

## Die W.E.St.-Skriptsprache

### Inhaltsverzeichnis

1	Anwendungsbereich, Nutzen und Vorteile.....	3
2	Grundkonzept.....	3
3	Ein erstes, einfaches Beispiel.....	4
4	Befehlsübersicht.....	4
5	Eine praktische Übung.....	6
6	Online am Modul.....	11
6.1	Verbinden und Daten auslesen.....	11
6.2	Offline erstelltes Skript laden oder Skript mit verbundenem Modul eingeben.....	12
6.3	Beobachtungsmodus.....	12
7	Funktionsreferenz.....	14
7.1	Mathematische Funktionen.....	14
7.1.1	DIR.....	14
7.1.2	ADD.....	14
7.1.3	SUB.....	14
7.1.4	MUL.....	14
7.1.5	DMUL.....	14
7.1.6	LIM.....	14
7.1.7	SQRT.....	14
7.1.8	SIN.....	15
7.1.9	ABS.....	15
7.1.10	NORM / NORML / UNORM.....	15
7.1.11	INTEG.....	16
7.1.12	PT1.....	16
7.1.13	MIN / MAX.....	16
7.2	Logik.....	17
7.2.1	SEL.....	17
7.2.2	GT / GE / LT / LE.....	17
7.2.3	AND / OR / NOT.....	17
7.2.4	RS.....	17
7.3	Zeitfunktionen.....	18
7.3.1	RAMP.....	18
7.3.2	TE.....	18
7.3.3	TA.....	18
7.3.4	FP / FN.....	19
7.3.5	FUR / FUS / FUT.....	19
7.4	Sonstige Funktionen.....	20
7.4.1	FUN2.....	20
7.4.2	SPAR.....	20
7.5	Modulspezifische Komplexfunktionen.....	21
8	Anwendungsbeispiele.....	21

8.1	Oszillierender Zylinderantrieb .....	21
8.1.1	Aufgabe .....	21
8.1.2	Lösung 1 .....	22
8.1.3	Lösung 2 .....	23
8.2	Gleichlauf von 2 Zylindern (Vorgabe der Bewegungsrichtung über Schalteingänge) .....	25
8.2.1	Aufgabe .....	25
8.2.2	Lösung .....	25
8.2.3	Skript .....	26
8.3	Aufteilung eines Sollwertsignals auf 2 Verstellpumpen .....	27
8.3.1	Aufgabe .....	27
8.3.2	Konzept .....	27
8.3.3	Lösung .....	28
8.3.4	Skript .....	29
8.4	Leistungsbegrenzungsregelung .....	30
8.4.1	Aufgabe .....	30
8.4.2	Lösung .....	30
8.4.3	Skript .....	31
8.5	Ansteuerung für einen Lagerprüfstand .....	32
8.5.1	Aufgabe .....	32
8.5.2	Lösung .....	32
8.5.3	Skript .....	33
8.6	Frequenzgangmessung .....	34
8.6.1	Aufgabe .....	34
8.6.2	Lösung .....	34
8.6.3	Skript .....	36
8.6.4	Ergebnisse .....	36

**W.E.St.** Elektronik GmbH

Gewerbering 31  
41372 Niederkrüchten

Tel.: +49 (0)2163 577355-0

Homepage: [www.w-e-st.de](http://www.w-e-st.de)  
E-Mail: [contact@w-e-st.de](mailto:contact@w-e-st.de)

Datum: 21.04.2026

## 1 Anwendungsbereich, Nutzen und Vorteile

Anwendungsorientierte Lösungen für die Elektrohydraulik: Dieses Motto charakterisiert das Erfolgsrezept unserer Produkte. Es bedeutet, dass die Funktion der Geräte einfach verständlich und zumeist intuitiv beherrschbar ist.

„Parametrieren statt Programmieren“ ist ein weiterer Leitgedanke. Der Anwender soll es auf diese Weise sehr leicht haben, da die Struktur des Programmes sinnvoll für die typischen Anwendungen der Hydraulik aufgebaut ist.

Es entsteht hier jedoch ein Konflikt zwischen einem einfachen Programmaufbau und der Flexibilität, spezielle Anforderungen realisieren zu können.

Die hier vorgestellte Skriptsprache löst dies, indem sie den festen Kern der Regelungsanwendung mit einer flexiblen Hülle ergänzt, die genauso einfach aufzusetzen ist.

Auf diese Weise werden die mit der Skriptsprache ausgestatteten Geräte noch vielseitiger und lassen sich problemlos an alle denkbaren Aufgaben anpassen.

Die weiteren Vorteile sind:

- SPS-Funktionalität kann durch Skriptfunktionen ersetzt werden
- Extrem schnelle Verarbeitung im Zyklus von 1 ms (beinhaltet nicht nur die Verarbeitung, sondern auch die Signal Ein- und Ausgabe)
- Einfachster Aufbau der Befehle, schnelle Einarbeitung
- Möglichkeit einer Offline-Simulation der Skripte auch ohne Hardware
- Auch Regelfunktionen sind realisierbar

Die Hydraulik stellt besondere Anforderungen an die Automatisierung, da bedingt durch die hohe Dynamik der Komponenten eine extrem schnelle Signalverarbeitung im ms-Zyklus zumindest vorteilhaft und oft unumgänglich ist. Selbst wenn die Geschwindigkeit der zu steuernden Antriebe eher gering erscheint, ist dies aus Gründen der Stabilität (Schwingneigung ist vorhanden bei Totzeiten) und Genauigkeit ein wichtiger Aspekt. SPSen erfüllen diese Anforderungen nur selten. Bedingt durch die Notwendigkeit, hier die Baugruppen mit besonderer Geschwindigkeit und Auflösung zu verwenden, sind spezialisierte Geräte meistens wirtschaftlicher und viel einfacher in der Handhabung.

Hier gibt es nun eine Möglichkeit gleichzeitig eine große Flexibilität zu behalten, und dies zu einem unschlagbaren Preis.

## 2 Grundkonzept

Ein Skript besteht aus einer Liste, in der vordefinierte Speicherzellen aufgelistet werden. Für jede dieser Zellen kann man eine Funktion angeben, mit welcher der Inhalt der Speicherzelle berechnet wird.

Während der Laufzeit werden diese Funktionen zyklisch aufgerufen und der Zelleninhalt neu berechnet.

Es gibt zwei Arten von Zellen, nämlich frei verwendbare (M1 ... Mxx) und fest mit Ausgängen verbundene Zellen. Der Inhalt Letzterer wird entweder direkt an physikalische Ausgänge des Gerätes weitergegeben oder dient als Eingangssignal in eine fest definierte interne Funktion, beispielsweise als Sollwert eines Reglers.

Die Benennung und Funktion der Zellen ist geräteabhängig, M... - Zellen gibt es immer.

Die aufgerufenen Funktionen können bis zu drei Parameter haben, die selber wiederum Speicherzellen oder physikalische Eingangssignale sind.

### 3 Ein erstes, einfaches Beispiel

Betrachten wir das folgende Skript:

```
M1          = GT  PIN14  PAR1
M2          = LT  PIN14  PAR2
...
LED_YR     = RS  M1     M2
```

In der ersten Zeile (M1) wird geprüft, ob das Eingangssignal an PIN14 größer als ein parametrierbarer Festwert (PAR1) ist.

In der zweiten Zeile (M2) wird geprüft, ob das Eingangssignal an PIN14 kleiner als ein zweiter Parameter ist.

Das Ausgangssignal, mit dem die rechte gelbe LED des Gerätes angesteuert wird, ist der Schaltzustand eines RS-Flipflop, welches mit den Ergebnissen dieser Vergleiche verbunden ist.

Wie man erkennt, können die Speicherzellen sowohl die Bedeutung eines Analogwertes wie auch einer boolschen Variable haben. Der Skript-Interpreter wertet einen Inhalt  $\geq 1,0$  als logisch „WAHR“ und Funktionen, die einen logischen Ausgangswert liefern, setzen die entsprechende Speicherzelle auf 0 oder 1,0.

Analoge Ein- und Ausgangssignale sind grundsätzlich im Bereich 0 ... 100% skaliert.

Setzt man bei dem obigen Beispiel also den Parameter PAR1 auf den Wert 50.0 und PAR2 auf 40.0, so erhält man einen Komparator mit Hysterese. Eine Spannung  $> 5\text{ V}$  an PIN 14 wird die rechte gelbe LED einschalten. Die LED leuchtet, bis die Spannung wieder unter 4 V gesunken ist.

### 4 Befehlsübersicht

Welche Funktionen stehen zur Verfügung? Hier soll zunächst ein Überblick gegeben werden. Eine detaillierte Beschreibung ist in Kapitel 7 zu finden.

Befehl:	Bedeutung:	Operand 1:	Operand 2:	Operand 3:
<b>Mathematik</b>				
DIR	direkte Zuweisung	Quelle	-	-
ADD	Addition	Summand 1	Summand 2	Summand 3 (optional)
SUB	Subtraktion	Minuend	Subtrahend	-
MUL	Multiplikation	Faktor 1	Faktor 2	Faktor 3 (optional)
DMUL	Multiplikation + Division	Faktor 1	Faktor 2	Divisor
LIM	Begrenzung	Eingangswert	Untere Grenze	Obere Grenze
SQRT	Wurzelfunktion	Eingangswert	-	-
SIN	Sinusfunktion	Eingangswert	-	-
ABS	Absolutwert	Eingangswert	-	-
NORM(L)	Normierung auf einen Bereich	Eingangswert	Stützpunkt X1	Stützpunkt X2
UNORM	Skalierung	Normierter Wert (u)	Stützpunkt Y1 (u=0)	Stützpunkt Y2 (u=1)
INTEG	Integrator	Eingangswert	Rücksetzen	Rücksetzwert (optional)
PT1	Tiefpass 1. Ordnung	Eingangswert	Zeitkonstante	Rücksetzen
MIN	Minimalwertauswahl	Wert 1	Wert 2	Wert 3 (optional)
MAX	Maximalwertauswahl	Wert 1	Wert 2	Wert 3 (optional)

<b>Logik</b>				
SEL	Umschaltung	Schalteingang (OP1)	Wert bei OP1 < 1	Wert bei OP1 >=1
GT	Vergleich: OP1 > OP2	Wert 1 (OP1)	Wert 2 (OP2)	-
LT	Vergleich: OP1 < OP2	Wert 1	Wert 2	-
GE	Vergleich: OP1 >= OP2	Wert 1	Wert 2	-
LE	Vergleich: OP1 <= OP2	Wert 1	Wert 2	-
AND	logisches "Und"	Wert 1	Wert 2	Wert 3 (ggf. "1" setzen)
OR	logisches "Oder"	Wert 1	Wert 2	Wert 3 (optional)
NOT	logische Negation	Eingangswert	-	-
RS	RS - Flipflop	Setzeingang	Rücksetzeingang	-
<b>Zeitfunktionen</b>				
RAMP	1 - Quadrantenrampe	Eingangswert	Rampenzeit	Rücksetzen
TE	Einschaltverzögerung	Eingangswert	Zeit	-
TA	Ausschaltverzögerung	Eingangswert	Zeit	Rücksetzen
FP	Flankenerkennung (steigt)	Eingangswert	-	-
FN	Flankenerkennung (fällt)	Eingangswert	-	-
FUR	Rechteckgenerator	Frequenz	Amplitude	Rücksetzen
FUS	Sinusgenerator	Frequenz	Amplitude	Rücksetzen
FUT	Dreieckgenerator	Frequenz	Amplitude	Rücksetzen
<b>Sonstiges (nicht bei jedem Gerät vorhanden)</b>				
FUN2	Zweitfunktionswert	-	-	-
SPAR	Parameter lesen / schreiben	Funktion auslösen	Index	Schreibwert (- für Lesen)

## 5 Eine praktische Übung

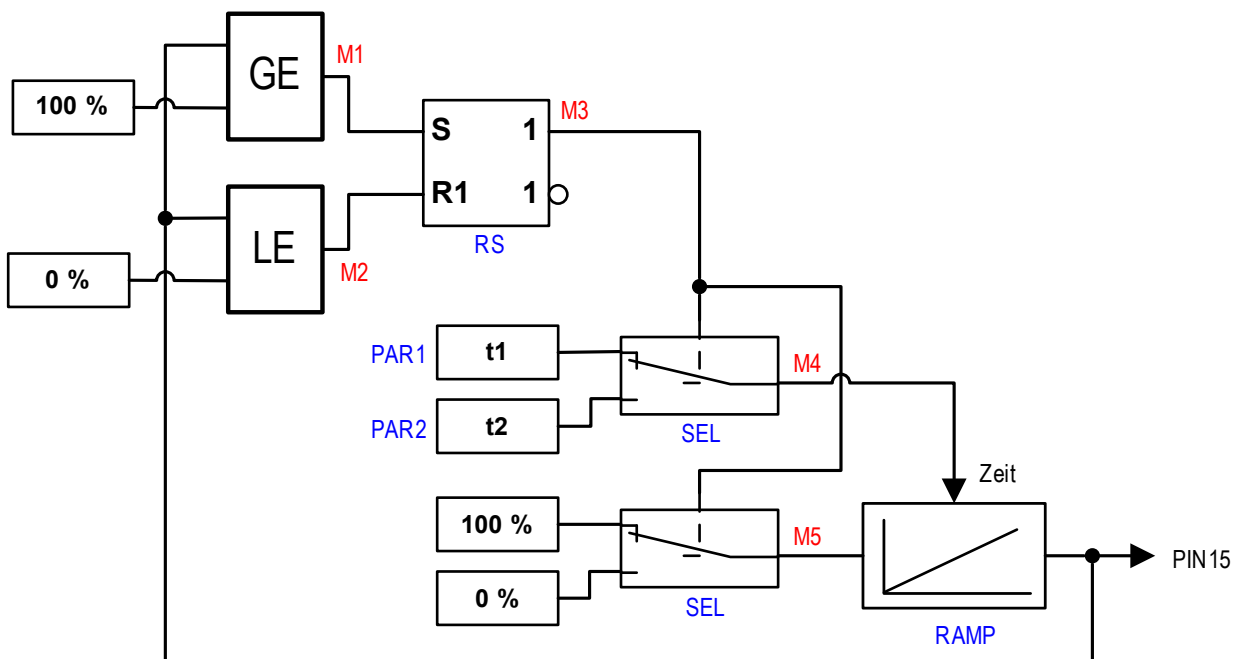
Dieses Beispiel können Sie direkt nachvollziehen. Das Programm „WESTScript“ stellen wir kostenlos zur Verfügung. Auch ohne angeschlossenes Gerät können Sie damit Ihre ersten Skripte ausprobieren.

