
1. EINLEITUNG	5
2. MOTIVATION.....	7
3. PROFIBUS.....	9
3.1. Protokollarchitektur und -Varianten	9
3.2. Übertragungstechnik DP/FMS	11
3.3. Datenübertragung	15
3.4. Anwenderschicht	19
4. DRIVECOM PROFIL 20.....	22
4.1. Datenstruktur	22
4.2. Geräteverhalten.....	23
4.3. Steuerwort	27
4.4. Statuswort.....	27
4.5. Parameterkommunikation	28
5. SERVOANTRIEB.....	30
5.1. Servomotor.....	31
5.2. Motorgeber	32
5.3. Servoregler.....	33
5.4. Positionieren	40
5.5. Sicherheitseinrichtungen.....	44
5.6. Auslegung	45
6. SOFTWARE.....	50
6.1. SIMATIC Step7	50
6.2. Lenze Global Drive Control	51
7. HARDWARE.....	52
7.1. CPU 314C-2DP	52
7.2. MDSKRS036-13.....	53
7.3. EVS9322-EP.....	54
7.4. EMF2133IB	55

8. UMSETZUNG	57
8.1. Folienvorzug	57
8.2. Profibus	59
8.3. Servoantrieb	66
8.4. SPS-Funktionsbaustein	81
9. TEST FOLIENVORZUG.....	84
10. ZUSAMMENFASSUNG	88
11. ABBILDUNGSVERZEICHNIS	89
12. FORMELZEICHENERKLÄRUNG	91
13. QUELLENANGABEN	92
14. SELBSTSTÄNDIGKEITSERKLÄRUNG.....	94
15. DANKSAGUNG	95
A. ANHANG	96
A.1. Inhalt CD-ROM.....	96
A.2. Dokumentation FB Folienvorzug	97
A.3. Petrinetz Parameterkommunikation	110

1. Einleitung

In heutigen Industrieanlagen werden zum Positionieren von Produkten und Werkzeugen vermehrt Servoantriebe eingesetzt. Der Vorteil von Servoantrieben gegenüber herkömmlichen Antrieben liegt in ihrer hohen Positioniergenauigkeit und ihrer hohen Dynamik, die einen schnellen Prozessablauf erlaubt.

Hinzu kommt, dass die automatisierten Anlagen heute häufig über keinen zentralen Motor mehr als Antriebseinheit verfügen. Häufig dienen mehrere Motoren als Antrieb. Diese verteilen sich dabei über die gesamte Anlage.

Um die Antriebe mit der zentralen Steuerung überwachen und steuern zu können, müssen diese miteinander verbunden werden. Der Informationsaustausch nimmt durch die ständige Weiterentwicklung jedoch stetig zu. Die Übertragung dieser steigenden Datenmenge wird heute meist mit Bus-Systemen ermöglicht. Mit ihnen können alle Elemente einer Anlage miteinander verbunden werden. Eines dieser Bus-Systeme ist der Profibus. Hierbei handelt es sich um ein System, welches in der Automatisierungstechnik weit verbreitet ist und von vielen Herstellern unterstützt wird.

Das Ziel dieser Diplomarbeit ist die Entwicklung einer Positioniersteuerung mit einem über den Profibus vernetzten Servoantrieb. Als zentrale Steuerung dient eine Speicherprogrammierbare Steuerung.

Nach kurzer Darstellung der Motivation dieser Arbeit am Beispiel eines derzeitig verwendeten Positionierantriebes und dessen Automatisierungsaufwand werden zunächst die Grundlagen der erforderlichen Elemente für eine Weiterentwicklung beschrieben. Hierbei handelt es sich um eine Profibus-DP-Kommunikation, das für Antriebssteuerungen normierte DRIVECOM-Profil sowie Grundlagen über Servoantriebe und Positionierprogramme. Danach folgt, als Beispiel für den Einsatz an Verpackungsmaschinen, ein Folienvorzug, der zum Verschließen von Schalen benötigt wird.

Unterstützt wird diese Arbeit von der Firma Ribbeck GmbH aus Bad Bramstedt. Die Firma Ribbeck GmbH ist tätig im Bereich der Verpackungs- und Lebensmitteltechnologie und vertreibt Verschluss- bzw. Versiegelungsmaschinen sowie Befüllungs- und Dosiereinrichtungen.

2. Motivation

Motivation zu dieser Arbeit ist der hohe Verdrahtungsaufwand in derzeitigen Verpackungsmaschinen und das Problem der ungenauen Positionierung. Für die Kommunikation wird neben einem bereits vorhandenen Profibus eine Vielzahl an Steuer- und Überwachungsleitungen eingesetzt.

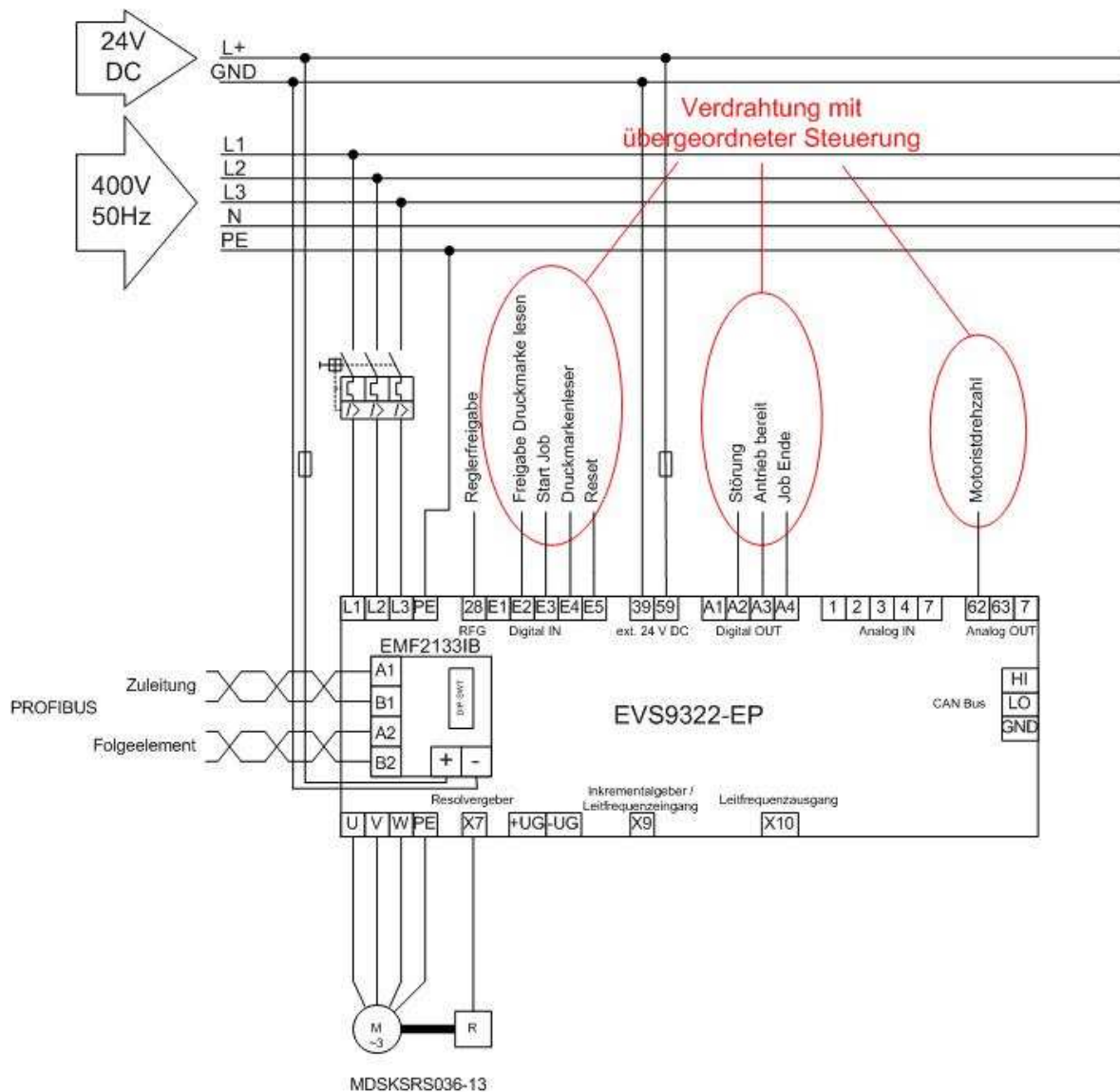


Abbildung 1 - Installation derzeitiger Positionierantriebe

Die Darstellung der Installation von derzeitigen Positionierantrieben zeigt, dass neben einigen digitalen Signalen ein analoges Signal mit einer übergeordneten Steuerung zu verbinden ist.

Die Positionierung kann nur abgeschätzt werden, da ein genaues Anpassen des Antriebes an die Maschine bisher fehlt.

Für dieses ungenaue Positionieren und die Geschwindigkeit erfolgt deren Vorgabe bereits über einen Profibus.

Mit dieser Diplomarbeit soll die Positionierung mit einem Servoantrieb auf eine Genauigkeit von $\pm 0,1$ Millimeter erfolgen und auf die zusätzlichen Leitungen

verzichtet werden, indem die Steuerung und Überwachung des Antriebes über den Profibus ermöglicht wird.

Als Beispiel dient der Einsatz einer Positionierung für einen Folienvorzug an Verpackungsmaschinen.

Bei der Umsetzung wurde dabei als Grundlage für die Kommunikation und das Positionierprogramm auf die derzeitige Installation verzichtet.

3. Profibus

Profibus steht für Prozess Field Bussystem. Er dient als Kommunikationssystem in der Feldebene von Automatisierungssystemen. Der Profibus ermöglicht den seriellen Datenaustausch zwischen den am Bus angeschlossenen Feldgeräten. Seine Vorteile liegen in einer möglichen Kostenersparnis. Diese entsteht durch einen reduzierten Verkabelungsaufwand bei dezentralen Peripheriestrukturen im Vergleich zu konventionellen Verbindungen. Ein weiterer Vorteil liegt in dem Konzept des offenen, firmenneutralen Feldbusstandards und der großen Anzahl verfügbarer Feldgeräte. Der Feldstandard Profibus entstand aus einem Verbundprojekt einiger Hersteller ab 1987 in Deutschland unter Führung des BMBF. Er ist zunächst in der DIN 19245 Teil 1 und Teil 2 normiert worden. Diese DIN wurde 1996 in die EN50170 überführt, aus der dann 1999 die IEC 61158/IEC 61784 entstand. /4/

In diesem Kapitel soll nun die Funktionsweise des Profibusses näher erläutert werden. Überwiegend sind /1/ und /2/ als Quellen verwendet.

3.1. Protokollarchitektur und –Varianten

Die Architektur des Profibus-Protokolls orientiert sich an den Standards der bereits bestehenden Kommunikationssysteme. So ist die Struktur des OSI-Referenzmodells zugrunde gelegt, welches aktuell der Norm ISO/IEC 7498-1:1994 entspricht.

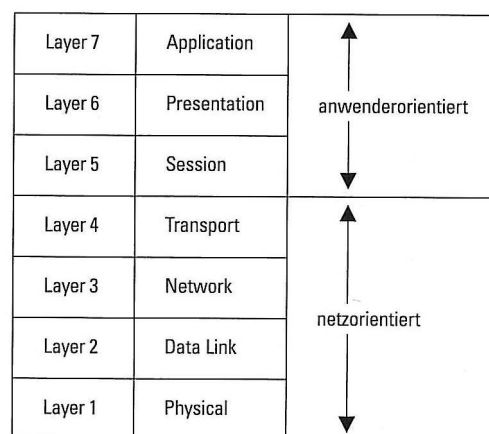


Abbildung 2 - Das ISO/OSI Modell für Kommunikationsstandards /1/

Die in Abbildung 2 dargestellten 7 Layer (Schichten) bilden zusammen das ISO/OSI-Referenzmodell. Es lässt sich in zwei Klassen aufteilen. Die 1. bis 4. Schicht bilden den netzorientierten und die 5. bis 7. Schicht den anwenderorientierten Teil. Hierbei regelt der netzorientierte Teil den Transport

der Informationen von einem Ort zum anderen. Der anwenderorientierte Teil stellt dem Anwender den Zugriff auf den Bus zur Verfügung.

Aufgrund der Anforderungen im Einsatzgebiet des Profibusses wird nur ein Teil der Schichten des Kommunikationsmodells berücksichtigt. So werden beim Profibus lediglich die Schichten 1, 2 und zum Teil auch 7 verwendet. Dies erhöht die Effizienz. Teile der fehlenden Schichten werden dann im Application Layer implementiert. Aufgrund der unterschiedlichen Ausprägung der Layer verfügt der Profibus über drei miteinander kompatible Varianten. Für die Kommunikation in der Zellebene dient der Profibus-FMS. Zum Nutzdatenaustausch in der Feldebene wird oftmals der Profibus-DP eingesetzt. Für Anwendungen in Ex-geschützten Bereichen gibt es die Variante Profibus-PA.

Profibus-FMS

Wie bereits erwähnt, dient der Profibus-FMS zur Kommunikation in der Zellebene. Ein Beispiel hierfür ist der Datenaustausch zwischen SPS und PC. Die im Application Layer stehenden FMS (Fieldbus Message Specification) stellen dem Anwender umfangreiche Kommunikationsdienste zur Verfügung. Das LLI (Lower Layer Interface) bildet hierbei für den FMS eine geräteunabhängige Verknüpfung zum Data Link Layer.

Profibus-DP

DP steht für dezentrale Peripherie. Der Profibus-DP Standard verwendet nur die ersten beiden Schichten des ISO/OSI-Referenzmodells. Durch diese schlanke Struktur ist ein schnellerer Datenaustausch möglich. Der Anwender bekommt hierbei durch das User-Interface einen direkten Zugang zum Data Link Layer. Dadurch werden die Schichten 7 bis 3 übergangen und sind nicht weiter ausgeprägt.

Da der Profibus-DP und -FMS mit derselben Datensicherung, Buszugriffsprotokoll und Übertragungstechnik arbeitet, können diese auch parallel auf einer Leitung angewendet werden.

Profibus-PA

Der Profibus-PA (Prozess Automation) ist von der Struktur identisch mit dem Profibus-DP. Sie unterscheiden sich jedoch in dem erweiterten PA-Protokoll für

die Datenübertragungstechnik. Dadurch wird ihr Einsatz in Ex-geschützten Bereichen ermöglicht.

PA-Geräte können mit dem Einsatz von sog. Segmentkopplern oder Profibus-PA-Links an DP-Netze angeschlossen werden.

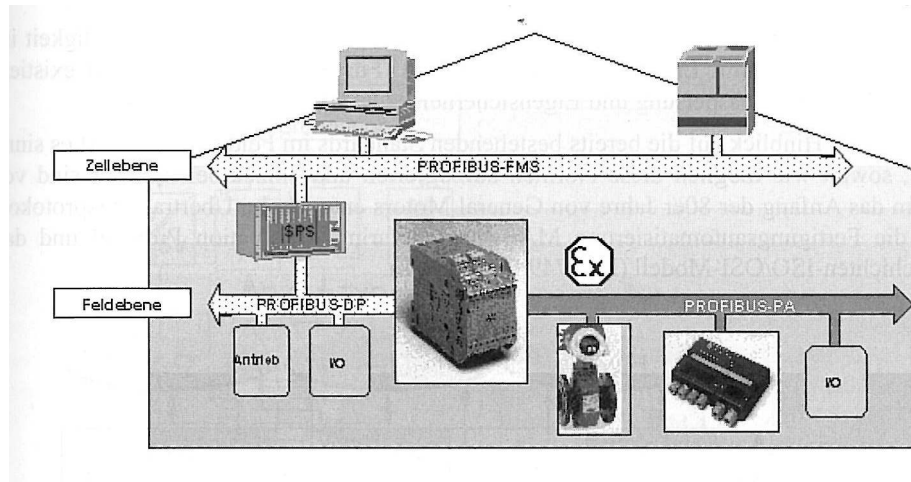


Abbildung 3 - Profibus-Anwendungsgebiete /2/

3.2. Übertragungstechnik DP/FMS

Das am häufigsten eingesetzte Übertragungsmedium für den Profibus DP/FMS ist eine geschirmte und verdrehte 2-Drahtleitung mit symmetrischer Datenübertragung nach dem Standard EIA RS485.

Übertragungsverfahren/Codierung

Die Übertragung der Daten basiert auf dem halbduplexen, synchronen RS485-Übertragungsverfahren. Die Bytes befinden sich dabei in einem 11-Bit-Zeichenrahmen, der im Non-Return-to-Zero-Code übertragen werden. Im Zeichenrahmen befinden sich neben den 8 Informationsbits zusätzlich ein Startbit, ein Paritätsbit und ein Stoppbit.

Während der Übertragung wird auf der Leitung-A das invertierte und auf der Leitung-B das nichtinvertierte Datensignal gesendet.

Der Ruhezustand der Leitung entspricht einem binären „1“-Signal.

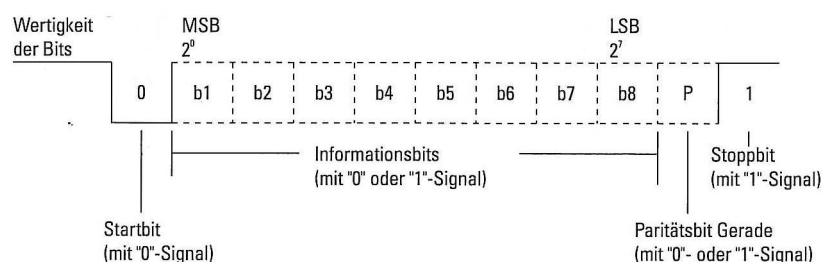


Abbildung 4 - Der Profibus UART-Zeichenrahmen /1/

Bus-Struktur

Die Struktur des Profibusses ist identisch mit dem RS485 Standard. Innerhalb eines Segmentes werden die Teilnehmer in Linienstruktur miteinander verbunden. Stichleitungen sind möglich. Hierbei ist zu beachten, dass nach der Norm bei einer Busgeschwindigkeit von 1,5MBit/s die Stichleitung eine Länge von 6 Metern nicht überschreiten sollte. Generell wird von dem Einsatz von Stichleitungen abgeraten. Ausnahmen bilden temporäre Stichleitungen von Programmier- oder Diagnosegeräten. Bei Busgeschwindigkeiten $>1,5\text{MBit/s}$ sind Stichleitungen nicht zulässig./1/ Da jedes angeschlossene Feldgerät eine Stromlast darstellt und das Signal abschwächt, sind nach dem RS485 Standard maximal 32 Teilnehmer zulässig. Durch den Einsatz von sog. Repeatern lassen sich weitere Teilnehmer anschließen. Ein Repeater bildet einen Teilnehmer eines Segmentes. Er verstärkt die Signale und stellt die Verbindung zu einem weiteren Segment dar. Beim Einsatz von Repeatern kommt es zu Verzögerungen der Bitsignale. Aus diesem Grund sind maximal 3 Repeater in Reihe nach der EN 50170 zulässig.

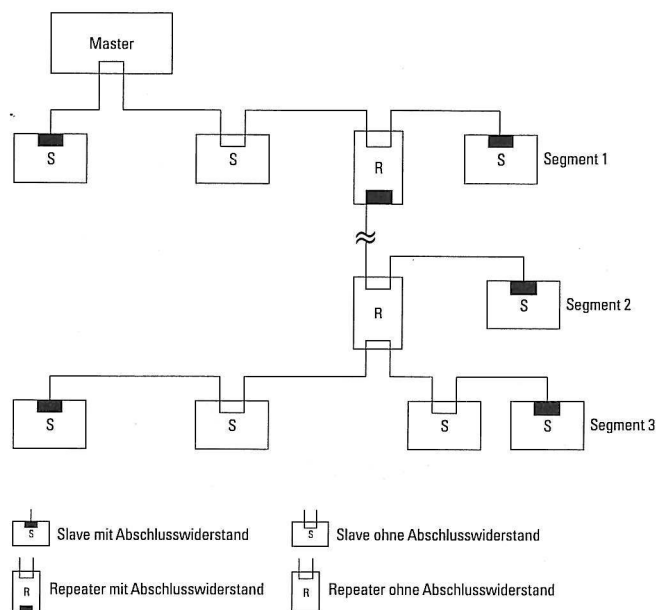


Abbildung 5 - Buskonfiguration mit Repeater /1/

Der Einsatz von Repeatern ermöglicht auch den Aufbau von Baum- oder Sternstrukturen in einem Profibus.

Busleitung

Die in einem Profibus eingesetzte maximal zulässige Leitungslänge (Segmentlänge) ist abhängig von der gewählten Übertragungsgeschwindigkeit. Im Bereich von 9,6kBit/s bis 12MBit/s sind, wie auch beim Standard RS485, in jedem Segment maximal 32 Teilnehmer möglich.

Baudrate (kBit/s)	9,6–187,5	500	1.500	12.000
Segmentlänge (m)	1.000	400	200	100

Abbildung 6 - Maximale Segmentlänge in Abhängigkeit der Baudrate /1/
Die in Abbildung 6 dargestellten Leitungslängen beziehen sich auf den in der Profibus-Norm spezifizierten Kabeltyp A (Abbildung 7).

Wellenwiderstand	135 bis 165 Ω , bei einer Messfrequenz von 3 bis 20 MHz
Kabelkapazität	< 30 pF pro Meter
Aderquerschnitt	> 0,34 mm ² , entspricht AWG 22
Kabeltyp	paarweise verdreht, 1 \times 2 oder 2 \times 2 oder 1 \times 4 Leiter
Schleifenwiderstand	< 110 Ω pro km
Signaldämpfung	max. 9 dB über die ganze Länge des Leitungsabschnittes
Abschirmung	Kupfer-Geflechschirm oder Geflechschirm und Folienschirm

Abbildung 7 - Spezifikation des Profibus-RS485-Kabels Typ A /1/
Bei Störung der Leitung wirkt sich dies auf beide Signale aus. Die Differenz beider Signale verändert sich dadurch nicht. Hierdurch findet eine wirksame Gleichtaktunterdrückung statt und verbessert die Störfestigkeit.
Zum Verbinden der einzelnen Busteilnehmer über die Busleitung werden in der Regel sog. 9-polige D-Sub-Steckverbindungen verwendet.

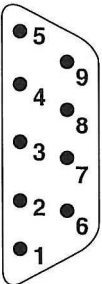
Ansicht	Pin-Nr.	Signalname	Bezeichnung
	1	SHIELD	Schirm bzw. Funktionserde
	2	M24	Masse der 24 V-Ausgangsspannung (Hilfsenergie)
	3	RxD/TxD-P	Empfangs-/Sendedaten-Plus B-Leitung
	4	CNTR-P	Signal für Richtungssteuerung-P
	5	DGND	Datenbezugspotential (Ground)
	6	VP	Versorgungsspannung-Plus
	7	P24	24V-Plus der Ausgangsspannung (Hilfsenergie)
	8	RxD/TxD-N	Empfangs-/Sendedaten-Minus A-Leitung
	9	CNTR-N	Signal für Richtungssteuerung-N

Abbildung 8 - Kontaktbelegung des 9-Poligen D-Sub-Steckers /1/

Busabschluss

An den Segmentenden des Profibusses befinden sich sog. Busabschlusswiderstände der Datenleitungen A und B. Dabei handelt es sich um einen Pulldown-Widerstand gegen das Datenbezugspotenzial und einen Pullup-Widerstand gegen das Versorgungsspannungs-Plus.

In dem Fall, dass kein Teilnehmer sendet, bilden diese Widerstände ein definiertes Ruhepotenzial auf der Busleitung welches einem binären „1“-Signal entspricht.

Viele Standard-Profibus-Stecker verfügen bereits über diese Kombination, so dass zur Aktivierung oftmals nur noch ein Schalter umgelegt werden muss.

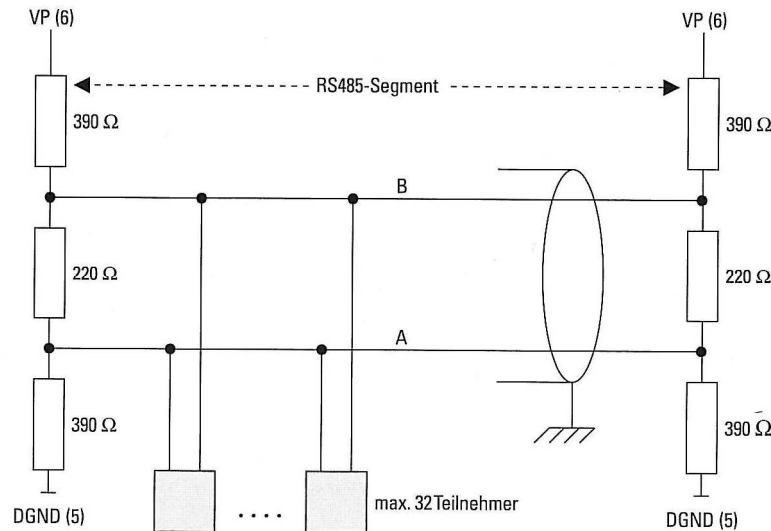


Abbildung 9 - Aufbau eines RS485-Bussegmentes /1/

Lichtwellenleiter

Eine Alternative zur 2-Drahtleitung bildet der Einsatz von Lichtwellenleitern. Hierdurch können Profibus-Systeme eine Länge von bis zu 15 km erreichen. Weitere Vorteile liegen in der optischen Datenübertragung, da diese unempfindlich gegenüber elektromagnetischen Störungen ist und grundsätzlich eine Potenzialtrennung gegeben ist.

Ein Nachteil liegt jedoch in der aufwendigeren Spleißtechnik und die durch LWL-Technik entstehenden Mehrkosten.

Unterschiede zu PA

Bei Anwendung des Profibus-PA kommt die Übertragungstechnik mit einer Baudrate von 31,25 kBit/sec nach IEC 61158-2 Standard zum Einsatz. Mit der Eigensicherheit und der Speisung der Feldgeräte über die Busleitung wird der Einsatz des Profibus-Systems in Ex-geschützten Bereichen ermöglicht.

Das Leitungsprotokoll arbeitet mit einer gleichstromfreien, bitsynchronen, manchestercodierten Übertragung.

Die Busleitung besteht, wie auch beim DP/FMS-Protokoll, aus einer 2-Drahtleitung. Auch hier sind maximal 32 Teilnehmer an jedem Segment zugelassen. Die Gesamtlänge eines Segmentes ist abhängig vom eingesetzten Kabeltyp.

Als Segmentabschluss dienen passive RC-Glieder.

3.3. Datenübertragung

Zur Steuerung der Datenübertragung und -verwaltung sowie des Buszugriffes dient die zweite Schicht des ISO/OSI-Referenzmodells. Für die Buszugriffssteuerung verwendet der Profibus ein hybrides Zugriffsverfahren. Zur Abwicklung der Datenübertragung (Field Data Link) und Datenverwaltung (Fieldbus Management) dienen die Dienste des Layer 2.

Hybride Buszugriffssteuerung

Um die Verwendung eines Profibusses mit mehr als einem Master zu ermöglichen, werden zwei Buszugriffsverfahren miteinander kombiniert. Dies stellt eine hybride Buszugriffssteuerung dar.

Eine Besonderheit des Profibusses ist, dass hierdurch auch die Kommunikation zwischen gleichberechtigten aktiven Teilnehmern (Master) sichergestellt ist. Hierbei erhält jeder Busteilnehmer innerhalb eines definierten Zeitfensters die Möglichkeit zu kommunizieren. Gleichzeitig ist durch den geringen Protokollaufwand ein einfacher und schneller Datenaustausch mit den passiven Peripheriegeräten (Slave) möglich. Bei Bedarf ist das System auch echtzeitfähig.

Token-Passing

Mit Hilfe des sog. Token-Passing-Verfahrens wird die Zugriffssteuerung der Master am Bus organisiert. Um zu verhindern, dass zwei oder mehrere Master gleichzeitig auf den Bus zugreifen, wird bei der Businitialisierung ein Token generiert. Die Master verfügen dabei über Werkzeuge, die gewährleisten, dass nur ein Token sich im Umlauf befindet. Weitere Token werden gelöscht.

Dieser Token wird nun zwischen den angeschlossenen Mastern herumgereicht. Wer im Besitz dieses Token ist, darf auf den Bus zugreifen. Mit der Token-Umlaufzeit ergibt sich aufgrund der Anzahl der aktiven Teilnehmer eine maximale Token-Haltezeit. Während der Token-Haltezeit ist jeder Teilnehmer mindestens einmal aktiv am Bus. Wenn er aktiv ist, hat er die Möglichkeit, seine Kommunikationsaufgaben abzuwickeln. Sollte das Zeitfenster hierfür nicht ausreichen, so muss er seine Kommunikation unterbrechen und nach Ablauf der Haltezeit den Token weiterreichen.

Master-Slave

Das Master-Slave-Verfahren steuert die Kommunikation zwischen den aktiven und den passiven Teilnehmern am Bus. Dabei ist es charakteristisch für dieses Verfahren, dass die Kommunikation zwischen Master und Slave stattfindet. Eine Kommunikation zwischen zwei Slaves ist nur indirekt über den Master möglich.

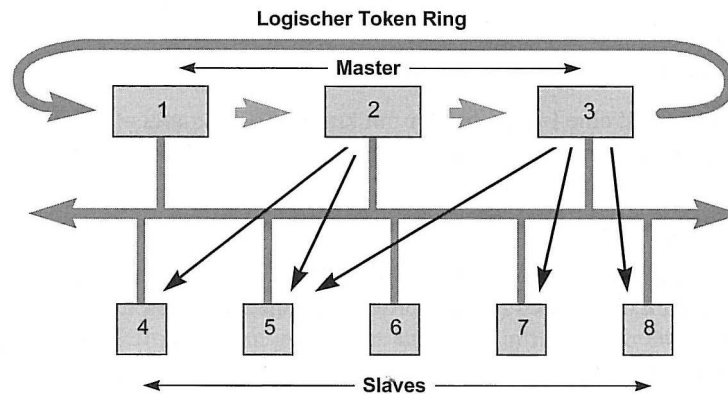


Abbildung 10 – Hybrides Profibus-Buszugriffsverfahren /2/

Wenn der Token-Ring aus nur einem Master besteht, handelt es sich um ein reines Master-Slave-System.

Datenaustausch

Um eine große Übertragungssicherheit zu ermöglichen, verwendet der Profibus fest vorgegebene Telegrammformate. Durch den Aufbau mit besonderen Start- und Stoppbits, sowie schlupffester Synchronisierung, Paritätsbit und Kontrollbit ist es möglich, gewisse Fehler zu erkennen. Durch den verwendeten Hamming-Abstand $HD=4$ können zusätzlich bis zu drei gleichzeitig auftretende Bit-Fehler erkannt werden.

