegler → Datenoperation → aktuellen Wert als Standard



1.3 Verlaufsdiagramm hinzufügen

Handling-Anleitung

Verlaufsdiagramm hinzufügen

- Frontpanel → Rechtsklick → Modern → Graph → Diagramm

mehrere Kanäle im Diagramm anzeigen

- Blockdiagramm → Rechtsklick → Programmierung → Cluster → Bündeln
- Komponenten Bündeln am Markierungskästchen aufziehen bis 3 Felder sichtbar
- Sensorwert, und beide Grenzwerte anschließen
- erst jetzt Komponente Bündeln ans Verlaufsdiagramm anschließen

Diagramm formatieren

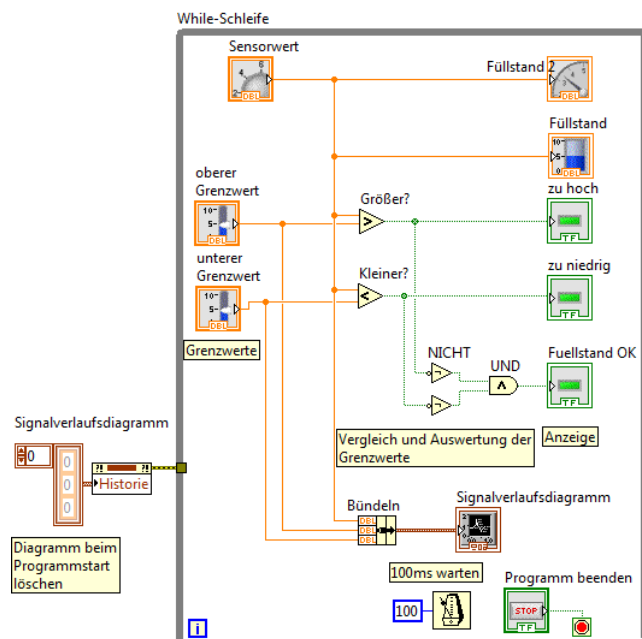
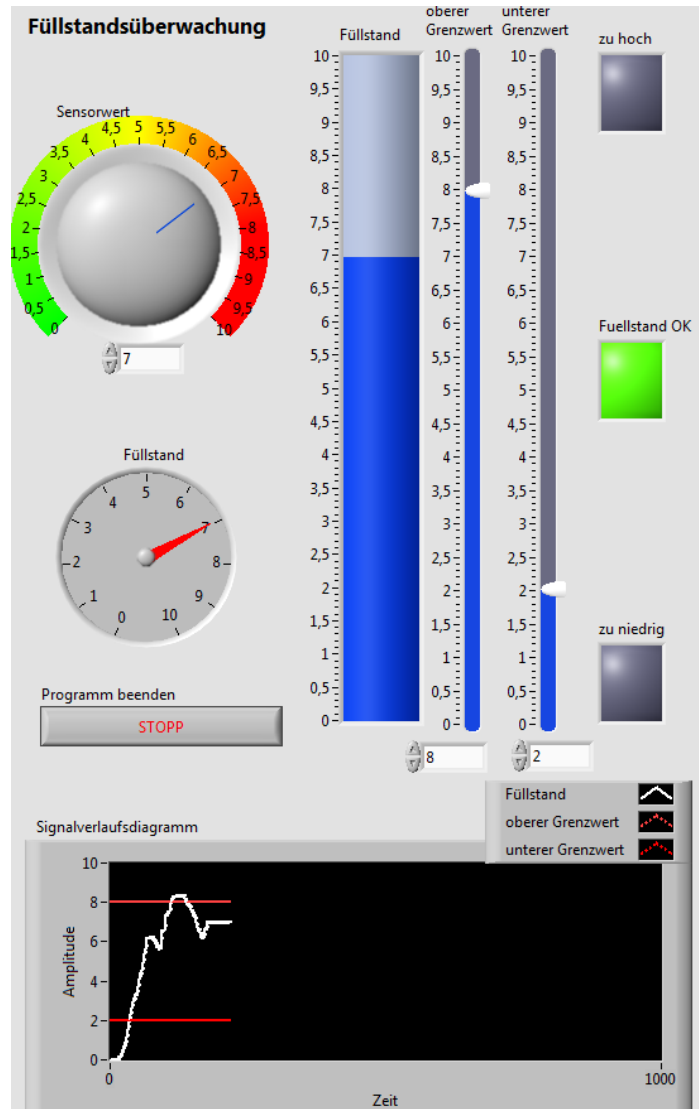
- Frontpanel → Rechtsklick auf das Diagramm → Eigenschaften
- Reiter Skalen → Amplitude (y-Achse) wählen → Autoskalierung ausschalten und gewünschte Min Max-Werte eingeben
- Reiter Plots → nacheinander Plot1, Plot2, Plot3 wählen: Name, Linienart und Farbe ändern

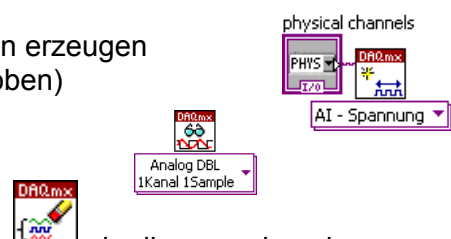
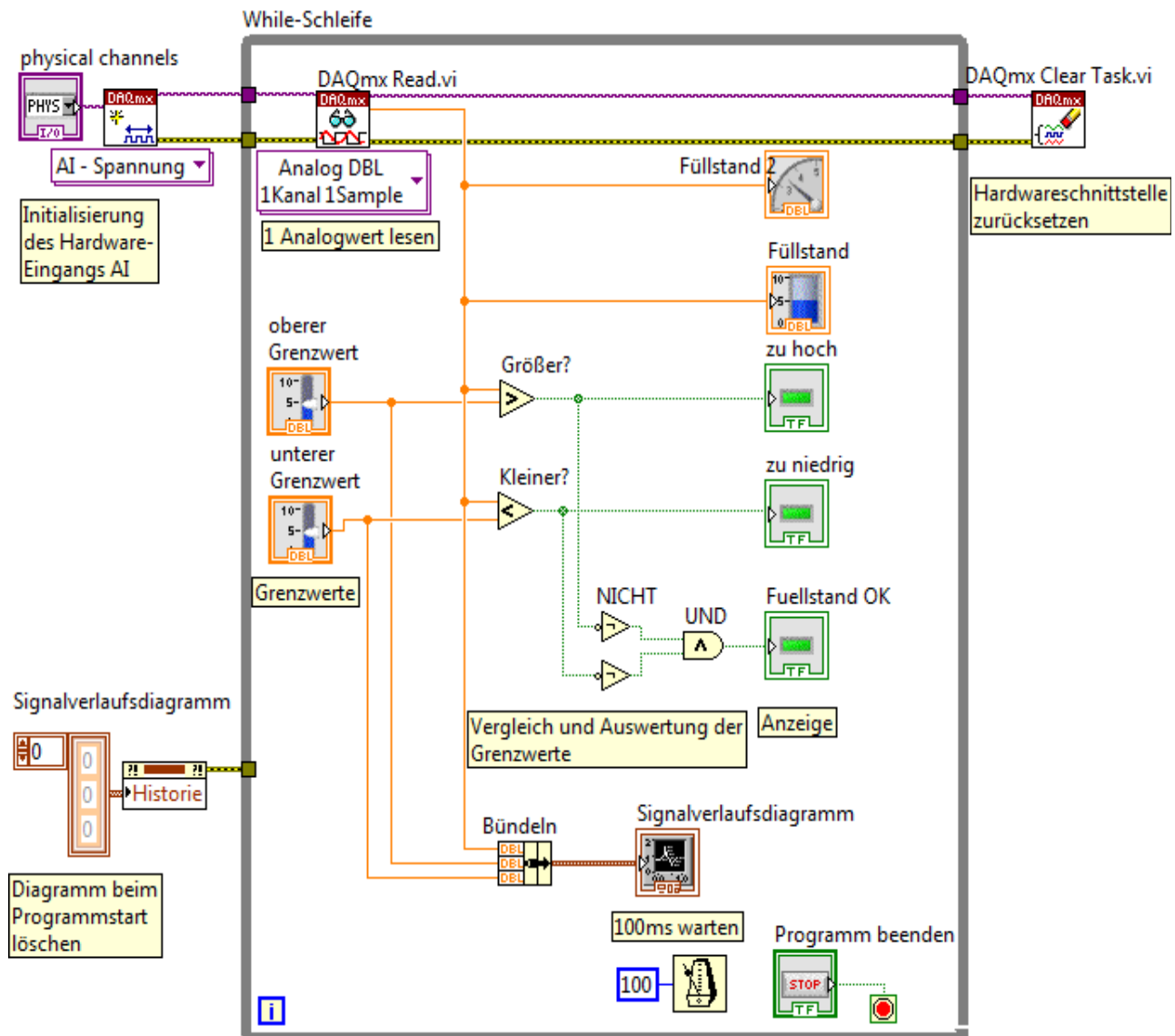
Diagramm beim Programmstart löschen

- Blockdiagramm → Rechtsklick auf das Diagramm → Erstellen → Eigenschaftsknoten → Historiendaten
- Komponente außerhalb der while-Schleife platzieren
- Rechtsklick → in schreiben ändern
- Rechtsklick auf den Eingang → Erstellen Konstante
- Ausgang der Komponente auf den while-Schleifen-Rand ziehen, jetzt wird das Nullsetzen der Historiendaten ausgeführt bevor die while-Schleife ausgeführt wird.

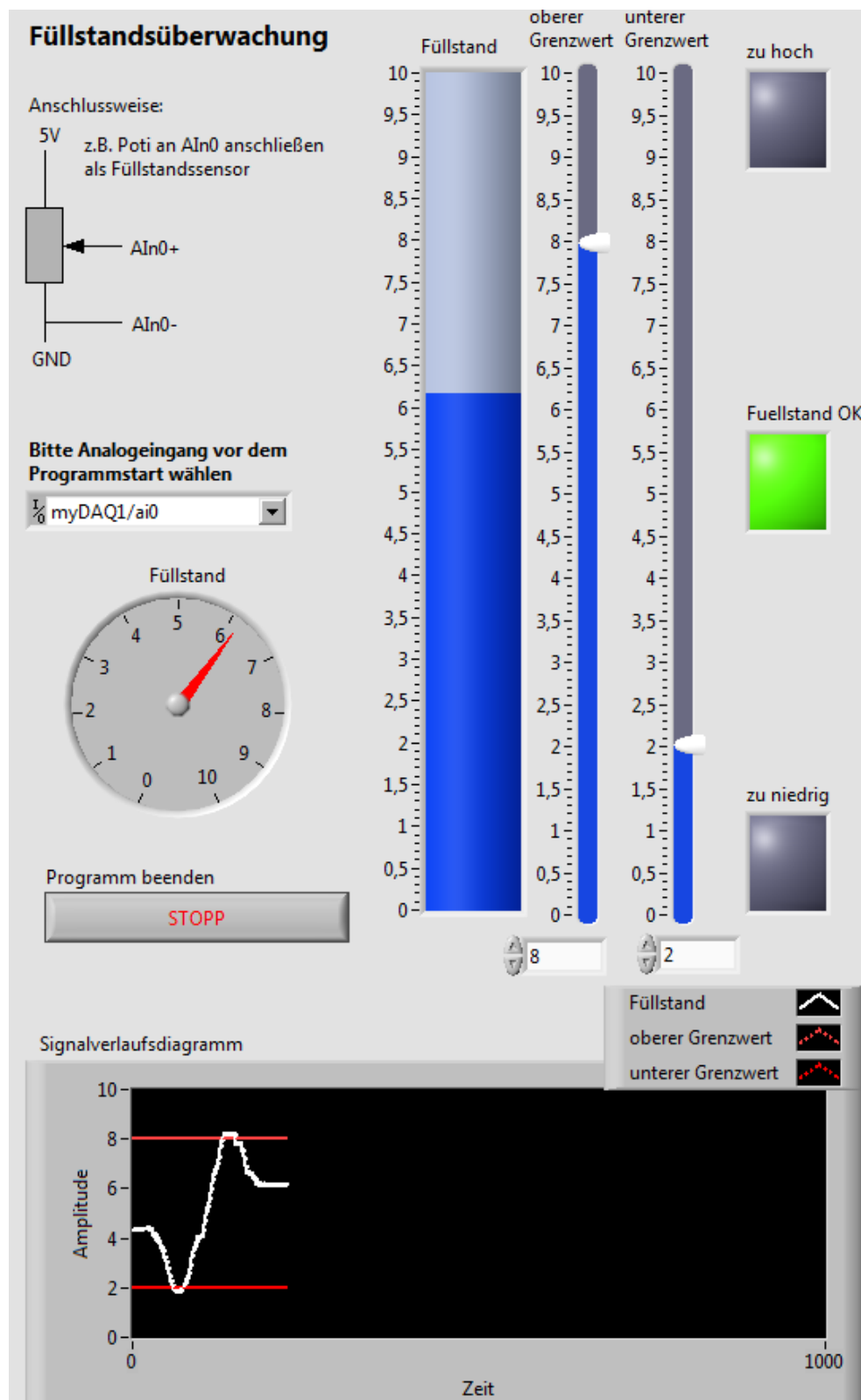
Timer hinzufügen

- Blockdiagramm → Rechtsklick → Programmierung → Timing → warten(Vielfaches)
- Rechtsklick auf den Eingang → Erstellen Konstante → 100
- → 100ms warten





DAQmx Lesen wird für jede neue Messung benötigt, daher Platzierung in der Schleife.
Task zurück erfolgt erst beim Programmende, also rechts außerhalb der Schleife.

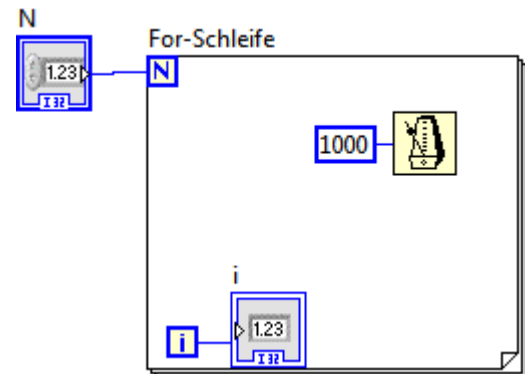


2 FOR-Schleife, Graph und Diagramm

2.1 FOR-Schleifen-Test

Testen Sie die Funktion der For-Schleife:

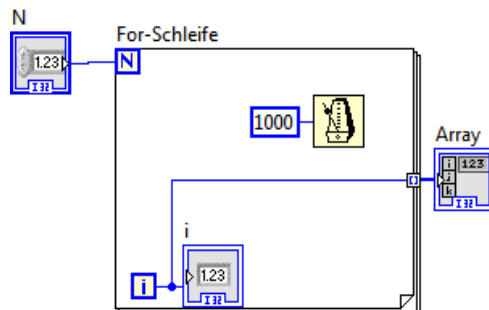
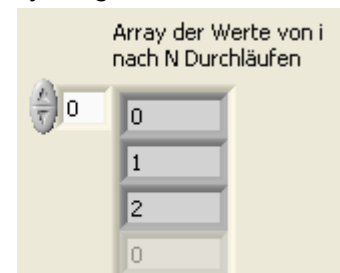
- Platzieren der For-Schleife auf dem Blockschaltbild
- Bei N ein Eingabefeld erstellen
- i soll angezeigt werden
- Zeitverzögerung wie in 1.4
- N gibt die Anzahl der Schleifendurchläufe an und ist eine Integerzahl (ganze Zahl)
- i ist die Laufvariable und zählt die Anzahl der aktuellen Schleifendurchläufe mit.



2.2 FOR-Schleifen-Test mit Array

Nach Programm-Ende sollen alle Werte der Schleife in einem Array dargestellt werden.

- Leitung an i anschließen und bis zum rechten Rand der For-Schleife ziehen → es entsteht ein Schleifentunnel.
- Am Schleifentunnel außerhalb der Schleife ein Anzeigeelement erstellen
 Dies wird automatisch ein Array aus Integer-Werten.
 Das Array mit der Maus nach unten „aufziehen“.



- Im Schleifentunnel werden die Werte von i gesammelt und nach Beendigung der Schleife an das Array ausgegeben.

2.3 Quadrat-Zahlen im Array darstellen


Nun sollen statt der Laufvariablen i im Array die Quadratzahlen von i dargestellt werden. Verwenden Sie x^2 und nicht den Formelknoten.




