

Hochschule Magdeburg-Stendal
Fachbereich Ingenieurwissenschaften und Industriedesign (IWID)
Institut für Elektrotechnik

Bachelorarbeit

zur Erlangung des Grades eines "Bachelor of Engineering"
im Studiengang Mechatronische Systemtechnik

**Thema: "Factory Acceptance Test des Prozessleitsystems PCS 7
und Anlagenfeld Simulation mit WinMOD"**

Eingereicht von: **Martin Bayer**

Angefertigt für: Actemium Controlmatic GmbH

Matrikel: E 2009

Ausgabetermin: 14. Juli 2014

Abgabetermin: 22. September 2014

Schulischer Betreuer: Herr Prof. Dr.-Ing. Ding

Betrieblicher Betreuer: Herr Dipl.-Ing. Jahn

.....
1. Prüfer

.....
2. Prüfer

FAT - Prozessleitsystem PCS 7 und Feldsimulation mit WinMOD

Inhalt:

Die Bachelorarbeit befasst sich mit der Erstellung einer universell nutzbaren Modulbibliothek im Simulationssystem WinMOD. Diese Bibliothek soll im Rahmen des Factory Acceptance Tests für die Evaluierung des Prozessleitsystems PCS 7 eingesetzt werden.

Themenschwerpunkte:

- Einführung in die Prozesssimulation
- Überblick Simulationssystem WinMOD
- Beschreibung der Softwaremodule
- Nutzungsbereiche der Softwaremodule
- FAT mit Softwaremodulen

Factory Acceptance Test des Prozessleitsystems PCS 7 mit Simulationssoftware WinMOD
Bayer, Martin
Hochschule Magdeburg – Stendal
Fachbereich Ingenieurwissenschaften und Industriedesign
Bachelorarbeit – Nr.: 20093335
88 S.; 74 B.; 61 Tab., 1 Anlage

Abstract

Das Prozessleitsystem PCS 7 beinhaltet softwaretechnische Lösungen für alle Hierarchieebenen der industriellen Automatisierung von der Unternehmensleitebene bis zur Anlagenfeldebene. Mit der Simulationssoftware WinMOD können Komponenten, Signale und Prozesse einer Industrieanlage nachgebildet werden. Zudem bietet WinMOD Schnittstellenlösungen für gängige Speicherprogrammierbare Steuerungen und eignet sich somit für den Factory Acceptance Test (FAT) von Prozessleitsystemen. Die FATs sollen sich hauptsächlich auf die Überprüfung von Signaladressierungen und von Ursache und Wirkung konzentrieren. Im Rahmen der vorliegenden Arbeit wurden ausgewählte PCS 7 – Projekte untersucht und auf deren Grundlage Aussagen getroffen wie zukünftig Simulationen effektiv erstellt werden können. Hierzu wurden mithilfe von WinMOD Simulationsmodule angelegt - welche reale Komponenten des Anlagenfeldes ersetzen – , ihr Aufbau beschrieben und in einer Bibliothek zusammengefasst. Im Anschluss wird auf die Verwendung der Bibliothek für verschiedene Anwendungsfälle eingegangen und eine Lösung präsentiert, welche das Erstellen einer Simulation für spezifische Projekte optimiert. Das Ergebnis zeigt auf, dass WinMOD sowohl neue Möglichkeiten der Durchführung für einen FAT liefert, wobei die Simulationserstellung nur einen geringen Mehraufwand bedeutet und sich die Durchführung der Simulation ergonomischer und kürzer gestaltet.

Factory Acceptance Test of process control system PCS 7 with simulation software
WinMOD
Bayer, Martin

Magdeburg – Stendal University of Applied Sciences
Department Engineering and Industrial Design
Bachelor thesis – No.: 20093335
88 p.; 74 f.; 61 t., 1 attachment

Abstract

The distributed control system PCS 7 contains software-based solutions for all levels of industrial automation from the enterprise management level down to the field level. To verify the functionality of created application software it is important to test it in a secure environment. With WinMOD it is possible to simulate components, signals and processes of an industrial plant. Further the WinMOD system includes interfaces to communicate with most programmable logic controllers. Therefore it is suitable for a factory acceptance test (FAT) of a distributed control system. In most cases the FAT will be used to proof signal allocations and the interaction of cause and effect. Some selected PCS 7 projects were used to get knowledge about how to compile simulations effectively. To replace real plant components there were made corresponding simulation modules with WinMOD which were collected in a library. It will be explained how to use this library and additionally presented a way to optimize the simulation creation process. The results show that WinMOD supplies more capabilities to realize a FAT. The accomplishing of the test is more ergonomic and efficient while the effort to create the simulation marginally creases.

Selbständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Bachelorarbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Ort, Datum

.....
Unterschrift des Verfassers

Inhaltsverzeichnis

1	Einleitung	8
2	WinMOD - Simulationssoftware	9
3	SIMATIC PCS 7 - Prozessleitsystem.....	10
4	Die Erstellung der Makrobibliothek.....	12
4.1	Makros	12
4.2	Aufbau PCS 7 – Projekte	14
4.2.1	Katalogisierung der Messstellen.....	14
4.2.2	Aktoren	15
4.2.3	Regelstrecken.....	16
4.3	Grundstruktur der Makros	18
4.3.1	Bedienfeld.....	18
4.3.2	Störverhalten.....	21
4.3.3	Betriebsartenumschaltung	24
4.3.4	Befehlseingabe am Makro	26
4.4	Umwandlung und Übertragung digitaler und analoger Werte.....	28
4.4.1	Analog – Digital – Wandlung.....	28
4.4.2	Analog – Digital - Wandlung in WinMOD	29
4.4.3	Externe Digitalformate	32
4.5	Makrovorlagen.....	33
4.5.1	Makrovorlagen der Aktoren	33
4.5.2	Regelstrecken.....	65
4.6	Anlegen der Makrobibliothek als Simulationsdatei.....	67
5	Realisierung eines FAT	68
5.1	Aufbau der Anlage.....	68
5.2	FAT am Beispiel eines Stellventils.....	68
5.2.1	Vorbereitungen und Testvarianten	69
5.2.2	Arten der Simulationserstellung und Simulationsarten.....	71
5.3	Gesamtauswertung.....	85
6	Zusammenfassung.....	87
7	Literaturverzeichnis.....	V

Formelzeichen

Formelzeichen	Benennung	Einheit
h	Höhe	mm
I	Stromstärke	mA
T	Temperatur	°C
t	Zeit	s
f	Frequenz	Hz

Abkürzungen

Abkürzung	Bedeutung
FAT	Factory Acceptance Test
SAT	Sight Acceptance Test
PCS	Process Control System
SPS	Speicherprogrammierbare Steuerung
I/O	Input/Output
PROFIBUS	Process Field Bus
AS	Automatisierungssystem
OS	Operatorsystem
CFC	Continuous Function Chart
FB	Funktionsbaustein
FC	Funktion
BuB	Bedienen und beobachten
HMI	Human Machine Interface
CPU	Central Processing Unit
APL	Advanced Process Library
PPO	Parameter – Prozessdaten – Objekt
PZD	Prozessdaten
VIK	Verband der Industriellen Energie- und Kraftwirtschaft
p.e.	pulse enable
RFG	Rampenfunktionsgenerator
NS	not stored
HIW	Hauptistwert
HSW	Hauptsollwert
BHKW	Blockheizkraftwerk
PID	proportional, integral, differenzial
UW	Ursache – Wirkung
EA	Engineering Assistenz

Abbildungsverzeichnis

Abb. 1: WinMod – System	9
Abb. 2: PCS 7, Komponenten	10
Abb. 3: Kommunikation Feldgeräte – Funktionsbausteine	11
Abb. 4: Makro, Entwicklungsansicht	13
Abb. 5: Makro, Simulationssicht	13
Abb. 6: Schema Regelstrecke	16
Abb. 7: Bedienfeld, Bereichseinteilung	18
Abb. 8: Bedienfeld, Bedienelemente	19
Abb. 9: Eingangssignal, Operanden	20
Abb. 10: Eingangssignal, Verschaltung	20
Abb. 11: Ausgangssignal, Operanden	20
Abb. 12: Ausgangssignal, Verschaltung 1	21
Abb. 13: Ausgangssignal, Verschaltung 2	21
Abb. 14: Störmeldung	22
Abb. 15: Störmeldung, Operanden	23
Abb. 16: Reset, Operanden	23
Abb. 17: Störung, Verschaltung	23
Abb. 18: Betriebsartenumschaltung	24
Abb. 19: Betriebsartenumschaltung 1, Operanden	25
Abb. 20: Betriebsartenumschaltung 1, Verschaltung	25
Abb. 21: Betriebsartenumschaltung 2, Operanden	26
Abb. 22: Betriebsartenumschaltung 2, Verschaltung	26
Abb. 23: Befehlseingabe, Fernbetrieb	27
Abb. 24: Befehlseingabe, Lokalbetrieb	27
Abb. 25: Befehlseingabe, Lokalbetrieb über Automatisierungssystem	27
Abb. 26: Analogwert – Verarbeitung	28
Abb. 27: Analogwert – Verarbeitung Beispielgrößen	29
Abb. 28: Analoge Stellgröße	29
Abb. 29: Analogwert – Verarbeitung WinMOD	30
Abb. 30: 2/2 Wegeventil	33
Abb. 31: Makro, AufZuVentil	35
Abb. 32: WinMOD – Element, Stellantrieb	36
Abb. 33: Stellventil	37
Abb. 34: 2/2 – Wegeventil, Gehäuseausführungen	38
Abb. 35: Stellventil, Makro	39
Abb. 36: Stellventil, Istwertbildung	40
Abb. 37: Motor_Dir, Makro	43
Abb. 38: Motor_Dir, Motorstart u. Zeitverzögerung	44
Abb. 39: Motor_Dir, Stromwert	44
Abb. 40: SIMCOCODE_Dir, Makro	46

Abb. 41: SIMOCODE, Ein/Ausschalten	47
Abb. 42: SIMOCODE, Notstart	48
Abb. 43: SIMOCODE, Störverhalten.....	48
Abb. 44: Prozessdaten – Bereiche des Standardtelegramm 20	49
Abb. 45: Zustandsdiagramm aus PROFIdrive.....	52
Abb. 46: PROFIdrive, Sollwertverarbeitung.....	53
Abb. 47: Rampenverläufe.....	54
Abb. 48: MotorSpdC_Tele1_20	55
Abb. 49: MotorSpdC_Tele1_20, Zustandsdiagramm	58
Abb. 50: MotorSpdC_Tele1_20, State Machine 1	59
Abb. 51: MotorSpdC_Tele1_20, State Element, Schritt3	59
Abb. 52: NS Befehl	59
Abb. 53: MotorSpdC_Tele1_20, Schrittkette1, Schritt3	60
Abb. 54: MotorSpdC_Tele1_20, Sollwertverarbeitung	60
Abb. 55: Rampenzeiten, Verschaltung.....	61
Abb. 56: MotorSpdC_Tele1_20, Drehrichtungsumkehr	62
Abb. 57: MotorSpdC_Tele1_20, Drehrichtungsumkehr, Verschaltung.....	63
Abb. 58: MotorSpdC_Tele1_20, Rampenfunktion	64
Abb. 59: MotorSpdC_Tele1_20, Prozesswerte	64
Abb. 60: Tank, Makro	65
Abb. 61: Tank, Verschaltung.....	65
Abb. 62: Temperatur, Makro	66
Abb. 63: Temperatur, Verschaltung	66
Abb. 64: Schema Anlagenteil Separator.....	68
Abb. 65: Eigenschaftsdialog Operand LCV211003	73
Abb. 66: Signalfluss- und Verarbeitung in Signalsimulation.....	75
Abb. 67: Bildbaustein, PID – Regler LIC211004	75
Abb. 68: Füllstandsregelkreis um Stellventil LCV211003	75
Abb. 69: WinMOD Simulationselemente, Signalsimulation	76
Abb. 70: Antriebs-/Messsystemsimulation und Prozesssimulation	77
Abb. 71: Makro VlvAn, Tank	77
Abb. 72: Simulationsfenster starten	78
Abb. 73: Makro Tank, Simulationsvorbereitung.....	78
Abb. 74: Simulationselemente, Test Regelkreis.....	78

Tabellenverzeichnis

Tabelle 1: Makro, Entwicklungsansicht, Erläuterungen	13
Tabelle 2: Makro, Simulationsansicht, Erläuterungen	14
Tabelle 3: Bedien- und Beobachtungs – Funktionsbausteine der PCS 7 – Projekte	15
Tabelle 4: Makrovorlagen zur Antriebssimulation.....	16
Tabelle 5: Makrovorlagen zur Regelstreckensimulation.....	18
Tabelle 6: Bedienfeld, Bereichseinteilung, Erläuterungen	18
Tabelle 7: Bedienfeld, Bedienelemente; Erläuterung.....	19
Tabelle 8: Vergleich reale und simulierte Störung.....	21
Tabelle 9: Störmeldung, Erläuterung.....	22
Tabelle 10: Betriebsartenumschaltung, Erläuterung	25
Tabelle 11: Normierung Messgröße - Digitalwert	29
Tabelle 12: Wertnormierung WinMOD	30
Tabelle 13: Parametrierung externes Analogformat „Siemens 0 ... 27648“.....	31
Tabelle 14: interne Analogformate.....	31
Tabelle 15: Parametrierung externes Digitalformates „Big Endian Word“	32
Tabelle 16: 2/2 – Wege – Schaltventil Signaleingänge.....	34
Tabelle 17: 2/2 – Wege – Schaltventil Signalausgänge	34
Tabelle 18: ausgewählte Parameter des Funktionsbausteins VlvL	35
Tabelle 19: Schaltventil, Operanden	35
Tabelle 20: AufZuVentil, Verschaltung	36
Tabelle 21: Stellventil, Signaleingänge	37
Tabelle 22: Stellventil, Ausgangssignale	38
Tabelle 23: ausgewählte Parameter des Funktionsbausteins VlvAnL	39
Tabelle 24: Stellventil, Operanden	40
Tabelle 25: Direktstarter (Klemme), Signaleingänge.....	41
Tabelle 26: Direktstarter, Signaleingänge	41
Tabelle 27: Direktstarter (Klemme), Parameter	42
Tabelle 28: Motor_Dir, Operanden	43
Tabelle 29: SMC_Dir, Steuerwort.....	45
Tabelle 30: SMC_Dir, Statuswort	45
Tabelle 31: SIMOCODE_Dir, Analogeingänge.....	46
Tabelle 32: SIMOCODE_Dir, Operanden	46
Tabelle 33: PROFIdrive Telegramm 1/20, Analogwerte	50
Tabelle 34: PROFIdrive Telegramm 1/20, Steuerwort	50
Tabelle 35: PROFIdrive Telegramm 1/20, Zustandswort	51
Tabelle 36: PROFIdrive Telegramm 1/20, NAMUR - Meldungen	51
Tabelle 37: MotorSpdC_Tele1_20, Operanden.....	55
Tabelle 38: MotorSpdC_Tele1_20, Netz 1 Schritt 3.....	60
Tabelle 39: MotorSpdC_Tele1_20, Sollwertverarbeitung, Erläuterung	61
Tabelle 40: Rampenzeiten, Verschaltung, Erläuterung	61

Tabelle 41: MotorSpdC_Tele1_20, Drehrichtungsumkehr, Operanden	63
Tabelle 42: Tank, Operanden	65
Tabelle 43: Tank, Verschaltung, Erläuterung.....	65
Tabelle 44: Temperatur, Erläuterungen.....	66
Tabelle 45: Temperatur, Verschaltung, Erläuterung	67
Tabelle 46: Stellventil LCV211003, Signale.....	69
Tabelle 47: Ursache – Wirkungs - Diagramm.....	70
Tabelle 48: Periphere Signale des Stellventils LCV211003	71
Tabelle 49: Eigenschaften Operand LCV211003.....	72
Tabelle 50.: Operandenliste, Test Stellventil.....	74
Tabelle 51: WinMOD – Kürzel für Signaltypen	74
Tabelle 52: Engineering – Liste, Test Stellventil (Teil 1)	80
Tabelle 53: Engineering – Liste, Test Stellventil (Teil 2).....	80
Tabelle 54: Messstellenliste, EA - Skript	82
Tabelle 55: EA Konfiguration (1)	83
Tabelle 56: EA Konfiguration (2)	83
Tabelle 57: Engineering – Liste, automatisch	84
Tabelle 58: Simulationserstellung – Simulationsart.....	85
Tabelle 59: Simulationserstellung – Simulationsart, Erläuterung.....	85
Tabelle 60: Simulationsart - Testvariante.....	86
Tabelle 61: Simulationsart – Testvariante, Erläuterung	86

1 Einleitung

Ein Unternehmen, welches Lösungen im Bereich der Automatisierungstechnik vorwiegend für verfahrenstechnische Anlagen anbietet, möchte zukünftig Factory Acceptance Tests (FAT) an dem Prozessleitsystem PCS 7 der Siemens AG mit dem Anlagenperipherie- und Prozesssimulations – Programm WinMOD der Firma Mewes & Partner GmbH durchführen. Der FAT birgt viele Vorteile für den Projektteur des Programms als auch für den Anlagenbetreiber. Insbesondere dient er der Qualitätssicherung der angebotenen Automatisierungslösung. So lassen sich Inbetriebnahmezeiten einsparen. Zum einen verkürzen sich die Stillstandszeiten im Falle von Anlagenerweiterungen oder -umbauten und zum anderen sinken die Personalkosten bei der Inbetriebnahme. Dem Anlagenbetreiber entstehen weniger Kosten und die Konkurrenzfähigkeit des Anbieters der Automatisierungslösung steigt gegenüber seinen Mitbewerbern. Zudem kann ein FAT gegenüber einem Sight Acceptance Test (SAT), also dem Vororttest der SPS – Anwendersoftware, als eher risikoarm eingestuft werden. Ein sich einem positiven FAT anschließender SAT birgt weniger Gefahren für Mensch, Umwelt und Anlage.

Für den Werkstest des Anwenderprogrammes ist eine Simulation unerlässlich. Auch bisher wurden durch das Unternehmen FATs vorgenommen. Während die Erstellung der Simulation keine Schwierigkeiten bereitete, war die Durchführung des Tests kompliziert und barg viele Fehlerquellen. Auch konnten wesentliche Bereiche des Anwenderprogrammes nicht getestet werden. Mit der Verwendung von WinMOD und der Erstellung von Simulationsbausteinen, welche in einer Bausteinvorlagenbibliothek zusammengefasst werden, soll sowohl die Simulationstiefe erweitert als auch der zeitliche Aufwand des FATs verringert werden.

Um diesen Anforderungen gerecht zu werden, wird dem Projektteur der Simulation für einen FAT ein möglichst leichter Einstieg geboten. Zugleich sollen die Komponenten der Bausteinvorlagenbibliothek projektuniversell einsetzbar sein. Der Bedarf einer Erweiterung dieser Bibliothek kann selbstverständlich nicht ausgeschlossen werden. Die Verwendung der Bausteine soll benutzerfreundlich und die Erstellung von geeigneten Simulationen unkompliziert und effektiv gestaltet werden.

2 WinMOD - Simulationssoftware

WinMOD ist ein System zur Anlagenperipherie- und Prozesssimulation. Mit WinMOD lässt sich eine Simulationsumgebung erstellen und unter Echtzeitbedingungen mit einem Automatisierungssystem verbinden. Die Anlagenfeld – Peripherie lässt sich in Teilen oder vollständig virtuell ersetzen. Somit kann eine Umgebung geschaffen werden, in welcher ein SPS – Anwenderprogramm getestet werden kann. [1]

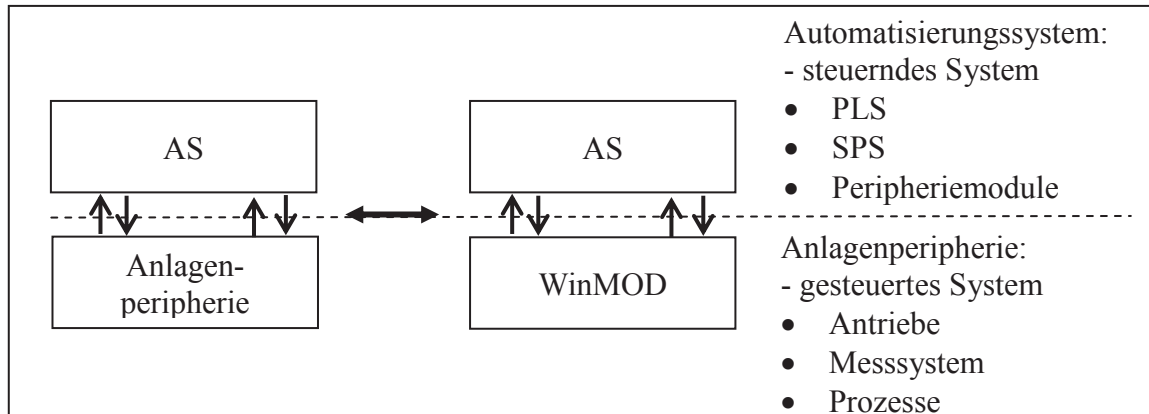


Abb. 1: WinMod – System

WinMOD – Konfigurationspakete stellen die Systemkopplung zwischen WinMOD und dem angeschlossenen Automatisierungssystem bereit. Die Konfigurationen sind dabei systembezogen und werden für alle gängigen Schnittstellen und Automatisierungssysteme angeboten. Für die Erstellung dieser Arbeit wurde die Konfiguration A750 zur Systemkopplung mit der SPS – Simulation S7 – PLCSIM genutzt. Die eigentliche Simulationsumgebung bleibt von der Schnittstelle weitgehend unbeeinflusst. [1]

Die Simulationstiefe lässt sich mit WinMOD frei konfigurieren. Die I/O – Abbildung oder Signalsimulation, bei welcher AS – Ausgänge beobachtet und AS – Eingänge simuliert werden, stellt dabei die einfachste Nutzung dar. Mit der Simulation von Antrieben und Messsystemen erfährt der Anwender bereits nach erfolgten Ansteuerungen Rückmeldungen aus dem simulierten System. Der zu simulierende Antrieb muss vom Anwender selber aufgebaut und angelegt werden. Durch die Schaffung von Verbindungen zwischen Antrieben und Messsystemen wird eine Prozesssimulation erreicht. Dies erlaubt das Testen von einfachen Regelkreisen bis hin zur Simulation aller Wechselwirkungen der Gesamtanlage. Zur Simulation von wiederkehrenden technologischen Elementen einer Anlage bietet WinMOD die Möglichkeit des Anlegens von Bausteinvorlagen, sogenannten Makrovorlagen (siehe 4.1 Makros, S.12). Sie werden in verschiedenartig strukturierten Bibliotheken abgelegt (siehe Kapitel 4.6). Mithilfe der Engineering Assistenz (siehe Kapitel 5.2.2.2.3) lässt sich aus diesen Elementen mit wenig Aufwand die Antriebssimulation einer gesamten Anlage erstellen.

3 SIMATIC PCS 7 - Prozessleitsystem

Der Einsatz eines Prozessleitsystems erlaubt das Führen einer verfahrenstechnischen Anlage. In ihm vereinen sich prozessnahe Komponenten, Bedien- und Beobachtungsstationen (auch Operatorstationen, kurz OS) sowie Engineering Komponenten. [2, p. 191]

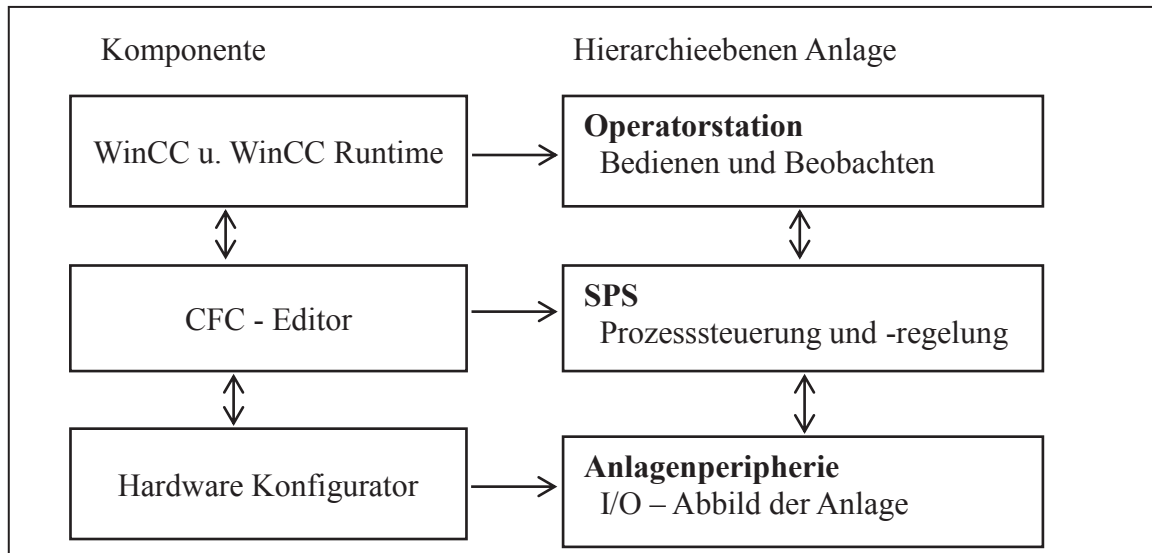


Abb. 2: PCS 7, Komponenten

Die prozessnahen Komponenten, z.B. eine SPS, kommunizieren direkt mit den Feldgeräten, steuern Aktoren an und nehmen Rückmeldungen des Anlagenfeldes auf. Mit der Applikation WinCC Runtime wird die Schnittstelle zwischen dem Bediener und dem Automatisierungssystem realisiert. Dieser erhält Einblick in alle visualisierten Systemzustände, Prozesse und Alarmer. Das Engineering definiert unter anderem die Schnittstellenadressierung zwischen Feldgeräten und der SPS. Hier wird das SIMATIC Tool „Hardware Konfigurator“ eingesetzt. Die Erstellung des SPS – Anwenderprogrammes findet hauptsächlich durch die Nutzung des grafisch arbeitenden Continuous Function Chart – Editors (CFC – Editor) statt. Hier werden speicherfähige „Funktionsbausteine“ (FB) sowie nicht speicherfähige „Funktionen“ (FC) zu CFC – Plänen verschaltet.

Funktionsbausteine werden in Familien eingeteilt. Zu nennen sind an dieser Stelle die Familien Channel, Drives und Control.

Channel – Kanaltreiberbausteine

Kanaltreiberbausteine dienen der Bearbeitung, Interpretation, Prüfung und Weiterleitung binärer, digitaler oder digitalisierter analoger Ein- und Ausgangssignale. Sie stehen logisch in direktem Kontakt zu den Feldgeräten.

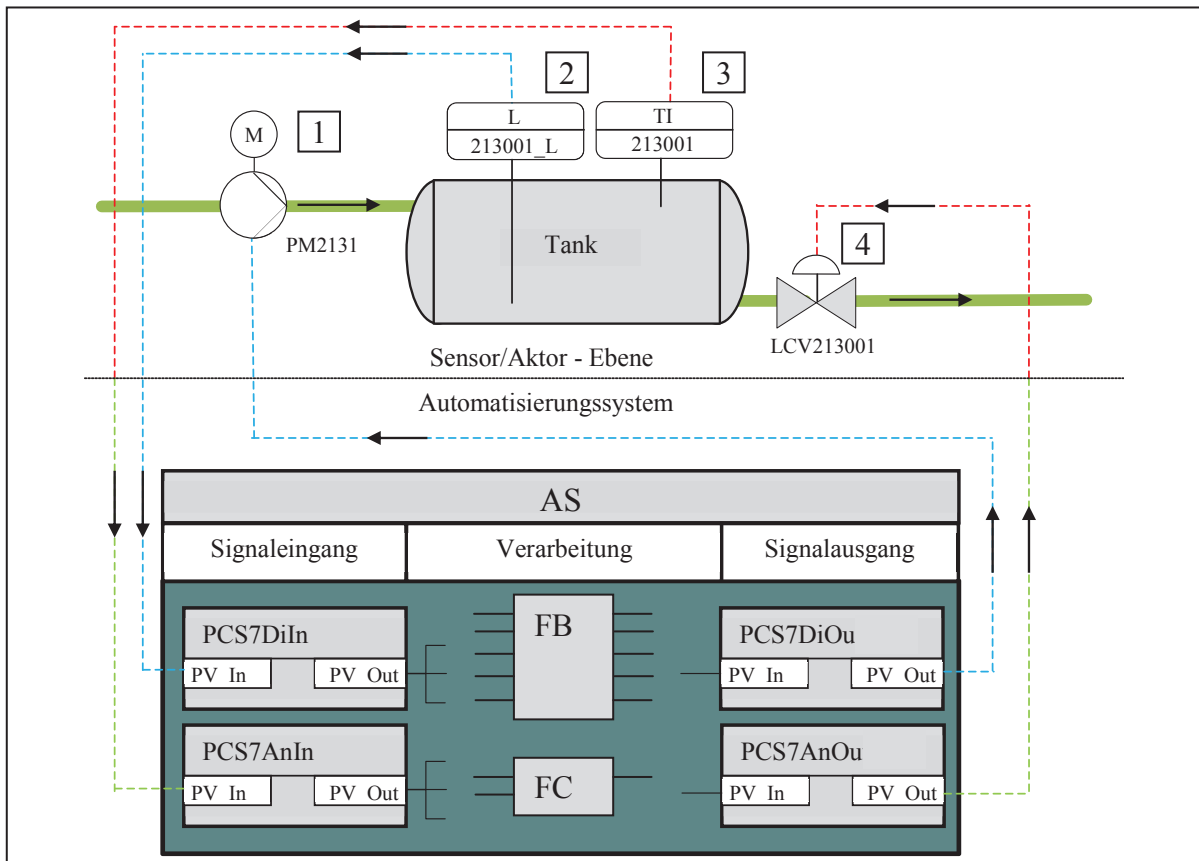


Abb. 3: Kommunikation Feldgeräte – Funktionsbausteine

- | | |
|---|----------------------------------|
| 1: Binärsignal schaltet Pumpe ein und aus | ----- Analogwert |
| 2: binäre Füllstandsmessung | ----- digitalisierter Analogwert |
| 3: analoge Temperaturmessung | ----- Binärwert |
| 4: Analogansteuerung eines Stellventils | |

Drives – Motor- und Ventilbausteine

Diese Funktionsbausteine sind bedien- und beobachtbar (BuB). Das bedeutet, dass sich ihr Status und ihre Bedienfunktionen in der Operatorebene wiederfinden. Ihre Visualisierung in der WinCC Runtime wird als Bildbausteinsymbol oder in der Detailansicht als Bildbaustein bezeichnet. Sie dienen der Ansteuerung von Motoren und Ventilen, werten deren Rückmeldungen aus und bieten umfangreiche Einstellmöglichkeiten zu Betriebsarten, Fehlererkennung und Meldung sowie Verriegelungsoptionen.

Controls - Reglerbausteine

Durch Abgleich von Soll- und Istwerten führen diese Funktionsbausteine den Stellwert eines Aktors nach. Durch eine geeignete Wahl der Regelparameter lässt sich eine gewünschte Regelgüte erzielen. Auch diese Bausteine verfügen über Bedien- und Beobachtungsfunktionen.

4 Die Erstellung der Makrobibliothek

4.1 Makros

Funktionale Einheiten, mit einem durch Signalelemente gebildeten Ein- und Ausgabeteil sowie einem durch Verschaltung von Simulationselementen realisierten Verarbeitungsteil, werden in WinMOD „Makros“ genannt. Einsatz finden sie insbesondere bei wiederkehrenden Aufgaben. Abgespeichert als Vorlage in einer Bibliothek sind sie projektübergreifend nutzbar. Alle aus der Makrovorlage erstellten Makros sind vom selben „Makrotypen“. [1, p. 9.1]

Neben dem verarbeitenden enthält ein Makro einen bedien- und beobachtbaren Teil: das Bedienfeld. Hierdurch kann das Makro als virtuelles Gerät genutzt werden.

