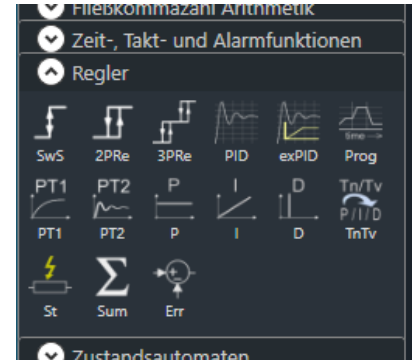


Regelungen, Listen und Programmregler

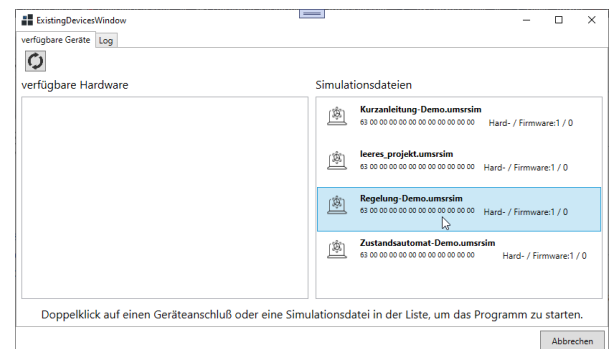
(Stand 11.2021, eine Teilbeschreibung des μ MSR50 Systems: <https://ing-brauns.de/umsr50>)

Dieses Dokument zeigt an einigen Beispielen, wie unter Verwendung der μ MSR-Soft-SPS, sich auch unterschiedlichste Regelungen umsetzen lassen. Dabei können parallel zu den anderen Automatisierungen oder IoT Abläufen, wie zuvor gezeigt, einzelne oder mehrere Regler konfiguriert werden. Wie ein einfacher 2 Punktregler umgesetzt wird wurde schon in der Kurzanleitung gezeigt, daher sollen hier primär Beispiele für einfache PID-Regler bis hin zu Programmreglern vorgestellt werden.



Alle hier vorgestellten Beispiele sind in der „Regelung-Demo.umsrsim“ - Datei, welche sich mit in der μ MSR-Testversion befindet.

Die folgenden Beispiele sind nur kurz stichpunktartig beschrieben. Die Grundlagen zum μ MSR50-System wurden in der Kurzanleitung beschrieben. Die Beispiele sind dabei nur eine kleine Auswahl unzähliger weiterer Möglichkeiten.

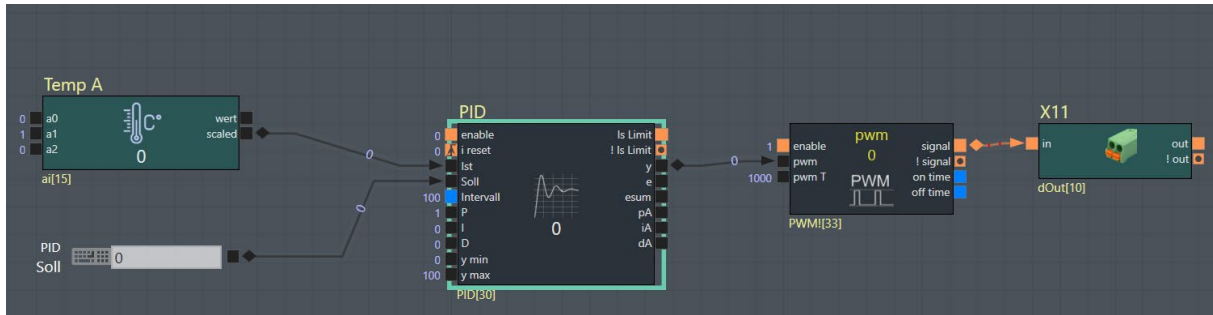


Inhalt

Beispiel 1 (Blatt 0, oben): Ein einfacher PID-Regler	2
Beschreibung PID-Regler Block	3
Beispiel 2 (Blatt 0, Mitte): Einen einfachen PID-Regler simulieren	4
Beispiel 3 (Blatt 0, unten): Einen einfachen PID-Regler aus einzelnen Strukturblöcken	5
Beispiel 4 (Blatt 1): PID-Regler mit Rampe und Holdback	6
Beispiel 5 (Blatt 2): Listen-Blöcke	7
Listen verknüpfen	7
Mehrdimensionale Liste	8
Beispiel 6 (Blatt 3): Programmregler	9
Programmregler im Detail	10
Beispiel 7 (Blatt 4): Beispiel PID-Regler mit variablen PID-Parametern	11

Beispiel 1 (Blatt 0, oben): Ein einfacher PID-Regler

Hier ein Beispiel für eine Temperaturregelung, z.B. Raumtemperatur, Wassertemperatur, Der Raum verfügt über eine Heizung, deren Leistung über PWM eingestellt werden kann und einem Temperatursensor, welcher mit dem „Temp A Anschluss“ verbunden ist.