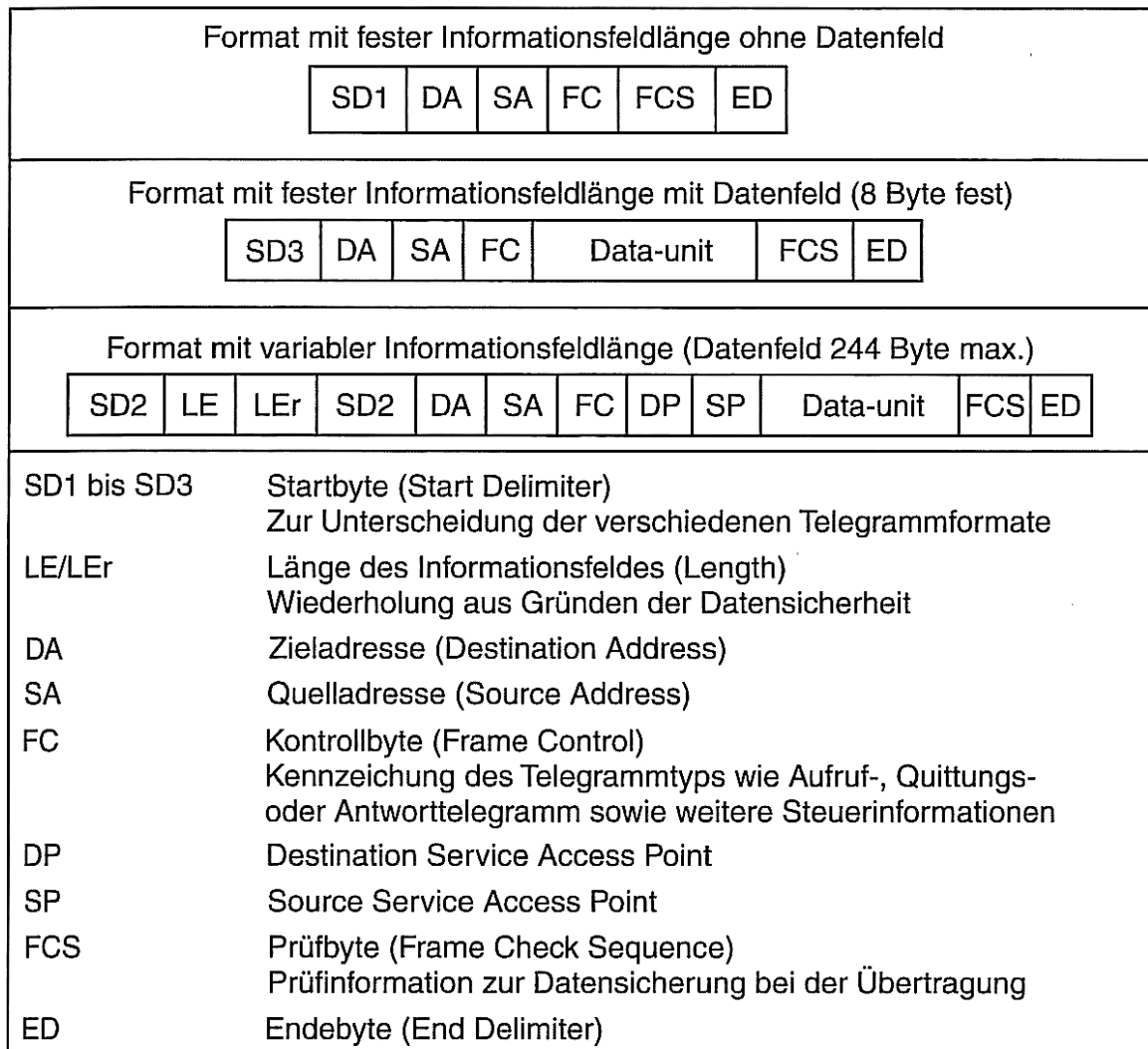


Abbildung 11 - Profibus-Telegrammformat /2/

Wird ein Telegrammfehler erkannt, so wird die Übertragung mindestens einmal wiederholt. Der Anwender kann bis zu 8 Wiederholungen einstellen.

Übertragungsprotokoll

Zur Datenübertragung und Kommunikationsaufbau dienen vier Übertragungsdienste.

SDN	Send Data with No Acknowledge (Daten-Sendung ohne Quittierung)
SDA	Send Data with Acknowledge (Daten-Sendung mit Quittierung)
SRD	Send and Request Data (Sendung mit Anforderung und Rückantwort)
CSRD	Cyclic Send and Request Data (Zyklischer SRD-Dienst)

Abbildung 12 - Datenübertragungsdienste /2/

Der SDN-Dienst dient hauptsächlich für eine Broadcast oder Multicast Kommunikation. Bei Broadcast werden alle am Bus angeschlossenen Geräte angesprochen, wohingegen beim Multicast nur eine bestimmte Gruppe aus mehreren Teilnehmern bestehend angesprochen wird. Um eine Kollision auf

dem Bus zu verhindern, wird hierbei auf eine Quittierung durch die Teilnehmer verzichtet.

Dieser Dienst dient z.B. zum synchronen Ansprechen von Busteilnehmern.

Bei den anderen Diensten handelt es sich um Punkt-zu-Punkt Kommunikation. Hierbei wird dem Anforderer (Requester), in der Regel ein Master, auf seine Anfrage an den Dienstbringer (Responder) ein Quittierungssignal oder eine Rückantwort gesendet.

Neben den elementaren Diensten SDA und SRD kann der Profibus-FMS Standard die Dienste SDN und CSRD verwenden. Dabei ist CSRD der einzige Dienst, der zyklisch ausgeführt wird.

Profibus-DP/PA Standard verwendet nur die Dienst SDA und SRD

Da der DP/PA Standard über keinen Application Layer verfügt, über die der Anwender auf den Data Link Layer zugreift, gibt es sog. Dienstzugriffspunkte. Diesen SAPs (Service Access Points) sind bestimmte Anwendungsdienste hinterlegt. Ein Teilnehmer kann über mehrere SAPs verfügen. Unterschieden wird dabei zwischen den eigenen Zugangspunkten (Local SAP) und denen der Partnerstation (Remote SAP).

Busparameter

Neben der Verwaltung der SAPs überwacht der Data Link Layer auch die Busparameter. Diese müssen bei allen Teilnehmern identisch sein. Sie werden in Abhängigkeit der Busgeschwindigkeit angegeben. Die Einheit für die Angabe ist tBIT.

Ein tBIT ist die sog. Bitlaufzeit. ($1 \text{ tBIT} = 1 / \text{Baudrate (bit/s)}$).

Die wichtigsten Busparameter und ihre Bedeutung:

Ttr	Die Soll-Token-Umlaufzeit (Target-Rotation-Time) ist die maximale zur Verfügung gestellte Zeit für den Tokenumlauf aller Busteilnehmer. In dieser Zeit bekommen alle aktiven Teilnehmer einmal das Recht (Token), Daten auf den Profibus zu senden. Die Differenz zwischen der Soll-Token-Umlaufzeit und der realen Token-Haltezeit eines Teilnehmers bestimmt, wie viel Zeit den anderen aktiven Teilnehmern für das Senden von Telegrammen noch zur Verfügung steht.
GAP-Faktor	Der GAP-Faktor (GAP = Lücke) legt fest, nach wie vielen Token-Umläufen versucht wird, einen neuen, aktiven Teilnehmer in den logischen Token-Ring aufzunehmen.
RETRY-Limit	Mit dem RETRY-LIMIT wird festgelegt, wie oft ein Telegramm nach fehlerhafter Quittierung oder Timeout maximal wiederholt wird.
Min_TSDR	Die minimale Protokoll-Bearbeitungszeit (min. station delay responder) ist die Zeit, die ein passiver Teilnehmer mindestens warten muss, um auf ein Telegramm antworten zu dürfen.
Max_TSDR	Die maximale Protokoll-Bearbeitungszeit (max. station delay responder) ist die Zeit, die ein passiver Teilnehmer maximal benötigen darf, um auf ein Telegramm zu antworten.
Tslot	Die Warte-auf-Empfang-Zeit (slot time) legt fest, wie lange der Sender maximal auf eine Antwort vom angesprochenen Teilnehmer wartet.
Tset	Die Auslösezeit (setup time) ist die Zeitspanne, die zwischen dem Empfang eines Telegramms und der Reaktion darauf im Teilnehmer ablaufen darf.
Tqui	Die Modulator-Ausklingzeit (Quiet-Time for Modular) beschreibt, wie lange ein sendender Teilnehmer nach dem Sendetelegrammende für das Umstellen auf Empfangen benötigen darf.
Tid1	Die Ruhezustandszeit 1 (idle time 1) legt fest, nach welcher Zeit ein sendender Teilnehmer nach dem Empfang einer Antwort frühestens wieder ein Telegramm senden darf.
Tid2	Die Ruhezustandszeit 2 (idle time 2) ist die Zeit, die ein Teilnehmer nach dem Senden eines unquittierten Telegramms (Broadcast) warten muss, um ein weiteres Telegramm zu versenden.
Trdy	Die Bereitschaftszeit (ready-time) beschreibt, nach welcher Zeit ein sendender Teilnehmer ein Antworttelegramm empfangen kann.

Abbildung 13 – Profibus-Busparameter /1/

3.4. Anwenderschicht

Profibus-FMS

Das Application Layer des ISO/OSI-Referenzmodells stellt für den Anwender die Dienste zur Verfügung, die ihm die Kommunikation auf dem Bus ermöglichen.

Ausgeprägt ist diese Schicht beim Profibus nur beim FMS-Standard. Hierbei werden unterschiedliche FMS-Profile angewendet. Diese stellen einen zuvor definierten Funktionsumfang zur Verfügung. Dadurch wird ermöglicht, dass auch die Geräte von unterschiedlichen Herstellern miteinander kommunizieren können.

Daher gibt es für unterschiedliche Anwendungsbereiche häufig unterschiedliche Profile. Ein Beispiel sind Gebäudeautomatisierungs-Systeme. Hier wird bei öffentlichen Ausschreibungen oft festgelegt, wie die Daten für Steuerung, Regelung oder auch Archivierung bearbeitet werden müssen. Ein weiteres Beispiel ist die Kommunikation zwischen SPS Steuerungen. Dabei ist festgelegt, welche Dienste den Austausch von Informationen der Teilnehmer unterstützen müssen.

Aufgrund seiner einfachen und umfangreichen Funktionalität findet der FMS-Standard in der Regel seine Anwendung in der Zellebene einer Produktionsanlage.

Profibus-DP/PA

Zur Kommunikation der Automatisierungsgeräte in der Feldebene wurde der DP/PA-Standard entwickelt. Er übergeht die Application Layer und greift aus Anwendersicht direkt auf die Data Link Layer zu. Durch dieses Vorgehen wird eine größere Geschwindigkeit beim Zugriff auf die Nutzdaten erreicht. Dies ist ein großer Vorteil in der Feldebene. Dabei werden die Nutzdaten mit Hilfe von standardisierten Diensten im User Interface, dem sog. Direct Data Link Mapper, der auf die Daten des Layer 2 direkt zugreift, zugänglich gemacht.

Eine Auswertung der Nutzdaten erfolgt hier durch den Anwender. Mit der Wahl bestimmter Dienstzugangspunkte werden Standarddienste ausgeführt. Dazu gehören beispielsweise die Diagnose, Parametrierung oder auch Konfiguration. Für den Nutzdatenaustausch wird der Default-SAP (Service Access Point) gewählt.

Im Laufe der Jahre wurden die Standard Dienste durch zusätzliche Dienste erweitert. Die ursprünglichen Dienste für Diagnose und zyklischen Datenaustausch sind in DP-V0 enthalten. Die nächste Version ist DP-V1. Hier sind neben den Diensten aus DP-V0 weitere Dienst für den azyklischen Datenaustausch und Alarmfunktionen enthalten. In der nächsten Erweiterung DP-V2 wird ein Dienst für einen quasi direkten Datenaustausch zwischen DP-Slaves hinzugefügt. Des Weiteren ist bei DP-V2 ein zeitlich konstanter Buszyklus für Echtzeitanwendungen möglich.

Zur Unterstützung gibt es aber auch beim DP-Standard definierte Profile. Sie dienen z.B. für die Steuerung von Antrieben oder Sicherheitseinrichtungen. In einem Profil ist die Bedeutung bestimmter Nutzdaten festgelegt.

Beim DP/PA Standard wird zusätzlich zwischen Master-Klasse-1 und Master-Klasse-2 unterschieden. Die Master-Klasse-1 dienen zum zyklischen Datenaustausch. Zur azyklischen Programmierung und Diagnose wird der Master-Klasse-2 verwendet. In der Praxis kommen auch Kombigeräte zum Einsatz.

Bei der Anwendung ist zu beachten, dass für jeden Slave nur ein Master Schreibzugriff auf die Nutzdaten erhält. Dabei handelt es sich um den Master, durch den der Slave zuvor parametrierung wurde. Hiermit wird verhindert, dass durch eine gleichzeitige Steuerung von unterschiedlichen Mastern es zu einem unkontrollierten Anlagenverhalten kommt.

GSD Datei

Zur Projektierung des Profibus-DP-Systems werden GSD-Dateien (Geräte Stamm Datei) verwendet. Diese werden während der Projektierung im Mastersystem mit Hilfe von Konfigurationswerkzeugen eingebunden. Sie enthalten Informationen für den Master über das angeschlossene Gerät. Dabei handelt es sich z.B. um Informationen, die der Master zur Initialisierung und Parametrierung der Slave-Geräte benötigt. Hierbei wird auch deutlich, welche Profile durch die Geräte unterstützt werden und es kann eine Auswahl des zu verwendenden Profils erfolgen.

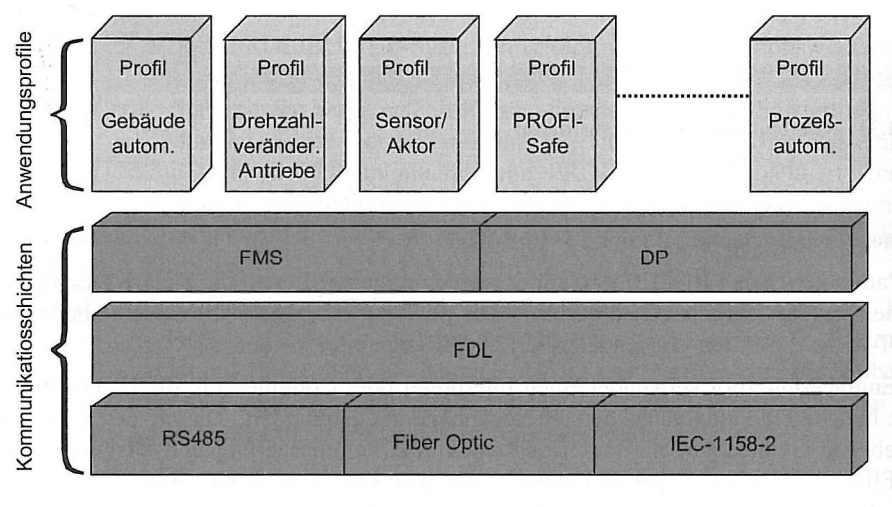


Abbildung 14 - Profibus-Profile /2/

4. DRIVECOM Profil 20

Wie bereits erwähnt, können bei einem Profibus-System sog. Profile verwendet werden. Hierdurch wird die Kommunikation mit der dezentralen Peripherie weitestgehend standardisiert. Profile gibt es dabei für die Gebäudetechnik, Sicherheitseinrichtungen, Antriebssysteme u.v.m.. Den größten Einsatzbereich von Profilen hat hierbei die Antriebstechnik. Dabei gibt es Profile, die speziell für den Profibus entwickelt wurden wie PROFIdrive, aber auch busunabhängige Profile. Das DRIVECOM Profile 20 bildet ein busunabhängiges Profil.

Im Folgenden, wird das DRIVECOM Profil 20 erläutert. Als Bezug sind die Quellen /8/ und /10/ zu beachten.

Dieses Profil beschreibt das Geräteverhalten sowie wichtige Parameter von drehzahlveränderlichen Antrieben. Die DRIVECOM Nutzergruppe e.V. ist ein Zusammenschluss von ca. 30 Unternehmen und Instituten. Entwickelt wurde das Profil auf dem Interbus. Sein Ziel ist die Definition eines Standards, der unabhängig von Bussystemen und Herstellern zum Einsatz kommt. Hierdurch ist es möglich, Antriebsregler unterschiedlicher Hersteller gegeneinander auszutauschen. Das Bussystem kann hierbei unberücksichtigt bleiben. Der Anwender kann das Bussystem frei wählen.

Das DRIVECOM Profil 20 dient zur Steuerung von Schritt-, Gleichstrom-, Servomotoren und Standardantrieben mit Frequenzumrichtern.

4.1. Datenstruktur

Das DRIVECOM Profil 20 arbeitet mit einem Prozessdaten- und einem Parameterdaten-Kanal. Die Prozessdaten werden hierbei zyklisch ausgetauscht.

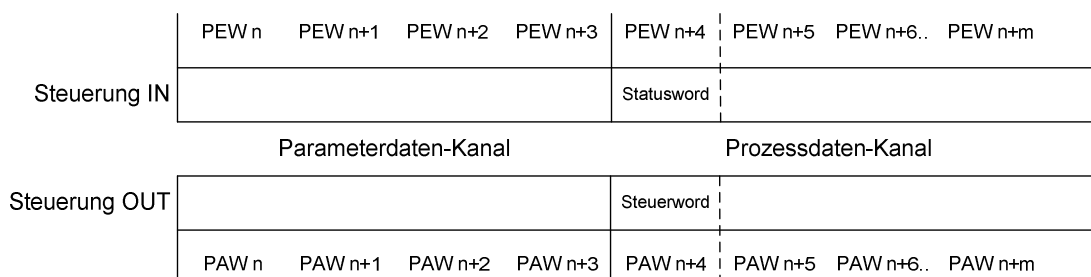


Abbildung 15 - Datenstruktur DRIVECOM

Prozessdaten-Kanal

Der Prozessdatenkanal dient zur Steuerung des Geräteverhaltens. Hierbei ist die DRIVECOM-Zustandsmaschine zu beachten.

Zur Informationsübertragung verfügt der Kanal über mindestens ein 16 Bit Statuswort sowie über ein 16 Bit Steuerwort. Abhängig vom Antrieb können weitere Prozessdatenwörter übertragen werden wie z.B. Soll- und Istwerte. Die Daten werden unquittiert und äquidistant übertragen.

Parameterdaten-Kanal

Der Parameterkanal erlaubt den Zugriff der Steuerung auf die Codestellen. Er dient damit zur Parametrierung des Antriebsreglers. Hierbei ist zu beachten, dass auf einige Codestellen nur zugegriffen werden kann, wenn der Regler gesperrt ist. Beim Zugriff auf die Codestellen wird dieser vom Antriebsregler bestätigt.

4.2. Geräteverhalten

Das Geräteverhalten eines Antriebssystems wird mit Hilfe der DRIVECOM-Zustandsmaschine definiert. Es ist vergleichbar mit einem Moore-Automaten. Die Ausgabe ist hierbei ausschließlich vom seinem Zustand abhängig und beschreibt hierbei ein internes oder ein externes Verhalten. Die Zustände können durch Zustandsübergänge wechseln. Einem Zustandsübergang geht immer ein Ereignis voraus. Diese Ereignisse können von außen (z.B. Steuerwort, Endlagenschalter) oder vom Antriebssystem (z.B. Grenzwertüberschreitung) ausgelöst werden. Informationen über den Zustand werden der Steuerung im Statuswort mitgeteilt.

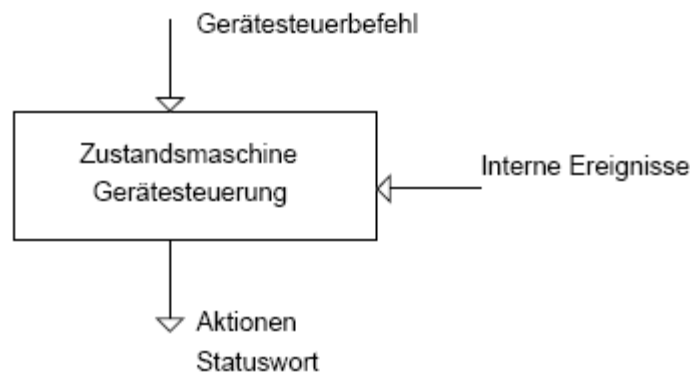
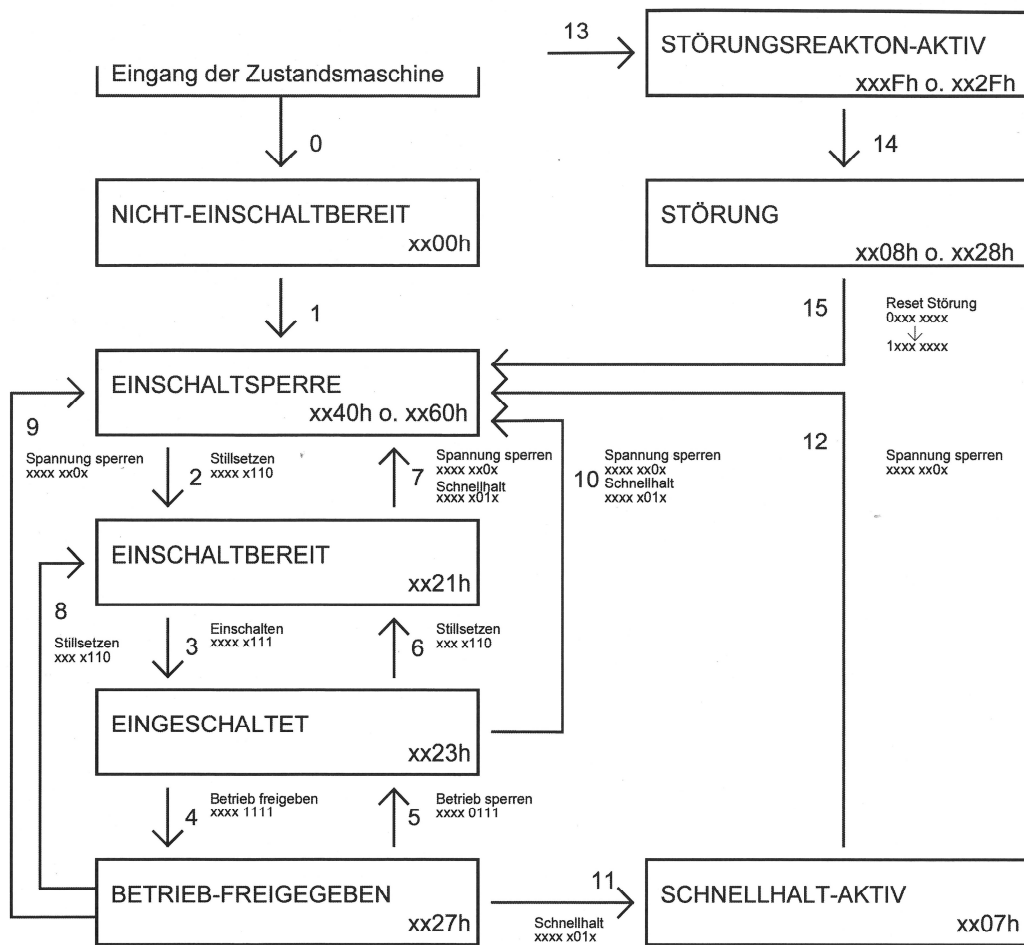


Abbildung 16 – Zustandsänderung /8/

Zustandsmaschine



Innerhalb der Zustände wird das 'Statuswort', an den Zustandsübergängen das 'Steuerwort' angegeben. Die Form xxxx xxxx zeigt nur die bits 7...0 in binärer Darstellung, xxxhx zeigt den hexadezimalen Ausdruck.

Abbildung 17 – Zustandsdiagramm /8/

Zustände

Nicht Einschaltbereit

Nach dem Einschalten wechselt der Antrieb selbstständig in den Zustand „Nicht Einschaltbereit“. Hierbei wird die Elektronik mit Spannung versorgt und es wird ein Selbsttest nach Abschluss der Initialisierung ausgeführt. Die Antriebsfunktion ist dabei gesperrt, d.h. es kann keine Energie an den Motor abgegeben werden. Es werden die gespeicherten Parameterwerte geladen.

Einschaltsperr

Nach Abschluss der Initialisierung und des Selbsttestes wechselt der Antrieb automatisch in den Zustand „Einschaltsperr“.

Auch in diesem Zustand sind die Antriebsfunktionen gesperrt. Es kann jedoch die Anwendung umparametriert werden.

Das System ist weiter gegen Einschalten gesperrt.

Einschaltbereit

Die Antriebsfunktionen sind in diesem Zustand weiter gesperrt. Eine Umparametrierung kann stattfinden.

Das Leistungsteil kann eingeschaltet werden.

Eingeschaltet

In diesem Zustand sind die Antriebsfunktionen weiter gesperrt. Es können jedoch die Anwendungen umparametriert werden und das Leistungsteil ist funktionsbereit.

Betrieb-Freigegeben

Die Antriebsfunktionen sind freigegeben und das Leistungsteil kann Energie an den Motor abgeben. Einige Parameter können nun nicht geändert werden, da der Antrieb für ihre Änderung gesperrt sein muss.

Schnellhalt-Aktiv

Der Antrieb wird in diesem Zustand durch einen vorgegebenen Ablauf in den Stillstand geführt. Danach wechselt er automatisch in den Zustand „Einschaltsperr“.

Störungsaktion-Aktiv

Wird durch den Antrieb eine Störung erkannt, so wechselt dieser in den Zustand „Störungsaktion-Aktiv“. Dabei führt der Regler eine fehlerabhängige Reaktion aus. Nach Abschluss wechselt er in den Zustand „Störung“. Während dieses Zustandes kann die Antriebsfunktion freigegeben sein.

Störung

In diesem Zustand ist die Antriebsfunktion gesperrt und der Regler kann umparametriert werden.

Zustandsübergänge

Stillsetzen

Mit diesem Gerätesteuerbefehl kann aus unterschiedlichen Zuständen in den Zustand „Einschaltbereit“ gewechselt werden. Dabei können während des Überganges unterschiedliche Aktionen ausgeführt werden. So wird z.B. das Leistungsteil ausgeschaltet oder die Antriebssperrfunktion ausgelöst.

Einschalten

Durch den Befehl „Einschalten“ wechselt der Antrieb vom Zustand „Einschaltbereit“ in den Zustand „Eingeschaltet“. Dabei wird das Leistungsteil eingeschaltet.

Spannungs-Sperren

Durch den Befehl „Spannungs-Sperren“ kann aus unterschiedlichen Zuständen in den Zustand „Einschaltsperr“ gewechselt werden. Dabei wird die Antriebssperrfunktion ausgeführt und das Leistungsteil wird ausgeschaltet.

Schnellhalt

Bei dem Befehl „Schnellhalt“ ist zu beachten, in welchem Zustand sich der Antrieb befindet. Wenn sich der Regler im Zustand „Betrieb-Freigegeben“ befindet, so wird durch diesen Befehl die Schnellhaltefunktion ausgeführt.

Aus dem Zustand „Eingeschaltet“ bewirkt dieser Befehl ein Ausschalten des Leistungsteils.

Betrieb-Freigeben

Hierbei wird vom Zustand „Eingeschaltet“ in den Zustand „Betrieb-Freigegeben“ gewechselt. Die Antriebsfunktionen werden freigegeben.

Betrieb-Sperren

Mit diesem Befehl wechselt der Antrieb aus dem Zustand „Betrieb-Freigegeben“ in den Zustand „Eingeschaltet“. Während des Überganges wird die Antriebssperr-Funktion ausgeführt.

Reset-Störung

Um nach einer Störung den Antrieb wieder in Betrieb nehmen zu können, muss zunächst der Befehl „Reset-Störung“ ausgeführt werden. Hierdurch wechselt der Regler aus dem Zustand „Störung“ in den Zustand „Einschaltsperr“.

Es ist zu beachten, dass dieser Befehl nur Sinn macht, wenn die Störung behoben ist und nicht wieder auftritt. Im anderen Fall wechselt der Antrieb immer wieder in den Zustand „Störung“.

Antriebssperrfunktion

Die Antriebssperrfunktion definiert das Verhalten bei einem Zustandsübergang. Sie ist herstellerepezifisch und parametrierbar.

Schnellhaltefunktion

Diese Funktion definiert das Verhalten im Zustand „Schnellhalt-Aktiv“. Sie ist wie die Antriebssperrfunktion parametrierbar.

4.3. Steuerwort

Die Abbildung 17 zeigt die Belegung der Bits im Steuerwort. Die mit „mandatory“ gekennzeichneten Bits sind dabei fest vorgegeben und dürfen bei einer zusätzlichen Konfiguration nicht verändert werden.

Bit	Name	mandatory
0	Einschalten	X
1	Spannung-Sperren	X
2	Schnellhalt	X
3	Betrieb-Freigeben	X
4	betriebsartabhängig	
5	betriebsartabhängig	
6	betriebsartabhängig	
7	Reset-Störung	X
8	Reserve	
9	Reserve	
10	Reserve	
11	herstellerspezifisch	
12	herstellerspezifisch	
13	herstellerspezifisch	
14	herstellerspezifisch	
15	herstellerspezifisch	

Abbildung 18 - Bitbelegung Steuerwort /8/

Durch bestimmte Kombinationen der Bits im Steuerwort können die Gerätesteuerbefehle ausgelöst werden.

Die Bits 4-6 dienen bei erweitertem Funktionsumfang zum Steuern der unterschiedlichen Betriebsarten. So bietet das DRIVECOM Profil 22 z.B. zusätzliche Funktionen in den Betriebsarten Lageregelung oder Geschwindigkeitsregelung an.

Mit den Bits 11-15 können herstellerspezifische Funktionen ausgelöst werden.

4.4. Statuswort

Zur Darstellung der Informationen über den Zustand des Reglers dient das Statuswort.

Bit	Name	mandatory
0	Einschaltbereit	X
1	Eingeschaltet	X
2	Betrieb-Freigegeben	X
3	Störung	X
4	Spannung-Gesperrt	X
5	Schnellhalt	X
6	EINSCHALTSPERRE	X
7	Warnung	
8	Meldung	
9	Remote	X
10	Sollwert-Erreicht	
11	Grenzwert	
12	betriebsartenabhängig	
13	betriebsartenabhängig	
14	herstellerspezifisch	
15	herstellerspezifisch	

Abbildung 19 - Bitbelegung Statuswort /8/

Neben den verbindlich vorgegeben Bitbelegungen gibt es auch hier frei belegbare Bits. Diese sind teilweise betriebsartabhängig oder herstellerspezifisch zu verwenden.

4.5. Parameterkommunikation

Zur Parametrierung der Codestellen des Antriebsreglers über einen Feldbus dient der Parameterkanal. Dieser setzt sich beim DRIVCOM Profil aus einem 8 Byte langen Telegramm zusammen.

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index High Byte	Index Low Byte	Data 4 / Error 4	Data 3 / Error 3	Data 2 / Error 2	Data 1 / Error 1

Abbildung 20 - Telegrammaufbau Parameterkanal /10/

Bei dem DRIVCOM Profil wird das Telegramm zyklisch mit übertragen. Dabei belegt es zusätzlich 4 Worte der Datenübertragung.

Servicebyte

Das Servicebyte dient zur Auftrags- und Antwortsteuerung für die Parameterkommunikation. Durch das Invertieren des Handshakes wird ein neuer Auftrag aktiviert.

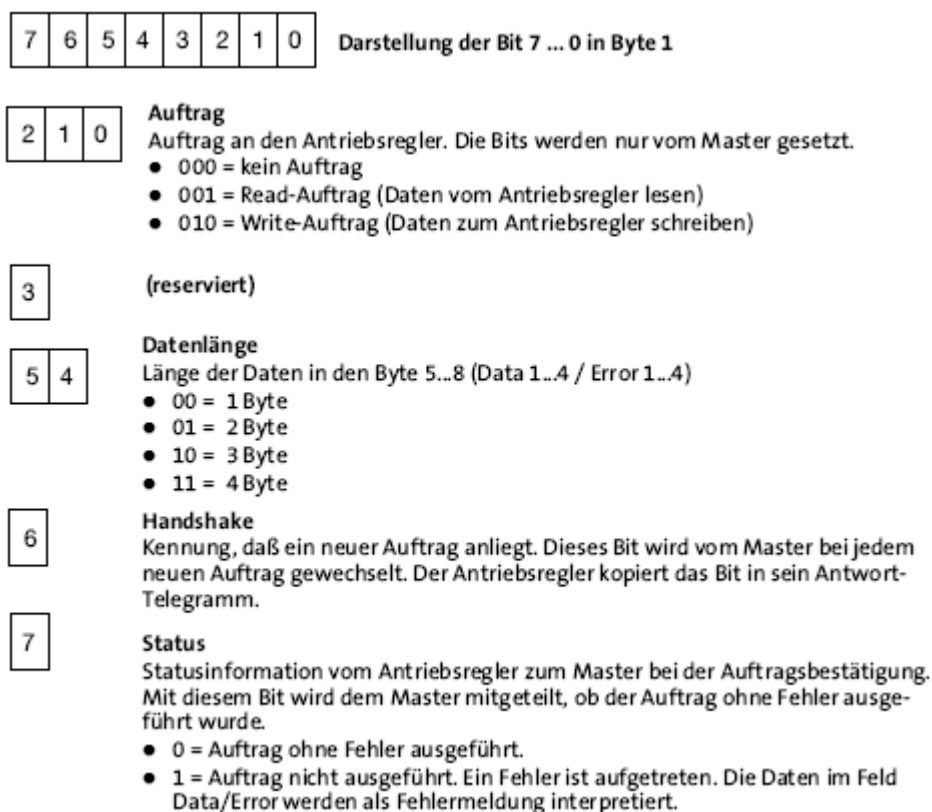


Abbildung 21 – Servicebyte /10/

Index/Subindex

Die Bytes für Subindex und Index enthalten die Adresse der Parameter im Antriebsregler. Dabei steht Index für den Parameter und Subindex für einen Unterpunkt des Parameters.

Daten

Die Daten enthalten die Informationen über den Parameter.

Bei einem Read-Auftrag können hieraus die Daten vom Antrieb gelesen werden. Beim Schreiben werden die Daten hier vorgegeben.

Fehlererkennung

In einem Fehlerfall beim Parametrieren oder Diagnostizieren dienen die Data-Bytes zur Fehlererkennung. Ein Beispiel hierfür ist, dass wenn ein Parameter neu geschrieben werden soll, dieser aber nur lesbar ist.