Aufbau eines Makros

Im Ein- und Ausgabebereich des Makros können Signalelemente platziert werden. Signalelemente dienen der Anzeige von Signalzuständen in unterschiedlichen Darstellungen sowie wichtiger Operandeninformationen. Sie werden in die Typen binär, digital und analog unterteilt. Unter ihrer Verwendung im Ein- bzw. Ausgangsbereich eines Makros werden die sogenannten „Makrosignale“ erzeugt. Signalelemente können in den sogenannten „Forcemodus“ gesetzt werden. Aus einem lesenden Signalelement wird dann ein schreibendes. Das geforcte Signalelement erhält eine Schwarzfärbung, um es deutlich von nicht – geforcten unterscheidbar zu machen.

Durch die Verschaltung von mathematischen und technischen Elementarfunktionen vom ODER – Baustein bis zum Verzögerungsglied 1. Ordnung, den sogenannten „Simulationselementen“, erhält das Makro seine funktionalen Eigenschaften. Ihre Darstellung erfolgt grafisch und ihre Verschaltung sowohl grafisch als auch textuell.

Eine Schnittstelle zwischen dem Anwender der Simulation und dem Makro kann auf zweierlei Arten hergestellt werden. Zum einen über die bereits erwähnten Signalelemente und zum anderen durch Elemente des Human Machine Interface (HMI), wie Leuchttaster und Schalter. Das Bedienen über HMI – Elemente gestaltet sich etwas übersichtlicher als über Signalelemente. Um die Bedienung über Signalelemente zur ermöglichen, müssen diese in den Force – Modus gesetzt werden. Diese Bedienart soll allerdings nachrangig genutzt werden und nur bestimmten Testsituationen vorbehalten bleiben oder um nicht beschalteten Makroeingängen einen gewünschten Signalzustand zuzuweisen.

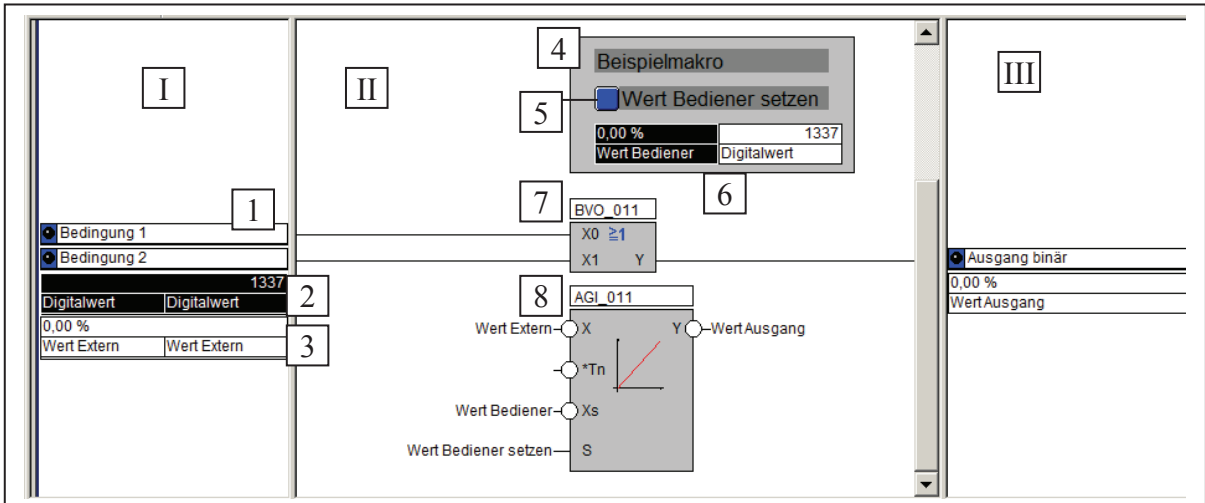


Abb. 4: Makro, Entwicklungsansicht

Tabelle 1: Makro, Entwicklungsansicht, Erläuterungen

Bereiche und Elemente	Erläuterung
(I) Eingangsbereich	Platzierung von Signalelementen für Makroeingänge
(II) Arbeitsbereich	Verschaltung funktionaler Elemente; Erstellung des Bedienfeldes
(III) Ausgangsbereich	Platzierung von Signalelementen für Makroausgänge
(1) Binärsignalelemente	Signalelement für binäre Signale
(2) Digitalsignalelement	Signalelement für digitale Signale
(3) Analogsignalelement	Signalelement für analog Signale
(4) Bedienfeld	Platzierung von Elementen, welche in der Simulationsansicht erscheinen
(5) Bedienelement	Befehlseingabe über Bedienfeld
(6) Signalelemente in Bedienfeld	Signalein-/ oder -ausgabe; kein Wechsel lesend/schreibend möglich (Forcemodus)
(7) Simulationselement 1	Beispiel für grafische Verschaltung
(8) Simulationselement 2	Beispiel für Verschaltung mit Operanden

Dem Anwender des Makros stellt sich dieses in der folgenden Ansicht dar. Hier finden sich die Signalelemente in ihrer Form als Makrosignale wieder. Die an die Makrosignale anfügbaren Signalelemente sind wiederum leer, da ihnen noch Operanden zugewiesen werden müssen.

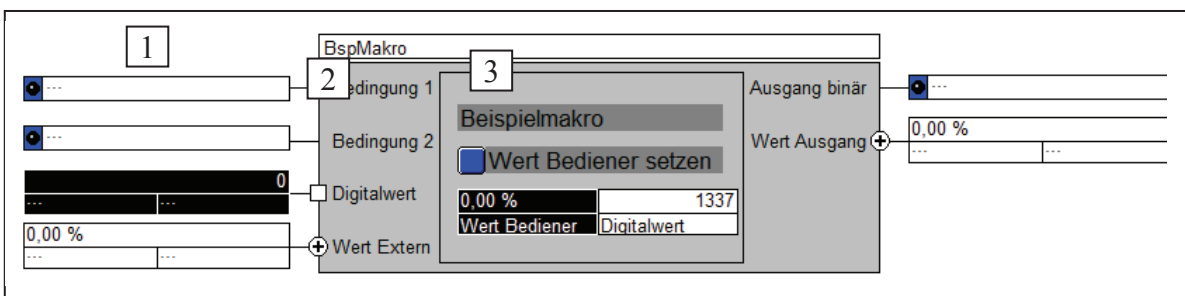


Abb. 5: Makro, Simulationssicht

Tabelle 2: Makro, Simulationsansicht, Erläuterungen

Elemente	Erläuterung
(1) Signalelemente	entsprechen Makrosignalen, forcen (de-) aktivierbar (Anzeigeelement oder Befehlseingabe)
(2) Makrosignale	Schnittstelle des Makros (binär, digital, analog)
(3) Bedienfeld	Bedienung und Beobachtung des Makros

4.2 Aufbau PCS 7 – Projekte

Zunächst sollte ermittelt werden, welcher Makros es bedarf und welche Grundstruktur sie aufweisen sollten. Hierzu wurden die CFC – Pläne zweier als repräsentativ eingestufte Projekte untersucht. Besonderes Interesse galt dabei solchen Plänen, die mit Feldgeräten (z.B. Motoren, Druckmessern) kommunizieren. Unabhängig davon, ob diese Pläne nur zur Ansteuerung oder nur bzw. auch zur Auswertung und Anzeige von Rückmeldungen Verwendung finden, werden diese in PCS 7 als „Messstelle“ bezeichnet.

Üblicherweise wird für jedes Feldgerät genau ein CFC – Plan verwendet, was sich aus der Projektierungsstruktur ergibt jedem CFC – Plan nur einen bedien- und beobachtbaren Baustein zuzuordnen. Diese Bausteine wurden so entwickelt, dass sie möglichst alle Funktionalitäten eines Feldgerädetypen, zum Beispiel eines nicht – drehzahlgeregelten direktstartenden Motors, abdecken. Für alle üblichen Beschaltungen eines solchen Motors wird somit nur ein bestimmter Motor - Funktionsbaustein benötigt. Somit wird jedem bedien- und beobachtbaren PCS 7 Funktionsbaustein, welcher ein Feldgerät repräsentiert, ein WinMOD – Makrotyp zugeordnet.

Bei der Erstellung der Makros soll die Messstelle als Definition der Grenzen eines Feldgerätes gelten. Die Makrovorlage erhält demnach alle Ein- und Ausgänge, welche einer Messstellenart der PCS 7 – Projekte zugeordnet werden konnten.

Die Regelung von Prozessen nehmen Regler – Funktionsbausteine vor, welche ebenfalls je einem CFC – Plan zugeordnet sind. Sie sind ebenfalls bedien- und beobachtbar. Das Makro soll der Art der zu Regelstrecke entsprechen.

4.2.1 Katalogisierung der Messstellen

Nach den verwendeten Bedien- und Beobachtungs – Funktionsbausteinen lassen sich die in den Projekten gefundenen Messstellen nach folgender Tabelle einteilen:

Tabelle 3: Bedien- und Beobachtungs – Funktionsbausteine der PCS 7 – Projekte

Messstelle	BuB – FB Familie	BuB - FB	Einsatzzweck
Anzeige	Monitoring	MonDiL	Binärmelder; z.B. Füllstandsgrenze
		MonAnL/MonAnS	Analogwertanzeige; z.B. Füllstandsanzeige
Aktoren	Drives	MotL	nicht – drehzahleregelter direktstartender Motor mit einer Drehgeschwindigkeit
		SMC_Dir	Motorsteuerung „SIMOCODE“ für Motoren mit Festdrehzahl
		MotSpdCL	drehzahleregelter Motor mit zwei Drehrichtungen
		VlvL	Auf/Zu – Schaltventil
		VlvAn	Stellventil
Regelung	Control	PIDConL	kontinuierlicher PID – Regler
Bedienung	Operate	OpAnL	Analogansteuerung; z.B. für Stellventil ohne Rückmeldung
		OpDiL	Binäransteuerung; z.B. Starten eines Motors ohne Rückmeldung

Die Messstellen „Anzeige“ können bei der folgenden Betrachtung zur Makroerstellung außen vorgelassen werden. Diese Feldgeräte enthalten simuliert keinen verarbeitenden Teil, da sie lediglich einen Signalstatus wiedergeben. Ähnlich verhält es sich mit den Messstellen „Bedienung“. Diese finden z.B. bei der Ansteuerung von Geräten Einsatz, welche kein Rückmeldesignal ausgeben.

Die eigentliche Aufmerksamkeit gilt den Aktoren und der Regelung.

4.2.2 Aktoren

Der MotSpdCL kommuniziert mit einem Frequenzumrichter. Die Anbindung erfolgt in den vorliegenden Projekten über den Feldbusstandard PROFIBUS unter Verwendung eines PROFdrive Telegrammprofils. Zur Nutzung des Telegrammprofils kam der Kanaltreiberbaustein FbDrive zum Einsatz.

Der SMC_Dir wurde speziell zur Kommunikation mit der Siemens Motorsteuerung SIMOCODE entwickelt, um einen Asynchronmotor als Direktstarter zu betreiben. Steuer- und Meldedaten werden zwischen der SPS und der Motorsteuerung ausschließlich über PROFIBUS oder PROFINET mittels eines bestimmten Telegrammprofils ausgetauscht. [3, p. 47]

Folgende Makrovorlagen müssen erstellt werden, um eine Simulation aller Aktoren der Projekte zu ermöglichen:

Tabelle 4: Makrovorlagen zur Antriebssimulation

Aktor	Makrotyp
2/2 - Wege – Schaltventil	AufZuVentil
Stellventil	Stellventil
nicht – drehzahl geregelter direktstartender Motor mit einer Drehgeschwindigkeit (Klemme)	Motor
nicht – drehzahl geregelter direktstartender Motor mit einer Drehgeschwindigkeit (SIMOCODE)	SIMOCODE_Dir
drehzahl geregelter Motor mit zwei Drehrichtungen (PROFIdrive – Profil Telegramm 1/20)	MotorSpdC_Tele1_20

4.2.3 Regelstrecken

Um Regelungen testen zu können, müssen Regelstrecken entworfen werden. Da nur das grundsätzliche Funktionieren des Regelns überprüft werden soll, können die Regelstreckenmodelle sehr stark vereinfacht werden. Dies erlaubt eine häufige Verwendung einmalig erstellter Regelstrecken.

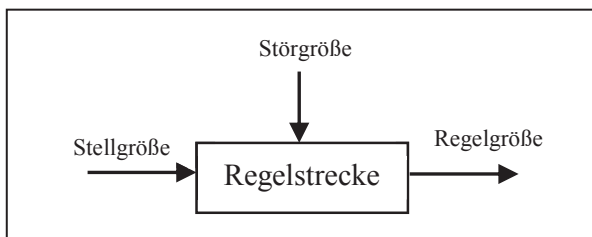


Abb. 6: Schema Regelstrecke

Folgende Arten von Regelungen treten in den Projekten auf:

- Temperaturregelung
- Durchflussregelung
- Füllstandregelung
- Druckregelung

Temperaturregelstrecke:

Stellgröße: z.B. Ventilöffnung oder im weiteren Sinne eine **Energiezufuhr/-abfuhr** in proportionaler Abhängigkeit zur Ventilöffnung

Regelgröße: Temperatur

Störgröße: Umgebungstemperatur

Infolge einer Energiezufuhr/-abfuhr steigt/sinkt die Temperatur bis sie einen statischen Endwert erreicht. Ohne Energiezufuhr oder –abfuhr nähert sich die Regelgröße wieder der Umgebungstemperatur an.

Durchflussregelstrecke:

Stellgröße: z.B. Ventilöffnung oder Drehzahl eines Motors

Regelgröße: Stoffdurchfluss

Störgröße: keine

Auf Störgrößeneinflüsse wie Viskositätsschwankungen oder Leitungsquerschnittsänderungen soll bei dieser Betrachtung verzichtet werden. Der Durchfluss verhält sich im einfachsten Fall proportional zur Stellgröße. Somit kann die Stellgröße direkt als Regelgröße zurückgeführt werden, was die Verwendung einer Regelstrecke überflüssig macht. Der einzige Einstellparameter wäre der Proportionalwert. Ein hinreichender Funktionstest der Regelung kann jedoch bereits durch Änderung des Sollwertes erreicht werden.

Füllstandsregelstrecke:

Stellgröße: z.B. Ventilöffnungen oder im weiteren Sinne ein **Stoffzufluss/-abfluss** in proportionaler Abhängigkeit zur Ventilöffnung

Regelgröße: Füllstand

Störgröße: Zufluss bzw. Abfluss

Infolge eines Stoffzuflusses/-abflusses steigt ein Füllstand. Der Abfluss soll nicht füllstandsabhängig sein.

Druckregelstrecke:

Stellgröße: z.B. Ventilöffnung oder im weiteren Sinne ein **Stoffzufluss** in Abhängigkeit von der Ventilöffnung und der Druckdifferenz

Regelgröße: Druck

Störgröße: Abfluss

Infolge eines Stoffzuflusses steigt ein Druck. Zur Vereinfachung soll angenommen werden, dass Stoffzufluss und Stoffabfluss druckunabhängig sind. Somit lassen sich die Füllstandsregelstrecke und die Druckregelstrecke zu einem Regelstreckentyp zusammenfassen.

Folgende Makrovorlagen müssen erstellt werden, um eine Simulation aller Regelstrecken der Projekte zu ermöglichen:

Tabelle 5: Makrovorlagen zur Regelstreckensimulation

Regelstrecke	Makrotyp
Temperatur	Temperatur
Füllstand/Druck	Tank

4.3 Grundstruktur der Makros

Bevor auf den Aufbau der einzelnen Makrotypen eingegangen wird, sollen zunächst wiederkehrende Elemente erläutert werden.

4.3.1 Bedienfeld

Das Bedienfeld ist Teil der Visualisierung eines Makros im Simulationsfenster. Mithilfe des Bedienfeldes kann ein virtuelles Gerät erzeugt werden. Dem Anwender einer Simulation ermöglicht es, Prozesse eines Makros bedien- und beobachtbar zu machen. Alle für die spätere Anwendung in der Simulation als relevant erscheinenden Befehlseingabe- und Beobachtungsfunktionen sowie weitere statische Informationen (erklärender Text, Symbole usw.) erscheinen im Bedienfeld. Erstellt wird es im Arbeitsbereich des Makros.

Das Bedienfeld teilt sich in 3 Bereiche ein:

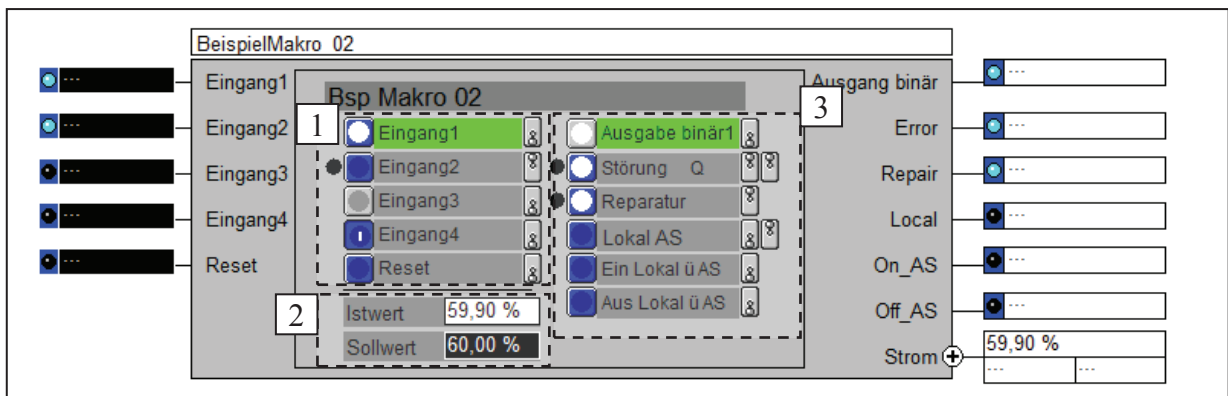


Abb. 7: Bedienfeld, Bereichseinteilung

Tabelle 6: Bedienfeld, Bereichseinteilung, Erläuterungen

Bereich	Erläuterung
(1) Befehlseingabe	Befindet sich das virtuelle Gerät in den Betriebsarten „Lokal“ oder „Lokal über AS“ (siehe 4.3.3 Betriebsartenumschaltung S. 24) so erfolgt hier das Setzen von Befehlen. Des Weiteren geben Leuchttaster und Textfeld den Status des jeweiligen Befehlseinganges wieder. Sie repräsentieren zugleich alle Makrosignaleingänge.
(2) Analogwerte	Hier findet das Eingeben von Analogwerten zur Parametrierung oder zur Sollwertvorgabe statt. Ein eingegebener Sollwert ist nur in der Betriebsart „Lokal“ oder „Lokal über AS“ (siehe 4.3.3 Betriebsartenumschaltung S. 24) aktiv. Wichtige Istwerte, welche ebenfalls am Makroausgang erscheinen, werden hier angezeigt.

Bereich	Erläuterung
(3) Meldungen	In diesem Bereich aufgeführte Signale erscheinen ebenfalls als Makrosignalausgänge. Eingaben in diesem Bereich werden direkt auf die Ausgänge geschrieben. Diese Eingaben haben keine direkten Auswirkungen auf das Verhalten des Makros. Eine Ausnahme bildet hier die Betriebsartenumschaltung (siehe 4.3.3 Betriebsartenumschaltung S. 24). Die Auswahl einer Lokalbetriebsart wird zum einen am Makroausgang „Local“ ausgegeben und bewirkt zum anderen eine Betriebsartenumschaltung.

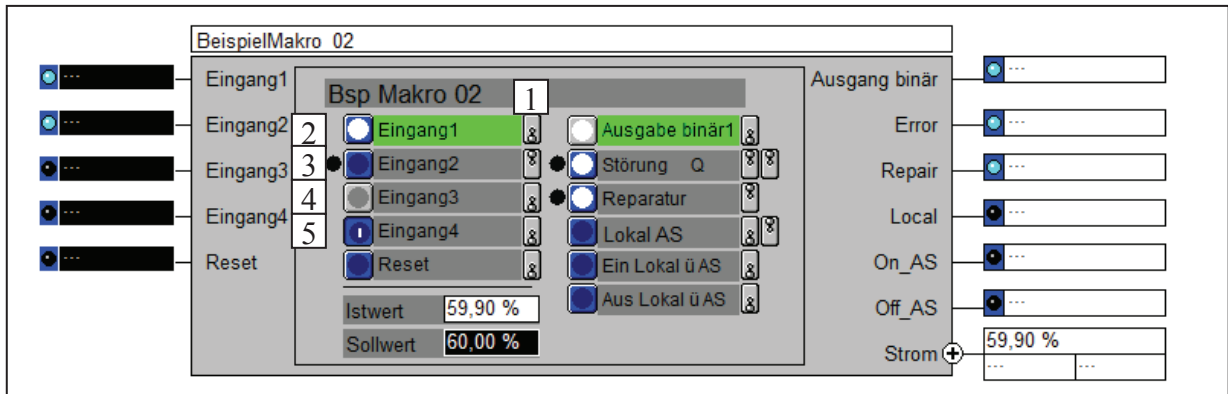


Abb. 8: Bedienfeld, Bedienelemente

Tabelle 7: Bedienfeld, Bedienelemente; Erläuterung

Element	Erläuterung
(1) Invertierungsschalter	<p>Der Invertierungsschalter erlaubt dem Anwender eine Parametrierung der Befehlseingabe- und Rückmeldungslogik. Bei aktivierter Invertierung erscheint ein schwarzer Punkt vor dem zugehörigen Leuchttaster.</p> <p>Jeder Befehlseingang und jede binäre Rückmeldung, welche keinem standardisierten Telegrammprofil zugeordnet werden kann, erhält einen Invertierungsschalter, um sicherstellen zu können, dass Befehle und Rückmeldungen stets korrekt ausgeführt bzw. verstanden werden.</p> <p>z.B.:</p> <p>Ventil1 Eingang „Open“ nicht invertiert: Open = 1 (Ventil öffnet) Open = 0 (Ventil schließt)</p> <p>Ventil2 Eingang Open invertiert: Open = 1 (Ventil schließt) Open = 0 (Ventil öffnet)</p>
(2) Beispieleingang 1	<p>Dieser Befehl ist aktiv: der Leuchttaster leuchtet und das Textfeld erscheint grün.</p> <p>Der Taster (eigentlich ein Schalter, da rastend) wurde <i>nicht</i> gedrückt: in dem Leuchter erscheint keine „1“. Der nicht – gedrückte Taster gibt den Wert logisch „0“ aus. Da sich das virtuelle Gerät im „Fernbetrieb“ befindet, erfolgt die Befehlseingabe über die Makrosignaleingänge. Die Lokalbedienung wird ignoriert.</p> <p>Am Makrosignal „Eingang1“ steht eine logische „1“ an. Da das Signal nicht durch den Invertierungsschalter invertiert wird, ist „Eingang1“ aktiv.</p>

Element	Erläuterung
(3) Beispieleingang 2	Dieser Befehlseingang wurde invertiert. Trotz der am Makrosignaleingang „Eingang2“ anstehenden logischen 1 ist dieser Befehl nicht aktiv.
(4) Beispieleingang 3	Eine Bedienung von grauen Tastern ist nicht möglich.
(5) Beispieleingang 4	Dieser Taster wurde gedrückt und gibt somit eine logische 1 aus. Der Wert wurde nicht invertiert. Da sich das virtuelle Gerät im Fernbetrieb befindet, ist diese Eingabe unwirksam. Der Befehl ist nicht aktiv.
(6) Reset	Reset quittiert eine quittierpflichtige Störung. Dieser Befehl ist betriebsartenunabhängig.

Verschaltung

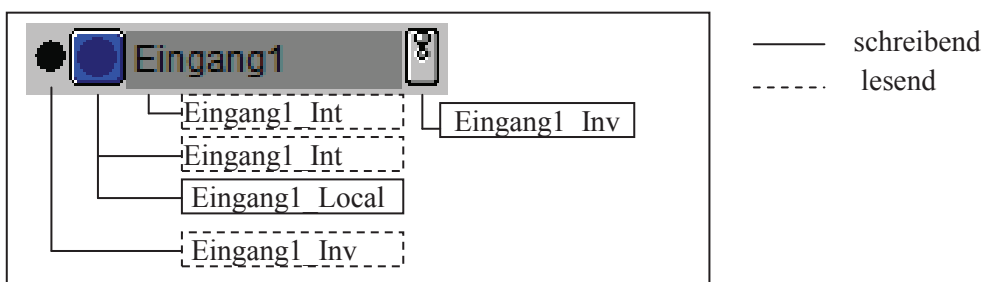


Abb. 9: Eingangssignal, Operanden

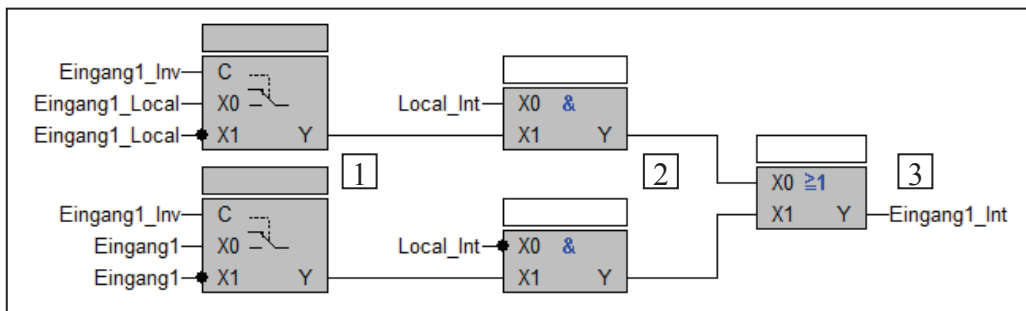


Abb. 10: Eingangssignal, Verschaltung

- 1: Invertierung von lokalen und externen Befehlen
- 2: betriebsartenabhängige „Weiterleitung“ der Befehlssignale
- 3: makrointern sind Befehle von der Invertierung entkoppelt

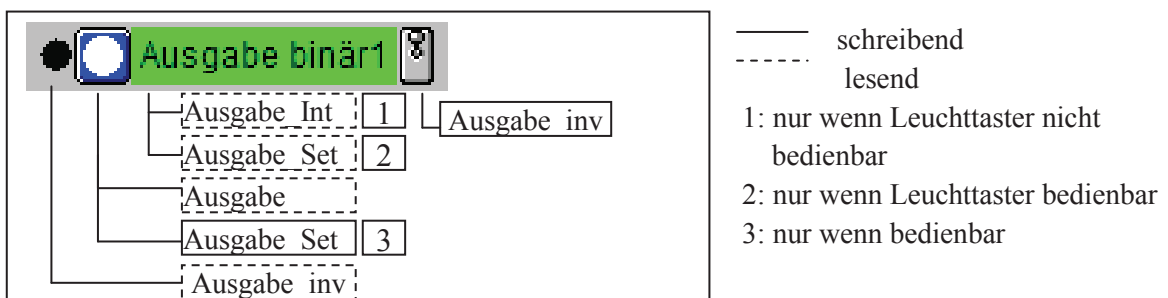


Abb. 11: Ausgangssignal, Operanden

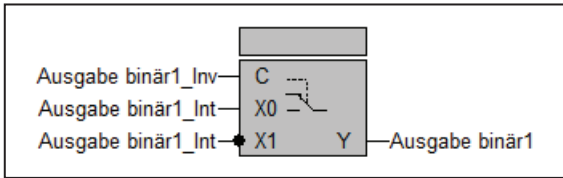


Abb. 12: Ausgangssignal, Verschaltung 1

Bei einem nicht – bedienbaren Leuchttaster wird das interne Signal invertiert.

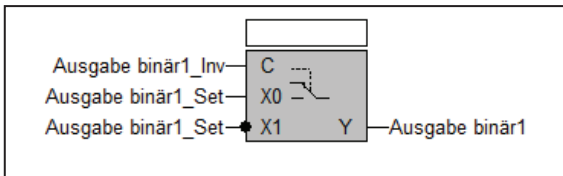


Abb. 13: Ausgangssignal, Verschaltung 2

Bei einem bedienbaren Leuchttaster wird das Signal des Tasters invertiert.

4.3.2 Störverhalten

Einige Feldgeräte besitzen die Möglichkeit Störungszustände zu erkennen. Ursachen für Störungen können vielfältig sein. Einzeln oder zusammengefasst zu Sammelstörungen werde sie als Störmeldung dem Automatisierungssystem mitgeteilt. Die Steuereinheit eines Feldgerätes und/oder einer SPS führen dann eine Fehleroutine durch. Störmeldungen können quittier- und nicht – quittierpflichtig sein. Nicht – quittierpflichtige Störmeldungen werden verlöschen, sobald die Störursache beseitigt wurde. Bei quittierpflichtigen Störmeldungen quittiert ein Reset – Befehl die Meldung, bevor sie verschwindet. [3, pp. 256 - 269]

Im simulierten Feldgerät werden Störungen nicht erkannt, sondern müssen vom Anwender gesetzt werden. Zudem steht ihm frei, ob die Störung quittierpflichtig ist oder nicht. Somit ist es auch unerheblich, ob das reale Feldgerät einen Reset – Befehl erst dann akzeptiert, wenn die Ursache der Störung behoben wurde oder nicht. Ein weiterer Unterschied zur realen Störungsbehandlung ist, dass das Setzen der Störung das Gerät nicht direkt in einen sicheren Zustand fahren lässt. Die Behandlung der Störung wird ausschließlich von der SPS vorgenommen.

Tabelle 8: Vergleich reale und simulierte Störung

Eigenschaft	Reale Störung	Simulierte Störung	Erläuterung
Störungsursache	erkannt durch Fehlerüberwachung der Steuereinheit des Feldgerätes	gesetzt durch Anwender	Fehleroutine des SPS – Anwenderprogramms nach Störmeldung steht im Vordergrund. Fehlerursache ist unerheblich
Quittierpflichtigkeit	wählbar	wählbar	kein Konflikt

Eigenschaft	Reale Störung	Simulierte Störung	Erläuterung
Fehlerroutine	sicherer Zustand für Gerät kann herbeigeführt werden Ausgabe einer Störmeldung	Ausgabe einer Störmeldung	Ausführung der Fehlerroutine des SPS – Anwenderprogramm steht im Vordergrund durch Forcen von Signalen des Steuergerätes kann Fehlerroutine der Steuereinheit simuliert werden

Störungen sind:

- anstehend: die Fehlerursache ist nicht behoben; es wird eine Störmeldung ausgegeben
- gegangen: die Fehlerursache wurde behoben, die Störung jedoch noch nicht quittiert; die Störmeldung wird weiterhin ausgegeben
- nicht – anstehend: es liegt kein Fehler vor; es wird keine Störmeldung ausgegeben

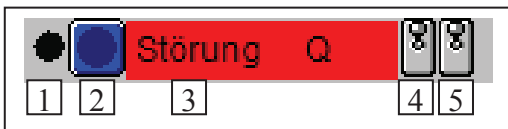




Abb. 14: Störmeldung

Tabelle 9: Störmeldung, Erläuterung

Element	Erläuterung
(1) Meldung invertiert	Erscheint der Punkt, so wird die Meldung logisch invertiert. Bei invertierter Meldung erscheint am Makroausgang: 0 = Störung anstehend/gegangen 1 = Störung nicht anstehend
(2a) Leuchte des Leuchttasters	gibt den Status des Makrosignalausganges „Error“ wieder
(2b) Taster	Fehler anstehend  Fehler behoben 
(3) Textfeld	gibt: 1. Störstatus wieder: Störung anstehend/gegangen: rot Störung nicht – anstehend: grau 2. Quittierpflichtigkeit wieder: Störung quittierpflichtig: Q Störung nicht quittierpflichtig: kein Q
(4) Invertierungsschalter	invertiert die Meldung
(5) Quittierschalter	setzt den Status der Quittierpflichtigkeit

Die Anzahl der Störmeldungsausgänge einer Makrovorlage soll sich an der maximalen Anzahl der Störmeldungseingänge des ihm zuzuordnenden Messstellentyps orientieren. Die Makrosignalausgänge der Störungen werden mit „Error“, „Error 2“, „Error 3“ usw. benannt.

