

**Hochschule für Angewandte Wissenschaften Hamburg
Fakultät Life Science**

**Modellierung und Entwicklung von Optimierungsansätzen zur
Verbesserung der Robustheit des Steuerungsprogramms eines
Mischwasserrückhaltebeckens (MRB) gegenüber unscharfen
Zustandsbedingungen**

Bachelorarbeit

Im Studiengang

Verfahrenstechnik

Vorgelegt von

**Björn Warncke
1762687**

Hamburg

Am 18. April 2018

1. Gutachter: Prof. Dr.-Ing.
2. Gutachter: Dipl.-Ing.

Marc Hölling
Lars Ebert

HAW Hamburg
Firma HAMBURG
WASSER

Thema der Bachelorarbeit

Modellierung und Entwicklung von Optimierungsansätzen zur Verbesserung der Robustheit des Steuerungsprogramms eines Mischwasserrückhaltebeckens (MRB) gegenüber unscharfen Zustandsbedingungen.

Stichworte

Simulation, WinMOD, Virtuelle Inbetriebnahme, Nachbildung, Mischwasserrückhaltebecken, PROFIBUS-DP, Softwaretest

Kurzzusammenfassung

Diese Ausarbeitung beschreibt die Erstellung und Interaktion einer Steuerung mit der Simulation eines Mischwasserrückhaltebeckens. Das implementierte Steuerungsprogramm wird untersucht und es werden sinnvolle Optimierungsansätze erarbeitet. Zur Nachbildung der Prozesse wird das Simulationssystem WinMod genutzt. Es ersetzt dabei die dezentralen Slaves der Speicherprogrammierbaren Steuerung, um die Ein- und Ausgangssignale der realen Anlage abzubilden. Zur Kommunikation wird das Feldbussystem PROFIBUS DP eingesetzt.

Björn Warncke

Title of the paper

Modeling and developing of optimization concepts focusing on the robustness of the control program of a combined sewage retention basin concerning uncertain state conditions.

Keywords

Simulation, WinMOD, virtual commissioning, reproduction, combined sewage retention basin, PROFIBUS-DP, software testing

Abstract

This elaboration describes the creation and interaction between a programmable logic control and a simulation of a combined sewage retention basin. The implemented control program is examined and meaningful optimization approaches are developed. The processes were reproduced by means of the simulation system WinMOD. The virtual model replaces the decentralized slaves of the programmable logic control, to represent the input and output signals of the real system. The fieldbus system PROFIBUS-DP is used for the communication.

Inhaltsverzeichnis

1	Einleitung.....	1
1.1	Aufgabenstellung und Motivation.....	1
1.2	Vorgehensweise und Methodik	2
2	Grundlagen	3
2.1	HAMBURG WASSER.....	3
2.1.1	Unternehmensvorstellung.....	3
2.1.2	Alsterentlastung und Mischwasserrückhaltebecken	3
2.2	WinMOD Software	6
2.2.1	Systemaufbau.....	6
2.2.2	Simulationsaufbau.....	8
2.2.3	Konventionen von HAMBURG WASSER	9
2.3	Hardwareaufbau.....	10
2.3.1	Steuerung	10
2.3.2	PROFIBUS-DP	11
2.4	Ergänzende Software	12
2.4.1	Control Builder Plus.....	12
2.4.2	CoDeSys	14
3	Erstellung einer WinMOD-Simulation	15
3.1	Vorbereitung zur Simulation	15
3.1.1	Nachstellung der Slave Struktur	15
3.1.2	Einbinden der Signale als Operanden	16
3.1.3	Definition der Analogwerte.....	20
3.2	Erstellung der Simulation	22
3.2.1	Makros und Makrobibliotheken.....	22
3.2.2	Aufbau eines Makros am Beispiel Ventil.....	23
3.2.3	Einbinden der Geräte mit der Engineering Assistenz.....	31
3.3	Erstellen des physikalischen Modells	36
3.4	Aufbau der Simulation.....	46
4	Steuerungsprogramm.....	49
4.1	Vorhandene Ablaufprogramme	49
4.1.1	Entwässerung	49
4.1.2	Be- und Entlüftung.....	54
4.2	Verifizierung der Simulation Vorort	55
4.3	Arbeiten mit der Test-SPS	57
5	Validierung des Steuerungsprogramms	58

5.1	Ablauf der Schrittketten	58
5.1.1	Schrittkette 1	58
5.1.2	Schrittkette 2	59
5.1.3	Schrittkette 3	60
5.2	Störungssimulation.....	60
5.3	Optimierungsansätze	61
6	Diskussion und Fazit	63
7	Ausblick.....	64
	Literaturverzeichnis.....	VI
A	Anhang.....	VIII

Tabellenverzeichnis

Tabelle 2.1: WinMOD Präfixkonventionen.....	9
Tabelle 2.2: SPS Präfixkonventionen	10
Tabelle 3.1: Fensterschemata Menü.....	48
Tabelle 3.2: Fensterschemata Anlagen- und Prozesssimulation.....	48

Abbildungsverzeichnis

Abbildung 1.1: Aufbau zum Simulationstest [eigene Darstellung]	2
Abbildung 2.1: Konzernstruktur HAMBURG WASSER (Stand 2017) [1, S.6]	3
Abbildung 2.2: Betriebsbild der Entwässerung [7, S.1]	5
Abbildung 2.3: Betriebsbild der Be- /Entlüftung [7, S.3]	5
Abbildung 2.4: WinMOD-Systemplattform [4, S.2]	6
Abbildung 2.5: Aufbau der WinMOD Anlage [8, S.4]	7
Abbildung 2.6: WinMOD-Systemsoftware [4, S.6]	8
Abbildung 2.7: Steuerungskonfiguration der Test-SPS [eigene Darstellung]	10
Abbildung 2.8: Hardware-Aufbau Feld 7 [9]	11
Abbildung 2.9: Auswahl der SPS im Targetwechsel [eigene Darstellung]	13
Abbildung 2.10: geänderte Hardware Konfiguration [eigene Darstellung]	13
Abbildung 2.11: logische Verknüpfung des Steuerungsprogrsmms [eigene Darstellung]	14
Abbildung 3.1: Einbinden des I/O-Moduls [eigene Darstellung]	15
Abbildung 3.2: Einfügen der Stationsadresse [eigene Darstellung]	16
Abbildung 3.3: Nachgestellte Slavestruktur [eigene Darstellung]	16
Abbildung 3.4: Dateienimport zur Signaleinbindung [eigene Darstellung]	18
Abbildung 3.5: Importformat zur Signaleinbindung [eigene Darstellung]	19
Abbildung 3.6: Feldzuweisungen zur Signaleinbindung [eigene Darstellung]	19
Abbildung 3.7: Signalstruktur einer Eingangs Baugruppe [eigene Darstellung]	19
Abbildung 3.8: Liste der internen Analogformate [eigene Darstellung]	20
Abbildung 3.9: Verarbeitung eines analogen Signals [eigene Darstellung]	21
Abbildung 3.10: Bit-Zuordnung des externen Analogformats [eigene Darstellung]	21
Abbildung 3.11: Ausschnitt aus Makrobibliothek mit Makro Schieber [eigene Darstellung]	22
Abbildung 3.12: verriegelte Wende-Schützschtaltung [eigene Darstellung]	23
Abbildung 3.13: Symbolzuordnung des Makros Schieber [eigene Darstellung]	24
Abbildung 3.14: Einbinden des Active Image [eigene Darstellung]	25
Abbildung 3.15: Einbinden des Analogsignals [eigene Darstellung]	25
Abbildung 3.16: Einbinden der Meldeleuchte [eigene Darstellung]	26
Abbildung 3.17: Konfiguration des Auto-/Hand-Schalters [eigene Darstellung]	26
Abbildung 3.18: Aufbau des EinschalterVorortVentil mit realen Vorbild [eigene Darstellung]	27
Abbildung 3.19: Übersicht des Makro-Verhaltens [eigene Darstellung]	28
Abbildung 3.20: Nennstromnormierung im Schieber Makro [eigene Darstellung]	29
Abbildung 3.21: Digitalisierung des Stromsignals im Schieber Makro [eigene Darstellung]	29
Abbildung 3.22: Spannungsversorgung und Fehlerprovokation im Makro [eigene Darstellung]	30
Abbildung 3.23: Aufbau des Makros Schieber [eigene Darstellung]	30
Abbildung 3.24: Arbeitsweise der (vorwärts) Engineering Assistenz [15, S.19]	31
Abbildung 3.25: Makro Import aus Datei [eigene Darstellung]	34
Abbildung 3.26: Makro Elementbibliotheken und Elementerzeugung [eigene Darstellung]	34
Abbildung 3.27: Operanden vervollständigen [eigene Darstellung]	35
Abbildung 3.28: 400V Spannungsversorgung von Feld 4 [eigene Darstellung]	36
Abbildung 3.29: Importe, Exporte und Modifikationen durch die Engineering Assistenz [6, S. 1-1]	36
Abbildung 3.30: Entleerungsbedingungen [eigene Darstellung]	37
Abbildung 3.31: Automatisches Befüllen [eigene Darstellung]	38
Abbildung 3.32: Pegelberechnung des Trennbauwerks [eigene Darstellung]	39
Abbildung 3.33: Speicherpegel [eigene Darstellung]	40
Abbildung 3.34: Pegelberechnung des Speicherpegels [eigene Darstellung]	40

Abbildung 3.35: Brauchwasserpegel [eigene Darstellung]	41
Abbildung 3.36: vereinfachte Darstellung des Speicherpegels [eigene Darstellung]	43
Abbildung 3.37: Volumenberechnung [eigene Darstellung]	43
Abbildung 3.38: Tabelle anhand der Pumpenkennlinie [eigene Darstellung]	44
Abbildung 3.39: Durchfluss [eigene Darstellung]	44
Abbildung 3.40: Physikalisches Testmodell [eigene Darstellung]	45
Abbildung 3.41: Aufbau des Makros Niveau [eigene Darstellung]	46
Abbildung 3.42: WinMOD Projektfenster [eigene Darstellung]	47
Abbildung 3.43: Fesnterschema All [eigene Darstellung]	47
Abbildung 4.1: Mischwassersiel und Zulauf zum Speicher des MRBs [eigene Darstellung]	50
Abbildung 4.2: Entwässerungsanlage [eigene Darstellung]	52
Abbildung 4.3: Schrittkette 1 [7]	52
Abbildung 4.4: Schrittkette 2 [7]	53
Abbildung 4.5: Schrittkette 3 [7]	53
Abbildung 4.6: Entlüftungsanlage [eigene Darstellung]	54
Abbildung 4.7: Schieberdarstellung des Steuerungsprogramms [7]	56
Abbildung 4.8: Pegelübersicht im Steuerungsprogramm [7]	56
Abbildung 5.1: Fehler im Steuerungsprogramm zum Starten der Schrittkette 2 [eigene Darstellung]	59
Abbildung 5.2: Optimiertes Steuerungsprogramm der Schrittkette 2 [eigene Darstellung]	60

Abkürzungsverzeichnis

Abkürzung	Bedeutung
AKS	Anlagenkennzeichnungssystem
AöR	Anstalt öffentlichen Rechts
AS	Automatisierungssystem
AWL	Anweisungsliste
CBP	Control Builder Plus
CDS	Control Development System
DP	Dezentrale Peripherie
FBS	Funktionsbausteinsprach
HSE	Hamburger Stadtentwässerung
KOP	Kontaktplan
mNN	Meter Normal Null
MRB	Mischwasserrückhaltebecken
mWS	Meter Wassersäule
PROFIBUS	Process Field Bus
SPS	Speicherprogrammierbare Steuerung
ST	Strukturierter Text

Symbolverzeichnis

Symbol	Einheit	Beschreibung
V	m ³	Volumen
h	m	Höhe
	mNN	
	mWS	
a	m	Breite
r	m	Radius
C	Mbit/s	Datenübertragungsrate

1 Einleitung

1.1 Aufgabenstellung und Motivation

Die vorliegende Bachelorarbeit wird bei der Hamburger Stadtentwässerung (HSE) durchgeführt. Im Rahmen dieser Abschlussarbeit wird der vollständige Prozess des Mischwasserrückhaltebeckens (MRB) so nachgestellt, dass sämtliche Ein- bzw. Ausgangssignale der Steuerung auf den simulierten Komponenten des MRBs abgebildet werden. Dabei werden alle Gerätekomponenten, wie Bedien- und Anzeigeelemente, Sicherungen, Pumpen usw., die im MRB vorhanden sind und das physikalische Modell des MRBs mit der Software WinMOD der Firma Mewes & Partner nachgebildet.

Als speicherprogrammierbare Steuerung (SPS) wird die AC500 der Firma ABB eingesetzt, welche über das Feldbussystem Profibus DP alle Signale der dezentralen Peripherie verarbeitet. Die Simulation ersetzt das reale MRB durch das virtuell nachgebildete Modell, in dem das Bussystem auf eine Rechneroberfläche mit Simulationssystem umgeleitet wird. In der Simulation sollen die Signale so steuerbar sein, dass insbesondere reale Ereignisse, wie Starkregen, Ausfall von Gerätekomponenten oder beliebige Testszenarien, über einen frei gewählten Zeitraum simuliert werden können.

Das MRB speichert bei Starkregen zur Vermeidung von Überläufen in nächstliegende Gewässer Mischwasser, um es nach dem Regenereignis kontrolliert in das Abwassernetz abzugeben. Die Anlage arbeitet im Automatikbetrieb. Die derzeit implementierte SPS ist wenig tolerant gegenüber unscharfen Zustandsbedingungen der Betriebsmittel. Die unscharfen Zustandsbedingungen der Betriebsmittel beziehen sich auf die für den Ablauf der Entwässerung des MRBs benötigten Geräte. Liegen eine oder mehrere Störungen vor, soll auch weiterhin eine Entleerung des Speichers sichergestellt werden. Die Abläufe dazu sollen im Steuerungsprogramm implementiert sein. Die Erfahrungen zeigen allerdings, dass bei Störungen der relevanten Betriebsmittel die Entleerung unterbrochen wird. Zur Steigerung der Robustheit und Reduzierung der Entstörungseinsätze soll das Steuerungsprogramm nun tolerierbare Zustände, wie den Ausfall einzelner Pumpen erkennen und den Anlagenbetrieb auch mit Einschränkungen gewährleisten.

Die Simulation hat den Vorteil das Kosten und Aufwand geringgehalten werden und das Steuerungsprogramm auch außerhalb von Starkregenereignissen getestet werden kann. Es kann je nach Detaillierungsgrad der gesamte Prozess sowie auch einzelne Signale und deren Einfluss auf den Gesamtprozess getestet werden. Des Weiteren können Änderungen sowie Optimierungsansätze im Steuerungsprogramm unmittelbar überprüft werden.

1.2 Vorgehensweise und Methodik

Begonnen wird mit den Vorarbeiten zur WinMOD Simulation, der Nachbau der Slave Struktur und das Bearbeiten der Signalliste, um diese in die simulierten Module zu laden. Anschließend wird das Steuerungsprogramm in die separate Test-SPS (typgleich der Vorort-SPS) implementiert um den richtigen Aufbau der Peripherie zu verifizieren. Gleiches wird zur Bestätigung auch mit der Vorort-SPS im MRB durchgeführt. Anhand der Dokumentationen wird die Simulation erstellt und diese an der Vorort-SPS des MRB getestet. Hierbei wird vor allem darauf geachtet, dass jedes Gerät auch über die vorhandenen Panels im MRB angesprochen werden können. Schließlich wird das Steuerungsprogramm an der separaten Test-SPS mit Hilfe der Simulation geprüft und optimiert.



(a) Test-SPS



(b) Vorort-SPS

Abbildung 1.1: Aufbau zum Simulationstest [eigene Darstellung]

Grundsätzlich wird versucht den Ablauf zu den Ergebnissen auch in der Thesis darzulegen. Für die Erstellung der Simulation wurde auf die Handbücher von Mewes & Partner und ein bereits erstelltes WinMOD-Projekt, Nachbildung eines Pumpwerkes, zurückgegriffen. Teile der Simulation wurden anhand der Stromlaufpläne, Funktionsbeschreibungen und Geräte- und Modulhandbüchern nachgebildet. Für die Hardware und Steuerungsprogrammsoftware waren neben der Literaturrecherche, vor allem zur Fehleranalyse, qualitative Erhebungen notwendig. Zur Optimierung des Steuerungsprogramms und in Teilen auch bei der Erstellung der Simulation war teilweise ein Trial-and-Error am erfolgreichsten.

2 Grundlagen

2.1 HAMBURG WASSER

2.1.1 Unternehmensvorstellung

Die Hamburger Stadtentwässerung AöR (HSE) und die Hamburger Wasserwerke (HWW) bilden durch den Zusammenschluss am 1. Januar 2006 den Gleichordnungskonzern HAMBURG WASSER (Abbildung 1.1), das zweitgrößte deutsche Trinkwasserver- und Abwasserentsorgungsunternehmen in kommunaler Hand. Die beiden Unternehmen blicken auf eine mehr als 170-jährige Geschichte mit der Ressource Wasser zurück. Die Trinkwasserversorgung sowie die Abwasserbeseitigung werden in Hamburg und dessen Metropolregion von mehr als 2000 Mitarbeitern sichergestellt und genügen dabei höchsten Qualitätsansprüchen. Dargelegt wird diese Aussage in der Umwelterklärung von HAMBURG WASSER [siehe 1, S. 6].

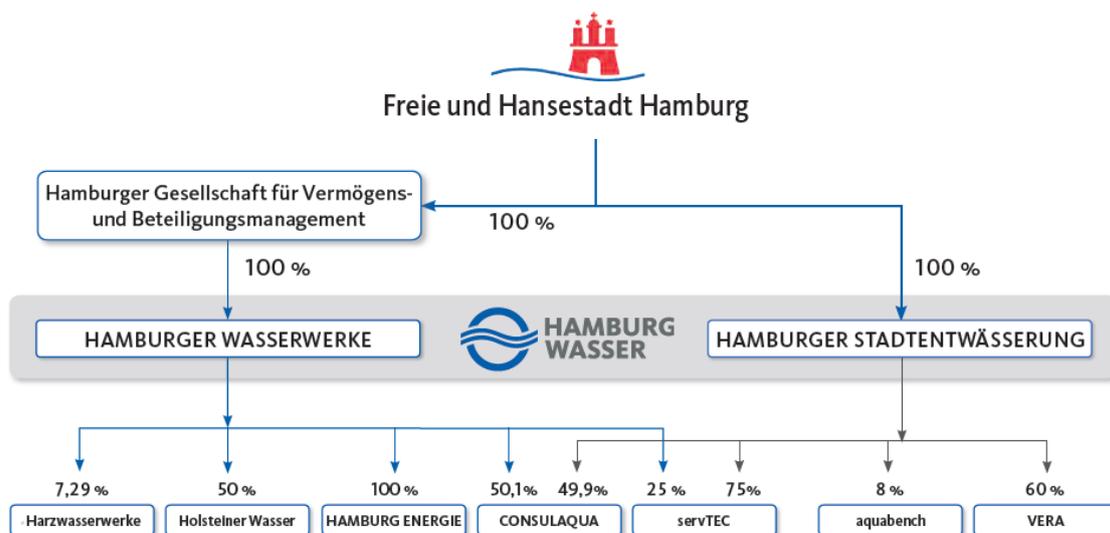


Abbildung 2.1: Konzernstruktur HAMBURG WASSER (Stand 2017) [1, S.6]

2.1.2 Alsterentlastung und Mischwasserrückhaltebecken

Zur Verbesserung der innerstädtischen Gewässergüte wurde 1982 das Alsterentlastungskonzept erstellt. Grundlage hierfür war ein umweltpolitisches Konzept der Umweltbehörde. Ziel ist es, die Alster und deren Nebengewässer vor Überläufen aus dem Mischwassersielnetz zu schützen. Der Grundgedanke sieht vor das Sielnetz in überschaubare Teileinzugsgebiete zu gliedern. Für jedes

Teilgebiet soll ausreichend Ableitungskapazität, sowie ausreichend Speicherkapazität bei Starkregenereignissen gewährleistet werden [siehe 2, S. 4].

Im Rahmen des Alsterentlastungskonzeptes wurde 1993 das Mischwasserrückhaltebecken (MRB) Schädlerstraße als erstes Rückhaltebecken Hamburgs in Betrieb genommen. Bei Überlastung des Sielnetzes kann das MRB bis zu 7400 m³ Mischwasser aufnehmen. Wird ein bestimmter Pegel im Netz überschritten erfolgt über den Zulauf die Beschickung des MRBs. Die kontrollierte Entleerung erfolgt nach Beendigung des Regenereignisses. Sollte die Menge des Ab- und Regenwassers die Kapazität des MRB überschreiten läuft über den Überlauf das Mischwasser in den anliegenden Gehölzgraben und von dort in die Wandse.

Das MRB verfügt insgesamt über sechs Pumpen, den zwei Hauptentleerungspumpen und die zwei Restentleerungspumpen zur Speicherentleerung. Für die Sohlenreinigung des Speicherbeckens befindet sich im Brauchwasserbecken die fünfte Pumpe und im Maschinenraum die Entwässerungspumpe. Diese dient der Entwässerung des Pumpenraums.

Die Anlage besitzt insgesamt 15 Schieber. Zusätzlich befindet sich im Zulauf das Zulaufschütz um bei einer Begehung des Speicherbeckens den Eintritt des Mischwassers zu verhindern. Die Entleerungspumpen verfügen über einen Saug- sowie Druckschieber und die beiden Hauptentleerungspumpen zusätzlich noch über einen Spülschieber. Bei der Brauchwasser- und Entwässerungspumpe handelt es sich um Tauchpumpen. Dadurch entfallen Saug- sowie Druckschieber. Das Brauchwasserbecken wird über ein Magnetventil versorgt. Für die Versorgung des Speicherbeckens mit Brauchwasser werden drei Absperrschieber eingesetzt. Zur Speicherreinigung stehen noch zwei weitere Absperrschieber zur Verfügung.

Es befinden sich neun teilweise redundant angebrachte Druckmesser zur Pegelbestimmung in der Anlage. Eine Radarmessung zur Volumenbestimmung ist oberhalb des Speicherbeckens angebracht. Schließlich gibt es noch zwei Schwimmschalter, einer davon bei der Entwässerungspumpe, sowie einer bei der Brauchwasserpumpe, als Redundanz zum Druckmesser im Brauchwasserbecken. Im Trennbauwerk sind insgesamt drei Druckmessungen, die beiden zum Zulauf angebrachten Pegelmessungen sind dabei redundant ausgeführt. Im gleichen Mischwassersiel am Ablauf befindet sich die dritte Messung. Die Durchflussmessung befindet sich oberhalb der Pumpen. Eine Messung dient der Bestimmung des Niveaus im Gehölzgraben. Die beiden redundanten Druckmessstellen werden zu einem Pegel verarbeitet, der für die Entwässerung des Speicherbeckens verantwortlich ist. Wie die beiden Messungen des Speicherbecken-Pegels, sind auch die Druckmessungen im Überlauf redundant ausgeführt. Der Ablauf des Entwässerungsprogramms findet sich in Abschnitt 4.1.1.

Die Entlüftung wird insgesamt über sieben Lüfter und fünf Luftklappen gesteuert. Der Traforaum sowie die Schaltanlage verfügen jeweils über einen Lüfter. Diese dienen dem Wärmeabtransport. Die Be- und

Entlüftung des Speichers und Speicherniedergangs wird von zwei Lüftern übernommen. Die Belüftung des Speichers verfügt über drei Luftklappen. Eine befindet sich direkt am Lüfter, die beiden anderen Luftklappen dienen zur Belüftung des oberen und unteren Speicherbeckens. Der Pumpenraum verfügt ebenfalls über einen Belüfter (einschließlich Luftklappe) und einen Entlüfter. Das Entlüftungsprogramm wird in Abschnitt 4.1.2 erläutert. Einen Überblick über die Geräte im MRB Schädlerstraße geben die beiden Abbildungen 2.2 und 2.3. (Betriebs-bilder des Steuerungsprogrammes).

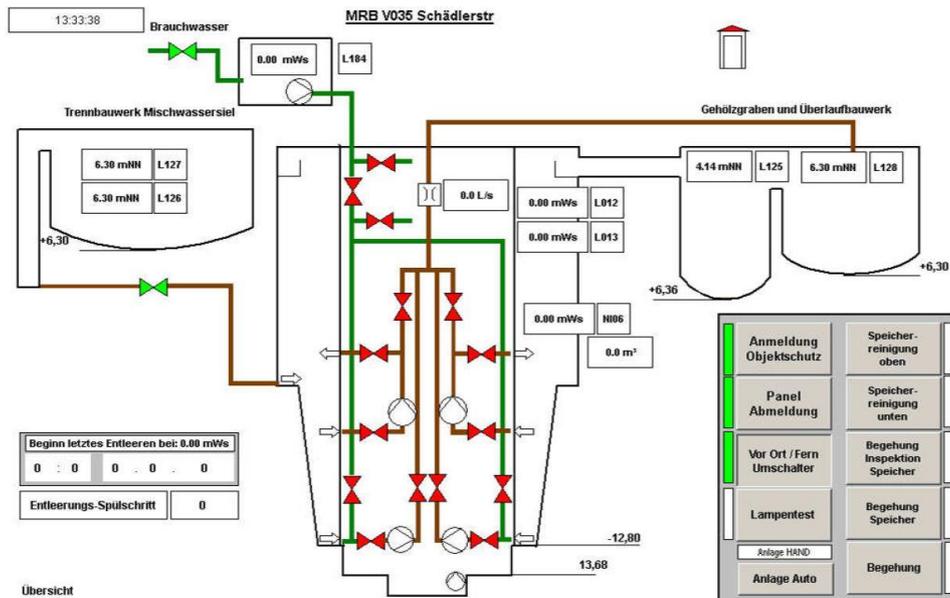


Abbildung 2.2: Betriebsbild der Entwässerung [7, S.1]

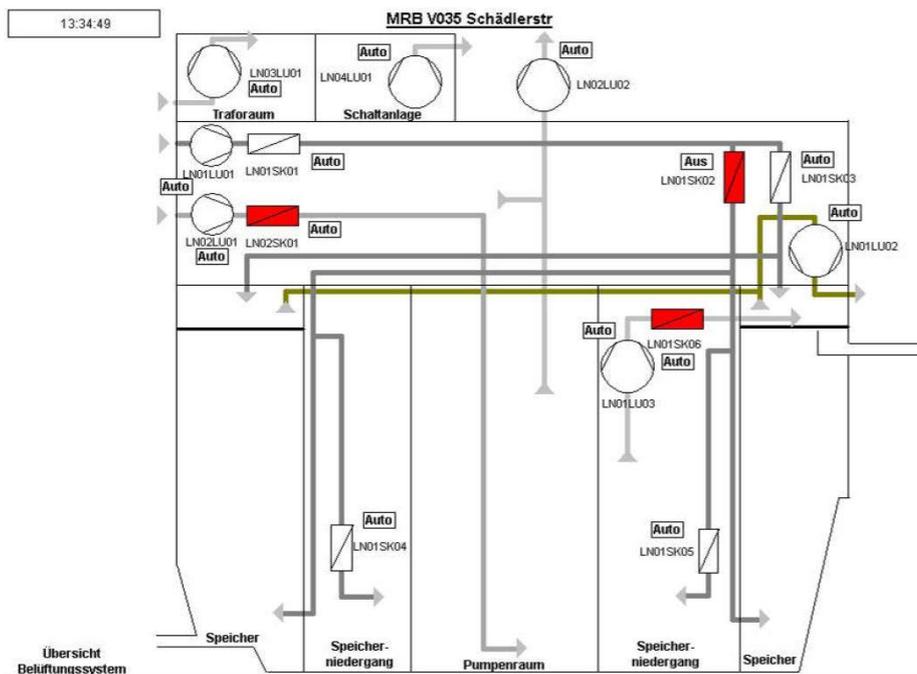


Abbildung 2.3: Betriebsbild der Be-/Entlüftung [7, S.3]

2.2 WinMOD Software

Folgend werden die soft- und hardwaretechnischen Grundlagen für die Modellierung und Simulation des MRBs Schädlerstraße beschrieben, die für die Überprüfung und Optimierung des vorhandenen Steuerungsprogramms nötig sind.