5. Servoantrieb

Servoantriebe sind eine besondere Form von drehzahlveränderlichen Antrieben. Dabei zeichnen sie sich durch ihre hohe Dynamik und genaue Regelung aus. Meist dienen sie zur Drehzahl-, Lage- oder Positionierregelung.

Zur Darstellung der Funktionsweise eines Positionierservoantriebes wird sich in diesem Kapitel im Wesentlichen auf die Quelle /4/, /5/ und /6/ bezogen.

Grundsätzlich handelt es sich bei Servoantrieben um sog. 4-Quadrant-Antriebe, d.h. sie können sowohl motorisch als auch generatorisch in beiden Drehrichtungen betrieben werden.

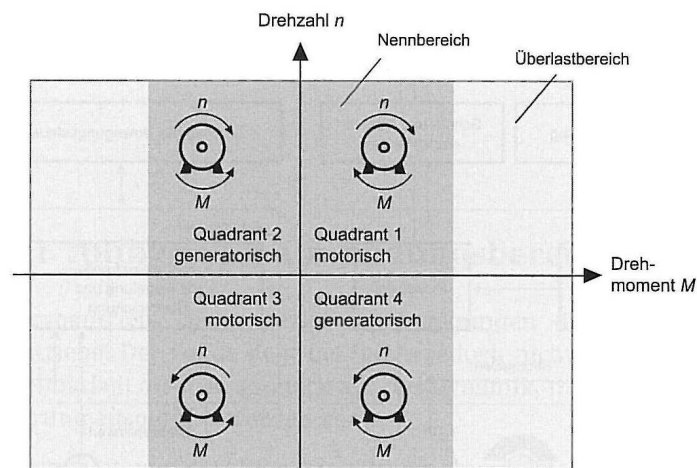


Abbildung 22 - 4-Quadranten Betrieb /5/

Der Aufbau eines Servoantriebes ist ähnlich wie bei drehzahlveränderlichen Antrieben. Als Energiewandler werden sog. Servomotoren eingesetzt. Diese unterscheiden sich durch ihre Bauform von herkömmlichen Motoren.

Um eine möglichst genaue Regelung zur erlangen, verfügen Servoantriebe über einen Motorgeber. Hierdurch bildet sich ein geschlossener Regelkreis.

Das Stellgerät wird durch einen Servoregler ersetzt. Dieser besteht aus dem Umrichter sowie einer Signalelektronik, die zur Regelung, Überwachung sowie zur Verarbeitung von Stellsignalen dient. Der Servoregler verarbeitet dabei die Signale von überlagerten Steuerungen und dem Motorgeber. Er verfügt bei modernen Systemen über einen Micro Controller zur digitalen Signalverarbeitung. Dies ermöglicht eine komplexe Konfiguration und Programmierung des Servoreglers.

Bei Bedarf verfügt der Antrieb zusätzlich über sog. Haltebremsen. Dies ist z.B. oftmals bei senkrechten Bewegungen der Fall.

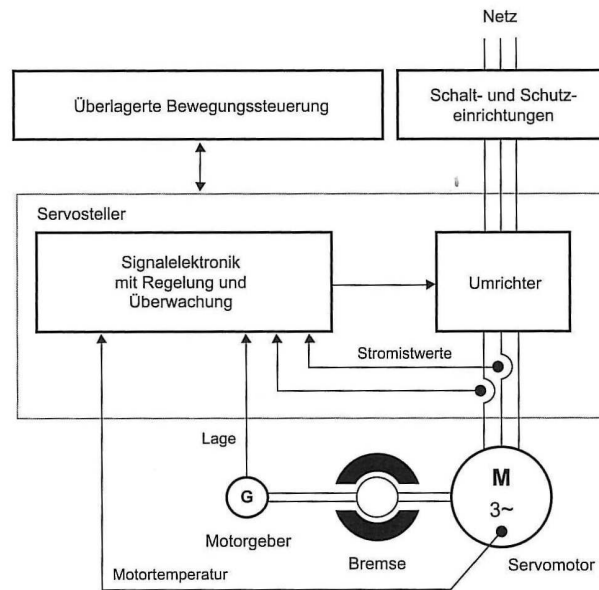


Abbildung 23 - Aufbau von Servoantrieben mit Synchronmotor /5/

5.1. Servomotor

Als Servomotor dienen sowohl Gleichstrom- als auch Synchron- oder Asynchronmotoren. Die ersten Servomotoren waren Gleichstrommotoren, da diese mit einer vergleichbar einfachen Regelung eingesetzt werden konnten. Mit der Weiterentwicklung der Leistungselektronik und der Entwicklung der Vectorregelung wurden auch die Drehstrommotoren als Servomotoren entdeckt. Ihr großer Vorteil gegenüber Gleichstrommotoren ist ihre Wartungsfreiheit.

Als Kenngröße von Servomotoren dient im Allgemeinen nicht die Leistungsangabe, sondern die Angabe des Motorbemessungsdrehmoments.

Für kleine bis mittlere Lasten bis ca. 200Nm werden heute häufig Synchronmotoren eingesetzt. Für größere Lasten bis ca. 2500Nm werden Asynchronmotoren verwendet.

Charakteristisch für einen Servomotor ist seine Bauform. Betrachtet man die allgemeine Drehmomentengleichung und stellt diese so um, dass man die Beschleunigung erhält, so ist zu erkennen, welche Eigenschaften für eine hohe Dynamik erforderlich sind.

$$J \frac{d\omega}{dt} = M_M$$

M_M	Motormoment
J	Massenträgheitsmoment
ω	Winkelgeschwindigkeit

$$\frac{d\omega}{dt} = \frac{M_M}{J}$$

Abbildung 24 - Drehmomentengleichung

Aus diesem Grund verfügt ein Servomotor über ein geringes Trägheitsmoment und über eine vergleichbar mechanisch steife Konstruktion mit einer kurzen kräftigen Welle. Hiermit wird eine hohe Überlastbarkeit für ein dynamisches Motordrehmoment erreicht. Servomotoren sind dabei um den Faktor 2 bis 3 kurzzeitig im Drehmoment überlastbar.

Durch die hohen Beschleunigungs- und Bremsmomente wird im Motor viel Energie in Wärme umgewandelt. Dabei erreichen die Gehäuse Temperaturen von bis zu 150°C.

Eine besondere Form der Servomotoren bilden die Direktantriebe. Bei diesen Antrieben kann auf Bewegungsumformer und Getriebe verzichtet werden. Die Bewegung wird dabei direkt an die Arbeitsmaschine übertragen. Linearmotoren, eine besondere Form der Synchronmotoren, sorgen hierbei z.B. für eine translatorische Bewegung. Ein weiterer Direktantrieb ist der sog. Torquemotor. Dieser arbeitet von seiner Funktion ähnlich wie Linearmotoren, erzeugt jedoch eine rotatorische Bewegung mit einer sehr geringen Drehzahl.

5.2. Motorgeber

Für eine genaue Regelung der Motoren verfügen Servoantriebe über Motorgeber. Diese dienen zur Rückführung relevanter Messgrößen und schließen den Regelkreis.

Zur Kommutierung von Blockstrommotoren (z.B. bürstenloser Gleichstrommotor) werden sog. Kommutierungsgeber verwendet. Sie geben Informationen zur Kommutierung, woraus sich zusätzlich Drehzahlen berechnen lassen.

Häufig verfügen Servoantriebe jedoch über Lagegeber. Diese geben Information über die Lage des Läufers.

Ein Beispiel ist der Resolvergeber. Dabei wird auf der sich mit dem Läufer bewegendem Welle eine Spule angebracht. Auf diese Spule wird eine Spannung induziert. An ihren Enden wird die Spule kurzgeschlossen und es fließt ein Kurzschlussstrom. Dieser Kurzschlussstrom induziert nun in zwei um 90° versetzt angeordneten Messspulen auf dem Ständer jeweils eine weitere Spannung. Aufgrund der Amplituden und Phasenverschiebung dieser beiden Spannungen zueinander, kann nun die Lage des Läufers bestimmt werden. Die Genauigkeit dieses Verfahrens ist meist ausreichend. Dabei ist die Auflösung des A/D Wandlers des Servoreglers mit ausschlaggebend.

Eine höhere Genauigkeit, wie sie z.B. bei Werkzeugmaschinen erforderlich ist, wird durch den Einsatz von Sin-Cos-Gebern erreicht. Dabei kommt ein optisches Messverfahren zum Einsatz.

Neben der Auflösung des A/D Wandlers ist hier die Anzahl der Inkremente/Umdrehung für die Genauigkeit ausschlaggebend.

Die Unterschiede der verschiedenen Motorgeber liegen neben der Messgröße und Messverfahren in der zu verwendenden Schnittstelle. Beim Resolvergeber werden mit einer analogen Schnittstelle zwei Spannungen gemessen, wohingegen beim Sin-Cos-Geber eine digitale Schnittstelle zur kontinuierlichen Zählung von Signalperioden erforderlich ist.

5.3. Servoregler

Der Servoregler bildet das Herzstück eines Servoantriebs. Er besteht aus dem Leistungsteil, der Schnittstelle für den Motorgeber, einer Steuerung und dem Regler. Um eine hohe Dynamik zu erreichen, wird die Vectorregelung verwendet. Diese führt zu einer dynamischen Stromregelung und damit zu einem dynamischen Drehmoment im Motor. Es ist dabei eine kurzzeitige hohe Überlastbarkeit des Leistungsteils erforderlich.

Zum Einsatz kommen digitale Regler, da mit ihnen die komplexe Vectorregelung beherrschbar ist.

Die Auslegung des Leistungsteils ist vom Motor abhängig. Bei Gleichstromservomotoren werden sog. Pulssteller verwendet, die, anders als die Stromsteller, nicht netzgeführt geregelt werden.

Für Drehstrommotoren kommen Umrichter zum Einsatz. Sie bestehen aus einem Gleichrichter, einem Zwischenkreis und einem Wechselrichter. Für den generatorischen Betrieb ist der Zwischenkreis mit einem Bremschopper und einem Bremswiderstand zu erweitern. Dabei wird die überschüssige Energie in Wärme umgewandelt. Durch den Einsatz von Rückspeisemodulen kann diese Energie auch in das Netz zurückgeführt werden.

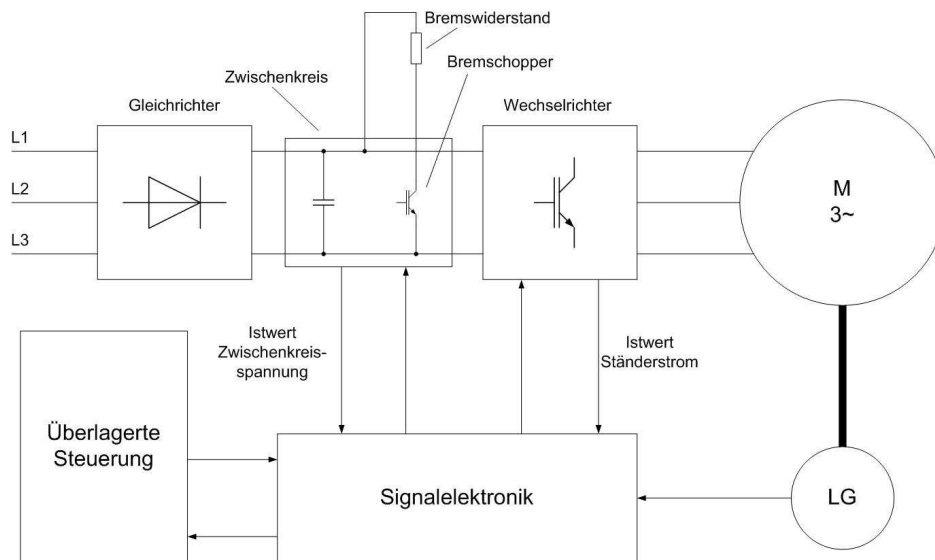


Abbildung 25 - Servoantrieb mit Servoregler (Frequenzumrichter)

Vectorregelung

Um den Läufer zu beschleunigen oder zu verzögern gilt es, das Drehmoment des Motors zu verändern. Hierzu wird der Strom des Motors geregelt. Zur Ansteuerung der Ventile des Umrichters, um einen entsprechenden Strom im Motor über die Strangspannungen einzustellen, arbeitet der Servoregler nach dem Prinzip der Pulsweitenmodulation. Die Regelung erfolgt durch Vectorregelung.

Grundlage der Vectorregelung ist die Betrachtung von Spannung, Strom und magnetischem Fluss als Raumzeiger. Diese beschreiben den stationären, geometrischen Zustand einer Größe auf dem Motor. Der jeweilige Raumzeiger wird dabei durch Überlagerung der einzelnen Zeiger in den Wicklungen gebildet. Zur mathematischen Beschreibung der Raumzeiger werden diese auf einem Koordinatensystem abgebildet.

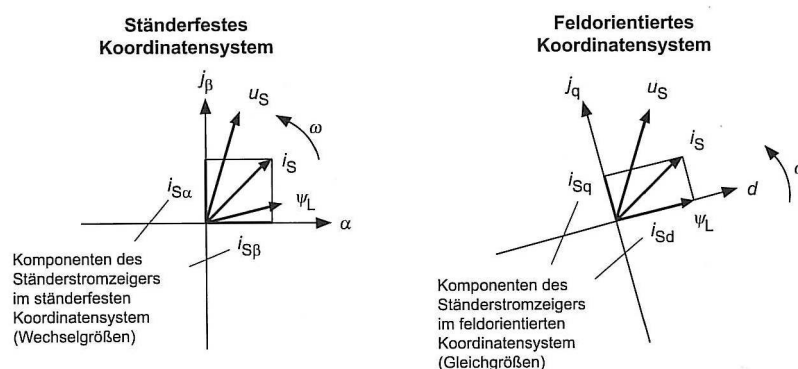


Abbildung 26 - Beschreibung des Strom-Raumzeigers in verschiedenen Koordinatensystemen /5/

Das ständerfeste Koordinatensystem ist fest mit dem Ständer verbunden und steht still. Die Raumzeiger weisen einen sinusförmigen Verlauf aus.

Das feldorientierten Koordinatensystem orientiert sich an dem Fluss. Dieser ist nach dem Magnetfeld des Läufers ausgerichtet und dreht sich mit der Frequenz der speisenden Spannung. In diesem Koordinatensystem weisen die Größen keinen sinusförmigen Verlauf mehr auf. Sie sind nun als Gleichgrößen zu sehen.

Mit Bezug auf das feldorientierte Koordinatensystem kann der Strom nun in zwei Komponenten zerlegt werden. Eine ist die feldbildende Komponente. Diese ist parallel zum Fluss ausgerichtet. Sie ist vergleichbar mit dem Erregerstrom der Gleichstrommaschine.

Beim Asynchronmotor regelt diese Komponente das magnetische Feld im Läufer.

Im Synchronmotor bildet der Permanentmagnet das Feld. Der Regler hält die Komponente auf Null.

Die zweite Komponente ist im rechten Winkel zum Fluss ausgerichtet. Dadurch wird sie auch als drehmomentbildende Komponente bezeichnet. Sie hat dieselbe Wirkung wie der Ankerstrom einer Gleichstrommaschine.

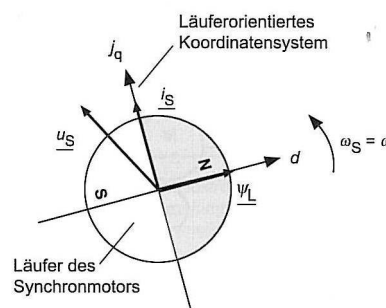


Abbildung 27 - Läuferflussorientiertes Koordinatensystem des Synchronmotors

/5/

Mit Hilfe dieser Betrachtung kann die Regelung nun wie bei einem Gleichstromservoantrieb erfolgen.

Die Größen werden hierbei jedoch mathematisch betrachtet und müssen mit Hilfe eines Motormodells auf die realen Größen umgerechnet werden. Hierzu ist es erforderlich, dass dem Servoregler Angaben über den angeschlossenen Motor gemacht werden, um ein Modell bilden zu können.

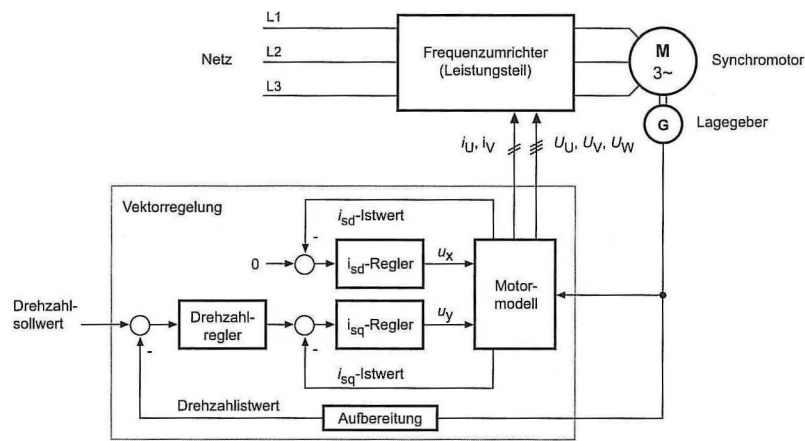


Abbildung 28 - Regelungsstruktur des Servoreglers beim Synchronmotor /5/

Reglereinstellungen

Moderne Antriebe haben heute digitale Regler. Diese arbeiten diskontinuierlich. Da jedoch die Abtastzeiten dieser Regler heute hinreichend klein und vergleichbar sind mit analogen Motorreglern, wird weitergehend von einer kontinuierlichen Regelung ausgegangen. Es wird jedoch darauf hingewiesen, dass bei einer genaueren Betrachtung dieser Umstand berücksichtigt werden sollte./5/

Zur Regelung von Drehmoment, Drehzahl und Lage verfügen Servoregler meist über eine kaskadierte Reglerstruktur.

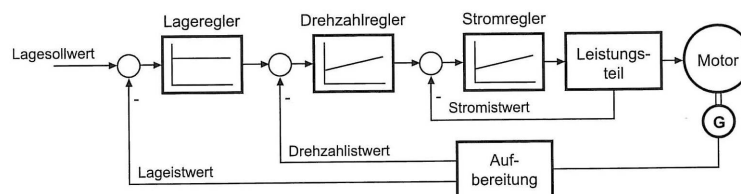


Abbildung 29 - Struktur der Kaskadenregelung bei elektrischen Antrieben /5/

Die innere Kaskade regelt den Strom und das Drehmoment. Zur Drehzahlregelung dient die mittlere Kaskade. Mit dem äußeren Regelkreis erfolgt die Lageregelung.

Die Einstellung der Regler erfolgt nun von innen nach außen.

Stromregelkreis

Der Stromregelkreis bildet sich aus dem Stromregler und der Strecke. Zur Strecke gehören das Leistungsteil und die Motorwicklungen.

Die Einstellung des inneren Regelkreises für die Stromregelung erfolgt bei modernen Systemen oft durch Selbstoptimierung. Hierzu dienen die vorgegebenen oder gemessenen Motorparameter.

Das Ziel beim Stromregelkreis ist ein gutes Führungsverhalten, dass mit einem PI-Regler eingestellt wird. Dieses Ziel ist sinnvoll, da die im Stromregelkreis auftretende Störung, die drehzahlabhängige induzierte Spannung, nur über eine geringe Dynamik verfügt und vom Regler problemlos beherrscht werden kann.

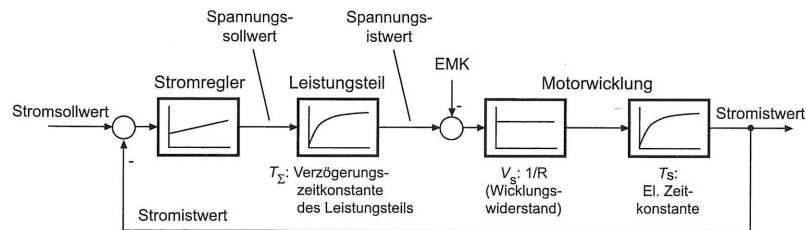


Abbildung 30 - Stromregelkreis am Beispiel eines Gleichstrommotors /5/

Das Verhalten der Strecke ist mit einem P-T1-Glied zu vergleichen.

Zur Bildung eines großen Frequenzbereichs, in dem der Istwert dem Sollwert folgen kann, erfolgt die Einstellung der Proportionalverstärkung und Nachstellzeit des PI-Reglers nach dem Betragsoptimum. Hierdurch wird eine hohe Grenzfrequenz erreicht. Zur Bestimmung der Einstellgrößen aus den Motorparametern und der Ersatzzeitkonstanten des Leistungsteils dienen dabei folgende Bestimmungsgleichungen:

$$K_p = \frac{T_s}{V_s \cdot 2T_\Sigma} \quad (I)$$

$$T_N = T_s \quad (II)$$

Die Ersatzzeitkonstante T_Σ beschreibt die Dynamik, mit der eine geforderte Spannungsänderung vom Regler an den Motorklemmen umgesetzt werden kann.

Durch das Einstellen der Reglerparameter nach dem Betragsoptimum liegt die Anregelzeit bei etwa dem 5-fachen der Ersatzzeitkonstanten. Hierdurch wird deutlich, wie die Dynamik der Stromregelung vom Leistungsteil abhängt. Aus diesem Grund werden bei Servoantrieben meist hochdynamische Leistungsteile verwendet.

Die Überschwingweite liegt bei unter 5%, durch das Einstellen nach dem Betragsoptimum. Hierdurch wird ein Schutz wirksam, der das Erreichen der Strombegrenzung verhindert.

Drehzahlregelkreis

Der Drehzahlregelkreis ist dem Stromregelkreis überlagert. Bei der Einstellung für die Drehzahlregelung kann der Stromregelkreis näherungsweise als P-T1-Glied betrachtet werden. Zur Regelung dient ein PI-Glied.

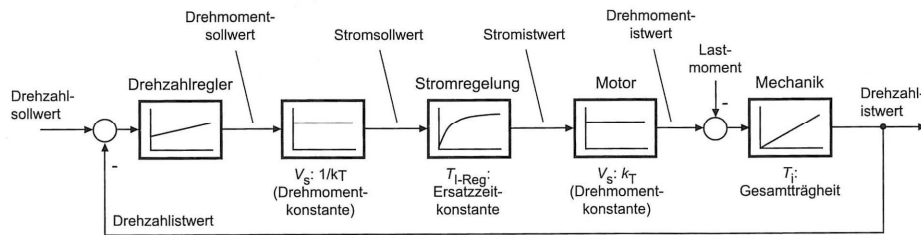


Abbildung 31 – Drehzahlregelkreis /5/

Als Störgröße tritt bei der Drehzahlregelung das Lastmoment auf. Dieses kann sich z.B. durch Reibung oder auch Unwucht deutlich verändern. Aus diesem Grund ist es das Ziel der Einstellung, ein möglichst gutes Störverhalten zu erreichen. Die Einstellung erfolgt dabei nach dem symmetrischen Optimum. Es wird erreicht, dass Störgrößen bis zu einer hohen Grenzfrequenz unterdrückt werden.

Durch die Einstellung nach dem symmetrischen Optimum lassen sich die Parameter des PI-Reglers durch folgende Bestimmungsgleichungen angeben:

$$K_p = \frac{T_i}{2T_{I-Reg}} \quad \text{mit } T_i = J \cdot s / \text{kgm}^2 \quad (\text{III})$$

$$T_N = 4T_{I-Reg} \quad (\text{IV})$$

Es wird dabei deutlich, dass sich die Parameter aus der Ersatzzeitkonstanten des Stromregelkreises und dem Trägheitsmoment des gesamten mechanischen Systems (bestehend aus Trägheitsmoment des Motors und der Arbeitsmaschine) ergeben.

Die Ersatzzeitkonstante des Stromregelkreises liegt dabei oft etwa beim 4-fachen der Ersatzzeitkonstanten des Leistungsteils. Hierdurch wird deutlich, wie die Dynamik der Drehzahlregelung von der Dynamik der Stromregelung abhängt.

Eine Angabe des Trägheitsmomentes des Gesamtsystems ist in der Regel nicht gegeben. Dies macht es erforderlich, das Trägheitsmoment messtechnisch zu bestimmen.

Viele Servoregler verfügen über Funktionen, mit denen das Trägheitsmoment automatisch bestimmt werden kann. Dies erfordert jedoch einen unerwünschten Hochlauf, so dass in der Praxis die Parameter oftmals manuell eingestellt werden.

Die Einstellung nach dem symmetrischen Optimum führt zu einer kurzen Anregelzeit und einem großen Überschwingen mit etwa 45% bei Betrachtung der Übertragungsfunktion.

Dieses „scharfe“ Verhalten macht den Einsatz hochauflösender Geber in Servoantrieben erforderlich, da es sonst bereits durch geringe Abweichungen zu einem unerwünschten Verhalten kommen kann. Dies ist besonders im Stillstand oder bei sehr geringen Drehzahlen zu beachten.

Lageregelkreis

Zur Einstellung des Drehzahlregelkreises verfügen moderne Servoregler heute meist über eine Automatik, die zur Ermittlung der Parameter dient. Für eine manuelle Einstellung der Lageregelung kann der Regelkreis für die Drehzahl näherungsweise als PT1-Glied betrachtet und mit einem P-Glied als Lageregler erweitert werden.

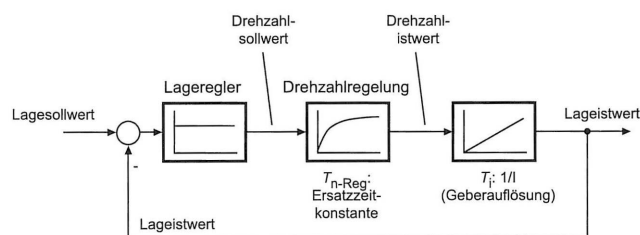


Abbildung 32 - Lageregelkreis /5/

Wenn davon ausgegangen wird, dass der Drehzahlregelkreis nach dem symmetrischen Optimum eingestellt wurde, kann näherungsweise die Ersatzzeitkonstante für die Drehzahlregelung mit dem 3-fachen der Ersatzzeitkonstanten des Stromregelkreises angenommen werden.

Um die Lageregelung nun mit einem guten Führungsverhalten zu optimieren, kann dieses nach dem Betragsoptimum eingestellt werden. Der Proportionalfaktor des P-Reglers ergibt sich dann aus der Auflösung des Lagegebers und der Ersatzzeitkonstanten des Drehzahlregelkreises.

$$k_v = \frac{T_i}{2T_{n-Reg}} \quad \text{mit } T_i = \frac{60}{\text{Auflösung des Gebers in U/min}} \quad (V)$$

Hierbei wird deutlich, dass die Dynamik der Lageregelung von der unterlagerten Drehzahlregelung abhängig ist.

Um den Schleppfehler, der sich als Regeldifferenz durch das P-Glied einstellt, gering zu halten, sollte die Verstärkung möglichst groß gewählt werden. Für den Schleppfehler gilt dabei die Beziehung:

$$F_{Schlepp} = \frac{n}{k_v} \quad (VI)$$

Funktionsumfang

Neben der komplexen Regelung des Antriebes verfügen moderne Servoregler über einen umfangreichen Funktionsumfang. Dazu zählen unter anderem Funktionen für die Diagnose und Inbetriebnahme. Zusätzlich können Schutz- und Überwachungsfunktionen vom Servoregler ausgeführt werden. Zur Anbindung in Automatisierungssystemen stehen Funktionen für die Kommunikation zur Verfügung. Dabei können unterschiedliche Sollwert- oder Steuerquellen ausgewählt werden.

Arbeitet der Antrieb als Positionierantrieb, kann die Sollwertvorgabe von einer überlagerten Steuerung oder von einem internen Verfahrsprogramm erfolgen. Bei einem Verfahrsprogramm ist es erforderlich, dieses zunächst im Servoregler zu programmieren. Anschließend kann es durch wenige Bits gesteuert werden. Die Programmierung erfolgt dabei meist mit Hilfe einer PC-gestützten Entwicklungsumgebung, die die jeweiligen Antriebshersteller liefern können. Mit dieser Entwicklungsumgebung erfolgt dabei auch die Diagnose und Inbetriebnahme.

5.4. Positionieren

Das Positionieren mit Antrieben wird ermöglicht durch das Steuern der Bewegung mit Positionieranwendungen. Dabei werden nacheinander unterschiedliche Lagesollwerte eingestellt, um einen vorgegebenen Verfahrsweg zu erhalten. Sollten diese steuernden Eingriffe nicht erfolgen, so würde der Antrieb bei Lagesollwertänderungen immer den kürzesten Weg fahren. Dieses könnte es zu einem undefinierten Verhalten der Arbeitsmaschine führen.

Um das gewünschte Verhalten und Verfahrsweg des Antriebes zu erhalten, werden Positioniersteuerungen verwendet.

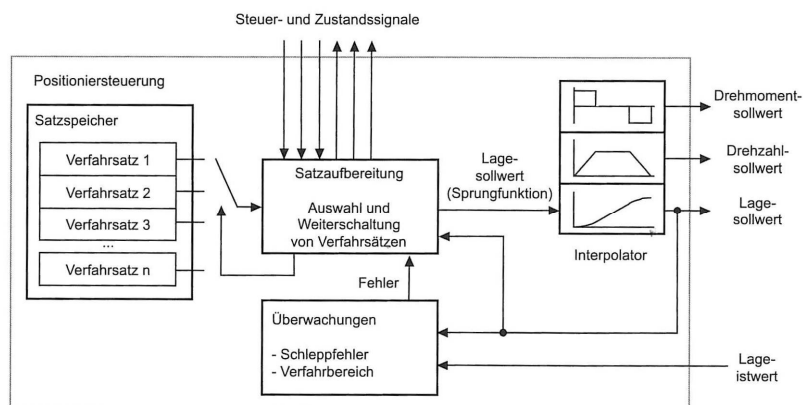


Abbildung 33 - Funktionelle Bestandteile eine Positioniersteuerung /5/

Der Fahrweg wird bei einer Positioniersteuerung mit Fahrätzen eingestellt. Dabei kann ein Fahrweg aus mehreren Fahrätzen bestehen, die dann nacheinander durchfahren werden.

Beim Durchlaufen eines Fahrweges werden die Fahrätze nicht direkt an der Regelung als Sollwerte eingestellt. Zunächst werden die Vorgaben aus den Fahrätzen aufbereitet, da sie neben der Vorgabe des Lageziels weitere Informationen enthalten.

Diese Aufbereitung koordiniert die Positioniersteuerung. Sie reagiert auf Fehler, die bei der Lageistwertüberwachung erkannt werden. Daneben verarbeitet sie die einzelnen Fahrwege und erstellt die entsprechenden Lagesollwertsprünge.

Der eigentliche Lagesollwertsprung für die Regelung wird von einem Interpolator generiert. Dieser Interpolator berücksichtigt den zeitlichen Verlauf des Lagesollwertes und leitet daraus zusätzliche Vorsteuerwerte für Drehzahl und Beschleunigung ab.

Fahrätze

Um das Verhalten des Antriebes einzustellen, enthalten die Fahrätze viele zusätzliche Informationen.

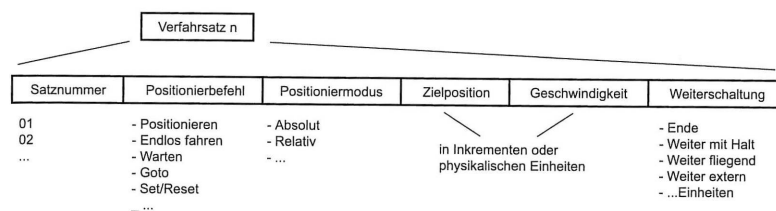


Abbildung 34 - Wichtige Bestandteile eines Fahratzes /5/

Neben der Zielposition enthalten sie bspw. Informationen, mit welcher Beschleunigung und Geschwindigkeit sich der Antrieb bewegen soll. Zusätzlich können Vorgaben über das Verhalten während der Bewegung gemacht werden. Hierbei kann z.B. auf Signale von außen gewartet werden, die einen Fahratz beenden und/oder zum Aufruf des nächsten Fahratzes führen. Dabei ist es wichtig, dass der aktive Fahratz Angaben enthält, welcher Fahratz im nächsten Aufruf aktiviert werden muss oder ob die Positionierung beendet wird.