Verschaltung

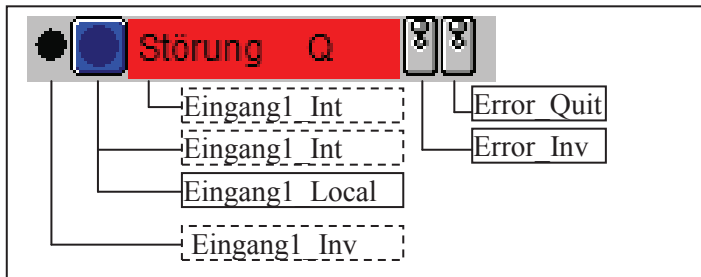


Abb. 15: Störmeldung, Operanden

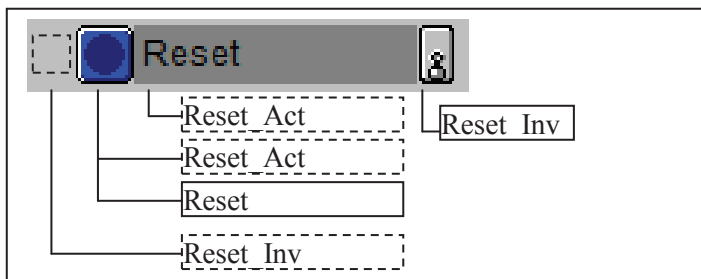


Abb. 16: Reset, Operanden

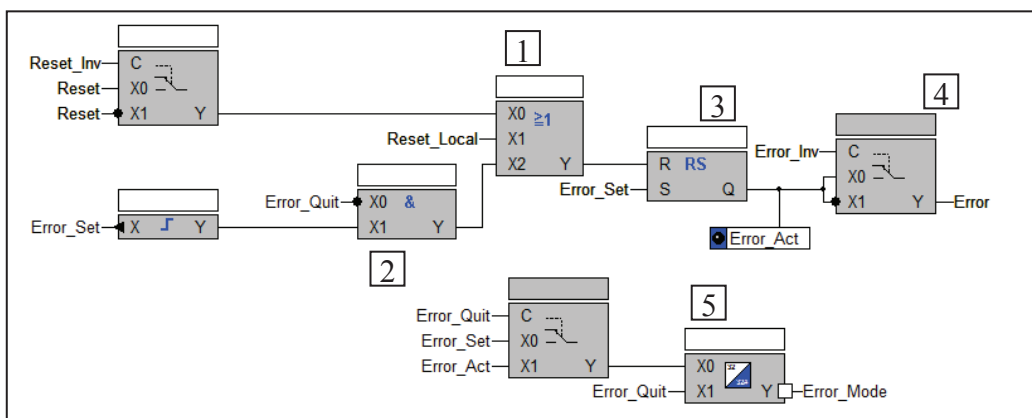


Abb. 17: Störung, Verschaltung

- 1: Invertierung des Resetsignals. Es wird nicht zwischen lokalem und externem Reset unterschieden.
- 2: „Error_Quit“ setzt den Modus „Quittierpflichtige Störung“ (Error_Quit = 1). Ist dieser Modus nicht aktiv, so findet die Quittierung bereits durch Beseitigung der Störungsursache („Error_Set“ \neg Schaltflanke negativ) statt.

3. „Error_Set“ setzt den Stöorzustand „Error_Act“, bis dieser über den Schalteingang „R“ wieder zurückgesetzt wird (setzdominant).
4. Abhängig von „Error_Inv“ wird der gesetzte Stöorzustand mit „Error“ = 0 (invertiert) oder „Error“ = 1 (nicht invertiert) ausgegeben.
5. Der Binär – Digitalkonverter gibt abhängig von dem gewählten Modus der Quittierpflichtigkeit und dem Stöorzustand einen digitalen Wert (0 – 3) aus, welcher das Element „Active Text (digital)“ den entsprechenden Status (siehe Tabelle 9: Störmeldung, Erläuterung) wiedergeben lässt.

4.3.3 Betriebsartenumschaltung

Es soll möglich sein, zwischen den Betriebsarten „Lokalbetrieb“, „Fernbetrieb“ und „Lokalbetrieb über Automatisierungssystem“ wechseln zu können.

- **Lokalbetrieb** bedeutet, dass sämtliche Befehle an der Vorortsteuerstelle gesetzt und direkt von der technologischen Einheit verarbeitet werden.
- Der **Fernbetrieb** ist die Ansteuerung durch das Automatisierungssystem.
- Beim **Lokalbetrieb über das Automatisierungssystem** werden die Befehle lokal an der Vorortsteuerstelle gesetzt, jedoch durch das Automatisierungssystem verarbeitet. Die eigentliche Ansteuerung findet somit, wie beim Fernbetrieb, über das Automatisierungssystem statt. Dies ist sinnvoll, wenn die Ansteuerung des Gerätes durch das Automatisierungssystem überwacht werden soll, um das Herbeiführen kritischer Zustände infolge einer unsachgemäßen Bedienung zu vermeiden. Allerdings kann das Gerät bei einem Ausfall des Automatisierungssystems nicht mehr bedient werden.

In den vorliegenden Projekten wird jeweils nur eine der Lokalbetriebsarten genutzt. Da die zu erstellenden Makros möglichst universell einsetzbar sein sollen, soll sowohl der Lokalbetrieb als auch der Lokalbetrieb über AS genutzt werden können.

Die Umschaltung der Betriebsarten erfolgt durch einen Schalter, der sich am Gerät befindet. Dem Automatisierungssystem wird der Betriebsartenstatus mitgeteilt.

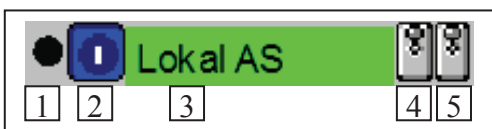




Abb. 18: Betriebsartenumschaltung

Tabelle 10: Betriebsartenumschaltung, Erläuterung

Element	Erläuterung
(1) Meldung invertiert	Wird die Betriebsart logisch invertiert, so wird dies durch einen Punkt gekennzeichnet. Bei invertierter Meldung erscheint am Makroausgang: 0 = Lokalbetrieb/Lokalbetrieb über AS 1 = Fernbetrieb
(2a) Leuchte des Leuchttasters	gibt den Status des Makrosignalausganges „Local“ wieder
(2b) Taster	eine Lokalbetriebsart gesetzt  Fernbetrieb gesetzt 
(3) Textfeld	Gibt wieder: 1. ob eine Lokalbetriebsart gewählt wurde: ja: grün nein: grau 2. welche Lokalbetriebsart gewählt wurde: Lokalbetrieb: „Lokal“ Lokalbetrieb über AS: „Lokal AS“
(4) Invertierungsschalter	invertiert die Meldung
(5) Lokalbetriebsartenschalter	Wechsel zwischen den Betriebsarten „Lokalbetrieb“ und „Lokalbetrieb über AS“

Die Meldung der Betriebsart erfolgt über das Makrosignal „Local“.

Verschaltung

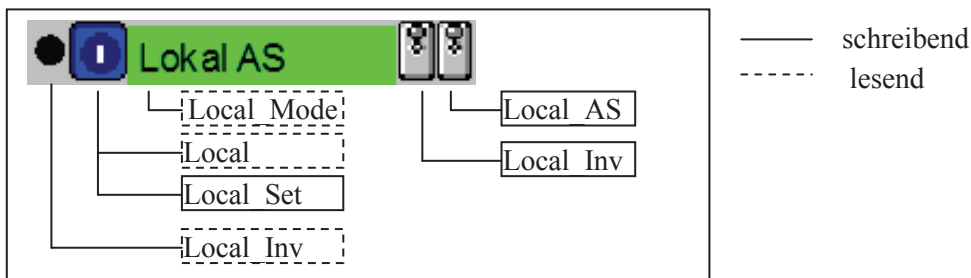


Abb. 19: Betriebsartenumschaltung 1, Operanden

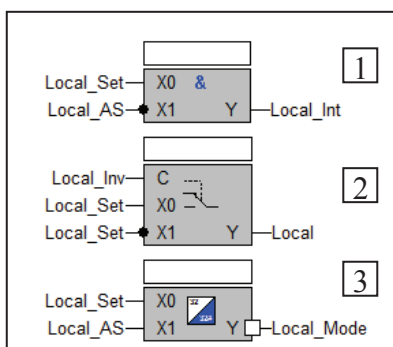


Abb. 20: Betriebsartenumschaltung 1, Verschaltung

1: In seiner weiteren Verwendung wird der Operand „Local_Int“ (interner Lokalbetrieb) genutzt, um zu entscheiden, ob das Gerät durch lokale oder externe Befehle gesteuert wird.

2: In Abhängigkeit der Schalterstellung „Local_Inv“ wird der gesetzte Lokalbetrieb (Local_Set) als logische 1 oder 0 über den Makroausgang „Local“ ausgegeben.

3: Der Binär – Digitalkonverter gibt abhängig von der gewählten Betriebsart einen digitalen Wert (0 – 3) aus. Der Modus wird über das Textfeld ausgegeben

Bei den Feldgeräten SIMOCODE und einem mit dem PROFIdrive Standardtelegramm 1 oder 20 angesteuerten Frequenzumrichter ist zusätzlich das Umschalten der Betriebsart aus dem Automatisierungssystem heraus möglich. [3, p. 63] [4, p. 9]

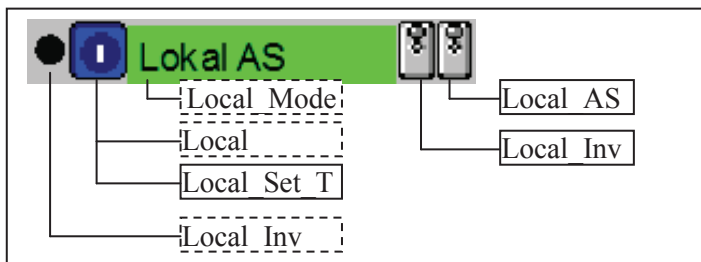


Abb. 21: Betriebsartenumschaltung 2, Operanden

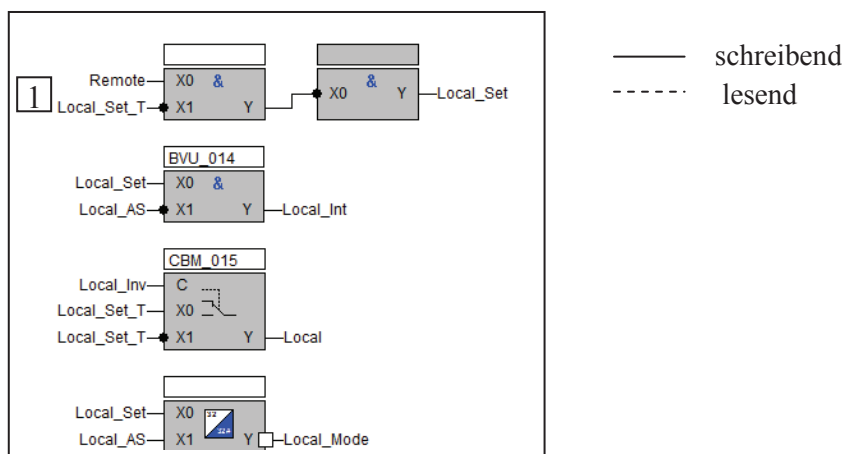


Abb. 22: Betriebsartenumschaltung 2, Verschaltung

Die Betriebsartenumschaltung wurde um die angegebene Verschaltung erweitert. Mit „Remote“ wird die Betriebsartenumschaltung aus dem Automatisierungssystem heraus realisiert.

4.3.4 Befehlseingabe am Makro

Grundsätzlich ist zu jeder Betriebsart die Anwahl eines jeden Schalters oder Tasters möglich, seine Ausführung jedoch nicht. Das Setzen von Rückmeldungen ist betriebsartenunabhängig. Es können nur solche Rückmeldungen im Bedienfeld gesetzt werden, welche nicht der internen Verschaltungslogik des Makros entspringen. Eine Änderung von Parametern ist jederzeit möglich.

Fernbetrieb

Die Befehlseingabe erfolgt über die Makrosignaleingänge.

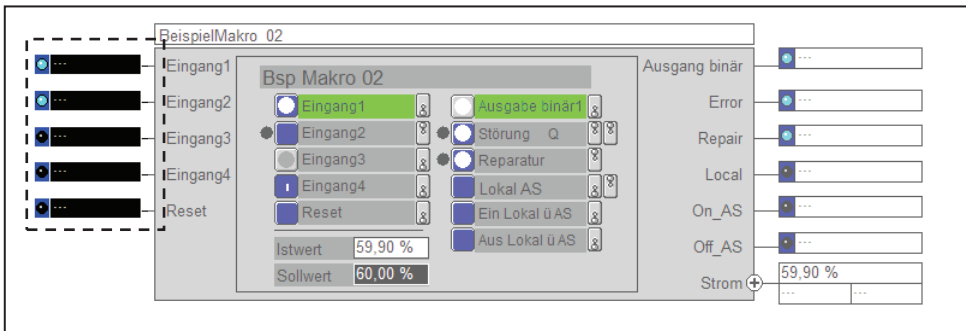


Abb. 23: Befehlseingabe, Fernbetrieb

Lokalbetrieb

Die Befehlseingabe erfolgt im Befehlseingabebereich des Bedienfeldes.

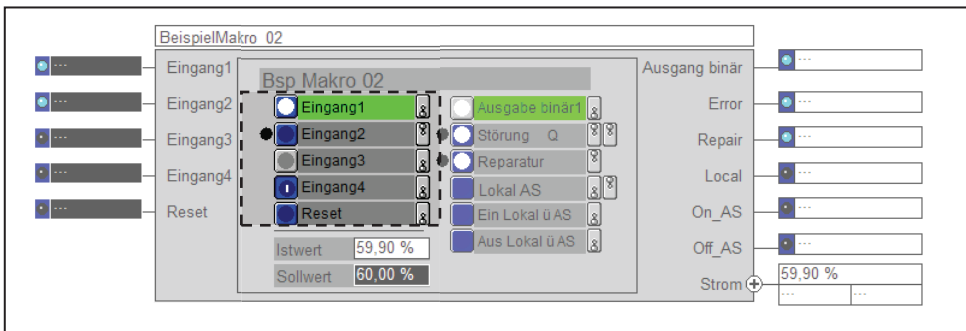


Abb. 24: Befehlseingabe, Lokalbetrieb

Lokalbetrieb über das Automatisierungssystem

Die Befehlseingabe erfolgt im Bereich „Meldungen“ des Bedienfeldes über Taster, die mit „AS“ gekennzeichnet sind.

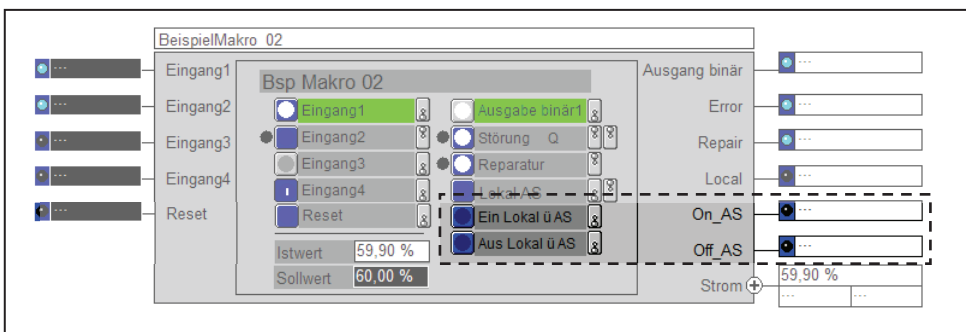


Abb. 25: Befehlseingabe, Lokalbetrieb über Automatisierungssystem

4.4 Umwandlung und Übertragung digitaler und analoger Werte

4.4.1 Analog – Digital – Wandlung

Die Verarbeitung einer analogen Messgröße in einer SPS kann nur durch ihren digitalisierten Wert ermöglicht werden.

Eine analoge Messgröße, z.B. eine Temperatur oder ein Füllstand, wird zunächst von einem Messumformer erfasst und anschließend in Spannungs- oder Stromsignale gewandelt, deren Stärke mit dem gemessenen physikalischen Wert normiert ist. Verschiedene Einheitsspannungs- oder -stromsignale sind hierfür definiert, z.B. das 4 ... 20mA Stromsignal. Ein Analog – Digitalumsetzer wandelt nun das analoge elektrische Signal in ein digitales um, um eine elektronische Verarbeitung durch die CPU zu realisieren.

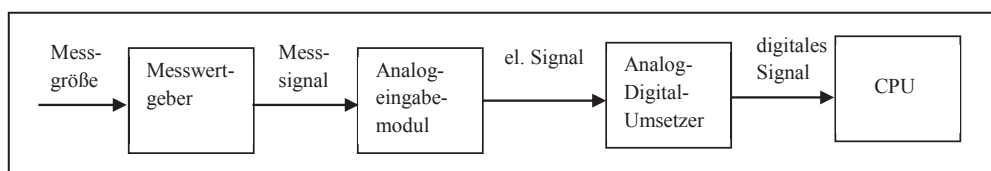


Abb. 26: Analogwert – Verarbeitung

Die folgenden Ausführungen und Zahlenwerte beziehen sich auf SIMATIC Analogmodule der Reihe ET200, zur AD- bzw. DA – Wandlung. Das digital – aufbereitete Signal wird stets als 16 Bit - Wort ausgegeben, wobei Bit 16 als Vorzeichenbit genutzt wird. Somit steht ein Wertebereich von $-32768 \dots 32767_{\text{Dez}}$ zur Verfügung.

Für die zu messende Größe werden ein Nennbereich sowie ein Über- und Untersteuerungsbereich festgelegt. Das elektrische und das digitale Signal werden für den Nennbereich normiert. Für digitalisierte Analogsignale von ET200 Analogmodulen ist dies der Bereich $0 \dots 27648$. Weiterhin werden ein Über- und ein Unterlaufbereich definiert.

Die Messgröße – und damit einhergehend das elektrische Signal sowie der Digitalwert – befinden sich im Regelbetrieb immer im Nennbereich. Werden die Bereiche des Über – oder Unterlaufes erreicht, so kann dies eine entsprechende Störmeldung erzeugen. Ein defektes Gerät könnte die Ursache sein. Der Über – und der Untersteuerungsbereich dienen als Toleranzband, um bei kurzzeitigem Überschwingen über die Grenze des Nennbereichs hinaus ein sofortiges Auslösen einer Störmeldung zu unterbinden.

Für die Messung des Füllstandes eines Tanks – der Füllstand bewegt sich in einem Nennbereich zwischen 0 und 3000mm – und der Übertragung des Messergebnisses durch ein 4 ... 20mA Einheits- Stromsignal ergibt sich folgende Tabelle:

Tabelle 11: Normierung Messgröße - Digitalwert

Füllstand in mm	Verhältniswert in %	Stromstärke in mA	Digitaler Wert als Dezimalzahl	Bereichsbezeichnung
3555	118,5	22,96	32767	Überlauf
			32512	
3527	117,6	22,81	32511	Übersteuerung
			27649	
3000	100	20	27648	Nennbereich
2250	75	16	20736	
0	0	4	0	
		4 – 0,579	-1	Untersteuerung
x			-4864	
x		1,185	-4865	Unterlauf
x			-32768	

Die Umformung eines gemessenen Füllstandes von 2250 mm in ein digitales Signal erfolgt nach folgendem Schema:

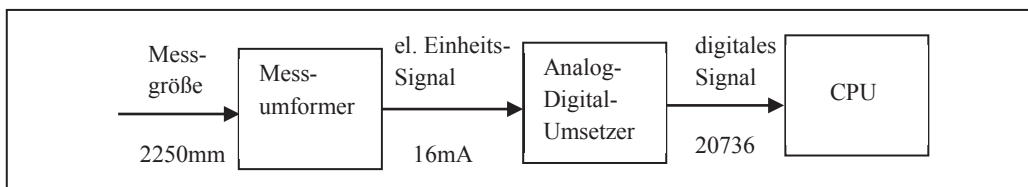


Abb. 27: Analogwert – Verarbeitung Beispielgrößen

Die Wandlung eines Digital – in ein Analogsignal, z.B. zur Vorgabe des Stellweges eines Stellventils, erfolgt invers nach gleichem Prinzip.

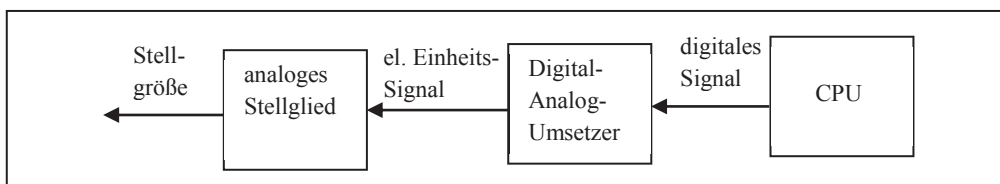


Abb. 28: Analoge Stellgröße

4.4.2 Analog – Digital - Wandlung in WinMOD

Dem im 4.4.1 Analog – Digital – Wandlung beschriebenen Prinzip folgend geschieht die Analogwertverarbeitung in WinMOD. So ist vorgesehen, dass die Mess- bzw. Stellgröße durch das Anzeigeformat „internes Analogformat“ und im Weiteren das Digitalsignal durch ein Anzeigeformat „externes Analogformat“ normiert werden.

Für das interne Analogformat müssen stets ein „physikalischer“ und ein „logischer“ Wertebereich definiert werden. Der physikalische Bereich entspricht dabei dem Nennbereich der physikalischen Messgröße. Dies kann eine physikalische Größe wie eine

Temperatur, aber auch der prozentuale Öffnungsweg eines Stellventils oder die prozentuale Drehzahl eines Motors sein. Der logische Wertebereich dient der Normierung.

Das externe Analogformat entspricht einem AD – oder DA – Umsetzer, welcher zwischen normierten Analog – und Digitalwerten wandelt. Dieses Anzeigeformat bildet die Schnittstelle zur CPU.

Die Messwertverarbeitung findet nach folgendem Schema statt:

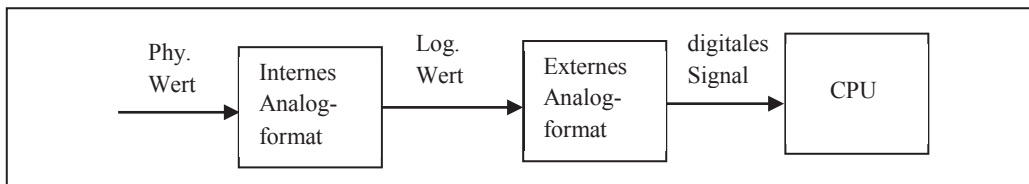


Abb. 29: Analogwert – Verarbeitung WinMOD

Folgende Tabelle erklärt die Skalierung zwischen den Formaten. Der physikalische Wertebereich liegt hier zwischen 0 und 250°C, der logische zwischen 0 und 100%. Das Digitalsignal wurde auf 0 und 27648 normiert.

Tabelle 12: Wertnormierung WinMOD

Internes Analogformat		Externes Analogformat	
Physikalischer Wert in °C	Logischer Wert in %	Digitalwert	
x	x	32765	
250	100	27648	
100	40	11059	
0	0	0	
x	x	-32766	

Die Normierung des Digitalsignals bezieht sich immer auf die Werte 0 und 100 % des logischen Wertebereichs des internen Analogformates. Die Grenzen des logischen Wertebereichs können nur in Ausnahmen überschritten werden. Bei Definition eines logischen Wertebereichs, dessen Grenzen außerhalb von +100 % oder -100 % liegen, werden darüber- oder darunterliegende Werte nur akzeptiert, wenn damit auch ein externes Analogformat beschrieben wird. Eine interne Weiterleitung ist ausgeschlossen. [1, p. 6.24]

Definieren des externen Analogformates „Siemens 0 ... 27648“

Zur Simulation der Analog – Digital – Umwandlung von SIMATIC Analogmodulen der Reihe ET200 wird das externe Analogformat „Siemens 0 ... 27648“ definiert. Es wurde wie folgt parametrisiert:

Tabelle 13: Parametrierung externes Analogformat „Siemens 0 ... 27648“

Parameter	Wert
Datengröße	Wort (16 Bit)
Bitreihenfolge	Big - Endian
Vorzeichen	Ja, Bit 16
Normierung	0 ... 27648

Nach diesem Schema wurden für die Digital – Analogwert – Normierung des SIMOCODE (siehe Tabelle 31) und des PROFIdrive – Profils (Tabelle 33) weitere externe Analogformate definiert.

Definieren des internen Analogformates „Prozent[%] +/-120“

Zur Ausgabe von Messgrößen wie einer Temperatur oder einem Druck müssen interne Analogformate individuell erstellt werden, um aussagekräftige Werte seitens des Automatisierungssystems zu erhalten. Unter Verwendung eines Prozentformates ist das Ablesen eines prozentualen Wertes bezogen auf den Nennbereich jedoch ebenfalls möglich.

Das Prozentformat beschreibt dabei die Art der Einstellung eines internen Analogformates, bei welchem der physikalische dem logischen Wertebereich gleichgesetzt wird.

Die internen Analogformate „Prozent [%] +/-100“ und „Prozent [%] 0...100“ sind in WinMOD bereits vordefiniert. Mit ersterem Format ist ein Operieren im bipolaren Nennbereich möglich, zweiteres ist auf den positiven also unipolaren Nennbereich beschränkt. Die Möglichkeit des Über – oder Untersteuerns unter Verwendung des externen Analogformates „Siemens 0 ... 27648“ ist in beiden Fällen nicht gegeben. Da zur Herbeiführung eines Fehlerzustandes diese Option bestehen soll, wird ein weiteres Format eingeführt: „Prozent[%] +/-120“.

Nachfolgende Tabelle zeigt die verschiedenen internen Analogformate sowie den unter Verwendung des externen Analogformates „Siemens 0...27648“ darstellbaren digitalen Bereich.

Tabelle 14: interne Analogformate

Internes Analogformat	Physikalischer Wertebereich in %	Logischer Wertebereich in %	Digitaler Bereich	Beispiel für entsprechendes Einheitsstromsignal
Prozent [%] +/-100	-100 ... +100	-100 ... +100	-27648 ... 27648	±20mA; bipolar
Prozent [%] 0...100	0 ... 100	0 ... 100	0 ... 27648	4 ... 20mA, unipolar
Prozent[%] +/-120	-120 ... +120	-120 ... +120	-32768 ... 32767; da 16 Bit begrenzt	±20mA; bipolar

4.4.3 Externe Digitalformate

Zur Aufnahme und Ausgabe digitaler Größen werden in PCS 7 die Kanaltreiber – Funktionsbausteine „FbDiIn“ und „FbDiOu“ genutzt. „Digital“ beschreibt hier die Zusammenfassung mehrerer Binärsignale zu einer Einheit. Die Bausteine arbeiten mit Digitalwerten des Word – Formates und der Bytereihenfolge „Big Endian“ und sind nicht vorzeichenbehaftet. [6, p. 1672]

Definieren des externen Digitalformates „Big Endian Word“

Die genannten Eigenschaften werden auf das externe Digitalformat „Big Endian Word“ angewandt. Da die Digitalsignale direkt und ohne Konvertierung übertragen werden sollen, gilt für das Zahlenformat die „Binärcodierung“.

Tabelle 15: Parametrierung externes Digitalformates „Big Endian Word“

Parameter	Parameterwert
Zahlenformat	Binärcodierung
Bytes Verarbeitungsbreite	2
Bytes vertausche (Big Endian)	ja
Vorzeichenbehaftet konvertieren	nein

4.5 Makrovorlagen

Im Folgenden sollen der Aufbau und die Funktion der zu simulierenden Feldgeräte und Regelstrecken sowie der diese ersetzenden Makros aus den Tabelle 4: Makrovorlagen zur Antriebssimulation und Tabelle 5: Makrovorlagen zur Regelstreckensimulation beschrieben werden.

4.5.1 Makrovorlagen der Aktoren

4.5.1.1 Ventile

Ventile dienen der Beeinflussung eines Volumenstromes durch Veränderung des Durchflussquerschnittes einer Leitung infolge der Verstellung eines beweglich angeordneten Drosselementes. [7, p. 95]

Neben Kriterien der Einteilung von Ventilen nach Wirkprinzipien, dem Stellglied, dem Stellantrieb oder dem Einsatzbereich ist aus Sicht des Projektors eines Prozessleitsystems insbesondere die Ansteuereinrichtung von Bedeutung, da sie die Schnittstelle zwischen den mechanischen Komponenten des Ventils und dem Automatisierungssystem darstellt. Die Ansteuereinrichtung wertet das Stellsignal aus und gibt den Ansteuerbefehl an den Stellantrieb weiter, welcher wiederum das Stellglied, also das Drosselement, positioniert. Weiterhin werden über die Ansteuereinrichtung Statusinformationen, wie z.B. die Position, zurückgegeben.

Nach Art der Ansteuerung kann im Wesentlichen zwischen stetig – und unstetig – stellend Ventilen unterschieden werden. [7, p. 95]

4.5.1.1.1 2/2 - Wege – Schaltventil

Schaltventile gehören zu den unstetig – stellenden Ventilen, was bedeutet, dass sie nur definierte Endlagen kennen. Zwischenstellungen dienen lediglich dem Zustandsübergang. Das Verweilen in der Zwischenstellung ist die Folge einer Störung im Ventil. Ihre Ansteuerung sowie ihre Stellungsrückmeldung erfolgen durch binäre Signale. Im energielosen (oder stromlosen) Zustand kehren Schaltventile häufig in eine Ruhelage zurück, deren Erreichen durch Rückstellfedern realisiert wird.

2/2 - Wege – Schaltventil

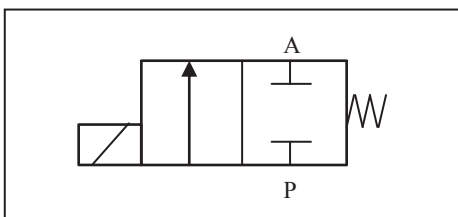


Abb. 30: 2/2 Wegeventil mit Federkraft-Rückstellung, Betätigung elektromagnetisch, stromlos geschlossen

Das 2/2 – Wege – Schaltventil besitzt 2 Leitungsanschlüsse und 2 definierte Endlagen: Auf und Zu. Der Schaltvorgang infolge einer Ansteuerung einer Magnetspule oder einer Pneumatikbetätigung vollzieht sich rasch. Wie jedes Schaltventil besitzt es eine Ruhelage: stromlos geöffnet oder stromlos geschlossen.

Verwendung in PCS 7 – Projekten

Zunächst wurde die Projektierung von 2/2 – Wege – Schaltventilen in den Beispielprojekten untersucht. Nachfolgend werden alle für die Erstellung des Makros wesentlichen Ein- und Ausgaben (aus Sicht der SPS) sowie Parametrierungen aufgeführt.

Signaleingänge

Tabelle 16: 2/2 – Wege – Schaltventil Signaleingänge

Binäreingänge	Erläuterung	Ausführung	Makrosignal Standardeinstellung
Auf	Rückmeldung „Auf“ bei logisch 1 (Schließer) oder logisch 0 (Öffner)	Schalter meist als Öffner: 1 = nicht geöffnet 0 = geöffnet	invertiert
Zu	Rückmeldung „Zu“ bei logisch 1 (Schließer) oder logisch 0 (Öffner)	Schalter meist als Öffner: 1 = nicht geschlossen 0 = geschlossen	invertiert
Lokalbetrieb	Meldet Vorortsteuerung an das Automatisierungssystem	Schalter als Schließer: 1 = Lokalbetrieb 0 = kein Lokalbetrieb	nicht invertiert
Lokal Öffnen	Öffnen des Ventils über Lokalsteuerung unter AS – Überwachung bei aktivem Lokalbetrieb	Taster: Positive Flanke = öffnen	nicht invertiert (Ausführung als Schalter)
Lokal Schließen	Schließen des Ventils über Lokalsteuerung unter AS – Überwachung bei aktivem Lokalbetrieb	Taster: Positive Flanke = schließen	nicht invertiert (Ausführung als Schalter)

Signalausgänge

Tabelle 17: 2/2 – Wege – Schaltventil Signalausgänge

Binärausgänge	Erläuterung	Ausführung	Makrosignal Standardeinstellung
Schalten	Öffnet oder schließt Ventil	Schalter/meist als Öffner	nicht invertiert

Funktionsbaustein VlvL

Tabelle 18: ausgewählte Parameter des Funktionsbausteins VlvL

Parameter	Erklärung	Bedeutung für Makro
MonTiDynamic	Nach erfolgter Ansteuerung, also dem Befehl das Ventil zu öffnen oder zu schließen, wird innerhalb eines bestimmaren Zeitraumes eine entsprechende Rückmeldung erwartet. Erfolgt diese nicht innerhalb des Zeitraumes, so wird eine Störmeldung ausgegeben. MonTiDynamic = 3s ist der geringste Parameterwert.	Die Schaltzeit muss parametrierbar sein. Als Standardeinstellung werden 0,25s gewählt.