2.2.1 Systemaufbau

WinMOD dient der Anlagenperipherie- und Prozesssimulation und wurde für die Modellierung und Simulation aufgrund vorhandener Lizenzen und bereits in der Vergangenheit erstellter Projektumsetzungen von HAMBURG WASSER vorgegeben. Die Systemplattform (Abbildung 2.4) besitzt die gleichen Schnittstellen äquivalent zur realen Anlage und Automatisierungssystem (AS). Es werden nicht nur Ein- und Ausgangssignale abgebildet, sondern auch die reale Anlage und deren Prozesse werden nachgebildet. Es besteht für das AS kein Unterschied zwischen der Simulation und der realen Anlage. Das Ziel von WinMOD ist es, die Automatisierungssoftware unabhängig von der realen Anlage zu testen, sowie die Funktionen und Funktionsabläufe unter Echtzeitbedingungen zu beobachten. Daher ist WinMOD besonders geeignet für virtuelle Inbetriebnahmen, Prüfungen und Validierung von Software und der Erstellung von Trainings-Simulatoren. In Abbildung 2.4 wird deutlich, dass durch die Anbindung der realen und virtualisierten Anlage am gleichen Feldbussystem ein einfaches Umschalten durch den Austausch der Busverbindung möglich ist.

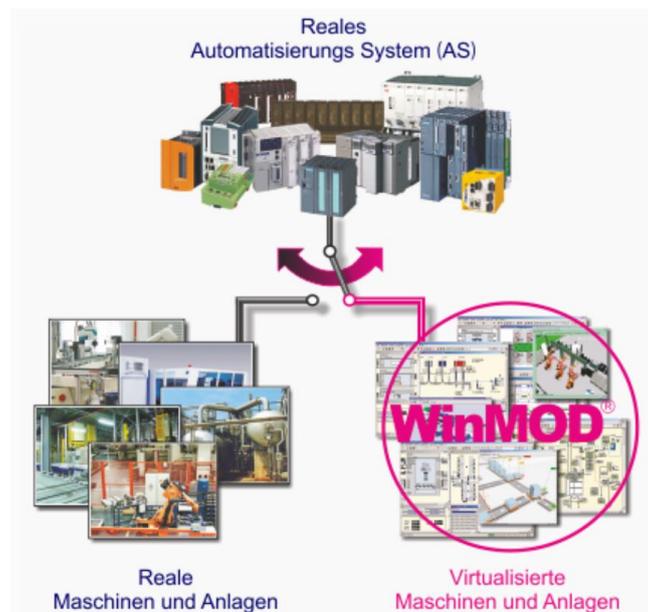


Abbildung 2.4: WinMOD-Systemplattform [4, S.2]

Die WinMOD Anlage kann durch fünf Bereiche ausreichend beschrieben werden. In Abbildung 2.5 sind diese Bereiche durch Buchstaben gekennzeichnet. In Bereich a, der WinMOD Konfiguration, wird die Systemkopplung gezeigt. Die Konfigurationen sind dabei systembezogen und werden für viele Automatisierungssysteme und Schnittstellen angeboten. Die Simulation ist weitgehend unbeeinflusst von der Schnittstelle. Der Bereich b stellt nun die WinMOD-Systemsoftware dar, in der die Simulation erstellt und ausgeführt wird. Der Bereich c beschreibt die möglichen Add-Ons, hierbei besonders hervorzuheben die WinMOD-Engineering Assistenz, bei der mit Hilfe von Text-Dateien Bausteine und Verbindungen importiert und exportiert werden können. Außerdem der WinMOD-Recorder mit dem Signale über eine Laufzeit visualisiert werden können. Über den WinMOD-3D-View wird eine dreidimensionale Simulation ermöglicht und via WinMOD-Net ist eine Simulation zwischen mehreren Rechnern möglich. Im vierten Bereich, d, werden die WinMOD-Grundlagen zusammengefasst. Bereitgestellte Bibliotheken und Vorlagen bilden die Grundlage der Anlage. Schließlich der letzte Bereich e, WinMOD-Workflows-Methoden-Integration. Hierin fällt der Import und Export von Daten. Beispielweise Operanden, diese Datenpunkte, in denen die Werte von Ausgängen abgelegt werden und aus denen Eingänge ihren Wert beziehen können.

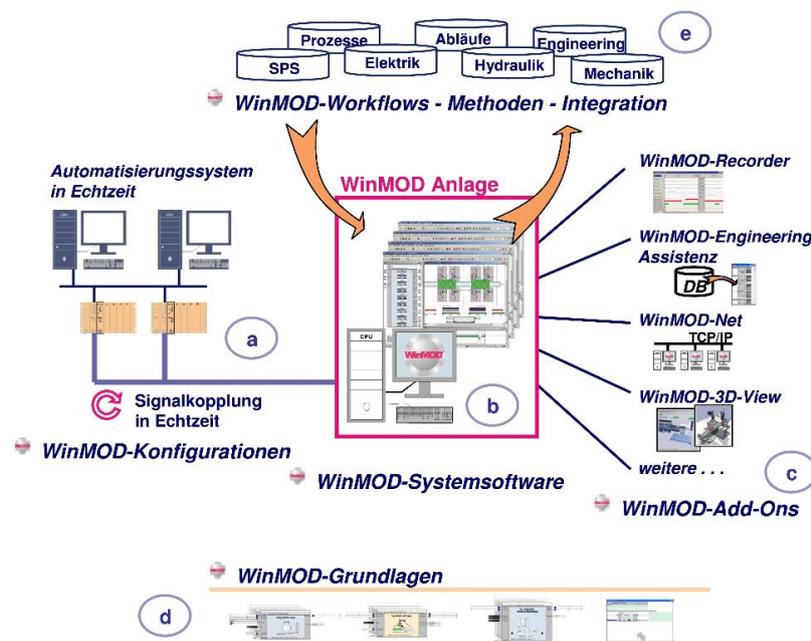


Abbildung 2.5: Aufbau der WinMOD Anlage [8, S.4]

Die Systemsoftware wird nochmals in drei weitere Bereiche unterteilt (Abbildung 2.6). Angefangen mit der Peripheriekopplung, über die Anlagensimulation, hin zur Prozesssimulation. Die Peripheriekopplung kann ähnlich wie bei der *Engineering Assistenz* mit Hilfe von einer erstellten Text-Datei konfiguriert werden. Aus einer Exportliste des Steuerungsprogramms in WinMOD werden Ein- und Ausgangssignale einfach generiert und implementiert. Näheres zur Erstellung der Peripherie folgt

im Kapitel 3.1.2. Zur Erstellung der Anlagen- beziehungsweise Prozesskomponenten wird auf die *Engineering Assistenz* zurückgegriffen. Für die Anzahl und Anordnung dieser Komponenten werden Dateien angefertigt und mit vorher erstellten Bibliotheken in ausgewählte Simulationsdateien geschrieben.

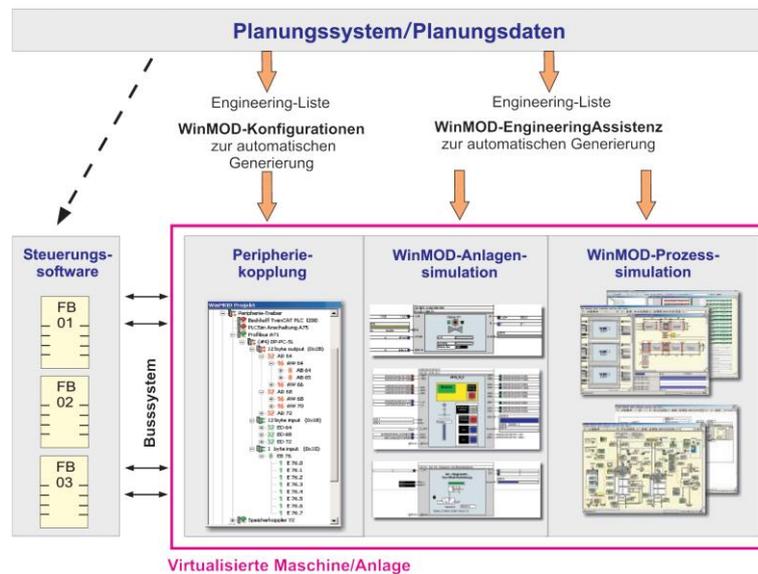


Abbildung 2.6: WinMOD-Systemsoftware [4, S.6]

2.2.2 Simulationsaufbau

Die Grundbestandteile der WinMOD-Simulation sind die Signalelemente und die Simulationselemente in logischer Verknüpfung. Je nach Simulationstiefe werden die Signalelemente mit lokalen oder vor allem globalen Operanden verknüpft. Unterschieden wird in vier Schichten.

In der Schicht 0 befindet sich die I/O-Simulation. WinMOD kommuniziert über die gewählte Konfiguration mit dem E/A-System. Die Binären und Analogen Signale sind als WinMOD-Operanden verfügbar und die analogen Signale werden entsprechend des Steuerungssystem formatiert. Die Signale können beobachtet und im Force-Modus manipuliert werden.

Die Funktionseinheiten bilden die 1. Schicht. Vor allem für den Zweck von Softwareprüfungen wird das Projekt in Funktionseinheiten strukturiert, zum Beispiel Antriebe und Messsysteme. Das Verhalten wird durch Simulationselemente und Makros simuliert.

Die 2. Schicht, die Funktionsgruppen beinhalten das Zusammenwirken mehrerer Funktionseinheiten zum Testen von Schrittabläufen.

Schließlich die Verfahrenssimulation als 3. Schicht hierfür ist eine tiefgehende Prozesskenntnis erforderlich [siehe 3, S. 11-2].

2.2.3 Konventionen von HAMBURG WASSER

Die interne und externe Kommunikation funktioniert in WinMOD über sogenannte Operanden als Abbildung von Signalen. Diese können lokal oder global vorkommen. Während lokale Operanden lediglich innerhalb einer Simulationsdatei gelten, sind globale Operanden im gesamten Projekt gültig. Die WinMOD und HAMBURG WASSER Konventionen sind nun Vereinbarungen die bezüglich der Benennung der im WinMOD-Projekt verwendeten Signale getroffen werden. Unterschieden werden kann zwischen dem Präfix und der Folgebezeichnung des Signals. Bei der internen Kommunikation handelt es sich um ein WinMOD-Signal und bei der externen Kommunikation um ein SPS-Signal. Über die Konventionen können globale Operanden eindeutig im WinMOD-Projekt erkannt und bestimmt werden. In Tabelle 2.1 sind die WinMOD Präfixkonventionen mit Beispiel und Bedeutung dargestellt. Die Folgebezeichnung der WinMOD Bezeichnung wird logisch bestimmt wie beispielsweise *tEin* für die Einschaltzeit.

Tabelle 2.1: WinMOD Präfixkonventionen

Präfix	Beispiel	Bedeutung
@	@Reset	Signal zur Simulationssteuerung z.B. Reset und setzen auf Anfang
%	%tEin	Parameter z.B. Einschaltzeit
#	#10%	Konstante z.B. 10%
\$	\$i24V	Physikalische Größe bzw. physikalischer Zustand z.B. Eingangssteuerspannung
f_	f_MV	Fehlerprovokation z.B. Motorvollschutz ausgelöst

Für das Präfix der externen Kommunikation mit der SPS wird eine zweistellige Bezeichnung genutzt. Die erste Stelle bezeichnet die Richtung des Signals, I steht für ein Eingangssignal der SPS und Q für ein Ausgangssignal. Die zweite Stelle bezeichnet die Art des Signals. X, als Datentyp Bool, wird für ein binäres und W, als Datentyp Word, für ein Analogsignal verwendet.

Zusätzlich zu dem Präfix wird zur kompletten Darstellung des Signals noch das Anlagenkennzeichen und die Signalbezeichnung des Fernwirksystems angefügt. Die Bezeichnung des Fernwirksystems kann in der Exceltabelle im Anhang X zu den Ein- und Ausgängen eingesehen werden. Für die Anlagenkennzeichnungen wird auf das Anlagenkennzeichnungssystem (AKS) von HAMBURG WASSER zurückgegriffen. Dieses besteht grundsätzlich aus fünf Stufen, wobei die fünfte Stufe die Zählzahl bildet, auf letzteres wird zu Gunsten der Nummerierung aus dem Fernwirksystem verzichtet. Die erste Stufe wird aus einem Buchstaben und drei folgenden Zählzahlen gebildet. Die weiteren Stufen bestehen aus zwei alphabetischen Zeichen und einer zweistelligen Zählzahl. So kann beispielhaft das Ausgangsbit des Einschaltbefehl der Hauptentleerungspumpe 1 zusätzlich zum Präfix Qx mit

V035SB01PA01PU01_339 bezeichnet werden. V035, steht für das MRB 035 oder auch MRB Schädlerstraße. Folgend SB01, das Speicherbecken eins der Anlage. Weiter mit PA01, das steht für Pumpenanlage eins und schließlich PU01, als Bezeichnung für die Pumpe eins. Die komplette Bezeichnung ist Qx_V035SB01PA01PU01_339. Das Komplette AKS befindet sich im digitalen Anhang unter der Datei AKS_1-4Stufe_V10.pdf. In Tabelle 2.2 dargestellt die Präfixkonventionen der SPS mit einem Beispiel und Bedeutung.

Tabelle 2.2: SPS Präfixkonventionen

Präfix	Beispiel	Bedeutung
Ix	Ix_V035SB01PA01PU03_	Eingangsbit
Iw	Iw_V035SB01PA01PU03_	Eingangsword Motorstrom Restentleerungspumpe 1
Qx	Qx_V035SB01PA01PU03_339	Ausgangsbit Betrieb Restentleerungspumpe 1

2.3 Hardwareaufbau

2.3.1 Steuerung

Die Speicherprogrammierbare Steuerung AC 500 des Herstellers ABB dient der Automatisierung von Prozessen. Hierbei werden binäre, digitale und analoge Ein- und Ausgangssignale auf entsprechenden Modulen erfasst und weiterverarbeitet. Eine funktionsfähige SPS setzt sich mehreren Modulen zusammen. Dazu gehören mindestens die CPU und ein Kommunikationsmodul. Um ein Ferneinwirken auf den Prozess zu ermöglichen wird zumeist noch ein Ethernet Modul angeschlossen. Je nach Feldbussystem können nun die eigentlichen I/O Module angeschlossen werden.

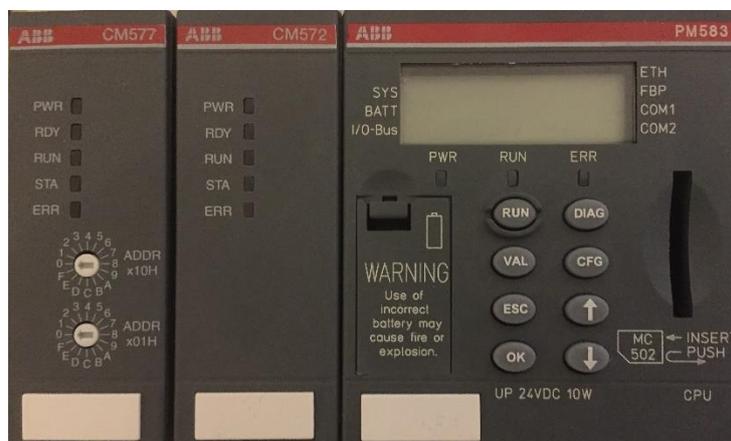


Abbildung 2.7: Steuerungskonfiguration der Test-SPS [eigene Darstellung]

Die dargestellte Steuerungskonfiguration der AC 500 (Abbildung: 2.7) entspricht der vorliegenden Test-SPS. Zentral die CPU der AC500, PM590 ETH, links der CPU das Kommunikationsmodul, der Profibus-Master CM 572 und linksseitig angebracht das Ethernet-Modul CM 577.

Über das Profibuskabel sind die anderen Hardware-Komponente, die Profibus Slaves angeschlossen. Es handelt sich ebenfalls um Baugruppen der Firma ABBILDUNG Die 21 I/O-Module, CI542 IO, sind auf neun Schaltschränke verteilt. Die binären Eingangs- (DI561), binären Ausgangs- (DO561) und analogen Eingangsmodulen (AI561) sind über den Slave mit dem Master verbunden. Beispielhaft in Abbildung 2.8 dargestellt.

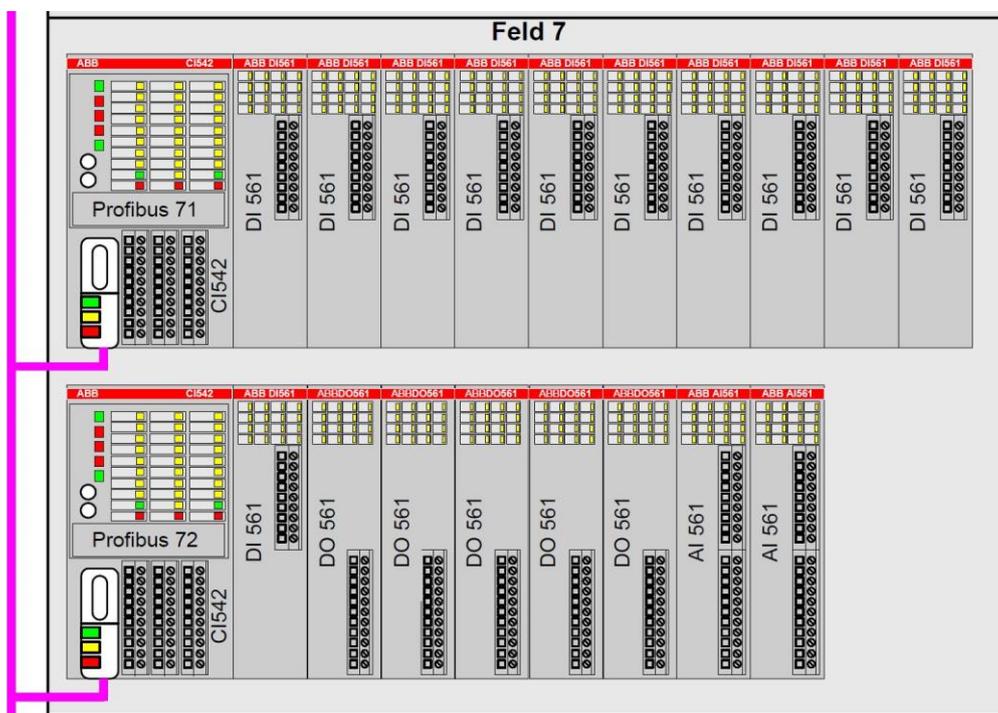


Abbildung 2.8: Hardware-Aufbau Feld 7 [9]

2.3.2 PROFIBUS-DP

Die vorliegende SPS arbeitet mit der sogenannten PROFIBUS-DP (dezentrale Peripherie) Feldbus-Kommunikation. PROFIBUS steht für „Process Field Bus“ und ist ein herstellerunabhängiger Standard für die Vernetzung von Feldgeräten. Eine geschirmte Zweidrahtleitung dient als Übertragungsmedium. Abhängig von der Leitungslänge ist die Übertragungsrate C. So können bei 100 m bis zu $12 \frac{\text{Mbit}}{\text{s}}$ übertragen werden. Bei einer Leitungslänge von 1000m reduziert sich die Übertragungsrate C auf nur noch $9,6 \frac{\text{kbit}}{\text{s}}$. Die maximale Teilnehmerzahl von Stationsadressen beträgt 127. Unterschieden wird

zwischen aktiven und passiven Teilnehmern. Der aktive Teilnehmer erhält temporären Zugriff auf den Bus und darf Datentelegramme senden. Nach dieser Zeit wird die Zugriffsberechtigung an den nächsten aktiven Teilnehmer weitergegeben, dieses Zugriffsverfahren wird „Token Passing“ genannt. Passive Teilnehmer erhalten keine Zugriffsberechtigung. Sind einem aktivem Teilnehmer (Master) passive Teilnehmer (Slave) zugeordnet, führt der Master den Datenaustausch mit den ihm zugeordneten Slaves durch, während er die Zugangsberechtigung hat. Die Hauptaufgabe des Systems ist die Aktualisierung des Prozessabbilds der Ein- und Ausgänge der Slavedaten. Eine Priorisierung einzelner Teilnehmer ist nicht vorgesehen und durch die hohen Übertragungsraten und daraus folgenden Datenübertragungszeiten im μs -Bereich nicht nötig. Für den Datendurchsatz entscheidend ist im Regelfall der Anwendungsprozess, mit Zykluszeiten von mehreren Millisekunden. Durch die Arbeitsweise, Übertragungsrate und Busaufbau werden Daten öfter übertragen und aktualisiert, als dass der Anwender darauf zugreifen kann. Es stehen dadurch zum Zeitpunkt des Zugriffs immer die neuesten Daten zur Verfügung [siehe 10, S. 172].

2.4 Ergänzende Software

2.4.1 Control Builder Plus

Der Control Builder Plus (CBP) ist ein Softwaretool zur einfachen und bequemen Konfiguration der AC500-Steuerung und deren verfügbaren Geräte:

- CPU
- I/O-Module
- Alle Schnittstellen und Feldbusse

Ein einfaches Beispiel zum Erstellen eines Projekts findet sich im Praktikumsbericht zur Einrichtung einer dezentralen Steuereinheit als Vorbereitung zum Testen eines Steuerungsprogramms [siehe 13] Änderungen bestehender Hardwarekonfigurationen, also Änderungen eingefügter Geräte lassen sich einfach über das Menü des jeweiligen Ziels ermöglichen. In Abbildung 2.9 ist das Menü zum Wechsel des CPU-Typknoten und in Abbildung 2.10 die neue Konfiguration nach auswählen des neuen Bauteils ersichtlich. Mit dem Targetwechsel (SPS auswählen) werden alle identischen Schnittstellen der alten und neuen SPS mit der vollständigen Konfiguration auf das neue Target übertragen.

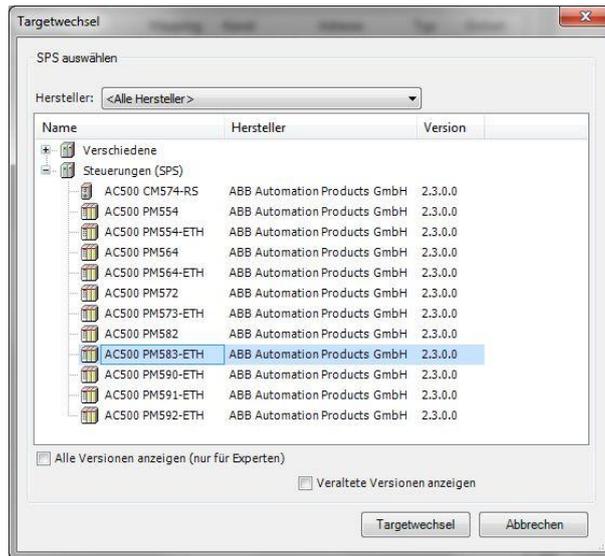


Abbildung 2.9: Auswahl der SPS im Targetwechsel [eigene Darstellung]

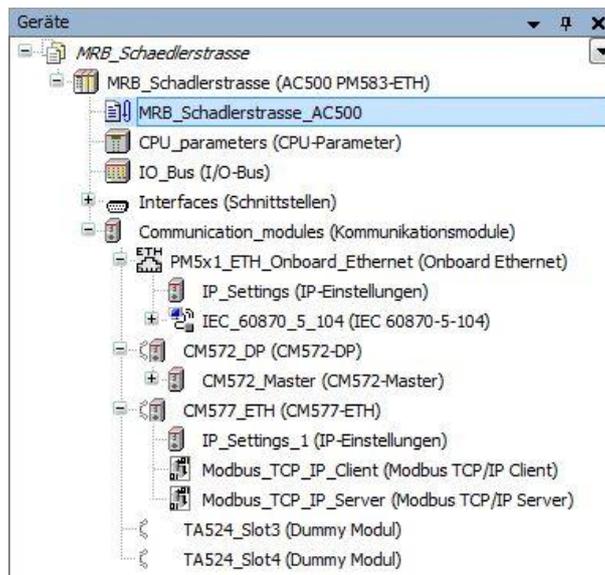


Abbildung 2.10: geänderte Hardware Konfiguration [eigene Darstellung]

Die Programmierung, Fehlersuche und Fehlerbehebung, sowie der allgemeine Zugriff auf das Programm erfolgt mit Hilfe des eingebundenen Plug-ins zur Steuerungs-Entwicklungsumgebung CoDeSys [siehe X]. Die Adress- und Variablendeklaration wird im CBP erstellt und in CoDeSys übernommen.

2.4.2 CoDeSys

CoDeSys steht für Controller Development System und ist eine Entwicklungsumgebung für Steuerungen nach SPS-Norm DIN EN 61131-3 (siehe 12, S. 2). Bei der deutschen Norm handelt es sich um die harmonisierte internationale Norm IEC 61131-3 zur Standardisierung der SPS-Programmierung mit herstellerunabhängigen Programmiersprachen. Die Norm involviert fünf Programmiersprachen. Die beiden textuellen Fachsprachen, die Anweisungsliste (AWL) und der strukturierte Text (ST) zur strukturierten Programmierung angelehnt an Pascal. Die beiden grafischen Fachsprachen, den Kontaktplan (KOP) und die Funktionsbausteinsprache (FBS). Schließlich die übergeordnete Ablaufsprache, die textuelle sowie grafische Elemente enthält [siehe 12, S. 19].

Vorgegeben durch die Norm wird ebenfalls der Befehlsvorrat für die Programmierung. Dazu gehört unter anderem die Signalabfrage. Funktional miteinander verbundene Variablen werden als Verknüpfungen bezeichnet. Beispiele sind die UND-, sowie ODER Verknüpfung. In den meisten Fällen setzt sich eine logische Funktion aus mehreren Grundverknüpfungen zusammen. Abbildung 2.11 zeigt eine logische Verknüpfung aus dem Steuerungsprogramm der Schädlerstraße. Die Variablen die den Eingang bilden, sind zwischengespeicherte interne Variablen aus weiteren zusammengesetzten Verknüpfungen. Das Ausgangssignal dient ebenfalls nur als Zwischenspeicher und wird an anderer Stelle weiterverarbeitet [siehe 12, S.58-80]. Ein Beispiel zur variablen Deklaration und Erstellung eines Funktionsplans findet sich in Anhang (Praxissemesterbericht). Sämtliche Signalverknüpfungen finden sich auch in WinMOD und dienen dort als Grundlage für die Simulationen und Makros, näheres dazu in Kapitel 3.

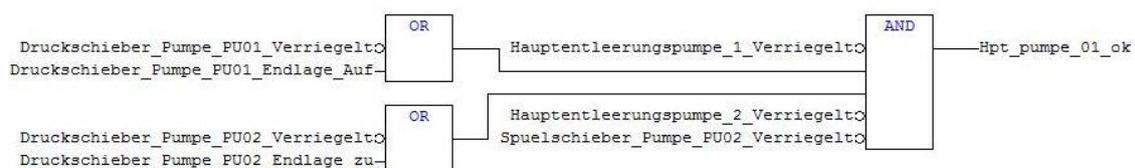


Abbildung 2.11: logische Verknüpfung des Steuerungsprogramms [eigene Darstellung]

3 Erstellung einer WinMOD-Simulation

Folgend wird die Vorgehensweise zur Erstellung der Simulation und der Aufbau des MRBs mit Hilfe der WinMOD Handbücher [siehe 7 und 8] erläutert. Die Grundelemente der WinMOD-Simulation werden im Beispiel Makro des Schiebers beschrieben. Beim Aufbau wurde sich an bereits bestehende Simulationsmodellen von HAMBURG WASSER orientiert.

