



Hochschule Karlsruhe
Technik und Wirtschaft
UNIVERSITY OF APPLIED SCIENCES

Hochschule Karlsruhe – Technik und Wirtschaft
Fakultät für Maschinenbau und Mechatronik

Inbetriebnahme der Maschinensteuerung einer Laserschneidemaschine

Bachelorarbeit (B. Eng.)

von
Alexander Dwarnicak
geb. am 21.09.1987
in Karlsruhe
Matr.-Nr.: 30651

Betreuer der Firma Institute of Materials and Processes
M. Sc. Dirk Löckmann

Betreuer der Hochschule Karlsruhe
Prof. Dr. –Ing. Rüdiger Haas

Karlsruhe, 01. 05. 2013 bis 31. 07. 2013

Erklärung

Ich versichere hiermit wahrheitsgemäß, die Abschlussarbeit selbstständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles einzeln kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde.

Karlsruhe, den 30. Juli 2013

Unterschrift:

Kurzfassung

Inbetriebnahme der Maschinensteuerung einer Laserschneidemaschine

Zur Steuerung der Laserschneidemaschine, im Institut of Materials and Processes der Hochschule Karlsruhe, wird eine neue NC Steuerung benötigt. Die Maschine wurde aus einer alten Turbosenator Fräsmaschine in verschiedenen Projekten und Abschlussarbeiten zu einer Laserschneidemaschine umgerüstet. Durch die verschiedenen Umbauten und die zum Teil sehr alte Hardware kam es immer wieder zu Ausfällen der Maschine. Da für die alte Steuerung weder eine Dokumentation noch geeignete Programmierschnittstelle zu finden ist, wird die Maschine nun mit einer neuen NC Steuerung sowie einer SPS ausgerüstet. Die Wahl fiel dabei auf die Software „TwinCAT“ der Firma Beckhoff, die auf jedem Windows Rechner installiert werden kann. Diese Software bietet neben der eigentlichen NC Steuerung auch eine Software SPS. Als Ein- und Ausgangsbaugruppen dienen Busmodule, die über das Industrial Ethernet „EtherCAT“ mit der Software - SPS verbunden sind.

Zur Achsbewegung können die vorhandenen Indramat Achsregler und Antriebe weiter verwendet werden, was die Umbaukosten reduziert.

Diese Arbeit beschreibt den prinzipiellen Aufbau einer Laserschneidemaschine und vermittelt die Grundbegriffe des Laserschneidens.

Es werden die verwendeten Bussysteme Profibus, EtherCat und Sercos II vorgestellt und deren Aufgabe und Konfiguration in der Maschinensteuerung erklärt.

Desweiteren werden die verschiedenen Komponenten der TwinCAT Software beschrieben. Der Schwerpunkt liegt dabei auf den Modulen TwinCAT PLC und TwinCAT NC I. Die TwinCAT PLC stellt die Software SPS bereit, während das TwinCAT NC I Modul für die NC-Code Interpretation und die Bahninterpolation zuständig ist.

Zur Maschinenbedienung wird eine Benutzeroberfläche gestaltet. Sie folgt der Norm ISO 9241, wodurch ein ergonomisches Bedienkonzept entsteht. Auf dem HMI werden alle relevanten Maschinenparameter sowie Bedienelemente für die Achsbewegungen und die LASER Steuerung angezeigt.

Ein Fehlermeldungssystem wird in die Benutzeroberfläche integriert. Dieses visualisiert Fehlermeldungen, Warnungen und Hinweise. Auftretende Meldungen werden archiviert.

Die Lasersteuerung wird in die Maschinensteuerung integriert. Dies ermöglicht den Laser aus dem NC Programm heraus zu steuern.

Abschließende Schnittversuche zeigen die Produktivitätssteigerung, die durch den Umbau der Steuerung erreicht wird.

Im Ausblick werden weitere Verbesserungen beschrieben, die in dieser Arbeit nicht umgesetzt werden konnten.

Abstract

Implementation of the numerical control for a laser cutting machine

To control the laser cutting machine, which is located in the institute of materials and processes, a new NC control is needed. The machine was converted, in various projects and theses, from an old Turbosenator milling to a laser cutting machine. By the various modifications and some very old hardware, it often results to failures of the machine. As for the old control neither a suitable programming interface nor documentation is yet to find, the machine is now equipped with a new NC control and a PLC. The choice falls on the software „TwinCAT“, which is produced by the company Beckhoff. It can be installed on any Windows computer. This software includes in addition to the actual control a software PLC. To connect the input and output modules to the PLC, an Industrial Ethernet „EtherCAT“ is used.

For axis movement the existing Indramat axis controllers and drives can be used, which reduces the cost of conversion. This thesis describes the basic structure of a laser cutting machine and explains the basic concepts of laser cutting. The bus protocols Profibus, EtherCAT and SERCOS II, used for the different connections, are introduced and explained. Also is their task and configuration in the TwinCAT machine control.

Furthermore the various components of the TwinCAT software are explained. The focus is on the modules TwinCAT PLC and TwinCAT NC I. TwinCAT PLC provides the software PLC, while the TwinCAT NC I module is responsible for interpretation of the NC code and path interpolation.

For machine operation a user interface is designed. It makes use of the standard ISO 9241, which creates an ergonomic operating concept. The HMI displays all relevant machine parameters and controls for axis motion and LASER control. Furthermore, an error message system is included in the dialog. It displays error messages, warnings and instructions on the user interface. Occurring messages are archived.

The laser controller is integrated into the machine control system, which allows controlling the laser out of the NC code. The final section attempts to show the increase in productivity, which is achieved by the conversion of the control.

The outlook describes further improvements, which could not be implemented in this work.

Nomenklatur

Lateinische Formelzeichen

d	mm	Strahldurchmesser
d_0	mm	Taillendurchmesser
f_P	Hz	Pulsfrequenz
t	s	Zeit
A	mm^2	Strahlquerschnittsfläche
E	W/mm^2	Leistungsdichte
F	m/min	Vorschubgeschwindigkeit
P	W	Leistung
P_{Pk}	W	Spitzenleistung
P_H	W	Pulsleistung
P_{av}	W	mittlere Leistung
Q	J	Pulsenergie

Griechische Formelzeichen

Θ	rad	Divergenz des Laserstrahls
τ_H	s	Pulsdauer
λ	nm	Wellenlänge

Besondere Zeichen

(1.1)	Gleichungsnummer, die erste Zahl gibt die Nummer des Kapitels an
[12]	Nummer im Quellenverzeichnis

Dimensionslose Kennzahlen

G	Güte
\ddot{U}	Überlapp

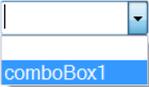
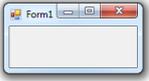
Abkürzungen

Abb.	Abbildung
ADS	Automation Device Specification
AS	Ablauf Sprache
ASCII	American Standard Code for Information Interchange
AWL	Anweisungsliste
CFC	Continuous Function Chart
CNC	Computerized Numerical Control

cw	continuous wave
d.h.	das heißt
DIN	Deutsches Institut für Normung
DMA	Direkt Memory Access
DP	Dezentrale Peripherie
Dr.	Doktor
EN	Europäischen Norm
EVA	Eingabe Verarbeitung Ausgabe
EW	Eingangswort
FB	Funktionsbaustein
FC	Funktion
FMMU	Fieldbus Memory ManagementUnit
FUP	Funktionsplan
GSD	Geräte Stammdatendatei
HDLC	High-Level Data Link Control
HMI	Human Machine Interface
HsKA	Hochschule Karlsruhe – Technik und Wirtschaft
I/O auch IO	Input Output
ID	Identifikator
IDN	Identifications Nummer
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IMP	Institute of Materials and Processes
Ing.	Ingenieur
IP	Internet Protocol
ISO	International Organization for Standardization
KOP	Kontaktplan
LAN	Local Area Network
LASER	Light Amplification by Stimulated Emission of Radiation (Lichtverstärkung durch stimulierte Emission von Strahlung)
LWL	Lichtwellen Leiter
M.Sc.	Master of Science
MDI	Manual Data Input
M-Funktionen	Maschinenfunktionen
MMT	Maschinenbau und Mechatronik
NC	Numerical Control
NC I	Numerical Control Interpolation
Nd YAG	Neodym-dotierter Yttrium Aluminium Granat
PC	Personal Computer
PCI	Peripheral Component Interconnect
PLC	Programmable Logic Controller
Prof.	Professor
PROFIBUS	Process Field Bus
PTP	Point to Point
R-Parameter	Rechen Parameter
RS	Rücksetzen Setzen
SERCOS	Serial Real-time Communication System

SIL	Safety Integrity Level
SPS	Speicherprogrammierbare Steuerung
ST	Strukturierter Text
TCP	Transmission Control Protocol
TFT	Thin Film Transistor
TwinCat	The Windows Control and Automation Technology
UDP	User Datagram Protocol
VDE	Verband der Elektrotechnik
vgl.	vergleiche
XGA	extended Video Graphics Array
z.B.	zum Beispiel

Windows Steuerelemente

label1	Label dienen der Anzeige von statischen Texten.
	Eine textBox kann sowohl zur Ausgabe von Informationen als auch der Eingabe von Text dienen.
	Die comboBox dient der Ein- und Ausgabe. Die Eingabe kann frei oder durch Auswahl eines der gelisteten Elemente erfolgen.
groupBox1	Die groupBox dient zur optischen und funktionalen Gruppierung von Steuerelementen auf der Dialogoberfläche. Dadurch wird die Übersicht erhöht. Innerhalb einer groupBox kann nur ein radioButton aktiv sein.
listBox1	Eine listBox dient zur Anzeige von Elementen. Die Elemente können vom Benutzer ausgewählt werden.
	Die WindowsForm bietet die Grundlage für den Dialog. Auf ihr werden die Steuerelemente platziert. Wird die Form geschlossen so wird das Programm beendet.
<input type="checkbox"/> checkBox1	Eine checkBox dient der Eingabe und Anzeige von booleschen Werten. Damit können Optionen an und abgeschaltet werden.
<input type="radio"/> radioButton1	Ein radioButton dient der Auswahl aus verschiedenen Optionen. Innerhalb einer Gruppierung kann nur ein radioButton aktiv sein.
	Ein Button dient der Eingabe. Durch anklicken wird der „_Click“ Event des Buttons aufgerufen in dem die Funktion hinterlegt wird.

Inhaltsverzeichnis

Nomenklatur	vii
Tabellenverzeichnis	xiii
Abbildungsverzeichnis	xv
1 Aufgabenstellung	1
2 Stand der Technik	3
2.1 Aufbau einer Laseranlage	3
2.2 Bussysteme	7
2.2.1 Profibus	7
2.2.2 Sercos II	7
2.2.3 EtherCAT	8
2.3 SPS Programmiersprachen	10
2.4 TwinCAT	13
2.5 Grundlagen Laserschneiden	18
3 Inbetriebnahme der Maschinensteuerung	21
3.1 Motivation der Arbeit	21
3.2 Komponenten	22
3.3 Namenskonvention	24
3.4 Zusatzfunktionen	27
3.4.1 Maschinenfunktionen	27
3.4.2 R-Parameter	29
3.4.3 Werkzeuge	29
3.4.4 Einzelsatz	30
3.4.5 Testmodus	30
3.4.6 Nullpunktverschiebung	30
3.5 TwinSAFE Programm	31
3.6 SPS - Programmierung	34
3.6.1 Spezielle Funktionsbausteine	35
3.6.2 Sicherheitsfunktionsbausteine	36
3.7 Benutzeroberfläche	38
3.7.1 Bedienkonzept	38
3.7.2 Bedienung der Benutzeroberfläche	40
3.7.3 Aufbau der Benutzeroberfläche	45
3.8 Implementierung des Benutzerdialogs	47
3.9 Kommunikation	50
3.10 Fehlermanagement	54
3.10.1 Fehlerreaktion	54

3.10.2 Fehlermanagement SPS	54
3.10.3 Fehlermanagement HMI	56
3.10.4 WindowsForm „frmFehlermeldungen“	57
3.11 Schnittversuche	59
3.11.1 Versuch 1	
Reduzierung der Hauptzeit	59
3.11.2 Versuch 2	
Verbesserung der Schnittqualität	62
3.11.3 Versuch 3	
Alternative Bearbeitungsstrategie	63
4 Zusammenfassung und Ausblick	65
4.1 Zusammenfassung	65
4.2 Ausblick	66
4.2.1 Vorschau	66
4.2.2 Mustern	66
4.2.3 Flaschenhalserkennung	67
4.2.4 Fehlererkennung im NC-Programm	67
4.2.5 Anpassung des Vorschubs	67
4.2.6 Dynamische Laserleistung bei Kurvenfahrt	67
4.2.7 Satzvorlauf	68
4.2.8 Parameterentwicklung	68
4.2.9 Parameterentwicklungsprogramm	68
Literaturverzeichnis	69
A Anhang	71
A.1 Bildanhang	71
A.1.1 Aufrufbaum	71
A.1.2 Automatik Dialog	73
A.1.3 Handbetrieb Dialog	74
A.1.4 MDI Dialog	75
A.1.5 Nullpunktverschiebung Dialog	76
A.1.6 Lasersteuerung Dialog	77
A.1.7 FehlerMeldungen Dialog	78
A.1.8 HMI_Laser Dialog	79
A.1.9 Oberes Panel	80
A.1.10 Unteres Panel	81

