

Verfasser: Markus Kuhnert
Hintere Rottach 7
87439 Kempten

Titel: Kreuztisch

Art der Arbeit: Diplomarbeit

Kurzzusammenfassung:

Eine alte Steuerung um einen Kreuztisch in X - und Y- Position zu verfahren wird modernisiert. Dazu werden bereits vorhandene Komponenten wie der Steuerschrank, Relais, Kabel usw. sortiert und zum Teil wieder verwendet. Der dazugehörige Kreuztisch bleibt bis auf die zwei neuen Synchronmotoren mit geändertem Riemenantrieb und der angepassten Verdrahtung der Endschalter nahezu unverändert.

Neu ist, dass die Motoren und deren Positionsgeber (Absolutwertgeber an der Y-Achse, Inkrementalgeber an der X-Achse) über eine schnelle DRIVE - CLIQ - Verdrahtung mit der Steuerung verbunden sind, welche eine genauere, einfachere und schnellere Kommunikation unter den Teilnehmern zulässt.

Die Steuerung soll geplant, aufgebaut und wenn fertig gestellt eine Kontur automatisch fahren können.

Die Hauptsteuerung übernimmt eine SIMATIC T - CPU von Siemens.

Projektiert wird mit STEP 7 und WinCC flexible, Motion mittels PLC open Bausteinen.

Erscheinungsjahr: 2016



Hochschule Anhalt
Anhalt University of Applied Sciences

emw

Fachbereich
Elektrotechnik, Maschinenbau
und Wirtschaftsingenieurwesen

Diplomarbeit

zur Erlangung des akademischen Grades
Diplom-Ingenieur (Fachhochschule)
Dipl.-Ing. (FH)

Markus Kuhnert

Vorname Nachname

Fernstudium Elektrotechnik, 2004, 4045299

Studiengang, Matrikel, Matrikelnummer

Thema:

**Planung, Aufbau und Inbetriebnahme einer
Steuerung für einen Kreutztisch mit modernen
Steuerungs- und Antriebskomponenten vom
Fabrikat Siemens.**

Prof. Dr. - Ing. Dr. h. c. H. Biechl

1. Prüfer(in)

Prof. Dr. - Ing. Jürgen Schwarz

2. Prüfer(in)

10.10. 2016

Abgabe am

Erklärung des Studenten:

Hiermit erkläre ich, dass die vorliegende Diplomarbeit selbstständig verfasst, in gleicher oder ähnlicher Fassung noch nicht in einem anderen Studiengang als Prüfungsleistung vorgelegt wurde und keine anderen als die angegebenen Hilfsmittel und Quellen, einschließlich der angegebenen oder beschriebenen Software, verwendet wurden.

Kempton, den 27.08.2016

Vorwort

Herr Prof. Dr. - Ing. Dr. h. c. H. Biechl, tätig an der Hochschule Kempten, bot mir auf meine Bewerbung hin das Thema für die Erstellung dieser Diplomarbeit an. Nachdem ich bereits eine Ausbildung zum Energieanlagenelektroniker bei den Stadtwerken Lindau absolvierte, fand ich dieses Thema mit einer SPS Steuerung sehr interessant. Bisher wurde die SPS nur am Rande angesprochen, und so freut es mich nun um so mehr, mich in die Materie einzuarbeiten.

An der Hochschule Kempten war ich bis zum Vordiplom eingeschrieben, und machte im selbigen Labor für elektrische Antriebe, in welchem ich heute meine Diplomarbeit erarbeite, bereits ein Praktikum. Herr Prof. Dr. - Ing. Dr. h. c. H. Biechl leitete damals die Versuche, wobei ich seine Vorlesungen Elektrotechnik Eins und Zwei besuchen durfte, und ihn seit daher persönlich kenne.

Ich danke Herrn Prof. Biechl, dass er mir diese Diplomarbeit ermöglichte, meinem Betreuer Herrn Dipl.- Ing. Geist für die hilfreichen Ratschläge und meinem geduldigen Werkstattleiter Herrn Heiland, welcher mir die Räumlichkeiten frei gehalten hatte, und täglich den Arbeitsplatz elektrisch freigeschalten hatte und dies zeitlich mit mir abgestimmt hatte.

Inhaltsverzeichnis

	Seite
1. Einleitung	1
1.1 Hochschulprofil	4
1.2 Laborprofil	5
1.3 Laborlageort, Versuchsort	6
2. Erste Arbeitsschritte	7
2.1 Entkernung	7
2.2 Einarbeitung in die STEP 7 von Siemens	7
2.3 Planung der neuen Steuerung	7
2.4 Verwendete Hardware	9
2.4.1 Das Programmiergerät	9
2.4.2 Montage und Verdrahtung der Hardware im Schaltschrank	9
2.4.3 Zentralbaugruppe für SPS-Technologieaufgaben	9
2.4.4 SITOP Modular 10 DC 24V Stromversorgung	13
2.4.5 Netzfilter und Netzdrossel	13
2.4.6 Control Unit CU320-2 DP	13
2.4.7 Smart Line Module	16
2.4.8 Double Motor Module	18
2.4.9 Antriebsmotoren	21
2.5 Verwendete Software	23
3. Inbetriebnahme Schwerpunkt "Handbetrieb"	24
3.1 Anlegen des S7 Projektes	24
3.2 Projektierung in der Hardware Konfiguration	24
3.3 Projektierung der Antriebskomponenten	24
3.4 Antriebstest über die Antriebssteuertafel des SINAMICS	25
3.5 Technologie-Datenbausteine anlegen	25
3.6 Visualisierung mit WinCC-Flexible	26
3.7 Entwicklung der Handsteuerung	26
3.7.1 Eigenschaften eines Templates	26
3.7.2 Integration des Technologie Templates	28
3.7.3 Zuordnung des Instanzdatenbausteins	28
3.7.4 Einbindung der PLC-Open Bausteine	28

3.7.5	Öffnen des OB1, Erstellen des Anwenderprogrammes "Handbetrieb"	29
3.7.6	Bedienung im Handbetrieb	29
4.	Inbetriebnahme Schwerpunkt "Automatikbetrieb"	30
4.1.1	Hardware-Endschalter Signale in die T-CPU einlesen	30
4.1.2	Referenziermöglichkeiten	31
4.1.3	Aktives Referenzieren	32
4.1.4	Hardwareaufbau, Referenzierung gespiegelte externe Nullmarke	32
4.1.5	Nullmarkenfehler, Fehlersuche	33
4.1.6	Fehlersuche	33
4.1.7	Startposition	36
4.2	Einbindung des Templates "MotionList Basic"	39
4.2.1	Eigenschaften des Templates "MotionListBasic"	39
4.2.2	Eigenschaften der Verfahrkontur	41
4.2.3	Festlegung eines geeigneten Koordinatensystems	42
4.2.4	Programmierung von Konturelementen	43
4.2.5	Absolute oder relative Programmierung, Endposition	43
4.2.6	Absolute und relative Programmierung, Gerade	43
4.2.7	Absolute und relative Programmierung, Kreisbogen	44
4.2.8	Absolute und relative Programmierung, Kreis	44
4.2.9	Speicherung einer Verfahrkontur	45
4.3	Abarbeiten einer Verfahrkontur	46
4.3.1	Verkettung der Verfahrkonturen	46
4.3.2	Konfiguration und Projektierung	46
4.3.3	Schnittstellen des FB540 "MotionList_Basic"	47
4.3.4	Erstellen des Anwenderprogramms	48
4.3.5	Bedienung im Automatikbetrieb	49
5.	Zusammenfassung und Ausblick	51
6.	Literaturverzeichnis	52
7.	Anlagen	54

Abkürzungsverzeichnis

Ω	Ohm
A	Ampere
AC	Wechselspannung
Bit	binary digit
Byte	= 8 Bit
CPU	Central Processing Unit
DB	Datenbaustein
DC	Gleichstrom
DI	Digitaler Eingang
DO	Digitaler Ausgang
DRIVE-CLIQ	Siemens Schnittstelle mit einer Übertragungsrate von 100 Mbit/s
E	Eingang
E/A	Eingang / Ausgang
FU	Frequenzumrichter
LED	Leuchtdiode
mm	Millimeter
OB1	Baustein in welchem die Anwendungssoftware hinterlegt ist, Organisationsbaustein genannt
P	elektrische Leistung
PC	Personal Computer
PLCopen	Zusammenschluss führender Steuerungshersteller
R	elektrischer Widerstand
SIMATIC	Bezeichnung für die T-CPU
SINAMICS	Bezeichnung für das Antriebssystem
SPS	Speicher programmierbare Steuerung
Step7	Software zur Programmierung der SIMATIC S7
S7T	Tool zur Parametrierung und Programmierung der Technologieobjekte
UDT	user defined type, Anwenderdefinierter Datentyp
V	elektrische Spannung (Volt)

1. Einleitung

Die Diplomarbeit, für die diese Dokumentation erstellt wurde, beruht darauf, dass eine vorhandene Steuerung des Kreuztisches veraltet ist, bzw. nicht mehr funktioniert.

Daher hat man sich im Labor für elektrische Antriebe der HS Kempten dafür entschieden diese vorhandene Steuerung mit neuen Steuerungskomponenten umzurüsten und diese dann für weitere Lehrmaßnahmen des Labors zur Verfügung zu stellen. Dies wurde dann zur Diplomarbeit ausgeschrieben, welche ich dann auf Anfrage erhalten habe. Der Kreuztisch selbst ist bis auf die Antriebsmotoren und dessen Positionsgeber unverändert geblieben.

"1.1 Hochschulprofil



Gründung 1977

Trägerschaft Staatlich

Ort Kempten (Allgäu)

Bundesland Bayern

Land Deutschland

Präsident Robert F. Schmidt

Studenten 6.257 WS 2016/17

Mitarbeiter ca. 330 (davon ca. 130 Professoren, ca. 200 nicht-Wissenschaftler" [1]

[1], Quelle: Wikipedia: Freie Enzyklopädie, Sitz in San Francisco, USA.

1.2 Laborprofil

**"IEES (Institut für elektrische Energiesysteme) und dem Bereich:
IRENE (Integration regenerativer Netze und Elektromobilität)**

**Anwendungsorientierte Forschung auf dem Gebiet der elektrischen
Energiesysteme**

- Modellbildung, Simulation, Regelung und Optimierung von Microgrids mit Batteriespeichern, FACTS und regenerativen Energiequellen
- Mathematische Modellbildung, Simulation und Optimierung von Komponenten und Systemen in der elektrischen Antriebstechnik und Mechatronik
- Numerische Magnetfeld-Berechnung
- Rechnergestützter Entwurf von Sondermaschinen
- Netzurückwirkungen von elektrischen Antrieben
- Prüfung von elektrischen Maschinen und Antrieben" [2]

Laborleiter:

Prof. Dr.-Ing. Dr. h.c. Helmuth Biechl
Telefon: (08 31) 25 23-253
Telefax: (08 31) 25 23-571
eMail: biechl@hs-kempten.de

[2], Quelle: Hochschule Kempten, Laborbeschreibung.

1.3 Laborort, Versuchsort:

Raum: T135a

Bahnhofstraße 61

87435 Kempten

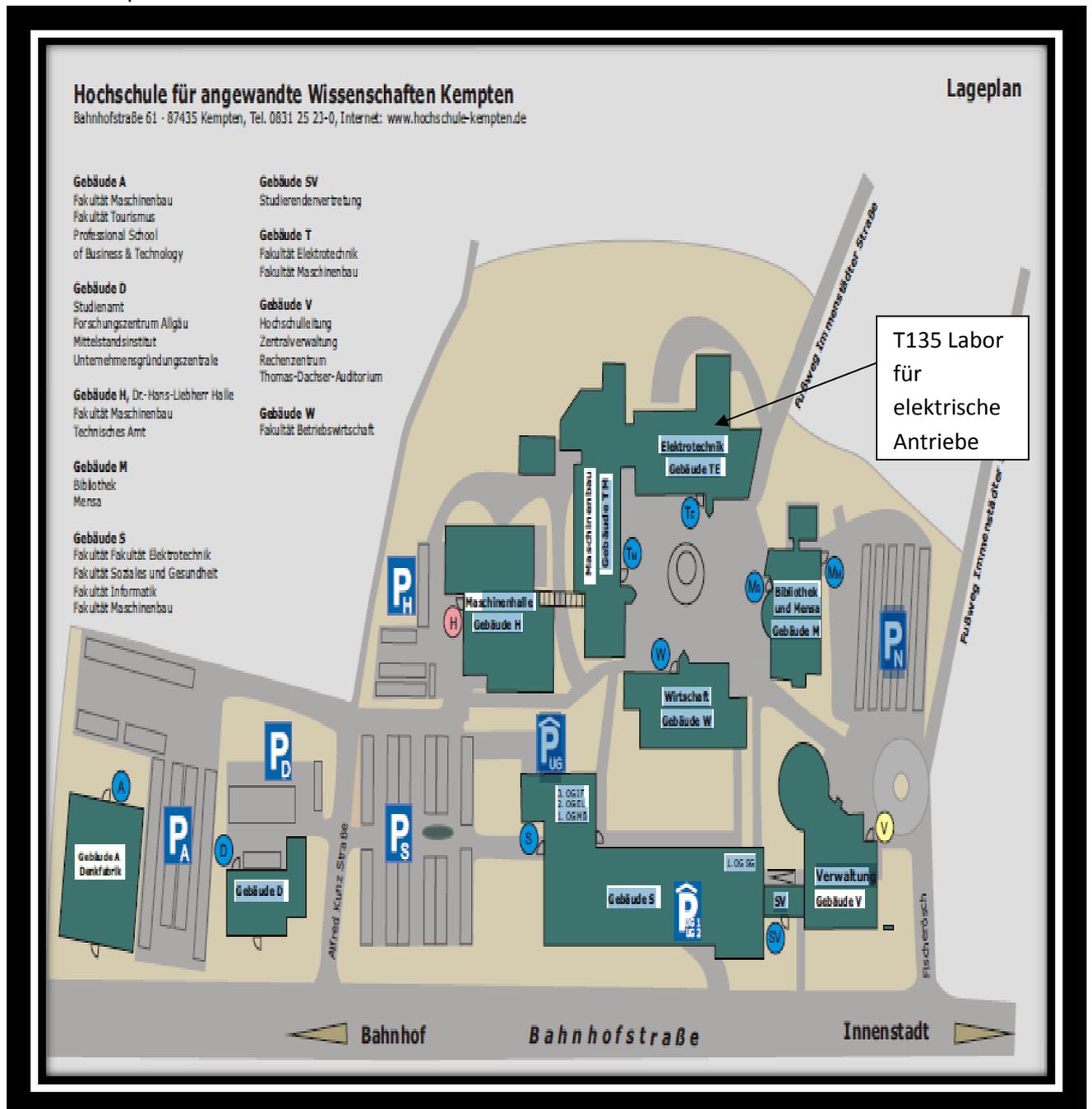


Abb. 2.0: Montageort des Kreuztisches an der HS Kempten, [2],Quelle: vgl. Hochschule Kempten, Lageplan.

2. Erste Arbeitsschritte

2.1 Entkernen und Sortieren

Da der bisherige Schaltschrank sowie einige darin befindliche Bauteile wieder benützt werden, wird der Schaltschrank von mir entkernt, und noch zu gebrauchende Teile aussortiert, alte Steuergeräte entsorgt.

2.2. Einarbeitung in die STEP 7 von Siemens

Die aktuelle Softwaresprache für komplexe Steuerungen bei Siemens ist momentan die S7. Um mich in das Themengebiet der speicherprogrammierten Steuerungen der S7 von Siemens einzufinden, baue ich anhand eines Übungsgerätes von der HS Kempten eine einfache Schaltung auf, mit der die Konfigurationsschritte erläutert werden. Begleitet wird diese Übung durch das Skript von Siemens, SIMATIC, Erste Schritte und Übungen mit STEP 7, Getting Started, Diese Dokumentation ist Bestandteil des Dokumentationspaketes. [3]

2.3 Planung der neuen Steuerung

Aktuelles Siemens Lehrmaterial ist bei Siemens fast ausschließlich online unter <http://support.automation.siemens.com> zu bekommen. Dort finde ich im Inbetriebnahmehandbuch (IH1) einen schematischen Plan, welcher einen möglichen Aufbau für meine Anforderungen beinhaltet. Für meine Anforderungen wird eine Steuerung benötigt, mit welcher zwei Achsen unabhängig voneinander verfahren werden können, gleichzeitig sollte diese Steuerung in Anlehnung an die alte Steuerung eine Handsteuerung und Automatiksteuerung betreiben können. Auch vereinfacht es sich für mich, bestimmte erste Funktionsfahrten im Handbetrieb zu fahren. Dieser zuvor erwähnte Plan dient mir als Grundlage für meinen Kreutztisch. In diesem Fall wird die Art der Verschaltung und der dazugehörenden Komponenten bei Siemens "SINAMICS S120" genannt. Im Plan ist ersichtlich wie und an welcher Position z. B.: Netzfilter, Netzdrossel, FU-CPU, DRIVE-CLiQ Verbindungen, Motoren, Laststrom, DC-Versorgungsspannung, Smart Line Module, Motor Module zu montieren sind. Siehe Bild 2.1 unten.

Diese Bauteile wurden bei der Siemens AG Geschäftsstelle Kempten mit deren Rücksprache geordert.

Als Hauptsteuerung kommt eine T-CPU zum Einsatz, diese besteht aus einer normalen S7 - CPU und einem Technologieteil. Diese T-CPU wird über einen PROFIBUS-DP (Drive) mit den SINAMICS S120 - Geräten verbunden, und wird über einen separaten PROFIBUS an das Programmiergerät angeschlossen. Siehe Bild 2.2 unten.

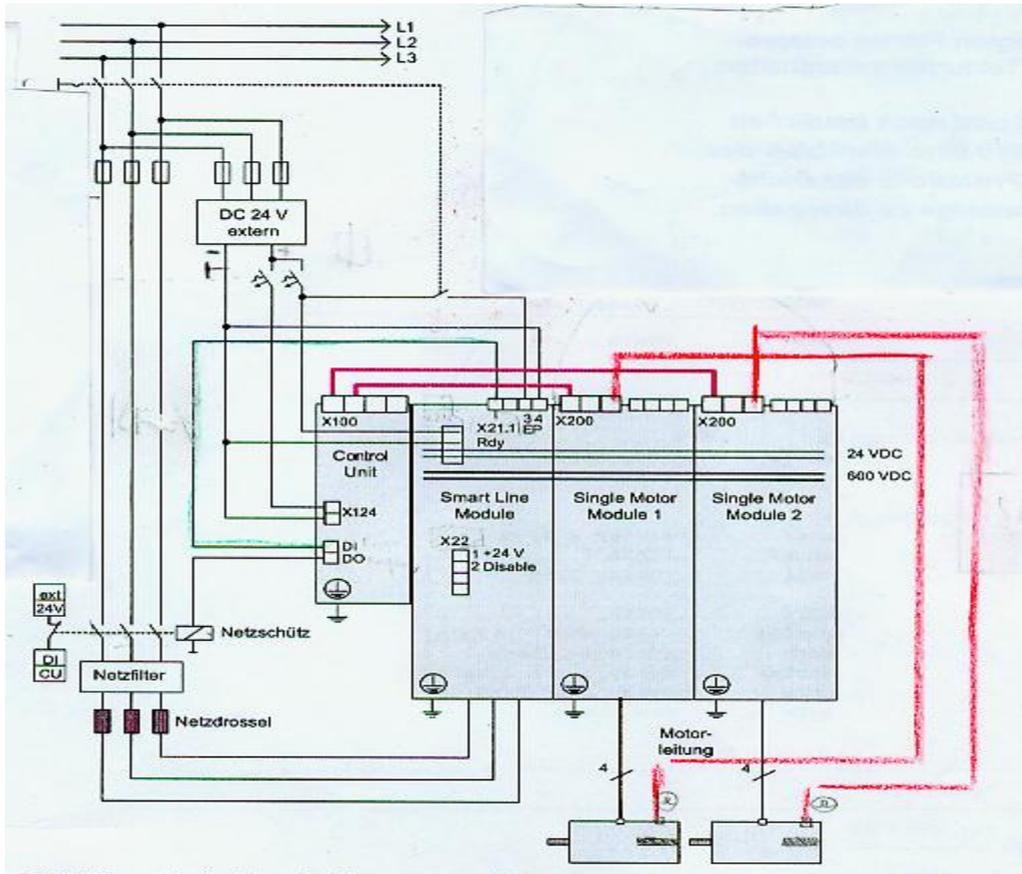


Abb. 2.1: möglicher Aufbau,
 [4], Quelle: vgl. Siemens AG: SINAMICS S120, Inbetriebnahmehandbuch, (IH1), 01/2011,
 6SL3097-4AF00-0AP1. S. 90.

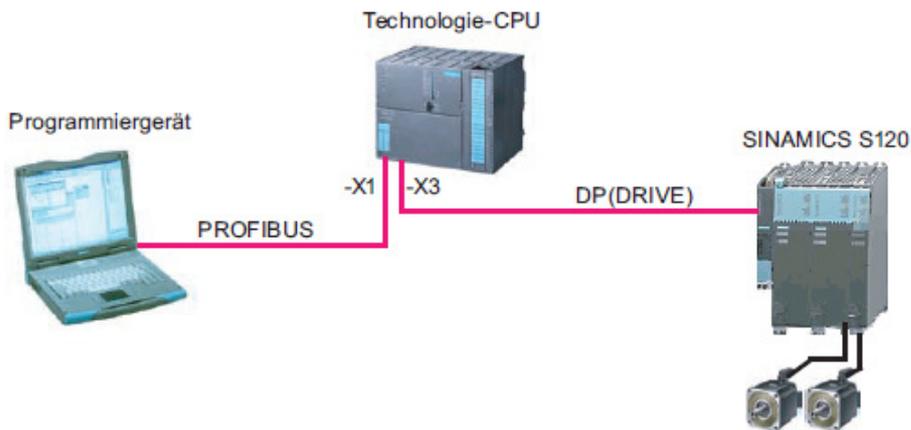


Abb. 2.2: Verbindung der Komponenten mittels Profibus,
 [5], (Quelle: Siemens AG: SIMATIC, S7-300, Anbindung SINAMICS S120 an die
 Technologie-CPU, Produktinformation, 09/2011, S. 15.

2.4 Verwendete Hardware

2.4.1 Das Programmiergerät

Als Programmiergerät dient ein Pentium IV Computer mit 2 GB Arbeitsspeicher und 1,2 GHz Prozessor mit 500 GB Festplatte. Betriebssystem ist XP Professional. Die Profibus Schnittstelle der T-CPU wird über einen USB-Ausgang des Computers mit dem Siemens SIMATIC S7 PC USB Adapter (6ES7 972-0CB20-0XA0) verbunden.

2.4.2 Montage und Verdrahtung der Hardware im Schaltschrank

Für die Montage der ganzen Leistungsbauteile im Booksize Format, Schraubschmelzsicherungen, 24V DC Stromversorgung, und Relais, welche auf der Hutschiene montiert sind, verbreitere ich das obere Drittel der Rückwand des Schaltschranks mit zusätzlichen Aluminiumblechen, welche ich mit 5 mm Ø Blindnieten anniete, und bringe in geeigneter Position die Hutschienen mit einem M5 Gewinde direkt auf die Blechrückwand an, welche aus einem 2,5 mm starkem, verzinkten Blech im unteren Zweidrittel, und im oberen Drittel aus 2,5 mm starkem Aluminiumblech besteht. Für die Hutschienenmontage werden die Gewindelöcher direkt in das Blech gebohrt und anschließend mit M5 Gewindebohrer geschnitten. Die Übersicht der Integration in der kompletten Steuerung, sowie des Kreuztisches ist im Anhang Abb. 2.4.2.1 - 2.4.2.3 zu sehen. Anschließend werden die Bauteile nach selbst erstelltem Verdrahtungsplan verkabelt und angeschlossen. Siehe Abbildung 2.4.2.4 - 2.4.2.8. Für den Verdrahtungsplan dient mir der obige ähnliche schematische Verdrahtungsplan Abb. 2.1 als Grundlage, welchen ich für meine Anforderungen noch erweitere. Am Kreuztisch werden die vier Endschalter der X- und Y-Achse, um eine Drahtbruchsicherheit zu gewährleisten noch als Öffner umverdrahtet. Angeschlossen werden die Endschalter über die graue Steuerleitung nach Abbildung 2.4.2.8 im Anhang, und wird am Schaltschrankfrontblech mit einem Flachstecker verbunden. An der Schaltschrankvorderseite befinden sich auch die beibehaltene Schalter für Handbetrieb in X- und Y-Position, Notaus, Hauptdreiphasennetzschalter, Freigabe Antrieb über das Lastralais K1, einzelne Freischaltung der beiden Achsen, Schlüsselschalter zwischen Hand- und Automatikbetrieb, und Temperaturkontrolllampe des Smart Line Moduls, welche alle neu verdrahtet wurden.