3.1 Vorbereitung zur Simulation

3.1.1 Nachstellung der Slave Struktur

Die Verbindung des Simulations-Rechners wird über die PCIe-Schnittstelle mit der externen Multislave- Karte, welche über das Profibuskabel mit der AC500 Steuerung verbunden ist, hergestellt. Hierzu wird die WinMOD-Konfiguration A740 Profibus Multislave benötigt. Diese stellt die logische sowie physikalische Verbindung des Feldbussystems Profibus-DP her. Um die Kommunikation WinMODs mit der SPS zu gewährleisten muss der Aufbau der realen Slavestruktur nachgebildet und deren Stationsadressen mit dem Master abgestimmt werden. Geladen wird der nötige Peripherie Treiber durch auswählen im Projektfenster (Umschalt+F4) und hinzufügen über die rechte Maustaste. Durch öffnen des Konfigurators und der Buskonfiguration können die Slaves eingefügt werden. Das dezentrale Peripherie Modul CI542 wird über drag and drop dem Profibus hinzugefügt und das eigentliche IO-Modul ausgewählt (Abbildung 3.1) und mit der Stationsadresse versehen (Abbildung 3.2), anschließend können die Eingangs- und Ausgangsmodule angefügt werden. Die nachzustellende reale Slavestruktur (Abbildung 3.3) kann dem Anhang X entnommen werden und liegt außerdem als CBP-Projekt vor.

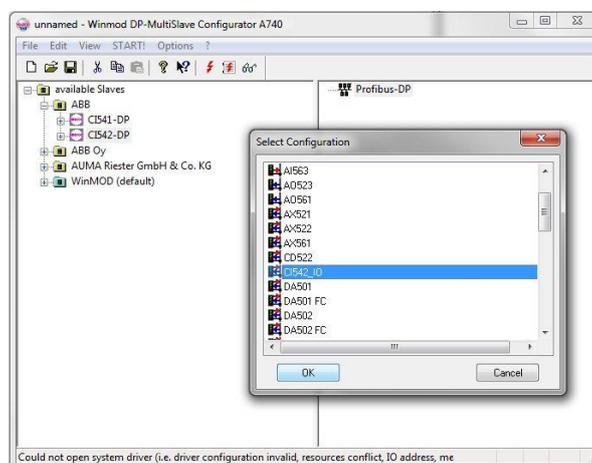


Abbildung 3.1: Einbinden des I/O-Moduls [eigene Darstellung]

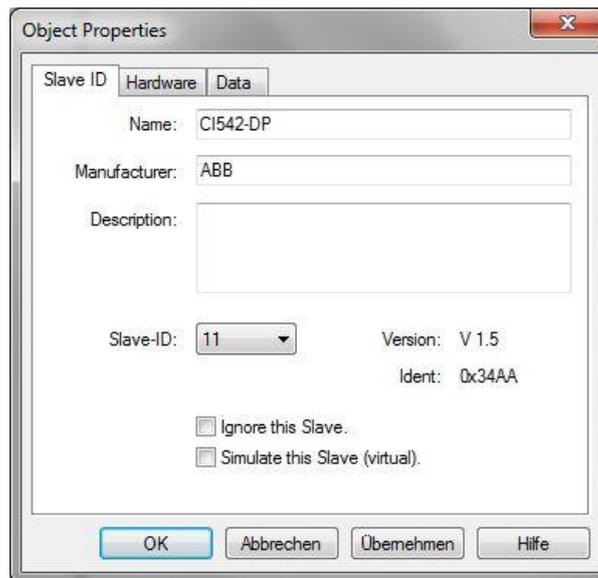


Abbildung 3.2: Einfügen der Stationsadresse [eigene Darstellung]

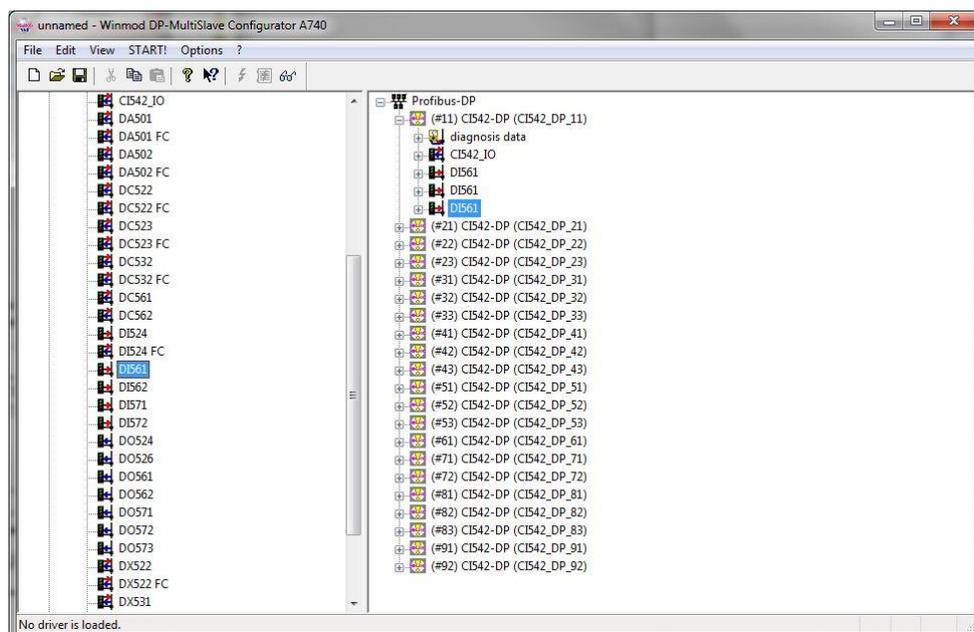


Abbildung 3.3: Nachgestellte Slavestruktur [eigene Darstellung]

3.1.2 Einbinden der Signale als Operanden

Um die Eingangs- und Ausgangssignale der SPS als Operanden in WinMOD zu importieren wird aus der Signalliste der SPS eine Microsoft Excel Tabelle (Engineering Datei) generiert. Beim Import dieser Tabelle werden die Signale Bit-Weise eingelesen. Folgende Formate sind definiert, das binäre 1-Bit Signal, das 8-Bit große Byte, das 16-Bit große Word (entspricht dem verwendeten analogen Format)

und das Doppelwort Format DWord mit 32 Bits. Die Module werden von oben nach unten anhand der Liste belegt. Zusätzlich zu den Prozesssignalen und den Reservesignalen der Ein- sowie Ausgangsmodule muss auch der Signalaufbau der IO-Module der Excel Tabelle hinzugefügt werden [siehe Anhang X], damit sämtliche Slavesignale eindeutig bestimmt sind.

Die Engineering Datei ist in mehrere Tabellenblätter unterteilt. Für den Operanden Import ist lediglich das erste Tabellenblatt vorgesehen, die Tabellenblätter Engineering, Engineering_Sort und Engineering_Energie bilden die Grundlage für den Import der globalen Operanden der Geräte und der Verbindungen von vordefinierten Makro-Dateien (siehe Abschnitt 3.2.2).

Die Signalliste der SPS wurde mit Hilfe der exportierten Daten aus dem Control Builder Plus erstellt. Die I/O Module sind mit internen Signalen belegt, die für den Datenaustausch ebenfalls berücksichtigt werden müssen. Diese hinzugefügten Signale werden einer Signalart zugeordnet und mit ABB_Intern benannt und durchnummeriert. Als Zwischenschritt wird die Signalliste noch in Eingangs- und Ausgangssignale getrennt. Mit Hilfe der Signalzuordnung können die Signale der Slaves getrennt nach Eingang oder Ausgang importiert werden. Zur Benennung der Signale wurden die im CBP genutzten Variablennamen genutzt. Die Signale werden von Oben nach Unten aufgefüllt. Deswegen muss die oben aufgeführte Struktur eingehalten werden um ein einfaches Importieren über den Treiber zu ermöglichen. Anschließend kann mit der Operandenzuordnung begonnen werden. Der grundsätzliche Spaltenaufbau der Signalliste entspricht der Feldzuordnung in WinMOD. Die jeweiligen Spalten dienen der Bestimmung und der zusätzlichen Informationen zu den Operanden.

- Signalart bezeichnet den Datentyp des Operanden, dieser wird mit einer Formel generiert. Bezogen auf die Adresse wird der dritte Buchstabe ausgelesen und einem Datentyp zugewiesen. Unterschieden wird in die bereits erwähnten vier verschiedenen Signalgrößen beziehungsweise Typen, X entspricht 1-Bit und wird mit BOOL deklariert, W wird mit WORD deklariert, B für BYTE und D für DWORD, letztere werden nur für die Bezeichnung der ABB_Intern Variablen gebraucht zur vollständigen Belegung der Slaves.
- Name ist die Benennung des Signals und entspricht der Variable im Hardwaremodul die vom CBP exportiert wird. Die Variable ist die Bezeichnung der Adresse.
- Adresse entspricht der Treiber Adresse des Operanden, diese wird ebenfalls aus dem CBP exportiert und dort automatisch generiert. Das Steuerungsprogramm der SPS greift auf die Adresse zu und kommuniziert über diese mit WinMOD.
- WinMOD-Symbol bezeichnet den Namen des Operanden, dieser wird mit Hilfe der eingefügten AKS, der Adresse und der Adresse des Prozessleitsystems (PLS) generiert. Den Anfang bilden die Buchstaben zwei und drei der Adresse, hieraus lässt sich die Signalart des Operanden ablesen. Der Mittelteil besteht aus dem Anlagenkennzeichen und benennt das Gerät. Zum

Schluss wird zur Differenzierung des Operanden mit gleichen AKS, die in die Liste eingefügte PLS-Adresse angehängt, z.B. Qx_V035SB01PA01PU03_339. Siehe auch Abschnitt 2.2.3 zur Bildung der HAMBURG WASSER Konventionen. Zusätzlich wird die Spalte Zusatzkommentar ausgelesen. Handelt es sich um ein Reservesignal, wird der Inhalt aus der Spalte Aggregat eingefügt.

- Kommentar bildet sich aus der Spalte Aggregat und dem Zusatzkommentar und dient dazu dem Operanden eine Erläuterung beizufügen, zur Erleichterung des Verständnisses.

Ist der Peripherie-Treiber eingerichtet und sind alle Signale der Slavestruktur in der Engineering-Liste aufgenommen, können die Operanden importiert werden. Damit die Signallisten zur Signal- und Operandenzuordnung von WinMOD verarbeitet werden können, müssen diese noch in eine Text Datei umgewandelt werden. Im Anhang X finden Sie die aus dem Tabellenblättern erstellten csv-Dateien. Anschließend kann, wie in den Abbildung 3.4 dargestellt, die jeweilige Datei importiert werden. Das Importformat (Abbildung 3.5) und die folgenden Feldzuweisungen sind wie in Abbildung 3.6 für die Signalzuordnung und für die Operandenzuordnung durchzuführen. Nach erfolgreichen Import der Operanden kann auf diese nun über das Projektfenster (Umschalt+F4) zugegriffen werden. Im Peripherie-Treiber kann die Signalzuordnung der Hardware mit den importierten Operanden nachverfolgt werden (Abbildung 3.7). Die Hardwarekonfiguration und deren Signalzuordnung kann, wenn einmal erstellt, auch exportiert, weiterbearbeitet und wieder importiert werden.

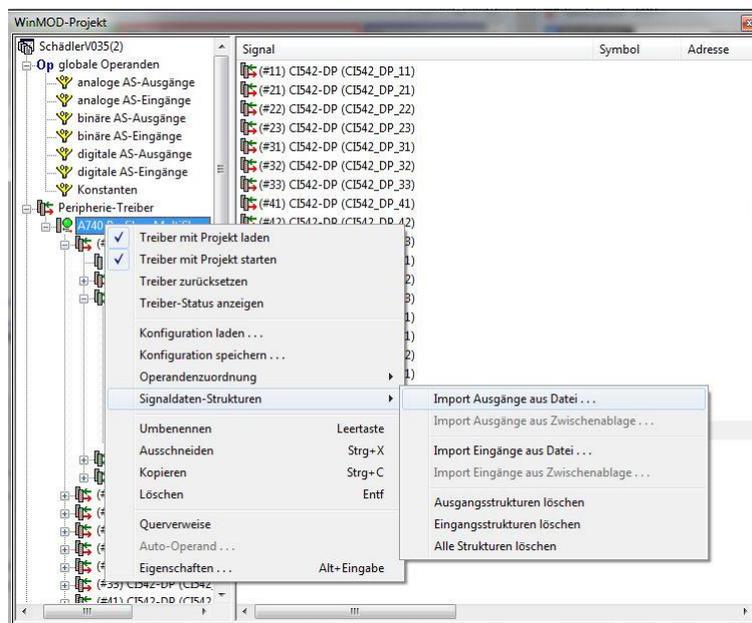


Abbildung 3.4: Dateienimport zur Signaleinbindung [eigene Darstellung]

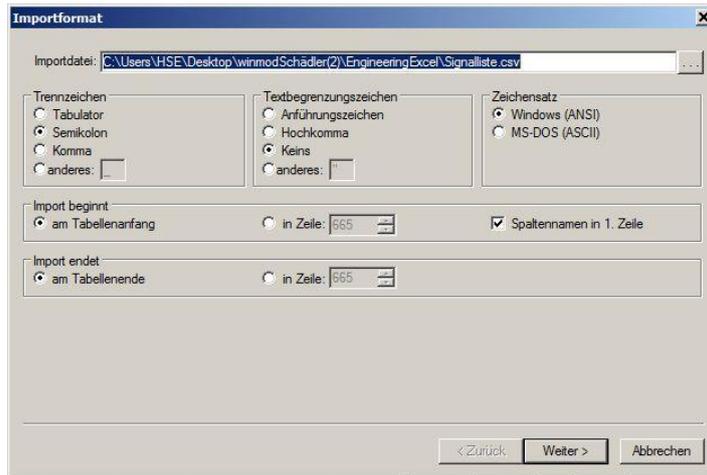
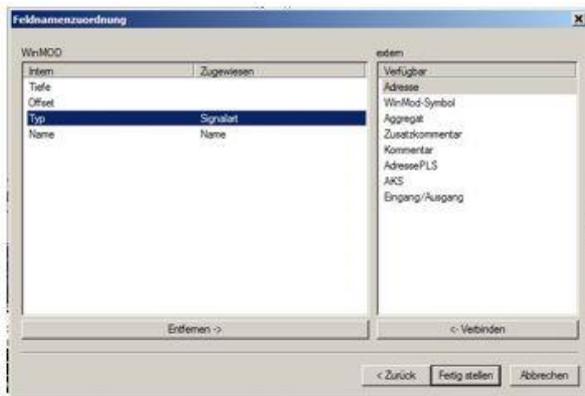
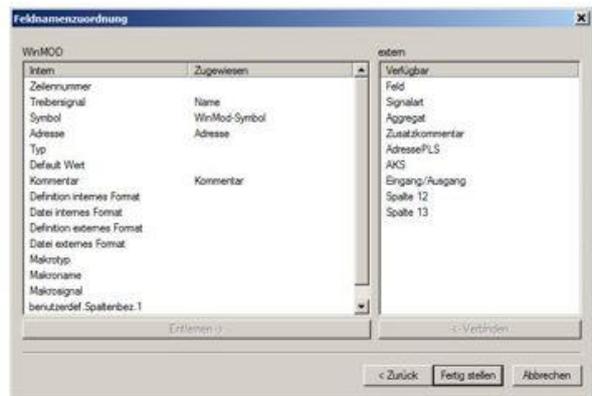


Abbildung 3.5: Importformat zur Signaleinbindung [eigene Darstellung]



(a) Signalzuordnung



(b) Operandenzuordnung

Abbildung 3.6: Feldzuweisungen zur Signaleinbindung [eigene Darstellung]

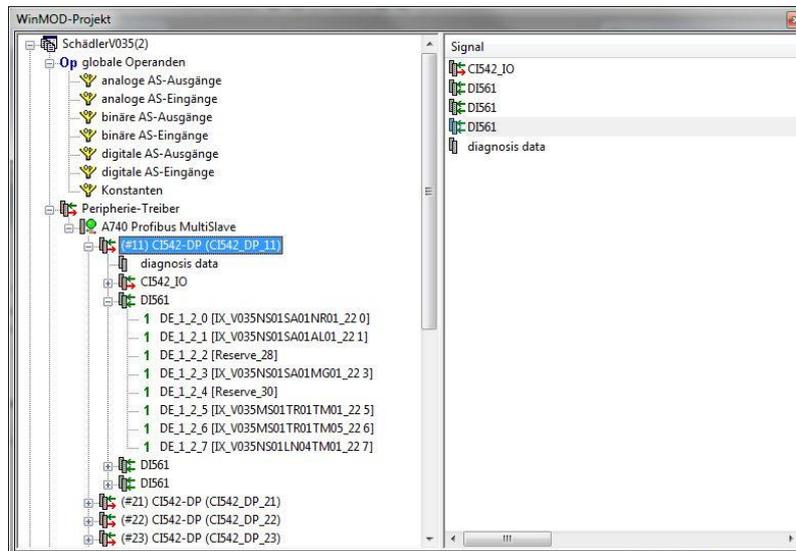


Abbildung 3.7: Signalstruktur einer Eingangs Baugruppe [eigene Darstellung]

3.1.3 Definition der Analogwerte

Zentrale Definitionen bezeichnen den Teil des Projekts in dem die Definitionen der internen sowie externen Analog- und Digitalformate angelegt werden. In diesem Projekt werden lediglich die Analogformate benötigt und definiert.

Die internen Analogformate stammen aus einer vorhandenen Bibliotheken Liste, sowie aus selbst erstellen Formaten, die für Simulation relevant sind. Die Namenstruktur für interne Analogformate ist so gewählt, dass sie möglichst leicht zuzuordnen ist. Beispielhaft hier das interne Analogformat für einen Strom im Wertebereich von null bis 1000 *current[A] 0...1000,00 SI*. Siehe auch Abbildung 3.8. Innerhalb der internen Analogformate wird noch zwischen dem logischen und physikalischen Wertebereich unterschieden. Während der logische Wertebereich immer in Prozent angegeben wird, kann der physikalische Wertebereich je nach Einheit variieren.

interne Analogformate	logisch	physikalisch	Einheit
area [m²] +/- 1.000,00 SI *	-100,00...+100,00 %	-1.000,00...+1.000,00 m²	m²
current [A] 0...1.000,00 SI *	0,00...+100,00 %	0,00...+1.000,00 A	A
current EU [mA] 4...20,00	0,00...+100,00 %	+4,00...+20,00 mA	mA
digits ABB [] 0...27.648	0,00...+100,00 %	0...+27.648	
digits EU [] 0...32.761	0,00...+100,00 %	0...+32.761	
dimensionless [] +/- 1.000,00	-100,00...+100,00 %	-1.000,00...+1.000,00	
dimensionless [] 0,00...100,00 SI	0,00...+100,00 %	0,00...+100,00	
distance [m] +/- 1.000,00 SI *	-100,00...+100,00 %	-1.000,00...+1.000,00 m	m
distance [m] +/- 10,00 SI	-1,00...+1,00 %	-10,00...+10,00 m	m
distance [m] +/- 100,00 m	-100,00...+100,00 %	-100,00...+100,00 m	m
distance [m] +/- 100,00 SI	-1,00...+1,00 %	-10,00...+10,00 m	m
distance [m] +/- 100,00 SI*	0,00...+100,00 %	-10,00...+10,00 m	m
distance [m] +/- 100,00 SI**	0,00...+100,00 %	-100,00...+100,00 m	m
distance [m] 0,00...1.000,00 SI	0,00...+100,00 %	0,00...+1.000,00 m	m
distance [m] 0...100,00 SI	0,00...+10,00 %	0,00...+100,00 m	m
distance [m] 0...100,00 SI*	0,00...+100,00 %	0,00...+100,00 m	m
Drehzahl	0,00...+100,00 %	0,00...+3.000,00 n	n
Prozent [%] +/- 100,00	-100,00...+100,00 %	-100,00...+100,00 %	%
Prozent [%] +/- 100,00 #2	-100,00...+100,00 %	-100,00...+100,00 %	%
Prozent [%] 0...100,00	0,00...+100,00 %	0,00...+100,00 %	%
Prozent [%] 0...100,00 #2	0,00...+100,00 %	0,00...+100,00 %	%
temperature [°C] -20...160,00	+2,53...+4,33 %	-20,00...+160,00 °C	°C
temperature [°C] -273,15...9.726,85	0,00...+100,00 %	-273,15...+9.726,85 °C	°C
temperature [°C] -50...400,00	+2,23...+6,73 %	-50,00...+400,00 °C	°C
temperature [°C] 0...100	+2,73...+3,73 %	0,00...+100,00 °C	°C
temperature [K] 0...10.000,00 SI *	0,00...+100,00 %	0,00...+10.000,00 K	K
temperature EU [°C] -20...160,00	0,00...+100,00 %	-20,00...+160,00 °C	°C
temperature EU [°C] -50...400,00	0,00...+100,00 %	-50,00...+400,00 °C	°C
test25m	0,00...+100,00 %	0,00...+25,00 m	m
testmeter	0,00...+100,00 %	0,00...+10.000,00 m	m
testmeter2	-100,00...+100,00 %	-100,00...+100,00 m	m
testmeter3	0,00...+100,00 %	-12,50...+7,50 m	m
testmeter4	0,00...+100,00 %	-12,50...+1.000,00 m	m
testmeter5	0,00...+100,00 %	-12,80...+12,20 m	m
time [h] 0...100,00 *	0,00...+100,00 %	0,00...+100,00 h	h
time [min] 0...6.000,00	0,00...+100,00 %	0,00...+6.000,00 min	min
time [s] 0...360.000,00 SI	0,00...+100,00 %	0,00...+360.000,00 s	s
velocity [m/s] +/- 1.000,00 SI *	-100,00...+100,00 %	-1.000,00...+1.000,00 m/s	m/s
volume [m³] +/- 1.000,00 SI *	-100,00...+100,00 %	-1.000,00...+1.000,00 m³	m³
volume flow [l/s] +/- 1.000,00*	-100,00...+100,00 %	-1.000,00...+1.000,00 l/s	l/s
volume flow [m³/s] +/- 1.000000	-100,00...+100,00 %	-1.000000...+1.000000 m³/s	m³/s

Abbildung 3.8: Liste der internen Analogformate [eigene Darstellung]

Das externe Analogformat bezieht sich auf die analoge Eingangskarte am Feldbuskoppler (Slave). Zur Verarbeitung der analogen Messgröße durch die SPS muss dieses Signal digitalisiert werden. Die Verarbeitung ist in Abbildung 3.9 erläutert.

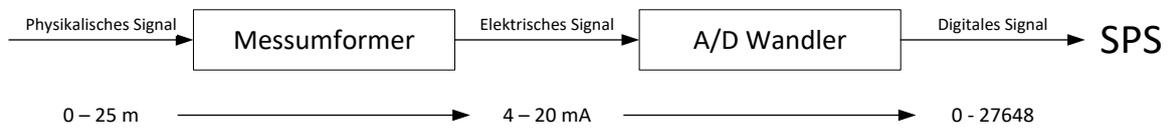


Abbildung 3.9: Verarbeitung eines analogen Signals [eigene Darstellung]

Die Messgröße wird von dem Messwertgeber an das analoge Eingabemodul übergeben und in ein elektrisches Signal, umgewandelt, normiert mit dem jeweiligen physikalischen Wert. Der Analog-Digital-Umsetzer setzt dieses elektrische Signal nun in ein digitales Signal um.

Die genaue Auflösung des eingesetzten AI-Moduls ist im Datenblatt ersichtlich [siehe 14], es handelt sich um eine 12-Bit plus Vorzeichen Übertragung. Es wird nur der Bereich von 4-20 mA genutzt, folglich unipolar und drahtbruchsicher gewählt. Die Analogwerterfassung des Steuerungsprogramms erfolgt nun im 16-Bit-Integer Format, das heißt ein Wertebereich von von -32768...+32767. Unipolare Analog-Baugruppen nutzen tatsächlich nur den Bereich von 0... 27648, 0 entspricht 4 mA und 27648 den 20 mA. Aus der 12-Bit Auflösung ergibt sich nun eine Einteilung von effektiv 3456 Stufen die auf den logischen Wert von 0...100% aufgeteilt werden können. Auf den Unter- sowie Übersteuerungsbereich wurde verzichtet.

Zur Erstellung des externen Analog-Formats in WinMOD muss wie in Abbildung 3.10 dargestellt die Bit-Zuordnung vertauscht werden um den richtigen Wertebereich zu definieren.

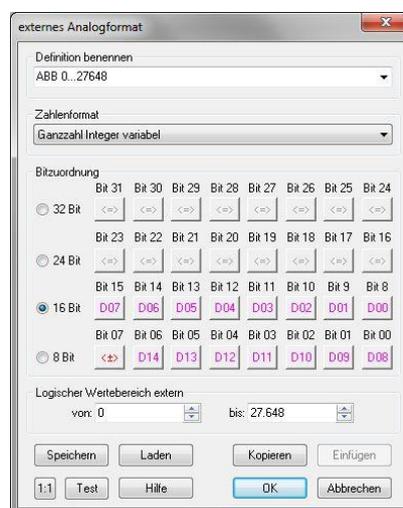


Abbildung 3.10: Bit-Zuordnung des externen Analogformats [eigene Darstellung]

3.2 Erstellung der Simulation

3.2.1 Makros und Makrobibliotheken

In WinMOD können Funktionsbausteine und Funktionsgruppen zu Makros zusammengefasst werden. Diese bestehen aus Verknüpfungen der Simulationselemente der Elementbibliothek und Operanden. In dem Makro werden Ein-, sowie Ausgänge und falls benötigt Konstanten lokal definiert und logisch intern miteinander verknüpft. Extern können diese dann mit globalen Signalen beschaltet werden. Sämtliche Geräte werden simulationsrelevant nachgestellt und als jeweilige Makrodatei gespeichert. Größtenteils sind Betriebsmittel in der Anlage mehrfach vorhanden, das MRB verfügt zum Beispiel über 15 Schieber. Die erstellten Makrodateien können nun einzeln über die Elementarbibliothek in die Simulationsdateien geladen oder in Makrobibliotheken eingefügt werden. Makrobibliotheken müssen erstellt beziehungsweise können auch sofern vorhanden ergänzt werden. Es wird auf bereits erstellte Makrobibliotheken zugegriffen, in diese werden die erstellten Makros je nach Zugehörigkeit gruppiert. Die verschiedenen Bibliotheken sind: *WinMOD Library Allgemein HSE*, *WinMOD Library Antriebe HSE*, *WinMOD Library Energie HSE*, *WinMOD Library Sensoren HSE* und *WinMOD Library Buttons and Display*. In Abbildung 3.11 ist ein Ausschnitt einer Makrobibliothek mit einem Schieber dargestellt.