Aufgabe: Ausgabe eines Sägezahnsignals an PIN15, die Spannung soll zyklisch von 0V bis 10V ansteigen. Die Zeiten des Anstiegs und Abfalls sollen einstellbar sein:



Nach einem Blick auf die verfügbaren Funktionen ist klar, dass der Rampengenerator ein guter Startpunkt wäre. Man schaltet sowohl sein Eingangssignal als auch seine Steigung um, wenn der obere bzw. untere Umkehrpunkt erreicht wird.

Eine mögliche Programmlogik hierfür sieht so aus:



Die blaue Schrift zeigt die Bezeichnungen der Elemente in der Skriptsprache, die rote eine mögliche Zuordnung der Speicherzellen.



## Übung:

Versuchen Sie nun das Funktionsschema der letzten Seite als Skript einzutragen.

Wählen Sie dazu für jede der zu verwendenden Zellen die richtige Funktion aus der Tabelle (S.3) aus und stellen Sie die Verbindungen durch Auswahl der Parameter her.

Im Folgenden sehen Sie eine Musterlösung.

## Musterlösung:

Skript

Online	Zeile	Funktion	Operand 1	Operand 2	Operand 3
	M1	GE	PIN15	100.0	
	M2	LE	PIN15	0.0	
	M3	RS	M1	M2	
	M4	SEL	M3	PAR1	PAR2
	M5	SEL	M3	100.0	0.0
	M6				
	M7				

... leere Zellen ...

	PIN15	RAMP	M5	M4	0.0
--	-------	------	----	----	-----

## Zu beachten:

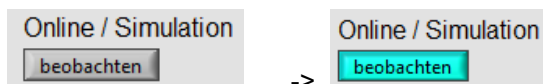
- Die festen Zahlen müssen exakt so geschrieben sein, wie hier zu sehen. Konstanten, die nicht im „Input“ Menü angeboten werden, müssen als Parameter angelegt werden.
- Die Rampenfunktion braucht einen dritten Operanden. Weil hier ein Rücksetzen nicht benötigt wird, wurde der Festwert „0.0“ = *False* eingetragen.
- Man kann die Kommandos und Parameter statt über das Kontextmenü durch eine direkte Texteingabe in die Zellen eingeben. Eine Syntaxüberprüfung erfolgt dann nicht.
- Die Zelle „PIN15“ ist gleichzeitig Ausgangssignal und Operand für andere Anweisungen.

Nach der Eingabe ist es zunächst sinnvoll, die Arbeit zu sichern.

Wählen Sie dazu den Menüpunkt Datei/Skript speichern. Skripte werden im CSV-Format gespeichert. Es handelt sich um eine einfache ASCII – Datei, die man auch mit einem Texteditor oder einer Tabellenkalkulation, z.B. EXCEL, lesen kann.

Nun ist aber die spannende Frage, ob das Programm auch funktioniert bzw. das gewünschte Signal erzeugt.

Um dies zu testen, wechselt man in den Simulationsmodus durch Betätigen der Schaltfläche

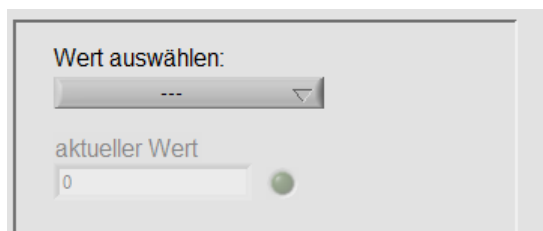


Es erscheinen zusätzliche Bedien- und Anzeigeelemente und in der Spalte *Online* sollten Sie „lebende“ Zahlen sehen. Die Funktion arbeitet aber noch nicht richtig, denn die Parameter PAR1 und PAR2 enthalten ja noch keine brauchbaren Zeiten. Der Voreinstellwert „0“ ergibt nur einen sehr schnellen Signalwechsel.

Zum Eintragen von Parameterwerten gibt es zwei Varianten:

- 1.) Sie klicken mit der Maus auf einen der PAR.. Einträge in der Skripttabelle. Es erscheint ein Dialogfenster mit der Anzeige des momentanen Parameterwertes. Sie können dann die Option „Ändern“ auswählen und einen neuen Wert eintragen. Diese Methode funktioniert übrigens auch im „Online“ – Betrieb am realen Modul.

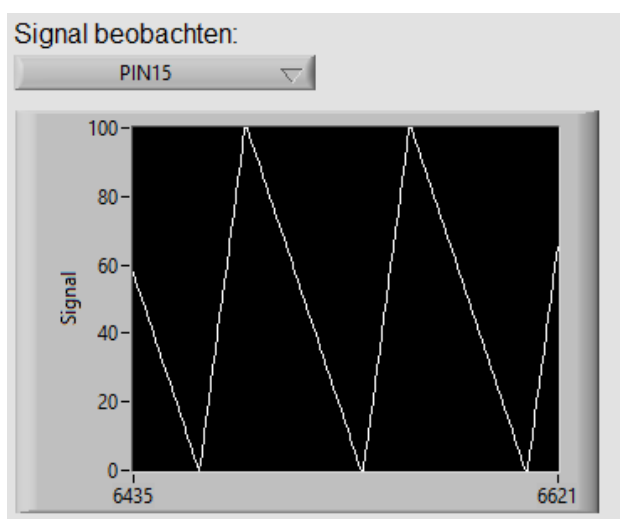
2.) Auf der rechten Seite sind diese Bedienelemente erschienen:



Über ein Pull down – Menü wählt man eine der hier simulierbaren Eingangsgrößen oder einen Parameter aus und hat sodann die Möglichkeit, einen Zahlenwert einzugeben. Um Schalteingänge bequemer simulieren zu können, gibt es das LED – Symbol rechts neben dem Zahlenfeld. Hier wird angezeigt, ob der aktuelle Wert einer ausgewählten Variablen  $\geq 1$  ist, also als „True“ interpretiert wird. Durch Mausklick auf dieses Feld kann man den Zustand schnell wechseln.

Nachdem Sie hier sinnvolle Werte, also z.B. PAR1 = 2 (s) und PAR2 = 5 (s) eingegeben haben, sieht man, dass sich der Zahlenwert in der Online – Spalte für PIN15 kleinschrittiger ändert und das Flipflop bei M3 im richtigen Takt schaltet.

Um einen besseren Eindruck vom Zeitverlauf zu bekommen, gibt es rechts oben ein kleines Signalverlaufsfenster. Wählen wir z.B. bei „Signal beobachten“ den PIN15 aus, erhalten wir eine mitlaufende Aufzeichnung



Es gibt hier einige Einstellmöglichkeiten über das Kontextmenü (rechte Maustaste).

Probieren Sie die Optionen „Autoskalierung X“ bzw. „Autoskalierung Y“ aus. Wenn man die automatische Skalierung in Y-Richtung abschaltet, können die Grenzen des dargestellten Bereichs auch nach Anklicken der Zahlenwerte der Achsskala (Minimum oder Maximum) direkt dort eingegeben werden.

Wechseln Sie das zu beobachtende Signal (z.B. von PIN15 auf M3), werden Sie feststellen, dass die Skalierung nicht mehr passt. Hier kann man z.B. wieder die Autoskalierung aktivieren.

Erneutes Betätigen der Schaltfläche Online/Beobachten stoppt die Simulation und die Elemente auf der rechten Fensterseite werden wieder ausgeblendet.

Wenn Sie nun ein wenig selber die Möglichkeiten der Skriptsprache erkunden möchten, wären dies Anregungen für Übungen:

- 1.) Erweitern Sie das Sägezahnskript so, dass die rechte LED anzeigt, wenn das Signal an PIN15 größer als 80% ist.
- 2.) Der Sägezahn soll auf „0“ zurückgesetzt werden, wenn ein Signal an Pin7 (Schalteingang) anliegt.
- 3.) Schreiben Sie ein neues Skript mit der folgenden Funktion: PIN 5 ist ein Schalteingang, PIN 6 ist ein Analogeingang. Man kann aber auch PIN6 als Schalteingang nutzen, indem man abfragt, ob das Signal > 50% ist. Programmieren Sie nun ein Skript, mit dem man den Ausgang an PIN16 mit einer parametrisierten Geschwindigkeit steigen lassen kann, wenn PIN5 betätigt wird und mit derselben Geschwindigkeit sinken lassen kann, wenn an PIN6 ein Signal > 50% anliegt. Dies entspricht der Funktion „Motorpotenziometer“.
- 4.) Stellen Sie sich vor, das Skript aus dem vorherigen Beispiel dient der Sollwertvorgabe in einem Positionsregler. Wenn Sie das System aktivieren, steht die Achse nicht unbedingt in der Position „0 mm“. Sie möchten also die augenblickliche Position einmalig als Sollwert übernehmen. Das soll geschehen, wenn der Eingang an PIN8 (Enable) eine positive Flanke liefert.

Ein Tipp noch, falls Sie Zeilen einfügen möchten oder den ganzen Block der Anweisungen verschieben wollen:

Öffnen Sie die CSV-Datei des Skriptes mit EXCEL o.Ä. und nutzen Sie die Möglichkeit, dort ganze Blöcke im Befehls- und Operandenbereich zu verschieben. Beachten Sie jedoch, dass die Spalte „Zeile“ dabei unverändert bleibt. Insbesondere das Löschen oder Einfügen kompletter Zeilen sollte man vermeiden.

Nach dem Durcharbeiten dieser Beispiele werden Sie sich schon recht gut in der Skriptprogrammierung zurechtfinden und vielleicht fallen Ihnen schon die ersten praktischen Anwendungen aus Ihrem Umfeld ein.

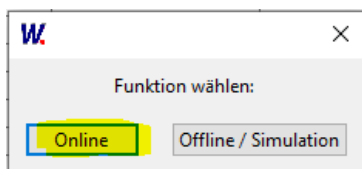
Im nächsten Kapitel soll es nun um den Schritt in die Praxis mit einem realen Modul gehen.

## 6 Online am Modul

### 6.1 Verbinden und Daten auslesen

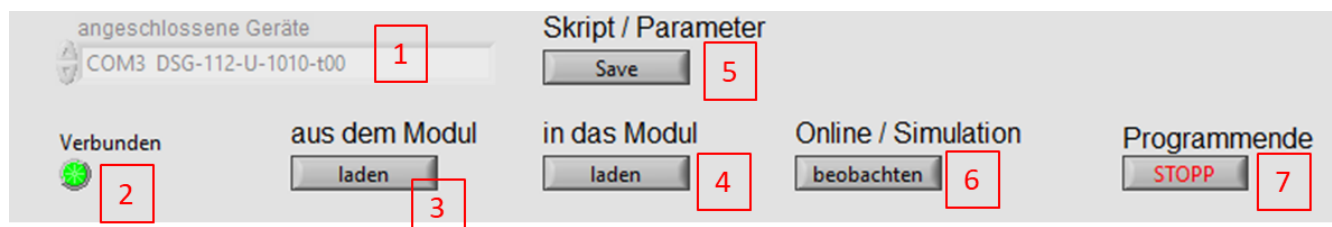
Zum Verbinden mit einem Modul muss eine USB – Verbindung bestehen, über die man sich auch per WPC – verbinden könnte, WPC darf aber nicht auf das betreffende Modul zugreifen.