Tabellenverzeichnis

3.1	Maschinenfunktionen	27
3.2	Auswertung von Versuch 1	61
3.3	Auswertung von Versuch 3	64

Abbildungsverzeichnis

2.1	Kavität eines Nd:YAG Lasers [16]	5
2.2	SPS Programm in Anweisungsliste (AWL)	10
2.3	SPS Programm in Strukturiertem Text (ST)	10
2.4	SPS Programm in Funktionsplan (FUP)	11
2.5	SPS Programm in Freigrafischem Funktionsplan (CFC)	11
2.6	SPS Programm in Kontaktplan (KOP)	12
2.7	SPS Programm in Ablaufsprache (AS)	12
2.8	Taskarbeitung bei maximal 50% Systemauslastung	13
3.1	M-Funktionsverarbeitung	28
3.2	Schneidkopf [16] und Werkzeugparameter	29
3.3	Stromstoßschaltung	31
3.4	TwinSAFE Programmierung	33
3.5	Aufbau des SPS Programms	34
3.6	Maschinenbediener in Bedienposition 1	38
3.7	Maschinenbediener in Bedienposition 2	39
3.8	Grundaufbau des Benutzerdialogs	41
3.9	Dialog Nullpunktverschiebung	42
3.10	Icon aus der Windows Umgebung (Ordner öffnen)	43
3.11	Icon nach ISO 55003-3 (Einzelsatz)	43
3.12	Icon im inaktiven Zustand (Referenzfahrt)	43
3.13	Icon im aktiven Zustand (Referenzfahrt)	43
3.14	Dialog der Lasersteuerung	44
3.15	Klassen	48
3.16	Netzstruktur der Laserschneidemaschine	50
3.17	Vorgang zum Starten eines NC-Programms	53
3.18	Fehlerrückmeldung	55
3.19	Gewählte Kontur, blaue Linien: Verfahrensweg in Bearbeitungsvorschub, rote Linie: Positionierbewegungen im Eilgang	59
3.20	Alte Strategie 50mm Rückzug	60
3.21	Neue Strategie mit 5mm Rückzug	61
3.22	Neue Strategie ohne Rückzug	61
3.23	Vergleichsschnitte nach alter Strategie (oben) und Neuer (unten)	62
3.24	Alternative Bearbeitungsstrategie	63

1 Aufgabenstellung

Im Institute of Materials and Processes (IMP) der Hochschule Karlsruhe soll ein Versuchsstand zur Ermittlung von Parametern zum Laserschneiden aufgebaut werden. Zu diesem Zweck wurde die Turbosenator Fräsmaschine umgebaut und mit einem Nd:YAG Laserschneidkopf der Firma Trumpf versehen. Im Rahmen dieser Bachelor Thesis soll nun die Steuerung der Maschine neu aufgebaut und programmiert werden. Dazu gehört die Programmierung der SPS sowie die Einbindung der NC – Steuerung in das SPS – Programm. Über M – Funktionen sollen die Schutztür, Maschinenabsaugung und der Laser angesteuert werden können. Es sollen alle relevanten Betriebsmodi einer Maschinensteuerung implementiert werden. Dazu gehören Handbetrieb, Tippbetrieb, MDI, Automatikbetrieb, Referenzfahrt, Nullpunktverschiebung, laden, bearbeiten und starten eines NC – Programms, Einzelsatz, Trockenfahrt, Laserparametereinstellung, Fehlermeldungssystem und deren Archivierung. Die Funktionen und wichtigen Parameter sollen über eine grafische Benutzeroberfläche aufrufbar und überwachbar sein. Dabei steht ein intuitives und ergonomisches Bedienkonzept im Fokus, welches auch ungeübten Bedienern einen einfachen Zugang ermöglicht. Bei der abschließenden Inbetriebnahme soll die Produktivitätssteigerung bewertet werden.

Im Einzelnen sind die folgenden Punkte zu bearbeiten:

- Entwerfen eines Bedienkonzeptes
- Erstellen einer grafischen Benutzeroberfläche
- Programmierung der oben genannten Funktionen der Maschinensteuerung
- Inbetriebnahme der Anlage
- Bewertung der Produktivitätssteigerung
- Dokumentation und Präsentation der Ergebnisse

2 Stand der Technik

2.1 Aufbau einer Laseranlage

Zum Aufbau einer Laseranlage gehören folgende Komponenten:

- Laserquelle
- Optische Komponenten zur Strahlführung und Fokussierung
- Achsen zur Bewegung des Laserstrahls
- Steuerung für die Ansteuerung der Achsen und der Laserquelle
- Versorgung von Achsen und Laserquelle

Die Laserschneidemaschine stammt aus dem Umbau einer alten 3-Achs-Fräsmaschine. Dabei wurde die Frässpindel durch einen Laserschneidkopf der Firma Trumpf ersetzt. Die Steuerung übernimmt die Bewegung der Achsen und damit auch des Laserschneidkopfes.

Der Laserschneidkopf dient zur Fokussierung des Laserstrahls, der durch das Lichtleitkabel an den Kopf geleitet wird. Durch die Linse wird das Licht gebündelt und erzeugt damit eine höhere Intensität im Brennpunkt. Dies ist wichtig um einen möglichst parallelen und dünnen Laserstrahl zu erhalten. So sind dünne und präzise Schnitte möglich. Vor der Linse befindet sich ein Schutzglas, welches eine Verschmutzung oder Beschädigung der Linse verhindert. Zusätzlich wird vor dem Schutzglas Druckluft eingeleitet, die, zusammen mit dem Laserstrahl, am unteren Ende des Schneidkopfes austritt. Sie erfüllt zwei Funktionen. Zum Einen übernimmt sie die Funktion von Sperrluft und verhindert das Eindringen von Schmutz und Gasen in Richtung der Fokussieroptik. Dies ist wichtig, da bereits eine geringe Verschmutzung des Schutzglases eine signifikante Reduzierung der Strahlleistung zur Folge hat. Zum anderen dient sie dazu das vom Laserstrahl geschmolzene Material abzutragen und aus dem Spalt herauszublasen und so ein erneutes Zusammenschmelzen des Schnittes zu verhindern.

Laser sind in vielen verschiedenen Leistungsklassen und Ausstattungen vorhanden und verfügen daher auch über ein breites Anwendungsspektrum.

- Oberflächenhärten
- Beschichten
- Schneiden
- Gravieren/Beschriften
- Perforieren
- Schweißen
- Hart/Weichlöten
- Sintern

- Rapid Prototyping (Laminated Object Modelling (LOM), Stereolithografie (STL oder SLA), Fuse Deposition Modelling (FDM), Selective Laser Melting (SLM))

Das Laserschneiden findet dabei Anwendung bei Kleinserien oder beim Schneiden von Rohren. Bei größeren Serien ist das Laserschneiden, im Vergleich zum Stanzen, teurer.

Ein weiteres großes Anwendungsgebiet ist die Medizintechnik.

- Augenoperation
- Verödung von Adern
- Hautkrankheiten
- Haarentfernung
- Tattoorentfernung
- Zertrümmern von Nierensteinen

Laser ist das Akronym für Light Amplification by Stimulated Emission of Radiation (Lichtverstärkung durch stimulierte Emission von Strahlung) und beschreibt das Grundprinzip auf dem diese Technologie basiert. Die Einteilung der verschiedenen Laserarten erfolgt nach der Art des stimulierten Materials. Man unterscheidet:

- Festkörperlaser
- Flüssigkeitslaser
- Gaslaser
- Freie-Elektronen-Laser

Bei dem in der Maschine verwendeten Nd:YAG Laser handelt es sich um einen Festkörperlaser. Nd:YAG steht für Neodym-dotierter Yttrium-Aluminium-Granat [18] und beschreibt die Zusammensetzung des Laserstabes. Verbreitete Gaslaser sind CO₂ Laser und He-Ne Laser. Die Kavität ist in [16] beschrieben als die funktionale Einheit, die zum Erzeugen des Laserstrahls dient (vgl. Abb. 2.1). Sie besteht aus Blitzlampen, Reflektoren und dem Laserstab. Die Lampen werden zur Anregung des Laserstabes verwendet. Sie sind parallel zum Laserstab, welcher sich in der Mitte befindet, angeordnet. Die Reflektoren dienen dazu, alles Licht aus den Blitzlampen auf den Laserstab zu lenken. Durch den Energieeintrag der Lampen werden im Inneren des Laserstabes Elektronen auf ein höheres Energieniveau gehoben. Dieser Vorgang wird Pumpen genannt. Fallen die Elektronen zurück auf ihr ursprüngliches Niveau, so wird ein Licht einer bestimmten Wellenlänge abgestrahlt. Damit erhält man ein sehr geringes Wellenlängenspektrum. Um die Intensität des Lichtes zu erhöhen, wird ein Resonator verwendet. Dieser besteht aus zwei Spiegeln, die sich auf beiden Seiten der Kavität befinden und das Licht innerhalb der Kavität hin und her werfen. Damit werden die Strahlen gebündelt. Strahlen, die nicht parallel zum Laserstab verlaufen, treten seitlich aus. Dadurch erhält man den geringen Divergenzwinkel des Laserstrahls. Einer der Resonatorspiegel ist entweder teildurchlässig oder es befindet sich eine Bohrung in dessen Mitte. Dadurch kann ein Teil des in der Kavität erzeugten Lichtes austreten. Dieses Licht kann zur Bearbeitung genutzt werden.

Der Laserstrahl wird mit Hilfe eines Lichtleiters zum Laserschneidekopf geleitet. Dies ermöglicht einen flexiblen Einsatz des Schneidkopfes. Bei CO₂ Lasern ist die Strahlleitung in einem

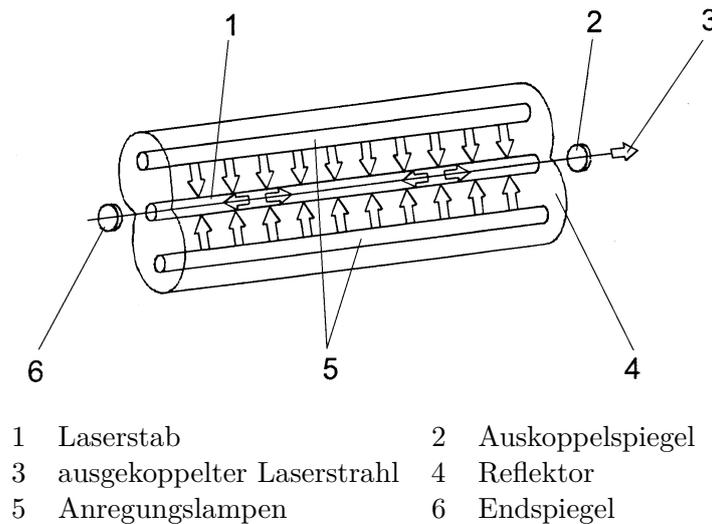


Abbildung 2.1. Kavität eines Nd:YAG Lasers [16]

Lichtleiter aufgrund der Wellenlänge nicht möglich. Hier werden Metallspiegel zur Strahlleitung verwendet.

Die Schneidbarkeit eines Materials hängt stark von der Oberflächenbeschaffenheit und den Adsorptionseigenschaften des Materials ab. Wird der Strahl zu stark reflektiert kann nicht genug Energie in das Material eingebracht werden um dieses zu schmelzen. Oft führt allerdings schon ein geringer Wärmeeintrag in das Material zu einer Änderung der Adsorptionseigenschaften, sodass dieses dann genug Energie aufnimmt, um es schneiden zu können.

Bei Q Switch Lasern wird zunächst die Kavität aufgeladen, bis ein gewünschter Wert erreicht ist. Dann wird eine Blende geöffnet, sodass das gesamte Licht auf einmal austreten kann. Dieses Vorgehen führt zu kurzen, sehr intensiven Impulsen.

Bei cw Lasern (continuous wave) wird dauerhaft Laserlicht erzeugt und an die Fokussieroptik geleitet.

Bei gepulsten Lasern erfolgt das Pumpen mit Hilfe von Blitzlichtlampen. Dadurch entsteht ein gepulstes Laserlicht, mit Pulsen hoher Energiedichte. Allerdings benötigen Blitzlichtlampen eine relativ lange Zeit um sich aufzuladen, wodurch die Pausenzeit im Vergleich zur Pulszeit sehr hoch ist.

Bei den Achsen handelt es sich um drei Kugelgewindespindeln, die entlang der Hauptachsen X, Y und Z angeordnet sind. Die Antriebe sind mit Inkrementalgebern ausgestattet. Damit kann die Positionsregelung erfolgen. Allerdings ist bei dieser Art von Geber, bei Maschinenstart die aktuelle Position unbekannt. Daher muss, nach Einschalten der Maschine zunächst eine Referenzfahrt durchgeführt werden. Die Achsen fahren dabei in eine festgelegte Richtung, bis sie den Referenzschalter erreichen. Die Lage der Referenzschalter ist bekannt, daher können von dort aus die Pulse gezählt und damit die Ist-Position berechnet werden.

