

Bernburg
Dessau
Köthen



Hochschule Anhalt
Anhalt University of Applied Sciences

emw

Fachbereich
Elektrotechnik, Maschinenbau
und Wirtschaftsingenieurwesen

Bachelorarbeit

zur Erlangung des akademischen Grades
Bachelor of Engineering (B. Eng.)

Yunzhi Xie

Vorname Nachname

Elektro- und Informationstechnik, 2013, 4060833

Studiengang, Matrikel, Matrikelnummer

Thema:

Aufbau des Plattentransportsystems mithilfe der speicherprogrammierbaren Steuerung (SPS)

Prof. Dr.-Ing. Anatoli Makarov

1. Prüfer/in

Dipl.-Ing. Ulf Heinisch

2. Prüfer/in

11.09.2017

Abgabe am

Selbstständigkeitserklärung

Hiermit erkläre ich, dass die Arbeit selbständig 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.

Köthen, 25.08.2017

Ort, Datum

Unterschrift des Studierenden

Sperrvermerk

Sperrvermerk: ja nein

wenn ja: Der Inhalt der Arbeit darf Dritten ohne Genehmigung der/des (Bezeichnung des Unternehmens) nicht zugänglich gemacht werden. Dieser Sperrvermerk gilt für die Dauer von X Jahren.

Köthen, 25.08.2017

Ort, Datum

Unterschrift des Studierenden

Abkürzungsverzeichnis

SPS	S peicher p rogrammierbare S teuerung
AC	A lternating C urrent (Wechselstrom)
DC	D irect C urrent (Gleichstrom)
CPU	C entral P rocessing U nit
FUP	F unktions p lan
KOP	K ontakt p lan
AWL	A nweisungs l iste
BCD	B inary C oded D ecimal
MW	M erker w ort
SR	S etzen und R ücksetzens
u to a	vom U rsprung in die A nfangsposition
u to z	vom U rsprung in die Z ielposition

Inhaltsverzeichnis

Selbstständigkeitserklärung.....	I
Sperrvermerk	I
Abkürzungsverzeichnis	II
1 Motivation und Zielsetzung.....	1
2 Hardware und Module	2
2.1 Struktur.....	2
2.2 Motor und dazugehörige Ausrüstung.....	6
2.2.1 DC-Motor-Geber-Modul.....	6
2.2.2 DC-Motor-Sensor-Modul	13
2.2.3 Umkehrsteuer des Motors	15
2.3 Simatic-Serie.....	17
2.3.1 Stromversorgungsbaugruppe PS307-5A.....	18
2.3.2 Zentralbaugruppe CPU315-2-DP und CPU314C-2-DP	19
2.3.3 Digitalbaugruppe SM323.....	20
2.3.4 Positionierbaugruppe FM357-2	21
3 Software und Programmierung.....	22
3.1 STEP7.....	22
3.2 Programmierung ohne Positionierbaugruppe	24
3.2.1 Programmierung bei der Zeitsteuerung.....	26
3.2.2 Programmierung bei der Signalsteuerung	34
3.3 Programmierung mit der Positionierbaugruppe.....	39
3.4 Vergleich von drei Programmen.....	42
4 Zusammenfassung und Ausblick	44
Anhang.....	a
Programm bei der Zeitsteuerung mit der Programmiersprache FUP	a
Abbildungsverzeichnis	i
Tabellenverzeichnis	ii
Literatur- und Quellenverzeichnis	iii
Internetverzeichnis	iii
Abbildungsquellenverzeichnis.....	iv

1 Motivation und Zielsetzung

Mit der Entwicklung von Wissenschaft und Technik wird die Automatisierung immer häufiger verwendet. Bei dieser Arbeit werden die Platten durch die Steuerung der Motoren von der Anfangsposition an einer Position der Stellage transportiert. Der ganze Vorgang wird mithilfe der SPS gesteuert. In diesem Fall werden die Siemens-Module ausgewählt.

Die Hardware von dem Plattentransportsystem in dieser Arbeit ist vorgegeben. Wenn aber die Datei der Hardware unbekannt ist, muss man zuerst die vorhandene Hardware kennen. Die Hardware wird laut der Anforderung ausgewählt. Die Programme sind wegen der unterschiedlichen Hardware etwas anders. Ein Programmierkonzept wird erstellt, dann wird ein Programm geschrieben.



Abb. 1.1: Modul des Plattentransportsystems

In diesem Praktikum werden diese Designs verglichen, um eine bevorzugte Ausführung auszuwählen.

2 Hardware und Module

2.1 Struktur

Das Programm basiert auf der Messung des Rahmens. Die Distanz zwischen der Anfangsposition und der Zielposition ermittelt den Motorlauf. Der erste Schritt ist es, die Stellage zu messen.

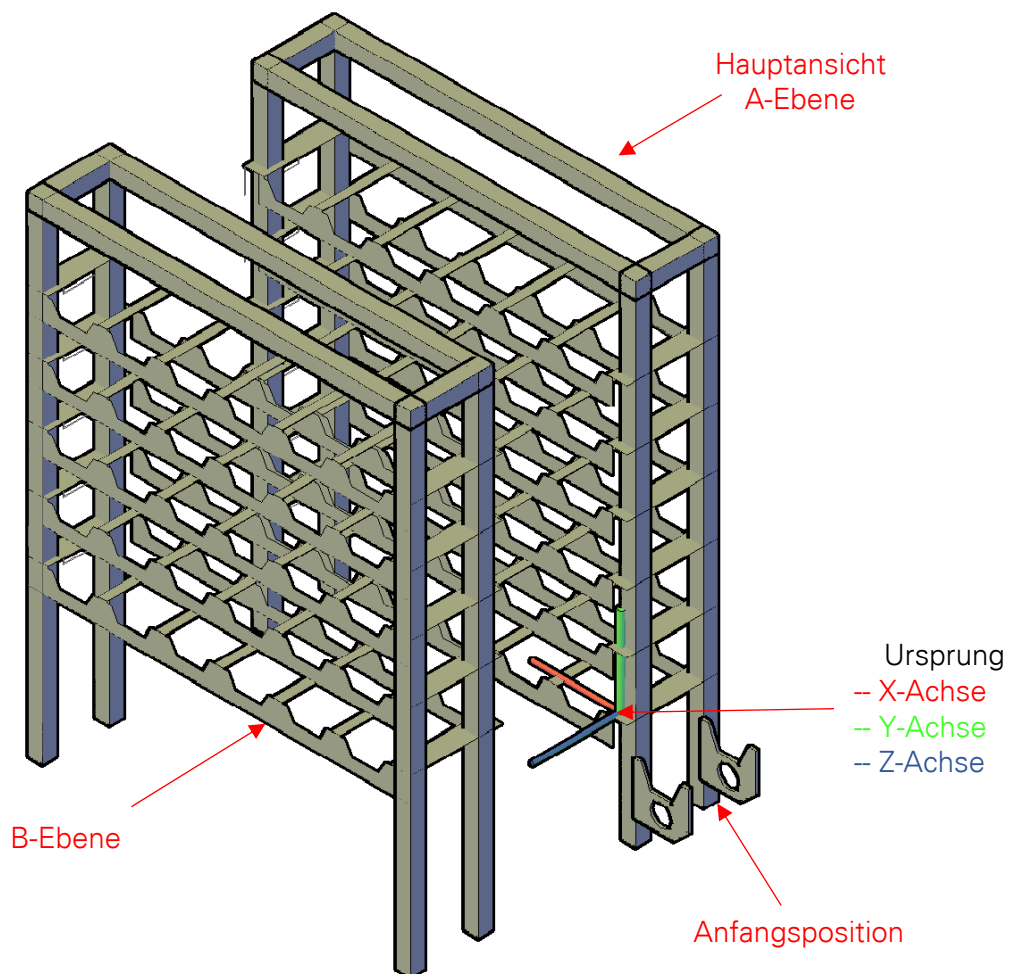


Abb. 2.1: 3D-Modul des Rahmens und Erstellung von Ursprung und Koordinaten

Wie die Abbildung 2.1 zeigt, wird die Anfangsposition als Ursprung festgelegt. Die Abstände zwischen den angrenzenden Zielpositionen sind regelmäßig, aber nicht gleich wie der Abstand zwischen der Anfangsposition und Zielposition. Damit die Berechnung einfach wird, ist der Ursprung nicht die Anfangsposition. Wenn der Abstand zwischen den angrenzenden

Zielpositionen mit 130 mm eingestellt ist, beträgt der Abstand zwischen dem Ursprung und der ersten Zielposition 93 mm.

Dann kann ein kartesisches Koordinatensystem erstellt werden. Die Länge des Rahmens ist die X-Achse, die Höhe des Rahmens die Y-Achse und der Abstand zwischen zwei Rahmen ist die Z-Achse.

Einheit der Messung [mm]

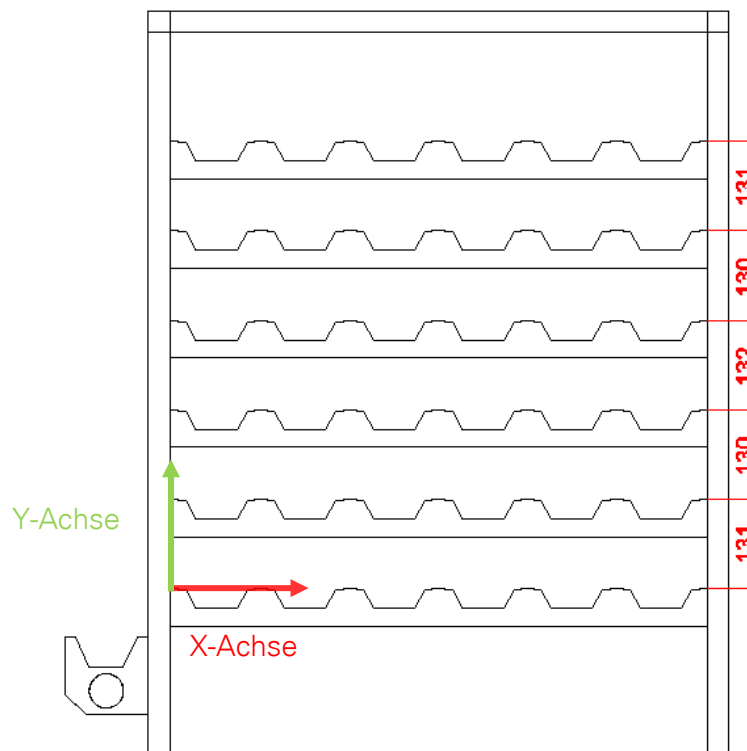


Abb. 2.2: Hauptansicht (Nur A-Ebene)

Anhand der Abbildung 2.2 kann man erkennen, dass es einige kleine Fehler aufgrund von Problemen bei der Installation der Rahmen gibt. Bei den Fehlern handelt es sich um 2 mm. Dadurch, dass der Maximalwert bei der Programmierung ausgewählt wird, können die Fehler kompensiert und vernachlässigt werden.

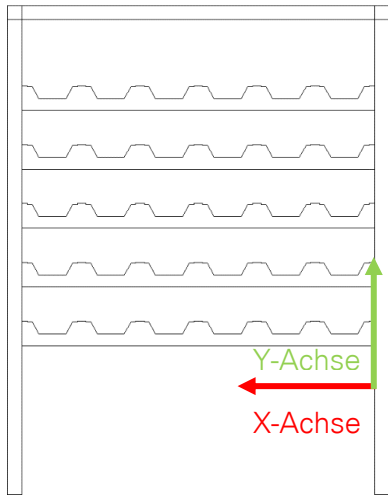


Abb. 2.3: Rückansicht (Nur B-Ebene)

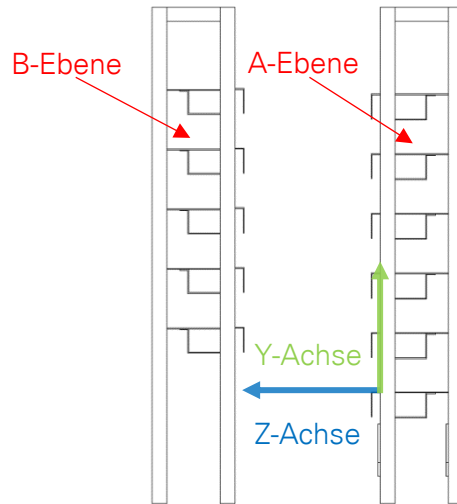


Abb. 2.4: Linksansicht

Die Hauptansicht, Rückansicht (Abbildung 2.3) und Linksansicht (Abbildung 2.4) werden verglichen. Die zwei Rahmen sind nicht symmetrisch, also muss man bei der Programmierung aufpassen.

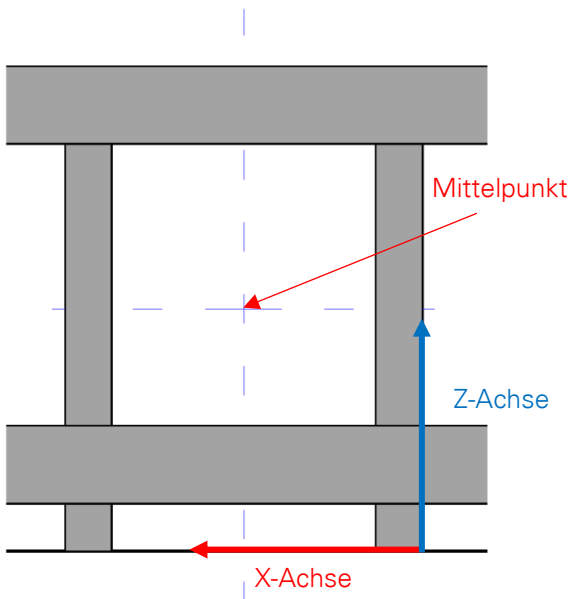


Abb. 2.5: Erstellung des Mittelpunkts auf dem Rahmen

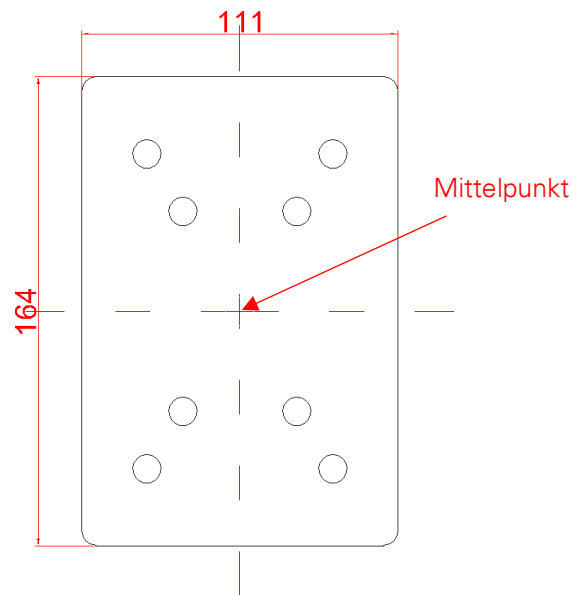


Abb. 2.6: Erstellung des Mittelpunkts auf der Platte

Der Mittelpunkt jeder Position wird als Zielpunkt erstellt. Die Erstellung des Mittelpunkts erfolgt wie in Abbildung 2.5 und Abbildung 2.6 gezeigt.

Der auf der Z-Achse gesteuerte Motor ist unabhängig von dem Bewegungsabstand, man muss nur die Daten des Bewegungsabstands auf der X-Achse und der Y-Achse messen. Auf der Basis der Koordinatenerstellung wird die Stellage gemessen, und graphisch dargestellt.

Einheit der Messung [mm]

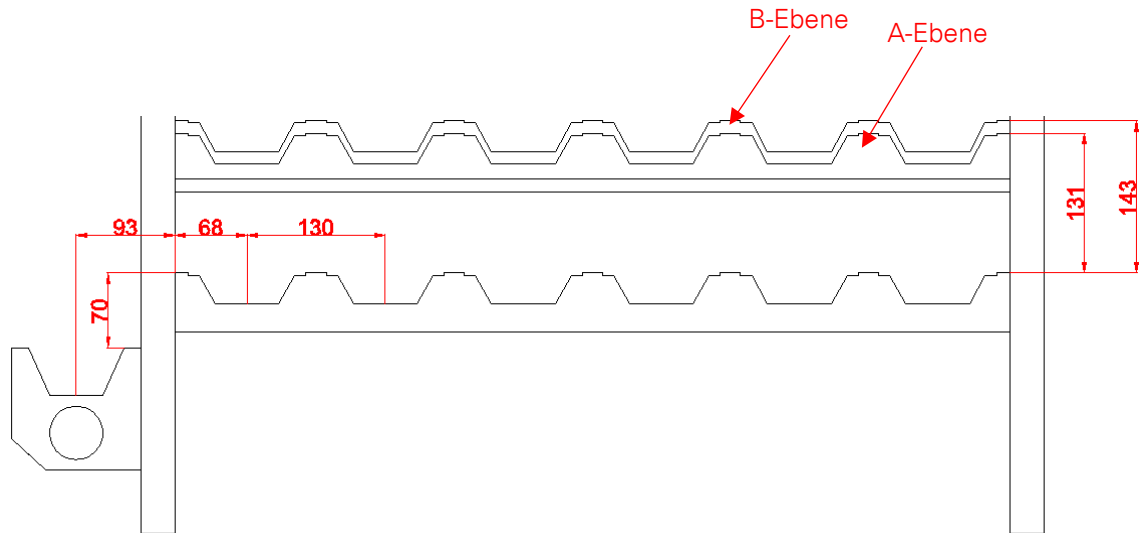


Abb. 2.7: Hauptdaten des Rahmens

Die Hauptdaten werden aus der Abbildung 2.7 gelesen und in Tabelle 1 dargestellt. Mithilfe der Daten der Tabelle können die Motoren für die X-Achse und Y-Achse gesteuert werden.

Bewegungsabstand	X-Achse [mm]	Y-Achse [mm]
Zwischen dem Ursprung und der Anfangsposition	93	70
Zwischen dem Ursprung und der ersten Zielposition (A-Ebene)	68	0
Zwischen dem Ursprung und der ersten Zielposition (B-Ebene)	68	143
Zwischen den angrenzenden Zielpositionen (X-Achse)	130	0
Zwischen den angrenzenden Zielpositionen (Y-Achse)	0	131
Bewegung für die Abholung und Setzung der Platten	0	20

Tabelle 1: Hauptdaten der Bewegungsabstände

2.2 Motor und dazugehörige Ausrüstung

Der Motor und die zugehörigen Ausrüstungen sind der Kern des Übertragungssystems. In diesem System gibt es insgesamt drei DC Motor-Module, das sind zwei DC-Motor-Geber-Module und ein DC-Motor-Sensor-Modul.

2.2.1 DC-Motor-Geber-Modul

Dieses Modul besteht aus einem DC Motor, einem Getriebe und einem Gebersystem. Mit Hilfe des Gebersystems können die Daten der Bewegung in die Richtung von X-Achse und Y-Achse überwacht werden. Dann kann man durch diese Daten den Motor steuern und die Platten in die Richtung von X-Achse und Y-Achse transportieren.

Abbildung 2.8 zeigt das DC-Motor-Modul. Hier werden die Produkte dieses Moduls von Dunkermotoren verwendet.