2.4.3 Zentralbaugruppe für SPS - Technologieaufgaben

Die T-CPU (CPU 315T-2 DP) besitzt vier integrierte schnelle Digitaleingänge, worauf bei der Inbetriebnahme (Referenzierung) noch weiter darauf eingegangen wird. Und besitzt acht weitere integrierte Digitalausgänge. Diese Eingänge reichten leider nicht aus, weshalb sie noch durch zwei weitere Steckkarten SM323 (8DI/8DO) und SM321 (16DI) erweitert wurde. Die Technologiefunktionen werden direkt über die T-CPU programmiert und parametrierbar, sie ist das Herzstück des kompletten Steuerungsaufbau, und steuert über den PROFIBUS-DP (Drive) die FU-CU. Sie wird auf einer speziellen Siemens Montageschiene montiert und

ebenfalls wie die anderen Leistungsbauteile damit direkt auf die Rückwand verschraubt. Die Montageansicht ist im Anhang Abb. 2.4.2.1 zu sehen.

Der SPS Programminhalt wird über das S7 Programm programmiert, die Technologie wird über ein integriertes dort zusätzlich geladenes Optionspaket S7T ebenfalls über das S7 Programm programmiert, parametrieren, und wird auch im S7 Programmmanager separat geöffnet.

"Übersicht:

- SIMATIC CPU mit integrierter Technologie-/Motion Control-Funktionalität
- Mit voller Funktionalität der Standard-CPU 315-2 PN/DP (außer CBA)
- Für branchenübergreifende Automatisierungsaufgaben im Serienmaschinen-Sondermaschinen- und Anlagenbau
- Ideal für synchronisierte Bewegungsabläufe wie Kopplung zu einem virtuellen/realen Master, Getriebegleichlauf, Kurvenscheibe, Bahninterpolation oder Druckmarkenkorrektur
- 3-D-Bahninterpolation mit verschiedenen Kinematiken
- Lage- und druckgeregelte Hydraulikachsen
- Einsatz als zentrale Steuerung in Fertigungslinien mit zentraler und dezentraler Peripherie
- Mit integrierter Peripherie für schnelle technologische Funktionen (z.B. Nockenschalten, Referenzpunkterfassung)
- PROFIBUS DP (DRIVE)-Schnittstelle für den taktsynchronen Anschluss von Antriebskomponenten
- PROFINET-Schnittstelle mit 2-Port-Switch
- PROFINET I/O-Controller für den Betrieb von dezentraler Peripherie an PROFINET
- Ein gemeinsames S7-Anwenderprogramm für Steuerungs- und Motion Control-Aufgaben (keine zusätzliche Programmiersprache für Motion Control erforderlich)
- Optionspaket „S7-Technology“ erforderlich (ab Version V4.2 SP3)
SIMATIC Micro Memory Card (8 Mbyte) zum Betrieb der CPU erforderlich.

Anwendungsbereich

Die CPU 315-3 PN/DP kommt in Maschinen zur Anwendung, bei denen eine Steuerung mit großem Programmspeicher und hoher Verarbeitungsgeschwindigkeit und gleichzeitig Motion Control-Anforderungen aufeinander treffen, z. B.:

- Verarbeitungs-/Montagelinien
- Kartonaufrichter

- Palettierer
- Handlingsysteme
- Fliegende Schere
- Abfüllanlagen
- Einschlagen/Einwickeln
- Etikettiermaschinen
- Walzenvorschub
- Portale mit Interpolation

Es können max. acht Achsen angesteuert werden. Neben genauer Einzelachspositionierung eignet sich die Baugruppe vor allem für komplexe Bewegungsabläufe, z.B. Kopplung zu einem virtuellen/realen Master, Getriebegleichlauf, Kurvenscheibe, Bahninterpolation oder Druckmarkenkorrektur.

Die CPU 315T-3 PN/DP eignet sich durch ihr Mengengerüst optimal für den Einsatz von SIMATIC Engineering Tools, z. B.:

- Programmierung mit SCL
- Schrittkettenprogrammierung mit S7-GRAPH

Aufbau

Die CPU 315T-3 PN/DP besitzt die volle Funktionalität der leistungsstarken CPU 315 sowie zusätzlich integrierte Funktionen für Technologie/Motion Control. Integrierte, schnelle E/A für technologische Funktionen wie Nockenschalten oder Referenzpunkterfassung runden die Baugruppe ab.

Die CPU 315T-3 PN/DP verfügt über:

- Mikroprozessor;
der Prozessor erreicht eine Bearbeitungszeit von ca. 0,05 μ s je Binäranweisung und 0,45 μ s je Gleitpunktoperation. Insbesondere bei Wort- und Doppelwortbefehlen sowie bei 32 bit-Festpunktbefehlen erreicht die CPU 315T-3 PN/DP sehr hohe Bearbeitungsgeschwindigkeiten.
- 384 Kbyte Arbeitsspeicher (entspricht etwa 128 K Anweisungen);
Der umfangreiche Arbeitsspeicher für ablaufrelevante Programmteile bietet Anwenderprogrammen ausreichend Platz. SIMATIC Micro Memory Cards (8 Mbyte) als Ladespeicher für Programm ermöglichen zusätzlich die Projektablage (inklusive Symbolik und Kommentaren) in der CPU und können zur Datenarchivierung und Rezepturverwaltung verwendet werden.
- Integrierte Ein-/Ausgänge
4 Digitalein- und 8 Digitalausgänge, nutzbar für technologische Funktionen, z.B. Referenzpunkterfassung (BERO) oder schnelle Nockenschaltensignale. Die Digitaleingänge lassen sich auch (eingeschränkt) im STEP 7-Anwenderprogramm einsetzen.
- Flexible Ausbaubarkeit;
max. 8 Baugruppen (einzeiliger Aufbau)

- **Kombinierte MPI/DP-Schnittstelle;**
Die erste integrierte Schnittstelle MPI/DP kann max. 32 Verbindungen gleichzeitig zu S7-300/400 oder Verbindungen zu PG, PC, OP aufbauen. Von den Verbindungen ist jeweils eine für PG und eine für OP fest reserviert. Mit der MPI ist es möglich, über "Globale Datenkommunikation" eine einfache Vernetzung mit bis zu 32 CPUs aufzubauen.
Diese Schnittstelle kann von einer MPI- zur PROFIBUS DP-Schnittstelle umprojektiert werden. Die DP-Schnittstelle kann als DP-Master oder als DP-Slave genutzt werden.
- **PROFIBUS DP (DRIVE)-Schnittstelle:**
Die PROFIBUS DP (DRIVE)-Schnittstelle unterstützt Taktsynchronität und bietet somit eine wesentliche Voraussetzung für die Beherrschung schneller und zeitkritischer Vorgänge, z.B. bei verteilten Achsen in Gleichlaufanwendungen. Die Schnittstelle kann nur als DP-Master eingesetzt werden. Sie dient zum Anschluss von Antriebskomponenten. Neben Antriebssystemen können auch noch DP-V0 Slaves eingeschränkt am DP (DRIVE)-Strang betrieben werden. Die anschließbaren Antriebe sind in den technischen Daten spezifiziert.
Eine Liste mit den am DP-Drive betreibbaren Antrieben finden Sie unter:
<http://support.automation.siemens.com>
- **PROFINET**
Die dritte integrierte Schnittstelle der CPU 315T-3 PN/DP ist eine PROFINET-Schnittstelle mit 2-Port-Switch, basierend auf Ethernet TCP/IP. Sie unterstützt folgende Protokolle:
 - S7-Kommunikation für den Datenaustausch zwischen SIMATIC-Steuerungen
 - PG-/OP-Kommunikation für Programmierung, Inbetriebnahme und Diagnose über STEP 7
 - PG-/OP-Kommunikation für Anbindung an HMI und SCADA
 - Offene TCP/IP-, UDP- und ISO-on-TCP (RFC1006)-Kommunikation über PROFINET
 - SIMATIC NET OPC-Server für die Kommunikation mit anderen Controllern und Peripheriegeräten mit eigener CPU

Funktion

Die CPU 315T-3 PN/DP verfügt über die volle Funktionalität der CPU 315-2 PN/DP (außer CBA). Zusätzlich stehen leistungsfähige Motion Control-Funktionen zur Verfügung.

Die Programmierung erfolgt mit STEP 7 sowie dem Optionspaket S7-Technology." [6]

[6], Quelle: Siemens AG, Produktbeschreibung, T-CPU allgemein.

2.4.4 Netzfilter und Netzdrossel

Für das 5 KW Smart line Motormodule wurden Netzdrossel und Netzfilter mitgeliefert. Da das Smart Line Modul sowohl Leistung aufnehmen, als auch Leistung wieder ins Versorgungsnetz zurückspeisen kann, werden durch diese Schutzeinrichtungen beide Seiten, Netz- als auch das Smart Line Modul vor elektrischen Störungen geschützt. Sie werden beide separat auf dem Schaltschrankboden verschraubt.

2.4.5 SITOP Modular 10 DC 24V Stromversorgung

Um die SPS sowie Kontrollleuchten, und die weiteren Geräte mit Niederspannung zu versorgen wird eine geregelte Standard Siemens Stromversorgung auf einer Hutschiene montiert. Sie liefert 24 V Ausgangsgleichspannung und ist bis zu 10 Ampere belastbar.

2.4.6 Control Unit CU320-2 DP

Um die SINAMICS S120 Geräte anzusteuern ist der T-CPU die FU-CU untergeordnet. Sie regelt das Smart Line Module sowie das Double Motor Module, welches letztendlich die beiden Synchronmotoren ansteuert. Sie besitzt einen Steckplatz für eine Compactflash-Card, welche alle Dateien beinhaltet um den parametrisierten Antrieb auch auf einer anderen CU betreiben zu können, falls diese erneuert werden sollte. Die Technologiefunktionen werden direkt über die T-CPU programmiert und parametrisiert. Sie wird auf der Hutschiene montiert.

"In der Control Unit CU320-2 laufen die Kommunikations-, Steuerungs- und Regelungsfunktionen für die Power Modules ab. Die Kommunikation zur übergeordneten Steuerung erfolgt entweder über PROFIBUS DP oder PROFINET.

Die Control Unit CU320-2 und die zugehörige CompactFlash Card können entweder einzeln oder als Control Unit Kit bestellt werden. Das Control Unit Kit besteht aus der Regelungsbaugruppe CU320-2, einer CompactFlash Card und der darauf hinterlegten Antriebs-Software.

Aufbau:

Die Control Unit CU320-2 hat standardmäßig folgende Anschlüsse und Schnittstellen:

- CU320-2 PN: 1 PROFINET-Schnittstelle mit zwei Ports (RJ45-Buchsen) mit Profil PR OFIdrive V4
- CU320-2 DP: 1 PROFIBUS-Anschaltung mit Profil PROFIdrive V4
- 4 DRIVE-CLiQ-Buchsen, welche die Kommunikation zu den weiteren und so DRIVE-CLiQ-Teilnehmern ermöglichen, z. B. Power Modules, Terminal Modules
- 12 parametrierbare Digitaleingänge (potenzialfrei)

- 8 parametrierbare bidirektionale Digitaleingänge/-Ausgänge (nicht potenzialfrei)
- 1 serielle RS232-Schnittstelle
- 1 Schnittstelle für das Basic Operator Panel BOP20
- 1 Slot für die CompactFlash Card, auf der Firmware und Parameter gespeichert sind
- 1 Slot zum Einbau einer Optionsbaugruppe für Schnittstellenerweiterung
- 2 Dreh-Codierschalter zum manuellen Einstellen der PROFIBUS-Adresse
- 1 Ethernet-Schnittstelle zur Inbetriebnahme und Diagnose
- 3 Messbuchsen und eine Bezugsmasse für die Unterstützung der Inbetriebnahme
- 1 Anschluss für die Elektronikstromversorgung über den DC-24-V-Einspeisestecker
- 1 PE/Schutzleiter-Anschluss
- 1 Masseanschluss

An der Control Unit CU320-2 befindet sich eine Schirmauflage für den Signalleitungsschirm der Optionsbaugruppe.

Der vorhandene Option-Slot dient der Erweiterung der Schnittstellen, z. B. Klemmenanzahl.

Der Status der Control Unit CU320-2 wird über mehrfarbige LEDs angezeigt. Die Firmware und die eingestellten Parameter befinden sich auf einer steckbaren CompactFlash Card und unterstützen so den Tausch der Control Unit ohne Hilfsmittel.

Integration:

Die Kommunikation zwischen einer Control Unit CU320-2 und den angeschlossenen Komponenten erfolgt über DRIVE-CLiQ.

Zu Diagnosezwecken kann das Basic Operator Panel BOP20 auch während des Betriebs auf die Control Unit CU320-2 aufgeschnappt werden.

Inbetriebnahme und Diagnose der Control Unit CU320-2 sowie der angeschlossenen Komponenten werden mit dem Inbetriebnahme-Tool STARTER und installiertem SINAMICS Support Package durchgeführt.

Technische Daten:

- Strombedarf max. bei DC 24 V, typ. ohne Berücksichtigung der Digitalausgänge	1 A
-Anschlussquerschnitt, max.	2,5 mm ²
-Absicherung, max.	20 A
-Digitaleingänge entsprechend IEC 61131-2 Typ 1	12 potenzialfreie Digitaleingänge 8 bidirektionale nicht potenzialfreie Digitaleingänge/-ausgänge

-Spannung	-3 ... +30 V
-Low-Pegel (ein offener Digitaleingang wird als „Low“ interpretiert)	-3 ... +5 V
-High-Pegel	15 ... 30 V
-Stromaufnahme bei DC 24 V, typ.	9 mA
-Verzögerungszeit der Digitaleingänge, etwa:	
L → H	50 µs
H → L	100 µs
-Verzögerungszeit der schnellen Digitaleingänge, etwa:	
L → H	5 µs
H → L	50 µs
-Anschlussquerschnitt, max.	1,5 mm ²
-Digitalausgänge (Dauerkurzschlussfest):	8 bidirektionale nicht potenzialfreie Digitalausgänge/-eingänge
-Spannung	DC 24 V
-Laststrom pro Digitalausgang, max.	500 mA
-Verzögerungszeit, typ./max.	
L → H	150/400 µs
H → L	75/100 µs
-Anschlussquerschnitt, max.	1,5 mm ²
-Verlustleistung	24 W
-PE-Anschluss	Schraube M5
-Masse-Anschluss	Schraube M5

Maße:

Breite	50 mm
Höhe	300 mm
Tiefe	226 mm

Gewicht, etwa	2,3 kg
Approbationen, gemäß	cULus" [6]

2.4.7 Smart Line Module

Das Smart Line Module ist für 5 kW Ausgangsleistung ausgelegt, und liefert für das Double Motor Module die benötigte Ausgangsspannung von 600V DC. Es ist von der Ausgangsleistung das kleinste Smart Line Module, und wird in der Bauform Booksize, ebenfalls auf der Hutschiene montiert geliefert. Das Smart Line Module wird über drei Phasen mit der 380V AC Netzspannung versorgt, und kann bei entsprechender Konfiguration auf Rückspeisung in das Versorgungsnetz eingestellt werden. Dann würde z. B. das Abbremsen eines Motors soviel Energie erzeugen, dass diese wieder ins Versorgungsnetz eingespeist werden kann. Dies kann bei einem größeren Industrieunternehmen mit mehreren Antriebskomponenten dieses Typs durchaus finanzielle Vorteile bieten.

"Übersicht:

Die Smart Line Modules sind kippsichere, netzgeführte Einspeise-/Rückspeiseeinheiten (Diodenbrücke für die Einspeisung; kippsichere, netzgeführte Rückspeisung über IGBTs) mit 100 % Rückspeisedauerleistung. Die Rückspeisefähigkeit der Modules kann über einen Digitaleingang (Smart Line Modules 5 kW und 10 kW) bzw. per Parametrierung (Smart Line Modules 16 kW, 36 kW und 55 kW) deaktiviert werden. Smart Line Modules sind für den Anschluss an geerdete TN-/TT- und ungeerdete IT-Netze geeignet.

Die Vorladung des Zwischenkreises erfolgt über integrierte Vorladewiderstände.

Zum Betrieb des Smart Line Modules ist die zugehörige Netzdrossel zwingend erforderlich.

Aufbau:

Die Smart Line Modules Bauform Booksize haben standardmäßig folgende Anschlüsse und Schnittstellen:

- 1 Netzanschluss über Schraubklemmen
- 1 Anschluss für die DC-24-V-Elektronikstromversorgung über den im Lieferumfang enthaltenen 24-V-Klemmenadapter
- 1 Zwischenkreisanschluss über integrierte Zwischenkreisschienen
- 2 PE/Schutzleiter-Anschlüsse
- 2 Digitaleingänge
(nur bei Smart Line Modules 5 kW und 10 kW)

[6], Quelle: Siemens AG, Produktbeschreibung, CU 320- 2DP.

- 1 Digitalausgang
(nur bei Smart Line Modules 5 kW und 10 kW)
Der Status der Smart Line Modules wird über zwei mehrfarbige LEDs angezeigt.
Der Signalleitungsschirm kann mittels einer Schirmanschlussklemme an dem Line Module angelegt werden, z. B. Typ KLBÜ 3-8 SC der Fa. Weidmüller.

Technische Daten:

-Netzspannung (bis 2000 m über NN)	3 AC 380 ... 480 V ± 10 % (im Betrieb -15 % <1 min)
-Netzfrequenz	47 ... 63 Hz
-SCCR (Short Circuit Current Rating)	65 kA in Verbindung mit den empfohlenen Sicherungen class J bzw. Leistungsschalter nach UL489 / CSA 22.2 No. 5-02
-Netzleistungsfaktor bei Bemessungsleistung:	
Grundschiwingung ($\cos \varphi_1$)	>0,96
Gesamt (λ)	0,65 ... 0,9
-Überspannungskategorie nach EN 60664-1	Klasse III
-Zwischenkreisspannung, etwa	1,35 \times Netzspannung
-Elektronikstromversorgung	DC 24 V, -15 %/+20 %
-Funk-Entstörung	
Standard	Keine Funk-Entstörung
Mit Netzfilter	Kategorie C2 nach EN 61800-3 Gesamtleitungslänge bis 350 m (geschirmt)
-Kühlart	
	Interne Lüftkühlung Externe Lüftkühlung, Leistungsteile mit verstärkter Lüftkühlung durch eingebauten Lüfter
-Umgebungs- bzw. Kühlmitteltemperatur (Luft)	0 ... 40 °C ohne Derating, >40 ... 55 °C
im Betrieb für netzseitige Komponenten, Line Modules und Motor Modules	
-Aufstellungshöhe	Bis 1000 m über NN ohne Derating, >1000 ... 4000 m über NN

2.4.8 Double Motor Module

Über das Double Motor Module werden die Motoren mit elektrischer Leistung für den Betrieb versorgt, gleichzeitig besitzen diese die Eingangspannung für die DRIVE - CLIQ Ausgänge der Positionsgeber der beiden Motoren. Die Versorgungseingangsspannung für das Double Motor Module liefert das Smart Line Module mit 600V DC. Ausgangsseitig liefern sie eine Frequenz und Amplituden gesteuerte drei Phasen 400V Wechselspannung. Das Double Motor Module wird in Booksize Bauform geliefert, und auf einer Hutschiene montiert.

"Die Double Motor Modules haben standardmäßig folgende Anschlüsse und Schnittstellen:

- 2 Zwischenkreisanschlüsse über integrierte Zwischenkreisschienen
- 2 Elektronikstromversorgungsanschlüsse über integrierte DC-24-V-Schienen
- 4 DRIVE-CLiQ-Buchsen
- 2 Motoranschlüsse über Stecker
(nicht im Lieferumfang enthalten)
- 2 Sicherer-Halt-Eingänge (1 Eingang pro Achse)
- 2 Sichere Motorbremsenansteuerungen
- 2 Temperatursensor-Eingänge
- 1 PE/Schutzleiter-Anschluss

Der Status der Motor Modules wird über zwei mehrfarbige LEDs angezeigt.

Die Schirmung ist im Gehäusekonzept integriert, was zu einer besseren Schirmanbindung führt.

- C-Type: optimiert für Dauerlast mit bis zu zweifacher Überlast (Continuous Motion)
- D-Type: optimiert für hochdynamische Aussetzlastspiele mit dreifacher Überlast (Discontinuous Motion)

Die Geräte der Bauform Booksize C-/D-Type sind für mehrachsige Anwendungen optimiert und werden direkt nebeneinander montiert. Die Verbindung für den gemeinsamen Gleichspannungszwischenkreis ist integriert. Die Entwärmung erfolgt über interne Luftkühlung.

Die Motor Modules Bauform Booksize C-/D-Type sind eine ersatzteilkompatible Weiterentwicklung der Booksize-Serie und bieten dabei folgende Vorteile:

- Der Platzbedarf unterhalb der Motor Modules ist durch ein verbessertes Design und einen neuen Motoranschluss-Stecker reduziert

[6], Quelle: Siemens AG, Produktbeschreibung, SMART LINE MODULE.

- Im neu entwickelten Motoranschluss-Stecker sind die Bremsleitungen und der PE-Anschluss direkt in den Stecker integriert
- Beim Double Motor Module liegen die Motoranschlüsse nebeneinander. Dies führt zu einer deutlich verbesserten Zugänglichkeit
- Der Lüfter kann einfach getauscht werden, ohne dass das Motor Module ausgebaut werden muss

Der Signalleitungsschirm kann mittels einer Schirmanschlussklemme an dem Motor Module aufgelegt werden, z. B. Typ KLBÜ 3-8 SC der Fa. Weidmüller.

Im Lieferumfang der Motor Modules sind enthalten:

- DRIVE-CLiQ-Leitung entsprechend der Breite des Motor Modules zum Anschluss an das nachfolgende Motor Module, Länge = Breite Motor Module + 0,06 m
- 2 Blindstopfen zum Verschließen unbenutzter DRIVE-CLiQ-Buchsen
- Steckbrücke zum Verbinden der DC-24-V-Schiene mit dem nachfolgenden Motor Module
- Stecker X21 und X22
- Gerätelüfter für die Leistungsteilentwärmung, die von den internen Spannungsebenen betrieben werden
- 1 Satz Warnhinweisschilder in 30 Sprachen
- 1 Schirmanschlussklemme

Technische Daten:

-Zwischenkreisspannung (bis 2000 m über NN)	DC 510 ... 720 V (Netzspannung 3 AC 380 ... 480 V)
-Ausgangsfrequenz:	
Regelungsart Servo	0 ... 650 Hz
Regelungsart Vector	0 ... 300 Hz
Regelungsart U/f	0 ... 600 Hz
-Elektronikstromversorgung	DC 24 V -15 %/+20 %
-Kühlart	Interne Luftkühlung Leistungsteile mit verstärkter Luftkühlung durch eingebauten Lüfter
-Zulässige Umgebungs- bzw. Kühlmitteltemperatur (Luft)	0 ... 40 °C ohne Derating, >40 ... 55 °C
im Betrieb für netzseitige Komponenten, Line Modules und Motor Modules	
Aufstellungshöhe	Bis 1000 m über NN ohne Derating,

>1000 ... 4000 m über NN

-Konformitätserklärungen

CE (Niederspannungs- und EMV-Richtlinie)

-Eignungsnachweis

cULus

-Safety Integrated

Safety Integrity Level 2 (SIL 2) nach IEC 61508, Performance Level d (PL d) und Kategorie 3 nach EN ISO 13849-1

-Ausgangsstrom

Bemessungsstrom I_N A 2 × 3

Bei S6-Betrieb (40 %) I_{S6}

D-Type A 2 × 4

Grundlaststrom I_H A 2 × 2,6

I_{max}

D-Type A 2 × 9

Typeleistung

Auf Basis I_N kW 2 × 1,6

Auf Basis I_H kW 2 × 1,4

-Zwischenkreisstrom I_d A 7,2

-Strombelastbarkeit

Zwischenkreisschienen A 100

DC-24-V-Schienen A 20

-Zwischenkreiskapazität μ F 220

-Strombedarf A 0,9

Bei DC 24 V, max.