Die Positionsregelung der Achsen erfolgt in drei Indramat Achsreglern. Je ein Achsregler ist mit einem Achsantrieb verbunden. Den Achsreglern vorgeschaltet befindet sich der Achsverstärker. Dieser wandelt die 400 V 3 Phasen Wechselspannung in 600V Gleichspannung um und stellt diese den Achsreglern zur Verfügung.

Im Interpreter werden die NC Sätze eingelesen, interpretiert und die Bewegung der Achsen berechnet. Die Bewegung wird, in Form von Positionswerten, zyklisch an die Achsregler übergeben. Dazu wird der Sercos II Bus verwendet. Die Achse nun die Aufgabe innerhalb des Zyklus die Position zu erreichen. Aus den Positionswerten berechnet jeder Achsregler die benötigte Motordrehzahl und steuert den angeschlossenen Motor an. Durch den Abstand der Stützpunkte auf der Kontur wird die Vorschubgeschwindigkeit vorgegeben. Zur Positionsregelung liefert die Achse ihre Position in Form von Inkrementalpulsen zurück. Aus der Anzahl der gefahrenen Pulse kann die aktuelle Position errechnet werden. Der zeitliche Abstand zwischen den Pulsen ergibt die Geschwindigkeit mit der sich die Achse bewegt. Vorschubgeschwindigkeit und Position werden in eine Kaskadenregelung gegeben. Diese Art der Regelung ermöglicht es eine präzise Bahn mit konstantem Vorschub zu fahren, da schon auf einen kleinen Schleppabstand gut geregelt werden kann.

Das zu schneidende Material wird fest auf den Maschinentisch eingespannt, welcher mit dem Grundgestell der Maschine verbunden ist. Der Laserschneidekopf wird dann über das Material bewegt. Die Bearbeitung erfolgt meist im 2,5-Achsen Betrieb. Dies bedeutet, dass die Z-Achse nur für eine Anfahrbewegung verwendet wird. Während des Schneidens wird der Strahl nur horizontal, in der XY-Ebene, bewegt. Dies ermöglicht eine einfachere Programmierung und entlastet den Interpreter und die Bewegungssteuerung.

2.2 Bussysteme

2.2.1 Profibus

Der Profibus (kurz für Process Field Bus) ist ein weit verbreiteter Feldbusstandard in der Automatisierungstechnik. Die Technologie dazu ist in der Norm IEC 61158 (Digital data communication for measurement and control - Fieldbus for use in industrial control systems) beschrieben. Die Profibus Technologie ist in drei Schichten unterteilt. Die unterste Schicht bildet die Bitübertragungsschicht. In ihr ist der physikalische Aufbau der Verbindung beschrieben. Dazu gehören

- Stecker und Kabel
- Netzstruktur
- Signalpegel

Die Datenübertragung erfolgt über eine geschirmte verdrehte Zweidrahtleitung. Damit ist, je nach Übertragungsrate, eine Übertragung über eine Strecke von 100m bis 1200m möglich. Wird ein Lichtleiter zur Übertragung verwendet beträgt die Reichweite bis zu 12km.

Im Stecker integriert befindet sich ein Busabschlusswiderstand. Dieser dient zur Versorgung des Buses. Wird dieser abgeschaltet, kann an dem Stecker der Bus erweitert werden. Mögliche Strukturen sind dabei Ring-, Stern- und Busstrukturen.

Die nächste Schicht bildet die Sicherungsschicht. In ihr ist der Telegrammaufbau und der Kommunikationsablauf geregelt. In jeder Busstruktur befindet sich ein Busmaster, der die Synchronisation aller Busteilnehmer übernimmt. Dies ist im Regelfall die Steuerung der Anlage. Andere Teilnehmer wie Sensoren und Aktoren werden als Slave an den Bus angeschlossen. Der Busmaster fordert über eine feste Adresse des Slaves Daten an oder schreibt in seinen Eingangspuffer. Die Kommunikation wird mit einem Handshake abgeschlossen. Danach folgt eine Sendepause, die verhindern soll dass sich noch Daten auf dem Bus befinden, die sich mit dem nächsten Kommunikationszyklus überschneiden könnten. Danach kann mit dem nächsten Slave kommuniziert werden. Sind mehrere Master an ein Bussystem angeschlossen, so wird das Tokenpasing angewendet um die Kommunikation auf dem Bus zu regeln. Der Master, der den Token besitzt, hat das Recht auf den Bus zu schreiben und Daten von den angeschlossenen Slaves anzufordern. Hat der Master die Kommunikation abgeschlossen gibt er den Token an den nächsten Master ab, der dann wiederum die Kommunikation mit den Slaves durchführen kann. Die Übertragungsrate ist zwischen 9600 baud und bis zu 12 Mbaud einstellbar. Die Daten werden dabei in Datentelegrammen versendet. Diese sollen die sichere Datenübertragung gewährleisten.

Frei nach [2].

2.2.2 Sercos II

Das Sercos interface II ist eine nach IEC 61491 und EN 61491 genormte Kommunikationsschnittstelle zwischen NC Steuerungen und Antrieben. Sie erfüllt dabei nicht nur die technischen Anforderungen, sondern ermöglicht durch das offene und standardisierte Design die Kommunikation zwischen Geräten verschiedener Hersteller. Das Sercos Interface findet Anwendung in allen Gebieten der Automatisierung und NC Bearbeitung in denen schnelle und

präzise Bewegungen notwendig sind. Durch ihn werden in kurzen gleichbleibenden Abständen Positionswerte an die Achsregler ausgegeben. Die Kommunikation erfolgt nach dem Master – Slave Prinzip, wobei der Master immer die numerische Steuerung ist. Der Kommunikationszyklus beginnt mit einer Synchronisationsphase, bei der die Achsen vom Master synchronisiert werden. Dies ist wichtig da dadurch gewährleistet wird, dass alle Bewegungen der Achsen synchron ablaufen. Nach der Synchronisation werden die Daten an die Achsen übermittelt. Nach der Synchronisation werden die neuen Positionsdaten an die Achsen übertragen. Jede Achse hat dabei eine feste Adresse. Nach den Achspositionen werden allgemeine Daten vom Master übertragen. Diese werden von allen Teilnehmern gleichzeitig empfangen. Anschließend hat jeder Slave einen definierten Zeitraum, in dem er Daten an den Master senden kann. In diesem konfigurierbaren Datenfeld können Daten aller Art übertragen werden. Um die Offenheit der Schnittstelle zu gewährleisten wurden vom Sercos Arbeitskreis ID Nummern für einzelne Parameter festgelegt. Daneben ist ein Adressbereich für Parameter vorgesehen, die nicht durch die ID Nummern abgedeckt sind. Die Zykluszeit ist einstellbar und liegt zwischen $65\mu\text{s}$ und 65ms . Die Übertragungsrates liegt dabei zwischen 2Mbit/s und 16Mbit/s . Physikalisch erfolgt die Kommunikation über eine Wellenleiter Ringstruktur. Lichtwellenleiter wurden gewählt um immun gegen Elektromagnetische Störungen in Kabelkanälen zu sein. Außerdem ist die LWL Technologie unempfindlich gegenüber Hochfrequenten Störungen, wie sie oft infolge von Leistungsregelungen auftreten.

Außerdem ist es möglich die Daten in örtlich getrennte Netze zu übertragen, was bei kabelgebundenen Übertragungen aufgrund der unterschiedlich vorherrschenden Spannungspegel nicht möglich ist.

Die Übertragungslänge beträgt in einem Plastik LWL bis zu 50m in einem Glas LWL bis zu 250m . Die maximale Anzahl von Teilnehmern ist auf 254 begrenzt. Kommunikationsfehler werden beim Sercos Interface durch das HDLC Protokoll erkannt. Werden die festen Kommunikationszeiten und Protokollängen genutzt um Fehler zu erkennen. Die Hammingdistanz ist dabei größer 4 . Servicedaten werden durch eine Handshake Prozedur abgesichert. Echtzeitdaten wie Soll- und Istwerte werden durch den nächsten Zyklus korrigiert. Solange arbeitet die Regelung mit dem letzten gültigen Wert. Schlägt die Kommunikation mehr als zwei Mal aufeinanderfolgend fehl, so wird die Anlage stillgelegt. Dies geschieht auch bei einem Kommunikationsabbruch.

Das Sercos II Interface wurde inzwischen durch das Sercos III Interface abgelöst. Die größten Änderungen sind dabei die Nutzung eines LAN Kabels. Dies ermöglicht Datenraten bis zu 100Mbit/s , das Anschließen von weiteren Interface Teilnehmern während des Betriebes, sowie das Anschließen eines PCs zu Diagnosezwecken. Die maximale Teilnehmerzahl wurde auf 511 erhöht. Durch den redundanten Aufbau der Datenleitungen kann der Betrieb trotz Drahtbruch aufrecht erhalten werden. Ein weiterer Vorteil ist, dass es die direkte Kommunikation zwischen Slaves ermöglicht.

Frei nach [3].

2.2.3 EtherCAT

Das EtherCAT basiert auf der IEEE 802.3 Ethernet Technologie. Allerdings bietet es gegenüber dieser Technologie weitere Verbesserungen, die sie für die Automatisierung verwendbar machen. Dazu gehören ein schlankes Übertragungsprotokoll, eine harte Echtzeitübertragung

und geringer Jitter¹. Als Basis für das Übertragungsprotokoll wird ein modifiziertes Ethernetprotokoll verwendet. Dies ermöglicht die Verwendung aller für die Ethernet Verbindung üblichen Hardware, wie Leitungen und Switche sowie die flexible Netztopologie der Ethernettechnologie. Der Vorteil des für die Automatisierung abgeänderten Protokolls liegt in der Reduzierung des Overheads, das ein Standard Ethernetprotokoll wie TCP/IP oder UDP/IP mit sich bringt. Diese Protokolle wurden ausgelegt um große Datenmengen transportieren zu können. In der Prozessautomatisierung kommen jedoch viele einzelne Teilnehmer mit kleinen Datenmengen zum Einsatz. Dies führt zu einer sehr geringen Nutzdatenrate. Statt jeden Teilnehmer einzeln anzusprechen verwendet EtherCAT daher ein umlaufendes Datenprotokoll. Jeder Netzwerkteilnehmer empfängt das Protokoll, liest die für ihn bestimmten Daten aus, schreibt Daten für die Steuerung in das Protokoll und sendet es weiter an den nächsten Teilnehmer. Um die Datenübertragung unabhängig von der Implementierung dieses Vorgangs durch die Verschiedenen Hersteller zu machen und eine Beschleunigung zu erreichen wird eine Fieldbus Memory Management Unit (FMMU) verwendet. Dieser Baustein wird in jeder I/O Baugruppe verwendet. Die FMMU ermöglicht es die Daten „on the fly“ zu lesen und zu schreiben und das Protokoll nur um Nanosekunden zu verzögern. Dazu wird ein Dynamik Memory Access (DMA) Modul verwendet. Dieses Modul hat neben seiner Geschwindigkeit noch den Vorteil, dass es direkt auf den adressierbaren Speicher zugreifen kann ohne, dass die CPU dadurch belastet wird.

Die Echtzeitübertragung ist wichtig, wenn sich in einem verteilten System Aktuatoren (z.B. Servomotoren) gleichzeitig koordiniert bewegen sollen. Für die Synchronisation werden verteilte Uhren (distributed clocks) verwendet. Dies erfolgt nach dem Master – Slave Prinzip. Der erste Teilnehmer im Bus versendet seine Systemzeit in einem Datenprotokoll. Alle anderen Teilnehmer übernehmen diese Zeit. Um die Signallaufzeit zu kompensieren wird eine Messung zu jedem Teilnehmer unternommen. Dadurch wird die Netztopologie bekannt und kann bei der Verteilung der Systemzeit mit berücksichtigt werden. Dadurch erhält man einen Jitter von deutlich unter 1µs und ist somit vertretbar. Durch die verteilten Uhren ist es auch möglich zeitkritische Messwerte zu versenden. Dies findet Anwendung, wenn ein Servomotor geregelt werden soll und die Geschwindigkeit dabei aus der Zeit zwischen zwei Impulsen errechnet wird. Bei einer hohen Abtastrate würde sich bereits ein kleiner Jitter durch große Geschwindigkeitssprünge bemerkbar machen. Daher wird bei diesen Signalen die Systemzeit an die Daten mit angehängt. Dies macht sie unabhängig von Signallaufzeiten und Jitter.

Frei nach [1]; [13].

¹Der Jitter beschreibt den geringen zeitlichen Versatz zweier synchroner Vorgänge

2.3 SPS Programmiersprachen

Die Programmiersprachen richten sich nach der IEC 61131-3. Dabei wird zwischen den textbasierten Sprachen wie Anweisungsliste und Strukturiertem Text und den grafischen Programmiersprachen wie Kontaktplan, Funktionsplan, Freigrafischer Funktionsplan unterschieden. Die Ablaufsprache stellt eine Mischform dar, da sie sowohl grafische als auch textbasierte Elemente enthält. Entnommen aus [14]; [15].