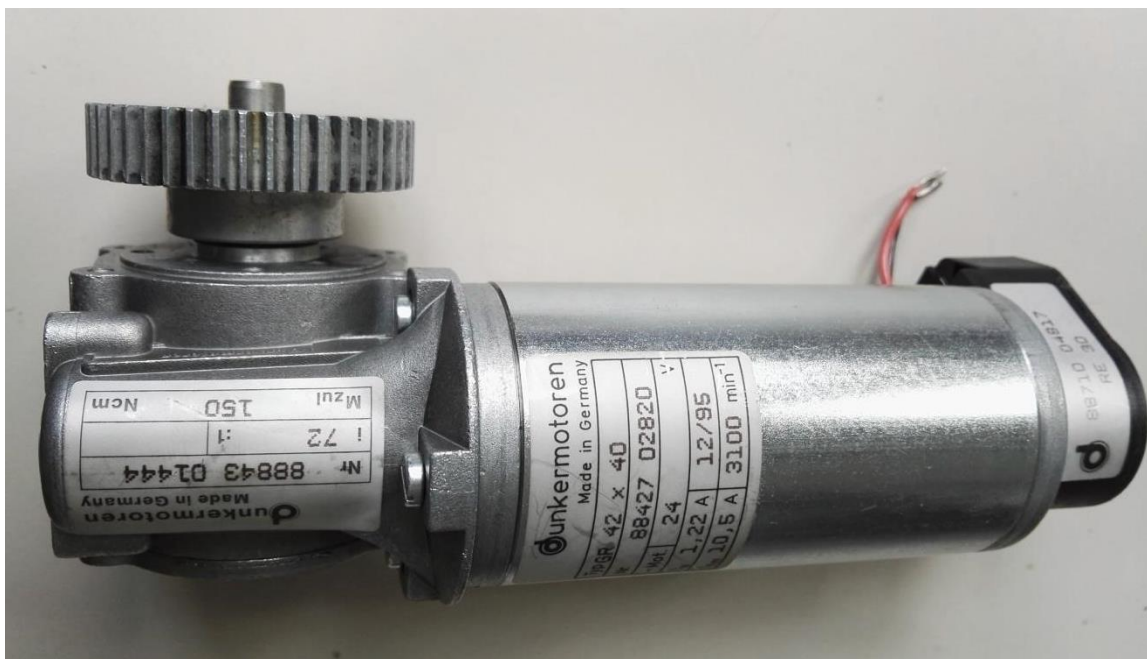


Abb. 2.8: DC-Motor-Geber-Modul

2.2.1.1 Produkt

Der DC-Motor GR-42×40-24V (Dunkermotoren)



Abb. 2.9: DC-Motor GR-42×40-24V (Dunkermotoren)

Quelle:

http://www.dunkermotoren.de/typo3temp/_processed_/c/5/csm_8842702010_ddef411635.jpg (16.07.2017)

Abbildung 2.9 zeigt den DC-Motor GR-42×40-24V aus der Gruppe der Dunkermotoren. Der DC-Motor ist ein bürstenbehafteter Gleichstrommotor, deren Baureihe GR/G ist. Er hat einen Durchmesser von 42 mm und eine Nennspannung von 24 V. Die maximale Abgabeleistung von dem Motor ist 32,3 W in der Stufe 40 W. Dieser Motor kann rechts sowie links laufen. Das heißt, der Manipulator kann in beiden Richtungen laufen, zum Beispiel in der negativen X-Achse und positiven X-Achse.

Parameter	Wert	Einheit
Nennspannung	24	V
Nennstrom	1,2	A
Nenndrehmoment	5,7	Ncm
Nenndrehzahl	3100	rpm
Leerlaufdrehzahl	3800	rpm
Maximale Abgabeleistung	32,3	W
Nennleistung	18,5	W
Anlaufstrom	5,68	A
Leerlaufstrom	0,17	A
Gewicht	0,49	kg

Tabelle 2: Technische Hauptdaten des DC-Motors GR-42×40-24V

Mit diesem Motor kann die elektrische Energie in kinetische Energie umgewandelt werden. Die technischen Hauptdaten des Motors für die Montage und das Programm sind in Tabelle 2 dargestellt.

Das Getriebe SG62-72:1 (Dunkermotoren)



Abb. 2.10: Getriebe SG62-72:1 (Dunkermotoren)

Quelle:

http://www.dunkermotoren.de/typo3temp/_processed_/c/csm_8884304005_fa133bd499.jpg (16.07.2017)

Die Drehzahl des DC-Motors GR-42×40-24V ist für dieses Plattentransportsystem nicht geeignet, weil die Motordrehzahl von 3100 rpm zu hoch ist. Mit dem Getriebe kann die Motordrehzahl anteilmäßig abfallen. Das Getriebe SG62-72:1 ist in Abbildung 2.10 dargestellt. Hier wird das Verhältnis von 72:1 verwendet. Ein Zahnrad wird mit der Getriebeausgangswelle verbunden, und greift in eine Zahnstange, damit eine Drehbewegung in eine geradlinige Bewegung umgesetzt werden kann.

Parameter	Wert	Einheit
Untersetzung	72:1	
Nenn Drehmoment	150	Ncm
Maximaler Wirkungsgrad	30	%
Gewicht	0,3	kg

Tabelle 3: Technische Hauptdaten des Getriebes SG62-72:1

In der Tabelle 3 kann man den maximalen Wirkungsgrad ablesen. Obwohl der Wirkungsgrad reduziert wird, wird die Frequenz der Signalerfassung ebenfalls reduziert. Dann können die Signale erkannt werden.

Der Geber RE30-3-500 (Dunkermotoren)



Abb. 2.11: Gebersystem RE30-3-500 (Dunkermotoren)

Quelle:

http://www.dunkermotoren.de/typo3temp/_processed_/6/1/csm_8871004720_0517cd801f.jpg (16.07.2017)

Abbildung 2.11 zeigt den Geber RE30-3-500. Er ist ein digitaler optischer Inkrementalgeber. Er liefert für die Übertragung der Positionsänderung zwei um 90° verschobene Signale, das heißt, es gibt zwei Signalkanäle. Für diesen Geber ist eine Versorgungsspannung von 5 V nötig. Aber diese Versorgungsspannung des Produkts unterscheidet sich von anderen Produkten, man muss eine separate Stromversorgung benutzen.

Der Geber ist ein Kernprodukt für die Signalsteuerung dieses Plattentransportsystems. Die Bewegungsdaten des Motors werden mithilfe des Gebers codiert und in ein Impulssignal umgewandelt, damit der Motorlauf mit den Bewegungsdaten gesteuert werden kann. Anhand der Tabelle 4 kann man erkennen, dass es 500 Impulse pro Umdrehung gibt. Man kann den Bewegungsabstand des Manipulators durch die Anzahl der Impulse überwachen.

Parameter	Wert	Einheit
Anzahl Kanäle	3	
Auflösung	500	
Geberversorgungsspannung	5	V
Ausgangsspannung high level	2.4	V
Abfallzeit	49	ns
Ausgangsspannung (high-level)	min. 2,4	V
Ausgangsspannung (low-level)	max.0,4	V

Tabelle 4: Technische Hauptdaten des Gebers RE30-3-500 für Montage und Programm

2.2.1.2 Anwendung und Parameterumrechnung

Die Parameter von dem Motor und dem Getriebe kann man in der Einführung lesen. Diese Parameter können nicht direkt in dem Programm der SPS verwendet werden. Man muss sie zuerst umrechnen.

Im ersten Schritt wird die Nenndrehzahl in die lineare Bewegungsgeschwindigkeit umgerechnet. Sie beeinflusst die Bewegung der Platte in die Richtung der X-Achse und Y-Achse.

Leerlaufdrehzahl von DC Motor GR42 $n_{GR42} = 3800rpm$

Nenndrehzahl von DC Motor GR42 $n_{GR42} = 3100rpm$

Untersetzung =72:1

$$n_{\text{leerlauf nach der Untersetzung}} = \frac{3800rpm}{72} = 52,778rpm$$

$$n_{\text{nennen nach der Untersetzung}} = \frac{3100rpm}{72} = 43,056rpm$$

(2.1)

Die aktuelle Leerlaufdrehzahl wird gemessen.

Die Zeit wird nach jeden zehn Umdrehungen des Zahnrads gemessen. Der Mittelwert bzw. der Durchschnitt werden nach zehn Messungen berechnet.

	1	2	3	4	5	6	7	8	9	10	\bar{t}
$t_{por\ 10\ U. [s]}$	11,44	11,49	11,49	11,46	11,61	11,48	11,61	11,50	11,45	11,51	11,50

Tabelle 5: Zeitmessung des Zahnrads pro zehn Umdrehungen

$$t_{por\ Umdrehn.} = 1,15\ s$$

$$n_{\text{aktuell leerlauf}} = \frac{60}{1,15s} = 52,17rpm$$

$$n_{\text{leerlauf nach der Untersetzung}} > n_{\text{aktuell leerlauf}}$$

Die aktuelle Leerlaufdrehzahl ist wenig kleiner als der Parameter, weil es nach der Verbindung des Zahnrads Störungen gibt. Beim normalen Betrieb sind die Störungen noch vorhanden. Die aktuelle Nenndrehzahl ist auch kleiner als die ideale Nenndrehzahl, also wird der Wert von 42 rpm gewählt.

Die Kreisbewegung wird durch das Zahnrad in eine lineare Bewegung umgewandelt.
Der Durchmesser des Zahnrads beträgt 45,80mm.

$$U = \pi \cdot d = 3,14 \cdot 45,80 = 143,89 \text{ mm}$$

Dieser Umfang des Zahnrads stellt die Bewegungsabstände pro Umdrehung dar.

Für das zeitabhängige Steuerungsprogramm wird die lineare Geschwindigkeit berechnet,

$$v_{linear} = U \cdot \frac{60}{n} = 143,89 \text{ mm} \cdot \frac{60}{42 \text{ rpm}} = 205,55 \text{ mm} \cdot \text{s}^{-1}$$

(2.2)

Bewegungsabstand	X-Achse [mm]	t [s]	Y-Achse [mm]	t [s]
Zwischen dem Ursprung und der Anfangsposition	93	0,452	70	0,341
Zwischen dem Ursprung und der ersten Zielposition (A-Ebene)	68	0,331	0	0,000
Zwischen dem Ursprung und der ersten Zielposition (B-Ebene)	68	0,331	143	0,696
Zwischen der angrenzenden Zielposition (X-Achse)	130	0,632	0	0,000
Zwischen der angrenzenden Zielposition (Y-Achse)	0	0	131	0,637
Bewegung für die Abholung und Setzung der Platten	0	0	20	0,097

Tabelle 6: Hauptdaten der Bewegungsabstände und Bewegungszeit

Beim zeitgesteuerten Programm wird die Datei in der Tabelle 6 verwendet.

Die andere Methode für die Programmierung wird durch die Signalerfassung des Gebers RE30-3-500 gesteuert.

Die Auflösung des Gebers beträgt 500 Impulse pro Umdrehung. Die Bewegungsabstände pro Impuls werden wie folgt berechnet.

$$s_{imp} = \frac{U}{500 \text{ Impulse}} = \frac{143,89 \text{ mm}}{500 \text{ Impulse}} = 0,288 \text{ mm} \cdot \text{Impulse}^{-1}$$

(2.3)

Durch den Erhalt der Anzahl des Impulssignals kann man die Bewegungsabstände steuern. Mithilfe der Tabelle 1 kann die Anzahl des Impulssignals (wie in Tabelle 7 gezeigt) berechnet werden.

Bewegungsabstand	X-Achse [mm]	Anzahl der Impulse von A	Y-Achse [mm]	Anzahl der Impulse von A
Zwischen dem Ursprung und der Anfangsposition	93	323	70	243
Zwischen dem Ursprung und der ersten Zielposition (A-Ebene)	68	236	0	0
Zwischen dem Ursprung und der ersten Zielposition (B-Ebene)	68	236	143	496
Zwischen den angrenzenden Zielpositionen (X-Achse)	130	451	0	0
Zwischen den angrenzenden Zielpositionen (Y-Achse)	0	0	131	454
Bewegung für die Abholung und Setzung der Platten	0	0	20	69

Tabelle 7: Hauptdaten der Bewegungsabstände und Anzahl der Impulse

Betrachtet man zum Beispiel das Abholen der Platten. Während der Bewegung des Manipulators vom Ursprung zur Anfangsposition erhalten die Kanäle A des Gebers 323 Impulssignale von Motor 1 und 243 Impulssignale von Motor 2.

Es gibt zwei um 90° verschobene Rechtecksignale aus RE 30. Nach einer Viertelperiode von Signal A wird das Signal B erhalten. Damit wird die Bewegung genauer überprüft.

2.2.2 DC-Motor-Sensor-Modul

Dieses Modul besteht aus einem DC-Motor und drei magnetischen Sensoren. Die Verwendung des Moduls ist einfacher als die Verwendung des DC-Motor-Geber-Moduls. Der Motorlauf wird mithilfe des Näherungsschalters gesteuert.

Der DC-Motor G30.2-24V (Dunkermotoren)



Abb. 2.12: DC-Motor G30.2-24V (Dunkermotoren)

Quelle:

http://www.dunkermotoren.de/typo3temp/_processed_/3/7/csm_8861506403_30cbd94e87.jpg (16.07.2017)

Der DC-Motor G30.2-24V (wie in Abbildung 2.12 gezeigt) ist ein bürstenbehafteter Gleichstrommotor mit Permanentmagneten. Er hat einen Durchmesser von 30 mm und eine Nennspannung von 24 V. Dieser Motor kann auch rechts sowie links laufen.

Parameter	Wert	Einheit
Nennspannung	24	V
Nennstrom	0,3	A
Nenndrehmoment	1	Ncm
Nenndrehzahl	3000	rpm
Leerlaufdrehzahl	5000	rpm
Maximale Abgabeleistung	4,1	W
Nennleistung	3,1	W
Anlaufstrom	0,77	A
Leerlaufstrom	0,08	A
Gewicht	0,08	kg

Tabelle 8: Technische Hauptdaten von DC-Motor G30.2-24V für Montage und Programm

Der Näherungsschalter SME-8-S-KED-24 (FESTO)



Abb. 2.13: Näherungsschalter SME-8-S-KED-24 (FESTO)

Quelle:

<https://www.landefeld.de/shop/media/festo/produkte/16bp0509.jpg> (23.07.2017)

Abbildung 2.13 zeigt den Näherungsschalter SME-8-S-KED-24 (FESTO). Er ist ein magnetischer Sensor. Der Näherungsschalter wird mit T-Nut zusammen benutzt. Mit Reed-Kontakt kann ein Signal erhalten werden, das heißt, es bewegt sich der Manipulator schon in dieser eingestellten Position.

Parameter	Wert	Einheit
Bauform		für T-Nut
Messprinzip		magnetisch Reed
Schaltausgang		kontaktbehaftet bipolar
Schaltelementfunktion		Schließer
Einschaltzeit	0,5	ms
Ausschaltzeit	0,03	ms

Tabelle 9: Technische Hauptdaten von Näherungsschalter SME-8-S-KED-2 (FESTO)

2.2.3 Umkehrsteuer des Motors

Da der Manipulator sich in beiden Richtungen bewegen muss, ist es wichtig für die Drehrichtungsumkehr des Motors. Der Vorgang kann einfach durch eine Umpolung von Plus- und Minuspol mithilfe des Relais realisiert werden. Durch dieses Verfahren wird der Motor gesteuert. Hier werden zwei Lutze-Relaisbausteine RPE-1E2/2W-DC-24V verwendet.

Pins von Relaisbausteine	Pins von Anlage
21 von 1k1	Plus-Pol des Motors
21 von 1k2	Minus-Pol des Motors
24	24 V Stromversorgung
22	0 V Stromversorgung

Tabelle 10: Pin-Belegung zwischen Relais und Anlage

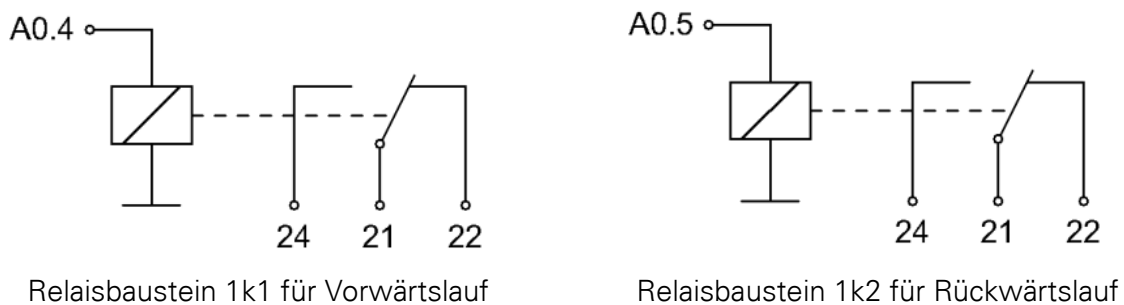


Abb. 2.14: Anschlussbild der Relaisbausteine für Vorwärtslauf und Rückwärtslauf (Motor 3)

Abbildung 2.14 zeigt das Anschlussbild der Relaisbausteine. Der Plus- und Minuspol von Motor 3 wird jeweils auf den Pin 21 der Relaisbausteine geschaltet. Pin 24 und Pin 22 sind mit der 24V-Stromversorgung verbunden. Vorwärts- und Rückwärtslauf können nicht gleichzeitig durchgeführt werden, deshalb werden beide Relaisbausteine miteinander verzahnt. Wenn der Relaisbaustein 1k1 eingeschaltet wird, schließt sich der Schalter im Relaisbaustein. Das Elektrodenpotential des Pluspols von Motor 3 ist höher als das Elektrodenpotential des Minuspols. Dann fließt der Strom von dem Pluspol in den Minuspol und der Motor 3 läuft vorwärts. Bei der Einschaltung des Relaisbausteins 1k2 ist das Elektrodenpotential des Minuspols von Motor 3 höher als das Elektrodenpotential des Pluspols und der Motor 3 läuft rückwärts.



Abb. 2.15: 4Q-PWM-Servoregler MTRG-25/2E (mattkeag)

Quelle:

https://www.mattke.de/fileadmin/_processed_/csm_MTRG_7405ddab58.gif (16.07.2017)

Abbildung 2.15 zeigt das Produkt 4Q-PWM-Servoregler MTRG-25/2E (mattkeag). Die Drehzahl und Umkehrrichtung von Motor 1 und Motor 2 kann mithilfe des 4Q-PWM-Servoreglers gesteuert werden, indem das Signal des Simatic-Moduls an den Eingang des Servoreglers gegeben und der Ausgang mit den Motoren angeschlossen wird. Der Servoregler wird mit der Simatic-Funktionsbaugruppe verbunden, und seine Daten werden bei der Programmierung eingestellt.

2.3 Simatic-Serie

Die Simatic-Serie sind Produkte von Siemens, sie werden hauptsächlich im Bereich der Industrieautomatisierung verwendet. Es gibt mehrere Baugruppen in der Simatic-Serie, z. B. Stromversorgungsbaugruppen, Zentralbaugruppen, Signalbaugruppen, Funktionsbaugruppen und Kommunikationsbaugruppen. Es ist flexibel, dass die verschiedenen Baugruppen je nach der Anforderung angewendet werden können.