Die Entscheidung, ob man online oder offline arbeiten möchte, kann nur bei Programmstart erfolgen:



Wählen Sie „Online“.

Es erscheinen neue Bedienelemente:

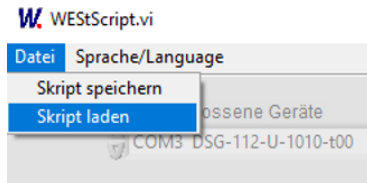


- 1.) Anzeige der aktiven Verbindung mit der Modulidentifikation. Falls mehrere potentielle Kandidaten (Geräte von W.E.St.) gefunden werden, muss man hier zunächst die gewünschte Verbindung auswählen. Gibt es nur einen möglichen Partner, wird die Verbindung sofort aufgebaut.
- 2.) Die grüne Anzeige bestätigt die erfolgreich aufgebaute Verbindung.
- 3.) Mit dieser Schaltfläche wird das momentane auf dem Modul befindliche Skriptprogramm in den Editor hochgeladen.
- 4.) Übertragung des Skripts aus dem Editor in das Modul. Achtung: Die Änderung ist unmittelbar wirksam.
- 5.) Diese Schaltfläche bewirkt, dass sowohl das Skript als auch die momentan eingestellten Parameter dauerhaft im EEPROM des Gerätes gespeichert werden. Es entspricht dem entsprechenden Knopf im WPC-Programm.
- 6.) Aktivierung des Beobachtungsmodus (s.u.)
- 7.) Das Programm soll nur über diese Schaltfläche beendet werden.

## 6.2 Offline erstelltes Skript laden oder Skript mit verbundenem Modul eingeben

Falls Sie eine Skriptdatei von Ihrem Rechner auf das Gerät übertragen möchten, läuft dies in mehreren Schritten ab:

1. Skript aus der Datei in den Editor laden:



Speichern: Sichert die angezeigte Skripttabelle in einer Datei

Laden: Holt das Skript aus einer Datei in die Tabelle

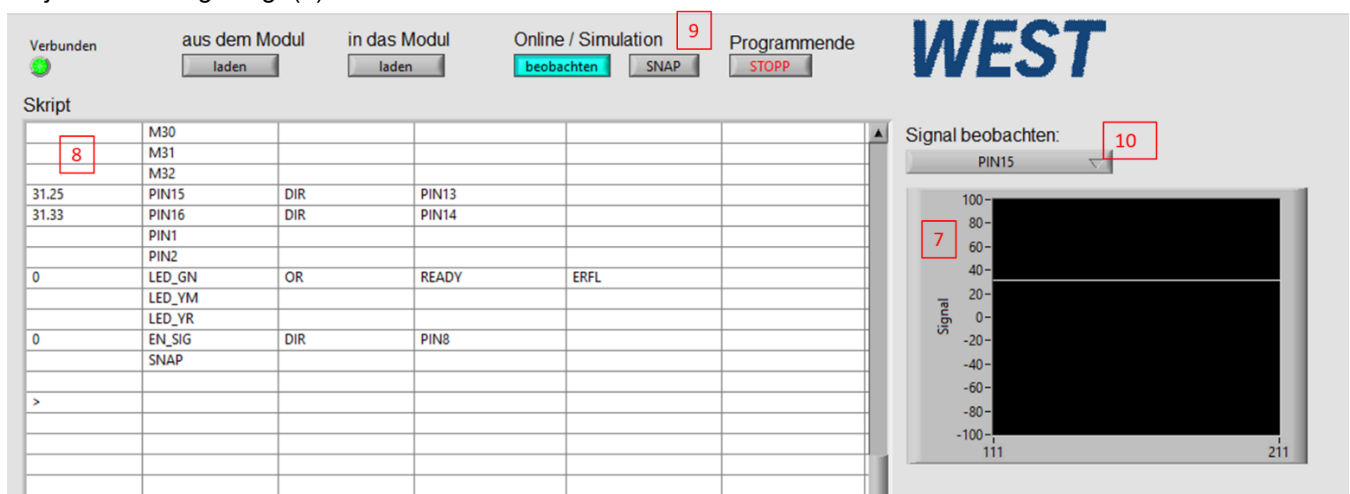
2. Mittels der Schaltfläche 4 (s.o.) übertragen Sie den Inhalt der Tabelle auf das Gerät. Sollten fehlerhafte Kommandos existieren, bricht die Übertragung an dieser Stelle ab.
3. Nach erfolgreicher Übertragung ist das geänderte Skript sofort aktiv. Sie können nun zunächst weitere Einstellungen (z.B. Parameter) vornehmen und die Funktion erproben. Vergessen Sie aber nicht das dauerhafte Speichern der Daten im nichtflüchtigen Speicher des Gerätes zum Abschluss der Aktivitäten über den Save Knopf in diesem Programm (5) oder WPC.

Direktes Editieren eines Skriptes im Online-Modus:

Nach dem Verbinden eines Moduls werden automatisch die Daten der Moduldefinition aus dem Gerät gelesen. Wie im Kapitel 5 beschrieben, kann man das Skript direkt in der angezeigten Tabelle ändern. Auch die Kontextmenüs können über einen Rechtsklick entsprechend aktiviert werden. Dies ist jedoch nur freigegeben, wenn kein Beobachtungsmodus aktiviert wurde (siehe folgender Abschnitt). Nach dem Ändern wird das Skript per Klick auf Taste 4 in das Modul geladen.

## 6.3 Beobachtungsmodus

Der Beobachtungsmodus dient der Inbetriebnahme und Überprüfung der Skriptfunktion. Wenn man diese Betriebsart über die Schaltfläche 6 aktiviert, werden in der Spalte „Online“ der Skripttabelle die aktuellen Werte für jede Zeile angezeigt (8):



The screenshot shows the WEST interface with the 'Online / Simulation' mode selected (indicated by a red box 9). The 'Skript' table is visible, with the 'Online' column highlighted (indicated by a red box 8). The 'Signal beobachten' window is open, showing a signal plot for PIN15 (indicated by a red box 10) with a value of 7 (indicated by a red box 7).

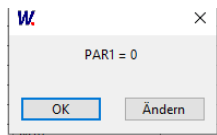
Skript	Online	Simulation	Programmierte
M30			
M31			
M32			
31.25 PIN15	DIR	PIN13	
31.33 PIN16	DIR	PIN14	
PIN1			
PIN2			
0 LED_GN	OR	READY	ERFL
LED_YM			
0 LED_YR			
EN_SIG	DIR	PIN8	
SNAP			
>			

Nochmaliges Betätigen der Taste 6 deaktiviert den Beobachtungsmodus.

Besondere Funktion, die im Beobachtungsmodus möglich sind:

## Parameteranzeige und Änderung

Beim (links)-Klick auf einen freien Parameter „PAR..“ in der Tabelle erscheint ein Dialogfenster, in dem der aktuelle Wert angezeigt wird und die Möglichkeit zur Änderung angeboten wird:



Beachten Sie bitte, dass Anzeige und Eingabe im Gleitpunktformat erfolgen, während WPC denselben Wert mit künstlich verschobenem Komma zeigt. Beispiel: Ein Wert von 1,23 würde hier ebenso dargestellt, im WPC jedoch als „123“.

## Signalschreiber

Im Beobachtungsmodus wird ein Streifenschreiber (7) sichtbar, in dem man den zeitlichen Verlauf eines der Signale darstellen kann. Hierzu wählt man über das Pull-Down Menü 10 ein interessierendes Signal aus. Die Skalierung der Y-Achse kann durch Rechtsklick auf deren Skala geändert werden: Autoskalierung deaktivieren, danach ist eine Änderung der Unter- und Obergrenze im Diagramm möglich, indem man direkt auf den Wert klickt und dort eine Zahl eingibt.

Der Signalschreiber an dieser Stelle ist als Hilfsmittel zur schnellen Beurteilung einzelner Signale gedacht. Möchte man mehrere Signale aufzeichnen, das Ergebnis speichern, usw. ist die Oszilloskopfunktion im WPC ein viel umfangreicheres und komfortableres Werkzeug.

## Schnappschuss

Möchte man die Situation bei sporadischen Vorgängen nachvollziehen, ist es hilfreich, wenn man eine Kopie der Online-Werte zum betreffenden Zeitpunkt anlegen kann. Hierfür gibt es eine spezielle Speicherzelle „SNAP“. Wenn der Wert dieser Variablen  $\geq 1,0$  wird, wird zu diesem Zeitpunkt eine Momentaufnahme der Online-Werte gespeichert. Diese Momentaufnahme kann man sich ansehen, wenn man den Schalter „SNAP“ (9) betätigt. Die Momentaufnahme wird bei jeder ansteigenden Flanke der Variablen „SNAP“ in der Tabelle überschrieben. Möchte man nur einen Zustand speichern, kann man z.B. die Funktion RS eintragen und nur den Setzeingang verbinden. Wählt man die Schnappschussansicht und findet nur Nullen in der Online-Spalte (inkl. der SNAP-Zeile) bedeutet dies, dass nach Start des Gerätes noch gar keine Aufnahme ausgelöst wurde.

## 7 Funktionsreferenz

### 7.1 Mathematische Funktionen

#### 7.1.1 DIR

Als Wert der Speicherzelle wird Operand 1 übernommen.

#### 7.1.2 ADD

Ergebnis = Operand 1 + Operand 2 + Operand 3

Die Angabe des 3. Operanden ist optional, d.h. wird hier nichts angegeben, so werden nur die ersten beiden Operanden addiert.

#### 7.1.3 SUB

Ergebnis = Operand 1 - Operand 2

#### 7.1.4 MUL

Ergebnis = Operand 1 \* Operand 2 \* Operand 3

Die Angabe des 3. Operanden ist optional, d.h. wird hier nichts angegeben, so werden nur die ersten beiden Operanden multipliziert.

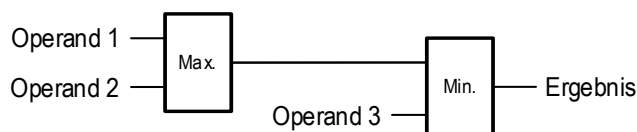
#### 7.1.5 DMUL

Ergebnis = Operand 1 \* Operand 2 / Operand 3

Möchte man nur eine Division durchführen, so ist Operand 2 = 1.0 einzustellen.

Sollte der Operand 3 den Wert „0,0“ besitzen, ist das Ergebnis der Division undefiniert. Um numerische Probleme in nachfolgenden Funktionen zu vermeiden, wird der Ausgang in diesem Fall ebenfalls auf „0,0“ gesetzt.

#### 7.1.6 LIM



Der bei Operand 1 angegebene Wert wird auf einen Bereich zwischen einer unteren Grenze (Operand 2) und einer oberen Grenze (Operand 3) beschränkt. Sollte man Operand 3  $\leq$  Operand 2 angeben, entspricht das Ergebnis Operand 3, bedingt durch die oben dargestellte Reihenfolge der Vergleiche.

#### 7.1.7 SQRT

Wurzelfunktion des Eingangssignals Bei negativem Eingangssignal wird dieses invertiert, die Wurzel gebildet und das Ergebnis wieder invertiert. Auf diese Weise wird die Wurzelfunktion in den negativ - reellen Bereich punktsymmetrisch fortgesetzt. Diese Art der Berechnung ermöglicht z.B. den sinnvollen Einsatz zur Durchflussberechnung anhand eines Differenzdruckes.

## 7.1.8 SIN

Sinusfunktion, Eingangssignal ist der Winkel im Bogenmaß [rad], ein Wert von 2 Pi entspricht 360°.

## 7.1.9 ABS

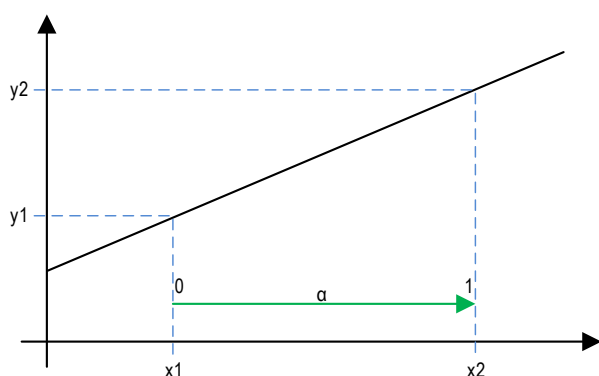
Am Ausgang wird der Absolutwert des Operanden 1 ausgegeben.

## 7.1.10 NORM / NORML / UNORM

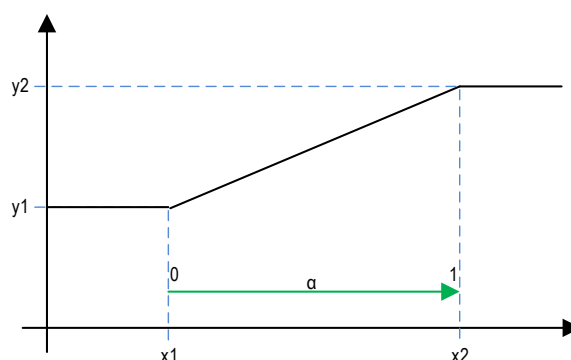
Funktionen zum Linearisieren und Umskalieren.