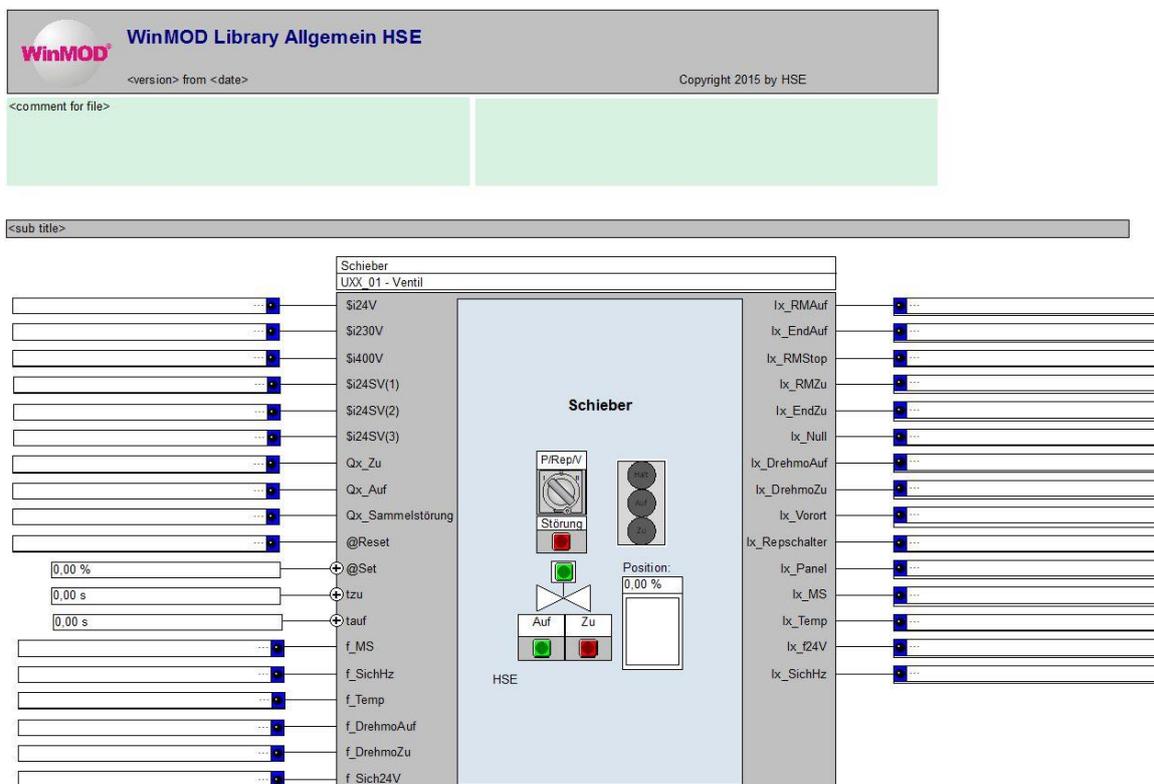


Abbildung 3.11: Ausschnitt aus Makrobibliothek mit Makro Schieber [eigene Darstellung]

Die WinMOD-Bibliotheken gehören zwar zu den Simulationsdateien, dienen aber lediglich nur zur Speicherung der Vorlagen. Die Signale der eingefügten Makros werden nicht belegt. Operanden werden erst in den vorgesehenen Simulationsdateien eingefügt.

3.2.2 Aufbau eines Makros am Beispiel Ventil

Das Makro wird auf Grundlage des Stromlaufplanes erstellt. Die Anlage besitzt insgesamt 15 Schieber, die Teil der Speichorentleerung sowie der Speicherreinigung sind. Zusätzlich verfügt die Anlage über ein Zulaufschütz, das durch Absperren des Zulaufes eine Speicherbegehung ermöglicht. Jede der Hauptentleerpumpen besitzt einen Saug-, Druck- und Spülschieber. Die beiden Restentleerpumpen einen Saug- und Druckschieber. Die Brauchwasserpumpe steuert den Zufluss über drei Absperrschieber. Zur manuellen Reinigung des Überlaufes oder des oberen Speicherbeckens stehen ebenfalls zwei Absperrschieber zur Verfügung. Die Schieber sind mit einem Motor ausgestattet und unterscheiden sich grundsätzlich in der Größe und der Stromaufnahme. Grundsätzliche Aufgabe des Schiebers ist das Öffnen oder Schließen einer Rohrleitung. Der angeschlossene Motor muss in der Lage sein die Drehrichtung zu ändern. Dazu müssen im Drehstrommotor zwei Außenleiter vertauscht werden. Technisch wird dies durch die verriegelte Wende-Schützschaltung realisiert (Abbildung 3.12). In der Simulation wird diese Aufgabe innerhalb des Makros *Schieber* umgesetzt. Mit Hilfe der Elementbibliothek werden interne, lokale Signale miteinander verknüpft. Eine Auflistung und Beschreibung aller Bausteine der Elementbibliothek findet sich im Anhang X, dem WinMOD-Handbuch.

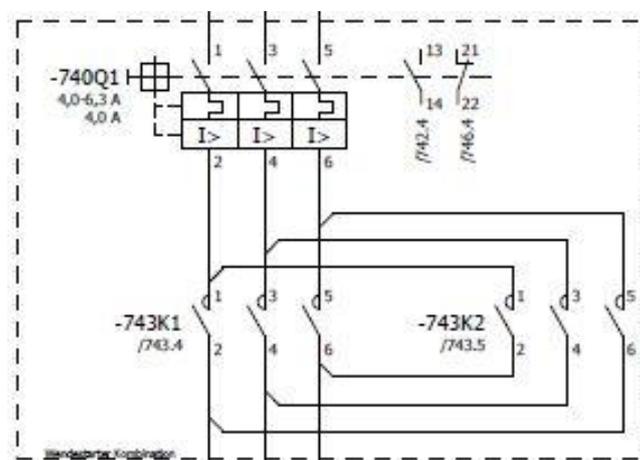


Abbildung 3.12: verriegelte Wende-Schützschaltung [eigene Darstellung]

Um innerhalb eines Makros Signale logisch miteinander zu verknüpfen müssen diese erst einmal angelegt und definiert werden. Möglich wird dies über ein Fenster zur Symbolzuordnung, das sich beim Aufruf des Makros auf der rechten Seite befindet (Abbildung 3.13). Für die lokalen Operanden ist ein Bezeichner, das Symbol, ein Kommentar, ein Datentyp mit Richtung (Prozesssignal/Konstante) und ein Typ (binär/analog/digital) sowie ein Defaultwert anzugeben. Die Analogsignale beziehen sich dabei auf ein in den zentralen Definitionen angelegtes internes Analogformat (siehe Abschnitt 3.1.3).

Symbol	Adresse	Kommentar
f_SichHz		Fehlerprovokation Heizungssicherung ausgelöst
#0 %		
#5,00%		
#95,00%		
#100,00%		
\$i24SV(1)		Spannungsversorgung Steuerungsmodul
\$i24SV(2)		Spannungsversorgung Steuerungsmodul
\$i24SV(3)		Spannungsversorgung Steuerungsmodul
\$i24V		Eingang Spannungsversorgung 24V
\$i230V		Eingang Spannungsversorgung 230V
\$i400V		Eingang Spannungsversorgung 400V
%Set		Arbeitspunkt
@Reset		Reset auf Arbeitspunkt
f_DrehmoAuf		Fehlerprovokation Drehmo öffnen ausgelöst
f_DrehmoZu		Fehlerprovokation Drehmo schließen ausgelöst
f_MS		Fehlerprovokation Motorschutz ausgelöst
f_Sich24V		Fehlerprovokation Steuerspannung ausgelöst
f_Temp		Fehlerprovokation Temperaturlöser
Iw_Motorstrom		Motorstrom
Ix_DrehmoAuf		Eingangssignal Drehmo öffnen ausgelöst
Ix_DrehmoZu		Eingangssignal Drehmo schließen ausgelöst
Ix_EndAuf		Eingangssigna Endlage Auf
Ix_EndZu		Eingangssigna Endlage Zu
Ix_f24V		Eingangssignal Steuerspannung fehlt
Ix_MS		Eingangssignal Motorschutz ausgelöst
Ix_Null		Eingangssignal Null
Ix_Panel		Eingangssignal Panelsteuerung
Ix_Repschalter		Eingangssignal Reperatursschalter
Ix_RMAuf		Eingangssignal Rückmeldung Auf (Vorort)
Ix_RMStop		Eingangssignal Rückmeldung Stop (Vorort)

Abbildung 3.13: Symbolzuordnung des Makros Schieber [eigene Darstellung]

Jedes Geräte-Makro verfügt über eine Aufteilung in das *Control-Panel* und *Behaviour*. Das Verhalten wird zur Übersicht in weitere Bereiche unterteilt. Auf dem *Control-Panel* werden die nach Außen sichtbaren Elemente abgelegt. Über den Reiter *Bearbeiten* lässt sich ein Bedienfeld einfügen und auf die gewünschte Größe anpassen. Diese Elemente können passive sowie auch aktive Elemente sein, welche ihre Farbe ändern, eine Bewegung darstellen oder Schaltvorgänge erlauben. Zur sichtbaren Identifizierung wird der Name des Makros als formatierbarer Text eingefügt. Des Weiteren wird eine von MS Visio erstellte bildliche Darstellung des Geräts als Active Image eingebunden. Die Abbildung 3.14 zeigt das Control Panel des Makros Schieber, mit Bezeichnung sowie bildlicher Darstellung.

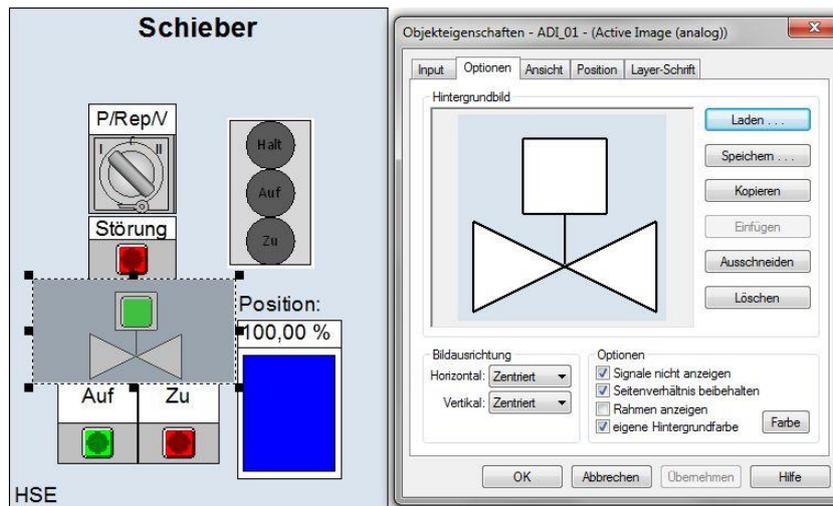


Abbildung 3.14: Einbinden des Active Image [eigene Darstellung]

Um die Position des Schieberblatts anzuzeigen (normiert in Prozent) wird ein Analogsignal, als grafische und numerische Anzeige eingefügt (Anzeige-Modus *Balken von oben*). Ist die Anzeige gefüllt (100%), ist der Schieber geschlossen (Abbildung 3.15). Bei null Prozent ist er komplett geöffnet.

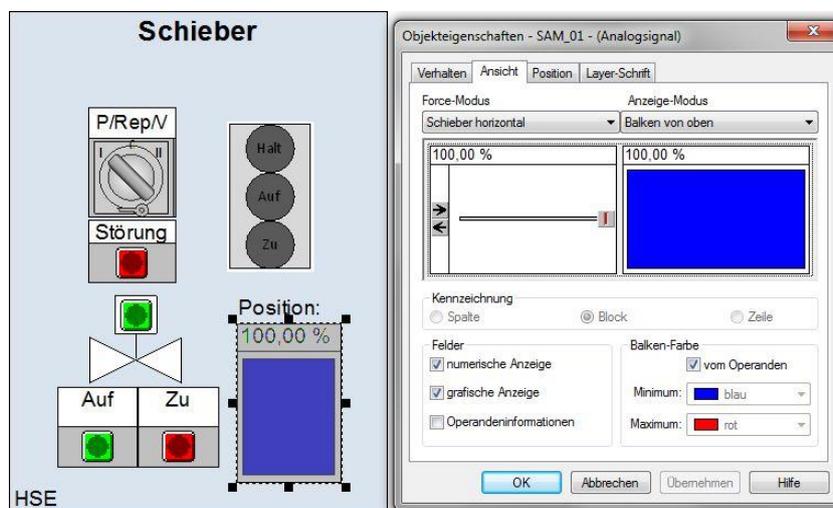


Abbildung 3.15: Einbinden des Analogsignals [eigene Darstellung]

Die verschiedenen Zustände werden über Leuchtaster abgebildet. Angezeigt werden Auf und Zu, eine gemeldete Störung und ob der Motor eingeschaltet ist. Die Abbildung 3.16 verdeutlicht die Konfiguration einer Meldeleuchte.

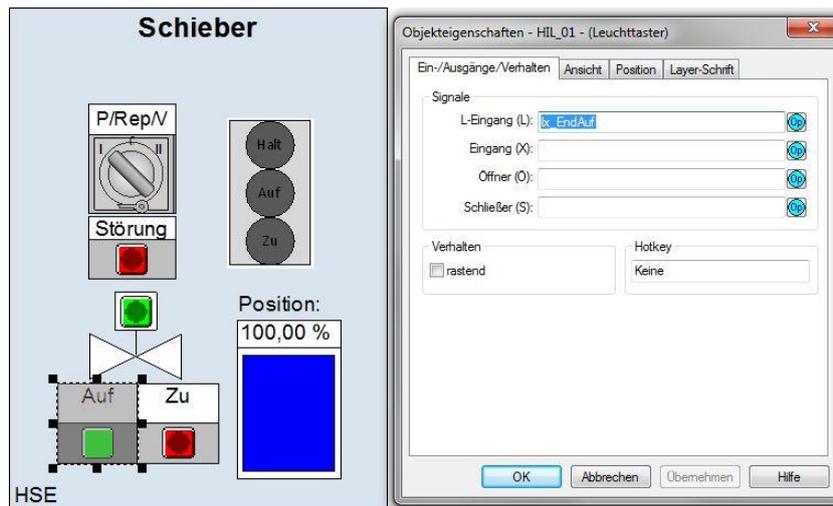


Abbildung 3.16: Einbinden der Meldeleuchte [eigene Darstellung]

Außerdem sind in das Bedienfeld zwei Makros eingebunden. Sie befinden sich in der WinMOD_Library_Buttons_Displays unter *Mehrstufenschalter mit mehreren Schaltflächen*. Der von Mewes & Partner erstellte *Auto-/Hand-Schalter* wird, wie in Abbildung 3.17 verdeutlicht, belegt. Steht der Schalter auf I ist die Panelsteuerung aktiviert. Für den Fernzugriff muss sich der Schalter ebenfalls in der Stellung I befinden. In der 0-Stellung wird der Reparaturschalter aktiviert. Damit wird der Stromkreis, für Arbeiten am Schieber bzw. Schieberantrieb, unterbrochen. Das Eingangssignal für die Nullstellung ist Hardwaretechnisch nicht auf die SPS geführt, wird logisch in der SPS gebildet, wenn weder die Panelsteuerung oder die auf Stellung II liegende Vorort-SPS aktiviert ist.

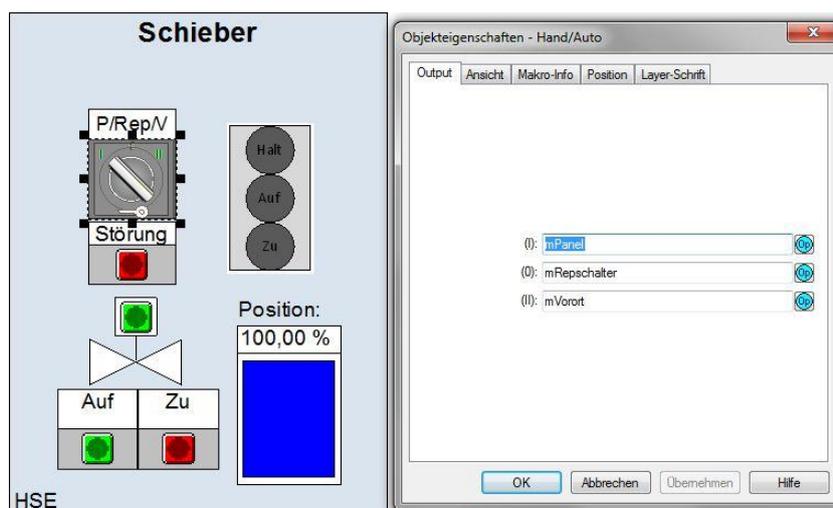


Abbildung 3.17: Konfiguration des Auto-/Hand-Schalters [eigene Darstellung]

Wird die Vorort-SPS aktiviert ist die Bedienung über den *EinschalterVorortVentil*-Schalter möglich. Hierbei handelt es sich um ein selbsterstelltes Schalter-Makro, optisch angelehnt an den realen Bedienschalter. Über den Einschalter werden die Eingangssignale für Auf, Zu und Halt an die SPS gesendet. Diese wiederum sendet das jeweilige Ausgangssignal zur Schiebersteuerung. Der *EinschalterVorortVentil*-Schalter wird mit Hilfe eines Controller Elements der *State Machine* und den zugehörigen *State Elements* realisiert (Abbildung 3.18). Über manuelle Signalschalter werden die State Elements angesprochen, diese befinden sich im selben Netz wie der State Controller und besitzen verschiedene Schrittnummern. Wird eine Schrittnummer betätigt, wird über den State Controller auf das digitale *Active Image* zugegriffen und das hinterlegte Bild der Schrittnummer geschaltet. Die Schrittnummern eins bis drei benötigen eine eingeschaltete Vorortsteuerung. Fehlt dieses Signal verbleibt die Schrittkette im Ausgangszustand (Schritt 0).

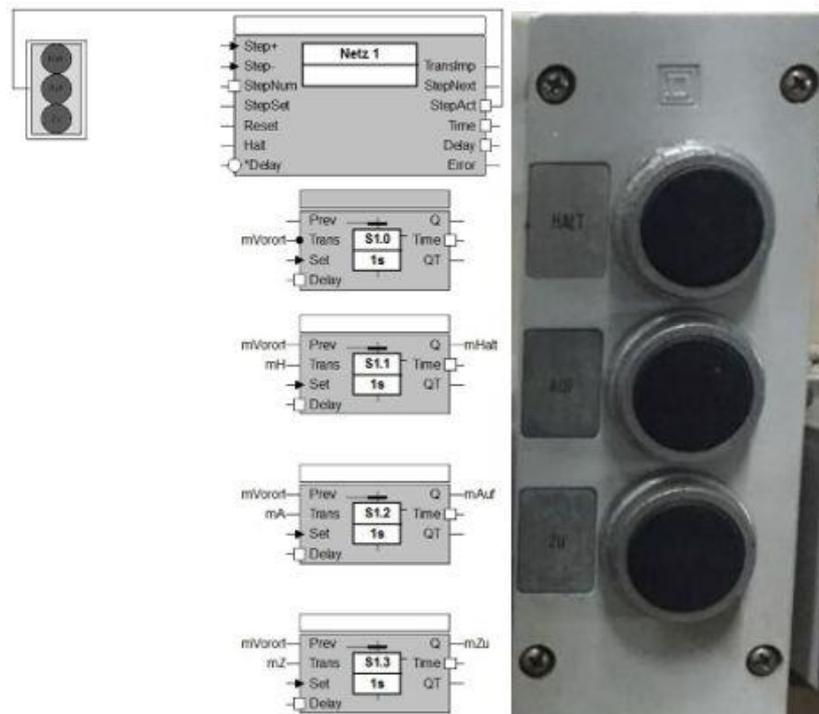


Abbildung 3.18: Aufbau des EinschalterVorortVentil mit realen Vorbild [eigene Darstellung]

Im Teil des Verhaltens (*Behaviour*) ist die logische Verschaltung der lokalen Operanden untergebracht. Die Eingangssignale befinden sich auf der linken, die Ausgangssignale auf der rechten Seite. Das Verhalten des Makros wird zur Übersicht in verschiedene Bereiche unterteilt (Abbildung 3.19).

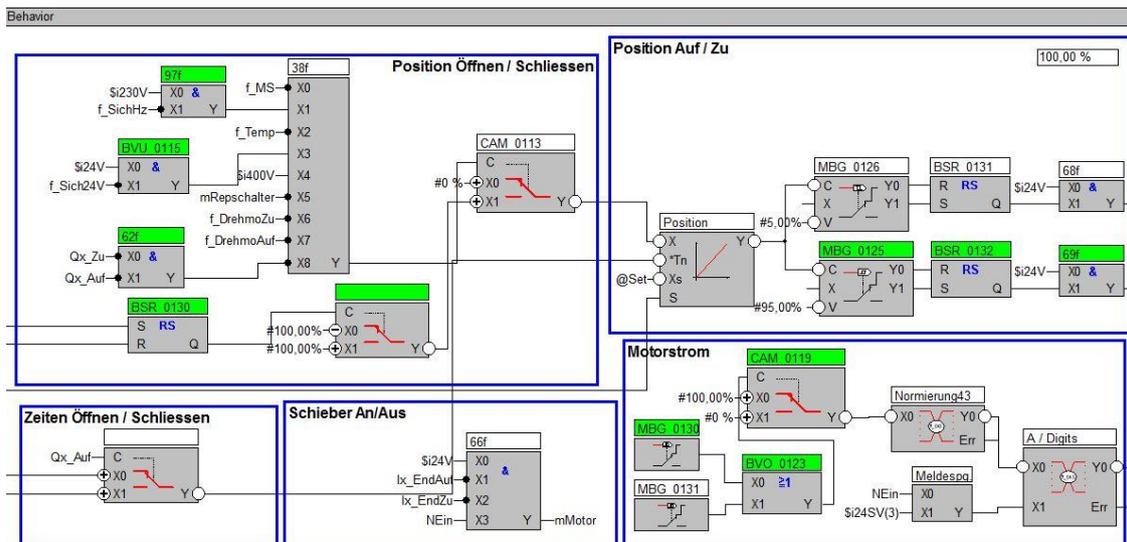


Abbildung 3.19: Übersicht des Makro-Verhaltens [eigene Darstellung]

Position Öffnen/Schließen beinhaltet den Signaleingang und die interne Störungsabfrage. Über den setzdominanten *RS-Flipflop* Baustein gelangt das Signal in den *Analogmultiplexer*. Ausgegeben wird hier je nach ankommenden Signalzustand das verschaltete Analogsignal, in diesem Fall 100% bei eingeschaltetem *RS-Flipflop* und -100% bei nicht vorhandenem Eingangssignal. Kommt es zu einer Störung des Schiebers wird im nachgeschalteten Multiplexer der Analogwert von 0% durchgeschaltet. Ausgehend von einem linearen Bewegungsverlauf des Schiebers wird dieser geöffnet, bei positivem beziehungsweise geschlossen, bei negativen analogem Eingangssignal. Kommt es zu einer Störung oder wird der Schieber nicht mehr angesteuert, stoppt der Schieber an der vorhandenen Position. Über den *@Reset* Operanden lässt sich die Position auf den in *@Set* festgesetzten Wert zurücksetzen. Das binäre Prozesssignal zum Zurücksetzen (*@Reset*) kann in der Projektoberfläche im SimControl-Fenster gesetzt werden. Das analoge Prozesssignal *@Set* findet sich in der Simulationsdatei zur Parametereinstellung der Geräte. Ebenso kann dort die Zeit zum Schließen oder Öffnen des Schiebers separat angegeben werden. Die zeitlich abhängige Position kann im Bedienfeld abgelesen werden. Diese wird in den beiden Grenzschaltern weiterverarbeitet. Die Endlagen sind mit einer Toleranz von fünf Prozent innerhalb des Fahrweges ausgegeben, um eventuelle Beschädigungen am Antrieb sowie Schieber zu verhindern. Der Motorstrom wird vereinfacht ohne Anlaufzeit dargestellt. Befindet sich der Schieber nicht in einer der Endpositionen, null oder 100 Prozent, wird der Nennstrom ausgegeben. Über den *FormulaX*-Baustein kann mit Hilfe des Prozesssignals %ISchieber das Signal auf den Nennstrom normiert werden (Abbildung 3.20). Im nächsten Schritt wird das analoge Stromsignal auf Digits gewandelt. Gleichzeitig wird geprüft, ob der Schieber in Bewegung und die Spannungsversorgung des Analog-Moduls gewährleistet ist (Abbildung 3.21).

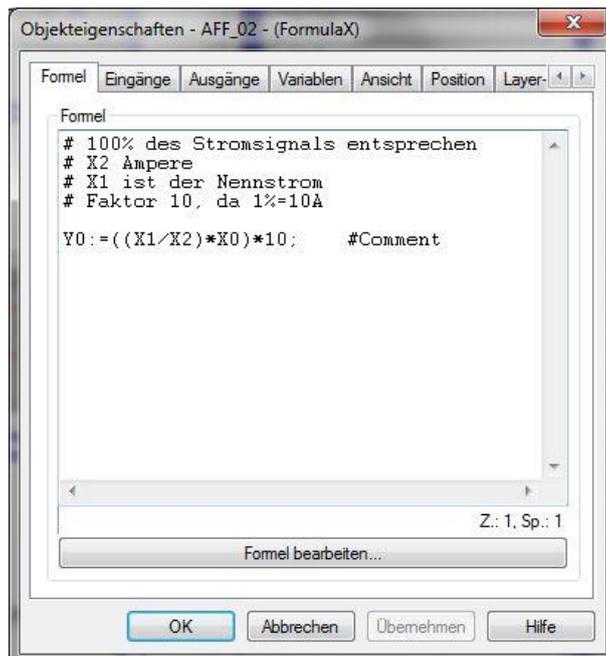


Abbildung 3.20: Nennstromnormierung im Schieber Makro [eigene Darstellung]

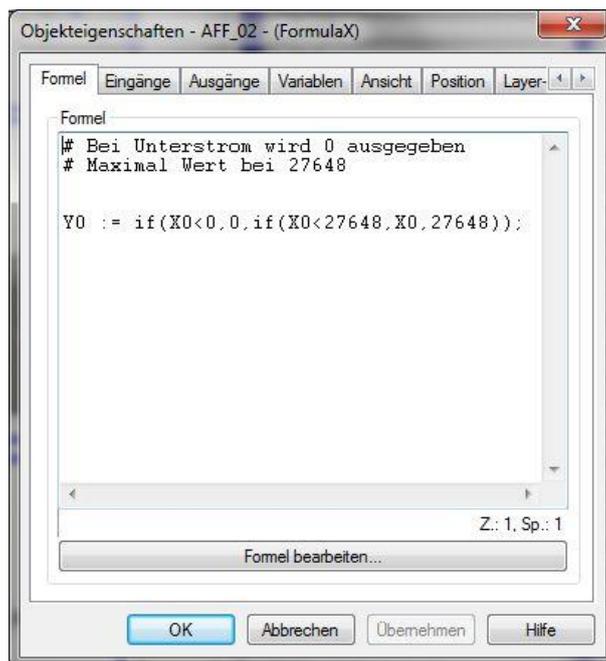


Abbildung 3.21: Digitalisierung des Stromsignals im Schieber Makro [eigene Darstellung]

Der Bereich *der Spannungsversorgung der Meldungen* (Abbildung 3.22) umfasst alle Eingangssignale zur SPS. Die Meldespannung bezieht sich auf die angeschlossene Feldperipherie, in diesem Fall die binären Ein- und Ausgangsmodule sowie das analoge Eingangsmodul und deren Spannungsversorgung. Die Meldespannung ist über die USV abgesichert. Einzelne Fehlerprovokationen der drei Modulspannungen $\$i24SV(1)$, $\$i24SV(2)$, $\$i24SV(3)$ können über das Makro

Spannungsversorgungmodule simuliert werden. Die Fehlerprovokation muss ebenfalls auf anliegende Meldespannung überprüft werden, damit das Signal verarbeitet werden kann. Besonderheit bei den möglichen aufkommenden Störungen des Schiebers sind neben dem Temperaturschalter (f_Temp), die Auslösung des Motorschutzes (f_MS) und die in dem Makro verarbeitete fehlende Steuerspannung (f_Sich24V). Die beiden Drehmo-Schalter zur Laufrichtung lösen aus sobald die Leistungsaufnahme einen bestimmten Punkt überschreitet. Abbildung 3.23 zeigt den kompletten Makro Aufbau des Schieber-Makros.

