

Otto Bubbers

Gottlieb Daimler Schule 2 Sindelfingen

[bubbers@gds2.de](mailto:bubbers@gds2.de)

Stand: 18.10.2021

**Computergestützte Messwerterfassung 1 und 2**

Unterrichtsmaterial zu BPE 17 & 62

Bildungsplan 2021 – Informatik TG

Hinweise

Die Kapitel 1 bis 10 stellen ein mögliches Vorgehen im Unterricht dar. Sie bauen zum größten Teil inhaltlich aufeinander auf. Da die Hardware-Ausstattung der Schulen sehr unterschiedlich ist, müssen viele Beispiele auf die Möglichkeiten der Schulen angepasst werden.

Kapitel 3 und 4 bieten hervorragende Möglichkeiten für Anwendungen und Querverbindungen zum Fach Mathematik. Kapitel 10, 12 und13 greifen wesentliche Bildungsplaninhalte des Profilfachs Umwelttechnik auf. Es können jedoch bei jedem Kapitel Anknüpfungspunkte zum Profilfach gefunden werden.

Die Kapitel 11 bis 18 stellen eine Ideensammlung zur inhaltlichen Fortführung der Grundlagen dar. Es ist nicht möglich, alle Beispiele in der im Bildungsplan vorgesehenen Zeit zu bearbeiten.

Welche Kapitel der BPE 17 bzw. der BPE 63 zuzuordnen sind, muss individuell entschieden werden. Z. B. ist es möglich, Kapitel 16 anhand eines einfachen Beispiels bereits in der Eingangsklasse mit einfließen zu lassen. Ferner bietet der Bildungsplan die Möglichkeit durch die Zuordnung des VIP-Bereichs gezielt Vertiefungen zu schaffen.

Die Computergestützte Messwerterfassung 1 sollte zeitlich am Ende der Eingangsklasse unterrichtet werden, um Vorkenntnisse der Schülerinnen und Schüler mit aufnehmen zu können.

Wann die BPE 63 unterrichtet wird, hängt auch von den ausgesuchten Beispielen und den Querverbindungen zu anderen Themen der Informatik (z. B. Mikrocontrollertechnik) ab.

Für Fragen und Anregungen steht der Autor gerne unter [bubbers@gds2.de](mailto:bubbers@gds2.de) zur Verfügung.

Inhaltsverzeichnis

[1 Grundeinstellungen in Labview 5](#_Toc85991479)

[1.1 Neues Projekt 5](#_Toc85991480)

[1.2 Grundsätzliche Arbeitsweise des Programms 5](#_Toc85991481)

[1.3 Grundeinstellungen der Menüs 6](#_Toc85991482)

[2 Spannung messen und Grenzwertüberwachung 7](#_Toc85991483)

[2.1 Simulierten Sensorwert anzeigen und Grenzwerte überwachen 7](#_Toc85991484)

[2.2 Erweiterung: Schieberegler für Grenzwerte, Standardwerte einstellen 8](#_Toc85991485)

[2.3 Verlaufsdiagramm hinzufügen 9](#_Toc85991486)

[2.4 Hardware hinzufügen: Potentiometer als Füllstandssensor 10](#_Toc85991487)

[3 Abstandsmessung mit Ultraschallsensor 12](#_Toc85991488)

[3.1 Aufgabenstellung und Vorbereitungen 12](#_Toc85991489)

[3.2 Aufgaben 12](#_Toc85991490)

[3.3 Lösung 13](#_Toc85991491)

[4 Füllstandsmessung mit Ultraschallsensor 14](#_Toc85991492)

[5 Abstandsmessung mit optischem Sensor 14](#_Toc85991493)

[5.1 Spannungswert des optischen Sensors auslesen 14](#_Toc85991494)

[5.2 Zusammenhang zwischen Spannung des Sensors und Abstand ermitteln 15](#_Toc85991495)

[5.3 Formelknoten zur Berechnung des Abstands einfügen 15](#_Toc85991496)

[6 Widerstand und Temperatur messen mit PT1000 16](#_Toc85991497)

[6.1 Einfache Widerstandsmessung 16](#_Toc85991498)

[6.2 Widerstandsmessung und Temperaturbestimmung mit PT1000 16](#_Toc85991499)

[6.3 Erweiterungen: Temperaturverlauf, T-Shirt- und Jackenwetter anzeigen 17](#_Toc85991500)

[7 FOR-Schleife, Graph und Diagramm 18](#_Toc85991501)

[7.1 FOR-Schleifen-Test 18](#_Toc85991502)

[7.2 FOR-Schleifen-Test mit Array 18](#_Toc85991503)

[7.3 Quadrat-Zahlen im Array darstellen 18](#_Toc85991504)

[7.4 Quadrat-Zahlen im Array darstellen. Array zu Beginn löschen. Eigenschaftsknoten erstellen. 18](#_Toc85991505)

[7.5 Quadratzahlen mit Signalverlaufsgraph außerhalb der Schleife 20](#_Toc85991506)

[7.6 Quadratzahlen mit Signalverlaufsgraph außerhalb der Schleife, Array-Werte und Graph bei Programmstart löschen 20](#_Toc85991507)

[7.7 Quadratzahlen mit Signalverlaufsdiagramm innerhalb der Schleife 20](#_Toc85991508)

[7.8 Quadratzahlen mit Signalverlaufsdiagramm innerhalb der Schleife Diagramm zu Beginn löschen 20](#_Toc85991509)

[8 Übung Polynomzeichner mit For-Schleife, Formelknoten, Array 21](#_Toc85991510)

[8.1 Datentypen 21](#_Toc85991511)

[8.2 Schleifen 21](#_Toc85991512)

[8.3 Formelknoten 21](#_Toc85991513)

[8.4 Array 21](#_Toc85991514)

[8.5 Schleifentunnel 22](#_Toc85991515)

[8.6 Grafische Darstellung 22](#_Toc85991516)

[9 Polynomzeichner mit ungeradzahligen Werten von X 23](#_Toc85991517)

[9.2 Mehrdimensionale Tabelle und Datei erzeugen 24](#_Toc85991518)

[10 LED-Kennlinien-Aufnahme 26](#_Toc85991519)

[10.1 Spannung ausgeben und Spannung messen 26](#_Toc85991520)

[10.2 Manuelle Kennlinienaufnahme 27](#_Toc85991521)

[10.3 For-Schleife und Werte sammeln im Array 28](#_Toc85991522)

[10.4 Automatische Kennlinienaufnahme mit 10 Messwerten 29](#_Toc85991523)

[10.5 Automatische Kennlinienaufnahme 100 Messwerte 31](#_Toc85991524)

[11 Erweiterungen und Alternativen 32](#_Toc85991525)

[11.1 Kennlinienaufnahme mit Rückkopplungsknoten und „Array erstellen“ 32](#_Toc85991526)

[11.2 Diagramm jedes Mal löschen 34](#_Toc85991527)

[11.3 Diagramm löschen wählbar / warten auf neue Messwertaufnahme 35](#_Toc85991528)

[12 Kennlinienaufnahme Solarzellen 37](#_Toc85991529)

[13 MPP-Suche mit 100 U-I-Messwerten eines Solarpanels 39](#_Toc85991530)

[13.2 Theorieteil 43](#_Toc85991531)

[14 Datenaustausch PC – Controller über die COM-Schnittstelle 44](#_Toc85991532)

[14.1 Datenübertragung COM-Schnittstelle Controller 44](#_Toc85991533)

[14.2 COM-Schnittstelle mit Umwandlung der Variablenformate 45](#_Toc85991534)

[14.3 Fernsteuerung einer RGB-LED-Lichterkette am Controller über Labview am PC 47](#_Toc85991535)

[15 Zustandsdiagramm in Labview 49](#_Toc85991536)

[15.1 5 Zustände mit Case erstellen 49](#_Toc85991537)

[15.2 Ausgänge steuern 50](#_Toc85991538)

[15.3 In Abhängigkeit von Eingaben oder Sensoren den Zustand wechseln 50](#_Toc85991539)

[15.4 Aktueller Zustand leuchtet (LED) 52](#_Toc85991540)

[16 Bedienen und Beobachten mit dem NI Data-Dashboard 53](#_Toc85991541)

[16.1 Online Anleitung 53](#_Toc85991542)

[16.2 Download der APP für Android, Ipad 53](#_Toc85991543)

[16.3 Handling-Anleitung NI Data-Dashboard 53](#_Toc85991544)

[17 Handling-Anleitung Labview: Datenübertragung NI-Dashboard 53](#_Toc85991545)

[17.1 Konzept des NI-Dashboards 53](#_Toc85991546)

[17.2 Erstellen eines Labview-Programms zur Data-Dashboard-Übertragung 55](#_Toc85991547)

[17.3 Zeitliche Abläufe im Chart und Kennlinien im XY-Graph anzeigen 58](#_Toc85991548)

[18 Anleitung PWM in Labview mit myDAQ 60](#_Toc85991549)

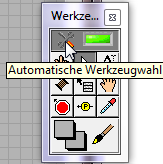
# Grundeinstellungen in Labview

## Neues Projekt

Labview-Startfenster → Datei → Neu

Menüleiste → Fenster → Nebeneinander

### Werkzeugpalette

Menüleiste → Ansicht → Werkzeugpalette

* in die Mitte zwischen beide Fenster
* Grundstellung: Automatische Werkzeugwahl
* Oft gebraucht:
  + Wert einstellen (am Frontpanel)
  + Text bearbeiten
  + Anschlüsse verbinden

## Grundsätzliche Arbeitsweise des Programms

### Frontpanel

* Eingaben des Anwenders am Bildschirm
* Ausgabe des Programms auf den Bildschirm

### Blockdiagramm

* Zu jeder Ein- und Ausgabemöglichkeit des Frontpanels gehört ein namensgleicher Block im Blockdiagramm
* Ausgabewerte der Blöcke sind rechts
* Eingabewerte für die Blöcke erfolgen links

## Grundeinstellungen der Menüs

* Rechtsklick auf Blockdiagramm
* Klick auf Doppelpfeil nach unten
* Sichtbare Paletten ändern  
  Haken bei
  + Programmierung
  + Mess-I/O
* Rechtsklick auf Frontpanel
* Klick auf Doppelpfeil nach unten
* Sichtbare Paletten ändern  
  Haken bei
  + Modern
  + System
  + Klassisch

# Spannung messen und Grenzwertüberwachung

## 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 Vergleicher und die logische Verknüpfung UND ermitteln die Bereiche.