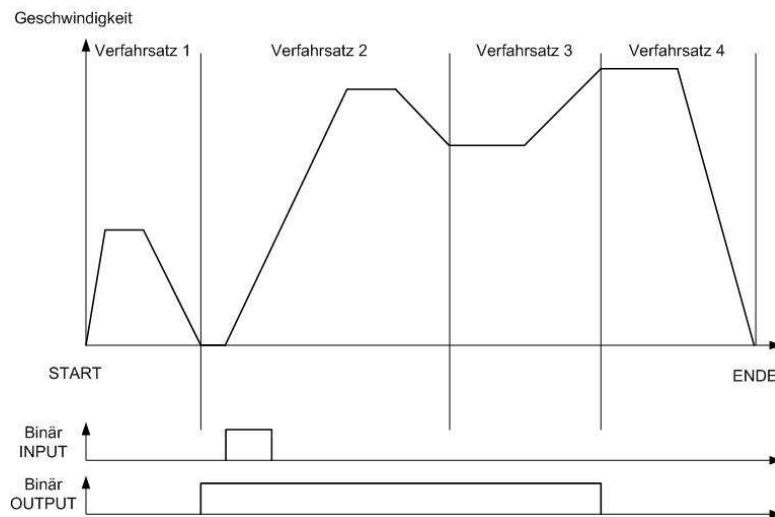


Abbildung 35 - Fahrweg bestehend aus mehreren Verfahrungsätzen
 Die Vorgaben zum Positionieren können dabei durch relative oder absolute Zielvorgaben erfolgen.

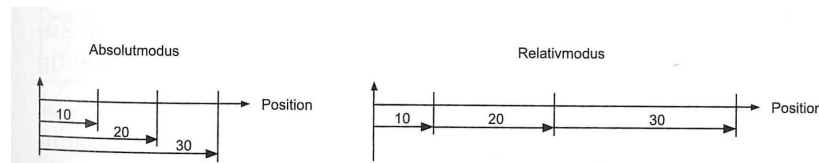


Abbildung 36 - Interpretation der Zielposition beim absoluten und relativen Positionieren /5/

Steuerung

Durch eine entsprechende Programmierung kann die Positioniersteuerung mit wenigen Signalen gesteuert und Rückschlüsse auf ihren Zustand erfolgen. Dabei werden in der Regel einige Standardsignale benutzt. Z.B. als Steuersignal gibt es „Start“ und „Stopp“ und als Zustandssignal „Bearbeitung läuft“ und „Startbereit“.

Steuersignale	Zustandssignale
Anwahl Betriebsart (mehrere Signale)	Achse ist referenziert
Tippen vorwärts	Achse fährt vorwärts
Tippen rückwärts	Achse fährt rückwärts
Schnell/langsam	Position erreicht und Halt
Satz überspringen	Aktuelle Betriebsart (mehrere Signale)
Anwahl Override (mehrere Signale)	Aktuelle Satznummer (mehrere Signale)
Anwahl Satz (mehrere Signale)	Bearbeitung läuft
Quittieren	Wartezeit läuft
Restweg löschen	Funktion beendet
Start	Startfreigabe erteilt

Abbildung 37 - Typische Steuer- und Zustandssignale einer Positioniersteuerung /5/

Mit diesen Signalen kann die Positioniersteuerung nun grundsätzlich gesteuert werden. Zusätzlich können weitere Signale programmiert werden, die dann zur Bearbeitung der einzelnen Verfahrssätze erforderlich werden.

Interpolator

Wie bereits erwähnt, werden Lagesollwertsprünge nicht direkt an dem Regler eingestellt. Zur Berechnung des wirkenden Lagesollwertsprunges dient der Interpolator, damit ein Überschwingen der Lageregelung verhindert wird. Dabei wird der Weg zwischen Start und Stopp des Sprunges interpoliert. Der Interpolator berechnet drei Bereiche: Beschleunigung, Konstantfahrt und Bremsen. Für die Berechnung dieser Bereiche müssen dem Interpolator zusätzlich einige Angaben über das Arbeitssystem gemacht werden. Er berücksichtigt dabei z.B. maximale und minimale Positionen, maximale Drehzahl und Beschleunigung sowie Getriebefaktoren und Referenzpositionen.

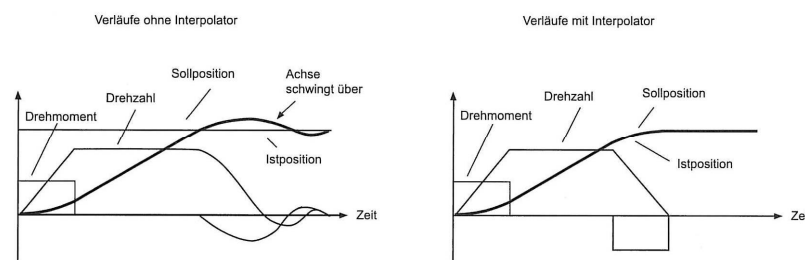


Abbildung 38 - Bedeutung des Interpolators für überschwingungsfreie Positioniervorgänge /5/

Überwachung

Zur Vermeidung von Fehler und Störungen bei der Positionierung werden Lageistwert und Aufbereitung ständig überwacht. Hierbei werden Grenzwerte für Position, Drehzahl des Motors, Geschwindigkeit der Arbeitsmaschine und Beschleunigungen beobachtet und mit vorgegebenen Werten verglichen.

Weiter muss hierbei der Lageistwert ständig aufgearbeitet werden. Der Lageistwert kann dabei direkt von der Motorwelle oder an der Arbeitsmaschine gemessen werden. Wenn der Lageistwert mit Hilfe des Lagegebers des Motors bestimmt wird, so müssen mögliche Getriebefaktoren bei der Umrechnung berücksichtigt werden. Die Positionierstrecke kann dabei aus mehreren Umdrehungen bestehen. Die Länge der Positionierstrecke bildet sich dabei aus der Auflösung des Gebers und aus der Genauigkeit der Systemberechnung. Bei einer 32Bit Berechnung und einem Geber mit 2048 Inkrementen/Umdrehung können damit $2^{32}/2048 = 2097152$ Umdrehungen erfasst werden./5/

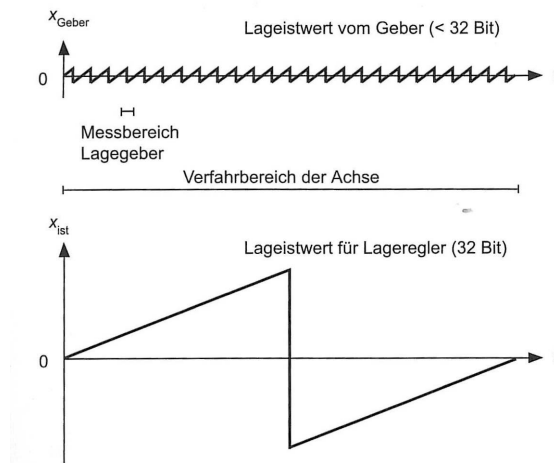


Abbildung 39 - Aufbereitung des gemessenen Lageistwertes für den Lagegeber
/5/

Nach dem Einschalten kann das System eine sog. Referenzfahrt ausführen. Dabei wird ein bestimmter Punkt auf dem Verfahrbereich angefahren und durch ein Referenzsignal erkannt. Die Arbeitsmaschine hat nun ihre Referenzposition erreicht. Bei Werkzeugmaschinen, die in der Regel eine hohe Genauigkeit erfordern, wird zusätzlich der Nullimpuls des Lagegebers beim Referenzieren berücksichtigt.

Bei Positionierungen für z.B. Werkstoffvorzüge, die in der Regel nur in eine Richtung arbeiten und für die ein unendlicher Verfahrbereich erforderlich ist, werden dann am Ende oder am Anfang des Fahrweges die Positionen zurückgesetzt. Eine Referenzfahrt ist hier in der Regel nicht erforderlich.

5.5. Sicherheitseinrichtungen

Sicherheitseinrichtungen versetzen Servoantriebe in einen sicheren Zustand. Wird z.B. eine Schutztür geöffnet, so muss gewährleistet sein, dass der Antrieb keine Bewegungen ausführen kann, die zu einer Gefährdung führen. Servoantriebe können nicht wie herkömmliche Antriebe einfach vom Netz geschaltet werden. Dabei gingen wichtige Informationen wie z.B. die Position verloren. Um dies zu verhindern, verfügen die Antriebe heute über eine Impulssperre.

Bei einer Impulssperre wird die Energieversorgung zum Motor unterbrochen, indem die Ansteuerimpulse im Stellgerät gesperrt werden. Bei einer weiteren sicheren Impulssperre wird die Stromversorgung für die Ventilansteuerung unterbrochen. Der Motor kann nun kein Drehmoment mehr erzeugen und keine gefährliche Bewegung mehr ausführen.

Zur Deaktivierung der Impulssperre muss eine Impulsfreigabe erfolgen und die Sperre für die Stromversorgung darf nicht aktiv sein. Diese Freigaben

können direkt von einer SPS an den Servoregler verdrahtet werden, oder auch über einen sicheren Feldbus erfolgen.

Eine weitere Einrichtung für die Sicherheit ist die Vorgabe von sicheren Bewegungen. Hier wird z.B. überwacht, ob der Antrieb innerhalb eines bestimmten Zeitfensters zum Stillstand kommt oder ein vorgegebenes Drehmoment überschritten wird. Wenn eine dieser Bedingungen nicht erfüllt ist, wird eine Impulssperre ausgelöst.

5.6. Auslegung

Bei der Auslegung eines Antriebes erfolgt zunächst die Wahl der Antriebsart. Dabei werden unterschiedliche Kriterien beurteilt. So muss betrachtet werden, ob es sich um eine Lineare- oder eine Rotationsbewegung handelt, die vom Motor ausgeführt werden soll. Ob es sich um eine konstante oder veränderliche Bewegung handelt? Welche Art von Netzversorgung steht zur Verfügung? Eigenschaften, die hierbei weiter für die Wahl eines Servoantriebes sprechen, sind seine hohe Dynamik und Überlastbarkeit gegenüber einem Standardantrieb. Dabei verfügt der Servoantrieb zusätzlich bei Bedarf über hohe Drehzahlen und die Möglichkeit für Positionierungen mit einer großen Genauigkeit.

Motor

Nach der Wahl der Antriebsart erfolgt die Auslegung des Motors. Eine Möglichkeit zur Auswahl des Motors ist dabei, aus einer Vielzahl an Motoren nacheinander die in Frage kommenden Modelle einzugrenzen.

Es erfolgt dabei zunächst eine mechanische Auslegung. Hierzu wird die anzutreibende Last betrachtet. Die relevanten Größen wie Lastdrehmoment und Massenträgheitsmoment müssen hierfür bestimmt werden.

Im nächsten Schritt erfolgt die Dimensionierung eines möglichen Getriebes. Dieser Schritt erübrigt sich bei Direktantrieben. Die Wahl des Getriebes erfolgt so, dass die Antriebsdrehzahl in den Bereich gängiger Motordrehzahlen von 700U/min bis 6000U/min fällt. Weiter soll das Getriebe dafür sorgen, dass für die Bewegung ein minimales Motordrehmoment erforderlich wird. Dies ist dann der Fall, wenn gilt:

$$i^2 = \frac{J_{Last}}{J_{Motor}} \quad \text{mit } i \text{ Getriebefaktor}$$

(VII)

Für Servoanwendungen ist es durch den Bewegungsablauf oftmals nicht möglich, dass Übersetzungsverhältnis genau zu bestimmen, da sich hierdurch unsinnige Motordrehzahlen einstellen können. Es wird dann erforderlich, einen Getriebefaktor nahe dem Optimum zu wählen, bei dem gleichzeitig ein großer Drehzahlarbeitsbereich abgedeckt wird.

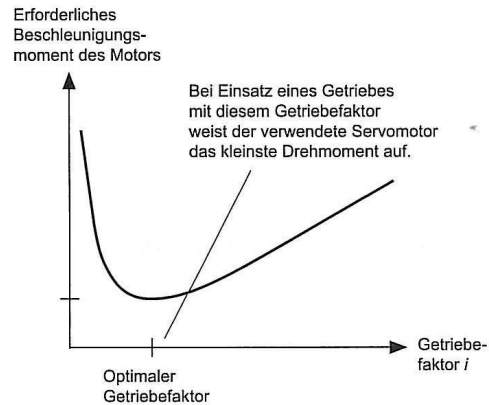


Abbildung 40 - Optimaler Getriebefaktor bei Servoanwendungen /5/

Nachdem das Getriebe bestimmt ist, werden die Lastgrößen auf die Motorseite des Getriebes umgerechnet. Es erfolgt nun die Bestimmung des erforderlichen Drehmoments des Motors.

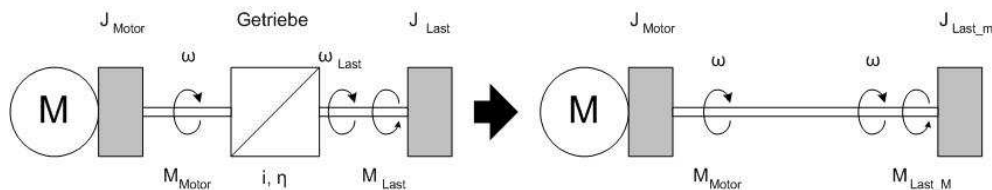


Abbildung 41 - Ersatzsystem für direkten Bezug auf Motorwelle

Die Ersatzgrößen für Lastmoment und Trägheit lassen sich berechnen mit

$$J_{Last_M} = \frac{J_{Last}}{i^2} \quad \text{und} \quad M_{Last_M} = \frac{M_{Last}}{i} \quad (\text{VIII})$$

Da bei Servoantrieben die Motordrehzahl und -drehmoment zeitlich veränderliche Größen darstellen, muss hierbei ihr Verlauf betrachtet werden. Es ergeben sich dabei mehrere Arbeitspunkte. Um nun einen Motor entsprechend der Anforderung auszuwählen, muss dessen Drehzahldrehmomentkennlinie verglichen werden. Erst wenn alle Arbeitspunkte innerhalb des zulässigen Arbeitsbereiches der Kennlinie liegen, kann der Motor verwendet werden.

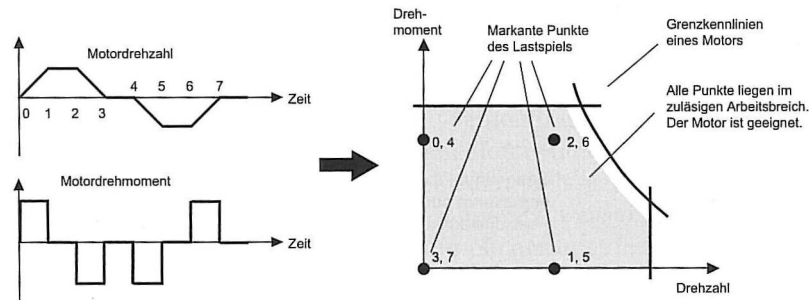


Abbildung 42 - Übertragung markanter Arbeitspunkte in das Drehzahl-Drehmoment-Diagramm des Motors /5/

Um ein schnelles Altern des Motors zu verhindern, muss im nächsten Schritt eine thermische Auslegung erfolgen. Durch die im Motor auftretenden Verluste entsteht Wärme, die zur Beschädigung der Wicklungsisolierungen führen kann. Hersteller erstellen hierzu sog. S1-Kennlinien von den Motoren. Mit Hilfe dieser Kennlinie wird beschrieben, welches Drehmoment bei welcher Drehzahl dauerhaft vom Motor abgegeben werden kann, ohne dass dieser überhitzt. Es wird die mittlere absolute Drehzahl und das Effektivmoment als Ersatzarbeitspunkt bestimmt.

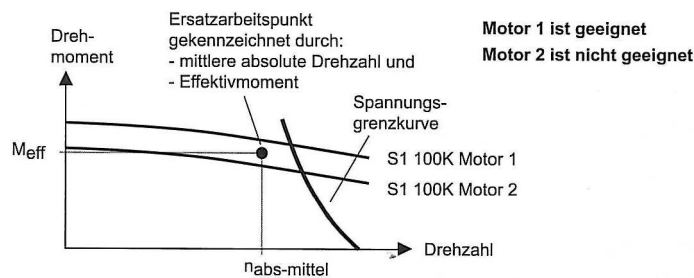


Abbildung 43 - Ersatzarbeitspunkt zur thermischen Auslegung /5/

Die Spannungsgrenzkurve wird gegeben durch die maximale Ausgangsspannung des Reglers und begrenzt das Motordrehmoment in Abhängigkeit von der Drehzahl.

Im nächsten Schritt erfolgt die Auslegung nach der baulichen Konstruktion. Hierbei wird der Motor z.B. nach Baugröße, Bauform, Belüftungsart, Schutzart, Kabelanschluss und Explosionsschutz bewertet. Einige dieser Größen sind dabei in Normen festgelegt.

Da die Hersteller oftmals Komplettsysteme anbieten, bestehend aus Motor und Motorgeber, erfolgt an dieser Stelle auch die Auswahl des Gebers. Bei der Auswahl des Gebers ist die benötigte Auflösung entscheidend. Die Auflösung ist jedoch nicht nur vom Geber abhängig, sondern auch von der Auswertelektronik des Reglers. Endgültig kann die Wahl des Gebers daher erst nach der Auswahl des Reglers erfolgen.

Sollten am Ende dieses Verfahrens noch mehrere Motoren zur Auswahl stehen, so erfolgt am Schluss eine Betrachtung nach wirtschaftlichen Aspekten.

Servoregler

Der Servoregler ist zwischen Motor und Netzversorgung angeordnet. Er stellt die für den Motor erforderliche Leistung zur Verfügung. Dabei sind ausgangseitig Spannungssystem, Spannungsamplitude, Stromamplitude und Erdungssystem durch den Motor bereits vorgegeben. Es müssen dabei die Spitzenwerte betrachtet werden.

Um eine Überhitzung der Halbleiterelemente im Leistungsteil zu verhindern, geben die Hersteller in der Regel neben den Nennspannungen einen Bemessungsausgangsstrom und Maximalstrom an.

Der Bemessungsstrom ist der Effektivwert bei Drehstromantrieben, den der Regler dauerhaft abgeben kann. Der Maximalstrom kann vom Regler kurzzeitig abgegeben werden. Oftmals wird dieser durch einen Überlastfaktor angegeben, der zwischen 1,5 und 3 bei Servoreglern liegt.

Der Strombedarf des Motors ergibt sich aus dem Drehmomentverlauf an der Motorwelle. Bei Synchronmotoren kann der Strom mit

$$I = \frac{M}{k_T} \quad \text{mit } k_T \text{ Drehmomentkonstante} \quad (IX)$$

ermittelt werden. Die Drehmomentkonstante kann dem Datenblatt des Motors entnommen werden.

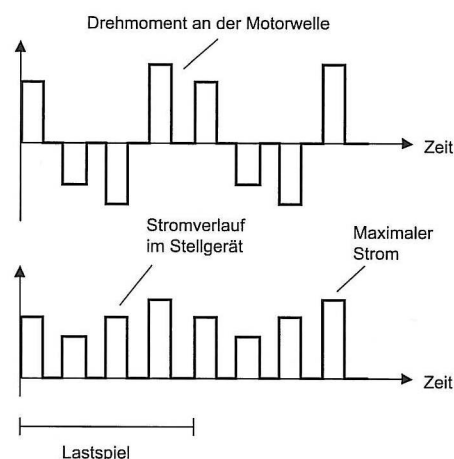


Abbildung 44 - Ermittlung des Stromverlaufs im Leistungsteil aus dem Drehmomentverlauf an der Motorwelle /5/

Der Effektivwert ergibt sich nun aus dem quadratischen Mittelwert des Stromverlaufes.

Für die Auslegung bei Servoantrieben sollte der Bemessungsstrom etwas größer sein als der Effektivstrom und der zulässige Maximalstrom größer als der auftretende Maximalstrom.

Bei Servoreglern, die als Kompletgerät ausgeführt sind, ist eine Auslegung der netzseitigen Einspeisung nicht erforderlich. Das Gerät verfügt dabei über einen Zwischenkreis und einen Gleichrichter, die dem Leistungsteil angepasst sind. Zusätzlich kann der Einsatz von Transformatoren erforderlich sein, wenn das Netz dem Gleichrichter angepasst werden muss. Dies ist z.B. beim Export in die USA der Fall.

In dem Fall, dass mehrere Achsen betrieben werden, macht es Sinn, in einem zusätzlichen Projektierungsschritt eine zentrale Einspeisung auszulegen. Dadurch werden mehrere Achsen von einem Zwischenkreis versorgt. Die Auslegung der Einspeisung erfolgt dann nach der Summenleistungsaufnahme aller angeschlossenen Motoren.

Um die Energie bei Bremsvorgängen aufzunehmen, ist ein Bremswiderstand zu dimensionieren. Dieser wandelt dann die überschüssige Energie im Zwischenkreis in Wärme um. Der Bremswiderstand wird mittels eines Bremschoppers beim Überschreiten einer vorgegeben Zwischenkreisspannung zugeschaltet.

Die Auslegung erfolgt dabei nach der Bemessungsleistung und Maximalleistung des Bremswiderstands. Es muss hierzu die tatsächlich auftretende mittlere Bremsleistung und Maximalleistung bestimmt werden.

Nach der lasttechnischen Auslegung des Servoreglers erfolgt zusätzlich eine Auslegung der Signalelektronik. Der Servoregler kann dabei oft durch Aufstecken von Modulen an einem Feldbus angeschlossen werden. Durch eine Technologiewahl können Positionierungen ausgeführt werden. Der Regler kann mit Sicherheitseinrichtungen für bestimmte Anwendungen erweitert werden.

6. Software

Zur Programmierung und Konfiguration des Profibusses und die zur Steuerung dienende SPS wird SIMATIC Step7 von der Firma Siemens verwendet. Hierbei handelt es sich um ein Softwarepaket für Automatisierungsprojekte.

Die Konfiguration und Programmierung, sowie Diagnose des Servoantriebes erfolgt mit Global Drive Control von der Firma Lenze. Es handelt sich hierbei um eine Entwicklungsumgebung für die Produktreihen Global Drive 8200 und 9300 von Lenze.

6.1. SIMATIC Step7

Bei der von Siemens erstellten SIMATIC Step7 Software handelt es sich um eine Entwicklungsumgebung, die der Norm IEC 61131-3 entspricht. Diese Norm dient zur weltweiten Standardisierung von speicherprogrammierbaren Steuerungen. Sie wurde als europäische und deutsche Norm DIN EN 61131 übernommen.

Mit Hilfe der Software lassen sich Hardwarekonfigurationen und Programmieraufgaben von Speicherprogrammierbaren Steuerungen leicht ausführen. Das zentrale Werkzeug bildet dabei der SIMATIC-Manager. Er verwaltet das Projekt.



Abbildung 45 - SIMATIC-Manager

Die SPS bedarf hierbei keinerlei mechanischer Schalter oder Stellschrauben mehr, die zur Konfiguration betätigt werden müssen. Alle Einstellungen an der Hardware können mittels der SIMATIC Step7 Hardwarekonfigurations-Umgebung vorgenommen werden.

Neben einer SPS ist es möglich, mit diesem Tool auch Bussysteme und deren Teilnehmer zu konfigurieren.

Für eine einfache und strukturierte Programmierung verfügt SIMATIC Step7 über eine Programmierumgebung. Mit Hilfe dieser Umgebung lassen sich übersichtlich die Programmbausteine in unterschiedlichen Programmiersprachen erstellen. Dem Anwender wird eine umfangreiche

Bibliothek zur Verfügung gestellt. Unterstützt werden die Sprachen KOP (Kontaktplan), FUP (Funktionsplan) und AWL (Anweisungsliste).

Zur Programmierung von komplexeren Aufgaben oder Datenverarbeitung dient S7-SCL. Diese Sprache entspricht der in der Norm IEC 61131-3 definierten Hochsprache ST (Structured Text).

6.2. Lenze Global Drive Control

Das Engineeringtool Global Drive Control von der Firma Lenze dient zur Inbetriebnahme, Bedienung, Parametrierung und Diagnose von Lenze Antriebsystemen der Baureihen Global Drive 8200 und 9300. Der Anwender erhält hiermit den Zugriff auf alle Parameter zum Programmieren.

Die Einstellungen eines Reglers lassen sich mit dieser Software übersichtlich darstellen. Im Onlinemodus ist die Software direkt mit dem Regler verbunden. Die Kommunikation erfolgt hierbei über eine LECOM (Lenze Kommunikation), eine Systembus-CAN Schnittstelle oder einen OPC-Busserver (Objekt Linking and Embedding for Process Control), über den der PC, bei Bedarf über einen Adapter mit dem Regler verbunden wird. Zusätzlich lassen sich die Einstellungen auch im Offlinemodus, bei dem keine Verbindung zum Regler besteht, parametrieren. Es ist möglich, z.B. zum Zwecke der Sicherung oder Vervielfältigung von Einstellungen, diese auf dem PC zu speichern.

Durch die mitgelieferte Oszilloskopfunktion steht dem Anwender zusätzlich eine umfangreiche Diagnosefunktion für 9300 Servo-Regler zur Verfügung.

Mit Hilfe des Funktionsblockeditors lassen sich die umfangreichen Funktionen der Regler auf eine einfache Art und Weise miteinander verknüpfen.

7. Hardware

Dieses Kapitel dient dazu, den Aufbau und die Zusammensetzung der einzelnen Baugruppen, die in dieser Diplomarbeit verwendet werden, zu erläutern.

7.1. CPU 314C-2DP

Die CPU 314C-2DP ist eine central processing unit. Hergestellt wird diese CPU von der Firma Siemens. Sie gehört zur Produktreihe SIMATIC.

Die CPU verfügt über eine integrierte Profibus-DP Schnittstelle. Sie kann hiermit als Master oder als Slave in einem Profibus-System konfiguriert werden.



Abbildung 46 - CPU 314C-2DP /13/

Zusätzlich verfügt die CPU über integrierte Ein- und Ausgänge. Dabei handelt es sich sowohl um analoge als auch um digitale Ein- und Ausgänge.

Der Leistungsumfang der CPU enthält zusätzliche Eigenschaften wie z.B.:

- 96 KByte Arbeitsspeicher, Ladespeicher MicoMemoryCard (MMC) steckbar 64KByte bis 8 MByte
- 256 Zähler
- 256 Zeiten
- 256 Merkerbytes
- 24 DI davon 16 für integrierte Funktionen nutzbar, alle auch als Alarmeingänge nutzbar
- 16 DO integriert, davon 4 schnelle Ausgänge
- 4 AI Strom/Spannung

- 2 AO Strom/Spannung
- 4 Impulsausgänge (2,5 kHz)
- 4-kanalig zählen und messen mit Inkrementalgebern 24 V (60Hz)
- integrierte Positionierfunktion

7.2. MDSKRS036-13

Beim MDSKRS036-13 handelt es sich um eine Drehstrom-Synchronmaschine. Ausgeführt ist diese als selbstkühlender Servo-Motor mit quadratischem Gehäuse und Kühlrippen.

Bemessungsdaten:	n_N	4000min ⁻¹	M_0	0,70Nm	M_{max}	3,10Nm
	M_N	0,60Nm	P_N	0,25kW	I_0	0,90A
	I_N	0,90A	I_{max}	5,00A	$U_{N,AC}$	245V
	f_N	200Hz				

Der Motor wird von der Firma Lenze hergestellt. Er gehört zur Baureihe Synchron-Servomotoren MD_KS. Es handelt sich hierbei um Synchronmotoren mit einer hohen Dynamik, einem geringen Massenträgheitsmoment und kompakter Bauform. Mit dieser Baureihe werden Nenndrehmomente von 0,6 bis 17 Nm sowie Spitzendrehmomente bis 52 Nm erreicht. Je nach Wahl handelt es sich dabei um einen selbstkühlenden oder fremdbelüfteten Motor.



Abbildung 47 - Lenze Servo-Motor der Baureihe MD_KS /15/

Im Gehäuse des Motors sind zusätzlich ein Resolvergeber und ein Temperatursfühler integriert.

Der Motor verfügt über zwei Schnittstellen: eine Leistungs- und eine Sensorschnittstelle. Die Leistungsschnittstelle stellt die Stromzuführung für die Wicklungen dar. An die Sensorschnittstelle sind die Kontakte des Resolvergebers und des Temperatursfühlers geführt.

7.3. EVS9322-EP

Der EVS9322-EP ist ein Servoregler für eine Motorbemessungsleistung von 0,75KW.

IN:	3/PE AC	400/480V	2,5A	2,1 kVA
	50-60Hz			
OUT:	3/PE AC	0-480V	I_N 2,5A	I_{max} 3,75A 60s
	2,1kVA	0-480Hz		

Für Positionieraufgaben verfügt er über ein Technologiepaket. Er gehört zur Familie der Servo-Umrichter 9300 von der Firma Lenze. Die Umrichter dieser Baureihe verfügen über eine intelligente Regelung mit unterschiedlich vorkonfigurierten Antriebsfunktionen. Der Leistungsbereich der Baureihe reicht von 0,37 bis 75 KW.



Abbildung 48 - Lenze Servo-Umrichter 9300 /16/

Als Standard verfügen dabei alle Varianten über eine einfache Drehzahl-, Drehmomentregelung und Gleichlauf als Leitantrieb oder Folgeantrieb. Um unterschiedlichen Anforderungen gerecht zu werden, verfügen die unterschiedlichen Typen jeweils über erweiterte Funktionen. Der Standard Typ ist der Servo-Umrichter. Er verfügt über die Standardfunktionen.

Der Servo-Positionierregler verfügt zusätzlich über eine komplette Positioniersteuerung. Verfahrenswege können dabei aus bis zu 32 Verfahrätzen gebildet werden.

Unabhängig von ihrem Funktionsumfang, verfügen die Servo-Umrichter 9300 über ein umfangreiches Angebot an Schnittstellen:

- Analoge Ein- und Ausgänge
- Digitale Ein- und Ausgänge
- Bis zu zwei Rückführsignale gleichzeitig

- Ein und Ausgangssignale für Inkrementalgeber
- Integrierte CAN-Bus Schnittstelle
- Steckplatz für Kommunikationsbaugruppe

Durch die intelligenten Antriebe können kleinere Teilprozesse einer Maschine mit Hilfe von Funktionsblockverknüpfungen gesteuert werden. Die Ein- und Ausgänge können dabei entsprechend frei programmiert werden.

7.4. EMF2133IB

Das EMF2133IB ist ein Kommunikationsmodul von der Firma Lenze. Mit Hilfe dieses Moduls lassen sich Antriebe wie z.B. ein Servo-Umrichter 9300 erweitern und in ein Profibus-System integrieren.



Abbildung 49 - Lenze EMF2133IB Profibusmodul /17/

Die Antriebe werden als Slave-Geräte in einem Profibus-System integriert. Dabei kann über die frontseitigen DIP-Schalter die Busadresse des Moduls eingestellt werden. Das Modul verfügt über eine automatische Erkennung der Busgeschwindigkeit. Es können in Abhängigkeit vom Grundgerät, auf welches das Modul montiert wird, bis zu 16 Prozessdatenwörter übertragen werden.

Durch den Anschluss einer externen Spannungsversorgung kann das Modul unabhängig vom Einschaltzustand des Grundgerätes am Profibus angesprochen werden.

Bei der Verwendung des EMF2133IB in Kombination mit DRIVECOM-Profil und einer Siemens SPS ist zu beachten, dass die Bytes des Steuer- und Statuswortes in umgekehrter Reihenfolge ausgetauscht werden. Dies bedeutet, dass das LOW-Byte des Kommunikationsmoduls dem HIGH-Byte des SPS Speichers entspricht und das LOW-Byte der SPS zum HIGH-Byte des Moduls gehört.

Ähnliches gilt für einen Wert des Prozessdatenkanals, dessen Datenformat 4 Bytes lang ist. Hier entspricht das LOW-Word des Moduls dem HIGH-Word der SPS.

Das Modul wird als Slave von einem Master am Bus konfiguriert. Dabei gibt der Master vor, welches Profil zum Steuern des Antriebes verwendet wird. Wird z.B. das DRIVECOM Profil 20 verwendet, so werden die Signale des Antriebes entsprechend der Zustandmaschine des DRIVECOM Profils aufbereitet. Umgekehrt werden die Signale der Steuerung für den Antrieb aufbereitet. Der Antrieb selbst erhält keine Informationen über das verwendete Steuerprofil.

8. Umsetzung

Dieses Kapitel soll die Entwicklung eines Positionierprogramms für profibusvernetzte Servoantriebe darstellen. Als Beispiel wird ein Folienvorzug in einer Verpackungsmaschine verwendet.

Die Umsetzung erfolgt mit einem Laboraufbau. Dieser besteht aus einer SPS 314C-2DP, einem EVS9322-EP und einem MDSKRS036-13. Auf ein Getriebe, Vorzugswelle und Folie wird aus Platzgründen verzichtet. Zunächst wird ein Profibus-System mit einem Master und einem Slave erstellt. Bei dem Master handelt es sich um eine SPS, die später zur Steuerung des Servoantriebs dienen soll. Der Servoantrieb bildet im Profibus-System den Slave. Danach wird der Servoantrieb konfiguriert und die Positionierung im Antrieb programmiert. Zum Abschluss erfolgt die Steuerung mit der SPS.