Interne Luftkühlung

-Verlustleistung

Maximale Verluste kW 0,1

Typische Verluste kW 0,05

-Kühlluftbedarf m^3/s 0,009

-Schalldruckpegel L_{pA} dB <60
(1 m)

-Motoranschluss 2 × Stecker (X1, X2) ,

U2, V2, W2		2 × (1,5 ... 6 mm ²)
-PE-Anschluss		Schraube M5
-Motorbremsenanschluss		Im Stecker des Motoranschlusses integriert (X1, X2), DC 24 V, 2 A
-Motorleitungslänge, max.		
Geschirmt	m	50
Ungeschirmt	m	75
Schutzart		IP20
-Maße		
Breite	mm	50
Höhe	mm	380
Tiefe	mm	270
Gewicht, etwa	kg	4,7 " [6]

2.4.9 Antriebsmotoren

Als Antriebsmotoren wurden jeweils ein Synchronmotor von Siemens geliefert. Diese unterscheiden sich aber hinsichtlich des Positionsgebers. So hat z. B. der Synchronmotor der Y-Achse einen Absolutwertgeber und kann aufgrund seines digitalen Ausgangswertes direkt mit der DRIVE - CLIQ - Verdrahtung an das Double Motor Module angeschlossen werden. Währenddessen der Synchronmotor an der X-Achse einen Inkrementalgeber besitzt und dessen analoges Ausgangspositionssignal noch einen SMC 20 Encoder benötigt, damit dieses Signal über die DRIVE - CLIQ - Verdrahtung an das Double Motor Module angeschlossen werden kann. Dieser SMC 20 Encoder, sowie dessen Anschlussleitungen mussten noch nachbestellt werden und waren im Lieferpaket ursprünglich nicht enthalten, was zu einer verspäteten Inbetriebnahme des Kreuztisches führte. Auch benötigt die X-Achse durch diesen enthaltenen Inkrementalgeber bei jedem Neustart der Motorsteuerung eine Referenzpunktfahrt, da die Positionsausgabewerte des Inkrementalgebers nicht statisch sind und mit jedem Ausfall der Netzspannung oder Abschalten der Motorsteuerung verloren gehen. Dies wird dann später in der Konfiguration der Achse softwaretechnisch mit eingepflegt und bedarf im OB1 der SPS Softwareprogrammierung einen eigenen Baustein. [6], Quelle: Siemens AG, Produktbeschreibung, DOUBLE MOTOR MODULE.

Die Montage und Anbindung mit dem Zahnriemenantrieb der beiden Motoren am Kreuztisch übernimmt ein Mitarbeiter vom Mechanikbereich der Hochschule Kempten. Dazu werden neue Zahnriemenscheibenaufnahmen gefertigt und die Motorhalterung angepasst.

"Technische Daten des Motors 1 an der X- Achse 1FK7042-2AF71-1AG1:

- Bemessungsdrehzahl: 3000 U/min
- Polzahl: 8
- Bemessungsdrehmoment (100 K): 2.6 Nm
- Max. Drehmoment: 8,2 Nm
- Bemessungsstrom: 1,9 A
- Bemessungsstrom Umrichter: 3 A
- Max. Bemessungsstrom Umrichter: 6 A
- Optimale Leistung: 0,8 kw
- Stillstandsrehmoment (60 K): 2,5 Nm
- Stillstandsrehmoment (100 K): 3 NM
- Stillstandsstrom (60 K): 1,8 A
- Stillstandssrom (100 K): 2,2 A
- Trägheitsmoment des Rotors: 2,9 Kgcm²
- Wirkungsgrad: 89 %
- Drehmomentkonstante: 1,38 Nm/A
- Spannungskonstante bei 20° C: 90 V/1000*min pro Minute
- Wicklungswiderstand bei 20° C: 4,67 Ω
- Drehfeldinduktivität: 35,0 mH
- Elektrische Zeitkonstante: 7,5 ms
- Mechanische Zeitkonstante: 2,15 ms
- Thermische Zeitkonstante: 30 min
- Wellentorsionssteifigkeit: 15500 Nm/rad
- Nettogewicht des Motors: 4,6 Kg
- Motortyp: Compact
- Achshöhe: 48 mm
- Kühlung: Luftkühlung
- Rundlauf toleranz: 0,04 mm
- Koaxialitätstoleranz: 0,08 mm
- Planlauf toleranz: 0,08 mm
- Schwinggrößenstufe: Stufe A
- Steckergröße: 1
- Schutzart: IP65
- Bauform gemäß Code I: IM BM (IM V1,IM V3)
- Temperaturüberwachung: Temperatursensor KTY84 in der Statorwicklung
- Elektrischer Anschluss: Stecker für Signale und Leistung drehbar
- Gehäusefarbe: Anthrazit RAL 7016
- Haltebremse: nicht vorhanden
- Wellenende: Glatte Welle
- Gebersystem: Inkrementalgeber sin/cos 1 Vpp
2048 S/R mit C- und D-Spur

Die Technischen Daten des 2. Motors, Y- Achse, 1FK7042-2AF71-1RG1:

Die Daten dieses Motors sind bis auf das Gebersystem genau identisch, daher steht hier nur die Angabe des Gebers:

- Gebersystem: Encoder AM20DQI: Absolutwertgeber 20 bit (Auflösung 1048576, gerberintern 512 S/R) + 12 bit Multiturn (Verfahrbereich 4096 Umdrehungen)" [6]

2.5 Die verwendete Software

In meinem Motorsteuerungsaufbau wird die Siemens S7 SPS Programmierung dafür benutzt um die SINAMICS S120 PLC - Open Technologie Bausteine in ein ablauffähiges Programm einzubinden. Dazu wird für die SPS die STEP7 Version V5.5 + Servicepack zwei benutzt. Ausgabestand ist K5.5.20.

Um für die Technologie CPU die Technologieobjekte zu Parametrieren und zu Programmieren bedarf es der S7-Technology Software Version 4.2.1.0 Optionspaket zu STEP7.

Verbindungsglied zwischen der SPS und der Technologie ist bei der SPS Programmierung das auf der Technologieseite zu findende "Technology Objects Management", abgekürzt "TOM" genannt. So bekommt z. B. die Y-Achse im späteren Verlauf den Technologiedatenbaustein 4 (DB4) zugewiesen, um als Verbindungsglied zwischen SPS und Technologie in der SPS Programmierung zu fungieren. Beide Achsen zusammen werden als Bahnobjekt bezeichnet und dazu wird in meiner Programmierung der Technologiedatenbaustein DB5 kreiert, welcher von nun an in meiner SPS Programmierung aufgerufen wird, um jenes Bahnobjekt, also beide Achsen zusammen ansprechen zu können.

Als dritte Software wird WinCC Flexible Version 2008 SPS Update 13 geladen, um später eine HMI-Bedienoberfläche erzeugen zu können, also ein am PC Bildschirm aufrufbares Bedienfeld, welches über Tastaturbefehle gesteuert werden kann. Dies würde der Überschaubarkeit dienen, und man bräuchte für die späteren Befehlseingaben am Schaltschrank weniger Knöpfe.

Das Grundskelett für meine Motorsteuerung des Kreuztisches würde jetzt stehen, wodurch ich nun zur Inbetriebnahme und Parametrierung meiner Steuerung fortfahren werde.

[6], Quelle:Siemens AG, Produktbeschreibung, SYNCHRONMOTOR.

3. Inbetriebnahme

Um die nun aufgebaute und voll verdrahtete Anlage in Betrieb zu nehmen benutze ich das Skript: SIMATIC S7-300, Anbindung SINAMICS S120 an die Technologie-CPU, welches ich unter <http://support.automation.siemens.com> gefunden hatte. [5]

3.1 Anlegen des S7 Projektes

Im S7 SIMATIC Manager wird ein neues Projekt mit Namen angelegt, "Kreuztisch 7.2014". Mit diesem Namen lässt sich das Projekt im S7 SIMATIC Manager jederzeit wieder öffnen.

3.2 Projektierung in der Hardware Konfiguration

In diesem Schritt wird dem Projekt die Hardware über den Hardware-Katalog "SIMATIC Technologie-CPU" zugewiesen. Dieser Hardware-Katalog ist im S7 Simatic Manager zu finden. Dazu füge ich eine Hardware aus dem Katalog, beginnend mit der gewählten Profilschiene per "Drag and Drop" in die Konfiguration ein, und vergebe die passenden Steckplatznummern. Dabei wird die Hardware genau nach Ihrer Bestellnummer ausgewählt, damit diese mit Ihrer Funktion auch genau übereinstimmt. Desweiteren werden nun die passenden Übertragungsgeschwindigkeiten eingestellt. Diese dürfen nicht höher sein als das was die USB- Profibusschnittstelle leisten kann. Andernfalls kann es, wie es mir passiert ist, vorkommen, dass auf den Profibus nicht mehr eingegriffen werden kann. Dann hilft es nur noch die T-CPU mit bestimmten Schalterbelegungen auf Urzustand zu löschen. Die verschiedenen Busteilnehmer mit dem passendem Bus werden zum Schluss noch über "Drag and Drop" miteinander verlinkt. Dieser Schritt wird S7 Simatic Manager "Routing" genannt. Desweiteren wird den verschiedenen Bussystemen die passenden Busgeschwindigkeiten zugewiesen.

3.3 Projektierung der Antriebskomponenten

Grundsätzlich kann die Projektierung der Antriebskomponenten im S7T Konfigurationsordner auf zwei verschiedene Arten durchgeführt werden:

Erstens durch die Automatische Konfiguration:

Die Topologie des Antriebssystems und die Komponenten des Antriebssystems, sofern Sie mit einer DRIVE-CLiQ Schnittstelle ausgerüstet sind, werden automatisch durch S7T Config erkannt.

Zweitens durch die Manuelle Konfiguration:

Hierbei muss die Struktur des Antriebssystems manuell angelegt werden und die vorhandenen Baugruppen müssen manuell aus den Konfigurationslisten ausgewählt werden.

Bei meiner Projektierung funktionierte das automatische Projektieren nicht, und musste alle Daten und Bauteilnummern selbst auswählen und eingeben.

Im letzten Schritt der Projektierung ist es notwendig die Kommunikationsschnittstelle über den PROFIBUS abschließend zu projektieren, dies ist ein Abgleich zwischen der Telegrammkonfiguration in S7T Konfiguration und in HW Konfiguration. Dabei werden die E/A-Adressen für den Datenaustausch über PROFIdrive zwischen SINAMICS und der Technologie-CPU festgelegt.

Diese komplette Projektierung wird nun über die T-CPU auf Antriebskomponenten geladen.

3.4 Antriebstest über die Antriebssteuertafel des SINAMICS

Nach nun fertig gestellter Projektierung sollte es möglich sein den Kreuztisch über die in dem S7-Technologiebereich bereitgestellten Antriebssteuertafel verfahren zu können. Diese Antriebssteuertafel erlaubt es die Antriebe im entscherten Bereich verfahren zu können. Daher ist es möglich ohne Beachtung der Endschalter und sonstigen Sicherheitsrelevanten Sicherheitseinrichtungen den Antrieb zu bewegen.

Leider ist mir dies auch nach zwei Wochen Fehlersuche und Auswertung der Fehlermeldungen nicht gelungen den Antrieb in Bewegung zu bringen. Daraufhin wandte ich mich an die Siemensniederlassung in Kempten, und bat um Rat. Herr Harald Stecker, Mitarbeiter der Siemens AG Kempten erklärte sich bereit, sich meine Steuerung anzusehen. Nach seiner persönlicher Begutachtung stellte sich heraus, dass die beiden DRIVE - CLIQ Leitungen zu den jeweiligen Motorpositionsgebern im Steckplatz getauscht werden müssten. An dies hatte ich nicht gedacht, da diese in der Parametrierung selbst Ihrem Motor zugewiesen werden. Nach dieser Fehlerbehebung war es nun möglich den Antrieb mit der der Antriebssteuertafel zu bewegen. Nachdem ich das Antriebssystem SINAMICS S120 projektiert und in Betrieb genommen habe, müssen die Achsen im Technologieteil der Technologie-CPU angelegt werden. Dadurch kann die CPU auf die Achsen zugreifen und die Steuerung der Achsen wird ermöglicht.

3.5 Technologie-Datenbausteine anlegen

Durch das Erzeugen der Technologie-Datenbausteine wird nun die Brücke zwischen den Antrieben im Technologiebereich und die Einbindung in das Anwenderprogramm im STEP7 Bereich geschlossen.

Dabei wird für die X-Achse der Datenbaustein 3 kreiert, die Y-Achse erhält den DB4, und beide Achsen zum Bahnobjekt zusammenwirkend bekommen den DB5 zugewiesen. Über diese DBs können nun die einzelnen Achsen und das Bahnobjekt im zu entwickeltem Anwenderprogramm aufgerufen werden. Mit dem DB5 ist es in der späteren Anwendersoftware möglich mit der X- und Y-Achse gleichzeitig im Automatikbetrieb eine Figur fahren zu können.

3.6 Visualisierung mit WinCC-Flexible

Um die Bedienung der Steuerung einfach und übersichtlich zu gestalten, möchte ich eine HMI-Oberfläche kreieren. Mit einer HMI-Oberfläche ist es möglich an einem PC-Bildschirm Bedien- und Anzeigeelemente zu entwickeln, wodurch dann per Mausklick z. B. die Bedienung der Achse, Bedienung eines Bausteins, Aktivierung eines zusätzlichen Verfahrtauftrags, Simulation der Achse usw. ausgeführt werden können. Und dies mit einer großen Übersichtlichkeit, welche durch die beschränkte Anzahl der Schalter auf dem Bedienblech des Schaltschranks nur schwer herzustellen ist. Da ich noch keine Erfahrungen mit WinCC-Flexible habe, arbeite ich mich mit einem Übungsbeispiel ein.

Thema: Erstellen Sie je einen digitalen Eingang, einen digitalen Ausgang der SPS, stellen Sie einen Zählwert da, und geben Sie einen Zeitsollwert als Variable vor. Dieses Beispiel diente zu Vorlesungszwecken als Übungsprojekt an der HS- Kempten, welches ich von meinem Betreuer bereitgestellt bekam.

Der Versuch über WinCC mittels einer HMI-Oberfläche die Bedienbefehle zu übernehmen und damit die Handschaltung des Kreuztisches in X- und in Y-Achse verfahren zu können, scheiterte leider daran, dass ich für meine Anwendung mindestens 356 Powertags benötigen würde, aber nur 128 Powertags in der vorhandenen Lizenzierung der WinCC-Flexible vorhanden sind. Die Powertags sind nötig um eine bestimmte Anzahl von Verschaltungen mit dem WinCC zu kreieren.

Nach Rücksprache mit meinem Betreuer an der HS Kempten, hat man sich entschieden den Kreuztisch ohne HMI Oberfläche aufzubauen. Darum war es leider nicht möglich die Ansteuerung der Befehle in meiner Steuerung über eine HMI Oberfläche zu bewerkstelligen.

Dies bedeutet für mich, dass ich die Befehle welche ich abgeben möchte durch eine Variation von Tastaturbefehlen herstellen werde. Da ich schon viel Arbeit in diesen Bereich investiert habe ist dies die sinnvollste Art damit ich das Projekt zum geplanten Zeitpunkt abgeben kann.

3.7 Entwicklung der Handsteuerung

3.7.1 Eigenschaften eines Templates

Bei der Entwurfsplanung der Handsteuerung bin ich auf der Homepage von Siemens: <http://support.automation.siemens.com> auf ein Template gestoßen, welches ich in meine Steuerung integrieren möchte.

Ein Technology Template ist ein Softwareobjekt oder ein Code-Baustein mit definierten Schnittstellen, der in andere Software-Projekte integriert werden kann und dort eine genau definierte technologische Aufgabe ausführt.

"Das Template "MC_MoveJog.2 vom März 2009" [7] hat folgenden Inhalt:

- Manuelles Verfahren von Achsen über Tipp-Tasten
- Vorgabe von Geschwindigkeit, Beschleunigung und Ruck für die Verfahrbewegung.

Diesem Template liegt eine Dokumentation " Technologie Template (MC_MoveJOG) Technologie -CPU " in PDF Format vor, in welche ich mich einarbeite, sowie ein STEP 7-Archiv (21365191_CPU31xT_MoveJog_CODE_v41.zip) welches die zum Technology Template gehörende Bausteine zur Einbindung in ein Anwenderprogramm beinhaltet.

Beides, Dokumentation und STEP 7-Archiv stehen in selbiger Seite zum Download zur Verfügung, unter <http://support.automation.siemens.com/WW/view/de/21365191>.

Ziel dieses Templates ist es die an die CPU angeschlossenen Achsen einzeln manuell Positionieren und Fahren sowie diese ohne großen Aufwand in positiver und negativer Richtung verfahren zu können. Die Richtungsvorgaben und Verfahrbefehle sollen nur so lange anhalten, wie die Tasten gedrückt gehalten werden. Geschwindigkeit, positive sowie negative Beschleunigung der einzelnen Achsen sowie der Ruck sind über Eingangsparameter im Baustein frei justierbar.

Das Template braucht nur noch mit einem zu entwickelndem Anwenderprogramm integriert werden und mit einer Achsennummer über den von mir zuvor im TOM vereinbarten Datenbaustein zugeordnet werden.

Weitere Eigenschaften, Ein-Ausgänge des Templates Namens "MC_MoveJog" sind:

- Die Wahl der Achse welche bewegt werden soll wird einfach durch Angabe der Nummer des im TOM gewählten DBs ausgewählt. (Eingang: Axis, Datentyp: INT)
- Die Wahl der Bewegungsrichtung wird über zwei getrennte Eingänge ermöglicht. (Eingänge: JOG_Pos und JOG_Neg, Datentyp: Bool)
- Die Sollgeschwindigkeit bei der Bewegung der Achse kann eingestellt werden. (Eingang: Velocity, Datentyp: Real)
- Vorgabe der Soll-Beschleunigung der Achse. (Eingang: Acceleration, Datentyp: Real)
- Vorgabe der Soll-Verzögerung der Achse. (Eingang: Deceleration, Datentyp: Real)
- Vorgabe des Sollwertes des Rucks der Achse. (Eingang: Jerk, Datentyp: Real)
- Der Baustein ermöglicht eine Anzeige des Achsbewegungsauftrages direkt am Bausteinausgang, so lässt sich gerade Entwicklungszeitraum und bei Erstversuchen im OB1 der Bausteinzustand im Verfahrbetrieb einfachst beobachten, und somit auch kontrollieren ob eine Funktion vorhanden ist. (Ausgang: Busy, Datentyp: Bool)
- Rückmeldung wenn die Sollgeschwindigkeit erreicht ist, dies wird am Ausgang angezeigt. (Ausgang: InVelocity, Datentyp: Bool)
- zeigt an das der Verfahrtauftrag durch einen externen Auftrag abgelöst wurde. Falls dieser Ausgang gesetzt ist, kann der Baustein nicht mehr über seine Eingänge kontrolliert werden. (Ausgang: ComandAborted, Datentyp: Bool)
- Signalisiert, dass bei einer Ausführung des Bewegungsauftrages ein Fehler aufgetreten ist. (Ausgang: Error, Datentyp: Bool)
- Zeigt den im Zusammenhang mit dem Ausgang "Error" stehenden Fehlercode zur Lokalisierung der Fehlerursache an. (Ausgang: ErrorID, Datentyp: Word).

Der Fehlercode lässt sich unter einer Fehlertabelle entschlüsseln, welche im SIMATIC Engineering Tools S7-Technology Funktionshandbuch zu entnehmen ist, z. B. Seite 958. Diese Vorgaben waren mir während meinen Entwicklungsversuchen stets sehr hilfreich.

Für die Entwicklung meiner Anwendersoftware werde ich zwei dieser FB 512 "MC_MoveJog" benötigen. Einmal um im Handbetrieb die X-Achse in beide Richtungen zu bewegen, und selbiges für die Y-Achse.

3.7.2 Integration des Technology Templates

In meinem Fall war das Template mit der Dokumentation zusammen als Zip File unter der Siemens Homepage zu downloaden. Die Datei ist ein STEP7-Archiv und enthält die zum Technologie Template benötigten Bausteine damit sie in ein Anwenderprogramm eingebunden werden kann.

Deaktivierung des Technologie Templates:

Das Template STEP7-Archiv muss als erstes über STEP7 deaktiviert werden. Die Funktionen Archivieren und Deaktivieren sind im Startmenü des STEP7 zu finden. Danach wird der komplette, S7-Programmordner aus dem STEP7-Archiv des Technologie Templates in das Wurzelverzeichnis meines STEP7-Projekts kopiert. Jetzt sind alle, für das Technologie Template notwendigen Bausteine, sofort in meinem Anwenderprogramm zur Verfügung.

Zur finalen Integration der Elemente des Technologie müssen nur noch die einzelnen Bausteine des Templates aus dem kopierten S7-Programmordner in den Bausteinordner meiner Anwendung kopiert werden. Daraufhin kann das Technologie Template durch einfachen Aufruf des Bausteins FB512"MC_MoveJog" im Anwenderprogramm genutzt werden. Dieser Baustein kann in den Programmiersprachen AWL und FUP aufgerufen werden. Ich selbst entscheide mich für die Programmiersprache FUP in meiner kompletten Programmierung, da ich diese am übersichtlichsten finde.

3.7.3 Zuordnung des Instanzdatenbausteins

Zu jeder Achse, welche parallel zu anderen Achsen mit Hilfe des FB512"MC_MoveJog" manuell bewegt werden soll, muss dem Funktionsbaustein FB512"MC_MoveJog" ein eigener Instanzdatenbaustein, bzw. ein Bereich in einem Multiinstanzdatenbaustein zugeordnet werden.

Durch einen einfachen Bausteinaufruf und die Übergabe der erforderlichen Parameter in den Bearbeitungsablauf des Anwenderprogramms wird der Baustein FB512"MC_MoveJog" des Technologie Templates integriert.

3.7.4 Einbindung der PLC-Open Bausteine

Desweiteren benötigt der Baustein FB512"MC_MoveJog" PLC-Open Technologiefunktionsbausteine um eine Bewegung ausführen zu können. Eine geeignete Auswahl von PLC-Open Bausteine sind im zuvor genannten Funktionshandbuch [8] zu finden. Ich entscheide mich aufgrund meiner Anforderungen für den Baustein FB401 MC_Power, mit der Technologiefunktion "MC_Power" kann eine Achse freigegeben bzw. gesperrt werden.

Beschreibung der Ein und Ausgänge des MC_Power:

- Eingang: Axis, in Int, Nummer des Datenbausteins
- Eingang: Enable, in Bool, Freischaltung oder Sperrung der Achse
- Eingang: Mode, in Int, Betriebsart der Achse wird gewählt
- Eingang: StopMode, in Int, Verzögerungsmuster wird gewählt
- Ausgang: Status, in Bool, Status der Freigabe der Achse
- Ausgang: Busy, in Bool, Auftrag in oder nicht in Bearbeitung

- Ausgang:Error, in Bool, Fehler oder kein Fehler
- Ausgang:ErrorID, in Word, Fehlercode zum Ausgang Error

3.7.5 Öffnen des OB1, Erstellen des Anwenderprogrammes "Handbetrieb"

Im SIMATIC Wurzelverzeichnis findet man im Projektfenster unter Programme den Ordner Bausteine, welcher geöffnet den OB1 (Organisationsbaustein) anzeigt. Dieser wird geöffnet und ein neues Anwenderprogramm erstellt.

In diesem OB1 schreibe ich das Anwenderprogramm, dabei entwerfe ich die Logik wie folgt:

- Automatikbetriebsschalter verriegeln
- den Notausschalter integrieren
- Freigabetaster einfügen
- Selbsthaltung Freigabetaster
- Freigabekontrollleuchte speisen und mit der Freigabe verknüpfen
- Handbetriebsschalter integrieren
- jeweils zwei Eingänge pro Achse für Rechts-und Linkslauf einfügen
- jeweils ein Achsenfreigabebaustein pro Achse einfügen

Das Anwenderprogramm ist im Anhang unter Abb. 3.7.5 zu finden. Die Endschalter sind in diesem Kapitel noch nicht mit eingebunden, sondern kommen erst beim Automatikbetrieb in Einsatz.