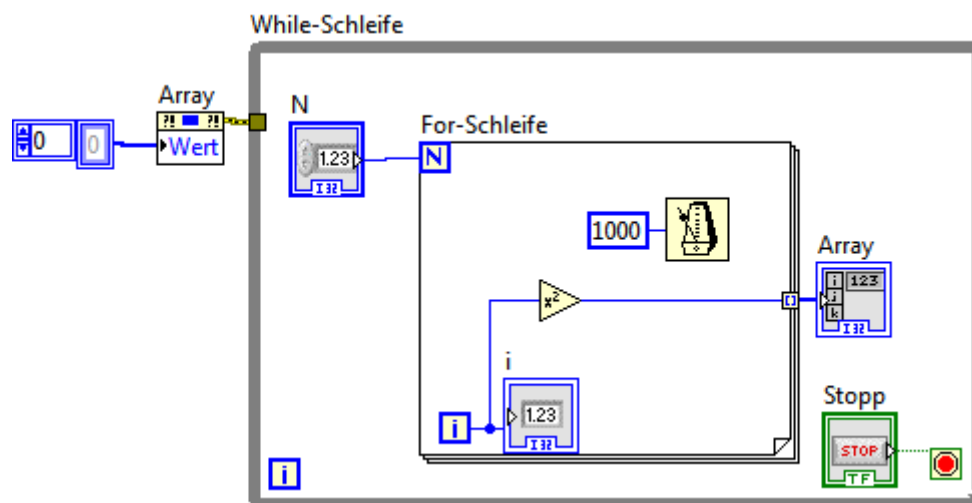
2.4 Quadrat-Zahlen im Array darstellen. Array zu Beginn löschen. Eigenschaftsknoten erstellen.

Die Zahlen im Array sollen zu Beginn des Programmstarts gelöscht werden:

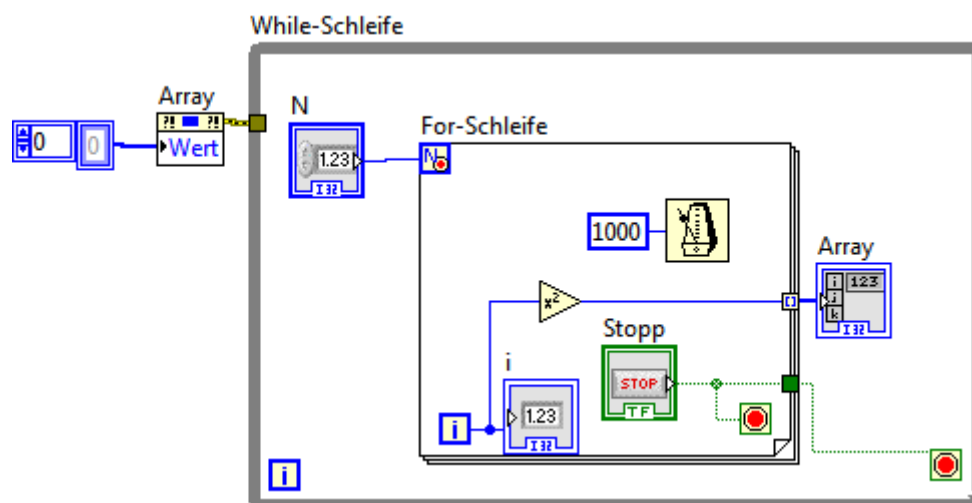
- Speichern und schließen Sie die Datei, anschließend wieder öffnen.
- Klicken Sie rechts auf das Array und erstellen Sie einen Eigenschaftsknoten → Wert
 Damit kann an jeder beliebigen Stelle des Programms auf das bestehende Array zugegriffen werden, um z.B. alle Werte zu löschen.
- Rechtsklick auf Wert → in Schreiben ändern
- Konstante 0 anschließen.
- Um eine gesicherte Reihenfolge in der Programm-Abarbeitung zu gewährleisten (zuerst löschen, dann Schleife ausführen, muss man „irgendeinen“ Ausgang des Eigenschaftsknotens (z.B. den Fehlerausgang) auf der linken Seite der Schleife anschließen.
- Sehen Sie sich den Programmablauf auch im Debug-Modus (Lampe) an

Array der Quadratzahlen


Array der Quadratzahlen




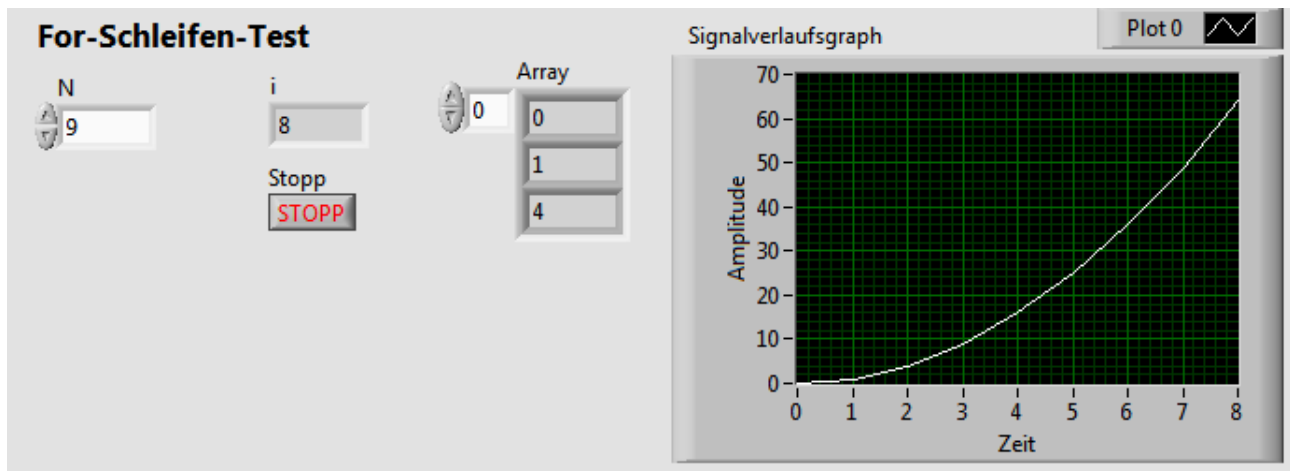
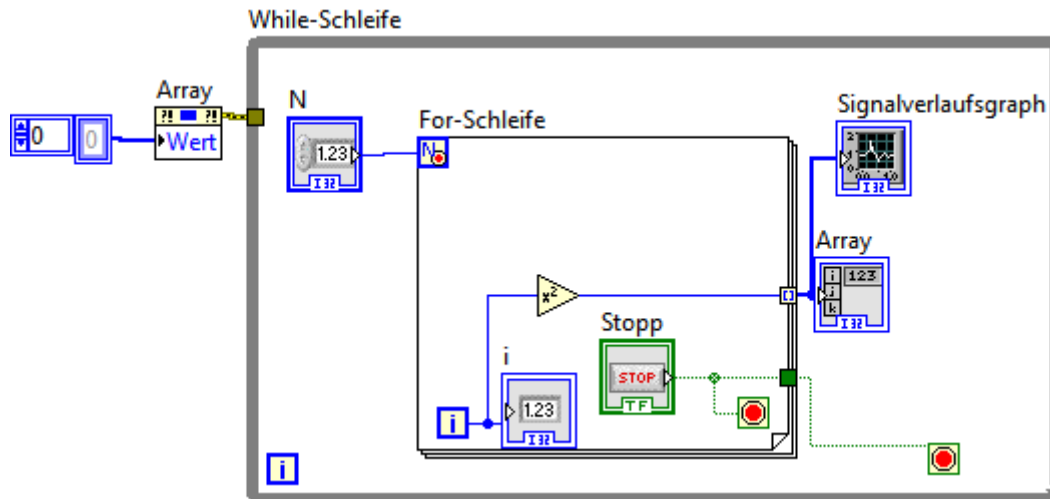
- Bei Programmstart werden die Werte des Arrays gelöscht.



2.5 Quadratzahlen mit Signalverlaufsgraph außerhalb der Schleife

Verwenden Sie einen Signalverlaufs-Graph (ausserhalb der FOR-Schleife) um die Ergebnisse anzuzeigen.

2.6 Quadratzahlen mit Signalverlaufsgraph außerhalb der Schleife, Array-Werte und Graph bei Programmstart löschen

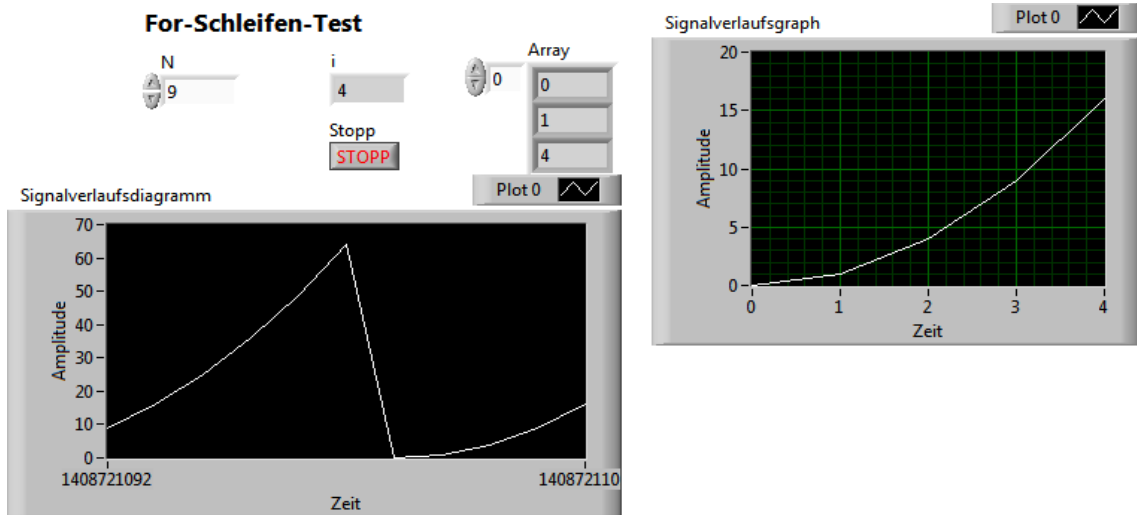


2.7 Quadratzahlen mit Signalverlaufsdigramm innerhalb der Schleife

Verwenden Sie ein Signalverlaufs-Diagramm (innerhalb der FOR-Schleife) und vergleichen Sie mit 2.5 (fügen Sie eine Zeitverzögerung in die Schleife ein).

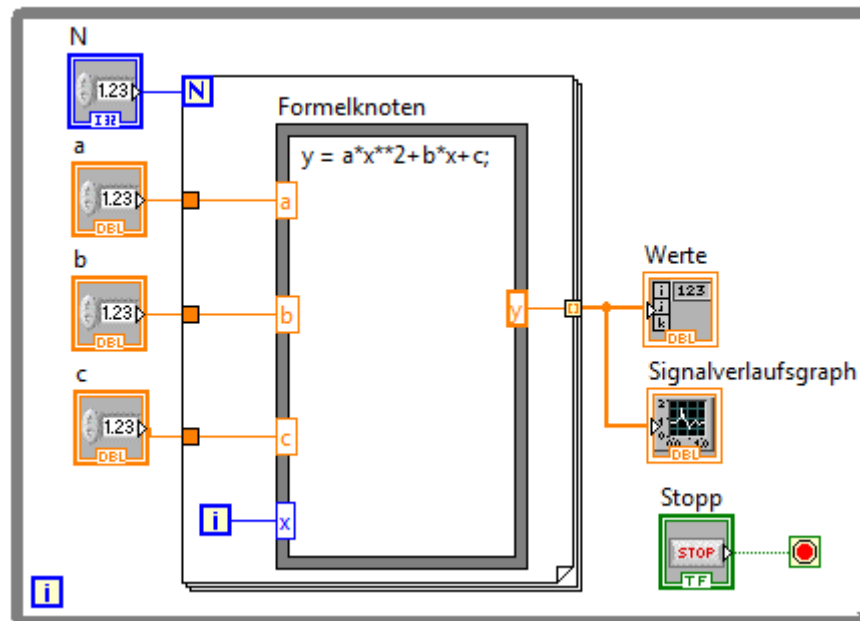
2.8 Quadratzahlen mit Signalverlaufsdigramm innerhalb der Schleife Diagramm zu Beginn löschen

Erstellen Sie einen Eigenschaftsknoten und setzen Sie die Eigenschaft „Historie“ auf Null.



Das Signalverlaufs-Diagramm zeichnet die aktuellen Werte auf und merkt sich diese.

3 Übung Polynomzeichner mit For-Schleife, Formelknoten, Array



3.1 Datentypen

Grüne Linien und Symbole:

Blaue Linien und Symbole:

Orange Linien und Symbole:

3.2 Schleifen

Unterschiede von while- und for-Schleifen:

N:

i:

3.3 Formelknoten

Handling-Anleitung

- Rechtsklick → Programmierung → Strukturen → Formelknoten
- An die Stelle, an der ein Eingang entstehen soll, auf den linken Rand des Formelknotens rechtsklicken → Eingang hinzufügen
- für alle Eingänge wiederholen, dann Variablennamen als Texte in die Eingänge schreiben (a, b, c, x)
- ebenso mit Ausgang verfahren (y)
- In den Formelknoten schreibt man seine C-Befehle als Text mit ; am Ende!
- + Plus * Mal ** Hoch
- Groß- und Kleinschreibung an Ein- und Ausgängen muss mit Gleichung übereinstimmen!
- An den Eingängen Bedienelemente erstellen und außerhalb der For-Schleife platzieren

3.4 Array

Handling-Anleitung

- Ausgang y auf den Rand der FOR-Schleife ziehen, dort entsteht ein „Tunnel“
- Rechtsklick auf den Tunnelausgang → Anzeigeelement erstellen → es entsteht ein Array.

- Im Frontpanel Array nach unten aufziehen → man sieht den Array-Inhalt

3.5 Schleifentunnel

- Ist der Schleifentunnel dunkel ausgefüllt, so liefert er einen Wert wie an den Eingängen
- Ist der Schleifentunnel hell (nicht ausgefüllt), so sammelt der Tunnel alle anfallenden Werte bei jedem Schleifendurchlauf und stellt diese nach Beendigung der Schleife als Array zur Verfügung.
- Auch der Schleifentunnel am Ausgang y kann nur einen (den letzten Wert) liefern, indem man mit dies mit Rechtsklick auf den Tunnel → Tunnelmodus einstellt.

3.6 Grafische Darstellung

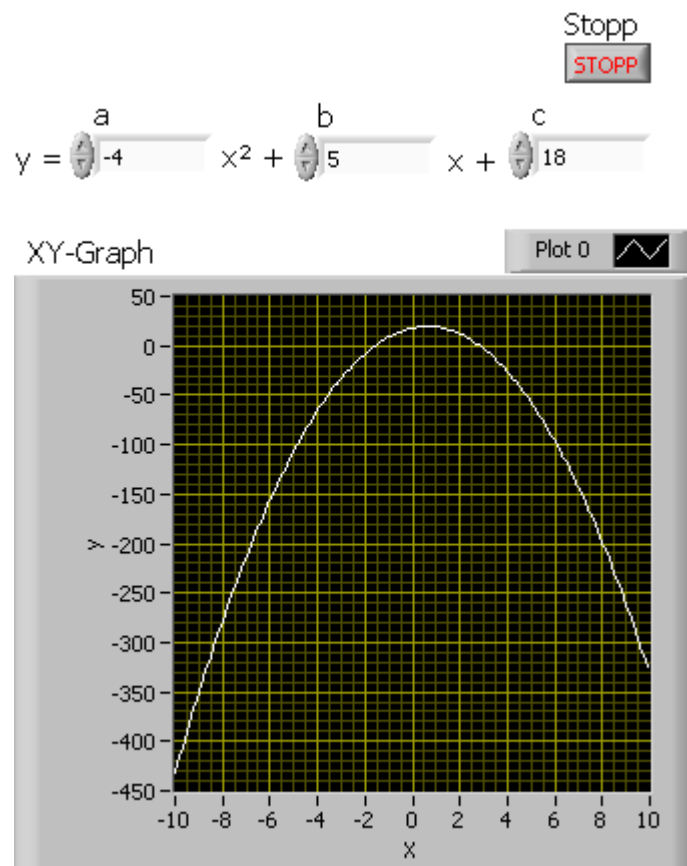
Schließen Sie parallel zum Array einen Siganlverlaufsgraphen und ein Signalverlaufsdigramm an. Starten Sie das Programm und erkennen Sie die Unterschiede zwischen beiden Darstellungen.