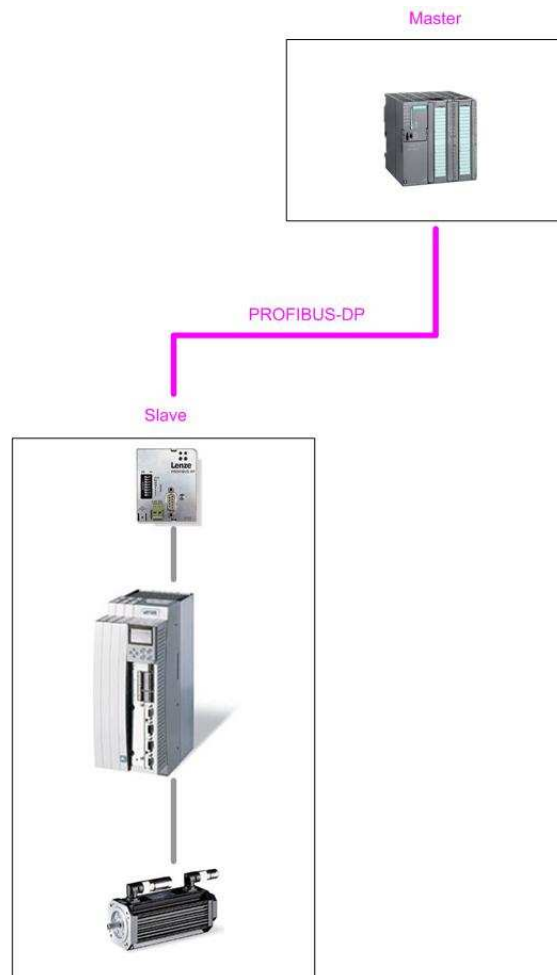


Abbildung 50 - Profibus vernetzter Servoantrieb und SPS

8.1. Folienvorzug

Bei einer Verschlussmaschine befinden sich zwischen einem Ober- und einem Unterwerkzeug Schalen. Diese Schalen sind mit Produkt gefüllt. Um diese Schalen nun luftdicht zu verschließen, sollen sie mit einer Folie versiegelt

werden. Die Folie befindet sich hierzu zwischen dem Oberwerkzeug und den Schalen. Wenn nun das Unterwerkzeug hochfährt, werden die Schalen darin eingeschlossen. Das Werkzeug ist so konzipiert, dass beim Zusammenfahren der obere Rand der Schalen gegen das Oberwerkzeug gedrückt wird. Folie und Schale werden dabei gegen eine heiße Siegelplatte gedrückt und die Folie wird mit der Schale verschweißt. In einem weiteren Schritt werden die Schalen mit der Folie im Werkzeug aus der Folienfläche ausgestanzt.

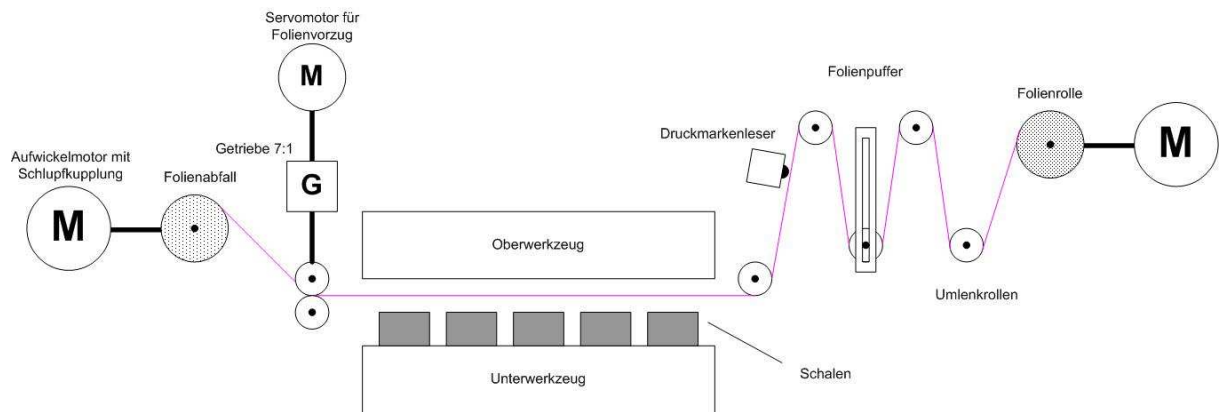


Abbildung 51 - Folienvorzug einer Verpackungsmaschine

Die Aufgabe des Folienvorzuges ist es nun, bei jedem Zyklus die Folie vorzuziehen. Dabei wird der Folienabschnitt um eine Werkzeuglänge vorgezogen, sodass zwischen Schalenoberkante und Oberwerkzeug wieder eine geschlossene Folienfläche vorhanden ist. Bei einer bedruckten Folie ist dabei entscheidend, dass später der Aufdruck der Folie zentriert auf der Schale zu finden ist. Dazu verfügen die Folien über so genannte Druckmarken. Diese befinden sich in regelmäßigen Abständen auf der Folie und können mit einem Druckmarkenleser erkannt werden. Sie kennzeichnen damit einen Foliendruck, der sich immer wiederholt (Rapport).

Eine Folie die nicht bedruckt ist, darf hingegen nur soweit vorgezogen werden, dass ein möglichst geringer Folienabfall entsteht.

Die Aufgabe der Positionierung ist nun unterschiedlich und abhängig davon, ob die Folie bedruckt ist oder nicht.

Wenn die Folie bedruckt ist, muss der Folienvorzug bis zur Druckmarke erfolgen. In diesem Fall wird auf ein externes Signal vom Druckmarkenleser reagiert.

Bei einer unbedruckten Folie entspricht der minimale Folienvorzug der Länge des Werkzeuges. Die Strecke ist bereits vorher bekannt und kann somit als Lagezielwert vorgegeben werden.

8.2. Profibus

Für die Kommunikation mit dem Servoantrieb wird der Profibus als Feldbus verwendet. Der Profibus besteht aus einer SPS CPU-314C-2DP, die als einziger Master am Bus betrieben wird, und einem Profibus Modulen EMF2133IB. Das Modul wird auf dem Servoregler EVS9322-EP montiert und ermöglicht somit der SPS die Kommunikation mit dem Servoregler über den Profibus.

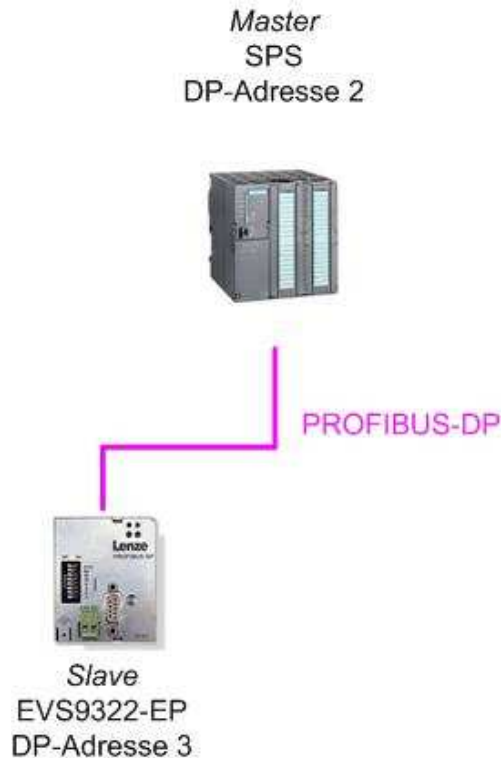


Abbildung 52 - Aufbau Profibus-DP Netz

Als Busübertragungsmedium wird eine geschirmte und verdrehte 2-Drahtleitung mit 9-poligem D-Sub Stecker verwendet.

Beim Aufbau und Verdrahten ist darauf zu achten, dass die Abschlusswiderstände an den Bussenden aktiviert sind und die Schirmung richtig angeschlossen ist. Dies sind die häufigsten Fehler, die bei der Verdrahtung auftreten können.

Projekt erstellen

Nachdem der Aufbau abgeschlossen ist, wird mit dem SIMATIC-Manager ein neues Projekt erstellt. Dieses neue Projekt verfügt als Standard über einen MPI-Bus. Multi Point Interface ist ein Bussystem von der Firma Siemens, welches zur Konfiguration und Datenaustausch von Baugruppen der SIMATIC-S7 Reihe verwendet wird.

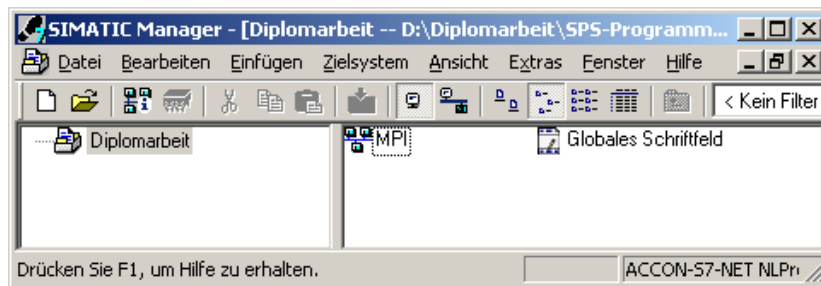


Abbildung 53 - Übersicht neues Projekt

Konfiguration

Über das Menü „Einfügen/Subnetz/Profibus“ kann dem System nun ein Profibus hinzugefügt und benannt werden. Der Profibus kann nun mit NetPro konfiguriert werden. NetPro ist ein Teil von Step7.

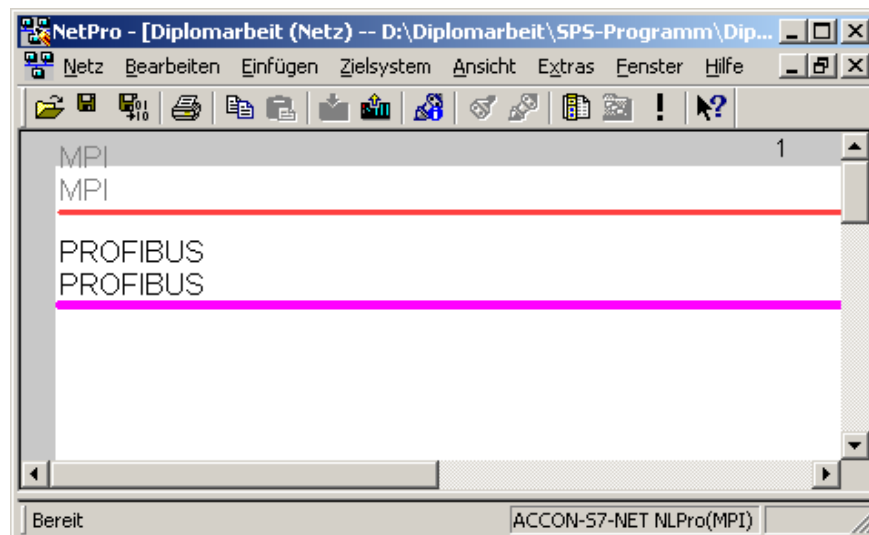


Abbildung 54 – NetPro

Mit NetPro werden die Objekteigenschaften des Profibus-Systems bearbeitet. Hier werden die Busgeschwindigkeit, maximale Anzahl der Busteilnehmer und ein Busprofil ausgewählt.

Die Angabe der maximalen Busteilnehmer dient zur Optimierung der Buszugriffssteuerung in Multi-Master-Systemen.

Für die Busgeschwindigkeit wird 12MBit/sec gewählt, da die Gesamtlänge des Netzes mit ca. 2 Metern deutlich unter der maximalen Länge für diese Geschwindigkeit liegt. Wenn die Länge des Netzes bekannt ist, kann diese zur Berechnung der Busparameter im Feld Optionen angegeben werden. Wenn die Länge nicht bekannt ist, wird die maximal Länge für die gewählte Geschwindigkeit in der Berechnung der Busparameter berücksichtigt.

Zur automatischen Berechnung der Busparameter werden unterschiedliche Algorithmen angewendet. Durch die Wahl des Busprofils werden diese ausgewählt.

Bei einer Wahl von DP werden zur Berechnung der Busparameter nur die Teilnehmer berücksichtigt, die innerhalb des Projektes bekannt sind und am Bus angeschlossen sind.

Mit der Wahl von Standard können bei der Berechnung der Busparameter weitere Teilnehmer berücksichtigt werden, die dem Projekt noch nicht bekannt sind, aber am Bus angeschlossen sind. Dies kann z.B. bei Multi-Master-Systemen der Fall sein. Die Anzahl der Teilnehmer kann hier über Optionen eingestellt werden.

Durch die Wahl von Benutzerdefiniert können die Busparameter manuell eingestellt werden.

Bei einer Wahl der Übertragungsgeschwindigkeit bis 1,5 MBit/sec kann als weiteres Profil Universell (DP/FMS) eingestellt werden. Hiermit können auch ältere SIMATIC-S5 Komponenten am Netz betrieben werden.

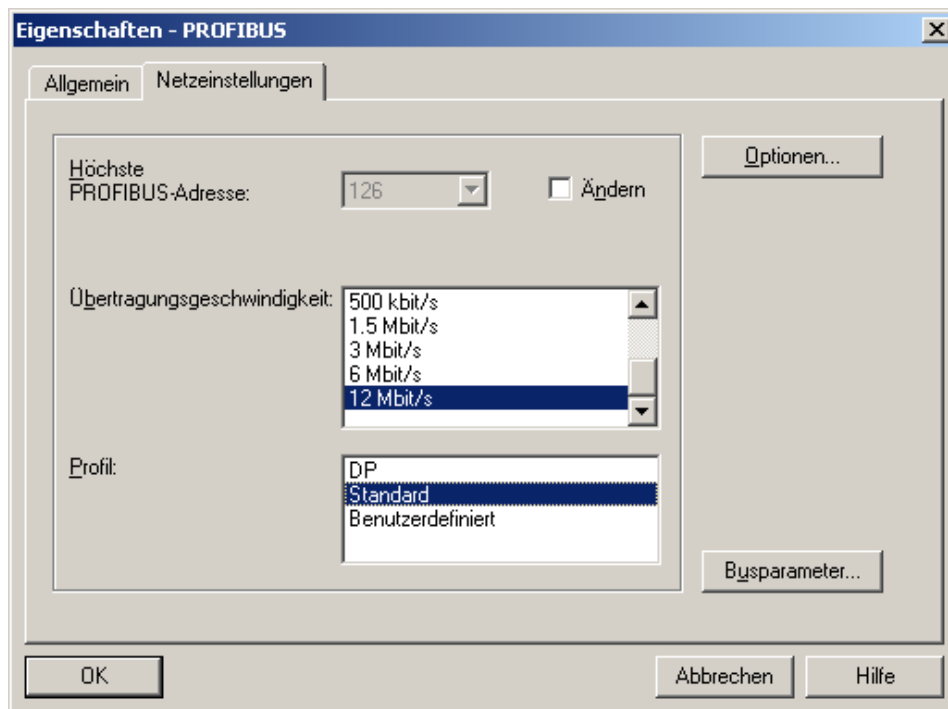


Abbildung 55 - Profibus Objekteigenschaften

Mit Optionen kann, wie erwähnt, die Länge des Netzes sowie die Anzahl der eingesetzten Repeater eingestellt werden. Zusätzlich können auch hier Einstellungen für Lichtwellenleiter als Übertragungsmedium vorgenommen werden. Als Angabe für die Gesamtlänge des Netzes werden hier 3m eingestellt. Das Netz verfügt über einen weiteren Slave-Teilnehmer, der nicht weiter projiziert wird.

Weiter kann, sobald der Bus über einen Master verfügt, unter Optionen die sog. Äquidistanz parametrisiert werden. Durch das Aktivieren des äquidistanten Betriebes des Netzes wird eine konstante Buszykluszeit eingestellt.

Bei Aktivierung dieses Betriebes ist es nicht zu empfehlen, die errechnete und vorgeschlagene Zeit zu verringern, da es hierdurch zu Kommunikationsfehlern kommen kann, wie z.B. bei einem azyklischen Datenaustausch, der dann nicht abgeschlossen werden könnte. Dies kann zu einem zeitlichen Überlauf führen. Der Buszyklus ist dabei nicht mehr konstant.

Weiter ist zu beachten, dass dieser Betrieb nur in Single-Master-Netzen möglich ist.

In diesem Projekt wird diese Betriebsart nicht aktiviert.

Master

Die Einstellungen des Profibus-Systems sind jetzt soweit abgeschlossen, dass im nächsten Schritt die Teilnehmer hinzugefügt werden können. Dazu wird im SIMATIC-Manager zunächst dem Projekt eine SIMATIC-300 Station hinzugefügt.

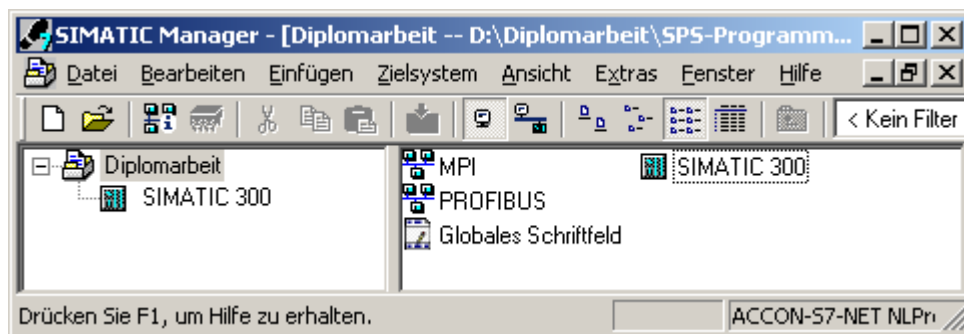


Abbildung 56 - Projektübersicht Diplomarbeit

Durch das Öffnen der SIMATIC 300 wird nun deren Hardwarekonfiguration möglich. Hierzu öffnet sich HW Konfig. Es handelt sich hierbei um ein weiteres Tool von Step7.

HW Konfig verfügt über einen sog. Hardware Katalog. Dieser enthält Informationen über unterschiedliche Hardwareelemente. Durch die Auswahl der entsprechenden Elemente erfolgt mit Hilfe der Software der Aufbau und die Konfiguration der Baugruppen und Busteilnehmer.

Der Hardware Katalog kann durch die Installation von GSD-Dateien erweitert werden. Hierdurch erhält HW Konfig Informationen über das Lenze EMF2133IB Profibus-Modul.

Zunächst wird dem Projekt ein Rack hinzugefügt. Auf diesem Rack kann die CPU 314-2DP aufsteckt werden. In der HW Konfig werden nun die Komponenten der CPU und Adressen dargestellt.

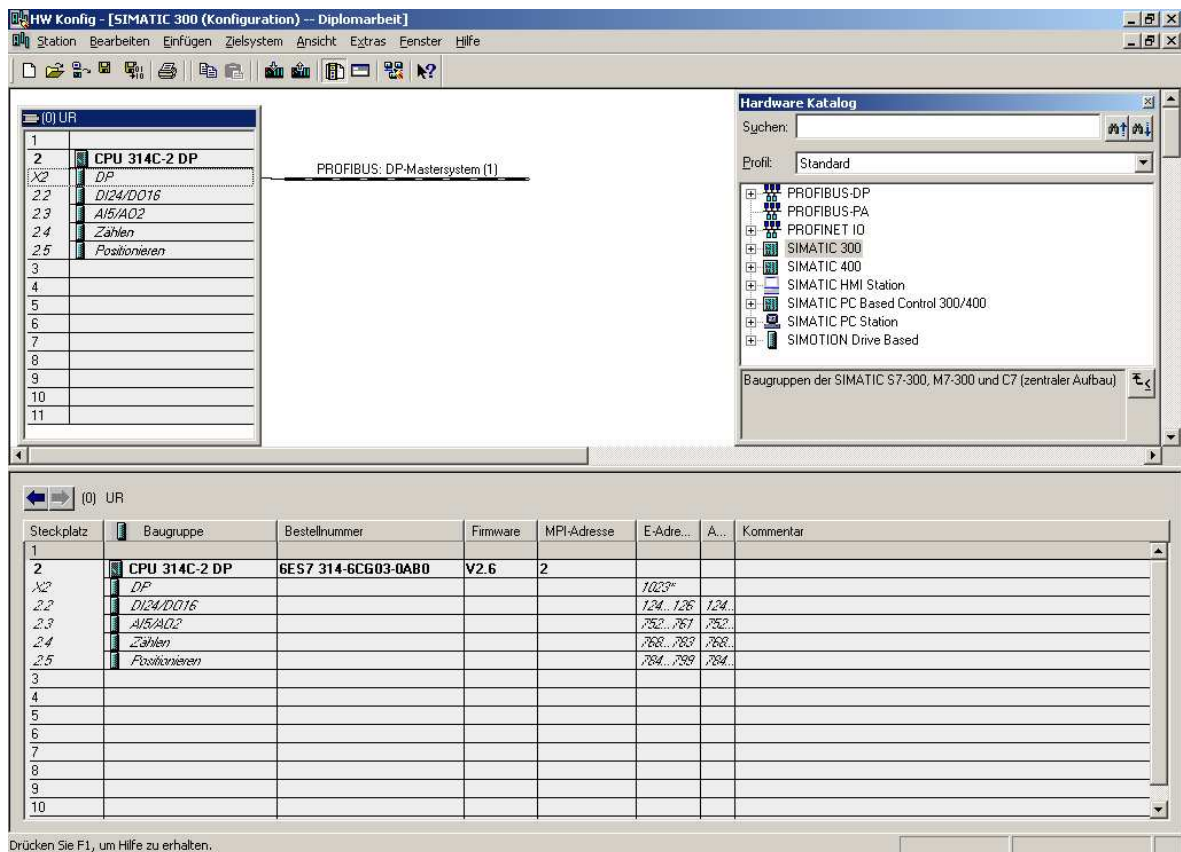


Abbildung 57 - HW-Konfig

Über die Objekteigenschaften der DP-Komponente kann die CPU an den Profibus angeschlossen und als Master Klasse 1 parametrierbar werden. Dem Master wird die Profibus-Adresse 2 zugewiesen.

Slave

Nachdem die SPS mit dem Profibus als Master verbunden ist, können weitere Slave-Geräte an den Bus angeschlossen werden.

Um das Lenze EMF2133IB Profibus-Modul hinzuzufügen, wird dieses im Hardware Katalog ausgewählt und per Drag&Drop an den Bus angehängt.

Beim Hinzufügen des Moduls wird diesem die Adresse des DP-Slaves 3 eingestellt. Hierbei ist es wichtig, dass im HW-Konfig die ausgewählte Adresse mit der am Modul eingestellten Adresse übereinstimmt. Weiter ist zu beachten, dass in einem Profibus-Netz jede Adresse nur einmal vergeben werden darf.

Nachdem diese Bedingungen berücksichtigt sind, muss weiter ausgewählt werden, in welchem Umfang der Datenaustausch durchgeführt werden soll. In diesem Beispiel wird aufgrund des Servoreglers EVS9322-EP, der über 4 Prozessdatenwörter verfügt, PAR(Kons.)+PZD (4W Kons.) gewählt. Dabei handelt es sich um einen sog. Parameterkanal bei dem 4 Datenwörter

ausgetauscht werden. Der Prozessdatenkanal PZD enthält ebenfalls 4 Datenwörter.

Zusätzlich wird hierdurch ausgewählt, dass die Steuerung mit Hilfe des DRIVECOM Profils 20 erfolgt.

Der Datenzugriff erfolgt konsistent.

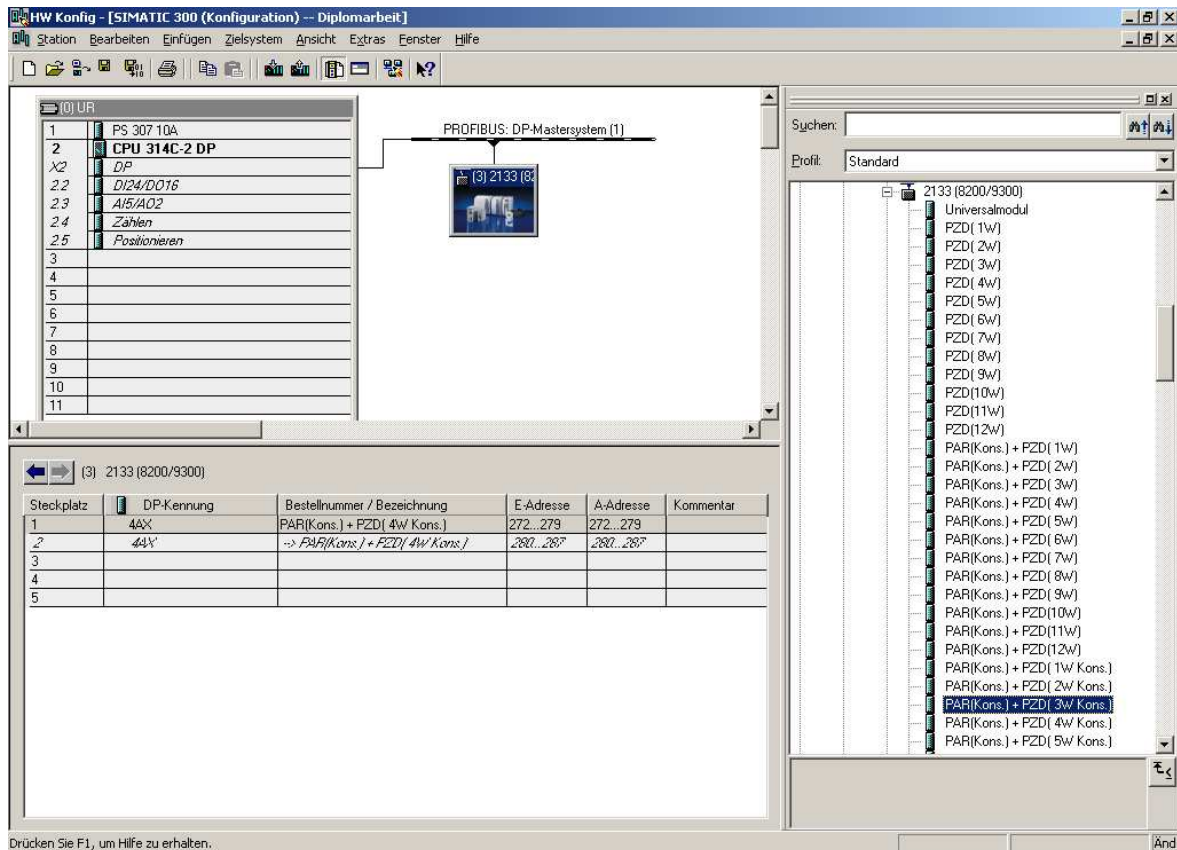


Abbildung 58 - DP-Slave Konfiguration

Busparameter

Nachdem die Konfiguration und Aufbau der Hardware abgeschlossen ist, können nun die errechneten Busparameter in den Profibus-Eigenschaften abgelesen werden.

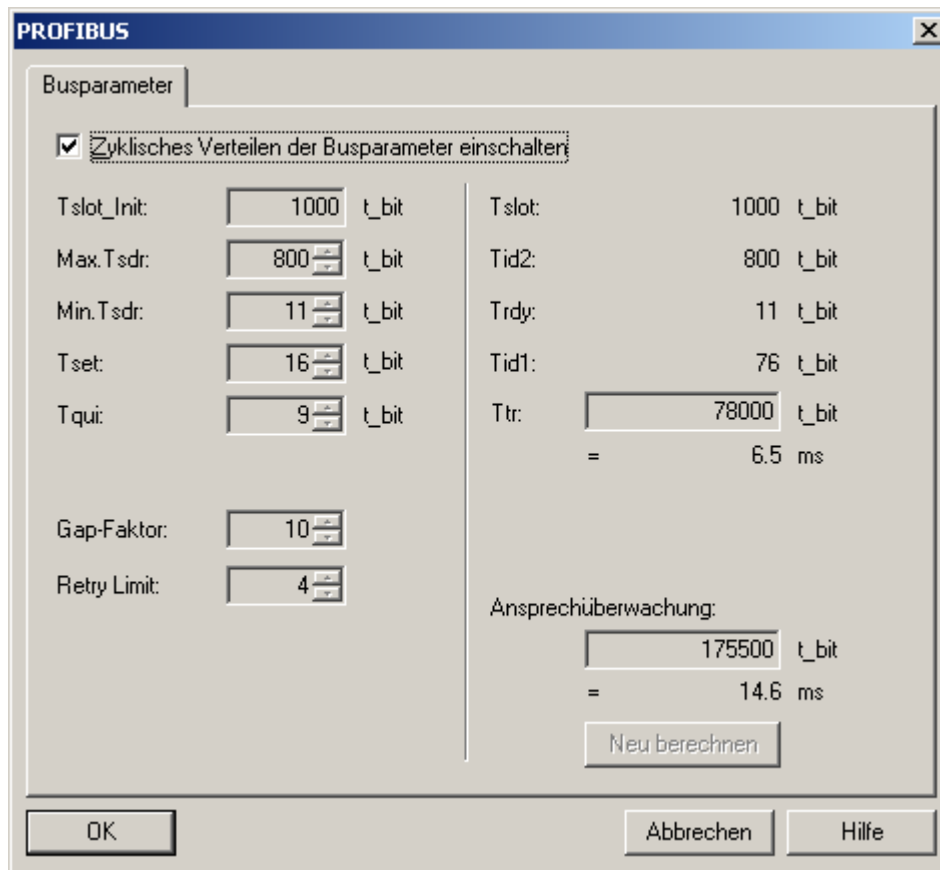


Abbildung 59 - Busparameter

Im Projekt wird die Token-Umlaufzeit mit 6,5ms berechnet.

Mit dem Aktivieren des zyklischen Verteilens der Busparameter über den Bus wird es möglich, sich mit einem Diagnose- oder Programmiergerät an den Bus anzuschließen. Dabei ist es nicht notwendig, die Busparameter zu kennen, da sie regelmäßig durch den Master an alle am Bus befindlichen Teilnehmer gesendet werden.

8.3. Servoantrieb

Der Folienvorzug wird mit einem positioniergesteuerten Servoantrieb durchgeführt.

Als Art der Positionierung wird die relative Positionierung gewählt, da es sich hierbei um einen endlosen Prozess handelt, abgesehen von dem Fall, dass die Folienvorzug leer ist. Dies wird bei einem Folienvorzug mit Hilfe der Folienvorzug erkannt.

Die Auslegung des Antriebssystems wird bei diesem Beispiel nicht durchgeführt, da er bereits durch den Hersteller der Verpackungsmaschine vorgegeben ist und bei der Konstruktion bestimmt wurde. Es handelt sich hierbei um den Synchronservomotor MDSKSRS036-13 mit dem Servoregler EVS9322-EP. Um den Regler am Profibus zu betreiben, wird er mit dem aufsteckbaren Kommunikationsmodul EMF2133IB erweitert.