### Handling-Anleitung

Vorbereitungen

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

Elemente plazieren

* Frontpanel → Modern → Numerisch → Drehregler
* Frontpanel → Modern → Numerisch → Rundinstrument

Komponten auf Blockdiagramm verbinden

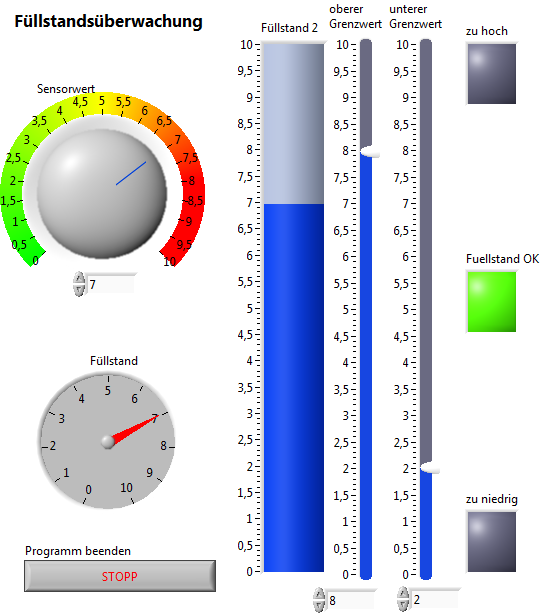
* Blockdiagramm → Rechtsklick → Programmierung → Strukturen → while-Schleife über den beiden Komponenten aufziehen
* Rechtsklick auf den roten Punkt der Schleife (Schleifenbedingung) → Erstellen → Bedienelement → Stopp-Button entsteht

Programm mit dickem weißem Pfeil unter der Menüzeile starten

Suchhilfen

* 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

## Erweiterung: Schieberegler für Grenzwerte, Standardwerte einstellen



### Handling-Anleitung

Feste zahlen der Grenzwerte durch Schieberegler ersetzen

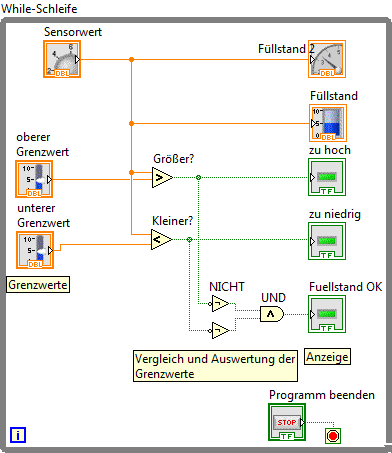
* Frontpanel → Rechtsklick → Modern → Numerisch → Schieber mit Zeiger
* Auf gleiche Größe wie den Füllstandstank ziehen

Zahlenanzeige sichtbar bei Schieberegler:

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

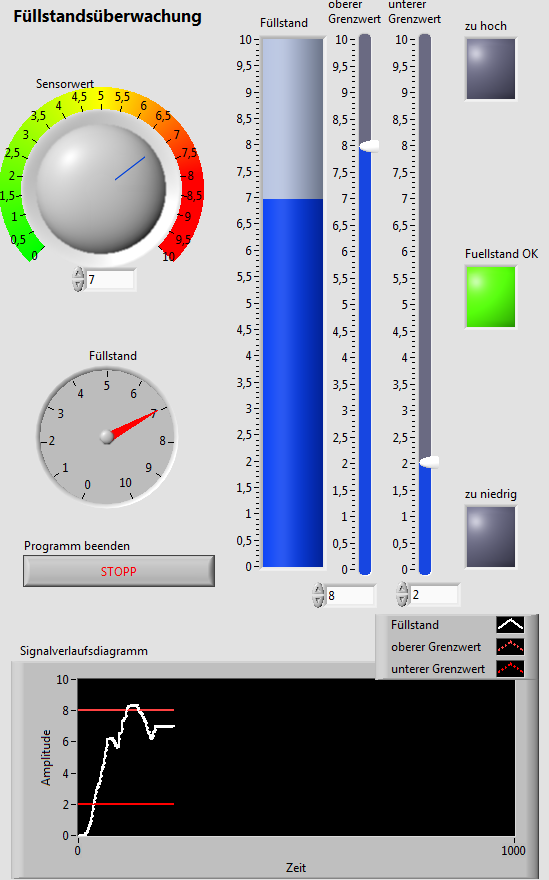
Nur eine Nachkommastelle bei Anzeige:

* 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

## Verlaufsdiagramm hinzufügen



### Handling-Anleitung

Verlaufsdiagramm hinzufügen

* Frontpanel → Rechtsklick → Modern → Graph → Diagramm

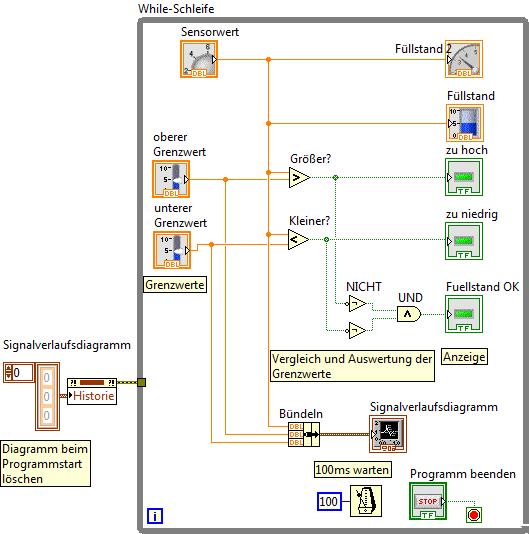
mehrere Kanäle im Diagramm anzeigen

* 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

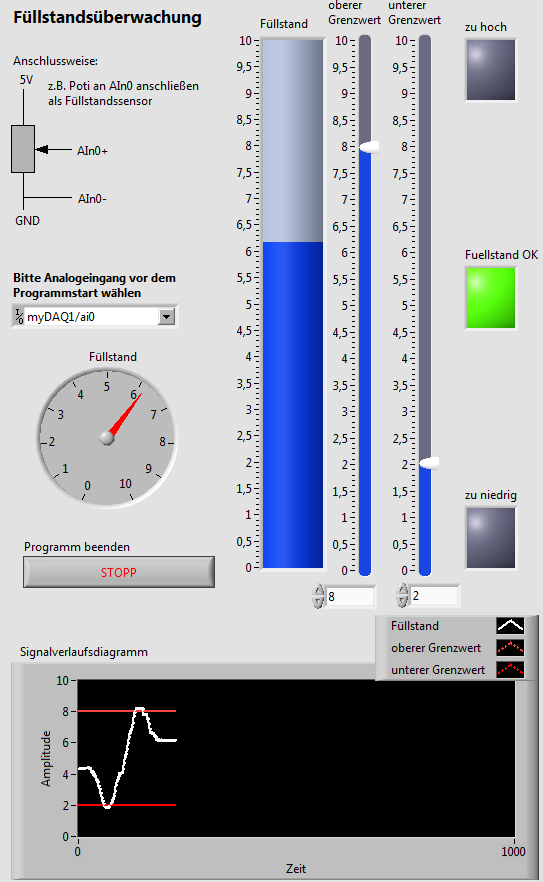
* Blockdiagramm → Rechtsklick auf das Diagramm → Erstellen → Eigenschaftsknoten → Historiendaten
* Komponente außerhalb der while-Schleife plazieren
* Rechtsklick → in schreiben ändern
* Rechtsklick auf den Eingang → Erstellen Konstante
* Ausgang der komponente auf den while-Schleifen-Rand ziehen,  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

Timer hinzufügen

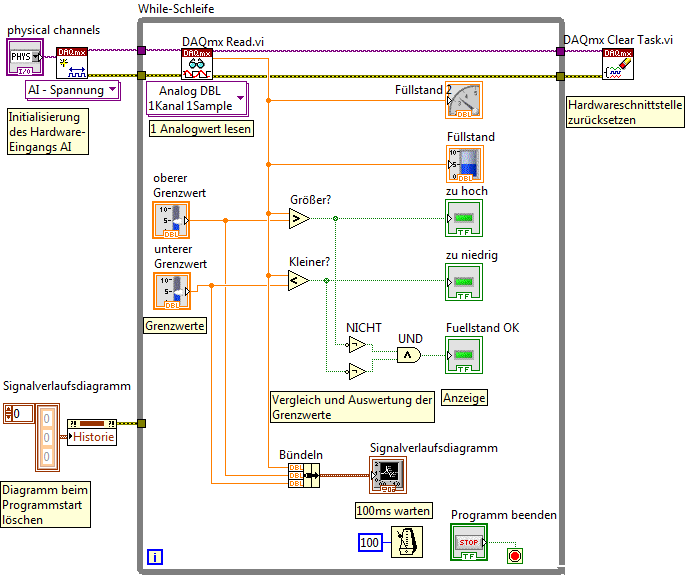
* Blockdiagramm → Rechtsklick → Programmierung → Timing → warten (Vielfaches)
* Rechtsklick auf den Eingang → Erstellen Konstante → 100
* → 100ms warten

## Hardware hinzufügen: Potentiometer als Füllstandssensor

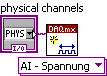
### Frontpanel



### Blockdiagramm



### Handling-Anleitung

Diese 3 Komponenten werden immer benötigt zum Lesen:

* Rechte Maus → MessIO → DAQmx → DAQmx VirtKan erzeugen
* Rechte Maus auf physikaleKanäle (2. Anschluss von oben)   
  → Erstellen Bedienelement
* Rechte Maus → MessIO → DAQmx → DAQmx Lesen
* Rechte Maus → MessIO → DAQmx → Task zurück

VirtKan erzeugen wird links außerhalb der Schleife plaziert da dies nur einmal am Programmanfang gemacht werden muss.

DAQmx Lesen wird für jede neue Messung benötigt, daher Plazierung in der Schleife.

Task zurück erfolgt erst beim Programmende, also rechts außerhalb der Schleife.

# Abstandsmessung mit Ultraschallsensor

## Aufgabenstellung und Vorbereitungen

In diesen Versuch wird ein Ultraschall-Entfernungsmesser in Betrieb genommen und mit einem Labview-Programm kalibriert.

Laut Datenblatt arbeitet der Sensor in einem Messbereich von 80 mm bis 1600 mm. Der Ausgang bildet diesen Messbereich linear auf eine analoge Spannung von 0 V bis 10 V ab.

Im dem dem Sensor beiliegenden Messprotokoll werden die Grenzen genauer spezifiziert, siehe Tabelle.

|  |  |  |
| --- | --- | --- |
| Messbereichsgrenzen | Abstand | Spannung |
| Untere Messwertgrenze | 80 mm | 0,03 V |
| Obere Messwertgrenze | 1600 mm | 9,93 V |

Daraus ergibt sich das angegebene U(s)-Diagramm.

### Gleichung s(U) erstellen

1600

s in mm

U inV

80

0,03

9,93

* Leiten Sie aus dem s(U)-Diagramm eine Geradengleichung der Form y(x) = mx + b ab   
  mit und erstellen Sie daraus die Gleichung s = m U + b mit den errechneten Zahlenwerten für m und b. (b bestimmen Sie z. B. über Δx = 9,93 und bekanntem m)

### Anschlussbelegung

Schließen Sie den Sensor an die myDAQ-Platine an:

|  |  |  |
| --- | --- | --- |
| 2 mm-Stecker-Leitung | Sensorleitung | myDAQ-Platine |
| rot | braun | +15 V |
| schwarz | blau | 0 V / GND |
| gelb | schwarz | Ain0+ |
|  |  | Ain0- an GND |

## Aufgaben

* Erstellen Sie ein Labview-Programm, das den Spannungswert am analogen Eingang ai0 fortwährend in einer while-Schleife misst und analog und digital anzeigt.
* Kalibrieren Sie den Sensor, indem Sie die Spannung mithilfe der oben gefundenen Formel in einen Abstand umrechnen. Stellen Sie den Abstand (zusätzlich zur Spannungsanzeige) mit einer Tankanzeige dar, die auch einen Zahlenwert anzeigt. Anzeige in mm ohne Nachkommastelle.
* Eine von zwei LEDs sollen leuchten, wenn der Messbereich über- oder unterschritten wird.  
  (Spannung zu groß oder zu klein)
* Mithilfe eines Timing-Moduls (warten) soll nur 2-mal pro Sekunde eine Messung erfolgen.
* Innerhalb der while-Schleife wird eine FOR-Schleife eingefügt, die dafür sorgt, dass 10 Messwerte gezählt und in einem Schleifentunnel gesammelt werden.

Anleitung: Datei speichern, dann FOR-Schleife auswählen und mit der Maus ein Fenster über allen Blocksymbolen außer stop aufziehen (innerhalb der while-Schleife). Auf die entstandenen Schieberegister (Pfeile bei den Leistungen, die in die FOR-Schleife führen) rechtsklicken und in Schleifentunnel umwandeln.