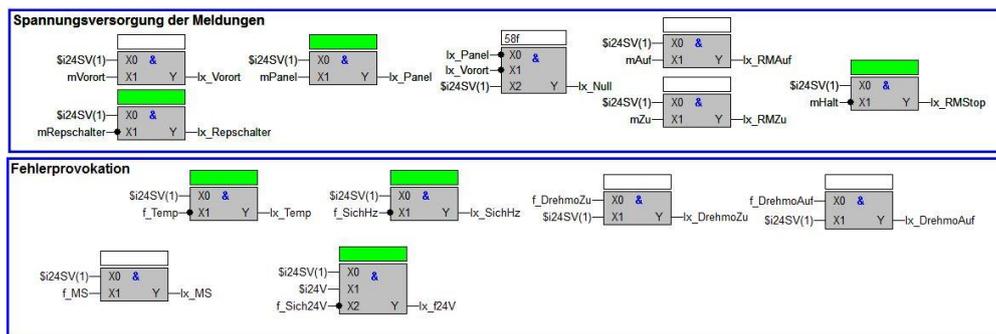


Abbildung 3.22: Spannungversorgung und Fehlerprovokation im Makro [eigene Darstellung]

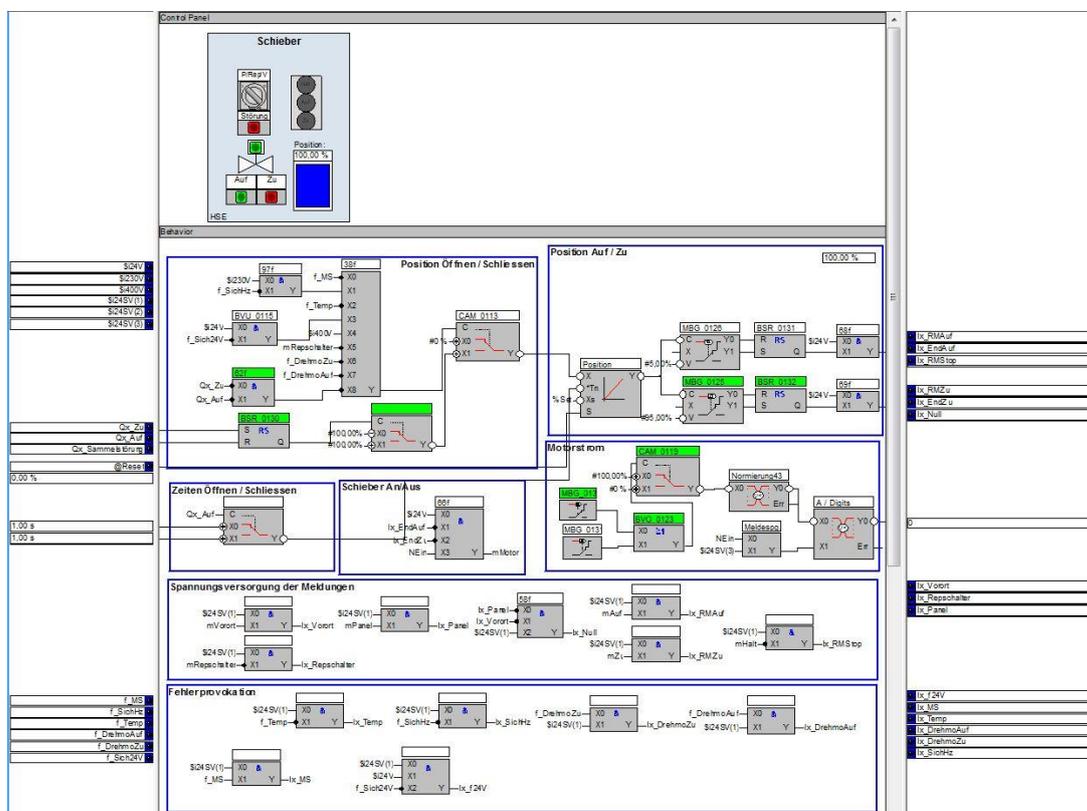


Abbildung 3.23: Aufbau des Makros Schieber [eigene Darstellung]

3.2.3 Einbinden der Geräte mit der Engineering Assistenz

Mit Hilfe der *Engineering Assistenz* wird das automatische Erzeugen der Komponenten und deren Verbindungen in eine Simulationsdatei realisiert. Die *Engineering Assistenz* nutzt dafür die Daten aus der *Engineering-Liste*. Anschaulich dargestellt in Abbildung 3.24 ist die Arbeitsweise der Assistenz.

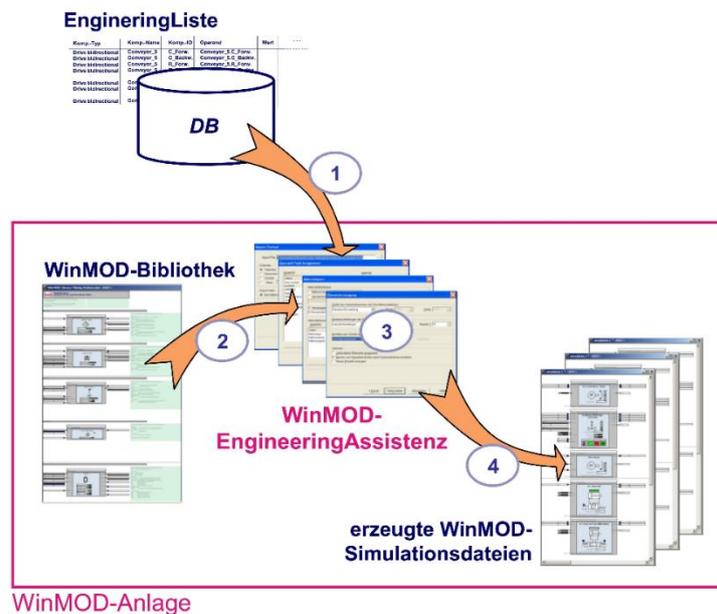


Abbildung 3.24: Arbeitsweise der (vorwärts) Engineering Assistenz [15, S.19]

Als erster Schritt (Punkt 1) greift das Tool auf die angelegte Engineering-Liste zu, mit entsprechenden Informationen zu dem vorgeschalteten WinMOD-Symbol, dem Makronamen, Makrotypen, sowie dem Makrosignal und dem Namen der Simulationsdatei in welcher die Daten geschrieben und die Makros erstellt werden. Außerdem werden hier die Analogsignale einem internen, für die interne Verarbeitung und Darstellung, und einem externen, für die Verarbeitung in dem Steuerungsprogramm, Analogformat (siehe Abschnitt 3.1.3) zugeordnet.

Zusätzlich bedient sich die Engineering Assistenz den in Punkt 2 dargestellten, angelegten Makrobibliotheken, in denen die erstellten Makros abgelegt sind (Abschnitt 3.2.1 und 3.2.2). Dafür müssen die Bezeichnungen des Makrotyps und Makrosignals in der Bibliothek als auch in der Engineering-Liste zwingend übereinstimmen.

Der dritte Punkt beschreibt nun die Oberfläche der *WinMOD-Engineering Assistenz*. Hier werden die Einstellungen für den Zugriff auf die in Punkt 1 und 2 beschriebenen Dateien festgelegt. Abschließend werden die in Punkt 4 gezeigten, Simulationsdateien mit den Makros gefüllt.

In der Simulation des MRBs werden grundsätzlich alle für den Ablauf der Anlagensimulation benötigten Signale mit Hilfe von Makros verarbeitet. Diese nach Stromlaufplänen erstellten Makros stellen die Geräte der Anlage dar und sind in den Makrobibliotheken zur weiteren Verarbeitung durch die *Engineering Assistenz* abgelegt. Für die Engineering-Liste werden die in Abschnitt 3.1.2 erwähnten Tabellenblätter Engineering, EngineeringSort und Engineering-Energie ebenfalls genutzt. Die für den Operanden-Import genutzte Signalliste wird zur Erstellung der Tabelle *Engineering* genutzt. Dazu werden aus der Signalliste per Formel die nicht benötigten Reservesignale entfernt, als leere Zeile ausgegeben und schließlich die Leerzeilen per Filter nicht mehr angezeigt. Zur weiteren Verarbeitung wird der Inhalt der Zeilen und Spalten in die Liste *EngineeringSort* kopiert und ergänzt oder ersetzt. Die *EngineeringSort* Liste wird wie folgt unterteilt.

- Signaltyp beschreibt wie in der Signalliste die Signalart. Der Signaltyp kann nun aber direkt als Eingangs- oder Ausgangssignal definiert werden. Dafür werden die ersten beiden Buchstaben des WinMOD-Symbols ausgelesen und damit der Signaltyp bestimmt. Handelt es sich beim zweiten Buchstaben um ein X wird ein B für binäres Signal geschrieben. A für Analogsignal wird ausgegeben, wenn es sich beim zweiten Buchstaben um ein W handelt. Falls es sich um ein Eingangssignal handelt, wird dem B beziehungsweise A ein I angehängt. Andernfalls ein O für Ausgangssignal. Hierfür wird der erste Buchstabe ausgelesen. Für die Simulation werden nur binäre und analoge Eingangs- sowie binäre Ausgangssignale benötigt. Die Unterscheidung I und O dient dem Einlesen und der Zuordnung des Signals in der Simulation. Die Benennung geschieht über das WinMOD-Symbol.
- *WinMOD-Symbol* bezeichnet den Namen des Operanden. Dieser ist aus dem Signaltyp, dem AKS und dem Signalnamen des Fernwirksystems zusammengesetzt. Den Anfang bilden die Buchstaben zwei und drei der Adresse. Hieraus lässt sich die Signalart des Operanden ablesen. Der Mittelteil besteht aus dem Anlagenkennzeichen und benennt das Gerät. Zum Schluss wird zur Differenzierung des Operanden mit gleichen AKS, der in die Liste eingefügte Signalname des Fernwerksystems angehängt, z.B. Qx_V035SB01PA01PU03_339. Siehe auch Abschnitt 2.2.3 zur Bildung der HAMBURG WASSER Konventionen.
- *Makroname* bezeichnet ein Gerät oder Gerätekomponente die einem Makrotyp zugeordnet wird. Der Makroname wird im Allgemeinen über das AKS definiert und benannt. Die auf das Makro verschalteten Signale beinhalten üblicherweise die gleiche Namenssyntax. Ausnahmen können integrierte Geräte oder Gerätekomponente in erstellte Makros bilden. So

sind das Magnetventil zur Schmiermittelversorgung und die Fettpumpe (z.B. V035SB01PA01VN01, V035SB01PA01FP01) der Hauptentleerungspumpen (V035SB01PA01PU01) in ein Makro integriert.

- *Makrotyp* ist die Typenbezeichnung des Makros und bezieht sich auf das in der Makrobibliothek abgelegte Makro. Die beiden baugleichen Hauptentleerungspumpen besitzen den gleichen Makrotyp Förderpumpe_Typ2 oder auch die Restentleerungspumpen den Typ Förderpumpe_Typ3.
- *Makrosignal* ist die Zuordnung des Signals auf die nach außen sichtbaren leeren Signale, die intern mit dem Makro verschaltet werden. Zum Beispiel wird das Ausgangssignal zum Öffnen des Saugschiebers der ersten Hauptentleerungspumpe (QX_V035SB01PA01SI11_47 1) mit dem Makrosignal Qx_Auf verschaltet. Beim Ansteuern des Signals durch die SPS wird gleichzeitig Qx_Auf und damit das Makro Schieber angesteuert und simuliert.
- *Adresse* ist äquivalent zum Operanden-Import die Adresse des Signals. Diese entspricht der Treiber Adresse des Operanden.
- *Kommentar* dient der Erläuterung des Operanden zur Erleichterung des Verständnisses
- *WinMOD Datei Name* gibt den Namen der Simulationsdatei an in welcher die Engineering Assistenz die Makros aus der Engineering-Liste und der Makrobibliothek ablegt. Die Simulationsdateien werden vorher erstellt und vor dem hinzufügen der Makros leer in das Projekt geladen. Sämtliche Signale und Funktionen die sich in der Simulationsdatei befinden werden beim Einfügen über die *Engineering Assistenz* überschrieben.
- *Internes Signal* beschreibt das interne Analogformat mit dem WinMOD das Signal verarbeitet. Das interne Signalformat (*digits ABB [] 0...27468*) entspricht dem bereits definiertem externen Signalformat (siehe Abschnitt 3.2.3).
- Externes Signal entspricht dem zur analogen Signalverarbeitung nötigen Format (siehe Abschnitt 3.2.3). In diesem Fall *ABB 0...27468*.

Wie bereits beim *Operanden_Import* wird aus der *EngineeringSort* Datei eine csv-Datei erstellt. Über den Menüreiter *Datei / Engineering Assistenz aus Datei* wird der Import gestartet (Abbildung 3.25).

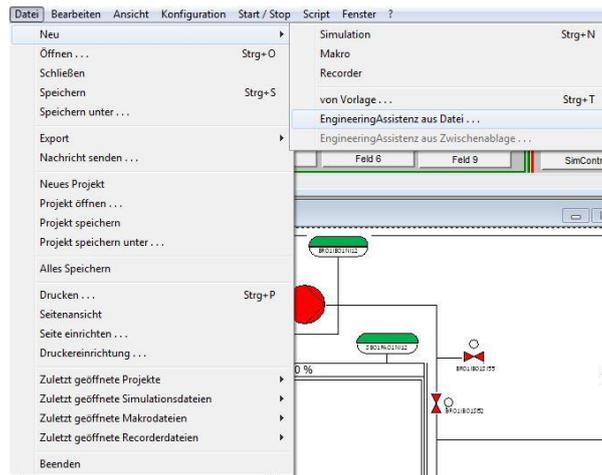
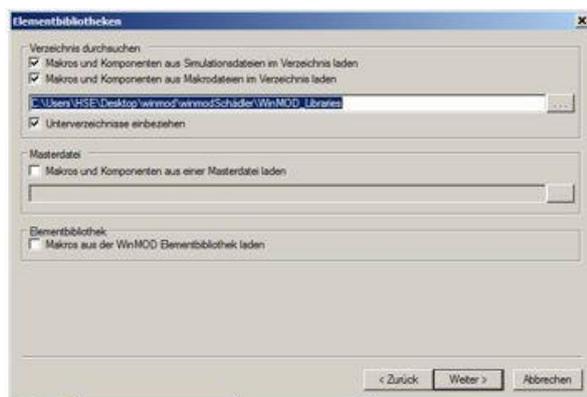


Abbildung 3.25: Makro Import aus Datei [eigene Darstellung]

Es erscheint ein Auswahlfeld aus dem die erstellte csv-Datei ausgewählt wird. Die Operandenfeld-Zuordnung benötigt nur eine Angabe des WinMOD-Symbols, da dieses verschaltet wird. Falls noch nicht bei der Operandenzuordnung der Signale geschehen, können hier die weiteren Operandeninformationen angegeben werden. Die restlichen Spalten werden bei der Makrofeldzuweisung zugeordnet. Infolge dessen sind die Elementbibliotheken wie in Abbildung 3.26 (a) und die Elementerzeugung wie in Abbildung 3.26 (b) durchzuführen. Die Elementbibliotheken geben das Verzeichnis an in dem die Makrobibliotheken abgelegt sind.



(a) Elementbibliotheken



(b) Elementerzeugung

Abbildung 3.26: Makro Elementbibliotheken und Elementerzeugung [eigene Darstellung]

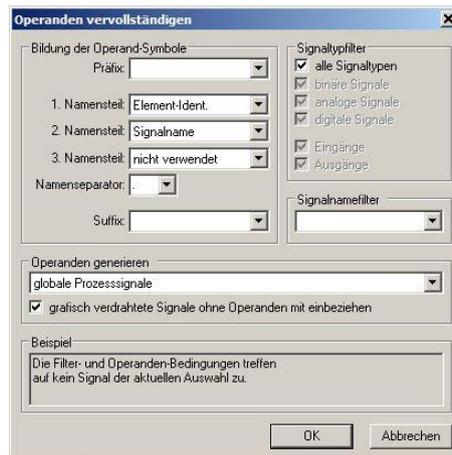


Abbildung 3.27: Operanden vervollständigen [eigene Darstellung]

Die Abbildung 3.27 zeigt den letzten Schritt zur Vervollständigung der Engineering Assistenz, das Fenster *Operanden vervollständigen*. Über die Option *Operanden vervollständigen* werden die nicht durch die *EngineeringSort* belegten von außen sichtbaren Signale durch automatisch generierte Operanden belegt. Diese werden mit dem Makronamen und als Suffix mit dem nach außen verschalteten internen Signal gebildet z.B. die Motorschutzauslösung der ersten Hauptentleerungspumpe V035SB01PA01PU01.f_MS. Diese Signale dienen dem Ablauf der Simulation als Verbindung der Simulationsdateien und der Makros. Hierzu werden einzelne automatisch erzeugte Operanden an Makros ersetzt und andere der durch *Operanden vervollständigen* simulationsinternen Signale eingefügt. Dies geschieht vor allem bei der allgemeinen Energieversorgung der Geräte, die über Spannungssignale realisiert wird. Aus Gründen der Übersicht wurde deswegen das Tabellenblatt Engineering-Energie erstellt und zwei Simulationsdateien für den Geräte-Import erstellt. Die Datei *Geräte* beinhaltet sämtliche für den physikalischen Simulationsablauf nötigen Geräte, Pumpen, Schieber, Lüfter, Luftklappen sowie Messeinrichtungen. Die zweite Datei *Geräte-Energie* enthält die für die Energieversorgung notwendigen Last- und Leistungsschalter, Sicherungen, Transformatoren sowie den Über-/ Unterspannungsschutz. Die Energieversorgung der Geräte erfolgt über den Transformator und wird mit 400V eingespeist. Einzelne Felder werden über den jeweiligen Lastschalter versorgt, wobei die Leitung die Spannungsversorgung aller im Feld befindlichen Geräte gewährleistet. In diesem Fall besitzen alle Anschlussgeräte des Feldes dasselbe Eingangssignal der 400V Spannung. Die in Abbildung 3.28 dargestellte 400 Volt Spannungsversorgung von Feld 4 verdeutlicht diesen Umstand.

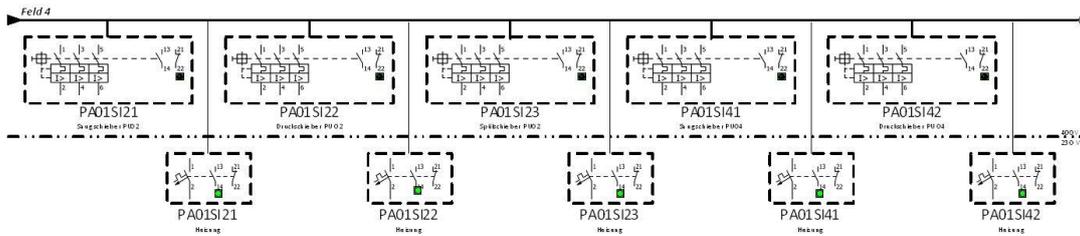


Abbildung 3.28: 400V Spannungsversorgung von Feld 4 [eigene Darstellung]

Das Austauschen der Signale kann entweder in der WinMOD-Simulation selbst, durch Austausch der Operanden, erfolgen oder aber über die Engineering-Liste. Allerdings müssten bei der direkten Anpassung der Engineering-Liste gleichzeitig sämtliche automatische erzeugten Operanden in die Liste aufgenommen werden. Neben der *vorwärts Engineering Assistenz* ist auch die *rückwärts Engineering Assistenz* möglich. Alle verknüpften Signale werden exportiert und analog der Zuordnung zum Import in eine csv-Datei geladen.

Die Engineering Assistenz ermöglicht ein einfaches Bearbeiten und Anpassen der Operanden innerhalb der Simulation oder über eine externe Modifikation der Engineering-Liste. Siehe dazu Abbildung 3.29 Importe, Exporte und Modifikationen durch die Engineering Assistenzen.

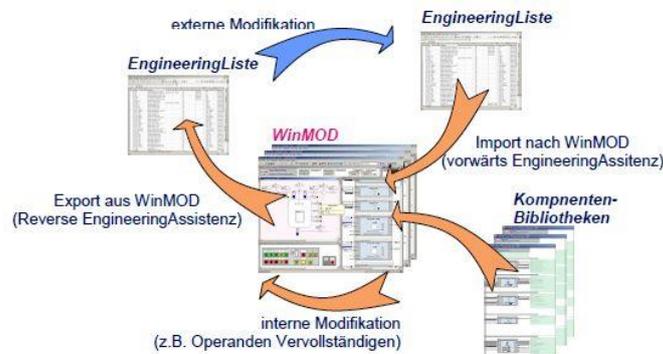


Abbildung 3.29: Importe, Exporte und Modifikationen durch die Engineering Assistenz [6, S. 1-1]

3.3 Erstellen des physikalischen Modells

Die Entwässerung des Speicherbeckens wird über die verschiedenen Niveaus gesteuert. Ein gleichzeitiger Zu- und Abfluss ist nicht möglich. Im Vordergrund steht die Überprüfung des Steuerungsprogramms sowie dessen Optimierung. Auf ein Durchflussmodell wurde verzichtet. Stattdessen wird für den Start der Simulation ein Wert (variabel einstellbar) des Speicherbecken-Pegels

vorgegeben. Die Pumpen laufen mit konstanter Drehzahl. In diesem Zusammenhang wurde die Abnahme des Pegels als linear angenommen.

Die relevanten Pegel befinden sich im Speicherbecken, im Trennbauwerk und im Brauchwasserbecken. Steigt der Pegel im Trennbauwerk über 6,80 m Normal Null (mNN) wird die Entwässerung unterbrochen. Der Brauchwasserpegel ist relevant für die Sohlenreinigung. Bei Störungen der Brauchwasserpumpe oder Zulaufventils wird die Sohlenreinigung unterbrochen oder ausgesetzt.

Zur Entleerung des Speicherbeckens wird überprüft ob eine der Pumpen läuft. Dazu muss die Drehzahl über 150 Umdrehungen pro Minute liegen. Ebenfalls ist ein offener Druckschieber erforderlich. Abbildung 3.30 zeigt die Entleerungsbedingungen. Neben den beiden Hauptentleerungspumpen ermöglichen auch die beiden Restentleerungspumpen die Entleerung.

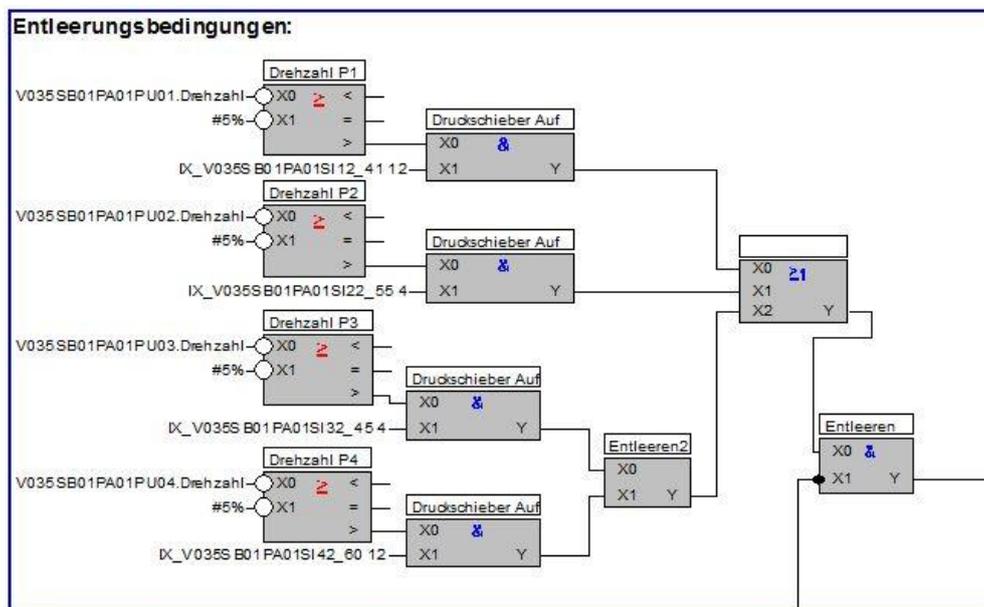


Abbildung 3.30: Entleerungsbedingungen [eigene Darstellung]

Das automatische Befüllen (Abbildung 3.31) beendet den Entleerungsvorgang und lässt den Pegel steigen. Beim Überschreiten des Überlaufpegels von 7,25 mNN füllt sich das Speicherbecken schneller als es sich entleeren würde, beziehungsweise bei Erreichen eines Pegels von 6,80 mNN wird die Entleerung unterbrochen (siehe Abschnitt 4.1.1). Im unteren Bereich der Abbildung 3.31 befindet sich der logische Aufbau zur Simulation des Trennbauwerk Pegels im Mischwassersiel. Zur Vereinfachung wird ein linearer Anstieg des Pegels angenommen und lässt sich über das globale Signal @SetPegel auf null setzen. Die Zeit zum Erreichen des änderbaren Niveaus ist variabel und hier auf 10 Sekunden bestimmt. Der Pegel zum Überlauf in das Speicherbecken @ÜberlaufTB ist einstellbar zwischen 7,25 mNN und 7,80 mNN. In der Simulation gesetzt auf 7,25 mNN. Der Testpegel @TestpegelTB lässt sich

sinnvoll mit zwei Werten bestimmen. Einen Wert oberhalb des gesetzten Überlaufs. Dabei steigt der Pegel im Speicherbecken. Zum anderen einen Wert oberhalb der 6,80 mNN Grenze. Dieser sorgt dafür, dass die Entleerung erst nach Ablauf einer festgelegten Zeit unterbrochen wird. Um lediglich die Unterbrechung der Entwässerung zu testen, ohne dass durch den steigenden Pegel die Messgeräte im Speicherbecken angesprochen werden, wird der Wert auf 6,85 mNN gesetzt. Die genannten Pegel beziehen sich auf die in der gesamten Anlage benannte Nulllinie. Hierbei liegt der für die Messeinrichtung relevante Ursprung bei 6,30 mNN. Diese werden im Testmodell abgezogen um innerhalb des Messbereiches des Analogwertes von 0-3 m zu bleiben. Innerhalb der Formelblocks (Abbildung 3.32) werden die Eingangssignale auf das interne Format normiert. Dabei entspricht ein Prozent einem Meter. Liegt der Testpegel oberhalb des Überlaufs wird über den Analogwertvergleich ein Befüllen des Speichers ausgelöst. Der Betrieb der Brauchwasserpumpe wirkt sich ebenfalls auf den Pegel im Speicherbecken aus, da während der Sohlenreinigung das Speicherbecken mit Brauchwasser befüllt wird.

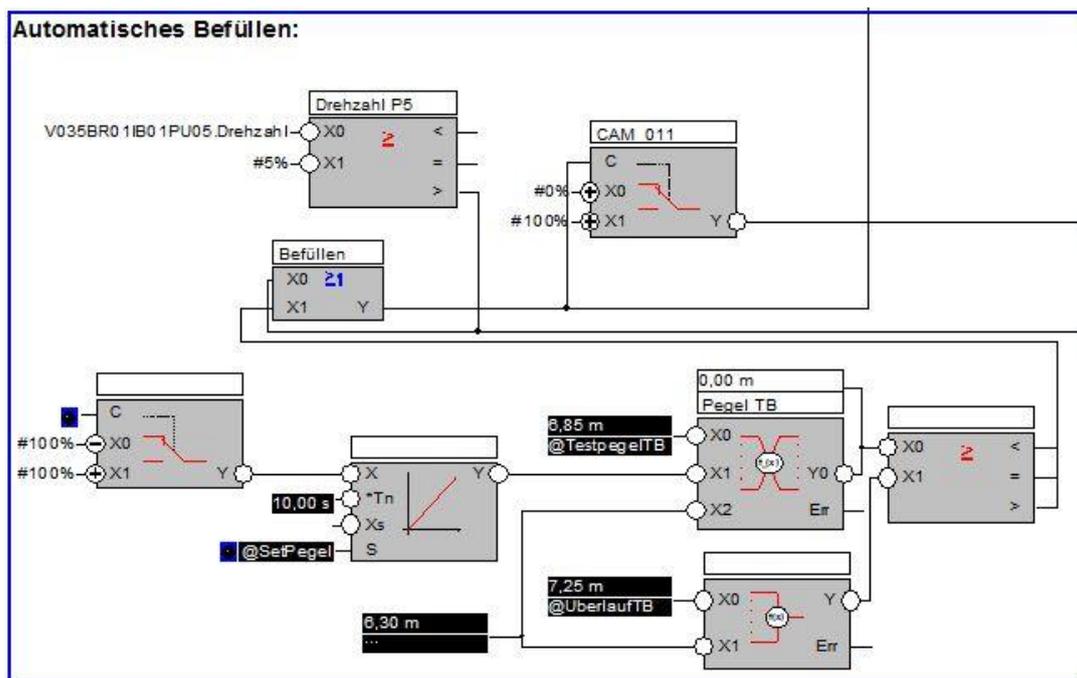


Abbildung 3.31: Automatisches Befüllen [eigene Darstellung]

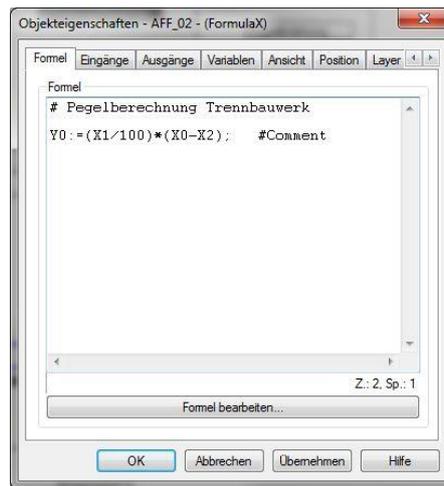


Abbildung 3.32: Pegelberechnung des Trennbauwerks [eigene Darstellung]

Auch für den Speicherpegel wird eine lineare Änderung des Füllstandes angenommen. Zur weiteren Vereinfachung ist die zeitliche Änderung des Pegels in beide Richtungen identisch. Diese ist mit vier Stunden ausreichend groß gewählt, so dass keine Schaltpunkte übersprungen werden. Im Steuerungsprogramm definierte Zeiten, wie zum Beispiel der Spülvorgang mit der zweiten Hauptpumpe, werden dadurch ebenso abgedeckt. Über @SetPegel wird in diesem Fall der Speicherpegel nicht auf den Arbeitspunkt X_s „Null“ gesetzt, sondern auf den gesetzten Speicher-Testpegel der Simulation. Zunächst auf 12 m Wassersäule (mWS) gesetzt. Auch aufgrund der Übersicht bezieht sich der Testpegel sowie die Speicherhöhe nicht auf mNN wie beim Brauchwasserbecken, sondern auf den Ursprung des Speicherbeckens. Die eingesetzte Speicherhöhe wird mit Hilfe des Speichervolumens bestimmt. Abbildung 3.33 zeigt den Speicherpegel und deren Änderung. Das Element *Speicherpegel* zeigt den eingegebenen Testpegel, der jeweils durch die Pegeländerung verringert oder erhöht wird. Die *Pegeländerung* normiert das eintreffende Signal auf das interne Meter-Format und beschränkt den steigenden Pegel auf die gesetzte Speicherhöhe, sowie den sinkenden Pegel auf null (Abbildung 3.34).

