

BACHELORTHESIS
Christopher Schade

Entwicklung einer integrativen Automatisierungslösung für den Betrieb von Grundwasserbrunnen im Bestandsnetz, sowie standardisierter Softwarefunktionen

FAKULTÄT TECHNIK UND INFORMATIK
Department Informations- und Elektrotechnik

Faculty of Engineering and Computer Science
Department of Information and Electrical Engineering

Christopher Schade

Entwicklung einer integrativen
Automatisierungslösung für den Betrieb von
Grundwasserbrunnen im Bestandsnetz, sowie
standardisierter Softwarefunktionen

Bachelorthesis eingereicht im Rahmen der Bachelorprüfung
im Studiengang Informations- und Elektrotechnik
am Department Informations- und Elektrotechnik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. Frerk Haase
Zweitgutachter : Christopher Krull

Abgegeben am 18. Februar 2022

Thema der Bachelorthesis

Entwicklung einer integrativen Automatisierungslösung für den Betrieb von Grundwasserbrunnen im Bestandsnetz, sowie standardisierter Softwarefunktionen

Stichworte

Automatisierung, SPS, TIA-Portal, objektorientierte Steuerungen, Objektmodell, standardisierte Softwarefunktionen, HMI, WinCC, Trinkwasserversorgung, Grundwasserbrunnen, Lübeck, TraveNetz GmbH

Kurzzusammenfassung

Diese Arbeit umfasst die Neuentwicklung einer Brunnensteuerung, die mittels zu entwickelnder Softwarefunktionen umgesetzt wird. Dabei wird die Software auf die Integrierbarkeit in die Bestandsanlage optimiert. Die Programmierung erfolgt nach aktuellen Programmierstandards der SPS-Programmierung unter Erstellung und Nutzung von Objektmodellen. Der verfahrenstechnische Prozess wird über ein Human Machine Interface visualisiert. Ziel ist es eine Steuerung auf Grundlage standardisierter Programmbausteine zu schaffen, mit der intuitiv, von mehreren Nutzerebenen aus, auf den Brunnen zugegriffen werden kann. Dabei soll das Programm so aufgebaut sein, dass es für weitere Projekte mit wenigen Anpassungen übernommen werden kann.

Title of the paper

Development of an integrative automation solution for the operation of groundwater wells in the existing network, as well as standardized software functions

Keywords

Automation, PLC, TIA-Portal, object-oriented programming, objectmodel, standardized software functions, HMI, WinCC, drinking water supply, groundwater wells, Lübeck, TraveNetz GmbH

Abstract

This thesis includes the development of a well control by means of software functions to be developed. The software is optimized for integration in the existing installation. The programming takes place according to current programming standards of PLC-programming under creation and usage of objectmodels. The process is visualized by a Human Machine Interface. The aim is to create a well control by using standardized program modules, that allows a intuitive access to the well by several user levels. The program should be structured in a way that it can be adopted for further projects with just a few adjustments.

Inhaltsverzeichnis

Abbildungsverzeichnis	iii
Tabellenverzeichnis	iv
Abkürzungsverzeichnis	v
1 Einleitung	1
1.1 Unternehmen	1
1.2 Zielsetzung	1
1.3 Aufbau der Arbeit	1
2 Grundlagen	3
2.1 Trinkwasserversorgung	3
2.1.1 Wasserversorgungsnetz	3
2.1.2 Reinwasserbehälter und -pumpen	3
2.1.3 Filteranlagen	5
2.1.4 Grundwasserbrunnen	5
2.2 Automatisierungstechnik	6
2.2.1 Speicherprogrammierbare Steuerung S7-1500	6
2.2.2 Analogwertverarbeitung	8
2.2.3 Industrielle Kommunikation über Profinet	10
2.2.4 Betrieb von Kommunikationsstrecken über DSL und SHDSL	10
2.2.5 Frequenzumrichter	11
2.3 Messtechnik	14
2.3.1 Messwerverfassung Durchfluss	14
2.3.2 Messwerverfassung Druck	15
2.3.3 Messwerverfassung Leitwert	16
2.3.4 Messwertübertragung	17
2.4 Regelungstechnik	18
2.5 SPS-Programmierung	20
2.5.1 Programmbausteine	20
2.5.2 Regelung kontinuierlicher Prozesse	21
2.6 Human Machine Interface	22
2.6.1 Siemens Comfort Panel	22
2.6.2 WinCC Comfort	22
2.6.3 Datenaustausch mit SPS	23
3 Projektplanung	24
4 Bestandsaufnahme	25
4.1 Rohrleitungs- und Instrumentenfließbild des Brunnens	25
4.2 Aufnahme der Messtechnik	26
4.3 Aufnahme weiterer Signale aus dem Feld	26
4.4 Aufnahme der Bestandsvisualisierung	27
4.5 Aufbau des Kommunikationsnetzes	28
4.6 Zugriffsstrukturen auf die Steuerung des Brunnens	29
4.7 Datenaustausch zwischen Brunnen und Werk	29
5 Konzeptionierung und Umsetzung	31
5.1 Entwicklung eines Objektmodells	31
5.1.1 Analogwertverarbeitung	31

5.1.2 Digitalwertverarbeitung	36
5.1.3 Zähler	38
5.1.4 Frequenzumrichter mit Direktansteuerung	40
5.1.5 Datenmodell	43
5.2 Entwicklung eines Visualisierungskonzepts	44
5.2.1 Farbgebung	44
5.2.2 Bilder	45
5.2.3 Objekt-Popups	48
5.2.4 Adressmultiplexing	51
5.3 Brunnenspezifische Funktionen	52
5.3.1 Freigaben und Grenzen	52
5.3.2 Berechnung von Abstich und Pegel	53
5.3.3 Berechnung von Ruhepegel und Absenkung	55
5.3.4 Berechnung des Pumpenwirkungsgrads	57
5.3.5 Durchflussregelung	58
6 Modultests	61
6.1 Erstellung von Testprotokollen	61
6.2 Vorgehensweise	61
7 Zusammenfassung	65
8 Ausblick	66
Literaturverzeichnis	67
Anhang	70
Eidesstattliche Erklärung	71

Abbildungsverzeichnis

1	Wassergewinnung nach Wasserarten 2016	3
2	Charakteristischer Verlauf des Tagesverbrauchs	4
3	Übersicht Brunnentechnik	6
4	CPU 1510-1 PN mit Ein- und Ausgabebaugruppen	7
5	Zyklische Programmabarbeitung	7
6	Mengenabbildungen für Skalierung und Normierung von Analogsignalen	9
7	Linien- und Ringstruktur mit Ethernet-Extendern	11
8	Schematische Darstellung eines Frequenzumrichters als Blockschaltbild	11
9	Vereinfachter Aufbau eines dreiphasigen Wechselrichters	12
10	Strangspannungen einer symmetrischen Last bei Grundfrequenztaktung	12
11	Typische Anschlussklemmen von Frequenzumrichtern	13
12	Prinzip der magnetisch-induktiven Durchflussmessung	14
13	Aufbau Drucksensor mit metallischer Membran	15
14	Aufbau Leitwertsensor für die induktive Leitwertmessung	16
15	Anschluss von 2-Draht-Messumformern an Analogeingangsgruppen	17
16	Anschluss von 4-Draht-Messumformern an Analogeingangsgruppen	17
17	Standardregelkreis mit Rückführung der Regelgröße	18
18	Vereinfachte Veranschaulichung des präemptiven Multitaskings	21
19	SIMATIC HMI TP700 Comfort	22
20	Server-Client-Prinzip für SPS und HMI	23
21	RI-Fließbild der Brunnenanlage	25
22	Bilder der Bestandsvisualisierung	27
23	Vereinfachte Netzwerkübersicht	28
24	Übersicht über die Steuerungsebenen	29
25	Put-Get-Kommunikationsbausteine	30
26	Beispielhafte Darstellung einer Alarmparametrierung	32
27	Aufruf und Verschaltung des Analogwertverarbeitungs-FC	35
28	Aufruf und Verschaltung des Digitalwertverarbeitungs-FC	38
29	FUP-Netzwerk zum Setzen und Rücksetzen des Mitternachtbits	39
30	Struktureller Aufbau der erstellten Klassen	44
31	Anlagenübersicht in der Prozessvisualisierung	46
32	Kennwertbild in der Prozessvisualisierung	47
33	Energie- und Leistungsdaten in der Prozessvisualisierung	47
34	Bilder zur Datenverwaltung in der Prozessvisualisierung	48
35	Messwert-Popups	49
36	Popups zur Einflussnahme auf die Förderung	50
37	Sicherheitsabfrage für das Resetten der Betriebsstunden	50
38	Adressmultiplexing	51
39	Veranschaulichung der relevanten Größen für den Trockenlauf	55
40	Förderhöhe	57
41	Beispiel: Testfälle und Akzeptanzkriterien	61

Tabellenverzeichnis

1	Signalwerte bei Umwandlung eines analogen Stromsignals I_E [Sei15, S. 36]	8
2	Auflistung der verbauten Messtechnik	26
3	Wichtige Signale aus dem Feld	26
4	Bausteinschnittstelle FC Analogwertverarbeitung	33
5	Bausteinschnittstelle FC Digitalwertverarbeitung	36
6	Wahrheitstabelle XOR	36
7	Bausteinschnittstelle FC Impulszähler	38
8	Bausteinschnittstelle FC FU-Ansteuerung	40
9	Bausteinschnittstelle FC FU-Freigabe	52
10	Bausteinschnittstelle FC Berechnung Abstich/Pegel	53
11	Bausteinschnittstelle FC Berechnung Ruhepegel/Absenkung	56
12	Bausteinschnittstelle FC Berechnung Pumpenwirkungsgrad	57
13	Bausteinschnittstelle FC Regler	59
14	Testfälle Messwertskalierung	63
15	Testfälle Istwertskalierung FU	63

Abkürzungsverzeichnis

AI	Analogeingang
AQ	Analogausgang
ASM	Asynchronmaschine
AWL	Anweisungsliste
CPU	Central Processing Unit
DB	Datenbaustein
DI	Digitaleingang
DQ	Digitalausgang
E+H	Endress und Hauser Messtechnik GmbH + Co.KG
E/A-Felder	Ein- und Ausgabefelder
FB	Funktionsbaustein
FC	Anwenderfunktion einer SPS
FU	Frequenzumrichter
FUP	Funktionsplan
HMI	Human Machine Interface
IDB	Instanzenbaustein
KOP	Kontaktplan
N.N.	Normalnull
NLS	Netzleitstelle
OB	Organisationsbaustein
PLC	Programmable Logic Controller
Siemens	Siemens AG Digital Industries
SPS	Speicherprogrammierbare Steuerung
ST/SCL	Strukturierter Text bzw. Structured Control Language
SWL	Stadtwerke Lübeck
TIA-Portal	Totally Integrated Automation Portal
Travenetz	TraveNetz GmbH
U-Pumpe	Unterwasserpumpe
UDT	User Defined Data Type - Benutzerdefinierter Datentyp
WW	Wasserwerk

1 Einleitung

Diese Bachelorarbeit beschäftigt sich mit der Entwicklung einer Automatisierungslösung für die Steuerung von Grundwasserbrunnen. Im Zuge geplanter Modernisierungsmaßnahmen soll die Software der Brunnensteuerung erneuert werden. Durch die Entwicklung standardisierter Softwarefunktionen soll die Möglichkeit geschaffen werden, eine unternehmenseigene Software-Bibliothek für Automatisierungsprojekte anzulegen.

1.1 Unternehmen

Im Rahmen eines dualen Studiums wurde in Zusammenarbeit mit der TraveNetz GmbH diese Bachelorarbeit erstellt.

Die TraveNetz GmbH ist eine Tochter der Stadtwerke Lübeck GmbH und der Schleswig-Holstein Netz AG.[Tra21b] Als Versorgungsunternehmen in den Sparten Strom, Wasser, Gas und Fernwärme stellt sie wichtige Versorgungsinfrastruktur für den Wirtschaftsraum Lübeck und Umland. Das Versorgungsnetz der Sparten erstreckt sich auf eine Gesamtlänge von 6560 km. [Tra21a]

1.2 Zielsetzung

Ziel dieser Arbeit ist die Entwicklung einer funktionsfähigen Software zur Steuerung von Grundwasserbrunnen. Diese soll unter objektorientierten Gesichtspunkten erstellt werden. Das Programm soll auf Grundlage bibliotheksfähiger, standardisierter Softwarefunktionen aufgebaut sein. Ein besonderes Augenmerk liegt auf den Programmbausteinen zur analogen Messwertverarbeitung, der Ansteuerung von Frequenzumrichtern und dem Handling digitaler Eingangssignale. Für diese Funktionen sollen Schnittstellen zur Prozessvisualisierung geschaffen werden.

Die Standardbausteine sollen in der Hochsprache SCL programmiert werden.

Weiterhin besteht das Ziel, ein benutzerfreundliches Human Machine Interface zu entwerfen, das auf dem Rohr- und Instrumentenfließbild der Anlage basiert und über Popups strukturiert und organisiert wird.

Die Software soll so organisiert sein, dass sie nahtlos in die Bestandsanlage integrierbar ist.

1.3 Aufbau der Arbeit

Für die zielführende Entwicklung der Software ist ein umfassendes Grundlagenwissen erforderlich. Vor allem im Bezug auf Trinkwasserversorgung und Automatisierungstechnik sind einige Erläuterungen notwendig. Weiterhin wird auf die Funktionsweise von Frequenzumrichtern eingegangen und typische Eigenschaften dieser Geräte erläutert. Weitere Ausführungen betreffen die industrielle Kommunikation über Profinet und die Messtechnik.

Weiterführend werden Maßnahmen des angewandten Projektmanagements behandelt. Dabei liegt das Hauptaugenmerk auf der Festlegung und der Organisation von Arbeitspaketen und Meilensteinen.

Im Anschluss wird auf die Bestandsanlagen eingegangen. Aus der Bestandsaufnahme folgen die Anforderungen an die Integrierbarkeit der zu erstellenden Software. Sie ermöglicht ein tiefgreifenderes technisches Verständnis für die Funktionsweise und Abläufe der Grundwasserförderung und der beteiligten Anlagen.

Die Konzeptionierung umfasst die Umsetzung des Projektes auf Grundlage der Anforderungen, die an die Software gestellt werden und unter Beachtung der Voraussetzungen, die aus der Bestandsaufnahme folgten.

Um die Funktion des Programms testen und die Einsatzfähigkeit im Feld beurteilen zu können, werden im Anschluss an die Umsetzung und Konzeptionierung Modultests durchgeführt. Der Rahmen dieser Arbeit ermöglicht es nicht, alle Modultests detailliert aufzuführen zu können. Um dennoch diesen Teil des Projekts beleuchten zu können, wird die Erstellung von Testprotokollen und die Vorgehensweise bei Modultests beschrieben.

Auswertung und Fazit sollen die Ergebnisse des Projektes beurteilen.

Zum Abschluss soll ein Ausblick mögliche Erweiterungsoptionen aufzeigen.

2 Grundlagen

2.1 Trinkwasserversorgung

Die wichtigste Trinkwasserressource in Deutschland ist das Grundwasser. Im Jahr 2016 wurden rund 70 % des Trinkwassers aus Grundwasserreserven und Quellwasser gefördert.[Umw21]

Wassergewinnung nach Wasserarten 2016 *

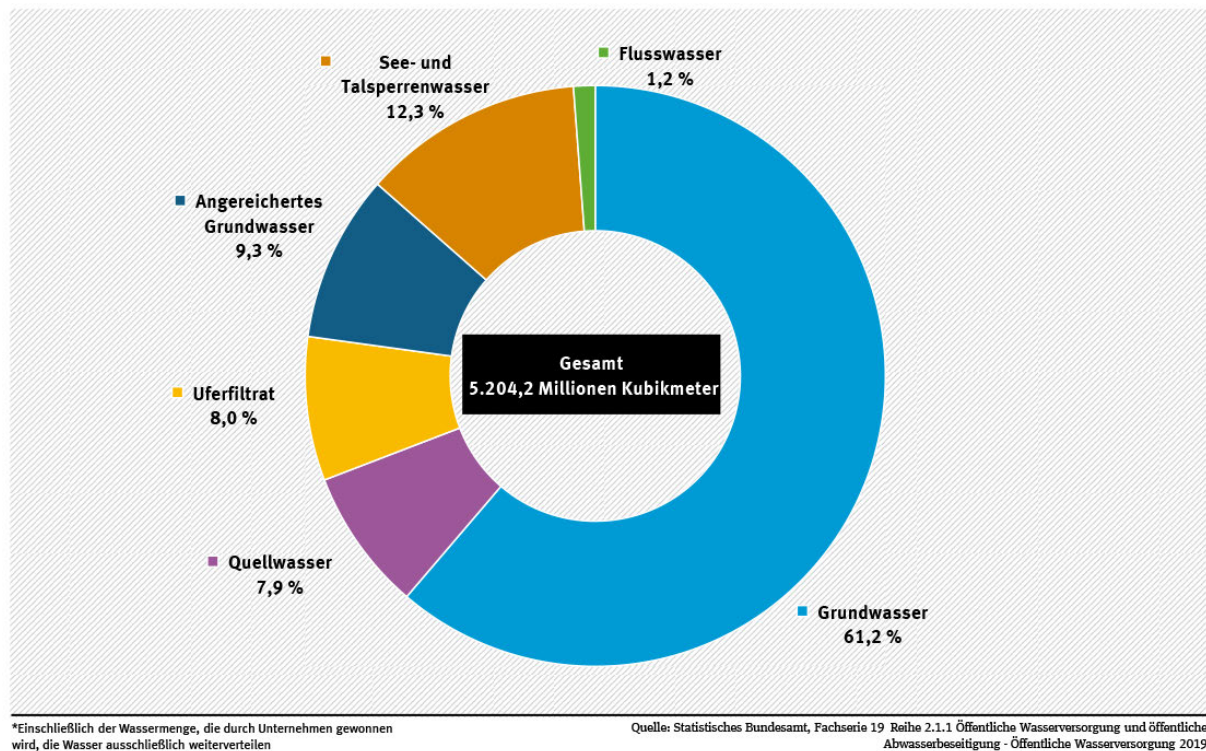


Abbildung 1: Wassergewinnung nach Wasserarten 2016 [Umw21]

In Schleswig-Holstein erfolgt die Wassergewinnung ausschließlich über die Entnahme von Grundwasser. [Lan21]

Über Rohrleitungen wird das geförderte Rohwasser¹ in umliegende Wasserwerke transportiert, wo dieses zu Reinwasser² aufbereitet wird. Von dort aus wird das Versorgungsnetz gespeist.

2.1.1 Wasserversorgungsnetz

Das größtenteils unterirdische Versorgungsnetz der Sparte Wasser umfasst eine Ausdehnung von 743 km und versorgt täglich etwa 200.000 Personen im Großraum Lübeck. [Tra21a]

Die vier Wasserwerke Klein Disnack, Kleinensee, Vorwerk und Geschendorf stellen die Trinkwasserversorgung mit jährlich mehr als 120 Millionen Kubikmetern Reinwasser sicher.

2.1.2 Reinwasserbehälter und -pumpen

Gespeist wird das Wasserversorgungsnetz aus den Reinwasserbehältern der Wasserwerke. Diese dienen dem Ausgleich der täglichen Verbrauchsschwankungen. Der charakteristische Tagesver-

¹Unbehandeltes Wasser

²Zu Trinkwasser aufbereitetes Wasser

brauch an Wasser kann der Abbildung 2 entnommen werden.

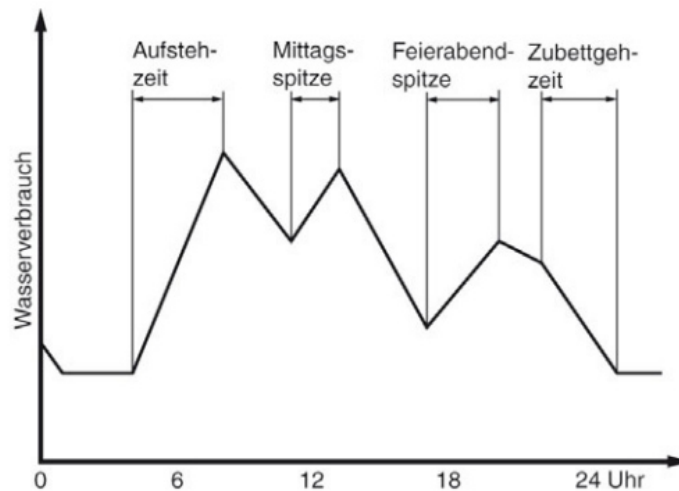


Abbildung 2: Charakteristischer Verlauf des Tagesverbrauchs [Kar13, S. 18]

Die Behälter sind in unmittelbarer Nähe zum Wasserwerk, oder Teil dessen. Unterschieden wird zwischen Hoch- und Tiefbehältern. Unter Hochbehältern sind u.a. Wassertürme zu verstehen. Diese sorgen für einen gleichmäßigen Versorgungsdruck und ermöglichen somit einen ebenso gleichmäßigen Pumpenbetrieb. Wartungs- und Instandhaltungsmaßnahmen sind im Vergleich zu Tiefbehältern allerdings aufwändiger und oftmals teurer. In Lübeck gibt es noch einen Wasserturm. Dieser ist momentan aufgrund von Wartungsarbeiten nicht in Betrieb.

Tiefbehälter, wie sie im Großraum Lübeck bevorzugt gebaut werden, haben nur geringen Einfluss auf den Versorgungsdruck. Dieser muss hier ausschließlich durch die Reinwasserpumpen erzeugt werden.[Kar13]

Die Reinwasserpumpen transportieren das Reinwasser aus den Behältern in das Versorgungsnetz. Durchgesetzt hat sich hier der Einsatz von Kreiselpumpen verschiedenster Bauformen. Das Kernstück der Kreiselpumpe ist das Laufrad, das bei Betrieb einen Sog am Saugstutzen erzeugt. Der Saugstutzen befindet sich auf der Seite des Reinwasserbehälters. Das Gegenstück, der Druckstutzen, ist netzseitig verbunden.

Die Kreiselpumpen werden durch Drehstrom-Asynchronmaschinen (kurz: ASM) über Antriebswellen betrieben. Die Leerlaufdrehzahl n berechnet sich für ASM über die Erregerfrequenz f und die typenabhängige Polpaarzahl p .

$$n = \frac{f}{p} \quad (1)$$

$$n = \frac{f \cdot 60}{p} \quad (2)$$

Mittels Gleichung 1 wird die Drehzahl der ASM und somit der Kreiselpumpe in s^{-1} ermittelt. Um die Drehzahl in min^{-1} zu erhalten, muss Gleichung 2 genutzt werden.

2.1.3 Filteranlagen

Zur Aufbereitung von Rohwasser werden Filteranlagen eingesetzt. Hier gibt es unterschiedliche Möglichkeiten der Filtration. In den Wasserwerken der SWL werden Eisen- und Manganfilter eingesetzt.

Bevor das Rohwasser die Filter durchfließt, wird es im sog. Oxidator mit Sauerstoff versetzt, welches ein Ausflocken von Eisen und Mangan bewirkt. Diese Flocken werden in den Filtern aus dem Wasser entfernt. Nach dem Prozessschritt der Filtration gilt das Wasser als Reinwasser und wird in die Reinwasserbehälter geleitet.

Um die Eisen- und Manganflocken aus den Filtern zu waschen, werden diese in regelmäßigen Abständen gespült. Das zur Spülung genutzte Wasser wird nicht in den Trinkwasserprozess zurückgeführt, sondern in Absetzbecken geleitet. Dort findet die Sedimentation der ausgespülten Anteile statt. Der sich absetzende Spülschlamm findet u.a. in Klärwerken Verwendung.

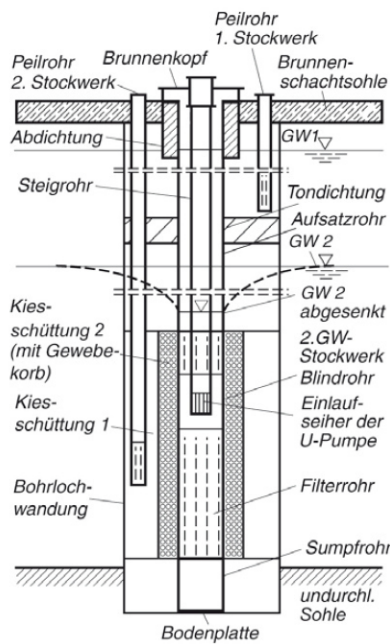
2.1.4 Grundwasserbrunnen

Die Förderung des Rohwassers geschieht durch Grundwasserbrunnen. Als Grundwasser wird nach DIN 4049-1 bis DIN 4049-3 Hohlräume ausfüllendes, unterirdisches Wasser bezeichnet. [Kar13, S. 27 f.]

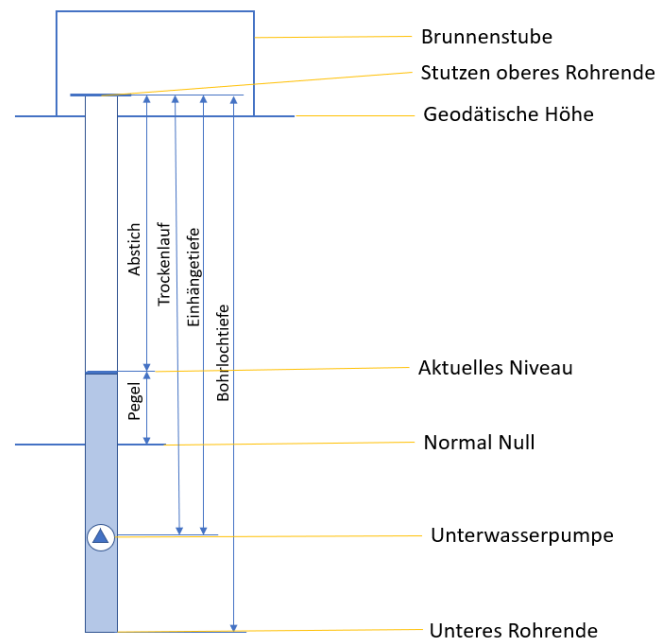
Neben der Sensorik ist das Kernstück jedes Brunnens die Unterwasserpumpe (kurz: U-Pumpe), die das Grundwasser fördert. Sie wird in ein Filterrohr eingesetzt und fördert das Grundwasser über ein Steigrohr (siehe Abbildung 3a) aus dem Brunnen heraus. Um das Trockenlaufen der Pumpe zu verhindern, wird sie in einer Tiefe von drei bis fünf Metern unterhalb des erwartbar tiefstem Wasserstands eingebaut. Die Besonderheit von U-Pumpen liegt in der konstruktiven Vereinigung von Pumpe und Antrieb.

Wichtige Höhenangaben in der Brunrentechnik sind der Pegel, der Abstich und der Trockenlauf. Der Abstich bezeichnet den Höhenunterschied zwischen dem Stutzen des oberen Rohrendes und dem aktuell anstehendem Niveau des Wasserstandes. Der Pegel gibt an, wie sich das Niveau des Wassers zu einer Reverenzebene verhält. Das Maß Normalnull (kurz: N.N.) stellt solche eine Reverenz dar und wird für die weitere Betrachtung verwendet. Pegel und Abstich ergeben zusammen die Höhendifferenz zwischen eigemessenem oberem Ende des Rohrstutzens und N.N.

Der Trockenlauf ist ein Niveau das über der Pumpe liegt. Hinter dem Begriff Trockenlauf verbirgt sich ein Grenzwert, der das Ziel verfolgt die Pumpe davor zu bewahren Luft anzusaugen. Häufig wird dieser Grenzwert so gesetzt, dass von der Einhängtiefe ein fester Wert abgezogen wird. In Abbildung 3b sind die Begrifflichkeiten visualisiert. Die eingezeichnete geodätische Höhe gibt die Höhenlage der Kreiselpumpe gegenüber eines beliebigen Bezugspunktes an. [KSB21]



(a) Aufbau von Vertikalbrunnen
[Kar13, S. 61]



(b) Höhenangaben im Brunnen

Abbildung 3: Übersicht Brunnentechnik

2.2 Automatisierungstechnik

Unter Automatisierung wird das Steuern und Regeln eines Prozesses auf Grundlage von Mess- und Stellsignalen verstanden. Das Ziel einer Automatisierung ist ein selbstständig ablaufender Prozess, bei welchem die Notwendigkeit manueller Eingriffe auf ein Minimum reduziert werden soll.

Die Automatisierungstechnik setzt sich aus den Fachbereichen der Messtechnik, der Regelungstechnik und der Steuerungstechnik zusammen.