* Stellen Sie die 10 Messwerte in einem Signalverlaufsgraph dar. (Im Blockdiagramm außerhalb der FOR-Schleife!).   
  Stellen Sie die 10 Messwerte auch in einem Array dar. (Rechstklick auf Schleifentunnel: Estellen → Anzeigeelement)
* Fügen Sie ein Signalverlaufsdiagramm ein. (Im Blockdiagramm in der FOR-Schleife!)
* Der Wertebereich beider Diagramme soll 80mm bis 1600mm betragen.  
  (Eigenschaften → Autoskalierung Y-Achse aus)
* Erklären Sie kurz die Unterschiede beider Diagramme.
* Kommentieren Sie Ihr Programm (Texte einfügen)

## Lösung

### Frontpanel mit Erklärungen

### Blockdiagramm mit Kommentaren

# Füllstandsmessung mit Ultraschallsensor

### Aufgabe

### Frontpanel mit Erklärungen

### Blockdiagramm mit Kommentaren

# Abstandsmessung mit optischem Sensor

## Spannungswert des optischen Sensors auslesen

### Funktion

Der optische Sensor GP 2Y0A02 von Sharp benötigt eine Spannungsversorgung von 5V. Er gibt zwischen dem gelben Anschluss und der schwarzen Leitung eine Spannung aus, die abhängig ist vom Abstand eines Objekts, das sich vor dem Sensor befindet. Der Arbeitsbereich beträgt 20 cm bis 150 cm.

### Handling-Anleitung

Den Sensor wie rechts angegeben anschließen.

Zur Spannungsmessung mit Labview werden 3 Komponenten benötigt:

* Blockdiagramm → Rechtsklick → MessI/O → DAQmx → Datenerfassung → Virtuellen Kanal erzeugen: Diese Komponente stellt eine Verbindung zwischen der Messbox DAQmx und Labview her. Es wird ein „Task“ erzeugt.
* Blockdiagramm → Rechtsklick → MessI/O → DAQmx → Datenerfassung → Lesen: Diese Komponente liest einen analogen Spannungswert vom eingestellten Kanal ein.
* Blockdiagramm → Rechtsklick → MessI/O → DAQmx → Datenerfassung → Task zurück: Diese Komponente setzt den Task zurück und gibt die USB-Schnittstelle wieder frei z. B. für andere Programme.

Einstellen der Messart

* An der Komponente „virtuellen Kanal erzeugen“ auf den kleinen Pfeil des Auswahlfeldes klicken und Analoge Erfassung → Spannung einstellen
* An der Komponente „Lesen“ auf den kleinen Pfeil des Auswahlfeldes klicken und Analog → Ein Kanal → Ein Sample → DBL einstellen

Einstellen des Eingangs (Kanals) und der verwendeten Messbox durch den Anwender

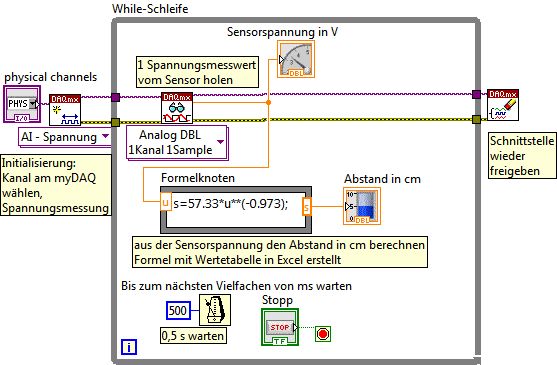
* Rechtsklick auf den Eingang „Physikalische Kanäle“ an der Komponente „Virtuellen Kanal erzeugen“ → Erstellen → Bedienelement

## Zusammenhang zwischen Spannung des Sensors und Abstand ermitteln

* Mit Labview die Spannung des Sensors bei verschiedenen Abständen messen.
* Abstand Sensor – weißes Blatt mit Meterstab messen
* Messtabelle in Excel oder calc erzeugen
* aus den Messwerten ein X,Y-Diagramm erstellen mit der Spannung auf der X-Achse.

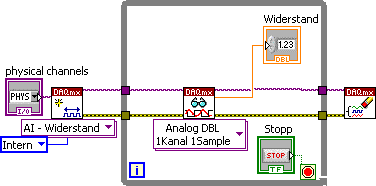
## Formelknoten zur Berechnung des Abstands einfügen

In einen Formelknoten kann eine kleines C-Programm oder auch nur eine Gleichung eingefügt werden.

* Blockdiagramm → Rechtsklick → Programmierung → Strukturen → Formelknoten
* Rechtsklick auf den linken Formelknotenrand → Eingang hinzufügen,   
  u in den Eingang schreiben
* Rechtsklick auf den rechten Formelknotenrand → Ausgang hinzufügen,   
  S in den Eingang schreiben
* Formel als Text in den Formelknoten schreiben  
  Kommazahlen mit Punkt!!!!!  
  mal ist \*  
  hoch ist \*\*  
  Formel mit ; abschließen!!! (Ende eines Befehls in der Programmiersprache C)  
  Groß- und Kleinschreibung der Variablen s und u beachten!!!
* Tankanzeige erstellen und formatieren, Messbereich maximal 150 oder 200
* Eingang und Ausgang des Formelknotens anschließen

# Widerstand und Temperatur messen mit PT1000

## Einfache Widerstandsmessung



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)

## Widerstandsmessung und Temperaturbestimmung mit PT1000

### Auflösung der PT1000-Widerstandsgleichung nach der Temperatur

*R* = *R*0 · (1 + *α* · *t*) wobei *α* = 3,85 · 10-3 / °C , *R*0 Widerstand bei 0°C: 1000 Ohm

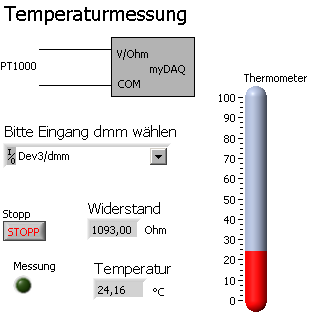
*R* / *R*0= 1 + *α* · *t*

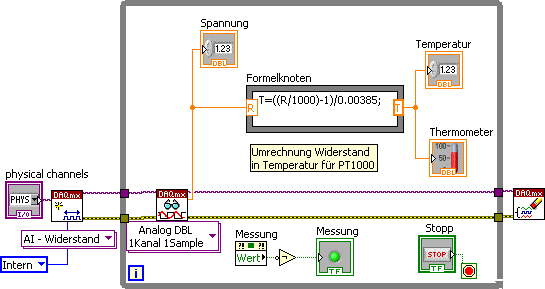
*R / R0 - 1 = α · t*

*t = (R / R0 -1) / α*

*t = (R / 1000 – 1) / 0,00385 °C wenn R der gemessene Widerstand ist.*

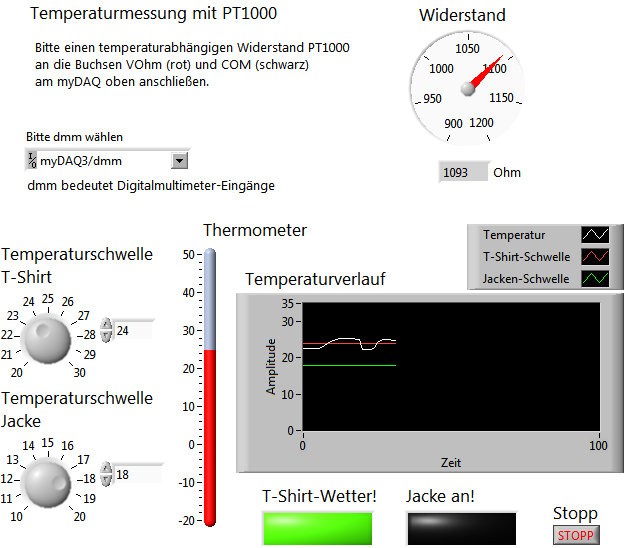
### Blockdiagramm





## Erweiterungen: Temperaturverlauf, T-Shirt- und Jackenwetter anzeigen

### Frontpanel und gewünschte Funktion



Im dargestellten Frontpanel sind die Temperaturschwellen von 24°C für T-Shirt-Wetter und 18°C für Jackenwetter eingegeben.

Wenn die gemessene Temperatur größer als 24°C ist, leuchtet die grüne LED T-Shirt-Wetter.

Wenn die gemessene Temperatur kleiner als 18°C ist, leuchtet die rote LED „Jacke an“!

### Blockdiagramm

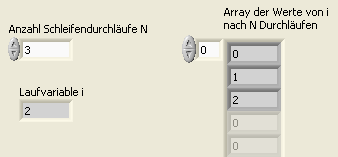
# FOR-Schleife, Graph und Diagramm

## FOR-Schleifen-Test

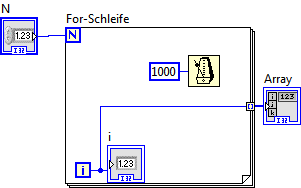
Testen Sie die Funktion der For-Schleife:

* 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.

## 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.

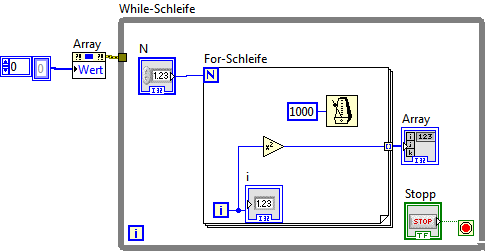
## Quadrat-Zahlen im Array darstellen

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

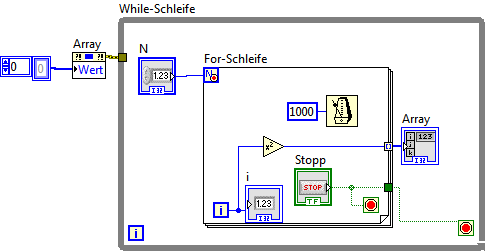
## Quadrat-Zahlen im Array darstellen. Array zu Beginn löschen. Eigenschaftsknoten erstellen.

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 Progamm-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



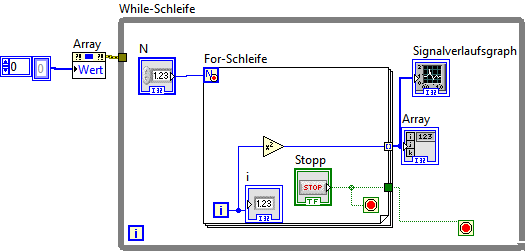
* Bei Programmstart werden die Werte des Arrays gelöscht.

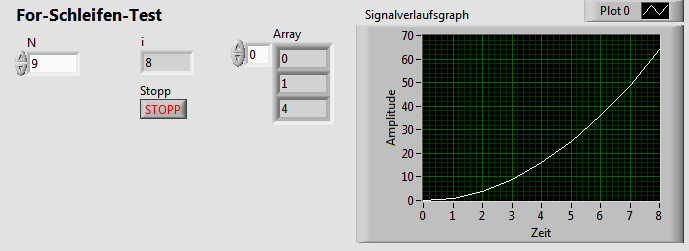


## Quadratzahlen mit Signalverlaufsgraph außerhalb der Schleife

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

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





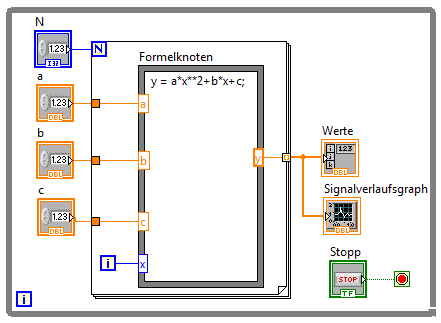
## Quadratzahlen mit Signalverlaufsdiagramm innerhalb der Schleife

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

## Quadratzahlen mit Signalverlaufsdiagramm innerhalb der Schleife Diagramm zu Beginn löschen

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

# Übung Polynomzeichner mit For-Schleife, Formelknoten, Array



## Datentypen

Grüne Linien und Symbole:

Blaue Linien und Symbole:

Orange Linien und Symbole:

## Schleifen

Unterschiede von while- und for-Schleifen:

N:

i:

## 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 Ausgangen muss mit Gleichung übereinstimmen!
* An den Eingängen Bedienelemente erstellen und außerhalb der For-Schleife plazieren

## 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

## 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 Rechtklick auf den Tunnel → Tunnelmodus einstellt.

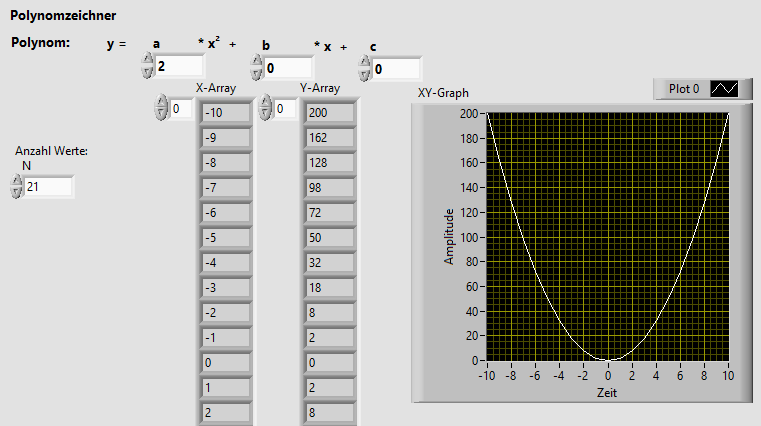
## Grafische Darstellung

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

Signalverlaufsdiagramm

Signalverlaufsgraph

# Polynomzeichner mit ungeradzahligen Werten von X



### Aufgaben

Ergänzen Sie die erhaltene Datei mit den Arrays, der Komponente Bündeln und der XY-Graph-Anzeige.