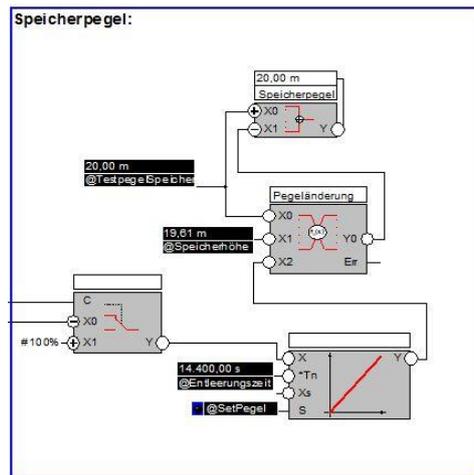


Abbildung 3.33: Speicherpegel [eigene Darstellung]

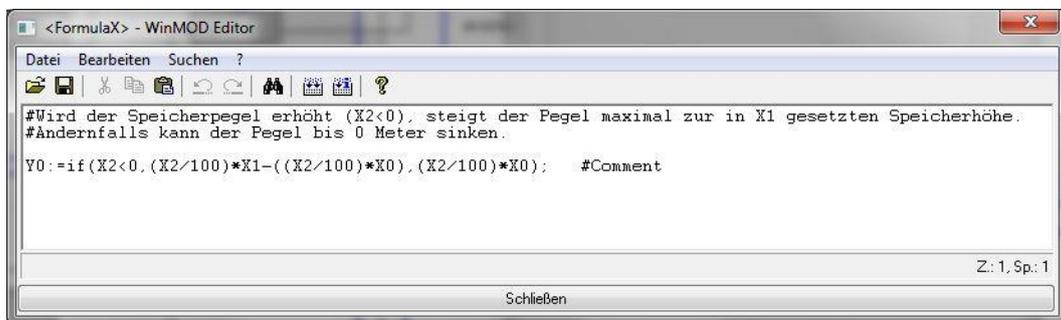


Abbildung 3.34: Pegelberechnung des Speicherpegels [eigene Darstellung]

Der ebenfalls linear abfallende oder steigende Pegel des Brauchwasserpegels entleert sich bei laufender Brauchwasserpumpe während der Sohlenreinigung. Schaltet das Magnetventil steigt der Pegel, davon ausgehend, dass sich das Brauchwasserbecken schneller befüllt als entleert. Die Entleerungszeit des Brauchwasserbeckens wird auf 30 Minuten gesetzt. Dies ist erforderlich, um bei einer Sohlenreinigung gleichzeitig auch ein Befüllen des Brauchwasserbeckens über den Zulauf zu ermöglichen. Das analoge Pegelsignal welches im zugehörigen Makro weiterverarbeitet wird, wird äquivalent zum Speicherpegel gebildet. Über das @SetPegel Signal wird der Pegel auf den gesetzten Wert von 2,10 mWS gesetzt. Die Beckenhöhe ist aus der Beschreibung entnommen und beträgt 3 mWS, bezogen auf den Grund des Brauchwasserbeckens (Abbildung 3.35).

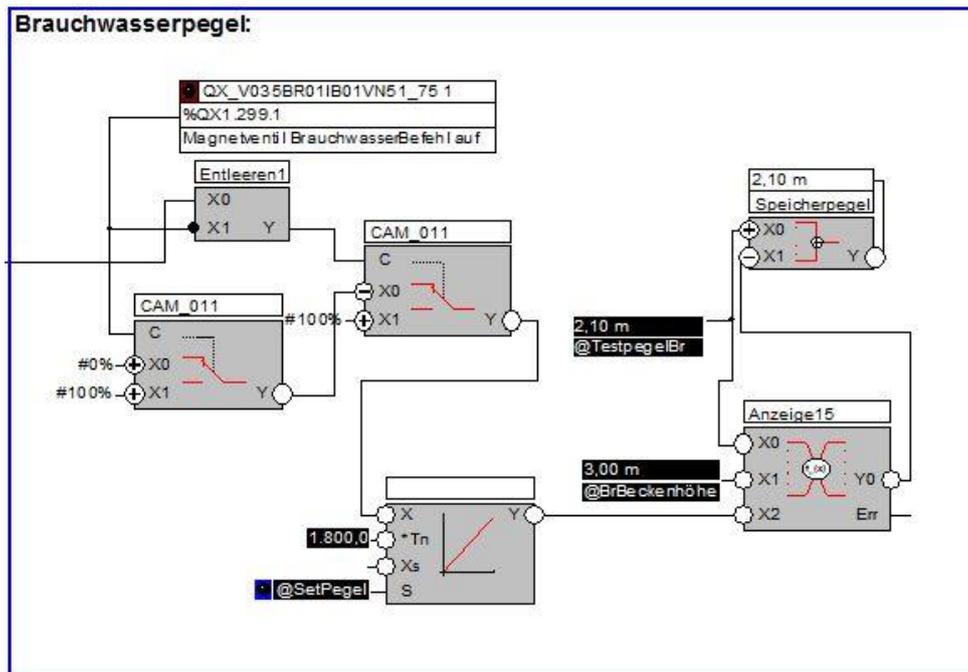


Abbildung 3.35: Brauchwasserpegel [eigene Darstellung]

Für die Speicherhöhe und als Vergleichswert für das Steuerungsprogramm und Simulation, wird das Volumen des Speicherbeckens bestimmt. Dies entspricht einer Näherung und nicht dem exakten Wert. Hierfür wird das Becken in drei beziehungsweise vier Volumina aufgeteilt. Der Kegelstumpf I, sowie Zylinder II, sowie der Innenzylinder III und IV. Das ergibt für das Gesamtvolumen von 7400 m³.

$$V_{ges} = (V_I - V_{III}) + (V_{II} - V_{IV}) \quad (3.1)$$

Die Volumina V_{II} , V_{III} und V_{IV} werden mit Hilfe der Zylinder Volumenformel:

$$V = \pi \cdot r^2 \cdot h \quad (3.2)$$

Das Volumen des Kegelstumpfes V_I kann über die zugehörige Volumenformel ermittelt werden.

$$V = \frac{h \cdot \pi}{3} \cdot (R^2 + (R \cdot r) + r^2) \quad (3.3)$$

Die nötigen Strecken ergeben sich aus der Bauzeichnung (siehe 19, Seite 1) und sind bereits in der vereinfachten Darstellung Abbildung 3.36 berücksichtigt. Für die Gesamthöhe des Speicherbeckens wird nun die Höhe des Zylinders V_{II} bzw. V_{IV} benötigt. Zusammengesetzt mit dem berechenbaren Volumen des Kegelstumpfes $V_I - V_{III}$ kann die Höhe des Zylinders h_z berechnet werden und anschließend mit der gegebenen Höhe des Kegelstumpfes h_k zu Gesamthöhe h_{ges} addiert werden.

$$h_{ges} = h_z + h_k = \frac{V_{ges} - (V_I - V_{III})}{\pi \cdot (R^2 - r_i^2)} + h_k = 19,61 \text{ m} \quad (3.4)$$

mit

$$V_I - V_{III} = \frac{h_k \cdot \pi}{3} \cdot (R^2 + (R \cdot r) + r^2) - \pi \cdot r_i^2 \cdot h_k = 2063,88 \text{ m}^3 \quad (3.5)$$

Oberhalb der Höhe des Kegelstumpfes h_k kann das jeweilige Volumen zur variablen Höhe jederzeit berechnet werden. Unterhalb von h_k ist das Volumen abhängig von der variablen Höhe h bezogen auf den jeweils zugehörigen Radius R_k . Um das Problem dieser Abhängigkeit zu umgehen wird aus der Schräge eine lineare Funktion gebildet und ein Rotationskörper um die Y-Achse erstellt. Dafür muss zunächst aus der linearen Funktion die Umkehrfunktion gebildet werden.

$$y = \frac{h_k}{a} \cdot x - \frac{h_k}{a} \cdot r \quad (3.6)$$

wird zu

$$x = \frac{a}{h_k} \cdot y + r \quad (3.7)$$

Anschließend wird das höhenabhängige Volumen des Kegelstumpfes V_k bestimmt.

$$V_k = \int_0^h \left(\frac{a}{h_k} \cdot y + r \right)^2 dy \quad (3.8)$$

Daraus ergibt sich für das Volumen $V_k(h)$ bereits zusammengefasst mit dem inneren Zylindervolumen $V_i(h)$.

$$V_k(h) - V_i(h) = \pi \cdot h \cdot \left[\frac{1}{3} \cdot \left(\frac{a}{h_k} \right)^2 \cdot h^2 + \left(\frac{a}{h_k} \right) \cdot r \cdot h + r^2 - r_i^2 \right] \quad (3.9)$$

Zur Kontrolle des Ergebnisses wird anschließend noch der das Volumen $V_I - V_{III}$ mit dem Volumen $V_k(h) - V_i(h)$ an der Stelle h_k verglichen.

$$V_k(h_k) - V_i(h_k) = V_I - V_{III} = 2063,88 \text{ m}^3 \quad (3.10)$$

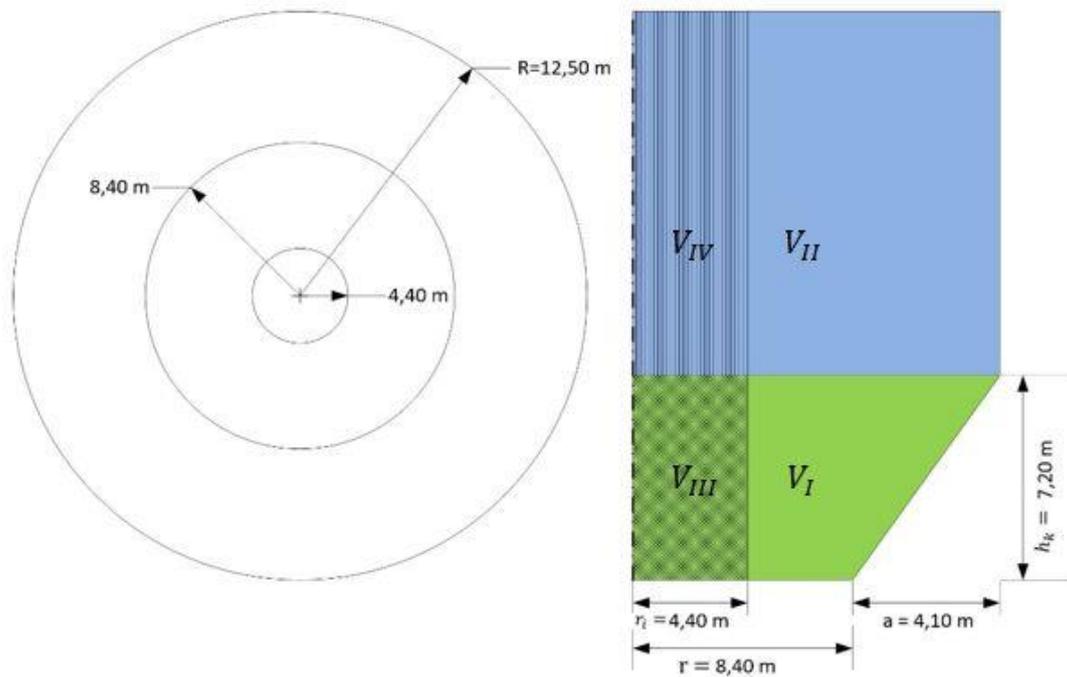


Abbildung 3.36: vereinfachte Darstellung des Speicherpegels [eigene Darstellung]

In Abbildung 3.37 wird die Volumenberechnung in Abhängigkeit des momentanen Speicherpegels dargestellt. Die Volumina des Kegelstumpfes und des Zylinders werden ebenso separat berechnet und schließlich addiert. Für den Kegelstumpf gilt, ab einer Höhe von 7,20 mWs wird h gleich h_k gesetzt und das Volumen $V_I - V_{III}$ ausgegeben. Entsprechend gilt für das Zylindervolumen unterhalb der Höhe des Kegelstumpfes h_k wird der Wert null ausgegeben.

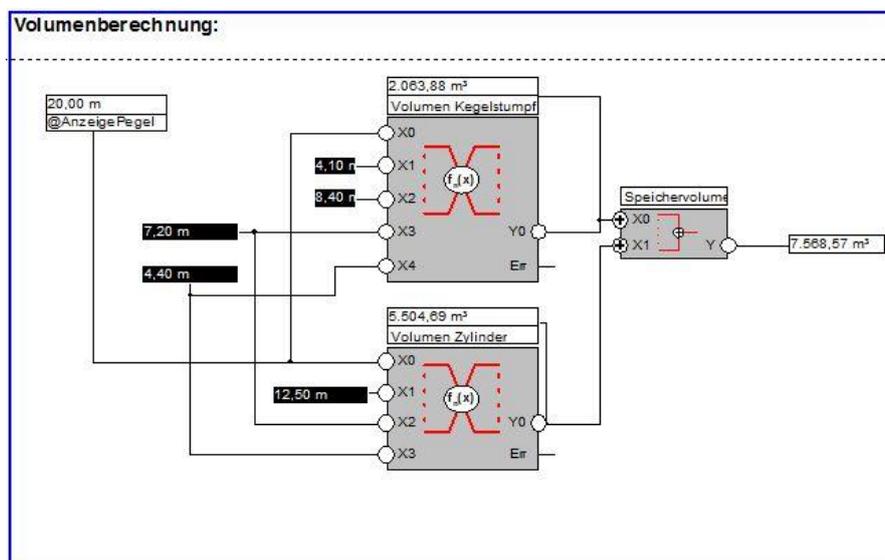


Abbildung 3.37: Volumenberechnung [eigene Darstellung]

Der Durchfluss spielt für die Simulation und das Steuerungsprogramm keine Rolle. Lediglich bei Nutzen von Frequenzumrichtern als Sanftanläufer ist die genauere Betrachtung des Durchflusses sinnvoll. Innerhalb dieser Simulation wurden mit Hilfe der Pumpenkennlinien Tabellen definiert. Je nach Pegel und vorhandenem Vordruck wird der idealisierte Durchfluss ausgegeben (Abbildung 3.38). Unterschieden wird zwischen den beiden Drehzahlen, wenn mit Hilfe der Hauptpumpen das Speicherbecken geleert wird. Laufen die beiden Restentleerungspumpen zur Entwässerung wird ein Durchfluss ähnlich der Hauptpumpen angenommen. Das Durchflussmodell wird in Abbildung 3.39 abgebildet.

Eingang (m)	Ausgang (l/s)
0,00	0,00
10,00	250,00
14,50	350,00
20,05	455,00

Abbildung 3.38: Tabelle anhand der Pumpenkennlinie [eigene Darstellung]

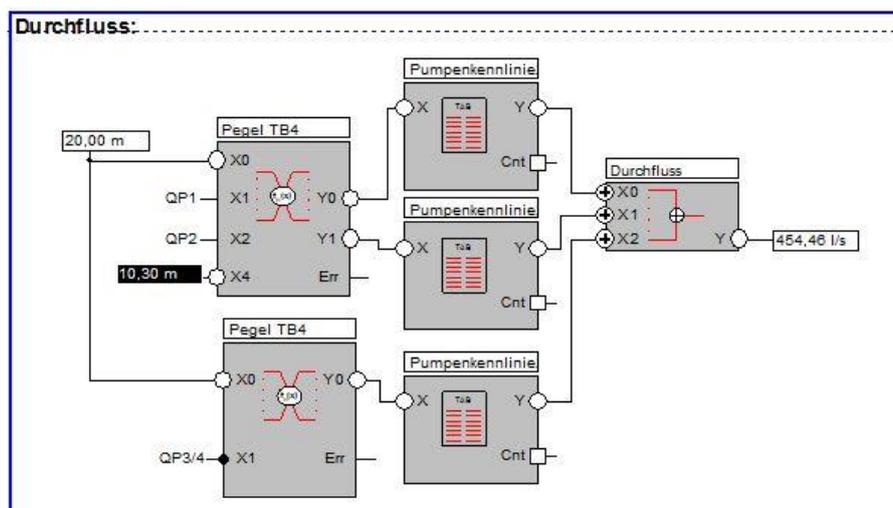


Abbildung 3.39: Durchfluss [eigene Darstellung]

Die Pegel des Überlaufs und Gehölzgraben, sowie der Pumpensumpf im Maschinenraum, wurden nicht weiter betrachtet. Diese können aber direkt manipuliert werden um eine Reaktion auszulösen. Sie

spielen aber lediglich eine untergeordnete Rolle beim Test des Steuerungsprogramms, ähnlich der Entlüftung der Anlage.

Sämtliche Operanden zur Berechnung innerhalb des physikalischen Testmodells wurden global definiert und sind im Simulationsfenster *Parameter* (siehe auch Abschnitt 3.4) Werten zugeordnet. Diese Werte sind der Dokumentation sowie technischen Zeichnungen bezüglich des MHBs Schädlerstraße entnommen. Wie auch bei den Makros wurde darauf geachtet, das physikalische Testmodell möglichst allgemeingültig zu erstellen. Das Testmodell kann demzufolge auch für die Simulation ähnlicher MRB verwendet werden und ermöglicht ein Anpassen sämtlicher Operanden. Ausnahme bildet das Durchflussmodell bei der die erstellten Tabellen von spezifischen Pumpenkennlinien stammen. Das gesamte physikalische Testmodell ist in Abbildung 3.40 dargestellt.

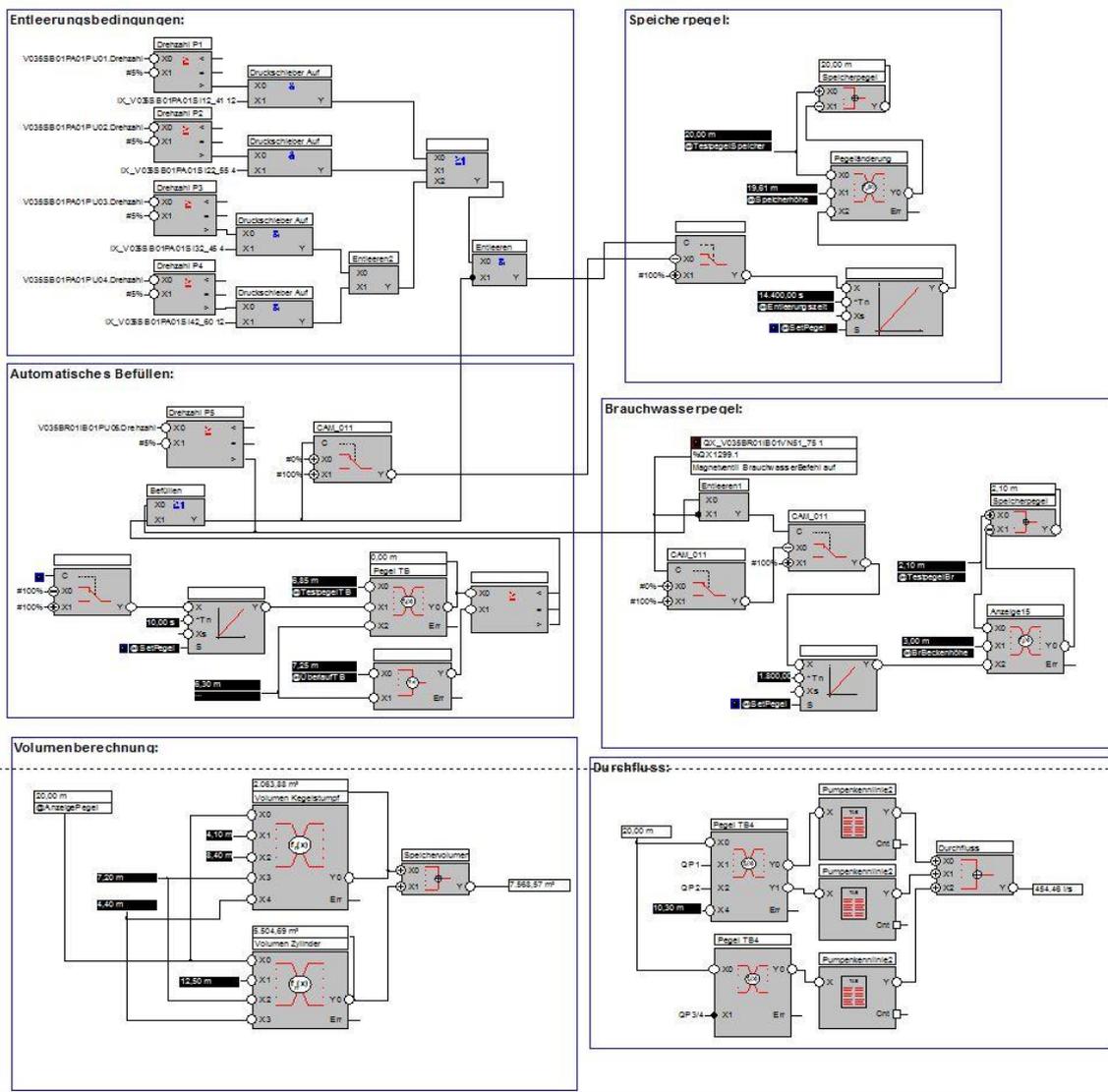


Abbildung 3.40: Physikalisches Testmodell [eigene Darstellung]

Das Anpassen der für den Ablauf benötigten Pegel und deren Skalierung zur Analogwertverarbeitung (siehe Abschnitt 3.1.3) geschieht im jeweiligen Niveau-Makro. Das interne analoge Signal entspricht dem Format *distance [m] +/-100,00 SI**, ein Prozent entspricht einem Meter. Neben den Wertebereichen zur Skalierung können auch die Normierungswerte im Steuerungsprogramm abgelesen werden. Im Falle des Speicherpegels sind das 25 m. Mit Hilfe der Normierung wird das Eingangssignal so weiterverarbeitet das die 25 m einem Wert von 100 % entsprechen. Im nächsten Schritt wird das interne Meter-Format in das interne Digit-Format umgewandelt. Auf einen Zwischenschritt in dem das Pegelsignal von Meter in ein Stromsignal zwischen 4 und 20 mA gewandelt wird, wird verzichtet. Das interne Format *digits ABB [] 0...27648* entspricht dem externen Format und kann direkt vom Eingangssignal abgelesen werden. Da auch die Makros allgemeingültig genutzt werden sollen, wird der Normalisierungsfaktor im Simulationsfenster Parameter für jedes Makro Niveau einzeln definiert. Abbildung 3.41 zeigt den internen Aufbau mit den Schritten zur Normalisierung und Digital-Wandlung.

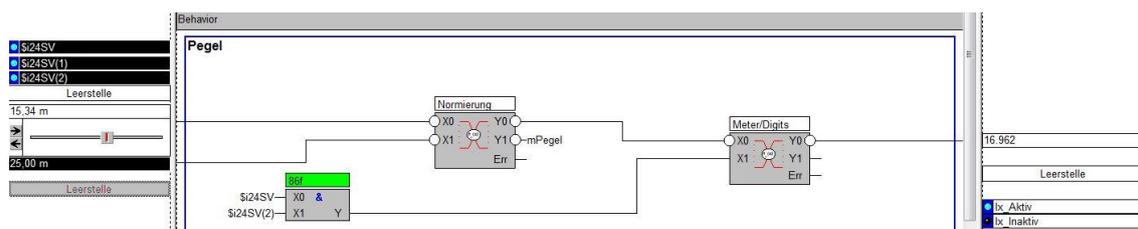


Abbildung 3.41: Aufbau des Makros Niveau [eigene Darstellung]

3.4 Aufbau der Simulation

Die Projektoberfläche für die Simulation ist unterteilt in Fensterschemata. Einen Überblick verschafft das über Umschalt+F4 aufrufbare Projektfenster. Hier erhält man Zugriff auf die globalen Operanden dem Peripherie-Treiber, die Zentralen Definitionen und der Zugriff auf die einzelnen Simulationsfenster den Projekt-Dateien (Abbildung 3.42)

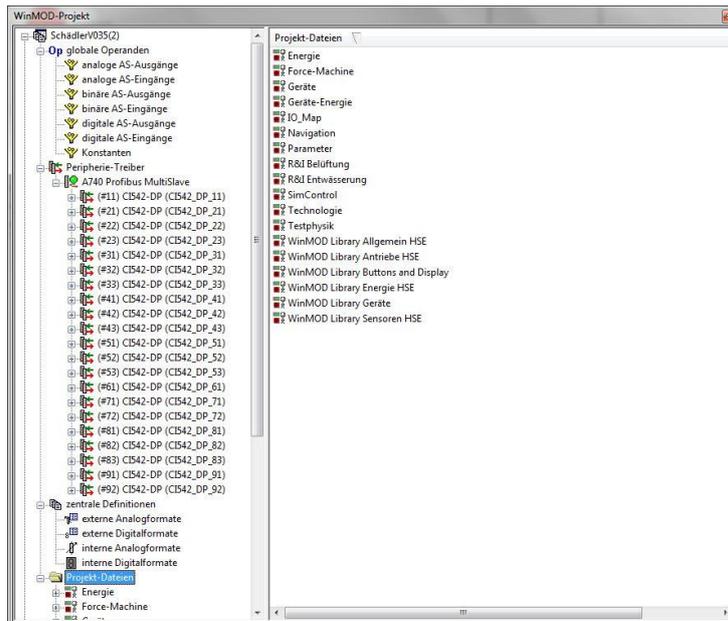


Abbildung 3.42: WinMOD Projektfenster [eigene Darstellung]

Die Fensterschemata werden über das Menü Fenster definiert und lassen sich über die Schaltfläche Fensterschema anwählen. In einem Fensterschemata wird die Anordnung der unterschiedlichen Simulationsdateien zueinander abgespeichert. Die Schaltflächen zum Anwählen sind in dem Navigationsfenster abgelegt. In Abbildung 3.43 ist das Fensterschemata All dargestellt. Bei diesem Fensterschema ist im unteren linken Bereich das Rohrleitungs- und Instrumentenfließschema der Entwässerung mit Signalanzeigeelementen und im rechten unteren Bereich die Geräte Simulationsdatei, in der die Geräte des MRBs implementiert und abgebildet sind. Eine Schalthandlung und deren Auswirkung der Datei Geräte.sim kann in der zum technologiegehörigen Simulationsdatei R&I Entwässerung beobachtet werden.