Bei diesem Plattentransportsystem werden die geeigneten Baugruppen zuerst aus den bestehenden Baugruppen ausgewählt, dann werden die Baugruppen nach der Art des Programms bestimmt. Wie Tabelle 11 zeigt, werden verschiedene Simatic-Module ausgewählt, weil es verschiedene Steuerungsarten gibt.

Steuerungsart	Zeitsteuerung	Signalsteuerung	
Stromversorgungsbaugruppe	PS 307 6ES7307-1EA01-0AA0	PS 307 6ES7307-1EA01-0AA0	PS 307 6ES7307-1EA01-0AA0
Zentralbaugruppe	CPU 315-2 DP 6ES7315-2AH14-0AB0	CPU 314C-2 DP 6ES7314-6CH04-0AB0	CPU 314C-2 DP 6ES7314-6CH04-0AB0
Signalbaugruppe	SM 323 6ES7323-1BH01-0AA0	-	-
Funktionsbaugruppe	-	-	FM 357-2 6ES7357-4AH01-0AE0

Tabelle 11: Auswahl von Simatic-Modulen anhand der Steuerungsarten

In den folgenden Abschnitten werden die Baugruppen kurz erläutert.

2.3.1 Stromversorgungsbaugruppe PS307-5A



Abb. 2.16: Stromversorgungsbaugruppe PS307-5A

Quelle:

https://support.industry.siemens.com/dl/files/imagepool/products/969/1969/v1/P_KT01_XX_01099i.jpg (24.07.2017)

Die Stromversorgungsbaugruppe PS307-5A (wie in Abbildung 2.16 gezeigt) sind Eingabebe-
gräte, die als Laststromversorgung verwendet werden. Die Wechselspannung von 230 V an
1-phasigem Wechselstromnetz wird in Gleichspannung von 24 V umgewandelt, damit die
Stromversorgungsbaugruppen andere Baugruppen mit der Spannung von 24 V versorgen
können.

Parameter	Wert	Einheit
Eingang	1-phasig AC	
Eingangsnennspannung U_e	120/230	V
Eingangsnennstrom I_e	2,3/1,2	A
Ausgang	DC	
Ausgangsnennspannung U_a (DC)	24	V
Stromnennwert I_a Nenn	5	A
Kurzzeitiger Überlaststrom	20	A

Tabelle 12: Technische Hauptdaten der Stromversorgungsbaugruppe PS307-5A

2.3.2 Zentralbaugruppe CPU315-2-DP und CPU314C-2-DP



Abb. 2.17: Zentralbaugruppe CPU315-2-DP



Abb. 2.18: Zentralbaugruppe CPU314C-2-DP

Quelle:

<https://mall.industry.siemens.com> (24.07.2017)

Die Zentralbaugruppe ist die Kernbaugruppe der Simatic-Serie. Durch diese Baugruppe kann man die Anlagen mit den erfassenden Signalen steuern. Die Zentralbaugruppe CPU315-2-DP und CPU314C-2-DP sind die Produkte der Zentralbaugruppe der CPU31x-Serie und CPU31xC-Serie.

Einen Unterschied kann man im Aussehen erkennen (wie in Abbildung 2.17 und Abbildung 2.18 gezeigt), nämlich, dass die Zentralbaugruppe der CPU31x-Serie keine integrierten digitalen und analogen Eingänge bzw. Ausgänge besitzt. Deshalb muss diese Zentralbaugruppe mithilfe der Signalbaugruppen verwendet werden.

Es gibt noch einen wichtigen Unterschied. Die Zentralbaugruppen der CPU31x-Serie verfügen über schnellere Zähler. Mit der Drehzahl des Motors nach der Untersetzung und Auflösung des Gebers kann man die Frequenz der Signalerfassung berechnen. (Siehe Gleichung 2.1)

$$f_{\text{Signalerfassung}} = \frac{43 \cdot 500}{60} \text{ Hz} = 358 \text{ Hz}$$

(2.4)

Die normalen Zähler können das Signal von hoher Frequenz nicht erkennen. Deswegen müssen die schnellen Zähler oder Frequenzzähler verwendet werden. Also wird die Zentralbaugruppe der CPU31xC-Serie bei der Signalsteuerung ausgewählt.

Die Funktion der Zentralbaugruppe der CPU31x-Serie kann die Anforderungen bei der Zeitsteuerung erfüllen. Hier wird die bestandene Zentralbaugruppe CPU315-2-DP verwendet.

Die Kennung „DP“ bedeutet, dass die Zentralbaugruppe die PROFIBUS-DP-Schnittstelle besitzt. Die PROFIBUS-DP-Schnittstelle kann als Master oder Slave parametrierbar werden und eine Übertragung von bis zu 12 Mbit/s haben.

2.3.3 Digitalbaugruppe SM323



Abb. 2.19: Zentralbaugruppen CPU314C-2-DP

Quelle:

https://mall.industry.siemens.com/collaterals/files/44/jpg/P_ST70_XX_00216i.jpg (25.07.2017)

Die Digitalbaugruppe SM323 (wie in Abbildung 2.19 dargestellt) ist eine Signalbaugruppe, die acht digitale Eingänge und acht digitale Ausgänge hat. Die digitalen Signale werden als Spannung durch die digitalen Eingänge gelesen. Die Spannung von high-level wird als 1 gelesen. Im Gegensatz dazu wird die Spannung von low-level als 0 gelesen. Die Baugruppen können digitale Signale senden, um die Anlage zu steuern.

2.3.4 Positionierbaugruppe FM357-2



Abb. 2.20: Positionierbaugruppe FM357-2

Quelle:

https://support.industry.siemens.com/cs/images/products/P_NC01_XX_00351i.jpg

(25.07.2017)

Abbildung 2.20 zeigt die mikroprozessorgesteuerte Positionierbaugruppe FM357-2. Diese Baugruppe kann maximal vier Achsen ansteuern, also wird die Funktionsbaugruppe als Mehrachsbaugruppe bezeichnet. Der Servo- und Schrittantrieb kann mit der Positionierbaugruppe für Einzelachsen oder im Achsverbund positioniert werden.

Die Programmierung mit dieser Positionierbaugruppe soll mit dem besonderen Baustein durchgeführt werden. Die technischen Daten, z. B. Drehzahl des Motors, Auflösung des Gebers und Bewegungsabstand, werden in dem Programm eingestellt. Die Motoren werden anhand des Programms mit diesen Daten gesteuert.

Wenn die Stromversorgung angeschlossen ist, kann der Geber mit der Versorgungsspannung von 5 V durch diese Positionierbaugruppe versorgt werden.

3 Software und Programmierung

Weil unterschiedliche Simatic-Module verwendet werden, kann man mit den unterschiedlichen Arten für das Transportsystem programmieren. Aber der grundlegende Prozess ist im Wesentlichen gleich.

3.1 STEP7

STEP7 ist eine Engineering-Software für die Programmierung zur Steuerung der Baugruppen der Simatic-Serie.

Die Baugruppen können nach der Konfiguration gesteuert werden. Abbildung 3.1 zeigt die Konfiguration bei der Zeitsteuerung. Die Baugruppe FM357-2 im roten Bereich bzw. Steckplatz 7 wird für die Signalsteuerung verwendet.

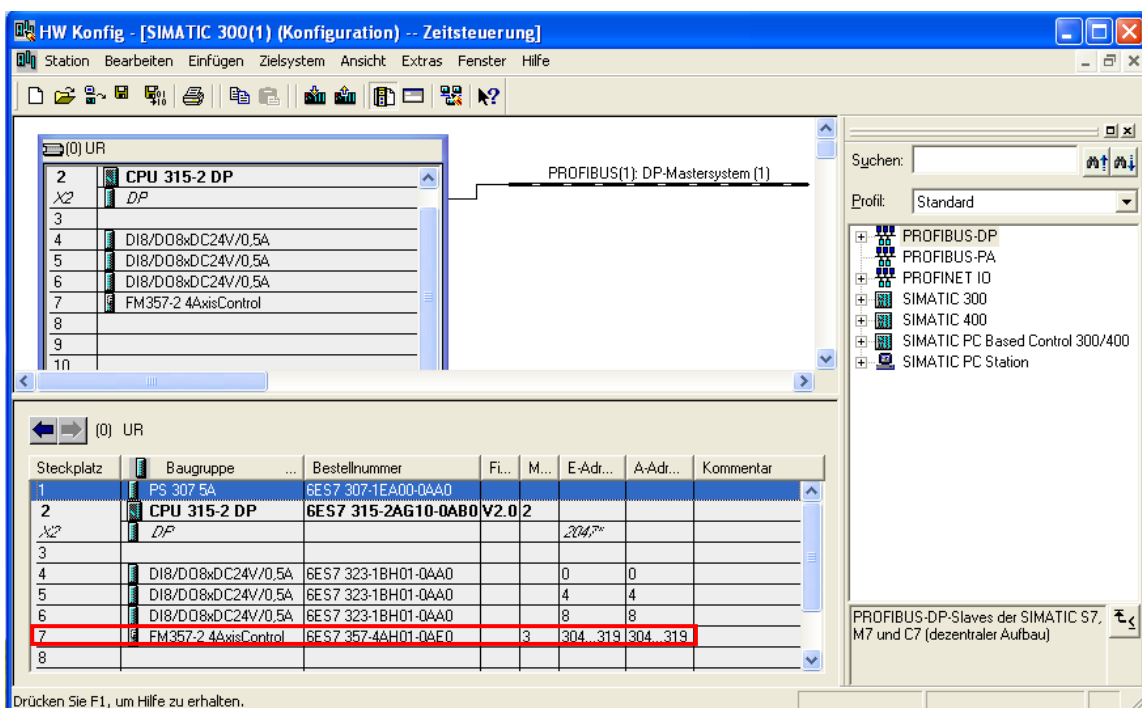


Abb. 3.1: HW Konfiguration des Simatic-Managers

In Siemens STEP 7 gibt es folgende Programmiersprachen:

- FUP -> Funktionsplan: Grafische Programmiersprache
- KOP -> Kontaktplan: Grafische Programmiersprache
- AWL -> Anweisungsliste: Textuelle Programmiersprache

Die Programmierung mit FUP stellt den logischen Aufbau eines SPS-Programms übersichtlich dar. Bei der Programmierung mit KOP kann man die Zustandsänderung von Kontakten in dem SPS-Programm erkennen. Die Programmdarstellung ist ähnlich wie ein elektrisches Schaltbild. Die Programmiersprache AWL ist eine maschinennahe, textbasierende Programmiersprache, damit können die grafischen Programmiersprachen FUP und KOP in die Programmiersprache AWL umgewandelt werden, aber das Reverse ist unbedingt notwendig.

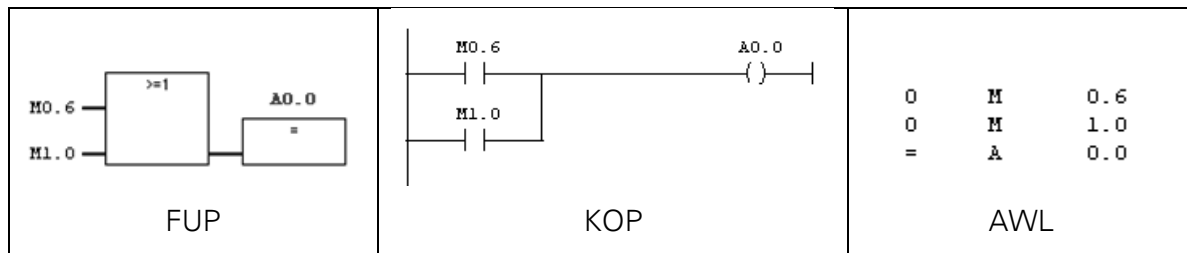


Abb. 3.2: Beispiel der Programmiersprachen FUP, KOP und AWL

Ein Beispiel ist in Abbildung 3.2 dargestellt. FUP zeigt die Operationsfunktion, und KOP zeigt eine Parallelschaltung. In der AWL werden die Anweisungen in der Reihenfolge geschrieben. Die Programme von drei Programmiersprachen haben die gleiche Bedeutung. Wenn ein Signal von M0.6 oder M1.0 kommt, ist A0.0 gleich 1.

Beim Schreiben des Programms ist die Programmiersprache C einfacher als die Programmiersprache von STEP 7. Aber die Programmiersprache STEP 7 hat gegenüber der Programmiersprache C den Vorteil, dass man während des laufenden Programms die Zustände der Eingänge und Ausgänge eines jeden Elements beobachten kann.

Um den Zustand einfach beobachten und die Anweisung von logischen Operationen schreiben zu können, wird die grafische Programmiersprache FUP in diesem Programm verwendet.

3.2 Programmierung ohne Positionierbaugruppe

Ohne die Positionierbaugruppe kann man nicht durch die Einstellung in dem Programm einfach programmieren. Man muss mithilfe der Zähler und Timer programmieren.

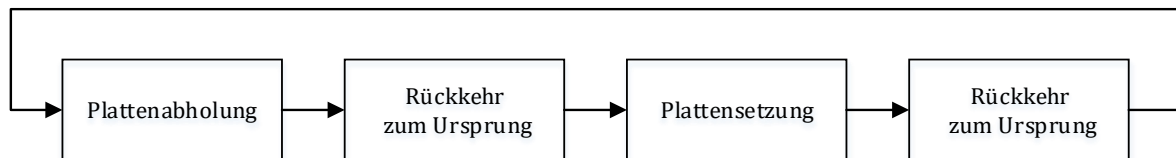


Abb. 3.3: Flussdiagramm

Der Transportprozess wird in vier Hauptschritte unterteilt.

1. Plattenabholung: Der Manipulator bewegt sich vom Ursprung in die Anfangsposition und holt die Platte ab.
2. Rückkehr zum Ursprung: Der Manipulator bewegt sich von der Anfangsposition zum Ursprung.
3. Plattensetzung: Der Manipulator bewegt sich vom Ursprung in die Zielposition und setzt die Platte.
4. Rückkehr zum Ursprung: Der Manipulator bewegt sich von der Zielposition zum Ursprung.

Bisher ist eine Schleife fertig, danach wird der erste Schnitt wiederholt, und führt eine neue Schleife durch.

Der Abstand zwischen dem Ursprung und der Anfangsposition ist nicht gleich wie der Abstand zwischen dem Ursprung und der Zielposition bzw. zwischen den zwei Zielpositionen. Um einfach berechnen zu können, wird der Bewegungsvorgang von der Anfangsposition in die Zielposition unterteilt. Zuerst bewegt sich der Manipulator von der Anfangsposition zum Ursprung, dann bewegt er sich vom Ursprung in die Zielposition.

Wenn sich der Manipulator in die positive X-Achsen-Richtung bewegt, wird diese Drehrichtung von Motor 1 als Vorwärtslauf-Richtung eingestellt. Im Gegensatz dazu wird die Drehrichtung von Motor 1 als Rückwärtslauf-Richtung bei der Bewegung des Manipulators in die

negative X-Achsen-Richtung eingestellt. Die Einstellung der beiden anderen Motoren erfolgt auf die gleiche Weise.

Die drei Motoren dürfen bei diesem Programm nicht gleichzeitig laufen. Also gibt es in den einzelnen Hauptschritten einige kleine Schritte, z. B. Ablaufreihenfolge der Motoren, Feinbewegung von Motor 2 bei der Plattenabholung und Plattensetzung. Es gibt insgesamt vierzehn Schritte. Abbildung 3.4 zeigt ein Zustandsflussdiagramm.

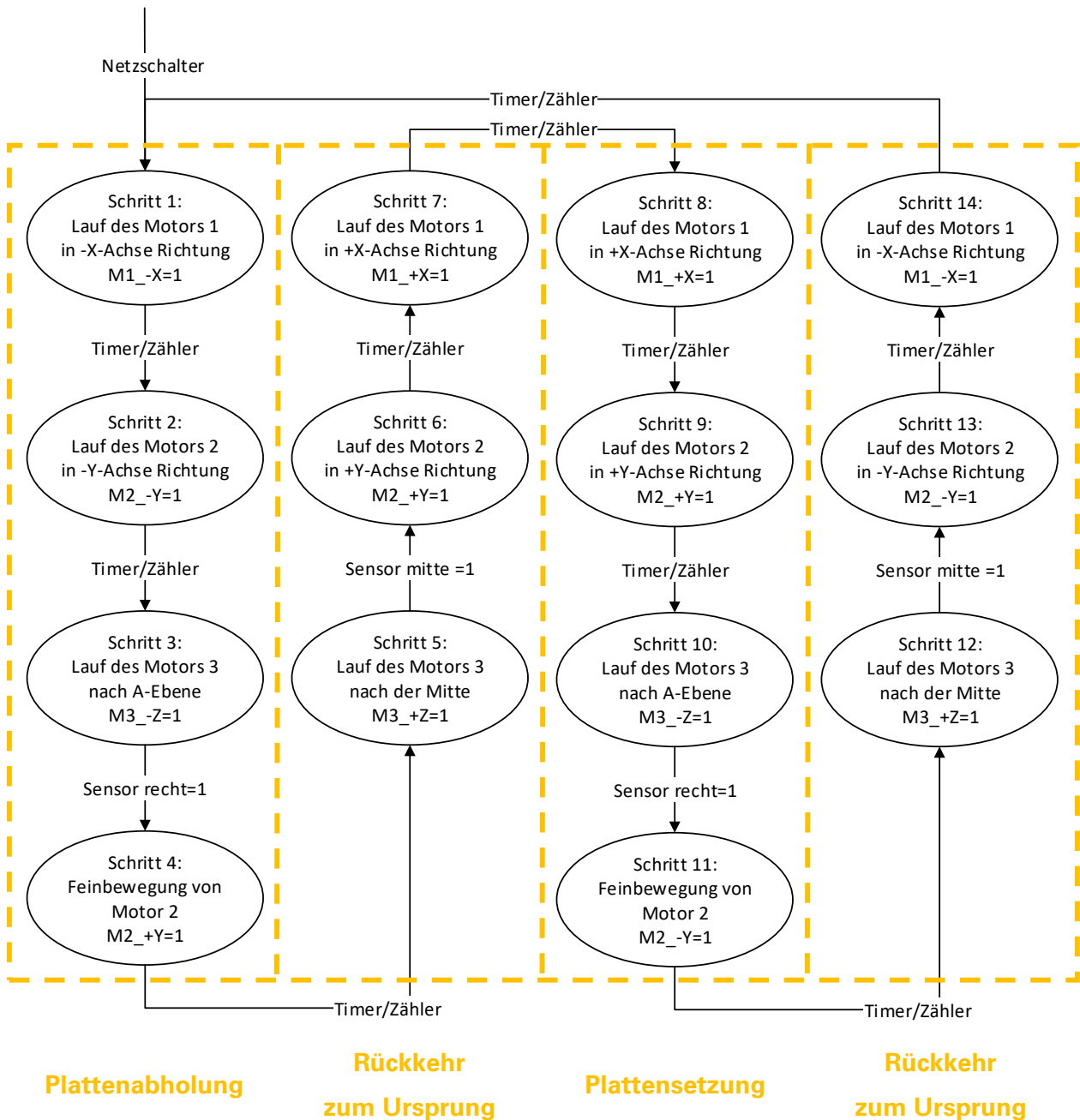


Abb. 3.4: Zustandsflussdiagramm

Die Plattenabholung wird nachfolgend erläutert:

Als Erstes läuft der Motor 1 beim Drücken des Netzschalters in die negative X-Achsen-Richtung. Nachdem sich der Manipulator in einem bestimmten Abstand bewegt hat, läuft der Motor 2 in die negative Y-Achsen-Richtung. Wenn der Manipulator an die Anfangsposition gelangt ist, stoppt der Motor 2 und Motor 3 läuft in die negative Z-Achsen-Richtung bis zum Eintritt des Signals des rechten Sensors. Dann gibt es eine Feinbewegung von Motor 2 zur Plattenabholung.