Testen Sie das Programm und ermitteln Sie die Aufgabe der Konstanten unter i durch Ändern der Werte.

Stellen Sie nacheinander mehrere Ihnen aus der Mathematik bekannte Funktionen dar.

### Anleitungen

* Arrays erstellen durch Rechtsklick auf Schleifenknoten → Erstellen Anzeigeelement
* Array auf Blockdiagramm nach unten ausziehen
* xy-Graph auf Frontpanel erzeugen
* Bündeln bei Rechtsklick Blockdiagramm → Programmierung → Cluster

## Mehrdimensionale Tabelle und Datei erzeugen

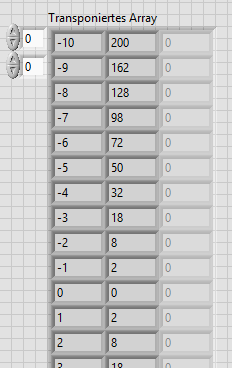
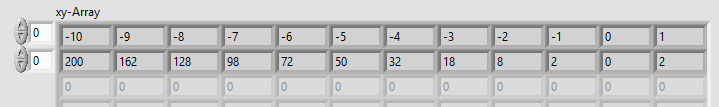
### Aufgaben

Ergänzen Sie Ihr Programm, testen Sie die Funktion und erklären Sie die Aufgabe der Komponenten.

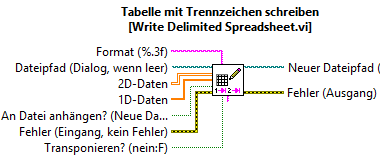
* Array erstellen und transponieren bei Rechtsklick Blockdiagramm   
  → Programmierung → Array
* Tabelle schreiben bei Rechtsklick Blockdiagramm → Programmierung   
  → Datei IO, Anschluss der Leitung an   
  2D-Daten
* Bei Beendigung des Programms mit dem Stopp-Button, werden Sie aufgefordert, einen Dateinamen und den Pfad anzugeben. Wählen Sie einen Ordner in Ihrem Homeverzeichnis und benennen Sie die Datei tabelle.txt.

### Bildschirmkopie der Datei (Ausschnitt)

### Bildschirmkopie XY-Array ohne / mit Transponieren



### Transponierte Textdatei schreiben

* Geben Sie Werte als Tabelle mit x- und y-Spalten in die Datei aus, indem Sie eine Konstante True (T) an Transponieren? anschließen.
* Tragen Sie in eine Konstante am Eingang Dateipfad h:\ordnerpfad\tabelle.txt ein.
* Schließen Sie ein Anzeigeelement an Neuer Dateipfad an, damit Sie sehen, wohin gespeichert wird.
* Öffnen Sie die Datei auf dem Netz auch mit dem Tablet.
* Öffnen Sie die Datei auch mit Excel.

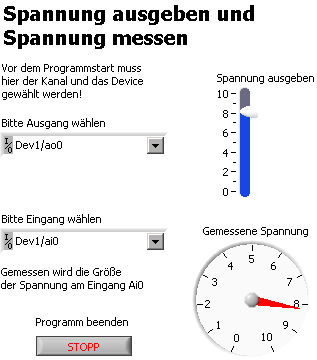
### Daten in Exceldatei schreiben

* Schließen Sie zusätzlich die Komponente Blockdiagramm   
  → Programmierung → Datei IO → Werte schreiben an.
* Konfigurations-Einstellungen
  + Dateiname: Laufwerk:\pfad\dateiname
  + Wenn Datei existiert Datei überschreiben
  + Dateiformat Excel
  + Spalten für x-Werte nur eine Spalte
* Öffnen Sie die Datei auf dem Netz auch mit dem Tablet.
* Öffnen Sie die Datei auch mit Excel.
* Erstellen Sie in Excel ein XY-Diagramm und eine Trendlinie mit Funktionsgleichung.

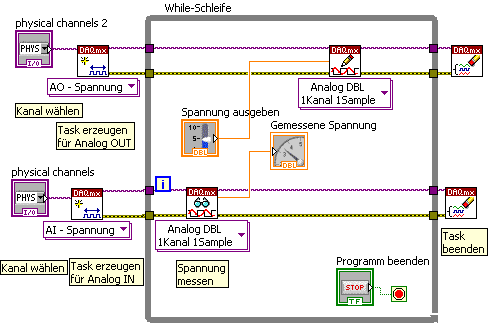
# LED-Kennlinien-Aufnahme

## Spannung ausgeben und Spannung messen

### Frontpanel

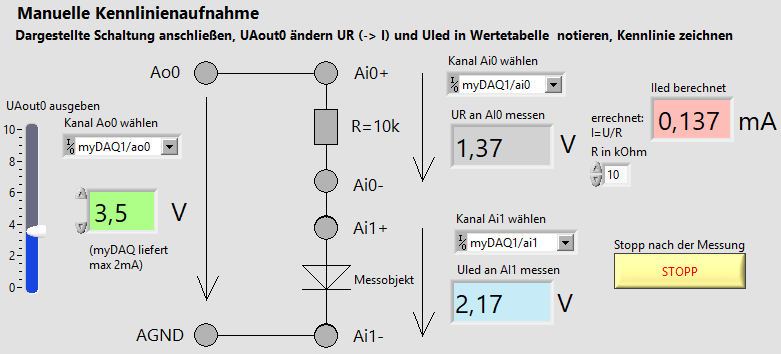
* Zum Ausgeben und zum Messen benötigt man jeweils einen Kanal
* Beim Ausgeben muss man AO-Spannung angeben und die Komponente „schreiben“ verwenden.

### Blockdiagramm



## Manuelle Kennlinienaufnahme

### Frontpanel und Erklärung des Prinzips



Eine LED wird in Reihe mit einem Vorwiderstand an eine Gesamtspannung angeschlossen.

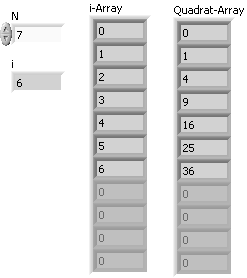
Die Gesamtspannung wird links am Schieber verändert. Vergrößert man die Gesamtspannung, so beginnt die LED zu leuchten, sobald die Durchlassspannung der LED überschritten ist. Dann wird auch die Spannung am Widerstand größer null, da nun ein Strom durch die Reihenschaltung fließt.

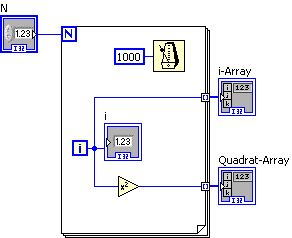
Mit I = U\_R / R kann die Größe des fließenden Stromes berechnet werden. Die Spannung an der LED wird ebenfalls gemessen.

Die Wertepaare von I und U\_LED können nun für verschiedene Gesamtspannungen in ein I(U)-Diagramm eingetragen werden. Daraus erhält man später die Kennlinie der LED.

### Blockdiagramm

## For-Schleife und Werte sammeln im Array





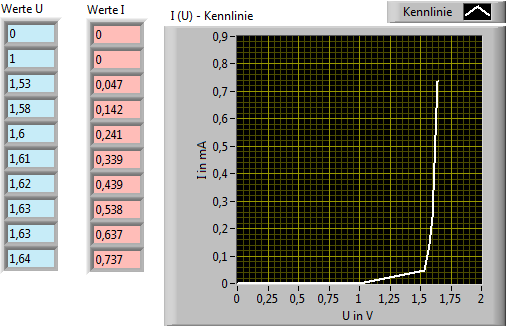
* Die FOR-Schleife wiederholt ein Programm so oft, wie bei N angegeben ist.
* I zählt dabei von 0 bis N-1 (Bsp oben: N=7, i=6 beim Programmende)
* führt man eine Leitung auf den rechten Schleifenrand, so entsteht ein Tunnel, in dem die Werte gespeichert werden.
* Schließt man ein Anzeigeelement außen an den Tunnel an, so erstellt LV ein Array. (Man muss die Anzeige am schwarzen Rechteck nach unten ziehen, damit man die Felder sieht)
* Die Werte werden erst nach dem Schleifendurchlauf in den Arrays dargestellt.

## Automatische Kennlinienaufnahme mit 10 Messwerten

### Frontpanel: Bedienung



### Frontpanel: Wertetabelle und I(U)-Kennlinie



### Blockdiagramm

* 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

## Automatische Kennlinienaufnahme 100 Messwerte

### Frontpanel



### Blockdiagramm

* 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 Uges = 0 bis 9,9V in 0,1V-Schritten durchgeführt.

# Erweiterungen und Alternativen

## Kennlinienaufnahme mit Rückkopplungsknoten und „Array erstellen“

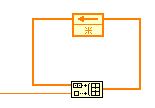
In den vorhergehenden Aufgaben wurden die x- und y-Werte von der For-Schleife gespeichert und zu einem Array zusammengefügt.

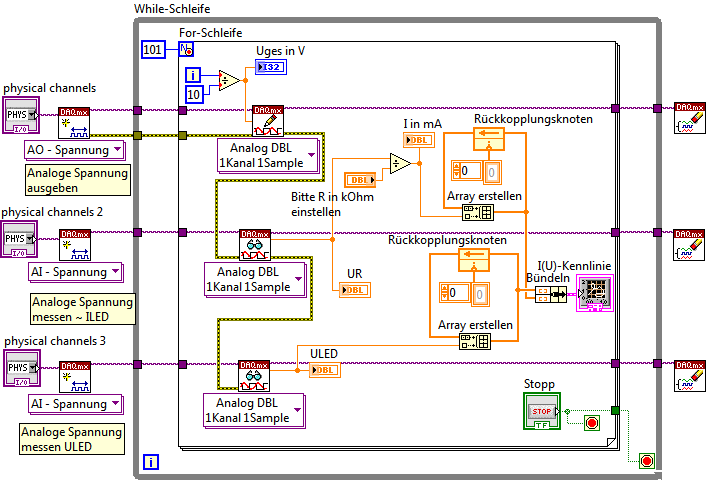
Diese hat den Nachteil, dass auf in der I(U)-Kennlinie die Werte erst dann angezeigt werden, wenn die gesamte Messreihe aufgenommen ist.