- Aus der Toolleiste wird der „Temp A“, ein PID Regler, ein PWM-Block und ein Digitalausgang auf die Arbeitsfläche gezogen und wie dargestellt verbunden.
- Jetzt nur noch die üblichen Einstellungen am PID Regler vornehmen
- Für die Sollwertvorgabe evtl. noch ein Eingabefeld im Diagrammdesigner hinzufügen.
- Würde der Regler jetzt auf einer realen μ C-Hardware laufen, wäre er jetzt fertig

Beschreibung PID-Regler Block

Wie bei alle Blockfunktionen im μ MSR50-System sind alle Optionen und Funktionseingänge auf der linken und alle Ergebnisse und abfragbaren Zustände auf der rechten Seite angeordnet.

Alle Ergebnisse und abfragbaren Zustände können im Diagramm dargestellt oder als Eingang für weitere Berechnungen verwendet werden.

Alle Optionen und Funktionseingänge können entweder mit Ausgängen anderer Blöcke verbunden werden. Wenn die Eingänge nicht mit anderen Blöcken verbunden sind, können sie oben links in den Einstellungen angepasst werden.

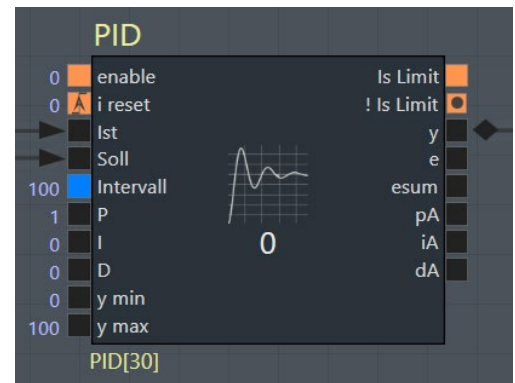
Bis auf wenige optionale Parameter (hier z.B. die ILimit) entspricht der Einstellungsbereich immer der linken Seite eines Blocks. Somit ist es möglich, jeder Option einen festen Wert zu geben oder diesen dynamisch durch andere Blöcke berechnen zu lassen.

Zu den PID-Regler-Block Optionen:

Zeile 1: Ist die BlockID, über diese lassen sich z.B. blattübergreifende Verknüpfungen und einfache API-Zugriffe festlegen.

Zeile 2 und 3: Sind beschreibende Metainformationen die bei allen Blöcken vorhanden sind.

Enable: Ist ein digitaler Eingang bzw. Option, welche den Regler einschaltet. Er kann z.B. dauerhaft an sein, über ein Designer-Block oder mit anderen Blockfunktionen bedingt geschaltet werden.



I reset: Optional kann es sinnvoll sein den I-Anteil des Reglers unter bestimmten Bedingungen extern zurückzusetzen

Ist / Soll: Der vorgegebene Soll-Wert und der Ist-Wert der Regelstrecke. In diesem Beispiel sind diese Eingänge mit anderen Blöcken verbunden, weshalb sie hier nicht verändert werden können.

P,I,D: Die Reglerparameter.

Y min, y max: Begrenzen den Ausgangswert des Reglers auf dieses Intervall.

ILimit: I-Anteil-Begrenzung

Zu den PID-Regler-Block Ergebnissen:

Is Limit gibt an, dass intern die I-Anteilbegrenzung wirkt.

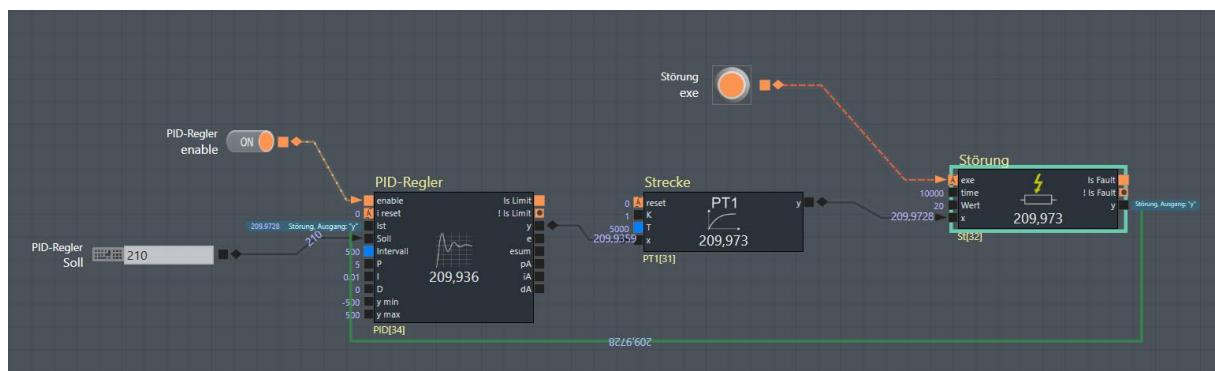
y ist die Regelgröße für das Stellglied.

e ist der Fehler / die Abweichung von Soll – Ist, esum ist die Fehlersumme des I-Anteils.

pA, iA, dA sind die P-, I- und D-Anteile der Regelgröße.