Der Motordrehzustand wird durch die logische Verknüpfung bestimmt, wie in Tabelle 13 dargestellt.

Motordrehzustand	Logische Verknüpfung
Vorwärtslauf von Motor 1 (+x)	S7+S8
Rückwärtslauf von Motor 1 (-x)	S1+S14
Vorwärtslauf von Motor 2 (+y)	S6+S9+S4
Rückwärtslauf von Motor 2 (-y)	S2+S13+S11
Vorwärtslauf von Motor 3 (+z)	S3+S10
Rückwärtslauf von Motor 3 (-z)	S5+S12

Tabelle 13: Bedingung des Motordrehzustands

Für beide Steuerungsarten ist der Ablauf fast gleich. Bei der Zeitsteuerung wird hauptsächlich der Timer verwendet, und der Zähler spielt bei der Signalsteuerung die wichtigste Rolle.

3.2.1 Programmierung bei der Zeitsteuerung

Die Programmierung bei der Zeitsteuerung ist hauptsächlich abhängig vom Timer, um die Motoren zu steuern. Hier wird der speichernde Einschaltverzögerungs-Timer ausgewählt.

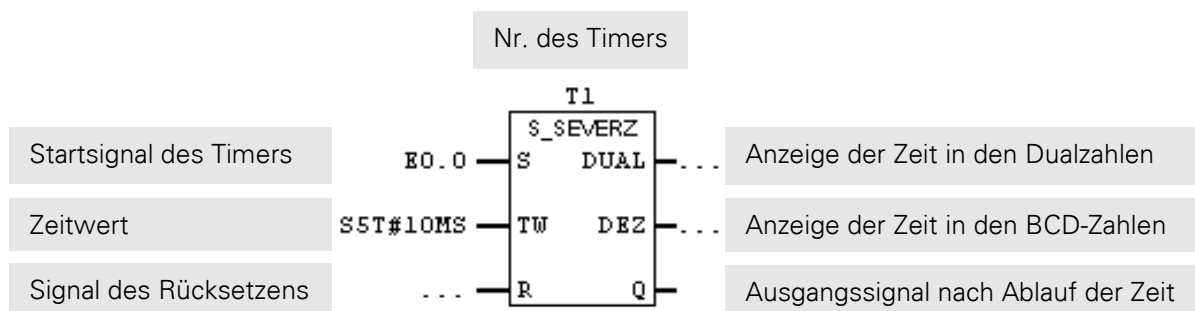


Abb. 3.5: Schema des Einschaltverzögerungs-Timers

Abbildung 3.5 zeigt ein Schema des Einschaltverzögerungs-Timers. Wenn der Eingang eine steigende Flanke erfasst, gibt der Ausgang ein Dauersignal nach dem Ablauf der eingestellten Zeit aus. Bei einer Signalerfassung im Rücksetzen wird das Dauersignal ausgeschaltet, und der Timer wird zurückgesetzt. Die Genauigkeit für den Zeitwert ist nicht hoch. Die kleinste Zeitbasis ist 10 ms, deshalb wird der Zeitwert vernachlässigt, wenn er kleiner als 10 ms ist.

Der Zeitwert in dem Hauptprogramm wird anhand der Tabelle 6 berechnet und eingestellt.

Symbol	Adresse	Datentype	Kommentar
M1_+x	A 0.0	BOOL	Motor1 läuft in +x-Richtung
M1_-x	A 0.1	BOOL	Motor1 läuft in -x-Richtung
M2_+y	A 0.2	BOOL	Motor2 läuft in +y-Richtung
M2_-y	A 0.3	BOOL	Motor2 läuft in -y-Richtung
M3_+z	A 0.4	BOOL	Motor3 läuft in +z-Richtung
M3_-z	A 0.5	BOOL	Motor3 läuft in -z-Richtung
Netzschalter	E 8.7	BOOL	Netzschalter
Sensor l	E 8.2	BOOL	Sensor der linken Position von Motor 3 (z-Achse)
Sensor m	E 8.1	BOOL	Sensor der mittleren Position von Motor 3 (z-Achse)
Sensor r	E 8.0	BOOL	Sensor der rechten Position von Motor 3 (-z-Achse)

Tabelle 14: Symboltabelle bei der Zeitsteuerung von STEP 7

Tabelle 14 ist eine Symboltabelle bzw. Zuordnungstabelle bei der Zeitsteuerung. Dann wird das Programm nach dem Zustandsflussdiagramm in der Reihenfolge geschrieben. Für jeden Schritt ist das SR-Glied bei der Aussetzung des Netzschalters E8.7 rückgesetzt.

Für den Lauf jedes Schrittes wird das SR-Glied verwendet. Ein Beispiel ist in Abbildung 3.5 aufgeführt:

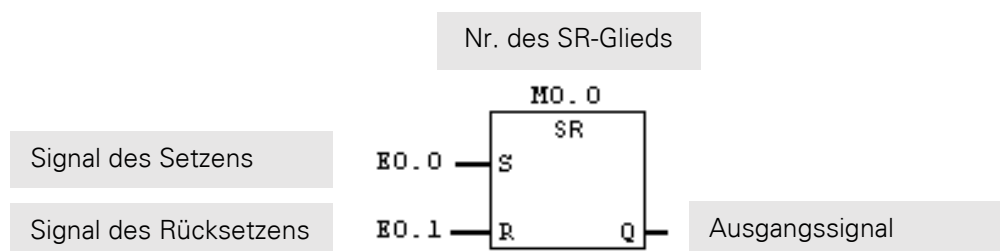


Abb. 3.6: Schema des SR-Glieds

Während des Eintritts eines Signals E0.0 in den Eingang wird das SR-Glied M0.0 gesetzt. Das heißt, es läuft der entsprechende Schritt. Das SR-Glied M0.0 wird ausgeschaltet, wenn ein Signal an dem Rücksetzeingang erfolgt.

Im Folgenden werden die wichtigen Schritte erklärt.

Der Manipulator bewegt sich vom Ursprung in die Anfangsposition in die negative X-Achsen-Richtung, das ist der erste Schritt. Die Anweisung dieses Vorgangs wird in Abbildung 3.7 gezeigt.

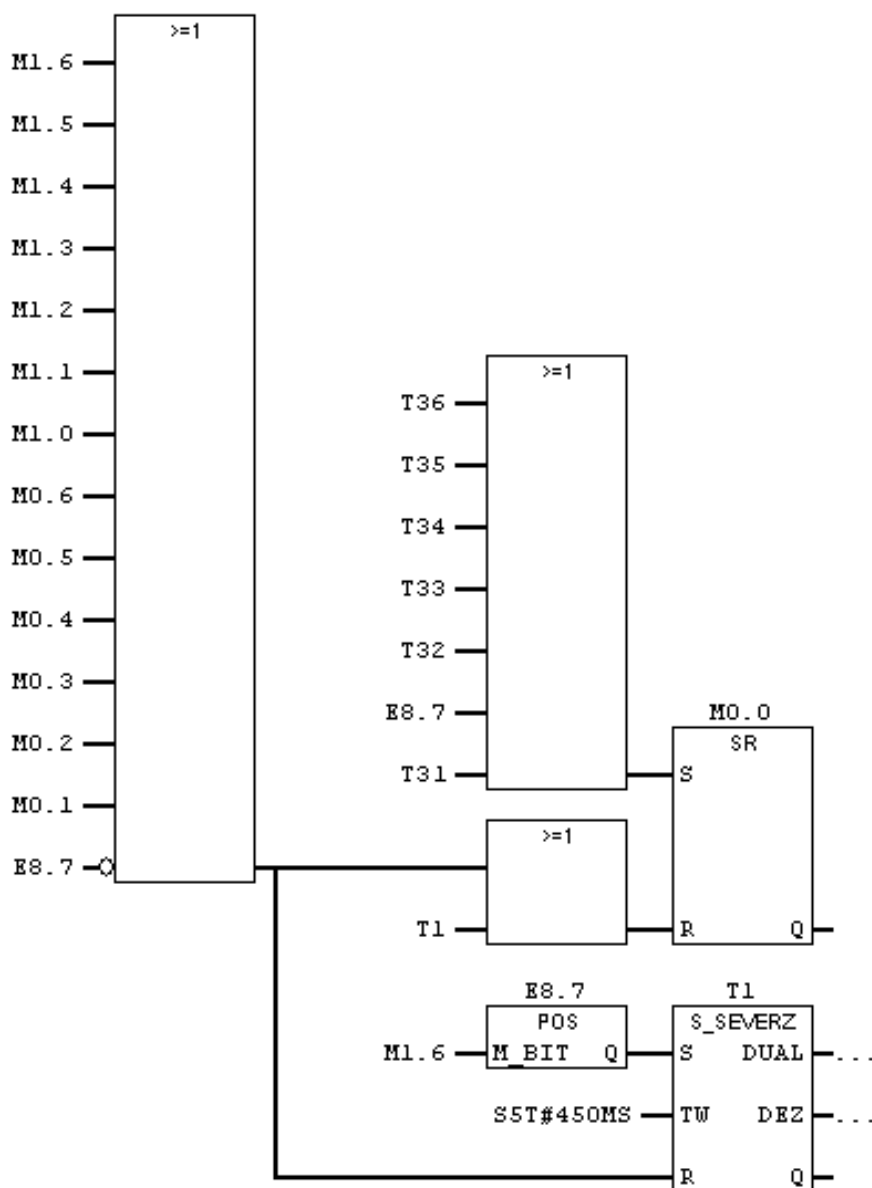


Abb. 3.7: Anweisung von Hauptprogramm: 1. u to a (-x)

Das SR-Glied M0.0 wird bei der Einschaltung des Netzschalters E8.7 und des Timers im Schritt 14 rückgesetzt. Wenn der Timer T1 eingeschaltet wird, wird das SR-Glied M0.0 gleichzeitig rückgesetzt.

Die Startbedingung des Timers ist der Signalwechsel von 0 auf 1. Für den Timer T1 wird die Startbedingung als der Zeitpunkt der Einschaltung des Netzschalters und Ausschaltung des SR-Glieds M1.6 erstellt. Bei der Ausschaltung des SR-Glieds M1.6 ist der Zeitpunkt des Endes eines Zyklus. Hier wird die Startbedingung mit der Operation POS durchgeführt.

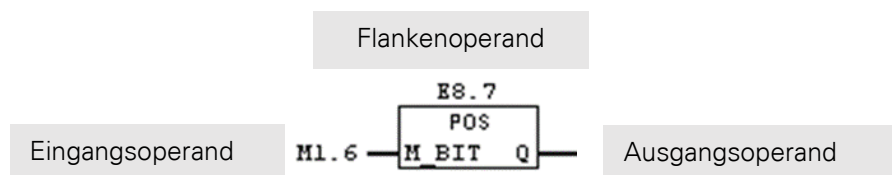


Abb. 3.8: Schema der POS-Operation

Wie in Abbildung 3.8 zu sehen ist, benötigt die POS-Operation zwei Operanden, bzw. Eingangsoperand und Flankenoperand. Dann wird am Ausgang der POS-Operation kurzzeitig ein Impuls abgegeben.

Tabelle 15 ist eine Wahrheitstabelle der Operation POS.

Eingangsoperand	Flankenoperand	Ausgangsoperand
0	0	0
0	1	1
1	1	0
1	0	0

Tabelle 15: Wahrheitstabelle der Operation POS

Zuerst wird der Netzschalter E8.7 und das SR-Glied M1.6 ausgeschaltet. Das heißt, es wird der Timer T1 bei der Einschaltung des Netzschalters gestartet, weil das SR-Glieds M1.6 ausgeschaltet wird. Der Ausgangszustand ist inzwischen 0. Bei der Einschaltung des SR-Gliedes M1.6 wird der Eingangsoperand und Flankenoperand als 1 geschrieben, und es ändert sich der Ausgangsoperand der Operation POS mit 0. Wenn das SR-Glied M1.6 ausgeschaltet wird, ist das Ergebnis nach der Operation 1. Gleichzeitig erfasst der Timer T1 eine steigende Flanke noch einmal.

Die sechzehn SR-Glieder werden miteinander verzahnt. Der Schalter der SR-Glieder wird auch als Rücksetzbedingung der anderen SR-Glieder und des Timers eingestellt.

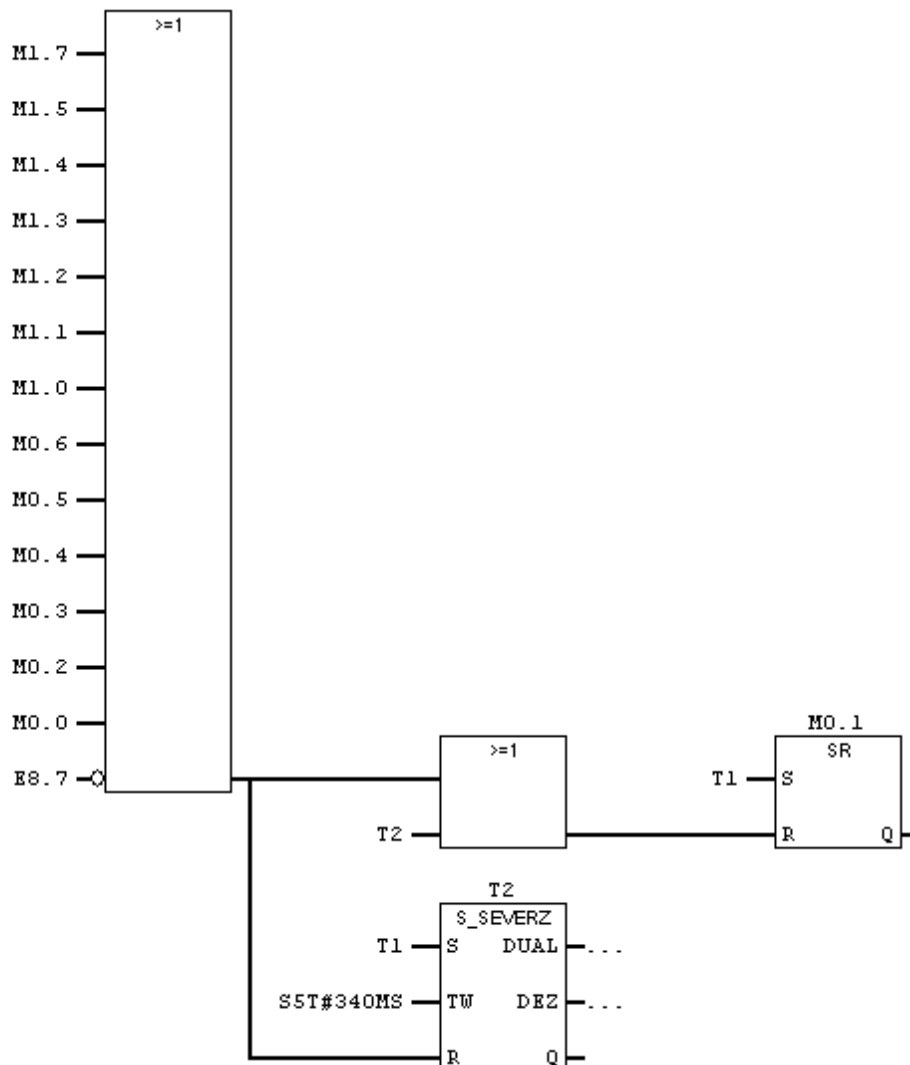


Abb. 3.9: Anweisung von Hauptprogramm: 2. u to a (-y)

Abbildung 3.9 zeigt die Anweisung der Bewegung des Manipulators vom Ursprung in die Anfangsposition in die negative Y-Achsen-Richtung. Die Startbedingung des SR-Glieds M0.1 und des Timers T1 ist die Einschaltung des Timers T1, und die Rücksetzbedingung ist die Einschaltung des Timers T2 und Verzahnung von anderen SR-Gliedern. Im Hauptprogramm werden die meisten Anweisungen der Schritte so geschrieben.

Die Steuerung von Motor 3 ist abhängig vom Eintritt des Signals der magnetischen Sensoren. Abbildung 3.10 zeigt ein Beispiel der Anweisung für die Steuerung von Motor 3. Also ist

die Rücksetzbedingung des SR-Glieds M0.2 ein Signal von dem rechten Sensor E8.0. Gleichzeitig ist das Signal die Startbedingung des SR-Gliedes M0.3 und des Timers T3.

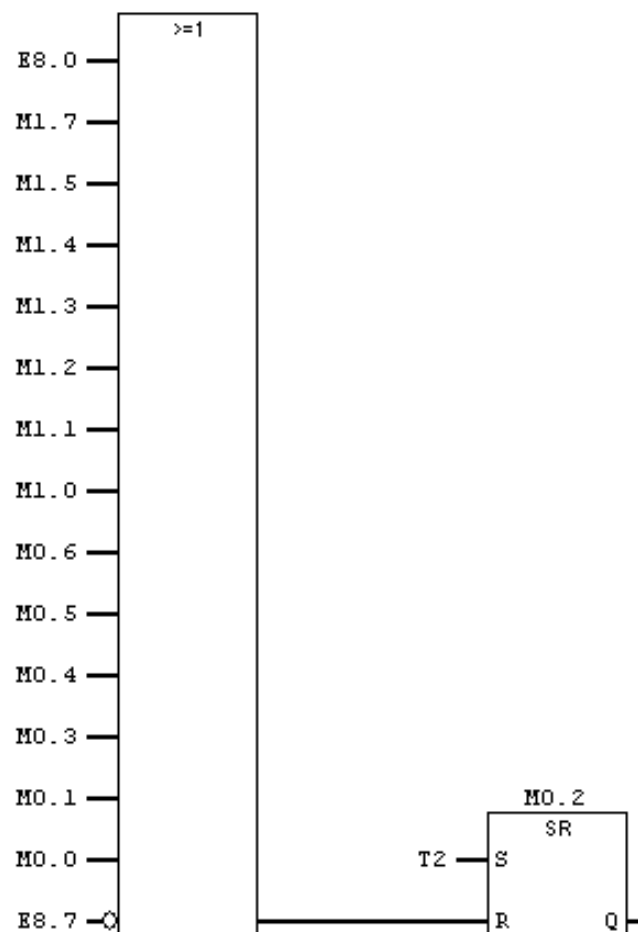


Abb. 3.10: Anweisung von Hauptprogramm: 3. u to a (-z)

Die Bewegungsabstände vom Ursprung in die Zielposition in die X-Richtung sind nicht gleich. Mit der Plattensetzung wird der Abstand immer größer. Diese Änderung wird durch eine Schleifenanweisung erreicht. Bei der Zeitsteuerung kann die Schleifenanweisung einfach mithilfe des Zählers geschrieben werden. Das SR-Glied M1.0 schaltet sich jeweils ein, der Zähler zählt ein Mal. Bis der Zähler sechs Mal zählt, wird der Zähler rückgesetzt, dann wird diese Schleife beendet. Die Vergleichsfunktion $CMP==1$ wird verwendet, um den entsprechenden Timer zu schalten. Das heißt, dass sich der Manipulator wegen der verschiedenen Zeitwerte von sechs Timern in die entsprechende Zielposition bewegen kann.