3.7.6 Bedienung im Handbetrieb

Nach fertig gestellter Programmierung und anschließendem Laden des erstellen Anwenderprogrammes in die T-CPU lässt sich nach Betätigen des Dreiphasenhauptschalters die Anlage mit Spannung versorgen. Am 24 V Netzteil liegt jetzt die Spannung von 230 V AC an, welches nun selbst die 24 V DC Versorgungsspannung für die SPS Steuerung, FU CPU, Smartlinemodule, Doublemotormodule liefert. Am Frontbedienblech des Schaltschranks lässt sich am oberen Schlüsselschalter Hand- oder Automatikbetrieb wählen. Dort stellt man auf Handbetrieb. Mit dem Taster Freigabe (am mittleren Frontbedienblech des Schaltschranks) werden nun die drei Phasen freigeschalten, und die 380 V AC stehen nun dem Smartlinemodule zur Verfügung, welches die weitere Laststromverteilung über das Doublemotormodule zu den Motoren übernimmt. Als nächsten Schritt werden nun die beiden Taster " Freigabe Motor1" und " Freigabe Motor2" betätigt, womit dann über die SPS-Steuerung die X- und Y-Achse freigeschalten werden. Nun lässt sich mit den vier Bewegungsrichtungstasten auf dem Frontbedienblech des Schaltschranks der Kreuztisch in alle vier Richtungen in einer Ebene bewegen, Anfangs experimentiere ich noch mit den ganzen Parametern für Beschleunigung, Ruck, und Geschwindigkeit bis ein geeignetes Fahrbild entstanden ist.

Als nächsten Schritt wird die Einbindung der Endschalter benötigt, damit im Automatikbetrieb die Achsen nicht überfahren werden, und der Kreuztisch dabei zerstört werden würde. Gleichzeitig werden diese mit dem Referenzierbetrieb gekoppelt.

4. Inbetriebnahme Schwerpunkt "Automatikbetrieb"

4.1.1 Hardware-Endschalter Signale in die T-CPU einlesen

Die Überwachung der mechanischen Endlage einer Achse erfolgt in der Regel über Hardware-Endschalter, die von der Achse betätigt werden, sobald sich diese der Endlage nähert.

Da bei meinem Kreuztisch zwei lineare Achsen vorhanden sind, sind zur Überwachung der Endlagen insgesamt vier Hardware-Endschalter notwendig, um die Endlage in positiver und negativer Richtung getrennt voneinander überwachen zu können.

Dabei findet bei der Technologie-CPU die Verarbeitung der Endschaltersignale für die jeweilige Achse in der integrierten Technologie statt. Für die Übertragung der Signale an die integrierte Technologie stehen mehrere Möglichkeiten zur Verfügung.

Das Einlesen in die T-CPU kann auf verschiedene Arten erfolgen:

- als Direkten Anschluss der Hardware-Endschalter an der integrierten Technologie über die digitalen OnBoard-Eingänge der T-CPU
- als Dezentralen Anschluss der Hardware-Endschalter an der integrierten Technologie über die digitalen Eingänge einer am DP (Drive= angeschlossenen dezentralen Peripherie ET 200)
- oder die Nutzung der digitalen Eingänge des Antriebsgerätes

Für den ersten Punkt habe ich mich für meine Anwendung entschieden, welcher gleichzeitig der einfachste und zugleich auch der schnellste ist, da eine weitere Übertragung über ein Bussystem eingespart wird.

Dabei ist zu erwähnen, dass die vier Digitaleingänge, welche in der T-CPU integriert sind schnelle Eingänge sind, und daher die Endschaltersignale schnellst möglichst zur Auswertung und naheliegendem Stoppbefehl an der SINAMICS zur Verfügung stehen, um weiteren Schaden durch unsachgemäßes Weiterfahren der Achsen zu verhindern, falls diese schon auf, oder über den Endschalter gefahren sind.

Die vier digitalen Eingänge auf der Technologie CPU können direkt in der integrierten Technologie ausgewertet und dort für technologische Funktionen, wie auch bei mir für die Referenzpunkterfassung verwendet werden.

Dabei erfolgt die Auswertung der Endschaltersignale der OnBoard-Eingänge in der integrierten Technologie über die Peripherieadresse der OnBoard-Eingänge. Daher muss im ersten Schritt die Peripherieadresse ermittelt werden.

Dies geschieht durch Öffnen des HW Konfig und anschließendem markieren des Steckplatzes der T-CPU. Dort werden dann die Peripherieadressen der OnBoard Ein- und Ausgänge der T-CPU angezeigt. Durch doppelklicken auf den Steckplatz der OnBoard Ein- und Ausgänge der T-CPU kann die Peripherieadresse geändert werden. Um die OnBoard Ein- und Ausgänge nutzen zu können ist eine Peripherieadresse zu wählen die größer oder gleich 64 ist. Adressen kleiner 64 können nur im SPS-Teil der T-CPU genutzt werden und stehen für technologische Funktionen in der integrierten Technologie nicht zur Verfügung, bzw. es können daher keine schnellen Zuweisungen an die SINAMICS erfolgen.

Die nun erstellte Peripherieadresse muss an der Maske für die Einstellung der Hardware - Endschalter an der entsprechenden Achse in S7T Config eingetragen werden. Dazu wird in S7T Config die Projektierung der entsprechenden Achse geöffnet die Parametrieremaske der Begrenzungen gewählt.

Die Hardware-Endschalter werden dort aktiviert und die im HW Konfig erstellte Peripherieadresse sowie die Bitnummern der Eingangsklemmen werden eingetragen. Danach wird diese Eingabe gespeichert, übersetzt und in die T-CPU geladen. Die Zustände

der Hardware-Endschalter können über den Technologiedatenbaustein der entsprechenden Achse oder über die Expertenliste der Achse in S7T Config beobachtet werden.

4.1.2 Referenziermöglichkeiten

Um den Istwert eines Gebers nach dem Einschalten der Versorgungsspannung oder einem Austausch des Gebers mit der Position der Maschine abzugleichen wird die Funktion "Referenzieren" benötigt. Dies ist nötig, damit die Steuerung weiß wo die zugeordneten Achsen stehen, und weitere Verfahrtaufträge mit der richtigen Voraussetzung abgesetzt werden können.

Dabei müssen Inkrementalgeber bei jedem Neustart der Anlage referenziert werden, Absolutwertgeber nur nach einem Austausch des Gebers oder dem Verdrehen der Spindel ohne Geber.

Folgende Referenziermöglichkeiten sind möglich:

- Aktives Referenzieren
- Passives Referenzieren
- Direktes Referenzieren

- **Beim aktiven Referenzieren** wird eine spezielle Verfahrbewegung durchgeführt, bereits aktive Fahraufträge werden dazu abgebrochen.

Folgende Referenziermöglichkeiten sind beim aktiven Referenzieren möglich:

- Referenzieren mit Referenznocken und Geber-Nullmarke
- Referenzieren nur mit Referenznocken
- Referenzieren nur mit Geber-Nullmarke
- Positionswert korrigieren
- Internes Achskoordinatensystem korrigieren

-**Beim passiven Referenzieren** wird keine Bewegung der Achse ausgelöst, aktive Fahraufträge werden dazu nicht abgebrochen, die notwendige "Referenzpunktfahrt" wird in diesem Fall durch externe Einwirkung oder durch das Anwenderprogramm realisiert.

Folgende Referenziermöglichkeiten sind beim Passiven Referenzieren möglich:

- Referenzieren mit Referenznocken und Geber-Nullmarke
- Referenzieren nur mit Referenznocken
- Referenzieren nur mit Gebernullmarke

-**Beim direkten Referenzieren** wird die Achsposition gesetzt ohne Berücksichtigung von Referenznocken (Geber-Nullmarken oder Referenznocken). Soll eine exakte Position auf der Achse zugewiesen werden, so muss die Achse während des Vorgangs still stehen.

4.1.3 Aktives Referenzieren

Meine X-Achse hat einen Inkrementalgeber mit folgenden Daten:

- mit Geber-Nullmarke
- ohne Referenznocken
- Verwendung des Endschalers als Referenznocken

Aufgrund dieser Konstellation wähle ich als Referenziermodi:

-Aktives Referenzieren mit Referenznocken und Geber Nullmarke.

Da ich bei meiner X-Achse keinen Referenznocken habe, nehme ich meinen Endschalter als Referenznocken. Dies ist in der Technologie S7 Config Hilfedatei erläutert. Damit die ganze Steuerung aber nicht stehen bleibt, wenn auf den Endschalter gefahren wird, muss noch in der Expertenliste auf P0495 " Geber Nullmarke" eingestellt werden. Damit ist der Endschalter zugleich als externes Nullmarkenersatzsignal konfiguriert. Zusätzlich sollte noch im S7T Config im Dialogfeld der X-Achse die Richtung angegeben werden in welche die Referenzierung startet, und wo die Nullmarken auftauchen, vor oder nach dem Endschalter, und es muss noch der Hardwareendschalter als Referenznocken ausgewählt werden. Desweiteren können noch maximale Weglängen ausgewählt werden, ab wann die nächste Nullmarke erscheinen müsste.

Erläuterung: Aktives Referenzieren mit Referenznocken und Geber-Nullmarke:

Wenn im gesamten Verbahrbereich der Achse der Geber mehrere Nullmarken besitzt bzw. die Nullmarke mehrfach durchfährt, wird zum eindeutigen Referenzieren ein zweites Signal verwendet. Diese Zusatzinformation liefert ein Referenznocken, und in meinem Fall ein Endschalter. Mit Erreichen des Endschalers wird die Referenzierfunktion für den nächsten Nullmarkendurchgang freigegeben. Ein Nullmarkendurchgang sollte einmal pro Motorachsenumdrehung erscheinen.

Die Achse fährt in angegebener Richtung, (ausgewählt im S7T Config Dialogfeld X-Achse, Referenzieren) für die Referenzpunktfahrt los. Nach Erreichen des Endschalers verlangsamt sich die Achsengeschwindigkeit, die Achse fährt wieder aus dem Endschalter heraus, und fährt zurück, bis die nächste Nullmarke des Gebers erkannt wird. Nun ist die Referenzierung abgeschlossen für die X-Achse. Die Position kann eindeutig zugewiesen werden.

4.1.4 Hardwareaufbau, Referenzierung gespiegelte externe Nullmarke

Als Hardwareaufbau dient mir eine Übersicht, des Siemens Skriptes " Anbindung SINAMICS S120" Abb. 4.1.4 [5]. Dort finde ich ein Verdrahtungsbild " Referenzieren mit einer auf die T-CPU gespiegelten externen Nullmarke", welches ich als Muster nehme.

Funktionsablauf: Für das Referenzieren der Achse soll eine externe Nullmarke (Bero) verwendet werden. Dazu muss einer der schnellen integrierten digitalen Eingänge des SINAMICS S120 genutzt werden. Zusätzlich wird das Signal auch in der Technologie CPU an den zuvor erwähnten ebenfalls schnellen Eingängen eingelesen. Im Unterschied zur ungespiegelten Variante wird nun in der T-CPU ebenfalls der Status des Bero (belegt / unbelegt) ausgewertet. Dadurch ist für das Referenzieren auch eine Richtungsumkehr möglich.

Im Funktionstest stellte sich nun heraus, dass die Nullmarken, welche mindestens einmal pro Motorumdrehung des Motors in der X-Achse erscheinen sollten, fehlen.

4.1.5 Nullmarkenfehler, Fehlersuche

Istzustand:

Der Schlitten der X-Achse fährt in den Endschalter Fahrtrichtung links, wie in der Konfiguration vorgegeben ein, verzögert seine Geschwindigkeit, fährt mit verzögerten Geschwindigkeit wieder richtungsgedreht aus dem Endschalter heraus verlässt den Endschalter und fährt nun so lange bis eine Nullmarke zu erkennen ist. Leider fährt dieser trotz mehrmaligen Umdrehungen des Motors auf der X-Achse weiter, und erkennt keine Nullmarke die dieser als zweites Lageortsignal benötigt. Der Schlitten fährt nun im verzögerten Geschwindigkeitsmodus weiter, bis er in Rechtsrichtung den zweiten Endschalter erreicht, dieser auslöst, und die Anlage zum stehen veranlasst. Dies ist ungefähr eine Strecke von 1,4 m, in welcher die Nullmarke hätte mindestens 10 mal erkannt werden müssen.

Sollzustand:

Normalerweise sollte der Schlitten den Endschalter richtungsverkehrt wieder verlassen, und rechtsfahrend bei der nächsten Nullmarke stehen bleiben. Also ca. nach 10 cm, ist eine Motorumdrehung und zusätzlich abhängig vom Übersetzungsverhältnis des Zahnriemenantriebes, wobei dies ein untersetztes Verhältnis ist.

4.1.6 Fehlersuche

Im S7TCofig, der Expertenliste, sowie in den Fehlermeldungen welche im onlinezustand zur Steuerung angezeigt werden lässt sich kein Fehler lokalisieren.

Um die Fehlermöglichkeiten einzugrenzen beschließe ich einen Versuchsaufbau zu kreieren um nachzuprüfen ob das Signal bei laufendem Betrieb gesendet wird oder nicht. Im Verdacht steht auch der SM 20 Encoder, welcher das vom Inkrementalgeber aussendende Analogsignal in ein Binäres umwandelt. Um auszuschließen, dass dieser ein Problem verursacht, möchte ich das Signal noch vor diesem Encoder abgreifen.

Doch um auch ein Verbindungsproblem selbst auszuschließen prüfe ich im ersten Schritt die einzelnen Litzen der Sensorleitung zwischen dem SM 20 Encoder und Inkrementalgeber auf Durchgang. Dazu baue ich mir aus starren Kupferkabeln Litzen geeignete Spitzen, um diese an den Voltmeter anzubringen, da ich mit den Spitzen der Messleitungen des Voltmeters nicht in die dünnen Pinnbuchsen des Steckers gelange. Der Voltmeter besitzt die Einstellung Durchgangsprüfung mit Signalton, womit sich dann der Durchgang an beiden Kabelenden sobald diese mit dem Durchgangspiepser kurzgeschlossen werden, durch das akustische Signal feststellen lässt.

Zusätzlich suche ich einen Verdrahtungsplan mit dem ich die geprüften Schritte nachvollziehen, und gleichzeitig auf Papier abhaken kann. Diesen finde ich wiederum unter dem online Siemens Supportzentrum. Das Anschlussbild ist im Anhang "vgl" Abb. 4.1.6 zu finden. Da es sich hierbei um einen 17 poligen Rundstecker am Motor, und den 24 poligen Flachstecker am SM 20 Encoder handelt, also mit unterschiedlichen Pinnzahlen, war es für mich wichtig auch zu wissen welche Litzen dort verdrahtungstechnisch zusammenspielen. Diese Kontrolle zeigte, dass das Signalkabel in Ordnung ist. Zusätzlich untersuche ich noch die Stecker und Buchsen selbst auf Schäden, oder sonstigem Auffälligem, sowohl Motor, sowie Encoderseitig.

Das Signalkabel lässt sich somit als Fehlerquelle ausschließen, und ich werde nun im zweiten Schritt den zuvor erwähnten Versuchsaufbau planen und aufbauen.

Da ich desweiteren ausschließen möchte, dass der Encoder eventuell einen Kurzschluss hat, und das Nullmarkersignal unterdrücken könnte, baue ich den Versuchsaufbau so auf, dass

dieser Autark vom Encoder funktioniert. Daher ist es notwendig den Inkrementalgeber gleichzeitig mit einer Versorgungsspannung zu speisen, um wiederum ein Ausgangssignal vom selbigen zu erhalten. Die Höhe der Versorgungsspannung messe ich zuvor aus der angeschlossenen Sensorleitung mit eingeschalteten Komponenten heraus, sie liegt bei 5V DC. Dazu musste ich das Flachsteckergehäuse am SMC 20 Encoder zum Messen öffnen. In meinem Versuchsaufbau kommt darum eine Konstantspannungsquelle zum Einsatz welche mir die 5V DC liefert. Dieser Aufbau ist im Anhang Abb. 4.1.6.1 zu sehen.

Die Ergebnisse zeige ich mit dem vierkanal Laboroszilloscope Tektronix TPS 2014 100Mhz auf. Dabei messe ich mit dem Kanal eins auf Pin drei das R-Signal (Nullimpuls / Referenzsignal) und mit Kanal zwei das C-Signal (Kommutierungslage: C / D Spur) auf Pin 5 jeweils am Rundstecker. Am Flachstecker, an welchem ich das Signal abgreife ist es der Pin 17 für das R-Signal und Pin 19 für das C-Signal. Mit einem gleichmäßigem Drehen von Hand an dem Riementrieb der X-Achse lässt sich nun ein aussagekräftiges Bild entwickeln. Dabei ist zu sehen, dass der Nullimpuls vorhanden ist, aber nicht klar verlaufend in seinem Funktionsablauf ist, der Umriss verzerrt ist. Das C-Signal hingegen ist immer kontinuierlich gleich konstant fortlaufend. Dieses Bild des R-Signales ist ohne Änderung am Systemaufbau einmal klar und einmal verzerrt, wobei ich den Grund für die Ursache nicht lokalisieren, weder noch irgendwie verstärken oder vermindern kann. Vergleichsmessungen in diesem Bereich und damit gesammelte Erfahrungen hatte ich bisher noch nie gehabt, weshalb ich vermutete, dass der Inkrementalgeber vermutlich einen Schaden haben könnte.

Das Ergebnis ist den Bildern Abb. 4.1.6.2 und 4.1.6.3 im Anhang zu sehen.

Als dritte Messung möchte ich nochmals das R-Signal bei laufendem Betrieb abfassen, und messe dabei am Flachstecker des Datenkabels nochmals an Pin 17, welchen ich zuvor schon einmal geöffnet hatte, bei der vorherigen Ermittlung der 5V DC Versorgungsspannung. Auch bei dieser Messung waren die Ergebnisse dieselben wie bei der vorherigen Messung. Dies bestärkt meine Vermutung nochmals.

Aufgrund dieser Vermutung stellte mir die Hochschule Kempten einen weiteren Inkrementalgeber. Dieser hatte, obwohl er dieselbe Bauteilenummer besitzt, unterschiedliche elektronische Bauteile am Anschlussflansch beim optischen Vergleich beider Bauteile.

Diesen Tauschinkrementalgeber baute ich dann ein, und wiederholte die Messungen, welche wieder dasselbe Ergebnis aufzeigten. Ich versuchte zur Sicherheit nochmal in einem Fahrversuch ob sich etwas änderte, aber der Fehler war immer noch derselbe, es sind auch keine eventuell neue Fehlermeldungen im S7T- Config zu lesen gewesen, welche einen weiteren Aufschluss hätten geben können.

Da ich mit meinen Erfahrungen und Kenntnissen der Ursache für den Fehler nicht auf den Grund komme, suche ich ob es über Siemens direkt eventuell Hilfe geben könnte. Als erstes versuchte ich einen Rat bei Herrn Stecker der Siemens AG in Kempten zu bekommen, welcher mir bei der Handbetrieb Inbetriebnahme bereits bestens weitergeholfen hatte. Doch dieser meinte, dass er bei diesem Problem, welches sehr ins Detail geht, mir nicht weiterhelfen könne.

Letztendlich versuchte ich es über das Siemens Servicecenter, von welchem ich unter <http://support.automation.Siemens.com> im Internet die Telefonnummer gefunden hatte, und schilderte dort telefonisch meinen Fall. Auf dieses Gespräch hin, und mit Angabe meiner T-CPU wurde ein Fall geöffnet mit der Nummer SR=1-4337373172 welches zugleich auf meinem E-Mail-Konto bestätigt wurde, wobei sich die nächsten Tage ein Fachberater, welcher in diesem Bereich involviert war, sich telefonisch melden würde.

Herr Barth von der Sinamics -Hotline meldete sich dann die nächsten Tage bei mir telefonisch, konnte mir dann aber rein informell nicht weiterhelfen, woraufhin wir uns geeinigt hatten, dass ich ihm die komplette Konfiguration meiner Siemenssteuerung als Archivierte Siemens STEP 7 Datei als Anhang in einer Mail zukommen lasse.

Es stellte sich heraus, dass die Datenmenge für eine E-Mail zu groß ist, und nicht als Anhang in einer Mail versendet werden kann. Ich informierte mich daher welche Möglichkeiten es gibt, anderweitig größere Datenmengen über das Internet zu versenden. CD-Rom als Datenträger mit dem Postweg wäre zu zeitaufwendig, und kam nur als letztes Mittel in Betracht. Nach einigen Recherchen mit dem Google Internetsuchdienst bin ich dann auf die Möglichkeit der Datenübertragung mit einer Cloud gestoßen. Dabei werden die Daten in einen Onlinespeicher geladen, und von dort vom Empfänger abgerufen. Selbiges hatte ich schon von der Archivierung von Bildern über eine Cloud gehört. In Praxistest Foren mit unter auf Chip.de suchte ich mir eine geeignete, einfach zu Bedienende und kostenlosen Cloud für meinen Zweck, und bin dabei auf WeTransfer als Dienstleister gestoßen.

" **WeTransfer** (englisch *we transfer*, deutsch ‚wir übertragen‘) ist ein 2009 eingeführter Filehosting-Dienst. Der primäre Zweck von WeTransfer ist nicht das Teilen von Dateien mit anderen, wie es konkurrierende Dienste anbieten, sondern der reine Versand großer Dateien an einen oder mehrere Empfänger. Eine Besonderheit ist, dass man sich nicht registrieren muss, um den Dienst zu nutzen. Es genügt das Hinterlegen der eigenen E-Mail-Adresse und der des Empfängers. Nach Angaben der Betreiber werden Dateien eine Woche nach dem Hochladen vom Server gelöscht, eine Datei darf die Größe von 2 Gigabyte und die Zahl der Adressaten von 20 nicht überschreiten. WeTransfer gab bekannt, dass im November 2013 zwei Millionen Personen pro Tag den Dienst zum Übertragen von Dateien genutzt hätten. Anders als bei anderen Filehosting-Diensten bietet WeTransfer keine Anwendungssoftware (App) für Smartphones und verschiedene Betriebssysteme an, sondern funktioniert nur über den Webbrowser. Hat man eine Datei über das Webportal zu WeTransfer – verschlüsselt – übertragen, erhält der Empfänger eine Nachricht und kann über einen nicht öffentlichen Web-Link diese Datei dann herunterladen. Hat er dies getan, wird der Absender darüber informiert, und der Vorgang ist abgeschlossen.

Die Verschlüsselung beim Hochladen und beim Speichern der Datei im von WeTransfer angemieteten Rechenzentrum ist, wie bei allen Cloud-Anbietern, keine Garantie für Datenschutz, Datensicherheit und Diskretion. Obgleich sich das Unternehmen in den Niederlanden befindet, bleiben die Daten nach Firmenangaben nicht notwendigerweise innerhalb Europas, sondern können auch in anderen Kontinenten gespeichert oder zwischengespeichert werden.

WeTransfer finanziert sich durch Werbung in Form von bildschirmfüllenden, hochauflösenden Fotografien der Werbekunden („Wallpapers“) und durch regelmäßig zahlende Kunden, die durch das Abonnement Sonderprivilegien erhalten, wie größere Datenvolumina, eigene E-Mail-Adressen und Passwortschutz. Die direkte Konkurrenz sind große Cloud-Dienste wie Dropbox, Google Drive oder Hightail (vormals YouSendIt).

WeTransfer wurde im Dezember 2009 in Amsterdam von den Niederländern Ronald Hans (Spitzname Nalden) und Bas Beerens gegründet. Beerens leitet eine Werbeagentur und schrieb zur Einsparung von Kurier- und Versandkosten eine Software, um große Dateien von und zu Kunden zu übertragen. Ronald Hans kommt aus der Bloggerszene, wo ihm die Idee kam, großformatige Werbung unter seine Blogs zu legen – heute die optische Basis von WeTransfer. Die Gründer betonen, WeTransfer ausschließlich aus Eigenkapital zu betreiben." [9]

Ich persönlich finde diesen im Internet gefundenen, kostenlosen Dienst sehr praktisch, und anwenderfreundlich, weshalb ich diesen auch für weitere Zwecke gerne wieder nehmen werde.