Das Makro „AufZuVentil“

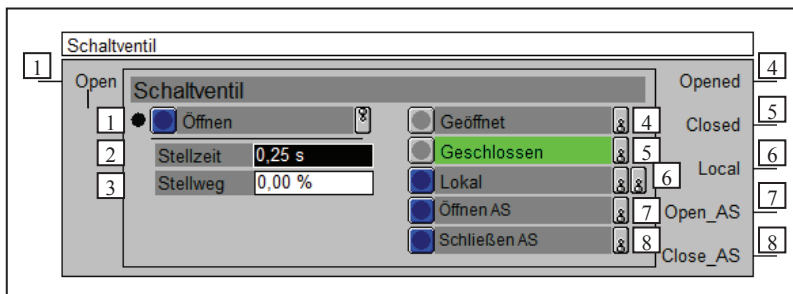


Abb. 31: Makro, AufZuVentil

Tabelle 19: Schaltventil, Operanden

Element	Operanden	Erläuterung
(1) Ventil öffnen/schließen	Open_x	Öffnen (Open = 1) oder Schließen (Open = 0) des Ventils (wenn nicht invertiert)
(2) Stellzeit	Stellzeit	Stellzeit einstellbar (0 – 99999 s); Stellzeit für Stellweg zwischen 0 % und 100 %
(3) Stellweg	Stellweg	Anzeige des Stellweges; Anzeigeformat: Prozent [%] +/-100
(4) Ventil geöffnet	Opened_x	Ausgabe Ventil geöffnet (Opened = 1; Stellweg \geq 95 %) oder nicht geöffnet (Opened = 0; Stellweg < 95 %) (wenn nicht invertiert)
(5) Ventil geschlossen	Closed_x	Ausgabe Ventil geschlossen (Closed = 1; Stellweg \geq 5 %) oder nicht geschlossen (Opened = 0; Stellweg < 5 %) (wenn nicht invertiert)
(6) Betriebsart	Local_x	Wahl und Ausgabe der Betriebsart gemäß Kap. 4.3.3 Betriebsartenumschaltung (Schaltung 1)
(7) Öffnen über AS	Open_AS_x	Lokaler Steuerbefehl zum Öffnen (Open_AS = 1) des Ventils mit AS – Überwachung (wenn nicht invertiert); (Ausführung als Schalter)
(8) Schließen über AS	Close_AS_x	Lokaler Steuerbefehl zum Schließen (Close_AS = 1) des Ventils mit AS – Überwachung (wenn nicht invertiert); (Ausführung als Schalter)

Verschaltung

Zur Simulation des Verhaltens eines „AufZuVentils“ wurde das WinMOD – Element „Stellantrieb“ verwendet.

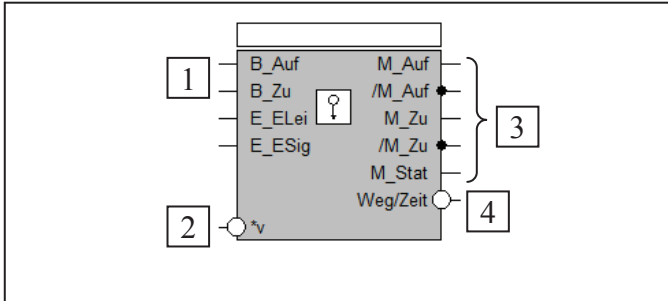


Abb. 32: WinMOD – Element, Stellantrieb

Durch die Ansteuerung eines Einganges (Auf oder Zu) (1) erscheint nach einer parametrierbaren Stellzeit (2) die entsprechende Rückmeldung (3). Innerhalb eines bestimmten Bereiches des Stellweges (4), befindet sich das Ventil in einer Zwischenstellung (5-95%) und ist weder geöffnet noch geschlossen.

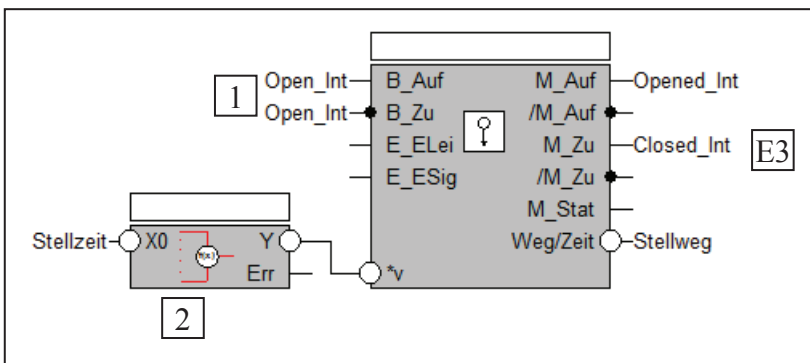


Tabelle 20: AufZuVentil, Verschaltung

Das Signal „Open_Int“ öffnet oder schließt das Ventil (1). Bei der Berechnung der Stellzeit muss zunächst der Kehrwert der Stellzeit (2) ausgegeben werden. Die Position und der Stellweg werden zurückgegeben (3).

Die Befehls- und Statussignalinvertierung folgen den Vorlagen aus Abschnitt 4.3.1 „Bedienfeld und die Betriebsartenumschaltung der Vorlage 1 aus dem Abschnitt 4.3.3 „Betriebsartenumschaltung“.

4.5.1.1.2 Stellventil

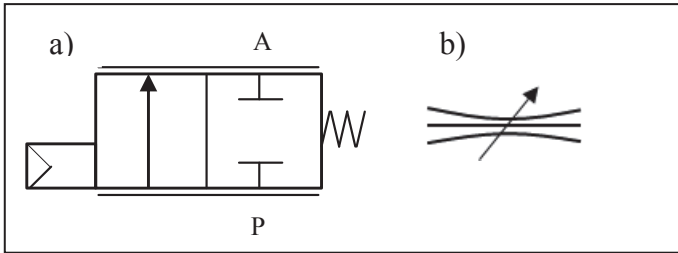


Abb. 33: Stellventil a) Aufbaubeispiel: 2/2 – Wege – Ventil, pneumatische Vorsteuerung [8, p. 108], Federkraftrückstellung b) Funktion: Drosselventil [9, p. 9]

Das Stellventil, auch Drosselventil genannt, ist ein stetig – stellendes Stromventil. Die Positionierung des Stellgliedes durch den Stellantrieb ist dem Einheitsstromsignal proportional. Die Rückmeldung des Stellweges an das Automatisierungssystem erfolgt ebenfalls durch ein Analogsignal. Als Stellantrieb wirken häufig Elektromagnete, welche gegen eine Rückstellfeder arbeiten. Im energielosen Zustand drückt die Feder das Ventil in eine bestimmte Endlage: Stellventile werden als stromlos offen oder stromlos geschlossen ausgeführt. [10, p. B15] Neben dem Elektromagnetismus werden Hilfsenergien pneumatisch oder hydraulisch bereitgestellt. In diesen Fällen können Pilotventile eingesetzt werden. Dabei handelt es sich um 2/2 – Wege – Schaltventile, welche im geöffneten Zustand das zum Stellen benötigte Medium hindurchlassen. Werden sie geschlossen, so fährt das Stellventil in seine Ruhelage. Eine weitere sicherheitsgerichtete Maßnahme stellen Endlagenschalter dar, welche zusätzlich zur Analogmessung die Endlagen des Stellventils überwachen.

Seinen Einsatz findet dieses Ventil vor Allem bei der Regelung des Durchflusses oder des Druckes von Fluidströmen.

Verwendung in PCS 7 – Projekten

Im Folgenden werden alle für diese Ventilart verwendeten Ein – und Ausgangssignale (aus Sicht der SPS) und relevante Parametrierungen aufgeführt.

Signaleingänge

Tabelle 21: Stellventil, Signaleingänge

Analogeingänge	Erläuterung	Bedeutung für Makro
Öffnungsweg Ist	Wert im Nennbereich zwischen 0 und 100%; Je nach Ausführung des Ventils geben die Grenzen des Bereiches eine vollständige Öffnung oder Schließung an.	Um das Verhalten außerhalb des Nennbereichs zu testen kann das neu definierte Anzeigeformat Prozent[%] +/-120 verwendet werden.

Signalausgänge

Tabelle 22: Stellventil, Ausgangssignale

Analogausgänge	Erläuterung	Bedeutung für Makro
Öffnungsweg Soll	Wert zwischen 0 und 100%; Je nach Ausführung des Ventils führt eine Änderung des Sollwertes zu einer weiteren Öffnung oder Schließung.	Eine Verwendung des Anzeigeformates Prozent[%] +/-120 ist möglich. Sollte jedoch eine Ansteuerung >100 Prozent vorgenommen werden, so kann dieser Wert simulationsintern nicht weitergegeben werden. Der Ist – Öffnungsweg wird dann höchsten 100% betragen.

Ergänzung

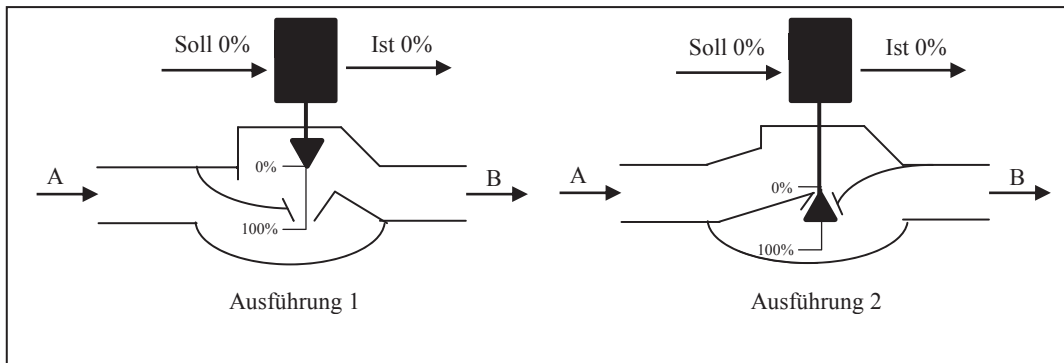


Abb. 34: 2/2 – Wegeventil, Gehäuseausführungen

Im Falle der Gehäuseausführung 1 entspricht ein Stellweg von 0% einem geöffneten und 100% einem geschlossenen Ventil. In der Ausführung 2 ist dies genau umgekehrt. Eine Sollwertstellung von 0% führt also im einen Fall dazu, dass das Ventil komplett geöffnet und im anderen Fall komplett geschlossen ist. Bei einigen Ventilantrieben lässt sich diese Logik umkehren, sodass unabhängig von der Gehäuseausführung das Stellsignal 0% immer zum Schließen des Ventils führt. Auch wenn diese Funktion nicht gegeben ist, führt dies bei einem Signaltest nicht zu Konflikten. Da nie der Öffnungsweg, sondern nur der Ausfahrweg des Stellgliedes überwacht wird, erwartet das Automatisierungssystem als Rücklesewert immer seinen ausgegeben Sollwert. Die Gehäuseausführung muss demnach nicht beachtet werden. Simulationsintern kann aus dem Ist – Stellwert jedoch nicht in jedem Fall auf den Öffnungszustand des Ventils rückgeschlossen werden. Im Einzelfall ist dies zu beachten.

Funktionsbaustein VlvAnL

Tabelle 23: ausgewählte Parameter des Funktionsbausteins VlvAnL

Parameter	Erläuterung	Bedeutung für Makro
MonTiDynamic	Nach erfolgter Ansteuerung, also dem Befehl das Ventil zu öffnen oder zu schließen, wird innerhalb eines bestimmaren Zeitraumes eine entsprechende Rückmeldung erwartet. Erfolgt diese nicht innerhalb des Zeitraumes, so wird eine Störmeldung ausgegeben. MonTiDynamic = 45s ist der geringste Parameterwert.	Die Stellzeit muss parametrierbar sein. Die höchstzulässige Trägheit von 45s des Stellgliedes soll sich im Makro nicht wiederfinden. Als Standardeinstellung wird 1s gewählt.

Zusammenfassung

In keinem Projekt werden Endlagenschalter verwendet; Pilotventile werden ausschließlich als eigenständige Einheiten betrachtet, welche bereits durch das Makro „AufZuVentil“ dargestellt werden können. Ebenso existieren keine expliziten Signale zur Stromlosschaltung eines Stellventils. In jedem Fall wird lediglich ein Analogsignal ausgegeben und/oder zurückgelesen, weshalb auch im Makro nur diese Signale berücksichtigt werden sollen.

Das Bedienen eines Stellventils im Lokalbetrieb ist möglich; ein Betriebsartenwechsel zwischen Lokal- (über Automatisierungssystem) und Fernbetrieb ist hingegen nicht vorgesehen. Somit soll das Makro über das Bedienfeld bedient werden können, auf die Ausgabe des Signals „Local“ wird jedoch verzichtet.

Das Makro „Stellventil“

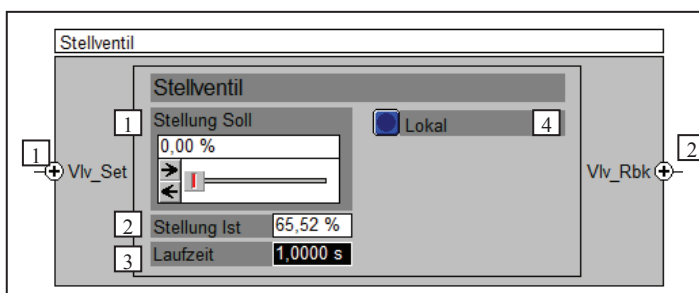


Abb. 35: Stellventil, Makro

Tabelle 24: Stellventil, Operanden

Element	Operanden	Erläuterung
(1) Sollstellung	Vlv_Set x	Externe/lokale Sollwertvorgabe: -100 %– 100 %
(2) Iststellung	Vlv_Rbk	Ausgabe des Stellweges: -100% – 100 %
(3) Stellzeit	Tn	Stellzeit einstellbar (0 – 99999 s); Stellzeit für Stellweg zwischen 0% und 100 %
(4) Betriebsart	Local	Wahl und Ausgabe der Betriebsart oder nicht geöffnet (Opened = 0; Stellweg < 95 %) (wenn nicht invertiert)

Verschaltung

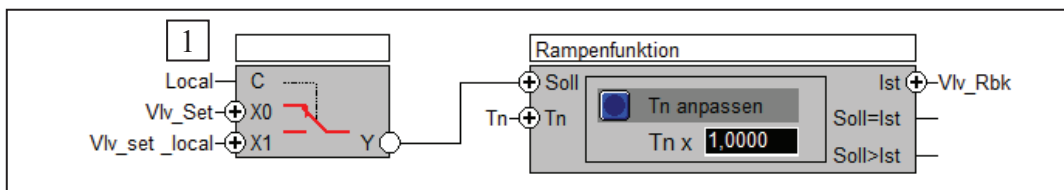


Abb. 36: Stellventil, Istwertbildung

Der Analogmultiplexer (1) erlaubt die Umschaltung der Sollwertquelle. Der Sollwert wird der Rampenfunktion übergeben, welche in Abhängigkeit von der Stellzeit „Tn“ den Istwert bildet.

4.5.1.2 Motoren

In verfahrenstechnischen Anlagen werden Motoren in ihrer Verwendung als Pumpen oder Kompressoren meist zur Realisierung von Stofftransporten oder als Rührer zur Stoffvermischung eingesetzt. Das Einsatzspektrum ist ebenso breit gefächert wie ihr technischer Aufbau. Im Zuge der immer leistungsfähiger werdenden Rechentechnik können heute mit dem Einsatz von Frequenzumrichtern auch Asynchronmotoren für schwierige regelungstechnische Aufgaben genutzt werden.

Wie bereits bei den Ventilen ist für den PCS 7 Projektteur insbesondere die Ansteuereinrichtung von Bedeutung. Welche technische Lösung sich hinter der Schnittstelle Ansteuereinrichtung – Antrieb befindet ist somit nicht relevant. So kann beispielsweise selbst ein Heizer wie ein direktstartender Motor betrachtet werden, da in beiden Fällen das Setzen eines Bits das Gerät einschaltet und der Zustand „Eingeschaltet“ durch Setzen eines Bits zurückgegeben wird.

Auf die Berechnung einer realitätsnahen Motorphysik soll verzichtet werden, da für den FAT lediglich die korrekte Übertragung und Anzeige von Prozessgrößen wie Strom oder Leistung von Interesse sind. Das Testen von Skalierungseinstellungen und der Meldeerzeugung nach Überschreiten zulässiger Grenzwerte ist so jedoch möglich und ausreichend. Die Kommunikation findet über Klemme oder Feldbus statt.

4.5.1.2.1 Direktstartender Motor

Ein Motor mit nur *einer* Festdrehzahl und *einer* Drehrichtung wird als Direktstarter bezeichnet.

Direktstartender Motor (Klemme)

Nachfolgend werden alle für die Erstellung des Makros wesentlichen Ein – und Ausgaben (aus Sicht der SPS) sowie Parametrierungen aufgeführt.

Signalausgänge

Tabelle 25: Direktstarter (Klemme), Signaleingänge

Binärausgänge	Erläuterung	Ausführung	Makrosignal Standardeinstellung
Ein	Einschaltung des Motors	Schalter als Schließer: 0 = Motor aus 1 = Motor ein	nicht invertiert
Reset	Quittierung eine Störung	Taster pos. Flanke= quittieren	nicht invertiert (Taster)
Kontrollleuchte	Spannungsversorgung und Kontakt zu Automatisierungssystem besteht	Schalter als Schließer: 0 = Leuchte aus 1 = Leuchte ein	nicht invertiert

Signaleingänge

Tabelle 26: Direktstarter, Signaleingänge

Binärausgänge	Erläuterung	Ausführung	Makrosignal Standardeinstellung
Betrieb	Motor ist eingeschaltet	Schalter als Schließer: 0 = Motor ausgeschaltet 1 = Motor eingeschaltet	nicht invertiert
Störung	Ausgabe einer (meiste nicht quittierpflichtigen) Störmeldung	Schalter als Öffner: 0 = Störung 1 = keine Störung	invertiert, nicht quittierpflichtig
Reparatur	Rückmeldung eines sich am Gerät befindenden Schalters. Schaltet den Motor stromlos. Der Störmeldung ähnlich wird der Reparaturschalter in der Simulation nur der Anzeige dienen.	Schalter als Öffner: 0 = Reparaturbetrieb 1 = kein Reparaturbetrieb	invertiert
Lokalbetrieb	Meldet Vorortsteuerung an das Automatisierungssystem	Schalter als Schließer: 0 = kein Lokalbetrieb 1 = Lokalbetrieb	nicht invertiert
Lokal Ein	Einschalten des Motors über Lokalsteuerung unter AS – Überwachung bei aktivem Lokalbetrieb	Taster: Pos. Flanke = einschalten	nicht invertiert (Ausführung als Schalter)

Binärausgänge	Erläuterung	Ausführung	Makrosignal Standardeinstellung
Lokal Aus	Schließen des Ventils über Lokalsteuerung unter AS – Überwachung bei aktivem Lokalbetrieb	Taster: Pos. Flanke = ausschalten	nicht invertiert (Ausführung als Schalter)
Nothalt	Rückmeldung eines sich am Gerät befindlichen Tasters. Stoppt den Motor. Der Störmeldung ähnlich wird der Nothalt in der Simulation nur der Anzeige dienen.	Rastender Taster als Öffner: 0 = Nothalt 1 = kein Nothalt	invertiert (Schalter)
Störung 2/3	weitere Störmeldungen	Schalter als Öffner: 0 = Störung 1 = keine Störung	invertiert, nicht quittierpflichtig
Analogeingänge	Erläuterung	Bedeutung für Makro	
Strom	Wert zwischen 0 und 100%; wobei 100% den Maximalwert des gewählten Messbereichs des Amperemeters darstellt und nicht die Obergrenze des zulässigen Stromwertes. [11]	Prozentuale Obergrenze für Stromwert muss einstellbar sein.	

Funktionsbaustein MotL

Tabelle 27: Direktstarter (Klemme), Parameter

Parameter	Erklärung	Bedeutung für Makro
MonTiDynamic	Nach erfolgter Ansteuerung, also dem Befehl den Motor ein oder auszuschalten, wird innerhalb eines bestimmaren Zeitraumes eine entsprechende Rückmeldung erwartet. Erfolgt diese nicht innerhalb des Zeitraumes, so wird eine Störmeldung ausgegeben. MonTiDynamic = 3 s ist der geringste Parameterwert.	Die Schaltzeit muss parametrierbar sein. Da das Anziehen des Netzschützes sowohl den Motor startet als auch die Rückmeldung „Betrieb“ verursacht, wird die Standardzeit auf 0 s gesetzt.

Das Makro „Motor_Dir“

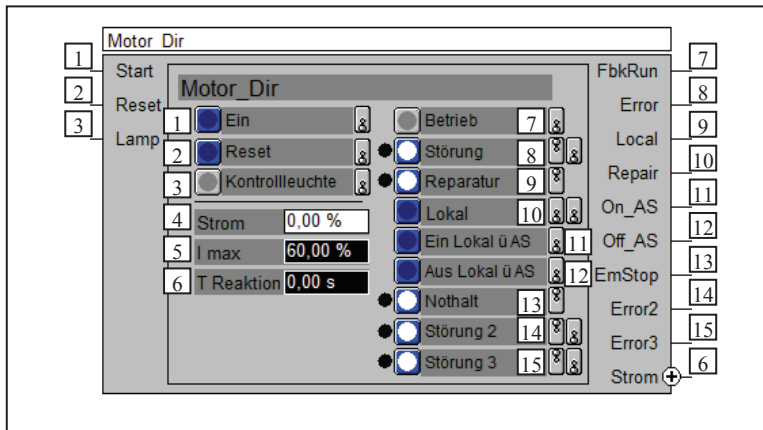


Abb. 37: Motor_Dir, Makro

Tabelle 28: Motor_Dir, Operanden

Element	Operanden	Erläuterung
(1) Start	Start_x	ein und ausschalten des Motors
(2) Reset	Reset_x	Quittieren jeder Störmeldung
(3) Kontrollleuchte	Lamp_x	Anzeige des Signals
(4) Iststrom	Strom	Ausgabe des aktuellen Stromwertes; Anzeigeformat: „Prozent[%] +/-120,00“
(5) Maximalstrom	Strom_Max	Eingabe eines maximalen Stromwertes; nach Einschalten des Motors wird dieser als Iststrom ausgegeben.
(6) Reaktionszeit	T_Reak	Einstellung einer Ein- und Ausschaltverzögerung des Motors (0 – 99999 s)
(7) Betrieb	FbkRun_x	Rückmeldung des Betriebszustandes (Motor ein- oder ausgeschaltet)
(8) Störung	Error_x	Ein- und Ausgabe einer Störmeldung gemäß Kap. 4.3.2 Störverhalten
(9) Reparatur	Repair_x	Ein- und Ausgabe einer Reparaturmeldung, schaltet den Motor nicht stromlos
(10) Betriebsart	Local_x	Wahl und Ausgabe der Betriebsart gemäß Kap. 4.3.3 Betriebsartenumschaltung (Schaltung 1)
(11) Einschalten über AS	On_AS	Lokaler Steuerbefehl zum Einschalten des Motors mit AS – Überwachung; (Ausführung als Schalter)
(12) Ausschalten über AS	Off_AS	Lokaler Steuerbefehl zum Ausschalten des Motors mit AS – Überwachung; (Ausführung als Schalter)
(13) Nothalt	Emstop_x	Ein- und Ausgabe einer Nothaltmeldung
(14) Störung 2	Error2_x	Ein- und Ausgabe einer Störmeldung gemäß Kap. 4.3.2 Störverhalten
(15) Störung 3	Error3_x	Ein- und Ausgabe einer Störmeldung gemäß Kap. 4.3.2 Störverhalten

Verschaltung

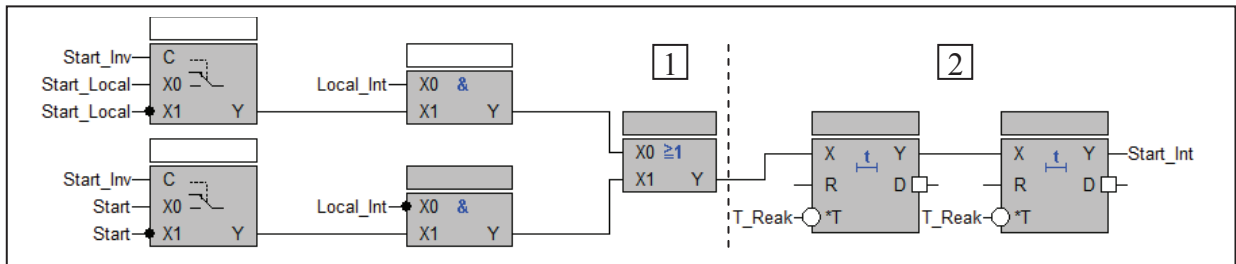


Abb. 38: Motor_Dir, Motorstart u. Zeitverzögerung

Der erste Teil des Schaltvorganges folgt dem Standardeingabeverhalten (siehe Abb. 10). Das Hintereinanderschalten der WinMOD Elemente „binäre Zeitverzögerung“ (2) mit den Parametrierungen „Einschaltverzögerung“ bzw. „Ausschaltverzögerung“ bewirken eine Reaktionsverzögerung um T_{Reak} .

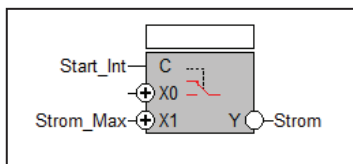


Abb. 39: Motor_Dir, Stromwert

Der Wechsel des Stromwertes zwischen 0 % und dem Maximalwert für den aus- bzw. eingeschalteten Motor erfolgt durch einen Analogmultiplexer. Verzögerungen oder Stromspitzen beim Anlauf sind nicht vorgesehen.

Die Befehls- und Statussignalinvertierung folgt den Vorlagen aus Kap. 4.3.1 „Bedienfeld“, Stör- und Resetverhalten der Vorlage des Kap. 4.3.2 „Störverhalten“ und die Betriebsartenumschaltung der Schaltung 1 aus Kap. 4.3.3 „Betriebsartenumschaltung“.

4.5.1.2.2 Direktstartender Motor (SIMOCODE pro)

SIMOCODE pro ist eine modular aufgebaute Motorsteuerung von Siemens zum Betrieb von Antrieben mit Festdrehzahl. Sie dient der Reduzierung des Verkabelungsaufwandes durch Einbindung von Hauptstromkreis, Steuerstromkreis und Automatisierungsebene und bietet durch projektierbare Steuer- und Überwachungsfunktionen Schutz für den angeschlossenen Antrieb auch bei Ausfall des Leit- oder Bussystems. Zudem ersetzt sie die Steuerkreishardware. [12, p. 25] Mit der Parametrierung der „Steuerfunktion“ kann ein Asynchronmotor in verschiedenen Schaltungsvarianten z.B. als Wendestarter oder mit einer Stern/Dreieckumschaltung betrieben werden. [12, p. 184] Das Hinzufügen von Erfassungsmodulen erlaubt die Messung von z.B. Strom- oder Leistungswerten des Motors. [12, p. 100]

Die logische Verbindung zwischen SIMOCODE pro und dem Leitsystem PCS 7 kann durch die Verwendung von eigens entwickelten Funktionsbausteinen hergestellt werden. Jeder dieser Funktionsbausteine realisiert *eine* Steuerfunktion. In den vorliegenden

Projekten wurde der Funktionsbaustein „SMC_Dir“ verwendet. Dieser steuert einen als Direktstarter parametrisierten SIMOCODE pro an und empfängt dessen Meldedaten.