Beispiel 2 (Blatt 0, Mitte): Einen einfachen PID-Regler simulieren

Das erste Beispiel wäre nur auf einer realen Hardware mit einer verbundenen Regelstrecke anwendbar. Um dennoch mit der Testversion den Funktionsumfang zu testen, gibt es die Möglichkeit eine Regelstrecke im Designer aufzubauen. Diese Möglichkeit kann auch für Regelungstechnikschulungen nützlich sein. Wenn die Parameter der Regelstrecke bekannt sind kann diese Funktionalität genutzt werden, um ein reales System zu simulieren um z.B. optimale Reglerstrukturen und Parameter zu bestimmen.



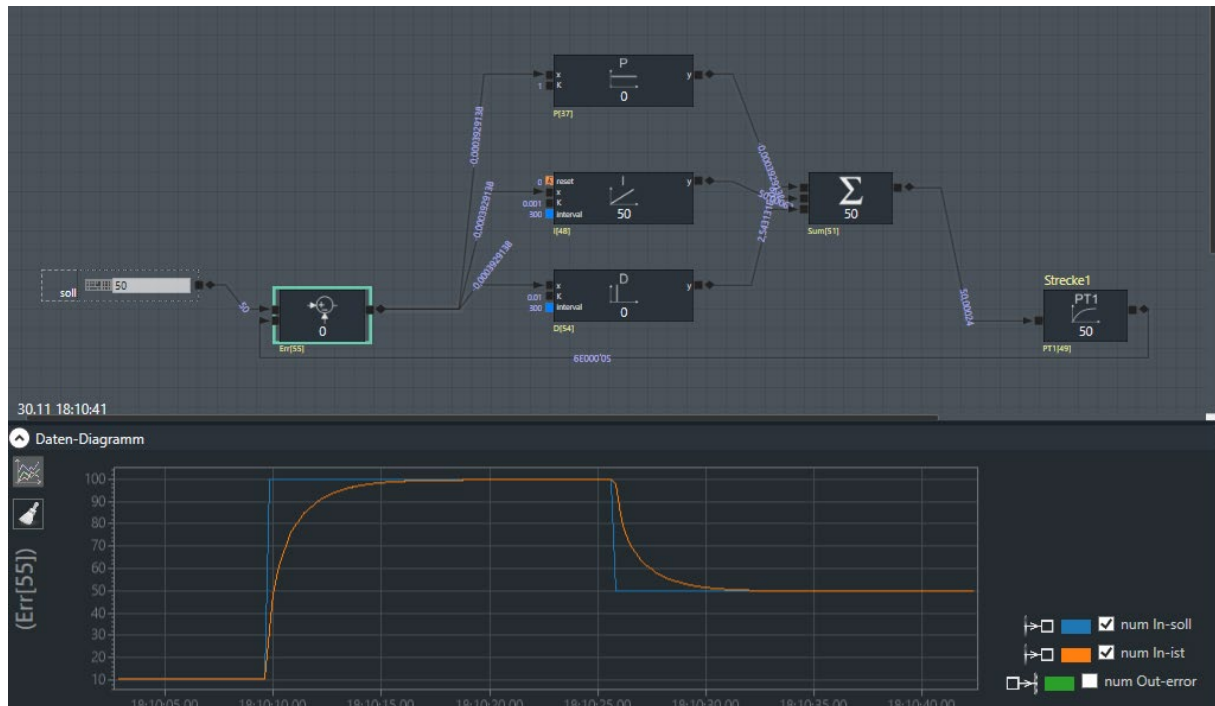
- Für dieses Beispiel wird wieder ein PID-Regler-Block aus der Tooleiste auf die Arbeitsfläche gezogen.
- Als Regelstrecke wird ein PT1-Glied verwendet. Ein PT1-Glied beschreibt z.B. das typische Verhalten bei einer einfachen Temperaturregelung oder einer Drehzahlregelung eines Verbrennungsmotors.
- Um das Verhalten dieser Regelschleife einfach zu testen, wird der Ausgang des PT1-Glieds nicht direkt mit dem PID-Ist-Eingang verbunden, sondern zusätzlich noch eine Störungs-Block-Funktion eingefügt.
- Diese Störungs-Block-Funktion addiert für eine angegebene Zeit einen eingestellten Wert auf den Istwert. Nach Ablauf dieser Zeit wird dieser Wert wieder abgezogen.
- Klickt man nun den Störungs-Taster an, erhält man folgende Ausgabe im Diagramm:



- Nach Belieben können die Parameter nun verändert und der Einfluss auf das Regelverhalten untersucht werden. Dazu kann man sich wie gewohnt alle Parameter der Blöcke im Diagramm unten ansehen. So z.B. auch die P-, I- und D-Anteile des Reglers.