Signalverlaufsdigramm

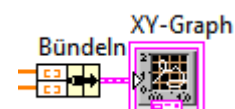
Siganlverlaufsgraph

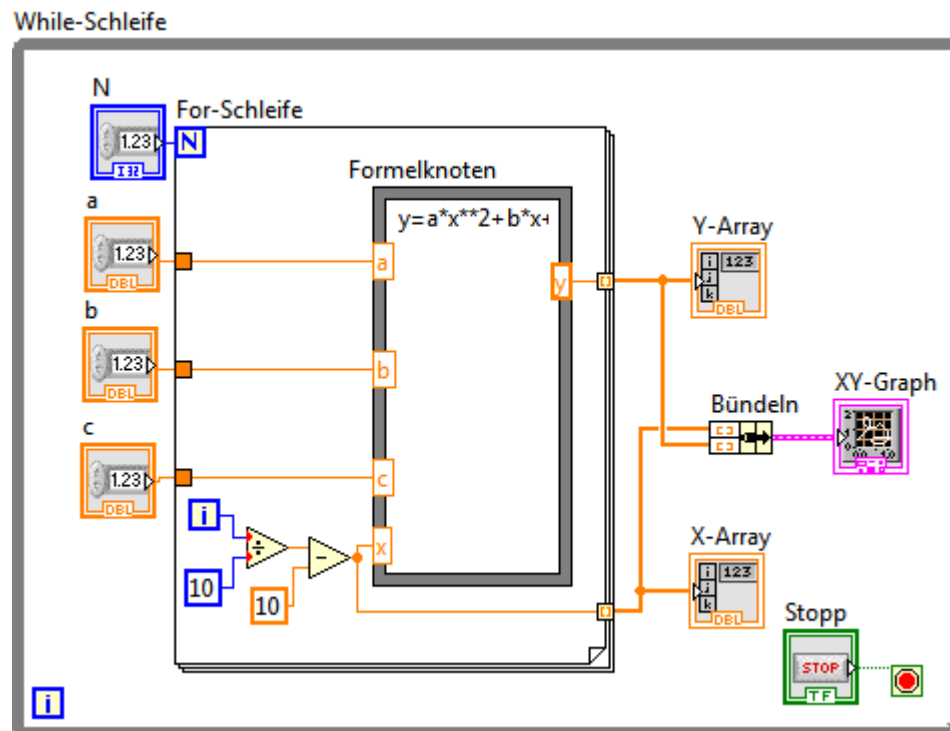
3.7 Polynomzeichner mit ungeradzahligen Werten von X

- Die Vorfaktoren der allgemeinen Funktionsgleichung können eingegeben werden.
- Die Zeichnung erfolgt immer im Bereich zwischen -10 und + 10 in 0,1-Schritten.



- Zur Anzeige verwenden Sie den XY-Graph mit der Komponente Bündeln zum Anschluss der x- und y-Werte

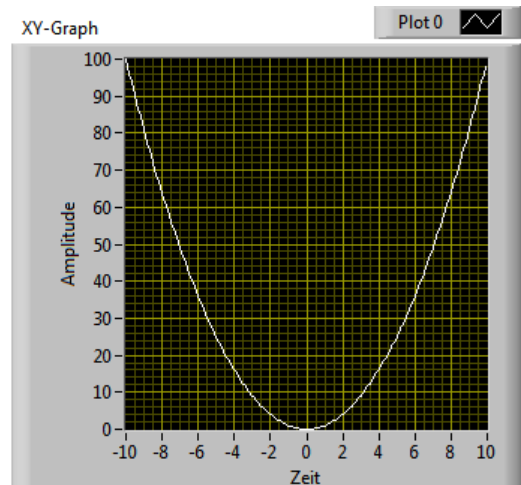




N
200
a
1
b
0
c
0

Stopp
STOPP

X-Array	Y-Array
-10	100
-9,9	98,01
-9,8	96,04
-9,7	94,09
-9,6	92,16
-9,5	90,25
-9,4	88,36

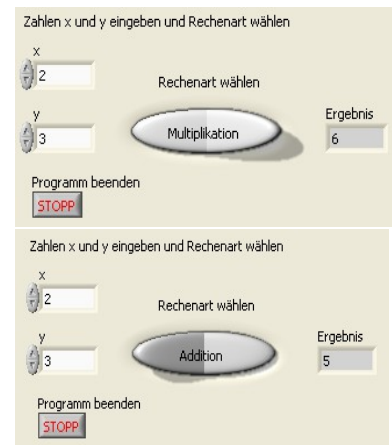


4 Case-Struktur

4.1 Case mit 2 Alternativen: True und False

Zwei Zahlen werden eingegeben.

Es wird die Summe (+) oder das Produkt (*) ausgegeben, je nachdem, in welcher Stellung der Schalter ist.



4.2 Case mit mehreren Alternativen und Textring

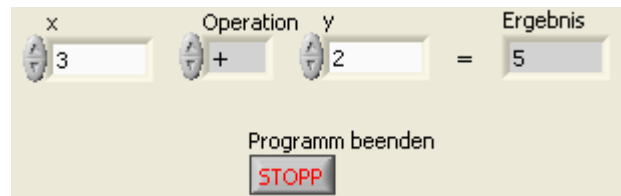
Über ein Bedienelement kann ausgewählt werden, ob man die Operation +, -, * oder / durchführen will.

Dazu wird ein Text-Bedienelement → Textring und eine CASE-Struktur gewählt.

Rechte Maus auf Textring
 → Eigenschaften → Wertebereich einstellen und bei → Objekte bearbeiten die 4 Fälle +, -, *, / eingeben.

Als Case gibt man 0, 1, 2, 3 ein.

Tipp: Textring und Case erst jetzt mit einer Leitung verbinden → Die 4 Werte 0,1,2,3 werden automatisch als 4 Cases übernommen.



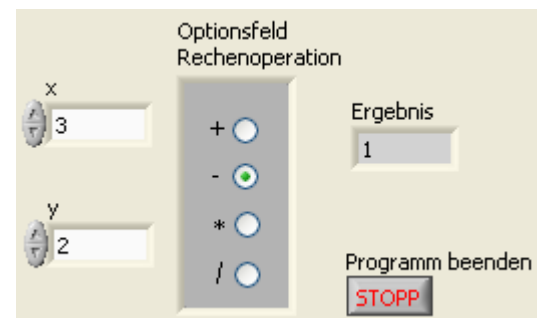
4.3 Case mit mehreren Alternativen und Optionsfeld

Wie oben, aber es soll über ein Bedienelement, bei dem man 1 aus 4 Buttons anklicken kann (Optionsfeld) die Rechenoperation ausgewählt werden.

Optionsfeld: Modern → Boolesch → Optionsfelder

Optionsfeld etwas größer ziehen.
 Eine Option markieren → STRG-C (Kopieren) → STRG-V → einfügen, STRG-V → einfügen
 → man hat 4 Optionen.

Texte von Optionsauswahl 1,2,3,4 ändern in +, -, *, /



5 Kennlinie automatisch aufnehmen

5.1 Spannung ausgeben und Spannung messen

- Zum Ausgeben und zum Messen benötigt man jeweils einen Kanal
- Beim Ausgeben muss man AO-Spannung angeben und die Komponente „schreiben“ verwenden.
-

Spannung ausgeben und Spannung messen

Vor dem Programmstart muss hier der Kanal und das Device gewählt werden!

Bitte Ausgang wählen
 I/O Dev1/ao0

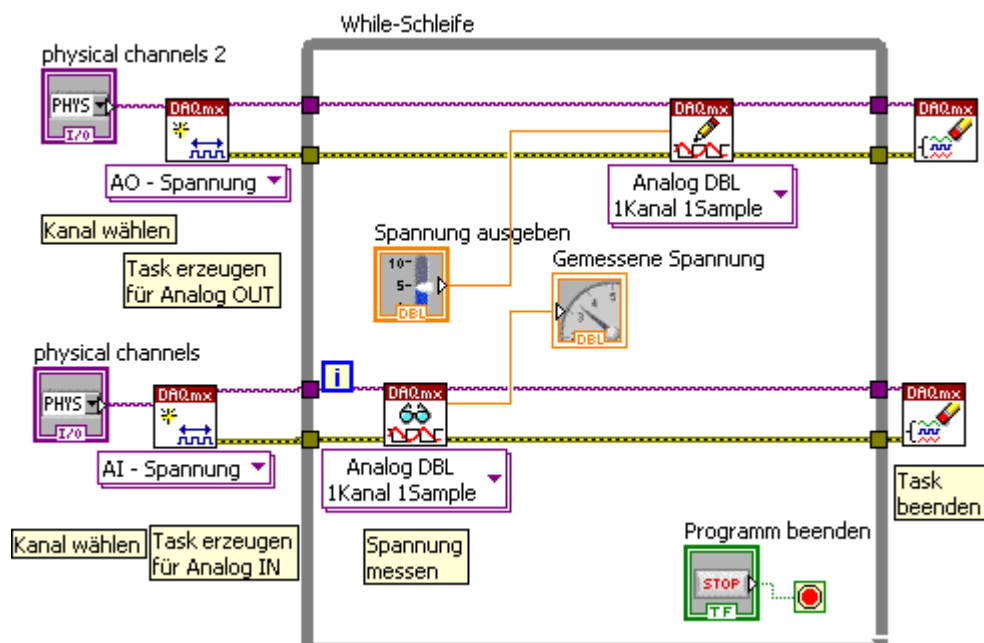
Bitte Eingang wählen
 I/O Dev1/ai0

Gemessen wird die Größe der Spannung am Eingang Ai0

Program beenden
 STOPP

Spannung ausgeben
 10
8
6
4
2
0

Gemessene Spannung
 10
9
8
7
6
5
4
3
2
1
0

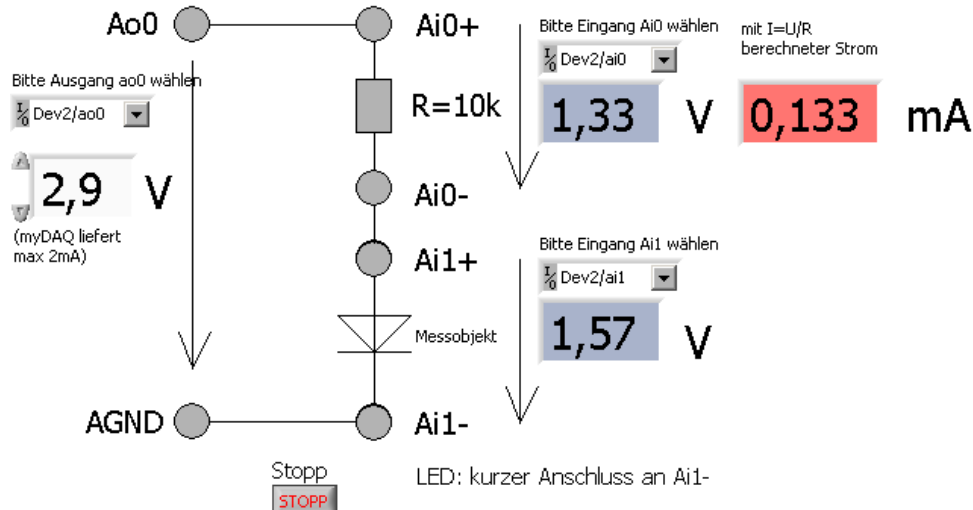


5.2 Manuelle Kennlinienaufnahme

Manuelle Kennlinienaufnahme

Vorgehen: Spannung am Eingang erhöhen und Strom und LED-Spannung beobachten und in einer Messtabelle notieren

Vor Programmstart
 Ein- und Ausgänge wählen



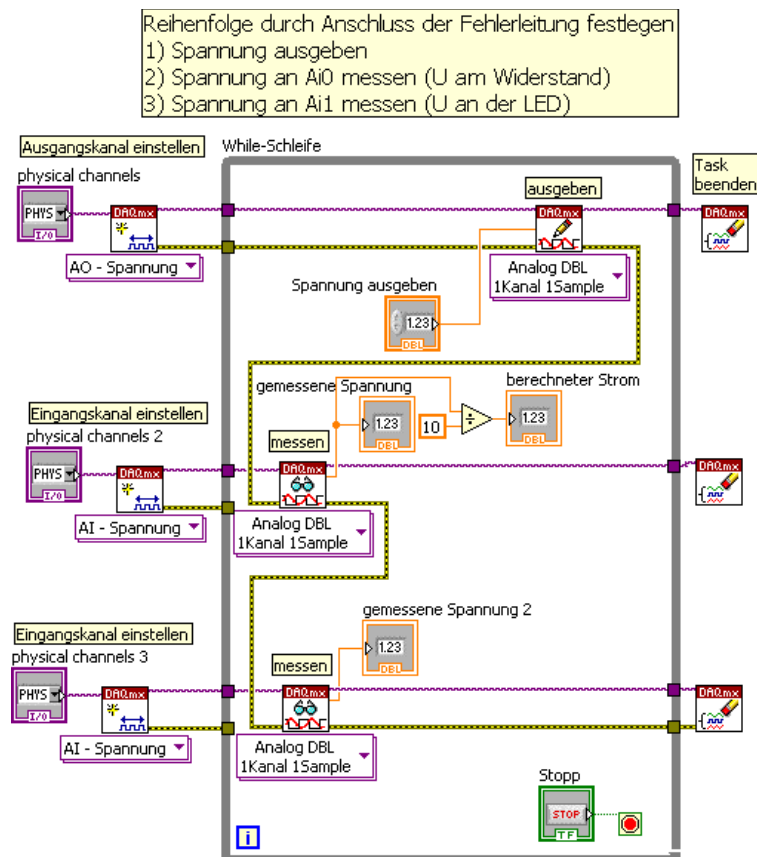
Mit dieser Schaltung kann die Kennlinie einer LED aufgenommen werden.

Vorgehen:

- Ausgangsspannung Ao0 ändern
- gemessene Spannung an der LED (Ai1) und
- den aus der Spannung Ai0 berechneten Strom beobachten

Die Reihenfolge der Messungen von Ai0 und Ai1 wird durch die Fehlerleitung festgelegt.

Die Ein-Ausgabebox kann 2 Spannungen nicht gleichzeitig messen!