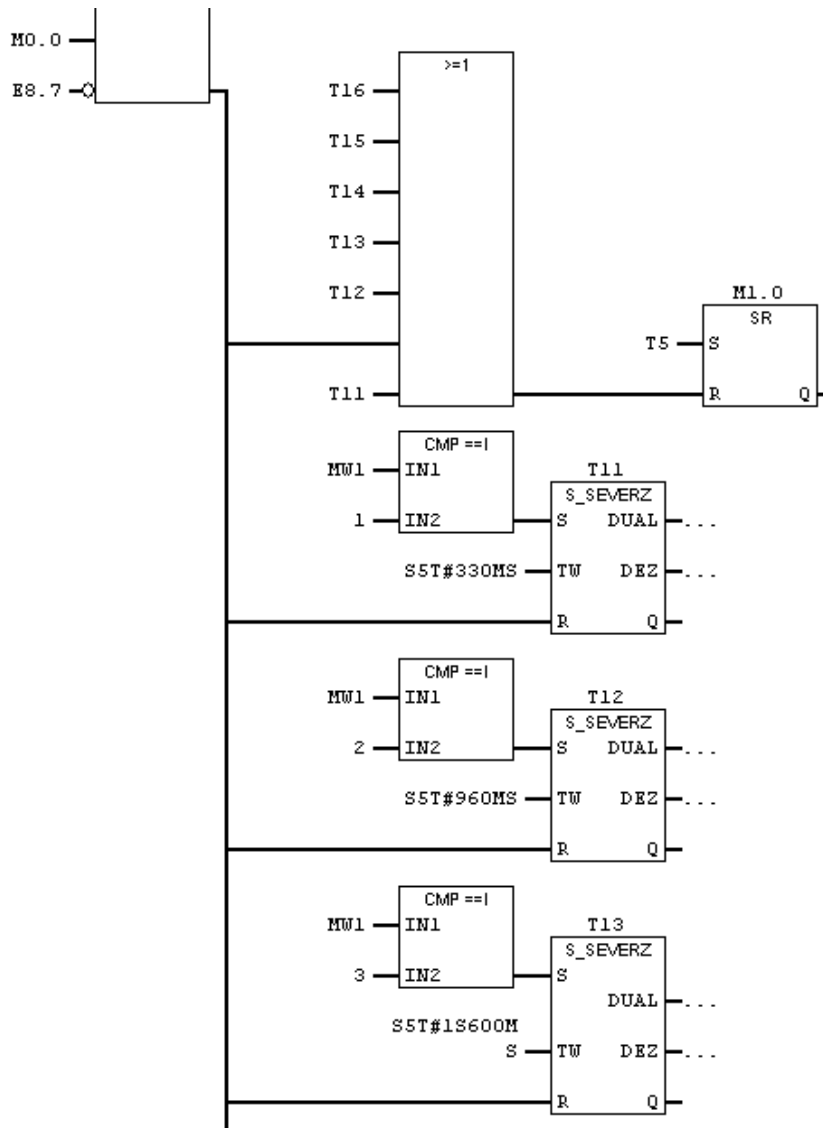
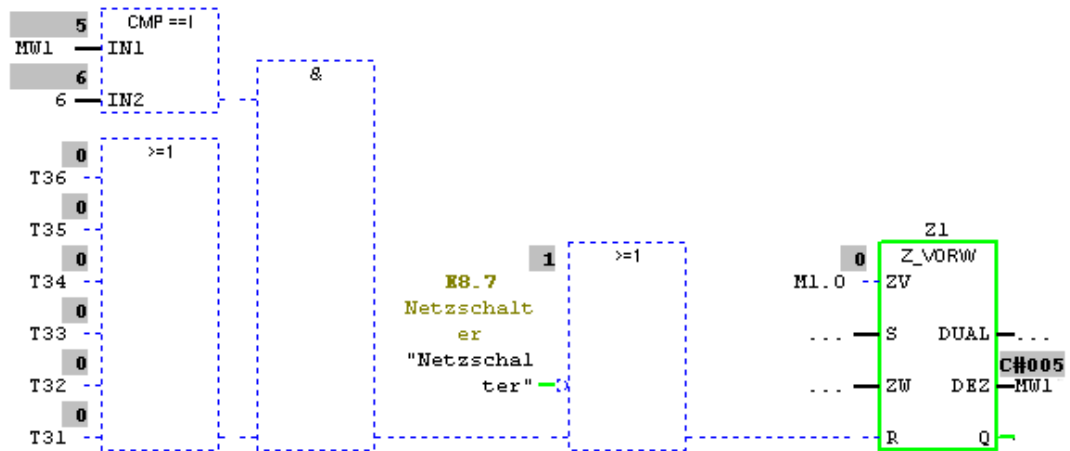


Abb. 3.11: Teil der Anweisung von Hauptprogramm: 8. u to z (+x)

Weil die Anweisung von Schritt 8 zu lang ist, zeigt die Abbildung 3.11 nur einen wichtigen Teil der Anweisung dieses Schritts. Die Anweisung von Schritt 14 wird ebenso geschrieben. Die Startbedingungen des Timers von Schritt 9 und Schritt 13 unterscheiden sich von Schritt 8. Es gibt noch eine Bedingung, die die Rückbedingung des letzten SR-Glieds ist. Für Schritt 9 ist die Startbedingung des Timers die Einschaltung des Timers von Schritt 8. Für Schritt 9 ist die Startbedingung des Timers der Eintritt des Signals von Sensor m. Das SR-Glied M1.1 steuert den Vorgang der Bewegung vom Ursprung in die Zielposition in die positive Y-Richtung. Der Zähler Z2 von Schritt 9 zählt einmal am Ende der Schleife von Schritt 8. Wenn diese Schleife von Schritt 9 beendet ist, hat jede Position in der A-Ebene eine Platte.

Netzwerk 21: Zähler +X

Zähler der X-Achsen für Schleife



Netzwerk 22: Zähler +Y

Zähler der Y-Achsen für Schleife

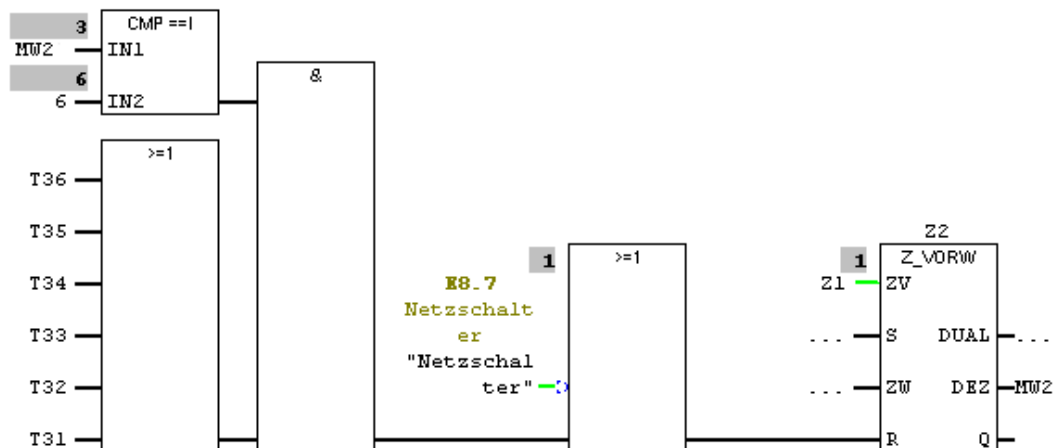


Abb. 3.12: Anweisung und Zustand der Zähler für Schleife

Abbildung 3.12 zeigt die Anweisung der Einstellung der Zähler für die Schleife und den Zustand der Zähler der X-Achsen und Y-Achse.

Außer der Ausschaltung des Netzschalters und der Vergleichsoperation wird die Rücksetzbedingung der Zähler mit einer UND-Verknüpfung eingestellt. Die Bedeutung der Einschaltung des Timers von Schritt 14 ist die, dass sich der Zähler nach Ende eines ganzen Platten-transportzyklus zurücksetzen muss. Der Zustand in Abbildung 3.12 zeigt, dass die Platte bisher schon an der Zielposition in der fünften Zeile und der dritten Spalte transportiert wurde.

In diesem Programm wird die Anweisung der Steuerung von Motoren geschrieben, damit man die Ablaufreihenfolge von Motoren bei dem Programmlauf beobachten kann.

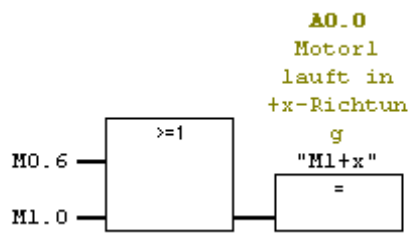


Abb. 3.13: Anweisung des Vorwärtslaufs von Motor 1

Ein Beispiel ist in Abbildung 3.13 dargestellt. Bei der Einschaltung des SR-Glieds M0.6 oder M1.0 läuft der Motor 1 vorwärts.

Das ganze Programm bei der Zeitsteuerung wird im Anhang dargestellt.

3.2.2 Programmierung bei der Signalsteuerung

Bei der Programmierung der Signalsteuerung ohne Positionsbaugruppe wird hauptsächlich der Zähler verwendet.

Bei Schritt 1 und Schritt 7 bzw. Schritt 2 und Schritt 6 bewegt sich der Manipulator in gleichen Abständen, aber in die entgegengesetzten Richtungen. Deswegen wird der Vorwärts- und Rückwärtszähler gewählt. Weil die Zentralbaugruppe CPU315-2-DP keinen schnellen Zähler hat, wird der normale Zähler verwendet, um die gleiche Funktion zu simulieren.

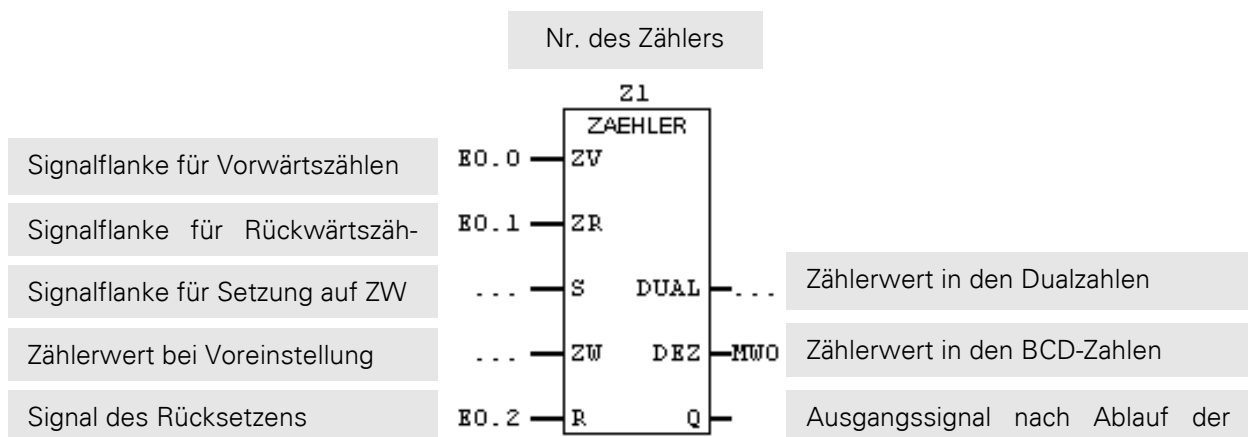


Abb. 3.14: Schema des Vorwärts- und Rückwärtszählers

Am Vorwärtszählen-Eingang und am Rückwärtszählen-Eingang erfolgt ein Impulssignal über eine steigende Flanke. Der Zählerwert am Ausgang wird im Zahlenformat von BCD-Zahlen gespeichert. Das Merkerwort muss mit dem eingestellten Wert verglichen werden, aber beim Vergleich sind die zwei Werte im Zahlenformat von Integer-Zahlen. Der Wert über 9 der BCD-Zahlen und Integer-Zahlen ist nicht gleich. Also muss der Merkerwort MW0 im Zahlenformat von Integer-Zahlen umgewandelt werden. Dazu wird die Umwandlungsfunktion von BCD_I verwendet. Die Funktion ist in Abbildung 3.13 dargestellt. Der Wert der BCD-Zahlen wird am Eingang eingegeben und den Wert der Integer-Zahlen erhält man am Ausgang.

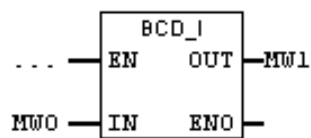


Abb. 3.15: Schema des Vorwärts- und Rückwärtszählers

Das Programm ist fast gleich wie das Programm bei der Zeitsteuerung. Der Unterschied besteht in der Rücksetzbedingung. Die Rücksetzbedingung des SR- Gliedes ist bei der Zeitsteuerung die Einschaltung nach dem Ablauf der eingestellten Zeit. Und bei der Signalsteuerung ist es der Vergleich zwischen dem Merkerwort des Zählers und dem eingestellten Wert.

Symbol	Adresse	Datentype	Kommentar
M1_+x	A 0.0	BOOL	Motor1 läuft in +x-Richtung
M1_-x	A 0.1	BOOL	Motor1 läuft in -x-Richtung
M2_+y	A 0.2	BOOL	Motor2 läuft in +y-Richtung
M2_-y	A 0.3	BOOL	Motor2 läuft in -y-Richtung
M3_+z	A 0.4	BOOL	Motor3 läuft in +z-Richtung
M3_-z	A 0.5	BOOL	Motor3 läuft in -z-Richtung
G1_A	E 0.0	BOOL	Geber 1 von Motor 1
G2_A	E 0.1	BOOL	Geber 2 von Motor 2
Sensor r	E 8.0	BOOL	Sensor der rechten Position von Motor 3 (-z-Achse)
Sensor m	E 8.1	BOOL	Sensor der mitten Position von Motor 3 (z-Achse)
Sensor l	E 8.2	BOOL	Sensor der linken Position von Motor 3 (z-Achse)
Netzschalter	E 8.7	BOOL	Netzschalter

Tabelle 16: Symboltabelle bei der Signalsteuerung von STEP 7

Man kann aus dem Vergleich zwischen Tabelle 14 und Tabelle 16 erkennen, dass die Einstellung der Symboltabelle für beide Steuerungen neben zwei zusätzlichen Eingängen des Gebers gleich ist. Die Anweisungen von beiden Steuerungen sind ähnlich. Die Werte für Vergleichsfunktion werden nach Tabelle 7 eingestellt. Der Schritt 2 des Hauptprogramms ist wie in Abbildung 3.16 dargestellt, dass das SR-Glied durch die Vergleichsfunktion zurücksetzt.

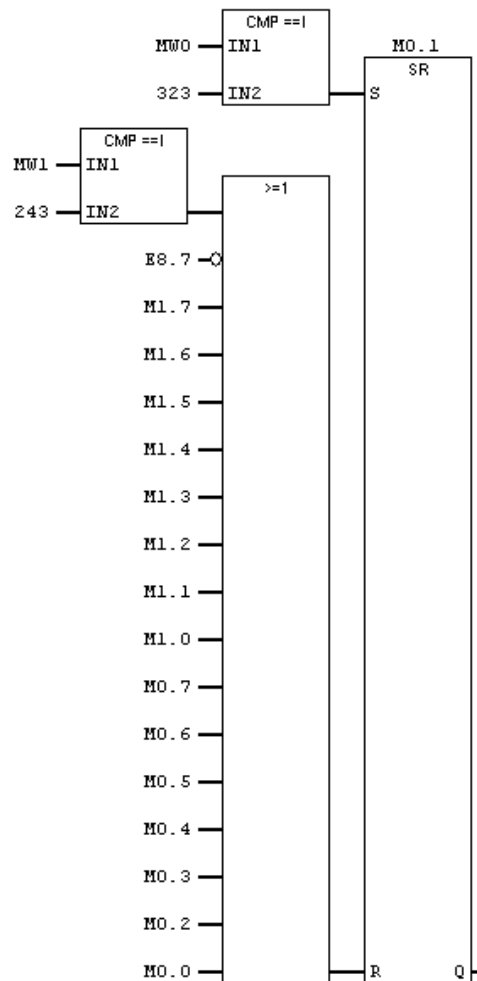


Abb. 3.16: Anweisung des Hauptprogramms bei der Signalsteuerung: 2. u to a (-y)

Die Schleifenanweisung ist komplexer als die Anweisung bei der Zeitsteuerung.

Der Zählerstand kann Zahlenwerte innerhalb von 0 - 999 haben. Aber nach der Programmiernorm 61131-3 gibt es die Zählerwertgrenze von -32768 bis 32767. Wenn nur ein Zähler verwendet wird, ist der Zählerwert der Bewegung in der letzten Zielposition größer als der Maximalwert von 999. Außer dem Abstand zwischen dem Ursprung und der ersten Zielposition ist der Abstand zwischen den angrenzenden Zielpositionen gleich.

Man kann Schritt 8 in zwei kleine Schritte einteilen:

- Schritt 8.1: Bewegung vom Ursprung in die erste Zielposition
- Schritt 8.2: Bewegung von der ersten Zielposition in die weitere Zielposition

Die Anweisung von Schritt 8.1 ist ähnlich wie Schritt 1. Bei der Programmierung von Schritt 8.2 werden die korrelativen Zähler wie in Abbildung 3.17 verwendet. Der Hauptzähler wird bei der Bestimmung des Abstands zwischen den angrenzenden Zielpositionen verwendet. Wenn der Zähler Z50 einmal einschaltet, wird der Zählerwert des verbundenen Zählers plus 1.