Installation

Die elektrische Installation erfolgt nach folgendem Schema:

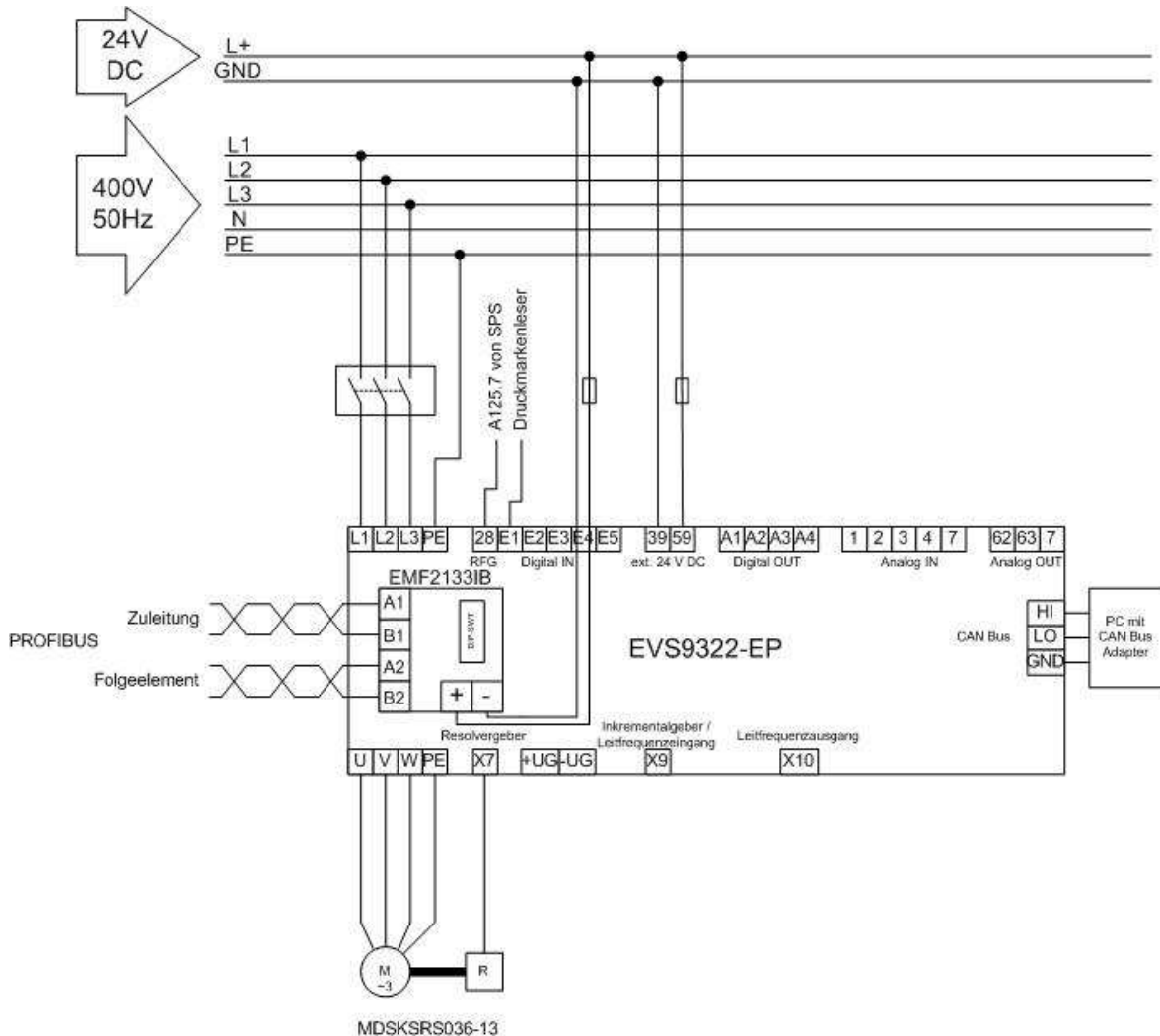


Abbildung 60 - Installation Servoantrieb

Der Anschluss der Motor- und Resolverleitung erfolgt durch spezielle Herstellerleitungen. Diese verfügen neben der erforderlichen Aderanzahl und dem Querschnitt über eine Schirmung. Sie dienen für eine EMV-gerechte (elektromagnetische Verträglichkeit) Verdrahtung.

Die „Klemme 28“ dient für die Reglersperre. Um die Reglersperre zu deaktivieren und damit die Stromversorgung des Leistungsteils freizugeben, muss diese Klemme auf „HIGH“ gesteuert werden. In der Praxis kann hiermit eine Sicherheitseinrichtung überwacht werden, die zu einer automatischen Reglersperre führt, wenn die Sicherheit nicht gegeben ist.

In diesem Beispiel wird die „Klemme 28“ beim Laboraufbau direkt von der SPS gesteuert. Um die Reglerfreigabe einzuschalten, muss der Ausgang A125.7 der SPS auf „HIGH“ gesteuert werden.

Der Schaltimpuls des Druckmarkenlesers wird am digitalen Eingang E1 des Reglers angeschlossen.

Mit Hilfe der externen 24V DC Versorgung kann die Signalelektronik auch ohne Stromversorgung aus dem 400V 50Hz Netz ihre Überwachungsfunktionen ausführen.

Der Regler kann dabei weiter über den CAN-Bus vom PC erreicht werden. Auf dem PC wird mit Global Drive Control der Antrieb diagnostiziert und parametriert.

Das EMF2133IB baut die Verbindung zum Profibus auf.

Inbetriebnahme

Die Inbetriebnahme des Servoantriebes erfolgt mit Global Drive Control.

Nach dem ersten Einschalten befindet sich der Regler in seinen Werkseinstellungen.

Im ersten Schritt wird nun die Grundkonfiguration des Reglers eingestellt. Dazu wird die Codestelle C0005 auf den Wert 20213 eingestellt. Die Grundkonfiguration dient als Grundlage für die weitere Parametrierung des Reglers.

Durch den Wert C0005=20213 arbeitet der Regler mit relativer Positionierung und wird über den Profibus gesteuert. Das Kommunikationsmodul wird mit einer externen 24V DC Spannungsquelle versorgt.

Der nächste Schritt passt den Regler an das Versorgungsnetz an. Hierzu wird in der Codestelle C0173/0 der Wert 1 gewählt.

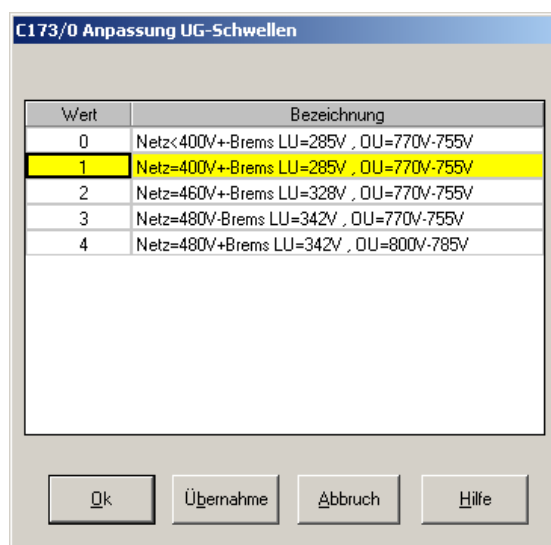


Abbildung 61 - C173/0

Der Regler ist nun so eingestellt, dass er an einem 400V 50/60Hz Netz mit und ohne Bremswiderstand betrieben werden kann. Die Werte LU=285V und OU=770-755V geben an, dass der Antrieb eine Störung meldet sobald die

Zwischenkreisspannung die Werte unter- oder überschreitet. Wird die Zwischenkreisspannung des Antriebes während seiner Bewegung überschritten, ist der Einsatz eines Bremswiderstandes erforderlich.

Im nächsten Schritt wird die Motorseite des Reglers konfiguriert. Hierbei bietet Global Drive Control die Möglichkeit, einen Lenze Motor aus einer im Regler hinterlegten Datenbank auszuwählen. Dies wird mit der Codestelle C0086 durchgeführt. Der Motor MDSKSRS036-13 hat den Wert C0086=108.

Die Datenbank verfügt dabei über motorspezifische Werte für den Regler. Mit C86=108 werden folgende Werte parametrisiert:

C0006	Motortyp	Servoregelung Synchronmotor in Sternschaltung
C0011	Maximaldrehzahl (Bezugsgröße)	3000rpm
C0022	Maximalstrom (Drehmomentbegrenzung)	1,35A
C0070	P-Verstärkung Drehzahlregler	5 Hierbei handelt es sich um eine robuste Vorgabe. Für ein optimales Verhalten muss der Wert später an die Last angepasst werden.
C0071	Ti-Zeit Drehzahlregler	20ms Anpassung wie bei C0070
C0075	P-Verstärkung Stromregler	0,75
C0076	Ti-Zeit Stromregler	1,5ms
C0081	Bemessungsleistung	0,25kW
C0084	Ständerwiderstand	19,8 Ohm
C0085	Streuinduktivität	30,5 mH
C0087	Bemessungsdrehzahl	4000rpm
C0088	Bemessungsstrom	0,9A
C0089	Bemessungsfrequenz	200Hz
C0090	Bemessungsspannung	245V
C0091	cos phi	1

Die Temperaturüberwachung wird mit den Codestellen C0583, C0584 und C0594 beschrieben und durch die Codestelle C0086 eingestellt.

Weiter wird durch die Codestelle C0086 als Rückführsystem der fest im Motorgehäuse montierte Resolvergeber ausgewählt.

$$\text{Auflösung Lageistwert} = \frac{360^\circ}{2^{11} \text{Bit Wandlungsbreites A/D Wandlers am Resolvereingang}} = \frac{360^\circ}{2048 \text{ Schritte}} = 0,1758^\circ / \text{Schritt}$$

Wenn der Wert C0086 nicht bekannt ist oder es sich um keinen Lenze Motor handelt, können die Parameter direkt eingestellt werden. Die Werte können dann dem Motortypenschild, Motordatenblatt entnommen oder berechnet werden.

Zur Anpassung an die Maschine kann im nächsten Schritt eine physikalische Einheit definiert werden. Intern arbeitet der Regler mit „units“. Durch die Maschinenanpassung werden Millimeter als physikalische Einheit gewählt. Für die Berechnungen ist die Angabe des Getriebefaktors und der Vorzugskonstanten nötig.

C1202 Getriebe Zähler	7
C1203 Getriebe Nenner	1
C1204 Vorzugskonstante	Pi·57mm=179,0707813mm/Umdrehung

Die Vorzugskonstante gibt an, wie viele „units“, nun Millimeter, bei einer Umdrehung an der Getriebeabtriebsseite von der Folie zurückgelegt werden.

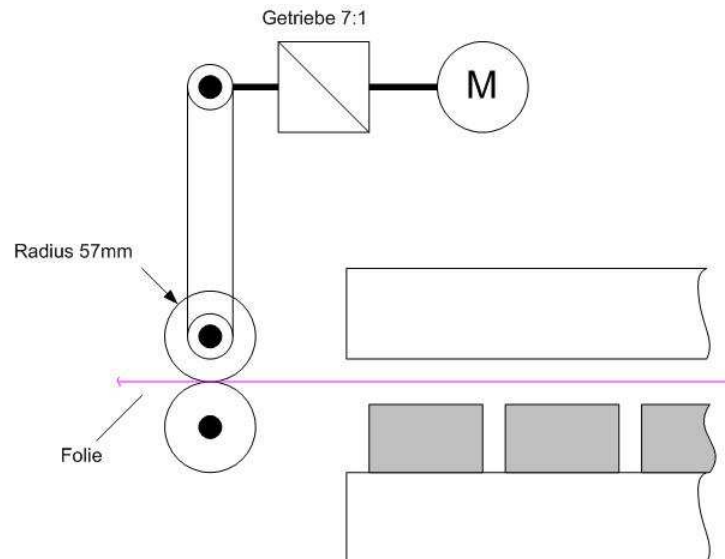


Abbildung 62 - Maschinenparameter

Es kann nun die maximale Folienvorzugsgeschwindigkeit berechnet werden. Sie ergibt sich aus der Maximaldrehzahl C0011, dem Getriebefaktor und der Vorzugskonstanten.

$$v_{\max} = \frac{n_{\max}}{i} \cdot k_{\text{Vorzug}} = \frac{3000}{7} \frac{1}{\text{min}} \cdot \frac{1 \text{min}}{60 \text{sek}} \cdot 179,0707813 \frac{\text{mm}}{\text{Umdrehung}} = 1279,077009 \frac{\text{mm}}{\text{sek}}$$

Als Nächstes erfolgt in der Codestelle C1240 die Vorgabe der maximalen Vorzugsgeschwindigkeit. Sie sollte gerade unterhalb der berechneten liegen. Gewählt wird $v_{\max}=1200\text{mm/sek}$.

Dieser Wert dient als Bezugsgröße für den Geschwindigkeitssollwert während der Positionierung.

Die Bezugsgröße für die maximal benötigte Beschleunigung/Verzögerung wird in der Codestelle C1250 angegeben. Um zu verhindern, dass bei einem Vorzug die Folie reißt, ist es das Ziel, die maximale Beschleunigung zu begrenzen ohne die minimale Folienvorzugszykluszeit zu verlängern.

Die maximale Verzögerung und die maximale Beschleunigung sind vom Betrag gleich.

Die maximale Beschleunigung berechnet sich aus der minimalen Zykluszeit, der gewählten maximalen Geschwindigkeit und der Strecke, die während eines Zykluses zurückgelegt werden muss.

Die Strecke ist durch die Konstruktion des Werkzeuges auf 1200mm festgelegt. Dieser Wert entspricht gleichzeitig der maximalen Distanz zwischen zwei Druckmarken.

Da sich die Folie, wenn sie zu lange unter dem heißen Oberwerkzeug liegt, verformt, wird eine möglichst kleine Vorzugszeit gewählt. Die Folie wird dann unmittelbar vor dem Zusammenfahren der Werkzeuge vorgezogen. Die minimale Vorzugszeit ergibt sich durch die Konstruktion der gesamten Maschine und der Abfolge der Einzelschritte während eines Maschinenzyklus. Sie liegt bei ca. 1,5sec.

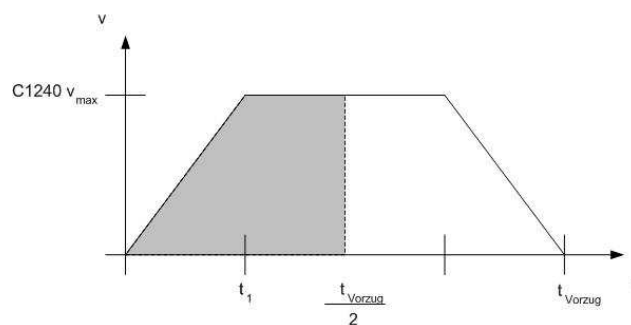


Abbildung 63 – Fahrprofil

$$\begin{aligned}\frac{s}{2} &= \frac{v_{\max} t_1}{2} + v_{\max} \left(\frac{t_{\text{Vorzug}}}{2} - t_1 \right) \\ s &= v_{\max} t_1 + 2v_{\max} \left(\frac{t_{\text{Vorzug}}}{2} - t_1 \right) \\ s &= v_{\max} t_1 + v_{\max} t_{\text{Vorzug}} - 2v_{\max} t_1 \\ s &= v_{\max} (t_{\text{Vorzug}} - t_1) \\ \rightarrow t_1 &= t_{\text{Vorzug}} - \frac{s}{v_{\max}}\end{aligned}$$

Es kann nun am Punkt $v(t_1)=v_{\max}$ die Beschleunigung bestimmt werden

mit $v(t) = at$

$$\rightarrow a = \frac{v(t)}{t}$$

$$a_{\max} = \frac{v(t_1)}{t_1} = \frac{v_{\max}}{t_{\text{Vorzug}} - \frac{s}{v_{\max}}}$$

$$\text{mit } v_{\max} = 1200 \frac{\text{mm}}{\text{sec}}$$

$$t_{\text{Vorzug}} = 1,5 \text{ sec}$$

$$s = 1200 \text{ mm}$$

$$\rightarrow a_{\max} = 2400 \frac{\text{mm}}{\text{sec}^2}$$

Die maximale Beschleunigung 2400 mm/sec^2 wird nun in der Codestelle C1250 eingetragen.

Die Vorzugszeit kann aufgrund der unterschiedlichen Reißfestigkeit der Folien variieren. Aus diesem Grund sind die Parameter Geschwindigkeit und Beschleunigung während der weiteren Programmierung als Variable zu berücksichtigen, die durch den Benutzer verändert werden können.

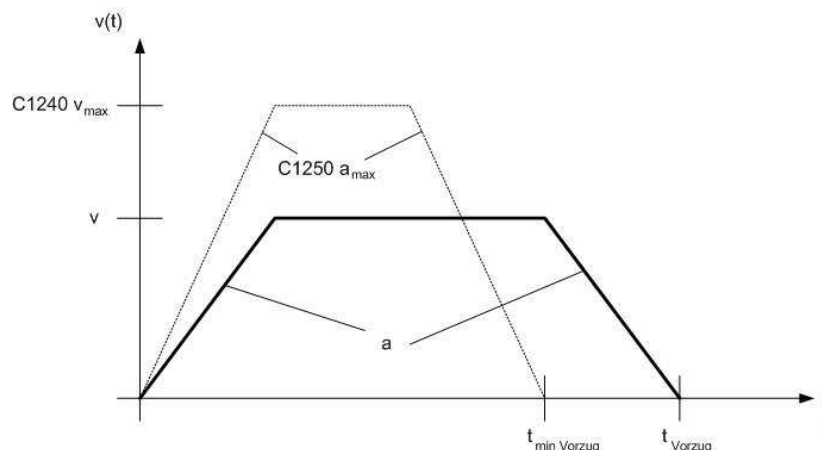


Abbildung 64 - variable Vorzugszeit

Eine große Vorzugszeit, kann die Leistung der Verpackungsmaschine beeinflussen.

Positionierprogramm

Zur Erstellung des Positionierprogramms wird zunächst der Fahrweg festgelegt. Dieser wird dann in einzelne Fahrsätze zerlegt.

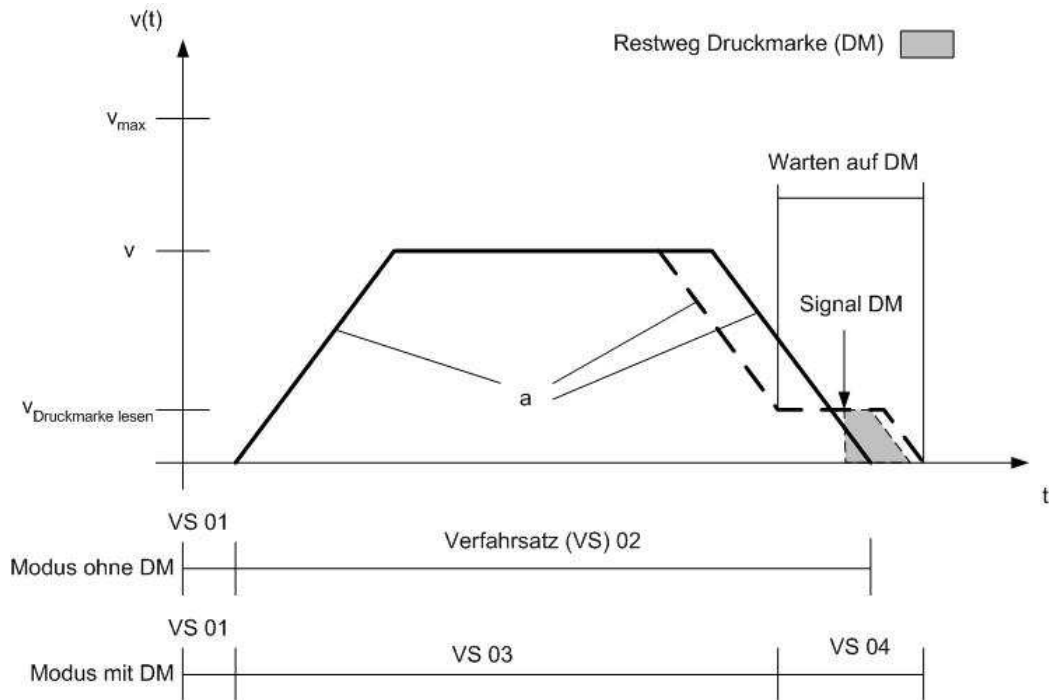


Abbildung 65 - Fahrwege für Folienvorzug

Beim Erstellen des Fahrweges wird berücksichtigt, ob die Maschine mit einer bedruckten Folie betrieben wird oder nicht.

Mit Hilfe des Dialogs Positionierung werden die Fahrsätze programmiert. In der Codestelle C1211 wird der Wert 1 eingetragen. Hiermit wird der Fahrsatz ausgewählt mit dem die Positioniersteuerung das Programm startet.

Verfahrensatz 01

Während der Abarbeitung des Verfahrensatzes durch die Positioniersteuerung im Servoregler wird die aktuelle Lage als Referenz gesetzt.

Dieser Verfahrensatz steuert den Modus. Wenn die SPS vorgibt, dass die Folie über eine Druckmarke verfügt und auf diese reagiert werden soll, wird dem Verfahrensatz 01 als Folgeverfahrensatz 03 vorgegeben. Dies ist der Fall, wenn das Signale PFI 01 „High“ ist.

In dem Fall, dass die Folie über keine Druckmarke verfügt, wird im nächsten Schritt der Verfahrensatz 02 aktiviert. PFI 01 ist „Low“.

Parameter	Value	Unit	Other
Warten	inaktiv	0-Pegel	
Schalten	inaktiv	0-Pegel	
PS-Mode	Referenz setzen		
Lage-Ziel	Real-Null		
Verfahrgeschw.	v-max		
Beschleunigung	a-max		
Verzögerung	a-max		
Endgeschw.	Stillstand		
TP-Fenster	Fahrbereich		wenn PS-CANCEL oder kein TP
TP-Restweg	Ziel = TP-Lage		Prg.-Ende
Schalten	inaktiv	0-Pegel	
Wartezeit	inaktiv		Sprung nach
Verzweigung 1	PFI 01	1-Pegel	PS 03
Verzweigung 2	inaktiv	0-Pegel	Prg.-Ende
Stückzahl	inaktiv		Prg.-Ende
Sprung	PS 02		

Abbildung 66 - Verfahrensatz 01

Verfahrsatz 02

Der Verfahrsatz 02 steuert den Modus ohne Druckmarke. Das Lageziel entspricht der Strecke. Die Stecke wird von der SPS vorgegeben. Der Parameter Lage-Ziel VTPOS-No 001 C1301/1, Verfahrgeschwindigkeit VTVEL-No 001 C1302/01, Beschleunigung/ Verzögerung VTACC-No 001 C1303/1 beschreiben den Verfahrweg des Verfahrsatzes und werden über den Profibus verändert.

Am Ende des Verfahrsatzes wird das Positionierprogramm beendet.

The screenshot shows a software interface for configuring a motion control program (PS 02). The interface is organized into several sections:

- Control Elements:** 'Warten' and 'Schalten' are both set to 'inaktiv' with a '0-Pegel' level. Navigation buttons include '<=', '<=', 'Gehe zu ...', '=>', and '=>|'.
- Mode and Target:** 'PS-Mode' is set to 'Relativer PS'. 'Lage-Ziel' is 'VTPOS-No 001' with a value of '0.0000 units'.
- Velocity and Acceleration:** 'Verfahrgeschw.' is 'VTVEL-No 01' at '10.00 %vmax'. 'Beschleunigung' and 'Verzögerung' are both 'VTACC-No 01' at '10.00 %amax'.
- End Conditions:** 'Endgeschw.' is 'Stillstand'. 'TP-Fenster' is 'Fahrbereich'. 'TP-Restweg' is 'Ziel = TP-Lage'.
- Branching and Jumping:** 'Wartezeit', 'Verzweigung 1', 'Verzweigung 2', 'Stückzahl', and 'Sprung' are all set to 'inaktiv'. 'Sprung' is set to 'Prg.-Ende'.
- Buttons:** A vertical stack of buttons on the right includes 'PS einfügen', 'PS entfernen', 'PS zurücksetzen', 'Dokumentation ...', 'Drucken ...', and 'Steuerung ...'. At the bottom right are 'Hilfe' and 'zurück'.

Abbildung 67 - Verfahrsatz 02

Verfahrensat 03

Für den Betrieb mit einer Druckmarke wird der Verfahrensat 03 von der Positioniersteuerung aktiviert. Dieser enthält die gleiche Verfahrensgeschwindigkeit, Beschleunigung und Verzögerung wie der Verfahrensat 02.

Das Lageziel wird in der Codestelle VTPOS-No 002 C1301/2 vorgegeben. Dabei ist es um 100mm geringer als VTPOS-No 001. Der Grund hierfür ist, dass in der Praxis die Distanz zwischen zwei Druckmarken eine Toleranz aufweist. Durch diese Programmierung wird berücksichtigt, dass die Distanz bis zu $\pm 50\text{mm}$ variieren kann.

Die Endgeschwindigkeit dieses Verfahrensatzes entspricht der Geschwindigkeit, mit der die Druckmarke gelesen wird.

PS 03	
Warten	inaktiv 0-Pegel
Schalten	inaktiv 0-Pegel
PS-Mode	Relativer PS
Lage-Ziel	VTPOS-No 061 1099.9999 units
Verfahrensgeschw.	VTVEL-No 01 30.00 %vmax
Beschleunigung	VTACC-No 01 100.00 %amax
Verzögerung	VTACC-No 01 100.00 %amax
Endgeschw.	VTVEL-No 02 20.00 %vmax
TP-Fenster	Fahrbereich wenn PS-CANCEL oder kein TP
TP-Restweg	Ziel = TP-Lage Prg.-Ende
Schalten	inaktiv 0-Pegel
Wartezeit	inaktiv Sprung nach
Verzweigung 1	inaktiv 0-Pegel Prg.-Ende
Verzweigung 2	inaktiv 0-Pegel Prg.-Ende
Stückzahl	inaktiv Prg.-Ende
Sprung	PS 04

Abbildung 68 - Verfahrensat 03

Verfahrensat 04

Dieser Verfahrensat dient zum Auslesen der Druckmarke. Die Beschleunigung/Verzögerung entspricht den anderen Verfahrensätzen. Als Verfahrensgeschwindigkeit ist eine geringe Geschwindigkeit gewählt, um ein Überfahren der Druckmarke, bei dem der Druckmarkenleser nicht reagiert, zu verhindern.

Zum Lesen der Druckmarke wird eine relative Positionierung mit „Touch Point“ gewählt. Dieser „Touch Point“ ist mit dem digitalen Eingang E1 verbunden. Die Positioniersteuerung wartet nun auf eine positive Flanke am Eingang E1, der mit dem Druckmarkenleser verbunden ist.

Die Gesamtlänge der Strecke dieses Verfahrensatzes entspricht neben der Toleranz zusätzlich einer ganzen Vorzugslänge. Dies dient für den Fall, dass eine neue Folie aufgelegt worden ist. Hierbei kann es vorkommen, dass der Antrieb die Druckmarke überfährt ohne auf sie zu reagieren. Es wird dann so lange weiter gefahren, bis die nächste Druckmarke erkannt wird.

Um zu verhindern, dass sich der Motor rückwärts dreht und die Folie entspannt, wird ein Restweg von 50mm abgefahren. Da dieser im Regelfall auch im vorigen Folienvorzug gefahren wurde, entspricht die Gesamtstrecke des Folienvorzuges wieder der Distanz zwischen zwei Druckmarken.

Nach Abschluss des Verfahrensatzes wird das Positionierprogramm beendet, und kann für den folgenden Folienvorzug wieder neu gestartet werden.

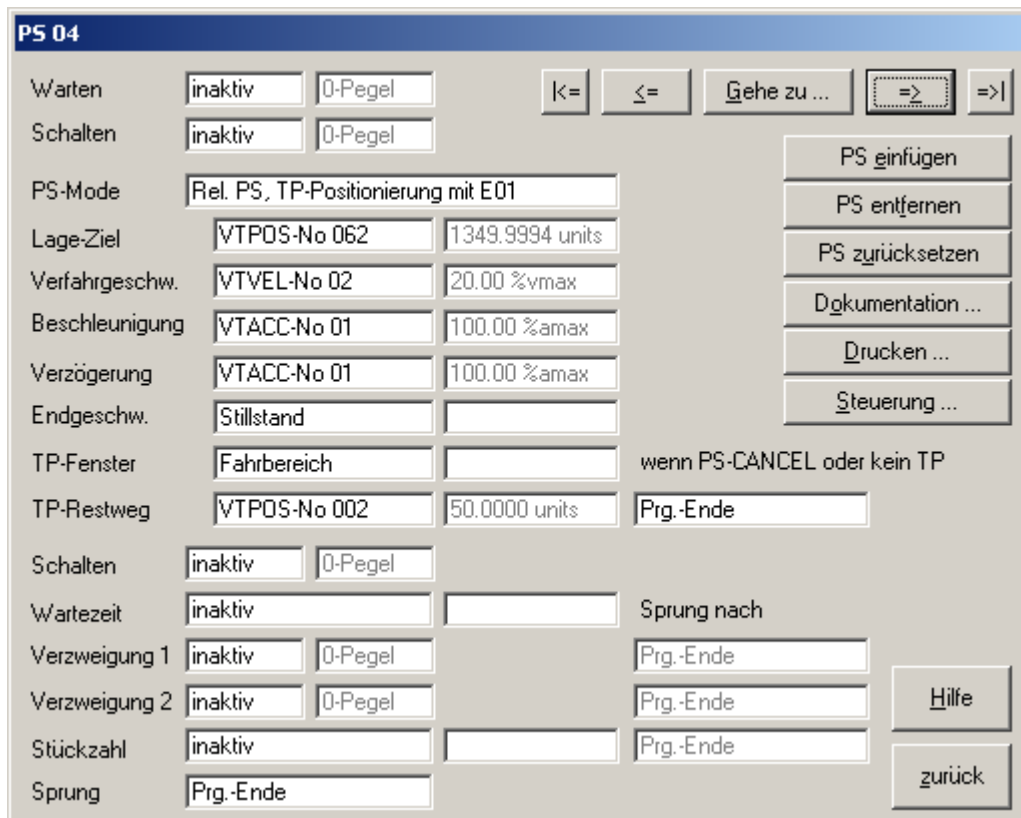


Abbildung 69 - Verfahrssatz 04

Bussteuerung

Der Antrieb wird nun weiter so konfiguriert, dass die zusätzlichen Signale für die Programmsteuerung über den Bus empfangen werden. Es wird dabei berücksichtigt, dass die Steuerung des Antriebes mit dem DRIVECOM PROFIL 20 erfolgt. Hierdurch sind einige der Bits des Steuerwortes bereits fest belegt. Die Signale des Steuerwortes werden vom EMF2133IB so aufbereitet, dass der Antrieb von außen durch das Profil in Betrieb genommen werden kann. Nach innen kann es dabei einige Abweichungen geben. Zusätzlich ist zu beachten, dass das HIGH-Wort und LOW-Wort eines Doppeldatenwortes in umgekehrter Reihenfolge gesendet wird und somit zur Verfügung steht. Dies wird jedoch in der SPS berücksichtigt.

Steuerwort Bit 5	POS-PRG START	Mit einer negativen Flanke an diesem Signal wird das Programm im Servoregler gestartet.
Steuerwort Bit 6	POS-PRG STOP	Dieses Signal ist „Low“ aktiv. Wenn das Signal aktiv ist, wird das Programm unterbrochen und zurückgesetzt.

Steuerwort Bit 11	POS-PRG RESET	„High“ aktiv. Bei Aktivierung wird das Programm zurückgesetzt.
Steuerwort Bit 12	MANUAL	„High“ aktiv. Handfahren aktiv
Steuerwort Bit 13	MANUAL-NEG	„High“ aktiv Handfahren in negative Richtung
Steuerwort Bit 14	MANUAL-POS	„High“ aktiv Handfahren in positive Richtung
Steuerwort Bit 15	DM-MODUS	Dieses Signal steuert das Signal PIF 01. Wenn das Signal „High“ ist, ist die Druckmarkenerkennung aktiv.

Die Vorgaben für Geschwindigkeit, Beschleunigung/Verzögerung und die Strecke werden durch die Parameterkommunikation geändert. Folgende Codestellen werden hierzu über den Profibus verändert.

C1301/1	Strecke
C1302/1	Verfahrgeschwindigkeit
C1302/2	Verfahrgeschwindigkeit Druckmarke lesen
C1303/1	Beschleunigung/Verzögerung

Im nächsten Schritt werden die zusätzlichen Statusmeldungen, die vom Servoregler über den Profibus gesendet werden, konfiguriert. Einige der Statusmeldungen sind bereits durch das DRIVECOM Profil 20 vordefiniert.

Statuswort Bit 14	RUNNING	Während des Programmablaufes ist dieses Signal „High“.
Statuswort Bit 15	POS-ENDED and POS-IN-TARGET	Wenn das Programm beendet ist und das Lageziel erreicht ist, wird dieses Signal „High“.

Prozessdatenwort 1	MCTRL-NACT	Dieses Signal stellt die relative Drehgeschwindigkeit der Motorwelle da. Es ist normiert auf $\pm 16384 = \pm 100\%$
Prozessdatenwort 2+3	POS-ACT	Aktueller Lageistwert Der Wert ist angegeben als Winkelsignal mit der Einheit [inc.]. Der Servoregler rechnet hierbei mit 65536 [inc./Umdrehung].

Zur Berechnung des aktuellen Lageistwertes in der SPS werden zusätzlich über den Parameterkanal folgende Codestellen ausgelesen:

C1204	Vorschubkonstante
C1202	Getriebe Zähler
C1203	Getriebe Nenner

8.4. SPS-Funktionsbaustein

Zur Steuerung des Servoantriebs wird ein Funktionsbaustein (FB) mit einem Instanzen Datenbaustein erstellt.