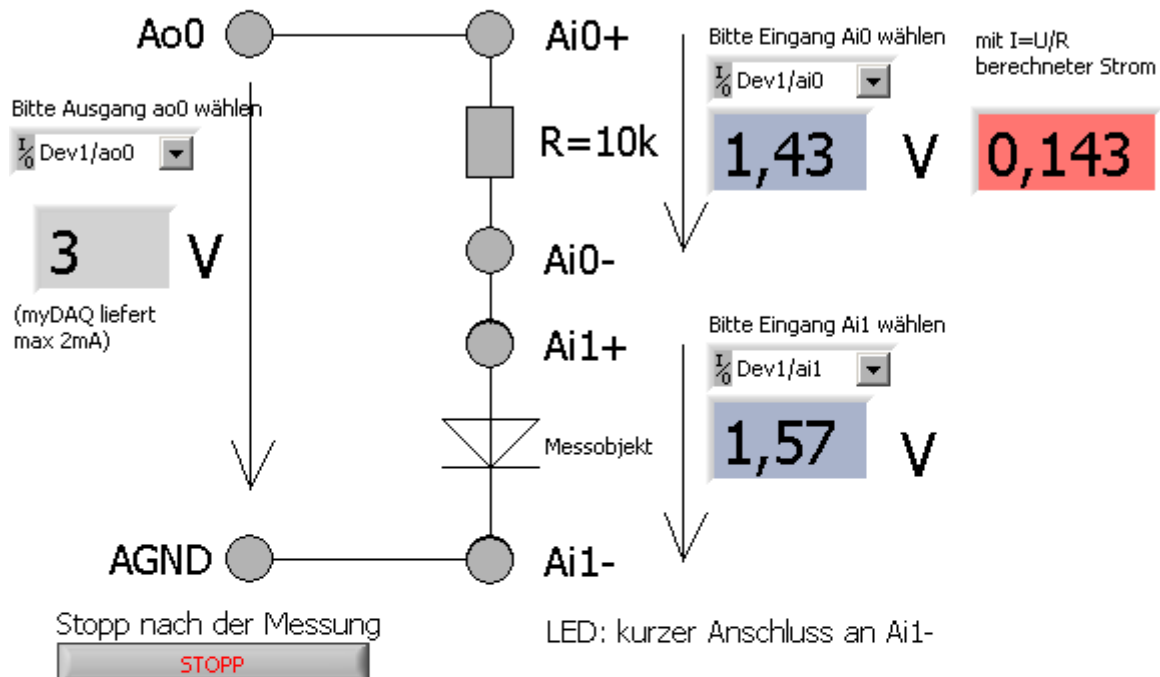


5.3 Automatische Kennlinienaufnahme mit 10 Messwerten

Automatische Kennlinienaufnahme mit 10 Messwerten

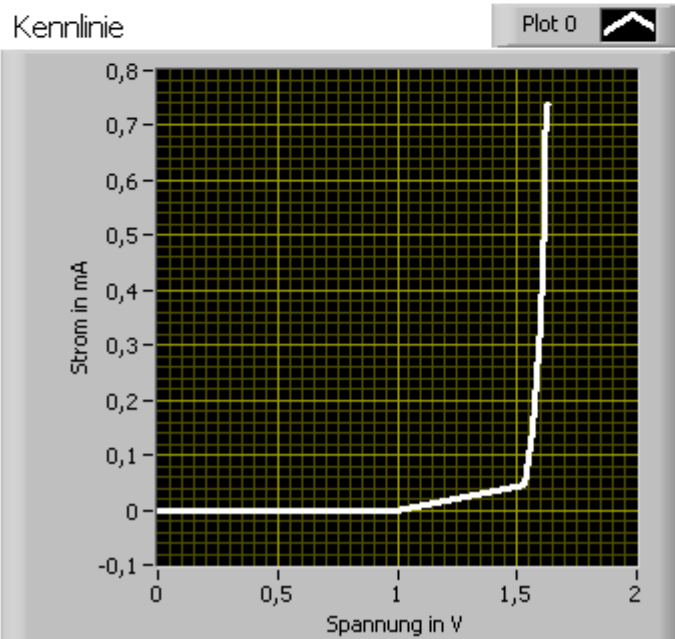
Spannung am Eingang erhöht sich automatisch in 1V-Schritten. Strom und LED-Spannung wird gemessen und nach der Messung in einer Messtabelle notiert

Vor Programmstart
 Ein- und Ausgänge wählen

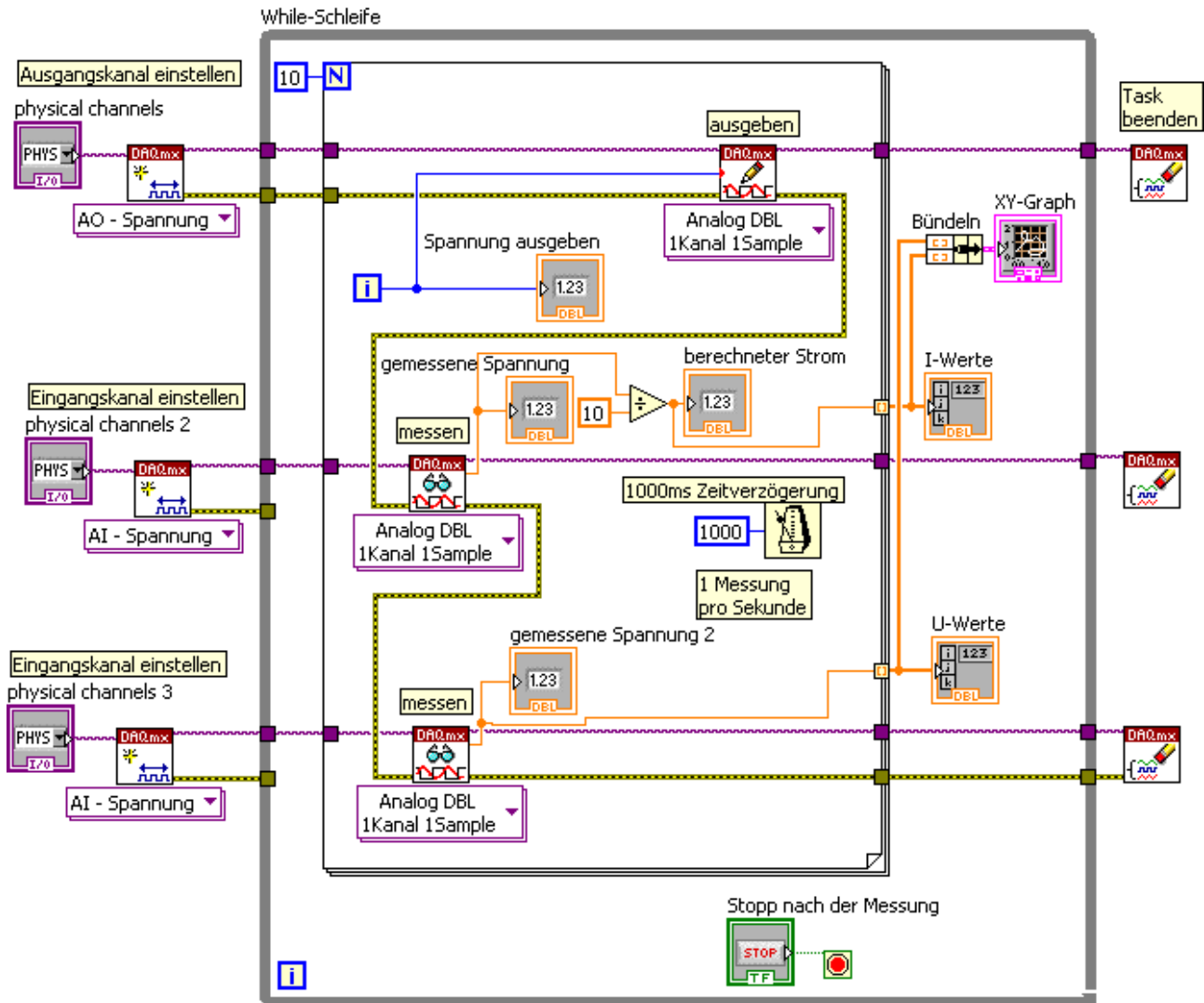


10 Messwerte sichtbar
 nach der Messung

U-Werte	I-Werte
-0,00	-0,000
1,00	-0,000
1,53	0,047
1,57	0,143
1,58	0,241
1,60	0,340
1,61	0,439
1,61	0,539
1,62	0,638
1,62	0,737



Reihenfolge durch Anschluss der Fehlerleitung festlegen
 1) Spannung ausgeben
 2) Spannung an Ai0 messen (U am Widerstand)
 3) Spannung an Ai1 messen (U an der LED)

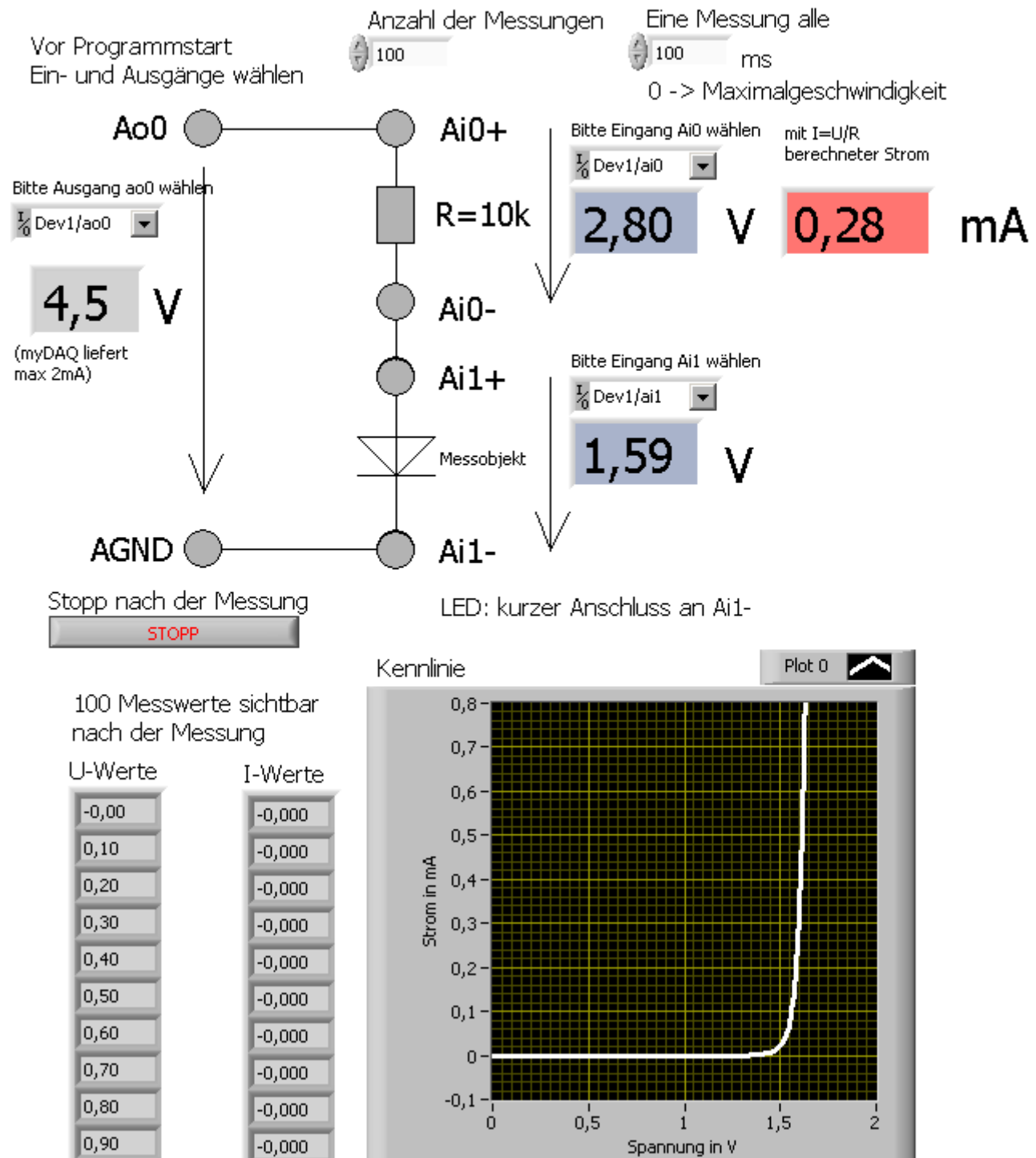


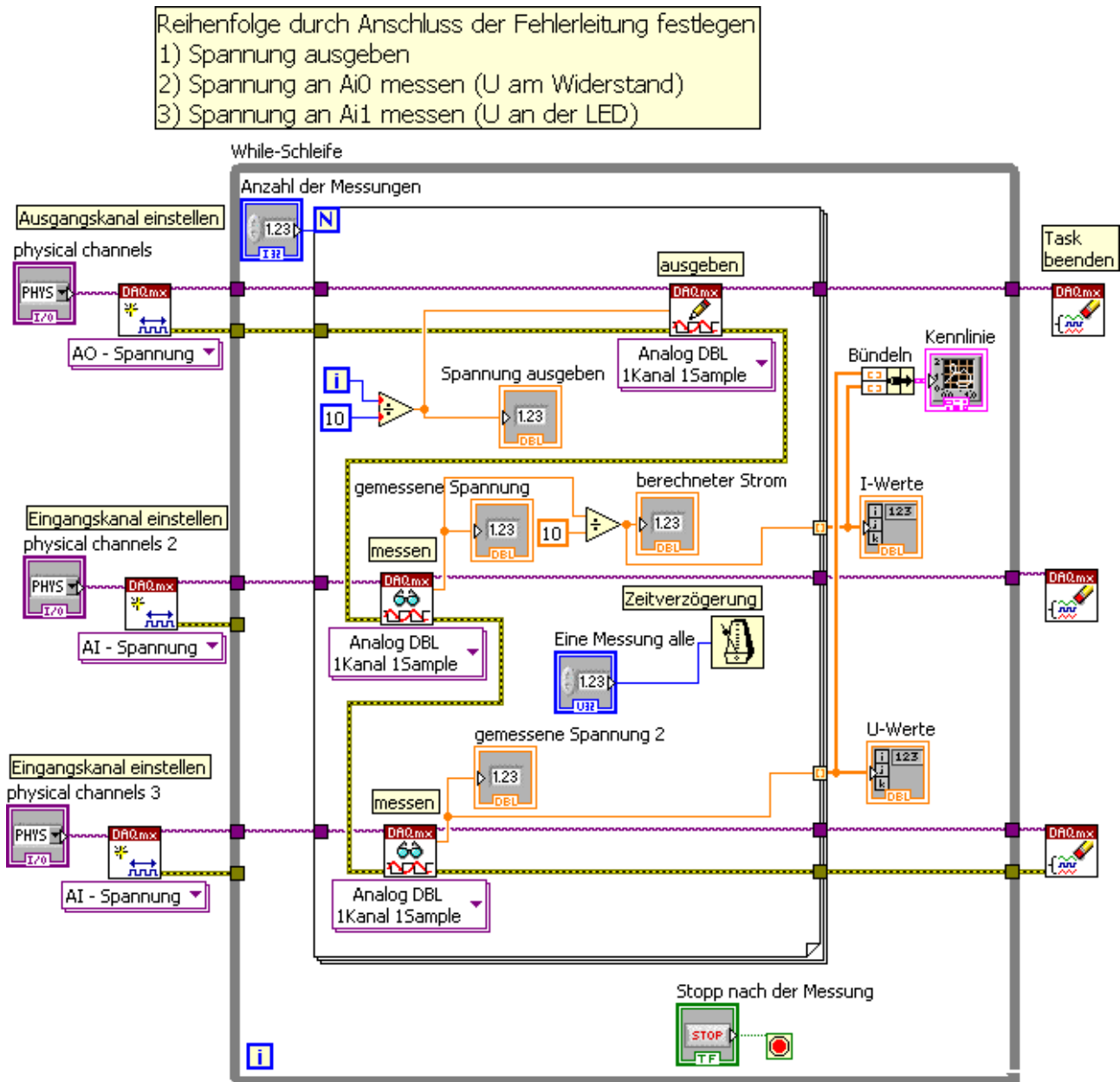
- Die For-Schleife erhöht die Spannung automatisch von 0 bis 9V. Zur Anzeige der Spannung wird das Bedienelement am Eingang in ein Anzeigeelement umgewandelt.
- Das Metronom erzeugt bei jedem Schleifendurchlauf eine Zeitverzögerung von 1000ms damit man den Messvorgang sieht.
- Die Messwerte von U und I werden außerhalb der Schleife in 2 Arrays gespeichert und anschließend im XY-Graph dargestellt.
 Den Graph findet man bei Rechtsklick → Modern → Graph → XY-Graph

5.4 Automatische Kennlinienaufnahme 100 Messwerte

Automatische Kennlinienaufnahme mit 100 Messwerten

Spannung am Eingang erhöht sich automatisch in 0,1V-Schritten. Strom und LED-Spannung wird gemessen und nach der Messung in einer Messtabelle notiert





- Anzahl der Messungen einstellbar → N
- Zeitverzögerung einstellbar in ms, wenn man 0 eingibt, wird das Programm mit Maximalgeschwindigkeit abgearbeitet.
- Bei jedem Schleifendurchlauf wird die Gesamtspannung um 0,1V erhöht. Dies ist möglich, weil i automatisch um 1 erhöht wird, aber i wird durch 10 geteilt.
- Bei N=100 werden 100 Messungen mit $U_{ges} = 0$ bis 9,9V in 0,1V-Schritten durchgeführt.

5.5 Kennlinienaufnahme mit Rückkopplungsknoten und „Array erstellen“

In den vorhergehenden Aufgaben wurden die x- und y-Werte von der For-Schleife gespeichert und zu einem Array zusammengefügt.