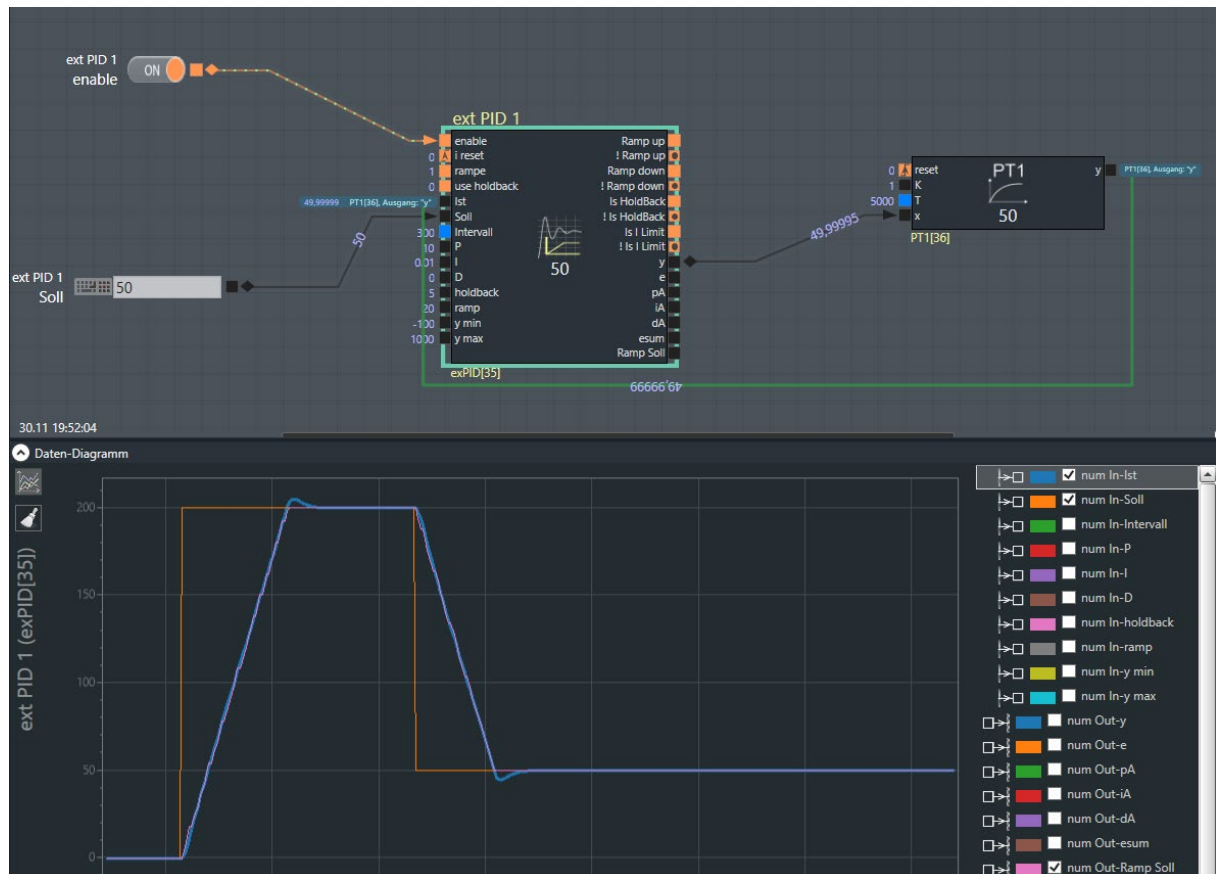
Beispiel 3 (Blatt 0, unten): Einen einfachen PID-Regler aus einzelnen Strukturblocken

Sollte die Struktur des klassischen PID-Reglers nicht geeignet sein, lassen sich auch andere Reglerstrukturen aus einzelnen Blöcken aufbauen. Die untere Grafik zeigt den PID zusammengesetzt aus P-, I- und D-Glieder:



Beispiel 4 (Blatt 1): PID-Regler mit Rampe und Holdback

Neben dem PID-Regler gibt es noch den erweiterten PID-Regler als Funktionsblock. Dieser Reglerblock bietet zusätzlich eine Rampenfunktion, sowie eine optionale Holdback-Option. D.h. der Sollwert wird nicht so schnell, wie es die Reglerparameter hergeben, erreicht, sondern entlang einer definierten Rampe. Ist die Holdback-Option eingeschaltet würde der Regler die Rampe erst dann fortsetzen, wenn der Istwert den vorgegebenen Rampenwert erreicht hat.