Beim Aufruf des FB enthält dieser folgende Ein- und Ausgabeelemente:

PZD_Adr	INPUT Word	Wortadresse der Prozessdaten im SPS Speicher
Druckmarke_lesen	INPUT Bool, HIGH aktiv	Aktivieren der Druckmarkenerkennung
Handfahren	INPUT Bool, HIGH aktiv	Aktivieren des Handfahren
Handfahren_pos	INPUT Bool, HIGH aktiv	Wenn Handfahren aktiv, dann Drehrichtung positiv
Handfahren_neg	INPUT Bool, HIGH aktiv	Wenn Handfahren aktiv, dann Drehrichtung negativ
POS_START	INPUT Bool	Mit einer positiven Flanke wird das Positionierprogramm gestartet.
POS_Stop	INPUT Bool, HIGH aktiv	Ein aktives Positionierprogramm wird unterbrochen und zurückgesetzt.
Error_RESET	INPUT Bool, HIGH aktiv	Sollte der Antrieb eine Störung gemeldet haben, kann, sobald die Störung behoben ist, der Antrieb mit diesem Signal zurücksetzt werden und sich wieder betriebsbereit melden.
Freigabe	INPUT Bool, HIGH aktiv	Wenn der Antrieb eingeschaltet ist, kann er hiermit betriebsbereit geschaltet werden.
Geschw	INPUT Int	Relative Geschwindigkeit für Verfahrensweg. Die Angabe enthält zwei Nachkommastellen. (Bsp. 85,37% = 8537) (Im FB begrenzt auf 10,00...100,00%)

Geschw_DM	INPUT Int	Relative Geschwindigkeit für Druckmarke lesen. Die Angabe enthält zwei Nachkommastellen. (Im FB begrenzt auf 10,00%...Geschw)
Beschl_Verz	INPUT Int	Relative Beschleunigung/Verzögerung. Die Angabe enthält zwei Nachkommastellen.
Strecke	INPUT Int	Vorgabe der Werkzeuglänge mit der Einheit Millimeter. Die Angabe enthält eine Nachkommastelle. (Bsp. 987,5mm = 9875)
Eingeschaltet	OUTPUT Bool, HIGH aktiv	Der Antrieb befindet sich im DRIVECOM Zustand „Eingeschaltet“
DRIVE_RDY	OUTPUT Bool, HIGH aktiv	Betrieb ist freigegeben
ERROR	OUTPUT Bool, HIGH aktiv	Es liegt eine Störung am Antrieb an.
RUNNING	OUTPUT Bool, HIGH aktiv	Das Positionierprogramm wird ausgeführt.
JOB_FIN	OUTPUT Bool, HIGH aktiv	Das Positionierprogramm wurde ohne Unterbrechung beendet.
MCTRL_NACT	OUTPUT Real	Relative Istgeschwindigkeit
POS_ACT	OUTPUT Real	Lageistwert angegeben in Millimetern

Zur Kommunikation mit der Peripherie werden die Systemfunktionen SFC 14 und SFC 15 verwendet. Der SFC 14 dient für das Einlesen der Daten mit konsistenter Datenübertragung. Zum Schreiben mit konsistenter Datenübertragung wird der SFC 15 verwendet.

Durch die konsistente Datenübertragung wird verhindert, dass die Daten im Busspeicher der SPS gleichzeitig gelesen und geschrieben werden.

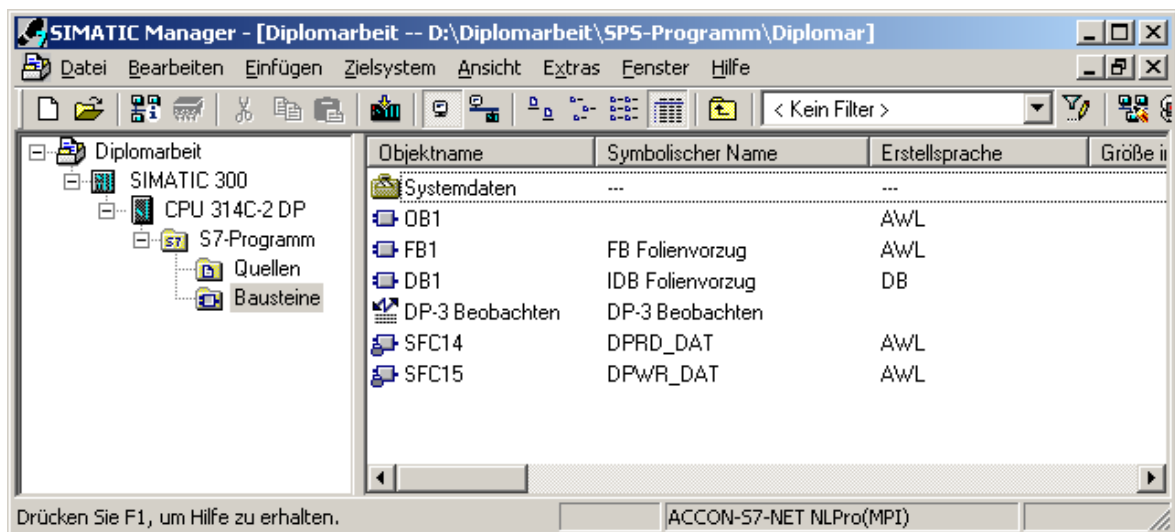


Abbildung 70 - Projekt Folienvorzug

Der Funktionsbaustein liest zunächst die Prozessdaten vom Antrieb ein und wertet diese aus. Dabei werden die Ausgabeelemente des FB in jedem Aufruf neu erstellt.

Es wird hierbei berücksichtigt, dass das LOW-Byte und HIGH-Byte getauscht werden müssen. Dies geschieht durch die Anordnung der Bytes im Speicher des Funktionsbausteins.

Nachdem der FB den aktuellen Zustand des Antriebes kennt, werden die Prozessdatenwörter für den Antrieb erstellt. Hierbei handelt es sich nur um das Steuerwort, da keine weiteren Informationen für den Antrieb über die Prozessdaten übermittelt werden. Das Steuerwort ist von dem Statuswort und den Eingangselementen des Funktionsbausteins abhängig. Es wird in jedem FB Aufruf neu angepasst. Hiernach werden die Prozessdatenwörter zum Antrieb geschrieben.

Zur Parameterkommunikation enthält der Funktionsbaustein ein Telegramm. Dieses enthält alle Informationen, die zum Auslesen oder Schreiben der einzelnen Parameter erforderlich sind.

Die Parameterkommunikation erfolgt azyklisch. Die Parameter werden nacheinander mit Hilfe eines Petrinetzes mit dem Antrieb ausgetauscht.

Durch Aktivieren „ERROR_Reset“ wird neben dem Rücksetzen einer Störung im Antrieb die Parameterkommunikation unterbrochen. Wird „ERROR_Reset“ zurückgesetzt, wird die Kommunikation wieder neu gestartet.

Im Anhang befindet sich der Quellcode des Funktionsbausteins. Die Netzwerke 1 bis 13 dienen zur Kommunikation der Prozessdatenwörter.

Zur Parameterkommunikation dienen die Netzwerke 14 bis 17. Das Netzwerk 16 und 17 enthält das Petrinetz, das sich ebenfalls im Anhang befindet.

9. Test Folienvorzug

Zum Testen des Systems wird eine Variablenbeobachtungstabelle erstellt. Dabei werden die Ein- und Ausgangelemente des Funktionsbausteins gesteuert und beobachtet.

Zusätzlich wird hier der Ausgang A125.7 gesteuert und auf „HIGH“ geschaltet. Die Reglersperre der Klemme 28 wird damit deaktiviert.

Um den Antrieb anzusteuern und den Betrieb freizugeben, müssen nun die Peripherieadresse der Prozessdatenwörter vorgegeben und das Signal „Freigabe“ auf HIGH gesteuert werden. Der Antrieb meldet dann „Eingeschaltet“=HIGH und „DRIVE_RDY“=HIGH und ist betriebsbereit.

Operand	Symbol	Anzeigeformat	Statuswert	Steuwert
1 A 125.7	"Klemme 28 RFG"	BOOL	true	true
2 A 125.0	"Druckmerkenleser"	BOOL	false	
3				
4 DB1.DBW 0	"IDB Folienvorzug".PZD_Adr	HEX	VW#16#0118	VW#16#0118
5 DB1.DBX 2.0	"IDB Folienvorzug".Druckmarke_Lesen	BOOL	false	
6 DB1.DBX 2.1	"IDB Folienvorzug".Handfahren	BOOL	false	
7 DB1.DBX 2.2	"IDB Folienvorzug".Handfahren_pos	BOOL	false	
8 DB1.DBX 2.3	"IDB Folienvorzug".Handfahren_neg	BOOL	false	
9 DB1.DBX 2.4	"IDB Folienvorzug".POS_START	BOOL	false	
10 DB1.DBX 2.5	"IDB Folienvorzug".POS_Stop	BOOL	false	
11 DB1.DBX 2.6	"IDB Folienvorzug".Error_RESET	BOOL	false	
12 DB1.DBX 2.7	"IDB Folienvorzug".Freigabe	BOOL	true	true
13 DB1.DBW 4	"IDB Folienvorzug".Geschw	DEZ	1000	
14 DB1.DBW 6	"IDB Folienvorzug".Geschw_DM	DEZ	1000	
15 DB1.DBW 8	"IDB Folienvorzug".Beschl_Verz	DEZ	1000	
16 DB1.DBW 10	"IDB Folienvorzug".Strecke	DEZ	2000	
17				
18 DB1.DBX 12.0	"IDB Folienvorzug".Eingeschaltet	BOOL	true	
19 DB1.DBX 12.1	"IDB Folienvorzug".DRIVE_RDY	BOOL	true	
20 DB1.DBX 12.2	"IDB Folienvorzug".ERROR	BOOL	false	
21 DB1.DBX 12.3	"IDB Folienvorzug".RUNNING	BOOL	false	
22 DB1.DBX 12.4	"IDB Folienvorzug".JOB_FIN	BOOL	false	
23 DB1.DBW 14	"IDB Folienvorzug".MCTRL_NACT	GLEITPUNKT	0.02441406	
24 DB1.DBW 18	"IDB Folienvorzug".POS_ACT	GLEITPUNKT	362.7856	
25				

Abbildung 71 - Antrieb Betrieb freigegeben

Es wird nun die Geschwindigkeit (100,00%), Beschleunigung/ Verzögerung (100,00%) und Strecke (1200,0mm) vorgegeben.

Wenn das Signal POS_START von LOW auf HIGH gesteuert wird, startet der Servoantrieb sein Positionierprogramm. Im Modus „Ohne Druckmarke“ erreicht er genau die vorgegebene Position 1200,0mm.

	Operand	Symbol	Anzeigeformat	Statuswert	Steuerwert
1	A 125.7	"Klemme 28 RFG"	BOOL	true	true
2	A 125.0	"Druckmerkenleser"	BOOL	false	
3					
4	DB1.DBW 0	"IDB Folienvorzug".PZD_Adr	HEX	VW#16#0118	VW#16#0118
5	DB1.DBX 2.0	"IDB Folienvorzug".Druckmarke_lesen	BOOL	false	
6	DB1.DBX 2.1	"IDB Folienvorzug".Handfahren	BOOL	false	
7	DB1.DBX 2.2	"IDB Folienvorzug".Handfahren_pos	BOOL	false	
8	DB1.DBX 2.3	"IDB Folienvorzug".Handfahren_neg	BOOL	false	
9	DB1.DBX 2.4	"IDB Folienvorzug".POS_START	BOOL	true	true
10	DB1.DBX 2.5	"IDB Folienvorzug".POS_Stop	BOOL	false	
11	DB1.DBX 2.6	"IDB Folienvorzug".Error_RESET	BOOL	false	
12	DB1.DBX 2.7	"IDB Folienvorzug".Freigabe	BOOL	true	true
13	DB1.DBW 4	"IDB Folienvorzug".Geschw	DEZ	10000	10000
14	DB1.DBW 6	"IDB Folienvorzug".Geschw_DM	DEZ	1000	
15	DB1.DBW 8	"IDB Folienvorzug".Beschl_Verz	DEZ	10000	10000
16	DB1.DBW 10	"IDB Folienvorzug".Strecke	DEZ	12000	12000
17					
18	DB1.DBX 12.0	"IDB Folienvorzug".Eingeschaltet	BOOL	true	
19	DB1.DBX 12.1	"IDB Folienvorzug".DRIVE_RDY	BOOL	true	
20	DB1.DBX 12.2	"IDB Folienvorzug".ERROR	BOOL	false	
21	DB1.DBX 12.3	"IDB Folienvorzug".RUNNING	BOOL	false	
22	DB1.DBX 12.4	"IDB Folienvorzug".JOB_FIN	BOOL	true	
23	DB1.DBD 14	"IDB Folienvorzug".MCTRL_NACT	GLEITPUNKT	0.06713867	
24	DB1.DBD 18	"IDB Folienvorzug".POS_ACT	GLEITPUNKT	1200.0	
25					

Diplomarbeit\SIMATIC 300\...\S7-Programm RUN Abs < 5.2

Abbildung 72 - Status: Folienvorzug erfolgreich beendet

Die Lage und das unterbrechungsfreie Beenden wird zurück an die SPS gesendet und an den Ausgangselementen JOB_FIN und POS_ACT dargestellt.

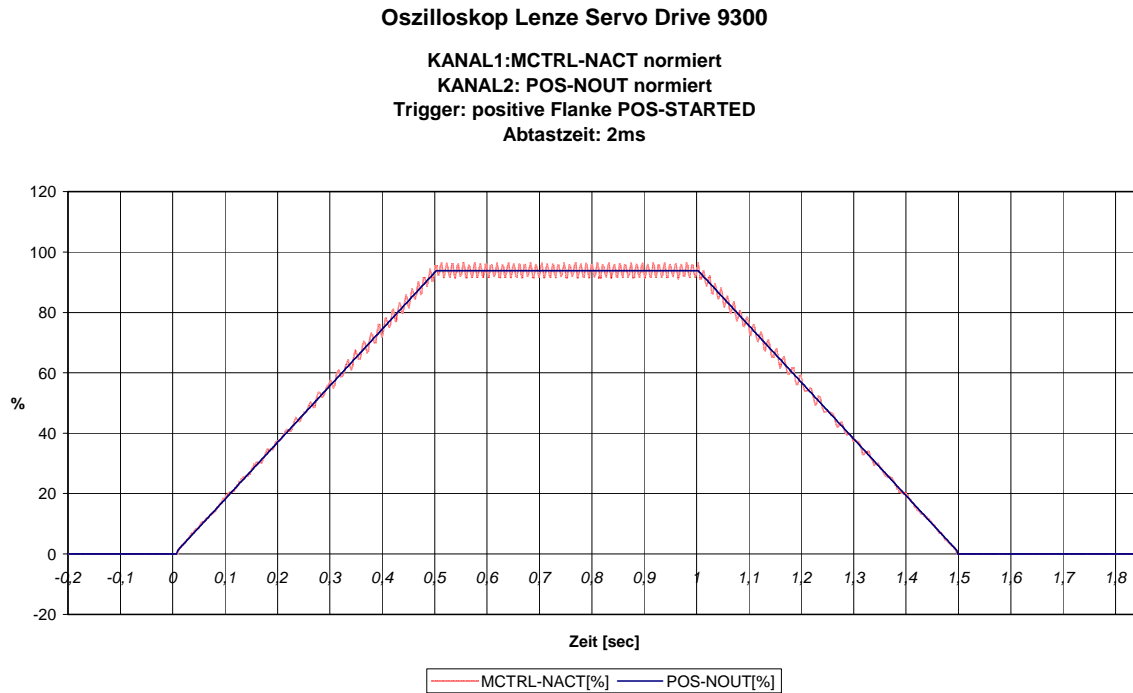


Abbildung 73 - Aufzeichnung Verfahrenweg

Auf dem mit Hilfe der Oszilloskopfunktion des Servoreglers aufgezeichnetem Verfahrenweg ist die Drehzahlsvorgabe des Interpolators (POS-N OUT) und die Motoristdrehzahl (MCTRL-NACT) zu erkennen. Hierbei wird deutlich, dass die Motoristdrehzahl der Drehzahlvorgabe folgt. Sie schwingt hierbei mit etwa 2% um den Vorgabewert. Die Sollwertsprünge werden entsprechend der Vorgabe des Verfahrenweges durch den Interpolator aufbereitet und an der Langregelung vorgegeben.

Die Zykluszeit des Folienvorzuges entspricht dem vorgegebenen Wert 1,5sec.

Beim wiederholten Anfahren einer Positionsvorgabe ist zu erkennen, dass in 10 von 10 Positionierungen die Zielvorgabe erreicht wird.

Es ist jedoch der interne Rundungsfehler für den Vorschub des Servoreglers zu beachten. Dieser entsteht, weil vorgegebene Lage-Ziele eine gebrochene Anzahl an Inkrementen ergeben können. Die interne Berechnung des Servoreglers berücksichtigt jedoch nur den ganzzahligen Anteil eines Wertes. Durch Multiplikation der Lage-Auflösung und des Lagewertes ist der Rundungsfehler zu erkennen. /11/

$$\text{Vorschub} = \text{Lage} - \text{Auflösung} \cdot \text{Lagewert}$$

$$= 2561,8470 \frac{\text{inc}}{\text{unit}} \cdot 1200,0 \text{unit} \quad 1[\text{unit}] = 1[\text{mm}]$$

$$= 3074216,45 \text{inc}$$

$$\text{Rundungsfehler} = 0,45 \text{inc}$$

Die Lage-Auflösung ergibt sich dabei aus der internen Auflösung des Winkelsignals, der Vorschubkonstanten und dem Getriebefaktor.

$$\begin{aligned}\text{Lage - Auflösung} &= \text{Auflösung Winkelsignal} \cdot \frac{\text{Getriebefaktor}}{\text{Vorschubkonstante}} \\ &= 65536 \frac{\text{inc}}{\text{Umdrehung}} \cdot \frac{7}{179,0708 \frac{\text{unit}}{\text{Umdrehung}}} \\ &= 2561,8470 \frac{\text{inc}}{\text{unit}}\end{aligned}$$

Mit dem Rundungsfehler ergibt sich eine Abweichung von $0,175\mu\text{m}$. Dabei ist die vorgegebene Genauigkeit von $\pm 0,01\text{mm}$ erfüllt. Die Toleranz des Getriebes und der Vorzugwelle sind bei dieser Betrachtung nicht berücksichtigt.

Beim relativen Positionieren ist nun jedoch zu beachten, dass sich die Fehler bei jedem Fahren summieren.

Bei dem hier erstellten Folienvorzug kann dieser Fehler jedoch unberücksichtigt bleiben.

10. Zusammenfassung

Im Rahmen dieser Arbeit wurde am Beispiel eines Folienvorzuges der Einsatz eines Profibus vernetzten Servoantriebs mit Positioniersteuerung umgesetzt.

Es wurde gezeigt, dass die Positioniersteuerung eines Servoantriebs über den Profibus erfolgen kann. Hierdurch wurde der Verdrahtungsaufwand verringert. Dies ermöglicht zusätzlich den Verzicht auf Ein- und Ausgangsbaugruppen der zentralen Steuerung.

Durch das Positionierprogramm und den Datenaustausch erfolgt eine genaue Positionierung.

Der Benutzer kann die Position genau vorgeben, ohne dafür die Geschwindigkeit und Zeit einstellen zu müssen.

An dieser Stelle wird noch einmal darauf hingewiesen, dass beim Einsatz dieses Systems in der Praxis die Parameter der Regelkreise an die reale Belastung angepasst werden müssen. Erst so kann gewährleistet werden, dass diese zu einem optimalen Verhalten des Motors führen.

In Zukunft werden, aufgrund der gezeigten Einsparungen und sehr guten Positioniergenauigkeit, vermehrt Bus-Systeme zur Kommunikation mit Servoantrieben eingesetzt werden. Dabei dienen die Antriebe meist zur Steuerung und Überwachung auch von Einzelprozessen.

Das Projektieren eines Gesamtprozesses mit einem Folienvorzug kann durch das Verwenden des Funktionsbausteins leicht durchgeführt werden. Dies erleichtert die Programmierung und Installation zusätzlich.

11. Abbildungsverzeichnis

Abbildung 1 - Installation derzeitiger Positionierantriebe.....	7
Abbildung 2 - Das ISO/OSI Modell für Kommunikationsstandards /1/	9
Abbildung 3 - Profibus-Anwendungsgebiete /2/	11
Abbildung 4 - Der Profibus UART-Zeichenrahmen /1/	11
Abbildung 5 - Buskonfiguration mit Repeater /1/	12
Abbildung 6 - Maximale Segmentlänge in Abhängigkeit der Baudrate /1/.....	13
Abbildung 7 - Spezifikation des Profibus-RS485-Kabels Typ A /1/	13
Abbildung 8 - Kontaktbelegung des 9-Poligen D-Sub-Steckers /1/	13
Abbildung 9 - Aufbau eines RS485-Bussegmentes /1/	14
Abbildung 10 - Hybrides Profibus-Buszugriffsverfahren /2/	16
Abbildung 11 - Profibus-Telegrammformat /2/	17
Abbildung 12 - Datenübertragungsdienste /2/	17
Abbildung 13 - Profibus-Busparameter /1/	19
Abbildung 14 - Profibus-Profile /2/	21
Abbildung 15 - Datenstruktur DRIVECOM	22
Abbildung 16 - Zustandsänderung /8/	23
Abbildung 17 - Zustandsdiagramm /8/.....	24
Abbildung 18 - Bitbelegung Steuerwort /8/	27
Abbildung 19 - Bitbelegung Statuswort /8/	27
Abbildung 20 - Telegrammaufbau Parameterkanal /10/	28
Abbildung 21 - Servicebyte /10/	28
Abbildung 22 - 4-Quadranten Betrieb /5/.....	30
Abbildung 23 - Aufbau von Servoantrieben mit Synchronmotor /5/	31
Abbildung 24 - Drehmomentgleichung	31
Abbildung 25 - Servoantrieb mit Servoregler (Frequenzumrichter).....	34
Abbildung 26 - Beschreibung des Strom-Raumzeigers in verschiedenen Koordinatensystemen /5/	34
Abbildung 27 - Läuferflussorientiertes Koordinatensystem des Synchronmotors /5/ ..	35
Abbildung 28 - Regelungsstruktur des Servoreglers beim Synchronmotor /5/.....	36
Abbildung 29 - Struktur der Kaskadenregelung bei elektrischen Antrieben /5/	36
Abbildung 30 - Stromregelkreis am Beispiel eines Gleichstrommotors /5/	37
Abbildung 31 - Drehzahlregelkreis /5/	38
Abbildung 32 - Lageregelkreis /5/.....	39
Abbildung 33 - Funktionelle Bestandteile eine Positioniersteuerung /5/.....	40
Abbildung 34 - Wichtige Bestandteile eines Verfahrssatzes /5/.....	41
Abbildung 35 - Verfahrenweg bestehend aus mehreren Verfahrssätzen	42
Abbildung 36 - Interpretation der Zielposition beim absoluten und relativen Positionieren /5/.....	42
Abbildung 37 - Typische Steuer- und Zustandsignale einer Positioniersteuerung /5/..	42
Abbildung 38 - Bedeutung des Interpolators für überschwingfreie Positioniervorgänge /5/.....	43
Abbildung 39 - Aufbereitung des gemessenen Lageistwertes für den Lagegeber /5/ ..	44
Abbildung 40 - Optimaler Getriebefaktor bei Servoanwendungen /5/	46
Abbildung 41 - Ersatzsystem für direkten Bezug auf Motorwelle	46
Abbildung 42 - Übertragung markanter Arbeitspunkte in das Drehzahl-Drehmoment- Diagramm des Motors /5/.....	47
Abbildung 43 - Ersatzarbeitspunkt zur thermischen Auslegung /5/.....	47
Abbildung 44 - Ermittlung des Stromverlaufs im Leistungsteil aus dem Drehmomentverlauf an der Motorwelle /5/.....	48
Abbildung 45 - SIMATIC-Manager	50
Abbildung 46 - CPU 314C-2DP /13/.....	52
Abbildung 47 - Lenze Servo-Motor der Baureihe MD_KS /15/	53
Abbildung 48 - Lenze Servo-Umrichter 9300 /16/	54
Abbildung 49 - Lenze EMF2133IB Profibusmodul /17/.....	55
Abbildung 50 - Profibus vernetzter Servoantrieb und SPS	57
Abbildung 51 - Folienvorzug einer Verpackungsmaschine.....	58
Abbildung 52 - Aufbau Profibus-DP Netz	59
Abbildung 53 - Übersicht neues Projekt	60

Abbildung 54 – NetPro	60
Abbildung 55 - Profibus Objekteigenschaften	61
Abbildung 56 - Projektübersicht Diplomarbeit.....	62
Abbildung 57 - HW-Konfig	63
Abbildung 58 - DP-Slave Konfiguration	64
Abbildung 59 - Busparameter.....	65
Abbildung 60 - Installation Servoantrieb.....	67
Abbildung 61 - C173/0	68
Abbildung 62 - Maschinenparameter	70
Abbildung 63 – Fahrprofil	71
Abbildung 64 - variable Vorzugzeit.....	72
Abbildung 65 - Fahrwege für Folienvorzug	73
Abbildung 66 - Verfahrssatz 01	74
Abbildung 67 - Verfahrssatz 02.....	75
Abbildung 68 - Verfahrssatz 03.....	76
Abbildung 69 - Verfahrssatz 04.....	78
Abbildung 70 - Projekt Folienvorzug	83
Abbildung 71 - Antrieb Betrieb freigegeben	84
Abbildung 72 - Status: Folienvorzug erfolgreich beendet	85
Abbildung 73 - Aufzeichnung Verfahrweg.....	86

12. Formelzeichenerklärung

a	Beschleunigung/Verzögerung
a_{\max}	maximale Beschleunigung/Verzögerung
f_N	Bemessungsfrequenz
F_{Schlepp}	Schleppfehler Lageregelung
i	Getriebefaktor
I	Strom
I_0	Stillstandsstrom
I_{\max}	maximal Strom
I_N	Bemessungsstrom
J	Trägheitsmoment
J_{Last}	Lastträgheitsmoment
J_{Last_M}	Lastträgheitsmoment getriebemotorseitig
J_{Motor}	Motorträgheitsmoment
K_P	Proportionalverstärkung
k_T	Drehmomentkonstante
k_V	Proportionalverstärkung Lageregler
k_{Vorzug}	Vorzugskonstante
M_0	Stillstands Drehmoment
M_{Last}	Lastdrehmoment
M_{Last_M}	Lastdrehmoment getriebeseitig
M_M	Motormoment
M_{\max}	maximal Drehmoment
M_N	Bemessungsdrehmoment
n	Drehzahl
n_{\max}	maximale Drehzahl
n_N	Bemessungsdrehzahl
P_N	Bemessungsleistung
s	Strecke
T_{Σ}	Ersatzzeitkonstante des Leistungsteils
T_i	Gesamträgheitszeitkonstante
$T_{I\text{-Reg}}$	Verzögerungszeitkonstante Stromregelkreis
T_N	Nachstellzeit
$T_{n\text{-Reg}}$	Verzögerungszeitkonstante Drehzahlregelkreis
T_S	Streckenzeitkonstante
t_{Vorzug}	Zeit für Folienvorzug
$U_{N,AC}$	Bemessungswechselspannung
v	Geschwindigkeit
$v_{\text{Druckmarke lesen}}$	Geschwindigkeit beim Druckmarke lesen
v_{\max}	maximale Geschwindigkeit
V_S	Streckenverstärkung
ω	Winkelgeschwindigkeit

13. Quellenangaben

- /1/ Josef Weigmann, Gerhard Kilian: Dezentralisieren mit Profibus DP/DPV1; Publicis Corporate Publishing; Erlangen; 3. Auflage 2002
- /2/ Gerhard Schnell, Bernhard Wiedemann (HRSG.): Bussysteme in der Automatisierungs- und Prozesstechnik; Vieweg+Teubner Verlag; Wiesbaden; 7.Auflage 2008
- /3/ Ralf Kories, Heinz Schmidt-Walter: Taschenbuch der Elektrotechnik; Verlag Harri Deutsch; Frankfurt am Main; 6. Auflage 2004
- /4/ Manfred Schulze: Elektrische Servoantriebe; Fachbuchverlag Leipzig im Carl Hanser Verlag; München; 1.Auflage 2008
- /5/ Jens Weidauer: Elektrische Antriebstechnik; Publicis Corporate Publishing; Erlangen; 2008
- /6/ Holger Lutz, Wolfgang Wendt: Taschenbuch der Regelungstechnik; Verlag Harri Deutsch; Frankfurt am Main; 7.Auflage 2007
- /7/ <http://de.wikipedia.org/wiki/Profibus> [29.10.2008]
- /8/ Antriebstechnik Gerätesteuerung; Profil-Nummer 20; vom 30. Januar 1995; Herausgegeben vom DRIVECOM Nutzergruppe e.V. Blomberg
- /9/ SIMATIC Programmieren mit Step7 Handbuch; Siemens AG Automation and Drive; Nürnberg; Ausgabe 03/2006
- /10/ Kommunikationshandbuch Profibus EMF2133IB; Lenze Drive Systems GmbH; Hameln; Ausgabe 06/2006
- /11/ Systemhandbuch Global Drive Servo-Positionierregler 9300; Lenze Drive Systems GmbH; Hameln; Ausgabe 03/2003
- /12/ SIMATIC S7 300 CPU31XC und 31X Technische Daten Gerätehandbuch; Siemens AG Automation and Drive; Nürnberg; Ausgabe 12/2006
- /13/ <https://www.automation.siemens.com/bilddb/index.asp?aktPrim=0&nodeID=10000736&lang=de> [19.12.2008]
- /14/ <http://akb.lenze.de/akb/infopool.nsf/html/index-inhalt> [08.01.2009]

- /15/ [http://www.lenze.de/lenze.de/de/active/020/Products/020/Servo drives/020/Servomotoren/020/Synchron-Servomotoren MDxKS/Produkt/Synchron-Servomotoren MDxKS .de.jsp?cid=0b0164e08017f8ac](http://www.lenze.de/lenze.de/de/active/020/Products/020/Servo%20drives/020/Servomotoren/020/Synchron-Servomotoren%20MDxKS/Produkt/Synchron-Servomotoren%20MDxKS.de.jsp?cid=0b0164e08017f8ac) [08.01.2009]
- /16/ [http://www.lenze.de/lenze.de/de/active/020/Products/020/Servo drives/010/Servo Umrichter/030/Servo Umrichter 9300/Servo Umrichter9300.de.jsp?cid=0b0164e08017f8a3](http://www.lenze.de/lenze.de/de/active/020/Products/020/Servo%20drives/010/Servo%20Umrichter/030/Servo%20Umrichter%209300/Servo%20Umrichter9300.de.jsp?cid=0b0164e08017f8a3) [08.01.2009]
- /17/ [http://www.lenze.de/lenze.de/de/active/020/Products/015/Industrielle Kommunikation/080/Profibus/InduKomm Profibus.de.jsp?cid=0b0164e08017f80a](http://www.lenze.de/lenze.de/de/active/020/Products/015/Industrielle%20Kommunikation/080/Profibus/InduKomm/Profibus.de.jsp?cid=0b0164e08017f80a) [08.01.2009]
- /18/ Handbuch/Manual Global Drive Control (GDC); Lenze Drive Systems GmbH; Hameln; GDC DE/EN 3.2
- /19/ Günter Wellenreuther, Dieter Zastrow; Automatisieren mit SPS, Theorie und Praxis; Vieweg Verlag; Wiesbaden; 3.Auflage 2005

14. Selbstständigkeitserklärung

Hiermit versichere ich, Kay Christian Schröder, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §25(4) ohne fremde Hilfe selbständig 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.

Bad Bramstedt, 19.Januar 2009

Kay Christian Schröder

15. Danksagung

An dieser Stelle möchte ich mich noch einmal bei der Firma Ribbeck GmbH, speziell bei Herrn Ribbeck für seine ideelle sowie materielle Unterstützung bedanken. Mein Dank gilt auch seinen Mitarbeitern, sowie insbesondere meinem betreuenden Professor der HAW Hamburg Herrn Prof. Dr. rer. nat. Henning Hasemann für die Unterstützung und Hilfsbereitschaft beim erstellen dieser Arbeit.

Ein besonderer Dank gilt meinen Eltern Bernd und Brigitte Schröder. Sie haben mir dieses Studium ermöglicht und mich dabei immer unterstützt.

A. Anhang

A.1. Inhalt CD-ROM

Ein Exemplar dieser CD-ROM ist beim Prüfer hinterlegt und kann dort eingesehen werden.