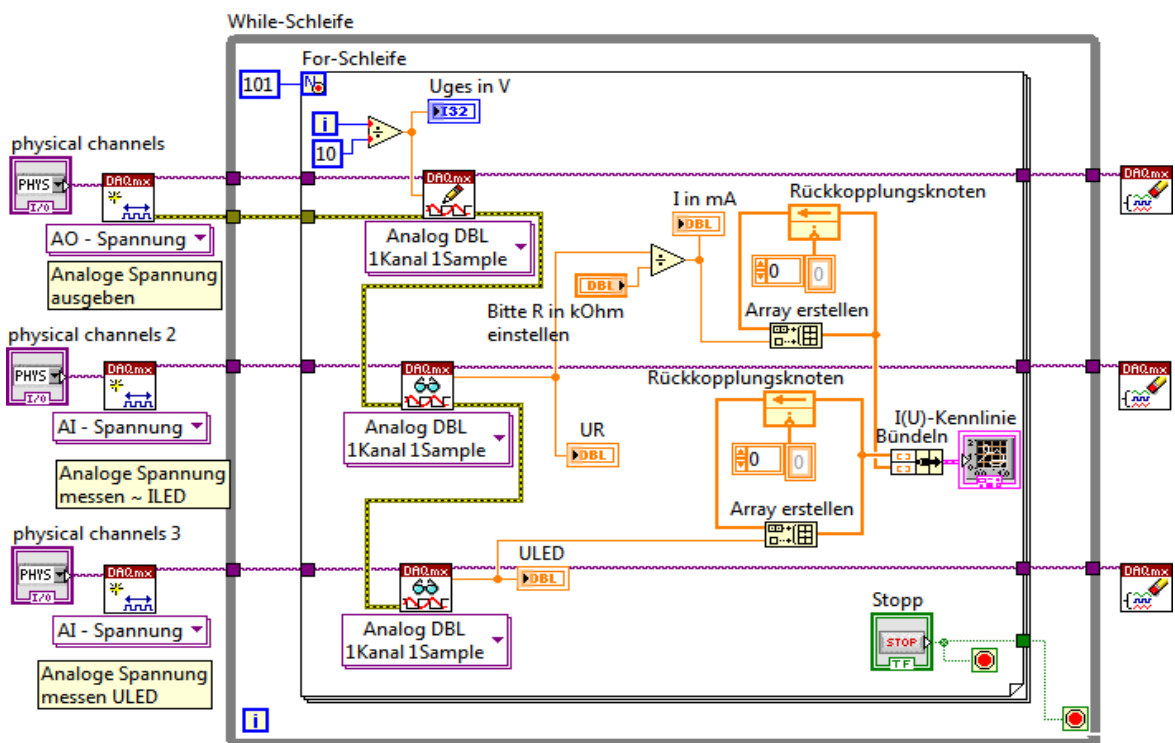
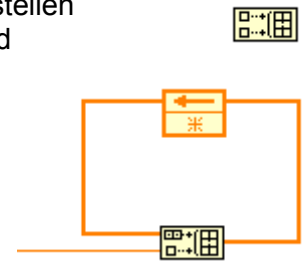
Diese hat den Nachteil, dass auf in der I(U)-Kennlinie die Werte erst dann angezeigt werden, wenn die gesamte Messreihe aufgenommen ist.

In diesem Beispiel erzeugen wir uns das Array selbst mit der Komponente „Array erstellen“:

Handling-Anleitung

Array erstellen

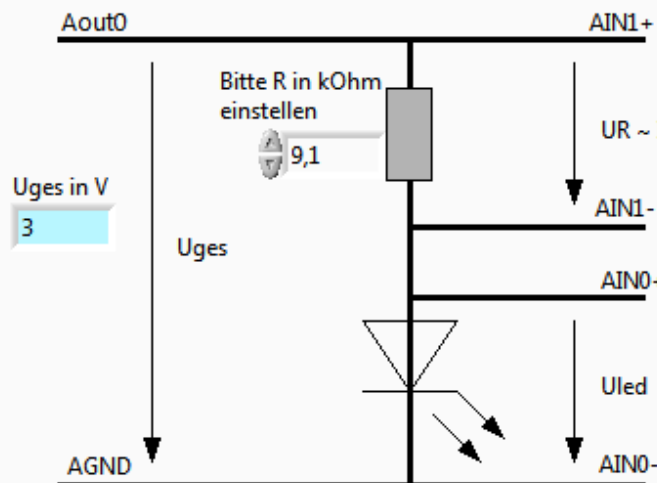
- Rechtsklick im Blockdiagramm → Programmierung → Array → Erstellen
- mit der Maus nach unten aufziehen bis 2 Kästchen entstanden sind
- Leitung mit I-Wert an den unteren Array-Eingang anschließen
- Vom Array-Ausgang eine Leitung zurück an den oberen Eingang führen. Es entsteht automatisch ein Rückkopplungsknoten, der das bisher entstandene Array im nächsten Schleifendurchlauf zurück auf den Eingang führt.
- Der neue Wert am unteren Array-Eingang wird dann zum bisherigen Array hinzugefügt.
- Ebenso mit dem U-Wert verfahren.



Kennlinienzeichner

Bitte AOut0 wählen

myDAQ2/ao0



Bitte AIN1 wählen

myDAQ2/ai1

UR in V 0,23 I in mA 0,02477

Bitte AIN0 wählen

myDAQ2/ai0

ULED in V 2,47

Array ILED

0	-8,69955
	-5,18325
	-0,00012
	-0,00015
	-5,18325
	-0,00012
	-0,00012

Array ULED

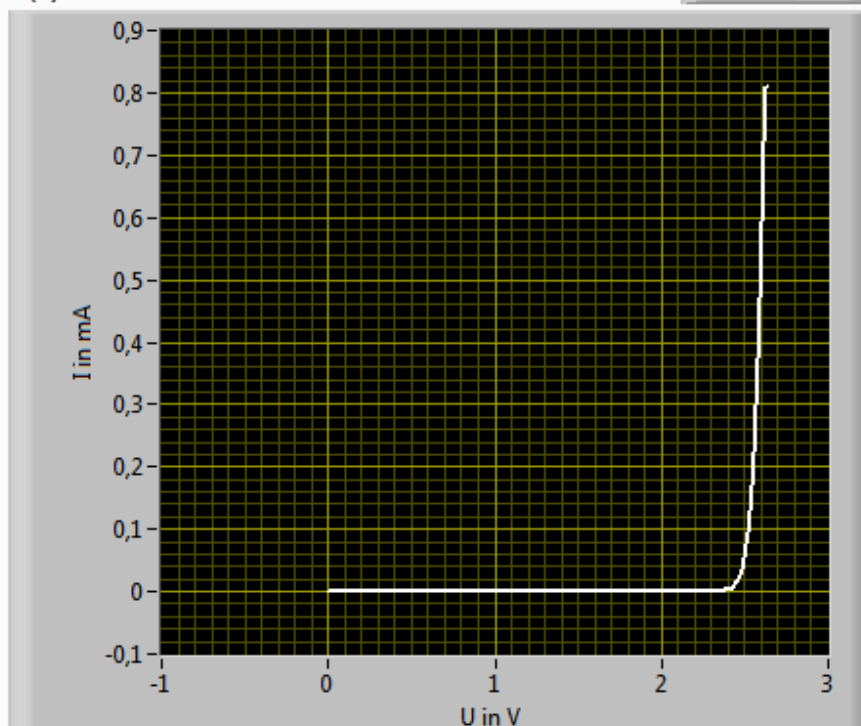
0	-0,00111
	0,09936
	0,19887
	0,29935
	0,39822
	0,49838
	0,59821

Stopp

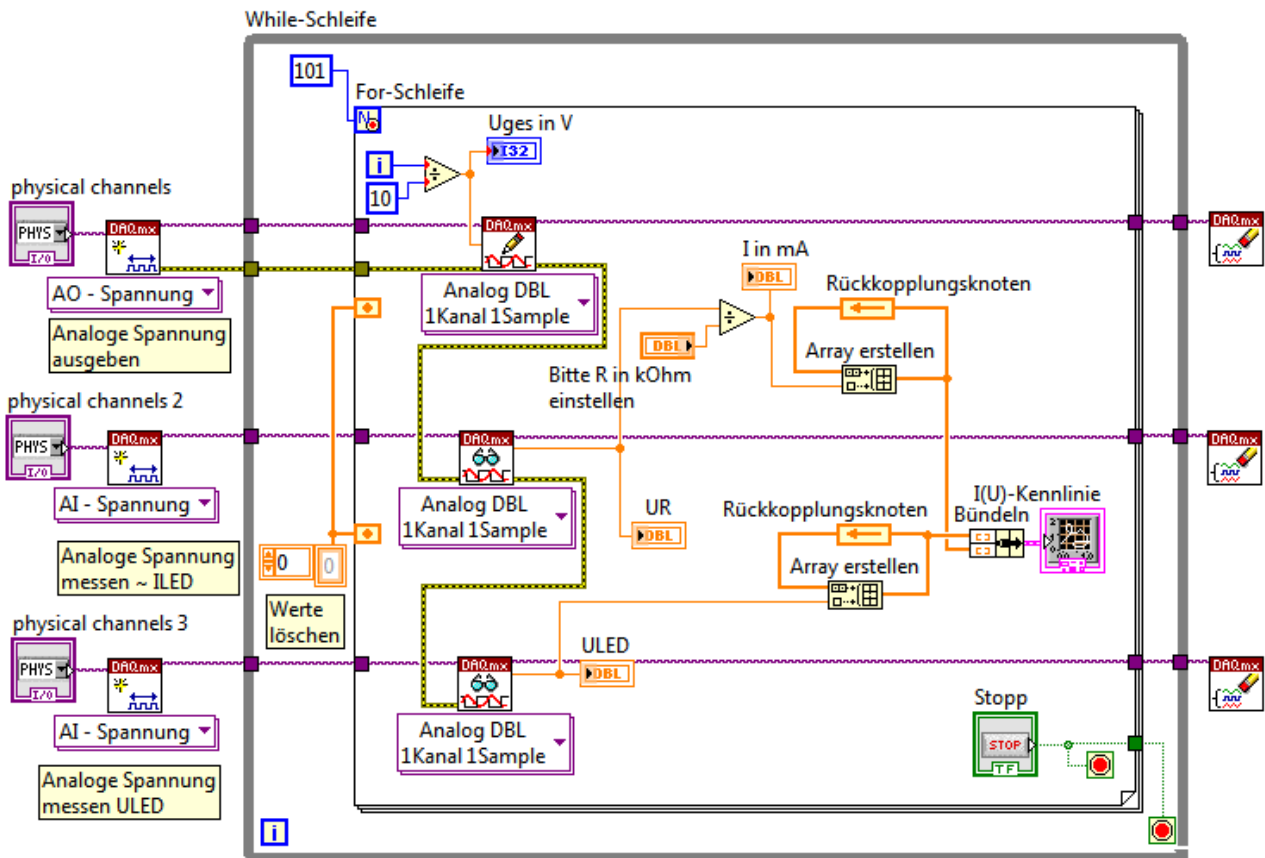
STOPP

I(U)-Kennlinie

Plot 0



5.6 Diagramm jedes Mal löschen



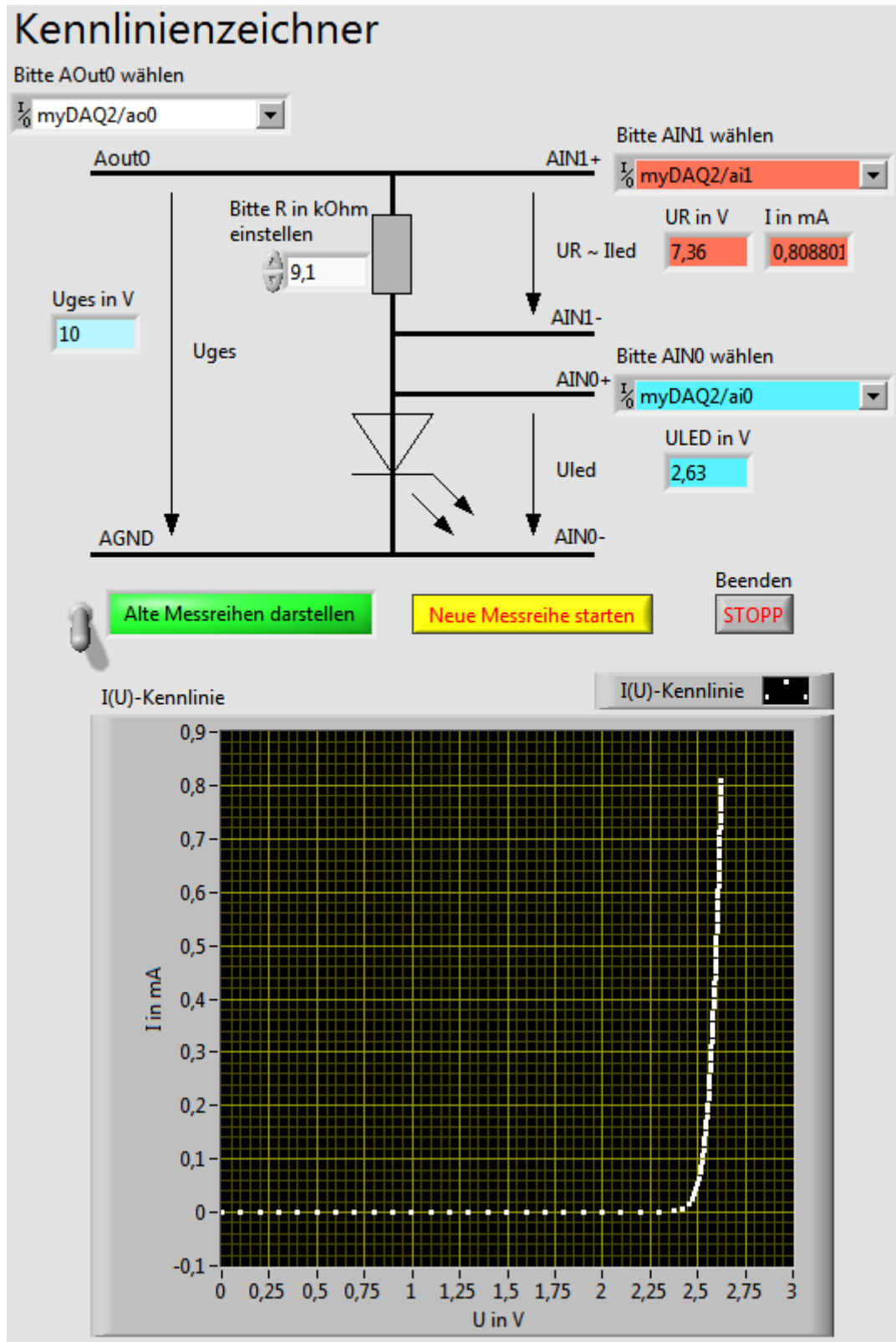
Die Array-Werte speichert der Rückkopplungsknoten. Wenn man das Diagramm bei einer neuen Messung neu entstehen soll, muss man die Werte im Rückkopplungsknoten löschen. Dies geschieht mit dem Initialisierungsanschluss unten.