Nachdem nun die Datenübertragung erfolgreich von statten gegangen war, konnte Herr Barth von der Siemens-Hotline mein STEP 7 Programm einschließlich der Konfiguration bei sich simulieren, und gab mir einige Vorschläge, welche Probleme womöglich vorliegen könnten.

Darunter wurde auch erwähnt, dass ich meinen linken Hardware Endschalter der X- Achse als Referenznocken verschaltet habe. Was aber von mir so gewollt war.

Desweiteren erkannte er, dass der Profibus nicht takt synchron arbeitet, und in der Konfiguration des Profibuses drei Werte umgestellt werden sollten.

Wie sich herausstellte war dies die Ursache für den Fehler weshalb mein Inkrementalgeber nullwertimpuls von der Steuerung nicht erkannt wurde. Der Profibus war nicht richtig justiert.

Nachdem der Fehler behoben war, machte ich einen Fahrttest mit der X-Achse des Kreuztisches, und wiederholte die Referenzfahrt, so wie diese ursprünglich vorgesehen war. Nun funktionierte alles so wie gewollt, der Schlitten fährt in den linken Endschalter der X-Achse, verlangsamt sich sofort, macht einen Richtungswechsel, fährt aus dem Endschalter wieder heraus, und fährt bis zum nächsten Nullwertsignal weiter, und bleibt dann stehen. Die Referenzierung wie ursprünglich geplant, funktioniert. Nachdem diese geklappt hat, baute ich den ursprünglichen Inkrementalgeber wieder zurück, und nach einem Fahrttest zeigte sich, dass auch dieser einwandfrei funktioniert, diesen Geber lies ich dann auch in der Schaltung eingebaut.

4.1.7 Startposition

Im nächsten Schritt möchte ich, dass beide Schlitten nach der Referenzierung an eine voreingestellte Startposition fahren. Damit beide Schlitten für eine zukünftige Figur immer dieselbe Ausgangsposition haben. Es gibt dafür eine Auswahl an Technologiefunktionen die sich unterscheiden zwischen:

"I. Absolutem Positionieren, mit FB 410 "MC_MoveAbsolute"

d.h. Die Technologiefunktion "MC_MoveAbsolute" startet eine Positionierbewegung einer Achse zu einer absoluten Position.

- Mit den Eingangs-Parametern Velocity , Jerk, Acceleration und Deceleration bestimmen Sie das dynamische Verhalten beim Bewegungsvorgang.
- Die Funktion endet mit dem Erreichen der Zielposition.
- Bei Moduloachsen kann die Drehrichtung vorgegeben werden.
- Über den Eingangs-Parameter Mode kann angegeben werden, ob eine aktive Bewegung abgelöst, ob die Bewegung angehängt oder ob die Bewegung überschleifend angehängt wird.

[9], Quelle: Wikipedia, freie Enzyklopädie, Sitz in San Francisco, USA.

Anwendbar auf Positionierachsen oder Gleichlaufachsen

Voraussetzungen

- Achse lagegeregt freigegeben
- Achse referenziert, falls in der Konfiguration "Referenzieren erforderlich" gewählt wurde
- Kein MC_Stop-Auftrag in Bearbeitung
- Für anhängende und überschleifende Bewegungen (Mode = 1, 2) gilt:
 - Kein "MC_CamIn" bzw. "MC_GearIn" gestartet
 - Kein Basisgleichlauf aktiv

II. Relatives Positionieren mit FB 411 "MC_MoveRelative"

Zweck:

- Die Technologiefunktion "MC_MoveRelative" startet an einer Achse eine Positionierbewegung zu einer Position relativ zur Startposition. Befindet sich die Achse beim Auftragsstart bereits in Bewegung und wurde mit Mode = 0 eine "ablösende Bewegung" gewählt, so wird die Startposition verwendet, die systemintern bei Beginn der Auftragsbearbeitung vorliegt. Dabei ist zu berücksichtigen, dass zwischen der Bearbeitung der Technologiefunktion und der Ausführung des Auftrags eine Reaktionszeit liegt, die abhängig von der Auslastung und der Zykluszeit ist.
- Mit den Eingangs-Parametern Velocity , Jerk, Acceleration und Deceleration bestimmen Sie das dynamische Verhalten beim Bewegungsvorgang.
- Über den Eingangs-Parameter Mode kann angegeben werden, ob eine aktive Bewegung abgelöst, ob die Bewegung angehängt oder ob die Bewegung überschleifend angehängt wird.

Anwendbar auf: Positionierachsen oder Gleichlaufachsen

Voraussetzungen:

- Achse lagegeregt freigegeben
- Kein MC_Stop-Auftrag in Bearbeitung
- Für speichernde und überschleifende Bewegungen gilt:
 - Kein "MC_CamIn" bzw. "MC_GearIn" gestartet
 - Kein Basisgleichlauf aktiv

III. Relatives Positionieren zu aktueller Zielposition mit FB 412 "MC_MoveAdditive"

Zweck:

- Die Technologiefunktion "MC_MoveAdditive" startet an einer Achse eine Positionierbewegung mit vorgebbaren Dynamikwerten zu einer Position relativ zur Zielposition eines aktuellen Positionierauftrags. Damit kann das Ziel eines vorhergehenden Auftrags um einen definierten Weg korrigiert werden.
- Mit den Dynamik-Parametern Velocity , Jerk, Acceleration, Deceleration bestimmen Sie das dynamische Verhalten beim Bewegungsvorgang.
- Die Achse bleibt an der Zielposition stehen.
- Ein MC_MoveAdditive-Auftrag löst den laufenden Auftrag ab.

Anwendbar auf Positionierachsen oder Gleichlaufachsen Voraussetzungen:

- Die Achse muss lagegeregt freigegeben sein.
- Die Achse muss referenziert sein, wenn bei der Konfiguration "Referenzieren erforderlich" gewählt wurde und
 - die Achse in Bewegung ist
 - oder der "MC_MoveAdditive" einen laufenden Bewegungsauftrag ablöst (Ausnahme "MC_MoveVelocity").
- Achse muss nicht referenziert sein,
 - wenn die Achse steht
 - wenn ein laufender MC_MoveVelocity-Auftrag abgelöst wird.
- Es ist kein MC_Stop-Auftrag in Bearbeitung.

Hinweis:

Wenn eine der folgenden Voraussetzungen gegeben ist, verhält sich "MC_MoveAdditive" wie "MC_MoveRelative":

- Die Achse steht bei Auftragsstart
- Eine kontinuierliche Funktion wird abgelöst, d. h. die Zielposition ist nicht definiert. Die Zielposition ist dann vom Ablösezeitpunkt abhängig und ergibt sich aus Distanz und der Lage zum Start des Vorgangs.

Technologiefunktionen

Einschränkungen für Moduloachsen:

Die Anwendung auf Moduloachsen ist nur eingeschränkt auf kleine Wegstrecken (in Bezug auf die Modulolänge) möglich, die die aktuelle Modulolänge nicht überschreiten dürfen:

Neue Zielposition = (alte Zielposition + Distanz) modulo Modulolänge

- Distanz muss kleiner als eine Modulolänge sein.
- Der Restweg des zu verfahrenen Weges muss kleiner als eine Modulolänge sein.
- Distanz plus Restweg des zu verfahrenen Restweges vom ablösenden Auftrag muss kleiner als eine Modulolänge sein."[8]

Resümee:

Für diese Aufgabe wähle ich aus dem STEP 7- Archiv den Funktionsbaustein FB410 "MC_MoveAbsolute", welchen ich aus dem Funktionshandbuch SIMATIC Engineering Tools S7-Technology ausgesucht hatte.

Mit diesem Baustein ist es möglich auf die bereits gefahrene Strecke, welche sich beim Referenzieren ergeben hatte mit der von mir definierten Strecke für die gewünschte Startposition aufzuschließen.

Er hat dieselben Ein- und Ausgänge welche bereits bei dem Funktionsbaustein "MC_MoveJog" der Handsteuerung bekannt sind. Zusätzlich besitzt er aber noch den Eingang "Position" für die Streckeneingabe zu meinem gewünschten Startpunkt, und "Direction" für die gewünschte Fahrtrichtung.

Somit ist es mir nach einigen Versuchen und Anbringung einer Startmarkierung an welcher ich den Schlitten haben möchte gelungen die geeignete Startposition nach dem Referenzieren automatisch ohne weitere Tastaturbetätigung anzufahren. Dies ermöglichte mir die Verkettung der "Done" Ausgänge der jeweils vorherigen Funktionsbausteinen in meinem erstelltem Anwenderprogramm.

4.2 Einbindung des Templates "MotionList Basic"

Da die Hardwarekonfiguration bereits im Bereich Handsteuerung sowie im Bereich Referenzieren abgeschlossen wurde, kann ich mich für das kommende Kapitel Automatikbetrieb ganz auf das Anwenderprogramm und die zu integrierenden Funktionsbausteine konzentrieren.

Im ersten Schritt möchte ich wieder auf ein Template zurückgreifen, welches eventuell bei Siemens zur Verfügung gestellt wird. Da der Kreuztisch in vielen industriellen Bereichen zum Einsatz kommt, fällt mir die Suche nicht schwer, und stoße bei <http://support.automation.siemens.com/WW/view/de/59259273> auf ein passendes Template mit dem Namen "MotionList Basic". Dabei nutze ich den Suchdienst Google als Hilfe um in der ganzen Vielfalt an Dokumenten das passende mit meinen Aufgaben zu finden. Diesem Template liegt wiederum eine Dokumentation im PDF Format bei [10], sowie ein STEP 7 Archiv (59259273_MotionListBasic_CODE_v42.zip) welches die zum Technology Template gehörende Bausteine zur Einbindung in mein Anwenderprogramm beinhaltet, und beides ist zum Download auf selbiger Seite zu finden.

Dabei soll mir das Template durch seine vorgefertigten Funktionseigenschaften dazu dienen eine bereits bewährte Steuerung aufzubauen. Auch besitzt dieses Template wieder die Möglichkeit, bzw. ist vorkonfiguriert für den Betrieb mit einer HMI-Oberfläche. Leider kann ich diese Funktion aufgrund der fehlenden Software nicht betreiben, aber es ist vorteilhaft, das diese Funktion später mit Hilfe dieser ebenfalls vorhandenen Dokumentation einfacher zu bewerkstelligen ist, falls man die Steuerung oder Funktionsaufgaben erweitern, bzw. vereinfachen, übersichtlicher gestalten möchte. Als typische Anwendungsgebiete für das "MotionList Basic" gelten die Definition und Abfahren einfacher Konturen mit Hilfe eines Bahnobjektes der Technologie-CPU, z. B. für das Auftragen von Klebstoff auf ein Werkstück oder das einfache Ausschneiden von Blechen mit Hilfe eines Lasers oder eines Wasserstrahles.

4.2.1 Eigenschaften des Templates "MotionListBasic"

Dieses Template bietet noch weitere Vorteile wie z.B.:

- bereitstellen einer Verfahrtable um gekettete Bewegungsabläufe zu realisieren
- Anlegen, Verwalten und speichern einer Verfahrtable in einem Datenbaustein in der Technologie-CPU.
- automatisches abarbeiten einer definierten Verfahrtable mit Hilfe eines Funktionsbausteines, der die Ansteuerung der erforderlichen Verfahrbefehle in der Technologie-CPU übernimmt.
- und wie bereits erwähnt die Unterstützung der Bedienung dieses Technology Templates durch die Integration deiner HMI-Bedienoberfläche zur Veränderung und Verwaltung der Verfahrtable und zur Beobachtung der vorgegebenen Verfahrbewegung.
- die Vorgabe der gewünschten Kontur ist einfach zu bewerkstelligen, durch dieses Technologie Template kann einfach und schnell die gewünschte zusammengesetzte Kontur aus Geraden und Kreisbögen in einem Datenbaustein hinterlegt werden.
- die Bausteine "FB 541 HMI_List", "FB 543 "HMI_DB_Explorer" und "FB 544 Delete_DB" würden im Zusammenspiel mit einer ebenfalls im Template erhaltenen HMI-Bedienoberfläche alle Funktionen für die Eingabe und die Verwaltung der in den Datenbausteinen im Technologie-CPU hinterlegten Konturen zur Verfügung stellen. Diese Vorteile werden bei mir aufgrund der Vereinfachung und nicht Benutzung der

HMI-Oberfläche nicht verwendet. Befehle werden bei mir direkt die den Datensatz des erzeugten Konturelementes UDT 541 "ConturListData" geschrieben.

- durch dem im Technology Template befindlichen Funktionsbaustein 540 "MotionList_Basic" ist es sehr einfach die in einem Datenbaustein hinterlegten Verfahrbewegung zeilenweise zu interpretieren und abzufahren.

Dabei ist dieses Technology Template auf die im Anwenderprogramm noch zu bewerkstelligten Aufgaben angewiesen wie z. B:

- der Überwachung der Achsdynamik wie das Referenzieren, und Einbindung der Endschalter
- Überwachung der der Geräte, Resetmöglichkeiten der Geräte bei einem Fehlerfall
- Einbindung des Not-Aus-Schalters.
- Anfahren der Grundstellung
- Freischalten der einzelnen Achsen

Komponenten des Technology Templates:

In dem Software-Paket des Technology Templates sind alle notwendigen STEP 7 Bausteine enthalten welche für die im Datenbaustein hinterlegten Kontur notwendig sind um eine Verfahrbewegung aufzunehmen.

Der Funktionsbaustein 540 "MotionList_Basic" enthält die vollständige Funktionalität um eine Verfahrbewegung auszuführen mittels einer in einem Datenbaustein hinterlegten Kontur.

Die Funktionsbausteine:

- FB 541 "HMI_List" für Verwaltungsfunktionen der Kontur
- FB 542 "HMI_3D_Movement" für die Berechnung der Darstellung der Kontur
- FB 543 "HMI_DB_Explorer" zur Suche der Konturen
- FB 544 "Delete_DB" zum Löschen eines Kontur-Datenbausteines

werden bei mir Aufgrund des Fehlens der HMI Funktion nicht in Anspruch genommen. Somit sind alle benutzten Funktionsbausteine des Technology Templates für meinen Automatikbetrieb auf den FB 540 "MotionList_Basic" beschränkt, welcher mir mit seinen vielseitigen Ein-und Ausgängen einen guten Grundstock für meine zu realisierenden Steuerung bietet.

4.2.2 Eigenschaften der Verfahrkontur

Für das definierte Verfahren der Achsen der Maschinenkinematik bildet die sogenannte Verfahrkontur die Grundlage. Über die Verfahrkontur wird der Pfad auf einem Werkstück vorgegeben, entlang dessen sich die Achsen der Maschinenkinematik bewegen sollen, um die Aufgabe zu erfüllen, welche von der Maschine gefordert wird.

Mit Geradenstücken und Kreisbögen lassen sich damit komplexe Verfahrkonturen bewerkstelligen. Dabei könnte das Template Verfahrkonturen im dreidimensionalen Raum vornehmen, wobei die Verfahrkonturen im zweidimensionalen Raum ausreichen, ich also auf einer X-Achse den Schlitten bewege, und auf dieser senkrecht verlaufend den Schlitten auf der Y-Achse in einer Ebene bewege. Und falls beide kombiniert bewegt werden kann ein Kreis- oder Diagonale dargestellt werden. Zusätzlich gibt es im Konturelement einer Verfahrkontur die Konturelementspezifischen Zusatzwerte "H-Funktion" und "M-Funktion" mit dessen Verwendung z. B. während der Abarbeitung der Kontur ein Laser oder Wasserstrahl ein- oder ausgeschaltet werden kann.

Die Verfahrkontur besteht dabei immer aus einem zusammenhängenden Pfad, entlang dessen sich die Achsen der Maschinenkinematik bewegen sollen.

Für den Start eines Konturelementes gilt:

- die aktuelle Position der Maschinenachsen ist der Ausgangspunkt für das erste Konturelement einer Verfahrkontur, daher ist mein Startpunkt nach dem Referenzieren immer genau definiert vorgegeben.
- der Endpunkt des vorherigen Konturelementes einer Verfahrkontur ist der Startpunkt für ein neues Konturelement einer Verfahrkontur

Auf Grund dieser beiden Bedingungen ist es ausreichend dass für jedes Konturelement jeweils nur ein Typ des Konturelementes anzugeben ist, wie z. B.: Gerade, Kreisbogen, oder den gewünschten Endpunkt eines Konturelementes.

Für das Template sind bestimmte Anfahrstrategien vorgesehen:

Da als Startpunkt für das erste Konturelement einer Verfahrkontur immer die aktuelle Position der Maschinenachsen verwendet wird, wird empfohlen bei der Definition einer Verfahrkontur auch eine genaue Anfahrstrategie zu berücksichtigen.

Wird z. B. die Bearbeitung eines Werkstückes während einer Verfahrkontur abgebrochen, so stehen die Maschinenachsen oft in einer Position, welche für die neue Verfahrkontur ungeeignet ist. Daher sollte als erstes Konturelement einer Verfahrkontur immer eine Gerade in Richtung des gewünschten Startpunktes der Kontur vorgegeben werden.

Möchte man jedoch ganz sicher gehen, dann sollte die Anfahrstrategie auf insgesamt drei Geraden erweitert werden um eine Kollision mit dem Werkstück oder der Maschine zu verhindern:

- durch die erste Gerade werden die Achsen der Maschinenkinematik auf eine Sicherheitsebene über dem Werkstück positioniert, in der alle Achsen ohne Kollision mit einem Werkstück verfahren werden können

- durch eine zweite Gerade werden die Achsen in der Sicherheitsebene über dem Startpunkt der Kontur positioniert
- durch eine dritte Gerade wird dann abschließend der eigentliche Startpunkt der Kontur auf dem Werkstück angefahren.

Diese Option, den genauen Anfahrtpunkt bei abgebrochenem Verfahrtauftrages zu definieren habe ich in meiner Steuerung nicht mit eingebunden, und erwähne sie hiermit um bei einer eventuellen Erweiterung diesen Schritt im Anwenderprogram mit einzubeziehen.

4.2.3 Festlegung eines geeigneten Koordinatensystems

Das Koordinatensystem ist notwendig um eine Achse oder Schlitten in einem definiertem Bereich selbstständig zu bewegen, diesem System ein Bezugspunkt zu geben und es dient zur eindeutigen Bestimmung der Position von Punkten und Objekten in einem geometrischen Raum.

Es gibt für die Interpolation in der Technologie-CPU drei verschiedenen Koordinatensysteme:

- Das Maschinen-Koordinatensystem bzw. -Achsen-Koordinatensystem (MCS). In diesem Koordinatensystem sind die einzelnen Achsen der Maschine enthalten, sie müssen kein Kartesisches Koordinatensystem bilden, müssen nicht im 90° Winkel zueinander ausgerichtet sein, sondern sind passend zur gewünschten Kinematik im Raum verteilt.
- Das Basis-Koordinatensystem (BCS) wird als grundlegendes Koordinatensystem zur Programmierung einer Verfahrkontur dargestellt. Es wird als rechtshändiges, kartesisches Koordinatensystem angegeben, daher sind die Achsen zueinander jeweils im 90° Winkel angeordnet. Dies kann mit der "rechten-Hand-Regel" verdeutlicht werden:
 - Daumen zeigt in Richtung der X-Achse
 - Zeigefinger zeigt in Richtung der Y-Achse
 - Mittelfinger zeigt in Richtung der Z-Achse
- Das Object Coordinate System (OCS), welches ebenfalls ein rechtshändiges kartesisches Koordinatensystem darstellt, das durch Verschiebung und Drehung aus dem Basis-Koordinatensystem gebildet werden kann. Man kann damit schnell und einfach eine Bearbeitung von Werkstücken erreichen, welche nicht parallel zum Basis-Koordinatensystem (BCS) liegen. Dabei wird lediglich das Basis-Koordinatensystem so verschoben und eventuell verdreht, dass das daraus resultierende Objekt-Koordinatensystem (OCS) parallel zu der Ebene des Werkstückes zum liegen kommt, welches bearbeitet werden soll.

Für meine Anwendung entscheide ich mich für das Basis-Koordinatensystem.

4.2.4 Programmierung von Konturelementen

Die Programmierung einer Verfahrkontur geschieht durch die Aneinanderreihung von verschiedenen Konturelementen. Über die Auswahl des Element-Typs und der Parametrierung der Endposition bzw. zusätzlicher Positions- oder Parameterwerte werden die Konturelemente definiert. Dabei erfolgt die Definition der Endpositionen der Konturelemente im Basis-Koordinatensystem (BCS) oder im Objekt-Koordinatensystem (OCS), also in einem rechtshändigen, kartesischen Koordinatensystem.

4.2.5 Absolute oder relative Programmierung, Endposition

Um die **Endposition** zu programmieren gibt es zwei verschiedene Möglichkeiten.

- **Die absolute Programmierung:** als Endposition wird die absolute Position im Basis-Koordinatensystem (BCS) bzw. Objekt-Koordinatensystem (OCS) programmiert, daher werden die genauen Koordinaten in X-, Y- und Z-Richtung im entsprechenden Koordinatensystem angegeben.
- **Die relative Programmierung:** als Endposition des Konturelements wird die relative Position im Basis-Koordinatensystem (BCS) bzw. Objekt-Koordinatensystem (OCS) programmiert, daher wird der Abstand der Endposition zum Startpunkt des Konturelements in X-, Y- und Z-Richtung im entsprechenden Koordinatensystem angegeben.

Es sind beide Programmierarten vollkommen gleichbedeutend. Es stehen jedoch für eine absolute und relative Programmierung unterschiedliche Befehle zur Verfügung. Je nach Situation ist die eine oder andere Programmierung besser. So kann z. B. die relative Programmierung mit $X=10$ leicht zur Anwendung gebracht werden, wenn mit Hilfe einer Geraden eine Distanz von 10mm in Richtung der X-Achse des Koordinatensystems überbrückt werden soll. Dies ist auch der Grund warum ich mich in meiner Programmierung des Konturelements für die relative Programmierung entschieden habe.

4.2.6 Absolute und relative Programmierung, Gerade

Um eine lineare Bewegung zwischen zwei Punkten zu realisieren wird das Konturelement-Typ "**Gerade**" benutzt. Dabei kann die Gerade beliebig im Raum (3D) positioniert werden.

- **Absoluten Programmierung:** bei einer Geraden wird der Endpunkt der Geraden als absolute Position im Koordinatensystem (BCS oder OCS) angegeben.
- **Relative Programmierung:** bei einer Geraden wird der Abstand des Endpunkts der Geraden bezogen auf den Startpunkt der Geraden im Koordinatensystem (BCS oder OCS) angegeben.

4.2.7 Absolute und relative Programmierung, Kreisbogen

Eine zirkulare Bewegung zwischen zwei Punkten kann über den Konturelement-Typ "**Kreisbogen**" realisiert werden. Dabei kann der Kreisbogen beliebig im Raum (3D) positioniert werden. Jedoch gilt für eine vollständige Definition eines Kreisbogens, dass zusätzlich zur Programmierung des Kreis-Endpunkts noch die Vorgabe eines Zwischenpunkts auf dem gewünschten Kreisbogen bzw. die Vorgabe des Kreis-Mittelpunkts zusammen mit der Auswahl der gewünschten Bogensegments notwendig ist. Es wird für die Verfahrbewegung des Konturelements "Kreisbogen" eine Ebene zwischen Startpunkt, Mittelpunkt und Endpunkt bzw. Startpunkt, Zwischenpunkt und Endpunkt des Kreises aufgespannt, in der der Kreisbogen realisiert wird. Es wiederum unterschieden zwischen absoluter und relativer Programmierung:

- Dabei wird bei der **absoluten Programmierung** eines Kreisbogens sowohl der Endpunkt, als gleich auch der Mittelpunkt bzw. ein Zwischenpunkt auf der Kreisbahn als absolute Position im Koordinatensystem (BCS oder OCS) angegeben.
- bei der **relativen Programmierung** eines Kreisbogens wird entweder der Abstand vom Startpunkt zum Endpunkt + Abstand Startpunkt zum Mittelpunkt, oder der Abstand vom Startpunkt zum Endpunkt + Abstand zum Zwischenpunkt angegeben.

Wird dabei die relative Variante: Abstand vom Startpunkt zum Endpunkt + Abstand vom Startpunkt zum Mittelpunkt gewählt, und liegen Startpunkt, Mittelpunkt und Endpunkt zusammen auf keiner gemeinsame Linie, so muss noch definiert werden welcher Kreisausschnitt gemeint ist, der mit dem kurzen Bogensegment (Öffnungswinkel kleiner 180°) oder der mit dem langen Bogensegment (Öffnungswinkel größer 180°).