Es kommt häufig vor, dass man Werte linear umrechnen möchte. Messsignale werden als Normsignale übertragen und müssen in eine physikalische Einheit umgerechnet werden, Einheiten müssen umgerechnet werden, Kennlinien müssen durch mehrere lineare Abschnitte angenähert werden usw. Diese Operationen oder Teile davon lassen sich in der Regel durch eine lineare Gleichung der Form  $y = m \cdot x + b$  beschreiben und sind somit relativ einfach durch die elementaren Funktionen Addition und Multiplikation zu realisieren.

Etwas unpraktisch wird es, wenn statt der Steigung  $m$  und dem  $y$ -Achsenabschnitt  $b$  zwei Wertepaare  $x_1/y_1$  und  $x_2/y_2$  gegeben sind. In diesem Fall müssen  $m$  und  $b$  zunächst aus diesen berechnet werden.



Linearisierung ohne Begrenzung (mit Extrapolation)



Reine Interpolation (Begrenzung an den Stützpunkten)

Ein anderer Ansatz für eine solche Aufgabe besteht darin, zunächst den Bereich zwischen den Stützpunkten in eine normierte Hilfsvariable umzuwandeln und diese in einem zweiten Schritt in die Zielvariable umzuwandeln.

Die obigen Skizzen zeigen, wie dies gemacht wird:

Mit  $\alpha = \frac{u-x_1}{x_2-x_1}$  als Zwischengröße wird das Eingangssignal  $u$  zunächst in einen normierten Wert  $\alpha$  umgewandelt ( $0 - 1$  für  $x_1 \dots x_2$ ), dann kann der Ausgangswert  $Y$  daraus gebildet werden:  $y = \alpha \cdot (y_2 - y_1) + y_1$ .

Die Funktionen NORM und NORML sorgen für den ersten Schritt, d.h. die Umrechnung in den normierten Zwischenwert. Die Funktion NORML begrenzt das Ergebnis auf den Wertebereich  $0,0 - 1,0$ , so dass in Kombination mit dem zweiten Schritt eine reine Interpolation stattfindet und der durch die Stützstellen definierte Bereich nicht über- oder unterschritten wird.

Dieser zweite Schritt wird von der Funktion UNORM durchgeführt.

### Ein praktisches Beispiel:

Ein Drucksensor mit einem Messbereich von 100 bar an PIN6 eines Moduls misst den Druck in einem (Plunger-)Zylinder, mit dem eine Arbeitsplattform angehoben wird. Wenn die Plattform unbelastet ist, wird ein Druck von 32 bar (= PAR1) gemessen; wenn die Plattform mit 8000 kg Gewicht (=PAR3) belastet ist, beträgt der Druck 87 bar (=PAR2).

Um die zusätzliche Belastung während des Betriebs zu ermitteln, schreiben Sie:

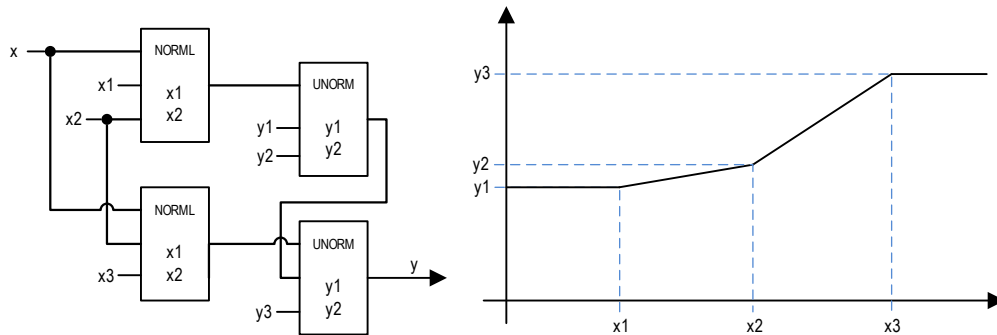
M1 = NORM PIN6 PAR1 PAR2

(Bei einem Messbereich von 100 bar entspricht das Signal PIN6 (%) dem Druck in bar)

M2 = UNORM M1 0.0 PAR3

## Weitere Anwendungsmöglichkeit:

Interpolation mit mehreren Stützpunkten

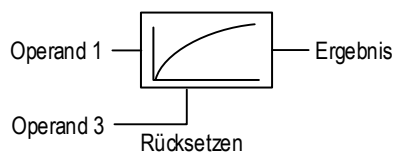


### 7.1.11 INTEG

Integrator: Das Signal an Operand 1 wird aufintegriert (Standardsteigung 1/s, d.h. der Ausgang steigt um diesen Wert pro Sekunde). Operand 2 dient zum Zurücksetzen, solange dieser Wert  $\geq 1$  ist, wird der Ausgang mit dem Signal des Operanden 3 beaufschlagt (0,0 wenn unverbunden). Nach Wegnahme des Reset läuft der Integrator von diesem Wert aus weiter.

### 7.1.12 PT1

Zeitdiskrete Implementierung eines Verzögerungselements 1. Ordnung. Diese Funktion kann man zum Filtern von Signalen verwenden.



Der Operand 2 gibt die Zeitkonstante in Sekunden an.

Die Zeitkonstante wird intern auf 0,01 s begrenzt, falls man kleinere Werte angibt, wird das Eingangssignal unverzögert an den Ausgang weitergegeben. Gleiches geschieht, wenn der Operand 3 (Rücksetzen)  $\geq 1,0$  ist.

### 7.1.13 MIN / MAX

Auswahl des minimalen oder maximalen Wertes der Eingangssignale. Der kleinste oder größte der drei Operanden wird an den Ausgang weitergeleitet.

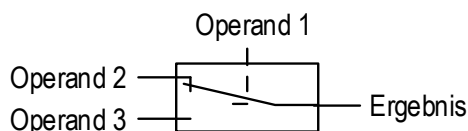
Der dritte Operand ist optional, d.h. wenn hier nichts angegeben wird, werden nur die ersten beiden Operanden verarbeitet.

## 7.2 Logik

Hier gilt die Festlegung, dass Operanden als Eingangswerte als WAHR interpretiert werden, falls sie  $\geq 1,0$  sind.

Als Ergebnis wird entweder 1,0 oder 0,0 ausgegeben. Dies kann man als Operator für andere Logikbefehle nutzen oder auch als Zahlenwert verwenden.

### 7.2.1 SEL



Nur Operand 1 wird als Bool'sche Variable aufgefasst. Der Inhalt von Operand 2 oder 3 wird unverändert als Ergebnis übernommen.

### 7.2.2 GT / GE / LT / LE

GT: Ergebnis = Operand 1 > Operand 2

GE: Ergebnis = Operand 1  $\geq$  Operand 2

LT: Ergebnis = Operand 1 < Operand 2

LE: Ergebnis = Operand 1  $\leq$  Operand 2

### 7.2.3 AND / OR / NOT

Logische Standardverknüpfungen Boole'scher Variablen (nicht Bitmuster).

Die AND – Verknüpfung erfordert immer die Angabe dreier Operanden. Sollen nur zwei verknüpft werden, ist der dritte fest auf „1,0“ zu setzen.

OR: Die Angabe des 3. Operanden ist optional, d.h. wird hier nichts angegeben, so werden nur die ersten beiden Operanden verknüpft.

### 7.2.4 RS

Wahrheitstabelle: (0 steht hier für *false* / 1 für *true*)

Alter Ausgangswert	Operand 1 (S)	Operand 2 (R)	Ergebnis
0	0	0	0
1	0	0	1
0 oder 1	1	0	1
0 oder 1	0	1	0
0 oder 1	1	1	0

## 7.3 Zeitfunktionen

Übersicht über diese Funktionen und deren Operanden:

Zeitfunktionen				
Befehl:	Bedeutung:	Operand 1:	Operand 2:	Operand 3:
RAMP	1 - Quadrantenrampe	Eingangswert	Rampenzeit	Rücksetzen
TE	Einschaltverzögerung	Eingangswert	Zeit	-
TA	Ausschaltverzögerung	Eingangswert	Zeit	Rücksetzen
FP	Flankenerkennung (steigt)	Eingangswert	-	-
FN	Flankenerkennung (fällt)	Eingangswert	-	-
FUR	Rechteckgenerator	Frequenz	Amplitude	Rücksetzen
FUS	Sinusgenerator	Frequenz	Amplitude	Rücksetzen
FUT	Dreieckgenerator	Frequenz	Amplitude	Rücksetzen

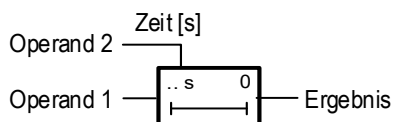
### 7.3.1 RAMP

Diese Funktion führt das Ergebnis dem Wert von Operand 1 nach, jedoch mit einer maximalen Geschwindigkeit, die mit Operand 2 bestimmt wird. Die Angabe erfolgt dabei in 100/s, d.h. bei einer sprunghaften Änderung des Eingangswertes um 100 (%) wird der Ausgang innerhalb der durch Operand 2 in Sekunden angegebenen Zeit die Änderung nachvollziehen.

Falls der Operand 3 verbunden und „true“ ist, wird der Eingangswert an Operand 1 direkt ohne Verzögerung als Ergebnis übernommen. Möchte man den Ausgang hierdurch zu „0.0“ setzen, ist also gleichzeitig eine „0.0“ durch Operand 1 vorzugeben.

### 7.3.2 TE

Standard – Einschaltverzögerung:

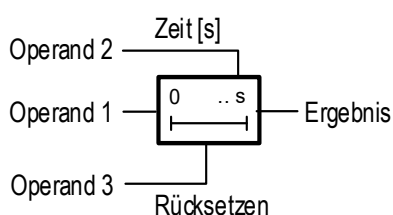


Der Ausgang schaltet auf „1“, wenn der Eingang (Operand 1) für mindestens die Zeit, welche mit Operand 2 vorgegeben wird, ununterbrochen „1“ war. Sollte sich Operand 2 während der Laufzeit ändern, verlängert oder verkürzt sich die Zeit entsprechend. Sollte die Änderung nach dem Ablauf der Verzögerungszeit stattfinden, hat dies keine Auswirkung auf das Ergebnis, der Ausgang bleibt gesetzt.

Bei einem Signalwechsel auf „0“ am Eingang wird das Ergebnis am Ausgang sofort ebenfalls zu „0“.

### 7.3.3 TA

Standard – Ausschaltverzögerung mit Erweiterung (Rücksetzeingang):



Der Ausgang schaltet sofort auf „1“, wenn der Eingang (Operand 1) gesetzt wird.

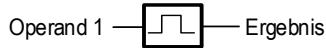
Wenn der Operand 1 danach für mindestens die Zeit, welche mit Operand 2 vorgegeben wird, ununterbrochen „0“ war, wird der Ausgang ebenfalls zurückgesetzt. Sollte sich Operand 2 während der Laufzeit ändern, verlängert oder verkürzt sich die Zeit entsprechend.

Ein sofortiges Rücksetzen von Ablauf der Verzögerung kann durch Operand 3 erreicht werden. Sollte Operand 1 und Operand 3 gleichzeitig „1“ sein, hat Operand 1 Vorrang und das Ergebnis am Ausgang bleibt „1“.

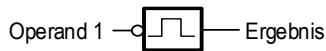
### 7.3.4 FP / FN

Flankenerkennung des Eingangssignals „Operand 1“:

FP: Eine steigende Flanke am Eingang setzt den Ausgang für einen Verarbeitungszyklus auf „1“.



FN: Eine fallende Flanke am Eingang setzt den Ausgang für einen Verarbeitungszyklus auf „1“.



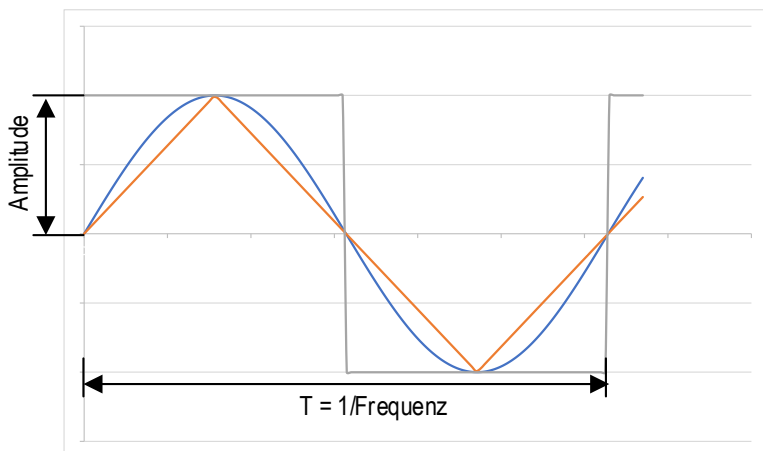
### 7.3.5 FUR / FUS / FUT

Diese drei Funktionen stellen Funktionsgeneratoren dar.