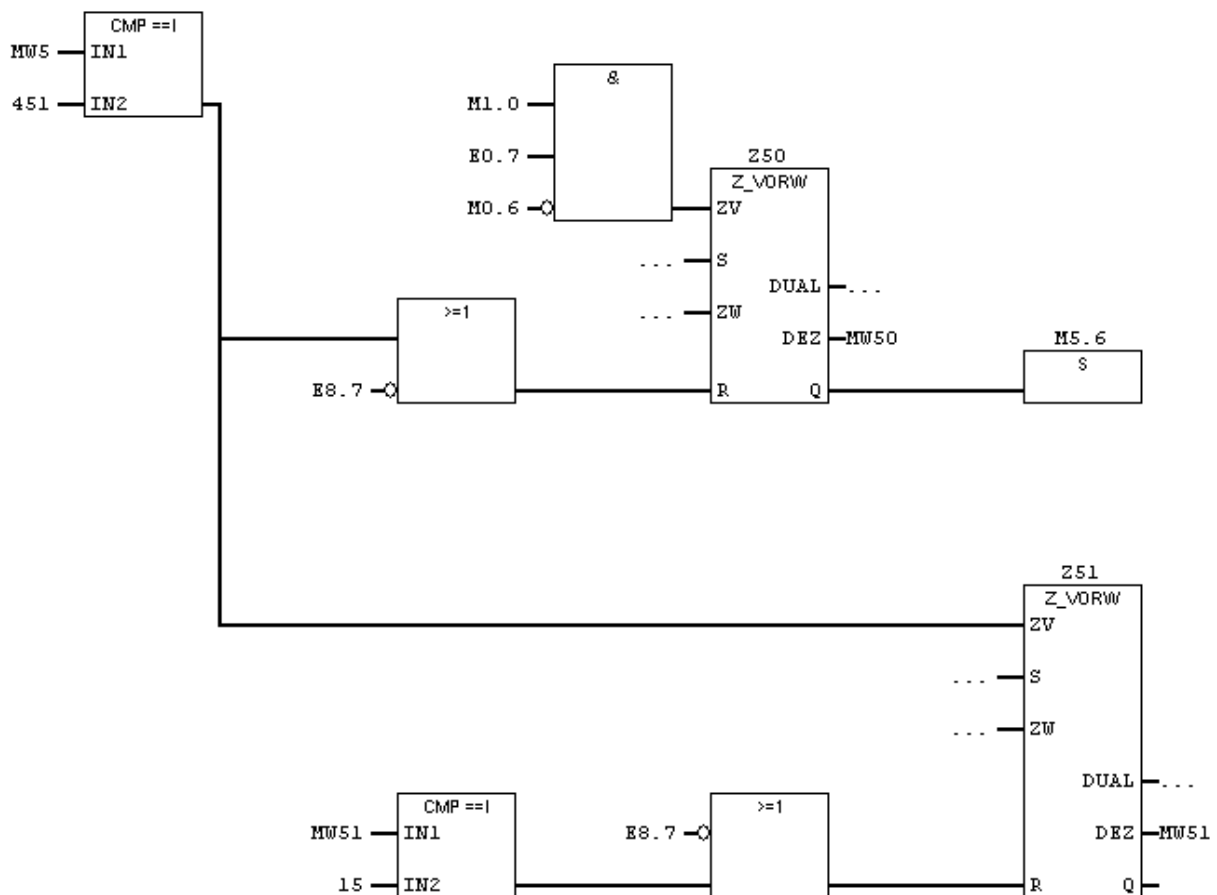


Abb. 3.17: Rücksetzen der Adresse M5.0

Die Anweisung, die in der nachfolgenden Abbildung 3.18 gezeigt wird, bestimmt die Position des Manipulators in der entsprechenden Achse. Wenn das Merkerwort gleich des eingestellten Wertes ist, schaltet sich der Merker M5.1 ein und der Manipulator stoppt gleichzeitig in der entsprechenden Position. Weil der Zähler vor dem Ende des Zyklus nicht rückgesetzt wird, wird der Zählerwert kumulativ addiert. Die Dreieckszahl wird wie der eingestellte Wert in der Vergleichsfunktion verwendet, d. h. 1, 3, 6, 10, 15.



Abb. 3.18: Zuweisung des Merkers

Damit die Rücksetzbedingung des SR-Glieds M1.0 in dem Hauptprogramm einfach geschrieben werden kann, wird der Zustand der Adresse M5.0 als die Rücksetzbedingung eingestellt. Die Anweisung ist in Abbildung 3.19 dargestellt. Wenn der Zustand 1 ist, setzt das SR-Glied auf M1.0. Bei der Einschaltung des SR-Glieds M1.0 wird der Zustand der Adresse M5.0 zurückgesetzt. Der Zustand der Adresse M5.0 wird durch die ODER-Verknüpfung von den Operationen POS gesetzt. Wenn sich der Manipulator jedes Mal in eine Zielposition bewegt, wird der Zustand der Adresse M5.0 einmal gesetzt.

Merker sind Adressregister innerhalb der CPU. Ein Merker entspricht einem eingestellten Wert. Dieser Wert ist die Ursache für die Verwendung von mehreren Merkern und Zählern.

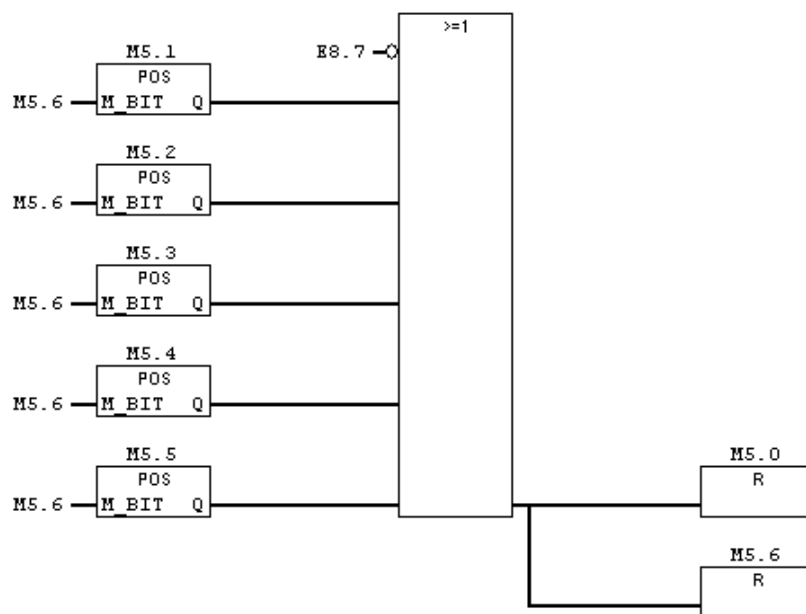


Abb. 3.19: Rücksetzen der Adresse M5.0

Neben der Bedingung der Zuweisung für die Steuerung von Motor 2 sind die Anweisungen für die Schleife fast gleich. Die Anweisung ist gleich wie die Anweisung bei der Zeitsteuerung. Die Änderung der Position in der Y-Achse ist abhängig von der Anzahl des Endes des Plattentransport-Zyklus.

3.3 Programmierung mit der Positionierbaugruppe

Die Programmierung mit der Positionierbaugruppe ist einfach. Wie Abbildung 3.19 zeigt, kann man durch die Funktion „Parametrieren“ folgende Daten einstellen, um die Bausteine zu erstellen.

- P-Parameter
- GUD-Programm
- NC-Programme
- Maschinendaten
- Werkzeugkorrekturwerte
- Nullpunktverschiebungen

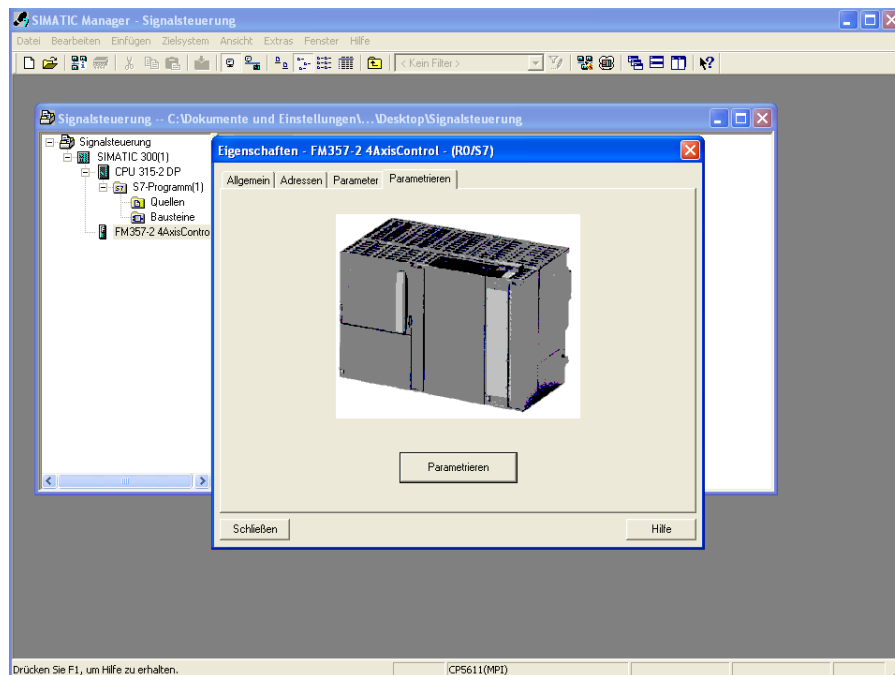


Abb. 3.20: Dialogfeld der Eigenschaft von FM357-2

Die folgende Abbildung 3.21 zeigt ein Beispiel für die Einstellung des Parameters des Motors. Man kann die Daten von den verschiedenen Motoren in die vier Achsen einstellen. In diesem Plattentransportsystem werden nur die Parameter des Motors für die X-Achse und die Y-Achse eingestellt, weil der Motor 3 durch den Sensor gesteuert wird.

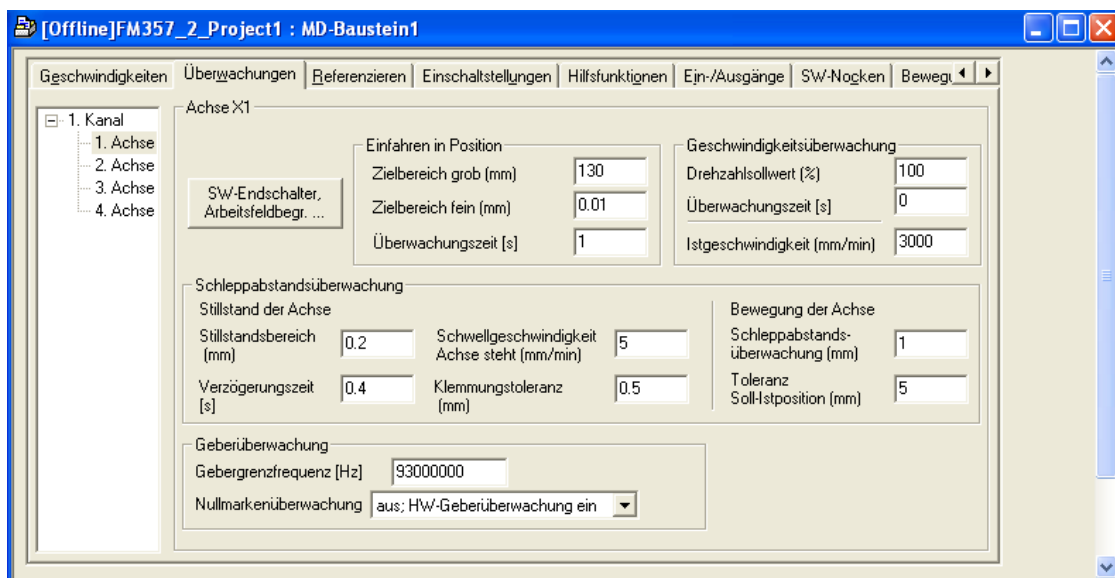
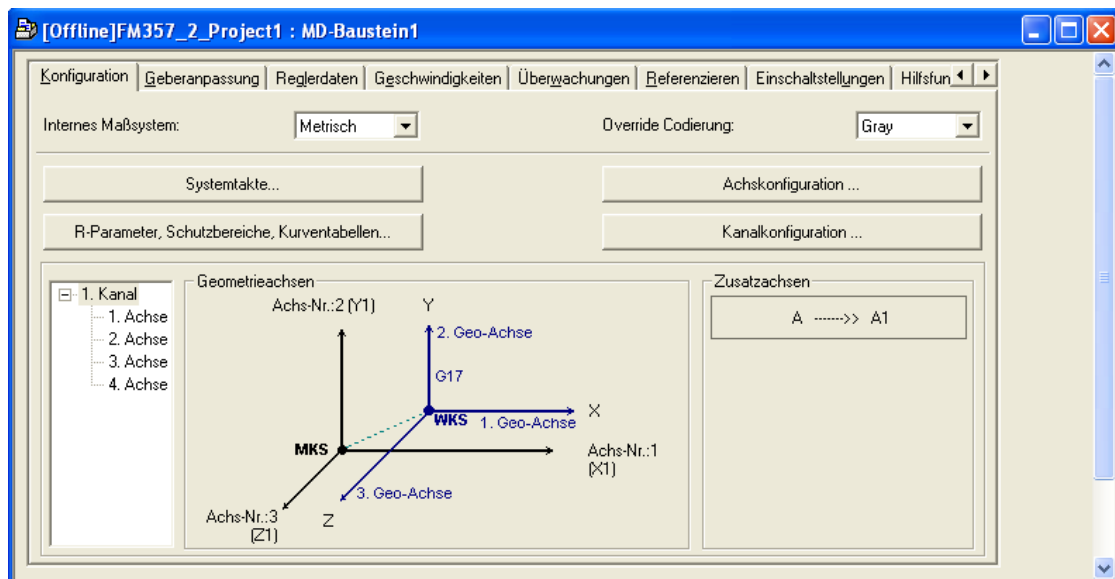


Abb. 3.21: Dialogfeld des Parametrierens der Maschinendaten

Abbildung 3.21 zeigt das Parametrieren bei der Überwachung. Der Parameter kann durch die Messung und Bedienungsanleitung erfasst werden. Dann wird der Parameter von der Bewegungsbahn, dem Motor bzw. Geber in dem Dialogfeld eingegeben.

Nach dem Parametrieren werden einige Globalbausteine erzeugt. Diese Bausteine können in der Bibliothek des Simatic-Managers gefunden und direkt aufgerufen werden. Ein Beispiel ist in Abbildung 3.22 dargestellt. Die Positionsdaten werden am Eingang von diesem FB-Baustein eingegeben, damit die Bewegung des Manipulators gesteuert wird.

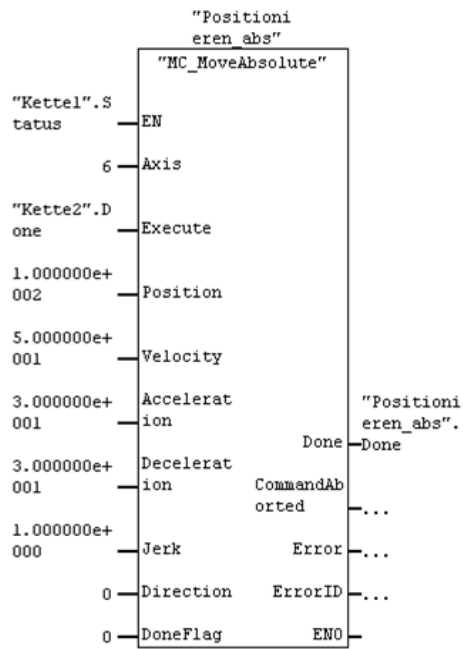


Abb. 3.22: Baustein im Simatic-Manager

Quelle:

<https://wenku.baidu.com/view/136cd7c658f5f61fb7366682.html> (04.07.2017)

3.4 Vergleich von drei Programmen

Die drei Programmierarten haben ihre eigenen Vor- und Nachteile.

Programmierungsarten	Programmierung ohne die Positionierbaugruppe		Programmierung mit der Positionierbaugruppe
	Programmierung bei der Zeitsteuerung	Programmierung bei der Signalsteuerung	
Komplexität des Programms	Einfach	Sehr komplex	Komplex
Programmfehler	Einfach zu finden	Nicht einfach zu finden	Einfach zu finden
Bewegungsabstandsfehler	Ja	Ja, klein	Gering, vernachlässigt

Tabelle 17: Vergleich von drei Programmen

Das Programm bei der Zeitsteuerung ist einfacher als andere Programme. Bei der Programmierung werden die grundlegenden Anweisungen, auch Schleife-Anweisung, verwendet. Wegen des einfachen Programms können die Programmfehler einfach gefunden werden. Der Debugger ist auch einfach. Aber es gibt Bewegungsabstandsfehler wegen der Zeitbasis. Obwohl die Hauptprogramme bei der Signalsteuerung und bei der Zeitsteuerung ähnlich sind, ist die Schleife-Anweisung komplex. Die Schleife-Anweisung in der Programmiersprache von SPS ist nicht so kurz zu schreiben wie in Programmiersprache C++. Mehrere Merker werden in dem Programm verwendet, und die Bedingung ist auch komplex. Weil das Programm komplex ist, besteht die Möglichkeit, dass Programmfehler auftreten. Die Genauigkeit dieses Programms ist höher als das Programm bei der Zeitsteuerung. Die Anzahl der Impulse weisen bei der Berechnung kleine Fehler auf. Die Bewegung bei der Signalsteuerung ist genauer als bei der Zeitsteuerung.

Die Schwierigkeit bei der Programmierung mit der Positionierbaugruppe ist keine Schleife-Anweisung, sondern die Erstellung und Anwendung des Bausteins. Das Programmierkonzept des Programms mit der Positionierbaugruppe ist einfacher als das des Programms ohne Positionierbaugruppe. Man muss die Positionierbaugruppe schnell kennen, um besser zu programmieren. Die Genauigkeit des Programms kann anhand der Abbildung 3.21 eingesehen werden. Der Bewegungsabstandsfehler ist gering, er kann vernachlässigt werden.

Wenn die Genauigkeit nicht anspruchsvoll ist, wird das Programm bei der Zeitsteuerung empfohlen. Weil die Bewegungsbahn einfach ist, ist das Programm auch einfach. Heute wird häufiger die Positionierbaugruppe für die automatisierte Bewegungsführung vom einfachen Positionieren bis zur Bahnsteuerung verwendet. Wenn die Bewegungsbahn noch komplizierter eingestellt wird, ist die Verwendung der Positionierbaugruppe am besten.

4 Zusammenfassung und Ausblick

Die Auswahl der Hardware ist in dieser Arbeit etwas anders. Normalerweise wird die Hardware zuerst laut des Programmierkonzepts ausgewählt. Aber in dieser Arbeit wird das Programm nach der vorhandenen Hardware geschrieben. Deshalb wird das Programmierkonzept eingeschränkt. Aber es wird versucht, verschiedene Möglichkeiten zum Programmieren herauszufinden und zu untersuchen. Außerdem muss man die Hardware vor der Auswahl gut kennen, weil alle Daten unbekannt sind. Nicht nur die Kenntnisse über die Steuerungstechnik, sondern auch Kenntnisse aus anderen Bereichen werden in dieser Arbeit angewendet, z. B. Digitaltechnik, Leistungstechnik.

In dieser Arbeit wird ein veraltetes Gerät verwendet, das für das Erlernen und die Anwendung der grundlegenden Programmierung gut geeignet ist. Aber heutzutage wurden für die praktische Anwendung einige Geräte eliminiert und neue Geräte werden verwendet. Die Funktion der Hardware ist fast gleich, z. B. zeigt Abbildung 4.1 die Programmierung mit der Zentralgruppe CPU1516. Das Programm mit der Zentralgruppe CPU 1516 ist ähnlich wie das Programm mit der Zentralgruppe CPU314C. Durch die Installation werden die Globalbausteine verwendet. Mit denen wird das Programm über das Positionieren geschrieben. Wie mit den neuen Geräten programmiert wird, wird in der Praxis und täglichen Arbeit erlernt.