2.2.1 Speicherprogrammierbare Steuerung S7-1500

In modernen automatisierten Anlagen sind speicherprogrammierbare Steuerungen (kurz: SPS) heute unabdingbar. Ihre Grundfunktionen sind das Steuern, Regeln und die Bereitstellung von Daten für die Visualisierung des Prozesses, sowie die Kommunikation mit anderen Geräten. Mittels ihrer programmierbaren Speicher sind schnelle und flexible Anpassungen der Anlagensteuerung möglich. Durch ihre anwenderorientierten Systeme und durch den großen Funktionsumfang, sowie der Möglichkeit eigene Funktionen zu entwickeln, haben die SPSen Relais- und Schüttschaltungen in industriellen Anlagen ersetzt.

Für die Automatisierung der Brunnenanlage wird eine SPS der Firma Siemens eingesetzt. Die CPU 1510-1 PN stammt aus der S7-1500-Serie und stellt den aktuellen Stand der Technik dar.

Die SPSen der S7-1500-Serie ermöglichen eine robuste und zuverlässige Automatisierung. Aufgrund ihrer Vielzahl an Kommunikationsschnittstellen, Technologieobjekten und Zusatzbaugruppen, sowie der umfassenden Unterstützung externer Hard- und Software durch GSD-Dateien, sind sie weltweit in industriellen Anlagen im Einsatz. Durch die Vielseitigkeit der Serie, spezi-



Abbildung 4: CPU 1510-1 PN mit Ein- und Ausgabebaugruppen [Sie21c]

ell im Hinblick auf redundante und fehlersichere Systeme, und die ausgeprägte Konnektivität, eignen sich diese auch für große und sehr komplexe Anlagen und Netze. Durch ganzheitlich projektierte Industrial Security-Konzepte der 1500er-Steuerungen werden zudem Anlagen, Systeme, Maschinen und Netzwerke umfassend vor Cyber-Angriffen geschützt. [Sie21a]

Typisch für SPSen ist die zyklische Programmabarbeitung auf Grundlage der Prozessabbilder. Als Prozessabbild wird ein Speicherbereich für die Signalzustände der Eingangs- bzw. Ausgangssignale verstanden, auf den die Steuerung zugreift. [TU 21]

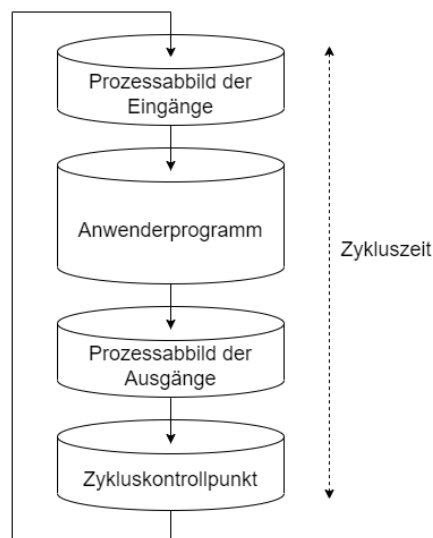


Abbildung 5: Zyklische Programmabarbeitung

Bei Hochfahren der CPU startet das Betriebssystem die Zykluszeitüberwachung. Der Ablauf jedes Zyklus ist prinzipiell gleich. Zu Beginn liest die CPU das Prozessabbild der Eingänge aus den Eingabebaugruppen ein. Anschließend wird das Anwenderprogramm ausgeführt. In diesem werden Funktionen der Reihenfolge nach abgearbeitet. Diese richtet sich nicht nach einer numerischen Priorisierung, sondern nach der Reihenfolge, in der die Funktionen aufgerufen werden. Damit folgt die Programmabarbeitung dem FIFO³-Prinzip. Nach Abarbeitung des Anwenderprogramms schreibt die CPU die Werte aus dem Prozessabbild der Ausgänge in die

³First in first out: zuerst aufgerufene Objekte werden auch als erstes abgearbeitet

Ausgabebaugruppen. Am Ende eines Zyklus werden anstehende Aufgaben wie z.B. Laden oder Löschen eines Bausteins durch die CPU vorgenommen. Danach kehrt sie zurück an den Start der Zykluszeit, startet die Zykluszeitüberwachung neu und durchläuft die Schrittkette erneut. Veranschaulicht ist dieser Prozess in Abbildung 5.

2.2.2 Analogwertverarbeitung

Analogbaugruppen ermöglichen der SPS die Verarbeitung kontinuierlicher Größen. Unterschieden werden die Baugruppen hauptsächlich nach ihrer Betriebsart als Analogeingang (kurz: AI) oder Analogausgang (kurz: AQ).

An den AI-Baugruppen können Spannungs- oder Stromsignale angelegt werden. Die Spannungssignale haben typischer Weise einen Wertebereich von $0V$ bis $10V$. Häufiger werden jedoch die Stromsignale von $4mA$ bis $20mA$ genutzt, da sie von den leitungsbedingten Spannungsabfällen nicht beeinflusst werden. [Sei15, S. 35]

Die angelegten Signale werden in der AI-Baugruppe auf einen Analog-Digital-Umsetzer geführt, der aus dem Analogsignal ein, von der CPU nutzbares, digitales Signal erzeugt. Dieses wird, um Störgrößen zu minimieren, über einen Optokoppler der CPU zugeführt. Diese bildet daraus ein Datenwort von 16 Bit Länge.

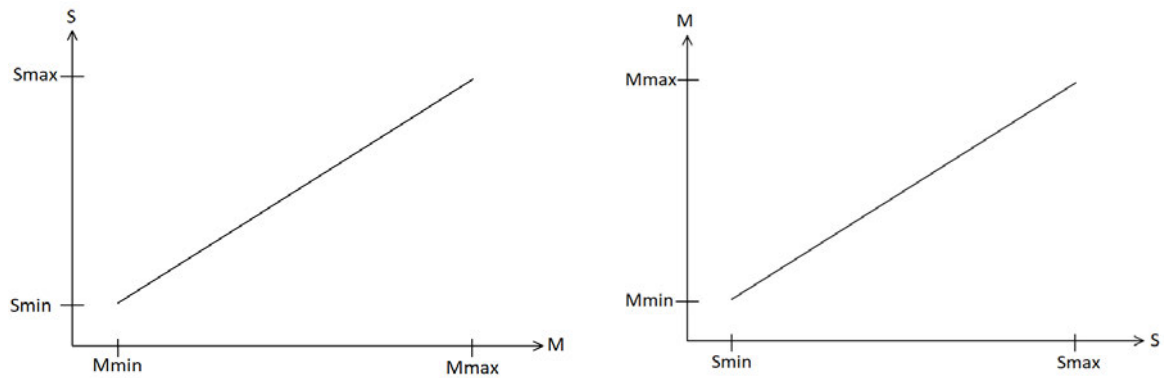
Die Datenwörter setzen sich zusammen aus 15 Datenbits mit der Wertigkeit 0 bis 14, und einem Vorzeichenbit der Wertigkeit 15.

I_E	x_W	VZ	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
[mA]	dez.																
>22,81	32767	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
22,81	32511	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
20,0005	27649	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	1
20,0	27648	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0
16,0	20736	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0
4,0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3,9995	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1,1852	-4864	1	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0
<1,852	-32768	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabelle 1: Signalwerte bei Umwandlung eines analogen Stromsignals I_E [Sei15, S. 36]

Mit Hilfe der Datenwörter können dezimale Signalwerte von -32768_{10} bis 32767_{10} abgebildet werden. Der im fehlerfreien Zustand erwartbare Wertebereich ist allerdings deutlich kleiner und liegt im Bereich von 0_{10} bis 27648_{10} , was genau dem Bereich von $4mA$ bis $20mA$ entspricht. Auch die Analogausgangs-Baugruppen (kurz: AQ-Baugruppen) nutzen diese Datenwörter. Ihre Funktionsweise ist genau umgekehrt zu der der AI-Baugruppe. Ein von der CPU erzeugtes Datenwort wird über einen Optokoppler auf einen Digital-Analog-Umsetzer gegeben, der daraus ein Analogsignal erzeugt, welches an das, an die AQ-Karte angeschlossene Gerät übermittelt wird.

Nur in seltenen Fällen sind die Signalwerte der Datenwörter aussagekräftig für die Steuerung. Sie werden viel eher dazu genutzt, auf eine physikalische Größe zu schließen. Diese können auf die Signalwerte abgebildet werden. Ebenso ist es umgekehrt möglich.



(a) Lineare Abbildung der Messwerte auf Signalwerte (b) Lineare Abbildung der Signalwerte auf Messwerte

Abbildung 6: Mengenabbildungen für Skalierung und Normierung von Analogsignalen

Zur Berechnung physikalischer Größen aus Signalwerten, werden diese skaliert und normiert. Dabei wird ein Wert x aus der Menge der physikalischen Werte M , genau auf einen Wert y aus der Menge der Signalwerte S abgebildet.

$$\forall x \in M \exists! y \in S \quad (3)$$

Diese Abbildung erfolgt zumeist linear. Die Skalierung und Normierung geschieht also durch das Aufstellen einer Geradengleichung.

Der Wert M_{min} gibt den kleinsten physikalischen Wert an. M_{max} ist folglich der größte physikalische Wert. S_{min} ist der Signalwert, auf den M_{min} abgebildet wird. S_{max} bildet den Messwert M_{max} ab.

Um aus den physikalischen Werten die Signalwerte zu bestimmen, wird für die in Abbildung 6a gezeigte Funktion die Geradengleichung aufgestellt.

$$S = \frac{S_{max} - S_{min}}{M_{max} - M_{min}} \cdot (M - M_{min}) + S_{min} \quad (4)$$

Umgekehrt berechnen sich die physikalischen Werte aus den Signalwerten nach der Geradengleichung für Abbildung 6b.

$$M = \frac{M_{max} - M_{min}}{S_{max} - S_{min}} \cdot (S - S_{min}) + M_{min} \quad (5)$$

Die in beiden Gleichungen auftretende Verschiebung auf der Abszisse bewirkt die Verschiebung der Gerade in den Koordinatenursprung.

2.2.3 Industrielle Kommunikation über Profinet

Die Kommunikation zwischen den Geräten des industriellen Netzwerkes geschieht u.a. über Profinet IO. Profinet ist eine Industrial Ethernet-Lösung die Echtzeitkommunikation unterstützt. Es basiert auf der TCP/IP-Protokollgruppe und ermöglicht die Integration von Feldbus-Systemen. Dadurch wird die Anbindung dezentraler Baugruppen bewerkstelligt.

Im Profinet-Netzwerk wird jedem Gerät eine eindeutige IP-Adresse zugeordnet, die als Identifizierung des Geräts dient. Zusätzlich erweitert die IP-Adresse den Funktionsumfang der Geräte um u.a. Konfigurations-, Diagnose oder auch Parametrieranwendungen.

Unterschieden wird nach IO-Controller, IO-Device und IO-Supervisor. Unter IO-Controller werden Geräte verstanden, die verantwortlich die Automatisierung steuern. Sie haben eine Master-Funktion inne. IO-Devices sind Feldgeräte, die einem IO-Controller untergeordnet sind und eine Slave-Funktion erfüllen. Sie bilden die Schnittstellen zum Prozess über analoge oder digitale Eingangs- und Ausgangssignale. IO-Supervisor sind im Bezug auf Siemens-Steuerungen das TIA-Portal oder der SIMATIC Manager, die als Engineering-Tool die IO-Devices parametrieren und überwachen. [Wik22]

Zusätzlich zur IP-Adresse müssen den Geräten Profinet-Namen zugewiesen werden. Sie dienen zur Identifizierung der Geräte im Profinet-Netzwerk. Sie ist vergleichbar mit der MAC-Adresse, die aus der Netzwerktechnik bekannt ist. Die Namen müssen nach Profinet-Namenskonvention eingerichtet werden.[Sie21b]

Neben dem eingeschränkten Umfang zugelassener Zeichen, ist bei den Namen darauf zu achten, dass diese sich klar von der IPv4-Schreibweise abgrenzen und nicht mit der Zeichenfolge *port-xyz* ($x,y,z = 0 \dots 9$) beginnen. Zugewiesen werden die Namen bei der Inbetriebnahme der Geräte.

Unter Nutzung des Producer-Consumer-Schemas⁴ werden zyklisch Daten zwischen IO-Controller und IO-Device ausgetauscht. Dabei werden die Ausgangssignale an die Peripherie geschickt, sowie die Eingangssignale an den IO-Controller. Für Diagnosemeldungen, Parametrier- oder Konfigurationsdateien wird ein azyklischer Datenaustausch zwischen IO-Controllern und IO-Devices aufgebaut. Ebenso werden azyklische Alarmer vom Peripheriegerät an den Controller geschickt. Für diesen Dienst wird auf UDP/IP zurückgegriffen, da dieses eine schnellere Übertragung ermöglicht. Nachteil von UDP/IP ist, dass keine Acknowledge-Dateien gesendet werden. Dadurch ist eine fehlerfreie Übertragung nicht gewährleistet. Es wird daher für Übertragungen verwendet, bei der ein höheres Augenmerk auf schnelle, statt auf fehlersichere Übertragungen gelegt wird.

2.2.4 Betrieb von Kommunikationsstrecken über DSL und SHDSL

Für die Übertragung auf längeren Strecken von bis zu 20 km wird u.a. auf DSL (Digital Subscriber Line) bzw. SHDSL (Single-Pair Highspeed Digital Subscriber Line) zurückgegriffen. Als Übertragungsmedium werden hier Kupferdoppeladern mit einem Durchmesser von 0,25 bis 0,8 mm verwendet. Beide Übertragungsstandards sind insofern mit Profinet vereinbar, dass sie beliebige Übertragungsprotokolle der höheren OSI⁵-Schichten unterstützen. Entsprechend sind

⁴Producer und Consumer sind Rollen in der Prozesssynchronisation. Während der Producer Datensätze erzeugt, verarbeitet der Consumer diese Daten.

⁵Das ISO/OSI-Referenzmodell (OSI: Open System Interconnection) beschreibt die Kommunikation von Geräten über Protokollstrukturen und Schichten, bei denen die einzelnen Schichten klar abgegrenzte Aufgaben erfüllen. Jede Kommunikation durchläuft die einzelnen Schichten zur Übertragung von Daten.

sowohl TCP/IP-, als auch UDP/IP-Verbindungen möglich.

Über Ethernet-Extender kann das Ethernet-Netzwerk über die 100 m-Beschränkung der IEEE 802.3 hinaus genutzt werden. Weiterhin steigern sie die Übertragungsrate über SHDSL auf bis zu 15,32 MBit/s, abhängig von der Übertragungsstrecke und der Qualität der Verbindungsleitungen. Mittels der Ethernet-Extender können sowohl Linien- als auch einfache Ringstrukturen aufgebaut werden. Tiefer gehende Ausführungen sollen nicht Thema dieser Arbeit sein.

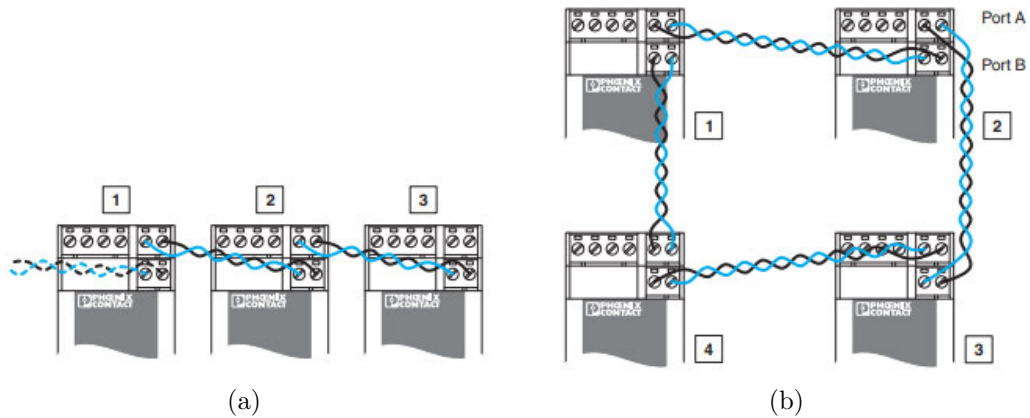


Abbildung 7: Linien- und Ringstruktur mit Ethernet-Extendern [PHO21]

2.2.5 Frequenzumrichter

Die Aufgabe von Frequenzumrichtern (kurz: FU) besteht darin, den angeschlossenen elektrischen Maschinen eine veränderliche Frequenz von Spannung und Strom zur Verfügung zu stellen. Dafür wird die Netzfrequenz umgerichtet. Elektrisch unterteilt wird der FU in den Eingangs-, Zwischen- und Ausgangskreis.

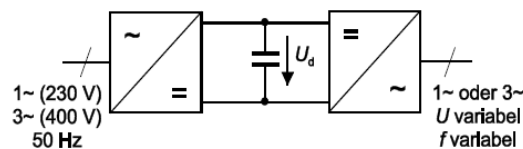


Abbildung 8: Schematische Darstellung eines Frequenzumrichters als Blockschaltbild [Pro20, S. 273]

Der Ausgangskreis ist als Drei-Phasen-Wechselrichters (kurz: 3-Ph-WR) ausgelegt (siehe Abbildung 9).

Die Frequenzumrichtung erfolgt im Ausgangskreis durch Grundfrequenztaktung. Dafür werden die einzelnen Schalter der Schalterpaare des 3-Ph-WR in bestimmten Abständen geschaltet. Bei symmetrischen, im Stern angeschlossenen Lasten folgen dadurch die in Abbildung 10 gezeigten Spannungsverläufe.

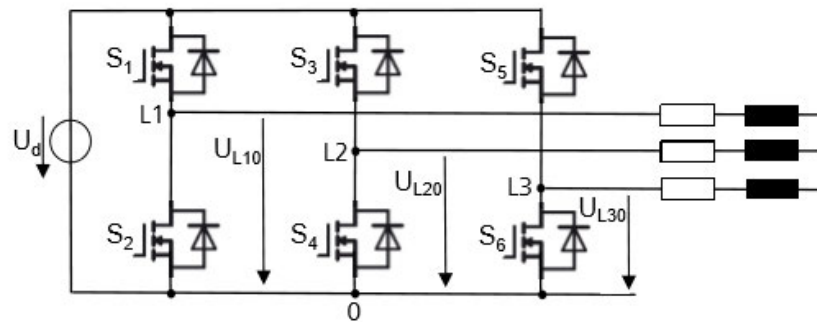


Abbildung 9: Vereinfachter Aufbau eines dreiphasigen Wechselrichters

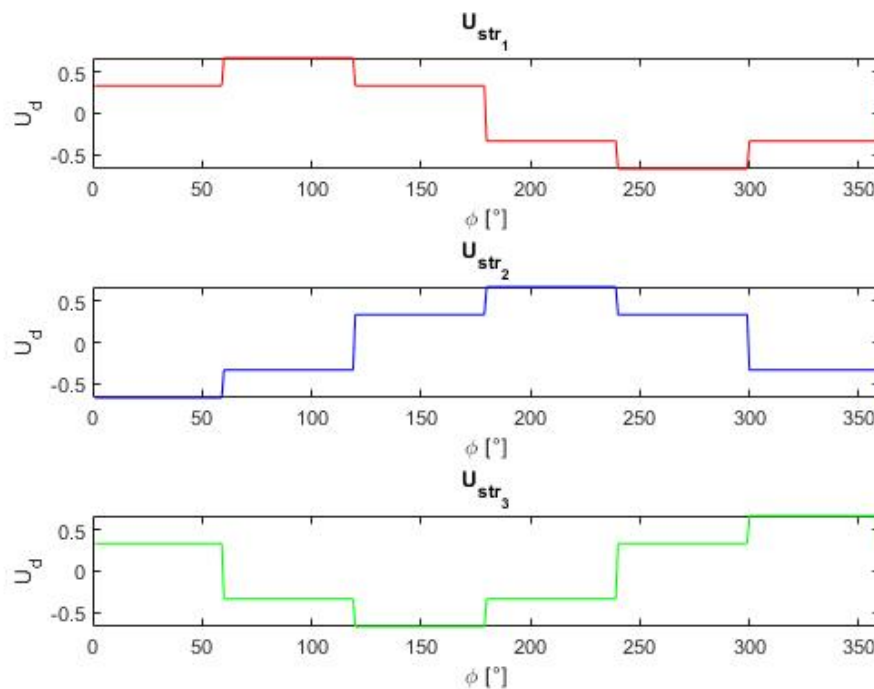


Abbildung 10: Strangspannungen einer symmetrischen Last bei Grundfrequenztaktung

Durch Vernachlässigung der ohmschen Lastanteile folgen sinusähnliche Laststromverläufe. Mit Hilfe der Grundfrequenztaktung besteht also durch schnelleres oder langsames Schalten die Option, die Frequenz der Spannungen und der Ströme zu verändern. Nicht beeinflussbar durch dieses Verfahren ist allerdings die Amplitude der Spannungen. Diese kann durch verschiedene andere leistungselektronische Verfahren, wie z.B. dem Unterschwingungsverfahren angepasst werden. Auf die leistungselektronische Betrachtung der Funktionsweise von Frequenzumrichtern wird in dieser Arbeit nicht weiter eingegangen.

Die typischen Schnittstellen von FUs können am Klemmenplan aus Abbildung 11 erläutert werden.

Der FU bildet ein einstellbares Frequenzband auf 0 bis 100% ab. Der Sollwert, den der FU einstellen soll, wird als Analogsignal an den FU gegeben. Im beispielhaften Klemmenplan (siehe Abbildung 11) ist dafür die Klemme 2 vorgesehen. Das Analogeingangssignal kann über einen Wechselschalter als Spannungs- oder Stromsignal festgelegt werden. An Klemme 3 wird die Mas-

se des Analogeingangskreises angelegt.

Neben zahlreichen programmierbaren Digitaleingängen (Klemmen 11 bis 16) und anderen Kommunikationsschnittstellen, die nicht einheitlich für die Gesamtheit der Frequenzumrichter gilt, kann über die Klemme 7 ein Analogausgang ausgelesen werden. Dieser gibt den Ausgangsfrequenzwert an, der derzeit am angeschlossenen Motor eingestellt ist. Der Motor ist an den Leistungsklemmen U2, V2 und W2 mit dem FU verbunden. Die umzurichtende Netzspannung wird an U1, V1 und W1 angelegt.

Weiterhin sind oftmals Relais- oder auch Digitalausgänge auf Klemmen gelegt, die Sammelstörungen anzeigen sollen. Hinter einer Sammelstörung können sich u.a. Störabschaltungen, oder auch das Auftreten von Überstrom oder Unterspannung verbergen.

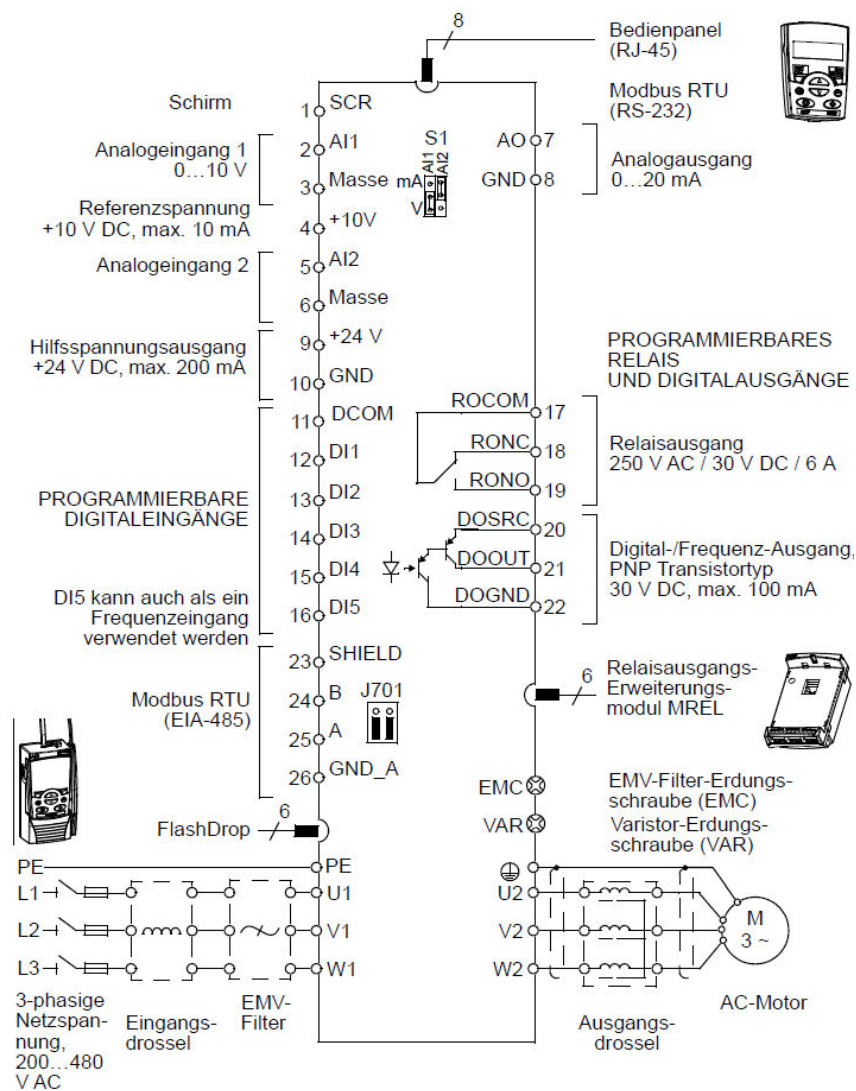


Abbildung 11: Typische Anschlussklemmen von Frequenzumrichtern [ABB11, S. 57]

Die o.g. Digitaleingänge werden zumeist für folgende Befehle genutzt:

- Start-Stop-Befehle,
- Drehrichtung (vorwärts bzw. rückwärts),
- Auswahl von Fest- bzw. Konstantdrehzahl und
- Beschleunigungs- und Verzögerungsauswahl unter Verwendung interner Rampenzeiten.

Die für die Digitaleingänge benötigte Hilfsspannung kann im gezeigten Klemmenplan an Klemme 9 angelegt werden. Es ist aber auch möglich, eine externe Spannungsquelle zu nutzen. Zusätzlich ist es bei einigen FUs möglich, diese über Feldbussignale anzusteuern. Dabei wird die Kommunikation von dem FU-internen Programm und der verbauten SPS übernommen.

2.3 Messtechnik

Die Messtechnik umfasst die Erfassung physikalischer Größen mittels geeigneter Verfahren und Geräte. Dabei soll in diesem Abschnitt besonders auf die Erfassung von Durchfluss, Druck und Leitwert eingegangen werden. Die Auswahl der in den folgenden Unterkapiteln vorgestellten Messprinzipien orientiert sich an der Bestandsanlage.

2.3.1 Messwerterfassung Durchfluss

Für die Durchflussmessung wird ein, auf dem Faraday'schen Induktionsgesetz beruhendes Messverfahren angewandt. Voraussetzung für die fehlerfreie Durchflussmessung von Flüssigkeiten, mittels des magnetisch-induktiven Messverfahrens, ist ein Leitwert $\geq 5 \mu S/cm$.

An der Messstelle wird ein Strom angelegt, durch welche sich ein Magnetfeld ausbildet. Im fließenden Messstoff wird eine Spannung U_e induziert. Diese ist proportional zur Fließgeschwindigkeit und wird über Messelektroden erfasst. Unter Zuhilfenahme des Rohrquerschnitts kann der Volumenstrom ermittelt werden. [End17]

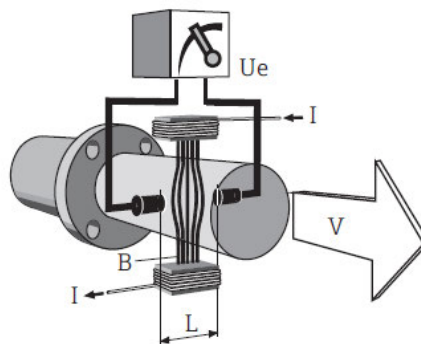


Abbildung 12: Prinzip der magnetisch-induktiven Durchflussmessung [End17]

Über folgende Gleichungen kann der Durchfluss Q berechnet werden.

$$U_e = B \cdot L \cdot v \quad (6)$$

$$Q = A \cdot v \quad (7)$$

Durch Einsetzen der Gleichung 6 in Gleichung 7 folgt:

$$Q = A \cdot \frac{U_e}{B \cdot L} \quad (8)$$

Dabei bezeichnen die Formelzeichen folgende physikalische Größen:

- B : Magnetische Induktion (Magnetfeld)
- L : Elektrodenabstand
- v : Durchflussgeschwindigkeit
- Q : Volumenfluss
- A : Rohrleitungsquerschnitt
- I : Stromstärke

Es gibt weitere Messverfahren für die Messung von Volumenströmen von Flüssigkeiten, die u.a. auf der Verwendung von Ultraschall, Differenzdruck oder Wärmeleistungsmessung beruhen. Diese werden hier jedoch nicht weiter behandelt.