FUR = **F**unktionsgenerator **R**echteck

FUS = **F**unktionsgenerator **S**inus

FUT = **F**unktionsgenerator **D**reieck (**T**riangle)



Befehl:	Bedeutung:	Operand 1:	Operand 2:	Operand 3:
FUR	Rechteckgenerator	Frequenz	Amplitude	Rücksetzen
FUS	Sinusgenerator	Frequenz	Amplitude	Rücksetzen
FUT	Dreieckgenerator	Frequenz	Amplitude	Rücksetzen

Nach dem Rücksetzen laufen die Generatoren wie oben gezeigt los, d.h. Sinus und Dreieck beginnen mit dem Anstieg von „0“, das Rechtecksignal mit der positiven Hälfte.

## 7.4 Sonstige Funktionen

### 7.4.1 FUN2

Einige Funktionen können zwei Rückgabewerte liefern. Der Zugriff auf den zweiten Wert erfolgt dabei durch diese Funktion, die sich immer auf die auf die Funktion in der vorangehenden Zeile bezieht.

TE	bereits verstrichene Zeit nach dem Aktivieren [s], bleibt beim Erreichen der eingestellten Verzögerungszeit stehen
TA	verbleibende Zeit bis zur Abschaltung in [s], ist gleich der Verzögerungszeit bei aktivem Eingangssignal
FUR/FUS/FUT	Sägezahnsignal von -1.0 bis 1.0
SPAR	Schreiben / Lesen abgeschlossen (asynchrone Verarbeitung!)

Weitere Verwendung bei modulspezifischen Komplexfunktionen möglich, siehe zugehörige Dokumentation.

### 7.4.2 SPAR

Skriptgesteuertes Lesen oder Schreiben von Parametern.

Das Vorhandensein einer Verschaltung zu dem dritten Parameter bestimmt, ob gelesen oder geschrieben wird.

Mit dieser Funktion ist es möglich, die Parameter der modulinternen Komplexfunktionen oder fest programmierter Bestandteile der Firmware aus dem Skript zu lesen oder zu beschreiben. Auf diese Weise kann man z.B. elegant eine Parameterumschaltung oder einen Abgleich der Nullposition während des Betriebs realisieren.

Eine steigende Flanke an Operand 1 löst die Funktion aus. Ein Dauersignal bewirkt lediglich, dass der Rückgabewert gehalten wird, ein neues Lesen oder Schreiben erfordert einen Signalwechsel.

Wenn kein dritter Operand vorhanden ist, liest die Funktion den Parameter, der über einen Index (Operand 2) zugeordnet wird. Die gültigen Indizes sind der jeweiligen Moduldokumentation zu entnehmen.

Der Rückgabewert beim Lesen entspricht dem eingestellten Parameter in dem Format, in dem auch die Eingabe in der Parametertabelle geschieht (z.B. 0,0 - 10000,0 für 0 – 100%). Sollte der Index nicht vorhanden sein, liefert die Funktion den Wert -1,0 zurück.

Zum Schreiben ist der dritte Parameter zu verbinden, der Wert ist im selben Format vorzugeben. Der Rückgabewert beim Schreiben gibt an, ob die Funktion erfolgreich abgeschlossen wurde (= 1,0). Werte außerhalb der Grenzen für den betreffenden Parameter werden nicht übernommen.

Beachten Sie bitte, dass dieser Befehl asynchron abläuft und nicht in jedem Fall im selben Zyklus abgeschlossen ist, in dem er angestoßen wurde.

Anwendungsbeispiel: Umschalten eines Parameters in Abhängigkeit von einem Schalteingang

M1 = SPAR PIN5 PAR11 PAR1

M2 = NOT PIN5

M3 = SPAR M2 PAR11 PAR2

PAR11 enthält den Parameterindex, PAR1 und PAR2 die beiden Werte, zwischen denen umgeschaltet werden soll.

Da die SPAR – Funktion bereits intern eine Flankenerkennung durchführt, kann man als Parameter 1 direkt den Eingang bzw. den negierten Eingang verbinden.

## **7.5 Modulspezifische Komplexfunktionen**

Über die hier beschriebenen Funktionen hinaus gibt es je nach Modul eine unterschiedliche Anzahl von Funktionen, die eine umfangreichere Berechnung mit nur einem Kommando durchführen und die über einen eigenen Parametervorrat verfügen. Die können Regler, Rampen, Kennlinienbausteine usw. sein.

Diese Funktionen werden in der jeweiligen Moduldokumentation beschrieben.

Eine Offline-Simulation dieser Funktionen ist leider nicht möglich.

## **8 Anwendungsbeispiele**

Im Folgende sollen exemplarisch einige typische und einige anspruchsvollere Anwendungen gezeigt werden, als Anregung und Vorlage für eigene Projekte. Die zu den Beispielen gehörenden Skripte stellen wir Ihnen gerne als Dateien zur Verfügung. Auch helfen wir gerne bei der Realisierung Ihrer Anwendung, sprechen Sie uns gerne an.

### **8.1 Oszillierender Zylinderantrieb**

#### **8.1.1 Aufgabe**

Ein Zylinder soll zyklisch zwischen zwei Endlagen hin- und herfahren. Dabei kommt es nicht auf eine exakte Positionierung an. Die Geschwindigkeit soll in den beiden Bewegungsrichtungen unterschiedlich eingestellt werden können. Es werden hier zwei Varianten untersucht, nämlich mit Nutzung eines analogen Rückführsignals (Wegaufnehmer) oder bei Verwendung von Schaltsignalen, z.B. aus induktiven Näherungsinitiatoren.

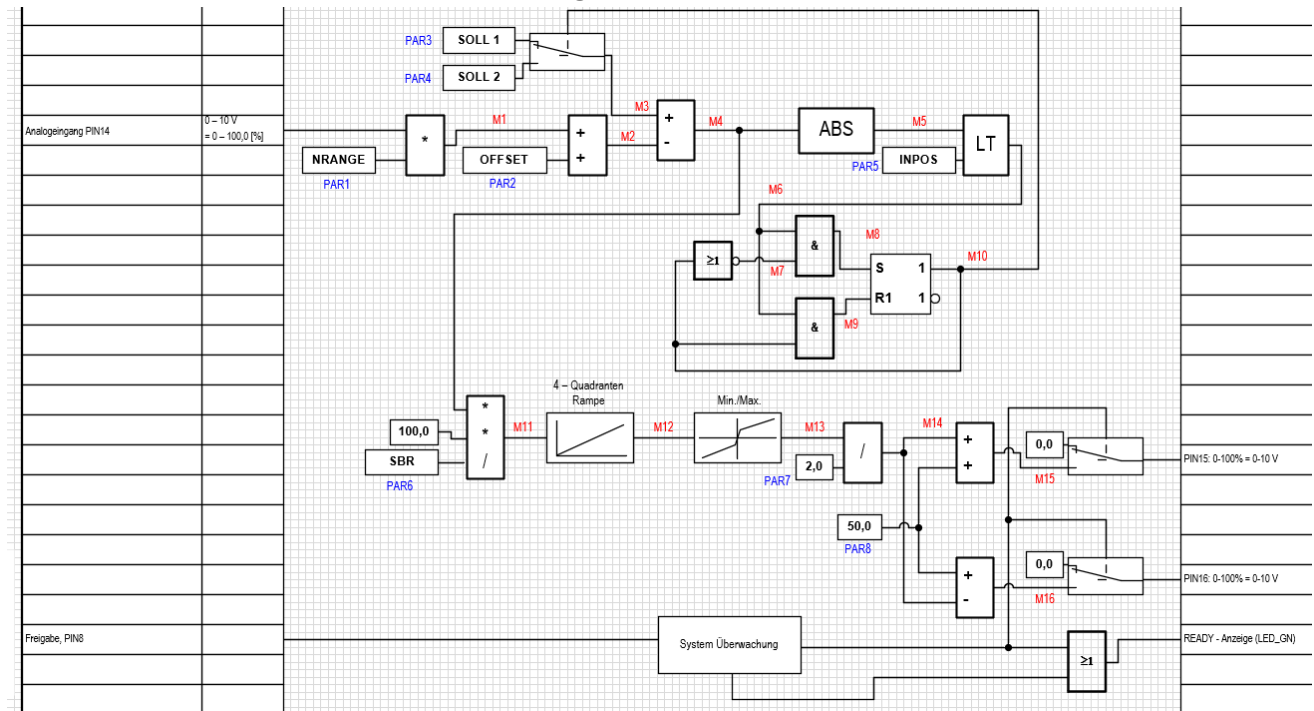


## 8.1.3 Lösung 2

Verwendung eines analogen Sensors.

Vorteile: Bei richtiger Einstellung kein Überfahren der Endposition. Definierter Verzögerungsvorgang ohne Stöße und klarem Haltepunkt.

### 8.1.3.1 Funktionsplan und Beschreibung



Diese Aufgabe erfordert schon ein relativ umfangreiches Skript.

Der untere Teil ab der Rampe ist identisch zum vorherigen Beispiel.

Beginnend vom Eingang des Wegsensors kann die Funktion folgendermaßen beschrieben werden: Zunächst wird das Signal durch Multiplikation mit der Messlänge und Addition eines Offsetwertes auf einen physikalischen Wert in mm skaliert (M2). Es erfolgt eine Differenzbildung zu einem Sollwert (M3), der über einen Schalter ausgewählt wird. Verfolgen wir zunächst die Erzeugung des Schaltsignals (M10). Das Signal kommt aus einem Flipflop, das über einen Puls an M6 umgeschaltet wird. Dieser Impuls entsteht, wenn der Absolutwert der Regelabweichung (M5) kleiner wird als ein eingestellte „INPOS“-Abweichung, als die geforderte Annäherung an den Umschaltpunkt. Weil direkt im nächsten Zyklus der Sollwert stark verändert sein wird, ist die Abweichung nur für einen Verarbeitungsschritt unter der Schaltschwelle.

Die Regelabweichung wird durch einen einstellbaren Bremsweg (PAR6) geteilt, hiermit bestimmt man die Länge des Bremsvorgangs bzw. die Größe der Verzögerung. Wenn die Regelabweichung genauso groß ist, wie der Bremsweg, wird der Ausgang (M10) den Wert 100 (%) annehmen. Ist er kleiner, wird das Ventil wegabhängig geschlossen. Falls M11 größer als 100% ist, erfolgt ohne weitere Maßnahmen eine Begrenzung auf +/- 100% durch die Weitergabe an die Komplexfunktion der 4-Quadrantenrampe. Diese wird verwendet, um eine definierte Beschleunigung zu erhalten. Zu beachten ist, dass die Zeiten im 2. und 4. Quadranten auf den Minimalwert eingestellt werden sollten, denn das Bremsen geschieht hier weg- und nicht zeitabhängig.

## 8.1.3.2 Skript

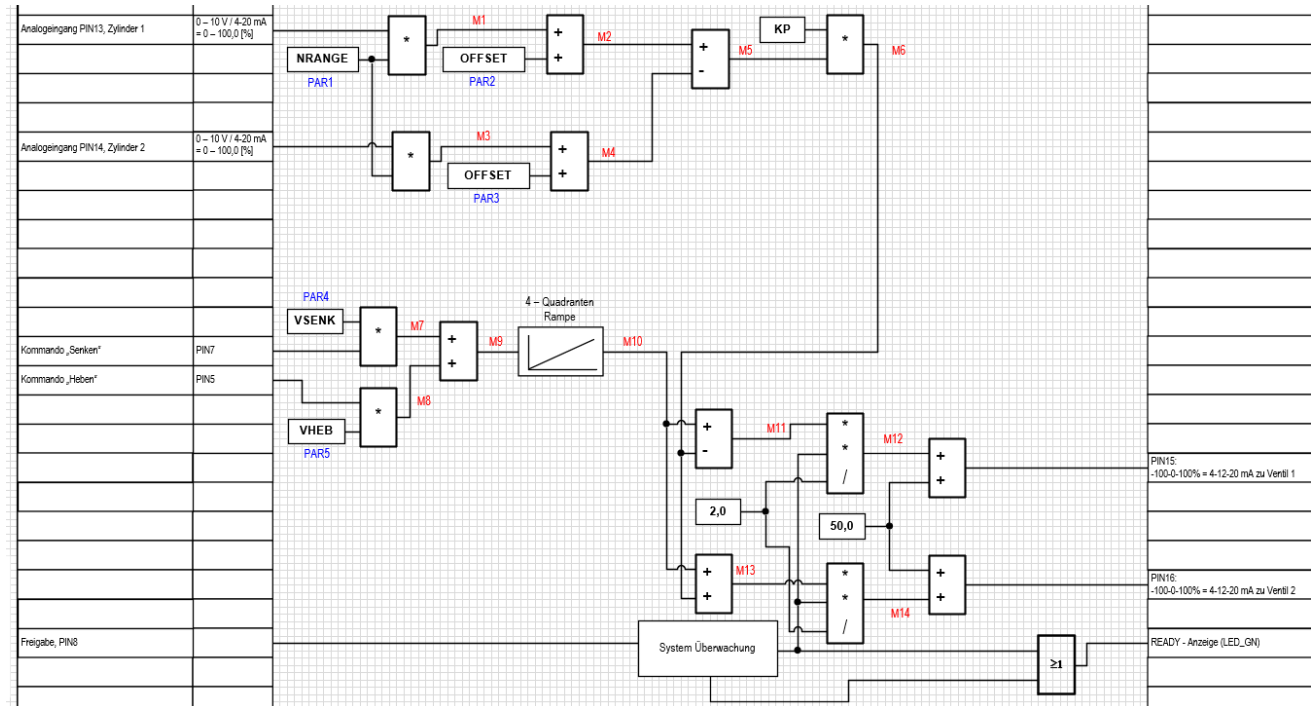
M1	MUL	PIN14	PAR1	-
M2	ADD	M1	PAR2	-
M3	SEL	M10	PAR3	PAR4
M4	SUB	M3	M2	-
M5	ABS	M4	-	-
M6	LT	M5	PAR5	-
M7	NOT	M10	-	-
M8	AND	M6	M7	-
M9	AND	M6	M10	-
M10	RS	M8	M9	-
M11	DMUL	M4	100.0	PAR6
M12	RAMP4Q	M11	-	-
M13	MINMAX1	M12	-	-
M14	DMUL	M13	1.0	PAR7
M15	ADD	M14	PAR8	-
M16	SUB	PAR8	M14	-
PIN15	SEL	READY	0.0	M14
PIN16	SEL	READY	0.0	M15
LED_GN	OR	READY	ERFL	-
EN_SIG	DIR	PIN8	-	-