4.2.8 Absolute und relative Programmierung "Kreis"

Mit dem Konturelement-Typ "Kreis" können auch zirkulare Bewegungen mit "mehreren Umdrehungen" realisiert werden. Dabei wird die Lage des Kreises über den Kreismittelpunkt definiert. Über die Definition eines Winkels (kann auch größer als 360° sein) wird dabei die Verfahrbewegung auf dem Kreisbogen vorgenommen. Dabei legt das Vorzeichen des Winkelwertes die Bewegungsrichtung auf dem Kreisbogen fest.

Es wird definiert, dass diese Kreise immer nur in einer Ebene liegen können, aber nicht beliebig in einem Raum. Um diese Einschränkung umgehen zu können, und Kreisbewegungen dennoch im Raum zu gestalten, kann man jedoch mit einem Funktionsbaustein FB 480 "MC_SetCartesianTransform" und dem benutzen des Objekt-Koordinatensystems dieses nach Wunsch verschieben, oder drehen. Dabei muss die Kreisbewegung ebenfalls in OCS ausgeführt werden.

Um das Konturelement Kreis zu programmieren ist auch hier wieder die absolute- oder relative Programmierung möglich:

- zur **absoluten Programmierung** eines Kreises wird der Mittelpunkt als absolute Position im Koordinatensystem (BCS oder OCS) angegeben.

- zur **relativen Programmierung** eines Kreises wird für den Mittelpunkt der Abstand des Mittelpunkts bezogen auf den Startpunkt des Kreises im Koordinatensystem (BCS oder OCS) angegeben.

Zusammenhang zwischen Winkelwert und Bewegungsrichtung:

Über das Vorzeichen des Winkels wird die Bewegungsrichtung der zirkularen Bewegung bestimmt. Dabei ist z. B. eine Bewegung in Urzeigersinn mit einem negativem Vorzeichen des Winkels zu verstehen.

4.2.9 Speicherung einer Verfahrkontur

Damit die T- CPU die gewünschte Verfahrkontur abrufen und abarbeiten kann, ist es notwendig die Verfahrkontur mit den passenden Parameter in einem geeignetem Datenbaustein in der CPU abzulegen. Dabei sind Parameter mit verschiedenen Bereichen einzubinden:

- Interne Parameter welche durch das Technology Template verwaltet werden wie z. B. der Eingang: "Execute" Befehl zur Ausführung der Kontur
- Allgemeine Parameter, welche sämtliche Voreinstellungen beinhalten, wie z. B. Geschwindigkeit, Ruck, Beschleunigung usw.
- Parameter zum Ort und welche Kontur verfahren werden soll, in welchem alle Angaben zur Position vorliegen, wie z. B. eine Gerade mit Start- Endposition, ein Kreisabschnitt mit Start, Mittelpunktposition und Endpunkt, usw.

Für diese Aufgaben mache ich mir den UDT 541 "ContouListData" ein Datenbaustein, welcher als UDT (user defined type) generiert ist zu nütze, und bereits durch das integrieren des Templates in mein STEP 7 Projekt im Ordner Bausteine vorhanden ist. In diesem bereits für das Template vorgefertigten programmierten Datenbaustein sind alle benötigten Datensatzabfragen bereits beinhaltet, wie z. B.:

- Konturelement-Typ wie z. B.: Kreis, Gerade
- Endpunkt, Zusatzpunkt
- Konturelementspezifische Zusatzwerte (M-und H Funktion, um z. B. einen Laser einzuschalten während der Bewegung) welche bei diesem Kreuztischaufbau jedoch nicht verwendet werden.

Dieser Inhalt wiederholt sich hundert mal, und somit ist es später möglich 100 verschiedene aufeinander abfolgende Verfahrbefehle zu fahren.

4.3 Abarbeiten einer Verfahrkontur

Für die Ausführung der gewünschten Verfahrkontur steht in der Technologie- CPU eine entsprechende Technologiefunktion zur Verfügung, welche die entsprechende Abarbeitung der Bewegung durchführt.

Um die Abarbeitung der Verfahrbewegung zu starten wird die entsprechende Technologiefunktion in der Technologie-CPU gestartet. Man erhält über die Statussignale der Technologiefunktion eine genaue Aussage über den aktuellen Zustand der Bewegung. So kann je nach Kombination der Signale "Busy" und "Activ" der genaue Status des Bewegungsauftrags für die Interpolierte Bewegung ermittelt werden.

4.3.1 Verkettung der Verfahrkonturen

Um eine gleichmäßige Gesamtkontur, welche aus vielen einzelnen Verfahrkonturen besteht in einem harmonischem Ablauf verfahren zu können, ist es notwendig, dass die Konturen aneinander gekettet, mit einander verschliffen werden. Dafür fasst der Auftragspuffer der Technologie- CPU immer genau zwei Bewegungsbefehle pro abzuarbeitende Bewegung. Dies wird im folgenden Schema gezeigt.

- I. Die erste Abarbeitung der Bewegung wird in der Technologie- CPU gestartet. Dazu wird der der Befehl gegeben, in meinem Fall durch das Betätigen des Links Tasters auf dem Frontblech des Schaltschranks, und von der T- CPU sofort zur Ausführung gebracht, und durch die Ausgänge ("Busy", "Aktiv") des Funktionsbausteins "MotionList_Basic" angezeigt.
- II. Bereits während dieser erste Befehl für die erste Abarbeitung läuft, muss sofort die Abarbeitung für die zweite Bewegung gestartet werden, so dass diese mit dem Ende der ersten Abarbeitung verschliffen werden kann. Dabei sollte es zu keinem Einbruch der Bahngeschwindigkeit kommen.
- III. Sobald am Ausgang "Done" des Funktionsbausteins das "True" Signal für die erste Abarbeitung erscheint, die zweite Abarbeitung in Bewegung ist, wird bereits die dritte Abarbeitung eingelesen.
- IV. Dieser Zyklus setzt sich solange fort, bis keine weiteren Verfahrkonturbefehle mehr vorliegen.

4.3.2 Konfiguration und Projektierung

Um mit dem Funktionsbaustein FB 540 eine Figur zu fahren reicht es nicht jede Achse einzeln zu dirigieren, sondern es müssen beide im Verbund betätigt werden, darum wird in der Parametrierung beim Anlegen der Maschinenachsen die Bahninterpolation gewählt. Somit erfolgt die Ansteuerung der Maschinenachsen durch das Technology Template mit Hilfe eines Bahnobjekts.

Dabei übergibt das Technology Template die gewünschte Achsposition im Basis-Koordinatensystem (BCS) oder Objekt-Koordinatensystem, an das Bahnobjekt, welches über die im Bahnobjekt enthaltene kinematische Transformation die Ansteuerung der Maschinenachsen übernimmt. Damit kann das Technology Template jede enthaltene Kinematik der S7- Technologie verwenden.

Für die weitere Projektierung wird noch die gewünschte Kinematik ausgewählt und damit die Kinematik-Transformation des Bahnobjekts festgelegt, sowie über den Kinematik-Typ auch der "Bearbeitungsraum", in meinem Fall zweidimensional in der Ebene festgelegt. Als Maschinenkinematik wird das Kartesische Portal ausgewählt.

4.3.3 Schnittstellen des FB540 "MotionList_Basic"

Der Funktionsbaustein FB 540 "MotionList_Basic" hat folgende **Eingänge**:

- AxesGroup: Nummer des Bahnobjekts, welches durch das Technology Template beeinflusst werden soll. Eingabe in INT.
- DB_Contour: Nummer des Datenbausteins, der die gewünschte Kontur bzw. Verfahrtablette enthält. Eingabe in INT.
- Execute: Annahme des Start und Stopp Befehls für die Abarbeitung der im Datenbaustein hinterlegten Kontur bzw. Verfahrbewegung. Eingabe in BOOL.
- Velocity: Vorgabe der Geschwindigkeit. Eingabe in REAL.
- Acceleration: Vorgabe der Beschleunigung. Eingabe in REAL.
- Deeleration: Vorgabe der Verzögerung. Eingabe in REAL.
- Jerk: Vorgabe des Rucks. Eingabe in REAL.
- CoordSystem: Auswahl des Koordinatensystems BSC oder OCS. Eingabe in INT.
- DynamicAdaption: Dynamikanpassung an die Konfigurierten Dynamikgrenzwerte der Bahnachsen. Dabei kann man wählen ob die Konfigurierten Dynamikgrenzwerte der einzelnen Bahnachsen benützt werden, oder nicht. Eingabe in INT
- Tolerance: Es wird die Zulässige Abweichung der einzelnen Anfahrpunkte eingestellt. Eingabe in INT.
- WaitIN: Eingang für ein Unterbrechungssignal, welches die Verfahrbewegung stoppt. Eingabe in BOOL.
- MotionInterrupt: Eingang für ein Unterbrechungssignal mit der Möglichkeit, die gestoppte Verfahrbewegung an der Unterbrechungsstelle wieder fortzuführen. Eingabe in BOOL.
- MotionContinue: Eingangssignal welches das MotionInterrupt wieder fortsetzt. Eingabe in BOOL.
- MotionAbort: Verfahrbewegung wird abgebrochen, und kann nicht mehr fortgesetzt werden. Eingabe in BOOL.
- VelocityOverride: Damit wird die Geschwindigkeitsübersteuerung eingestellt. Eingabe in REAL.

Ausgänge des FB 540:

- Done: Kontur ist abgeschlossen. Ausgabe in BOOL.
- Busy: In Verfahrbetrieb. Ausgabe in BOOL.

- Active: Befehl für den Fahrbetrieb liegt an. Ausgabe in BOOL.
- Stop: Verfahrbewegung gestoppt. Ausgabe in BOOL.
- CommandAborted: Verfahrbewegung wurde von außerhalb gestoppt, z. B. durch Erreichen eines Endschalters. Ausgabe in BOOL.
- Error: Bearbeitungsfehler. Ausgabe in BOOL.
- ErrorID: Fehlercode, dieser kann über eine Fehlertabelle welche im Funktionshandbuch genauer erläutert werden. Ausgabe in WORD.
- ErrorSource: zusätzlicher Fehlercode zur Lokalisierung des Fehlers innerhalb des Bausteins. Ausgabe in WORD.
- ActSetVelocity: momentane Verfahr­geschwindigkeit. Ausgabe in REAL.
- ActSetAcceleration: momentane für die Ausführung eingestellte Beschleunigung. Ausgabe in REAL.
- ActSetDeceleration: momentane für die Ausführung eingestellte Verzögerung. Ausgabe in REAL.
- ActSetJerk: momentan eingestellter Ruck für die Ausführung der Verfahr­bewegung. Ausgabe in REAL.
- ActualContourDB: momentane Nummer des Datenbausteins, welcher gerade zur Verfahr­bewegung benützt wird. Ausgabe in INT.
- ActualListLine: momentane Zeilennummer der Verfahrkontur. Ausgabe in INT.
- ActualCoordSystem: momentan benütztes Koordinatensystem. 0=BCS und 1=OCS. Ausgabe in INT.
- M_Funtion_OUT: das Aufzeigen, wenn z.B. gerade ein Laser, Bohrer zusätzlich eingesetzt wird. Ausgabe in INT.
- H_Funktion_OUT: das Aufzeigen, wenn z.B. gerade ein Laser, Bohrer zusätzlich eingesetzt wird. Ausgabe in REAL.

4.3.4 Erstellen des Anwenderprogramms

Beim Erstellen des Anwenderprogramms baue im OB1 auf der bereits vorhandenen Handsteuerung auf, und knüpfe mein neues Anwenderprogramm im Automatikbetrieb an das vorherige Programm der Handsteuerung an.

Beide Achsen, X-Achse, sowie Y-Achse erhalten für die Freigabe jeweils den Baustein "MC Power". Dazu kommt nun im Automatikbetrieb das Referenzieren der X-Achse der Achse mit dem Inkrementalgeber, welches ich mit dem Baustein "MC_Home" abwickle. Zum setzen, und nicht zum Referenzieren wird der selbige Baustein auch für die Y- Achse mit Absolutwertgeber verwendet. Dazu wird lediglich der Eingang Mode des Bausteins verändert und dieser auf Absolutwertgeberjustage gestellt.

Nachdem beide Achsen gesetzt sind, lasse ich sie jeweils mit dem Baustein "MC_MoveAbsolute" auf die jeweilige Startposition fahren.

Als weitere Option benötige ich für etwaige Fehler noch die Möglichkeit, wenn diese durch die Fehlerausgabe des FB540 identifiziert, und beseitigt sind, dass ich danach die Anlage wieder in Betrieb nehmen kann. Dies geschieht durch die Bausteine "MC_Reset" jeweils für die einzelnen Achsen, sowie einen zusätzlichen für das Bahnobjekt, welches den Verbund beider Achsen darstellt, und als eigene Funktion gesehen werden kann. Auch das Bahnobjekt bekommt einen "MC_Reset" zugewiesen.

4.3.5 Bedienung im Automatikbetrieb

Als ersten Schritt muss die Steuerung mit dem Dreiphasendrehschalter eingeschaltet werden, damit die T-CPU mit Spannung versorgt wird.

Im Zweiten Schritt muss der Computer zum Programmieren eingeschaltet werden, um eine Figur im Datenbaustein UDT zu programmieren. Um eine Figur fahren zu können wird zuerst im UDT 20-Baustein der T-CPU die Tabelle des Datensatzes mit den entsprechenden Daten gefüllt. Ich wähle in meinem Beispiel eine Gerade und direkt folgend eine Kreisbahn, um zu sehen ob die Aufträge wie gewünscht ineinander verkettet sind, und dies auch optisch gut nach zu verfolgen ist. Es ist möglich diese Einträge jederzeit durch überschreiben wieder zu ändern. Der UDT 20 Baustein ist im SIMATIC Manager, welchen ich geöffnet habe, im Ordnerverzeichnis CPU 315-2DP unter S7- Programm, Bausteine zu finden. Für eine Kontur wähle ich im UDT 20 folgende Konturelemente aus:

- I. Für eine Gerade trage in dem Feld "ContourCommand" die Befehlsnummer des Befehls ein. Diese Befehlsnummern sind in der Applikationsbeschreibung dieses Template "MotionList Basic" [10] zu finden, und im Anhang als Tabelle ersichtlich. Abb. 4.3.4. Als Befehlsnummer für eine Gerade wähle ich einen Befehl, der es mir erlaubt eine Gerade relativ zum Startpunkt zu setzen. Für diesen Fall wähle ich die Befehlsnummer drei mit der Bezeichnung "MoveLinearRelative", und trage die Nummer drei in das ContourCommand Feld ein. Als nächsten Schritt muss noch der Zielpunkt vorgegeben werden. Startpunkt ist der Punkt an welchem die X- und Y-Achsen bereits stehen, und die Achsen nach dem Referenzieren automatisch anfahren werden. Für den Zielpunkt trage ich im "Feld EndPoint" für X, und Y in der jeweiligen Zeile die passende Werte ein. Die Zeile für den Y-Wert bleibt leer, da ich die Achsen nur auf einer Ebene bewege, und diese für einen dreidimensionalen Raum wäre. Dieser Endpunkt soll mir ein Diagonale Linie nach rechts oben auf meinem Kreuztisch abfahren.
- II. In der darauffolgenden Konturliste, welche mit nach unten scrollen im Datenbaustein UDT 20 als nächstes aufzurufen ist, werde ich die Kontur für einen Kreisbogen eintragen. Als Befehlsnummer wähle ich die dreizehn mit der Bezeichnung "MoveCirclesRelative". Dies ist der Befehl zur Erzeugung einer zirkularen Verfahrbewegung in einer vorgegebenen Ebene des Koordinatensystems unter Angabe des Abstands des Mittelpunkts des Kreisbogens, bezogen auf den Startpunkt des Kreisbogens, und des Öffnungswinkels. Mit dem Befehl können auch zirkuläre Verfahrbewegungen mit mehreren Umdrehungen durch Angabe eines Öffnungswinkels welcher größer als 360° ist erzeugt werden. Dabei ist die Bewegungsrichtung der zirkulären Verfahrbewegung in der vorgegebenen Ebene über das Vorzeichen des Öffnungswinkels festgelegt. Die Befehlsnummer 13 trage ich die Zeile "ContourCommand" für den Kreisbogenauftrag ein. Im Feld "Endpoint" trage ich in der X- Zeile den Wert des Winkels ein, und kann durch wählen des Vorzeichens die Drehrichtung vorgeben. Im Feld "AuxPoint" trage ich nun noch in die X- Zeile die Entfernung vom Startpunkt (=Endpunkt der vorherigen Geraden) zum Kreismittelpunkt in X- Richtung ein, und in die Y- Zeile trage ich die Entfernung vom Startpunkt zum Kreismittelpunkt in Y- Richtung ein.
- III. Eine weitere Diagonale Richtung ursprünglichen Startpunkts füge ich in die dritte Konturliste ein. Nun ist meine komplette dreiteilige Kontur programmiert, und wird nach dem Startbefehl ausgeführt.

Diese Änderung im Datenbaustein UDT 20 wird nun noch in die Technologie- CPU geladen, und steht jetzt bereit zum Abruf.

Als nächsten Schritt in der Bedienung wird der Schlüsselschalter von Hand auf Automatik gestellt. Der Freigabetaster für das Dreiphasenschütz betätigt, welcher das 380 V Versorgungsnetz für das Smartlinemodule durchschaltet, und wie im Handbetrieb auch die einzelnen Achsen am Freigabeschalter auf dem Bedienblech des Schaltschranks betätigt.

Sobald die Freigabe für die X- Achse betätigt wird, führt diese Achse die Referenzierung aus um ihre genaue Position zu ermitteln in welcher Sie momentan steht. Beide Achsen X- und Y-Achse fahren nach durchgeführter Positionsermittlung automatisch an den Startpunkt, welcher im Anwenderprogramm vorgegeben wurde. Ab jetzt ist der Kreuztisch und seine Steuerung bereit eine Figur zu fahren.

Startbefehl erhält man durch betätigen des "X- Achse links" Tasters im Automatikbetrieb. Im Automatikbetrieb habe ich die vier Richtungstaster anderen Funktionen zugeordnet.

Daher kann ich mit dem Betätigen des Tasters "X- Achse rechts" im Automatikbetrieb nach einer Fehlerbehebung des Achsenbetriebs die Reset-Funktion aller drei Bahnobjekte auslösen. Zusätzlich sind Fehler auch über die S7- Technologie online über den Computerbildschirm ersichtlich um etwaige Fehler besser zu erkennen, sowie auch in den Ausgängen der einzelnen Funktionsbausteinen im Anwenderprogramm. Auch ist es möglich Fehler direkt in der S7- Technologie zu zurückzusetzen.

Damit sind alle Funktionen und somit die Bedienung des Kreuztisches im Automatikbetrieb beschrieben.

5. Zusammenfassung und Ausblick

Nach einigen Anfangsschwierigkeiten bei der Inbetriebnahme ist es nun möglich genau definierte Figuren mit dem Kreuztisch fahren zu können. Vorteilhaft ist, dass die Steuerung noch einige Zusatzfunktionen bietet, welche ich im Moment noch nicht benutze und daher eine Erweiterung sehr einfach zu gestalten ist, wie z. B.:

- H_ und M_ Funktion des Templates welches im UDT 20 über den Funktionsbaustein FB 540 "MotionList_Basic" im Automatikbetrieb aufgerufen werden kann, wodurch eine Fräse oder Laser im Verfahrbetrieb einfach dazu geschaltet werden kann.
- Einfache Integration einer HMI- Bedienoberfläche, um die Übersichtlichkeit und etwaige Erweiterung im Bedienbereich zu vereinfachen.
- Erweiterung zum Dreidimensionalen Raum. Die Z- Achse ist im Template schon eingebunden, und muss nur noch aktiviert werden. Damit lässt sich dann z. B. wie man es von einem Kreuztisch in der Praxis kennt, sehr einfach die Höhe über einem Werkstück einstellen, um damit z. B. Gravur Schilder mit einem Fräskopf zu fertigen.

Vorteilhaft bei der Konfiguration der Steuerung fand ich die vielen Funktionsbausteine welche die Einbindung der vielen Funktionen sehr vereinfachen, wie es auch beim Referenzieren der Fall war.

Ich denke, dass dieser Steuerungsaufbau sehr zeitgemäß ist, da ich dies auch aus der Praxis in einer Käserei kenne und ich dort ebenfalls mit der SPS- Steuerung, dem Profibus und den dazugehörigen Frequenzumrichtern für den Antrieb der Antriebsmotoren für sämtliche Abläufe zu tun hatte.

Dieser Aufbau ist aufgrund seiner verwendeten modernen Systemkomponenten sehr aufschlussreich über heutige Steuerungsvarianten in der Praxis, und ich denke für das Labor in dem die Steuerung aufgebaut ist, wird sie sehr hilfreich für die Studenten sein, um einen Praxisnahen Eindruck zu gewinnen.

Ich selbst konnte viel dazulernen, und freue mich, dass ich diese Steuerung aufbauen durfte. Dabei sind viele kleine Details und Probleme auf mich zugekommen, welche ich alleine durch die Theorie niemals erfahren hätte. Dies hat mir sehr viel Erfahrungen und Praxiswissen gelehrt.

6. Literaturverzeichnis

[1] Wikipedia: Freie Enzyklopädie, Sitz in San Francisco, USA, Suchbegriff: "HS-Kempton";
https://de.wikipedia.org/wiki/Hochschule_f%C3%BCr_angewandte_Wissenschaften_Kempton
(15.07.2016, 15:30 Uhr), gespeichert als PDF.

[2] Hochschule Kempten, Laborbeschreibung und Lageplan;
<http://www.hochschule-kempten.de/hochschule/labore-elektrotechnik/elektrische-energiesysteme/forschung.html>
(15.07.16, 18.00 Uhr), gespeichert als PDF.

http://www.hochschule-kempten.de/fileadmin/fh-kempten/HK/anfahrt_lage/Gebaeudeplan_farb_2012.pdf
(15.07.2016, 18:13 Uhr), heruntergeladen als PDF.

[3] Siemens AG: Erste Schritte und Übungen mit STEP 7, Getting Started, Dokumentation, 03.2006, 6ES7810-4CA08-8AW0;
https://cache.industry.siemens.com/dl/files/511/18652511/att_53987/v1/S7gsv54_d.pdf
(16.07.2016, 9:59 Uhr), heruntergeladen als PDF.

[4] Siemens AG: SINAMICS S120, Inbetriebnahmehandbuch, (IH1), 01/2011, 6SL3097-4AF00-0AP1. S. 90;
https://cache.industry.siemens.com/dl/files/703/49084703/att_61722/v1/IH1_012011_deu_de-DE.pdf
(24.06.2016, 14:51 Uhr), heruntergeladen als PDF.

[5] Siemens AG: SIMATIC, S7-300, Anbindung SINAMICS S120 an die Technologie-CPU, Produktinformation, 09/2011, A5E00480377-04. S. 15;
<https://support.industry.siemens.com/cs/document/30119684/simatic-s7-300-anbindung-sinamics-s120-an-die-technologie-cpu?dti=0&pnid=14383&lc=de-WW>
(28.06.16, 18:50 Uhr), heruntergeladen als PDF.

[6] Siemens AG, Produktbeschreibung:
<https://mall.industry.siemens.com/mall/de/WW/Catalog/Products/10216008>
(T-CPU allgemein, 28.06.2016, 19:00 Uhr), gespeichert als PDF.

<https://mall.industry.siemens.com/mall/de/WW/Catalog/Product/6ES7315-6TH13-0AB0>
(T-CPU spezifisch, 28.06.2016, 19:30 Uhr), heruntergeladen als PDF.

<https://mall.industry.siemens.com/mall/DE/WW/Catalog/Products/10045387>
(CU320-2DP, 28.06.2016, 20:00 Uhr), gespeichert als PDF.

<https://mall.industry.siemens.com/mall/de/WW/Catalog/Products/10268325>
(SMART LINE MODULE, 28.06.2016, 20:30 Uhr), gespeichert als PDF.

<https://mall.industry.siemens.com/mall/de/WW/Catalog/Products/10264887>

(DOUBLE MOTOR MODULE, 28.06.2016, 21:00 Uhr), gespeichert als PDF.