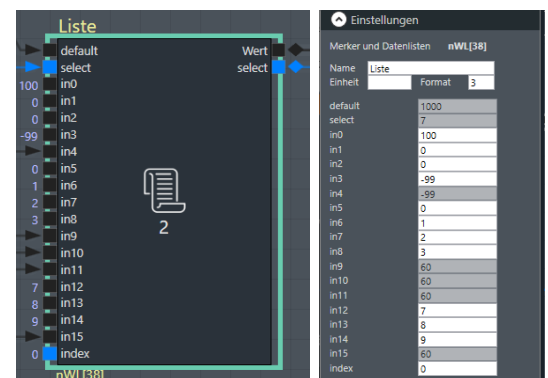


Beispiel 5 (Blatt 2): Listen-Blöcke

Zur Vorbereitung auf das folgende Beispiel, den Programmregler, sollen an dieser Stelle die Listenfunktionen vorgestellt werden. Diese befinden sich in der Merker und Datenliste-Toolbox. Hier gibt es Listen für die einzelnen Datenformate des µMSR50-Systems (Digitale, Ganzzahl, Kommawerte und Schaltzeitpunkte). Die Anwendung ist dabei immer dieselbe und wie hier am Beispiel von numerischen Werten vorgestellt.

Nebenstehend ein Beispiel für eine Datenliste. Diese Liste kann 16 Werte enthalten, welche entweder fest eingestellt werden oder durch Verknüpfungen mit anderen Blöcken versorgt werden.

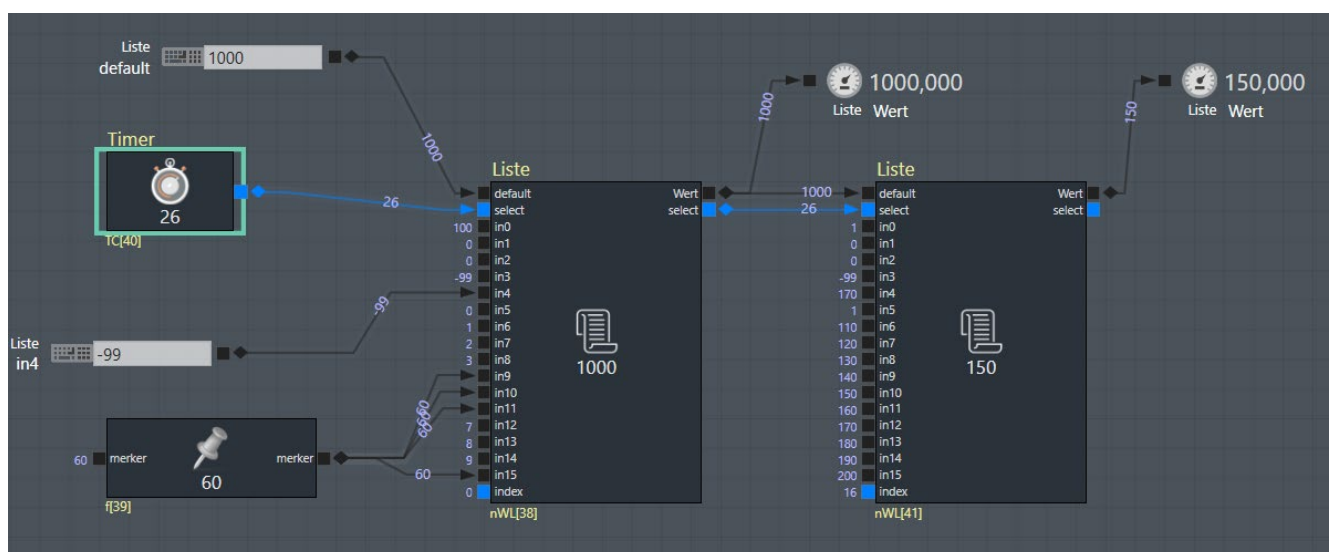
Der Ganzzahleingang „select“ bestimmt, welcher Wert am Ausgang anliegt. Der Index-Eingang legt die Nummer des in0 fest. Sollte der select-Eingang einen anderen Wert als index .. index+15 haben, wird der default-Eingang als Ausgangswert verwendet. Im Prinzip ist dieser Block wie ein Multiplexer mit 16 Eingängen zu sehen.