## 8.2 Gleichlauf von 2 Zylindern (Vorgabe der Bewegungsrichtung über Schalteingänge)

### 8.2.1 Aufgabe

Bei diesem Beispiel soll eine denkbar einfache Gleichlaufregelung zweier Zylinder realisiert werden. Als Eingangssignale werden lediglich Schaltsignale (auf/ab) und die aktuelle Position beider Zylinder verarbeitet. Ausgangsseitig ist die Ansteuerung zweier Proportionalventile über Ausgänge 4-12-20 mA vorgesehen.

### 8.2.2 Lösung



Im oberen Teil des Plans wird zunächst eine Skalierung der Eingangssignale vorgenommen, die in M2 und M4 in physikalischer Einheit, z.B. mm vorliegen. Es erfolgt eine Differenzbildung zur Regelabweichung M5. Die Proportionalverstärkung KP bestimmt den Einfluss des Gleichlaufanteils.

Zur Bildung des eigentlichen Stellsignals bzw. dessen Mittelwertes (M10) wird ausgenutzt, dass digitale Signale wie Zahlen in eine Multiplikation verwendet werden können. Der Zahlenwert für VSENK (PAR4) muss negativ sein. Eine Rampe sorgt für stoßfreies Anfahren und Abbremsen. Der Gleichlaufanteil wird für Zylinder 1 vom mittleren Stellsignal abgezogen und für Zylinder 2 hinzuaddiert. Die bipolaren Stellsignale (M11 / 13) werden anschließend auf den Wertebereich der Ausgangssignale umskaliert. Wieder wird eine Multiplikation genutzt, um das READY - Signal der Fehlerüberwachung zum Abschalten der Bewegung einzuschleusen.

## 8.2.3 Skript

M1	MUL	PIN13	PAR1	-
M2	ADD	M1	PAR2	-
M3	MUL	PIN14	PAR1	-
M4	ADD	M3	PAR3	-
M5	SUB	M2	M4	-
M6	MUL	PAR6	M5	-
M7	MUL	PAR4	PIN7	-
M8	MUL	PIN5	PAR8	-
M9	ADD	M7	M8	-
M10	RAMP4Q	M9	-	-
M11	SUB	M10	M6	-
M12	DMUL	M11	READY	2.0
M13	ADD	M10	M6	-
M14	DMUL	M13	READY	2.0
PIN15	ADD	M12	50.0	-
PIN16	ADD	M14	50.0	-
LED_GN	OR	READY	ERFL	-
EN_SIG	DIR	PIN8	-	-

## 8.3 Aufteilung eines Sollwertsignals auf 2 Verstellpumpen

### 8.3.1 Aufgabe

Man möchte zwei Pumpen mit unterschiedlichem Fördervolumen parallel fördern lassen. Extern wird die Entscheidung getroffen, ob beide Pumpen oder nur eine Pumpe fördern soll, das Modul überwacht anhand des momentanen Sollwertsignals, ob beide Pumpen laufen müssen. Die beiden Pumpen haben einen unterschiedlichen Wirkungsgrad, so dass Pumpe 1 die Grundlast weitestgehend übernehmen soll und Pumpe 2 erst ab einer gewissen Anforderung eine größere Fördermenge liefern soll.

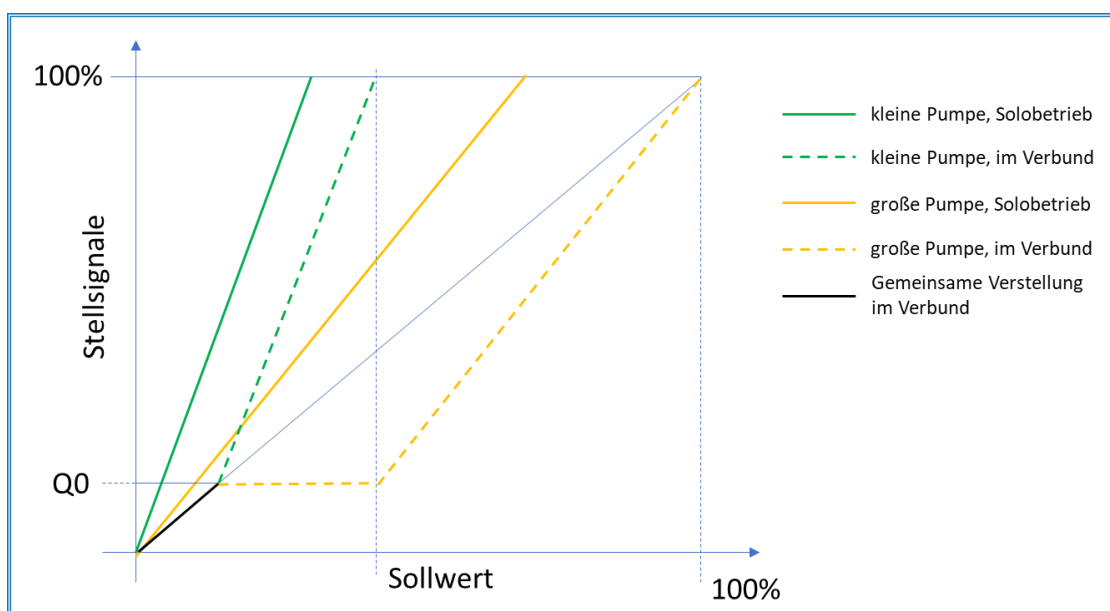
### 8.3.2 Konzept

Egal, ob nur eine Pumpe arbeitet oder beide Pumpen im Verbund, soll das Sollwert – Eingangssignal in das Modul immer denselben resultierenden Gesamtvolumenstrom ergeben. Mit anderen Worten: Die Steigerung des Volumenstroms pro % - Signalanstieg am Eingang soll immer gleich sein. Wenn nur eine Pumpe läuft, wird also schon vor dem Erreichen von 100% Sollwert die Grenze erreicht sein, an der die Fördermenge nicht weiter steigen kann und das Stellsignal an diese Pumpe 100% erreicht. Damit der Anwender rechtzeitig reagieren kann, soll bei 90% Stellsignal ein DO dies signalisieren, so dass eine zweite Pumpe manuell oder automatisch zugeschaltet werden kann.

Das Diagramm auf der nächsten Seite zeigt das zu realisierende Verstellgesetz.

Die durchgezogenen Linien zeigen die Ausgangssignale bei Einzelbetrieb. Bedingt durch die unterschiedlichen Fördermengen sind die Steigungen unterschiedlich und die Fördergrenze (100% Stellsignal) werden bei unterschiedlichen Sollwerten erreicht.

Wenn beide Pumpen laufen, sollen diese bis zu einer Mindestmenge (Q1), z.B. 10 %, zunächst im Verbund aufgeschwenkt werden und dann aufgrund des besseren Wirkungsgrades zunächst die kleinere Pumpe 1 die weitere Steigerung der Liefermenge übernehmen. Erst wenn diese voll angesteuert wird, übernimmt anschließend die größere Pumpe die weitere Erhöhung bis zum Erreichen des Maximalpunktes (100/100).





## 8.3.4 Skript

M1	MUL	PIN13	READY	-
M2	LIM	M1	0.0	M13
M3	ADD	PAR2	PAR3	-
M4	SUB	M1	M2	-
M5	DMUL	M4	M3	PAR2
M6	ADD	M2	M5	-
M7	LIM	M6	0.0	100.0
M8	MUL	M7	PIN5	-
M9	SUB	M6	M8	-
M10	DMUL	M9	PAR2	PAR3
M11	ADD	M10	M2	-
M12	AND	PIN7	PIN5	-
M13	SEL	M12	0.0	PAR1
M14	NOT	M12	-	-
M15	GT	M8	PAR4	-
M16	GT	M11	PAR4	-
M17	OR	M15	M16	-
PIN15	DIR	M8	-	-
PIN16	LIM	M11	0.0	100.0
PIN2	AND	M14	M17	-
LED_GN	OR	READY	ERFL	-
EN_SIG	DIR	PIN8	-	-

## 8.4 Leistungsbegrenzungsregelung

### 8.4.1 Aufgabe

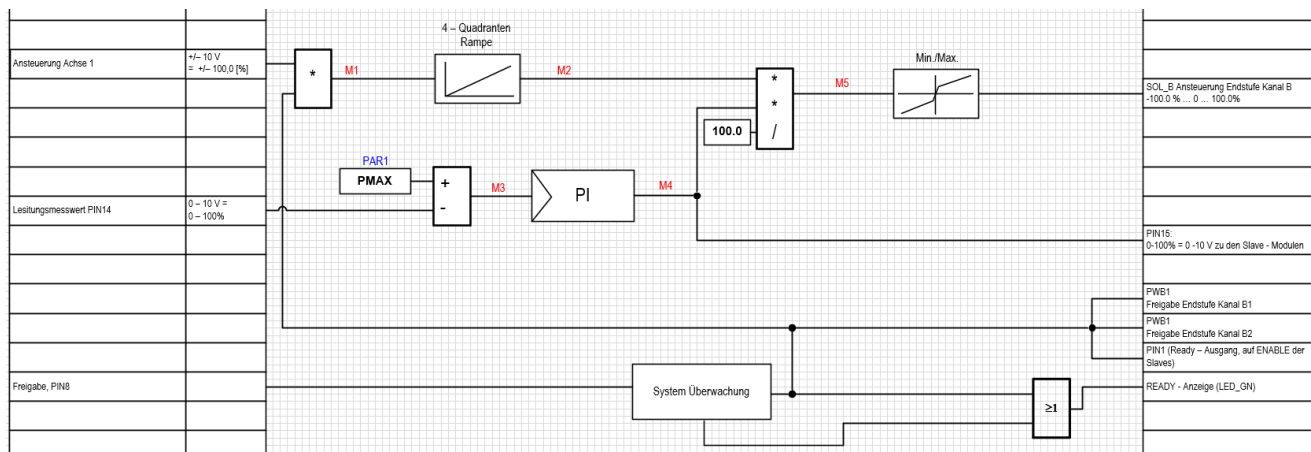
Eine Pumpe mit begrenzter Leistung (Antriebsleistung kleiner als die System – Eckleistung) speist parallele Verbraucher, die über Proportionalventile angesteuert werden. Für einen einzelnen Verbraucher ist in diesem Fall die Verwendung einer PAM-195-P-S3 möglich. Bei mehreren parallelen Verbrauchern ist eine andere Lösung erforderlich, bei der ein Begrenzungsregler auf alle beteiligten Achsen einwirkt.

Hierzu wird ein Gerät als „Master“ konfiguriert, d.h. es enthält den Regler und gleichzeitig die Ansteuerung einer der beteiligten Achsen, die übrigen Geräte erhalten eine reduzierte Funktionalität, d.h. sie empfangen das Ausgangssignal des Reglers aus dem Master und begrenzen die Ansteuerung des angeschlossenen Verbrauchers entsprechend.