Für Steuer- und Meldedaten werden jeweils zwei Byte genutzt. Folgende Tabellen zeigen die Bitbelegung der Steuer- und Statusworte:

Tabelle 29: SMC_Dir, Steuerwort [13, p. 63]

Bit	Steuerbefehl	Erläuterung
0	-	-
1	Aus	schaltet Motor aus
2	Ein (als Taster)	positive Flanke schaltet Motor ein, wenn kein Aus Befehl ansteht
3	Test	führt einen Funktionstest am Gerät durch und kann als Resetbefehl genutzt werden [12, p. 325]
4	Notstart	setzt Überlastauslösung vor Ablauf der Abkühlzeit zurück; mit „Ein“ kann Motor sofort erneut gestartet werden [12, p. 340]
5	Betrieb über Bus	umschalten zwischen Fernbetrieb und einer Lokalbetriebsart [12, p. 174]
6	Reset	Quittiert Störmeldung [12, p. 328]
7	-	
8	Benutzerspezifische Ausgänge	Zusatzbefehle

Tabelle 30: SMC_Dir, Statuswort [12, p. 63]

Bit	Statuswort	Erläuterung
0	-	-
1	Aus	Motor ausgeschaltet
2	Ein	Motor eingeschaltet
3	Vorwarnung Überlast	Strom übersteigt kurzzeitig eingestellten Grenzwert (nicht quittierpflichtig) [12, p. 502]
4	-	-
5	Fernbetrieb	Betriebsart Fernbetrieb aktiv
6	Sammelstörung	Störung erkannt; Störmeldung geht, wenn Fehlerursache beseitigt und/oder Störmeldung quittiert wurde [12, p. 492]
7	Sammelwarnung	nicht quittierpflichtiger Gerätestatus
8	-	-
9	Testposition	Test erfolgreich: Hauptstromkreis ist vom Netz getrennt [12, p. 330]
10 - 15	-	-

In einem Beispielprojekt wurden Stromerfassungsmodule genutzt:

Tabelle 31: SIMOCODE_Dir, Analogeingänge

Analogeingänge	Erläuterung	Normierung	Bedeutung für Makro
Strom	Wert zwischen 0 und 100 %; gibt den aktuellen Maximalwert der gemessenen Strangströme aus; 100% entsprechen dem „Einstellstrom“/ Nennstrom;	0 % ... 100 % \triangleq 0 ... 65535 [12, p. 542]	Prozentuale Obergrenze für Stromwert muss einstellbar sein; Externes Analogformat „Big Endian 0 ... 65535“ wird definiert

Das Makro „SIMOCODE_Dir“

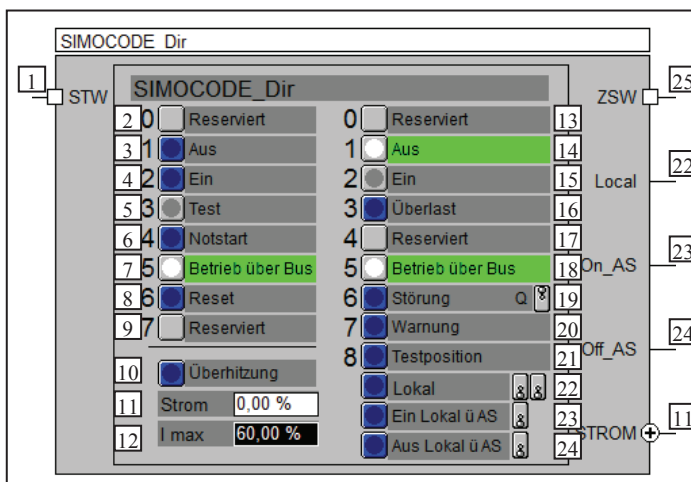


Abb. 40: SIMOCODE_Dir, Makro

Tabelle 32: SIMOCODE_Dir, Operanden

Element	Operanden	Erläuterung
(1) Steuerwort	STW	Anzeigeformat: „16 Bit hexadezimal“
(2) Reserviert	STW_Bit0	bei Bedarf Befehlsanzeige
(3) Aus	Off_x	Ausschalten des Motors (Taster)
(4) Ein	On_x	Einschalten des Motors (Taster)
(5) Test	Test_x	Anzeige des Signals; keine Funktion hinterlegt
(6) Notstart	EmStart_x	setzt „Überhitzung“ zurück
(7) Betrieb über Bus (Eingang)	Remote	Betriebsart „Fernbetrieb“, wenn „Lokalbetrieb“ nicht gesetzt (siehe Kap. 4.3.3 Betriebsartenumschaltung Schaltung 2)
(8) Reset	Reset_x	Quittieren jeder Störmeldung; betriebsartenunabhängig; Reset über Bus nur wirksam, wenn Motor aus
(9) Reserviert	STW_Bit7	bei Bedarf Befehlsanzeige
(10) Überhitzung	Overheat	setzen und rücksetzen von „Überhitzung“; erzeugt Störmeldung; Störmeldung auch bei anstehender Überhitzung quittierbar; schaltet Motor aus, verhindert Neustart
(11) Iststrom	STROM	Ausgabe des aktuellen Stromwertes Anzeigeformat: „Prozent[%] +/-120,00“

Element	Operanden	Erläuterung
(12) Maximalstrom	Strom_Max	Eingabe eines maximalen Stromwertes; Nach Einschalten des Motors wird dieser als Iststrom ausgegeben.
(13) Reserviert	ZSW_Bit0	Eingabe Bit 0 des Zustandswort
(14) Aus	ZSW_Off	Anzeige Motor ausgeschaltet
(15) Ein	ZSW_On	Anzeige Motor eingeschaltet
(16) Vorwarnung Überlast	Warning1	Eingabe einer Warnung
(17) Reserviert	ZSW_Bit4	Eingabe Bit 4 des Zustandswort
(18) Betrieb über Bus (Ausgang)	Remote	Ausgang gleich Eingang
(19) Störung	Error_x	Ein- und Ausgabe einer Störmeldung gemäß Kap. 4.3.2 Störverhalten
(20) Warnung	Warning2	Eingabe einer Warnung
(21) Testposition	Testpos	Statuseingabe; keine Funktion hinterlegt
(22) Betriebsart	Local_x	Wahl und Ausgabe der Betriebsart gemäß Kap. 4.3.3 Betriebsartenumschaltung (Schaltung 2)
(23) Einschalten über AS	On_AS	lokaler Steuerbefehl zum Einschalten des Motors mit AS – Überwachung; Ausführung als Schalter
(24) Ausschalten über AS	Off_AS	lokaler Steuerbefehl zum Ausschalten des Motors mit AS – Überwachung; Ausführung als Schalter
(25) Zustandswort	ZSW	gebildet aus Statusbit 0 – 8; Anzeigeformat: „16 Bit hexadezimal“

Verschaltung

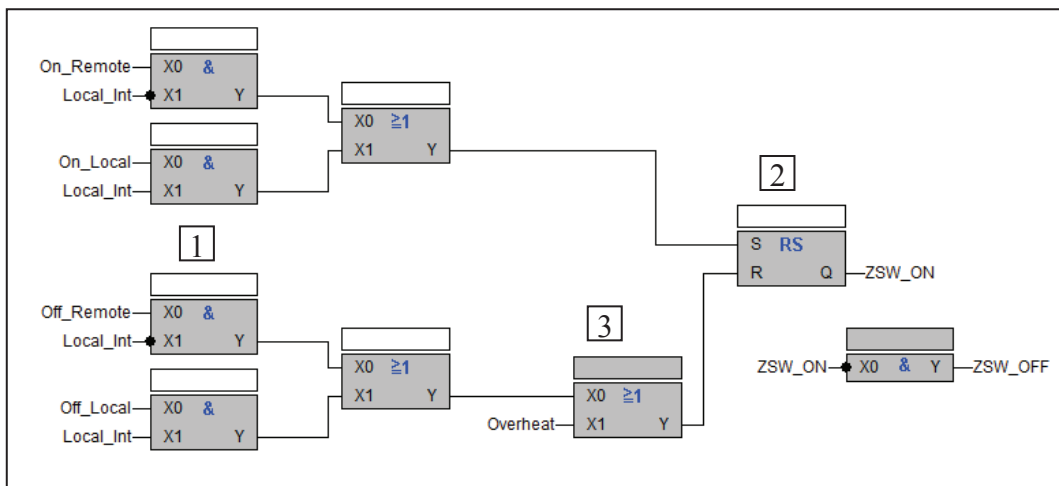


Abb. 41: SIMOCODE, Ein/Ausschalten

Das Beziehen der Steuerquelle folgt dem Beispiel aus Kap. 4.3.3 Schaltung 2. Das Ein- bzw. Ausschalten (1) des Motors erfolgt durch positive Flanken der Signale „On_x“ bzw. „Off_x“. Der Flipflop (2) ist rücksetzdominant. Das anstehende Signal „Overhead“ (3) schaltet den Motor aus und verhindert ein Wiedereinschalten.

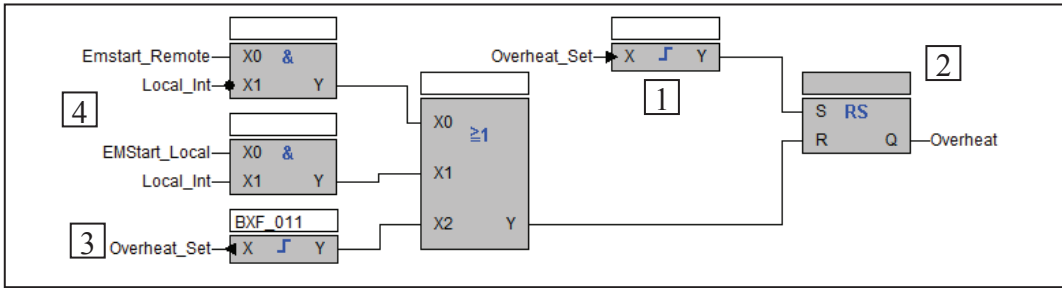


Abb. 42: SIMOCODE, Notstart

Das Einschalten (positive Flanke) von „Überhitzung“ (siehe Abb. 40: SIMCOCODE_Dir, Makro) erzeugt einen Impuls (1). Der Status „Overheat“ wird auf logisch 1 gesetzt (2). Das Ausschalten (negative Flanke) von „Überhitzung“ erzeugt einen Impuls (3). Der Status „Overheat“ wird auf logisch 0 gesetzt. Das Beziehen der Steuerquelle folgt dem Beispiel aus Kap. 4.3.3 Schaltung 2. Bei aktiviertem Notstart wird der Status „Overheat“ auf logisch 0 gesetzt (4).

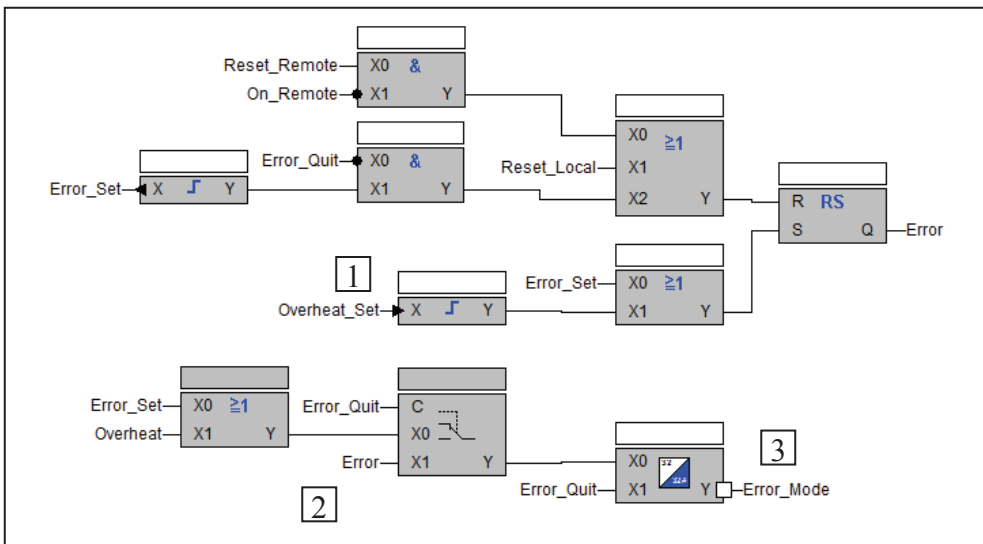


Abb. 43: SIMOCODE, Störverhalten

Der Grundaufbau des Störverhaltens gleicht dem Beispiel aus Abb. 17. Sie wurde um einige Signale der „Überhitzung“ erweitert. Das Einschalten (positive Flanke) der „Überhitzung“ erzeugt einen Impuls und setzt „Error“ auf logisch 1. Das Signal „Overheat“ (2) färbt, wie auch „Error_Set“ die Statuszeile „Störung“ rot (3). Somit ist „Überhitzung“ als Sammelstörung ausgegeben.

Die Verarbeitung des Stromwertes geschieht wie im Makro „Motor_Dir“ (Abb. 39: Motor_Dir, StromwertVerzögerungen oder Stromspitzen beim Anlauf sind nicht vorgesehen).

Die Statussignalinvertierung folgt den Vorlagen aus Abschnitt 4.3.1 „Bedienfeld“ und die Betriebsartenumschaltung der Vorlage 2 aus dem Abschnitt 4.3.3 „Betriebsartenumschaltung“.

4.5.1.2.3 Drehzahl geregelter Motor mit zwei Drehrichtungen

Die Drehzahlregelung von Asynchronmotoren kann durch Frequenzumrichter vorgenommen werden. Eine standardisierte Kommunikation zwischen SPS und der Antriebsregelung kann mit dem PROFIBUS Telekommunikationsprofil PROFIDrive hergestellt werden.

Dieses Gerätekommunikationsprofil wurde von verschiedenen Antriebsherstellern entwickelt, um den Integrations- und Inbetriebnahmeaufwand für den Anwender von Antriebsgeräten in der Automatisierungstechnik zu vereinfachen, indem der Austausch von Steuerbefehlen und Statusmeldungen sowie von Soll- und Istwerten zwischen Antriebseinheiten und der übergeordneten SPS vereinheitlicht wird. Trotz Standardisierung könnten Antriebsregelungen in Details Abweichungen in ihrem Verhalten aufweisen. Alle Ausführungen dieses Kapitels beziehen sich auf den Siemens SINAMICS G120C.

Um das PROFIDrive-Profil in unterschiedlichen Anwendungsbereichen einsetzen zu können, sind verschiedene Parameter – Prozessdaten – Objekt – Typen (PPO – Typen) definiert und für den jeweiligen Einsatzbereich optimiert. Sie unterscheiden sich im Wesentlichen in ihrer Struktur und Art der Steuerbefehle und Statusmeldungen sowie der Art und Anzahl der übertragenen Soll – und Istwerte.

Für Aufgaben der Verfahrenstechnik kann der PPO – Typ Standardtelegramm 20 genutzt werden. Der Kanaltreiberbaustein FbDrive, welcher in der PCS 7 Advanced Process Library (APL) ab der Version 8 zur Verfügung steht, bildet die logische Schnittstelle zwischen dem SPS – Anwenderprogramm und der Antriebssteuerung des Frequenzumrichters. Er wird zusammen mit den Telegrammtypen 1 oder 20 verwendet, welche sich nur in der Anzahl der übertragenen Istwerte unterscheiden.

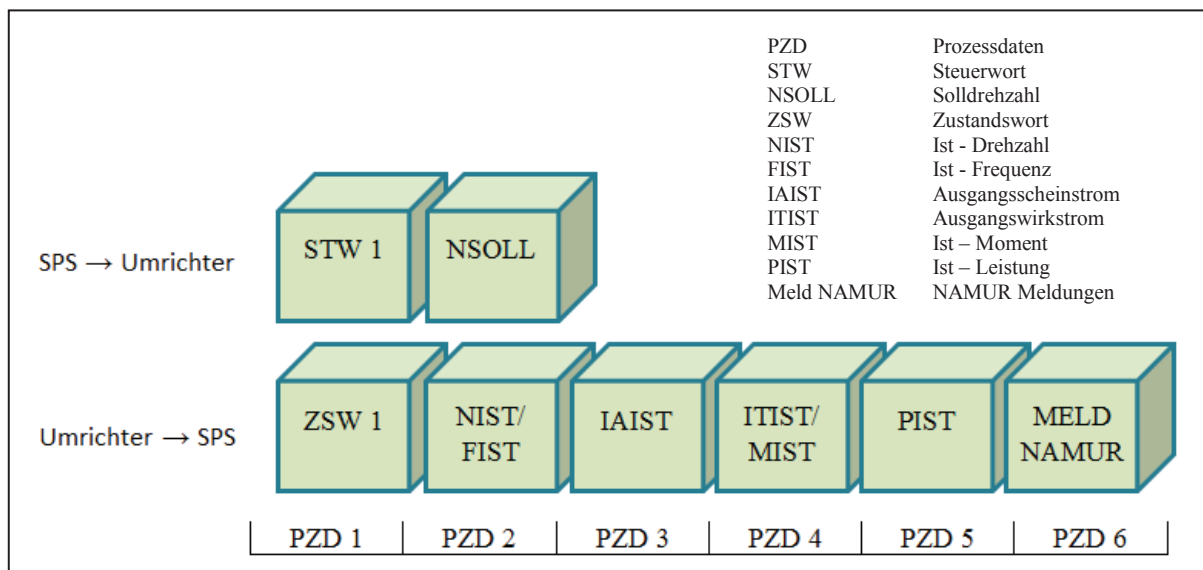


Abb. 44: Prozessdaten – Bereiche des Standardtelegramm 20

Die PZD 1 beider Senderrichtungen und das PZD 6 sind 16 Bit Datenworte mit binären Steuerbefehlen bzw. Statusmeldungen. Alle übrigen Datenworte sind Analogwerte mit 16 – Bit – Auflösung (Bit 15 als Vorzeichenbit). Der Telegrammtyp 1 überträgt nur die PZD 1 und PZD 2.

Tabelle 33: PROFIdrive Telegramm 1/20, Analogwerte [5, p. 9]

Analogwerte	Erläuterung	Normierung	Bedeutung für Makro
PZD 2 bis PZD 5	Wert zwischen -200 % und 200 %	0% ... 100% \triangleq 0 ... 16384	Da auf eine realitätsnahe Berechnung der physikalischen Werte verzichtet werden kann, sollen sich alle Istwerte am Wert der Soll Drehzahl orientieren. Das externe Analogformat „Big Endian -32768 ... 32767“ und das interne Analogformat „Prozent[%]+_-200,00“ werden definiert.

Steuer – und Zustandsworte

Das Steuerwort 1

Tabelle 34: PROFIdrive Telegramm 1/20, Steuerwort

Bit	Steuerwort (Bit = 1)	Erläuterung
0	EIN	siehe Abb. 45: Zustandsdiagramm aus PROFIdrive Version 4.1
1	Kein AUS 2 (kein Austrudeln)	siehe Abb. 45: Zustandsdiagramm aus PROFIdrive Version 4.1
2	Kein AUS 3 (kein Schnellhalt)	siehe Abb. 45: Zustandsdiagramm aus PROFIdrive Version 4.1
3	Freigabe Betrieb	siehe Abb. 45: Zustandsdiagramm aus PROFIdrive Version 4.1
4	Freigabe Rampenausgang	= 0: schnellstmöglicher Bremsvorgang = 1: Hoch-/Rücklauframpe werden berechnet [3, p. 96]
5	Freigabe Steilheitsbegrenzer	= 0: „Einfrieren“ des Hochlaufgebers [3, p. 96]
6	Freigabe Sollwert	= 0: bremsen gemäß Rücklauframpe = 1: beschleunigen gemäß Hochlauframpe [3, p. 96]
7	Fehlerquittierung (0→1)	Quittiert Störmeldung
8	Tippbetrieb 1 EIN	Festdrehzahl 1 [3, p. 178]
9	Tippbetrieb 2 EIN	Festdrehzahl 2 [3, p. 178]
10	Betrieb über Bus	umschalten zwischen Fernbetrieb und einer Lokalbetriebsart
11	Invertierung Sollwert	Drehzahl wird gemäß Rücklauframpe auf 0 abgesenkt und anschließend gemäß Hochlauframpe erhöht
12	Reserviert	Reserviert
13	Reserviert	Reserviert
14	Reserviert	Reserviert
15	Parametersatz 2	Wechsel des Befehlssatzes

Das Zustandswort 1

Im Zustandswort werden die Statusmeldungen des Umrichters ausgegeben.

Tabelle 35: PROFIdrive Telegramm 1/20, Zustandswort

Bit	Zustandswort (Bit = 1)	Erläuterung
0	Einschaltbereit	siehe Abb. 45: Zustandsdiagramm aus PROFIdrive Version 4.1
1	Betriebsbereit	siehe Abb. 45: Zustandsdiagramm aus PROFIdrive Version 4.1
2	Betrieb freigegeben	siehe Abb. 45: Zustandsdiagramm aus PROFIdrive Version 4.1
3	Störung	Störung erkannt; Störmeldung geht, wenn Fehlerursache beseitigt und/oder Störmeldung quittiert wurde bzw. nach Neustart [3, p. 259]
4	Kein AUS 2	kein Austrudeln oder keine zwangsläufige Netztrennung
5	Kein AUS 3	kein Schnellhalt oder keine externe Verriegelung
6	Einschaltsperr	siehe Abb. 45: Zustandsdiagramm aus PROFIdrive Version 4.1
7	Warnung	nicht quittierpflichtiger Gerätestatus [3, p. 256]
8	Frequenz oder Drehzahl im Sollwertbereich	Istwert = interne Sollwertvorgabe [14, p. 653]
9	Steuerung angefordert	keine Steuerung angefordert
10	Frequenz oder Drehzahl erreicht oder überschritten	einstellbarer Wert wurde überschritten; gibt meist einen Gutzustand aus [14, p. 279]
11	Eingestellte Strom- oder Drehmomentgrenze nicht erreicht	einer der einstellbaren Werte wurde überschritten; gibt meist einen Gutzustand aus
12	Reserviert	-
13	Keine Motor Überlast	Motor - Übertemperatur
14	Drehrichtung positiv	Motorrechtslauf
15	Parametersatz 2 aktiv	Rückmeldung 2. Befehlssatz aktiv

NAMUR – Meldungen

Die NAMUR – Meldungen sind das Störungswort nach VIK – NAMUR – Definition. Sie sollen aus der Vielzahl der durch den Umrichter detektierbaren Störmeldungen den wesentlichen Teil übersichtlich zusammenfassen.

Tabelle 36: PROFIdrive Telegramm 1/20, NAMUR - Meldungen [5, p. 12]

Bit	Bedeutung (Bit=1)
0	Fehler Umrichterinformationselektronik / Softwarefehler
1	Netzfehler
2	Zwischenkreisüberspannung
3	Fehler Umrichterleistungselektronik
4	Übertemperatur Umrichter
5	Erdschluss
6	Überlast

Bit	Bedeutung (Bit =1)
7	Kommunikationsfehler
8	Externe Sicherheitsabschaltung
9	Fehler Drehzahlerfassung
10	Fehler interne Kommunikation
11	Fehler Zwischenkreisspannung
12	Reserviert
13	Reserviert
14	Reserviert
15	Sonstiger Fehler

Motorbetrieb

Der Motorbetrieb unterteilt sich in die Ein-/Ausschalterschrittfolge und die Sollwertbildung. Auf die Ein-/Ausschalterschrittfolge wirken die Bits 0 – 3 des Steuerwortes, auf die Sollwertbildung die Bits 4 – 6.

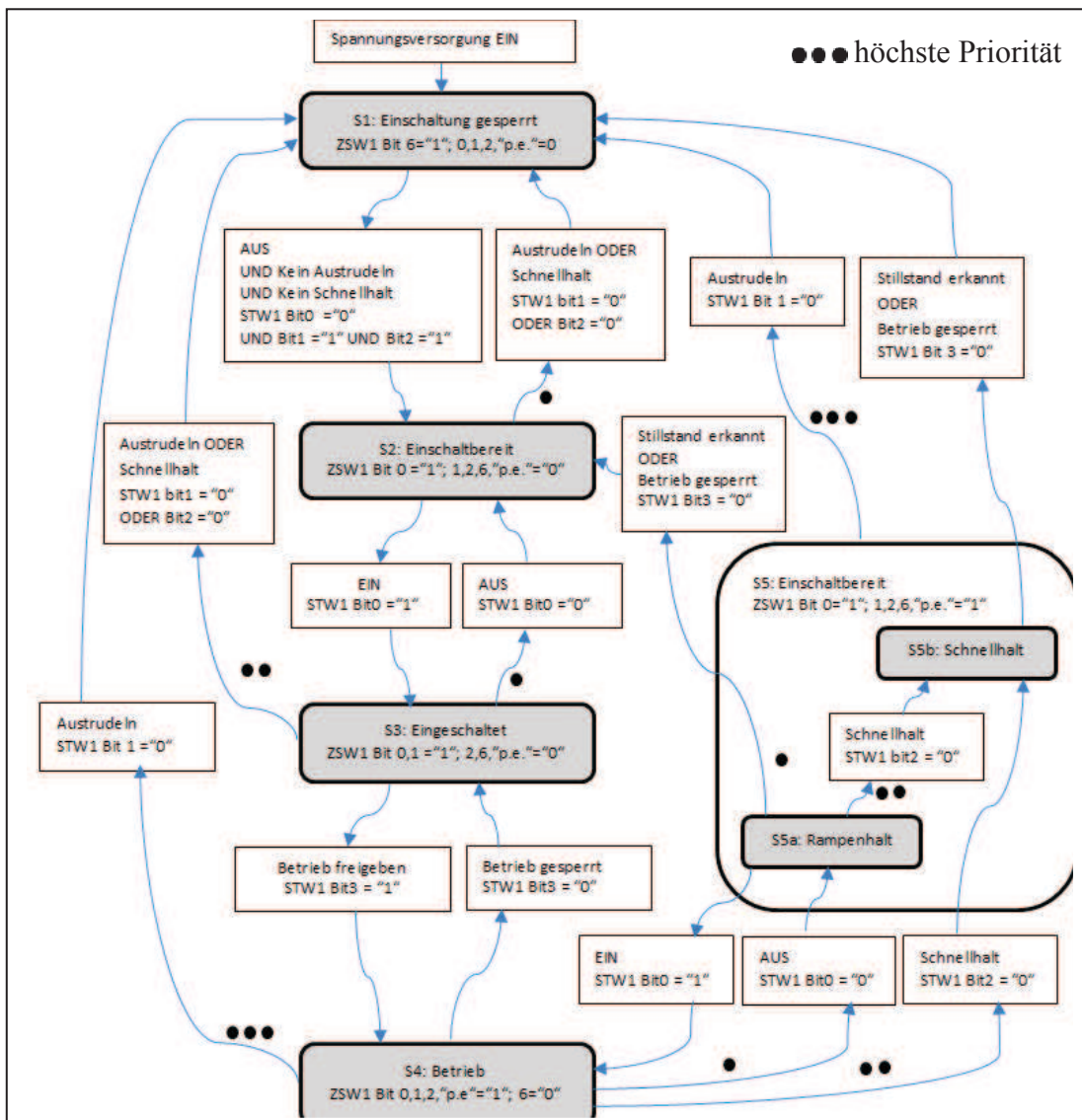


Abb. 45: Zustandsdiagramm aus PROFIdrive Version 4.1, Ein-/Ausschalterschrittfolge [5, p. 22]

Das Diagramm beschreibt den gestuften Ein-/Ausschaltvorgang des Motors unter Wirkung der Bits 0 bis 3 des Steuerwortes, welche als Weiterschaltbedingungen fungieren. Während der Ausführung eines Schrittes wird das entsprechende Bitmuster (Bits 0,1,2 und 6) in das Zustandswort geschrieben. In Schritten, in denen der Pulsgenerator also ausgeschaltet wird (p.e. = 0), wird der Motor bis zum Stillstand austrudeln. Wenn p.e. = 1, so wird, unter Beachtung der weiteren Sollwertverarbeitung, mit der jeweiligen Rampenfunktion verfahren.

An den Ein-/Ausschaltvorgang schließt sich die Sollwertverarbeitung an. Erst im Schritt 4 „Betrieb“ stehen die Funktionen der Sollwertverarbeitung zur Verfügung. Die Bits 4 – 9 und Bit 11 des Steuerwortes führen diese Funktionen aus. [15, p. 51]

Folgende Darstellung zeigt das Zusammenwirken der Befehle (Bit 7 und 8 sind nicht dargestellt).

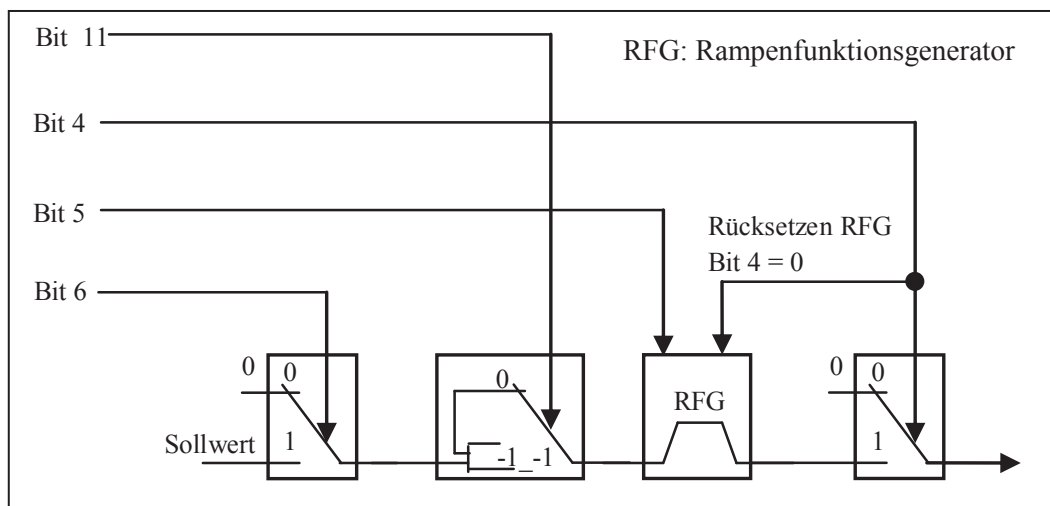


Abb. 46: PROFIdrive, Sollwertverarbeitung [5, p. 23]

Der Rampenfunktionsgenerator (RFG), auch Hochlaufgeber genannt, dient der ruckfreien Drehzahlverstellung eines Asynchronmotors. Dabei wird die Frequenz des antreibenden Stromes linear erhöht oder gesenkt, bis die Sollfrequenz erreicht ist. Die Drehzahl des Motors folgt der Kennlinie mit einer durch trägheits- und lastmomentbeaufschlagten Verzögerung.

Der RFG dient der Verminderung von zu hohen Momenten bei Brems- und Beschleunigungsvorgängen, wodurch die Mechanik der angetriebenen Maschine geschont wird. Außerdem sind Beschleunigungs- und Bremsdauer des Motors hierdurch weitestgehend lastunabhängig und können frei eingestellt werden.

Parametrierbar sind die Hochlaufzeit, die Rücklaufzeit und die Rücklaufzeit für den Schnellhalt. Die Hochlaufzeit kennzeichnet die Beschleunigung des Motor, die Rücklaufzeiten das Abbremsen. Es wird die Zeit für eine Drehzahländerung von 100% (0 % → 100 % bzw. 100 % → 0 %) parametrierbar.

Die Abb. 47: Rampenverläufe“ zeigt einen möglichen Drehzahlverlauf bei folgender Parametrierung des RFG.

- Hochlaufzeit = 10s
- Rücklaufzeit = 5s
- Rücklaufzeit für Schnellhalt = 3s

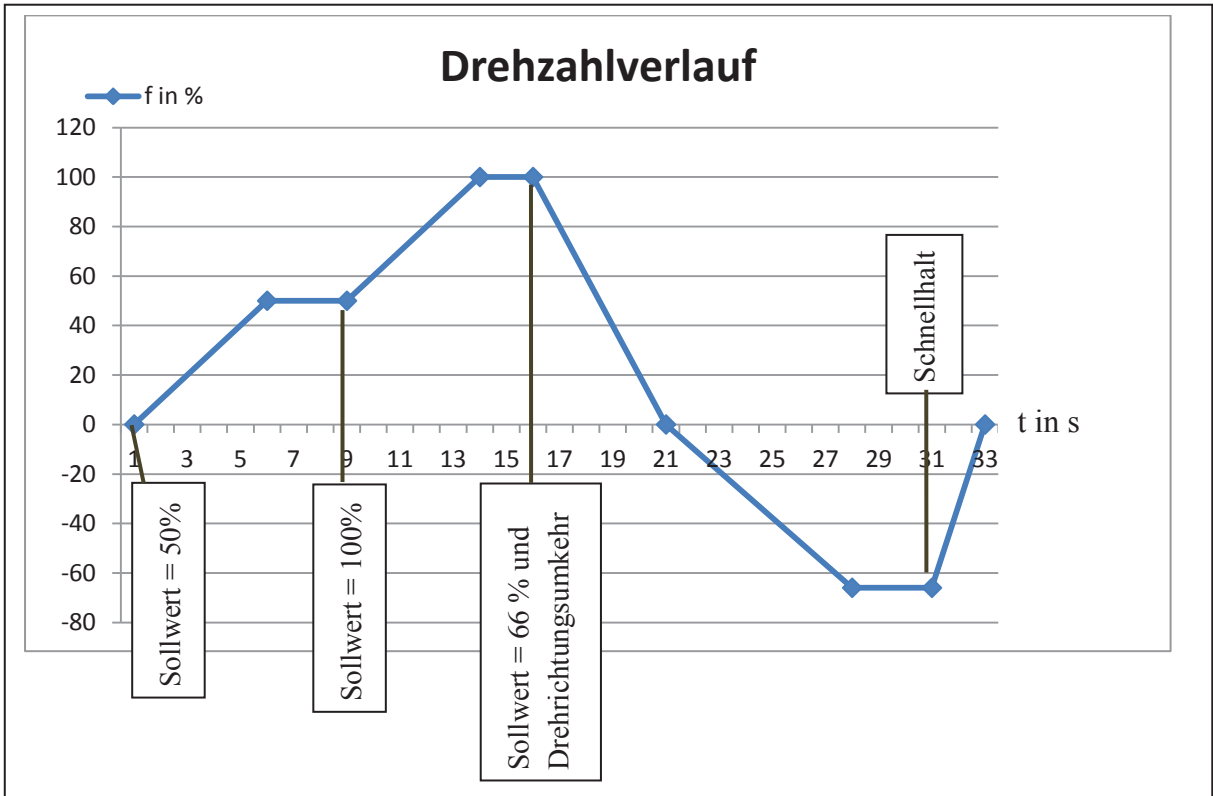


Abb. 47: Rampenverläufe

Die Sperrung des RFG setzt die Ausgangsfrequenz auf 0 Hz (Gleichspannung), wodurch der Motor schnellstmöglich zum Stillstand gebracht wird.