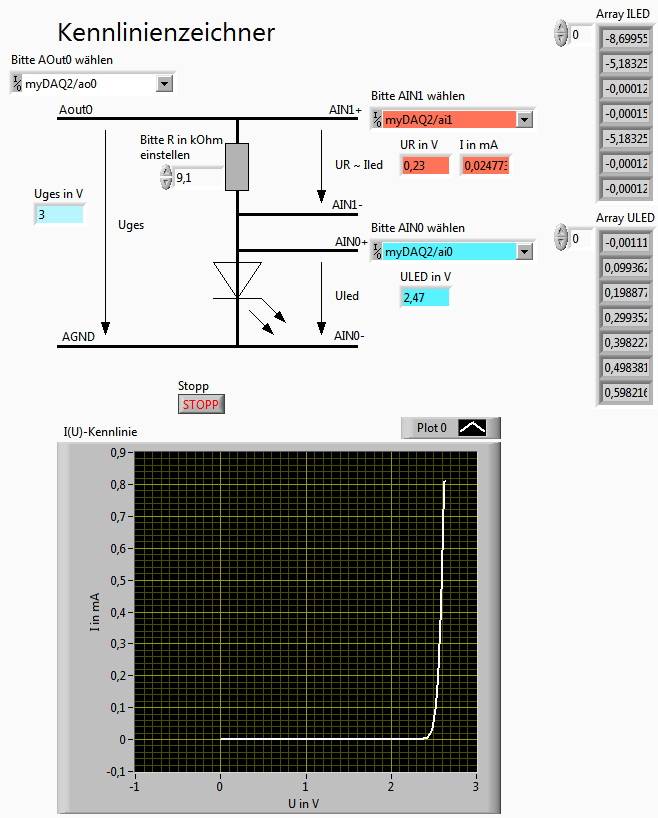
In diesem Beispiel erzeugen wir uns das Array selbst mit der Komponente „Array erstellen“:

### Handling-Anleitung

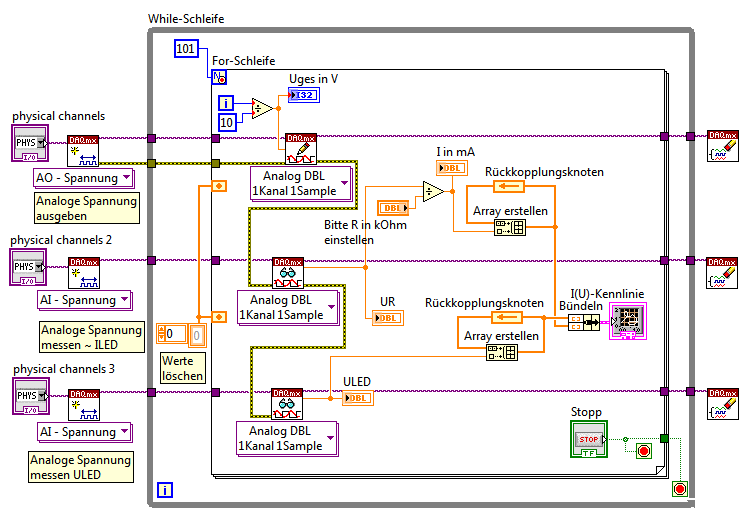
Array erstellen

* 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.





## Diagramm jedes Mal löschen



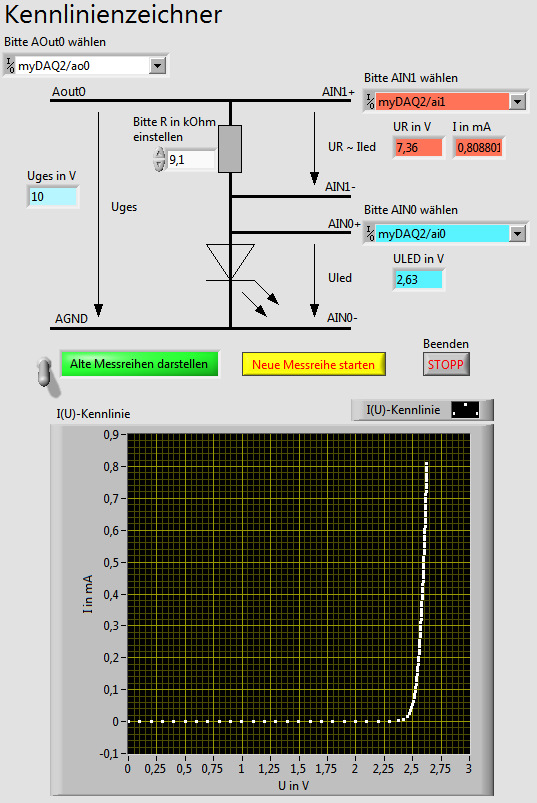
Die Array-Werte speichert der Rückkopplungsknoten. Wenn man das Diagramm bei einer neuen Messung neu entstehen soll, muss man die Werte im Rückkopplungsknoten löschen. Dies geschieht mit dem Initialisierungsanschluss unten.

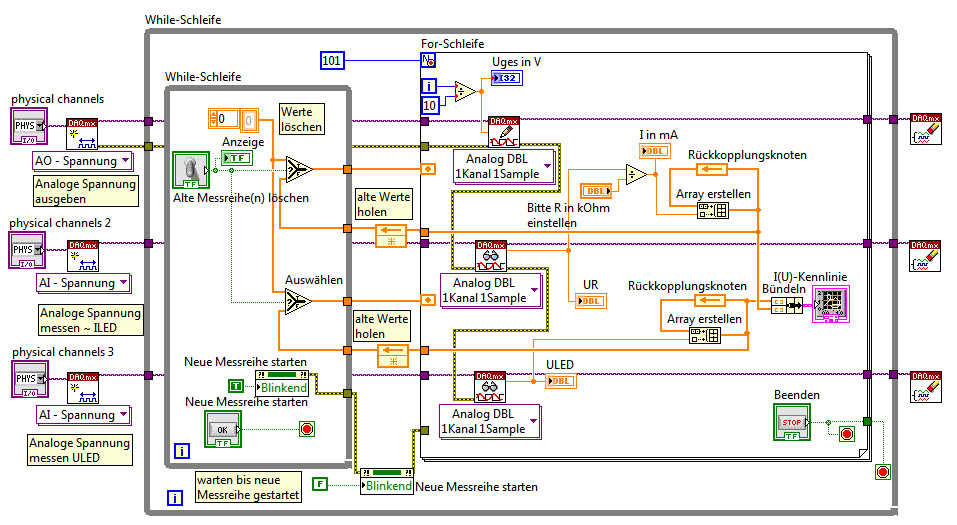
### Handling-Anleitung

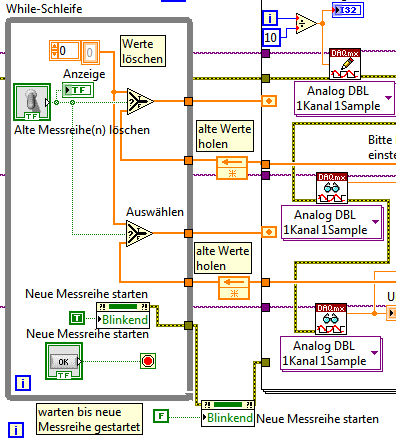
Rückkopplungsknoten initialisieren

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

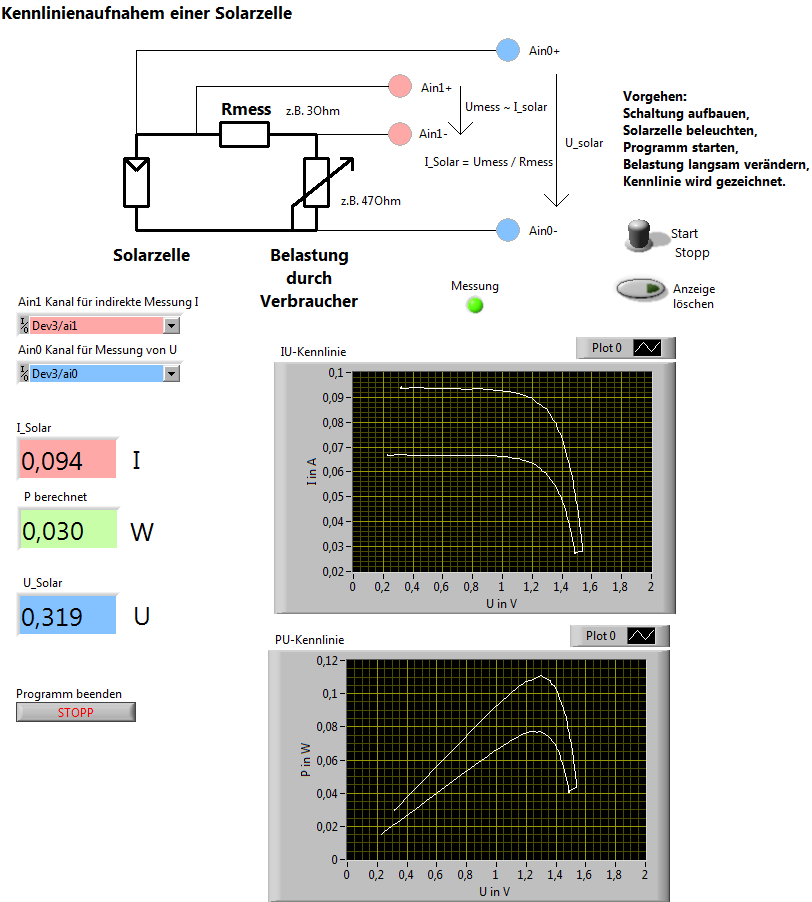
## Diagramm löschen wählbar / warten auf neue Messwertaufnahme



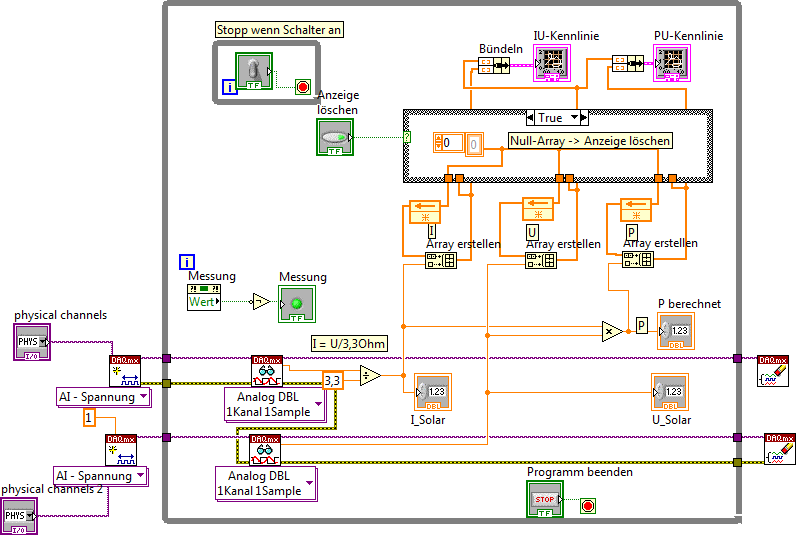




# Kennlinienaufnahme Solarzellen



Erklärungen folgen



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

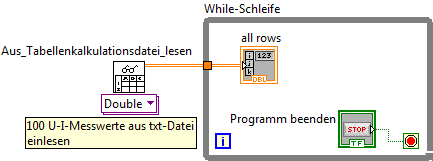
### Allgemeines:

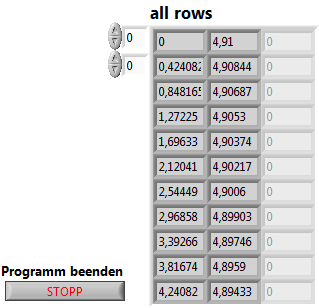
* Beschriften Sie alle Komponenten immer so, dass Sie später verstehen, welche Aufgabe die Komponenten haben. z. B. im Bild von 13.1.3: U-Array, I-Array usw.
* Speichern Sie jede Teilaufgabe extra ab. 13.1.2 erhält Nr 1, 13.1.3 erhält Nr 2 usw.
* Achten Sie auf saubere Leitungsführung und ordnen Sie die Komponenten auf dem Frontpanel und dem Blockdiagramm möglichst so an, wei bei den einzelnen Teilaufgaben angegeben.
* Sie benötigen die Datei 100UI\_Messwerte\_Kennlinie\_Solarzelle.txt
* Fragen Sie, wenn Sie nicht weiterkommen

### 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.

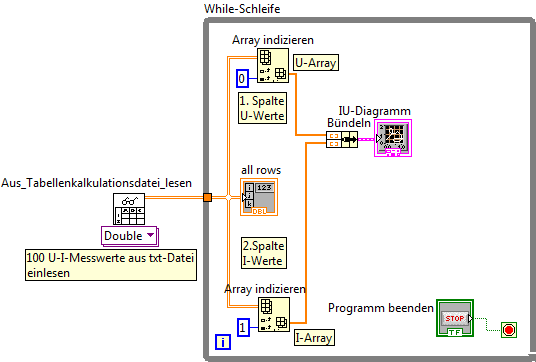


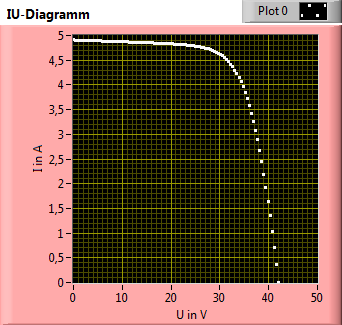


### Spalten aus der Tabelle extrahieren und I-U-Kennlinie erstellen

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.