Listen verknüpfen

Mit der Möglichkeit einen Default-Wert und einen index für den in0 anzugeben, können jetzt mehrere Listen zu einer Liste verknüpft werden. Dazu das folgende Beispiel.

Der Ausgangswert der ersten Liste wird als Default-Wert für die zweite Liste verwendet. Der select-Wert wird einfach von der ersten Liste auf die zweite Liste übergeben. Die zweite Liste bekommt die index-Nummer 16. Somit ist es jetzt möglich 32 Werte zu speichern. In dem Beispiel unten wird ein Timer, welcher von 0 bis 31 zählt, mit dem select-Eingang der ersten Liste verbunden.



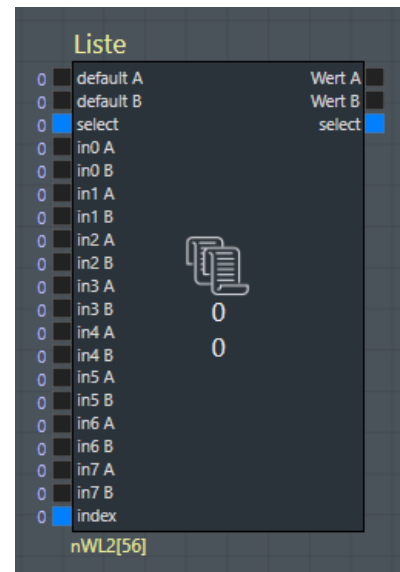
Der Ausgang-Wert der zweiten Liste gibt nun nacheinander die Werte der ersten und zweiten Liste aus. Mit dem gleichen Muster lassen sich jetzt beliebig viele Listen verknüpfen.

Mehrdimensionale Liste

Nun gibt es Aufgaben, wo nicht unbedingt 16 Werte benötigt werden, dafür jedoch mehrere zu einem Zeitpunkt. Ein Beispiel ist die 2D Positionierung. Hier werden jetzt zu einem Zeitpunkt die Werte für die x und y Koordinate benötigt. Das lässt sich mit 2 Listen umsetzen. Werden jedoch nicht mehr als 8 Positionen benötigt, kann für diese Aufgabe der „num.Werteliste2“-Block verwendet werden.

Der Unterschied zur zuvor beschriebenen Liste ist, dass diese Liste für einen bestimmten „select“-Wert zwei Ausgangswerte besitzt. Daher gibt es eben auch einen in0-A und in0-B Wert der dann als Wert A und Wert B ausgegeben wird.

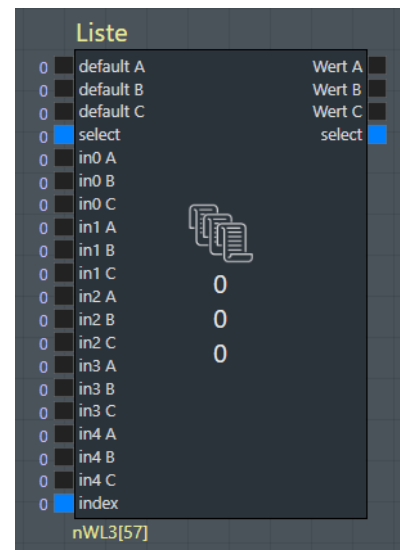
Die zuvor beschriebene Möglichkeit der Verknüpfung mehrerer Listen besteht auch bei dieser Listenart.



Häufig gibt es die Anforderung, dass zu einem Zeitpunkt drei Werte eingestellt werden sollen. Typische Beispiele können 3D-Positionsaufgaben sein. Auch können z.B. die PID-Parameter eines Reglers in einer Liste gespeichert und in Abhängigkeit von bestimmten Bedingungen dem Regler zur Verfügung gestellt werden.

Sicherlich könnten dafür auch drei separate Listen verwendet werden, wobei wenn z.B. nur 5 Vektoren benötigt werden, reicht eben auch die eine Liste.