Das Makro „MotorSpdC_Tele1_20“

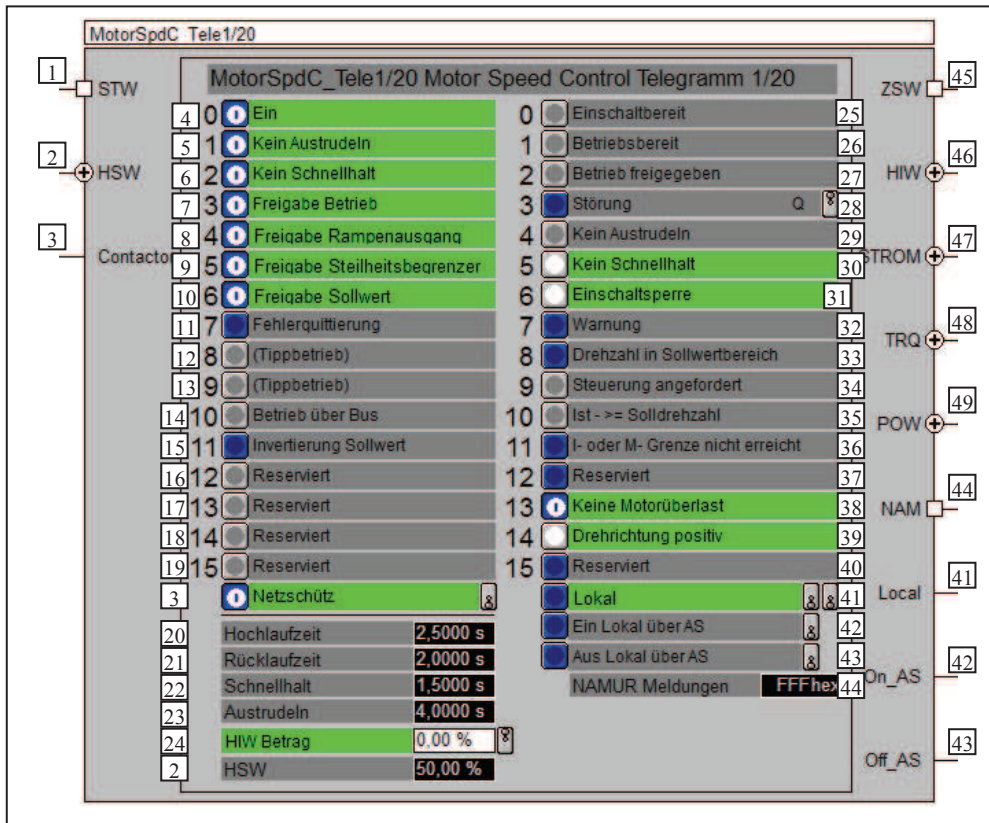


Abb. 48: MotorSpdC_Tele1_20

Tabelle 37: MotorSpdC_Tele1_20, Operanden

Element	Operanden	Erläuterung
(1) Steuerwort	STW	Anzeigeformat: „16 Bit hexadezimal“
(2) Hauptsollwert	HSW	Drehzahlsollwert; Anzeigeformat: „Prozent[%] +/-200,00“
(3) Netzschütz	Contactora_x	muss für Ein-/Ausschaltvorgang gesetzt sein (s. Abb. 49); keine Störungsquittierung
(4) Ein	STW_Bit0_x	Einschalten des Motors (s. Abb. 49)
(5) kein Austrudeln	STW_Bit1_x	= 0: austrudeln bis Stillstand (s. Abb. 49)
(6) kein Schnellhalt	STW_Bit2_x	= 0: Schnellhalt (s. Abb. 49)
(7) Freigabe Betrieb	STW_Bit3_x	= 0: austrudeln bis Stillstand = 1: Hochlauf bis Solldrehzahl
(8) Freigabe Rampenausgang	STW_Bit4_x	= 0: Motor stoppt ohne Verzögerung = 1: Hoch-/Rücklaufrahmen werden berechnet
(9) Freigabe Steilheitsbegrenzer	STW_Bit5_x	= 0: Drehzahl wird eingefroren = 1: Hoch-/Rücklaufrahmen werden berechnet
(10) Freigabe Sollwert	STW_Bit6_x	= 0: bremsen gemäß Rücklaufrahmen = 1: beschleunigen gemäß Hochlaufrahmen
(11) Fehlerquittierung	STW_Bit7_x	Quittieren jeder Störmeldung; betriebsartenunabhängig
(12) Tippbetrieb 1	STW_Bit8_x	bei Bedarf Befehlsanzeige (keine Funktion hinterlegt)
(13) Tippbetrieb 2	STW_Bit9_x	bei Bedarf Befehlsanzeige (keine Funktion hinterlegt)

Element	Operanden	Erläuterung
(14) Betrieb über Bus	STW_Bit10_x	Betriebsart „Fernbetrieb“, wenn „Lokalbetrieb“ nicht gesetzt (siehe Kap. 4.3.3, Schaltung 2)
(15) Invertierung Sollwert	STW_Bit11_x	Drehzahl wird gemäß Rücklauftrampe auf 0 abgesenkt und anschließend gemäß Hochlauftrampe erhöht
(16) Reserviert	STW_Bit12_x	bei Bedarf Befehlsanzeige
(17) Reserviert	STW_Bit13_x	bei Bedarf Befehlsanzeige
(18) Reserviert	STW_Bit14_x	bei Bedarf Befehlsanzeige
(19) Reserviert	STW_Bit15_x	bei Bedarf Befehlsanzeige
(20) Hochlaufzeit	HLZ	Dauer Drehzahländerung 0 → 100 %; Einstellbereich 0 – 99999 s
(21) Rücklaufzeit	RLZ	Dauer Drehzahländerung 100 → 0 %; Einstellbereich 0 – 99999 s
(22) Schnellhalt	Q_RLZ	Dauer Drehzahländerung 100 → 0 %; Einstellbereich 0 – 99999 s
(23) Austrudeln	Coast_Stop	Dauer Drehzahländerung 100 → 0 ; Einstellbereich 0 – 99999 s
(24) Hauptistwert (Betrag)	HIW_Betrag	Auswahl Ausgabe der Drehzahl als Betrag oder vorzeichenbehaftet; Anzeige der Drehzahl
(25) Einschaltbereit	ZSW_Bit0	(s. Abb. 49)
(26) Betriebsbereit	ZSW_Bit1	(s. Abb. 49)
(27) Betrieb freigegeben	ZSW_Bit2	(s. Abb. 49)
(28) Störung	ZSW_Error	Ein- und Ausgabe einer Störmeldung gemäß Abs. 4.3.2 „Störverhalten“
(29) kein Austrudeln	ZSW_Bit4	= 0, wenn Austrudeln aktiv
(30) kein Schnellhalt	ZSW_Bit5	= 0, wenn Schnellhalt aktiv
(31) Einschaltsperr	ZSW_Bit6	(s. Abb. 49)
(32) Warnung	ZSW_Warnung	Eingabe einer Warnung
(33) Drehzahl in Sollwertbereich	ZSW_Bit8	Eingabe manuell
(34) Steuerung angefordert	STW_Bit10	= 1, Betrieb über Bus
(35) Ist >= Vergleichsdrehzahl	ZSW_Bit10	= 1, wenn Drehzahl ≠ 0 (Vergleichsdrehzahl = 0%)
(36) I- oder M-Grenze nicht erreicht	ZSW_Bit11	Eingabe manuell
(37) Reserviert	ZSW_Bit12	Eingabe manuell
(38) keine Motorüberlast	ZSW_Bit13	Eingabe manuell
(39) Drehrichtung positiv	ZSW_Direct_Pos	= 1, wenn Drehrichtung positiv
(40) Reserviert	ZSW_Bit15	Eingabe manuell
(41) Betriebsart	Local_x	Wahl und Ausgabe der Betriebsart gemäß Abs. 4.3.3 Schaltung 2
(42) Einschalten über AS	On_AS	Lokaler Steuerbefehl zum Einschalten des Motors mit AS – Überwachung (Ausführung als Schalter)
(43) Ausschalten über AS	Off_AS	Lokaler Steuerbefehl zum Ausschalten des Motors mit AS – Überwachung (Ausführung als Schalter)
(44) NAMUR Meldungen	NAM	Eingabe manuell: 0 - FFFF _{Hex} ; Anzeigeformat: „16 Bit hexadezimal“

Element	Operanden	Erläuterung
(45) Zustandswort	ZSW	gebildet aus Statusbit 0 – 15: 0 – FFFF _{Hex} ; Anzeigeformat: „16 Bit hexadezimal“
(46) Hauptistwert	HIW	Ausgabe der aktuellen Drehzahl, Betrag oder vorzeichenbehaftet; Anzeigeformat: „Prozent[%] +/-200,00“
(47) Strom	STROM	Ausgabe des aktuellen Stromwertes Anzeigeformat: „Prozent[%] +/-200,00“; gleich dem Wert der Drehzahl (nur positiv)
(48) Moment	TRQ	Ausgabe des aktuellen Momentenwertes Anzeigeformat: „Prozent[%] +/-200,00“; gleich dem Wert der Drehzahl (nur positiv)
(49) Leistung	POW	Ausgabe des aktuellen Leistungswertes Anzeigeformat: „Prozent[%] +/-200,00“; gleich dem Wert der Drehzahl (nur positiv)

Verschaltung

Ein-/Ausschaltvorgang

Als Grundlage der Programmierung dieses Vorganges dient das PROFIdrive Zustandsdiagramm Abb. 45. Das Diagramm soll als Ablaufkette mit jedem dargestellten Zustand und jeder Rampenfunktion nachgebildet werden. Um die Programmierung zu erleichtern, wurde das Diagramm zunächst leicht abgewandelt. So wurde der Schritt 1 (Einschaltung gesperrt) in zwei Schritte unterteilt, da in diesem 2 verschiedene Rampenfunktionen verwendet werden können: der Schnellhalt und das Austrudeln. Das Austrudeln wird in der Programmierung als eigenständige Rampenfunktion gehandhabt. Ebenso wurde der Schritt 5 in zwei Schritte aufgeteilt: Rampenhalt und Schnellhalt. Weiterhin wichen die Prioritätssymbole zusätzlichen Befehlen.

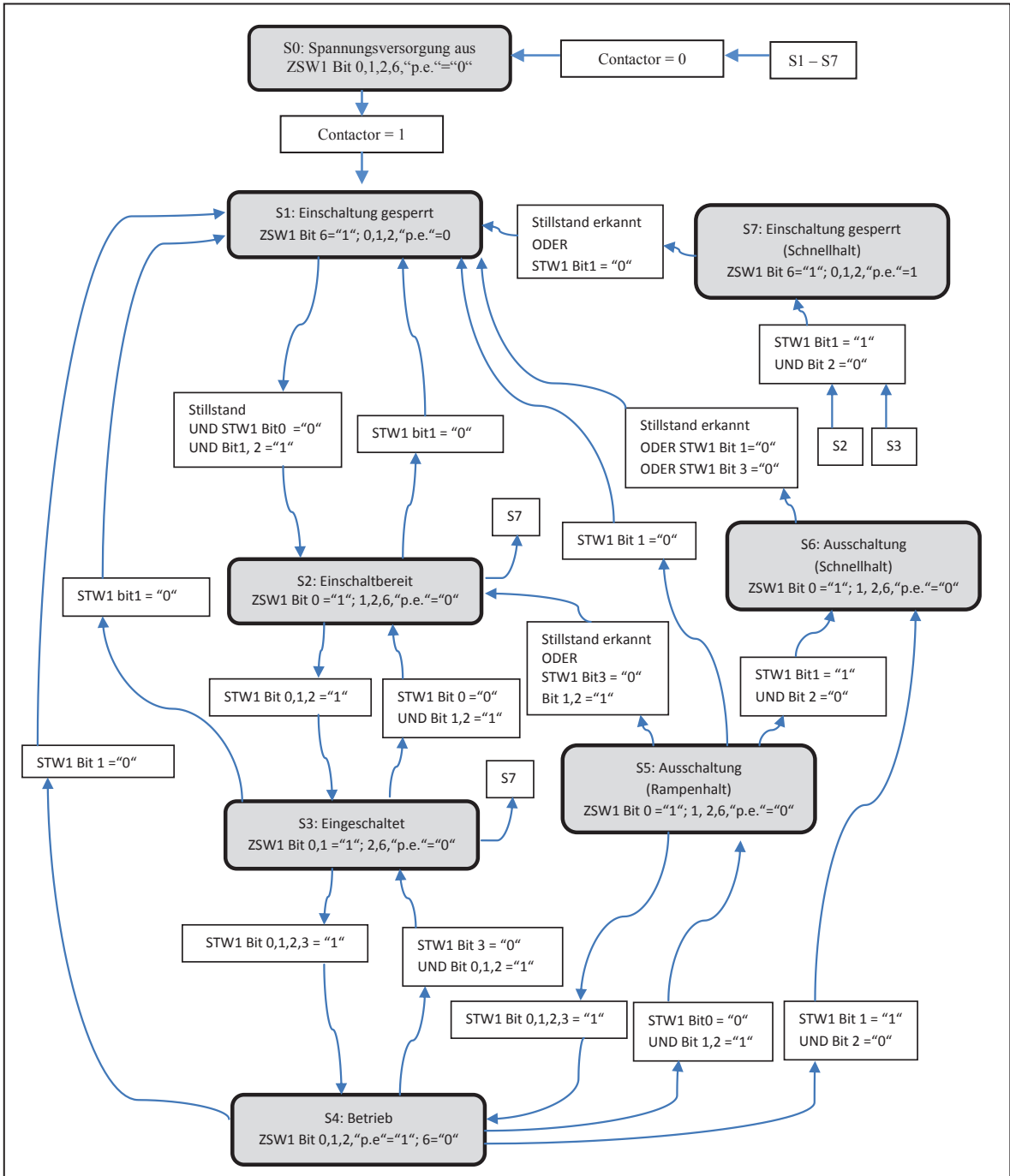


Abb. 49: MotorSpdC_Tele1_20, Zustandsdiagramm

WinMOD bietet eigens entwickelte Elemente zur Schrittkettenprogrammierung:

State Machine

Die State Machine ist das Kernstück des Steuernetzes. Die State Machine stellt das Betriebsartenschaltwerk, visualisiert die aktuellen Schritte und kontrolliert die 1 aus n Bedingungen. [1, p. 8.219]

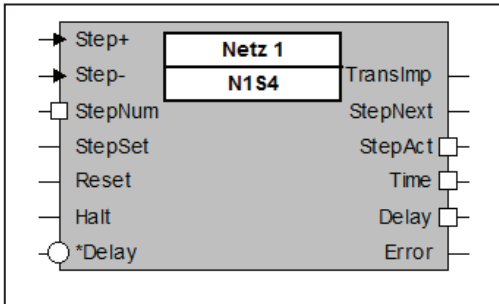


Abb. 50: MotorSpdC_Tele1_20, State Machine 1

Die Schrittkette erhielt die Nummer 1.

State Element

Das State Element repräsentiert einen Schritt in der Ablaufkette. Am Eingang „Trans“ können Transitionen festgelegt werden.

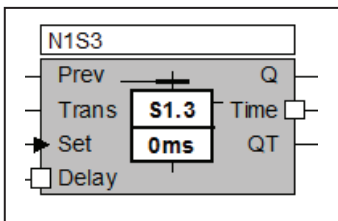


Abb. 51: MotorSpdC_Tele1_20, State Element, Schritt3

Jeder Schritt der Kette erhielt ein State Element, dem die entsprechende Netz- und Schrittnummer zugewiesen wurde.

NS Befehl

Not Stored – Befehle werden mit dem State Element – Ausgang „Q“ verschaltet. Am Ausgang des NS – Befehls steht ein Operand. Da Operanden von verschiedenen State Elementen gleichsam beschrieben werden (Schritt 1 aktiv: Motor_On = 1; Schritt 2 inaktiv: Motor_On = 0), kann es zu logischen Kurzschlüssen kommen, da ein Eingang nur von einer Stelle beschrieben werden kann. Dies verhindert der NS Befehl, indem nur solche NS Befehls – Ausgänge beachtet werden, die dem gerade aktiven Schritt zugeordnet sind.

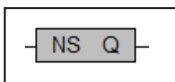


Abb. 52: NS Befehl

Beispiel Schritt 3

Abb. 53 zeigt beispielhaft die Verschaltung eines Schrittes, welche sich aus dem Zustandsdiagramm Abb. 49 ableitet.

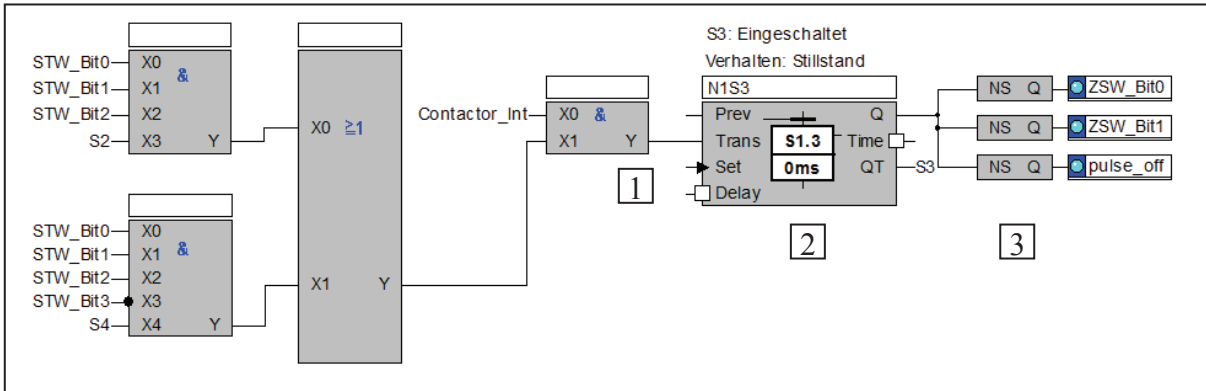


Abb. 53: MotorSpdC_Tele1_20, Schrittkette1, Schritt3

Tabelle 38: MotorSpdC_Tele1_20, Netz 1 Schritt 3

Element	Erläuterung
(1) Transition	Weiterschaltung aus Schritt 2 oder 4 (S2; S4) in den Schritt 3 bei erfüllten Weiterschaltbedingungen
(2) State Element	aktiv, wenn Trans = 1; wenn aktiv, Q = 1 und QT = 1; mit QT verschalteter Operand „S3“ dient als Weiterschaltbedingung in Schritte 0,1,2,4 und 7
(3) NS Befehle	Zustandsausgabe Schritt 3

Alle weiteren Schritte wurden nach demselben Schema erstellt.

Sollwertverarbeitung

Die Verschaltung zur Sollwertverarbeitung leitet sich aus der Abb. 46: PROFIdrive, Sollwertverarbeitung ab.

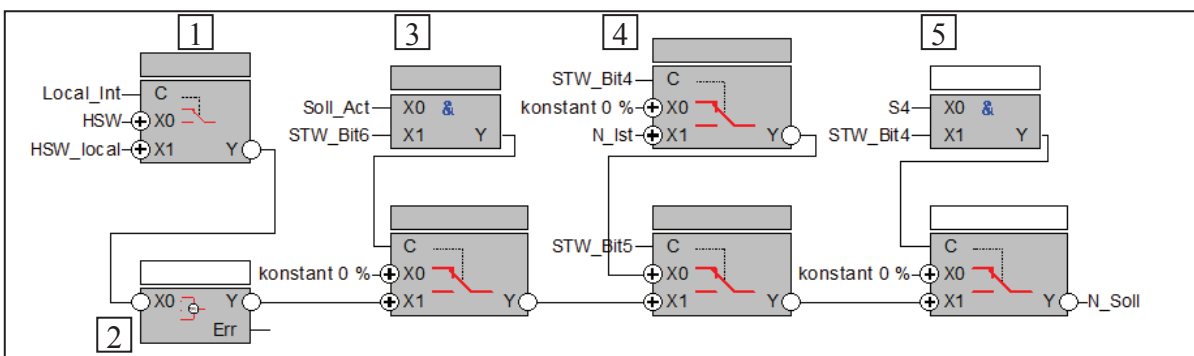


Abb. 54: MotorSpdC_Tele1_20, Sollwertverarbeitung

Tabelle 39: MotorSpdC_Tele1_20, Sollwertverarbeitung, Erläuterung

Element	Erläuterung
(1) Sollwertquelle	Umschaltung der Sollwertquelle auf „Lokal“ mit Local_Int = 1
(2) Formel	intern wird Sollwert als Betrag weiterverarbeitet
(3) Soll_Act; STW_Bit6	Soll_Act: siehe Abb. 56 wenn Soll_Act und STW_Bit6 = 1, dann wird HSW weitergeleitet
(4) STW_Bit4; STW_Bit5	wenn STW_Bit5 = 0 und STW_Bit4 = 1, dann wird HSW „eingefroren“ und weitergeleitet; wenn STW_Bit4 = 0, dann wird internen Sollwert = 0% weitergeleitet
(5) STW_Bit4; Schritt 4	Befindet sich die Schrittkette 1 im Schritt 4 und ist STW_Bit4 = 1, dann wird HSW weitergeleitet; HSW wird intern als N_Soll weitergeleitet

Rampenzeiten

Im realen Frequenzumrichter sind Rampenzeiten (Hoch-/Rücklaufzeit; Schnellhalt) durch Eingriff über z.B. ein umrichtereigenes Bedienpanel parametrierbar. In der Simulation soll auch die Zeit des Austrudeln parametrierbar sein. Verzichtet wurde hingegen auf die Verstellbarkeit der Stoppzeit durch das Gleichstrombremsen (Steuerbit 4).

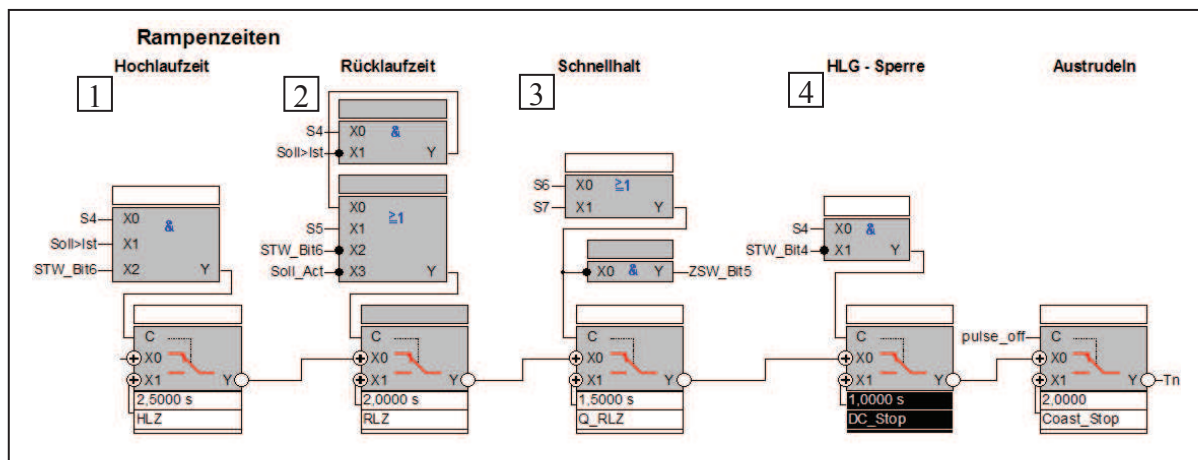


Abb. 55: Rampenzeiten, Verschaltung

Tabelle 40: Rampenzeiten, Verschaltung, Erläuterung

Element	Erläuterung
(1) Hochlaufzeit	aktiv, wenn sich die Schrittkette 1 im Schritt 4 befindet, der Sollwert über dem Istwert liegt und das Steuerbit 6 = 1 ist
(2) Rücklaufzeit	aktiv, wenn Rampenhalt oder Drehrichtungsumkehr aktiv sind, oder wenn Steuerbit 6 = 0, oder wenn während des Betriebes die Drehzahl gesenkt werden soll

Element	Erläuterung
(3) Schnellhalt	aktiv, wenn sich die Schrittkette 1 im Schritt 6 oder 7 befindet wenn aktiv; ZSW_Bit5 = 0
(4) Hochlaufgeber Sperre	aktiv, wenn sich die Schrittkette 1 im Schritt 4 befindet und Hochlaufgeber gesperrt wird (STW_Bit4 = 0)
(5) Austrudeln	aktiv, wenn pulse_off = 1 (siehe Abb. 49 MotorSpdC_Tele1_20, Zustandsdiagramm)

Drehrichtungsumkehr

Die Drehrichtungsumkehr des Motors wird durch das Bit 11 des Steuerwortes ausgelöst. Bevor die Drehrichtung umgekehrt werden kann, muss der Motor über die Rücklauframpe zum Stillstand gebracht werden.

Es ergibt sich folgendes Zustandsdiagramm:

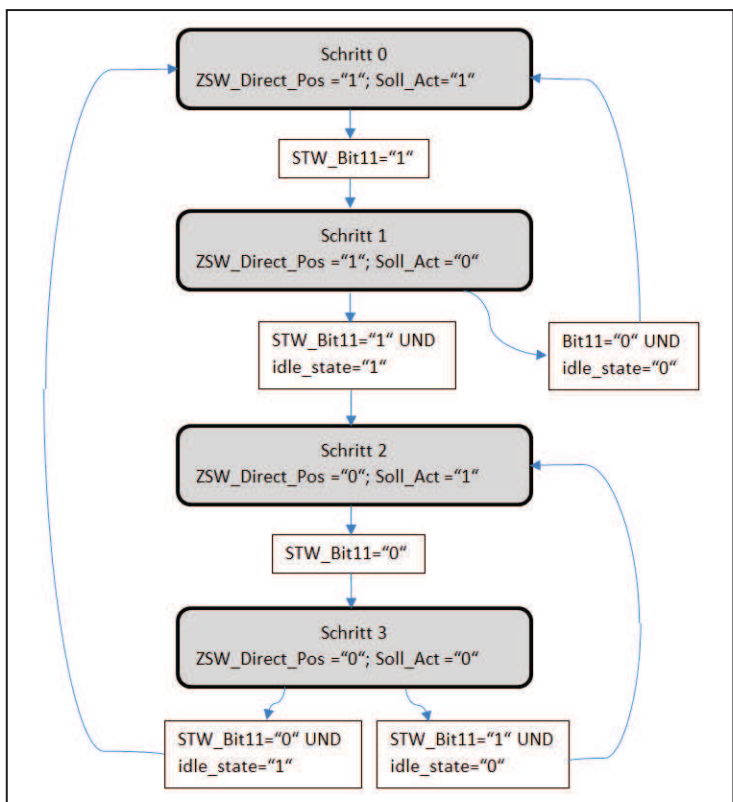


Abb. 56: MotorSpdC_Tele1_20, Drehrichtungsumkehr

Tabelle 41: MotorSpdC_Tele1_20, Drehrichtungsumkehr, Operanden

Operand	Erläuterung
ZSW_Direct_Pos	= 0: Rückmeldung Drehrichtung ist negativ = 1: Rückmeldung Drehrichtung ist positiv
Soll_Act	= 0: Drehzahlsollwert HIW inaktiv; Motor bremst gemäß Rücklauframpe bis zum Stillstand ab = 1: Drehzahlsollwert HIW aktiv; Motor beschleunigt gemäß Hochlauframpe bis Sollwert erreicht
idle_state	= 0: Drehzahl des Motors \neq 0 = 1: Drehzahl des Motors = 0
STW_Bit11	Steuerbefehl: = 0: Solldrehrichtung positiv = 1: Solldrehrichtung negativ

Zur Programmierung wurden die WinMOD Schrittketten – Elemente genutzt. Das Schaltnetz erhielt die Nummer 2.

Alle Schritte und Transitionen lassen sich direkt aus der Abb. 56 übernehmen.

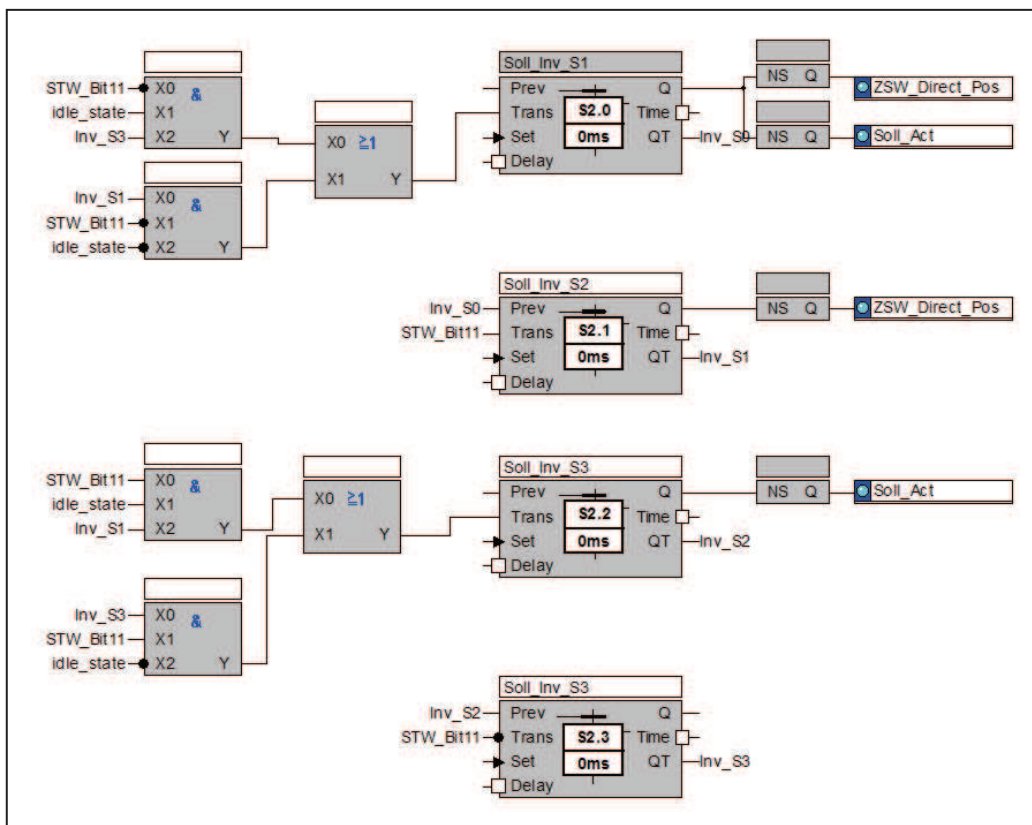


Abb. 57: MotorSpdC_Tele1_20, Drehrichtungsumkehr, Verschaltung

Istwertbildung

Nach Bearbeitung des Sollwertes und der Rampenzeit bildet das Makro „Rampenfunktion“, welches in das Makro MotorSpdC_Tele1_20 integriert wurde, den Istwert.

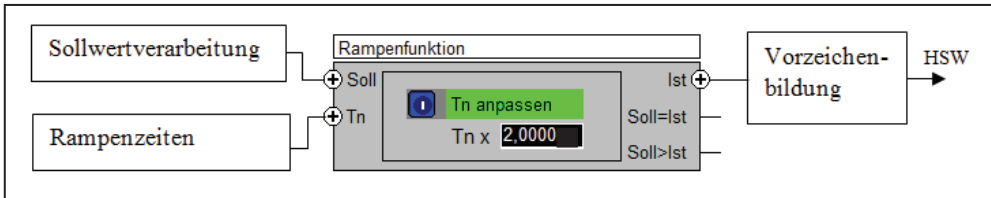


Abb. 58: MotorSpdC_Tele1_20, Rampenfunktion

Aufgrund der für den Hauptistwert und den Hauptsollwert verwendeten Anzeigeformate (Prozent[%]+_200,00) ist eine Anpassung der Nachstellzeit mit dem Anpassungsfaktor 2 erforderlich.

Prozesswerte

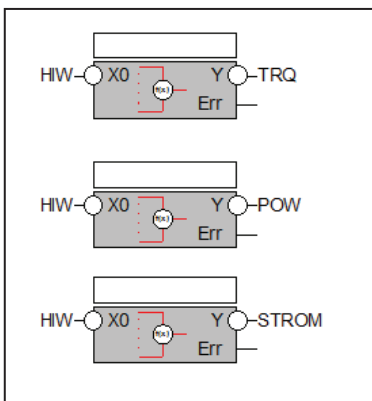


Abb. 59: MotorSpdC_Tele1_20, Prozesswerte

Die Prozesswerte Moment, Leistung und Strom entsprechen immer dem Hauptistwert HIW. Die Abb. 59 zeigt, wie dieser auf entsprechende Operanden geschrieben wird. Da die Simulation kein physikalisch exaktes Motormodell beinhalten, sondern sich auf die Signalübertragung konzentrieren soll, ist diese Berechnung hinreichend.