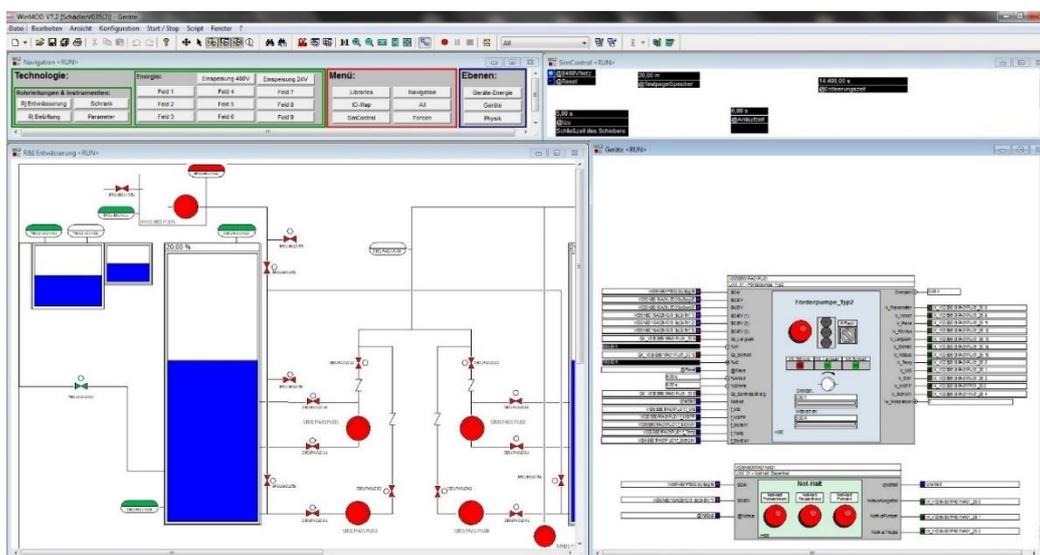


Abbildung 3.43: Fensterschema All [eigene Darstellung]

Die Navigationsleiste und das rechts oben sichtbare Sim Control Fenster sind in jedem definiertem Schemata sichtbar. Über die Navigation können die verschiedenen Schemata angewählt werden. Sim Control beinhaltet globale Operanden die nur der Steuerung der Simulation dienen, hier kann zum Beispiel der Pegel des Speicherbeckens eingestellt werden. Es handelt sich um virtuelle Signale die nicht durch einen Prozess abgebildet werden. Jedes Fenster kann für sich im gesamten Projekt gestartet werden. Elemente in den Fenstern können erst nach starten der Simulation genutzt werden. Die Tabellen 3.1 und 3.2 beinhalten die in den Fensterschemata anwählbaren Simulationsdateien und deren Inhalt.

Tabelle 3.1: Fensterschemata Menü

Fensterschema	Simulationsdatei	Inhalt der Simulationsdatei
Libraries	Alle Bibliotheken	Bibliotheken Allgemein, Antriebe, Buttons and Display, Geräte, Energie und Sensoren
IO-Map	IO-Map.sim	Eingangs- / Ausgangssignale der SPS
SimControl	SimControl.sim	Simulations-Steuersignale
Navigation	Navigation.sim	Fensterschema
All	*.sim	Grundposition aller Fenster
Forcen	Force-Machine.sim	Aktives Forcen ausgewählter Signale

Tabelle 3.2: Fensterschemata Anlagen- und Prozesssimulation

Fensterschema	Simulationsdatei	Inhalt der Simulationsdatei
Geräte-Energie	Geräte-Energie.sim	Geräte zur Energieübertragung des MRB
Geräte	Geräte.sim	Geräte des MRB (Pumpen, Schieber, Messeinrichtungen)
Physik	Testphysik.sim	Physikalisches Testmodell zur Speicharentleerung
RI_Entwässerung	R&I Entwässerung.sim	R&I Schema der BE- / Entwässerung
RI_Belüftung	R&I Belüftung.sim	R&I Schema der BE- / Entlüftung
Schrank	Technologie.sim	Aufbau der Schaltschränke
Parameter	Parameter.sim	Parametrierung der Geräte
Einspeisung 400V	Energie.sim	Einspeisung 400V und 400V Feldeinspeisung
Einspeisung 24V	Energie.sim	Einspeisung 24V und 24V Feldeinspeisung
Feld 1-9	Energie.sim	Feld-Spannungsversorgung

Die Simulationsdateien wurden alle für das Projekt selbst erzeugt. Die in der Simulation als Metadatei hinterlegten Grafiken wurden mit Microsoft Visio, auf Basis der Bediengerätebilder, Stromlaufpläne und Fotos erstellt.

4 Steuerungsprogramm

Das bestehende Steuerungsprogramm basiert auf der Anforderungsdokumentation (siehe 16) zum MRB Schädlerstraße. Im Rahmen der Erneuerung 2013 wurde die vorhandene SPS ausgetauscht und durch eine neue der Firma ABB ersetzt. Die Programmierung erfolgte mit CoDeSys. Im Folgenden wird der Ablauf des Programms, die Vorgehensweise zur Verifizierung der Signale und dem Arbeiten mit externer SPS beschrieben.

4.1 Vorhandene Ablaufprogramme

4.1.1 Entwässerung

Das MRB funktioniert als riesiges Sammelbecken, das bei Starkregenereignissen die Überlastung des Mischwassersielnetzes verhindern und eine Belastung der Wandse als Zufluss der Alster verhindern soll. Innerhalb des Siels befindet sich ein variabler Schweller der den Zulauf zum Speicherbecken über eine Drallkammer, zur Erhöhung der Fließgeschwindigkeit um Ablagerungen zu verhindern, ermöglicht. Der Zulauf befindet sich im sogenannten Trennbauwerk. Dort befindet sich auch die für die Entwässerung wichtige Pegelmessung. Diese ist redundant ausgelegt. Steigt der Pegel im Mischwassersiel über 6,80 m MNN (mNN) wird eine begonnene Speicherentleerung unterbrochen. Hintergrund ist, dass das im MRB gespeicherte Wasser bei Beendigung des Starkregenereignisses in das selbe Zulaufsiel zurückgeführt wird. Deshalb ist auch eine zeitliche Steuerungskomponente für den Entwässerungsvorgang des Speicherbeckens vorhanden. Der automatische Abpumpvorgang beginnt erst, wenn der Pegel im Becken sich innerhalb von zwei Stunden nicht verändert hat. Die Abbildung 4.1 verdeutlicht den Zulauf des MRBs über das Mischwassersiel. Zu sehen ist auch der Gehölzgraben, der den Ablauf zur Wandse darstellt, falls das Volumen des Speicherbeckens nicht ausreicht.

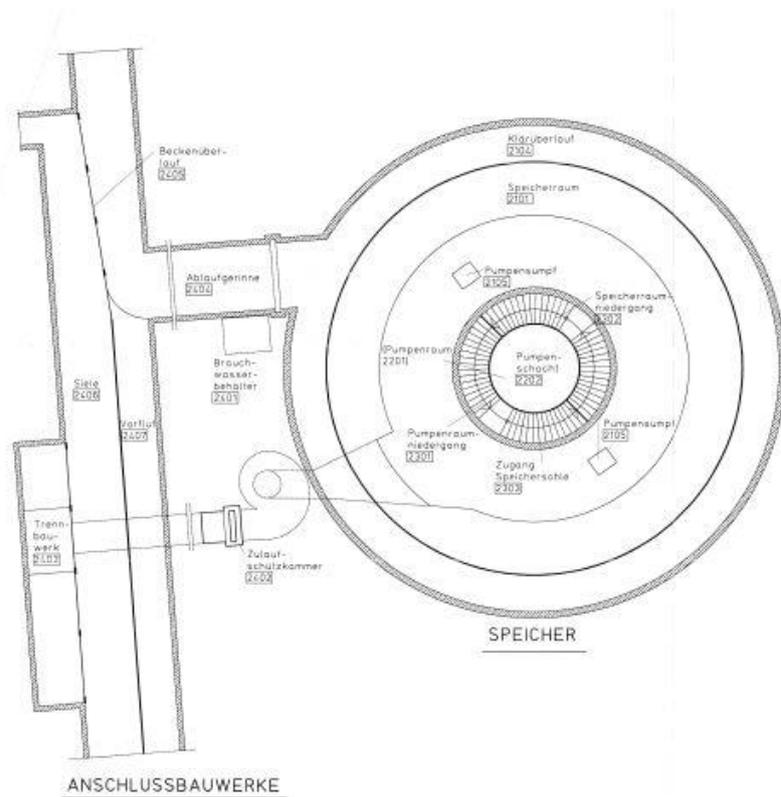


Abbildung 4.1: Mischwassersiel und Zulauf zum Speicher des MRBs [eigene Darstellung]

Über die Drallkammer wird das mit bis zu $10 \frac{\text{m}}{\text{s}}$ fließende Mischwasser in die Speicherkammer des MRBs geleitet. Im Zulauf zur Drallkammer befindet sich das Zulaufschütz, welches im Regelfall immer geöffnet ist. Im Speicherbecken befinden sich drei Pegelmessungen für das Niveau. Eine Radarmessung zur Volumenbestimmung und zwei Druckmesser für die Schaltpunkte des Pegels. Die Speicherentleerung läuft über mehrere Schritte, mit Hilfe von zwei Hauptpumpen und zwei Restentleerungspumpen. Die beiden Hauptpumpen werden mit zwei Drehzahlen gefahren, folgend mit „kleiner“ und „großer“ bezeichnet. Die unterschiedlichen Drehzahlen werden über zwei getrennte Wicklungen mit unterschiedlichen Polaritäten erreicht. Diese haben die Möglichkeit von verschiedenen Geschwindigkeitsverhältnissen und sind nicht wie bei der Dahlander-Schaltung an ein bestimmtes Verhältnis gebunden. Die beiden Restentleerungspumpen laufen im Parallelbetrieb. Zur Verringerung des Anlaufstroms sind die Pumpen in Stern-Dreieck geschaltet. Beim Einschalten der Motoren kann der Anlaufstrom das Siebenfache des Nennstroms erreichen. Durch die Stern-Dreieck-Anlaufschaltung kann die Leistungsaufnahme zu Beginn auf ein Drittel des Anlaufstroms reduziert werden. Dabei ist die Strangspannung in Sternschaltung statt der sonst genutzten 400V auf 230V reduziert und wird dann, mechanisch über ein Zeitrelais als Schützschtaltung, auf Dreieckschaltung umgestellt.

Falls das Speicherbeckenvolumen nicht mehr ausreicht, befindet sich oberhalb des Überlaufs die Vorklärung. Die Vorklärung kann man sich als grobes Sieb vorstellen um Spinnstoffe daran zu hindern in den Gehölzgraben zu gelangen und von dort aus in den Zufluss zur Alster. In dem Überlauf befinden sich ebenfalls zwei redundant aufgebaute Pegelmessungen mit deren Hilfe die Abschlagsmenge berechnet werden kann.

Bei gleichbleibendem Pegel im Speicherbecken wird nach zwei Stunden die Entleerung begonnen. Überschreitet der Pegel im Trennbauwerk einen Wert von 6,80 mNN wird der Entleerungsvorgang unterbrochen. Beginnt die Entleerung, fahren die Saugschieber der Haupt- und Restentleerungspumpen sowie die Druckschieber einer der Haupt- und der beiden Restentleerungspumpen in die geöffnete Grundstellung. Bei einem Wasserstand von mehr als 10,30 m Wassersäule (mWS) beginnt eine der Hauptpumpen den Entwässerungsvorgang in „kleiner“ Drehzahl. Die Pumpen werden je nach Betriebsdauer alternierend gewählt um ein ähnliches Betriebsstundenverhältnis zu gewährleisten. Unterhalb der 10,30 mWS schaltet die Pumpe auf „große“ Drehzahl um den geringeren Vordruck zu kompensieren. Im Anhang X befinden sich die zugehörigen Pumpenkennlinien der beiden Drehzahlen der *KSB/KRPE 300-630* Pumpe. Erreicht das Niveau im Speicherbecken 3,10 mWS, öffnet sich der Spülschieber der nicht für den Entleerungsvorgang genutzten Pumpe. Diese startet in der „kleinen“ Drehzahl und leitet das gleichzeitige Spülen im Becken ein. Das Wasser wird von der Spülpumpe im Kreis gefördert und hält das Wasser in Bewegung, so dass sich keine Ablagerungen absetzen können. Wird ein Pegel von 2,30 mWS gemessen schaltet die Entleerungspumpe ab. Der zugehörige Druckschieber geschlossen und der Spülschieber geöffnet. Mit dem Start des Spülvorgangs beginnen die beiden Restentleerungspumpen die Speicherentleerung. Bei einem Pegel von 0,80 mWS wird der Spülvorgang beendet. Erreicht das Niveau 0,30 mWS stoppt die Speicherentleerung. Die Absperrschieber SI52, SI53, SI54 öffnen. Zwecks Sohlenreinigung wird Brauchwasser in das Speicherbecken gepumpt bis der Pegel wieder 1,30 mWS erreicht. Anschließend schließen die Absperrschieber und ein 15-minütiger Spülvorgang mit beiden Hauptpumpen beginnt. Der Brauchwasserbeckenwasserstand wird über ein Magnetventil gesteuert (SI51). Erreicht der Pegel 2,00 mWS öffnet das Ventil für den Zulauf des Brauchwasserbeckens. Bei einem Wasserstand von 2,10 mWS wird der Zulauf unterbrochen und das Magnetventil wird nicht mehr angesteuert. Falls das Pegel-Messgerät ausfällt befindet sich noch ein Schwimmschalter im Becken, der die Überwachungsfunktion des Pegelmessgeräts übernehmen kann (Schaltkontakt). Nach beenden des Spülvorgangs stoppen die Hauptpumpen und deren Saug- sowie Spülschieber schließen sich. Die Restentleerungspumpen leeren das Speicherbecken. Im letzten Schritt schließen die

restlichen geöffneten Schieber. Das entleerte MRB befindet sich wieder in der geschlossenen Grundstellung. In Abbildung 4.2 ist die Entwässerungsanlage veranschaulicht.

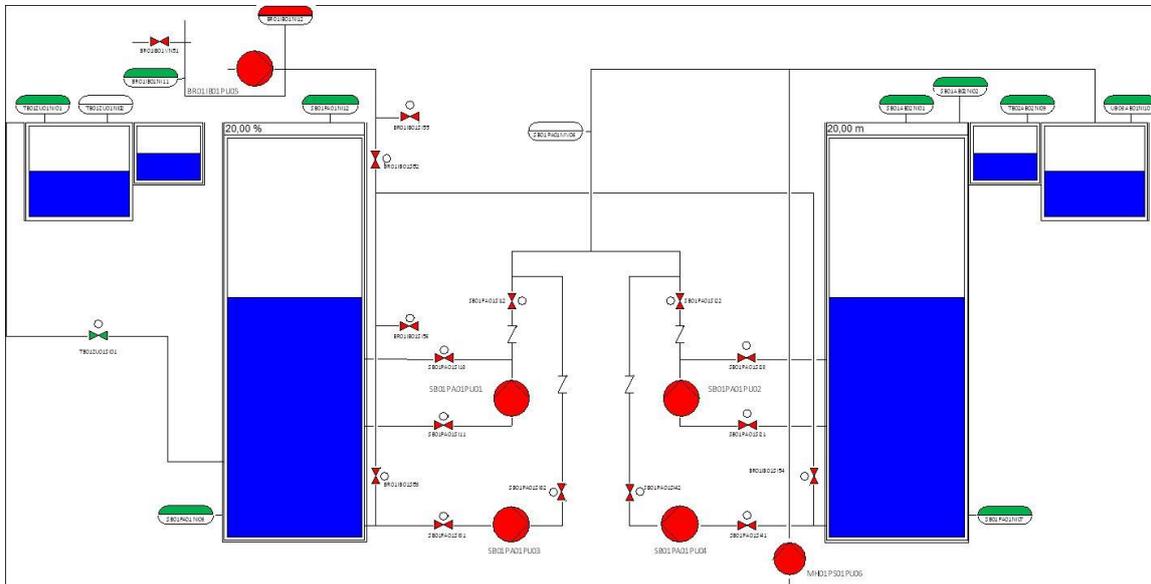


Abbildung 4.2: Entwässerungsanlage [eigene Darstellung]

Der Ablauf zur Speicherentleerung wurde in CoDeSys in Schritte aufgeteilt und jeweils fortlaufend programmiert. Abbildung 4.3 zeigt die Schritte 0 - 32 sowie den Schritt 99 Störung. Fällt während des Entleerungsvorgangs eine Pumpe oder nötiger Schieber aus, schaltet das Programm nach 10 Minuten Wartezeit in diesen Schritt. Durch manuelle Quittierung wird zurück in die Grundstellung 0 gefahren.

Keine Pumpe gestört bzw. Störung einer oder beider Restentleerungspumpen	
0	Grundstellung
1	Schieber in Grundstellung fahren auf
2	1.Hauptentleerungspumpe kleine Drehzahl ein
3	1.Hauptentleerungspumpe große Drehzahl ein
4	Spülschieber 2.Hauptentleerungspumpe auf
5	2.Hauptentleerungspumpe kleine Drehzahl (Spülen) ein
6	Hauptentleerungspumpe große Drehzahl (Entleeren) auf
7	Druckschieber 1.Hauptentleerungspumpe zu
8	Spülschieber 1.Hauptentleerungspumpe auf
9	1.Hauptentleerungspumpe kleine Drehzahl (Spülen) ein
10	Restentleerungspumpen ein
11	Druckschieber Restentleerungspumpen zu
12	Restentleerungspumpen aus
13	Brauchwasserschieber bei Hpt Pumpen auf
14	Brauchwasserschieber bei Hpt Pumpen auf
15	Brauchwasserschieber auf
16	Brauchwasserpumpe ein
17	Brauchwasserpumpe aus
18	Brauchwasserschieber zu
19	Brauchwasserschieber bei Hpt Pumpen zu
20	2.Hauptentleerungspumpe kleine Drehzahl (Spülen) ein
21	1.Hauptentleerungspumpe kleine Drehzahl (Spülen) ein
22	Beide Hauptentleerungspumpe kleine Drehzahl aus
23	Spülschieber Beide Hauptentleerungspumpe zu
24	Druckschieber Restentleerungspumpen auf
25	Restentleerungspumpen ein
26	Druckschieber Restentleerungspumpen zu
27	Restentleerungspumpen aus
28	Druckschieber 1.Hauptentleerungspumpe auf
29	1.Hauptentleerungspumpe kleine Drehzahl ein
30	1.Hauptentleerungspumpe kleine Drehzahl aus
31	Druckschieber 1.Hauptentleerungspumpe zu
32	Schieber in Grundstellung fahren zu
99	Störung

Abbildung 4.3: Schrittkette 1 [7]

Ebenso sind Schrittketten bei Störung einer Hauptpumpe beziehungsweise Störung beider Hauptpumpen hinterlegt.

Liegt zu Beginn der Speicherentleerung die Störung einer Hauptpumpe vor, wird der Vorgang mit der Schrittkette 2, Schritt Nummer 100 fortgesetzt. Die Entleerung ist äquivalent zur ersten Schrittkette, lediglich die Spülvorgänge werden mit nur einer Pumpe realisiert. Bei einem Ausfall der Brauchwasserpumpe wird die Sohlenreinigung in beiden Fällen einfach übersprungen. Schrittkette 2, mit den Schritten 100 - 122 ist in Abbildung 4.4 ersichtlich.

100	Schieber in Grundstellung fahren auf
101	1.Hauptentleerungspumpe kleine Drehzahl ein
102	1.Hauptentleerungspumpe große Drehzahl ein
103	Spülschieber 1.Hauptentleerungspumpe auf
104	Druckschieber 1.Hauptentleerungspumpe zu
105	Restentleerungspumpen ein
106	Druckschieber Restentleerungspumpen zu
107	Restentleerungspumpen aus
108	Druckschieber Restentleerungspumpen auf
109	1.Hauptentleerungspumpe kleine Drehzahl aus
110	Brauchwasserschieber bei Hpt Pumpen auf
111	Brauchwasserschieber auf
112	Brauchwasserpumpe ein
113	Brauchwasserpumpe aus
114	Brauchwasserschieber zu
115	Brauchwasserschieber bei Hpt Pumpen zu
116	1.Hauptentleerungspumpe kleine Drehzahl ein
117	1.Hauptentleerungspumpe kleine Drehzahl aus
118	Spülschieber 1.Hauptentleerungspumpe zu
119	Restentleerungspumpen ein
120	Druckschieber Restentleerungspumpen zu
121	Restentleerungspumpen aus
122	leer

Abbildung 4.4: Schrittkette 2 [7]

Sind beide Hauptpumpen gestört erfolgt die Speicherentleerung lediglich mit Hilfe der beiden Restentleerungspumpen, sämtliche Spül- und Reinigungsvorgänge entfallen. Dargestellt die Schritte 200 – 204 der Schrittkette 3 in Abbildung 4.5.

Beide Hauptentleerungspumpen gestört	
200	Druckschieber 2.Hauptentleerungspumpe zu
201	Druckschieber 1.Hauptentleerungspumpe zu
202	Restentleerungspumpen ein
203	Druckschieber Restentleerungspumpen zu
204	Restentleerungspumpen aus

Abbildung 4.5: Schrittkette 3 [7]

Kommt es während des Betriebes zu Störungen soll die Entwässerung weiterhin gewährleistet sein und automatisch mit der nächsten Schrittkette weitergefahren werden.

Falls eine notwendige Entwässerung des Pumpenraumes nötig ist, schaltet die sich dort befindliche Entwässerungspumpe PU06, über den angebrachten Schwimmschalter in deren Pumpensumpf, zu.

4.1.2 Be- und Entlüftung

Grundsätzlich ist die Entlüftung nicht Teil der Prüfung des Programms. Das Hauptaugenmerk liegt auf den automatischen Entwässerungsvorgang. Da die Entlüftung jedoch auch Teil der Modellierung der WinMOD-Simulation ist, soll die Funktion kurz beschrieben werden. In Abbildung 4.6 ist die Lüftungsanlage veranschaulicht.

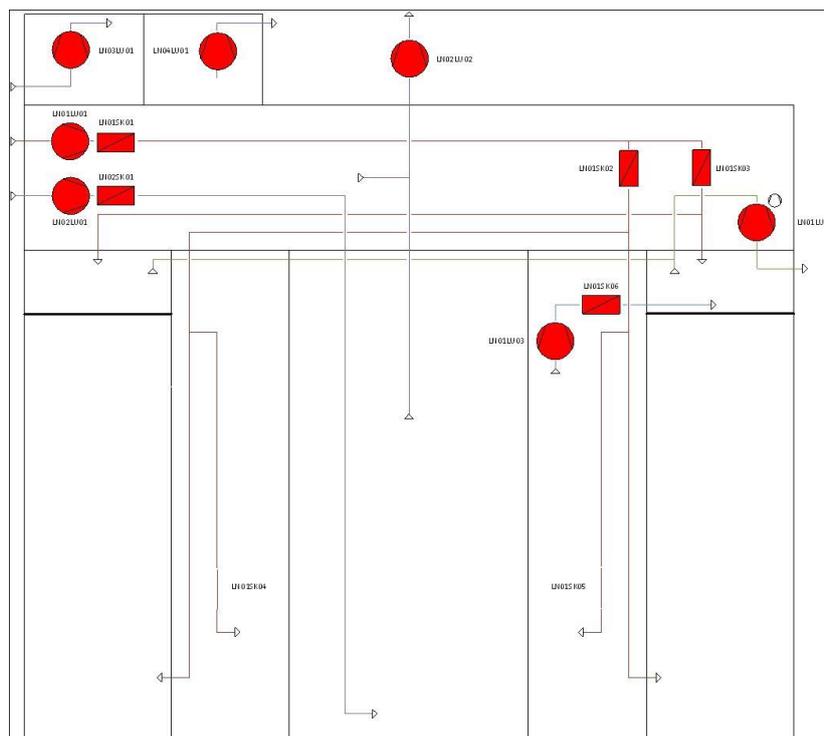


Abbildung 4.6: Entlüftungsanlage [eigene Darstellung]

Wird über das Bedienpanel die Begehung der Inspektionsebene des MRBs angemeldet, werden die beiden Luftklappen des Zulüfters Speicher (LN01 SK01, LN01 SK02) geöffnet, der Zulüfter Speicher (LN01 LU01) und der Ablüfter Speicher (LN01 LU02) eingeschaltet. Bei einer Begehung des Speicherbeckens (Anmeldung am Panel) wird zusätzlich noch die dritte Luftklappe des Speicher-Zulüfters zum Speicher (LN01 SK03) geöffnet, sowie der Ablüfter des Speicherniedergangs (LN01 LU03)

eingeschaltet. Die beiden Lüftungsklappen des Speicherniedergangs (LN01 SK04, LN01 SK05) sind ausgebaut und der Zulüfter sowie Ablüfter des Speichers verfügen nur noch über eine Drehzahl. Ebenso ist ein zyklischer Lüftungsbetrieb des Speichers eingerichtet. Dabei werden Zul- und Ablüfter des Speichers zu einer festgeschriebenen Start- und Laufzeit angesteuert.

Die beiden Lüfter des Betriebsgebäudes (LN02 LU01, LN02 LU02) und die Luftklappe des Zulüfters (LN02 SK01) werden bei einer Begehung (Anmeldung über Panel) des Betriebsgebäudes angesteuert. Außerhalb einer Begehung wird die Belüftung eingeleitet, sobald der Thermostat des Ablüfters (LN02 LU02) auslöst.

Bei Auslösung einer der Brandschutzklappen (LN02 SK02, LN02 SK03) in der Zuluftleitung, wird die Lüftungsanlage sofort abgeschaltet und sämtliche Lüftungsklappen geschlossen.

Der Lüfter im Traforaum (LN03 LU01) löst aus, sobald einer der einstellbaren Trafotemperaturüberwachungen (MS01 TR01 TM01, MS01 TR01 TM05) anspricht.

Der Lüfter (LN04 LU01) im Niederspannungs-Schaltraum wird eingeschaltet sobald eine der Temperaturüberwachungen in den 8 Schaltschränken auslöst.

4.2 Verifizierung der Simulation Vorort

Nach Anschluß des Simulations PC erfolgt der Test des Steuerungsprogramms. Hierfür werden einzelne Signale im WinMOD Programm gesetzt (geforct). Über die im MRB vorhandenen Panels werden zur Validierung der Simulation die Geräte bedient (Betriebsart Hand) und beobachtet.

Die Multi-Slavekarte der WinMOD-Simulation wird direkt über PROFIBUS mit der Vorort-SPS verbunden. Anschließend wird über die Panelhandsteuerung jedes Gerät angesteuert und überprüft.

Mit Betätigen des *Auf* Button wird über die SPS das Ausgangssignal zum Öffnen des Schiebers an die Simulation gesendet. Die Eingangssignale werden zurück zur SPS geschickt, verarbeitet und können direkt am Panel abgelesen werden. Der offene Schieber ist grün, der geschlossene rot hinterlegt. Die Öffnungs-/ Schließvorgänge werden in der jeweiligen Farbe blinkend dargestellt. Befindet sich der Schieber zwischen den Endlagen und in Ruhestellung (Stopp) wird er weiß angezeigt (Abbildung 4.7).