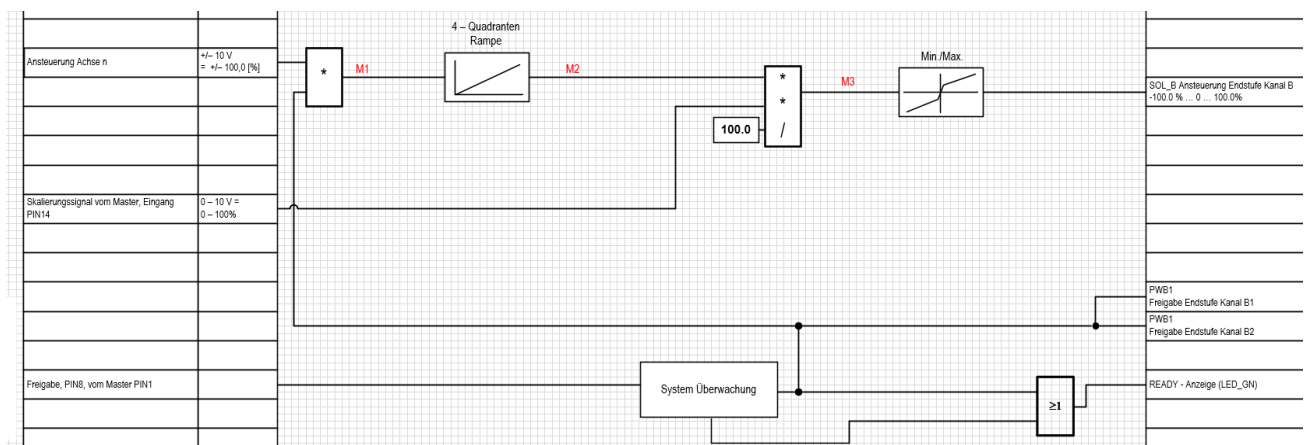
### 8.4.2 Lösung

Es kommt hier die DSG-112-P zum Einsatz, die über eine integrierte Endstufe verfügt. Pro Gerät wird ein Ventil mit zwei Magnetspulen angesteuert.

#### Teil 1: Master



#### Teil 2: Slave



Die Skripte für Master und Slave sind ähnlich aufgebaut. Der Ansteuersollwert von PIN9/10 (bipolar!) wird zunächst mit dem READY multipliziert, damit die Rampe bei einer Abschaltung wieder auf „0“ zurückläuft. Es folgt die im Modul vorgegebene Komplexfunktion „4-Quadrantenrampe“. Deren Ausgang wird bei beiden Geräten mit dem Ausgang des Leistungsreglers skaliert, beim Master ist die Verknüpfung intern realisiert, der Slave empfängt diesen Faktor über einen Analogeingang. Der Leistungsregler gibt normalerweise ein Signal von 100% aus und ist auf diesen Wert nach oben begrenzt. Bei Überschreitung seines Sollwertes sinkt das Ausgangssignal und begrenzt die Ansteuerung aller Ventile im gleichen Verhältnis. Es ist wichtig, dass der Eingriff nach der Rampe erfolgt, damit er verzögerungsfrei geschieht und vor der linearisierenden Min./Max.-Funktion. Mit deren Ausgangssignal kann die modulinterne Endstufe beaufschlagt werden, ein positives Signal wirkt auf Kanal B1, ein negatives auf B2.

## 8.4.3 Skript

### Teil 1: Master

M1	MUL	PIN910	READY	-
M2	RAMP4Q	M1	-	-
M3	SUB	PAR1	PIN14	-
M4	PI_1	M3	50.0	-
M5	DMUL	M2	M4	100.0
PIN15	DIR	M4	-	-
SOL_B	MINMAX1	M5	-	-
PIN1	DIR	READY	-	-
LED_GN	OR	READY	ERFL	-
PWB1	DIR	READY	-	-
PWB2	DIR	READY	-	-
EN_SIG	DIR	PIN8	-	-

Zusätzlicher Hinweis: Der Regler bekommt hier einen festen Rückführwert von 50%. Zusammen mit der Paramtereinstellung PID1:YR = 5000 (50%) bedeutet das: Sein Ausgangssignal wird auf  $50\% - 50\% = 0\%$  bis  $50\% + 50\% = 100\%$  begrenzt. Diese Einstellung ist für eine korrekte Funktion erforderlich.

### Teil 2: Slave

M1	MUL	PIN910	READY	-
M2	RAMP4Q	M1	-	-
M3	DMUL	M2	PIN14	100.0
SOL_B	MINMAX1	M3	-	-
LED_GN	OR	READY	ERFL	-
PWB1	DIR	READY	-	-
PWB2	DIR	READY	-	-
EN_SIG	DIR	PIN8	-	-

## 8.5 Ansteuerung für einen Lagerprüfstand

### 8.5.1 Aufgabe

Wälzlager sollen in einem Dauerlauf mit einer pulsierenden Kraft belastet werden. Der Kraftverlauf soll sinusförmig sein, mit einstellbarer Frequenz und Amplitude. Die Belastung wird durch einen Hydraulikzylinder aufgebracht.

Zur Ansteuerung wird ein schnelles Regelventil mit Nullschnittschieber verwendet. Es kommt eine Kraftmessdose zum Einsatz, die ein ausreichend schnelles Messsignal liefert. Der Anwender gibt den Anteil der statischen und der dynamischen Kraft (als Amplitude) an.

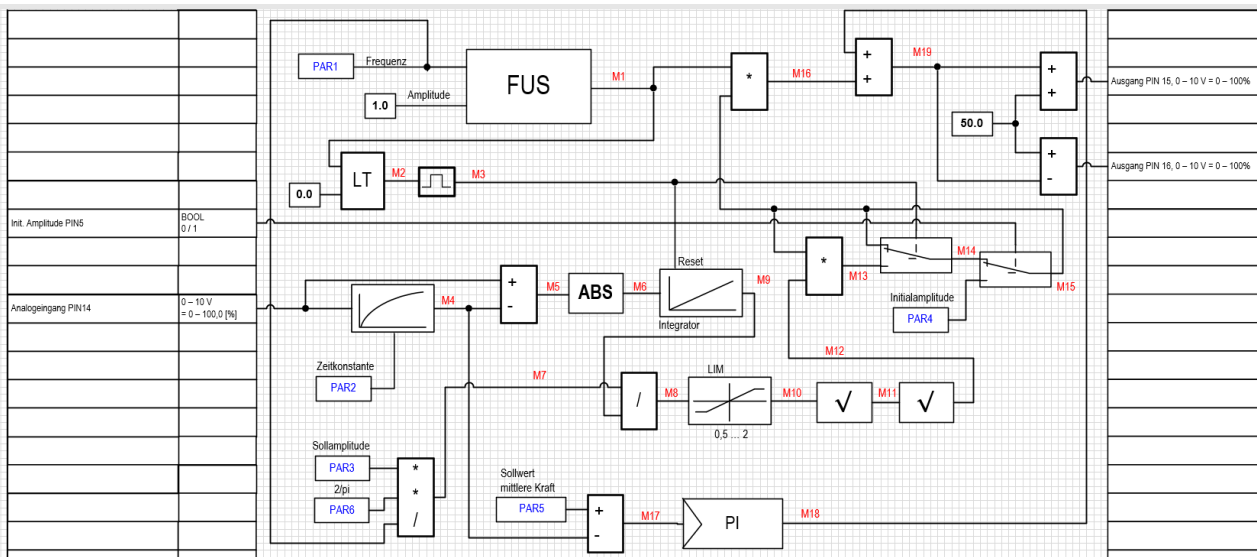
### 8.5.2 Lösung

Es handelt sich hier um eine Regelung von zwei Größen, der statischen und der dynamischen Kraft. Das Ventil wird mit einem Sinussignal angesteuert, dessen Amplitude so eingeregelt wird, dass sich die gewünschte Kraftamplitude ergibt und das mit einem Offset versehen wird, über den man die statische Kraftkomponente einstellt.

Zur Ermittlung dieser Komponente wird das Eingangssignal der Messung zunächst durch einen Tiefpass mit niedriger Grenzfrequenz vom Wechselanteil befreit (M4). Es folgt ein PI-Regler, den man relativ langsam einstellt und der mit dem Signal M18 den Mittelwert der Ansteuerung vorgibt.

Der Wechselanteil (M5) ergibt sich als Differenz der Gesamtkraft abzüglich der statischen Komponente. Es folgt eine Aufintegration des Betrages dieses Anteils über eine Periode. Zu diesem Zweck wird über M2 und M3 ein Impuls generiert, sobald das Sinussignal des Generators (M1) unter Null fällt. Dies findet einmal pro Vollwelle statt.

Das Integral  $\int_0^T |\sin(\omega \cdot t)| dt = \frac{4}{\omega} = \frac{2}{f \cdot \pi}$  würde bei einer Amplitude von „1“ bei M9 anstehen, wenn der Impuls kommt. Somit ist der Sollwert für dieses Signal gleich der gewünschten Amplitude mal  $\frac{2}{f \cdot \pi}$ . In M8 wird der Quotient aus Soll- und Istwert gebildet. Beim Erreichen der gewünschten Amplitude wird dieser Wert „1“. Man kann M8 auch als Korrekturfaktor auffassen, mit dem die augenblickliche Ansteuerung multipliziert werden müsste, um die gewünschte Amplitude zu erhalten. Der Regler kann einmal pro Durchgang (Vollwelle) eine Korrektur anbringen, damit diese nicht zu kräftig ausfällt, wird zunächst eine Signalbegrenzung vorgenommen (M10) und dann durch zweimaliges Wurzelziehen eine weitere Reduktion bzw. Annäherung an den Faktor 1,0 bewirkt. Die Schaltung aus M13/M14/M15 schleust die variable Verstärkung in den Signalfluss ein. Über M15 kann von außen auf einen Initialwert der Amplitude geschaltet werden, z.B. unmittelbar nach Start des Systems oder nach Wechsel des Betriebspunktes. Diese Art der Regelung ähnelt dem von W.E.St. patentierten Konzept des MR – Reglers und bewirkt eine schnelle und stabile Annäherung an den Zielwert. Zu beachten ist, dass bedingt durch die Schaltung eine Korrektur nur in dem Moment erfolgt, in dem über eine komplette Sinuswelle integriert wurde. Durch die getauschte Bearbeitungsreihenfolge M8 / M9 wird erreicht, dass in diesem Moment das Resultat der Integration aus dem vorangehenden Schritt für die weitere Berechnung genutzt wird. Der Ausgang des Integrators wird im Durchgang des Impulses auf „0“ zurückgesetzt.



## 8.5.3 Skript

M1	FUS	PAR1	1.0	-
M2	LT	M1	0.0	-
M3	FP	M2	-	-
M4	PT1	PIN14	PAR2	-
M5	SUB	PIN14	M4	-
M6	ABS	M5	-	-
M7	DMUL	PAR3	PAR6	PAR1
M8	DMUL	M7	1.0	M9
M9	INTEG	M6	M3	-
M10	LIM	M8	PAR7	2.0
M11	SQRT	M10	-	-
M12	SQRT	M11	-	-
M13	MUL	M15	M12	-
M14	SEL	M15	M13	-
M15	SEL	PIN5	M14	PAR4
M16	MUL	M1	M15	-
M17	SUB	PAR5	M4	-
M18	PI_1	M17	-	-
M19	ADD	M16	M18	-
PIN15	ADD	M19	50.0	-
PIN16	SUB	M19	50.0	-
LED_GN	OR	READY	ERFL	-
EN_SIG	DIR	PIN8	-	-

Da die Werte  $2/\pi$  und 0,5 nicht als feste Konstanten zur Verfügung stehen, werden die Parameter PAR6 und PAR7 verwendet, die entsprechend zu setzen sind.

## 8.6 Frequenzgangmessung

### 8.6.1 Aufgabe

Der Frequenzgang einer Regelstrecke mit Proportional-Druckventil soll ermittelt werden. Das Ventil hat eine integrierte Endstufe und wird mit einem Signal 0-10 V angesteuert. Der Druck wird mit einem Sensor erfasst, dessen Signal zur Auswertung dem Gerät als Eingangssignal zur Verfügung steht. Ziel ist die Erstellung eines Bode-Diagramms für verschiedene Signalamplituden.

Dieses Beispiel lässt sich abwandeln, so dass Frequenzgänge beliebiger Strecken gemessen werden können. Es ist nicht auf hydraulische Systeme beschränkt. Begrenzender Faktor ist jedoch die eigene Abtastzeit von 1 ms. Wie man im Abschnitt 8.6.3 sehen wird, sind sinnvolle Messungen bis zu einer maximalen Frequenz von etwa 100 Hz möglich. Dies setzt jedoch voraus, dass man oberhalb von 10 Hz eine Kompensation mit dem Frequenzgang der Baugruppe durchführt.

### 8.6.2 Lösung

Das hier zum Einsatz kommende Programm ist schon relativ aufwändig und stellt die Leistungsfähigkeit der Skriptprogrammierung unter Beweis. Es wird das sogenannte Korrelationsverfahren angewandt. Die Strecke wird mit einem Sinus-Testsignal beaufschlagt, dessen Frequenz langsam gesteigert wird (Sweep). Hierzu kommt der Baustein „FUS“ zum Einsatz.