Anweisungsliste (AWL)

Bei der Anweisungsliste enthält jede Zeile des Programms eine Steueranweisung. Diese kann, ähnlich Assembler Code, direkt vom Prozessor interpretiert werden. Eine Steueranweisung besteht aus einem Operator und je nach Operation aus einem oder mehreren Operanden. Die Abarbeitung des Programms erfolgt Zeilenweise.

```

0001 LD      Eingang_1
0002 OR      Eingang_2
0003 AND     Eingang_3
0004 ST      Ausgang_1
0005
0006
0007
0008

```

Abbildung 2.2. SPS Programm in Anweisungsliste (AWL)

Strukturierter Text (ST)

Strukturierter Text ist eine Weiterentwicklung der Anweisungsliste und ähnelt den Hochsprachen wie Basic und C. Neben einfachen Verknüpfungsoperationen und Sprüngen ist es in ST möglich Schleifen und Verzweigungen zu programmieren. Das Programm wird durch diese mächtigen Anweisungen übersichtlicher und damit weniger fehleranfällig. Auch ist die Verwendung von komplexen Datentypen wie Enumerationen, Arrays und Strukturen möglich. Neben dem direkten Variablenzugriff ist es möglich die Variablen über Pointer zu dereferenzieren. Die Abarbeitung des Programms erfolgt zeilenweise.

```

0001
0002 IF (Eingang_1 OR Eingang_2) AND Eingang_3 THEN
0003     Ausgang_1 := TRUE;
0004 ELSE
0005     Ausgang_1 := FALSE;
0006 END_IF
0007
0008

```

Abbildung 2.3. SPS Programm in Strukturiertem Text (ST)

Funktionsplan (FUP)

Beim Funktionsplan werden die einzelnen Operationen durch Symbole dargestellt und miteinander verknüpft. Die Eingänge der Operationen befinden sich auf der linken Seite der Symbole, die Ausgänge auf der Rechten. Die Anordnung der Verknüpfungen erfolgt in Netzwerken. Der Funktionsplan bietet eine übersichtliche Programmierung, gerade wenn es sich um die bitweise Verknüpfung von Ein- und Ausgängen handelt. Die Bearbeitung des Programms erfolgt dabei in den Netzwerken von links nach rechts.

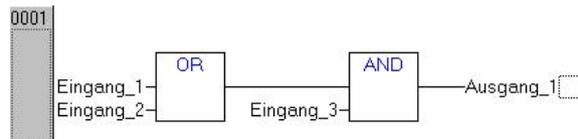


Abbildung 2.4. SPS Programm in Funktionsplan (FUP)

Freigrafischer Funktionsplan (CFC)

Der Freigrafische Funktionsplan ist eine Weiterentwicklung des Funktionsplans. Er arbeitet nicht in einzelnen Netzwerken, sondern mit frei platzierbaren Elementen. Dadurch ist es z.B. möglich Rückkopplungen zu erstellen.

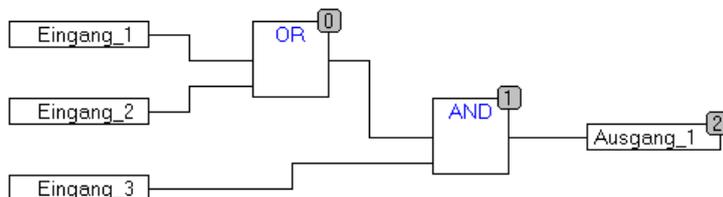


Abbildung 2.5. SPS Programm in Freigrafischem Funktionsplan (CFC)

Kontaktplan (KOP)

Der Kontaktplan ist die grafische Darstellungsart eines Stromlaufplanes. Allerdings ist die Darstellung um 90° gedreht, sodass die Strompfade von links nach rechts verlaufen. Dies hat den Ursprung in der Anfangszeit der Programmierung in der die Strompfade zeilenweise mit ASCII Code dargestellt wurden. Mit KOP ist es möglich einen Stromlaufplan schnell in eine SPS zu übertragen. Bitverknüpfungen erfolgen dabei, ähnlich einer Verdrahtung mit Kabeln, durch die Reihen- bzw. Parallelschaltung der Kontakte. Die Bearbeitung erfolgt in Netzwerken, die von links nach rechts interpretiert werden.

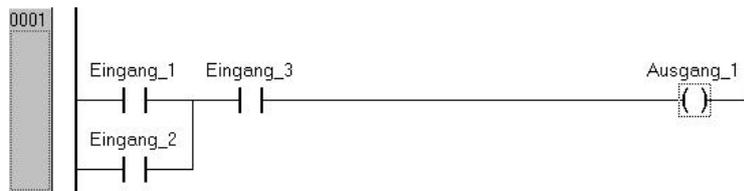


Abbildung 2.6. SPS Programm in Kontaktplan (KOP)

Ablaufsprache (AS)

Die Ablaufsprache stellt eine Mischform aus textbasierten und grafischen Programmiersprachen dar. Bei ihr steht die strukturierte Darstellung von aufeinanderfolgenden Ereignissen im Vordergrund. Die einzelnen Schritte werden in Rechtecken als Black – Box dargestellt. Sie sind über Transitionen und deren Bedingungen miteinander verbunden. Die Programmierung innerhalb des Schrittes wird erst nach dem Öffnen sichtbar. Die Programmiersprache innerhalb des Schrittes ist frei wählbar. Die Abarbeitung des Programms erfolgt Schrittweise von oben nach unten, sobald die nächste Transitionsbedingung erfüllt ist. Es ist allerdings möglich Parallel-, Alternativzweige und Sprünge in den Ablauf einzufügen.

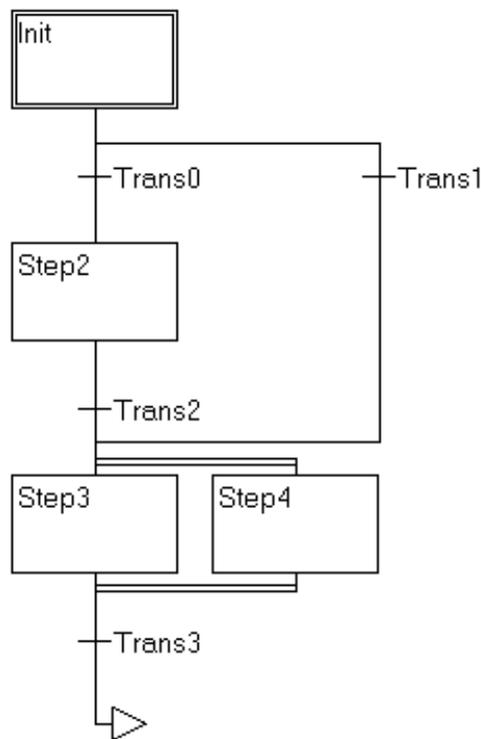


Abbildung 2.7. SPS Programm in Ablaufsprache (AS)

Für diese Arbeit wurde die Programmiersprache Strukturierter Text gewählt. Sie bietet die schnelle und übersichtliche Programmierung von komplexen Programmstrukturen.

2.4 TwinCAT

TwinCAT ist eine Software die von der Firma Beckhoff entwickelt und vertrieben wird. Sie kann auf jedem beliebigen Windows Rechner installiert werden. Eine Intel Architektur ist dabei von Vorteil. Je nach Anwendung können sich zusätzliche Anforderungen an Schnittstellen und Peripheriegeräte des Rechners ergeben. TwinCAT dient als übergeordnete Software der Konfiguration und dem Betrieb von Beckhoff Geräten. Sie beinhaltet den SystemManager über den die angeschlossenen Geräte lokalisiert, verknüpft und konfiguriert werden können. Um die Echtzeitfähigkeit zu gewährleisten, wird ein Echtzeitkernel auf der Ziellplattform installiert. Dieser regelt die Verteilung der Rechenzeit zu deterministisch bestimmbar Zeiten. Damit bekommt jeder Task ein gewisses Zeitfenster in dem er seine Aufgaben abarbeiten kann. Die Verteilung erfolgt dabei nicht, wie in einem Windows System üblich, nach der angeforderten Rechenleistung, sondern zu festgelegten Zeiten. Dadurch ist eine feste Zykluszeit möglich. Kann ein Task innerhalb eines Zeitfensters nicht komplett ausgeführt werden, so wird er durch das Betriebssystem unterbrochen und im nächsten Zeitfenster fortgesetzt. In der Windows Rechenzeit können andere Programme wie der Benutzerdialog ausgeführt werden.

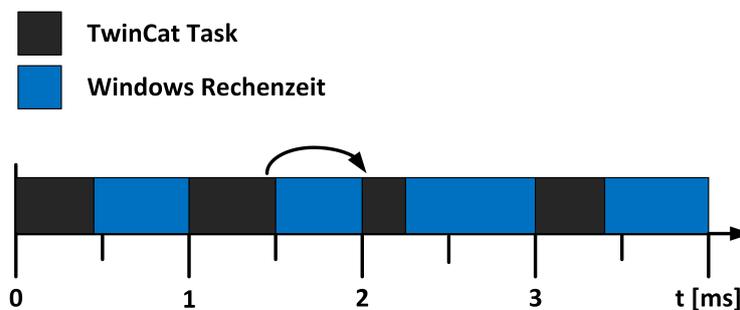


Abbildung 2.8. Taskabarbeitung bei maximal 50% Systemauslastung

TwinCAT stellt auch die grundlegende Kommunikation über verschiedene Bussysteme und Ads bereit. Das System kann durch weitere Softwaremodule erweitert werden. Zur Konfiguration dieser Module dient der TwinCAT SystemManager.

TwinCAT PLC enthält eine Software SPS, die über EtherCAT mit Ein- und Ausgangsbaugruppen kommunizieren kann. Auch ein Meldesystem und eine grafische Benutzeroberfläche sind hier integriert.

TwinCAT PTP bietet eine Punkt zu Punkt Sollwertgenerierung die zur Ansteuerung von ein-dimensionalen Servoachsen verwendet wird.

TwinCAT Camming ermöglicht die Koppelung von Achsen über eine virtuelle Kurvenscheibe.

TwinCAT NC I beinhaltet einen 3D Interpolator für eine 3-Achs Bewegung mit bis zu 5 Hilfsachsen.

TwinCAT CNC ermöglicht eine 3D 5-Achs Interpolation.

Frei nach [14]; [13].

Für diese Arbeit wird TwinCAT mit dem Modulen PLC und NC I verwendet. Daher wird auf diese Module noch einmal genauer eingegangen.

TwinCAT PLC

Diese Software SPS nutzt die Rechnerleistung und Hardwarearchitektur moderner I-PC aus, um eine leistungsstarke SPS mit geringen Zykluszeiten zu erzeugen. Da das System in das Windowssystem integriert ist, werden dadurch Zykluszeiten von unter $15\mu\text{s}$ möglich [14]. Sie ist damit im Vergleich zu einer Hardware SPS ca. vier mal schneller [14]. Desweiteren erleichtert die Rechnerarchitektur die Berechnung von Gleitkommaoperationen, die in einer Hardware SPS sehr viel Zeit in Anspruch nehmen.

Die Kommunikation mit den Ein- und Ausgangsbaugruppen findet über EtherCAT statt. Dies ermöglicht die Peripheriegeräte dicht am Einsatzort zu platzieren und den Verdrahtungsaufwand gering zu halten.

Die SPS kann aber auch ohne diese Baugruppen betrieben werden. Erst durch eine Verknüpfung der SPS Variablen mit den Ein- und Ausgängen der realen Baugruppen im TwinCAT SystemManager werden diese mit einbezogen. Diese Unabhängigkeit ermöglicht es z.B. eine Maschinenbedienung über ein HMI, dessen Eingaben von der SPS verarbeitet werden, zu realisieren ohne, dass dafür reale Tester und Schalter verwendet werden.

Für die Programmierung der SPS steht ein eigenes Softwarepaket PLC Control zur Verfügung.

PLC Control bietet die Entwicklungsumgebung für die SPS Programmierung in TwinCAT. Neben den Editoren für die verschiedenen Sprachen bietet PLC Control auch die Möglichkeit sich im laufenden SPS Betrieb in die SPS einzuloggen, dort den Code zu beobachten und zu debuggen. Funktionen wie Breakpoint, Einzelschritt, das Schreiben und Forcen von Variablen, sowie die Variablenbeobachtung erleichtern das Debuggen.

Frei nach [14]; [13].

In einem neu erstellten Projekt befindet sich zunächst nur ein Programm mit dem Namen „MAIN“. Dieses Standard-Programm wird als erstes von der PLC Task aufgerufen und bearbeitet. Zur besseren Übersicht und Strukturierung des Programms können Funktionsbausteine (FB) und Funktionen (FC) angelegt werden. Jedes Programm, Funktionsbaustein oder Funktion gliedert sich in zwei Abschnitte. In einen Teil werden die Variablen definiert, die innerhalb des Bausteins gültig sind, sowie Eingangs- und Ausgangsvariablen über die der Baustein mit dem übergeordneten Bausteinen kommunizieren kann.