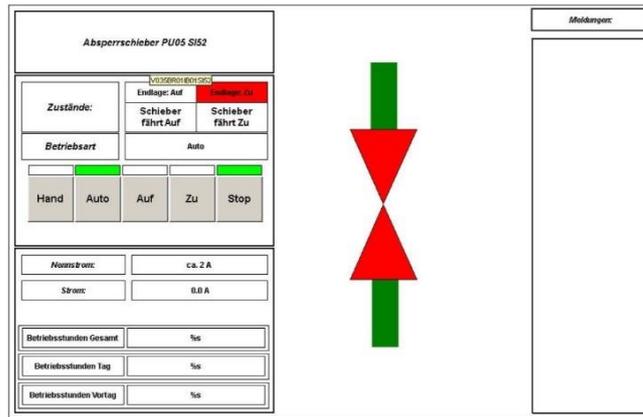


Abbildung 4.7: Schieberdarstellung des Steuerungsprogramms [7]

Am Panel erfolgt im Prozessbild (Pegelübersicht) die Prüfung und Verifizierung der Analogsignale (Abbildung 4.8). Hierzu werden in der Simulation die internen Analogsignale auf einen bestimmten Wert gefordert und über das externe Ausgangssignal am Panel abgelesen.

V035TB02AB02NI09 Ablauf Gehoelzgraben	L125	Revision mit Messwert	Revision ohne Messwert	4.14 mNN
V035SB1AB02NI01 Ablaufbauwerk Überlauf 1 L012	L012	Revision mit Messwert	Revision ohne Messwert	0.00 mNN
V035SB1AB02NI02 Ablaufbauwerk Überlauf 2 L013	L013	Revision mit Messwert	Revision ohne Messwert	0.00 mNN
V035BR01IB01NI11 Brauchwasser Pegel L184	L184	Revision mit Messwert	Revision ohne Messwert	0.00 mNN
V035MH01PS01NI01 Pumpensumpf (Speicher)	NI06	Revision mit Messwert	Revision ohne Messwert	0.00 mWs
V035SB01PA01NI12 Speicher	NI07	Revision mit Messwert	Revision ohne Messwert	0.00 mWs
V035SB01PA01NI06 Speicher (Radarmessung)	L185	Revision mit Messwert	Revision ohne Messwert	0.00 mWs
V035TB01ZU01NI02 Trennbauwerk	L127	Revision mit Messwert	Revision ohne Messwert	6.30 mNN
V035UB01AB01NI10 Überlaufbauwerk (Mischwassersiel)	L128	Revision mit Messwert	Revision ohne Messwert	6.30 mNN
V035TB01ZU01NI01 Zulauf Gehoelzgraben	L126	Revision mit Messwert	Revision ohne Messwert	6.30 mNN

Abbildung 4.8: Pegelübersicht im Steuerungsprogramm [7]

Im Anschluss erfolgt in ersten Tests die Überprüfung der Schrittketten an der Vorort-SPS des MRBs Schädlerstraße. Dazu wird der Speicherbeckenpegel manuell in WinMOD eingestellt und die Schrittkette von Hand am Panel gestartet. Das Verhalten der Schrittketten wird geprüft indem der Pegel um die Schaltpunkte variiert wird und einzelne Störsignale ausgelöst.

4.3 Arbeiten mit der Test-SPS

Um die Validierung des Steuerungsprogramms zu erleichtern wird eine Test-SPS eingesetzt. Damit besteht die Möglichkeit die Validierung auch außerhalb des MRBs Schädlerstraße durchzuführen. Mit Hilfe des Control Builder Plus wird die Hardwarekonfiguration der Vorort-SPS an die vorliegende Test-SPS angepasst [siehe Anhang]. Der Unterschied bezieht sich lediglich auf die CPU-Einheit, statt der PM590 ETH wird für das Testen des Steuerungsprogramms auf die PM583 ETH zurückgegriffen. Diese verfügt über einen kleineren Programmspeicher von 1 MB statt der 2 MB der PM590. Der Speicher ist mit 1 MB ausreichend bemessen. Die Test-SPS verfügt über kein Panel. In dem unter CoDeSys erstellten Steuerungsprogramm ist aber ein Abbild des Panels implementiert, so dass eine Validierung des Steuerungsprogramms trotzdem möglich ist. Nötige Parameter werden äquivalent zum MRB Schädlerstraße per Hand nachgetragen um anschließend die Schrittketten sowie Störfälle zu testen.

5 Validierung des Steuerungsprogramms

Das bisher genutzte Steuerungsprogramm ist wenig tolerant gegenüber unscharfen Zustandsbedingungen und generiert dadurch eine hohe Störanfälligkeit. So bricht der Entleerungsvorgang mitunter ohne ersichtlichen Grund ab oder es erfolgt bei Störungen innerhalb der Schrittketten kein automatisches Umschalten zur nächsten Schrittkette. Mit Hilfe der erstellten WinMOD-Simulation und der eingerichteten Test-SPS kann das Steuerungsprogramm getestet werden. Im Wesentlichen liegt das Hauptaugenmerk auf der Entwässerung. Auf die Entlüftung wird kein Einfluss genommen. Zuerst wird der störungsfreie Ablauf geprüft. Hierbei wird der Pegel im Speicherbecken so eingestellt, dass er oberhalb des Umschaltpunkts zur „großen“ Drehzahl liegt. Der Start der Schrittkette zur Entwässerung des Speicherbeckens erfolgt manuell. Ansonsten würde das Programm erst starten, nachdem der Pegel sich über einen Zeitraum von zwei Stunden nicht geändert hat. Beim manuellen Start der Entleerung ist zu berücksichtigen, dass der Pegel im Trennbauwerk sich einmalig verändert hat. Um zu verhindern das einzelne Schaltpunkte übersprungen, wird die Entleerungszeit auf mindestens 4 Stunden gesetzt.

Um die fehlerfreie Funktion des Steuerungsprogramms zu gewährleisten, wird der Ablauf der Schrittketten mehrmals manuell ausgelöst. Dabei werden vor Beginn der Entleerung in der Simulation einzelne Fehlersignale gesetzt, um eine Störung einzelner oder mehrerer Geräte zu provozieren und den Ablauf aller Schrittketten zu verifizieren. Führt die Fehlerprovokation dazu, dass die Schrittkette nicht korrekt abgearbeitet wird, muss das CoDeSys-Programm auf eventuelle Fehler geprüft und gegebenenfalls korrigiert werden.

Anschließend wird die Reaktion auf Störungen während des Entleerungsvorgangs untersucht. Die Vorgabe ist eine Unterbrechung der angefangenen Schrittkette und eine weiterführende Entleerung über die nächst mögliche Schrittkette.

5.1 Ablauf der Schrittketten

5.1.1 Schrittkette 1

Der Pegel wird oberhalb des Schaltpunktes zur „großen“ Drehzahl gewählt, damit die gesamte Schrittkette zur Entleerung ablaufen kann. Zuerst erfolgt die Überprüfung des automatischen Anlaufs. Dazu wird der Pegel in der Simulation zwei Stunden nicht verändert und die Entwässerung nicht manuell gestartet. Somit ist sichergestellt, dass die Schrittkette 1 bis zur Entleerung des Speicherbeckens abgearbeitet wird.

Die Schrittkette wird mehrmals durchlaufen und in der Abarbeitung der einzelnen Schritte können keine Fehler festgestellt werden. Es zeigte sich, dass das alternierende Schalten der Pumpen zur gleichmäßigen Nutzung innerhalb der Testumgebung ebenfalls funktionierte.

5.1.2 Schrittkette 2

Zu Beginn wird in der Simulation eine Störung für die Hauptentleerungspumpe PU01 ausgelöst in dem das Signal des Motorschutzes f_MS gefordert wird. Ziel ist es mit der Störung die Abarbeitung der Schrittkette 2 zu starten.

Der Versuchsablauf hat ergeben, dass nicht wie erwartet Schrittkette 2 sondern Schrittkette 3 abgearbeitet wird. Um das Problem zu analysieren wird der Programmabschnitt in dem der Fehler ausgelöst wird im Online-Modus untersucht. Dabei wird ein Programmfehler festgestellt (Abbildung 5.1).

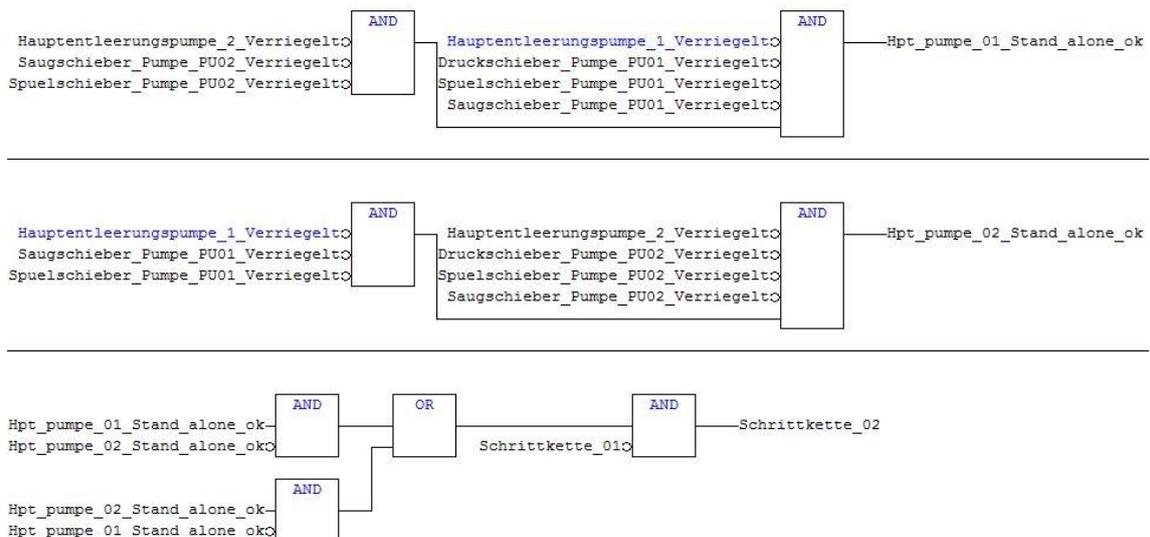


Abbildung 5.1: Fehler im Steuerungsprogramm zum Starten der Schrittkette 2 [eigene Darstellung]

Durch die Negierung und dem UND-Baustein wird die Variable Hpt.OX_standalone_OK nur gesetzt, wenn beide Hauptpumpen und deren Schieber störungsfrei laufen. Die Variable Schrittkette 2 kann folglich bei Störung der Hauptpumpen oder Schieber nicht gesetzt werden. Indem die Negierung entfernt und der AND-Baustein durch einen OR-Baustein ersetzt wurde konnte der Fehler behoben werden.

Die Entleerung in Schrittkette 2 startet nun erfolgreich bei Störung einer Pumpe oder den zugehörigen Schiebern

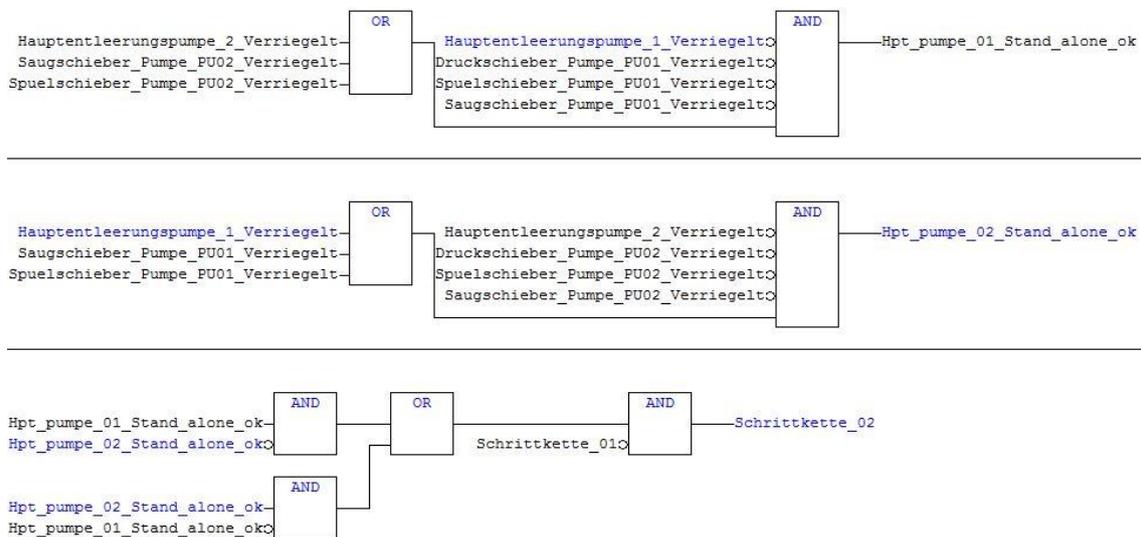


Abbildung 5.2: Optimiertes Steuerungsprogramm der Schrittkette 2 [eigene Darstellung]

5.1.3 Schrittkette 3

Der Test der dritten Schrittkette erfolgte, indem in der Simulation die Störung f_MS (Motorschutz) beider Hauptentleerungspumpen ausgelöst wurde. Durch die Änderung des Ablaufprogramms zur Schrittkette 2 traten keine Probleme auf und die Schrittkette wurde korrekt beendet.

5.2 Störungssimulation

Sofern eine Störung vor Beginn der Entleerung des Speicherbeckens auftritt, kann durch anpassen des Steuerungsprogramms der Ablauf der Schrittketten gewährleistet werden. In den meisten Fällen tritt die Störung allerdings während des Ablaufs der Schritte innerhalb der Schrittkette auf. Das Unterbrechen einer Schrittkette während des Durchlaufs wird mit Hilfe der WinMOD-Simulation und der Fehlerprovokation des Motorschutzes (f_MS) der Hauptpumpe durchgeführt. Eigentlich müsste nach Ablauf einer definierten Zeit, hier zehn Minuten die Schrittkette stoppen und in den Schritt 99 *Störung* wechseln. Anhand einiger Versuche konnte der Schrittwechsel nur teilweise bestätigt werden. In anderen Fällen lief der aktuelle Schritt weiter, bis er manuell unterbrochen wurde. Wie bereits in Abschnitt 4.2 beschrieben, soll die automatische Entleerung mit Hilfe der nächst möglichen Schrittkette durchgeführt werden. Allerdings ist für ein Fortfahren des gestörten Entleerungsvorgangs

eine manuelle Quittierung erforderlich. Demzufolge ist die automatische Speicherentleerung nach Störung fehlerhaft. Betrachtet man das Steuerungsprogramm wird deutlich das bei fehlerhaften Schiebern, die Verriegelung der Pumpe angesprochen wird und nach der definierten Zeit von zehn Minuten die Störung ausgelöst wird. Ein Blick auf den Baustein zum Störungsschritt 99 bei der Hauptpumpe PU01 macht deutlich das die Variable *Hauptentleerungspumpe_1_keine_Laufmeldung* gesetzt sein muss. Diese Variable wird nur ausgelöst, wenn im zugehörigen *Antriebsmodul_Schnell_Langsam* die Variable KLM_S geschaltet wird. In der Überwachung der Laufmeldung wird deutlich warum der Schrittwechsel nicht vollzogen wird. Damit KLM_S auf True gesetzt wird, muss auch das Ausgangssignal zum Einschalten der Pumpe geschaltet sein. Sobald eine Störung forciert wird und das Eingangssignal die SPS erreicht, entfällt der Einschaltbefehl der Pumpe. Somit steckt das Ablaufprogramm in vorliegenden Schritt fest. Wird für die simulierte Störung keine Laufmeldung der Pumpe benötigt, zum Beispiel beim auslösen des Motorschutzes f_{MS} , fehlt die Weiterverarbeitung zum Störungsschritt 99.

Ein Abbruch der Entwässerung in dem der Mischwassersiel-Pegel im Trennbauwerk über einen Wert von 6,80 mNN gesetzt wird, führt in jeder Schrittkette erfolgreich zum Abbruch der Entwässerung. Sämtliche Geräte fahren in Grundstellung, Schritt 0.

5.3 Optimierungsansätze

Neben den bereits durchgeführten und validierten Lösungen zum fehlerfreien Ablauf der Schrittketten muss der Ablauf der Entleerungs-Schrittkette auch bei aufkommender Störung während eines Entleerungsvorgangs gewährleistet sein. Statt in den manuell quittierpflichtigen Schritt 99 zu schalten sollte äquivalent zum Abbruch durch das Überschreiten des Mischwassersiel-Pegels zurück in die Grundstellung gefahren werden. Nach erneutem Anlauf läuft die nächstmögliche Schrittkette ab. Die Störung bleibt dabei quittier pflichtig. Des Weiteren ist über einen Verzicht auf Laufmeldungen, durch das Stromsignal, nachzudenken. Ein einfaches binäres Rückmeldungssignal ist hier sinnvoller. Beim Ausbleiben des Signals kann eine Störung abgegeben werden und es wird in die Grundstellung gefahren. Auch sollten in Schritt 1 lediglich die Schieber der genutzten Entleerungspumpe öffnen und nicht auch für den späteren Verlauf genutzte Schieber wie Saug- und Druckschieber der Restentleerungspumpen. Die Volumenberechnung hat keinen Einfluss auf die Entwässerung, jedoch zeigt sich das diese fehlerhaft im Steuerungsprogramm implementiert ist. Unterhalb der 7,20 mWS wird nicht beachtet, dass bei der Verwendung der Volumenformel des Kegelstumpfes neben der Höhe auch der Radius angepasst werden muss. Zur Optimierung kann die hier in Abschnitt 3.3 vorgestellte Berechnung des Kegelstumpfes genutzt werden.

Ein weiteres Problem ist softwaretechnisch nicht zu lösen, die ABB SPS ermöglicht nur eine Fernwirkübertragung über einen Weg. Ein benötigter zweiter Weg wird nicht unterstützt. Wird ein Zweitweg dazu geschaltet und über eine der zwei IPs eine Anfrage an die SPS in der Schädlerstraße gesendet, wird die Antwort nicht an die sendende IP adressiert, sondern an die Andere. Das System erkennt das als Unterbrechung der Kommunikation und startet eine Generalabfrage. Daraus ergibt sich eine Endlosschleife von Kommunikationsunterbrechung und Generalabfrage, da die Antwort immer an die falsche IP geschickt wird. Eine Lösung dieses Fehlers ist nur über den Austausch der SPS möglich.

6 Diskussion und Fazit

Das im Rahmen dieser Arbeit realisierte WinMOD-Projekt ermöglicht den Funktionstest des Steuerungsprogramms des MRBs Schädlerstraße Vorort sowie über eine dafür eingerichtete externe Test-SPS. Die Modellierung der Optimierungsansätze wird mit Hilfe von WinMOD ermöglicht. Durch das Forcen von Signalen, wie dem Setzen des Pegels oder das Erzwingen von Störungen über die Fehlerprovokationen, kann der Einfluss direkt im Steuerungsprogramm und dessen Visualisierung abgelesen werden. Das Ansteuern der Signale aller Komponenten per Handsteuerung Vorort, sowie das Testen des störungsfreien Ablaufs der einzelnen Schrittketten zeigen, dass das erstellte WinMOD-Simulationsmodell geeignet zur Bewertung und Optimierung des implementierten Steuerungsprogrammes ist. Die approximierten Komponenten und Flussmodelle sind idealisierte Nachbildungen und decken den nötigen Detaillierungsgrad ab.

Bei der Nachbildung des Flussmodells sind mehrere Möglichkeiten zur Implementierung in Betracht gezogen worden. Da Zu- und Abfluss nicht gleichzeitig ablaufen können, wurde sich für ein einfaches pegel- und zeitabhängiges Modell entschieden. Dadurch wird ein direktes manipulieren der Pegel und der Entleerungszeit ermöglicht. Die Entwässerungszeit wurde allgemein festgesetzt und so gewählt, dass einzelne Schritte einer Schrittkette nicht übersprungen werden.

Die Simulation ist durch die Makrobibliotheken und die Engineering-Listen so individuell anpassbar, dass auch zukünftig bei Änderung der Komponenten, durch anpassen der Parameter eine Simulation noch möglich ist.

Es zeigte sich recht schnell, dass das Programm sehr unübersichtlich gestaltet ist. Einzelne Optimierungen bzw. Ausbesserungen von Fehlern ermöglichen zwar den störungsfreien Ablauf des Schrittprogramms, allerdings durch das Abfragen einzelner zusätzlicher Signale zu keiner zeitlich ansprechender Lösung führt.

Das Stromsignal soll eigentlich nur zur Auswertung von Störungen dienen. Das Steuerungsprogramm benötigt das Stromsignal nicht nur für die Visualisierung, sondern setzt auch einen minimalen bzw. maximalen Eingangsstrom voraus für den Programmablauf. Dies führt bei Störungen der Pumpen zum fehlerhaften Ablauf der Schrittketten. Die Laufmeldung der Pumpe ist gekoppelt an den analogen Stromwert und das Ausgangssignal. Bei Störung der Pumpe entfällt das Ausgangssignal und das Steuerungsprogramm kann nicht weiterverarbeitet werden. Um eine Verbesserung des Steuerungsprogramms gegenüber den unscharfen Zustandsbedingungen zu gewährleisten müssten die Antriebsmodule sämtlicher Geräte angepasst werden. Eine Verarbeitung des Stromsignals ist nur bei Änderung des Anlaufverhaltens durch das Verwenden von Frequenzumrichtern sinnvoll. Des Weiteren sollte eine manuelle Quittierung zur Neuabfrage der jeweiligen Schrittkette entfernt und

lediglich das Alarmsignal an die Leitzentrale durchgeschaltet werden. Auch hierbei müssten alle Schrittketten angepasst werden.

Auch zeigte sich, dass ein Teil der Dokumentationen fehlerhaft war. So gibt es zum Beispiel keine eindeutige Benennung der Messgeräte zur Pegelbestimmung. Auch stimmen die Signalzustände nicht mit dem Stromlaufplan überein. Durch die Vorgabe die Ein- und Ausgangsmodule nur mit den Signalen jeweils eines Gerätes zu konfigurieren, wird die Signalliste unübersichtlich. Weniger als ein Drittel der Signale dient der Steuerung des MRBs.

In Folge des doch sehr verschachtelten Steuerungsprogramms und unnötiger Voraussetzungen zum Ablauf der Entwässerung sollte von einem Überarbeiten des Steuerungsprogramms abgesehen werden. Durch den hohen zeitlichen Aufwand zur Optimierung des Steuerungsprogramms sollte ein neues implementiert werden. Ferner sollte wegen der angesprochenen Problematik der Fernwirkübertragung über zwei Wege, ein Austausch der SPS angestrebt werden.

7 Ausblick

Grundsätzlich ist eine weitere Überarbeitung des Steuerungsprogramms in der Schädlerstraße nötig, zumindest die in Abschnitt 4.4 erwähnten Optimierungsansätze sollten in das Steuerungsprogramm übernommen werden um den Ablauf der Entleerung des MRB zu gewährleisten. Bei Implementierung eines neuen Steuerungsprogrammes kann dieses mit Hilfe der erstellten WinMod-Simulationsdateien und Makrobibliotheken getestet werden. Die Engineering-Liste zum Operanden-Import muss gegebenenfalls bei Austausch der SPS an den neuen Treiber angepasst werden. Gerade in Anbetracht dessen, dass in näherer Zukunft auch bei anderen MRB ein Austausch der Elektronik notwendig wird wäre ein auf einen Standard geschriebenes Steuerungsprogramm sinnvoll. Der Ablauf der Schrittketten könnte beibehalten werden. Da das Simulationsprogramm möglichst allgemein und variabel gestaltet wurde kann dieses nicht nur für das MRB Schädlerstraße genutzt, sondern auch für andere MRB genutzt werden.

Literaturverzeichnis

- [1] HAMBURG WASSER: *Umwelterklärung 2016*. Website 2017- Online erhältlich unter <https://www.hamburgwasser.de/privatkunden/unser-wasser/umweltschutz/umwelterklaerung/> abgerufen am 22.02.2018
- [2] hamburg.de: *Beseitigung von kommunalem Abwasser in Hamburg, Lagebericht 2005/2006*. Website 2017- Online erhältlich unter <http://www.hamburg.de/abwasser-downloads/> abgerufen am 27.02.2018
- [3] Mewes & Partner GmbH: *WinMOD-Systemplattform Version 7.2*. Website. 2017. – Online erhältlich unter <http://www.winmod.de/de/index.php?page=winmod-setup> abgerufen am 01.02.2018
- [4] Mewes & Partner GmbH: *Prospekt_WinMOD-Systemsoftware*. Website. 2015. – Online erhältlich unter http://www.winmod.de/de/uploads/Prospekt_WinMOD-Systemsoftware_de.pdf abgerufen am 30.09.2017
- [5] Mewes & Partner GmbH: *Handbuch WinMOD-Konfiguration A740*. Website. 2017. – Online erhältlich unter <http://www.winmod.de/de/index.php?page=winmod-handbuecher> abgerufen am 05.10.2017
- [6] Mewes & Partner GmbH: *WinMOD_Handbuch_EngineeringAssistenz_de*. Website. 2017. – Online erhältlich unter <http://www.winmod.de/de/index.php?page=winmod-handbuecher> abgerufen am 22.01.2018
- [7] HAMBURG WASSER: *6_3_Bediengerätbilder*. Unveröffentlichtes Dokument. 2015
- [8] Mewes & Partner GmbH: *WinMOD-Systemplattform Version 7.0*. Website. 2011. – Online erhältlich unter <http://www.winmod.de/de/index.php?page=winmod-setup> abgerufen am 05.10.2017
- [9] HAMBURG WASSER: *6_1_Automatisierungsstruktur*. Unveröffentlichtes Dokument. 2015
- [10] Berger, Hans: *Automatisieren mit SIMATIC*. Publicis MCD Verlag, 2000. – ISBN 3-89578-132-0
- [11] Popp, Manfred: *PROFIBUS-DP/DPV1*. Hüthig GmbH, 2000. – ISBN 3-7785-2781-9
- [12] Wellenreuther, Günter; Zastrow, Dieter: *Automatisieren mit SPS-Theorie und Praxis*. Vieweg+Teubner, 2011. – ISBN 978-3-8348-1504-0
- [13] Warncke, Björn: *Praktikumsbericht zur Einrichtung einer dezentralen Steuereinheit als Vorbereitung zum Testen eines Steuerungsprogramms*. Unveröffentlichtes Dokument. 2017
- [14] ABB: *AI561 Analoges Eingangsmodul*. Website 2012. Online erhältlich unter <http://new.abb.com/products/1TNE968902R1101/ai561s500-analog-input-mod-4ai-u-1l2bitsign-24vdc-2-wire> abgerufen am 22.01.2018
- [15] Mewes & Partner GmbH: *WinMOD_Einfuehrung_Prozess_de*. Website. 2014. – Online erhältlich unter <http://www.winmod.de/de/index.php?page=winmod-setup> abgerufen am 05.10.2017
- [16] HAMBURG WASSER: *Funktionsbeschreibung Schädlerstr. Unveröffentlichtes Dokument. 2015*
- [17] HAMBURG WASSER: *Dokumentation_HSE_MRB_Schädlerstr. Unveröffentlichtes Dokument. 2015*
- [18] HAMBURG WASSER: *Objektschlüsselnummernplan*. Betriebsanleitung MRB Schädlerstraße. 1993
- [19] HAMBURG WASSER: *Umbau MRB-Schädlerstraße*. Technische Zeichnungen. 2013

Versicherung über die Selbstständigkeit

Ich versichere hiermit, dass ich die vorliegende Bachelorarbeit mit dem Thema „Modellierung und Entwicklung von Optimierungsansätzen zur Verbesserung der Robustheit des Steuerungsprogramms eines Mischwasserrückhaltebeckens (MRB) gegenüber unscharfen Zustandsbedingungen“ ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Ort, Datum

Unterschrift

A Anhang

Der Anhang dieser Bachelorarbeit befindet sich auf der beigefügten CD und ist beim Erstgutachter Prof. Dr.-Ing. Marc Hölling oder dem Zweitgutachter Dipl.-Ing. Lars Ebert einzusehen.

- A 1. 01_AKS: AKS_14Stufe_V10.pdf
- A 2. 01_AKS: AKS_5Stufe.xls
- A 3. 02_EngineeringListe: Engineering_Schädlerstraße.xlsx
- A 4. 03_Stromlaufplan: EPLAN_Pumpwerk_Typ1.pdf
- A 5. 04_Berechnung: Volumenberechnung
- A 6. 05_WinMOD: WinMOD_Mischwasserrückhaltebecken