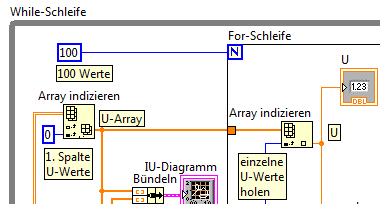
### P-U-Kennlinie hinzufügen

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

### 100 einzelne Werte von U und I extrahieren und anzeigen

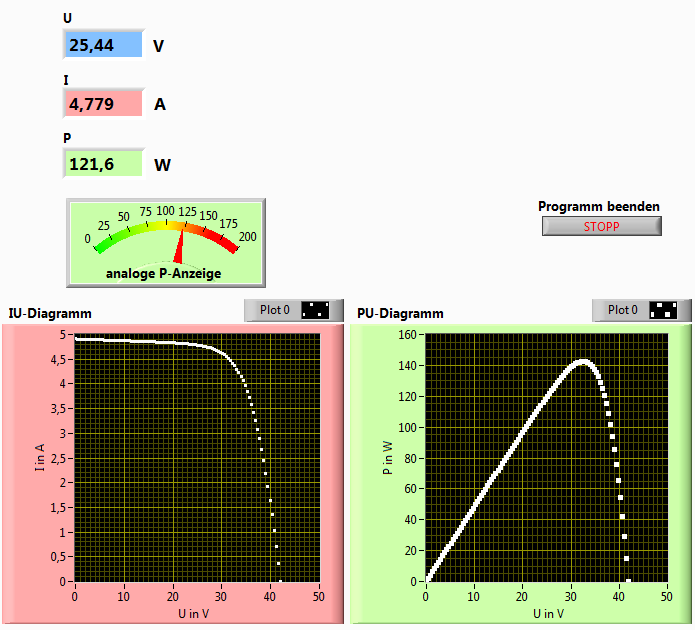
Löschen Sie die Anzeige der Wertetabelle (all rows) auf dem Bildschirm. (Platz schaffen!)

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

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

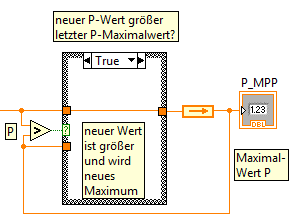


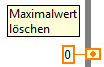
Anleitung: Indizierung deaktivieren bei den Schleifentunneln von U und I durch Rechtsklick auf die Schleifentunnel.

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

### MPP der Leistung feststellen

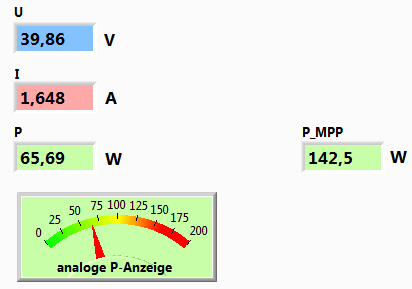
In der For-Schleife soll der maximale Wert (MPP) der Leistung ermittelt werden. Dazu wird der gerade angezeigte alte Wert mit dem neuen Wert verglichen. Je nachdem, welcher der beideb Werte größer ist, wird der neue oder alte Wert als Maximalwert übernommen.





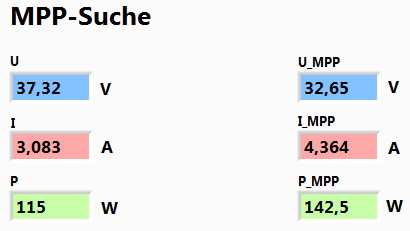
Erst wenn Ihr Programmteil funktioniert, testen Sie folgendes: Klicken Sie auf den Rückführungsknoten rechts und wählen Sie "Initialisierung eine Schleife nach hinten verschieben". Dann Rechtsklick auf Initialisierungsanschluss und Konstante erstellen.

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



### 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!!)



### 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.

### Diagramme neu zeichnen

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

## Theorieteil

### Datentypen

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

### Schleifen

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

### Arrays

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

# Datenaustausch PC – Controller über die COM-Schnittstelle

## Datenübertragung COM-Schnittstelle Controller

### 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*,OUTP); **// Port0 auf Ausgabe**

**while**(1U) **// 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(empfbyte); **// anzeigen**

}

delay\_ms (100); **// Zeitverzoegerung**

}**//while**

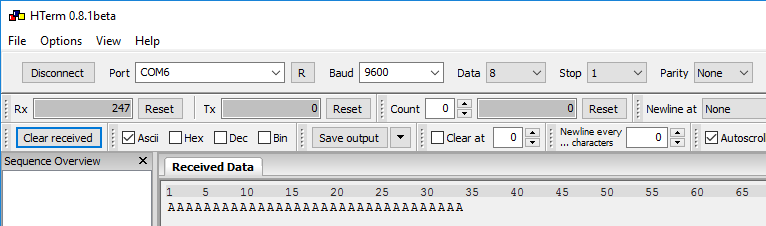
}**//main**

### 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.

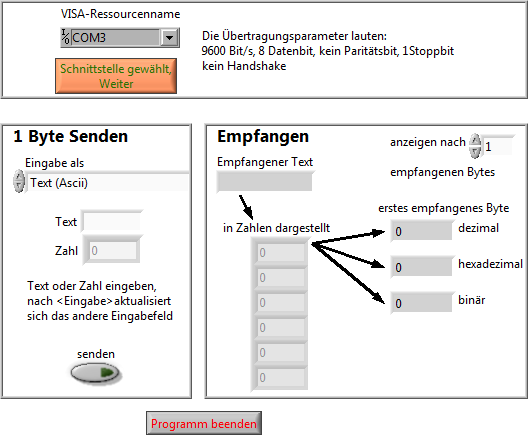
### Hterm-Programm

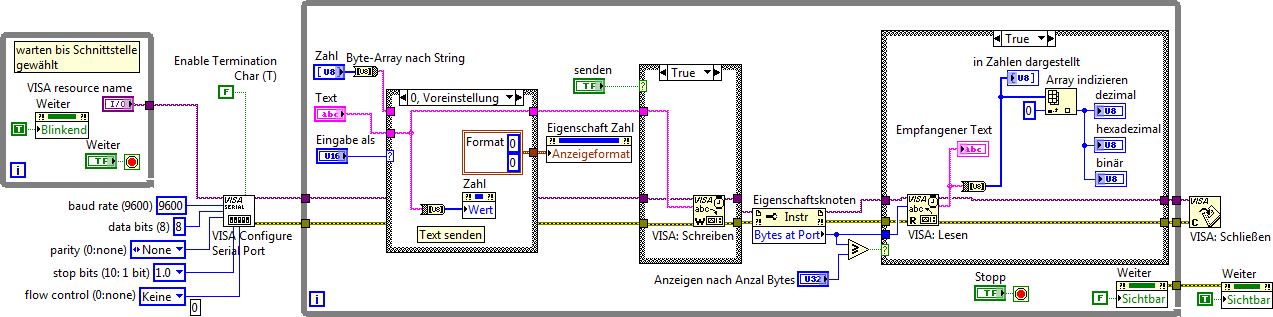
Empfangene Zeichen werden als Ascii-Zeichen dargestellt: AAAA…

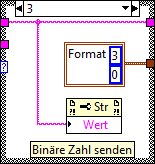
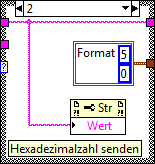
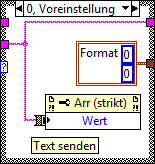
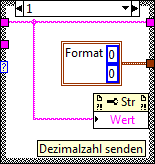
Protokoll: 9600 Bit/s, 8 Datenbit, 1 Sopbit, kein Paritätsbit

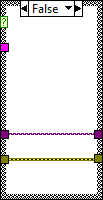
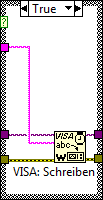
USB-Verbindung der Entwicklungsumgebung Dave wird für die Datenübertragung verwendet um am PC als COM6 für die Com-Verbindung verwendet.

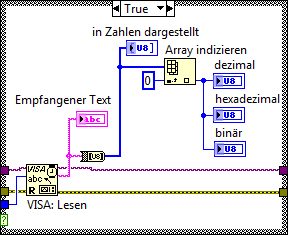
## COM-Schnittstelle mit Umwandlung der Variablenformate

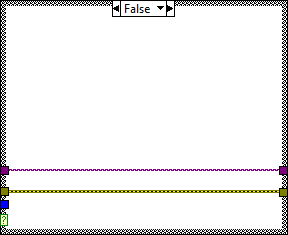






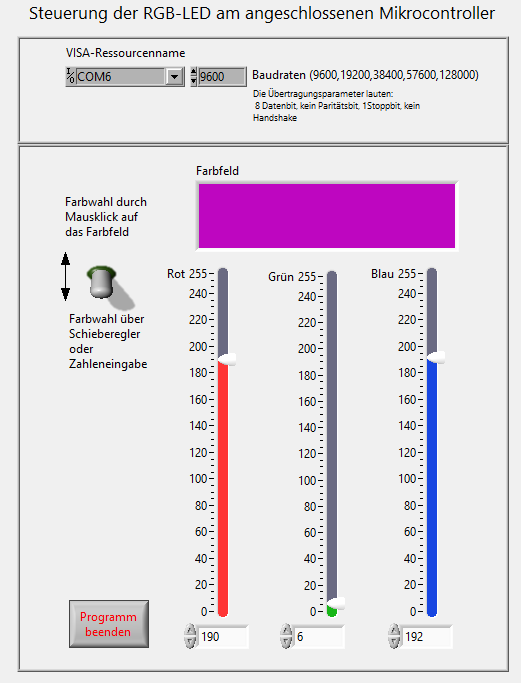






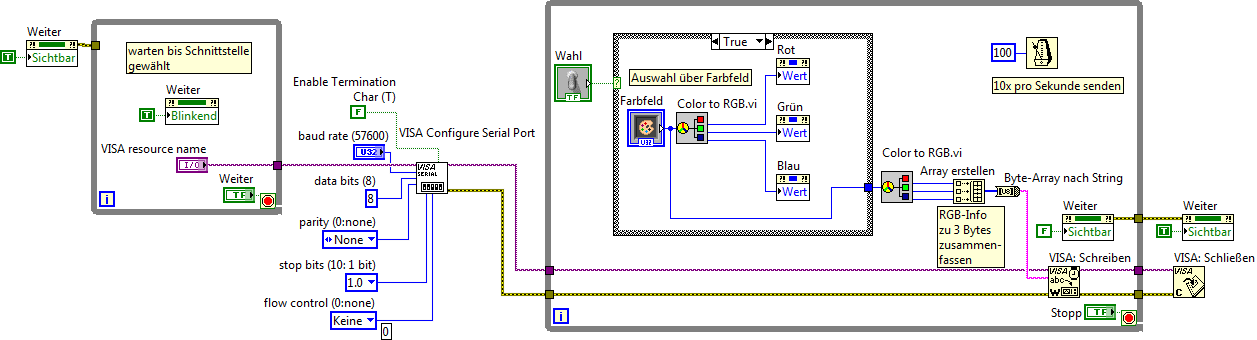
## Fernsteuerung einer RGB-LED-Lichterkette am Controller über Labview am PC