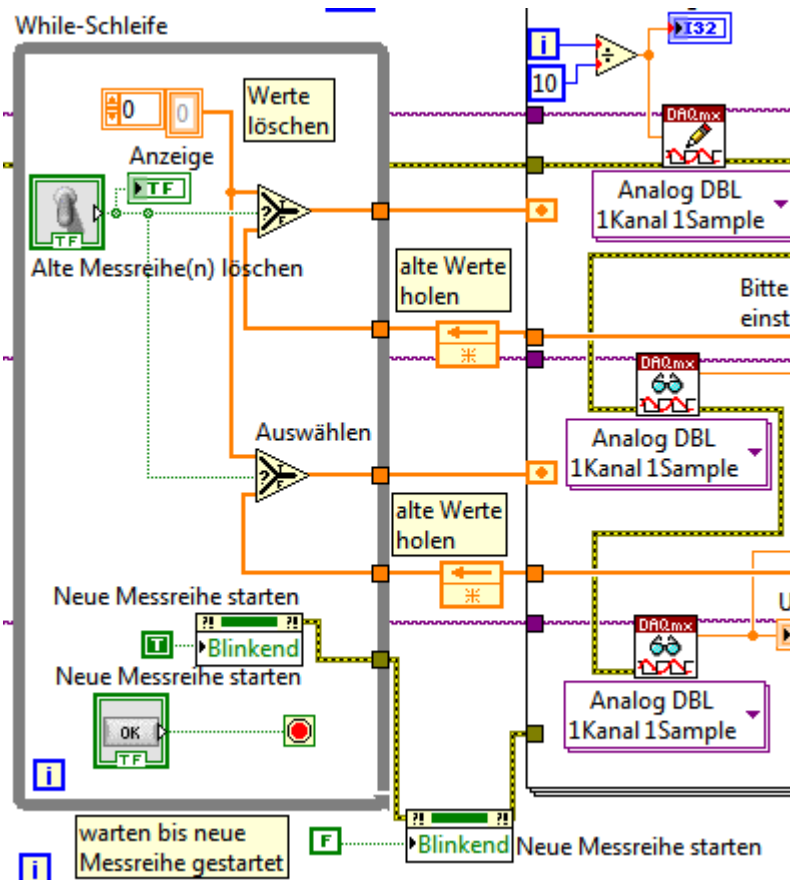
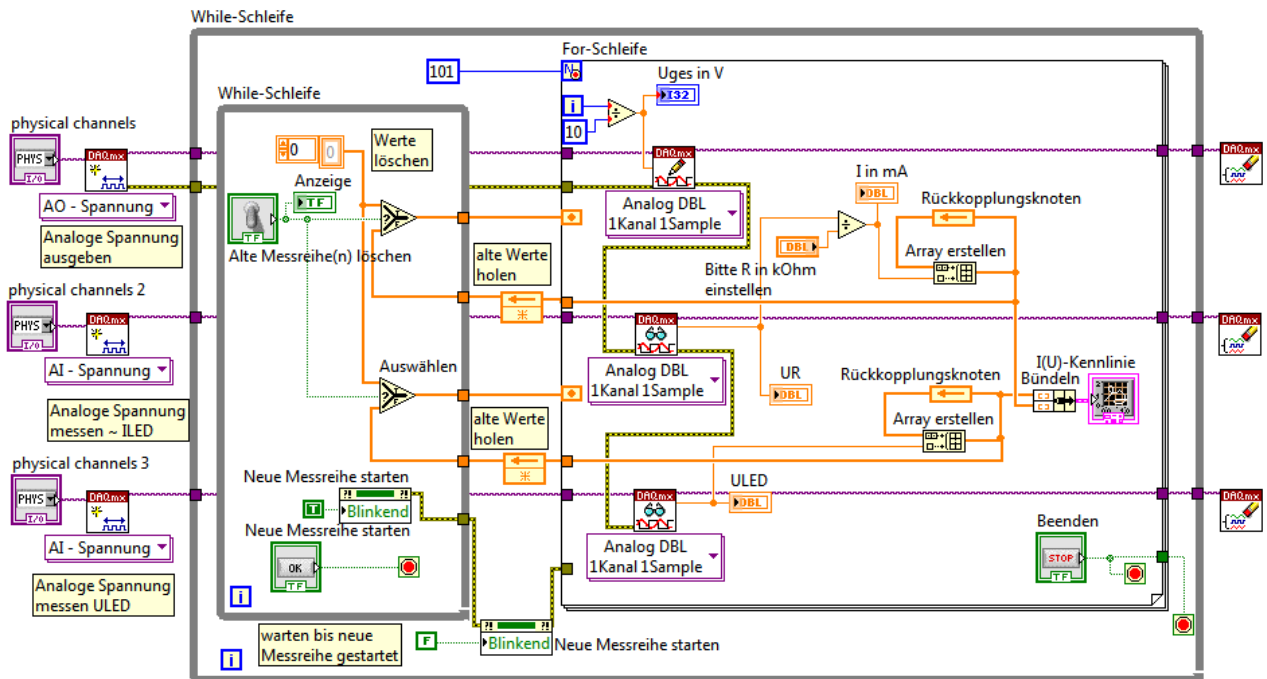
Handling-Anleitung

Rückkopplungsknoten initialisieren

- Rechtsklick auf den Initialisierungsanschluss → Erstellen → Konstante
- Initialisierung eine Schleife nach hinten verschieben
 → Die Initialisierung wandert an den linken Schleifenrand.
- Dort die erzeugte Konstante anschließen

5.7 Diagramm löschen wählbar / warten auf neue Messwertaufnahme





6 Widerstand und Temperatur messen

6.1 Einfache Widerstandsmessung

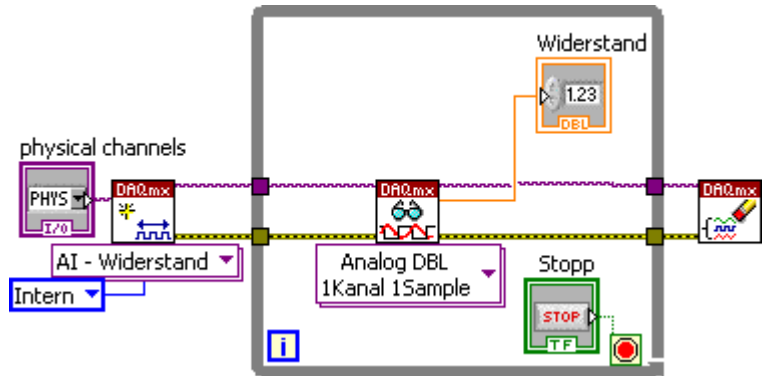
Widerstandsmessung

Bitte Eingang dmm wählen

$\frac{1}{10}$ Dev2/dmm

Stopp
 STOPP

Widerstand
 812,00 Ohm



In der myDAQ-Box ist auch ein Digitalmultimeter enthalten. Dies sind die farbigen 4mm-Buchsen an der unteren Seite der Box. Zwischen den Eingängen Hi (V, Ohm) und COM kann man den Widerstand messen. Dazu gibt die Box einen konstanten, bekannten Strom aus und misst die am Widerstand abfallende Spannung. Mit $R = U/I$ berechnet die Box den Wert des angeschlossenen Widerstands.

Bei der Komponente DAQmx-Kanal erzeugen muss man einstellen:

- AI-Widerstand
- Interne Erregerquelle (bedeutet: das Gerät liefert einen Strom zur Widerstandsmessung)

6.2 Widerstandsmessung und Temperaturbestimmung mit PT1000

$R = R_0 \cdot (1 + \alpha \cdot t)$ wobei $\alpha = 3,85 \cdot 10^{-3} / ^\circ\text{C}$, R_0 Widerstand bei 0°C : 1000 Ohm

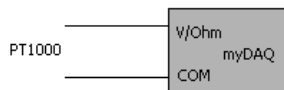
$R / R_0 = 1 + \alpha \cdot t$

$R / R_0 - 1 = \alpha \cdot t$

$t = (R / R_0 - 1) / \alpha$

$t = (R / 1000 - 1) / 0,00385 \text{ } ^\circ\text{C}$ wenn R der gemessene Widerstand ist.

Temperaturmessung



Bitte Eingang dmm wählen

$\frac{1}{10}$ Dev3/dmm

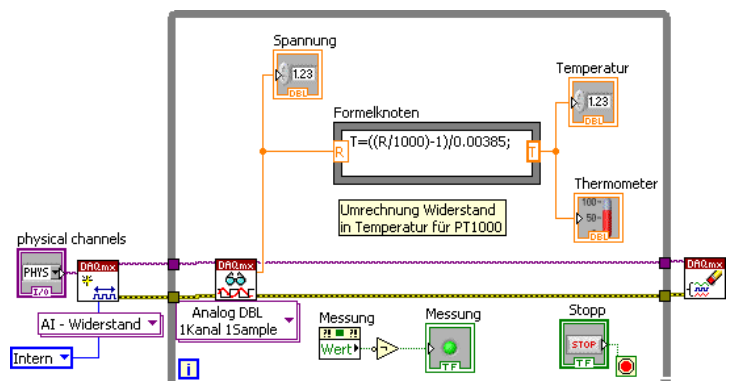
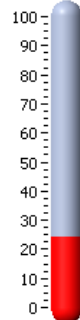
Stopp
 STOPP

Widerstand
 1093,00 Ohm

Messung

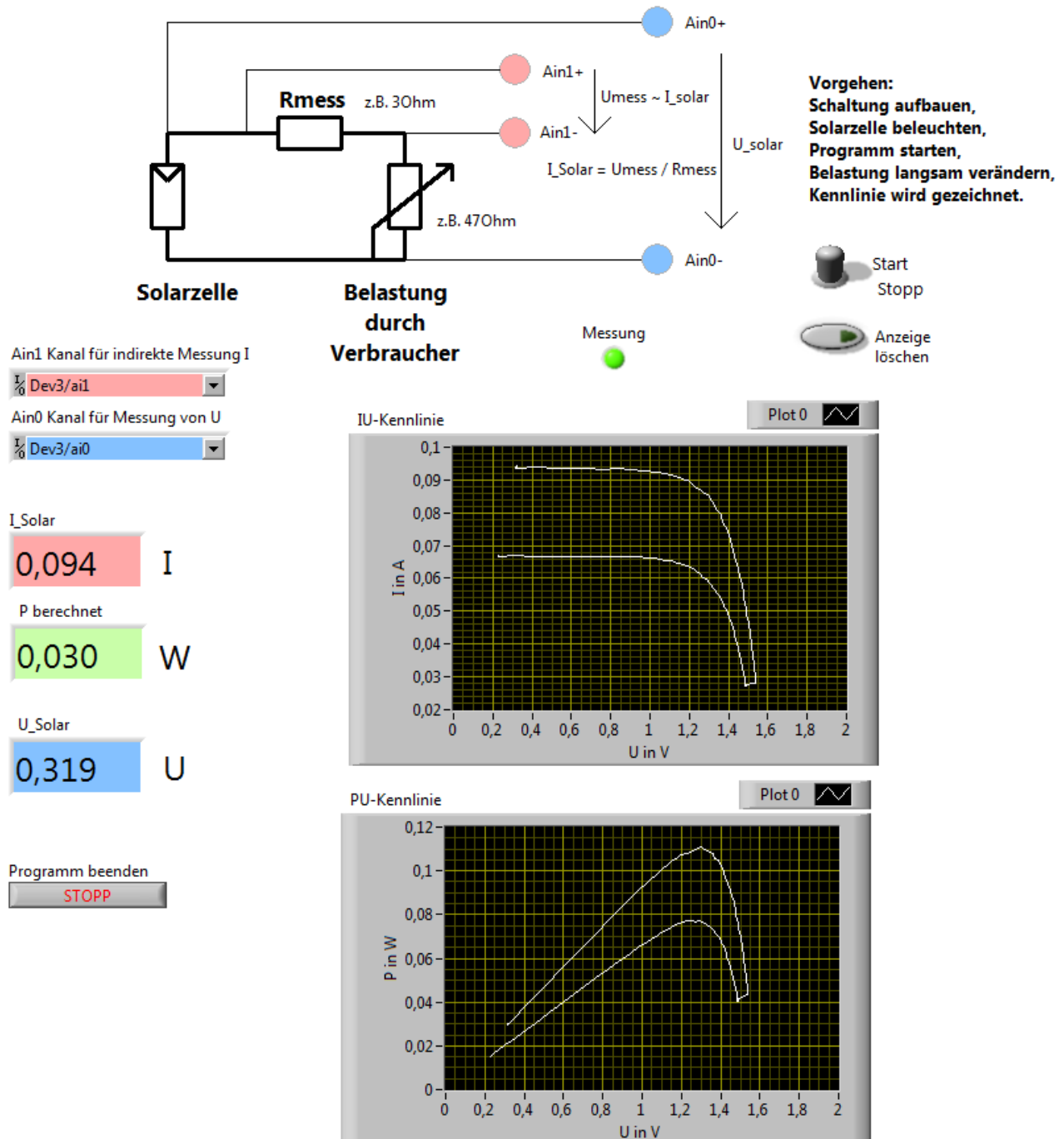
Temperatur
 24,16 $^\circ\text{C}$

Thermometer

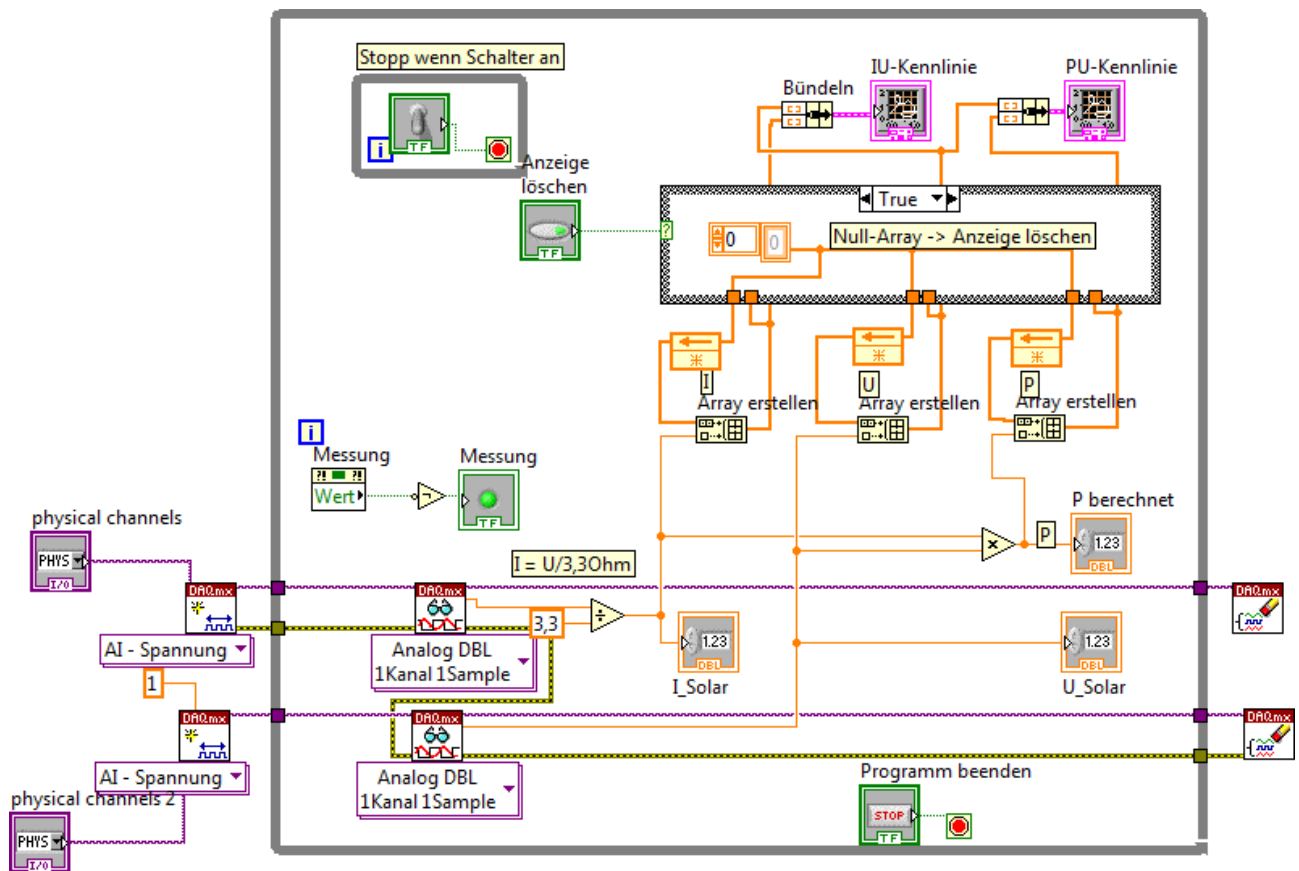


7 Kennlinienaufnahme Solarzellen

Kennlinienaufnahme einer Solarzelle



Erklärungen folgen



7.1 MPP-Suche mit 100 U-I-Messwerten eines Solarpanels

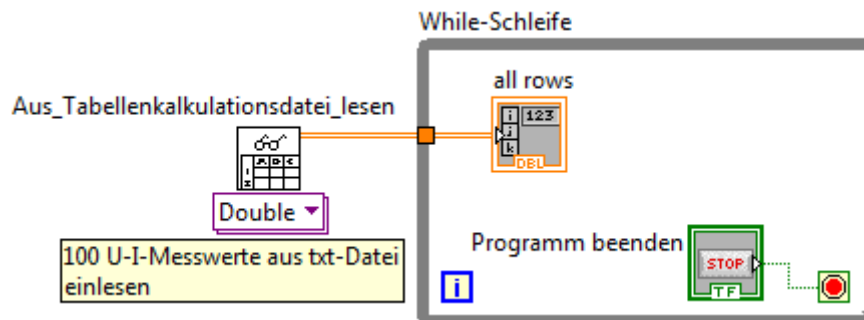
7.1.1 Allgemeines:

- Beschriften Sie alle Komponenten immer so, dass Sie später verstehen, welche Aufgabe die Komponenten haben. z.B. im Bild von 7.1.3: U-Array, I-Array usw.
- Speichern Sie jede Teilaufgabe extra ab. 7.1.2 erhält Nr 1, 7.1.3 erhält Nr 2 usw.
- Achten Sie auf saubere Leitungsführung und ordnen Sie die Komponenten auf dem Frontpanel und dem Blockdiagramm möglichst so an, wie bei den einzelnen Teilaufgaben angegeben.
- Sie benötigen die Datei 100UI_Messwerte_Kennlinie_Solarzelle.txt
- Fragen Sie, wenn Sie nicht weiterkommen

7.1.2 Daten aus Datei lesen und in einer Tabelle darstellen

100 Spannungs/Strom Messwertpaare befinden sich als Text-Tabelle in der Datei 100UI_Messwerte_Kennlinie_Solarzelle.txt.

Verwenden Sie die Komponente "aus Tabellenkalkulationsdatei lesen" und zeigen Sie die Werte wie unten dargestellt auf dem Bildschirm an.



all rows

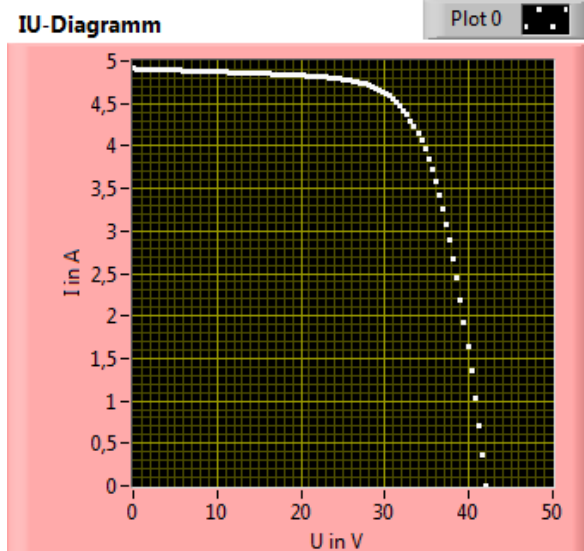
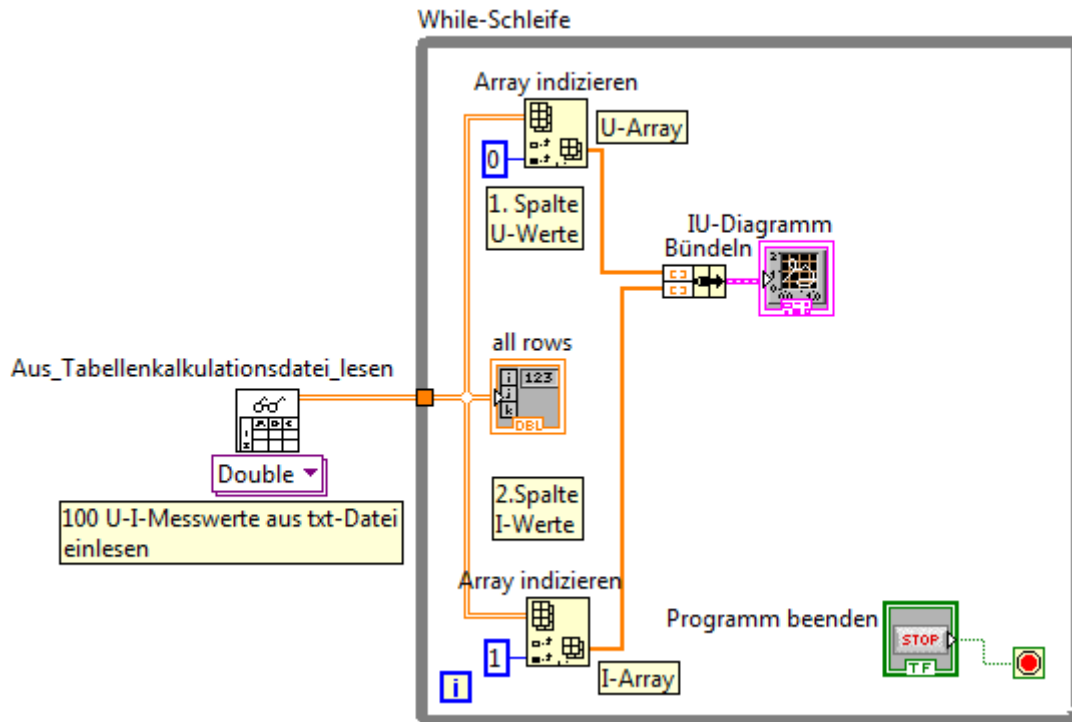
0	0	4,91	0
0	0,424082	4,90844	0
	0,848165	4,90687	0
	1,27225	4,9053	0
	1,69633	4,90374	0
	2,12041	4,90217	0
	2,54449	4,9006	0
	2,96858	4,89903	0
	3,39266	4,89746	0
	3,81674	4,8959	0
	4,24082	4,89433	0

Programm beenden
STOPP

7.1.3 Spalten aus der Tabelle extrahieren und I-U-Kennlinie erstellen

Mit den Komponenten Array indizieren können Sie einzelne Reihen aus der Tabelle herauslösen. Achten Sie darauf, wo Sie die Indizierungs-Konstante (0 oder 1) anschließen (Zeilen oder Spalten?)

Stellen Sie die Werte als IU-Kennlinie dar.



7.1.4 P-U-Kennlinie hinzufügen

Multiplizieren Sie das U-Array und das I-Array
 fügen Sie die PU Kennlinie neben die IU-Kennlinie hinzu.



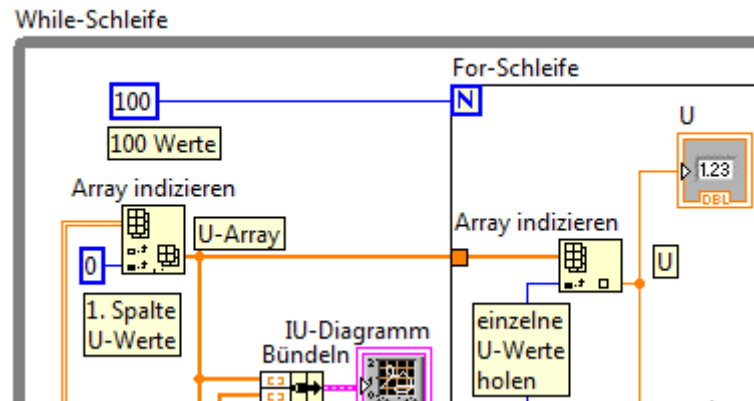
miteinander und

7.1.5 100 einzelne Werte von U und I extrahieren und anzeigen

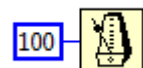
Löschen Sie die Anzeige der Wertetabelle (all rows) auf dem Bildschirm. (Platz schaffen!)

Mit weiteren Komponenten "Array indizieren" können Sie einzelne Werte des U-Array und I-Arrays extrahieren.

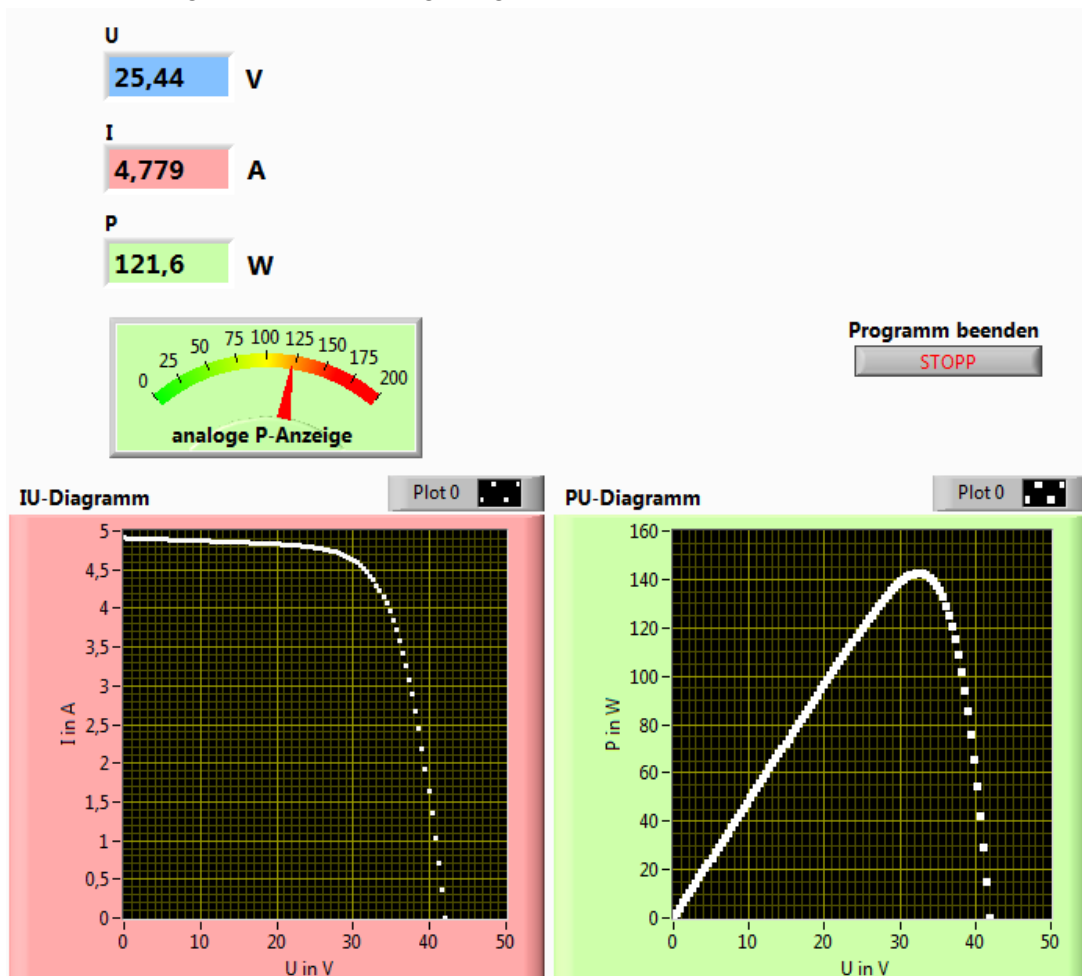
Verwenden Sie eine For-Schleife, um nacheinander die 100 Werte von U, I und P in 10 Sekunden auf dem Bildschirm auszugeben.



Anleitung: Indizierung deaktivieren bei den Schleifentunneln von U und I durch Rechtsklick auf die Schleifentunnel.

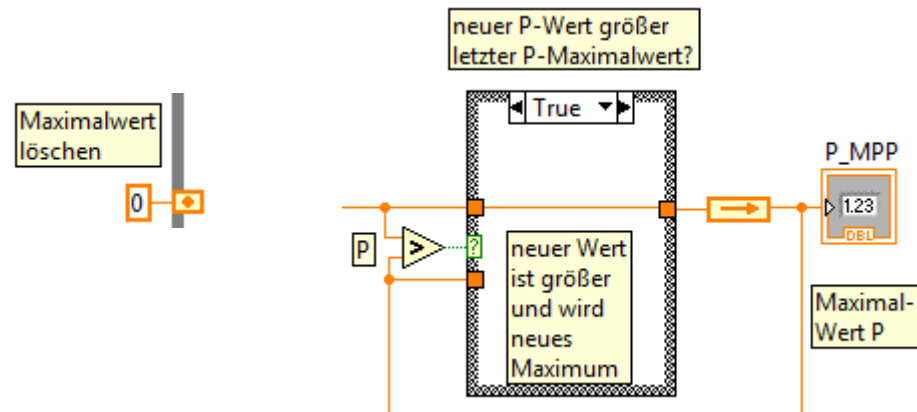


Die aktuellen Werte von U, I und P werden angezeigt, P soll zusätzlich mit einem analogen Instrument angezeigt werden.



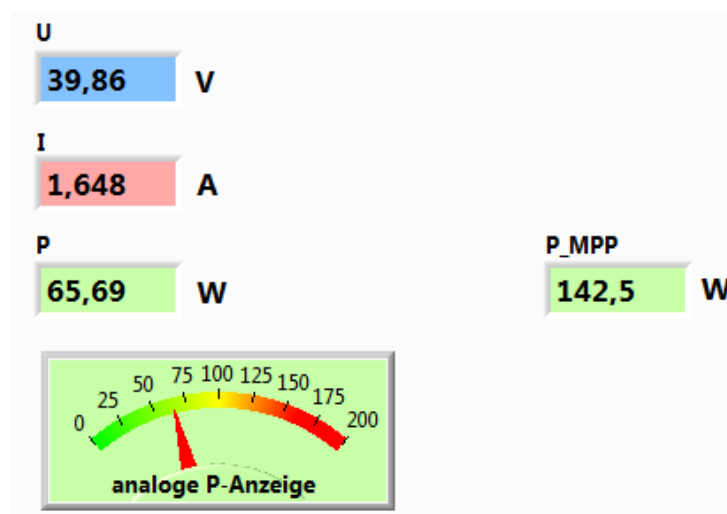
7.1.6 MPP der Leistung feststellen

In der For-Schleife soll der maximale Wert (MPP) der Leistung ermittelt werden. Dazu wird der gerade angezeigte alte Wert mit dem neuen Wert verglichen. Je nachdem, welcher der beiden Werte größer ist, wird der neue oder alte Wert als Maximalwert übernommen.



Erst wenn Ihr Programmteil funktioniert, testen Sie folgendes: Klicken Sie auf den Rückführungsknoten rechts und wählen Sie "Initialisierung eine Schleife nach hinten verschieben". Dann Rechtsklick auf Initialisierungsanschluss und Konstante erstellen.

beschreiben Sie die Auswirkungen, wenn man diese Initialisierung an der For-Schleife durchführt bzw. wenn man die beschriebene Aktion (Rechtsklick..) noch einmal macht und die Initialisierung an der while-Schleife durchführt.



7.1.7 Werte von I und U im MPP von P anzeigen

Zeigen Sie zusätzlich die Werte von U und I im MPP der Leistung an.
 (also nicht die absoluten Maximalwerte von U und I!!)

MPP-Suche			
U		U_MPP	
37,32	V	32,65	V
I		I_MPP	
3,083	A	4,364	A
P		P_MPP	
115	W	142,5	W

7.1.8 Programm beenden mit Stopp-Button

Wie Sie sicherlich schon bemerkt haben, lässt sich das Programm nicht mehr mit dem Stopp-Button beenden. Informieren Sie sich in der Hilfe zur For-Schleife, wie man diese abbrechen kann.

Ändern Sie dann Ihr Programm so ab, dass die Beendigung mit dem Stopp-Button jederzeit funktioniert.

7.1.9 Diagramme neu zeichnen

Ändern Sie Ihr Programm so ab, dass die Diagramme bei jedem Durchlauf der Messwerte neu gezeichnet wird.

7.2 Theorieteil

7.2.1 Datentypen

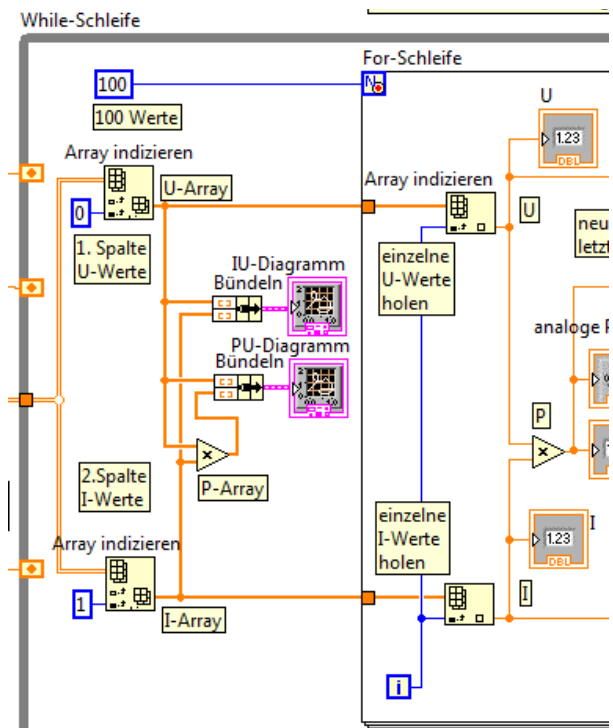
Markieren Sie 3 verschiedene Datentypen im Blockdiagramm und begründen Sie, von welchem Typ diese sein könnten.
 Was bedeutet die unterschiedliche Strichbreite?
 Was kann es bedeuten, wenn eine Verbindung gestrichelt dargestellt wird?