2.3.2 Messwerterfassung Druck

Der Prozessdruck wird mit Hilfe einer metallischen Prozessmembrane erfasst. Die durch den Druck ausgelenkte Membrane überträgt ihre Elongation über eine Füllflüssigkeit auf eine Wheatstonesche Messbrücke. Die Ausgangsspannung der Brücke ist proportional zum anliegenden Prozessdruck. In hochwertigen Drucksensoren wird zusätzlich die Temperatur gemessen, um die Umgebungseinflüsse zu berücksichtigen. [End21]

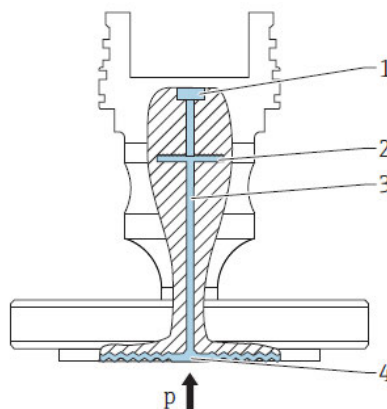


Abbildung 13: Aufbau Drucksensor mit metallischer Membran [End21]

Die Sensorelemente aus Abbildung 13 sind folgend bezeichnet:

- 1: Messelement
- 2: Innenliegende Prozessmembrane
- 3: Kanal mit Füllflüssigkeit
- 4: Metallische Prozessmembrane
- p : Druck

2.3.3 Messwerterfassung Leitwert

Die Leitwertmessung erfolgt, ähnlich wie die Durchflussmessung, auf Grundlage einer induktiven Messung. Im Sensor wird durch einen Oszillator ein wechselndes Magnetfeld in der Primärspule erzeugt. Der im Messstoff induzierte Strom erzeugt ein Magnetfeld in der Sekundärspule. Der Induktionsstrom ist abhängig vom Leitwert des Mediums.

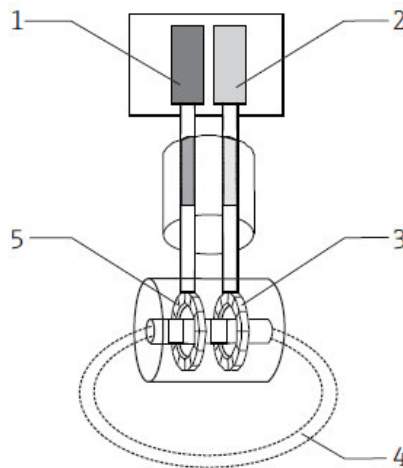


Abbildung 14: Aufbau Leitwertsensor für die induktive Leitwertmessung [End18]

Dargestellt sind in Abbildung 14:

- 1: Oszillator
- 2: Empfänger
- 3: Sekundärspule
- 4: Stromfluss durch das Medium
- 5: Primärspule

Im Gegensatz zu anderen Verfahren, werden hier keine Elektroden verwendet und dadurch Polarisierungseffekte vermieden. Weiterhin ist die Messung vollständig galvanisch vom Medium getrennt. [End18]

2.3.4 Messwertübertragung

Für die drahtgebundene Messwertübertragung können als Stromgeber 2-Draht- oder 4-Draht-Messumformer verwendet werden. Beide müssen für die Messwernerfassung mit Spannung versorgt werden. Dem 2-Draht-Messumformer wird über das Analogeingabemodul die Spannung zugeführt. Daher auch die Bezeichnung des „passiven“ Messumformers. Die einfache Verdrahtung nach Abbildung 16 bringt eine erhöhte Störanfälligkeit durch den Einfluss elektro-magnetischer Felder, leitungsbedingter parasitärer Kapazitäten und Induktivitäten, sowie durch den ohmschen Leitungswiderstand mit sich. Durch Schirmen, Erden und Verdrillen der 2-Draht-Leitung können Störeinflüsse minimiert werden. Weiterhin ist für den Betrieb passiver Messumformer die Bürde eine wichtige Größe. Sie gibt den maximalen Widerstand der Stromschleife an. Sie setzt sich aus dem Widerstand des Messumformers und allen Widerständen der Stromschleife zusammen. Sollte der Widerstand größer als die Bürde sein, würde der 2-Draht-Messumformer mit einer zu niedrigen Spannung versorgt werden, wodurch die Qualität der Messdaten stark negativ beeinflusst würde.

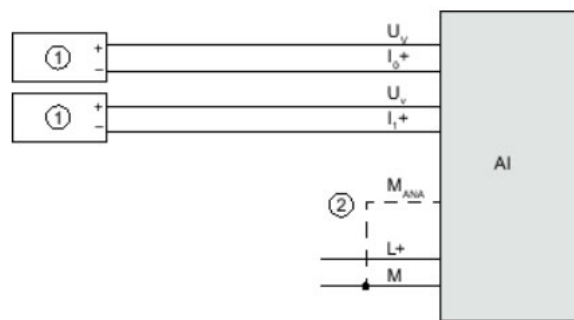


Abbildung 15: Anschluss von 2-Draht-Messumformern an Analogeingangsgruppen [Sie14, S. 70]

4-Draht-Messumformer hingegen besitzen eine externe Spannungsversorgung, weshalb sie auch als „aktive“ Messumformer bezeichnet werden.

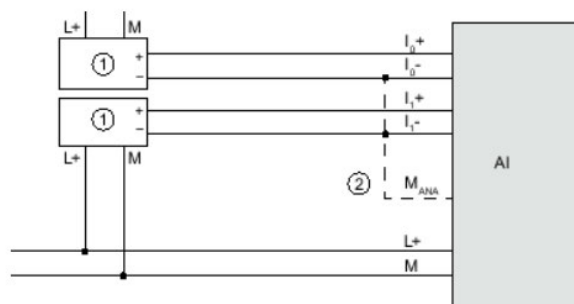


Abbildung 16: Anschluss von 4-Draht-Messumformern an Analogeingangsgruppen [Sie14, S. 71]

Durch die Trennung von Last- und Versorgungskreis ist das Messsignal von 4-Draht-Messumformern weniger störungsbelastet, als die von 2-Draht-Messumformern.

2.4 Regelungstechnik

Unter Regelungstechnik wird die gezielte Beeinflussung von zeitlich veränderlichen Prozessen verstanden, sodass sie in einer vorgegebenen Weise ablaufen. [Lun13, S. 1] Dabei wird zwischen kontinuierlichen und diskreten Regelungen unterschieden. Auf die zuletzt genannten wird in dieser Arbeit nicht eingegangen.

Um die Prozesse wie gewünscht zu beeinflussen, werden Regelkreise aufgestellt (siehe Abbildung 17). Der charakteristische Aufbau dieser mathematischen Modelle besteht aus einer Regelstrecke, die das Verhalten des Prozesses abbildet, einem Regler, der Einfluss auf die Strecke übt, und einer Rückführung des Reglerausgangs, die einem fortlaufenden Soll-Istwert-Vergleich, in Form einer Regelabweichung $e(t)$, dient. Dargestellt werden die Modelle als Blockschaltbilder, wie in Abbildung 17, oder als Signalflussgraphen, die jedoch für einfache Regelungen eher ungeeignet sind.

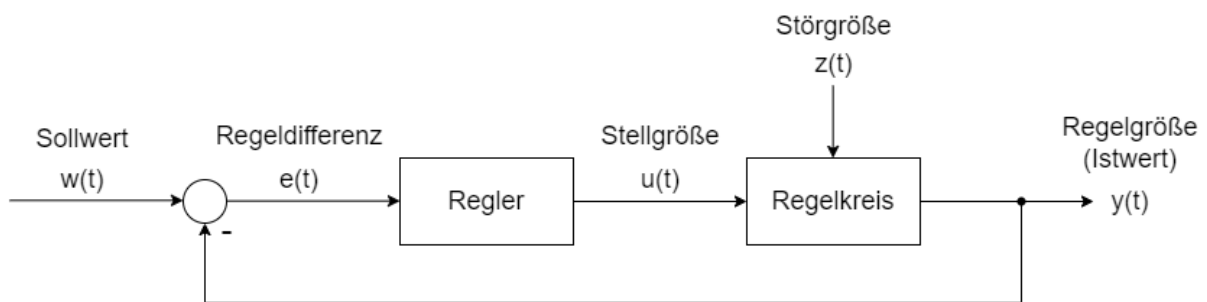


Abbildung 17: Standardregelkreis mit Rückführung der Regelgröße

Ziel einer idealen Regelung ist es, die Regelgröße $y(t)$ auf den Sollwert $w(t)$ anzupassen. In vielen Bereichen werden Eingrößenregelungen eingesetzt, die auf die Anpassung einer einzigen Prozessgröße ausgelegt sind.

Im Vergleich zu Steuerungen können Regelungen, bei falsch ausgelegten Reglern, instabil werden. Jedoch können diese unter bestimmten Umständen stabilisiert werden. Sollte das Modell einer Steuerung eine Instabilität erreichen, kann dieses nicht mehr gesteuert werden. Weiterhin wirkt eine gut ausgelegte Regelung unbekanntem Störgrößen $z(t)$ entgegen.

Die zu regelnden Prozesse müssen bestimmte Anforderungen erfüllen. Bezogen auf die zeitliche Betrachtung des Prozesses, muss dieser sowohl zeitkontinuierlich, als auch zeitinvariant sein. Als zeitkontinuierlich wird ein System verstanden, dessen Definitionsbereich der auftretenden Zeitfunktion entweder die gesamte Zeitachse, oder ein Intervall der Zeitachse umfasst.

Zeitinvariante Systeme zeichnen sich dadurch aus, dass sie auf zeitlich verschobene Eingangssignale immer mit, um die selbe Zeit T_t verschobene Ausgangssignale antworten. Mit $T_t > 0$ folgt:

$$u_1(t) = u_2(t - T_t) \mapsto y_1(t) = y_2(t - T_t) \quad (9)$$

Eine weitere wichtige Eigenschaft dynamischer Systeme ist die Kausalität. Diese besagt, dass die Eingangsgröße zur Zeit \bar{t} nur für zukünftige Zeitpunkte $t \geq \bar{t}$ das Verhalten des Systems beeinflussen kann. Grundsätzlich ist die Regelung von allen bekannten physikalischen Prozessen

immer kausal, da der Einfluss eines gegenwärtigen Zustands keine Rückwirkung auf die vergangenen Zustände bewirken kann. Der Begründung dafür ist aus dem zweiten Hauptsatz der Thermodynamik⁶ abzuleiten.

Um das Verhalten von Reglern beschreiben zu können werden Buchstabencodes eingeführt.

- P: Proportionalverhalten
- I: Integratives Verhalten
- D: Differenzierendes Verhalten

P-Regler wirken als Multiplikator auf das anstehende Signal. Ihre Übertragungsfunktion $G_R(s)$, die das Verhalten von Ausgang zu Eingang des Reglers beschreibt, leitet sich aus dem Blockschaltbild des Regelkreises in Abbildung 17 ab.

$$u(t) = K_P \cdot e(t) \quad (10)$$

Durch Transformation in den Bildbereich lässt sich die folgende Übertragungsfunktion bestimmen:

$$G_R(s) = \frac{U(s)}{E(s)} = K_P \quad (11)$$

Der Faktor K_P wird Proportionalitätsfaktor genannt, da die Regler-Ausgangsgröße $U(s)$ proportional zu Regler-Eingangsgröße $E(s)$ ist.

Der Einfluss von I-Reglern kann folgendermaßen beschrieben werden:

$$u(t) = K_I \int e(t) dt \quad (12)$$

Für den Bildbereich folgt damit die Übertragungsfunktion:

$$G_R(s) = \frac{K_I}{s} = \frac{1}{T_I \cdot s} \quad (13)$$

K_I ist der Kehrwert der Integrationszeit T_I .

Reine D-Regler werden nicht verwendet, da sie nicht auf konstante Regelabweichungen reagieren. Durch verschiedenartige Verschaltungen der Grundregelungen kann auf viele Regelstrecken Einfluss genommen werden. Die meist verwendeten Regler werden als PID-Regler bezeichnet. Typische Regelstrecken sind PT_n-Strecken. Das n steht für die Anzahl der Polstellen der Regelstrecke im Bildbereich. Sie bestimmen die Systemdynamik und sind daher essenziell wichtig in der Betrachtung und der Auslegung der Regelung. Dabei kann die Ordnung des Systems ebenfalls aus diesem Wert abgeleitet werden. Beispielsweise ist eine PT₂-Strecke ein System zweiter

⁶Kurzgefasst sagt der zweite Hauptsatz der Thermodynamik im Bezug auf Zeit aus, dass sich alle Systeme zeitlich vorwärts bewegen, solange sie zu einem Maß der energetischen Unordnung streben.

Ordnung, das aus einer Differenzialgleichung gebildet wird, deren höchste Ableitung die zweite ist.

2.5 SPS-Programmierung

Für die Erstellung der Anwenderprogramme auf Siemens-Steuerungen wird das *Totally Integrated Automation Portal* (kurz: TIA-Portal) verwendet. Es ist ein Engineering-Tool, das die Verwaltung und Programmierung der Projektdateien ermöglicht. Weiterhin werden mit diesem Tool die zu projektierenden Geräte konfiguriert und auch Kommunikationsnetzwerke werden hier erstellt.

Zur Programmierung stehen verschiedene Sprachen zur Verfügung. Neben dem grafischen *Funktionsplan* (kurz: FUP) und *Kontaktplan* (kurz: KOP), sind die meist verwendeten Sprachen die Assembler-ähnliche *Anweisungsliste* (kurz: AWL) und die Hochsprache *Strukturierter Text* (kurz: ST), welche auch als *Structured control Language* (kurz: SCL) bekannt ist.

2.5.1 Programmbausteine

Die Programmierung von SPSen basiert auf der Verwendung unterschiedlicher Programmbausteine. Diese unterteilen sich in programmierbare Bausteine, die Funktionen ausführen und speichernde Programmbausteine, die sog. Datenbausteine (kurz:DB).

Bei den programmierbaren Bausteinen stehen Funktionsblöcke (kurz: FB) und Funktionen (kurz: FC) zur Verfügung. Der Unterschied beider liegt in der Verwaltung und Nutzung der Speicherstrukturen. Mit jedem Aufruf eines FBs wird ein Instanzdatenbaustein (kurz: IDB) erzeugt, in dem die verwendeten Datenpunkte abgelegt werden. Dadurch wird eine funktionsinterne Datenverwaltung ermöglicht. FCs haben temporären Speicher, der während des Aufrufs zur Verfügung steht. Darin werden die temporären Variablen des aufgerufenen FCs abgelegt. Nach dessen Abarbeitung wird der Speicherplatz neu vergeben.

Für FCs und FBs können Schnittstellen nach außen geschaffen werden, die Dateneingänge (Inputs) und Datenausgänge (Outputs) zur Verfügung stellen. Mit diesen Schnittstellen kann in dem FC gearbeitet werden.

Um die programmierten Bausteine nutzen zu können, müssen sie in Organisationsbausteinen (kurz: OB) aufgerufen werden. Sie sind die Schnittstellen des Anwenderprogramms zum Betriebssystem der SPS. Gestartet werden diese nur bei bestimmten Ereignissen durch das Betriebssystem. Wichtige OBs sind u.a der OB100 und der OB1. Der OB100 ist nach Neustart der CPU für den Anlauf des Betriebssystems zuständig. Im OB1 wird das Anwenderprogramm aufgerufen und zyklisch abgearbeitet. Weitere wichtige Organisationsbausteine sind die Cyclic Interrupt OBs, da diese in einstellbaren Zeitabständen ausgeführt werden. Sollte nach Ablauf der eingestellten Zeit ein anderer OB bearbeitet werden, wird dieser für den Cyclic Interrupt OB ausgesetzt. Genutzt werden diese OBs für alle Funktionen, die eine feste Zykluszeit erfordern.

Für den Einsatz der Cyclic Interrupt OBs und um bei Auftreten einer Systemstörung das Anwenderprogramm unterbrechen zu können, wird das sog. präemptive Multitasking genutzt. Dabei werden die OBs mit einer Priorität versehen. Bei Auftreten einer Störung wird der niedriger priorisierte OB ausgesetzt und der höher priorisierten OB von der CPU ausgeführt. Sobald das Programm des höher priorisierten OBs abgearbeitet ist, wird das des niedriger priorisierten

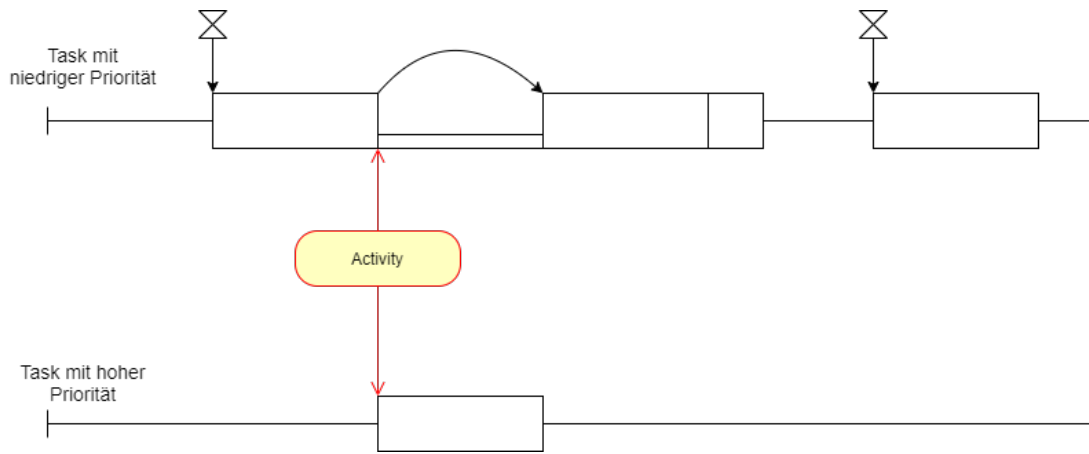


Abbildung 18: Vereinfachte Veranschaulichung des präemptiven Multitaskings

OBs fortgesetzt. Der OB 121 und OB 122 haben mit Prioritätslevel 29 die höchste Priorität. Die niedrigste Priorität mit Level 1 hat der OB 1 und somit das Anwenderprogramm. Veranschaulicht wird das Prinzip des präemptiven Multitaskings in Abbildung 18.

2.5.2 Regelung kontinuierlicher Prozesse

Die Regelung kontinuierlicher Prozesse geschieht in der Programmierung über von Siemens vorgefertigte Technologiebausteine. Der Baustein *CONT_C* ist speziell für diesen Anwendungsfall konzipiert. Es handelt sich um einen konfigurierbaren PID-Regler, über den Einfluss auf den Prozess genommen werden kann. Die einzelnen Regleranteile sind an- und abwählbar und konfigurierbar. Allerdings wird die Verwendung als reiner D-Regler ausgeschlossen. Das Hinterlegen eines mathematischen Streckenmodells ist nicht notwendig. Es handelt sich um einen Eingrößenregler, dessen Regelgröße auch als Peripherievariable angelegt werden kann. Diese wird vom Baustein auf 0 bis 100 % skaliert. Aus dem Soll- und dem Istwert kann somit die Regeldifferenz berechnet werden, die auch am Bausteinausgang abgefragt werden kann. Über ein Totband kann die Ausregelung der Regeldifferenz in einem bestimmten Bereich um den Sollwert herum verhindert werden. Sinnvoll ist dies für um den Sollwert schwingende Systeme, um ein Überansprechen des Reglers zu vermeiden.

Die Stellgröße kann mittels einstellbarer Stellwertgrenzen eingeschränkt werden. Dadurch ist es notwendig, speziell für Regler mit I-Anteil, Anti-Windup⁷-Maßnahmen ergreifen zu können. Hier wird das sog. Clamping zur Verfügung gestellt, wodurch der I-Anteil bei Erreichen der Stellwertbegrenzung und damit der Sättigung des Reglers, „eingefroren“ wird.

Weiterhin besteht die Option einen Handbetrieb einzustellen, bei der ein Festwert als Stellgröße vorgegeben wird. Durch internes Anpassen der Regleranteile wird eine stoßfreie Umschaltung zum Automatikbetrieb gewährleistet.[Sie19b]

Ebenso möglich ist es, eine Störgrößenaufschaltung vorzunehmen.

Das Blockschaltbild des Technologiebausteins ist dem Anhang zu entnehmen. Der Anhang zur

⁷Windups treten bei Reglern mit Stellgrößenbeschränkung und I-Anteil auf. Geht die Regelung in Sättigung kann die Strecke für einen gewissen Zeitraum nicht ausgeregelt werden. Dabei steigt der I-Anteil an. Sollte zu einem späteren Zeitpunkt ein Sollwert angenommen werden, der nicht zur Sättigung des Reglers führt, kann der übergroße I-Anteil ein starkes Schwingen oder die Instabilität des Reglers bewirken. Um dies zu umgehen werden Anti-Windup-Maßnahmen erhoben.

Arbeit befindet sich auf CD und kann beim Erstgutachter eingesehen werden.

2.6 Human Machine Interface

Die Hauptaufgabe des Human Machine Interface (kurz: HMI) ist das Bedienen und Beobachten von Prozessen. Das HMI gibt dem Bediener die Möglichkeit, bei Bedarf in den Prozess einzugreifen, aber auch, um z.B. Einstellungen vorzunehmen.

Der Nutzereingriff erfolgt auf Grundlage von Anlagenübersichten, Buttons und anderen digitalen Bedienelementen, die im HMI projiziert werden. Sie sind mit Datenpunkten der SPS verknüpft, auf die über die Bedienung zugegriffen wird.

2.6.1 Siemens Comfort Panel

Die Comfort Panels entstammen einer Produktreihe der Firma Siemens für Touch- und Key Panels. [Sie21d] Die Panels sind in unterschiedlichen Größen verfügbar. Im Folgenden wird jeweils vom Sieben-Zoll Panel ausgegangen, das in Abbildung 19 zu sehen ist.



Abbildung 19: SIMATIC HMI TP700 Comfort [Aut21]

Über das Panel ist es möglich, aktiv, von außen auf Prozesse einzuwirken. Gleichmaßen dient es der Beobachtung des Prozesses. Ermöglicht wird dies u.a. über Kurven-, Trend- oder auch Füllstandanzeigen und anderweitige Beobachtungselemente. Einfache Möglichkeiten Prozessdaten darzustellen sind Ein- und Ausgabe-Felder (kurz: E/A-Felder). Die Panels bieten die Möglichkeit Daten zu archivieren und damit über einen längeren Zeitraum zugänglich zu machen.

2.6.2 WinCC Comfort

WinCC Comfort ist ein im TIA-Portal integriertes Tool zur Erstellung von Prozessvisualisierungen. Mit der Ausführung *Comfort* können für Siemens Panels und HMIs der Produktstufe *Basic*, *Comfort* und *Mobile* Visualisierungen projiziert werden. Über verschiedene vorgefertigte Bedien- und Beobachtungselemente können umfassende HMIs erstellt werden. Hier findet auch die Verknüpfung der Prozessvariablen der SPS mit der Visualisierung statt.

Weiterhin können hier Alarm- und Warnmeldungen erzeugt werden, die über das HMI quittierbar sind. Unterschieden wird zwischen Analog- und Bitmeldungen, die in Meldeklassen und -gruppen organisiert werden.

Externe Erweiterungen sind mit WinCC möglich. Diese reichen von der Archivierung der Prozessdaten in SQL⁸-Datenbanken bis hin zur Kommunikation über OPC UA⁹-Server.

2.6.3 Datenaustausch mit SPS

Die Verbindung zwischen HMI und SPS erfolgt allgemein über Ethernet. Genutzt wird hierfür oftmals eine Profinet-Anbindung. Die Kommunikation selbst erfolgt über das Server-Client-Prinzip. Als Server wird ein Programm bezeichnet, das ausgewählte Daten der SPS, sog. Items, in zyklischen Abständen anderen Programmen zu Verfügung stellt, ohne genau zu wissen, wer diese nutzt. [Sei21, S. 45] Dieses Programm ist Teil der SPS.

Der Client ist das Gegenstück zum Server. Er kann auf die Items zugreifen, die der Server zur Verfügung stellt. Dabei kann der Client in Form der HMI lesend und schreibend auf die Items zugreifen. Die beschriebenen Items werden durch den Server zurück an die SPS gegeben, die diese dann übernimmt. Wichtig ist hierbei, dass es sich bei Server und Client nicht um Geräte, sondern um Rollen handelt. Dies bewirkt, dass beispielsweise eine CPU gleichzeitig Server und Client sein kann.

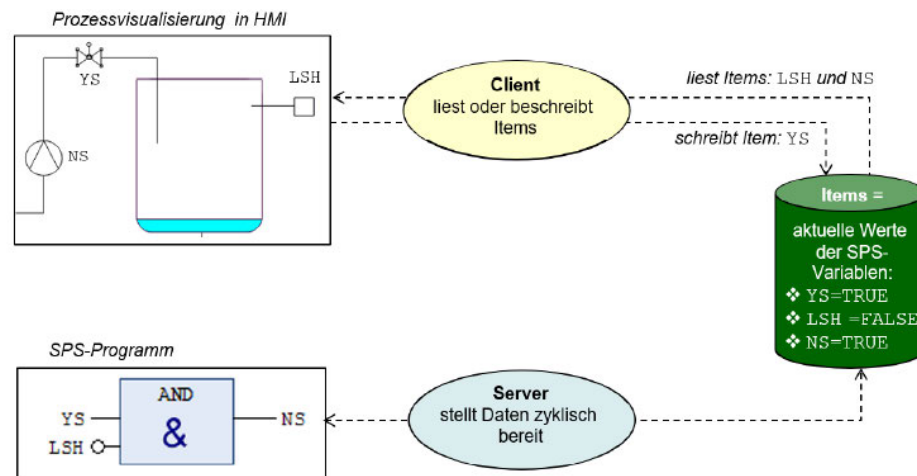


Abbildung 20: Server-Client-Prinzip für SPS und HMI [Sei21, S. 45]

Die zyklische Kommunikation wird azyklisch angestoßen. Dies geschieht bei Aufrufen von Bildern und anderen HMI-Elementen, die spezielle Daten erfordern. Diese Daten werden dann vom Server zyklisch zur Verfügung gestellt.

⁸Structured Query Language: Programmiersprache zur Erstellung und Verknüpfung von Datenbanken

⁹Open Platform Communications Unified Architecture: Standard-Kommunikationsprinzip für den Datenaustausch zwischen Systemen unterschiedlicher Hersteller in der Automatisierungstechnik [Sei21, S. 267]

3 Projektplanung

Nach Abgrenzung des Projektumfangs und dem Erhalt eines Lastenhefts wurde eine Projektplanung erarbeitet. Dafür wurde das Projekt in einzelne Abschnitte unterteilt. Diese umfassen die Vorplanung, den Softwareentwurf, die eigentliche Programmierung, das HMI-Bedienkonzept und die Inbetriebnahme. Für diese übergeordneten Abschnitte wurden Unterabschnitte und Aufgabenpakete definiert. Diese wurden unter Zuhilfenahme eines Projektplans zeitlich sortiert. Die Bearbeitungsdauer für die einzelnen Aufgabenpakete wurde grob geschätzt. Um das Projekt bis zum angedachten Zieldatum fertigzustellen, wurden die Bearbeitungszeiten pessimistisch geschätzt. Außerdem sind für das gesamte Projekt ausreichende Zeitreserven geplant. Diese sind genau so groß dimensioniert, wie die eigentlich angedachte, aber geschätzte Bearbeitungsdauer des Projekts. Die geplanten 121 Projektstage setzen sich also zusammen aus 60 Tagen eigentlicher Bearbeitungsdauer und 61 Tagen Zeitreserve.

Die verhältnismäßig große zeitliche Reserve soll zudem das, aus dem Projektmanagement von Softwareprojekten, bekannte 90 %-Phänomen¹⁰ vermeiden, bzw. ausgleichen.

Die Ausführung des Projektes fand nach einer abgewandelten Form des V-Modells statt. In wöchentlichen Abstimmungsgesprächen wurden die Projektfortschritte vorgestellt und diskutiert. Bei Differenzen zwischen Programm und Anforderungen wurden hier zudem Lösungsansätze erarbeitet, die anschließend softwareseitig umgesetzt wurden.

Der Zeitplan ist dem Anhang beigelegt. Der Anhang der Arbeit befindet sich auf CD und kann beim Erstgutachter eingesehen werden.

¹⁰Das 90 %-Phänomen sagt aus, dass bei Abschätzung des Projektfortschritts auf 90 % für die verbleibenden 10 % nochmal die selbe Zeit benötigt wird. Es ist eine Abwandlung des Paretoprinzips. Als Standardmaßnahme dagegen wird die Projektzeit verdoppelt.

4 Bestandsaufnahme

Um einen Überblick über den Brunnen und die dazugehörigen Strukturen zu bekommen, wurde zu Beginn des Projektes eine Bestandsaufnahme durchgeführt. Viele Informationen hat der zugehörige Stromlaufplan vermitteln können. Andere konnten aus der Begehung der Anlage gewonnen werden. Weiterhin bestand die Möglichkeit die Software und Visualisierung des Wasserwerks nachzuvollziehen.