Bei sinusförmiger Ansteuerung einer Strecke kann der Real- und Imaginärteil ihres Frequenzganges bestimmt werden durch:

$$Re = \frac{4 \cdot \int_0^T y(t) \cdot \sin(\omega \cdot t) dt}{\pi \cdot \hat{u} \int_0^T \sin(\omega \cdot t) dt} = \frac{2 \cdot \int_0^T y(t) \cdot \sin(\omega \cdot t) dt}{T} \quad \text{und}$$

$$Im = - \frac{2 \cdot \int_0^T y(t) \cdot \cos(\omega \cdot t) dt}{T}$$

Wobei  $y(t)$  das Ausgangssignal der Strecke, also das Eingangssignal des Gerätes und  $\hat{u}$  die Amplitude des Ansteuersignals darstellen.

Es folgt für den Amplitudengang:

$$|G(j\omega)| = \sqrt{Re^2 + Im^2}$$

Und für den Phasenverlauf:

$$\varphi = \tan^{-1} \left( \frac{Im}{Re} \right)$$

Diese abschließenden Berechnungen können auch im Nachgang offline durchgeführt werden. In dem Modul ist die Ansteuerung des Prüflings durch ein Sinussignal steigender Frequenz und die Aufintegration des Messwertes bzw. die laufende Bestimmung der Größen „Re“ und „Im“ zu realisieren. Wenn man eine Messung durchführen möchte, zeichnet man mit WPC diese Werte und die Frequenz auf und speichert sie anschließend zur Weiterverarbeitung in einer \*.csv-Datei. Dieses Vorgehen hat den Vorteil, dass man keine höherfrequenten Informationen aufzeichnen muss, sondern bereits vorverarbeitete Daten, so dass die Aufzeichnungsgeschwindigkeit kein begrenzender Faktor ist.

Umseitig ist der Funktionsplan eines geeigneten Skriptes dargestellt.

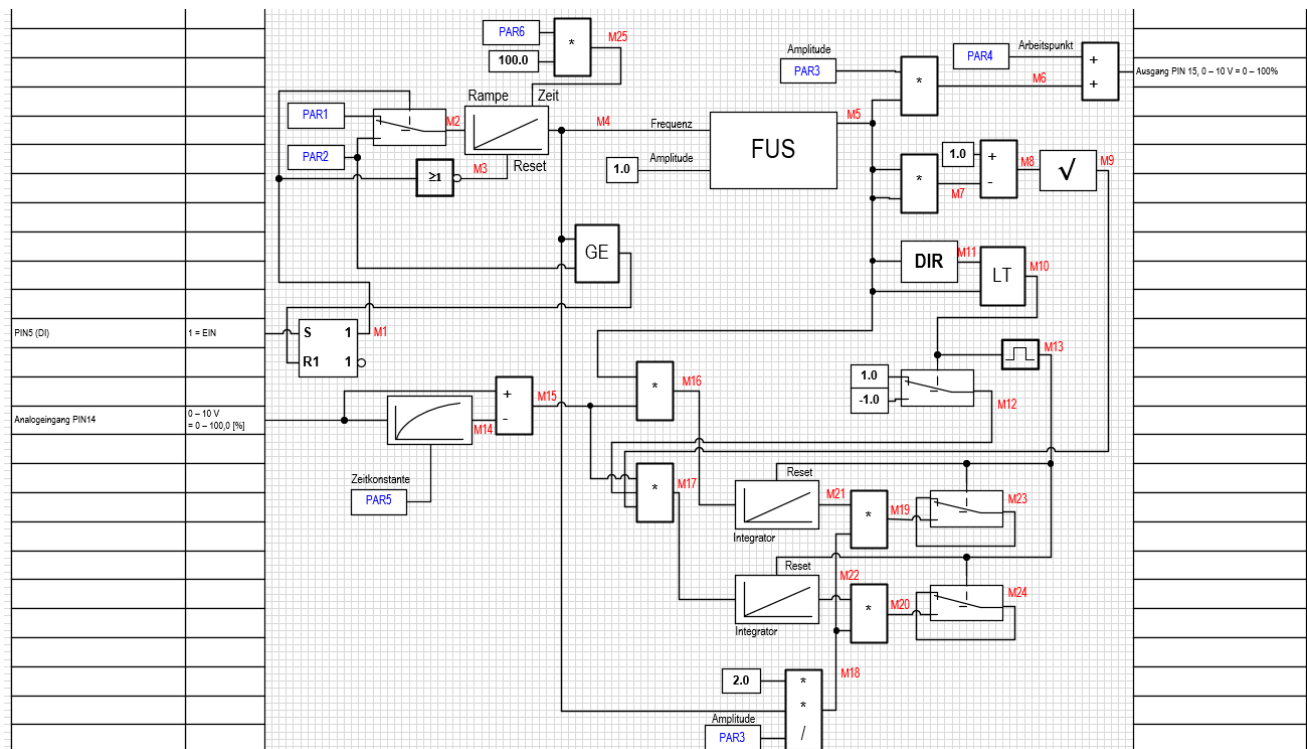
Im oberen Teil ist die Erzeugung des Sweep-Signals, realisiert mit Rampe und Sinusgenerator, zu sehen. Ein Sweep wird mit Setzen des Eingangs an PIN5 gestartet. Zum Schluss springt die Frequenz wieder auf den Startwert zurück. Die Frequenzgrenzen werden durch PAR1 (Start) und PAR2 (Ende) festgelegt. Die mit PAR6 einzustellende Geschwindigkeit sollte sehr langsam sein, damit man von stationären Verhältnissen ausgehen kann.

Bei M5 liegt während der Messung ein Sinussignal mit der Amplitude „1“ vor. Mit PAR3 wird die Amplitude am Prüfling bestimmt und mit PAR4 der Offset bzw. Arbeitspunkt.

Für die Bildung des Integranden in der Formel des Imaginärteils ist es erforderlich, neben der sinusförmigen Ansteuerung den Cosinus zu bilden. Dies wird hier unter Ausnutzung des Satzes von Pythagoras ( $\cos^2(\alpha) + \sin^2(\alpha) = 1$ ) in M9 realisiert, wobei M9 den Betrag des Cosinus enthält und der Schalter nach M12 das Vorzeichen mitbringt. Der Cosinus ist dann negativ, wenn der Sinus fällt, denn er ist die Ableitung der Sinusfunktion. Einmal pro Vollwelle wird der Impuls M13 ausgelöst, der die Aktualisierung des Ergebnisses auslöst.

In M15 ist die Antwort der Strecke, befreit vom stationären Anteil, enthalten. Zu diesem Zweck ist die Zeitkonstante des PT1 zur Mittelung des Eingangswertes extrem sehr hoch einzustellen, so dass M14 als konstant angesehen werden kann ( $> \text{Faktor } 100 \text{ höher als die Periodendauer bei der kleinsten Messfrequenz}$ ). In den beiden Integratoren werden die beiden oben angegebenen Integrale gebildet, nach Erreichen der Zeit T wird das Ergebnis mit den übrigen Faktoren M18 multipliziert und dann in den als Signalspeicher genutzten Sektoren gespeichert.

Der Anwender definiert die Signale M4 (Frequenz), M23 (Realteil) und M24 (Imaginärteil) im WPC als zu beobachtende Skriptvariablen SC:A /:B /:C und zeichnet diese Werte während der Messung auf.



## 8.6.3 Skript

M1	RS	PIN5	M26	-
M2	SEL	M1	PAR1	PAR2
M3	NOT	M1	-	-
M4	RAMP	M2	M25	M3
M5	FUS	M4	1.0	-
M6	MUL	M5	PAR3	-
M7	MUL	M5	M5	-
M8	SUB	1.0	M7	-
M9	SQRT	M8	-	-
M10	LT	M11	M5	-
M11	DIR	M5	-	-
M12	SEL	M10	1.0	-1.0
M13	FP	M10	-	-
M14	PT1	PIN14	PAR5	-
M15	SUB	PIN14	M14	-
M16	MUL	M5	M15	-
M17	MUL	M5	M12	M9
M18	DMUL	2.0	M4	PAR3
M19	MUL	M21	M18	-
M20	MUL	M22	M18	-
M21	INTEG	M16	M13	-
M22	INTEG	M17	M13	-
M23	SEL	M13	M23	M19
M24	SEL	M13	M24	M20
M25	MUL	PAR6	100.0	-
M26	GE	M4	PAR2	-
PIN15	ADD	PAR4	M6	-

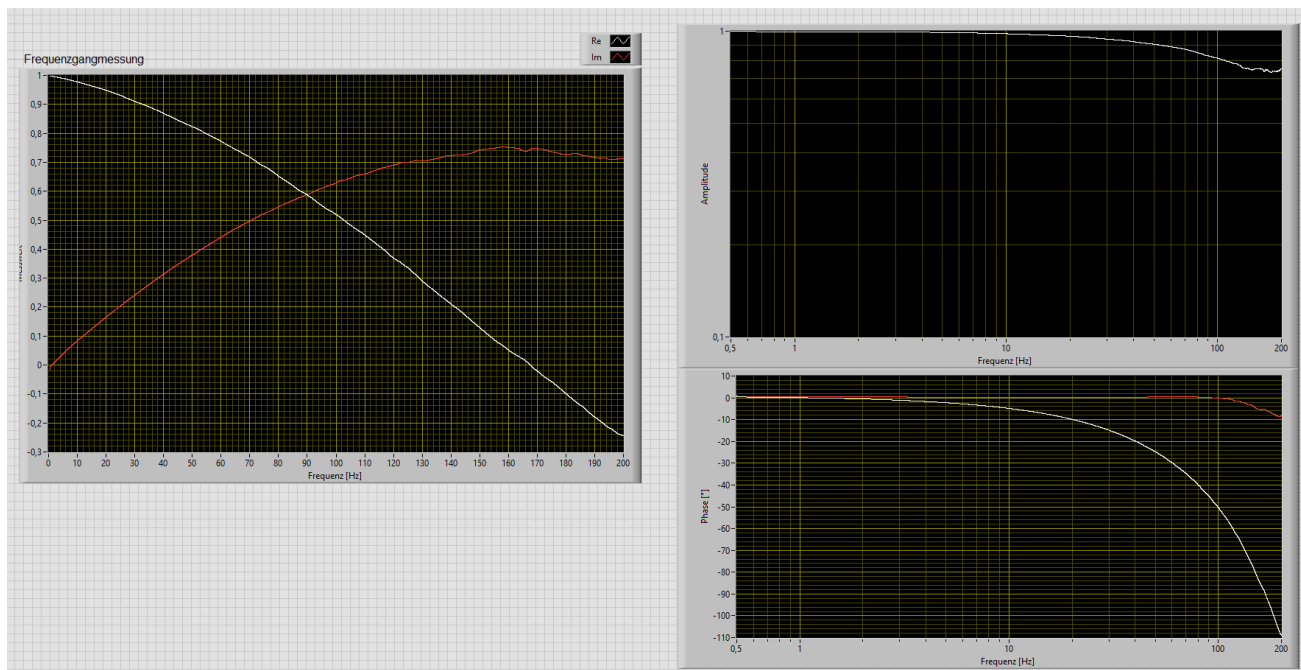
## 8.6.4 Ergebnisse

Zu Testzwecken wurde dieses Beispiel auf einer Baugruppe mit direkt verbundenem Aus- und Eingang betrieben. Somit misst das Skript also den Frequenzgang der Baugruppenperipherie.

Zur Auswertung wurden die obenerwähnten Signale mit dem Oszilloskop des WPC aufgezeichnet und in einer \*.csv – Datei gespeichert. Erwartungsgemäß fluktuierten die Integralwerte etwas, so dass eine Glättung ihrer Verläufe durch eine Filterung erfolgte, anschließend wurden Amplituden- und Phasengang berechnet.

In dieser Darstellung sieht man links die gefilterten Rohwerte und rechts das erzeugte Bodediagramm.

Der Rohwert des Imaginärteils wurde zur besseren Darstellung an der X-Achse gespiegelt.



Wie man sieht, wurde ein Bereich von 0,5-200 Hz vermessen.

Der Phasenverlauf (weiße Kurve) zeigt einen deutlichen Abfall oberhalb von 10 Hz.

Stellt man die X-Achse auf eine lineare Skala um, so bemerkt man, dass es sich um einen linearen Abfall handelt.

Dieser Effekt wird durch die zeitdiskrete Abtastung und die damit verbundene Totzeit erzeugt. Es wurde empirisch ermittelt, dass sich das Verhalten durch ein Totzeitelement mit einer Verzögerung von 1,4 ms beschreiben lässt. Wenn man eine entsprechende Kompensation des Phasengangs vornimmt ergibt sich die rote Kurve, die bis 100 Hz keinen nennenswerten Abfall zeigt. Die Amplitude ist bis dorthin auf ca. 0,8 gesunken.

Bis zu einer Frequenz von 100 Hz ist es also möglich, eine sinnvolle Messung von Strecken mit diesem einfachen Aufbau durchzuführen.

Auf Nachfrage stellen wir das hier verwendete, einfache LabView-VI zur Auswertung gerne als Quelle oder in kompilierter Form zur Verfügung.