7.2.2 Schleifen

Erklären Sie die Unterschiede von while- und for-Schleifen. Welche Bedeutung haben die Angaben N und i?

7.2.3 Arrays

Erklären Sie den Begriff Array.
 Array indizieren ist das umgekehrte zu Array erstellen. Geben Sie an, wie man aus den einzelnen U-Werten in der For-Schleife wieder ein Array erstellen kann.



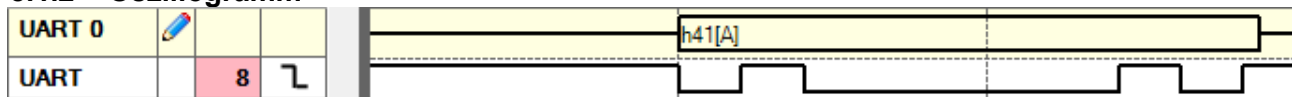
8 Datenaustausch PC – Controller über die COM-Schnittstelle

8.1 Datenübertragung COM-Schnittstelle Controller

8.1.1 Ascii „A“ dauernd senden, 1Byte empfangen und binär und als Ascii darstellen

```
/* Test serielle Kommunikation, Am PC HTerm starten mit 9600Baud und Connect
drücken */
#include <XMC1100-Lib.h> // Hilfsfunktionen fuer XMC1100
uint8_t sendebyte, empfbyte;
int main(void) // Hauptprogramm
{
    delay_ms(500); // Start LCD-controller
    rs232_init(); // Com-Schnittstelle initialisieren
    lcd_init(); // LCD initialisieren
    port_init(P0, OUP); // Port0 auf Ausgabe
    while(1) // Endlosschleife
    {
        sendebyte = 'A'; // Code vom Ascii-Zeichen A bestimmen
        //sendebyte = 0x41; // alternativ als Hexzahl
        rs232_put ( sendebyte ); // senden
        //rs232_print ( char *text )
        empfbyte = rs232_get(); // 1 Byte abholen
        if (empfbyte != 0)
        {
            port_write(P0, empfbyte); // und als Dualzahl an LEDs ausgeben
            lcd_setcursor(1,1); // links oben
            lcd_char(emfbyte); // anzeigen
        }
        delay_ms (100); // Zeitverzoeigerung
    } //while
} //main
```

8.1.2 Oszillogramm

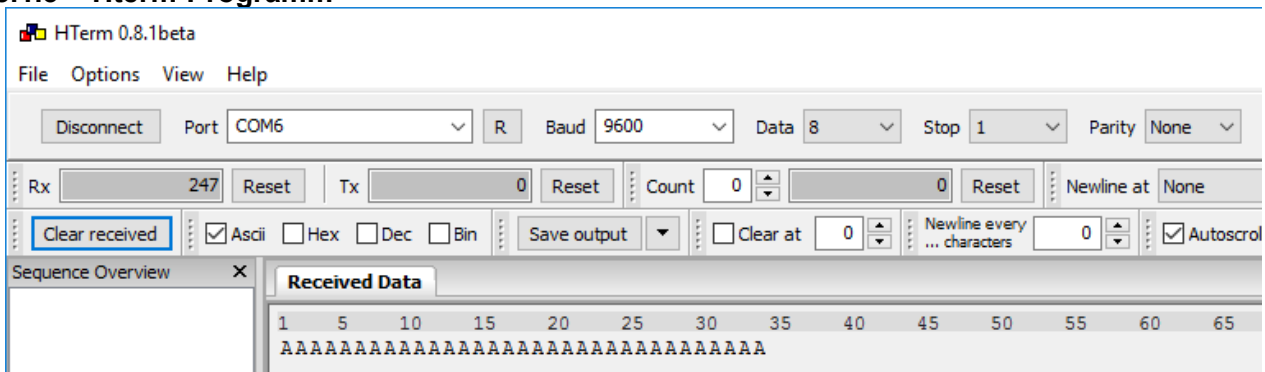


Grundzustand 1, dann Startbit 0, dann 10000010, dann Stoppbit=Grundzustand der Leitung

Links steht das zuerst empfangene, LSB → Binär 01000001 = hex 41 = Ascii „A“

Beim Controller wird dies über die USB-Schnittstelle und über P1.2 übertragen, dort kann man oszilloskopieren.

8.1.3 Hterm-Programm



Empfangene Zeichen werden als Ascii-Zeichen dargestellt: AAAA...

Protokoll: 9600 Bit/s, 8 Datenbit, 1 Sopbit, kein Paritätsbit

USB-Verbindung der Entwicklungsumgebung Dave wird für die Datenübertragung verwendet um am PC als COM6 für die Com-Verbindung verwendet.

8.2 COM-Schnittstelle mit Umwandlung der Variablenformate

VISA-Ressourcenname
 Die Übertragungsparameter lauten:
 9600 Bit/s, 8 Datenbit, kein Paritätsbit, 1Stoppbit
 kein Handshake

**Schnittstelle gewählt,
Weiter**

1 Byte Senden

Eingabe als

Text

Zahl

Text oder Zahl eingeben,
nach <Eingabe> aktualisiert
sich das andere Eingabefeld

senden

Empfangen

Empfangener Text

anzeigen nach
empfangenen Bytes

erstes empfangenes Byte

in Zahlen dargestellt

0

dezimal

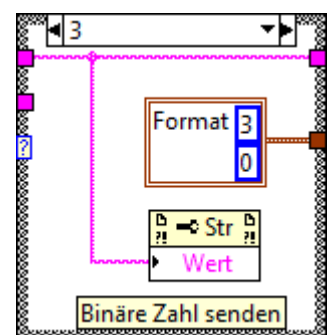
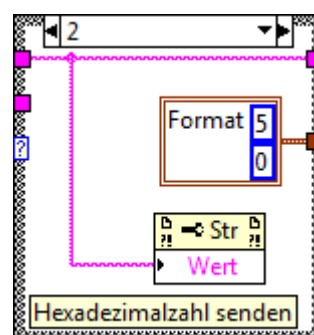
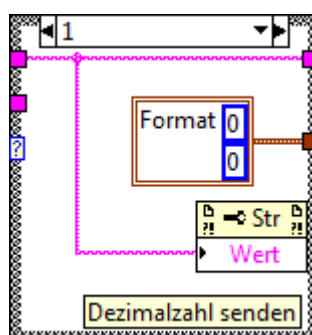
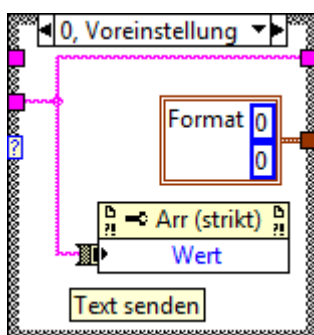
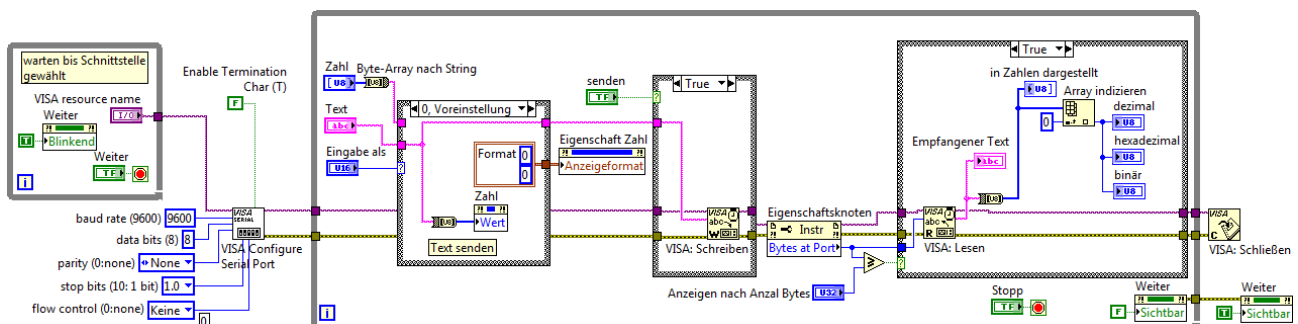
0

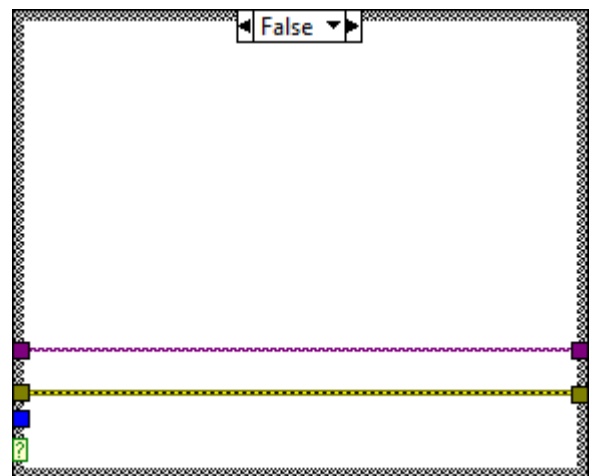
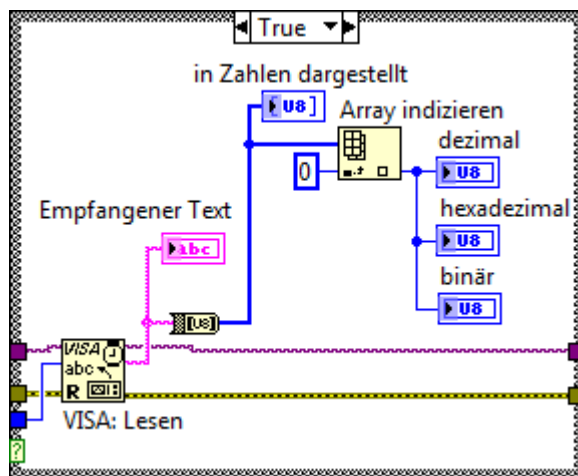
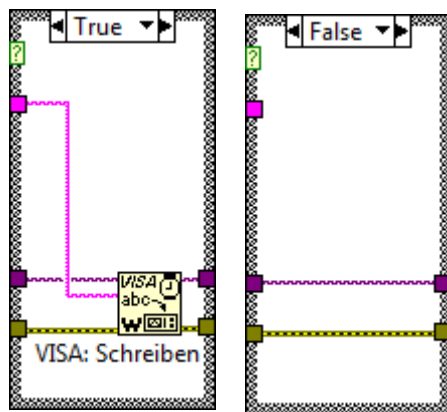
hexadezimal

0

binär

Programm beenden

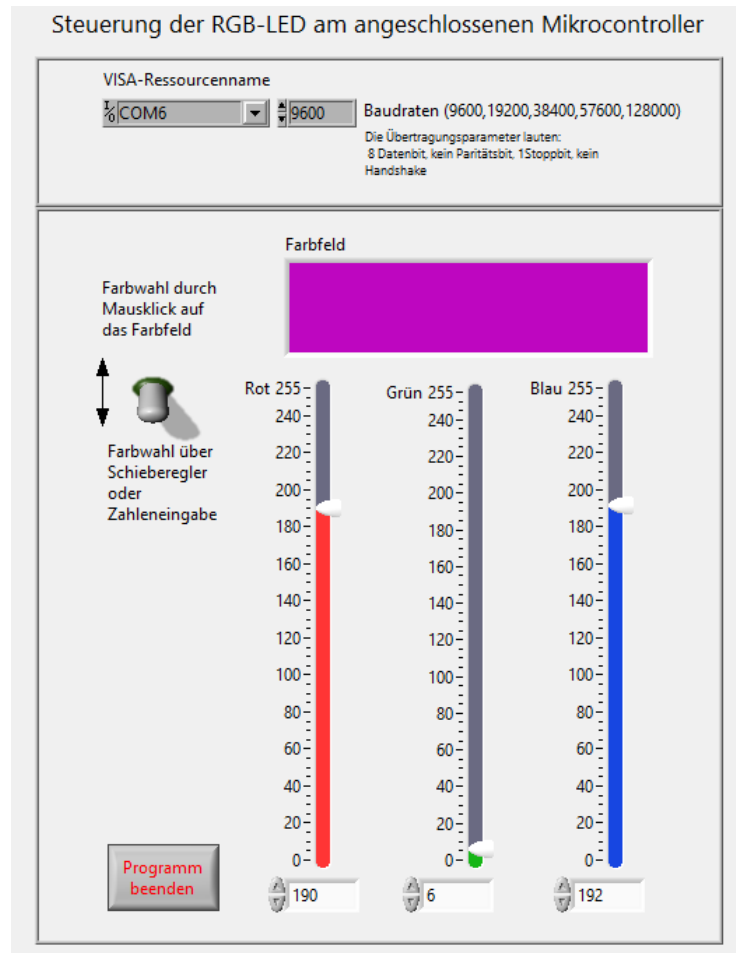




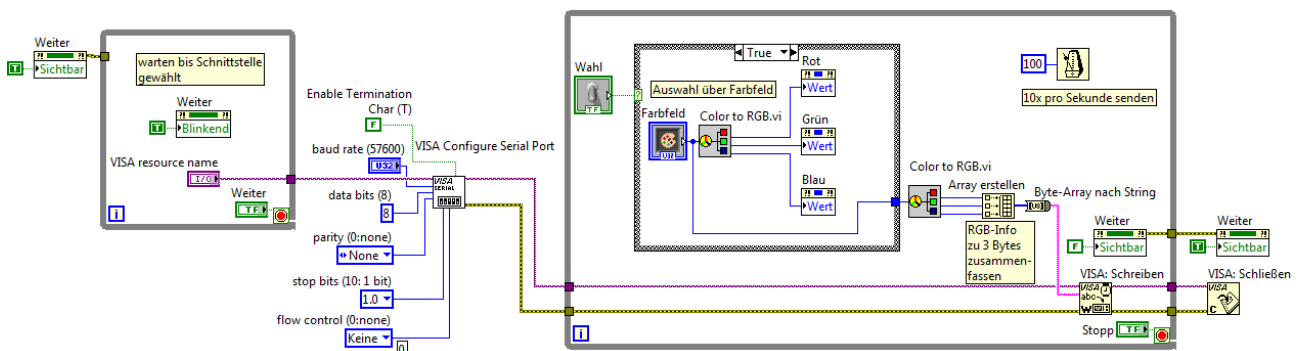
8.3 Fernsteuerung einer RGB-LED-Lichterkette am Controller über Labview am PC

8.3.1 Frontpanel

- C-Programm (unten) am Controller starten.
- Labview am PC starten
- Schnittstelle wählen.
- Farbe der RGB-LED mit Schiebereglern oder über das Farbfeld einstellen.
- Die Kommunikation erfolgt über die COM-Schnittstelle. 3 Bytes übertragen die Tastgrade der PWM-Signale, die der Controller erzeugt.



8.3.2 Blockdiagramm



8.3.3 Controller-Programm

```
/* Serielle Kommunikation: Labview-Programm steuert RGB-LED am Controller */
#include <XMC1100-Lib.h> // Hilfsfunktionen fuer XMC1100
uint8_t r_byte, g_byte, b_byte;
int main(void) // Hauptprogramm
{
    pwm1_init(); // PWM Kanal 0 initialisieren
    pwm2_init(); // PWM Kanal 1 initialisieren
    pwm3_init(); // PWM Kanal 1 initialisieren
    pwm1_start(); // Ausgabe starten
    pwm2_start(); // Ausgabe starten
    pwm3_start(); // Ausgabe starten
    pwm1_duty_cycle(0); // Tastgrade
    pwm2_duty_cycle(0);
    pwm3_duty_cycle(0);
    rs232_init();
    while(1U) // Endlosschleife
    {
        if (rs232_char_received() != 0) {
            r_byte = rs232_wait_get(); // Tastgrad für Rot vom PC holen
            g_byte = rs232_wait_get(); // Tastgrad für Gruen vom PC holen
            b_byte = rs232_wait_get(); // Tastgrad für Blau vom PC holen
        } // if
        pwm1_duty_cycle(~r_byte); // Tastgrade
        pwm2_duty_cycle(~g_byte); // Invertierung bei lowaktiver RGB-LED
        pwm3_duty_cycle(~b_byte);
    } //while
} //main
```