4.1 Rohrleitungs- und Instrumentenfließbild des Brunnen

Das Rohr- und Instrumentenfließbild (kurz: RI-Fließbild) des Brunnen dient der vereinfachten zeichnerischen Darstellungen des Förderprozesses, um die beteiligten Maschinen, Mess- und Regelinstrumente, sowie Leitungen und sonstige Armaturen schematisch darzustellen.

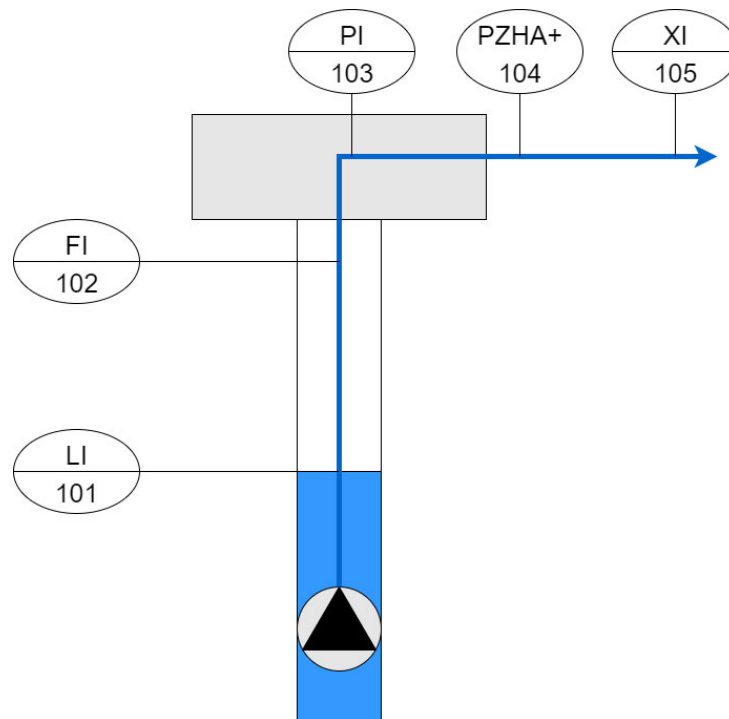


Abbildung 21: RI-Fließbild der Brunnenanlage

Die Messstellen sind durch die Ellipsen gekennzeichnet. Der Kennbuchstabencode macht Angaben über die physikalische Messgröße, der Messwertauswertung und Meldungs- und Alarmverhalten. Die unten stehende Nummer ist eine fortlaufende Messstellenbezeichnung. Die Buchstabenkürzel haben folgende Bedeutungen:

- LI: Niveaumessung mit Anzeige
- FI: Durchflussmessung mit Anzeige
- PI: Druckmessung mit Anzeige
- PZHA+: Druck-Notabschaltung mit Alarmauslösung bei Überschreitung eines oberen Grenzwerts

- XI: Leitwertmessung mit Anzeige

Über den Strich in der Mitte der Ellipsen wird angezeigt, ob das Messorgan vor Ort, oder aus der Ferne ausgewertet werden kann.

Aus dem RI-Schema geht hervor, dass das geförderte Rohwasser über ein handbetriebenes Stellventil auf eine Sammelleitung in Richtung des Wasserwerks geleitet wird.

Weiterhin ist die Unterwasserpumpe des Brunnens hier wiederzufinden.

4.2 Aufnahme der Messtechnik

Aus dem RI-Schema wurde bestimmt, welche verfahrenstechnischen Größen am Brunnen aufgenommen werden. Mit Hilfe des Stromlaufplans der Anlage konnten die, für die Messungen verbauten Geräte ermittelt werden.

Nr.	Bezeichnung	Funktion	Messbereich	Hersteller	Typ
1	Pegelsonde	Niveaumessung	0-20 m	OTT	PLS
2	Durchfluss	Durchflussmessung	0-150 m ³ /h	E+H	Promag W400
3	Wirkdruck	Wirkdruckmessung	0-5 bar	E+H	Cerabar S PMP71
4	Leitwert	Leitwertmessung	0-2000 μ S/cm	E+H	Indumax CLS50D

Tabelle 2: Aufzählung der verbauten Messtechnik

Die Pegelsonde der Firma OTT vom Typ PLS ist zuständig für die Niveaumessung im Brunnen. Aus dem Druck, der auf die Sonde wirkt, folgt die über der Sonde stehende Wassersäule. Die Messungen durch die Geräte der Firma Endress und Hauser (kurz: E+H) geschehen nach den in Abschnitt 2.3 beschriebenen Verfahren.

Die Verbindung mit der SPS erfolgt über Kupferdoppeladern, die an eine AI-Baugruppe angelegt sind.

Bei dem Gerät Promag W400 zur Durchflussmessung besteht die Besonderheit, dass dieses einen zusätzlichen Impulsausgang zur Verfügung stellt. Dieser wird mit einem DI-Eingang verbunden. Die Wertigkeit eines Impuls beträgt beim verbauten Gerät 0,025 m³. Die Impulse werden u.a. zur Mengenerfassung genutzt, indem die Pulse unter Zuhilfenahme ihrer Wertigkeit aufsummiert werden.

4.3 Aufnahme weiterer Signale aus dem Feld

Aus dem Stromlaufplan der Anlage folgten weitere relevante Signale. Die wichtigsten sind in der Tabelle 3 erfasst.

Nr.	Bezeichnung	Funktion	Signalart	Adresse
1	Not-Aus	Notauslösung	DI	E1.5
2	Hochwasser	Wasser im Brunnenschacht	DI	E2.1
3	Tür-/Dachlukenüberwachung	Meldet Station geschlossen	DI	E2.2
4	Druckschalter	Auslösung bei Überdruck	DI	E2.3
5	Wasserwerksmeldung Freigabe	Brunnenfreigabe	DI	E2.7

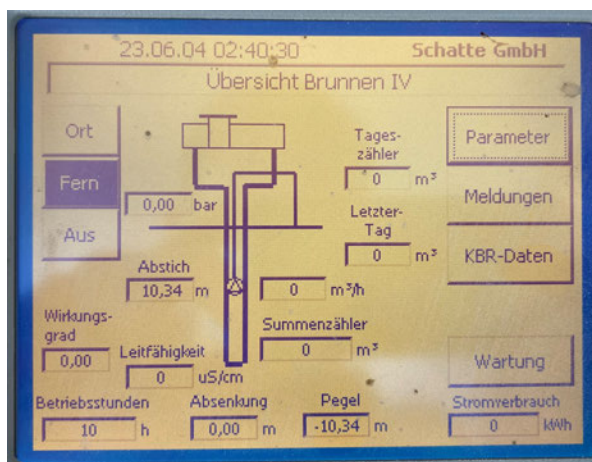
Tabelle 3: Wichtige Signale aus dem Feld

Die aufgeführten Signale nehmen aktiv Einfluss auf den Betrieb der Anlage. Andere vorliegende Signale dienen als Information, auf die nicht automatisiert reagiert wird. Diese melden Sicherungsfälle, machen Angaben zum Betrieb und Zustand der unterbrechungsfreien Spannungsversorgung, oder melden den Ausfall der Kommunikationsstrecke zum Wasserwerk.

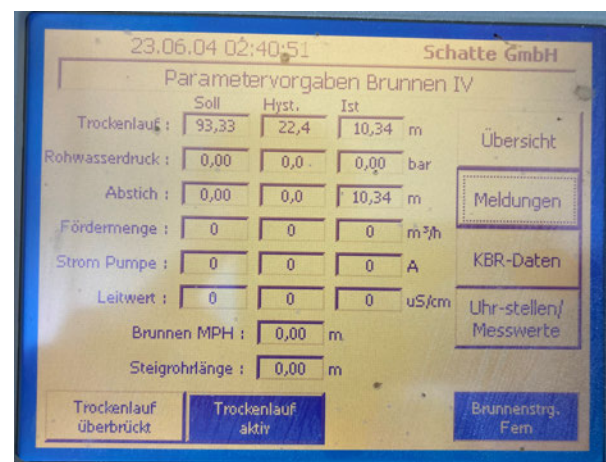
4.4 Aufnahme der Bestandsvisualisierung

Über ein am Brunnen verbautes Panel wird der Prozess visualisiert und ggf. beeinflusst. Die Bestandsvisualisierung umfasst eine Anlagenübersicht mit den wichtigsten Daten. Darunter fallen die Messwerte, aber auch Berechnungen und Zählwerte. Weiterhin sind auf diesem Bild Schaltflächen projektiert, über die verschiedene Zugriffsebenen aktiviert oder deaktiviert werden können. Auf die Zugriffsstruktur wird in Kapitel 4.6 detaillierter eingegangen.

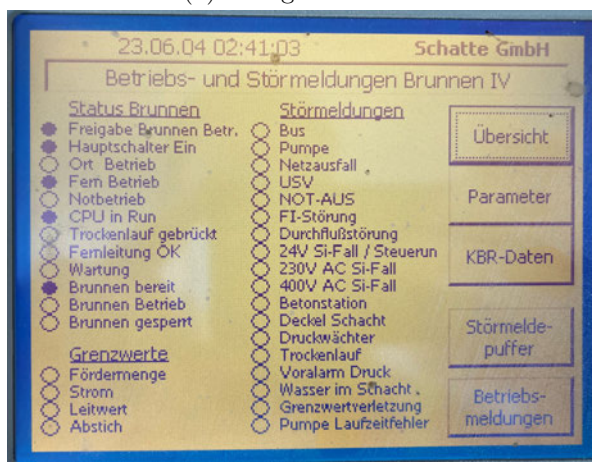
Auf allen Bildern der Visualisierung sind rechts Schaltflächen zu sehen, die der Umschaltung auf andere Bilder dienen. Einige Bilder bieten die Option auf Untermenüs zuzugreifen, wie z.B. auf den Störmeldepuffer.



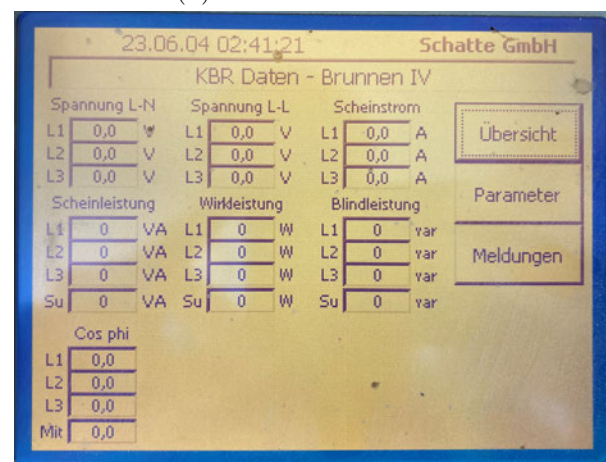
(a) Anlagenübersicht



(b) Parameterübersicht



(c) Meldungsübersicht



(d) Energiedatenübersicht

Abbildung 22: Bilder der Bestandsvisualisierung

Die vier Hauptbilder sind neben der Anlagenübersicht, die Meldungsübersicht, die Parameterübersicht und die Energiedatenübersicht.

Die Visualisierung zeichnet sich weiterhin durch den angebrachten Permanentbereich aus, der Uhrzeit und Datum, sowie den Firmennamen des Unternehmens anzeigt, welches die Visualisierung umgesetzt hat.

4.5 Aufbau des Kommunikationsnetzes

Das bestehende Kommunikationsnetz verbindet den Brunnen, das zugehörige Wasserwerk und die Netzleitstelle (kurz: NLS). Werksintern basiert es auf dem in Kapitel 2.2.3 beschriebenen Profinet. Die Netzleitstelle ist über ein Fernwirkprotokoll auf Basis von TCP/IP angebunden.

Im Wasserwerk sind verschiedene Koppelstellen eingerichtet. Die Brunnenzentrale verwaltet den Datenaustausch zwischen den Brunnen und der Zentral-SPS des Wasserwerks. Außerdem ist sie mit der Fernwirkzentrale verbunden, um die Brunnendaten an die NLS weiterzugeben.

Die Brunnenzentrale und die Fernwirkzentrale sind zwei SPSen, die nur die zugewiesenen Kommunikationsaufgaben umsetzen. Auf Grundlage fest hinterlegter Verbindungen verwalten sie nicht nur den Datenaustausch und die Verbindung der Teilnehmer, sondern stoßen auch die Übertragung aktiv an und managen die Daten, die übertragen werden sollen.

Der Aufbau des Kommunikationsnetzes zwischen den wichtigsten Kommunikationspartnern stellt die Abbildung 23 dar.

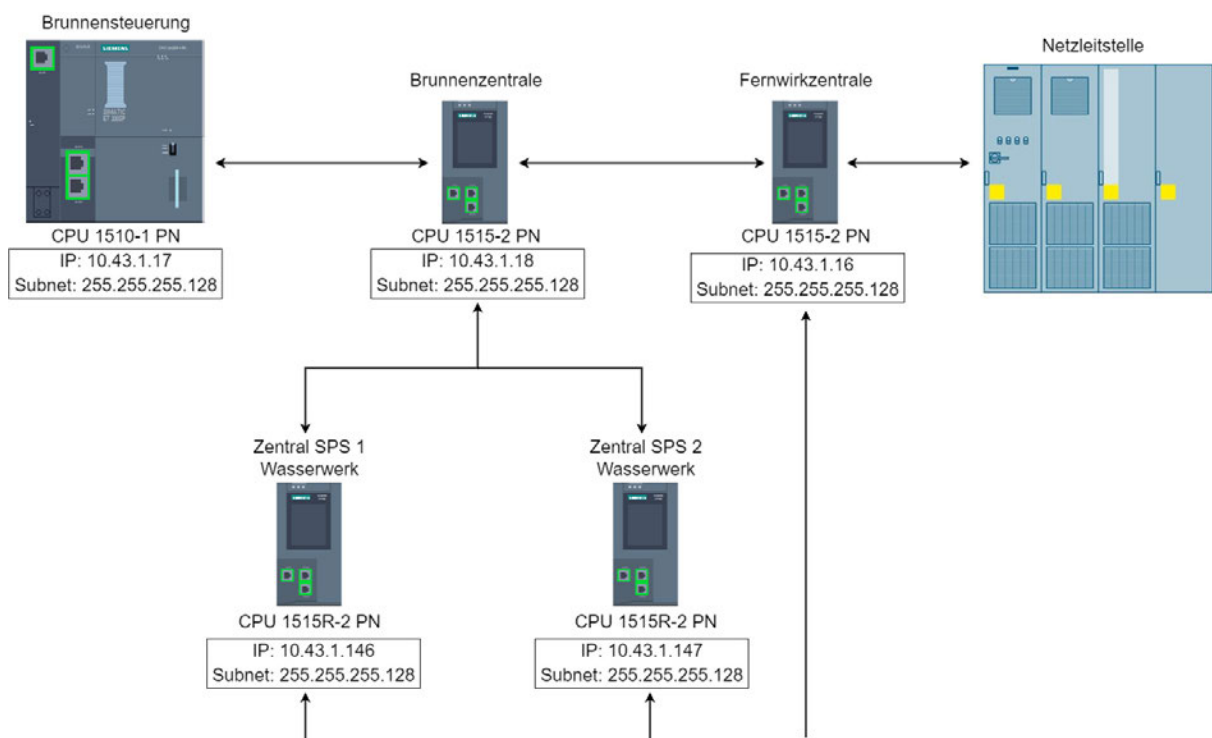


Abbildung 23: Vereinfachte Netzwerkübersicht

Das Netzwerk wird durch den Einsatz von Firewalls und VPN-Routern, sowie durch die Unterteilung in VLANs vor äußeren Zugriffen geschützt.

4.6 Zugriffsstrukturen auf die Steuerung des Brunnens

Wie an den Verbindungspfeilen in Abbildung 23 zu erkennen ist, verläuft die Kommunikation stets bidirektional zwischen den Kommunikationsteilnehmern. Der Zugriff auf die Steuerung kann über verschiedene Ebenen erfolgen. Unterschieden werden folgende Zugriffsebenen:

- Ortebene
- Nahebene
- Fernebene

Die Ortebene bezeichnet den Zugriff auf den Brunnen direkt über die im Brunnen verbaute SPS. Die Nahebene ermöglicht den Zugriff aus dem Wasserwerk heraus und die Fernebene bildet die Prozessleitebene in der Netzleitstelle ab.

Entsprechend müssen die erhobenen Daten in alle drei Ebenen kommuniziert werden. Aktiv auf den Prozess kann immer nur eine Ebene Einfluss nehmen.

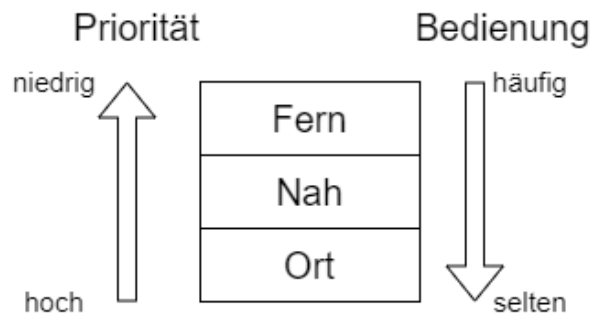


Abbildung 24: Übersicht über die Steuerungsebenen

Die Ebenen untereinander sind mit Prioritäten versehen. Die höchste hat die Ortebene, die niedrigste die Fernebene. Das Konzept erlaubt höher priorisierten Ebenen niedriger priorisierten die Steuerung zu entziehen. Entgegengesetzt ist dies nicht möglich. Dadurch wird eine höhere Sicherheit bei direkten Arbeiten am Brunnen erreicht.

Aufgrund der drei Steuerungsebenen liegen zu jeder Zeit drei Datensätze vor, zwischen denen unterschieden werden muss. Diese Datensätze beinhalten die wichtigsten Befehle und Meldungen für den Betrieb des Brunnens.

4.7 Datenaustausch zwischen Brunnen und Werk

Der Datenaustausch zwischen Brunnen und Wasserwerk wird durch die Brunnenzentrale organisiert und verläuft asynchron über vorgefertigte Kommunikationsbausteine. Diese sind nicht an den Bearbeitungszyklus der CPUs gebunden. Es sind sog. Put-Get-Bausteine, die dem Datenaustausch zweier Geräte in einem Netzwerk dienen. Die Geräte, die miteinander kommunizieren sollen müssen dem selben Profinet-Netzwerk angehören. [Sie20]

Bei diesen Bausteinen wird immer ein Partnergerät per IP-Adresse zugewiesen. Der Speicher dieses Partners wird, sofern die notwendigen Zugriffsrechte parametrisiert worden sind, gelesen und/oder geschrieben. Der Aufruf eines Put-Bausteins wirkt schreibend auf das Partnergerät,

der Get-Baustein wirkt lesend. Wichtig ist dabei, dass der Partner einen absoluten Bausteinzugriff erlaubt. Dieser ist notwendig, um auf die gewünschten Speicherbereiche zugreifen zu können. Über einen symbolischen Zugriff ist dies nicht möglich. Bei der absoluten Adressierung müssen die Speicheradressen in Form von Bytes und Bits angegeben werden. Beim symbolischen Zugriff kann der Anwender für jede dieser Speicheradressen Symbole, in Form von Bezeichnungen definieren, über die Daten aufgerufen werden können.

Bei Put-Bausteinen ist es grundsätzlich so, dass aus dem Speicher des Geräts, das die Kommunikation anstößt, Daten gelesen werden, die dann in die angegebenen Speicherbereiche des Partnergeräts geschrieben werden.

Der Get-Baustein wirkt entgegengesetzt. Die vom Partner gelesenen Daten müssen im Speicher des aufrufenden Geräts abgelegt werden.

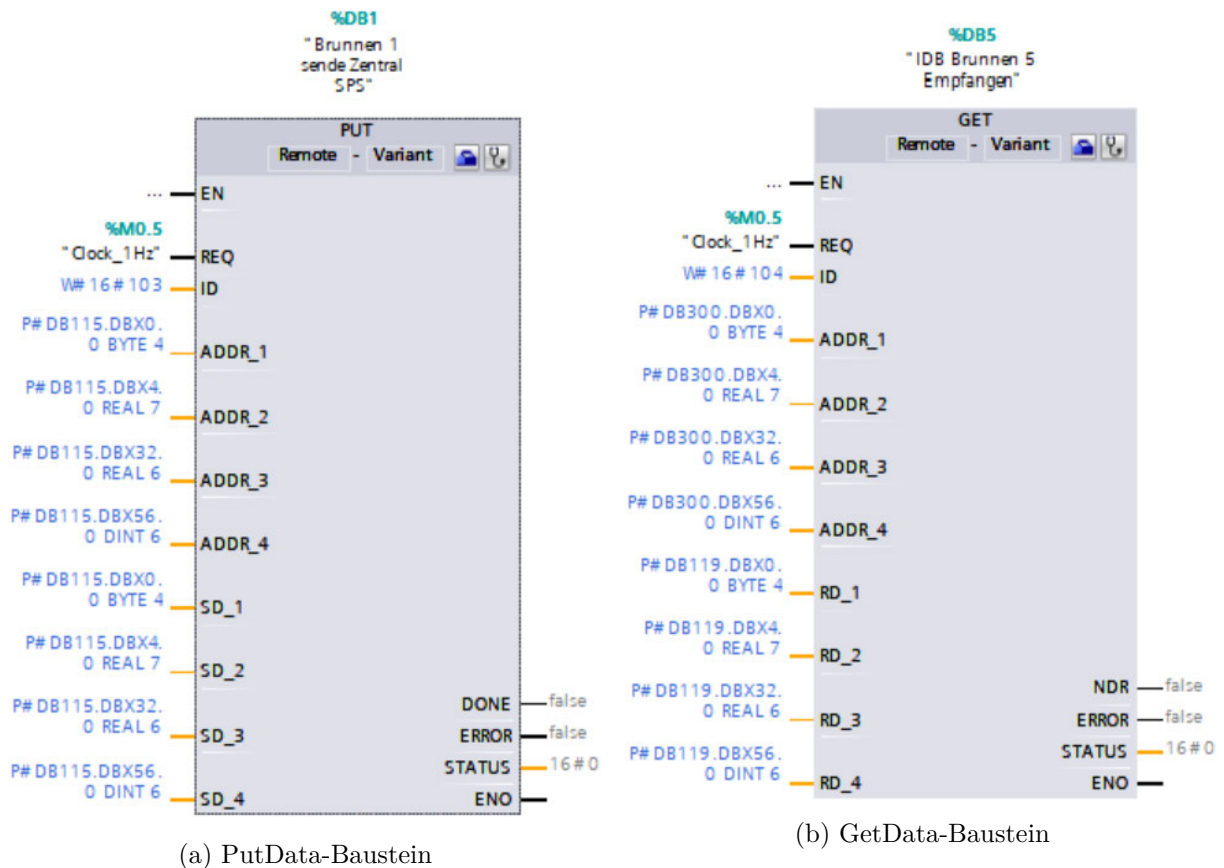


Abbildung 25: Put-Get-Kommunikationsbausteine

In der Steuerung des Brunnens und der Brunnenzentrale sind Kommunikations-DBs angelegt, auf die absolut zugegriffen werden kann. In diesen sind Sende- und Empfangsfächer eingerichtet. Alle Daten die ausgetauscht werden sollen, werden durch Funktionen der SPS in die Kommunikations-DBs kopiert. Dabei ist die SPS des Brunnens ein rein passiver Kommunikationspartner. Alle Übertragungen werden von der Brunnenzentrale angestoßen und durchgeführt. An dem Eingang *REQ* wird ein Taktmerkerbit angelegt, das bei Flankenwechsel die Kommunikation startet. An den *ADDR*-Eingängen werden die absoluten Speicherbereiche der Partner-CPU angelegt. Die Eingänge *SD_x* bzw. *RD_x* geben die Speicheradressen des kommunikationsführenden Geräts an. Die Speicheradressen werden als Pointer an die Bausteine übergeben.

5 Konzeptionierung und Umsetzung

5.1 Entwicklung eines Objektmodells

Die Softwareentwicklung geschieht unter dem Gesichtspunkt der objektorientierten Programmierung. Gegenüber der funktionalen Programmierung, die vor allem für die Entwicklung von Ablaufsteuerungen genutzt wird, bietet die Objektorientierung die Möglichkeiten einfacher Erweiterungen der Software und begünstigt eine starke Vernetzung der einzelnen Programnteile. Besonders in Hinblick auf verfahrenstechnische Anlagen, wie sie der Brunnen mitsamt des Grundwasserförderprozesses darstellt, ist diese Eigenschaft nicht zu vernachlässigen. Dabei schließt eine objektorientierte Programmierung die funktionale in keinsten Weise aus. Die Methoden der Klassen werden vollständig funktional programmiert und umgesetzt.

Das Zusammenspiel von objektorientierter und funktionaler Programmierung ist essenziell wichtig, um alle Bereiche des Prozesses bestmöglich abzudecken. Dabei wird dieser durch die Erstellung von Klassen in Bereiche aufgebrochen. Die in den jeweiligen Bereichen zu findenden Aufgaben und Problemstellungen werden durch die funktionalen Methoden der zuständigen Klassen behandelt.

5.1.1 Analogwertverarbeitung

Der FC Analogwertverarbeitung muss ein vielfältiges Aufgabenspektrum abdecken. Messumformer übertragen, wie bereits Kapitel 2.3.4 erklärt, oftmals ihre Daten über Stromsignale, die an die SPS über Analogeingangskarten angebunden sind. Diese Daten müssen skaliert und normiert werden. Andere Umformer verwenden Bussysteme, um ihre Daten zu übertragen. Häufig sind diese bereits skaliert und normiert. Der FC soll gängige Signaltypen verarbeiten können.

Weiterhin soll es möglich sein, messwertspezifische Alarm- und Warnmeldungen mit zugehörigen Hysteresen parametrieren zu können. Bei Überschreitung der Alarmgrenze soll ein Alarmbit gesetzt werden, das erst dann quittiert werden kann, wenn die Hysterese Grenze unterschritten wird. Die Hysterese wird genutzt, um u.a. Meldungen zeitlich versetzt zur Auslösung quittieren zu können. Der Unterschied zwischen Alarmen und Warnungen besteht darin, dass Alarme quittiert werden müssen. Sie zeigen kritische Anlagenzustände an. Warnungen hingegen müssen nicht quittiert werden. Auch sie werden bei einer Grenzwertverletzung ausgelöst. Die Warnungen sollen auf möglicherweise bevorstehende kritische Zustände hinweisen. An Abbildung 26 soll die Funktionsweise verdeutlicht werden. Zu sehen ist ein Messwert mit Alarm und Hysterese. Zum Zeitpunkt, in dem der Messwert die Alarmgrenze übersteigt, wird der Alarm ausgelöst. Gekennzeichnet ist dies mit dem Wirkpfeil (1). Der Alarm ist quittierbar, sofern der Messwert die Hysterese Grenze unterschreitet. Dieser Punkt ist mit einem Kreuz im oberen Graph markiert. Sobald das Quittierbit gesetzt und die Hysterese eingehalten wird, resettet der Alarm. In der Abbildung 26 ist dies durch den Wirkpfeil (2) gekennzeichnet.

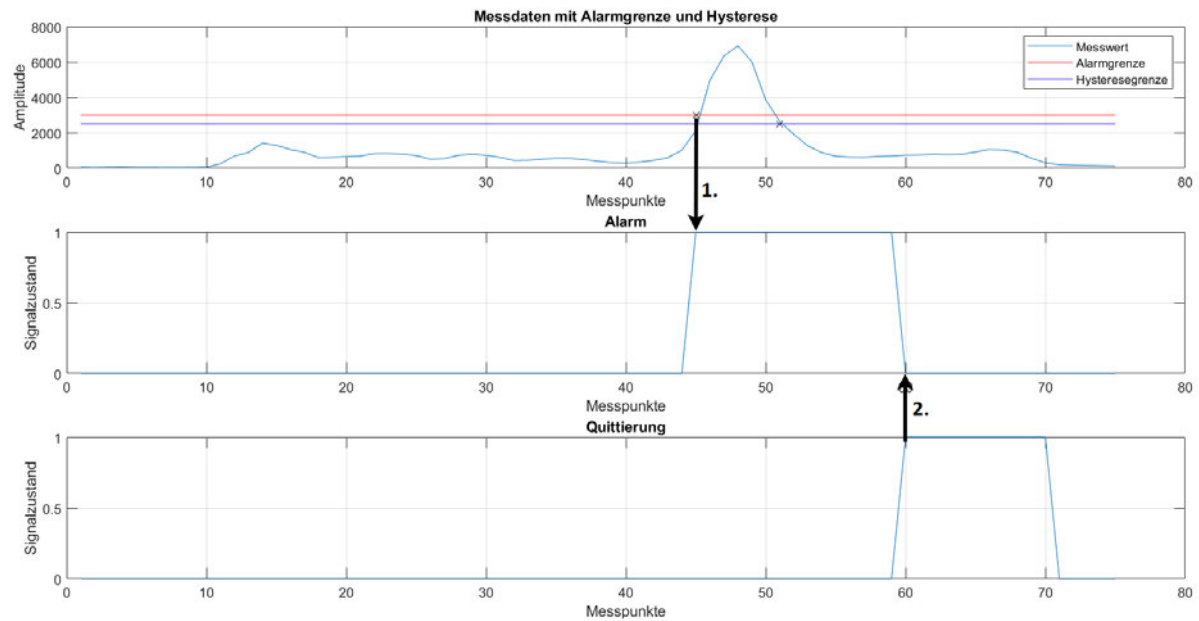


Abbildung 26: Beispielhafte Darstellung einer Alarmparametrierung

Die nachfolgende Tabelle 4 zeigt auf, welche Daten dem FC übergeben werden und welche dieser zurückgibt.