### Frontpanel



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

### Blockdiagramm



### 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**

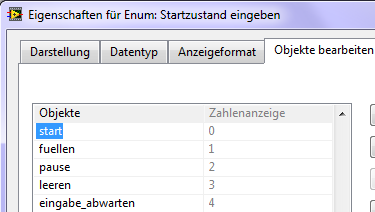
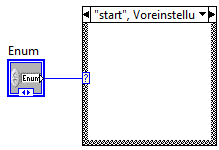
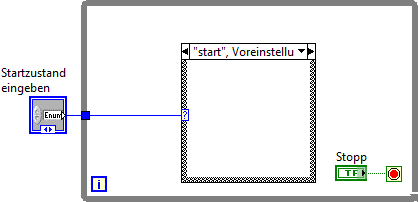
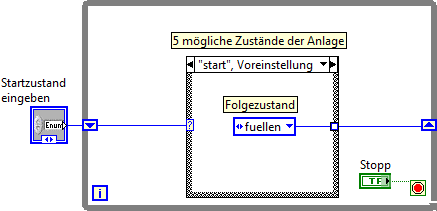
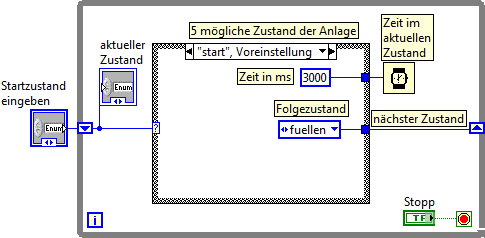
pwm3\_duty\_cycle(~b\_byte);

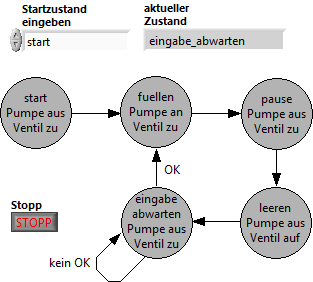
}**//while**

}**//main**

# Zustandsdiagramm in Labview

## 5 Zustände mit Case erstellen

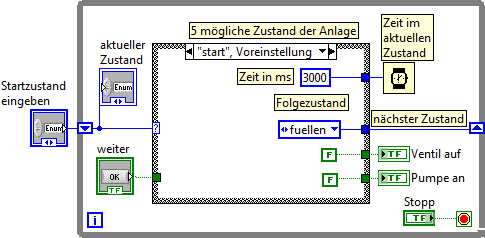
* Programmierung → Strukturen → Case-Struktur erstellen
* Programmierung → Numerisch → Enum-Konstante erstellen
* rechte Maus → in Bedienelement ändern
* rechte Maus → Eigenschaften → Objekte bearbeiten
* Als Objekte die Zustände eingeben:
* start, fuellen, pause, leeren, eingabe\_abwarten
* Nun wurden den Werten 0 bis 5 unsere Zustandsnamen zugeordnet.
* Das Enum-Bedienfeld nun an das ? der Case-Struktur anschließen: 2 von unseren 5 Zustände wurden automatisch als Cases übernommen.  
  rechte Maus auf die Case-Struktur   
  → Case danach einfügen (3x durchführen)   
  → auch die anderen Zustände wurden übernommen.
* While-Schleife einfügen, Enum-Eingabe wird außerhalb der Schliefe plaziert, hier wird später der vom Bediener der Startzustand eingegeben.
* Rechte Maus auf den Schleifentunnel  
  → durch Schieberegister ersetzen
* Schieberegister:  
  Der Wert der Enum-Variable wird im Zustand der Case-Struktur geändert und rechts aus der Schleife „herausgegeben“, zwischengespeichert und steht beim nächsten Schleifendurchlauf „links“ wieder zur Verfügung. Beim ersten Schleifendurchlauf wird der am Eingabefeld eingegebene Wert verwendet.
* Rechte Maus auf „?“ → Konstante erstellen  
  Diese Konstante im Case plazieren und rechts an das Schieberegister anschließen.  
  Diese Konstante gibt den Folgezustand an.
* Für alle 5 Cases die Folgezustände angeben.
* Mit einem Anzeigeelement wird der aktuelle Zustand angezeigt.
* Nach Programmstart sollen die Zustände automatisch alle 2s gewechselt werden:  
  Start → Füllen → Leeren → Füllen → Leeren



## Ausgänge steuern

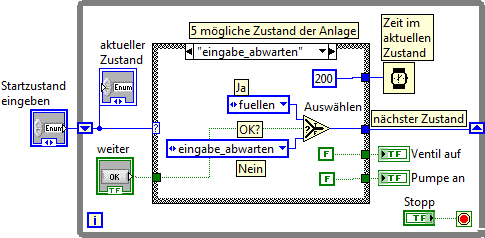
Die Ausgänge „Ventil“ und „Pumpe“ werden zustandsabhängig gesteuert.

In allen Zuständen werden daher entsprechende Konstanten (False F für aus und True T für an) angeschlossen.



## In Abhängigkeit von Eingaben oder Sensoren den Zustand wechseln

Beispiel: Zustand eingabe\_abwarten wird nur verlassen, wenn OK gedrückt wurde:

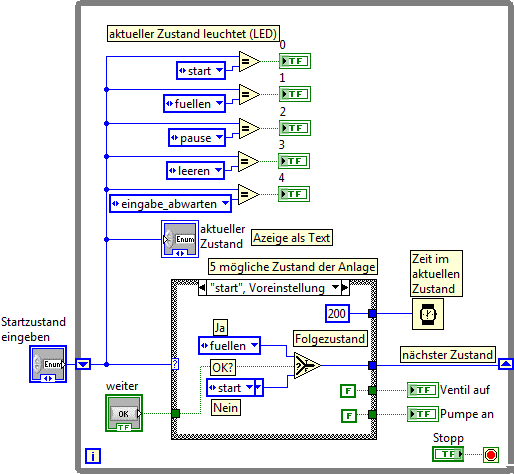
Die Komponente Auswählen hat die gleiche Funktion wie eine Case-Struktur mit True und False



|  |  |
| --- | --- |
| Zustand 1 | Zustand 2 |
| Zustand 3 | Zustand 4 |
| Zustand 5 | |

## Aktueller Zustand leuchtet (LED)





# Bedienen und Beobachten mit dem NI Data-Dashboard

## Online Anleitung

[Erste Schritte mit dem Data Dashboard for LabVIEW](http://www.ni.com/tutorial/13757/de/)

## Download der APP für Android, Ipad

Für Android nicht in den Google-Store gehen, dort ist nur die alte Version (1.0) zu finden.

Nach **ni data dashboard apk** suchen → neueste Version 2.3.0 (Stand 11/2018)

Zur Installation muss man bei den App-Einstellungen in Android die Installation „fremder Apps“ freigeben. Hersteller ist National Instruments (NI)

Fürs Ipad findet man die neueste Version im App-Store

## Handling-Anleitung NI Data-Dashboard

Ist für Android und Ipad weitgehend identisch und wir hier gut erklärt:

[Erste Schritte mit dem Data Dashboard for LabVIEW](http://www.ni.com/tutorial/13757/de/)

# Handling-Anleitung Labview: Datenübertragung NI-Dashboard

## Konzept des NI-Dashboards

  
Quelle: National Instruments, <http://www.ni.com/white-paper/14033/de/>

Auf dem Smartphone oder Ipad läuft die App Data Dashboard, die ähnlich einer Labview-Frontpanel-Darstellung alle gewünschten Werte grafisch darstellt und die Fernsteuerung mithilfe von Bedienelementen ermöglicht.

Das Dashboard ist lediglich eine grafische Oberfläche zur Darstellung von über WLAN, LAN übertragenen Daten.

Die eigentliche Steuerung erfolgt mit dem PC, auf dem das Labview-programm läuft. Dieser PC arbeitet auch als Sever für die Datenübertragung.

Im Labview-Programm werden Variablen erstellt, über die alle Werte zwischen den Apps Data-Dashboard und Labview ausgetauscht werden.

Auf der Folgeseite sind die beteiligten Komponenten in einem Blockschaltbild dargestellt.

**Labview-Projekt**

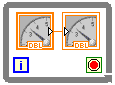
Frontpanel

Bibliothek mit **Umgebungsvariablen**

zum Datenaustausch über WLAN

Blockdiagramm

Bedienung durch Anwender



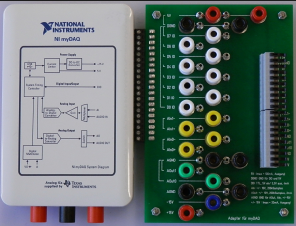
Steuerungs-Programm

VI

**PC = Labview-Server**

**Ein-/Ausgabe**

USB



**Hardware**

Versuchs-

aufbau

**WLAN-Router**

LAN

IP des PC

z. B. 192.168.0.248

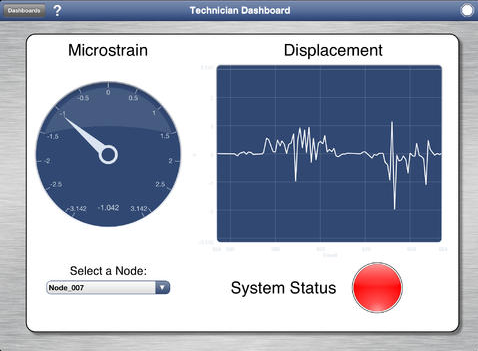
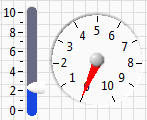
Netz-Name,PW

**Smartphone, Tablet**

WLAN: einloggen in Netzname mit PW

WLAN

App: **NI-Dashboard**



Bedienen und Beobachten durch Anwender

myDAQ

Datenaustausch über **Umgebungsvariablen**

**NI-Dashboard:**

- IP des PC

- Name Bibliothek mit **Variablen**

- Zuordnung **Variablen** zu Anzeige- und

Bedienelementen des Dashboards

**Labview:**

- Projekt anlegen (Neu!)

- Bibliothek mit **Variablen** erstellen

- Zuordnung **Variablen** zu Anzeige- und

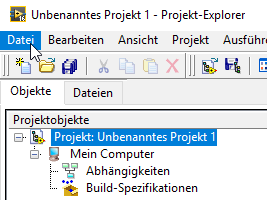
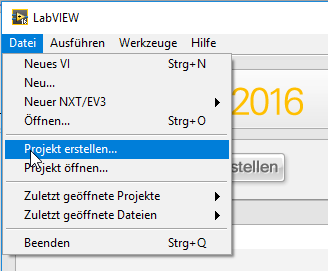
Bedienelementen im Blockdiagramm

Blockschaltbild mit den beteiligten Komponenten bei der Nutzung des NI-Data-Dashboards

## Erstellen eines Labview-Programms zur Data-Dashboard-Übertragung

### Projekt erstellen

Labview öffnen



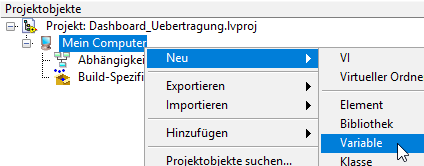
Datei → Projekt erstellen

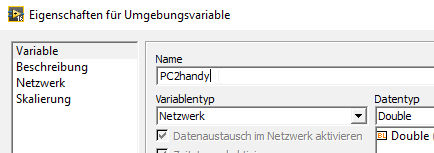
leeres Projekt → Fertigstellen

Im Projekt-Explorer: Datei → Speichern unter → Name vergeben

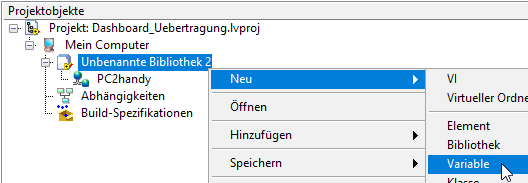
### Variablen zur Datenübertragung erstellen

Auf Mein Computer Rechtsklick → Neu → Variable



sinnvollen Variablen-Namen eintragen

Auf **unbekannte Bibliothek** Rechtsklick → Neu → Variable



weitere Variable(n) erstellen und sinnvoll benennen

Anmerkung: Falls Sie auf Mein Computer Rechtsklicken entsteht eine neue Bibliothek! Falls dies passiert ist, einfach durch Klick auf die ungewünschte Bibliothek und den Schere-Button diese wieder löschen.