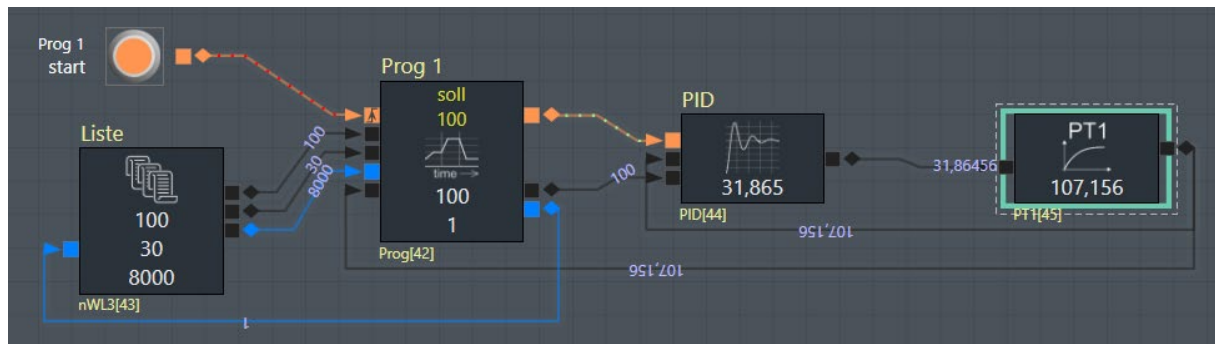
Weniger Blöcke vereinfachen die Diagrammstruktur, was vorteilhaft für die Abarbeitungsgeschwindigkeit, dem Ressourcenverbrauch und der Übersichtlichkeit ist



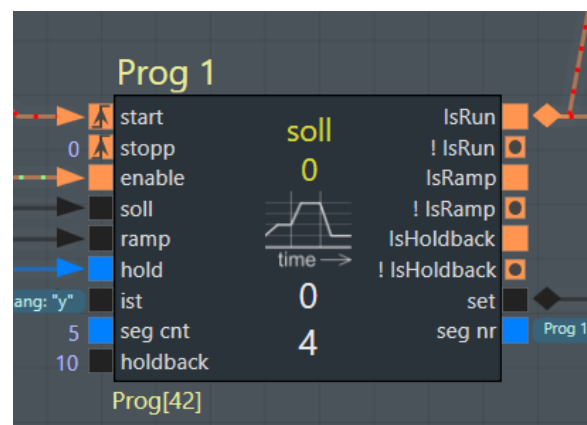
Beispiel 6 (Blatt 3): Programmregler

Ein Programmregler setzt bestimmte Werte eines Systems, wenn diese erreicht wurden wird eine Aktion ausgeführt und das System wird mit neuen Werten gesetzt. Typische Beispiele sind Temperaturprofile, Positionierungsaufgaben usw.

Programmregler können klassisch mit Schrittketten, unter Zuhilfenahme eines Zustandsautomaten oder mit der entsprechenden Programmregler-Blockfunktion, wie nachfolgend gezeigt, umgesetzt werden.



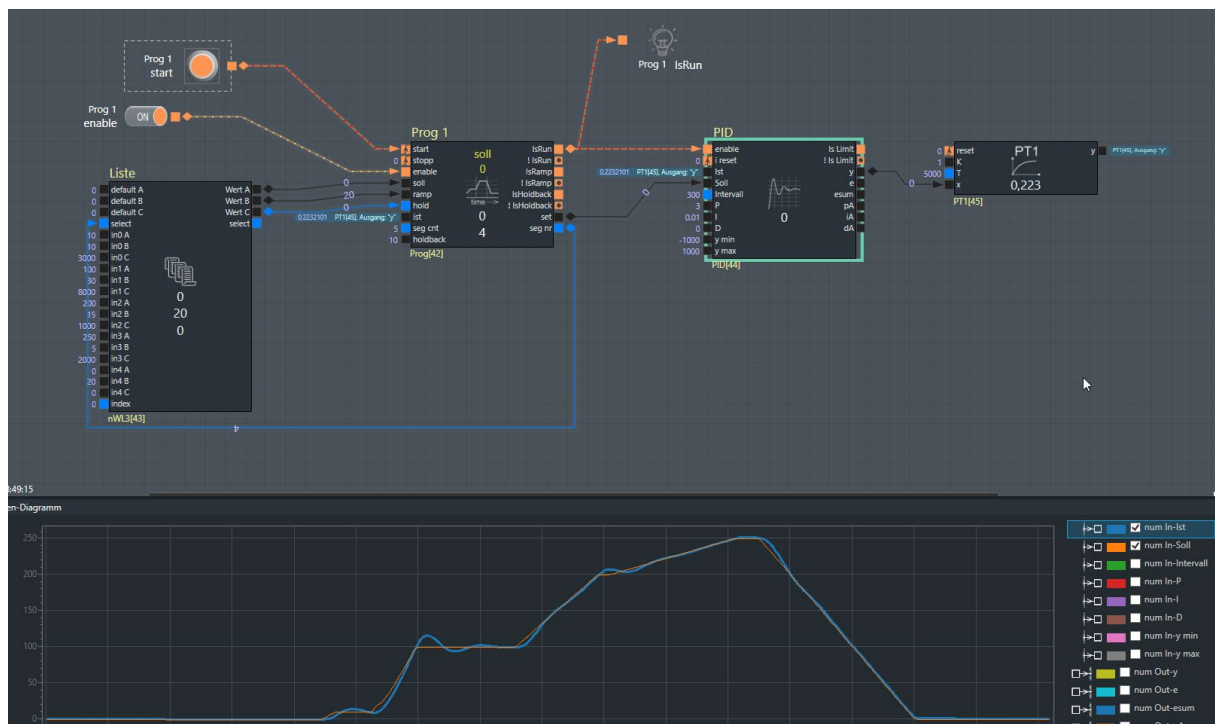
Der Programmregler bekommt als Eingangswert eine Sollwertvorgabe, sowie die Angabe eines Rampenwerts für das Erreichen dieses Sollwerts und eine Haltezeit für diesen Sollwert. Aus der Programmlaufzeit und den Vorgaben berechnet der Programmregler den aktuellen Sollwert, welcher einem Regler-Block als Sollwert übergeben werden kann. Als Regler kann dann eine beliebige Regler-Struktur eingesetzt werden. Im Beispiel wird dazu ein PID-Regler verwendet.