Mit der Input-Schnittstelle werden folgende Daten verknüpft:

1. Der aktuelle *Signalwert*, welcher je nach Übertragung an die Steuerung in unterschiedlichen Datentypen definiert sein kann. Die Flexibilität wird über den Datentyp „Variant“ ermöglicht. Dieser stellt einen Pointer dar, welcher das Auslesen des Datentyps der verknüpften Variable ermöglicht.
2. Der *Typ*, also die Art der Anbindung an die Steuerung. Diese kann über Anwenderkonstanten definiert werden.
3. Die Bezeichnung der Messstelle zur Identifikation und Anzeige im HMI.
4. Die SI-Einheit der entsprechenden Messstelle. Ebenfalls zur Anzeige im HMI.

Der Zugriff auf Variablen, welche über die InOut-Schnittstelle verknüpft sind, erfolgt lesend und schreibend.

Für die Analogwertverarbeitung relevant sind zum einen Parameter, welche über die Datenstruktur *MessDatenPara* an den FC übergeben werden. Zum anderen Aktualwerte, welche durch Abarbeitung des FCs in die Datenstruktur *MessDatenAktual* geschrieben werden.

Input			
Name	Datentyp	Defaultwert	Kommentar
Signalwert	Variant		aktueller Signalwert
Typ	Int		Art der Anbindung
MessBez	String		Bezeichnung der Messstelle
SI	String		SI-Einheit der Messstelle
InOut			
MessAktual	MessDatenAktual		Datenstruktur der Aktualwerte
MassPara	MessDatenPara		Datenstruktur der Parameter
Temp			
Warn_OGt	Real		Warnung obere Grenze abzüglich Hysterese
Warn_UGt	Real		Warnung obere Grenze zuzüglich Hysterese
Alarm_OGt	Real		Alarm obere Grenze abzüglich Hysterese
Alarm_UGt	Real		Alarm obere Grenze zuzüglich Hysterese
Val	Real		aktueller Signalwert als Real
Signal	Struct		Struktur der Signalwerte für verschiedene Datentypen
Constant			
(0)4_20_mA	Int	1	Signal zur Skalierung
Direkt	Int	2	Direktwert, z.B. als Bussignal
Drahtbruch	Int	-32768	Strom $I_E < 1,1852$ mA bei 16-Bit Datenwort

Tabelle 4: Bausteinschnittstelle FC Analogwertverarbeitung

Die letztliche Unterscheidung nach Art der Datenübertragung erfolgt über o.g. den Eingang *Typ*. Die darin enthaltenen Anwenderkonstanten ermöglichen eine Switch-Case-Verzweigung.

```

1 CASE Typ OF
2
3   4_20_mA:
4   REGION 4...20mA Signal
5     IF Signalwert = NULL THEN
6       Val := MessPara.Skalierung.Sign_min;
7     ELSIF TypeOf(Signalwert) = Word THEN //uebertragener Daten vom Typ Word
8       VariantGet(SRC := Signalwert,
9                 DST => Signal.Word);
10      Val := INT_TO_REAL(WORD_TO_INT(Signal.Word));
11    ELSE //Datentyp falsch
12      ;
13    END_IF;
14  END_REGION
15 .
16 .
17 .
18 END_CASE;
```

Listing 1: FC Analogwertverarbeitung: Unterscheidung nach Datenübertragung

Nach Unterscheidung des Signaltyps über die Switch-Case-Anweisung, wird in einer If-Verzweigung der Datentyp der an der Input-Schnittstelle *Signalwert* angelegten Variable ermittelt. Dies erfolgt unter Zuhilfenahme der Funktion *TypeOf()*.

Sofern diese ein gültiges Ergebnis liefert, z.B. *Word*, wird der Wert der Variablen über die Funktion *VariantGet()* ausgelesen und in einer temporären Variable zwischengespeichert. Im Anschluss wird diese in den Datentyp *Real* konvertiert, da die weitere Verarbeitung des Signalwerts diesen erfordert.

Am Beispiel eines 4 bis 20 mA Signals ist dies in Listing 1 von Zeile drei bis Zeile vierzehn dargestellt.

Für Stromsignale vom Typ 4-20 mA ist es notwendig, diese auf Drahtbruch zu prüfen. Dies geschieht unter Zuhilfenahme der Tabelle 1 aus Kapitel 2.2.2. Diese sagt aus, dass der Signalwert für Stromsignale, die kleiner als 1,1852 mA sind, -32768 beträgt. Ein Signal mit einem Strom kleiner des genannten Wertes wird als Drahtbruch gewertet.

```

1 //Drahtbrucherkenung bei Anbindung ueber 4 bis 20mA
2
3 IF Typ = (0)4_20_mA THEN
4     //Signalwert -32768 entspricht einem Strom I_E < 1,1852mA bei 16-Bit-Datenwort
5     IF Signal.Word = Drahtbruch AND NOT MessAktual.HMI-Simulation.Sim THEN
6         MessAktual.Aktual.Alarm.kDrahtbruch := FALSE;
7     ELSE
8         MessAktual.Aktual.Alarm.kDrahtbuch := TRUE;
9     END_IF;
10 END_IF;

```

Listing 2: FC Analogwertverarbeitung: Drahtbrucherkenung

Zu Testzwecken wird eine anwählbare Messwert-Simulation realisiert. Ist diese aktiv, erfolgt keine Überwachung auf Drahtbruch. Weiterhin wird der Signalwert, der sonst von der Peripherie gelesen wird, im FC überschrieben. Da im übrigen Programm die Messwert-Datenstruktur genutzt wird, lässt sich ein Test des kompletten SPS-Programms ohne angeschlossene Peripherie erreichen.

Eine weitere Funktion des FCs ist die Messwertskalierung und -normierung nach Gleichung 5. Dabei werden die in der Gleichung 5 genannten Variablen durch folgende Parameter ersetzt:

- $M \simeq$ Ergebnis: Skalierter Messwert
- $M_{max} \simeq$ Parameter: Physikalische Skalierungsgrenze Maximum
- $M_{min} \simeq$ Parameter: Physikalische Skalierungsgrenze Minimum
- $S \simeq$ Eingelesener Signalwert
- $S_{max} \simeq$ Parameter: Signal Skalierungsgrenze Maximum
- $S_{min} \simeq$ Parameter: Signal Skalierungsgrenze Minimum

Eine anschließende Überprüfung stellt die Einhaltung der gegebenen Skalierungsgrenzen sicher. Überschreitet der berechnete Messwert M die obere physikalische Skalierungsgrenze M_{max} , so wird M auf M_{max} begrenzt. Sollte andererseits der berechnete Messwert die untere physikalische Skalierungsgrenze M_{min} unterschreiten, so gilt $M = M_{min}$. Aus Tabelle 1 folgt, dass die Messwerte auch größer oder kleiner als die Skalierungsgrenzen sein können. Die Größen M_{min} und M_{max} stellen Begrenzungen dar, die u.a. das Anzeigen von Schleimengen¹¹ verhindern

¹¹ „Menge eines Mediums, die ein Messgerät (Zähler) nicht misst, z.B. Gasstrom, Wasserstrom“ [Rec]

sollen. In einigen Fällen kann dies aber durchaus gewollt sein. Dementsprechend muss die Überprüfung der Messwerte auf Skalierungsgrenzen für jeden Prozess einzeln bewertet werden.

Die Parametrierung der Messwerte über den PLC-Datentyp *MessDatenPara* erlaubt u.a. die Projektierung von Alarmen und Warnungen. Für jeden einzelnen Messwert können vier verschiedene Meldungen ausgewählt werden, um die Einhaltung von Grenzwerten zu überwachen, bzw. eine Über- oder Unterschreitung zu melden. Zu diesem Zweck stehen jeweils ein Alarm und eine Warnung zur Verfügung, die eine entsprechende Über- oder Unterschreitung kenntlich machen.

Für die Auswertung der Analogwertmeldungen am Brunnen erzeugt der FC Statuscodes für das HMI. Aus ihnen können anstehende Meldungen abgeleitet werden. Der Status zeigt immer nur einen Fehler bzw. Zustand an. Die Statuscodes sind als Statusbytes organisiert. Die Werte die sie annehmen können und was diese aussagen sind folgend aufgelistet:

- 16#00: Status ok
- 16#02: Status Drahtbruch
- 16#04: Status untere Alarmgrenze unterschritten
- 16#08: Status obere Alarmgrenze überschritten
- 16#10: Status untere Warnungsgrenze unterschritten
- 16#20: Status obere Warnungsgrenze überschritten

Der Bausteinaufruf mit zugehöriger Verschaltung ist in Abbildung 27 zu sehen.

1	CALL "Messwertverarbeitung"	%FC2200
2	Signalwert := "Niveausonde"	%IW4
3	Typ := "(0) 4_20_mA"	1
4	MessBez := 'Pegel'	'Pegel'
5	SI := 'm'	'm'
6	MessAktual := "DB_MessDatenAktual".MessAktual[1]	
7	MessPara := "DB_MessDatenPara".MessPara[1]	
8		

Abbildung 27: Aufruf und Verschaltung des Analogwertverarbeitungs-FC

5.1.2 Digitalwertverarbeitung

Für die Steuerung des Prozesses, aber auch zur programmierten Verwendung, ist es sinnvoll, Digital-Eingangssignale zu manipulieren. Dies beinhaltet in erster Linie Ein- und Ausschaltverzögerung und Signalinvertierung.

Der FC zur Digitalwertverarbeitung schafft entsprechende Optionen. Wie auch bei der Analogwertverarbeitung sind die Optionen an- bzw. abwählbar und über die Parameter einstellbar.

Nachfolgende Tabelle 5 zeigt die Schnittstelle des Bausteins.

Input			
Name	Datentyp	Defaultwert	Kommentar
Signal	Bool		DI-Signal der Peripherie
1sekTakt	Bool		Sekundenpuls des CPU Takts in 1sek. Schritten
InOut			
DEDatenAktual	DEDatenAktual		Datenstruktur der Aktualwerte
DEDatenPara	DEDatenPara		Datenstruktur der Parameter
Temp			
1sekFlanke	Bool		1 Sekunden Flanke
valinv	Bool		interne Signalauswertung: Invertierung
valET	Bool		interne Signalauswertung: Einschaltverzögerung
valAT	Bool		interne Signalauswertung: Ausschaltverzögerung

Tabelle 5: Bausteinschnittstelle FC Digitalwertverarbeitung

Am Eingang *Signal* wird das DI-Signal der Peripherie angelegt. Dieses wird anschließend bei Bedarf durch den FC manipuliert. Die Parametrierung der o.g. Möglichkeiten geschieht über die InOut-Variable *DEDatenPara*. Sie enthält die Parameter des jeweiligen Signals zur Selektion der Manipulationsmöglichkeiten, sowie der zugehörigen Zeitangaben für die Ein-, bzw. Ausschaltverzögerung. Das bearbeitete Signal des digitalen Eingangswertes wird in der Struktur *DEDatenAktual* abgelegt. Dort kann es für die weitere Verwendung abgegriffen werden.

Weiterhin beinhaltet die genannte Struktur den Rohwert des Signals, also den unbearbeiteten Peripheriewert, als auch Angaben zu den möglicherweise gewählten Ein- und Ausschaltverzögerungen.

Die Signalinvertierung erfolgt durch die XOR-Verknüpfung des DI-Signals mit dem Selektionsbit für die Invertierung, wie in Listing 3 dargestellt.

```

1 //Wem ausgewaehlt wird das Signal invertiert
2
3 valinv := Signal XOR DEDatenPara.Sel_inv;
```

Listing 3: FC Digitalwertverarbeitung: Invertierung

Digitales Eingangssignal	Selektionsbit	Manipuliertes Signal
0	0	0
0	1	1
1	0	1
1	1	0

Tabelle 6: Wahrheitstabelle XOR

Die Wahrheitstabelle 6 zeigt den Zusammenhang zwischen Selektionsbit, digitalem Eingangssignal und dem Ergebnis der Verknüpfung, also dem manipulierten Signal.

Für die zeitlich wirkenden Signalverzögerungen werden Timer benötigt. Bei Verwendung der von Siemens vorgefertigten *TON*-Timer wird ein Instanzaufruf getätigt, wodurch der FC seine Bibliotheksfähigkeit verlieren würde. Aus diesem Grund müssen die Timer ebenfalls programmiert werden.

Dies geschieht unter Zuhilfenahme eines Taktmerkerbits, welches die CPU zur Verfügung stellt. Dieses wird an den FC-Eingang *1sekTakt* angelegt. Für den FC wird der 1 Hz-Taktmerker verwendet. Die Taktmerkerbits erzeugen zu jedem Beginn der Periode eine steigende Flanke. Beispielsweise ist der 1 Hz-Taktmerker je eine halbe Sekunde 1 und anschließend eine halbe Sekunde 0, sodass nach einer vollen Sekunde wieder eine steigende Flanke erfassbar ist. Die Erfassung von Flankenwechseln des Taktmerkers ermöglicht das Zählen von Sekunden. Bei jeder steigenden Flanke wird eine Zählvariable um Eins erhöht. Der Wert der Zählvariablen entspricht den seit dem Start des Timers vergangenen Sekunden. Die Flankendetektion wird durch den Vergleich des aktuellen Zustands des Taktmerkerbits mit dem Zustand aus dem vorangegangenen Zyklus umgesetzt. Tritt hier eine Differenz auf, wird ein Flankenwechsel erkannt.

```

1 REGION @Flankenauswertung 1 sek Puls
2     //Flanke
3     1sekFlanke := 1sekTakt AND NOT DEDatenAktual.Intern.Flankenpuls
4
5     //Ruecksichern Status
6     DEDatenAktual.Intern.FlankePuls := 1sekTakt;
7 END_REGION

```

Listing 4: FC Digitalwertverarbeitung: Flankenauswertung

Bevor die jeweiligen Timer aber gestartet werden, müssen die Startbedingungen geprüft werden. Grundlegend wird kontrolliert, ob die jeweilige Signalverzögerung ausgewählt wurde und anschließend, ob der Timer nicht evtl. schon abgelaufen ist.

Die Einschaltverzögerung startet, sobald ein Signalwechsel von *FALSE* auf *TRUE* erfolgt.

```

1 IF DEDatenPara.Sel_EVerz AND valinv AND NOT DEDatenAktual.Intern.Altwert THEN
2     IF DEDatenAktual.Aktual.EVerzT < DEDatenPara.EVerzT THEN
3         valET := DEDatenAktual.Intern.Altwert;
4         IF 1sekFlanke THEN
5             DEDatenAktual.Aktual.EVerzT += 1; //Aktualwert Timer erhoehen
6         END_IF;
7     ELSIF DEDatenAktual.Aktual.EVerzT >= DEDatenPara.EVerzT THEN
8         //Timer abgelaufen
9         valET := valinv;
10    END_IF;
11 ELSE
12    (*Timer zuruecksetzen bei Signalabfall oder nicht ausgewaehlter
13    Einschaltverzoeigerung*)
14    DEDatenAktual.Aktual.EVerzT := 0;
15    valET := FALSE;
16 END_IF;

```

Listing 5: FC Digitalwertverarbeitung: Einschaltverzögerung

Die Ausschaltverzögerung funktioniert prinzipiell nach dem selben Schema, nur startet diese bei einem Signalwechsel von *TRUE* zu *FALSE*.

Der Aufruf und die Verschaltung des FCs geschieht wie in Abbildung 28 gezeigt.

1	<code>CALL "DigitaleEingaenge"</code>	%FC21...
2	<code>Signal := "Pumpe ein"</code>	%I3.2
3	<code>lsekTakt := "Clock_1Hz"</code>	%M0.5
4	<code>DEDatenAktual := "DB_DEDatenAktual".DEAktual[19]</code>	
5	<code>DEDatenPara := "DB_DEDatenPara".DAPara[19]</code>	
6		

Abbildung 28: Aufruf und Verschaltung des Digitalwertverarbeitungs-FC

5.1.3 Zähler

Für die Erfassung von Impulswerten besteht die Notwendigkeit einen Impulzzählerbaustein zu programmieren. Dabei werden vom FC zwei Zähler geführt. Der Zähler 1 soll vorrangig Betriebszählwerte für das HMI erfassen und ist dabei nur über das HMI zurücksetzbar. Der Zähler 2 ist als PLC-Zähler u.a. für den Aufruf von Programmfunktionen gedacht und bietet zusätzlich die Möglichkeit über die Steuerung zurückgesetzt zu werden. Dafür wird die in Tabelle 7 gezeigte Bausteinschnittstelle geschaffen.

Input			
Name	Datentyp	Defaultwert	Kommentar
ImpEin	Bool		Impulseingang
Reset_HMI_Cnt1	Bool		Rücksetzeingang Zähler 1 über HMI
Reset_HMI_Cnt2	Bool		Rücksetzeingang Zähler 2 über HMI
Reset_Strg	Bool		Rücksetzeingang Zähler 2 über die Steuerung
Wertigkeit	Real		Wertigkeit eines Pulses
InOut			
Cnt1	Real		Zähler 1: Zählwert
Cnt2	Real		Zähler 2: Zählwert
Flnk	Bool		Flankenmerker

Tabelle 7: Bausteinschnittstelle FC Impulzzähler

Die Impulserfassung erfolgt über die Detektion steigender Flanken am Eingang *ImpEin*. Dafür wird ein Flankenmerkerbit *Flnk* genutzt. Das Auftreten einer positiven Flanke erhöht die Zählwerte beider Zähler um die angegebene Wertigkeit.

```

1 IF ImpEin AND NOT Flnk THEN
2   Cnt1 += Wertigkeit;
3   Cnt2 += Wertigkeit;
4 END_IF;
5
6 //Zuweisung des letzten Zustands des Impulseingangs auf den Flankenmerker
7 Flnk := ImpEin;
```

Listing 6: FC Impulzzähler

Genutzt werden die Impulzzähler, um u.a. die Mengenzähler zu bedienen. Hier werden die Tageszählmenge, die Gesamtzählmenge und die Tageszählmenge des vorherigen Tages gespeichert.

Für die Tageszählmenge ist es notwendig auf die Lokalzeit der CPU zuzugreifen. Über ein Bit, das zu Mitternacht gesetzt wird, wird der Wert der Variablen *Zaehlwert_heute* in die Variable *Zaehlwert_gestern* geschrieben. Anschließend erfolgt die Rücksetzung des Tagesmengen Zählers. Mehrere FUP-Netzwerke setzen diese Anweisungen um. Das Netzwerk, welches das Mitternachtbit setzt, ist in Abbildung 29 dargestellt. FUP wird hier einmalig zur Veranschaulichung des komplexen booleschen Ausdrucks zum Setzen des Mitternachtbit genutzt. Dadurch wird Verständlichkeit erleichtert.

Aufgrund der zyklischen Programmabarbeitung ist eine sekundengenaue Abfrage notwendig. Es wird darauf verzichtet den Millisekundenbereich mit einzubeziehen, da dieser zu klein für den Programmzyklus ist. Die verwendeten Uhrzeitdaten werden von der CPU zur Verfügung gestellt..

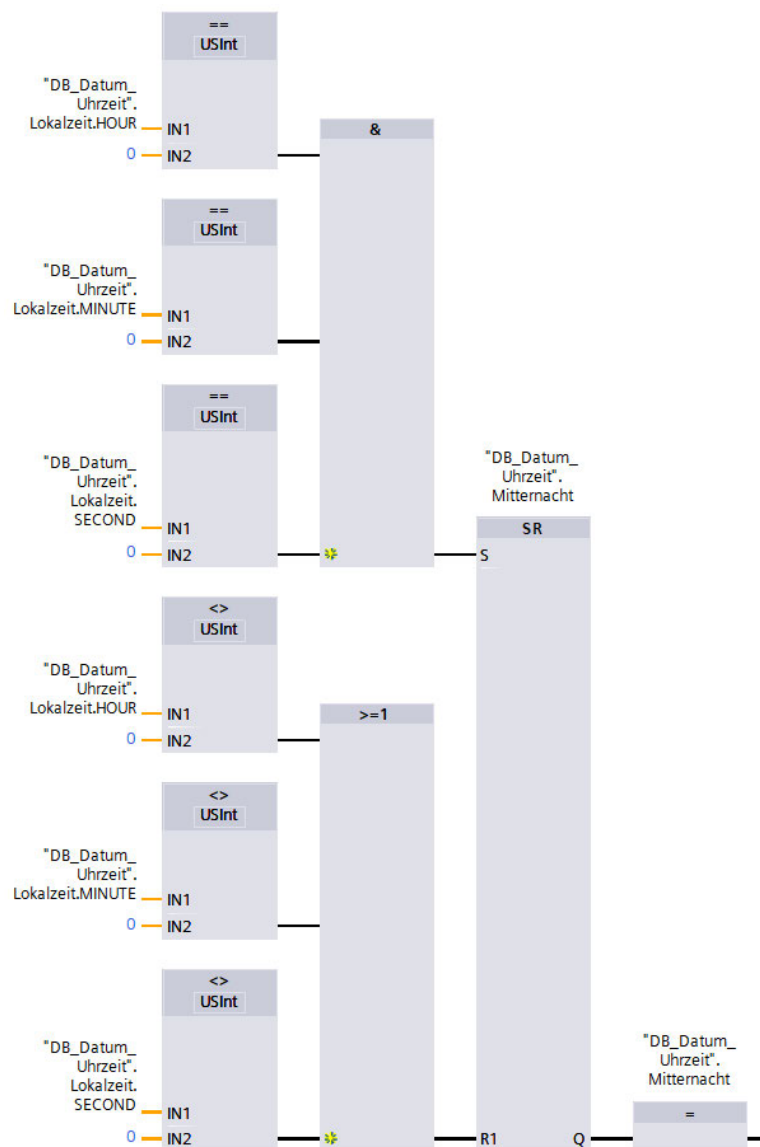


Abbildung 29: FUP-Netzwerk zum Setzen und Rücksetzen des Mitternachtbits

Der einmalige Aufruf des Impulzzähler-FCs setzt Tagesmengen zähler und Gesamt mengen zähler um. Der Zähler 1 entspricht dem Gesamt mengen zähler. Dieser kann nur durch das HMI

zurückgesetzt werden. Der Zähler 2 stellt dementsprechend den Tagesmengenähler dar. Durch das Anlegen des Mitternachtbits an den FC-Eingang *Reset_Strg* wird dieser stets zu Mitternacht resettet.

5.1.4 Frequenzrichter mit Direktansteuerung

Wie in Kapitel 2.2.5 beschrieben, können FUs über Analogsignale oder über Bussysteme mit der SPS kommunizieren. Aus der analogen Kommunikation ergeben sich Aufgaben zur Skalierung der Soll- und Istwerte. Diese und weitere Aufgaben setzt der FC zur Direktansteuerung für FUs um, die nicht über eine Busverbindung mit der SPS verfügen. Für FUs, die über Busverbindungen an die SPS angebunden sind, ist der FC nicht nutzbar. Die Funktion übernimmt die Sollwertverarbeitung, die Erzeugung eines Start-Befehls, die Überwachung eingestellter Signalverzögerungen und das Erstellen von Meldungen. Weiterhin wird das Monitoring von Betriebsstunden und Schaltspielen, sowie die Berechnung von wichtigen Motordaten für die angeschlossene ASM abgedeckt.

An der Schnittstelle (siehe Tabelle 8) werden die Eingangssignale *Freigb* und *Ein_Aus* erwartet. Die Verknüpfung der Freigabe erfolgt außerhalb des FCs, ist aber grundlegend für die Ansteuerung. Erst wenn sie erteilt ist, kann der FU gestartet werden. Gestartet wird, wenn sowohl die Freigabe, als auch das Einschaltsignal vorliegen.

Die benötigten Parameter werden der Datenstruktur *FUPara* entnommen. Der FC legt die ermittelten Aktualdaten in der Struktur *FUAktual* ab.

Der Sollwert des FUs, sowie der Startbefehl und die berechnete *Drehzahl* der angeschlossenen ASM sind als Output-Variablen angelegt. Die Drehzahl berechnet sich nach Gleichung 2.

Input			
Name	Datentyp	Defaultwert	Kommentar
Clk_1Hz	Bool		Taktsignal 1Hz
Freigb	Bool		Freigabesignal
Ein_Aus	Bool		Ein = 1, Aus = 0
Istw	Word		Istwert
Output			
Sign	Word		Signalwert 0 bis 27648 an FU
DQ=>FU-Start	Bool		Digitalausgang zu FU: Startbefehl
Drehzahl	Real		Istdrehzahl des angeschlossenen Motors
InOut			
FUPara	FUPara		Datenstruktur der Parameter
FUAktual	FUAktual		Datenstruktur der Aktualwerte
Temp			
Val_out	Int		Sollwert als Integer-Signalwert
Val_in	Int		Istwert als Integer-Signalwert
Pole	Int		Polpaarzahl der ASM
Sel_Ein	Bool		Auswahl: Einschaltverzögerung
Sel_Aus	Bool		Auswahl: Ausschaltverzögerung

Tabelle 8: Bausteinschnittstelle FC FU-Ansteuerung

Die für die o.g. Sollwertverarbeitung notwendigen Parameter sind der Variablen *FUPara* zu entnehmen. Die hier eingestellten prozentualen Sollwertgrenzen legen den Arbeitsbereich des FUs fest. Der eingestellte Sollwert wird vor der Skalierung auf diese Grenzen geprüft. Liegt der

Sollwert außerhalb des parametrisierten Bereichs wird er auf die jeweilige Bereichsgrenze gesetzt. Die anschließende Skalierung geschieht nach der Gleichung 4, die in Kapitel 2.2.2 erläutert wird. Der eingestellte Sollwert wird dadurch auf den zugehörigen Signalwert umgerechnet. Die Komponenten der Gleichung 4 werden durch folgende Parameter und Variablen ersetzt:

- $S \simeq$ Ergebnis: Sollwert des FUs als Signalwert
- $S_{max} \simeq$ Parameter: Signal Skalierungsgrenze Maximum
- $S_{min} \simeq$ Parameter: Signal Skalierungsgrenze Minimum
- $M \simeq$ Sollwert vom Typ *Real*
- $M_{max} \simeq$ Parameter: Physikalische Skalierungsgrenze Maximum
- $M_{min} \simeq$ Parameter: Physikalische Skalierungsgrenze Minimum

Im Anschluss an die Skalierung folgt die Kontrolle der Skalierungsgrenzen und ggf. die Anpassung des Signalwerts *Val_out*. Danach wird über die Funktion *INT_TO_WORD()* der Wert der temporären Variable *Val_out* auf den Output *Sign* zugewiesen.

Der vom FU zurückgegebene Signalwert enthält die aktuelle Frequenz oder Pumpendrehzahl als prozentualen Wert. Da dieser als 4-20 mA Signal vorliegt, muss dieses durch den FC über Gleichung 5 skaliert werden.

Wie in Kapitel 2.2.5 beschrieben, erfordern einige FUs ein digitales Startsignal. Dieses stellt der FC bereit. Die Funktion kann dem Listing 7 entnommen werden.

```

1 REGION @DQ-Startbefehl
2     IF Freigb AND Val_out <> 0 THEN
3         DQ=>FU-Start := TRUE;
4     ELSE
5         DQ=>FU-Start := FALSE;
6     END_IF;
7 END_REGION

```

Listing 7: FC FU-Ansteuerung: DQ-Startbefehl

Für den Betrieb der FUs ist es über den FC möglich Ein- und Ausschaltüberwachungen einzustellen. Dabei werden in *FUPara.Timer* maximale Einschalt- bzw. Ausschaltzeiten festgelegt, die vom FC auf Einhaltung kontrolliert werden.

Bei der Einschaltüberwachung wird eine Meldung generiert, sollte nach der maximal erlaubten Einschaltzeit kein Rückmeldesignal anliegen. Als Rückmeldung wertet der FC eine Frequenz- bzw. Drehzahlanpassung des FUs.