Der andere Teil dient zur Programmierung der Funktion des Bausteins. Bei der Erstellung eines Bausteins kann eine beliebige Sprache gewählt werden, die innerhalb des Bausteins gültig ist. Damit besteht die Möglichkeit für jeden Baustein die Sprache zu wählen, die sie für diesen Programmteil am geeignetsten ist.

Funktionen besitzen immer einen Rückgabewert. Sie dienen der Berechnung wiederkehrender Werte. Eine Funktion kann nur ein Mal instantiiert werden. D.h. ihre Gültigkeit reicht vom Aufruf bis zur Ausgabe des Rückgabewertes. Dies ist maximal ein SPS Zyklus.

Ein Funktionsbaustein kann beliebig oft instantiiert werden. Das bedeutet, dass bei einem Aufruf zunächst überprüft wird ob im aktuellen Gültigkeitsbereich bereits ein FB erzeugt wurde. Ist dies nicht der Fall so wird ein neuer FB im Speicher erzeugt. Existiert der FB bereits im Speicher, so wird dieser weiter verwendet und abgearbeitet. Die Gültigkeit des Bausteins reicht also vom jeweiligen Aufruf des Bausteins bis zum Beenden des SPS Programms. Dadurch ist es möglich in einem FB Funktionen und Abläufe zu programmieren die auch mehr als einen SPS Zyklus andauern. Sie können daher gut für die Strukturierung des Programmes verwendet werden. In dieser Arbeit wurden zum Beispiel jeweils ein FB für Hand-, MDI- und Automatikbetrieb angelegt.

Bibliotheken

Für dieses Projekt werden neben der Standard Bibliothek noch drei weitere Bibliotheken der TwinCAT Software verwendet. Dies sind die Bibliotheken tcMC, tcNCI und TcNcCfg.

In der Standard Bibliothek sind alle grundlegenden Funktionen und Funktionsbaustein hinterlegt wie Timer, Flankenauswertung und Zähler.

tcMC beinhaltet die Bausteine zur Achsbewegung im Handbetrieb, sowie das Stoppen und Freischalten der Achsen.

Die tcNCI Bibliothek dient zur Bedienung des NCI Moduls. Sie beinhaltet Funktionen zur Bedienung des Interpreters und der Interpolationsgruppe.

TcNcCfg dient zur Verwaltung der Interpolationsgruppe.

TwinCAT PLC HMI

[...]Das TwinCAT PLC HMI ist ein System zur Ausführung von Visualisierungen, die mit dem TwinCAT Programmiersystem erstellt wurden. Wenn ein Steuerungsprogramm entsprechende Visualisierungen enthält, werden diese nach dem Start von TwinCAT PLC HMI im Vollbildmodus dargestellt und der Benutzer kann darüber per Mausklick oder Tastatur die im zugrunde liegenden Projekt enthaltenen Steuer- und Überwachungsfunktionen bedienen. Dies ist auch möglich, wenn die TwinCAT PLC Projektdatei mit einem Leseschutz versehen ist. Der Anwender hat keine Möglichkeit, das Steuerungsprogramm zu editieren, Menüs und Funktionsleisten stehen nicht zur Verfügung, es handelt sich um eine reine 'Bedienung' der enthaltenen Visualisierungselemente.

Die wesentlichen Steuer- und Überwachungsfunktionen in einem Projekt müssen also beim Erstellen eines für die Bedienversion vorgesehenen Projektes auf Visualisierungselemente gelegt werden und damit im Online Modus bedient werden. Hierzu gibt es spezielle Eingabemöglichkeiten im Konfigurationsdialog eines Visualisierungselements.

Durch die nahtlose Integration in TwinCAT PLC Control bietet die Visualisierung mit TwinCAT PLC HMI folgende Vorteile:

- Es kann direkt mit den TwinCAT PLC Control Variablen des Steuerungsprogramms gearbeitet werden.
- Die Verwendung von Ausdrücken in der Visualisierungskonfiguration ist möglich
- Ein Platzhalterkonzept ermöglicht objektorientiertes Arbeiten.
- Die TwinCAT PLC Control Funktionen Trace und Rezepturen lesen/schreiben stehen auch in TwinCAT PLC HMI zur Verfügung

[...] [14]

Traceaufzeichnung

[...]Traceaufzeichnung bedeutet, dass der Werteverlauf von Variablen über einen bestimmten Zeitraum hin aufgezeichnet wird. Diese Werte werden in einen Ringspeicher geschrieben (Tracebuffer). Ist der Speicher voll, so werden die „ältesten“ Werte vom Speicheranfang her wieder überschrieben.

Maximal können 20 Variablen gleichzeitig aufgezeichnet werden. Da die Größe des Tracebuffers in der Steuerung einen fixen Wert besitzt, können bei sehr vielen oder sehr breiten Variablen (DWORD) weniger Werte aufgezeichnet werden. [...] [14]

TwinCAT Eventlogger

[...]Der TwinCAT Eventlogger ist ein Alarm- und Diagnosesystem für TwinCAT-basierte Steuerungen. Er hat die Aufgabe, alle auftretenden Events im TwinCAT-System zu verwalten, weiterzuleiten und in die TwinCAT-Logdatei zu schreiben. Events sind in diesem Zusammenhang Alarme, Warnungen, Hinweise oder Anweisungen. Die Quittierung einer Meldung ist möglich. Mithilfe eines Message-Formatters wird die Verbindung zwischen dem eigentlichen Event und dessen Meldetext hergestellt. Diese ist in einer externen Datenbank hinterlegt.[...] [14]

TwinCAT NC I

NC I steht für „Numerical Control Interpolation“ und stellt alle Funktionalitäten für die 3D Bahnsteuerung im TwinCAT System bereit. Es beinhaltet einen Interpreter, Sollwertgenerator und Lageregler, welche zusammen die Bewegung der Achsen steuern. NC I ermöglicht auch die Bildung von Interpolationsgruppen sowie die Zuschaltung und Synchronisierung von bis zu 5 Hilfsachsen. Desweiteren befinden sich im NC I Modul weitere Funktionalitäten, die in Werkzeugmaschinen Anwendung finden. Dazu gehören

- Nullpunktverschiebung
- M – Funktionen
- R – Parameter
- Werkzeuge

Das Interpreterelement basiert auf DIN 66025 und interpretiert Satzweise den NC Code. Aus den Positionsangaben im Code interpoliert er eine Bahn, welche die Punkte miteinander verbindet. Diese Bahn wird an den Sollwertgenerator übergeben.

Der Sollwertgenerator zerlegt die Bahn in Teilstücke und gibt diese, über den Sercos Bus, zu festen Zykluszeiten an die Achsregler weiter. Diese haben nun einen Zyklus Zeit die übermittelte Position zu erreichen. Der Abstand zweier Stützpunkte ergibt also die Geschwindigkeit mit der sich die Achse bewegen soll.

Der Lageregler schreitet ein, wenn die Achse die geforderte Position nicht innerhalb des Zyklus erreichen kann und sich dadurch ein Schleppabstand bildet. Um den Schleppabstand zu verkleinern wird er der Achse im nächsten Zyklus einen Offset auf den Stützpunkt legen, sodass die Achse schneller fährt.

Frei nach [14].

TwinCAT SystemManager

Der TwinCAT Systemmanager ist das zentrale Modul der TwinCAT Software. In ihm werden alle Module verwaltet. Jedes Modul bekommt eine Task zugewiesen, die vom SystemManager im Windows Betriebssystem angemeldet werden. Durch die Echtzeit Kernelerweiterung bekommt dadurch jeder Task, in gleichbleibenden Abständen, eine gewisse Rechenzeit zugewiesen. Reicht dieser Timeslot nicht aus so wird die Task unterbrochen. Die Ausführung wird in der freien Windows Rechenzeit fortgesetzt. Die Abarbeitung der Tasks geschieht normalerweise nach dem EVA-Prinzip. Befindet sich eine Task mit ihrer Rechenzeit an der Grenze ihres Timeslots kann dies zu stark unterschiedlichen Zeitpunkten der Ausgabe führen. Teilweise wird die Task die Bearbeitung im Timeslot abschließen können. Zu einem anderen Teil wird sie jedoch durch eine andere Task oder das Betriebssystem unterbrochen. Um diese Schwankungen zu minimieren gibt es die Option OI am Taskanfang. Dadurch wird das EVA Prinzip nicht mehr angewendet. Stattdessen werden am Taskanfang die Eingänge eingelesen, und zunächst nur die Verarbeitung der Daten vorgenommen, die für die Ausgabe relevant sind. Nach der Ausgabe werden allgemeine Berechnungen der Task ausgeführt. Eine weitere Aufgabe des SystemManagers ist die Verwaltung der Schnittstellen. Die Schnittstellen wie Profibus, EtherCAT und andere Bussysteme werden mit ihren entsprechenden Übertragungsmedien im SystemManager angemeldet. Der SystemManager kann dann automatisch nach angeschlossenen Geräten suchen. Angeschlossene Geräte wie EtherCAT Klemmen werden dann im SystemManager angezeigt und können konfiguriert werden. Auch die Konfiguration der Tasks ist im System Manager möglich. Bei der NC I Task können Antriebe angelegt und konfiguriert werden. Hier kann auch ein Interpreterelement angefügt werden, welches die Berechnung der Bahn übernimmt. Um den Interpreter mit den Antrieben zu verbinden wird, im Interpreter eine Interpolationsgruppe angelegt. In dieser Gruppe können die Antriebe den Achsen zugeordnet werden.

Frei nach [14].

Verknüpfung von Variablen

Die Verknüpfung von Variablen dient dazu, die einzelnen Tasks miteinander zu verbinden. Dazu können Ausgänge einer Task oder Kommunikationsschnittstelle mit den Eingangsvariablen einer anderen verknüpft werden. Die Variablen müssen dabei vom gleichen Typ sein. Bei der Laserschneidemaschinensteuerung werden hier die Ein- und Ausgänge der EtherCAT Baugruppen mit den Ein- und Ausgangsvariablen der SPS und die NC I Task mit den Sercos Achsen verknüpft.

Konfiguration der Tasks

In der Taskkonfiguration können grundlegende Eigenschaften wie Taskpriorität, Zykluszeit und Ads Port festgelegt werden. Dazu kommen Konfigurationen die für die jeweiligen Tasks charakteristisch sind. In der PLC Task wird hier der Projektpfad des SPS Programms angegeben und das Verhalten bei Programmstart eingestellt. In der NC I Task werden Antriebe definiert und Konfiguriert und ein NC Kanal für die Interpolation angelegt.

2.5 Grundlagen Laserschneiden

Frei nach [16]. Der Laserstrahl kann nach verschiedenen Kriterien beurteilt werden. Dadurch kann die Eignung des Laserstrahls für verschiedene Verwendungszwecke geprüft werden. Die Kriterien lassen sich den zwei Kategorien Strahlqualität und Leistung zuordnen. Die Strahlqualität wird durch verschiedene Faktoren bestimmt, die hauptsächlich durch den technischen Aufbau der Stahlleitung und Fokussierung bestimmt werden. Der Strahldurchmesser (d) gibt den Durchmesser des Laserstrahls an. Er muss nicht über die gesamte Strecke der Laserleitung konstant sein. Der Taillendurchmesser (d_0) gibt den kleinst möglichen Strahldurchmesser im Brennpunkt hinter der Fokussieroptik an. Je kleiner dieser Wert desto höher die Leistungsdichte im Brennpunkt. Die Divergenz (Θ) beschreibt den Winkel mit dem der Strahl von der Parallelität abweicht. Das Strahlparameterprodukt

$$\frac{1}{4} \cdot \Theta \cdot d_0 \quad (2.1)$$

ist ein Maß für die Strahlqualität. Je kleiner das Strahlparameterprodukt desto besser lässt sich der Laserstrahl fokussieren, wodurch die Strahlqualität steigt. Das Strahlparameterprodukt bleibt während der gesamten Ausbreitung des Laserstrahls konstant und ist charakteristisch für den Laser. Die Werte für das Strahlparameterprodukt wird in der Maßeinheit [mm·mrad] angegeben. Durch diesen Wert werden die minimale Strukturgröße und die möglichen Materialstärken im Wesentlichen festgelegt. Die Strahlquerschnittsfläche (A) beschreibt die Fläche, die vom Laserstrahl senkrecht durchquert wird. Die Fläche muss nicht über die gesamte Strecke der Laserleitung konstant sein. Im Fokus besitzt sie den kleinsten Wert. Die Leistungsdichte (E) ist ein Maß für die Leistung bezogen auf eine bestimmte Fläche.

$$E = \frac{P}{A} \quad (2.2)$$

Dabei ist P die Leistung, die auf eine bestimmte Fläche A trifft. Beim Laserschneiden wird eine sehr hohe Leistungsdichte benötigt. Dies geschieht zum einen durch die Erzeugung einer hohen Laserleistung zum anderen durch eine möglichst geringe Querschnittsfläche im Fokus. Die Wellenlänge (λ) bezeichnet den Abstand zweier benachbarter Maxima. Das Laserlicht eines Nd:YAG Lasers hat eine Wellenlänge von 1064nm. Die Pulsfrequenz (f_P) gibt die Anzahl der Laserpulse pro Sekunde an. Die Pulsdauer (τ_H) gibt die Länge eines Laserpulses an. Bei einfachen Rechteckimpulsen ist dies die Zeit zwischen den 50% – Werten der steigenden und fallenden Flanke. Die Laserleistung ist ein wichtiger Parameter des Lasers. Man unterscheidet

- Spitzenleistung,
- Pulsleistung und
- mittlere Leistung.