4.5.2 Regelstrecken

4.5.2.1 Tank

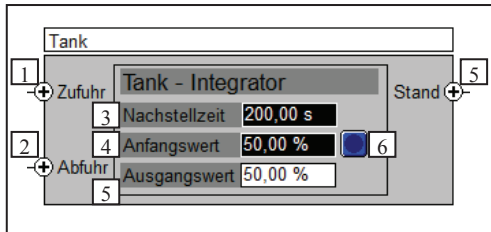


Abb. 60: Tank, Makro

Tabelle 42: Tank, Operanden

Operand	Erläuterung
(1) Zufuhr	Anzeigeformat: Prozent[%]+/-100,00
(2) Abfuhr	Anzeigeformat: Prozent[%]+/-100,00
(3) Tn	Anzeigeformat: 0...99999; bei berechneter Summe aus Zu- und Abfuhr von 100% füllt sich der „Tank“ innerhalb der eingestellten Zeit um 100%
(4) Integral_Xs	Anzeigeformat: Prozent[%]+/-100,00; Um Störmeldungen aufgrund eines leeren/vollen Tanks zu Beginn der Simulation zu vermeiden, kann ein Anfangswert vorgegeben werden.
(5) Stand	Anzeigeformat: Prozent[%]+/-100,00; berechneter Integralwert aus Eingangswerten „Zufuhr“ und „Abfuhr“
(6) Set_Xs	Taster; Setzen des Anfangswertes

Verschaltung

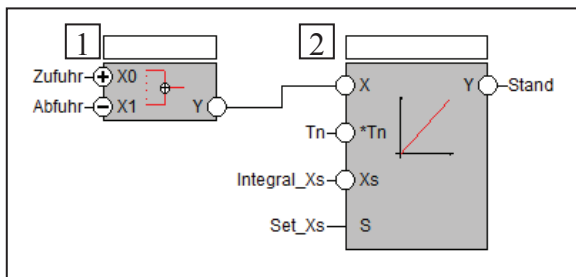


Abb. 61: Tank, Verschaltung

Tabelle 43: Tank, Verschaltung, Erläuterung

Element	Erläuterung
(1) Addition und Subtraktion	bildet Differenz zwischen Wert „Zufuhr“ und „Abfuhr“
(2) Integral	bildet den Integrationswert „Stand“: $\text{Stand}(t) = \frac{\text{Zufuhr}(t) - \text{Abfuhr}(t)}{Tn}$ Xs: Anfangswert S: setzen des Anfangswertes mit Set_Xs = 1

4.5.2.2 Temperatur

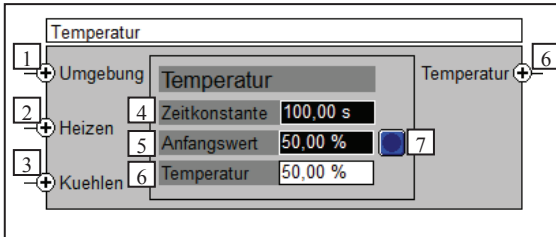


Abb. 62: Temperatur, Makro

Tabelle 44: Temperatur, Erläuterungen

Operand	Erläuterung
(1) Umgebung	Anzeigeformat: Prozent[%] +/-100,00; simuliert Umgebungstemperatur; ohne Energie Zu- oder Abfuhr gleicht sich der Wert „Temperatur“ des Elementes der Umgebungstemperatur an
(2) Heizen	Anzeigeformat: Prozent[%] +/-100,00; Anhebung des Wertes „Temperatur“
(3) Kuehlen	Anzeigeformat: Prozent[%] +/-100,00; Senken des Wertes „Temperatur“
(4) T1	Anzeigeformat: 0...99999; Zeitkonstante eines PT1 - Gliedes
(5) PT_Xs	Anzeigeformat: Prozent[%] +/-100,00; Um Störmeldungen aufgrund einer zu niedrigen/hohen Temperatur zu Beginn der Simulation zu vermeiden, kann ein Anfangswert vorgegeben werden.
(6) Temperatur	Anzeigeformat: Prozent[%] +/-100,00; berechneter Wert aus Eingangswerten „Umgebung“, „Heizen“ und „Kuehlen“
(7) Set_Xs	Taster; Setzen des Anfangswertes

Verschaltung

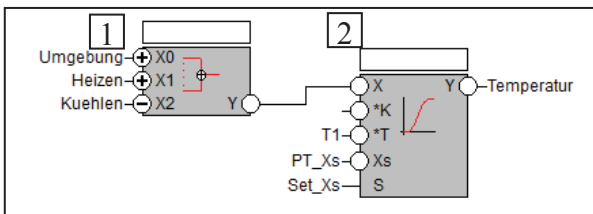


Abb. 63: Temperatur, Verschaltung

Tabelle 45: Temperatur, Verschaltung, Erläuterung

Element	Erläuterung
(1) Addition und Subtraktion	bildet Wert zu: $Y = \text{Umgebung} + \text{Heizen} - \text{Kuehlen}$
(2) PTn – Glied	bildet den Ausgangswert „Temperatur“: $Temp\acute{e}ratur(t) = (\text{Umgebung} + \text{Heizen} - \text{Kuehlen}) - T1 \cdot Temp\acute{e}ratur$ Xs: Anfangswert S: setzen des Anfangswertes mit Set_Xs = 1

4.6 Anlegen der Makrobibliothek als Simulationsdatei

Alle erstellten Makros liegen als Makrodateien (*.wmm) vor und können als Vorlagen genutzt werden. Zusätzlich wurden sie in die Simulationsdatei „Makrotemplates“ eingebunden. Aus Gründen der Nutzerfreundlichkeit empfiehlt es sich bei der Erstellung einer Simulation mit Makros sie aus dieser Vorlagendatei zu beziehen. Das direkte Einlesen der Makrodateien führte häufiger zu Fehlermeldungen, da diese angeblich nicht gefunden werden konnten. Beim Einlesen über die Vorlagendatei trat dieses Problem hingegen nie auf. Des Weiteren bietet die Vorlagendatei den entscheidenden Vorteil, dass sich diese wie eine gewöhnliche Simulationsdatei behandeln lässt. Die Signalelemente, welche die Stati der Makroanschlüsse visualisieren, lassen sich bereits vorab an der Makrovorlage platzieren. Bei anderen Einlesemethoden würde diese Arbeit für jedes Signalelement der Simulation einzeln anfallen. Zudem besteht hierbei die Möglichkeit des Forcens der Signalelemente der Vorlage. Der Eingang „Contactor“ des Makros MotorSpdC_Tele1_20, welcher das Netzschütz repräsentiert, muss zur Funktionsfähigkeit des Antriebes auf logisch 1 stehen. Ist der Eingang nicht verschaltet, muss dieser Eingang in der Simulationsdatei „geforcet“ auf logisch 1 gesetzt werden.

5 Realisierung eines FATs

In diesem Kapitel soll anhand eines realen, bereits abgeschlossenen Projektes die Durchführung von FATs erläutert werden. Bei dem Projekt handelt es sich um eine Erdölförderanlage. Dabei soll der Nutzen eines FATs untersucht und ein Einblick in die Arbeit mit WinMOD und zugleich mit den erstellten Makros geboten werden. Verschiedenen Methoden der Simulationserstellung werden aufgezeigt und abschließend verglichen.

5.1 Aufbau der Anlage

Eine Vielzahl von Pferdekopfpumpen fördert Rohöl zu Tage. Über Förderleitungen gelangt das Rohöl zur Aufbereitung zur eigentlichen Anlage. Separatoren trennen hier zunächst das Wasser vom Öl. Um diesen Vorgang zu beschleunigen, wird Warmwasser benötigt, welches in Rohrleitungen durch die Tanks geführt wird. Die Aufbereitung des Warmwassers findet in einem BHKW statt, dessen Gasversorgung ausschließlich durch das mitgeführte Erdgas gedeckt wird. Das BHKW stellt ebenfalls einen Teil der Energie zum Betrieb der Pumpen und zum Erzeugen der Instrumentenluft bereit. Die Instrumentenluft (Druckluft) wird zum Stellen und Schalten von Ventilen genutzt. Am Ende des Prozesses liegen die geförderten Medien getrennt vor und müssen gespeichert bzw. weitergeleitet werden. Gespeichert werden das Reinöl, das Lagerstättenwasser sowie Verunreinigungen, bis sie entweder von Tanklastwagen abgeholt oder zur Weiterverarbeitung über Fernleitungen zu weiteren Anlagen transportiert werden. Die Refination findet hier nicht statt.

5.2 FAT am Beispiel eines Stellventils

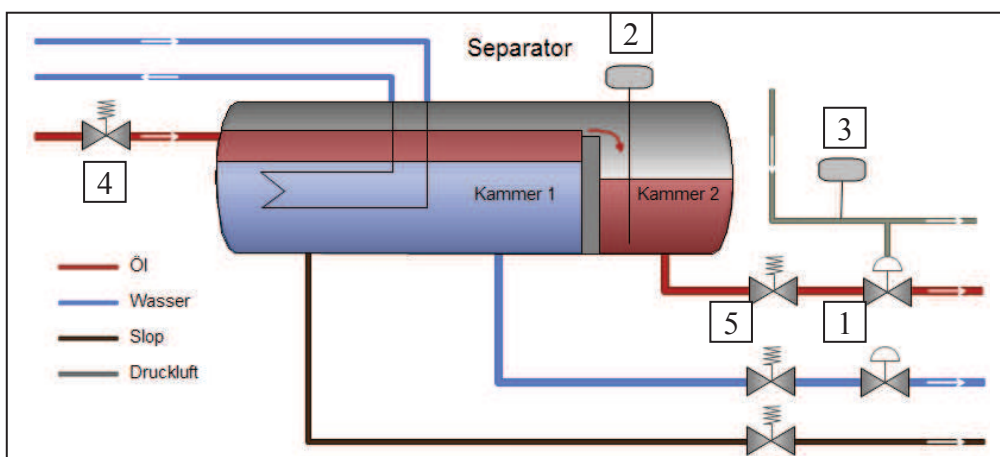


Abb. 64: Schema Anlagenteil Separator

- 1: Stellventil LCV211003
- 2: Füllstandmessung LIC211004
- 3: Drucküberwachung P911001

- 4: Magnetschaltventil Zulauf
- 5: Magnetschaltventil Ablauf

Das Stellventil befindet sich hinter dem Ablauf eines Separators. Rohöl hat einen sehr hohen Wasseranteil (daher auch „Nassöl“) und verschiedene Verunreinigungen (Slop). Das Nassöl fließt bei geöffnetem Ventil in die Kammer 1 des Separators und wird erwärmt. Aufgrund der geringeren Dichte von Öl gegenüber Wasser schwimmt dieses auf und schwappt über die Trennwand in Kammer 2. Das zu testende Ventil regelt den Füllstand in dieser Kammer über seinen Öffnungsweg. Ein Füllstandmesser gibt den Istwert des Füllstandes aus, woraus ein PID – Regler die Sollwertdifferenz sowie die Stärke der Änderung und das Integral der Regelabweichung erfasst. Daraus errechnet dieser die Sollstellung des Ventils. Da zum Fahren des Ventils als Hilfsenergie Druckluft verwendet wird, muss diese ebenfalls überwacht werden. Ein zu geringer Luftdruck erwirkt das Schließen des Ventils, um einen sicheren Zustand herbeizuführen. Da sowohl die Magnetventile des Zu- und des Ablaufs sowie das Stellventil den Durchfluss unterbinden, kommt der Prozess zum Erliegen.

5.2.1 Vorbereitungen und Testvarianten

5.2.1.1 Messstellenliste

Die Messstellenliste enthält Symbole zur Benennung aller Signale, die zwischen Automatisierungssystem und Anlagenfeld ausgetauscht werden sowie Adressierungen und Kommentare zur Verwendung der Signale.

Signale, deren korrekte Verschaltung getestet werden soll, sind der Messstellenliste zu entnehmen, da die in ihr aufgelisteten Adressen sowohl zur Hardware- als auch zur Softwareprojektierung genutzt werden. Dieser Liste entlehnt ist die Signalliste (oder auch Symbolliste), welche lediglich eine für die weitere Projektierung aufbereitete Form der Messstellenliste darstellt. Aus ihr geht die Adressierung aller im Leitsystem festgelegten Hardwareverschaltungen hervor. Hierbei können Adressierungsfehler auftreten.

5.2.1.2 Signaltest

Signale des Stellventils

Um die Richtigkeit der Verschaltung des Stellventils zu testen, genügen bereits die Signale, welche unmittelbar dem Ventil zuzuordnen sind.

Tabelle 46: Stellventil LCV211003, Signale

Symbol	Adresse	Erläuterung
LCV211003	AW 2306	Sollstellung
GI211003	EW 2000	Iststellung

Das Symbol benennt das Signal. Level Control Viscosity (LCV) beschreibt hier die Ansteuerung eines Stellventils zur Füllstandregelung, „GI“ die Anzeige der Stellung.

Der Buchstabe „A“ der Adresse steht für Ausgang, „E“ für Eingang. Das „W“ gibt an, dass das digitale Signal als Daten – Wort (Doppelbyte) versendet wird. Die Zahl gibt die Nummer des Lowbytes des Daten – Wortes an. Somit belegt das AW 2306 die Bytes 2306 und 2307.

5.2.1.3 Ursache – Wirkungs – Test

Signale für Verriegelung, Schutz und Rückmeldungen von Prozesswerten

Während bestimmter Prozesszustände muss ein technologisches Element in einen - für das System, den Menschen und die Umwelt - sicheren Zustand gefahren werden; ein Prozess kann erst unter bestimmten Voraussetzungen starten oder das Eingreifen durch den Bediener wird verhindert. Diese Ausnahmen von der einfachen Ansteuerung eines Aktors und der Ausführung des Befehls werden z.B. in einem Ursache – Wirkungs – Diagramm (UW – Diagramm) beschrieben.

Tabelle 47: Ursache – Wirkungs - Diagramm

SC-01061 3-Phasen-Separator 1 B2110			Wirkung					Legende: E=Einschalten O=Öffnen F=Freigabe R=Regeln A=Ausschalten C=Schließen S=Sperrern V=Verriegelt Z=Sicherheitsfunktion T=Time
Ursache			LAV211002	LAV211004	LCV211003	LCV211005	TCV211001	
R+I	TAG		Typ	Schaltpunkt				Bemerkungen
SC-01061	LIC211001			3-Phasen-Separator 1 Trennschicht Niveau	AI			
SC-01061	LI211001	ASL	DI	3-Phasen-Separator 1 Trennschicht Niveau Min			R	C
SC-01061	LIC211004		AI	3-Phasen-Separator 1 Reinöl Niveau		R		
SC-01061	L211006	ASLL	DI	3-Phasen-Separator 1 Nassöl Niveau Min Min		C		
SC-01061	L211003	ASHH	DI	3-Phasen-Separator 1 Nassöl Niveau Max Max				siehe SC-01060
SC-01061	L211005	ASLL	DI	3-Phasen-Separator 1 Reinöl Niveau Min Min		C		
SC-01061	P211001	ASHH	DI	3-Phasen-Separator 1 Druck Max Max				siehe SC-01060
SC-01061	P211001	ASLL	DI	3-Phasen-Separator 1 Druck Min Min				siehe SC-01060
SC-01061	TIC211001		AI	3-Phasen-Separator 1 Trennschicht Temperatur				R
SC-01066	L221001	ASHH	DI	Reinölbehälter Niveau Max Max		C		
SC-01071	L222002	ASHH	DI	Hochtank 4 Niveau Max Max		C		
SC-01064	L331002	ASHH	DI	Lawa-Behälter 4 (Vorhop) Niveau Max Max		C		je nach Wahlschalter HS331001
SC-01065	L332002	ASHH	DI	Lawa-Behälter 5 (V-KB) Niveau Max Max		C		je nach Wahlschalter HS332001
SC-01079	L334002	ASHH	DI	Hochtank 3 Niveau Max Max		C		je nach Wahlschalter HS331001
SC-01078	P910002	ASLL	DI	Instrumentenluft Druck Min Min		C	C	C

Im obigen UW – Diagramm werden in der Spalte „Wirkung“ alle Aktoren des Anlagenteils „3 – Phasen – Separator 1“ aufgeführt. Die Legende beschreibt die Art der Wirkung, welche hervorgerufen wird infolge von „Ursachen“. Die Ursache für das Schließen („C“) des Ventils „LCV211003“ aus dem Beispiel (der Name des Gerätes und des

Ansteuersignales sind identisch) ist die Feststellung eines zu geringen Drucks der Instrumentenluft durch den Drucksensor „P910002“. Der Sensor „LIC211004“ misst den Füllstand und gibt ihn anschließend aus. Auf Grundlage dieser Messung bestimmt das Automatisierungssystem den Öffnungsweg des Ventils („R“ = Regeln). Die Adressen dieser Messstellen sind ebenfalls der Messstellenliste zu entnehmen.

Tabelle 48: Periphere Signale des Stellventils LCV211003

Symbol	Adresse	Erläuterung
P910002	E 244.1	Instrumentenluft Druck Min Min
LIC211004	EW2020	3 – Phasen – Separator 1 Reinölniveau

5.2.1.4 Regelkreis – Test

Um das Ansprechen eines Reglers zu testen genügt bereits eine Manipulation der Regelgröße. Die Änderung des Stellwertes ist nun zu beachten. Geschieht die Manipulation der Regelgröße manuell, ist das Regelverhalten nicht überprüfbar. Somit muss eine Regelstrecke zwischen Stellgröße und Regelgröße geschaltet werden. Die Regelstrecke kann in einem CFC – Plan oder mit WinMOD realisiert werden

5.2.1.5 Unberücksichtigte Signale

An dieser Stelle werden die Grenzen des Tests bzw. die Tiefe der Simulation deutlich, da Signale und Prozessgrößen, welche den Prozess um das Stellventil herum beeinflussen, vom Test ausgeschlossen bleiben.

Nicht berücksichtigt werden z.B. das Zulaufventil, der Füllstand in Kammer 1 oder der Ablauf aus Kammer 1, um nur die Einflussfaktoren zu nennen, welche sich in mittelbarer Umgebung um das Ventil befinden. Mit dem Zusammenführen all dieser Größen wäre der Weg für eine Anlagensimulation bereitet. Dies soll jedoch nicht Gegenstand dieses FATs sein.

5.2.2 Arten der Simulationserstellung und Simulationsarten

5.2.2.1 FAT mit PCS 7 Simulationsoptionen

Auch ohne die Simulation von Signalzuständen an den Hardwareanschlüssen einer SPS ist ein Testen von Ursache und Wirkung mit PCS 7 möglich. So können Bildbausteine in der WinCC Runtime oder Funktionsbausteine in der CFC Runtime in einen Simulationsmodus versetzt und Signalzustände simuliert werden.

Um zu überprüfen, ob das Auslösen der Druckmeldung der Messstelle P910002 das Stellventil LCV211003 tatsächlich in den sicheren Zustand fahren lässt, wird der Druckmelder im CFC – Editor oder in der WinCC Runtime in einen Simulationsmodus

versetzt und die entsprechende Meldung ausgelöst. Auf gleichem Wege lässt sich auch die Füllstandsregelung simulieren.

Auswertung

Für diese Art der Simulation ist zunächst ein zusätzlicher nicht unerheblicher Aufwand bei der Erstellung der CFC Pläne vonnöten. Da für die meisten Anwendungsfälle jedoch bereits fertige Lösungen bestehen, bei denen auch kein zusätzlicher Projektierungsaufwand anfällt, kann dieser Punkt vernachlässigt werden. Das Erstellen der Simulation wird bereits durch das Projektieren mit PCS 7 abgeschlossen.

Entscheidender Nachteil ist natürlich, dass nur mit internen Signalen gearbeitet werden kann und ein Überprüfen der Richtigkeit der Adressierung nicht möglich ist.

5.2.2.2 FAT mit WinMOD

Voraussetzungen für weitere Schritte

Das PCS 7 – Projekt wurde erstellt. Da der Test mit PLCSIM durchgeführt wird, müssen sich PCS 7 und die WinMOD – Systemsoftware auf demselben Rechner befinden. Um die Systemkopplung zwischen WinMOD und PLCSIM herzustellen, muss die Peripherie – Treibersoftware A750 verwendet werden. Die in der PCS 7 Hardware Konfiguration projektierte Anlagenperipheriestruktur muss in WinMOD eingelesen werden, sodass unter anderem die PROFIBUS- und Signaladressierung bekannt gemacht wird: die „Signalstruktur“ wird angelegt.

5.2.2.2.1 Signalsimulation

Globale Operanden

Globale Operanden geben Signalen, die von der Simulation zur Steuerung gesendet werden, ihre Eigenschaften und interpretieren eingehende Signale. Ein von der Simulation empfangenes Signal und der zugehörige Operand bzw. der Operand eines zu sendenden und das von der Steuerung erwartete Signal müssen kompatibel sein.

Operanden weisen folgende Eigenschaften auf:

Tabelle 49: Eigenschaften Operand LCV211003

Eigenschaft	Eintragung	Erläuterung
Symbol	LCV211003	Symbolische Bezeichnung des Signals
Adresse	AW 2306	Datenausgangs – Wort 2306
Kommentar	3PH1 – Ausschleusungsventil	gibt hier Ort der Verwendung und Art der Messstelle an

Eigenschaft	Eintragung	Erläuterung
Typ	Analog	Da es sich bei der Adresse um ein Datenwort handelt, erkennt WinMOD, dass es sich bei dem Signaltypen um ein digitales oder analoges Signal mit einer Länge von 16 Bit handeln muss. Eindeutig kann WinMOD nur feststellen, dass es sich nicht um ein binär Signal handeln kann. Lautete die Adresse z.B. E244.1 wäre dies eindeutig ein Binärsignal. Im Beispiel muss die Einstellung „analog“ vorgenommen werden.
Richtung	AS – Ausgang	Aus der Adresse kann WinMOD bereits ableiten, dass es sich hier um einen AS – Ausgang handelt.
Default – Wert	0	Vorgabe eines Standardwertes
Internes Format	Prozent[%]+_ -100,00	Der Stellweg wird in der Simulation in Prozent ausgegeben. Die Verwendung des Anzeigeformates Prozent[%]+_ -120,00 ist möglich. Intern werden logische Werte über 100 % jedoch nicht weitergeleitet.
Externes Format	Siemens 0...27648	Das S7 – Format für Standardanalogsignale wird hier gewählt.

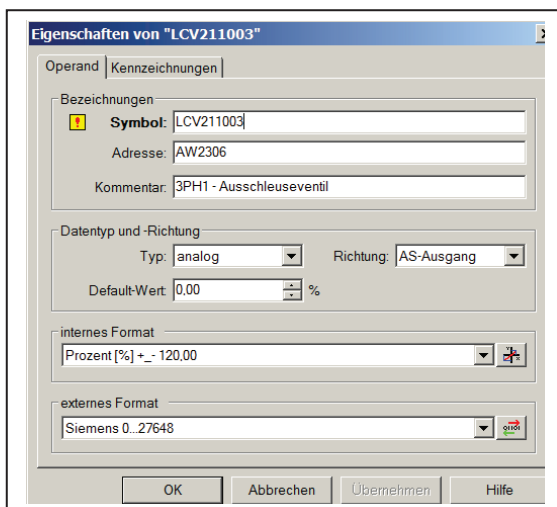


Abb. 65: Eigenschaftsdialog Operand LCV211003

Automatische Operandenerstellung mit Operandenimport

Die Operandenerzeugung findet für gewöhnlich durch einen Import einer „Operandenliste“ statt.

Für die Beispielsimulation ergibt sich folgende Tabelle:

Tabelle 50.: Operandenliste, Test Stellventil

Symbol	Adresse	Kommentar	Typ	Default	Def. Int. Format	Def. Ext. Format
LCV211003	AW2306	3PH1 - Ausschleuseventil	AM		Prozent[%]+_- 120,00	Siemens 0...27648
GI211003	EW2000	3PH1 - Ausschleuseventil	AM		Prozent[%]+_- 120,00	Siemens 0...27648
P910002	E244.1	Instrumentenluft Druck Min Min				
LIC211004	EW2032	3 - Phasen - Separator 1 Reinölniveau	AM	50	Prozent[%]+_- 120,00	Siemens 0...27648

Für die Analogwerte muss der Typ mit angegeben werden. Für das Binärsignal ist dies nicht zwingend erforderlich.

Tabelle 51: WinMOD – Kürzel für Signaltypen

Typ	Typkürzel
binär	BM
analog	AM
digital	DM

Ein nicht eingetragener Defaultwert wird bei jeder Art von Signal als „0“ erkannt. Auch die Angabe der Formate ist nicht zwingend. Bei fehlender Angabe wird das zuletzt verwendete Format eingetragen. Für Binärsignale existieren keine Anzeigeformate. Anzeigeformate liegen als Datei oder als „Definition“ vor. Durch das Einlesen einer Anzeigeformatdatei in ein Projekt wird dieses zu einer Anzeigeformatdefinition.

Bevor die Tabelle eingelesen werden kann, muss sie in einem geeigneten Format abgespeichert werden: txt, csv oder tab sind erlaubt. Bei Erstellung der Tabelle mit Excel sollte bei der anschließenden Speicherung das Format „Text (Tabstopp-getrennt) (*.txt)“ verwendet werden. Nur in diesem Fall können Umlaute verwendet werden. „Unicode Text“ wird hingegen von WinMOD nicht erkannt.

Mithilfe der „Operandenzuordnung“ werden nun die Symbole der Operandentabelle (siehe Tabelle 50) mit den Adressen der Signalstruktur verknüpft.

Ergebnis:

Nun können bereits Signale durch WinMOD gelesen und empfangen werden. Damit wurde eine Signalsimulation erzeugt.

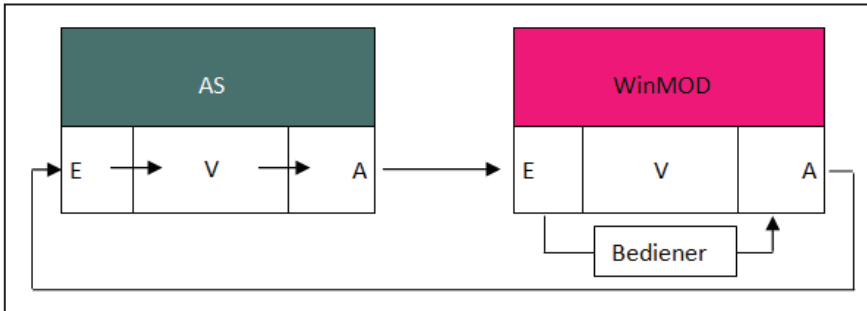


Abb. 66: Signalfluss- und Verarbeitung in Signalsimulation

Ob die Adressierung mit der Solladressierung der Messstellenliste übereinstimmt, kann nun bereits getestet werden; ebenso die Verriegelung.

Da sich die Signale GI211003 und LIC211004 nicht aus einem Prozess heraus ändern, muss der Bediener hier die „Verarbeitung“ der Signale realisieren. Das Signalelement des Rückgabewertes GI211003 muss also „geforcet“ werden. Um zu überprüfen, ob der PID – Regler LIC211004 anspricht, muss ebenso der simulierte Wert der Füllstandsmessung „geforcet“ werden. Alle Größen des Regelkreises lassen sich über den WinCC Bildbaustein des PID – Reglers in der WinCC Runtime einsehen.

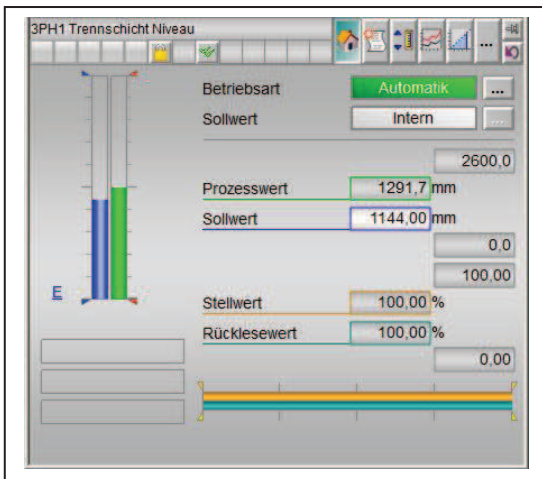


Abb. 67: Bildbaustein, PID – Regler LIC211004

Die folgende Darstellung verdeutlicht die verwendeten Begrifflichkeiten und ihre Position im Regelkreis.

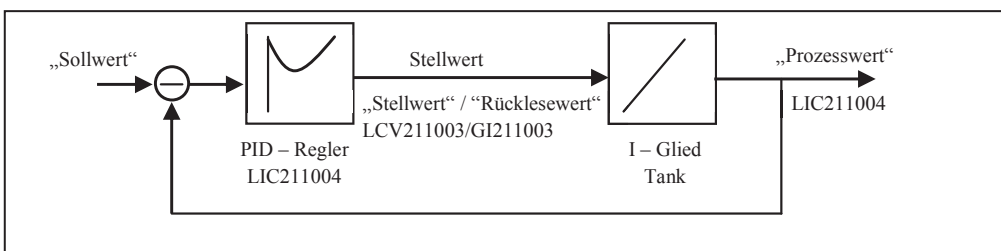


Abb. 68: Füllstandsregelkreis um Stellventil LCV211003

Das Setzen des Operanden LIC211004, welcher dem Prozesswert entspricht, auf einen Wert über dem Sollwert, wird den Regler dazu veranlassen, den Stellwert LCV211003 zu erhöhen, da der Ölstand sinken soll. Allerdings sinkt der „Ölstand“ nicht. Somit wird die Stellgröße durch den Regler auf ihren Maximalwert gestellt – das Ventil also geöffnet. Tatsächlich zeigt sich, dass dieser Prozesswert LIC211004 im Bildbaustein angezeigt wird; ebenso die Stellgröße im Operanden LCV211003. Nach der Ansteuerung des Ventils wird ein Rücklesewert erwartet, der dem Stellwert entspricht. Tut er dies nicht, so wird eine Meldung ausgegeben. Der Rücklesewert GI211003 wird manuell auf den Wert des Stellwertes gesetzt. Dieser Wert erscheint nun ebenfalls im Bildbaustein.

Folgende Abbildung zeigt die in der WinMOD – Simulation vorgenommenen Einstellungen. Sie entsprechen den Werten im Bildbaustein LIC211004, siehe Abb. 67.

100,00 %	
GI211003	EW2002
100,00 %	
LCV211003	AW2308
50,00 %	
LIC211004	EW2020
P910002	

Abb. 69: WinMOD Simulationselemente, Signalsimulation

Anschließend wird das Signal P910002 auf logisch 1 gesetzt: der Instrumentendruck ist zu gering. Infolgedessen schließt das Ventil wie erwartet.

Dieser Test war erfolgreich. Die Möglichkeit des Erzwingens von fehlerhaften Zuständen besteht ebenfalls. Wiche der Rücklesewert des Ventils von seinem Stellwert ab, so wäre hiermit eine Fehlfunktion simuliert.

Auswertung

- Der zeitliche Aufwand ist erheblich. Ein Stellventil besitzt lediglich 2 Anschlüsse, andere Aktoren 10 und mehr. Zunächst müssten diese gesucht und gruppiert werden. Auch mit den Filterfunktionen gestaltet sich diese Methode aufgrund der Vielzahl der Aktoren als recht mühselig. Zudem ist der Anwender ständig damit beschäftigt dem Automatisierungssystem die geforderten Rückmeldungen zu liefern, bevor durch deren Ausbleiben eine Störung erkannt wird und das technologische Element nicht mehr bedienbar ist. Bei Schaltventilen bleiben hier oft nur 2 bis 3 Sekunden.
- Der Aufbau eines funktionierenden Regelkreises, bei dem die Änderung des Stellwertes, die Regelgröße beeinflusst, kann nicht realisiert werden.

- Für das Beispielventil besteht lediglich ein Verriegelungssignal. Auch das Auffinden dieser Signale ist recht aufwendig und zudem aufgrund der meist eher kryptischen Bezeichnungen mit einem gewissen Fehlerpotential verbunden.

5.2.2.2 Manuelles Erstellen einer Antriebs-/Messsystem- und Prozesssimulation mit Makros

Mit dem Einsatz von Makros vertieft sich das Simulationsniveau hin zu einer Antriebs-/Messsystem- und Prozesssimulation, womit die beiden erstgenannten Nachteile der Signalsimulation minimiert werden.