Nach dem selben Prinzip arbeitet die Ausschaltüberwachung. Nach Vorliegen des Ausschaltbefehls oder Entzug der Freigabe muss der FU innerhalb der eingestellten maximalen Ausschaltzeit den Istwert auf 0% heruntergefahren haben. Auch hier wird bei Nichteinhaltung eine Meldung erzeugt. Die Meldungen der Einschalt- und Ausschaltüberwachungen entziehen die erneute Betriebsfreigabe des FUs und müssen dementsprechend quittiert werden.

Die beiden Überwachungen können selektiv ausgewählt werden. Die Umsetzung erfolgt über Timer, die auf Flankendetektion beruhen. Der dafür genutzte Taktmerker wird am Eingang

Clk_1Hz angelegt. Der folgende Programmcode aus Listing 8 zeigt die Umsetzung der Ausschaltüberwachung. Mit kleinen Anpassungen ist dieser Code auch auf die Einschaltüberwachung anwendbar.

```

1 REGION @Ausschaltueberwachung
2   IF Sel_Aus THEN
3     IF (FUAktual.Istw > 0) AND (NOT Freigb OR Ein_Aus OR (Val_out = 0))
4       AND NOT FUPara.Meldungen.AusschlStr THEN
5
6       //Flankendetektion des Timers
7       IF (Clk_1Hz AND NOT FUPara.Timer.Ausschalten.Flnk) THEN
8         FUPara.Timer.Ausschalten.Sec += 1;
9       END_IF;
10
11      //Ruecksichern des Taktmerkerzustands
12      FUPara.Timer.Ausschalten.Flnk := Clk_1Hz;
13
14      //Bedingung fuer Ausloesen der Ausschaltstoerung
15      IF FUPara.Timer.Ausschalten.Sec >=
16        FUPara.Ausschalten.Timer.Ausschaltzeit_max
17        AND (FUAktual.Istw > 0) THEN
18        // Zeit abgelaufen und Betriebsmeldung steht nicht an
19        FUPara.Meldungen.AusschlStr := TRUE;
20        FUPara.Meldungen.SmlStr := TRUE;
21
22        //Timer nach Ablauf der Zeit zuruecksetzen
23        FUPara.Timer.Ausschalten.Sec := 0;
24
25        ELSIF NOT FUPara.Meldungen.AusschlStr THEN
26          //Zeit noch nicht abgelaufen oder Meldung rechtzeitig
27          FUPara.Meldungen.AusschlStr := FALSE;
28        END_IF;
29      END_IF;
30
31      //Timer resetten sobald keine Betriebsmeldung mehr vorliegt
32      IF FUAktual.Istw = 0 THEN
33        FUPara.Timer.Ausschalten.Sec := 0;
34      END_IF;
35
36    END_IF;
37
38  END_REGION

```

Listing 8: FC FU-Ansteuerung: Ausschaltüberwachung

Das Setzen eines Quittierbits setzt die ausgelösten Meldungen zurück. Wie bereits oben angemerkt, wird vom FC das Monitoring von Schaltspielen und Betriebsstunden übernommen. Als Schaltung wird die Änderung des Betriebszustands des FUs verstanden. Entsprechend sind diese über Flankendetektionen zu erfassen.

```

1 REGION @Schaltspiele
2   //Flankendetektion
3   IF ((FUAktual.Istw <> 0) AND NOT FUPara.Counter.Flnk) OR
4     ((FUAktual.Istw = 0) AND FUPara.Counter.Flnk) THEN
5     FUPara.Monitoring.Schalt += 1;
6   END_IF;
7
8   //Ruecksichern des Betriebszustands
9   FUPara.Counter.Flnk := (FUAktual.Istw <> 0);
10 END_REGION

```

Listing 9: FC FU-Ansteuerung: Monitoring Schaltspiele

Die Betriebsstunden werden wiederum über den Einsatz eines internen Timers umgesetzt. Sie dienen u.a. als Richtwert für Wartungen des Motors und des FUs.

```

1 REGION @Betriebsstunden
2   IF FUAktual.Istw <> 0 THEN
3     //Flankendetektion
4     IF (Clk_1Hz AND NOT FUPara.Timer.BetriebsstundenTimer.Flnk) THEN
5       FUPara.Timer.BetriebsstundenTimer.Sec += 1;
6     END_IF;
7
8     //Ruecksichern des Taktmerkerzustands
9     FUPara.Timer.BetriebsstundenTimer.Flnk := Clk_1Hz;
10    END_IF;
11
12    //Berechnung der Betriebsstunden
13    IF FUPara.Timer.BetriebsstundenTimer.Sec = 3600 THEN
14      FUPara.Monitoring.Betr_std += 1;
15      FUPara.Timer.BetriebsstundenTimer.Sec := 0;
16    END_IF;
17 END_REGION

```

Listing 10: FC FU-Ansteuerung: Monitoring Betriebsstunden

Das Zurücksetzen der Monitoringdaten ist über das HMI realisierbar.

Neben den Monitoringdaten werden die Frequenz und die Drehzahl der angeschlossenen ASM zur Verfügung gestellt. Dafür werden die Nenndrehzahl und die Nennfrequenz in *FUPara.Motor* angegeben. Aus Gleichung 2 kann die Polpaarzahl der ASM bestimmt werden.

Weiterhin muss das am FU eingestellte Frequenzband angegeben werden, das auf die prozentualen Wert 0% bis 100% abgebildet wird. Über die bekannten Gleichungen zur Skalierung wird aus dem prozentualen Istwert der frequenzielle Istwert gebildet. Aus dem Frequenz-Istwert kann anschließend die Motordrehzahl in min^{-1} berechnet werden. Diese Werte der ASM dienen als zusätzliche Informationen und haben keinen Einfluss auf die Steuerung.

5.1.5 Datenmodell

Die objektorientierte Programmierung erfordert die Erstellung von Klassen, die über ihre Attribute und Methoden Aufgaben erfüllen. Dies kann über mehrere Wege umgesetzt werden. Der standardisierte Weg ist es, für jede Klasse einen FB zu erstellen. Aufgrund seines IDBs werden

hier die Attribute der Klasse festgeschrieben. Die Methoden werden anschließend im FB angelegt.

Um nicht für jedes Objekt einer Klasse einen einzelnen DB schaffen zu müssen, werden die Klassen aus anderen Programmkomponenten zusammengesetzt. Die Attribute der Klassen werden in PLC-Datentypen festgelegt. Zusammen mit speziell auf diese Datentypen angepassten FCs werden die Methoden der Klasse erzeugt. Die genutzten Objekte der Klasse werden nicht durch IDBs dargestellt, sondern sind als Array in Datenbausteinen organisiert.

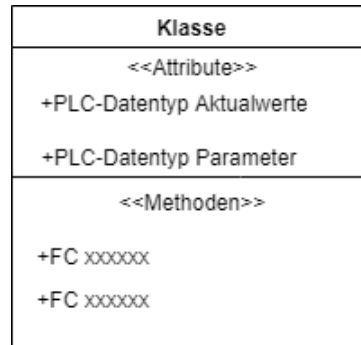


Abbildung 30: Struktureller Aufbau der erstellten Klassen

Die Klassenattribute werden dabei durch zwei PLC-Datentypen verkörpert. Einer der beiden stellt alle Aktualparameter¹² der zugehörigen Objekte dar. Hier sind alle Parameter zugeordnet, deren Speicherung nicht remanent¹³ sein muss. Die remanenten Daten sind im PLC-Datentyp der Parameter angeordnet. Dadurch wird der Speicherbedarf der Objekte stark minimiert. Grund dafür ist die Array-Struktur, in der die Objekte angeordnet sind. Da in DBs nur das gesamte Array remanent gesetzt werden kann, würden bei einer nicht vollzogenen Trennung auch alle Aktualparameter in die Remanenz überführt werden. Dies hätte eine hohe Speicherauslastung des Remanenzspeichers zur Folge.

Die Entscheidung, die Objekte in Arrays zu organisieren, beruht auf dem Bedien- und Datenkonzept des HMIs.

5.2 Entwicklung eines Visualisierungskonzepts

Das HMI ist die Schnittstelle des Prozesses zu den Bedienern. Um eine umfassende Visualisierung zu projektieren, wurden Richtlinien und Konzepte umgesetzt. Ziel der Visualisierung ist eine software-ergonomisch optimierte Darstellung der verfahrenstechnischen Anlage. Die Bedienung durch das HMI soll intuitiv sein.

5.2.1 Farbgebung

Für die Software-Ergonomie spielt die Farbgebung eine essenzielle Rolle. Als Grundfarbton wird ein helles Grau gewählt. Die Farben werden nach dem RGB¹⁴-Koordinatensystem bezeichnet. Das verwendete Grau hat die Koordinaten (192, 192, 192). Es wirkt sehr formal, wodurch die projektierten Anzeigen stark durch ihre Farbgebung in den Vordergrund gestellt werden. Die

¹²Absoluter Wert einer Variablen der zu einem bestimmten Zeitpunkt vorliegt

¹³Remanente Variablen erhalten ihre Daten auch bei Spannungsausfall

¹⁴Rot-Grün-Blau-Farbsystem

wichtigen Informationen heben sich somit hervor.

Es wird nur eine begrenzte Anzahl an Farben genutzt. Aus der VDI/VDE 3699 folgen Empfehlungen an Farben und Darstellungen für die Visualisierung verfahrenstechnischer Prozesse.[VDI14]

- Rot: Gefahr
- Orange: Warnung
- Gelb: Vorsicht
- Grün: Sicherheit, oder auch ein bzw. auf
- Weiß/Grau: Allgemeine Information, oder auch aus bzw. geschlossen
- Blau: Spezielle Information

Die Farbgebungen finden sich in allen Teilen der Visualisierung wieder. Dabei wird besonders auf Einheitlichkeit geachtet. Es soll beispielsweise vermieden werden, unterschiedliche Grüntöne für Anzeigen gleicher Bedeutung zu verwenden. Selbiges gilt für die anderen genannten Farben, aber auch für Buttons, Anzeigen und anderen Visualisierungselemente.

Es wird weitestgehend darauf verzichtet, das für Hintergründe eingesetzte Grau für Bildelemente zu nutzen. Dadurch heben sich die prägnanten Farben vom Hintergrund ab, sodass schnell der Zustand der Anlage erfasst werden kann.

Die Farbe von Ein- und Ausgabefeldern (kurz: E/A-Felder) gibt an, ob diese lesend oder schreibend genutzt werden können. Eingabefelder werden weiß dargestellt, reine Ausgabefelder grau.

5.2.2 Bilder

Die Visualisierung des Prozesses beruht auf der Verwendung von HMI-Bildern. Diese stellen alle wichtigen Informationen dar. Darunter sind z.B. die Messwerte, Berechnungen und Meldungen zu verstehen. In allen Bildern ist ein Permanentbereich am oberen Rand projektiert. Dieser zeigt das aktuelle Datum und die Uhrzeit an, sowie das Logo der SWL. Weiterhin ist hier eine Schaltfläche zu finden, bei deren Bedienung unten im Bild ein Slide-In-Bild¹⁵ eingeblendet wird. Dieses Slide-In dient als Umschaltleiste zwischen den einzelnen HMI-Bildern. Dies wird über die Bedienung der hier fest zugeordneten Funktionstasten ermöglicht.

Das Hauptbild der Visualisierung ist die Anlagenübersicht.

Angelehnt an das RI-Fließbild (Markierung 1) des Brunnens aus Abbildung 21 ist hier die Anlage mittig im Bild dargestellt. Die genutzten Symbole sind nach in Kapitel 5.2.1 aufgeführten Konvention dargestellt. Die gemessenen Werte, sowie die Messstellenbezeichnung und die SI-Einheit werden hier aufgeführt. Messstellenbezeichnung und SI-Einheit können optional über die Schaltfläche *Bezeichnungen einblenden/ausblenden* (Markierung 2) verborgen werden. Über die erneute Betätigung des Buttons werden die Bezeichnungen und SI-Einheiten wieder eingeblendet.

Weiterhin sind auf diesem Bild wichtige Größen wie Abstich und Pegel abgebildet (siehe Markierung 3).

¹⁵Bei Bedarf einblendbare Bilder, die an den Rändern des HMIs projektiert sind. Im inaktiven Zustand sind sie nicht sichtbar.

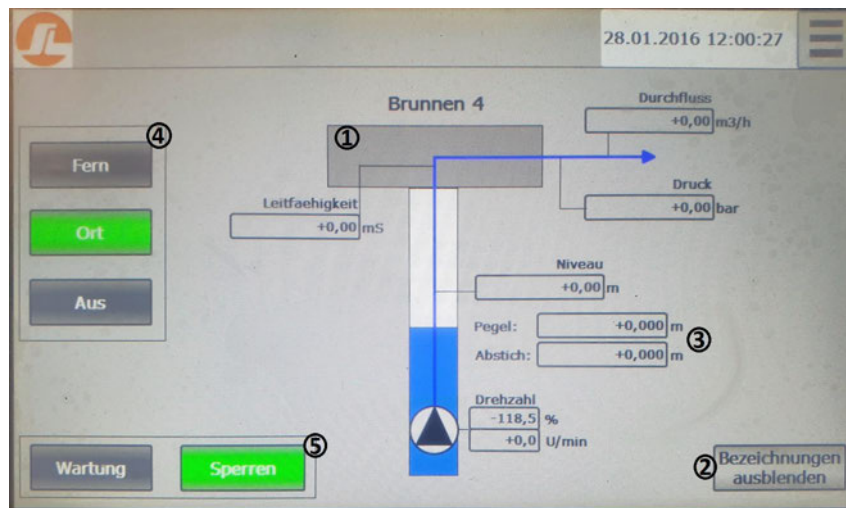


Abbildung 31: Anlagenübersicht in der Prozessvisualisierung

Die sich links befindenden Schaltflächen (Markierung 4) sind für die Umschaltung zwischen den Zugriffsebenen (siehe Kapitel 4.6) verantwortlich. Aus Anlagensicht besteht hier nur die Notwendigkeit zwischen Ort- und Fernzugriff zu unterscheiden.

Über die weiter unten zu sehenden Buttons (Markierung 5) *Wartung* und *Sperren* sind weitere Bedienmöglichkeiten gegeben. Durch Betätigen des *Wartung*-Buttons wird ein Bit in der SPS gesetzt. Dieses setzt die Freigabepfung der Anlage zu Teilen aus und erlaubt dadurch Anlagenzustände, die im Normalbetrieb zur Abschaltung führen würden.

Der *Sperren*-Button bewirkt bei Betätigung das Ausschalten des FUs und lässt einen erneuten Anlauf nicht zu, solange die Sperrung aktiv ist. Diese kann durch erneutes Klicken aufgehoben werden. Sowohl der *Wartung*-, als auch der *Sperren*-Button werden bei Aktivierung grün hinterlegt.

Ein weiteres Bild ist das der Anlagenkennwerte (siehe Abbildung 32). Visualisiert sind die Höhenangaben und Abstände im Brunnen. Verbildlicht werden diese durch Pfeile. Alle relevanten Daten der baulichen Ausführung der Anlage müssen hier eingetragen werden. Andere werden berechnet und dargestellt. Die Daten dienen der SPS u.a. zur Berechnung von Abstich und Pegel.

Auf dem Bild 33 zu den Energie- und Leistungsdaten werden eben diese dargestellt. Dadurch lassen sich aktuelle Energieverbräuche der Pumpe ablesen, ebenso aber auch die Werte für Spannung und Strom.

Weitere Bilder sind zur Verwaltung von Meldungen und Benutzerdaten angelegt. Für diese werden von Siemens vorgefertigte Visualisierungselemente verwendet. Über die Meldungsverwaltung werden Alarmer und Warnungen, sowie sonstige Meldungen erfasst. Dafür werden im HMI Meldegruppen erstellt, in denen festgelegt wird, ob diese quittiert werden müssen. Der Status der Meldungen wird über Buchstaben abgekürzt:

- *K*: kommend
- *G*: gehend
- *Q*: quittiert

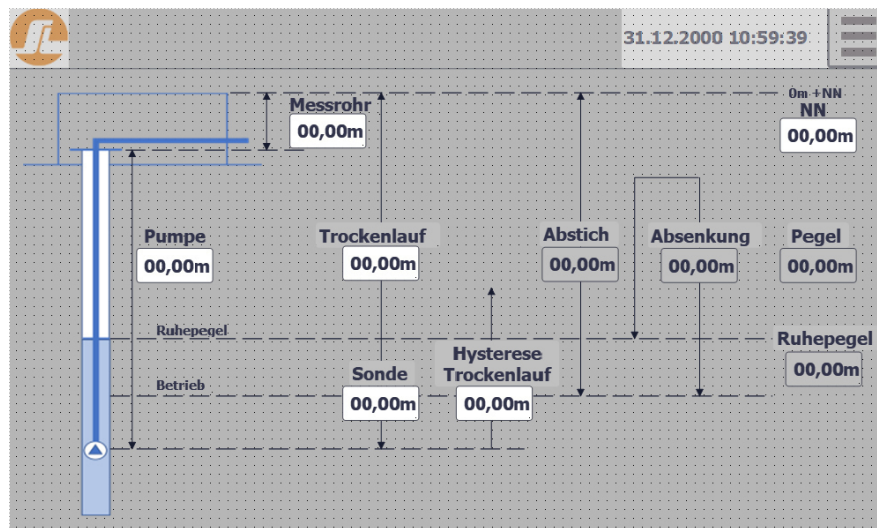


Abbildung 32: Kennwertbild in der Prozessvisualisierung

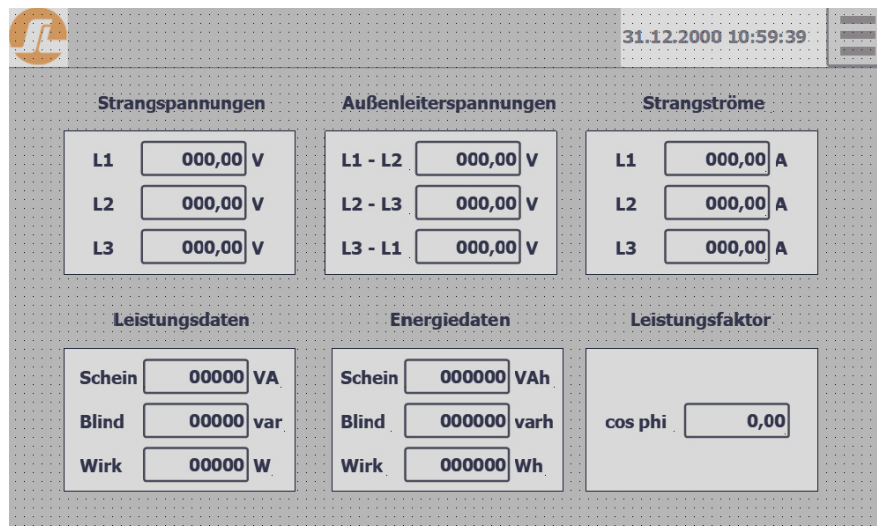


Abbildung 33: Energie- und Leistungsdaten in der Prozessvisualisierung

Im HMI erstellte quittierpflichtige Meldungen können über eine Schaltfläche quittiert werden. Ebenso kann über einen Button auf markierte Meldungen mit vorgefertigten Aktionen reagiert werden. Beispielsweise kann das Wechseln zu einem Bild das weitere Informationen zur Meldung gibt als eine solche Systemfunktion projektiert werden.[Sie19a] Ein weiterer Button zeigt Meldungsinformationen, sofern diese zuvor projektiert wurden.

Über die Benutzerverwaltung können neue Nutzer mit unterschiedlichen Zugriffsrechten angelegt werden. Alle Nutzer müssen einen Nutzernamen und ein Passwort festlegen. Die Standardnutzerklassen sind der Administrator und der Standardnutzer. Der Administrator darf auf die Benutzerverwaltung zugreifen, darf Überwachen, Bedienen und Parametrieren. Der Standardnutzer darf Überwachen und Bedienen. Benutzer, für die die Standardnutzer-Rechte nicht ausreichend sind, denen aber kein Administratorstatus gegeben werden soll, werden der Gruppe der Benutzer mit Parametrierrecht zugewiesen.

Sollte ein Benutzer auf für ihn gesperrte Optionen zugreifen wollen, wird ein Anmeldefenster geöffnet, dass das Einloggen eines Nutzers mit höherer Berechtigung ermöglicht. Durch Eingabe

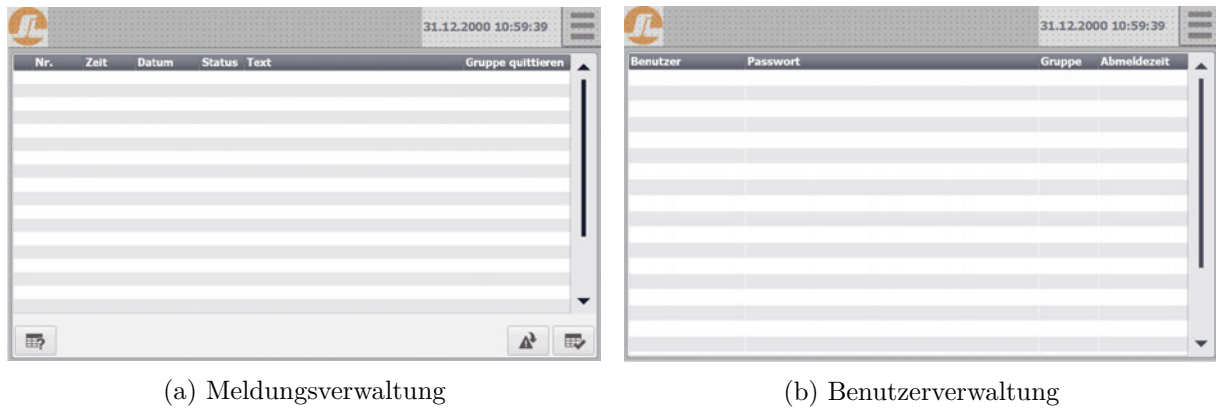


Abbildung 34: Bilder zur Datenverwaltung in der Prozessvisualisierung

der Nutzerdaten eines für die Aktion berechtigten Nutzers wird der Zugriff gewährt und der vorher eingeloggte Nutzer abgemeldet.

5.2.3 Objekt-Popups

Für eine hohe Übersichtlichkeit sieht das Visualisierungskonzept vor, alle objektspezifischen Daten und Informationen über Popup-Fenster zu verwalten. Diese sind mit einer einheitlichen Größe von 464x370 Pixel dargestellt.

Jedes Objekt benötigt ein Popup, sofern es zur Anzeige kommt. Bei Klicken auf einen in der Anlagenübersicht gezeigten Messwert wird ein Popup geöffnet, das den aktuellen Status, sowie den Messwert und den zeitlichen Verlauf der Messgröße zeigt. Zu sehen ist dieses in Abbildung 35a.

Das Popup 35b gibt eine Übersicht über selektierte und ausgelöste Alarmer und Warnungen. Die Kreise in der Spalte *Selektiert* zeigen ausgewählte Meldungen an. Die Anzeigen der Spalte *Ausgelöst* signalisieren ausgelöste Meldungen.

Im Popup der Abbildung 35c sind die Parameter der Messwertverarbeitung einstellbar. Über Schaltflächen werden Alarmer und Warnungen an- bzw. abgewählt. Die zugehörigen Grenzwerte und Hysteresen sind in E/A-Feldern zu hinterlegen. Einstellmöglichkeiten bzw. Parameter sind mit der entsprechenden Struktur verknüpft.

Das letzte Messwert-Popup ist in Abbildung 35d dargestellt. Dies dient der Einstellung von Softwareparametern. Hier werden die Skalierungsparameter eingegeben, die in der Analogwertverarbeitung als M_{min} und M_{max} verwendet werden. Über den Button *Simulation Ein/Aus* und das zugehörige E/A-Feld können Messwerte simuliert werden.

Die zweite große Gruppe an Popups dienen der Bedienung des FUs, der Regelung und der Pumpe. Die Abbildung 36a zeigt über die drei oberen Kreise den Status des FUs an.

Über die darunter liegenden Schaltflächen *Automatik* und *Hand* kann die Betriebsart der Durchflussregelung vorgegeben werden. Die Regelung des Durchflusses wird in Kapitel 5.3.5 erläutert. Für den Handbetrieb müssen weitere Parameter und Betriebsarten eingestellt werden. Die dafür benötigten Bedienelemente sind im Bereich *Handbetrieb* wiederzufinden. Die Veranschaulichung von Sollwert und Istwert des Volumenstroms wird von Balkendiagrammen übernommen. Die Verwendung von Balken soll hier einen schnellen Überblick geben, ob der Regler wie gewünscht arbeitet. Das darunterliegende E/A-Feld gibt die Regeldifferenz in % wider. Die Regelparameter,

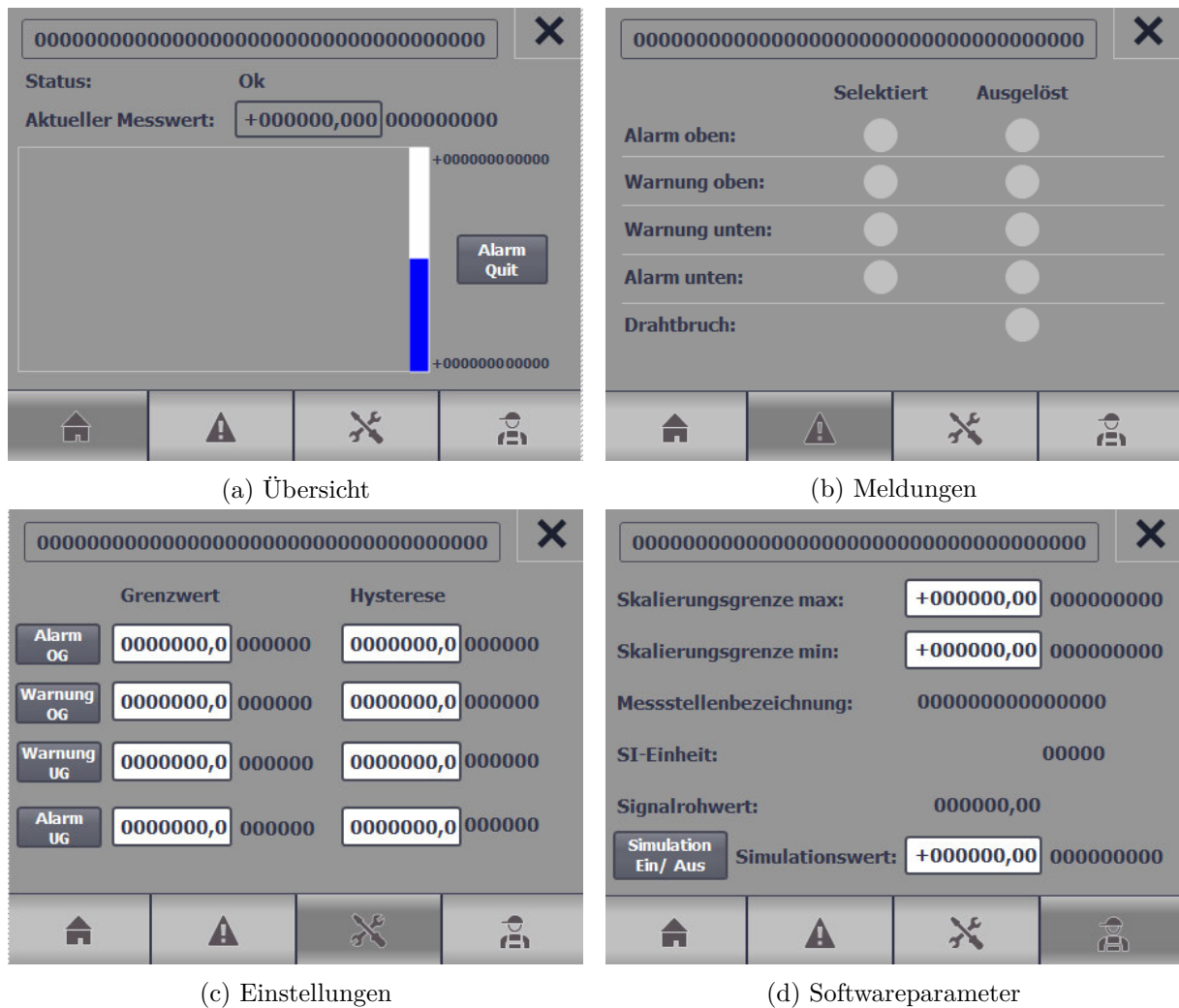


Abbildung 35: Messwert-Popups

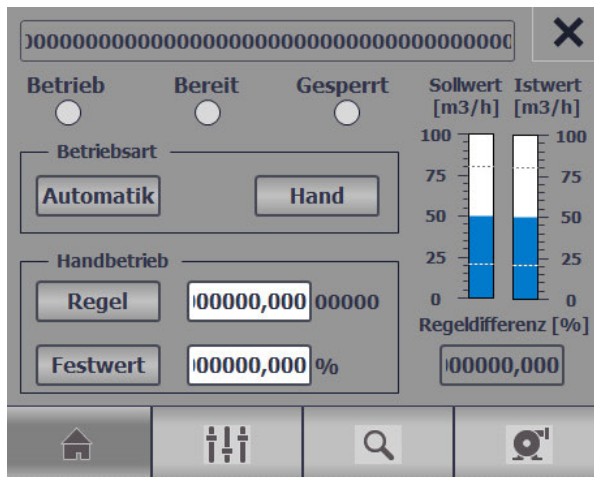
in Form von P-, I- und D-Anteil können im Popup der Abbildung 36b vorgegeben werden.