Die Spitzenleistung (P_{Pk}) ist der höchste Leistungswert, der während eines Laserpulses auftritt. Die Pulsleistung (P_H) ist die Leistung eines einzelnen Laserpulses. Die Pulsleistung beträgt meist mehrere Kilowatt. Die mittlere Leistung (P_{av}) gibt den Mittelwert der abgegebenen Leistung über einen längeren Zeitraum an. Trotz der hohen Pulsleistung ist die mittlere

Leistung mit maximal 200 W eher gering. Die mittlere Leistung ergibt sich aus Pulsleistung und Pulsdauer wie folgt.

$$P_{av} = P_H \cdot \tau_H \cdot f_P \quad (2.3)$$

Die Pulsenergie (Q) gibt die Energiemenge an, die während eines Laserpulses abgegeben wird.

$$Q = \int_0^{\tau_H} P_H dt \quad (2.4)$$

In der grafischen Darstellung eines Laserpulses entspricht die Pulsenergie der Fläche unter dem Puls. Die Leistungsdaten des Laserstrahls werden so gewählt, dass bei jedem Laserpuls ein Loch in das zu schneidende Material geschnitten wird. Durch die Aneinanderreihung vieler dieser Löcher entsteht letztendlich ein durchgängiger Schnitt. Die Energie, die für ein durchgängiges Loch benötigt wird, hängt hauptsächlich vom Adsorptionsverhalten des Materials und der Materialstärke ab. Weitere Faktoren, die eine Rolle spielen können, sind Wärmeleitfähigkeit und Konvektion. Das geschmolzene Material im Schnittspalt muss so lange flüssig bleiben, dass es vom Druckluftstrahl aus dem Spalt gedrückt wird. Einstellbare Parameter für einen Laserpuls sind Pulsleistung (P_H), Pulsdauer (τ_H) und Pulsfrequenz (f_P). Im Programm WinLas kann auch die Pulsform, also eine Pulsleistungskurve, erstellt werden. Die maximale mittlere Pulsleistung liegt bei 200 W, die maximale Pulsleistung bei 8000 W. Weitere Faktoren, die Einfluss auf den Schnitt haben, sind Fokuslage und Vorschub der Achsen. Der Vorschub der Achsen kann aus der Pulsfrequenz und dem Lochdurchmesser errechnet werden. Dazu wird ein Überlapp (\ddot{U}) festgelegt. Dieser Überlapp definiert die prozentuale Überschneidung zweier aneinandergrenzender Löcher. Der Überlapp hat Einfluss auf die Schnittkante des Schnittes. Ist er zu gering, so bilden sich Spitzen aus, die in den Schnitt hinein ragen. Diese Spitzen schmelzen jedoch, bedingt durch ihre geringe Abmessungen und der schlechten Wärmeleitung, zu einem großen Teil ab. Dennoch verkleinern sie den Schneidspalt und führen zu einem unsauberen Schnittbild. Hier kann ein Grenzwert festgelegt und durch Versuche den zugehörigen Überlapp ermittelt werden. Ist der Überlapp zu groß so wird der Schnittspalt durch den erhöhten Energieeintrag breiter. Dies hat insofern Auswirkungen auf das Schneidergebnis, dass feine Konturen ausglühen können. Außerdem reduziert sich dadurch der Vorschub. Es kann also eine Formel für den Vorschub (F) entwickelt werden, welche sich aus dem Durchmesser eines Loches (d), der Überlappung (\ddot{U}) und der Pulsfrequenz (f_P) zusammensetzt.

$$F = d \cdot \ddot{U} \cdot f_P \quad (2.5)$$

Bezieht man noch eine Güte (G) in die Berechnung mit ein, die angibt, wie weit die zwischen den Löchern entstehenden Spitzen in den Schnittspalt hineinragen dürfen so wird die Überlappung zu einer Funktion dieser Güte.

$$F = d \cdot \ddot{U}(G) \cdot f_P \quad (2.6)$$

Dies ermöglicht es z.B. einen Schnitt mit geringerer Güte, zugunsten einer höheren Schnittgeschwindigkeit, auszuführen. Oder im Gegenzug eine höhere Schnittqualität mit einem geringeren Vorschub und einer damit einhergehenden längeren Programmlaufzeit zu erreichen.

3 Inbetriebnahme der Maschinensteuerung

3.1 Motivation der Arbeit

Die ehemalige Turbosenator Fräsmaschine stammt aus einem Altbestand und wurde dem IMP überlassen. Da die 3-Achs-Frästechnologie inzwischen weitestgehend durch die 5-Achs Bearbeitung abgelöst ist und das IMP über mehrere dieser Maschinen verfügt, wurde der Umbau der Maschine zu einer Laserschneidemaschine beschlossen. Zwar konnte die Spindel durch einen Laserschneidkopf ersetzt werden, jedoch steht für die alte SPS und Steuerung keine Software mehr zur Verfügung um die Lasersteuerung in die Maschinensteuerung zu integrieren. Zudem erzeugt die fehlende Spindel oftmals Fehler, wodurch sich die Steuerung nicht einschalten lässt. Die Maschine ist somit nicht produktiv oder für den Lehrbetrieb einsetzbar. Eine Neukonzeptionierung und Aufbau der Steuerung ist damit unumgänglich.

Es wird nach einer kostengünstigen Lösung gesucht, bei der möglichst viele der Komponenten weiterverwendet werden können. Die Entscheidung fällt zugunsten der Beckhoff Software TwinCAT, welche sowohl eine PLC als auch die NC Steuerung enthält. Sie ist auf jedem Windows PC einsetzbar. Durch verschiedene Busmodule kann sowohl eine Sercos als auch eine Profibusverbindung hergestellt werden. Der Sercos Bus dient der Kommunikation mit den Achsreglern, die dadurch beibehalten werden können. Der Profibus ermöglicht die Einbindung der Lasersteuerung. Außerdem kann über den PC auch eine Benutzeroberfläche realisiert werden. Damit wird die Leistungsfähigkeit des PCs optimal ausgenutzt.

Die flexible Anbindung der Peripheriegeräte reduziert den Verdrahtungsaufwand und spart Kosten, besonders in größeren Anlagen mit verteilten Systemen. Der Konfigurationsaufwand bleibt überschaubar was das System zu einer kostengünstigen Alternative zu herkömmlichen Steuerungen macht. Durch den modularen Aufbau ist es möglich die Steuerung exakt an ihre Aufgabe anzupassen.

In dieser Arbeit soll die Realisierung eines solchen Projekts am Beispiel des Umbaus der Laserschneidemaschine veranschaulicht werden.

Beschrieben wird dabei das prinzipielle Vorgehen bei der Programmierung der SPS und der Parametrisierung der NCI. Außerdem erlaubt sie einen Überblick über die zugrundeliegenden Normen und Maschinenrichtlinien, die für die Umsetzung wichtig sind.

3.2 Komponenten

Die folgende Auflistung soll einen Überblick auf die verwendeten Komponenten bieten. Es wird sich dabei auf die wesentlichen Komponenten beschränkt.

Achsregler und Achsantriebe

3x Indramat Achsregler DDS2.1-W025-DS01-00
2x Indramat Digitaler AC-Servomotor MDD 065 C-N-060-N2L-095GB0 für X- und Y-Achse
Indramat Digitaler AC-Servomotor MDD 065 C-N-060-N2L-095GB1 für Z-Achse (gebremst)

Laserquelle

Trumpf HL204-P

Steuerung

Hematec MPC 15 Industrie Panel PC:
Intel Core2Duo T5450 (1.6GHz)
1024MB RAM
Beckhoff TwinCat mit den Modulen PLC und NCI
Beckhoff Servos II Interface PCI Steckkarte FC7501

Beckhoff EtherCat-Module

EK 1100 EtherCat-Koppler
EL6900 TwinSAFE PLC
2x EL1904 TwinSAFE 4-Kanal Safety Eingang
EL2904 TwinSAFE 4-Kanal Safety-Ausgang
2x EL1809 16-Kanal digital Eingang
EL3064 4-Kanal analog Eingang 0-10V
EL4011 1-Kanal analog Ausgang 0-20mA
EL2809 16-Kanal digital Ausgang

Oberfläche

Eingabe

18x Eaton RMQ-16 Q18D Taster mit Lasergravierten Symbolen
Eaton RMQ-16 Q18S1R Schlüsselschalter
Eaton M22-R47K Drehpotentiometer
GFT-105 Folientastatur mit Touch-Matrix
15" XGA TFT Touch-Display (1024x768) (Verbaut in Panel-PC MPC 15)

Ausgabe

15"XGA TFT Touch-Display (1024x768) (Verbaut in Panel-PC MPC 15)

Samsung 19"TFT Bildschirm (1024x768)

Software

Beckhoff TwinCat v2.11 mit den Modulen PLC und NCI

Microsoft Visual Studio 2008

Trumpf WinLas

Microsoft Windows XP SP3

Beckhoff TeNcEdit

3.3 Namenskonvention

Die Namenskonvention, welche Anwendung bei der Vergabe von Variablen- und Objektnamen findet dient der besseren Lesbarkeit des Codes. Sie ist so ausgerichtet, dass aus dem Namen des Objektes auf den Typ und die Verwendung geschlossen werden kann. Dadurch wird die Erstellung und Wartung des Codes erleichtert.

Die Konvention dieser Arbeit richtet sich weitestgehend an der in TwinCAT verwendeten Konvention, da somit ein durchgängiges System erreicht wird. Außerdem ist ein Verständnis des Programms für Kenner des TwinCAT Systems sehr schnell möglich.

Allgemeine Konventionen

Funktionsbausteine erhalten den Präfix „fb“.

Variablen erhalten einen Präfix, der ihren Typ wiedergibt z.B. „b“ für Boolvariablen und „i“ für Integer.

Eingänge erhalten neben den Präfix „in_“ und dem Typ noch die Art der Quelle sowie deren Funktion z.B. in_bTaster_NcStart. Ausgänge erhalten den Präfix „out_“ und dem Typ die Art des Stellglieds das sie Schalten z.B. out_bRelais_Absaugung.

Bei Variablen und Strukturen, welche die Kommunikation zwischen zwei TwinCAT Modulen übernehmen, setzt sich der Name aus der Richtung des Datenstroms, aus SPS Sicht, dem Datentyp und den beiden Modulen zusammen. Wobei die beiden Module mit dem Wort „to“ verbunden sind, sodass auch hier noch einmal die Richtung mit angegeben wird. z.B. in_stNcToPlc.

Variablen, die global deklariert, jedoch nicht mit einem Eingang oder Ausgang verknüpft sind, werden als Merker deklariert und erhalten den Präfix „m_“ gefolgt von Typ und Name.
m_fJogAbstand

Variablen vom Typ Bool, die zum Anzeigen eines Zustandes dienen erhalten den Suffix „_Flag“.
m_bEinzelsatz_Flag

Konvention bei Funktionsbausteinen

Funktionsbausteine, die einen Ablauf steuern, besitzen einen durchgängigen Aufbau. In der Variablendeklaration ist eine Eingangsvariable bExec angelegt. Mit ihr wird der Ablauf gestartet. In den Ausgangsvariablen wird mit bBusy angezeigt, ob das Programm innerhalb des Bausteins gerade abgearbeitet wird. Die Ausgangsvariable bFehler gibt an, ob bei der Bearbeitung des Bausteins ein Fehler aufgetreten ist.

Dadurch kann eine durchgängige Methode angewandt werden, um einen Funktionsbaustein aufzurufen und dessen Ausführung zu überwachen. Im Aufruf wird der Eingang bExec auf „true“ gesetzt. Dadurch wird der Ablauf des FBs gestartet. Der FB setzt seinen Ausgang bBusy auf „true“.

Bei jedem Programmzyklus wird nun im aufrufenden Baustein geprüft, ob die Bearbeitung abgeschlossen ist, während die Abarbeitung im aufgerufenen Baustein weiter voranschreitet. Ist die Bearbeitung im aufgerufenen Baustein abgeschlossen, so wird der Ausgang bBusy auf „false“ gesetzt. Dies wird vom aufrufenden Baustein erkannt. Der Baustein wird nun erneut aufgerufen. Dabei wird der Eingang bExec auf „false“ gesetzt. Dies stoppt die Ausführung im aufgerufenen FB und initialisiert diesen.