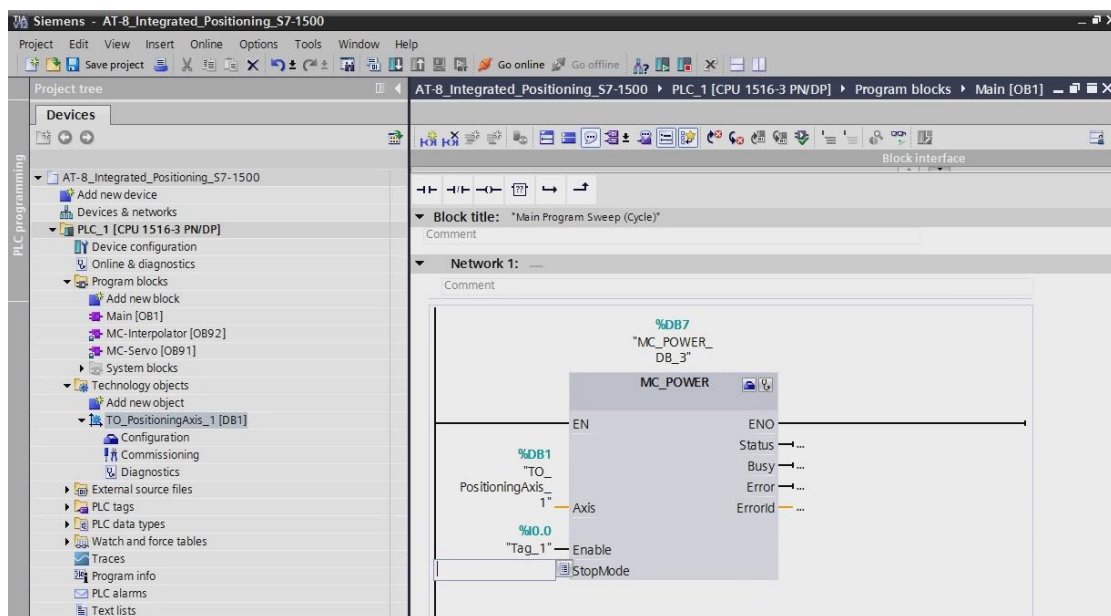


Abb. 4.1: Programmierung mit der Zentralgruppe CPU1516

Quelle:

<https://www.youtube.com/watch?v=ZRSuV3Zs2P0> (16.07.2017)

Anhang

Programm bei der Zeitsteuerung mit der Programmiersprache FUP

OB1 - <offline>

```

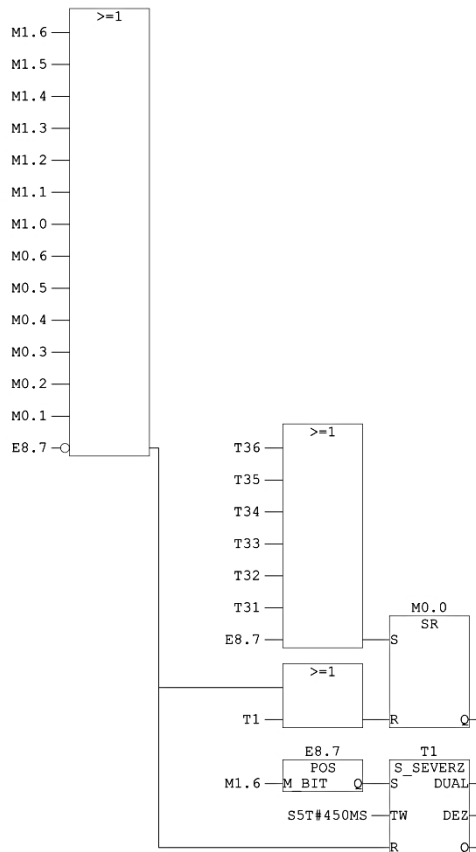
""
Name:                               Familie:
Autor:                               Version: 0.1
Zeitstempel Code:                   Bausteinversion: 2
                                     26.07.2017 12:59:43
Interface:                           12.07.2017 16:31:46
Längen (Baustein / Code / Daten): 02192 01998 00022

```

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)"

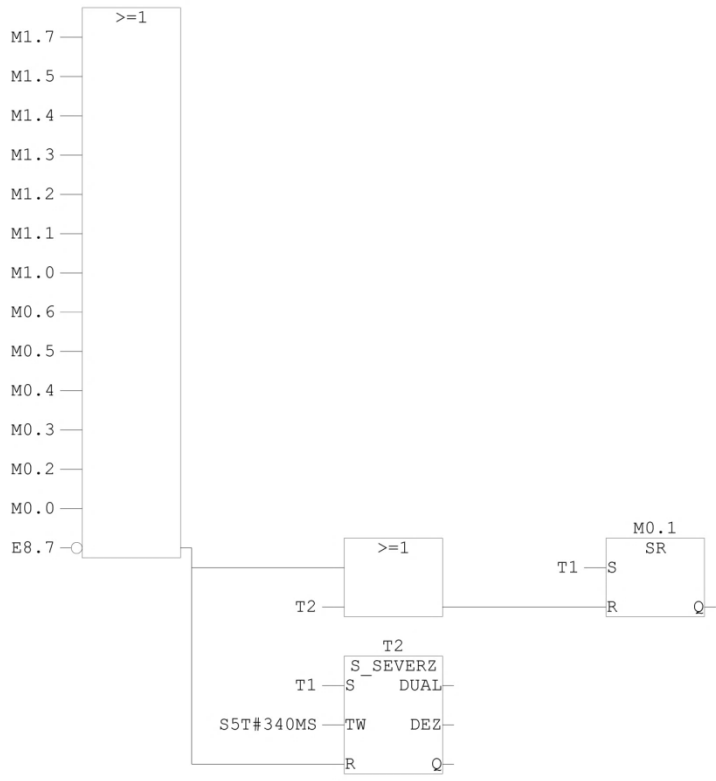
Netzwerk: 1 Hauptprogramm: 1. u to a(-x)



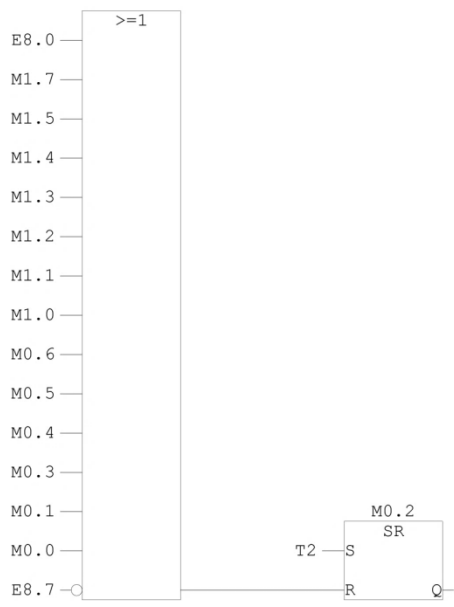
Symbolinformation

E8.7 Netzschalter Netzschalter

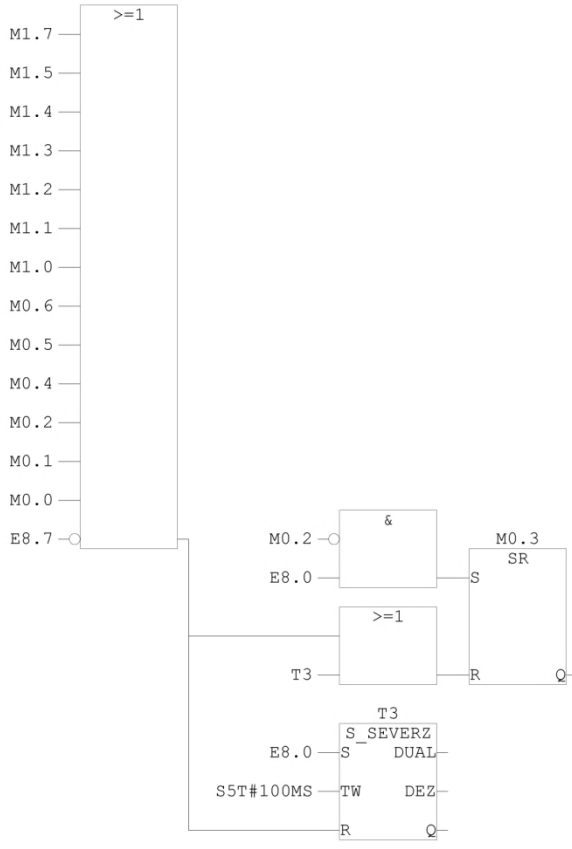
Netzwerk: 2 Hauptprogramm: 2. u to a(-y)



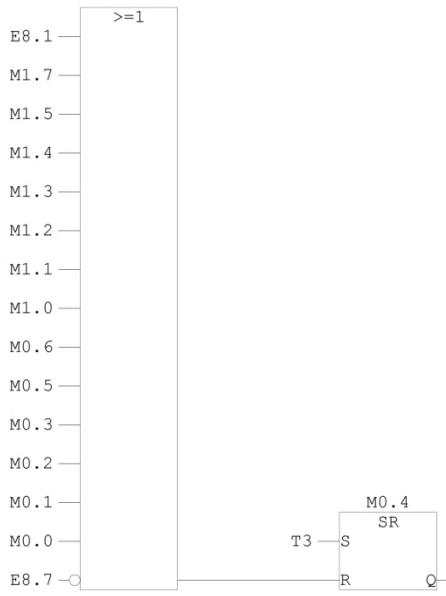
Netzwerk: 3 Hauptprogramm: 3. u to a(-z)



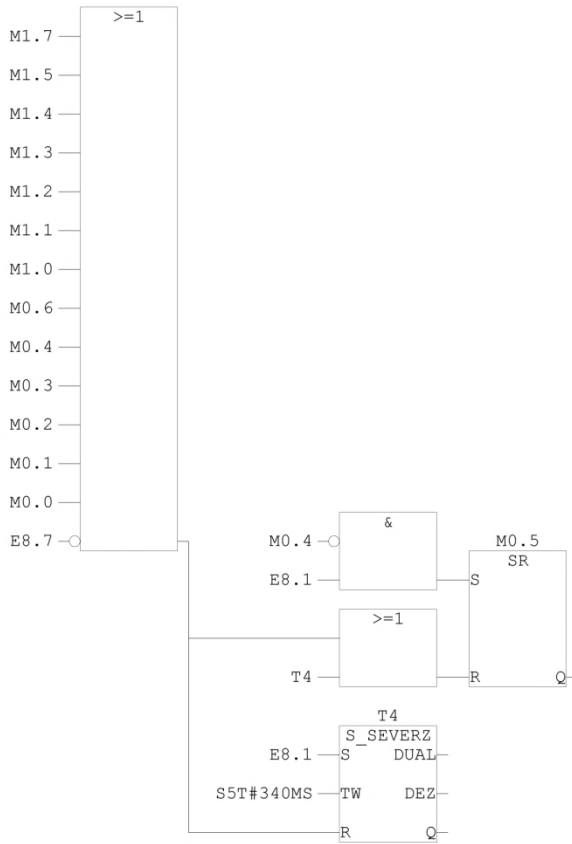
Netzwerk: 4 Hauptprogramm: 4. u to a(+y0)



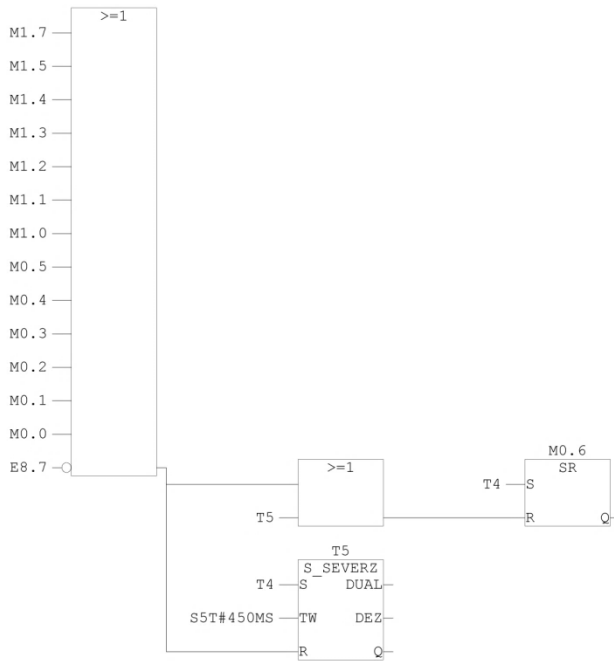
Netzwerk: 5 Hauptprogramm: 5. a to u(+z)



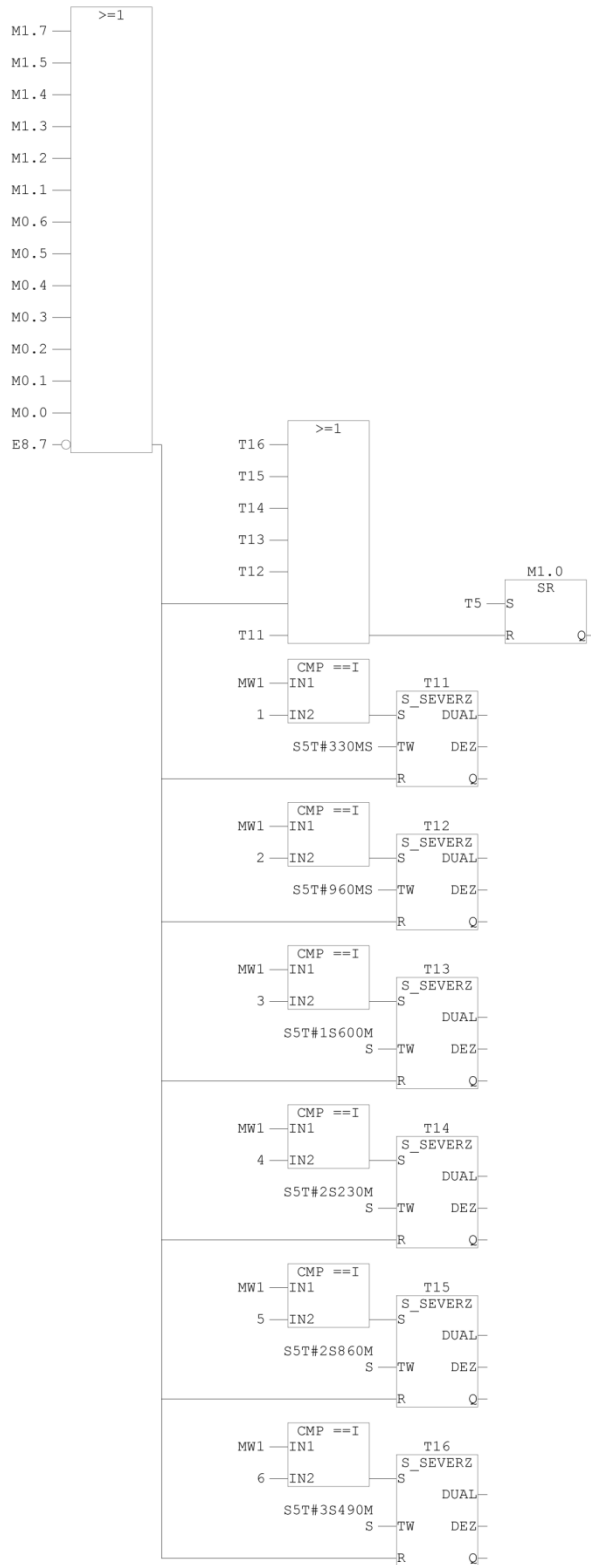
Netzwerk: 6 Hauptprogramm: 6. a to u(+y)



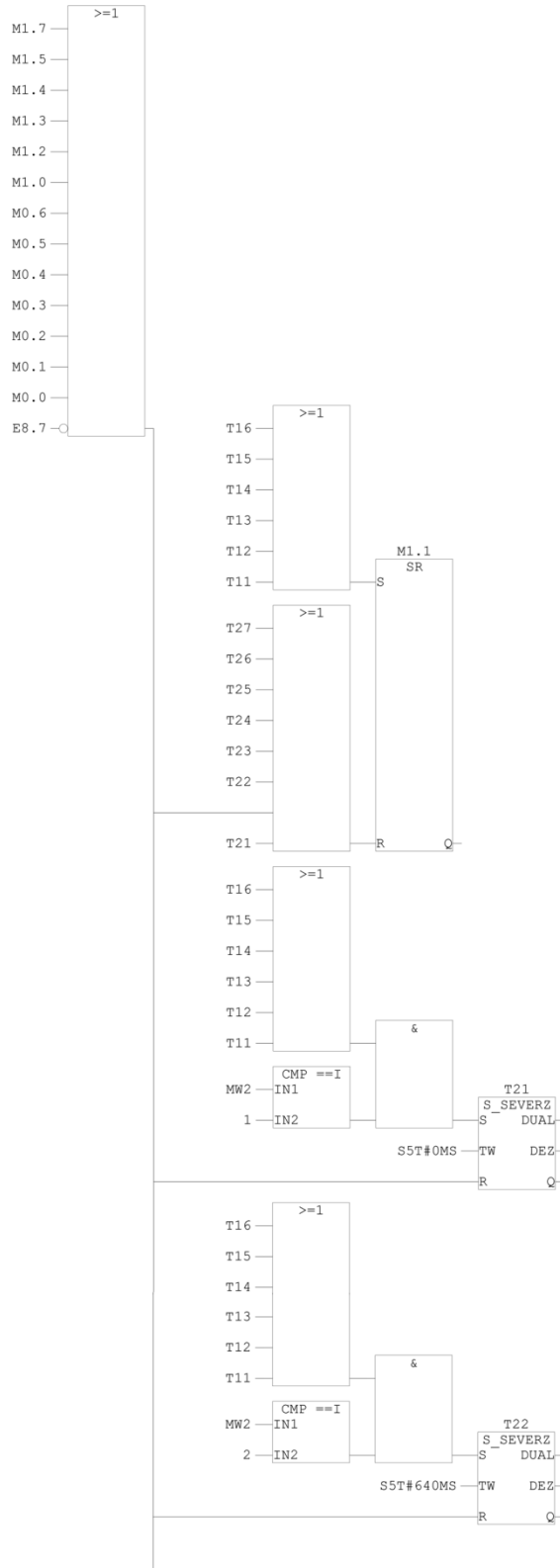
Netzwerk: 7 Hauptprogramm: 7. a to u(+x)

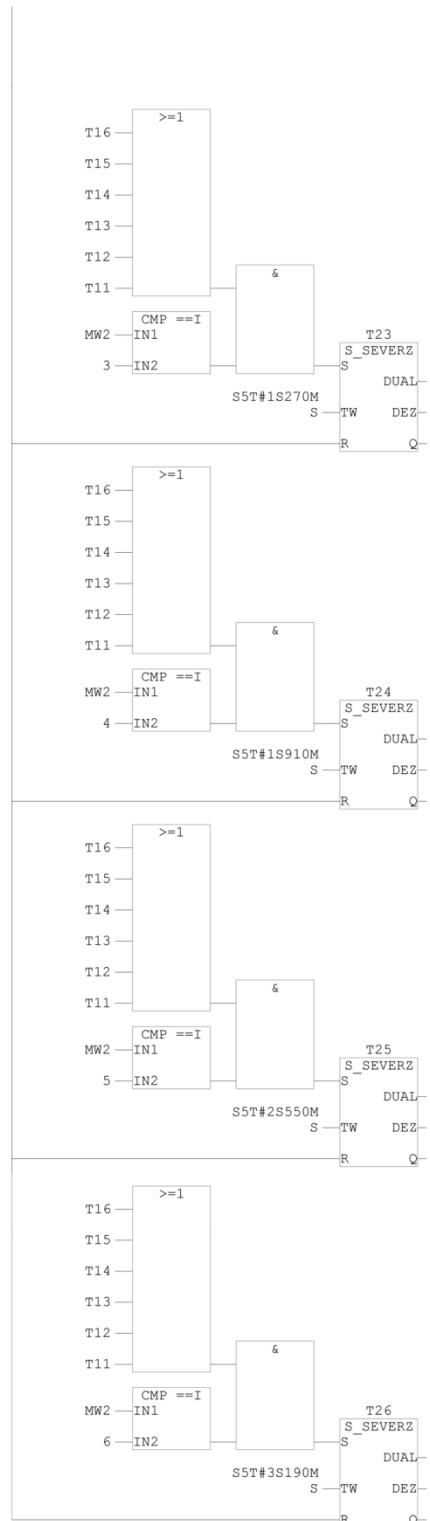


Netzwerk: 8 Hauptprogramm: 8. u to z(+x)