<https://mall.industry.siemens.com/mall/de/WW/Catalog/Product/1FK7042-2AF71-1AG1>
(SYNCHRONMOTOR 1FK7-1AG1 mit Inkrementalgeber, 29.06.2016, 17:00 Uhr)
heruntergeladen als PDF.

<https://mall.industry.siemens.com/mall/de/WW/Catalog/Product/1FK7042-2AF71-1RG1>
(SYNCHRONMOTOR 1FK7-1RG1 mit Absolutwertgeber, 29.06.2016, 17:30 Uhr),
heruntergeladen als PDF.

[7] Siemens AG: Technology Template: „MC_MoveJOG“ Technologie-CPU,
Dokumentation, 03/2009, V4.1, Beitrags-ID:21365191, und Programmarchiv;
https://support.industry.siemens.com/cs/document/21365191/technologie-cpus%3A-technologie-template-move_jog-?dti=0&lc=de-WW
(Dokumentation, 30.06.16, 19:31 Uhr), gespeichert als PDF.
(Programmarchiv, 30.06.16, 19:45 Uhr), gespeichert als ZIP-Datei.

[8] Siemens AG: SIMATIC Engineering Tools, S7-Technology, Funktionshandbuch, 10/2010
A5E00251797-07;
<https://support.industry.siemens.com/cs/document/30119663/simatic-engineering-tools-s7-technologie?dti=0&pnid=14383&lc=de-WW>
(02.7.2016, 9:23 Uhr), heruntergeladen als PDF.

[9] Wikipedia: Freie Enzyklopädie, Sitz in San Francisco, Suchbegriff: " We Transfer";
<https://de.wikipedia.org/wiki/WeTransfer>
(09.07.2016, 17:03), gespeichert als PDF.

[10] Siemens AG: Technology Template: „MotionList Basic“ Technologie CPU,
Applikationsbeschreibung 03/2012 und Programmarchiv;
<https://support.industry.siemens.com/cs/document/59259273/technologie-cpu%3A-technologie-template-%E2%80%9Emotionlist-basic%E2%80%9C?dti=0&lc=de-WW>
(Beschreibung, 16.07.2016, 17:30), gespeichert als PDF.
(Programmarchiv, 16.07.2016, 17:45), gespeichert als ZIP-Datei.

7. Anlagen

Abb. 2.4.2.1 Schaltschrank	I
Abb. 2.4.2.2 Kreuztisch Vorderansicht	II
Abb. 2.4.2.3 Kreuztisch Seitenansicht	II
Abb. 2.4.2.4 Leistungsteil	III
Abb. 2.4.2.5 Steuerungsteil	IV
Abb. 2.4.2.6 Steuerungsteil 1	V
Abb. 2.4.2.7 Steuerungsteil 2	V
Abb. 2.4.2.8 Graue Steuerleitung Endschalter	VI
Abb. 3.7.5 Anwenderprogramm	VII
Abb. 4.1.4 Aktives Referenzieren, gespiegelt	XV
"vgl." Abb. 4.1.6 Inkrementalgeberleitung	XVI
Abb. 4.1.6.1 Messversuchsaufbau	XVII
Abb. 4.1.6.2 Peak- Nullmarke	XVII
Abb. 4.1.6.3 Peak- Nullmarke verzerrt	XVIII
Abb. 4.3.4 Motion- Befehlsliste	XIX



Abb. 2.4.2.1 Schaltschrank

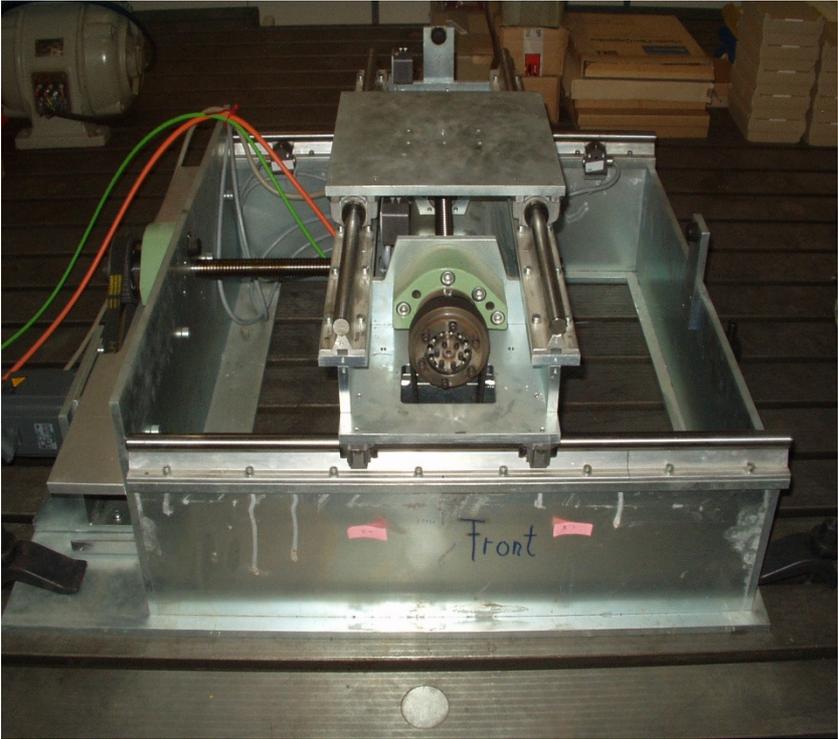


Abb. 2.4.2.2 Kreuztisch Vorderseite



Abb. 2.4.2.3 Kreuztisch Seitenansicht

Leistungsteil Kreuztisch

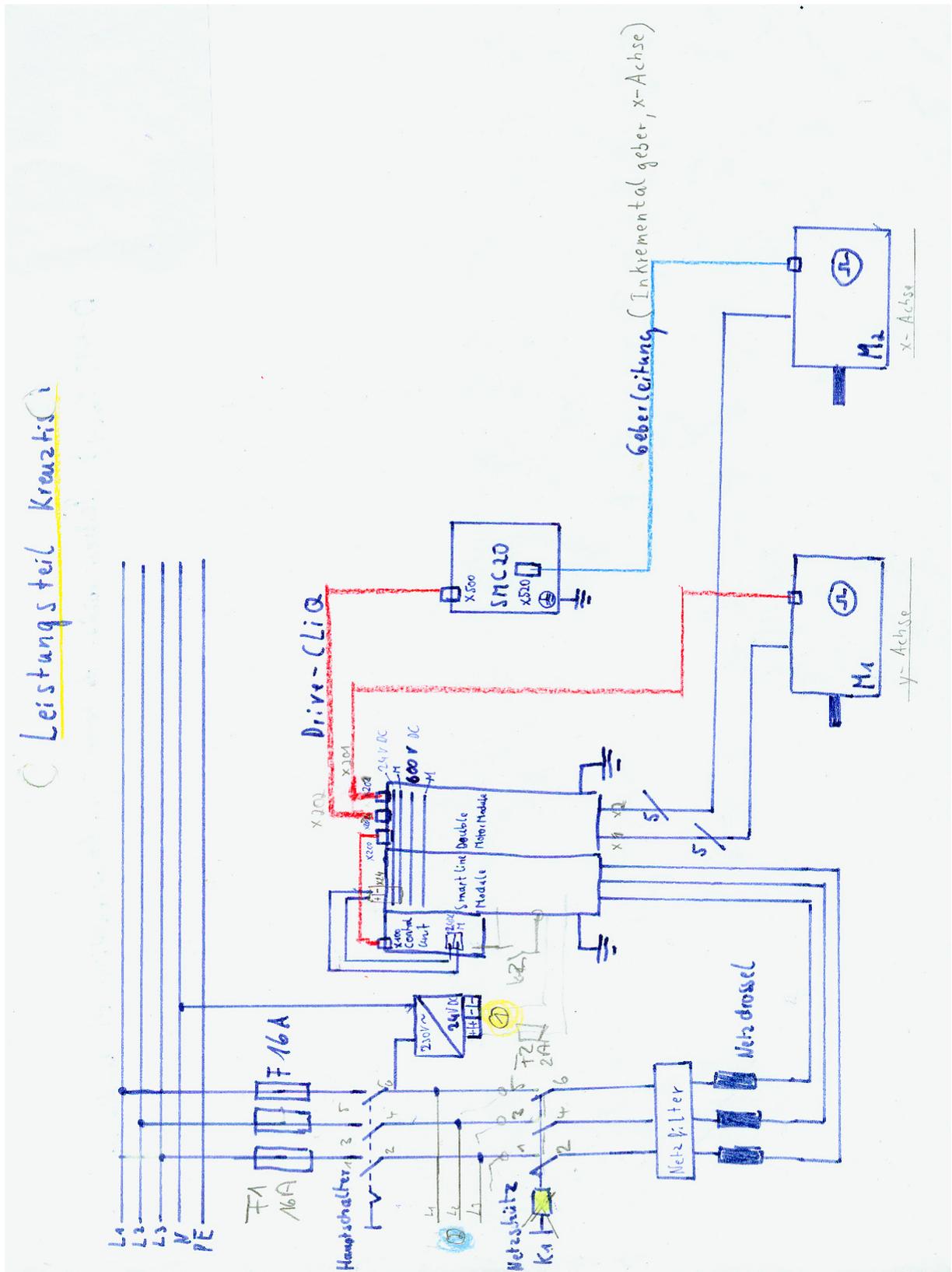


Abb. 2.4.2.4 Leistungsteil

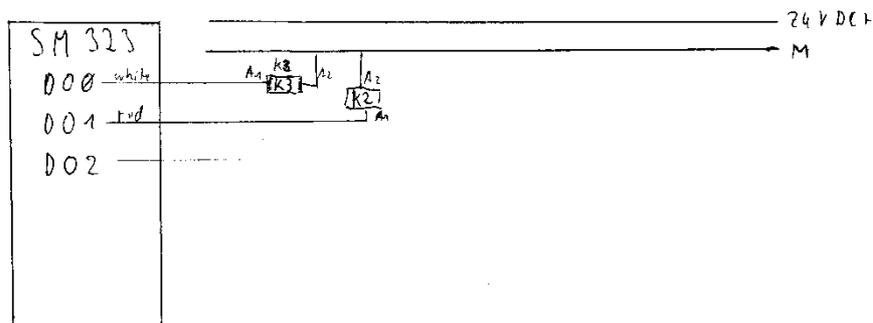
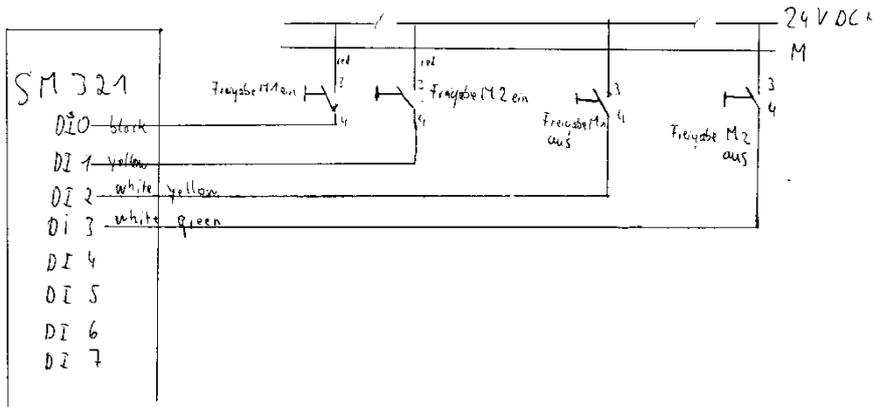


Abb. 2.4.2.6 Steuerungsteil 1

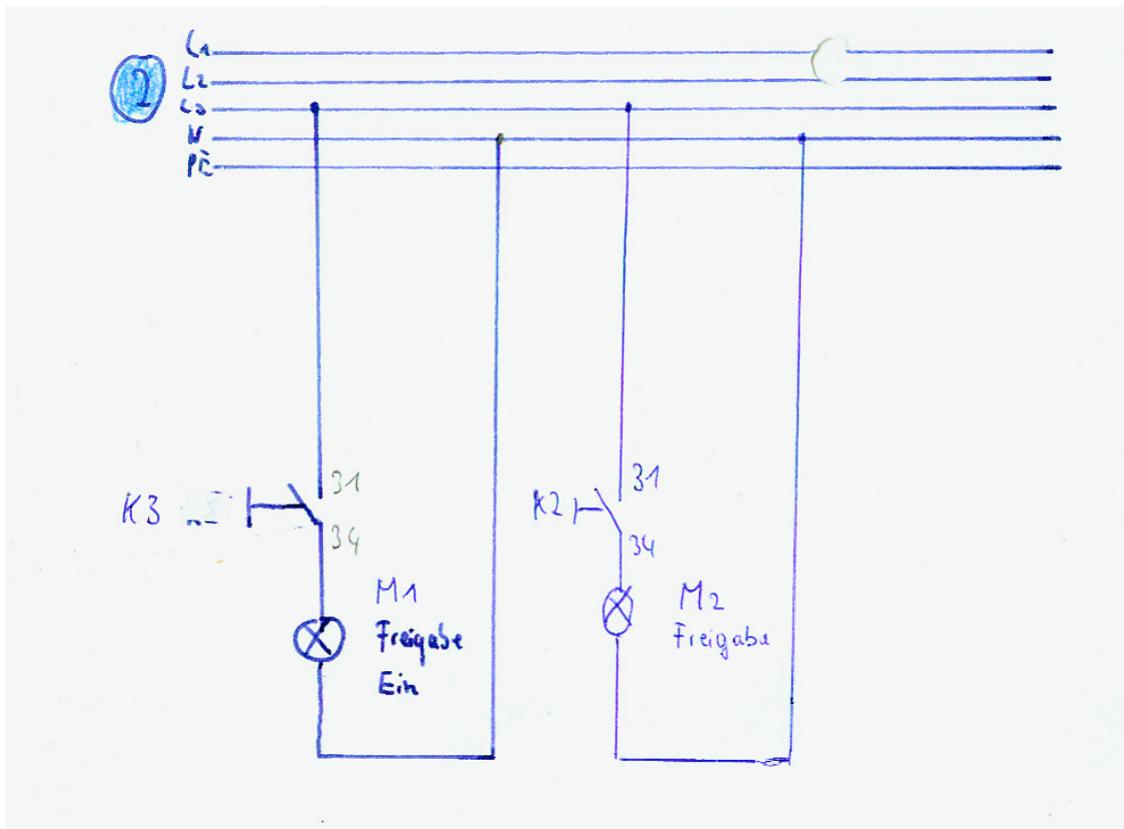


Abb. 2.4.2.7 Steuerungsteil 2

Pin	1	2	} 24V DC
	Rot Blau	Weiß Gelb	
Pin	14	15	} PE
	Grün Weiß	Braun Grün	
Pin	3	= XR	
	Grün Weiß		
Pin	4	= XL	
	Lilla		
Pin	5	= Y U	
	Braun		
Pin	6	= YO	
	Weiß		

Abb. 2.4.2.8 Graue Steuerleitung Endschalter

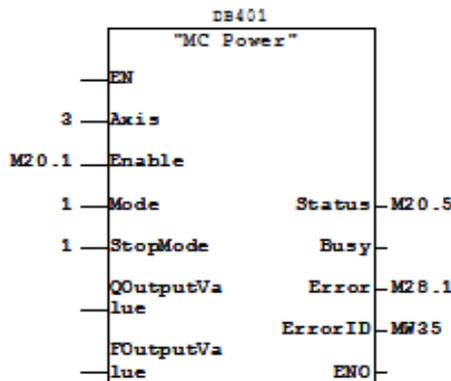
OB1 - <offline>

"CYCL_EXC" Cycle Execution
 Name: Familie:
 Autor: Version: 0.1
 Bausteinversion: 2
 Zeitstempel Code: 31.05.2016 15:15:23
 Interface: 15.02.1996 16:51:12
 Längen (Baustein / Code / Daten): 01910 01730 00026

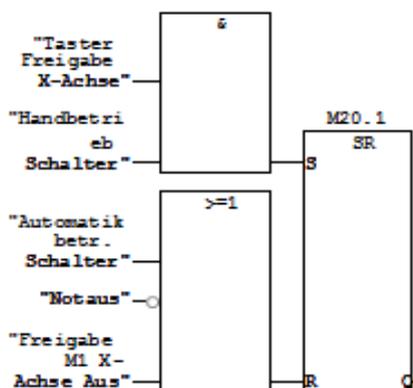
Name	Datentyp	Adresse	Kommentar
TEMP		0.0	
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date_And_Time	12.0	Date and time OB1 started

Baustein: OB1 "Main Program Sweep (Cycle)"

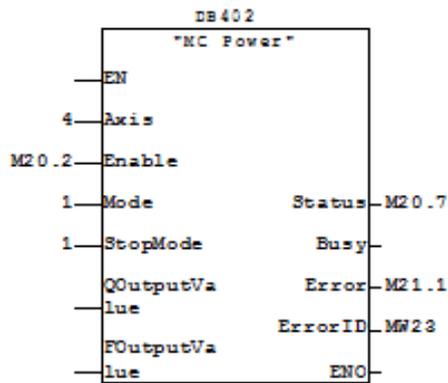
Netzwerk: 1 Baustein Freigabe X-Achse



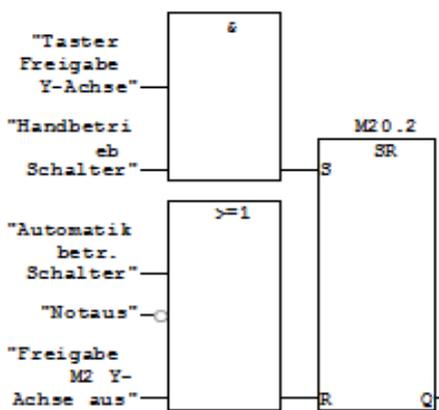
Netzwerk: 2 Taster selbshaltung Freigabe X-Achse Handbetrieb



Netzwerk: 3 Baustein Freigabe Y-Achse Handbetrieb

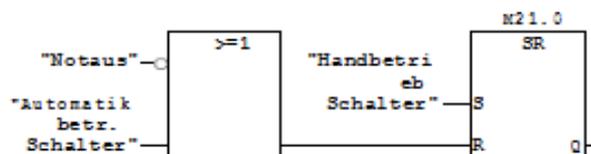


Netzwerk: 4 Taster selbsthaltung Freigabe Y-Achse



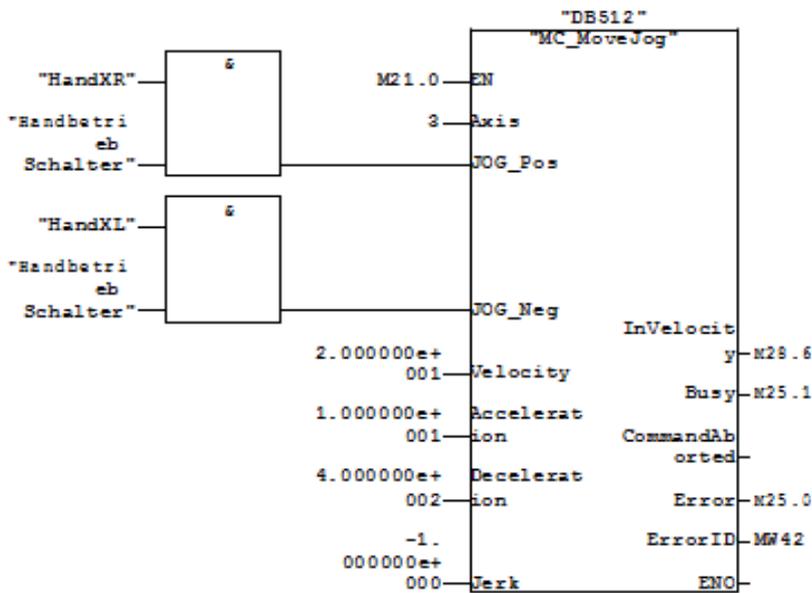
Netzwerk: 5 Handbetrieb+Notaus für Einschaltbedingung

Handbetrieb+Notaus



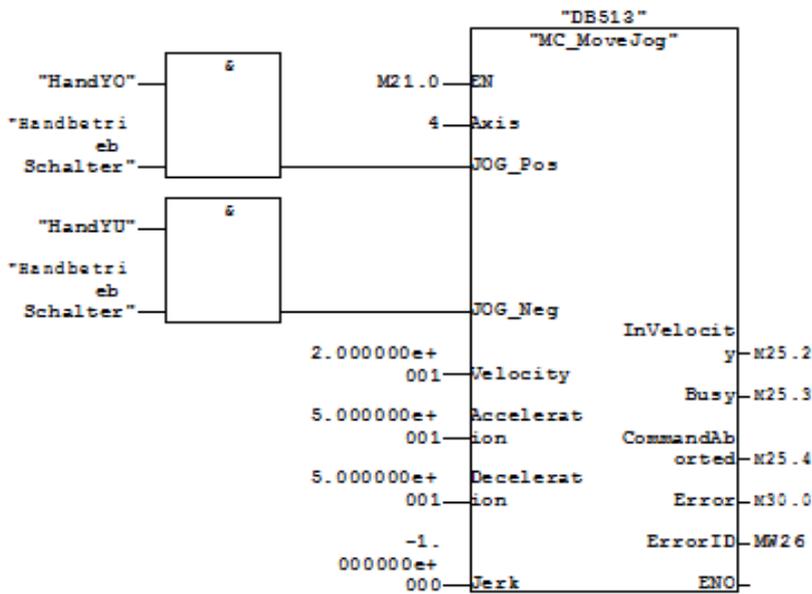
Netzwerk: 6 Template Baustein Handsteuerung X-Achse

X-Achse Handbetrieb

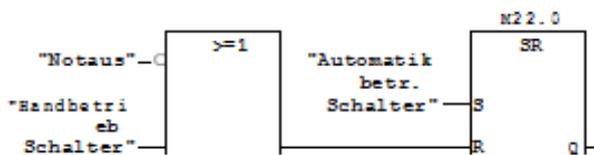


Netzwerk: 7 Template Baustein Handsteuerung Y-Achse

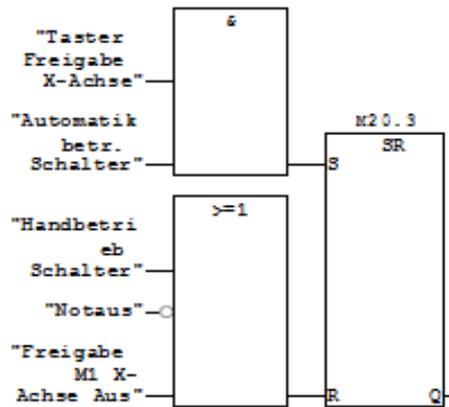
Y-Achse Handbetrieb



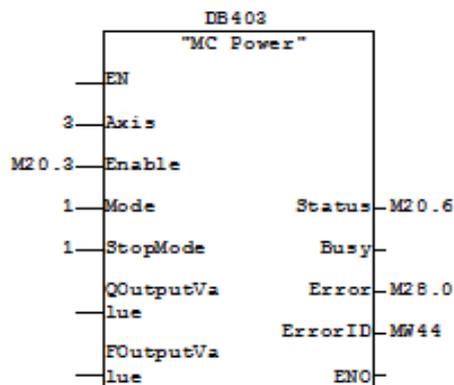
Netzwerk: 8 Automatikbetrieb+Notaus (Startvoraussetzung Automatikbetrieb)



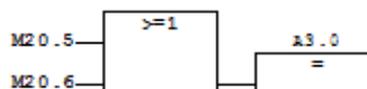
Netzwerk: 9 Voraussetzung Freigabe X-Achse Automatikbetrieb



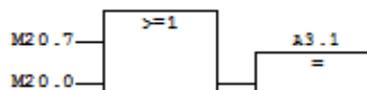
Netzwerk: 10 Baustein Freigabe X-Achse Automatikbetrieb



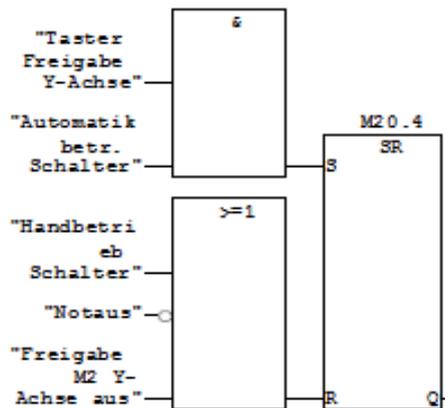
Netzwerk: 11 Verknüpfung Freigabekontrollleuchte X-Achse