Tritt während der Bearbeitung im aufgerufenen FB ein Fehler auf, so wird der Ausgang bBusy des FBs auf „false“ gesetzt. Gleichzeitig wird der Ausgang „bFehler“ auf „true“ gesetzt. Daher erfolgt jedes Mal, wenn der „bBusy“ Ausgang des aufgerufenen FBs auf „false“ gesetzt wurde, eine Überprüfung ob bei der Bearbeitung ein Fehler aufgetreten ist.

```

1  (* Abarbeitung des Funktionsbausteins starten *)
2  fbNcProgramm(bExec := TRUE);
3
4  (* Prüfen ob Bearbeitung abgeschlossen ist *)
5  IF NOT fbNcProgramm.bBusy THEN
6      (* Abarbeitung stoppen *)
7      fbNcProgramm(bExec := FALSE);
8
9      (* Prüfen ob ein Fehler aufgetreten ist *)
10     IF fbNcProgramm.bErr THEN
11         (* Fehlerbehandlung *)
12     END_IF
13 END_IF

```

Auch der innere Aufbau der Funktionsbausteine ist gleich. Zur Steuerung des Ablaufes wird eine „Case Of“ Anweisung mit dem Zähler „iState“ verwendet. Im ersten Schritt (Schritt 0) wird überprüft ob der Eingang „bExec“ gesetzt ist. Ist dies der Fall, so beginnt die Abarbeitung des Programms. Der Ausgang „bBusy“ wird auf „true“ gesetzt, der Ausgang „bFehler“ auf „false“. Danach wird der Zähler auf den Wert 10 gesetzt. Damit werden beim nächsten Programmzyklus die Anweisungen bearbeitet, welche sich in Case 10 befinden. Der große Abstand zwischen den Cases wird gewählt um im Nachhinein einfacher erweiternde Schritte dazwischen einfügen zu können.

Tritt ein Fehler auf, so wird der Fehler gemeldet indem der Ausgang „bFehler“ auf „true“ gesetzt und in Schritt 9999 gesprungen wird. Das Fehlermanagement wird in Kapitel 3.10 beschrieben.

Der Schritt 9999 ist immer der letzte Schritt in der Kette. In ihm werden alle intern aufgerufenen Funktionen und Bausteine gestoppt und der Ausgang „bBusy“ auf „false“ gesetzt.

Wird bExec auf „false“ gesetzt so wird iState auf 0 gesetzt. Die Abarbeitung kann dadurch zurückgesetzt und neu gestartet werden.

```

1  CASE iState OF
2  0:
3      (* Prüfen ob Abarbeitung gestartet werden soll *)
4      IF bExec THEN
5          bBusy := TRUE;
6          iState := 10;
7      END_IF
8  10:
9      (* Beispielschritt aus dem Funktionsbaustein fbLaser_AUS *)
10
11     (* Programmstart zurücksetzen *)
12     out_uiPlcToLaser0.11 := FALSE;
13
14     (* Prüfen ob Fehlerbits des Lasers gesetzt sind *)
15     IF in_uiLaserToPlc0.5 OR in_uiLaserToPlc0.4 THEN
16         (* ein Fehler ist aufgetreten *)
17
18         (* globales Fehlerbit setzten *)
19         m_bFehler := TRUE;
20         (* Fehlernummer in Puffer schreiben *)
21         udiErrorID := in_uiLaserToPlc4;
22         (* in letzten Schritt springen *)
23         iState := 9999;
24     END_IF
25
26     (* Prüfen ob Rücksetzten erfolgt ist *)

```

```
27     IF NOT in_uiLaserToPlc0.12 THEN
28         (* Schritt erfolgreich ausgeführt *)
29         iState := 20;
30     END_IF
31 20:
32     (* Anweisungen in Schritt 20 *)
33     ...
34
35 9999:
36     (* signalisiert dem aufrufenden Baustein,
37     dass die Abarbeitung abgeschlossen ist *)
38     bBusy := FALSE;
39
40 END_CASE
41
42 (* Rücksetzen der Schrittkette *)
43 IF NOT bExec THEN
44     iState := 0;
45 END_IF
```

3.4 Zusatzfunktionen

3.4.1 Maschinenfunktionen

Maschinenfunktionen (M-Funktionen) dienen dazu, der Maschine während der Ausführung des NC-Programms Steuerbefehle zu geben. Die Normung erfolgt in DIN66025-2 [5]. In Klasse 4 werden die M-Funktionen für Maschinen zum Brenn-, Plasma-, Laser-, Wasserstrahl-Schneiden, Drahterodieren festgelegt. Die folgende Tabelle 3.1 zeigt die relevanten Funktionen für das Laserschneiden, sowie weitere Maschinenfunktionen, die bei der Laserschneidemaschine umgesetzt werden. Es wird zwischen schnellen und Handshake Funktionen unterschieden. In

Tabelle 3.1. Maschinenfunktionen

M-Numemr	Funktion nach DIN66025-2 Klasse 4	Funktion Laserschneidemaschine
M03	Schneiden Aus	Schneiden Aus
M04	Schneiden Ein	Schneiden Ein
M06	–	Absaugung Ein
M07	–	Absaugung Aus
M08	–	Sperrluft Ein
M09	–	Sperrluft Aus
M11	–	Schutztür Schließen
M12	–	Schutztür Öffnen
M30	Programmende mit Rücksetzen	Programmende mit Rücksetzen

beiden Fällen sendet der Interpreter ein Signal und die Nummer der M-Funktion an die SPS, welche dann die Ausführung des Befehls und die Ansteuerung der Komponenten übernimmt. Bei der schnellen M-Funktion wird die Abarbeitung des NC Programms direkt nach aussenden des Signals fortgesetzt. Bei der Handshake Funktion wartet der Interpreter auf ein Signal der SPS das diese die Umsetzung der M-Funktion abgeschlossen hat, bevor er die Abarbeitung des NC Codes fortsetzt.

Außerdem findet noch eine Unterscheidung in before Move, after Move und dem Zusatz Auto-Reset statt. Before und after Move spielt eine Rolle, wenn sich im gleichen Satz wie dem der M-Funktion noch ein Bewegungssatz befindet. Bei einer before Move M-Funktion wird zuerst die M-Funktion ausgeführt und dann erst die Bewegung. Bei der after Move Funktion wird zuerst die Bewegung ausgeführt. M-Funktionen mit dem Zusatz AutoReset werden am Ende des Satzes automatisch zurückgesetzt. Sie sind also nur für einen Satz gültig.

Die Umsetzung der M-Funktionen erfolgt in der SPS. Erkennt der Interpreter eine M-Funktion im NC Programm so setzt er eine Flag. Diese Flag wird von der SPS zyklisch überwacht. Ist sie gesetzt wird über einen Funktionsbaustein die Nummer der M-Funktion ausgelesen, die an den Funktionsbaustein „fbM_Funktionen“ weitergegeben wird. Über eine CASE OF Anweisung wird dann die zugehörige Aktion zugeordnet und ausgeführt.

Alle M-Funktionen sind als before Move vom Typ Handshake ausgeführt. Dadurch werden sicherheitsrelevante Funktionen wie die Steuerung der Schutztür und die Lasersteuerung sicher ausgeführt, bevor die Verarbeitung fortgesetzt wird. Bei der Steuerung der Sperrluft und der Absaugung könnten auch schnelle M-Funktionen verwendet werden. Hier wurden die Handshake Funktionen gewählt um den Programmieraufwand gering zu halten. In Abb. 3.1 ist der schematische Ablauf der M-Funktionsverarbeitung dargestellt.

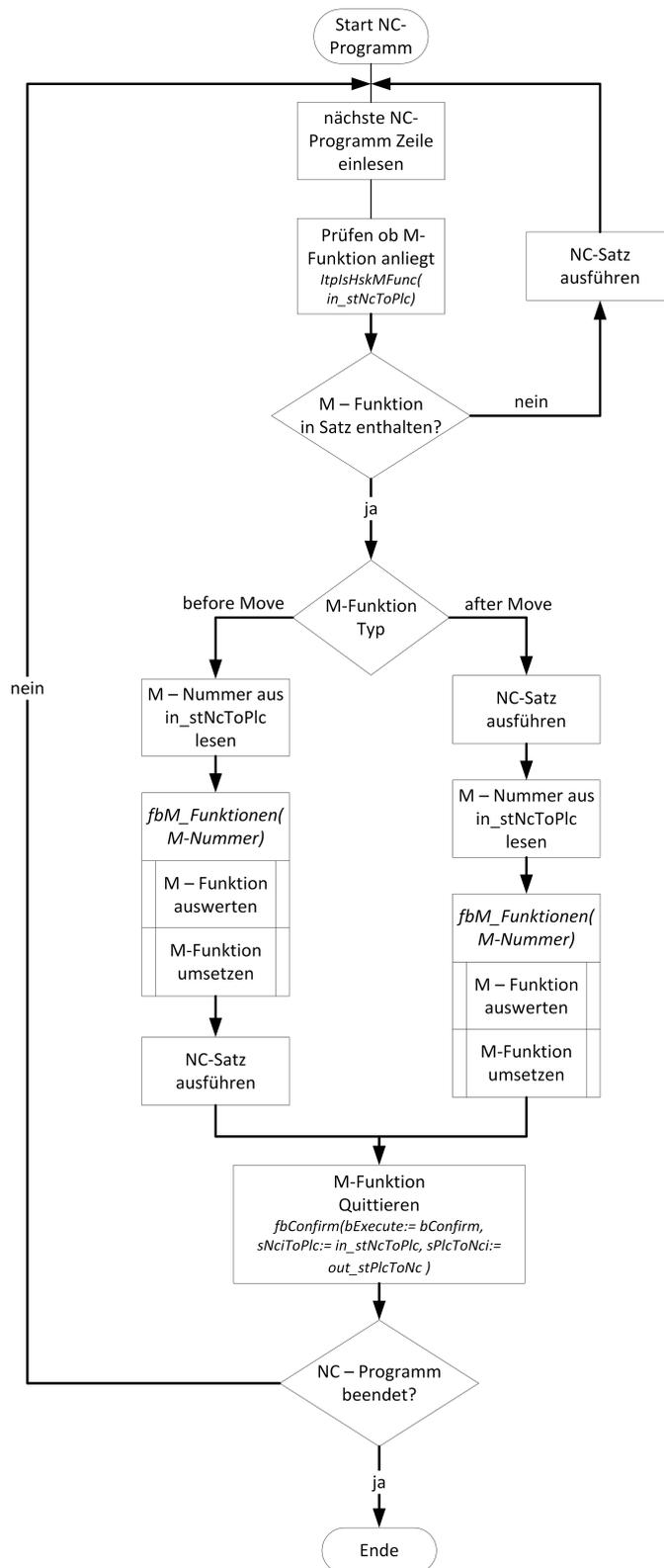


Abbildung 3.1. M-Funktionsverarbeitung

3.4.2 R-Parameter

R-Parameter, kurz für Rechenparameter, werden verwendet um mathematische Berechnungen im NC Programm auszuführen. Dies kann verwendet werden um eine Antastfunktion zu realisieren, bei der durch das Antasten zweier Kanten der Mittelpunkt des Werkstücks berechnet wird.

In der Steuerung der Laserschneidemaschine werden die R-Parameter dazu verwendet die Laserparameter aus dem NC Programm heraus zu steuern. Dazu wird im HMI die entsprechende Option aktiviert. Die im NC Programm hinterlegten Parameter R100 – R103 werden, während der Abarbeitung, in die Steuerung eingelesen. Diese Parameter stehen für Plusleistung, Pulsdauer und Pulsfrequenz. Dadurch ist es möglich diese Parameter während des Schnittes zu ändern.

3.4.3 Werkzeuge

In TwinCat NC I ist es möglich bis zu 255 Werkzeuge im Speicher zu hinterlegen. Es wird dabei zwischen Schaftfräser und Bohrer unterschieden. Bei Bohrern wird nur die Länge in die Bahnberechnung einbezogen, bei Schaftfräsern kann zusätzlich noch durch die Befehle G41 und G42 die Fräserradiuskorrektur aktiviert werden.

Bei der Laserschneidemaschine werden die Werkzeuge dazu verwendet den Laserstrahldurchmesser zu kompensieren. Die Werkzeuglänge stellt dabei ein Offset der Fokuslage dar. In den Laserparametern sind diese Daten hinterlegt und werden automatisch mit den Schneideparametern übernommen. Dadurch werden Schneidspaltbreite (d) und Fokuslage (f) automatisch dem gewählten Material angepasst.

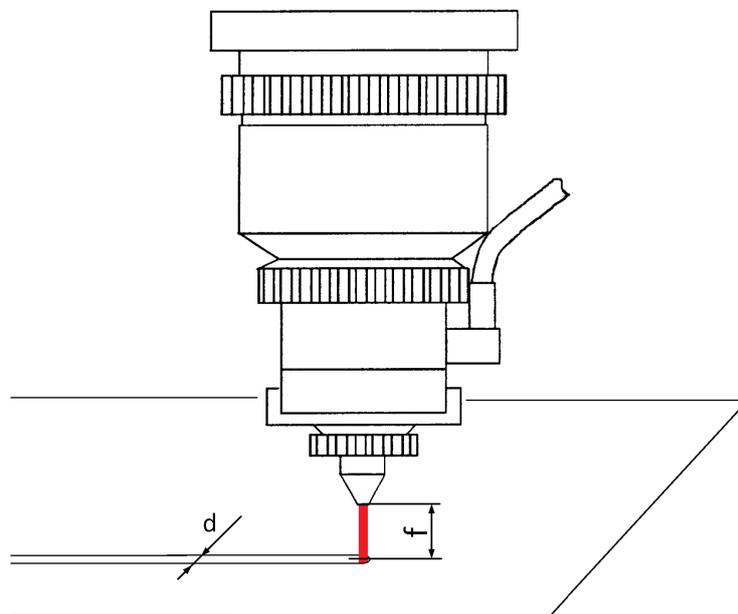


Abbildung 3.2. Schneidkopf [16] und Werkzeugparameter

3.4.4 Einzelsatz

Beim Einzelsatz hat der Benutzer die Möglichkeit das NC-Programm satzweise, also Zeile für Zeile, abarbeiten zu lassen. Die Satzweilerschaltung erfolgt durch den Maschinenbediener mit einem Tastendruck auf NC-Start. Dies ist im Einrichterbetrieb sehr hilfreich, da hierdurch das Programm sehr sicher auf Fehler im Ablauf untersucht werden kann. Die Schnittqualität wird dabei allerdings gering ausfallen, da sich durch die längeren Aufenthalte zwischen den Sätzen Einbrände ergeben.