Das Popup der Abbildung 36c gibt Informationen zu den wirksamen Regelparametern an der Pumpe und bietet einen Gesamtüberblick über den Momentanzustand.

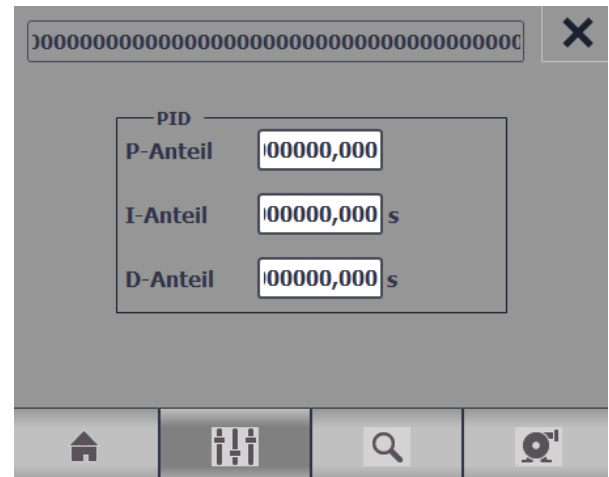
Über das vierte Popup (siehe Abbildung 36d) können die Motorkennndaten eingestellt werden. Weiterhin werden hier die Betriebsstunden und die erfassten Schaltspiele angegeben. Über die beiden *Reset*-Buttons können die Zählwerte zurückgesetzt werden. Um eine versehentliche Betätigung abzufangen werden Sicherheitsabfragen erzeugt. Diese sind immer nach dem Schema des in Abbildung 37 zu sehenden Popups aufgebaut.

Weiterhin wird die umgesetzte Arbeit in kWh und der aktuelle Strom durch die ASM dargestellt. Rechts daneben können die in Kapitel 5.1.4 beschriebenen Ein- und Ausschaltüberwachungen eingestellt werden. Die Kreise neben den dafür vorgesehenen E/A-Feldern leuchten rot, sollten Zeitüberschreitungen festgestellt werden.

Für alle gezeigten Popup-Bilder, sowohl für die der Messwerte, als auch für die zuletzt beschriebenen dienen die im unteren Bereich projizierten Schaltflächen zum Umschalten zwischen den Popups. Um zu veranschaulichen, welche Schaltfläche zu welchem Popup zugeordnet ist, werden die jeweiligen Buttons über die Gestaltung von den nebenstehenden abgehoben. Zu sehen ist



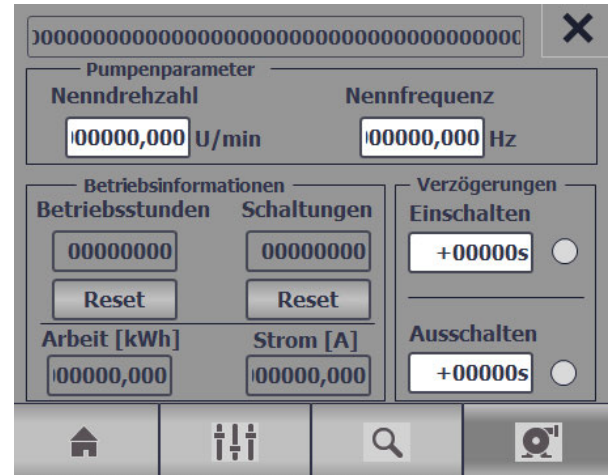
(a) Regler



(b) Regelparameter



(c) Übersicht Regelung



(d) Motorparameter

Abbildung 36: Popups zur Einflussnahme auf die Förderung

das sowohl in der Abbildung 35, als auch in der Abbildung 36.



Abbildung 37: Sicherheitsabfrage für das Resetten der Betriebsstunden

5.2.4 Adressmultiplexing

Die Popup-Bilder sind für die Bedienung, Parametrierung und für das Beobachten der einzelnen Objekte einer Klasse projiziert. Sie sind für die Anlagenfahrenden die Schnittstelle zu einem Objekt einer Klasse. Entsprechend müssen die Daten für jedes einzelne Objekt in den Popups dargestellt werden. Für jedes Objekt müsste somit eine Instanz der Popups erzeugt werden, was vor allem bei größeren Projekten einen erhöhten Projektierungsaufwand zur Folge hätten. Ebenso wäre die mit einem entsprechendem Speicherbedarf im Visualisierungsgerät verbunden. Um dies zu umgehen, werden die Daten der Objekte per Adressmultiplexing mit den Popup-Bildern verknüpft. Dies soll am Beispiel der Messwert-Popups erläutert werden.

Durch die in Kapitel 5.1.5 beschriebene Organisation der einzelnen Objekte in Arrays ist es möglich, die Daten eines Objekts durch die Adresse abzurufen. Dazu muss eine Multiplexvariable angelegt werden, die angibt, welches Objekt aus dem Array aufgerufen werden soll. Die im Popup hinterlegte Datenstruktur entspricht Variablen aus den definierten Strukturen, hier am Beispiel für die Messwerte, aus den Variablen der Datentypen *MessDatenAktual* und *MessDatenPara*. Bei Auswahl eines Objektes im HMI, wird eine ID im Ganzzahlformat zugewiesen, welche der Adresse der Messwertstruktur in dem entsprechenden Array entspricht. Über einen Pointer werden die Strukturen des hier beispielhaften Messwerts mit der Datenstruktur des Popups verknüpft. Bei Verwendung des Adressmultiplexings wird die Datenstruktur nur einmal angelegt, da der statische Zugriff auf einzelne Objekte nicht genutzt wird.

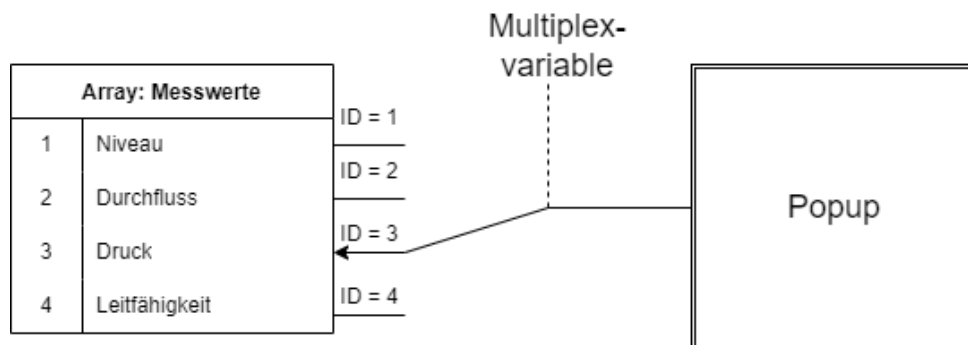


Abbildung 38: Adressmultiplexing

Die beschriebene Mehrfachnutzung von Popups durch virtuelle Kanäle reduziert den Bedarf an Speicher und den Projektierungsaufwand, da pro Klasse nur ein Popup erzeugt werden muss. Neben den Messwerten sind auch die anderen Objekte des Programms, wie z.B. die FU-Objekte, oder auch die Entitäten der Klasse der Digitalsignale in Arrays angeordnet. Somit ist das beschriebene Verfahren auf alle Informationen anwendbar. Ausgeschlossen davon sind jedoch spezifische Berechnungen, wie Abstich oder Pegel, da diese Informationen keiner Klasse angehören oder zugeordnet werden können.

5.3 Brunnenspezifische Funktionen

5.3.1 Freigaben und Grenzen

In Kapitel 5.1.4 ist beschrieben, dass für die Ansteuerung des FUs ein Freigabesignal vorliegen muss. Dieses wird von einem weiteren FC erzeugt, welcher prüft, ob alle Bedingungen für den Betrieb des FUs erfüllt sind. Dabei werden Meldungen des Brunnens, aber auch des zugehörigen Wasserwerks kontrolliert. Über die Bausteinschnittstelle ist es weiterhin möglich zusätzliche Bedingungen anzulegen. Dies geschieht über die Input-Variable *Stoerung*.

Input			
Name	Datentyp	Defaultwert	Kommentar
WW_Kom	Struct		Meldungen Wasserwerk
Pumpe_gesperrt	Bool		
Pumpe_SmlStr	Bool		Pumpe Sammelstörung
Wartung	Bool		Wartung aktiv
Freigb_ext	Bool		Externes Freigabesignal
Stoerung	Bool		etwaige Störung, die interne Freigabe verhindern
Trockenlauf	Bool		Trockenlauf derPumpe
Schachtflutung	Bool		Wasser im Brunnenschacht
Drehzahl	Bool		Unterschreitung der Minimaldrehzahl
Output			
Freigb	Bool		Freigabe
Temp			
Check	Int		Checksumme

Tabelle 9: Bausteinschnittstelle FC FU-Freigabe

Der FC wird in jedem Zyklus bearbeitet und erteilt oder entzieht die Freigabe. Dafür wird u.a. ein sog. Wegecheck durchgeführt. Dabei werden alle Einrichtungen, die das Wasser durchfließen muss, kontrolliert. Diese melden die Zustände der einzelnen Anlagenteile des Wasserwerks. Darunter zu verstehen sind die Wasserleitungen, die Filteranlagen und die Reinwasserbehälter, sowie der Pumpenraum. Für die Wasserleitungen des Rohwassers werden Meldungen übertragen, die anzeigen, ob ein Druckmaximum erreicht ist. Die Meldung der Filter gibt deren Betriebszustand wieder. Die Behälter werden auf Füllstand und Überlauf geprüft. Die Meldung des Pumpenraums gibt Informationen darüber, ob dieser geflutet ist, was z.B. auf einen Rohrbruch im Werk hindeuten könnte. Alle genannten Störungen werden als Bitmeldungen an den Brunnen übermittelt.

Die Anlagenteile des Wasserwerks werden einzeln geprüft. Bei bestandener Kontrolle wird die Zählvariable *Check* um Eins inkrementiert. Am Ende aller Wegechecks wird diese Variable mit der Anzahl aller Kontrollen verglichen. Bei nicht bestandenen Kontrollen sind Checksumme und Anzahl der Kontrollen verschieden. Die Freigabe wird in diesem Fall nicht erteilt bzw. entzogen. Sind Checksumme und Anzahl der Kontrollen gleich, so sind alle Teile des Wegechecks bestanden.

Die anschließende Kontrolle der Brunnenmeldungen betrifft die Pumpe und den Brunnenschacht. Die Pumpe wird auf anstehendes Sperrsignal, Trockenlauf und Minimaldrehzahl kontrolliert. Diese Überprüfung erfolgt nur, wenn die Pumpe eingeschaltet ist, da diese sonst niemals starten würde. Die Überprüfung des Sperrsignals dient der Sicherheit an der Anlage. Die Trockenlauf-

und die Drehzahlkontrolle werden vor dem Hintergrund durchgeführt, dass sowohl der Betrieb im Trockenlauf, als auch der Betrieb mit zu niedriger Drehzahl einen Schaden der Pumpe herbeiführen können.

Jede dieser Meldungen führt dazu, dass die nach dem Wegecheck intern gesetzte Freigabe zurückgesetzt wird. Allerdings geschieht dies nur, wenn keine Wartung aktiv ist. Bei gesetztem Wartungsbit führt nur der Trockenlauf zum Freigabeentzug.

```

1 REGION @Freigabeentzug
2   IF NOT Wartung THEN
3
4       IF Schachtflutung OR Drehzahl OR Stoerung THEN
5           Freigb := FALSE;
6       END_IF;
7
8   END_IF;
9
10  //Auch im Wartungsfall darf die Pumpe nicht trockenlaufen
11  IF Trockenlauf THEN
12      Freigb := FALSE;
13  END_IF;
14
15 END_REGION

```

Listing 11: FC FU-Freigabe: Freigabeentzug

5.3.2 Berechnung von Abstich und Pegel

Wichtige Betriebsdaten der Brunnenanlage sind Abstich und Pegel, die bereits in Kapitel 2.1.4 erklärt wurden. Diese sind entsprechend durch eine Funktion zu berechnen, um sie dem HMI zur Verfügung stellen zu können. Benötigte Werte sind den Eingängen der Bausteinschnittstelle (Tabelle 10) zu entnehmen.

Input			
Name	Datentyp	Defaultwert	Kommentar
Einhaengtiefe_Sonde	Real		in m
Einhaengtiefe_Pumpe	Real		in m
mWS	Real		von Sonde gemessener Druck in mWS
Abstand_Re_NN	Real		Abstand N.N. zu oberen Rohrstutzen
Trockenlauf_Grenze	Real		Bei Unterschreitung wird Trockenlaufbit gesetzt
Trockenlauf_Hysterese	Real		in m
Quit_Trockenlauf	Bool		
Output			
Abstich	Real		in m
Pegel	Real		in m
InOut			
Trockenlauf	Bool		
Temp			
Diff_Sonde_Pumpe	Real		Höhenunterschied zwischen Sonde und Pumpe

Tabelle 10: Bausteinschnittstelle FC Berechnung Abstich/Pegel

Wie in Abbildung 3b dargestellt, ist das eingemessene obere Ende des Rohrstutzens der Bezugspunkt der Berechnungen. Entsprechend wird der Abstich größer, je weniger Wasser im Brunnen steht. Dieser entspricht der Differenz zwischen der Einhängtiefe der Messsonde und der gemessenen Wassersäule. Die Berechnung des Abstichs kann der nachfolgenden Gleichung 14 entnommen werden.

$$\text{Abstich} = \text{Einhängtiefe Sonde} - \text{Wassersäule} \quad (14)$$

Aus Abbildung 3b folgt der Pegel als Höhendifferenz zwischen dem Abstand von Rohrende und N.N., sowie dem Abstich. Bezugspunkt des Pegels ist N.N. und ist entsprechend kleiner, wenn weniger Wasser im Brunnen steht.

$$\text{Pegel} = \text{Abstand Rohrende zu N.N.} - \text{Abstich} \quad (15)$$

Weiterhin übernimmt der FC das Setzen des Trockenlauf-Bits. Dies geschieht unter Verwendung des Abstichs. Ist der Abstich größer als der Wert der Trockenlaufgrenze, wird das Bit gesetzt. Dem nachfolgenden Listing 12 ist die Umsetzung der Trockenlaufüberwachung und der dazugehörigen Quittierung zu entnehmen.

```

1 REGION @Trockenlauf: Bit setzen
2   //Trockenlaufmeldung steht nicht an
3   IF NOT Trockenlauf THEN
4
5       //Trockenlaufschutz
6       IF Abstich >= Trockenlauf_Grenze THEN
7           Trockenlauf := TRUE;
8       ELSE
9           Trockenlauf := FALSE;
10      END_IF;
11
12  END_IF;
13 END_REGION
14
15 REGION @Trockenlauf: Quittieren
16   //Trockenlaufmeldung steht an
17   IF Trockenlauf THEN
18       //Quittierbar, wenn Abstich kleiner als Trockenlauf+Hysterese
19       IF Quit_Trockenlauf AND
20          (Abstich < (Trockenlauf_Grenze - Trockenlauf_Hysterese)) THEN
21           Trockenlauf := FALSE;
22
23       ELSE
24           Trockenlauf := TRUE;
25       END_IF;
26   END_IF;
27 END_REGION

```

Listing 12: FC Berechnung Abstich/Pegel: Trockenlauf

Die Quittierung der Trockenlaufmeldung geschieht, wie in Listing 12 gezeigt, nur dann, wenn der Abstich kleiner ist, als die Trockenlaufgrenze, abzgl. Hysterese.

Wie gezeigt, wird die Trockenlaufmeldung auf Grundlage des Abstichs gesetzt, bzw. zurückgesetzt. Der Abstich kann nur dann korrekt ermittelt werden, wenn über der Sonde eine Wassersäule steht. Daher wird empfohlen, die Sonde auf Höhe der Pumpe einzuhängen. Dadurch wird die korrekte Erfassung des Trockenlaufs sichergestellt.

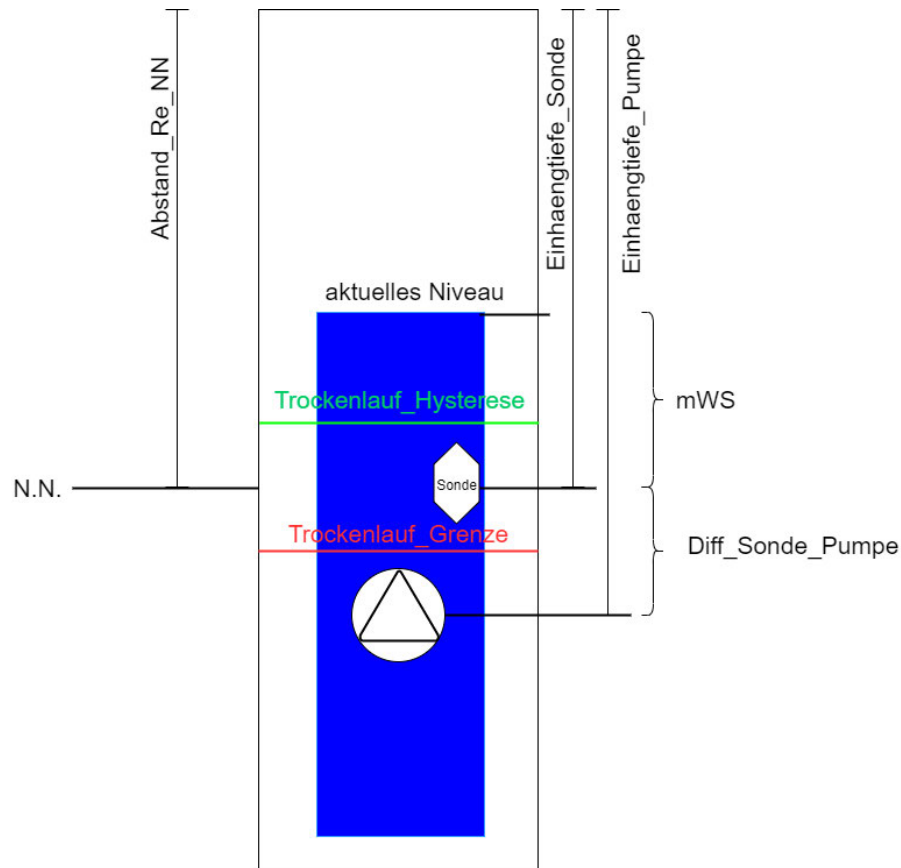


Abbildung 39: Veranschaulichung der relevanten Größen für den Trockenlauf

Das Einhängen der Sonde oberhalb der Pumpe, wie in Abbildung 39 dargestellt, hat den entscheidenden Nachteil, dass der Trockenlauf nicht erkannt wird, sollte die projektierte Grenze unterhalb der Sonde liegen. Um dies zu umgehen bestehen folgende Möglichkeiten:

1. Einhängen der Sonde auf Höhe der Pumpe
2. Trockenlauf-Grenze und Trockenlauf-Hysterese oberhalb der Sonde projektieren
3. Zusätzliche Abfrage zum Setzen des Trockenlaufbits, bei erfasster Wassersäule von 0 m

Das beste Ergebnis ist bei Umsetzung der erstgenannten Variante zu erwarten.

5.3.3 Berechnung von Ruhepegel und Absenkung

Der Betrieb der Pumpe bewirkt eine Absenkung des Ruhepegels. Sowohl die Absenkung, als auch der Ruhepegel sollen durch die Steuerung bestimmt werden. Dafür wird u.a. der Abstich benötigt.

Input			
Name	Datentyp	Defaultwert	Kommentar
Abstich	Real		in m
Betr	Bool		Pumpe in Betrieb
Clk_1Hz	Bool		1 Hz Taktmerker
Delay_Absenkung	Real		in Stunden
Delay_Ruhepegel	Real		in Stunden
Output			
Absenkung	Real		in m
Output			
Timer_Absenkung	Timer		
Timer_Ruhepegel	Timer		
Ruhepegel	Real		in m
Temp			
Sekunden_Absenkung	Real		
Sekunden_Ruhepegel	Real		

Tabelle 11: Bausteinschnittstelle FC Berechnung Ruhepegel/Absenkung

Der Ruhepegel wird eingelesen, wenn die Pumpe eine einstellbare Zeit nicht in Betrieb war. Diese Zeit kann über die Input-Schnittstelle *Delay_Ruhepegel* in Stunden angegeben werden. Die Erfassung des Ruhepegels ist dem nachfolgenden Listing 13 zu entnehmen.

```

1 REGION @Ruhepegel
2   //wird eingelesen , wenn Pumpe eine gewisse Zeit nicht in Betrieb war
3   Sekunden_Ruhepegel := Delay_Ruhepegel*3600;
4   IF NOT Betr THEN
5
6       //Starte Timer
7       IF (Clk_1Hz AND NOT Timer_Ruhepegel.Flnk) OR
8         (NOT Clk_1Hz AND Timer_Ruhepegel.Flnk) THEN
9           Timer_Ruhepegel.Passed += 1;
10          END_IF;
11
12      ELSE
13          //Timer zuruecksetzen
14          Timer_Ruhepegel.Passed := 0;
15          END_IF;
16
17      //Zeiten vergleichen
18      IF Timer_Ruhepegel.Passed = Sekunden_Ruhepegel THEN
19          Ruhepegel := Abstich;
20          Timer_Ruhepegel.Passed := 0;
21          END_IF;
22
23      //Ruecksichern der Flanke
24      Timer_Ruhepegel.Passed := Clk_1Hz;
25
26 END_REGION

```

Listing 13: FC Berechnung Ruhepegel/Absenkung: Ruhepegel

Die Berechnung der Absenkung wird durchgeführt, wenn die Pumpe für die parametrisierte Zeit durchgehend in Betrieb war. Sie berechnet sich aus der Differenz zwischen Ruhepegel und

momentanen Abstich. Sollte kein Ruhepegel vorliegen, beispielsweise, weil es der erste Start der SPS ist, wird die Berechnung der Absenkung vorerst ausgesetzt, um falsche Daten zu vermeiden.

5.3.4 Berechnung des Pumpenwirkungsgrads

Um den Wirkungsgrad der Pumpe feststellen zu können, wird neben der umgesetzten Wirkleistung der Pumpe, auch die Förderhöhe der Pumpe benötigt. Die Wirkleistung, sowie eine Vielzahl anderer Energie- und Leistungsdaten werden vom Energymeter¹⁶ zur Verfügung gestellt. Die Förderhöhe ist ein Begriff aus der Kreiselpumpentechnik und bezeichnet die auf das Fördermedium übertragene Förderleistung durch die Pumpe.[KSB]

Verbildlicht ist das Prinzip der Förderhöhe in Abbildung 40. Dabei kann der linke Behälter als Brunnen angenommen werden und der rechte als das Wasserwerk. Als Messpunkt am Brunnen dient das eingemessene obere Rohrende. Im Wasserwerk wird als Messpunkt der höchste Punkt der sog. Reinwasserschleife genutzt, da an diesem Punkt ein Druck von Null bar anliegt.

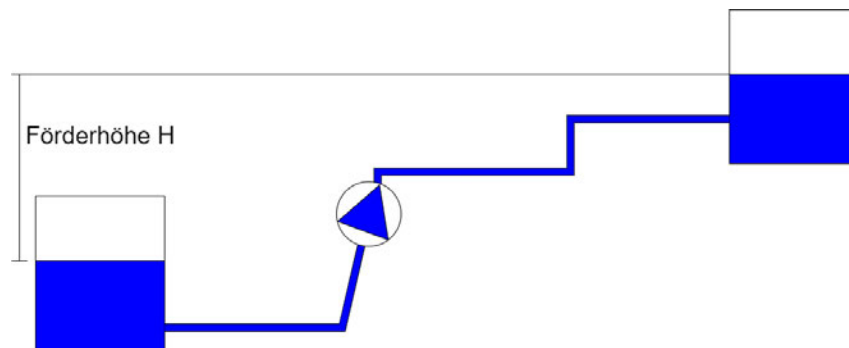


Abbildung 40: Förderhöhe

Die Messpunkte werden an der Bausteinschnittstelle (siehe Tabelle 12) angelegt.

Input			
Name	Datentyp	Defaultwert	Kommentar
Durchfluss	Real		
Dichte	Real		Dichte des Mediums in kg/m^3
P_Pumpe	Real		Gesamtwirkleistung der Pumpe in W
MessBr	Real		Einmesshöhe des Brunnen über N.N.
MesWW	Real		Einmesshöhe der Reinwasserschleife über N.N.
Output			
H	Real		Förderhöhe in m
n	Real		Wirkungsgrad
Const			
g	Real	9,81	Fallbeschleunigung in m/s^2

Tabelle 12: Bausteinschnittstelle FC Berechnung Pumpenwirkungsgrad

Die Förderhöhe bestimmt sich nach Abbildung 40 aus der Differenz der Einmesspunkte von Brunnen und Wasserwerk.

¹⁶AI Energy Meter 400VAC ST: AI-Baugruppe für den maschinennahen Einsatz zur Erfassung von über 200 Energie- und Messdaten

$$H = \text{Messpunkt Wasserwerk} - (\text{Messpunkt Brunnen} + \text{Abstich}) \quad (16)$$

Der Wirkungsgrad der Pumpe bestimmt sich anschließend durch Gleichung 17.

$$\eta = \frac{g \cdot \rho \cdot 3600 \cdot Q \cdot H}{P} \quad (17)$$

Die Formelzeichen stellen folgende Größen dar:

- η : Wirkungsgrad der Pumpe
- g : Fallbeschleunigung in m/s^2
- ρ : Dichte in kg/m^3
- Q : Durchfluss in m^3/h
- H : Förderhöhe in m
- P : Wirkleistung in W

Die Gleichung 17 vernachlässigt Rohrreibungsverluste. Anschließend an die Berechnung des Wirkungsgrads, wird eine Messwertkontrolle implementiert. Diese legt den Wirkungsgrad von 0% fest, sollte kein bzw. ein negativer Durchfluss erfasst werden.

5.3.5 Durchflussregelung

Die Regelung des Durchflusses ist eine der zentralen Aufgaben der Steuerung. Umgesetzt wird das durch den CONT_C-Baustein, der in Kapitel 2.5.2 erläutert ist. Die Regelung ist so aufgebaut, dass nach Vorgabe eines Durchflusses in m^3/h die Drehzahl der Pumpe so angepasst wird, dass sich dieser Sollwert einstellt. Erfasst wird der Durchfluss durch ein Messgerät, das nach dem, in Kapitel 2.3.1 beschriebenen Verfahren arbeitet. Dabei kann der Sollwert von allen drei Zugriffsebenen (siehe Kapitel 4.6) aus vorgegeben werden. Dabei wird zwischen drei Betriebsarten unterschieden. Wird der Brunnen von der NLS aus gesteuert, läuft die Regelung im Automatikbetrieb. Der aus der NLS kommende Sollwert wird damit von der Regelung angefahren. Die Steuerung aus dem Wasserwerk heraus erfolgt ebenfalls über den Automatikbetrieb. Vor Ort am Brunnen kann die Regelung im Handbetrieb verwendet werden. Dabei wird unterschieden zwischen Regel- und Festwertbetrieb. Im Regelbetrieb wird am HMI ein Durchflusssollwert eingestellt, der ausgeregelt wird. Im Festwertbetrieb wird eine feste Drehzahl der Pumpe vorgegeben, die daraufhin eingestellt wird.

Alle Informationen, die für den FC wichtig sind, werden über die Bausteinschnittstelle (siehe Tabelle 13) herangeführt. Der Reglerbaustein, inklusive des FCs, können für sämtliche kontinuierlichen Regelungen verwendet werden.

Die in der Variable *Vorg* vom PLC-Datentyp *Regelung* enthaltenen Daten werden zum Setzen der temporären Bits für das Anwählen der Regleranteile genutzt.