### Unbekannte Bibliothek umbenennen

durch Rechtsklick auf unbekannte Bibliothek → Öffnen

Dann Datei → Speichern unter → sinnvollen Namen vergeben

Bibliotheks-Fenster schließen

### Ergebnis:

### VI erstellen (Blockdiagramm und Frontpanel)

Rechtsklick auf „Mein Computer“ → Neu → VI

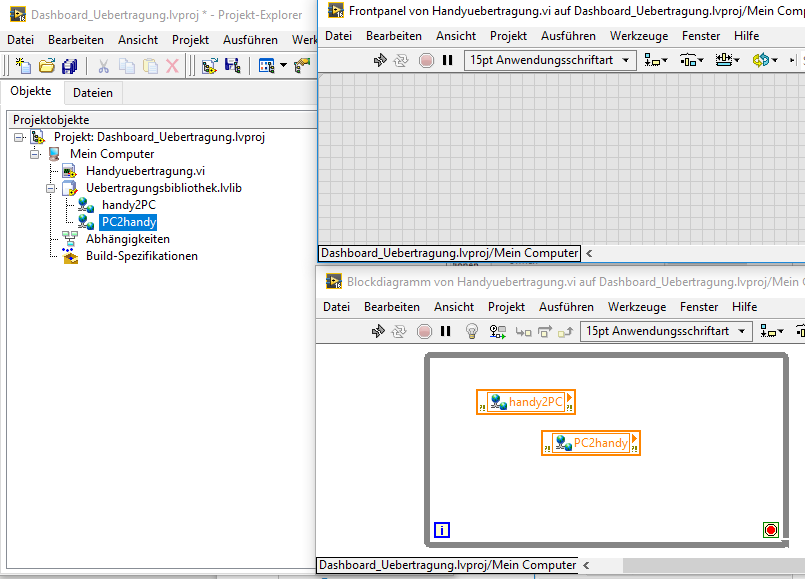
Im Blockdiagramm oder Frontpanel: Datei → Speichern unter → sinnvoller Name

Die 3 Fenster Projekt-Explorer, Blockdiagramm, Frontpanel auf dem Bildschirm anordnen

While-Schleife im Blockdiagramm erstellen

Variablen aus dem Projekt-Explorer ind die while-Schleife ziehen

Ergebnis:



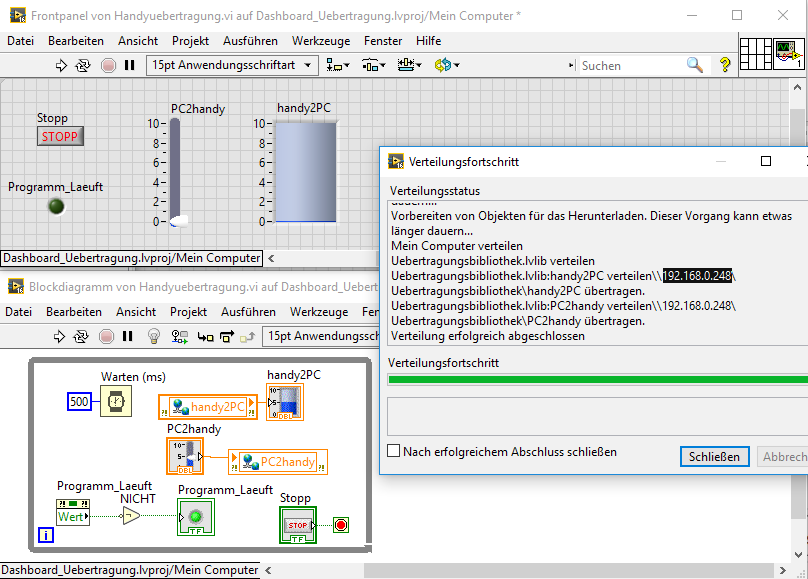
### Variablen mit der Steuerung verbinden

Variable mit Übertragungsrichtung Handy → PC: Anzeigeelement erstellen

Variable mit Übertragungsrichtung PC → Handy:   
- Rechtsklick auf die Variable → Zugriffsmodus → schreiben  
- Bedienelement erstellen

Timing: Datenübertragung alle 500 ms!

Anzeige-LED für Arbeitsgeschwindigkeit / Programm-läuft-Anzeige  
- LED auf Frontpanel erzeugen  
- Rechtsklick auf LED im Blockdiagramm → Erstellen → Eigenschaftsknoten → Wert  
- Eigenschaftsknoten und LED mit einem Boolschen NICHT verbinden



Programmstart

IP-Adresse

### Labview-Programm und -Server starten

Klick auf weißen Pfeil

Im Fenster Verteilungsfortschritt steht die IP-Adresse des PCs, die im Data-Dashboard auf dem Handy / Ipad bei den Variablen eingegeben werden muss.

Nach Klick auf Schließen startet das Programm

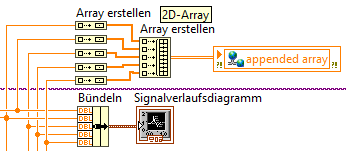
**Achtung:**

**Beim Test im heimischen Netzwerk muss die Windows-Firewall ausgeschaltet sein oder eine entsprechende Freigabe in den „Tiefen der Firewall“ eingetragen werden.**

## Zeitliche Abläufe im Chart und Kennlinien im XY-Graph anzeigen

### Unterschied zeitliche Abläufe für Frontpanel und Dashboard darstellen

Sollen auf dem Dashboard z. B. mehrere zeitabhängige Kurven in einem Diagramm dargestellt werden, so muß die Umgebungsvariable im Blockdiagramm erstellt werden.

Rechts sieht man die notwendigen Komponenten zur Darstellung auf dem PC und zur Darstellung auf dem Dashboard.

1) Für die Darstellung auf dem Frontpanel bündelt man die Siganle und schließt sie an ein Siganlverlaufsdiagramm an

2) Für die Übertragung an das Dashboard erstellt man aus jedem einzelnen Signal zunächst ein eigenes Array (im Bild 5x Komponente Array erstellen notwendig).  
Dann führt man diese Arrays (, die jeweils nur aus einem Wert bestehen,) auf eine weitere Komponente Array erstellen. Dadurch entsteht ein 2D-Array (also eine Tabelle)

### Umgebungsvariable für zeitliche Abläufe erstellen

- Rechtsklick auf den Ausgang des 2D-Arrays   
 → Erstellen → Umgebungsvariablenknoten   
 → neue Umgebungsvariable

Falls dies nicht funktioniert, siehe 17.3.3

Diese neue Umgebungsvariable kann nun im Projekt-Explorer in die eigene Bibliothek kopiert oder verschoben werden:



- bei beiden Bibliotheken mit + alle Variablen  
 anzeigen, Ergebnis rechts.

- Umgebungsvariable (hier appended array)  
 mit Drag and Drop (mit gedrückter Maustaste)   
 in eigene Bibliothek verschieben

<Strg> - <C> kopieren, <Strg> - <X> ausschneiden, <Strg> - <V> einfügen geht auch.

- Rechtsklick auf die Variable → Umbenennen

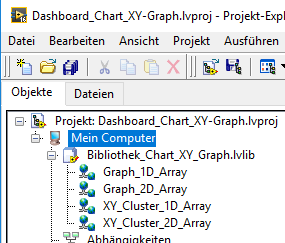
- Mit Klick auf leere Unbenannte Bibliothek 1 <Entf>   
 oder Rechtsklick → Aus Projekt entfernen   
 kann diese gelöscht werden. (sonst sieht man diese später im Dashboard)

### Umgebungsvariable aus bestehender Bibliothek kopieren

Falls das Real-Time-Modul nicht installiert ist, funktioniert das Erstellen des Umgebungsvariablenknoten nicht wie oben beschrieben.

Dann kopieren Sie die Bibliothek Bibliothek\_Chart\_XY\_Graph.lvlib (erhältlich beim Bub) in ihren Ordner.

Anschließend klicken im Projekt-Explorer auf „Mein Computer“ und fügen mit Rechtsklick → Hinzufügen → Datei (Ort und Name der Datei angeben) die Bibliothek zu Ihrem Projekt hinzu.

In der Bibliothek befinden sich die Umgebungsvariablen Graph\_1D\_Array für einen Kurvenverlauf und Graph\_2D-Array für mehrere Kurvenverläufe.

Mit der Variable XY\_Cluster\_1D\_Array erstellt man XY-Diagramme, z. B. IU-Kennlinien, mit XY\_Cluster\_2D\_Array können mehrere Kennlinien in ein Diagramm gezeichnet werden.

Das Kopieren und Umbenennen der Variablen ist in 17.3.2 erklärt.

### Variablen im Dashboard verwenden

Die 2D-Variable kann mit einem Graph oder Chart (Indicators) verwendet werden. Bei den Properties (Zahnrad) kann man bei Plots die gewünschte Anzahl an Kurven hinzugefügen und diese sinnvoll benennen.

### Variable erstellen für XY-Graph im Dashboard

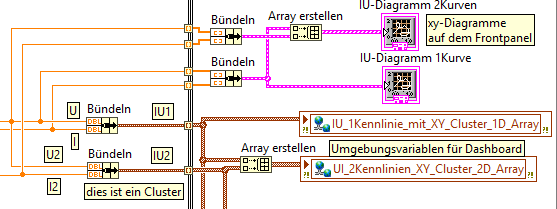
Sollen auf dem Dashboard z. B. eine oder mehrere Kennlinien in einem XY Graph dargestellt werden, so muß in Labview ein Cluster aus den Werten von X und Y (mit der Komponente bündeln) und anschließend ein Array dieser Clusterwerte erstellt werden.

Unten sieht man die notwendigen Komponenten zur Darstellung auf dem PC und zur Darstellung auf dem Dashboard.

1) Für die Darstellung auf dem Frontpanel erstellt man zuerst ein Array z. B. mit dem Schleifentunnel einer FORschleife und bündelt anschließend beide zur Darstellung einer Kennlinie auf dem XY-Diagramm.  
Für 2 Kennlinien müssen die gebündelten Werte der einzelnen Kennlinien zu einem Array zusammengeführt werden.

2) Für die Übertragung an das Dashboard bündelt man zuerst, führt dann auf den Tunnel der Schleife, wodurch ein Array entsteht, und schließt dann an die XY\_Cluster\_1D\_Array-Umgebungsvariable aus der Bibliothek oben an.

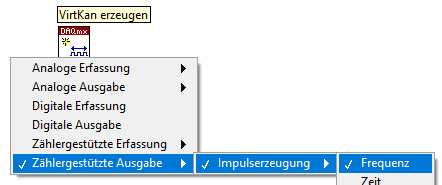
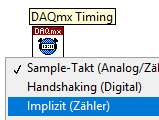
Für 2 Kennlinien werden die beiden Arrays zu einem weiteren Array zusammengefürht, es entsteht ein 2D-Array, an das die entsprechende XY\_Cluster\_2D\_Array-Umgebungsvariable aus der Bibliothek angeschlossen werden kann.



For-Schleifen-Rahmen

# Anleitung PWM in Labview mit myDAQ

[Englische Anleitung](http://www.ni.com/tutorial/2991/en/)



Komponenten wie abgebildet plazieren und verbinden

5 Bedienelemente an VirtKan erzeugen anschließend

Konstante am Sample-Modus Eingang DAQmx-Timing erstellen und auf kontinuierlich einstellen