Netzwerk: 9 Hauptprogramm: 9. u to z1(+y)

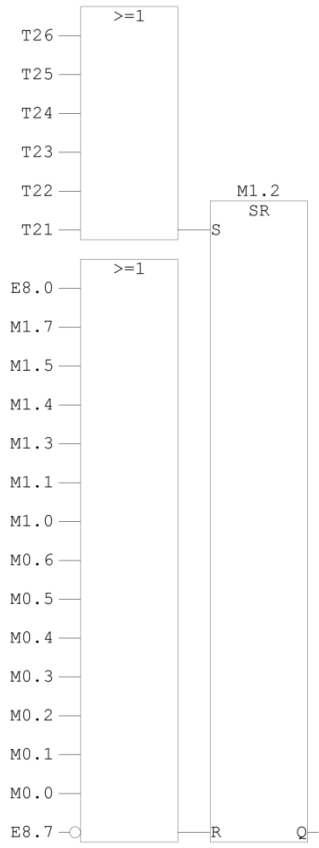




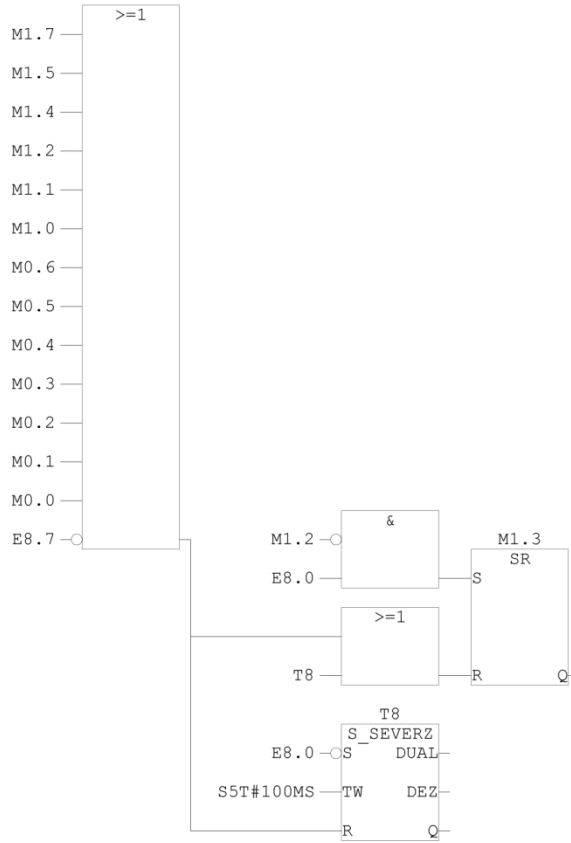
Symbolinformation

E8.7 Netzschalter Netzschalter

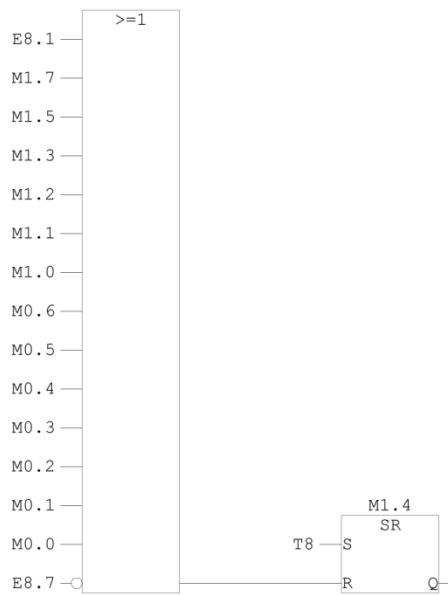
Netzwerk: 10 Hauptprogramm: 10. u to z1(-z)



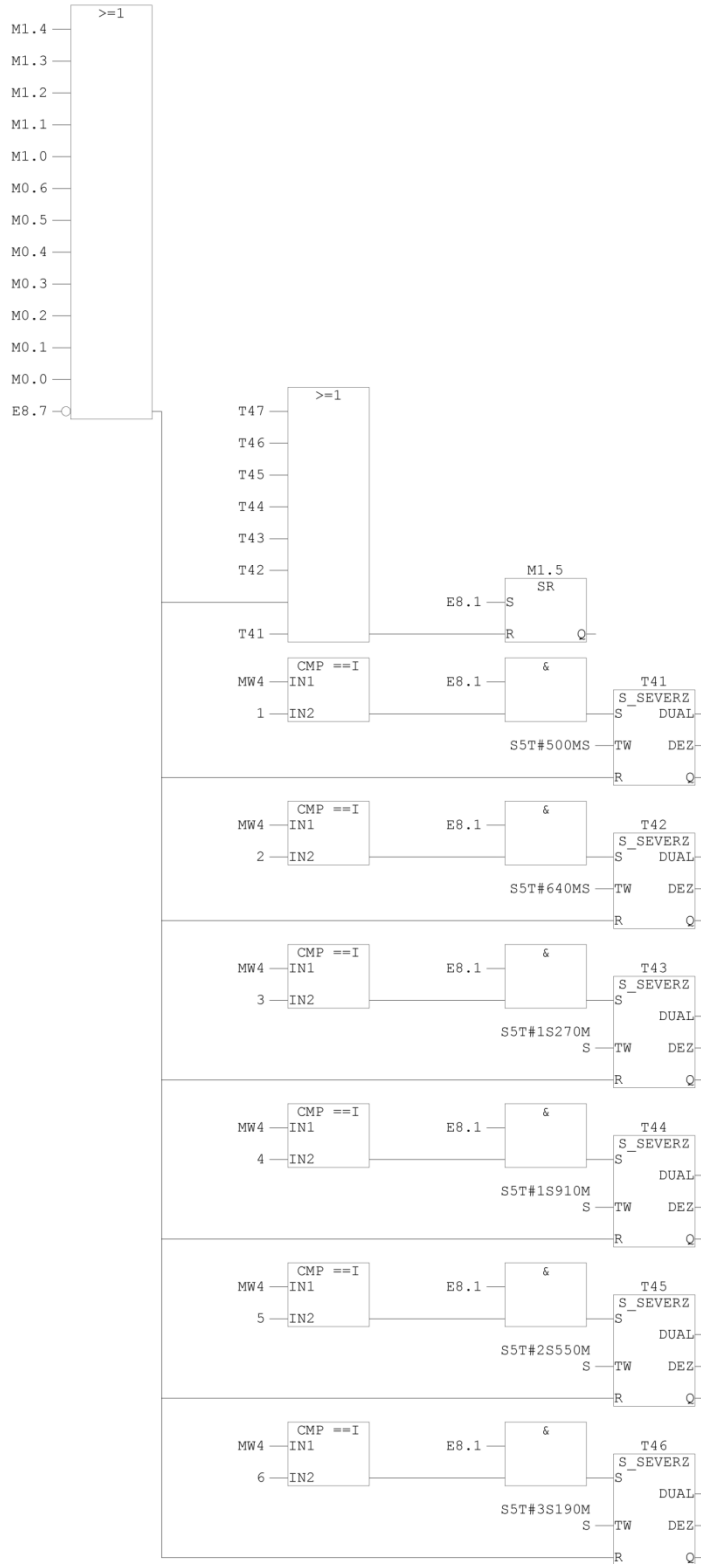
Netzwerk: 11 Hauptprogramm: 11. u to z1(-y0)



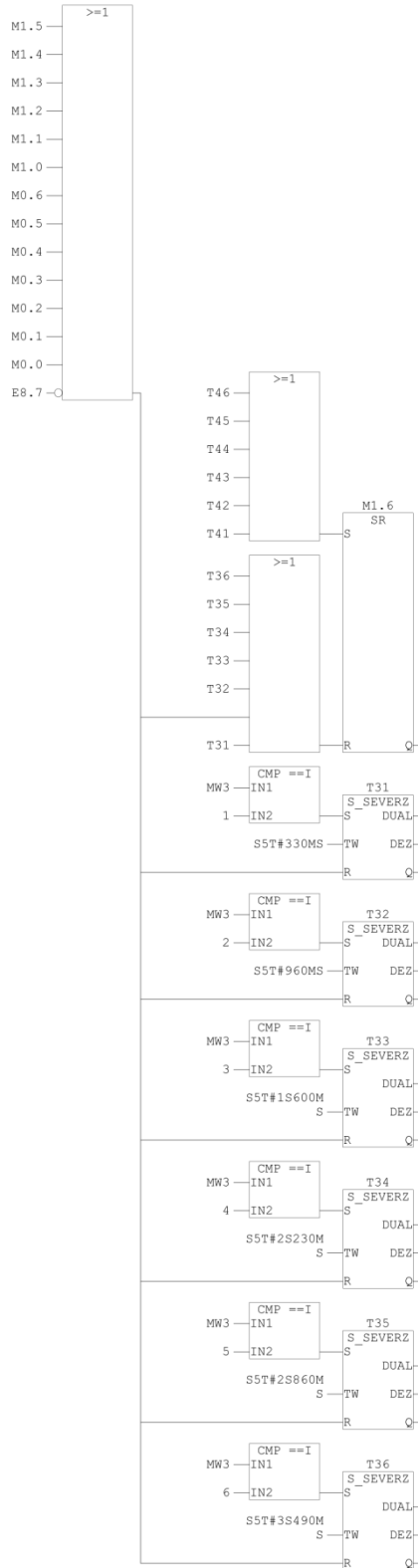
Netzwerk: 12 Hauptprogramm: 12. z1 to u(+z)



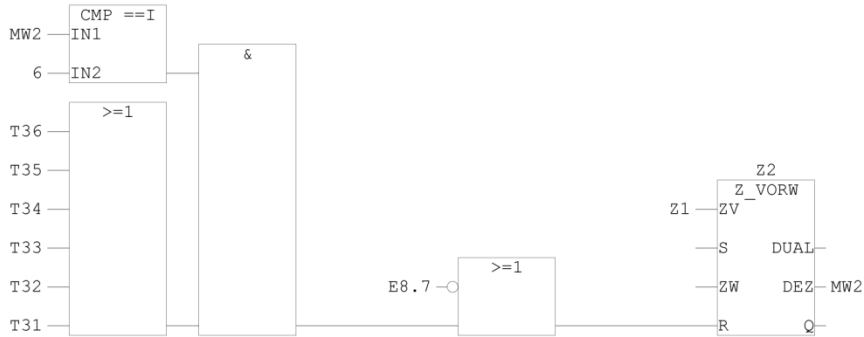
Netzwerk: 13 Hauptprogramm: 13. z1 to a(-y)



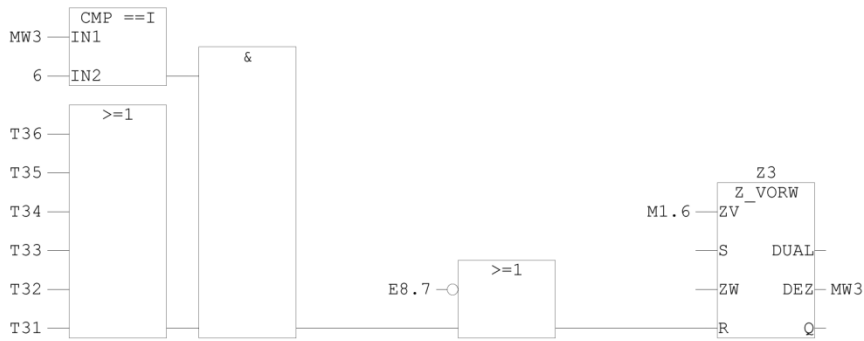
Netzwerk: 14 Hauptprogramm: 14. z1 to a(-x)



Netzwerk: 22 Zähler +Y



Netzwerk: 23 Zähler -X



Netzwerk: 24 Zähler -Y



Abbildungsverzeichnis

Abb. 1.1: Modul des Plattentransportsystems	1
Abb. 2.1: 3D-Modul des Rahmens und Erstellung von Ursprung und Koordinaten	2
Abb. 2.2: Hauptansicht (Nur A-Ebene)	3
Abb. 2.3: Rückansicht (Nur B-Ebene).....	4
Abb. 2.4: Linksansicht	4
Abb. 2.5: Erstellung des Mittelpunkts.....	4
Abb. 2.6: Erstellung des Mittelpunkts.....	4
Abb. 2.7: Hauptdaten des Rahmens	5
Abb. 2.8: DC-Motor-Geber-Modul.....	6
Abb. 2.9: DC-Motor GR-42×40-24V (Dunkermotoren).....	7
Abb. 2.10: Getriebe SG62-72:1 (Dunkermotoren)	8
Abb. 2.11: Gebersystem RE30-3-500 (Dunkermotoren).....	9
Abb. 2.12: DC-Motor G30.2-24V (Dunkermotoren)	13
Abb. 2.13: Näherungsschalter SME-8-S-KED-24 (FESTO)	14
Abb. 2.14: Anschlussbild der Relaisbausteine für Vorwärtslauf und Rückwärtslauf (Motor 3)	15
Abb. 2.15: 4Q-PWM-Servoregler MTRG-25/2E (mattkeag)	16
Abb. 2.16: Stromversorgungsbaugruppe PS307-5A.....	18
Abb. 2.17: Zentralbaugruppe CPU315-2-DP.....	19
Abb. 2.18: Zentralbaugruppe CPU314C-2-DP	19
Abb. 2.19: Zentralbaugruppen CPU314C-2-DP	20
Abb. 2.20: Positionierbaugruppe FM357-2.....	21
Abb. 3.1: HW Konfiguration des Simatic-Managers	22
Abb. 3.2: Beispiel der Programmiersprachen FUP, KOP und AWL	23
Abb. 3.3: Flussdiagramm.....	24
Abb. 3.4: Zustandsflussdiagramm	25
Abb. 3.5: Schema des Einschaltverzögerungs-Timers	26
Abb. 3.6: Schema des SR-Glieds	27
Abb. 3.7: Anweisung von Hauptprogramm: 1. u to a (-x)	28
Abb. 3.8: Schema der POS-Operation	29
Abb. 3.9: Anweisung von Hauptprogramm: 2. u to a (-y)	30
Abb. 3.10: Anweisung von Hauptprogramm: 3. u to a (-z).....	31
Abb. 3.11: Teil der Anweisung von Hauptprogramm: 8. u to z (+x).....	32
Abb. 3.12: Anweisung und Zustand der Zähler für Schleife	33

Abb. 3.13: Anweisung des Vorwärtslaufs von Motor 1	34
Abb. 3.14: Schema des Vorwärts- und Rückwärtszählers	34
Abb. 3.15: Schema des Vorwärts- und Rückwärtszählers	35
Abb. 3.16: Anweisung des Hauptprogramms bei der Signalsteuerung: 2. u to a (-y).....	36
Abb. 3.17: Rücksetzen der Adresse M5.0	37
Abb. 3.18: Zuweisung des Merkers	38
Abb. 3.19: Rücksetzen der Adresse M5.0	38
Abb. 3.20: Dialogfeld der Eigenschaft von FM357-2	39
Abb. 3.21: Dialogfeld des Parametrierens der Maschinendaten	40
Abb. 3.22: Baustein im Simatic-Manager.....	41
Abb. 4.1: Programmierung mit der Zentralgruppe CPU1516.....	44

Tabellenverzeichnis

Tabelle 1: Hauptdaten der Bewegungsabstände	5
Tabelle 2: Technische Hauptdaten des DC-Motors GR-42×40-24V	7
Tabelle 3: Technische Hauptdaten des Getriebes SG62-72:1	8
Tabelle 4: Technische Hauptdaten des Gebers RE30-3-500 für Montage und Programm.....	9
Tabelle 5: Zeitmessung des Zahnrads pro zehn Umdrehungen	10
Tabelle 6: Hauptdaten der Bewegungsabstände und Bewegungszeit	11
Tabelle 7: Hauptdaten der Bewegungsabstände und Anzahl der Impulse.....	12
Tabelle 8: Technische Hauptdaten von DC-Motor G30.2-24V für Montage und Programm...13	
Tabelle 9: Technische Hauptdaten von Näherungsschalter SME-8-S-KED-2 (FESTO)	14
Tabelle 10: Pin-Belegung zwischen Relais und Anlage	15
Tabelle 11: Auswahl von Simatic-Modulen anhand der Steuerungsarten.....	17
Tabelle 12: Technische Hauptdaten der Stromversorgungsbaugruppe PS307-5A.....	18
Tabelle 13: Bedingung des Motordrehzustands.....	26
Tabelle 14: Symboltabelle bei der Zeitsteuerung von STEP 7	27
Tabelle 15: Wahrheitstabelle der Operation POS.....	29
Tabelle 16: Symboltabelle bei der Signalsteuerung von STEP 7.....	35
Tabelle 17: Vergleich von drei Programmen	42

Literatur- und Quellenverzeichnis

Internetverzeichnis

[01]

Produkte – Dunkermotoren GmbH

<http://www.dunkermotoren.de/produkte/> (16.07.2017)

[02]

SME-8| Festo Deutschland

https://www.festo.com/cat/de_de/products_SM8 (23.07.2017)

[03]

MTRG25_2E_D_A1_03_DE_S

https://www.mattke.de/fileadmin/_migrated/content_uploads/MTRG25_2E_D_A1_03_DE_S.pdf (23.07.2017)

[4]

SPS Programmierung: Speicherprogrammierbare Steuerung SPS lernen

<https://www.sps-lehrgang.de/> (23.07.2017)

Abbildungsquellenverzeichnis

[01]

Abb. 2.9-Abb. 2.12

<http://www.dunkermotoren.de>

[02]

Abb. 2.13

<https://www.landefeld.de/shop/media/festo/produkte/16bp0509.jpg> (23.07.2017)

[03]

Abb. 2.15: 4Q-PWM-Servoregler MTRG-25/2E von mattkeag

https://www.mattke.de/fileadmin/_processed_/csm_MTRG_7405ddab58.gif (16.07.2017)

[04]

Abb. 2.16

https://support.industry.siemens.com/dl/files/imagepool/products/969/1969/v1/P_KT01_XX_01099i.jpg (24.07.2017)

[05]

Abb. 2.17-Abb. 2.19

<https://mall.industry.siemens.com> (24.07.2017)

[06]

Abb. 2.20

https://support.industry.siemens.com/cs/images/products/P_NC01_XX_00351i.jpg
(25.07.2017)

[07]

Abb. 3.22

<https://wenku.baidu.com/view/136cd7c658f5f61fb7366682.html> (04.07.2017)

[08]

Abb. 4.1

<https://www.youtube.com/watch?v=ZRSuV3Zs2P0> (16.07.2017)