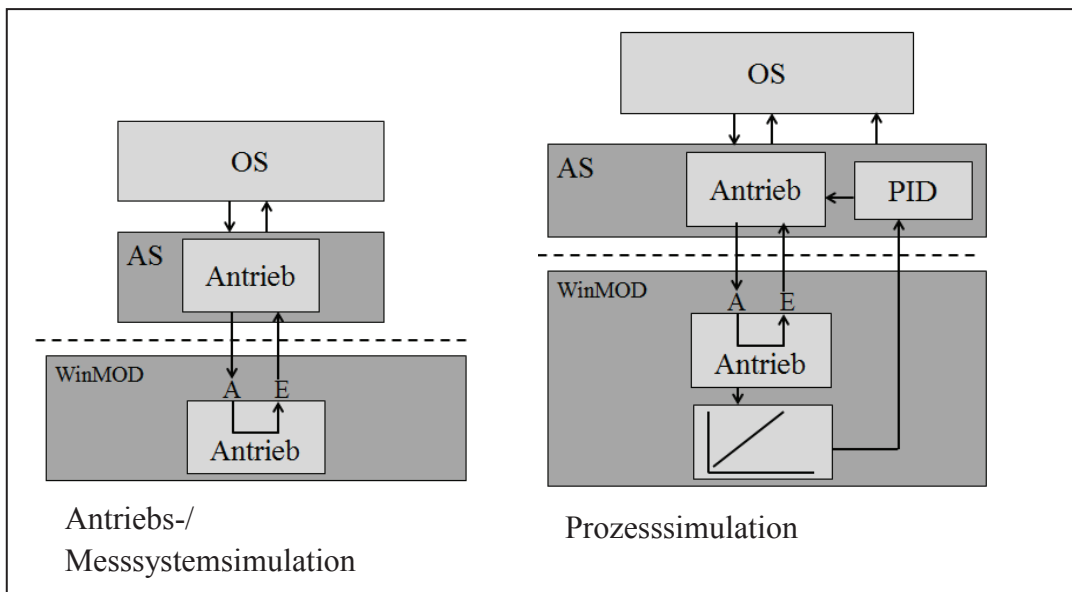


Abb. 70: Antriebs-/Messsystemsimulation und Prozesssimulation

Aus der Vorlagendatei „Makrotemplates“ oder einer projektspezifischen Kopie der Vorlagendatei werden die Makrotypen „VlvAn“ (Stellventil) und „Tank“ benötigt.

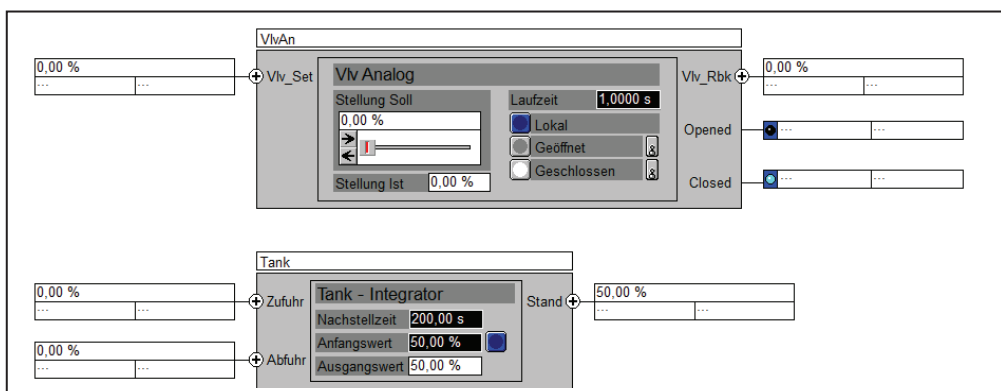


Abb. 71: Makro VlvAn, Tank

Da es sich bei dem Ventil um ein Ablaufventil handelt, muss der Tankzulauf manipuliert werden. Dazu muss das Simulationsfenster „Makrotemplates“ zunächst gestartet werden, um Änderungen in den Bedienelementen und an den Signalelementen vornehmen zu können.

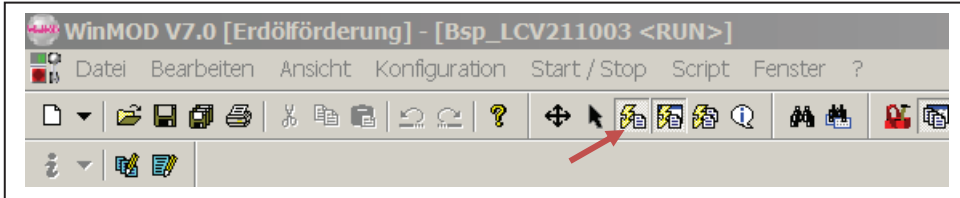


Abb. 72: Simulationsfenster starten

Ein Vortäuschen des Zulaufes geschieht durch Forcen des Signalelementes „Zufuhr“ am Makroeingang und der Vergabe eines Wertes zwischen 0 und 100%. Möchte man zu Beginn der Simulation eine Fehlermeldung aufgrund eines zu geringen Füllstandes vermeiden, lässt sich über das Bedienelement des Makros ein Anfangswert setzen. Standardmäßig liegt dieser bei 50%.

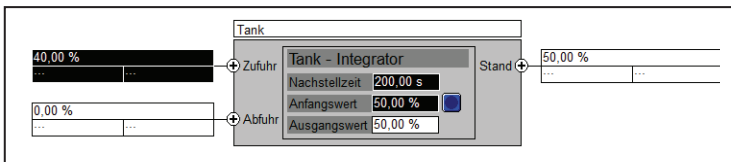


Abb. 73: Makro Tank, Simulationsvorbereitung

Per Copy and Paste werden die Vorlagen in ein anderes Simulationsfenster eingefügt und erhalten einen symbolischen Namen; dies ist der sogenannte Makroname. Abschließend werden den benötigten Signalelementen per Drag and Drop die Operanden zugewiesen und das Verriegelungssignal eingefügt.

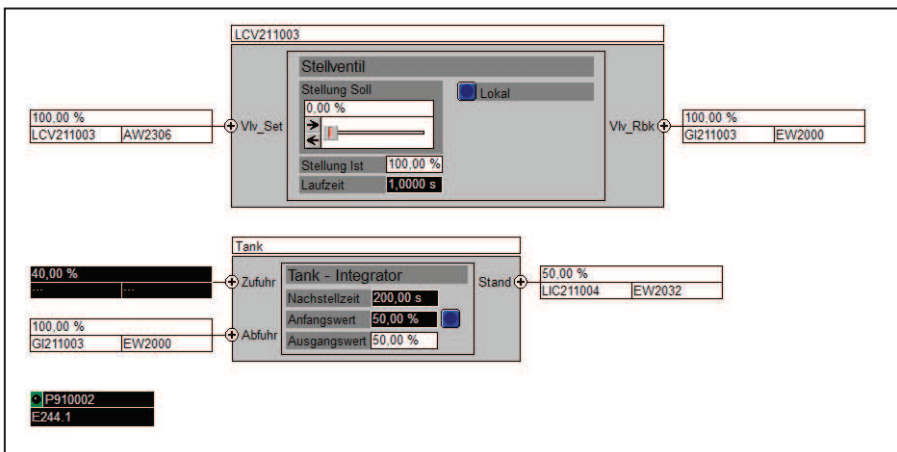


Abb. 74: Simulationselemente, Test Regelkreis

Mit dem Start der Simulation beginnt der Test. Durch Ablesen von Stell- und Rückführgrößen kann eingesehen werden, ob eine korrekte Adressierung vorliegt.

Ebenfalls ist die Funktionsfähigkeit des Reglers zu beobachten. Das Anheben und Senken der Zufuhr beeinflusst den Stellwert. Bei gleichbleibender Zufuhr nimmt der Prozesswert einen stationären Endwert ein.

Auswertung

Es ist festzustellen, dass diese Art der Simulation die Dauer des Testes wesentlich verkürzt, das Erstellen jedoch sehr viel Zeit in Anspruch nimmt und bei einer höheren Anzahl von zu testenden Signalen sehr gewissenhaft durchgeführt werden sollte. Mit der Verwendung der Regelstrecke „Tank“ kann das Regelverhalten getestet werden.

5.2.2.2.3 Simulationserstellung mit WinMOD Engineering Assistenz

Automatisches Erstellen einer Antriebs- und Messsystemsimulation

Das kostenpflichtigen Add – On „Engineering Assistenz“ birgt ein deutlichen Einsparpotential bei der Erstellung von Antriebs- und Messsystemsimulationen, da sowohl die Erzeugung der Makros als auch das Zuweisen der Operanden zu den Makrosignalen automatisiert wird.

Die Grundlage hierfür bildet eine Messstellenliste, welche zunächst in eine sogenannte Engineering – Liste umgewandelt wird. Diese enthält die Operandenliste und weitere Spalten, die der Makrozuordnung dienen: Makrotyp, Makroname, Makrosignal, Zieldatei. Signale, die keinem Makro zugeordnet werden sollen, können aus dieser Liste gelöscht werden.

- Der Makrotyp ist die Bezeichnung der Makrovorlage. Die Engineering Assistenz wird die Makros aus der Vorlagendatei beziehen, so wie es zuvor manuell geschah. Somit können auch hier Änderungen an den Vorlagen vorgenommen werden, welche dann Anwendung auf alle Makros dieses Typs finden.
- Der Makroname ist die Identifikation des Makros und kann somit dem Gerätenamen entsprechen. Für alle Operanden, welche einem einzigen Makro zugeordnet werden sollen, muss derselbe Makroname vergeben werden.
- Das Makrosignal bezeichnet einen Anschluss des Makros.
- Die Zieldatei benennt die Simulationsdatei, in welcher die Makros erzeugt werden.

Auf weitere Zuordnungs- und Einstellmöglichkeiten soll an dieser Stelle nicht weiter eingegangen werden.

Anwendung der Engineering Assistenz an einem Beispiel

Die bereits erstellte Operandenliste (siehe Tabelle 50) wird um die benötigten Spalten der Makrozuordnung erweitert. Da das Signal GI211003 sowohl das Ausgangssignal des Stellventils LCV211003 als auch eine Eingangsgröße der Regelstrecke darstellt, muss dieses zweimal eingetragen werden. Das Verriegelungssignal P910001 wurde aus der Engineering – Liste gelöscht.

Tabelle 52: Engineering – Liste, Test Stellventil (Teil 1)

Symbol	Adresse	Kommentar	Typ	Default	Def. Int. Format	Def. Ext. Format
LCV211003	AW2306	3PH1 - Ausschleuseventil	AM		Prozent[%]+_- 120,00	Siemens 0...27648
GI211003	EW2000	3PH1 - Ausschleuseventil	AM		Prozent[%]+_- 120,00	Siemens 0...27648
LIC211004	EW2020	3 - Phasen - Separator 1 Reinölniveau	AM	50	Prozent[%]+_- 120,00	Siemens 0...27648
GI211003	EW2000	3PH1 - Ausschleuseventil	AM		Prozent[%]+_- 120,00	Siemens 0...27648

Tabelle 53: Engineering – Liste, Test Stellventil (Teil 2)

Symbol	Makrotyp	Makroname	Makrosignal	Zieldatei
LCV211003	VlvAn	LCV211003	Vlv_Set	Erdölförderanlage
GI211003	VlvAn	LCV211003	Vlv_Rbk	Erdölförderanlage
LIC211004	Tank	LIC211004	Stand	Erdölförderanlage
GI211003	Tank	LIC211004	Zufuhr	Erdölförderanlage

Nach Erstellen der Engineering – Liste und dem Abspeichern im txt – Format „Tabstopp – getrennt“ erfolgt dessen Einlesen durch die Engineering Assistenz. Hierbei werden Operanden erzeugt und ihnen ihre Eigenschaften (Tabelle 52) zugewiesen. In der Simulationsdatei (z.B. „Makrotemplates“) wird über den Makrotypen (Tabelle 53) nach den Makrovorlagen gesucht und für jeden einem Makrotypen zugeordneten Makronamen wird ein Makro erzeugt. Über die Verknüpfung „Symbol“ – „Makrosignal“ finden die Operanden und Signalelemente der Makros zueinander. Der Operandenimport verbindet die Operanden mit ihren zugehörigen Signalen des Peripherietreibers.

Das Einfügen des Verriegelungssignals schließt die Erstellung der Simulation ab. Das Ergebnis gleicht der manuellen Erstellung einer Simulation mit Makros (Abb. 74).

Auswertung

Obwohl sehr viel Zeit bei der direkten Erstellung der Simulation, d.h. der Benennung und Zusammenstellung von Makros und der Verbindung der Signalelemente mit Operanden, eingespart werden kann, erscheint es zunächst noch sehr aufwendig, dies stattdessen in

Form einer Tabelle vorwegzunehmen. Ebenfalls müssen Signale gesucht und zugewiesen werden.

5.2.2.2.4 Erstellung Engineering – Liste mit Excel – Skript

Zwar wird seitens WinMOD eine Lösung zur Erstellung einer Engineering – Liste angeboten, jedoch benötigt die „Reverse Engineering Assistenz“ hierfür eine bereits fertige Simulation. Ansonsten wird darauf vertraut, dass der Projektteur diese Aufgabe allein bewältigt. Dies ist verständlich, da verlässliche Lösungen automatisierter Arbeitsabläufe eine normierte Umgebung voraussetzen, die mit der Vielzahl von Möglichkeiten der Signaldeklaration nicht gegeben ist.

Lösungsansatz

Durch die Auswertung der Messstellenlisten sollen Signale, welche ein und demselben Gerät und damit auch Makro zuzuordnen sind, zusammengefasst und das entsprechende Makrosignal bestimmt werden. Hierzu wurde ein Excel VBA – Skript erstellt und als *.xlsm – Datei („Excel – Vorlage mit Makros“; „EA_Liste_erstellen, Makrovorlagen“) abgespeichert. Diese Vorlage kann anschließend in Excel eingebunden werden, um das Script komfortabel über eine Schaltfläche auszuführen. Eine Bibliothek hilft dem Anwender des Makros bei den notwendigen Konfigurationen, die vorgenommen werden müssen, um die Messstellenliste in eine Engineering – Liste zu wandeln. Die Bibliothek ist in einer weiteren Excel – Datei abgelegt. Sie listet alle WinMOD – Makros und ihre zugehörigen Makrosignale sowie die standardmäßigen Anzeigeformate auf.

Einbinden der Excel – Makrovorlage „EA_Liste_erstellen“

- Öffnen der Excelvorlagen – Datei „EA_Liste_erstellen, Makrovorlagen“
- Anwahl: *Datei* → *Optionen* → *Menüband anpassen*
- Auswahl des Befehls *Makros*
- *Hinzufügen* der Makros *EA_Vorbereitung* und *EA_Liste_erstellen* in eine beliebige Registrierkarte, z.B. *EA_Liste*

Ergebnis: Zwei Schaltflächen wurden erstellt, über welche die Excel – Vorlage jederzeit aufgerufen werden kann.

Voraussetzungen der Messstellenliste

Der Gerätenamen bzw. das WinMOD – Makro erhält dieselbe Bezeichnung wie eines seiner Signale. Ist dies nicht gewünscht, müssen diese Makros im Anschluss umbenannt werden.

Die Zuordnung zu einem Gerät bzw. WinMOD – Makro erhalten die Signale durch ihre symbolischen Bezeichnungen. Die Zuordnung muss über eine Zahlenfolge gegeben sein, welche jedes Symbol trägt. Die Zahlenfolge muss mindestens 3 Ziffern umfassen.

Beispiel Stellventil: Die Signalsymbole LCV211003 und GI211003 verfügen über die gleiche Ziffernfolge und werden somit einem Makro zugeordnet. Der Makroname lautet dann „LCV211003“ oder „GI211003“.

Verfügen mehrere Geräte unterschiedlichen Typs über die gleiche Gerätenummer, so können sie trotzdem differenziert werden.

Anwendung

Folgendes Beispiel soll die Anwendung der Skripte erläutern:

Die Simulation besteht aus den Stellventilen „LCV211003“ (Füllstandreglung) und „PCV211005“ (Druckregelung) sowie einem Auf – Zu – Ventil „POV211003“.

Tabelle 54: Messstellenliste, EA - Skript

Symbol	Adresse	Kommentar
LCV211003	AW2306	3PH1 - Ausschleuseventil
GI211003	EW2000	3PH1 - Ausschleuseventil
PCV211005	AW2308	3PH1 - Druckregelventil
GI211005	EW2002	3PH1 - Druckregelventil
POV211003	A1006	Ausschleuseventil
G211003.1 OH	E1512	Ausschleuseventil Auf
G211003.2 OL	E1513	Ausschleuseventil Zu

- Öffnen einer neuen Excel – Datei, der Messstellenliste und der Bibliothek „EA_Bibliothek“
- Ausführen des Makros *EA_Vorbereitung* in der neuen Excel – Datei

Zwischenergebnis: Die Tabellen der Arbeitsmappe wurden umbenannt: Tabelle 1: *Engineering Liste*; Tabelle 2: *Konfiguration*. Die Tabelle *Konfiguration* wurde mit Spaltenköpfen versehen.

- Kopieren der Messstellenliste in die Tabelle *Engineering Liste*; nicht benötigte Signale (keinem Makro zuweisbar) können gelöscht werden
- Kopieren benötigter Makrotypen aus der *EA_Bibliothek* in die Tabelle *Konfiguration*

Tabelle 55: EA Konfiguration (1)

Makrotyp	Makrosignal	Zieldatei	Ext. Format	Int. Format	Default	Typ Operand	Zeichenkette	Spalte
VlvAn	Vlv_Set		Siemens 0...27648	Prozent [%] + - 120,00		AM		
	Vlv_Rbk		Siemens 0...27648	Prozent [%] + - 120,00		AM		
	Opened							
	Closed							
Vlv	Open							
	Opened							
	Closed							
	Local							
	Open_AS							
	Close_AS							

In die Spalte *Zeichenkette* soll das Makrosignal dem Symbol zugeordnet werden. Die Spalte *Spalte* zeigt an, in welcher Spalte der Tabelle *Engineering Liste* nach der Zeichenkette gesucht wird.

UND – Verknüpfung: Durch das Duplizieren der Zeilenköpfe *Zeichenkette* und *Spalte* können pro Makrosignal weitere Kriterien zur Verknüpfung von Makrosignal und dem Symbol eingefügt werden. Jede der Zeichenketten muss sich in *einer* Zeile der Tabelle *Engineering Liste* wiederfinden.

ODER – Verknüpfung: Durch das Duplizieren einer Zeile können pro Makrosignal weitere Kriterien eingefügt werden.

Das rot – markierte Makrosignal ist das „Hauptargument“. Je Makrotyp ist es das zuoberst aufgeführte Makrosignal. Jedoch lässt sich auch diese Zeile duplizieren. Das Hauptargument stellt für das Skript den Ausgangspunkt zur Erkennung eines Gerätes dar. Kann dieses Makrosignal keinem Gerät zugeordnet werden, so können auch weitere Signale des Gerätes nicht gefunden werden. Aus dem Symbol, welches über das Hauptargument gefunden wird, wird der Makroname gebildet.

Folgende Konfiguration kann für das Beispiel verwendet werden:

Tabelle 56: EA Konfiguration (2)

Makrotyp	Makrosignal	Zieldatei	Zeichenkette	Spalte	Zeichenkette	Spalte
VlvAn	Vlv_Set	BspDatei	LCV	Symbol		
	Vlv_Set	BspDatei	PCV	Symbol		
	Vlv_Rbk	BspDatei				
	Opened	BspDatei				
	Closed	BspDatei				
Vlv	Open	BspDatei	POV	Symbol		
	Opened	BspDatei	G	Symbol	Auf	Kommentar
	Closed	BspDatei	G	Symbol	Zu	Kommentar

Makrotyp	Makrosignal	Zieldatei	Zeichenkette	Spalte	Zeichenkette	Spalte
	Local	BspDatei				
	Open_AS	BspDatei				
	Close_AS	BspDatei				

Das Betätigen der Schaltfläche *EA_Liste erstellen* aktiviert das Skript, die Engineering Liste wird erzeugt und zusätzlich im Text – Editor (Datei – Format: „Text (Tabstopp – getrennt) (*txt)“ geöffnet..

Tabelle 57: Engineering – Liste, automatisch

Symbol	Makroname	Makrotyp	Makrosignal	Zieldatei
LCV211003	LCV211003	VlvAn	Vlv_Set	BspDatei
GI211003	LCV211003	VlvAn	Vlv_Rbk	BspDatei
PCV211005	PCV211005	VlvAn	Vlv_Set	BspDatei
GI211005	PCV211005	VlvAn	Vlv_Rbk	BspDatei
POV211003	POV211003	Vlv	Open	BspDatei
G211003.1.OH	POV211003	Vlv	Opened	BspDatei
G211003.2.OL	POV211003	Vlv	Closed	BspDatei

Nachdem Abspeichern der txt – Datei kann die Engineering – Liste zur Erstellung der Simulation mit der Engineering Assistenz genutzt werden.

Auswertung

Das Skript kann nur unter den genannten Voraussetzungen verwendet werden. Ein flexibler Einsatz ist somit nicht gegeben; da Messstellenlisten der zu betreuenden Projekte jedoch häufig ähnliche Strukturen aufweisen, kann das Skript trotzdem eine häufige Anwendung finden. Ein eventuelles Nachkorrigieren (z.B. Einstellung des *Konfigurators* unpassend) oder Vervollständigen der automatisch erstellten Engineering – Liste kann nicht ausgeschlossen werden. Um auf diesem Wege zugleich eine Prozesssimulation vorzubereiten muss die Messstellenliste entsprechend von Hand erweitert werden.

Die Erstellung der Engineering – Liste und damit auch der Simulation gestaltet sich nun jedoch sehr effizient.

5.3 Gesamtauswertung

Zusammenfassend stellen die Tabellen Tabelle 58 und Tabelle 60 Vergleich zwischen den verschiedenen Arten der Simulationserstellung sowie der Simulationen an. Da von einer fehlerfreien Durchführung aller Arbeitsschritte seitens des Projektors/Anwenders ausgegangen wird, bleiben nur der zeitliche Aufwand und die Machbarkeit der Durchführung als Kriterium.

Tabelle 58: Simulationserstellung – Simulationsart

		Simulationserstellung			
		PCS 7 Projektierung	WinMOD, ohne Engineering Assistenz	WinMOD, mit Engineering Assistenz	WinMOD, Engineering Assistenz u. Excel Skript
Simulationsart	PCS 7 Simulationsoptionen	+1 *1			
	Signalimulation		*2		
	Antriebs-/ Messsystemsimulation		*3	*4	*5
	Prozesssimulation		*6	*7	*8

Aufwand  gering  mittel  hoch  unmöglich, nicht notwendig

Tabelle 59: Simulationserstellung – Simulationsart, Erläuterung

	Erläuterung
*1	wenn mit PCS 7 Projektierung abgeschlossen
*2	mit Operandenimport abgeschlossen
*3	Makros manuell mit Operanden verschalten
*4	Verschaltung von Makros und Operanden in Excel
*5	wenn für Projekt anwendbar, Anzahl der Signale unerheblich
*6	Makros manuell mit Operanden und untereinander verschalten
*7	Verschaltung von Makros (Antrieb + Prozess) und Operanden in Excel
*8	wie 5 + manuelles Einfügen von Signalen der Regelstrecken in Excel

Tabelle 60: Simulationsart - Testvariante

		Simulationsart			
		PCS 7 Simulationsoptionen	Signal-simulation	Antriebs-/ Messsystem-simulation	Prozess-simulation
Testvariante	Ursache – Wirkungs - Test	*1	*2	*3	*4
	Signaltest	*5	*6	*7	*8
	Regelkreistest	*9	*10	*11	*12

Aufwand gering mittel hoch unmöglich, nicht notwendig

Tabelle 61: Simulationsart – Testvariante, Erläuterung

	Erläuterung
*1-*3	verursachende Signale ungeordnet
*4	Regelkreise geschlossen, Verriegelungssignale ungeordnet
*5	nicht möglich
*6	abhängig von Signalmenge des Projektes und Komplexität der Geräte
*7	Rückmeldungen erfolgen direkt nach Ansteuerung des Antriebes
*8	Rückmeldungen erfolgen direkt nach Ansteuerung des Antriebes
*9	Regelkreis geschlossen, unmittelbare Rückmeldung
*10	manuelle Eingabe aller Größen des Regelkreises (außer Stellgröße); nur Test des Ansprechens des Reglers
*11	manuelle Eingabe der Prozessgröße; nur Test des Ansprechens des Reglers
*12	unmittelbare Rückmeldung der Anlage

Somit lässt sich feststellen, dass beispielsweise ein Signaltest mit einer Antriebsimulation (unter Verwendung der erstellten WinMOD Makrovorlagen) durchgeführt werden sollte, da hier der zeitliche Aufwand am geringsten ist. Zur Erstellung dieser Simulationsart sollte die Engineering Assistenz zusammen mit dem entwickelten Excel Skript verwendet werden.

6 Zusammenfassung

Mit den erstellten Makros konnte die Grundlage für eine effiziente und weniger fehleranfällige Durchführung weiterer FATs geschaffen werden. Während nach Erwerb der Simulationssoftware WinMOD lediglich das rasche Erstellen einer Signalsimulation möglich ist, können nun Antriebs- und Messsystemsimulationen sowie Prozesssimulationen realisiert werden. Das Testen von Verriegelungen bleibt jedoch aufwendig.

Die Makrovorlagen können in vielen weiteren Projekten Einsatz finden. Eine Neugestaltung für ähnlich aufgebaute Antriebe ist nicht nötig, da eine Erweiterbarkeit der Makros um zusätzliche Signale gegeben ist. Unter Verwendung der Invertierungsschalter für eine eventuelle Umparametrierung eines Makros bleibt dem Anwender ein (fehleranfälliger) Eingriff in die Verschaltungslogik des Makros erspart. Das Duplizieren einer Makrovorlage erlaubt damit ein einfaches Erstellen mehrerer Varianten einer Vorlage.

Die Erweiterung der Makrovorlagen – Bibliothek ist ohne größere Vorüberlegungen umsetzbar. So können das Bedienkonzept und die notwendigen Verschaltungen zur Betriebsartenumschaltung oder des Stör- Resetverhaltens leicht übernommen werden. Die ergonomisch gestaltete Bedienoberfläche ist voraussichtlich auf alle zukünftig zu erstellenden Makros anwendbar.

Mit dem Einsatz des Excel Skriptes „EA_Liste_erstellen“ konnte zusätzlich die Erstellung von Antriebs- und Messsystemsimulationen erleichtert werden. Mit der automatisierten Erzeugung der Engineering – Liste wurde die Dauer der Simulationserstellung von der Anlagengröße entkoppelt. Voraussetzung zur Anwendung bleibt allerdings eine bestimmte Struktur des Aufbaus der Messstellenliste.

Ausblick

Das Erstellen einer Simulation, welche für den Test von Verriegelungsschaltungen optimiert ist, bedarf Kenntnisse über das Ursache – Wirkungs – Verhalten der Anlage. Ein solches Verhalten wird beispielsweise in Ursache – Wirkungs – Diagrammen beschrieben. Das Auslesen eines solchen Diagrammes würde den Weg bereiten, um Verriegelungssignale in der Simulation strukturiert zu gruppieren und sie im Idealfall dem jeweiligen Antrieb automatisch zuzuordnen. Das aufwendige und fehlerbehaftete Zusammentragen der einzelnen Signale würde somit entfallen. Grundlage hierfür wäre allerdings, dass die Beschreibung des Ursache – Wirkungs – Verhaltens immer von gleichem Aufbau ist. Bei einer Kausalstruktur von geringer Komplexität wäre eine einheitliche Darstellung und damit die Programmierung eines geeigneten Skriptes vorstellbar. Komplizierte Zusammenhänge werden jedoch häufig in Prosaform verfasst. Sollte das Verhalten der Anlage jedoch zumindest in Teilen durch UW – Diagramme

darstellbar sein, so wäre die Voraussetzung zur Umsetzung einer weitgehend automatisierten Simulationserstellung gegeben.

Die Erstellung einer Simulation zum Test eines Regelkreises als einen Teil der Prozesssimulation erfordert erweiterte Anlagenkenntnisse. Das Einlesen der benötigten Informationen aus Tabellen ist hier nicht gegeben. Somit muss dieser Schritt auch weiterhin manuell ausgeführt werden.

7 Literaturverzeichnis

- [1] Mewes & Partner, WinMOD - Systemhandbuch V7.0.
- [2] K. F. Früh, U. Maier und D. Schaeel, Handbuch der Prozessautomatisierung, Oldenbourg Verlag, 2008.
- [3] Siemens AG, Umrichter SINAMICS G120C Betriebsanleitung, FW 4.4, A5E02999804A AA, 2011.
- [4] Siemens AG, SIMOCODE pro Bibliothek SP3 für PCS 7 V7.0 Programmier- und Bedienhandbuch, 2011.
- [5] NAMUR, PROFIBUS Schnittstelle für Antriebe mit Frequenzumrichtern in der Verfahrenstechnik, 2010.
- [6] Siemens AG, SIMATIC Prozessleitsystem PCS 7 PCS 7 Advanced Process Library V8.0 A5E03709255-01, 2011.
- [7] H. Töpfer, Automatisierungstechnik.
- [8] H. Watter, Hydraulik und Pneumatik, 2008.
- [9] D. Will, Hydraulik, Grundlagen, Komponenten, Schaltungen, 2008.
- [10] Festo Didactic, Proportional hydraulics, 1996.
- [11] Ulrich Buhr Industrie-Elektronik, <http://www.unipower.de/APM380.pdf>.
- [12] Siemens AG, SIRIUS SIMOCODE pro PROFIBUS Systemhandbuch NEB631605021000/RS-AA006, 2013.
- [13] Siemens AG, SIMOCODE pro Bibliothek SP3 für PCS 7 V7.0 Programmier- und Bedienhandbuch.
- [14] Siemens AG, SINAMICS G120, 2012.
- [15] ABB, PROFIBUS DP Adaptermodul RPBA-01 Benutzerhandbuch 3AFE64635476 REV F DE, 2005.
- [16] Siemens AG, Analogwertverarbeitung Funktionshandbuch A5E03461438-02, 2013.

Anhang I

Makrotyp	Makrosignal	Zieldatei	Ext Format	Int Format	Default	Typ Operand
VlvAn	Vlv_Set		Siemens 0...27648	Prozent [%] +_- 120,00		AM
	Vlv_Rbk		Siemens 0...27648	Prozent [%] +_- 120,00		AM
	Opened					
	Closed					
Vlv	Open					
	Opened					
	Closed					
	Local					
	Open_AS					
	Close_AS					
MotSpdC_Tele1_20	STW		Big Endian Word			DM
	HSW		Big Endian - 32768 ... 32767	Prozent [%] +_- 200,00		AM
	ZSW		Big Endian Word			DM
	HIW		Big Endian - 32768 ... 32767	Prozent [%] +_- 200,00		AM
	STROM		Big Endian - 32768 ... 32767	Prozent [%] +_- 200,00		AM
	TRQ		Big Endian - 32768 ... 32767	Prozent [%] +_- 200,00		AM
	POW		Big Endian - 32768 ... 32767	Prozent [%] +_- 200,00		AM
	NAM		Big Endian Word			DM
	Local					
	On_AS					
	Off_As					
	Contactore					
MotL	Start					
	Reset					
	Lamp					
	RbkRun					
	Error					
	Local					
	Repair					
	On_AS					
	Off_AS					
	EmStop					
	Error2					

Makrotyp	Makrosignal	Zieldatei	Ext Format	Int Format	Default	Typ Operand
	Error3					
	STROM		Siemens 0...27648	Prozent [%] +_- 120,00		AM
SMC_DIR	STW		Big Endian Word			DM
	ZSW		Big Endian Word			DM
	STROM		Big Endian Word 0...65535	Prozent [%] +_- 120,00		AM
	Local					
	On_AS					
	Off_AS					
Tank	Stand		Siemens 0...27648	Prozent [%] +_- 120,00		AM
	Zufuhr		Siemens 0...27648	Prozent [%] +_- 120,00		AM
	Abfuhr		Siemens 0...27648	Prozent [%] +_- 120,00		AM
Temperatur_RS	Temperatur		Siemens 0...27648	Prozent [%] +_- 120,00		AM
	Umgebung		Siemens 0...27648	Prozent [%] +_- 120,00		AM
	Heizen		Siemens 0...27648	Prozent [%] +_- 120,00		AM
	Kuehlen		Siemens 0...27648	Prozent [%] +_- 120,00		AM