- Ausarbeitung Diplomarbeit
 - Deckblatt.PDF
 - Seite1.PDF
 - Seite2.PDF
 - Diplomarbeit.PDF
- SPS Programm
 - Diplomarbeit.ZIP
- Servo Programm
 - Diplomarbeit.GDC

A.2. Dokumentation FB Folienvorzug

```
FUNCTION_BLOCK "FB Folienvorzug"
```

```
TITLE =
```

```
//Dieser Funktionsbaustein steuert einen Servopositionierantrieb. Der
//Antrieb dient für den Folienvorzug einer Verpackungsmaschine. Dabei verfügt
//dieser über eine Druckmarkenerkennung, die bei Bedarf aktiviert wird. Die
//Kommunikation erfolgt über einen PROFBUS mit dem DRIVECOM Profil 20. Der
//Datenaustausch ist konsistent. Es werden vier Prozessdatenwörter und 4
//Parameterdatenwörter zwischen FB und Antrieb ausgetauscht. Es ist zu
//beachten, dass die Parameterdatenwörter direkt vor den Prozessdatenwörtern im
//Speicher der SPS liegen müssen.
VERSION : 0.1
```

```
VAR_INPUT
```

```
PZD_Adr : WORD ; //Peripherieadresse Prozessdatenwort
Druckmarke_lesen : BOOL ; //Modus Druckmarke lesen 1=Aktiv
Handfahren : BOOL ; //Aktiviert das Handfahren
Handfahren_pos : BOOL ; //bei aktivierten Handfahren Drehrichtung positiv
Handfahren_neg : BOOL ; //bi aktivierten Handfahren Drehrichtung negative
POS_START : BOOL ; //positive Flanke startet Positionierprogramm
POS_Stop : BOOL ; //Stoppen und Rücksetzen des Positionierprogramms
Error_RESET : BOOL ; //Stoerung rücksetzen
Freigabe : BOOL ; //Antrieb freigeben
Geschw : INT ; //relative Geschwindigkeit % inkl. zwei Nachkommastelle
Geschw_DM : INT ; //relative Geschwindigkeit DM lesen % inkl zwei Nachkommastelle
Beschl_Verz : INT ; //relative Beschleunigung/Verzoegerung % inkl. zwei Nachkommastellen
Strecke : INT ; //Werkzeuglaenge Angabe in mm inkl. einer Nachkommastelle
```

```
END_VAR
```

```
VAR_OUTPUT
```

```
Eingeschaltet : BOOL ;
DRIVE_RDY : BOOL ; //Betrieb ist freigegeben
ERROR : BOOL ; //Stoerung ist aktiv
RUNNING : BOOL ; //Positionierung aktiv
JOB_FIN : BOOL ; //Positionierung ohne Unterbrechung beendet
MCTRL_NACT : REAL ; //relative Geschwindigkeit in %
POS_ACT : REAL ; //Aktueller Lageistwert in mm
```

```
END_VAR
```

```
VAR
```

```
PAR_Telegramm : STRUCT //Telegramme für Parameterkommunikation
  Geschw : STRUCT
    Servicebyte : BYTE ;
    Subindex : BYTE ;
    Index : WORD ;
    Data : DWORD ;
  END_STRUCT ;
  Geschw_DM : STRUCT
    Servicebyte : BYTE ;
    Subindex : BYTE ;
    Index : WORD ;
    Data : DWORD ;
  END_STRUCT ;
  Beschl_Verz : STRUCT
    Servicebyte : BYTE ;
    Subindex : BYTE ;
    Index : WORD ;
    Data : DWORD ;
  END_STRUCT ;
  Strecke : STRUCT
    Servicebyte : BYTE ;
    Subindex : BYTE ;
    Index : WORD ;
    Data : DWORD ;
  END_STRUCT ;
  Vorschubkonstante : STRUCT
    Servicebyte : BYTE ;
```

```

Subindex : BYTE ;
Index : WORD ;
Data : DWORD ;
END_STRUCT ;
Getriebe_zaeher : STRUCT
Servicebyte : BYTE ;
Subindex : BYTE ;
Index : WORD ;
Data : DWORD ;
END_STRUCT ;
Getriebe_nenner : STRUCT
Servicebyte : BYTE ;
Subindex : BYTE ;
Index : WORD ;
Data : DWORD ;
END_STRUCT ;
END_STRUCT ;
PZD_receive : STRUCT //Prozessdaten <-- Antrieb
Statuswort : STRUCT
Meldung : BOOL ;
Remote : BOOL ;
Sollwert_erreicht : BOOL ;
Grenzwert : BOOL ;
DCTRL_FAIL_QSP : BOOL ;
reserviert_2 : BOOL ;
RUNNING : BOOL ;
POS_ENDED_and_POS_IN_TAR : BOOL ;
Einschaltbereit : BOOL ;
Eingeschaltet : BOOL ;
Betrieb_freigegeben : BOOL ;
Stoerung_TRIP : BOOL ;
Spannung_gesperrt : BOOL ;
Schnellhalt : BOOL ;
Einschaltsperrung : BOOL ;
Warnung : BOOL ;
END_STRUCT ;
PZDW_1 : WORD ;
PZDW_2 : WORD ;
PZDW_3 : WORD ;
END_STRUCT ;
PZD_send : STRUCT //Prozessdaten --> Antrieb
Steuerwort : STRUCT //Steuerwort -> Antr.-Regler
DRIVECOM_reserviert_1 : BOOL ; //Steuerwort Bit 08: -> DRIVECOM reserviert
DRIVECOM_reserviert_2 : BOOL ; //Steuerwort Bit 09: -> DRIVECOM reserviert
DRIVECOM_reserviert_3 : BOOL ; //Steuerwort Bit 10: -> DRIVECOM reserviert
POS_RESET : BOOL ; //Steuerwort Bit 11: -> Positionierung zuruecksetzen
POS_MANUAL : BOOL ; //Steuerwort Bit 13: -> Hersteller Bit 13, siehe Manual
POS_MANUAL_NEG : BOOL ; //Steuerwort Bit 12: -> Hersteller Bit 12, siehe Manual
POS_MANUAL_POS : BOOL ; //Steuerwort Bit 14: -> Hersteller Bit 15, siehe Manual
Modus_DM : BOOL ; //Steuerwort Bit 15: -> Hersteller Bit 15, siehe Manual
Einschalten : BOOL ; //Steuerwort Bit 00: -> DRIVECOM Einschalten
Spg_sperren : BOOL ; //Steuerwort Bit 01: -> DRIVECOM Spannung sperren
Schnellhalt : BOOL ; //Steuerwort Bit 02: -> DRIVECOM Schnellhalt
Betrieb_freigeben : BOOL ; //Steuerwort Bit 03: -> RFR (Reglerfreigabe)
HLG_sperren : BOOL ; //Steuerwort Bit 04: -> Quickstop
POS_START : BOOL ; //Steuerwort Bit 05: -> Programm starten mit positiver Flanke
POS_STOP : BOOL ; //Steuerwort Bit 06: -> Positionierung wird gestoppt und zurueckgesetzt
Reset_Stoerung : BOOL ; //Steuerwort Bit 07: -> Reset Stoerung
END_STRUCT ;
PZD_1 : WORD ;
PZD_2 : WORD ;
PZD_3 : WORD ;
END_STRUCT ;
PAR_receive : STRUCT //Parameterdaten <-- Antrieb
Servicebyte : BYTE ;
Subindex : BYTE ;
Index : WORD ;
Data : DWORD ;

```

```

END_STRUCT ;
PAR_send : STRUCT //Parameterdaten --> Antrieb
  Servicebyte : BYTE ;
  Subindex : BYTE ;
  Index : WORD ;
  Data : DWORD ;
END_STRUCT ;
Petrinetz : STRUCT //Zustandsspeicher Petrinetz
  Merker_1 : BOOL := TRUE;
  Merker_2 : BOOL ;
  Merker_3 : BOOL ;
  Merker_4 : BOOL ;
  Merker_5 : BOOL ;
  Merker_6 : BOOL ;
  Merker_7 : BOOL ;
  Merker_8 : BOOL ;
  Merker_9 : BOOL ;
  Merker_10 : BOOL := TRUE;
  Merker_11 : BOOL ;
END_STRUCT ;
ACT_TELE_pointer : WORD ; //Speicheradresse Byte des aktuellen Parametertelegramms
ACT_TELE : WORD ; //Nummer des Aktuellen Parametertelegramms
Vorschubkonstante : REAL ;
Getriebe_zaehler : REAL ;
Getriebe_nenner : REAL ;
END_VAR
VAR_TEMP
  return : INT ;
  PAR_Adr : WORD ;
  Lesebefehl : WORD ; //Parameterlesebefehl für 4 Byte Datenlaenge
  Schreibebehl : WORD ; //Parameterschreibebehl für 4 Byte Datenlaenge
END_VAR
BEGIN
NETWORK
TITLE =PZD Antrieb -> SPS
//Auslesen der Prozessdatenwörter mit dem SFC 14.
  CALL "DPRD_DAT" (
    LADDR          := #PZD_Adr,
    RET_VAL        := #return,
    RECORD         := #PZD_receive);

NETWORK
TITLE =Statuswort auswerten für Steuerung
// Stoerung
  O #PZD_receive.Statuswort.Meldung;
  O #PZD_receive.Statuswort.Stoerung_TRIP;
  O #PZD_receive.Statuswort.Warnung;
  O #PZD_receive.Statuswort.DCTRL_FAIL_QSP;
  = #ERROR;

// Positionierprogramm erfolgreich beendet
  U #PZD_receive.Statuswort.POS_ENDED_and_POS_IN_TAR;
  = #JOB_FIN;

// Positiniierprogramm wird ausgefuehrt
  U #PZD_receive.Statuswort.RUNNING;
  = #RUNNING;

// Der Betrieb des Servos ist freigegeben
  U #PZD_receive.Statuswort.Betrieb_freigegeben;
  = #DRIVE_RDY;

// Der Antrieb ist im DRIVECOM Zustand Eingeschaltet
  U #PZD_receive.Statuswort.Eingeschaltet;
  = #Eingeschaltet;

NETWORK
TITLE =Drehzahl berechnen

```

//Es wird die relative Drehzahl vom Motor übermittelt.

```
L #PZD_receive.PZDW_1;
ITD ;
DTR ;
L 1.638400e+004;
/R ;
L 1.000000e+002;
*R ;
T #MCTRL_NACT;
```

NETWORK

TITLE =Lageistwert berechnen

//Der Lageistwert wird in Inkrementen vom Servo übermittelt. Mit Hilfe von

//Vorschubkonstante und Getriebefaktor wird sie in [mm] umgerechnet.

//Es wird zusätzlich berücksichtigt, dass HIGH-Wort und LOW-Wort getauscht werden

//muessen.

```
L #PZD_receive.PZDW_3;
SLD 16;
L #PZD_receive.PZDW_2;
OD ;
DTR ;
L L#65536;
DTR ;
/R ;
L #Vorschubkonstante;
*R ;
L #Getriebe_zahler;
/R ;
L #Getriebe_nenner;
*R ;
T #POS_ACT;
```

NETWORK

TITLE =Modus Druckmarke / ohne Druckmarke

//Es wird die Druckmarkenerkennung bei Bedarf aktiviert.

```
U #Druckmarke_lesen;
= #PZD_send.Steuerwort.Modus_DM;
```

NETWORK

TITLE =Handfahren

//Steuerung des Handfahren.

//Wenn Handfahren aktiviert ist, kann das Positionierprogramm nicht gestartet

//werden.

```
U #Handfahren;
= #PZD_send.Steuerwort.POS_MANUAL;

U #Handfahren_pos;
= #PZD_send.Steuerwort.POS_MANUAL_POS;

U #Handfahren_neg;
= #PZD_send.Steuerwort.POS_MANUAL_NEG;
```

NETWORK

TITLE =Positionierung steuern

//Steuerung des Positionierprogramms.

//Mit STOP wird das Programm unterbrochen und zurückgesetzt.

```
UN #POS_START;
UN #ERROR;
= #PZD_send.Steuerwort.POS_START;

UN #POS_Stop;
= #PZD_send.Steuerwort.POS_STOP;
```

NETWORK

TITLE =Error_RESET

//Wenn der Antrieb eine Stoerung meldet kann sie zurueckgesetzt werden.

//Die Stoerung kann nur zurückgesetzt werden, wenn er das Positionierprogramm

//nicht ausführt und die Stoerung behoben ist.

```

    U( ;
    O #PZD_receive.Statuswort.Stoerung_TRIP;
    ON #RUNNING;
    ) ;
    U #Error_RESET;
    = #PZD_send.Steuerwort.Reset_Stoerung;

NETWORK
TITLE =Abbruch Positionieren
//Die Positionierung wird unterbrochen und zurueckgesetzt.
    O #POS_Stop;
    O #PZD_send.Steuerwort.Reset_Stoerung;
    = #PZD_send.Steuerwort.POS_RESET;

NETWORK
TITLE =Reglerfreigabe
//Freigabe des Antribs nach dem DRIVECOM Profil 20
    U #Freigabe;
    = #PZD_send.Steuerwort.Betrieb_freigeben;

NETWORK
TITLE =/QSP -> Antr.-Regler
//Zuweisung der Verknuepfung /Quickstop zum Antr.-Regler
//( diese Verknüpfung ist negiert: 0:= Quickstop zum Antr.-Regler
//
//          1:= kein Quickstop zum Antr.-Regler)
//
//Der Quickstop wird beim Servoantrieb nicht ausgeführt uns ist immer deaktiviert
    SET ;
    = #PZD_send.Steuerwort.HLG_sperren;

NETWORK
TITLE =DRIVECOM Einschaltsperr->Einschaltbereit->Eingeschaltet
//Wechsel vom Zustand Einschaltsperr in den Zustand Eingeschaltet der
//Zustandsmaschine nach dem DRIVECOM Profil 20
// => "Eingeschaltet"
    SET ;
    S #PZD_send.Steuerwort.Einschalten;
    S #PZD_send.Steuerwort.Spg_sperren;
    S #PZD_send.Steuerwort.Schnellhalt;
// => "Einschaltbereit"
    UN #PZD_receive.Statuswort.Einschaltbereit;
    UN #PZD_receive.Statuswort.Stoerung_TRIP;
    R #PZD_send.Steuerwort.Einschalten;

NETWORK
TITLE =Prozessdaten -> Antr.-Regler
//Prozessdaten werden zum Antrieb geschrieben
    CALL "DPWR_DAT" (
        LADDR          := #PZD_Adr,
        RECORD          := #PZD_send,
        RET_VAL         := #return);

NETWORK
TITLE =Werte Begrenzen
//Die Werte für die Strecke, Geschwindigkeit, Geschwindigkeit Druckmerkelesen und
//die Beschleunigung/Verzoegerung werden begrenzt.
// Steck 200,0...2000,0[mm]
    L #Strecke;
    L L#2000;
    <D ;
    SPBN g1;
    T #Strecke;
g1: L #Strecke;
    L L#20000;
    >D ;
    SPBN g2;
    T #Strecke;
g2: NOP 0;

```

```

// Geschwindigkeit 10,00...100,00[%]
L #Geschw;
L 1000;
<I ;
SPBN g3;
T #Geschw;
g3: L #Geschw;
L 10000;
>I ;
SPBN g4;
T #Geschw;
g4: NOP 0;

// Geschwindigkeit Druckmarke 10,00...Geschw[%]
L #Geschw_DM;
L 1000;
<I ;
SPBN g5;
T #Geschw_DM;
g5: L #Geschw_DM;
L #Geschw;
>I ;
SPBN g6;
T #Geschw_DM;
g6: NOP 0;

// Beschleunigung/Verzoegerung 10,00...100,00[%]
L #Beschl_Verz;
L 1000;
<I ;
SPBN g7;
T #Beschl_Verz;
g7: L #Beschl_Verz;
L 10000;
>I ;
SPBN g8;
T #Beschl_Verz;
g8: NOP 0;

NETWORK
TITLE =Telegramm fuer Parameterkommunikation
//Hier werden die Telegramme für die Parameterkommunikation erstellt.
//Parameter die zum Antrieb geschrieben werden, werden hier vorbereitet. Dabei
//wird berücksichtigt, dass die letzten 4 Ziffern für Nachkommastellen vorgesehen
//sind.
//Die Parameter die eingelesen werden, werden hier aufbereitet und im FB
//gespeichert.
//
//Das Datentyp für den Austausch ist DINT.
//
L W#16#32;
T #Schreibbefehl;

L W#16#31;
T #Lesebefehl;

// Telegramm Geschw Codestelle C1302/1
L #Schreibbefehl;
T #PAR_Telegramm.Geschw.Servicebyte;
L 1;
T #PAR_Telegramm.Geschw.Subindex;
L 24575;
L 1302;
-I ;
T #PAR_Telegramm.Geschw.Index;
L L#100;
L #Geschw;

```



```
ITD ;
*D ;
T #PAR_Telegramm.Geschw.Data;

// Telegramm Geschw_DM Codestelle C1302/2
L #Schreibbefehl;
T #PAR_Telegramm.Geschw_DM.Servicebyte;
L 2;
T #PAR_Telegramm.Geschw_DM.Subindex;
L 24575;
L 1302;
-I ;
T #PAR_Telegramm.Geschw_DM.Index;
L 100;
L #Geschw_DM;
*D ;
T #PAR_Telegramm.Geschw_DM.Data;

// Telegramm Beschl_Verz Codestelle C1303/1
L #Schreibbefehl;
T #PAR_Telegramm.Beschl_Verz.Servicebyte;
L 1;
T #PAR_Telegramm.Beschl_Verz.Subindex;
L 24575;
L 1303;
-I ;
T #PAR_Telegramm.Beschl_Verz.Index;
L 100;
L #Beschl_Verz;
*D ;
T #PAR_Telegramm.Beschl_Verz.Data;

// Telegramm Strecke Codestelle C1301/1
L #Schreibbefehl;
T #PAR_Telegramm.Strecke.Servicebyte;
L 1;
T #PAR_Telegramm.Strecke.Subindex;
L 24575;
L 1301;
-I ;
T #PAR_Telegramm.Strecke.Index;
L 1000;
L #Strecke;
*D ;
T #PAR_Telegramm.Strecke.Data;

// Telegramm Vorschubkonstante Codestelle C1204
L #Lesebefehl;
T #PAR_Telegramm.Vorschubkonstante.Servicebyte;
L 0;
T #PAR_Telegramm.Vorschubkonstante.Subindex;
L 24575;
L 1204;
-I ;
T #PAR_Telegramm.Vorschubkonstante.Index;
L #PAR_Telegramm.Vorschubkonstante.Data;
DTR ;
L 1.000000e+004;
/R ;
T #Vorschubkonstante;

// Telegramm Getriebe_zaeher Codestelle C1202
L #Lesebefehl;
T #PAR_Telegramm.Getriebe_zaeher.Servicebyte;
L 0;
T #PAR_Telegramm.Getriebe_zaeher.Subindex;
L 24575;
L 1202;
```

```

-I ;
T #PAR_Telegramm.Getriebe_zaeher.Index;
L #PAR_Telegramm.Getriebe_zaeher.Data;
DTR ;
L 1.000000e+004;
/R ;
T #Getriebe_zaeher;

// Telegramm Getriebe_nenner Codestelle C1203
L #Lesebefehl;
T #PAR_Telegramm.Getriebe_nenner.Servicebyte;
L 0;
T #PAR_Telegramm.Getriebe_nenner.Subindex;
L 24575;
L 1203;
-I ;
T #PAR_Telegramm.Getriebe_nenner.Index;
L #PAR_Telegramm.Getriebe_nenner.Data;
DTR ;
L 1.000000e+004;
/R ;
T #Getriebe_nenner;

NETWORK
TITLE =Telegrammkommunikation
//Die Parameterwerden nacheinander azyklisch mit dem Antrieb ausgetauscht. Dies
//erfolgt mit Hilfe eines Petrinetzes. Es werden insgesamt sieben Parameter
//ausgetauscht. Die Telegramme sind im Instanzen Datenbaustein hintereinander
//abgelegt.
U #Error_RESET;
SPBN RE;
L P##Petrinetz;
LAR1 ;
L W#16#102;
T DIW [AR1,P#0.0];
SPA end;
RE: NOP 0;

//T1
U #Petrinetz.Merker_1;
UN #Petrinetz.Merker_2;
R #Petrinetz.Merker_1;
S #Petrinetz.Merker_2;
SPBN p1;
L P##PAR_Telegramm;
SRW 3;
T #ACT_TELE_Pointer;
p1: NOP 0;

//T2
U #Petrinetz.Merker_10;
UN #Petrinetz.Merker_11;
R #Petrinetz.Merker_10;
S #Petrinetz.Merker_11;
SPBN p0;
L #ACT_TELE_Pointer;
SLW 3;
LAR1 ;
p0: NOP 0;

//T3
U #Petrinetz.Merker_2;
UN #Petrinetz.Merker_3;
U DIX [AR1,P#0.0];
R #Petrinetz.Merker_2;
S #Petrinetz.Merker_3;

```

//T4

```

U #Petrinetz.Merker_2;
UN #Petrinetz.Merker_3;
U DIX [AR1,P#0.1];
R #Petrinetz.Merker_2;
S #Petrinetz.Merker_3;
SPBN p3;
L DID [AR1,P#4.0];
T #PAR_send.Data;
p3: NOP 0;

```

//T5

```

U #Petrinetz.Merker_3;
UN #Petrinetz.Merker_4;
UN #Petrinetz.Merker_5;
U( ; //Handshakebit vom Antrieb gesetzt?
L #PAR_receive.Servicebyte;
L W#16#40;
UW ;
L W#16#40;
==I ;
) ;
R #Petrinetz.Merker_3;
S #Petrinetz.Merker_4;
S #Petrinetz.Merker_5;
SPBN p4;
L DIB [AR1,P#0.0];
L W#16#BF;
UW ;
T #PAR_send.Servicebyte; // rücksetzte Handshakebit zum Antrieb!
p4: NOP 0;

```

//T6

```

U #Petrinetz.Merker_3;
UN #Petrinetz.Merker_4;
UN #Petrinetz.Merker_7;
U( ; // Handshakebit vom Antrieb nicht gesetzt?
L #PAR_receive.Servicebyte;
L W#16#BF;
OW ;
L W#16#BF;
==I ;
) ;
R #Petrinetz.Merker_3;
S #Petrinetz.Merker_4;
S #Petrinetz.Merker_7;
SPBN p5;
L DIB [AR1,P#0.0];
L W#16#40;
OW ;
T #PAR_send.Servicebyte; // setze Handshakebit zum Antrieb
p5: NOP 0;

```

//T7

```

U #Petrinetz.Merker_4;
UN #Petrinetz.Merker_6;
R #Petrinetz.Merker_4;
S #Petrinetz.Merker_6;
SPBN p6;
L DIB [AR1,P#1.0];
T #PAR_send.Subindex;
L DIW [AR1,P#2.0];
T #PAR_send.Index;
p6: NOP 0;

```

//T8

```

U #Petrinetz.Merker_6;
U #Petrinetz.Merker_5;

```

```

UN #Petrinetz.Merker_9;
U( ; // Handshakebit vom Antrieb nicht gesetzt?
L #PAR_receive.Servicebyte;
L W#16#BF;
OW ;
L W#16#BF;
==I ;
) ;
R #Petrinetz.Merker_6;
R #Petrinetz.Merker_5;
S #Petrinetz.Merker_9;

//T9
U #Petrinetz.Merker_6;
U #Petrinetz.Merker_7;
UN #Petrinetz.Merker_9;
U( ; // Handshakebit vom Antrieb gesetzt?
L #PAR_receive.Servicebyte;
L W#16#40;
UW ;
L W#16#40;
==I ;
) ;
R #Petrinetz.Merker_6;
R #Petrinetz.Merker_7;
S #Petrinetz.Merker_9;

//T10
U #Petrinetz.Merker_9;
UN #Petrinetz.Merker_8;
U DIX [AR1,P#0.0];
R #Petrinetz.Merker_9;
S #Petrinetz.Merker_8;
SPBN p7;
L #PAR_receive.Data;
T DID [AR1,P#4.0];
p7: NOP 0;

//T11
U #Petrinetz.Merker_9;
UN #Petrinetz.Merker_8;
U DIX [AR1,P#0.1];
R #Petrinetz.Merker_9;
S #Petrinetz.Merker_8;

//T12
U #Petrinetz.Merker_8;
UN #Petrinetz.Merker_1;
UN #Petrinetz.Merker_2;
R #Petrinetz.Merker_8;
SPBN m002;
L #ACT_TELE;
L 5;
>I ;
SPBN m001;
S #Petrinetz.Merker_1;
L 0;
T #ACT_TELE;
SPA m002;
m001: SET ;
S #Petrinetz.Merker_2;
L #ACT_TELE;
INC 1;
T #ACT_TELE;
L #ACT_TELE_Pointer;
INC 8;
T #ACT_TELE_Pointer;
m002: NOP 0;

```

```

NETWORK
TITLE =Parameterkommunikation
//Die Datenwoerter fuer die Parameterkommunikation sind im Speicher der SPS
//direkt vor den Prozessdatenwoertern abgelegt.
//Es erfolgt auch hier eine Konsistente Datenübertragung
// T13
  U  #Petrinetz.Merker_11;
  UN #Petrinetz.Merker_10;
  R  #Petrinetz.Merker_11;
  S  #Petrinetz.Merker_10;
  SPBN p8;
  L  #PZD_Adr;
  L  8;
  -I ;
  T  #PAR_Adr;

  CALL "DPRD_DAT" (
    LADDR      := #PAR_Adr,
    RET_VAL    := #return,
    RECORD     := #PAR_receive);

  CALL "DPWR_DAT" (
    LADDR      := #PAR_Adr,
    RECORD     := #PAR_send,
    RET_VAL    := #return);
p8: NOP 0;

end: NOP 0;
END_FUNCTION_BLOCK

```

```

DATA_BLOCK "IDB Folienvorzug"
TITLE =
VERSION : 0.0

```

```

"FB Folienvorzug"
BEGIN
  PZD_Adr := W#16#0;
  Druckmarke_lesen := FALSE;
  Handfahren := FALSE;
  Handfahren_pos := FALSE;
  Handfahren_neg := FALSE;
  POS_START := FALSE;
  POS_Stop := FALSE;
  Error_RESET := FALSE;
  Freigabe := FALSE;
  Geschw := 0;
  Geschw_DM := 0;
  Beschl_Verz := 0;
  Strecke := 0;
  Eingeschaltet := FALSE;
  DRIVE_RDY := FALSE;
  ERROR := FALSE;
  RUNNING := FALSE;
  JOB_FIN := FALSE;
  MCTRL_NACT := 0.000000e+000;
  POS_ACT := 0.000000e+000;
  PAR_Telegramm.Geschw.Servicebyte := B#16#0;
  PAR_Telegramm.Geschw.Subindex := B#16#0;
  PAR_Telegramm.Geschw.Index := W#16#0;
  PAR_Telegramm.Geschw.Data := DW#16#0;
  PAR_Telegramm.Geschw_DM.Servicebyte := B#16#0;
  PAR_Telegramm.Geschw_DM.Subindex := B#16#0;
  PAR_Telegramm.Geschw_DM.Index := W#16#0;
  PAR_Telegramm.Geschw_DM.Data := DW#16#0;
  PAR_Telegramm.Beschl_Verz.Servicebyte := B#16#0;
  PAR_Telegramm.Beschl_Verz.Subindex := B#16#0;
  PAR_Telegramm.Beschl_Verz.Index := W#16#0;

```

```
PAR_Telegramm.Beschl_Verz.Data := DW#16#0;
PAR_Telegramm.Strecke.Servicebyte := B#16#0;
PAR_Telegramm.Strecke.Subindex := B#16#0;
PAR_Telegramm.Strecke.Index := W#16#0;
PAR_Telegramm.Strecke.Data := DW#16#0;
PAR_Telegramm.Vorschubkonstante.Servicebyte := B#16#0;
PAR_Telegramm.Vorschubkonstante.Subindex := B#16#0;
PAR_Telegramm.Vorschubkonstante.Index := W#16#0;
PAR_Telegramm.Vorschubkonstante.Data := DW#16#0;
PAR_Telegramm.Getriebe_zaeher.Servicebyte := B#16#0;
PAR_Telegramm.Getriebe_zaeher.Subindex := B#16#0;
PAR_Telegramm.Getriebe_zaeher.Index := W#16#0;
PAR_Telegramm.Getriebe_zaeher.Data := DW#16#0;
PAR_Telegramm.Getriebe_nenner.Servicebyte := B#16#0;
PAR_Telegramm.Getriebe_nenner.Subindex := B#16#0;
PAR_Telegramm.Getriebe_nenner.Index := W#16#0;
PAR_Telegramm.Getriebe_nenner.Data := DW#16#0;
PZD_receive.Statuswort.Meldung := FALSE;
PZD_receive.Statuswort.Remote := FALSE;
PZD_receive.Statuswort.Sollwert_erreicht := FALSE;
PZD_receive.Statuswort.Grenzwert := FALSE;
PZD_receive.Statuswort.DCTRL_FAIL_QSP := FALSE;
PZD_receive.Statuswort.reserviert_2 := FALSE;
PZD_receive.Statuswort.RUNNING := FALSE;
PZD_receive.Statuswort.POS_ENDED_and_POS_IN_TAR := FALSE;
PZD_receive.Statuswort.Einschaltbereit := FALSE;
PZD_receive.Statuswort.Eingeschaltet := FALSE;
PZD_receive.Statuswort.Betrieb_freigegeben := FALSE;
PZD_receive.Statuswort.Stoerung_TRIP := FALSE;
PZD_receive.Statuswort.Spannung_gesperrt := FALSE;
PZD_receive.Statuswort.Schnellhalt := FALSE;
PZD_receive.Statuswort.Einschaltsperrung := FALSE;
PZD_receive.Statuswort.Warnung := FALSE;
PZD_receive.PZDW_1 := W#16#0;
PZD_receive.PZDW_2 := W#16#0;
PZD_receive.PZDW_3 := W#16#0;
PZD_send.Steuerwort.DRIVECOM_reserviert_1 := FALSE;
PZD_send.Steuerwort.DRIVECOM_reserviert_2 := FALSE;
PZD_send.Steuerwort.DRIVECOM_reserviert_3 := FALSE;
PZD_send.Steuerwort.POS_RESET := FALSE;
PZD_send.Steuerwort.POS_MANUAL := FALSE;
PZD_send.Steuerwort.POS_MANUAL_NEG := FALSE;
PZD_send.Steuerwort.POS_MANUAL_POS := FALSE;
PZD_send.Steuerwort.Modus_DM := FALSE;
PZD_send.Steuerwort.Einschalten := FALSE;
PZD_send.Steuerwort.Spg_sperren := FALSE;
PZD_send.Steuerwort.Schnellhalt := FALSE;
PZD_send.Steuerwort.Betrieb_freigegeben := FALSE;
PZD_send.Steuerwort.HLG_sperren := FALSE;
PZD_send.Steuerwort.POS_START := FALSE;
PZD_send.Steuerwort.POS_STOP := FALSE;
PZD_send.Steuerwort.Reset_Stoerung := FALSE;
PZD_send.PZD_1 := W#16#0;
PZD_send.PZD_2 := W#16#0;
PZD_send.PZD_3 := W#16#0;
PAR_receive.Servicebyte := B#16#0;
PAR_receive.Subindex := B#16#0;
PAR_receive.Index := W#16#0;
PAR_receive.Data := DW#16#0;
PAR_send.Servicebyte := B#16#0;
PAR_send.Subindex := B#16#0;
PAR_send.Index := W#16#0;
PAR_send.Data := DW#16#0;
Petrinetz.Merker_1 := TRUE;
Petrinetz.Merker_2 := FALSE;
Petrinetz.Merker_3 := FALSE;
Petrinetz.Merker_4 := FALSE;
Petrinetz.Merker_5 := FALSE;
```

```
Petrinetz.Merker_6 := FALSE;
Petrinetz.Merker_7 := FALSE;
Petrinetz.Merker_8 := FALSE;
Petrinetz.Merker_9 := FALSE;
Petrinetz.Merker_10 := TRUE;
Petrinetz.Merker_11 := FALSE;
ACT_TELE_Pointer := W#16#0;
ACT_TELE := W#16#0;
Vorschubkonstante := 0.000000e+000;
Getriebe_zaeher := 0.000000e+000;
Getriebe_nenner := 0.000000e+000;
END_DATA_BLOCK

ORGANIZATION_BLOCK OB 1
TITLE = "Main Program Sweep (Cycle)"
VERSION : 0.1

VAR_TEMP
OB1_EV_CLASS : BYTE ;      //Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1 : BYTE ;       //1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY : BYTE ;     //Priority of OB Execution
OB1_OB_NUMBR : BYTE ;    //1 (Organization block 1, OB1)
OB1_RESERVED_1 : BYTE ;  //Reserved for system
OB1_RESERVED_2 : BYTE ;  //Reserved for system
OB1_PREV_CYCLE : INT ;    //Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE : INT ;     //Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE : INT ;     //Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME : DATE_AND_TIME ; //Date and time OB1 started
END_VAR
BEGIN
NETWORK
TITLE =

    CALL "FB Folienvorzug" , "IDB Folienvorzug" ;

END_ORGANIZATION_BLOCK
```

A.3. Petrinetz Parameterkommunikation