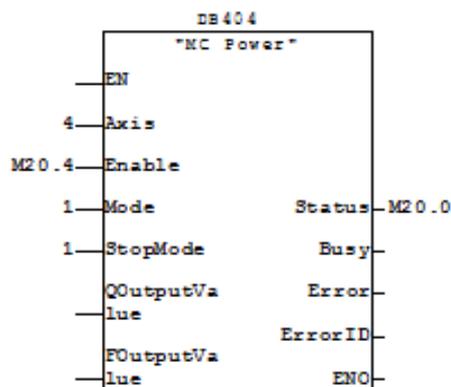
Netzwerk: 12 Verknüpfung Freigabekontrollleuchte Y-Achse



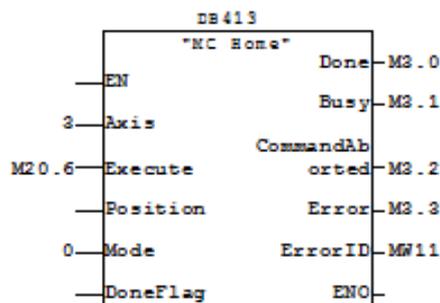
Netzwerk: 13 Voraussetzung Freigabe Y-Achse Automatikbetrieb



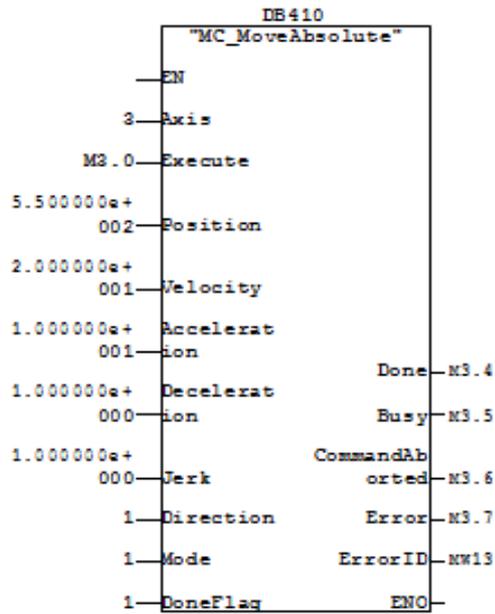
Netzwerk: 14 Baustein Freigabe Y-Achse Automatikbetrieb



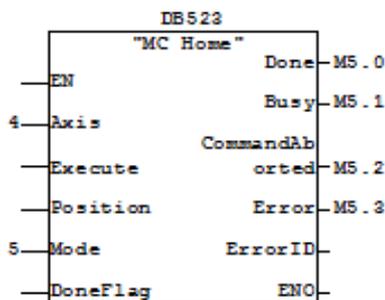
Netzwerk: 15 Referenzfahrt X-Achse Automatikbetrieb



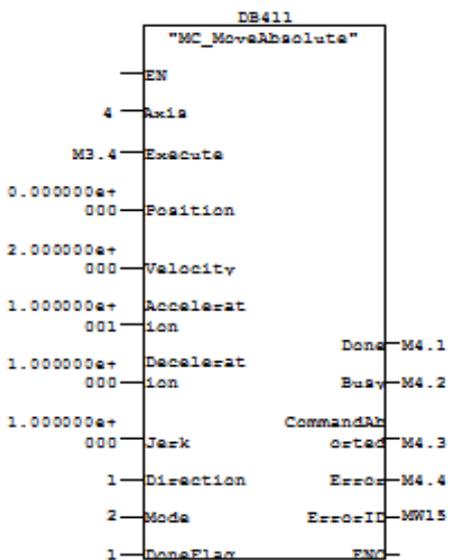
Netzwerk: 16 X-Achse positionieren zum Startpunkt



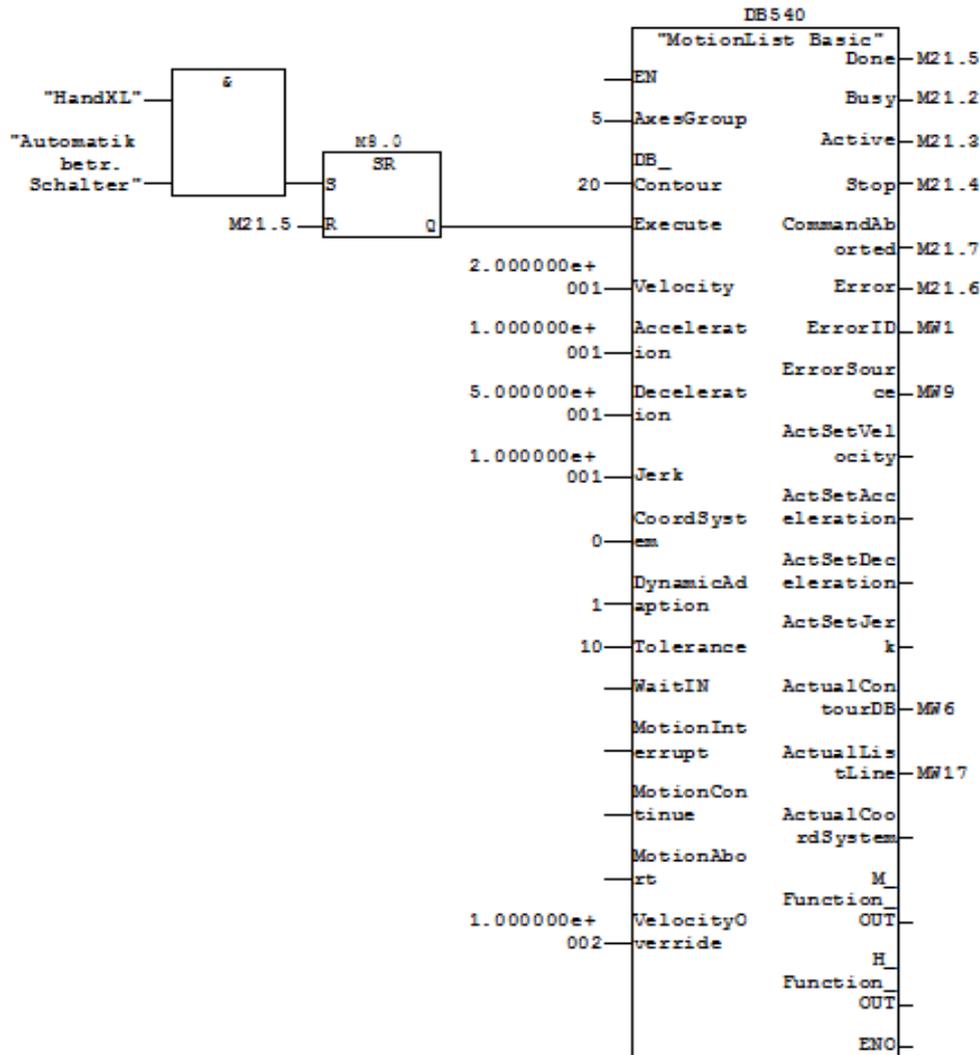
Netzwerk: 17 y-Achse setzen, (Referenzfahrt)



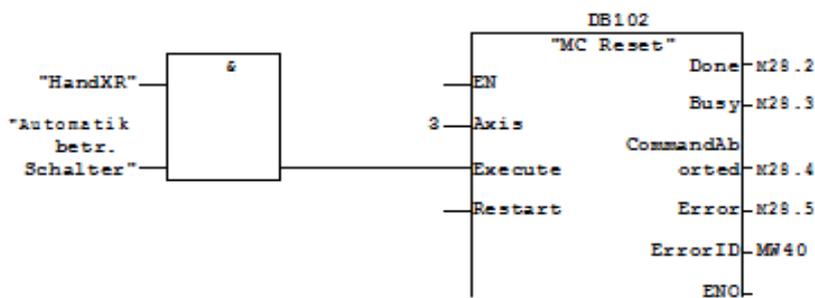
Netzwerk: 18 Y-Achse positionieren zum Startpunkt



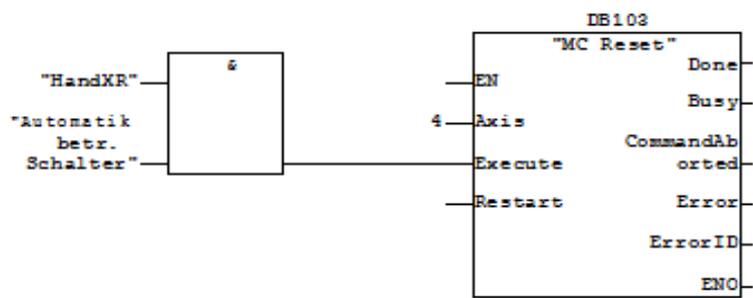
Netzwerk: 19 FB540 "MotionList-Basic"



Netzwerk: 20 X-Achse Reset im Automatikbetrieb



Netzwerk: 21 Y-Achse Reset im Automatikbetrieb



Netzwerk: 22 Bahnobjekt Reset im Automatikbetrieb

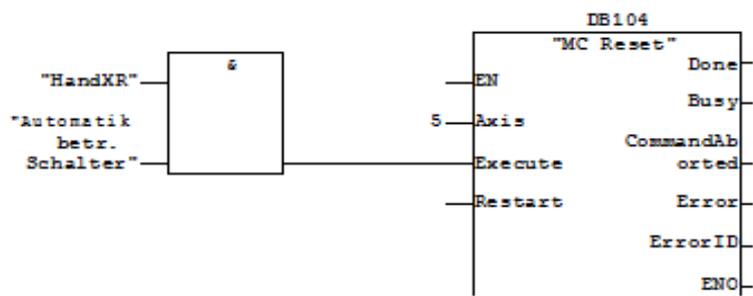


Abb. 3.7.5 Anwenderprogramm

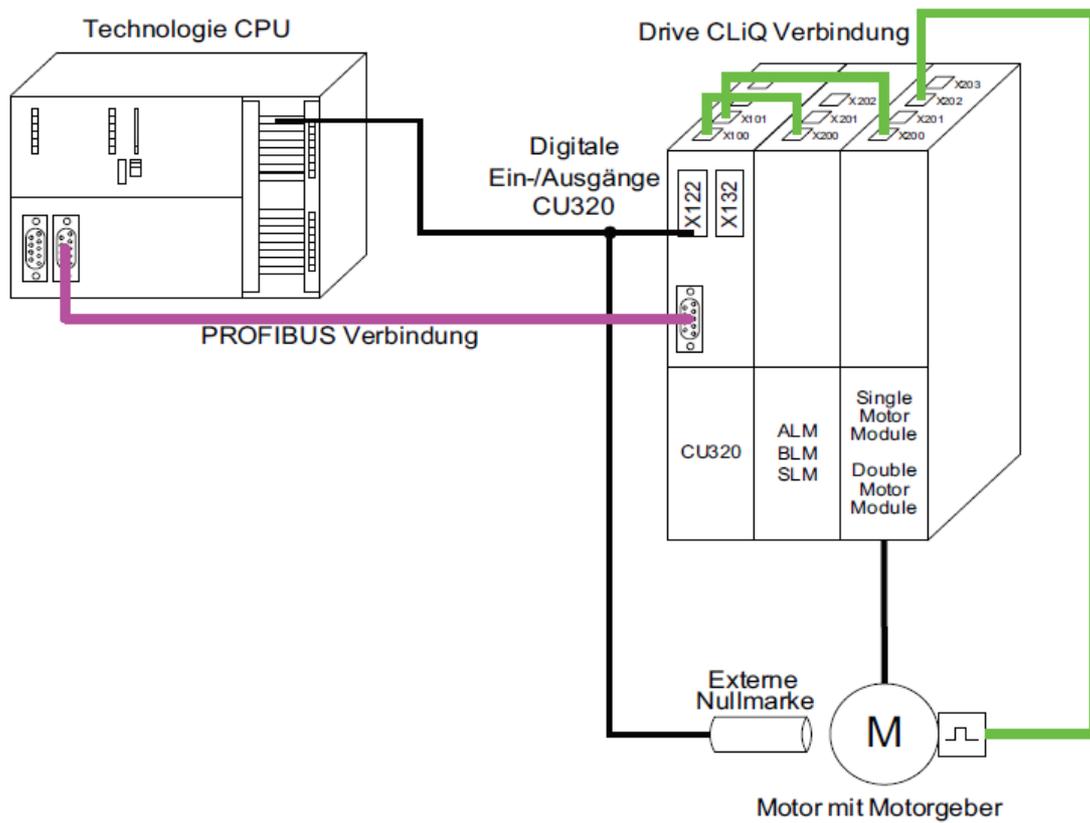


Abb. 4.1.4 Aktives Referenzieren, gespiegelt,
[5], Quelle: Siemens AG: SIMATIC, S7-300, Anbindung SINAMICS S120 an die
Technologie-CPU, Produktinformation, 09/2011, A5E00480377-04, S. 116.

Connector types

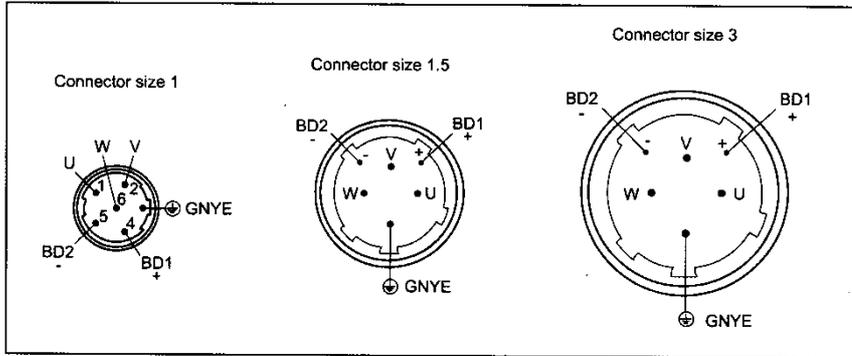


Figure 6-2 Power connector

Rundstecker (Zweifach)
Flachstecker

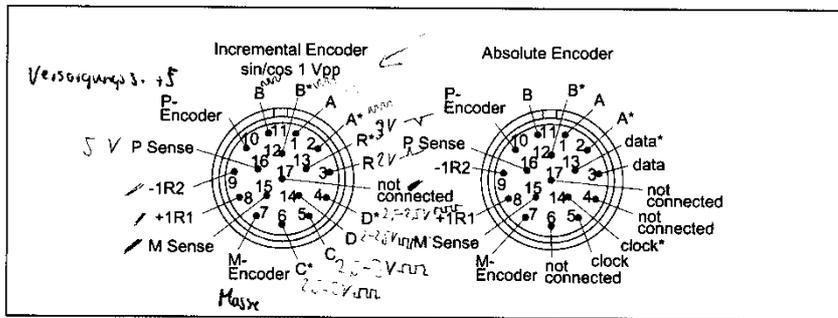


Figure 6-3 Signal connector (motor without DRIVE-CLiQ interface)

- 1 3
- 2 4
- 3 17
- 4 22
- 5 19
- 6 20
- 7 2
- 8 13
- 9 25
- 10 1
- 11 6
- 12 7
- 13 18
- 14 21
- 15 16
- 16 14
- 17 5; 8; 24
- 18
- 19
- 20

Power connection via terminal box

- The terminal assignment in the terminal box must be implemented according to the diagram.
- The PE conductor must be connected.
- Cable lugs must be used in accordance with DIN 46234.
- Connect optional brake (see figure).

Schirm | Schirm

alt am Flachstecker

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
5V													5V					



Abb. 4.1.6.1 Messversuchsaufbau

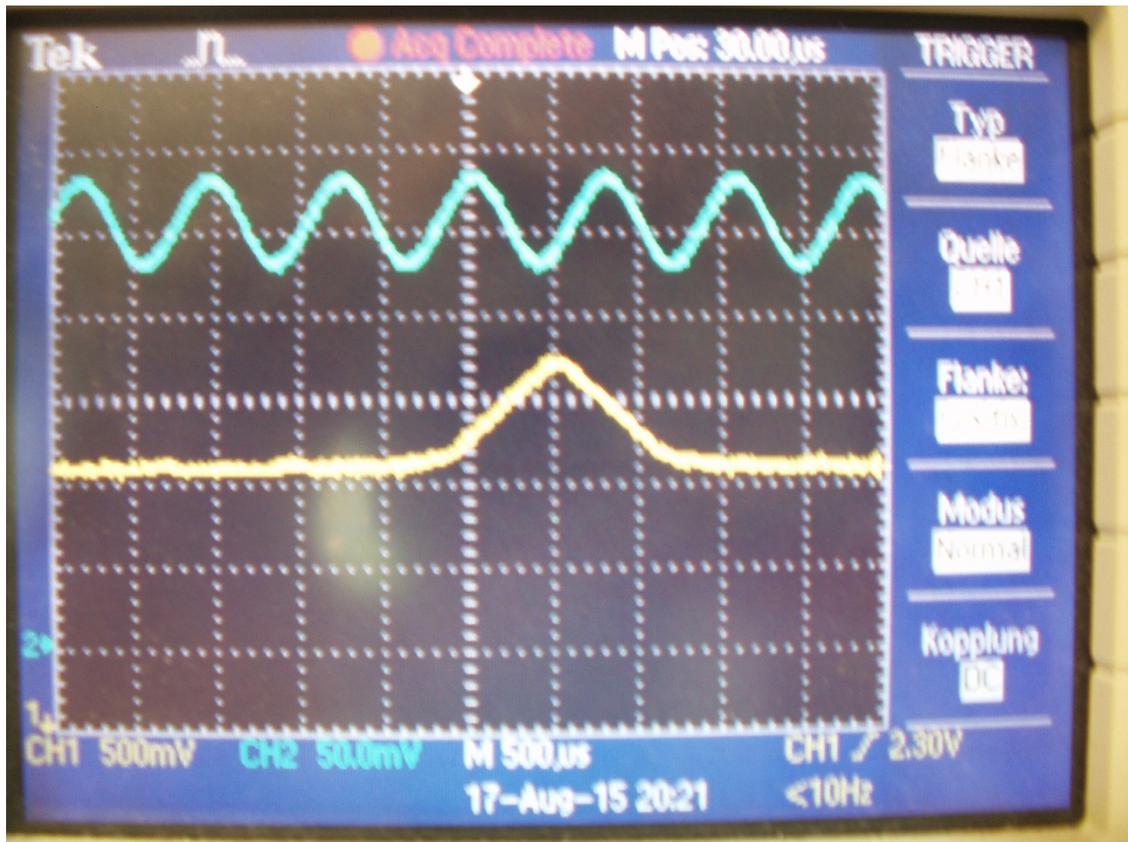


Abb. 4.1.6.2 Peak- Nullmarke

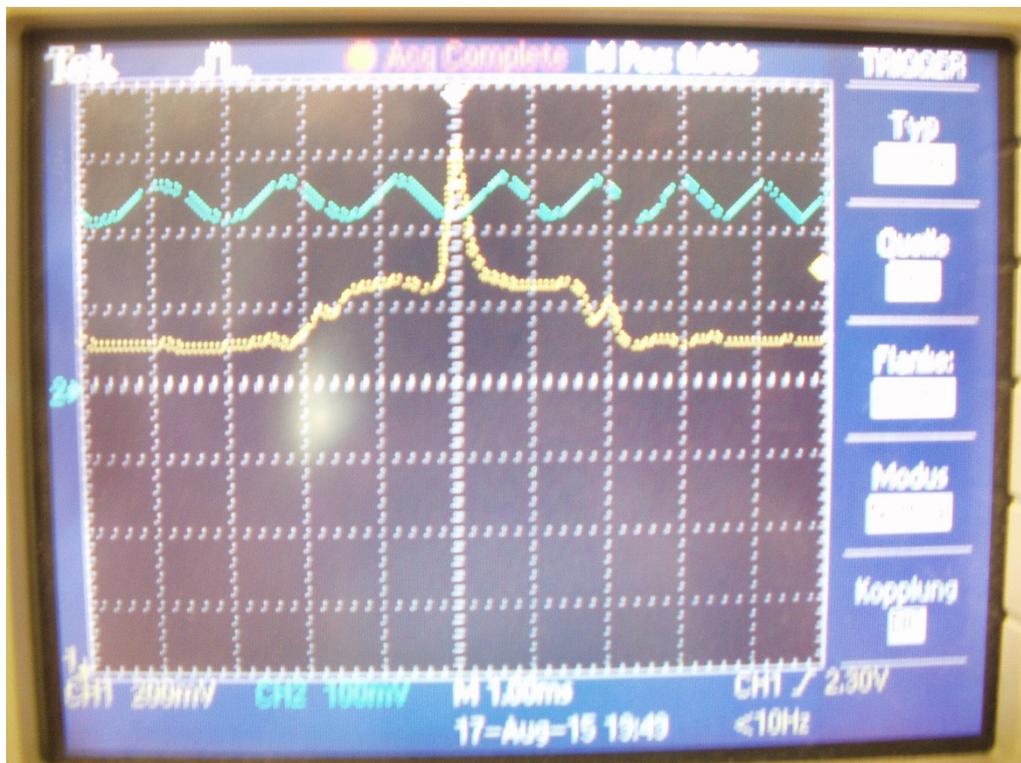


Abb. 4.1.6.3 Peak- Nullmarke verzerrt

Befehle, die in der „MotionList Basic“ genutzt werden können, in tabellarischer Form.

Tabelle 7-1 Im Technology Template vorhandene Befehle

ID	Befehl	Parameter							H-Fkt.	M-Fkt.
		EndPoint			AuxPoint					
		X	Y	Z	X	Y	Z			
0	NOP									
1	SetCartesianTransform	Trans X	Trans Y	Trans Z	Rot X	Rot Y	Rot Z			
2	MoveLinearAbsolute	Position X	Position Y	Position Z				H-Funktion	M-Funktion	
3	MoveLinearRelative	Distance X	Distance Y	Distance Z				H-Funktion	M-Funktion	
4	MoveCircAbs_AE	End Point X	End Point Y	End Point Z	Aux Point X	Aux Point Y	Aux Point Z	H-Funktion	M-Funktion	
5	MoveCircAbs_CE_S	End Point X	End Point Y	End Point Z	Aux Point X	Aux Point Y	Aux Point Z	H-Funktion	M-Funktion	
6	MoveCircAbs_CE_L	End Point X	End Point Y	End Point Z	Aux Point X	Aux Point Y	Aux Point Z	H-Funktion	M-Funktion	
8	MoveCircRel_AE	Distance X	Distance Y	Distance Z	Aux Point X	Aux Point Y	Aux Point Z	H-Funktion	M-Funktion	
9	MoveCircRel_CE_S	Distance X	Distance Y	Distance Z	Aux Point X	Aux Point Y	Aux Point Z	H-Funktion	M-Funktion	
10	MoveCircRel_CE_L	Distance X	Distance Y	Distance Z	Aux Point X	Aux Point Y	Aux Point Z	H-Funktion	M-Funktion	
12	MoveCirclesAbsolute	Arc			Center Point X	Center Point Y	Center Point Z	H-Funktion	M-Funktion	
13	MoveCirclesRelative	Arc			Distance C_X	Distance C_Y	Distance C_Z	H-Funktion	M-Funktion	
16	ExactStop_ON									
17	ExactStop_OFF									
18	WaitIN							H-Funktion	M-Funktion	
19	WaitTime	Wait Time [s]						H-Funktion	M-Funktion	
22	SetPlane_XY									
23	SetPlane_YZ									
24	SetPlane_ZX									

7.2 Motion-Befehle

ID	Befehl	Parameter							H-Fkt.	M-Fkt.
		EndPoint			AuxPoint					
		X	Y	Z	X	Y	Z			
25	SetTransitionMode							Transition Mode	Transition Param	
26	CallContourDB								DB Nummer	
27	SetVelocity							Velocity		
28	SetAcceleration							Acceleration		
29	SetDeceleration							Deceleration		
30	SetJerk							Jerk		
31	SetTolerance								Tolerance	
32	SetCoordSystem_BCS									
33	SetCoordSystem_OCS									
34	JumpToLine								Destination Line	
99	EndOfList									

Abb. 4.3.4 Motion- Befehlsliste, [10], Quelle: Siemens AG: Technology Template: „MotionList Basic“ Technologie CPU, Applikationsbeschreibung 03/2012, S.58- 59.