Input			
Name	Datentyp	Defaultwert	Kommentar
Messw	Real		Messwert
Reg_Low	Real		Reglergrenze unten
Reg_High	Real		Reglergrenze oben
Output			
Stellw	Real		Stellwert
InOut			
Vorg	Regelung		Enthält Befehle für die Bedienung der Regelung
Temp			
Stellgroesse_Soll	Real		Reglervorgabe abhängig von Ort/Fern
P_sel	Bool		P-Anteil angewählt
I_sel	Bool		I-Anteil angewählt
D_sel	Bool		D-Anteil angewählt

Tabelle 13: Bausteinschnittstelle FC Regler

```

1 REGION @Regleranteile
2   //P-Anteil (de-) aktivieren
3   IF Vorg.Regelanteile.P <> 0 THEN
4     P_sel := TRUE;
5   ELSE
6     P_sel := FALSE,
7   END_IF;
8
9   //I-Anteil (de-) aktivieren
10  IF Vorg.Regelanteile.I <> 0 THEN
11    I_sel := TRUE;
12  ELSE
13    I_sel := FALSE,
14  END_IF;
15
16  //D-Anteil (de-) aktivieren
17  IF Vorg.Regelanteile.D <> 0 THEN
18    D_sel := TRUE;
19  ELSE
20    D_sel := FALSE,
21  END_IF;
22 END_REGION

```

Listing 14: FC Regler: Regleranteile

Anschließend wird eine Instanz des CONT_C-Bausteins aufgerufen und parametrisiert. Dies geschieht wie in Kapitel 2.5.2 beschrieben.

```

1 "CONT_C_DB"(MAN_ON:=Vorg.Hand.Regel_Fest, //wird bei Fernzugriff zurueckgesetzt
2     PVPER_ON:=FALSE,
3     P_SEL:=P_sel,
4     I_SEL:=I_sel,
5     INT_HOLD:=FALSE,
6     I_ITL_ON:="FirstScan",
7     D_SEL:=D_sel,
8     CYCLE:=T#100ms,
9     SP_INT:=Vorg.Regelparameter.Stellgroesse_Soll,
10    PV_IN:=Messw,
11    MAN:=Vorg.Hand.Festwertbetr.Stellw_Aggregat,
12    GAIN:=Vorg.Regelparameter.P,
13    TI:=INT_TO_TIME(REAL_TO_INT(Vorg.Regelparameter.I*1000)), // in ms
14    TD:=INT_TO_TIME(REAL_TO_INT(Vorg.Regelparameter.D*1000)), // in ms
15    DEADB_W:=Vorg.Regelparameter.Totband,
16    LMN_HLM:=Reg_High,
17    LMN_LLM:=Reg_Low,
18    LMN=>Stellw,
19    LMN_P=>Vorg.Output_Regler.P_Anteil,
20    LMN_I=>Vorg.Output_Regler.I_Anteil,
21    LMN_D=>Vorg.Output_Regler.D_Anteil,
22    PV=>Vorg.Output_Regler.Istw,
23    ER=>Vorg.Output_Regler.Regeldiff);

```

Listing 15: FC Regler: CONT_C-Instanz

Im Automatikbetrieb wird der Handstellwert *Vorg.Hand.Festwertbetr.Stellw_Aggregat* nachgeführt. Der im Automatikbetrieb derzeitig berechnete Stellwert wird in diese Variable geschrieben, um bei der Umschaltung in den Festwertbetrieb sprunghafte Anstiege oder Abfälle zu vermeiden, da diese ungewollte Auswirkungen auf den Prozess haben können. Bei zu schnellem Anstieg kann es u.a. zu Druckschlägen kommen, die u.U. nachgelagerte Armaturen aufdrücken könnten.

Die Parameter, nach denen die Reglerbetriebsart eingestellt wird, werden durch andere FCs sortiert und gemanaged. Sie übernehmen die gesamte Verwaltung der Kommunikationsdaten zwischen den drei Zugriffsebenen.

Im Baustein, der den Regler-FC aufruft ist weiterhin so programmiert, dass nach der Umschaltung vom Hand- in den Automatikbetrieb, das Bit für den Hand-Regelbetrieb gesetzt wird. Die Umsetzung in AWL sieht wie folgt aus:

```

1 A  "DB_HMI_Panel_Bedienung".Ort_Fern
2 R  "DB_Regler".Regler_1.Hand.Regel_Fest

```

Listing 16: Aufruf FC Regler: Umschaltung Ort/Fern

6 Modultests

6.1 Erstellung von Testprotokollen

Die objektorientierte Software stellt Softwaremodule zur Verfügung, die als Klassen für die eingesetzten Feldgerätetypen verwendet werden. Die dafür entwickelten FCs müssen auf die Funktionalität getestet werden. In diesem Rahmen ist oftmals auch von White-Box-Tests die Rede, bei denen Testfälle mit Kenntnis der Funktionsweise erstellt werden und daher besonders geeignet für das Testen von Teilkomponenten ist. Es dient vor allem dazu, die innere Funktionsweise der FCs zu überprüfen.[Wik21]

Mittels dieser White-Box-Tests werden die entwickelten Standard-Bausteine, die die Methoden der Klassen darstellen, getestet. Für die Festlegung der Testfälle werden Modultest-Protokolle angefertigt, die sich am Programmcode orientieren. Neben Angaben zur Version des Softwaremoduls, werden hier Akzeptanzkriterien für die einzelnen Testfälle aufgestellt. Anhand dieser ist es nach Durchführung des White-Box-Tests möglich, die Funktionsweise auf Richtigkeit zu beurteilen.

Für das Aufstellen von Testfällen ist es notwendig, Akzeptanzkriterien zu schaffen, die eindeutig mit bestanden oder nicht bestanden beurteilt werden können. Dafür bieten sich insbesondere boolsche Testfälle an.

Die Testfälle werden in Diagrammen organisiert. Hier werden die Signalzustände für den jeweiligen Testvorgang festgelegt. Ein Beispiel einer solchen Tabelle ist in Abbildung 41 zu sehen.

Alarm unten Test-Nr	Testvorgang			Akzeptanzkriterium		Testergebnis	
	MessAktual.Aktual.Alarm.kDrahtbr	MessAktual.Aktual.Skal_Messw <= MessPara.Alarm.UG	MessPara.Alarme.Sel_UG	MessAktual.Alarm.UG	Mess.Aktual.Alarm.Sammel	MessAktual.Alarm.UG	Mess.Aktual.Alarm.Sammel
1	FALSE	FALSE	FALSE	FALSE	FALSE		
2	FALSE	FALSE	TRUE	FALSE	FALSE		
3	FALSE	TRUE	FALSE	FALSE	FALSE		
4	FALSE	TRUE	TRUE	FALSE	FALSE		
5	TRUE	FALSE	FALSE	FALSE	FALSE		
6	TRUE	FALSE	TRUE	FALSE	FALSE		
7	TRUE	TRUE	FALSE	FALSE	FALSE		
8	TRUE	TRUE	TRUE	TRUE	TRUE		

Abbildung 41: Beispiel: Testfälle und Akzeptanzkriterien

6.2 Vorgehensweise

Für das Testen der White-Box-Testfälle stellt Siemens ein Simulationsprogramm namens *PLC Sim* zur Verfügung. Dies ist ein Programm, das die Verhaltensweise einer Hardware-SPS simuliert. Über die Verbindung von TIA-Portal und PLC Sim, sowie der Verwendung von Force-Tabellen, in denen einzelne Variablen auf einen vorgegebenen Wert festgesetzt werden, ist es möglich, die Methoden einer Klasse losgelöst von deren Objekten zu testen. Die Verwendung von Whitebox-Tests lässt nur auf Fehler der einzelnen Funktionen schließen. Die korrekte Verschaltung zwischen FCs und Variablen kann nicht überprüft werden. Daher wird diese Testmethode erweitert. Dies wird im Folgenden erläutert.

Das Testen der Software geschieht über einen Testaufbau. Dieser besteht aus der CPU 1510-1 PN und dem Comfort Panel, sowie zugehöriger Spannungsversorgung und Sicherungen. Das Projekt und die Visualisierung werden auf SPS bzw. HMI-Panel geladen. Anschließend kann über

die Ethernet-Schnittstelle des PCs eine Online-Verbindung zur SPS aufgebaut werden. Über die Online-Verbindung besteht die Möglichkeit Daten zu „beobachten“. Die Beobachtungsfunktion zeigt die Aktualdaten ausgewählter DBs und FCs. Durch Bedienung des HMI-Panels ist die systematische Überprüfung der Variablenverknüpfungen möglich. Darunter fällt z.B. auch, dass die Daten der E/A-Felder korrekt an die Speicherbereiche der SPS angebunden sind. Diese Tests werden oftmals als Loop-Check bezeichnet und sind Teil der Funktionsprüfung.[Sei21, S. 235f.] Auf der Anlagenübersicht beginnend, wurde im ersten Schritt sichergestellt, dass die Messstellen richtig bezeichnet sind und die projektierten Buttons die verknüpften Bits setzen bzw. zurücksetzen. Ebenso sind die Bedien- und Beobachtungselemente der projektierten Popups und der anderen HMI-Bilder auf die korrekte Verschaltung mit den PLC-Variablen überprüft worden. Fehlerhafte Verknüpfungen waren in diesem Schritt korrigierbar.

Der zweite Schritt der Funktionsprüfung ist der Test der Zusatzlogik. Dieser umfasst die Kontrolle von Verriegelungen, Alarmen, aber auch von mathematischen Funktionen, wie z.B. der Skalierung und Normierung der Analogwerte. Diese Tests dienen als erweiterte Form der White-Box-Tests, da sie nicht nur die Funktionalität der einzelnen Softwaremodule, sondern die der Klasse als Ganzes bewerten. Hier finden die zuvor erwähnten Testprotokolle Anwendung. Die dort aufgeführten Testfälle werden durchgeführt.

Begonnen wurde mit der Funktionsprüfung der Messwertobjekte. Getestet wurde dabei am Objekt *Druck*. Durch die Allgemeingültigkeit der Messwert-Klasse, ist die Funktionsprüfung am Objekt *Druck* repräsentativ für diese Klasse. Die ersten Tests bezogen sich auf die Drahtbrucherkennung. Da am Testaufbau keine Feldgeräte an die AI-Karten angeschlossen waren, konnten hier auch keine Ströme gemessen werden. Die Drahtbruch-Meldung wurde wie erwartet gesetzt. Bei Einstellung einer Messwertsimulation löste die Drahtbruchmeldung nicht mehr aus. Somit erfüllt diese Teilfunktion die Anforderungskriterien.

Über Einstellung der Messwertsimulation und durch Parametrierung der Alarme, Warnungen und Hysteresen konnten diese im nächsten Schritt getestet werden. Besonderes wurde darauf geachtet, dass die Meldungen nur dann auslösen, wenn diese auch vom Nutzer selektiert sind.

Anschließend folgte die Überprüfung der Statuscode bei Auslösung einer Meldung.

Zuletzt wurde die Skalierung und Normierung der Analogwerte getestet. Die dafür erforderlichen Variablen wurden über das HMI parametrierung. Die im Prozess auftretenden Drücke befinden sich im Bereich von 0 bis 10 bar. Da die Skalierung aber auch für Zahlen mit Nachkommastelle getestet werden sollte, wurde die obere Skalierungsgrenze war auf 10,5 bar festgelegt, die untere auf 0 bar.

Dieser Test konnte nicht mit Hilfe der Messwertsimulation durchgeführt werden. Um die Signalwerte zu testen wurde auf die Verwendung der angesprochenen Forcetabellen zurückgegriffen. Diese setzte den Analogeingang *Druck* auf die im Testprotokoll vorgegebenen Signalwerte. Die Forcewerte müssen vom Typ *Word* sein. Die vom Testprotokoll vorgegebenen Signalwerte sind in der Tabelle 14 aufgeführt. Die angegebenen Akzeptanzkriterien sind abhängig von den eingestellten Skalierungsgrenzen.

Signalwert dezimal	Signalwert hexadezimal	Erwarteter skaliertes Messwert
0	16#0000	0 bar
13824	16#3600	5,25 bar
27648	16#6C00	10,5 bar
-4800	16#92C0	0 bar
32511	16#7EFF	10,5 bar

Tabelle 14: Testfälle Messwertskalierung

Für die genannten Skalierungsgrenzen wurden die aufgeführten Akzeptanzkriterien erfüllt.

Das Testen der Frequenzumrichter-Funktionen unterteilte sich in den Test der Freigabe, den Test der Direktansteuerung und den Test der Regelung. Für die Tests des Freigabe-FCs wurden die Wasserwerksmeldungen über den entsprechenden Kommunikations-DB geforced. Der Testaufbau ermöglichte die Simulation der brunnen-internen Meldungen. Auch die Freigabe bei gesetztem Wartungsbit wurde hier getestet. Die Tests zeigten, dass der FC die Freigabe verweigert, sollte das Trockenlaufbit gesetzt sein.

Für den Test der Regelung wurden über das HMI die Regelparameter eingestellt. Über die Parametrierung eines Durchfluss-Sollwerts konnte die Stellwertberechnung des Reglers geprüft werden. Da am Regler kein mathematisches Streckenmodell hinterlegt ist, wurde überprüft, ob sich der Stellwert bei Variation des Durchflusses ändert. Da in der Simulation des Durchflusses die Nachbildung des realen Prozesses nicht vollumfänglich möglich war, kann die korrekte Funktionsweise des Regler nicht gänzlich festgestellt werden. Die letzte Prüfung muss daher bei der Inbetriebnahme erfolgen.

Zuletzt folgten die Tests zur Direktansteuerung des FUs. Für die Kontrolle der Sollwertskalierung war ein Sollwert von 100 % vorgegeben. Diese musste auf einen Signalwert von 27648 skaliert werden. Für die Kontrolle der Istwert-Skalierung wurde der Analogeingang der Drehzahl auf die in Tabelle 15 geforced. Der FC erfüllte die dort angegebenen Akzeptanzkriterien.

Signalwert dezimal	Signalwert hexadezimal	Erwarteter skaliertes Messwert
0	16#0000	0 %
13824	16#3600	50 %
27648	16#6C00	100 %
-4800	16#92C0	0 %
32511	16#7EFF	100 %

Tabelle 15: Testfälle Istwertskalierung FU

Im Anschluss an die Kontrolle der Skalierungen wurde im nächsten Schritt das Setzen eines DQ-Startbefehls getestet. Bei Freigabe und einem eingestellten Sollwert größer Null wird das Startbit gesetzt.

Anschließend folgte der Test der Ein- und Ausschaltverzögerungen. Hier wurden im HMI Verzögerungszeiten eingegeben. Durch Forcen des Analogeingangs *PumpeDrehz_IW* konnte die Auslösung der Störmeldungen simuliert werden. Zuletzt wurden die verbleibenden Tests zu Kontrolle der Monitoringdaten und der Berechnung der Motorkeindaten durchgeführt.

Zuletzt wurden die Digitalsignale getestet. Dafür wurden die Forcetabellen genutzt. Über Parametrierung einer Signalinvertierung wurde das Bit gekippt. Die Einschaltverzögerung übernahm

nach Ablauf der Einschaltzeit den Signalrohwert in den *Status*-Wert, der im Programm weiterverwendet wird. Beim Test der Ausschaltverzögerung wurde nach Ablauf der Ausschaltzeit das *Status*-Bit zurückgesetzt.

7 Zusammenfassung

Im Rahmen dieser Arbeit wurde eine vollständig einsetzbare Software zur Steuerung von Grundwasserbrunnen entwickelt. Die dafür erarbeiteten Softwaremodule dienen der Umsetzung objektorientierter Klassen zur Verarbeitung analoger Messwerte, Digitalwerte und Frequenzumrichter. Die entwickelten Klassen ermöglichen aufgrund ihrer Bibliotheksfähigkeit den Einsatz in anderen Projekten und erfüllen somit eine grundlegende Anforderung, die zu Beginn des Projektes festgelegt wurde. Auf den Brunnen angepasste Funktionen, etwa zur Berechnung von Abstich, Pegel oder Trockenlauf, sowie zur Ermittlung von Pumpenwirkungsgrad und Förderhöhe konnten entwickelt werden. Die Berechnungen dieser Programmteile müssen im Feld getestet werden. Die fehlerfreie Funktion der beschriebenen Klassen konnte mittels der in Kapitel 6 beschriebenen Funktionstests weitestgehend festgestellt werden. Über die dort angebrachten Tests wurden Fehler analysiert und behoben. Dies betrifft sowohl den Programmcode, als auch die Verschaltung und Verknüpfung der einzelnen Objekte, Funktionen und Daten untereinander.

Die entwickelte Steuerung ist auf die Integrierbarkeit in die bestehenden Anlagen optimiert. Die in Kapitel 4 erläuterte Bestandsaufnahme bildete die Grundlage für die Erstellung von Datenbausteinen und Funktionen zur Kommunikation mit der SPS des Wasserwerks. Aufgrund der ausführlichen Bestandsdokumentation und den Einblick in die Programme und die Visualisierung des Wasserwerks, war es möglich die bestehende Zugriffsstruktur beizubehalten. Sie ermöglicht die Steuerung des Brunnen aus der Netzleitstelle, dem Wasserwerk und direkt am Brunnen. Dabei konnte die Zugriffsebene *Ort* um Funktionen erweitert werden, die über das Ein- und Ausschalten der Pumpe hinausgehen. Hervorzuheben ist hier vor allem die Vielzahl an Möglichkeiten zur Parametrierung der Objekte am projektierten HMI.

Das auf Basis von Objektpopups aufgebaute HMI, das als Schnittstelle zwischen Bediener und Prozess dient, ist intuitiv bedienbar. Die einheitliche Farbgebung der einzelnen Visualisierungselemente hebt die relevanten Daten hervor und ermöglicht eine gute Übersichtlichkeit.

Durch die ausführlichen Software-Tests konnte die korrekte Funktionalität der entwickelten Klassen festgestellt werden.

Vorstellbare Erweiterungen am Brunnen selbst sind u.a. die Verwendung von mobilen HMI-Panels. Der Zugriff auf die Steuerung über Bluetooth oder WLAN könnte eine komfortablere Bedienung und Beobachtung des Förderprozesses ermöglichen. Auch bietet die Verwendung von OPC UA Potenzial für einen leistungsfähigeren Datenaustausch zu externer Software, die u.a. die Betriebsdaten in SQL-Datenbanken verwaltet. Ebenso wäre über OPC UA eine vertikale Integration der Schichten der Autoamtsierungs-pyramide denkbar. In wie weit diese Integration in den nächsten Jahren im Bereich der Grundwasserversorgung notwendig sein wird, ist schwer abschätzbar. Jedoch ist in der Automatisierungstechnik der Trend zur vertikalen Vernetzung weit verbreitet.

Der zu Beginn des Projekts erstellte Zeitplan konnte eingehalten werden. Die gesetzten Meilensteine wurden indes nicht immer zum vorgesehenen Zeitpunkt abgeschlossen. Hier zeigte sich die fehlende Erfahrung bei der Erstellung und Einschätzung der Projektabschnitte, die allerdings durch eingeplante Zeitreserve vollständig ausgeglichen werden konnten.

8 Ausblick

Die Inbetriebnahme der Software bildet den nächsten Meilenstein der Modernisierungsmaßnahme. Diese konnte aufgrund interner Betriebsabläufe und der dadurch aufgetretenen zeitlichen und personellen Engpässe nicht im Rahmen der Bachelorarbeit durchgeführt werden. Neben der Parametrierung der einzelnen Objekte, wird hier die Funktion der Software endgültig getestet. Spätere Automatisierungsprojekte bieten weiterhin die Möglichkeit, die Bibliothek objektorientierter Funktionen zu erweitern, deren Grundstein mit dieser Arbeit geschaffen wurde.

Literaturverzeichnis

- [ABB11] ABB AUTOMATION PRODUCTS GMBH MOTORS & DRIVES (Hrsg.): Benutzerhandbuch ACS310 Frequenzumrichter. Wallstadter Straße 59, D-68526 Lauenburg, Deutschland: ABB Automation Products GmbH Motors & Drives, 12 2011
- [Aut21] AUTOMATION24: SIMATIC Comfort Panel Siemens TP700 Comfort - 6AV2124-0GC01-0AX0. <https://www.automation24.de/simatic-comfort-panel-siemens-tp700-comfort-6av2124-0gc01-0ax0>. Version: 2021. – [Online; Stand 10. Januar 2022]
- [End17] ENDRESS+HAUSER (DEUTSCHLAND) GMBH+CO. KG (Hrsg.): Technische Informationen Proline Promag 50W, 53W Magnetisch-induktives Durchflussmesssystem. Colmarer Straße 6, 79576 Weil am Rhein, Deutschland: Endress+Hauser (Deutschland) GmbH+Co. KG, 10 2017
- [End18] ENDRESS+HAUSER (DEUTSCHLAND) GMBH+CO. KG (Hrsg.): Technische Information Indumax CLS50D/CLS50 Induktiver Leitfähigkeitssensor für Standard-, Ex- und Hochtemperaturanwendungen. Colmarer Straße 6, 79576 Weil am Rhein, Deutschland: Endress+Hauser (Deutschland) GmbH+Co. KG, 10 2018
- [End21] ENDRESS+HAUSER (DEUTSCHLAND) GMBH+CO. KG (Hrsg.): Technische Informationen Cerabar PMP71B Prozessdruck- und Füllstandsmessung in Flüssigkeiten und Gasen. Colmarer Straße 6, 79576 Weil am Rhein, Deutschland: Endress+Hauser (Deutschland) GmbH+Co. KG, 03 2021
- [Kar13] KARGER, Rosemarie ; HOFFMANN, Frank (Hrsg.): Wasserversorgung Gewinnung - Aufbereitung - Speicherung - Verteilung. 14., vollst. akt. Aufl. 2013. Springer, 2013
- [KSB] KSB SE & CO.KGAA: Förderhöhe. <https://www.ksb.com/kreiselpumpenlexikon/foerderhoehe/186016>. – [Online; Stand 24. Januar 2022]
- [KSB21] KSB SE & CO.KGAA: Geodätische Höhe. <https://www.ksb.com/kreiselpumpenlexikon/geodaetische-hoehe/186912>. Version: 01 2021, Abruf: 15.10.2021
- [Lan21] LAND SCHLESWIG-HOLSTEIN: Trinkwasser und Coronavirus SARS-CoV-2. <https://www.schleswig-holstein.de/DE/Landesregierung/Themen/GesundheitVerbraucherschutz/Trinkwasser/trinkwasser.html>. Version: 01 2021, Abruf: 16.09.2021
- [Lun13] LUNZE, Jan 1.: Regelungstechnik 1 Systemtheoretische Grundlagen, Analyse und Entwurf einschleifiger Regelungen. 9., überarb. Aufl. Springer, 2013 (Springer-Lehrbuch)
- [PHO21] PHOENIX CONTACT GMBH & Co.KG (Hrsg.): Ethernet-Extender Anwenderhandbuch UM DE TC Extender... Flachsmarktstraße 8, 32825 Blomberg, Deutschland: PHOENIX CONTACT GmbH & Co.KG, 2021
- [Pro20] PROBST, Uwe ; VERLAG, Carl H. (Hrsg.): Leistungselektronik für Bachelors Grundlagen und praktische Anwendungen. 4., aktualisierte und erweiterte Auflage. Hanser, 2020
- [Rec] RECKNAGEL ONLINE: Schleimmenge. <https://www.recknagel-online.de/mehr/lexikon/schleimmenge.html>. – [Online; Stand 27. Januar 2022]

- [Sei15] SEITZ, Matthias 1.: Speicherprogrammierbare Steuerungen für die Fabrik- und Prozessautomation Strukturierte und objektorientierte SPS-Programmierung, Motion Control, Sicherheit, vertikale Integration. 4., überarbeitete und erweiterte Auflage. Fachbuchverl. Leipzig im Carl-Hanser-Verl., 2015 (Hanser eLibrary). <https://external.dandelon.com/download/attachments/dandelon/ids/DE0024D8251B10420E877C1257F530032BFD4.pdf>
- [Sei21] SEITZ, Matthias 1.: Speicherprogrammierbare Steuerungen in der Industrie 4.0 objektorientierter System- und Programmentwurf, Motion Control, Sicherheit, Industrial IoT. 5., aktualisierte und erweiterte Auflage. Hanser, 2021 (Hanser eLibrary)
- [Sie14] SIEMENS AG DIGITAL INDUSTRIES (Hrsg.): SIMATIC S7-1500, ET200MP, ET 200SP, ET 200AL, ET 200pro, ET 200eco PN Analogwertverarbeitung. Postfach 48 48, 90026 Nürnberg, Deutschland: Siemens AG Digital Industries, 06 2014
- [Sie19a] SIEMENS AG (Hrsg.): TIA Informationssystem: EventsReferenceWCCPdeDE. Postfach 48 48, 90026 Nürnberg, Deutschland: Siemens AG, 2019
- [Sie19b] SIEMENS AG DIVISION DIGITAL FACTORY (Hrsg.): SIMATIC S7-1200, S7-1500 PID-Regelung. Postfach 48 48, 90026 Nürnberg, Deutschland: Siemens AG Division Digital Factory, 11 2019. https://cache.industry.siemens.com/d1/d1-media/036/108210036/att_967326/v7/130065949707/de-DE/index.html#home. – [Online; Stand 11. Januar 2022]
- [Sie20] SIEMENS AG (Hrsg.): S7-Kommunikation zwischen SIMATIC S7-1200 und SIMATIC S7-1500 mit PUT/GET. Postfach 48 48, 90026 Nürnberg, Deutschland: Siemens AG, 11 2020
- [Sie21a] Siemens AG Digital Industries: SIMATIC S7-1500. <https://new.siemens.com/de/de/produkte/automatisierung/systeme/industrie/sps/simatic-s7-1500.html>. Version: 2021. – [Online; Stand 07. Januar 2022]
- [Sie21b] Siemens AG Digital Industries: Welche Regeln müssen bei der Namenszuweisung von SINAMICS Geräten mit PROFINET-Anschluss eingehalten werden, wenn diese mit S7 Technology oder SIMOTION SCOUT parametrisiert werden? <https://support.industry.siemens.com/cs/document/36062700/>. Version: 2021. – [Online; Stand 13. Januar 2022]
- [Sie21c] SIEMENS AG DIGITAL INDUSTRIES (Hrsg.): Gerätehandbuch SIMATIC ET 200SP CPU 1510-SP-1 PN (6ES7510-1DJ01-0AB0). Postfach 48 48, 90026 Nürnberg, Deutschland: Siemens AG Digital Industries, 05 2021
- [Sie21d] SIEMENS AG DIVISION DIGITAL INDUSTRIES (Hrsg.): SIMATIC HMI Bediengeräte Comfort Panels. Postfach 48 48, 90026 Nürnberg, Deutschland: Siemens AG Division Digital Industries, 05 2021
- [Tra21a] TRAVENETZ: Unser Netzgebiet. <https://www.travenetz.de/unternehmen/netzgebiet/>. Version: 01 2021, Abruf: 01.09.2021
- [Tra21b] TRAVENETZ: Unsere Gesellschafterstruktur und Beteiligungen im Überblick. <https://www.travenetz.de/unternehmen/gesellschafterstruktur-und-beteiligungen/>. Version: 05 2021, Abruf: 01.09.2021
- [TU 21] TU BERGAKADEMIE FREIBERG INSTITUT FÜR AUTOMATISIERUNGSTECHNIK: Versuch: SPS Speicherprogrammierbare Steuerungen. <https://tu-freiberg>.

de/sites/default/files/media/institut-fuer-automatisierungstechnik-6735/download/praktikum_sps.pdf. Version: 10 2021, Abruf: 22.10.2021

- [Umw21] UMWELTBUNDESAMT: Öffentliche Wasserversorgung. <https://www.umweltbundesamt.de/daten/wasser/wasserwirtschaft/oeffentliche-wasserversorgung#grundwasser-ist-wichtigste-trinkwasserressource>. Version: 08 2021, Abruf: 02.09.2021
- [VDI14] VDI/VDE-GESELLSCHAFT MESS- UND AUTOMATISIERUNGSTECHNIK: VDI/VDE 3699 Blatt 2: Prozessführung mit Bildschirmen - Grundlagen / Verein Deutscher Ingenieure. 2014 (3). – Forschungsbericht
- [Wik21] WIKIPEDIA: White-Box-Test — Wikipedia, die freie Enzyklopädie. <https://de.wikipedia.org/w/index.php?title=White-Box-Test&oldid=207317286>. Version: 2021. – [Online; Stand 27. Januar 2022]
- [Wik22] WIKIPEDIA: Profinet — Wikipedia, die freie Enzyklopädie. <https://de.wikipedia.org/w/index.php?title=Profinet&oldid=218925031>. Version: 2022. – [Online; Stand 10. Januar 2022]

Anhang

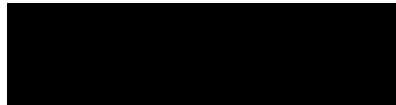
Der Anhang zur Arbeit befindet sich auf CD und kann beim Erstgutachter eingesehen werden.

Eidesstattliche Erklärung

Versicherung über die Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §16(5) APSO-TI-BM ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen habe ich unter Angabe der Quellen kenntlich gemacht.

Unterschrift :



Ort, Datum :