Nach Erreichen des Sollwerts und Ablauf der Haltezeit erhöht der Programmregler eine Nummer, d.h. er befindet sich im nächsten Programmschritt. Über die Programmnummer können jetzt, z.B. über die Listenblöcke, die nächsten Programmwerte vorgegeben werden.

Programmregler im Detail

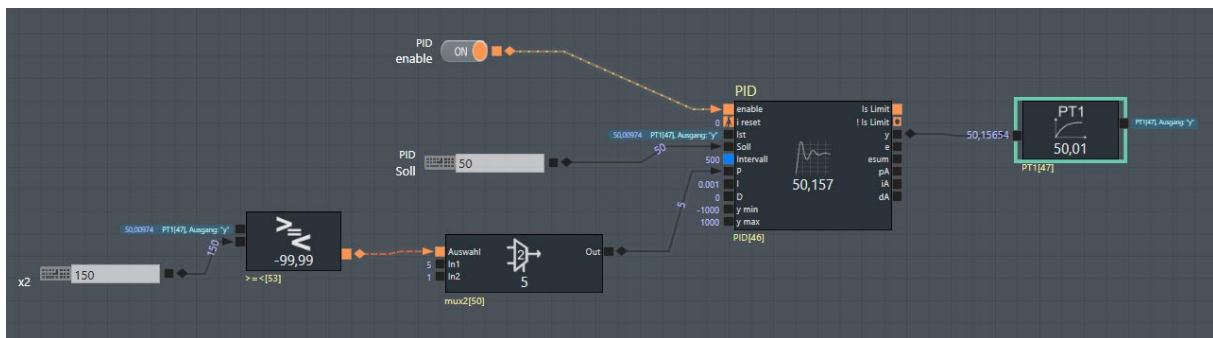
- Der Programmregler gibt seinen Sollwert an einen konfigurierten PID Regler der seinerseits die Regelstrecke, hier ein PT1-Glied, regelt.
- Der Istwert der Regelstrecke wird wieder dem PID Regler bereitgestellt. Hier muss jetzt jedoch der Istwert auch dem Programmregler (Prog1) bereitgestellt werden.
- Der Programmregler bekommt die Angabe das es sich um 5 Segmente handelt.
- Die eigentlichen Programmwerte, bekommt der Programmregler aus einer Liste, mit jeweils einer Angabe für den Sollwert, einer Rampe über die der Sollwert erreicht werden soll und eine Haltezeit an diesem Sollwert.
- Damit sich die Werte für jedes Segment ändern wird die Segmentnummer des Programmreglers mit dem select-Wert der Liste verbunden.
- Mit dem Klicken der Start-Taste wird nun das Programm abgearbeitet. Im Diagramm unten ist das Ergebnis dieser Konfiguration zu sehen.



Beispiel 7 (Blatt 4): Beispiel PID-Regler mit variablen PID-Parametern

Gelegentlich kommt es vor, dass die PID-Parameter in Abhängigkeit vom Ist-Wert oder anderer Größen, z.B. Außentemperatur angepasst werden sollen. Mit der Idee des μ MSR50-Systems alle Parameter eines Blocks / einer Funktion als Eingangswert offenzulegen, können diese nach Belieben dynamisch angepasst werden.

Hierzu folgendes Beispiel. Dort wird der P-Parameter mit Hilfe eines Multiplexers in Abhängigkeit vom aktuellen Ist-Wert angepasst. Natürlich könnte man hier auch über z.B. Listen alle PID-Parameter für beliebig viele Intervalle setzen.



Ich bedanke mich für Ihr Interesse und freue mich über jede Kritik und Anregung.

Sie erreichen mich unter info@ing-brauns.de