Für die Umsetzung des Einzelsatzes steht in der SPS Bibliothek „TcNci.lib“ ein Funktionsbaustein „ItpSingleBlock“ bereit. Wird dieser aktiviert so stoppt der Interpreter automatisch nach jedem Satz und wartet auf eine Flanke am „bTriggerNext“ Eingang des Bausteins.

3.4.5 Testmodus

Im Trockenlauf wird das NC-Programm ohne die Umsetzung der M-Funktionen ausgeführt. Es verfahren also lediglich die Achsen. Dadurch hat der Benutzer die Möglichkeit das Programm ablaufen zu lassen, ohne dass ein Schnitt ausgeführt wird. Dies ist Hilfreich um das Programm auf Kollisionen zu Prüfen oder wenn unsicher ist ob der Platz auf dem vorhandenen Material für die Kontur ausreichend ist.

Die Implementierung der Funktion erfolgt im Funktionsbaustein „fbM_Funktionen“. Vor dem Ausführen einer M-Funktion wird geprüft ob der Testmodus aktiviert ist. Ist dies der Fall so wird die M-Funktion quittiert, ohne die Funktionalität umzusetzen.

3.4.6 Nullpunktverschiebung

Zur Nullpunktverschiebung stehen 6 Nullpunkte bereit, die vom Anwender programmiert und im NC Code verwendet werden können. 4 Nullpunkte(G54 – G57) werden über das HMI eingestellt. 2 Nullpunkte (G58 und G59) werden direkt aus dem NC Code heraus erstellt und parametrisiert. Im NC Code können die Nullpunkte dann über den entsprechenden Befehl z.B. G54 aufgerufen werden. Die Nullpunktverschiebung ist so lange gültig, bis sie mit G53 abgewählt oder durch den Aufruf eines anderen Nullpunktes ersetzt wird.

3.5 TwinSAFE Programm

Die Verbindung der TwinSAFE Gatter wird im SystemManager vorgenommen. Dazu wird für jedes Gatter ein Funktionsbaustein mit der entsprechenden Funktion angelegt. An jedem FB stehen bis zu 8 Eingänge zur Verfügung, die einzeln aktiviert und als Öffner- oder Schließerkontakt konfiguriert werden können. Es ist nur in einem UND-Glied möglich einen Standard Eingang, sprich SPS – Ausgang, mit anderen TwinSAFE Eingängen zu verknüpfen. Bei allen anderen Gattern ist dies nicht erlaubt. Dies ist eine Sicherheitsmaßnahme, da Standard SPS-Eingänge nicht überwacht und damit im TwinSAFE System als unsicher angesehen werden. Kein TwinSAFE Ausgang darf durch einen Standard Eingang geschaltet werden, ohne mit mindestens einem TwinSAFE Eingang verknüpft und damit abgesichert zu sein.

Die Schaltung, die der TwinSAFE Programmierung zu Grunde liegt, ist eine Stromstoßschaltung. Sie wird verwendet, da die Spannung der Maschine mit nur einem Taster ein- und ausgeschaltet werden soll.

Beide RS-Glieder sind zu Beginn rückgesetzt. Der Taster ist nicht betätigt. Wird der Taster nun betätigt, wird der Ausgang des oberen RS-Glieds gesetzt. Dieser Ausgang wird verwendet um den Schütz zu schalten. Mit Loslassen des Tasters wird das zweite (untere) RS-Glied gesetzt. Die Schaltung befindet sich nun im stabilen, eingeschalteten Zustand.

Wird der Taster ein weiteres Mal betätigt so wird das obere RS-Glied zurückgesetzt. Der Ausgang schaltet auf 0, der Schütz fällt ab. Durch das Loslassen des Tasters wird das untere RS-Glied zurückgesetzt. Die Schaltung befindet sich wieder im ausgeschalteten Grundzustand.

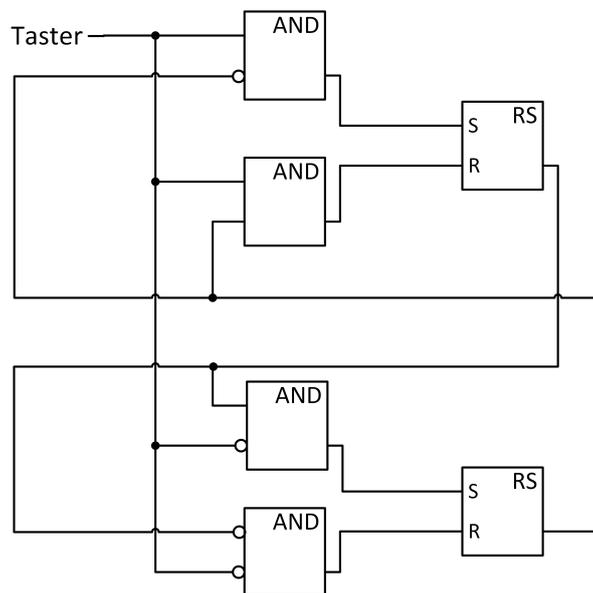


Abbildung 3.3. Stromstoßschaltung

Die Schaltung der Maschinensteuerung im TwinSAFE Modul wird um einen Not-Aus erweitert. Dieser ist Drahtbruchsicher als Öffner verdrahtet. Daher ist der Eingang des Not-Aus im Regelfall auf 1. Fällt der Not-Aus Eingang auf 0 so setzt er die beiden RS-Glieder zurück und somit auch den Ausgang des Schützes.

Um sicherzustellen, dass sich die Maschine im fehlerfreien Grundzustand befindet, wird die

Einschaltbedingung um weitere Eingänge erweitert. Dies sind

- Meldekontakt des Achsverstärkers
- Meldekontakt des Hauptschützes
- Meldekontakt des Trenntransformators
- Ausgang der SPS um Bereitschaft anzuzeigen (out_bSteuerung_Bereit)
- Not-Aus

Liefern alle Eingänge eine logische 1, so kann durch Betätigen des Tasters der Hauptschütz eingeschaltet werden. Der Ausgang des ersten RS-Gliedes wird neben dem Ausgang an den Hauptschütz auch an die SPS zurückgegeben, um hier den Zustand des Schützes anzuzeigen und im Programm zu verwenden. Desweiteren wird er dazu benutzt die Lasersteuerung einzuschalten. Dazu wird das Signal intern mit einem Ausgang der SPS, welcher signalisiert, dass die Lasersteuerung eingeschaltet werden soll, sowie dem Not-Aus zusammengeschaltet. Liefern alle drei Signale eine logische 1, so wird der Ausgang und damit der Schütz für die Lasersteuerung, geschaltet.

Wird in der SPS ein Fehler festgestellt so wird der Ausgang out_bSteuerung_Bereit zurückgesetzt. Dies soll, ähnlich dem Not-Aus, den Hauptschütz ausschalten. Dazu wird es negiert über ein ODER- Glied mit dem Not-Aus verknüpft. Da TwinSAFE es nicht zulässt einen normalen SPS Eingang direkt mit einem TwinSAFE ODER-Glied zu verknüpfen, muss der Ausgang der SPS zunächst mit einem Signal innerhalb des TwinSAFE Netzwerks mit einem UND-Glied verknüpft werden. Dazu wird das Ausgangssignal des ersten RS-Gliedes verwendet. Dies ist möglich, da das out_bSteuerung_Bereit Signal anliegen muss, um das erste RS-Glied setzen zu können. Somit sind beide Signale im fehlerfreien Zustand auf logisch 1. Durch die Negation des SPS – Ausgangs ist das Ergebnis der UND Verknüpfung logisch 0. Fällt der SPS – Ausgang auf logisch 0 so wird er durch die Negation zu logisch 1, was zusammen mit dem gesetzten RS-Glied eine logische 1 am Ausgang der UND-Verknüpfung erzeugt. Diese wirkt über das ODER-Glied auf die Rücksetzeingänge der RS-Glieder.

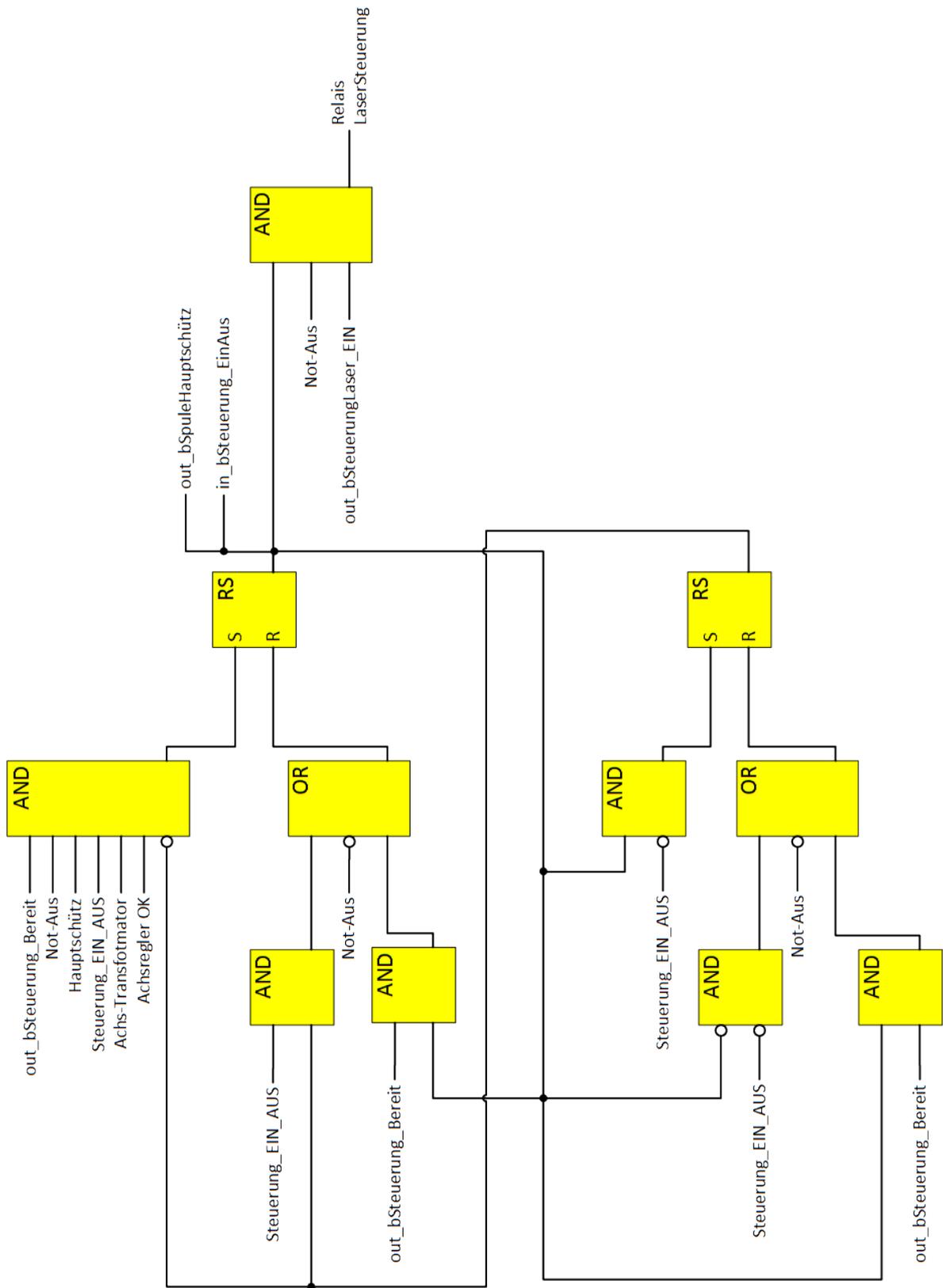


Abbildung 3.4. TwinSAFE Programmierung

3.6 SPS - Programmierung

Die Funktionsbausteine sind als Zustandsmaschine ausgeführt. Dadurch kann immer nur ein Zustand aktiv sein. Außerdem wird gewährleistet, dass alle Schritte abgearbeitet werden und die Reihenfolge der Abarbeitung eingehalten wird. Ein Sonderfall stellt dabei das Auftreten eines Fehlers dar, der zur Unterbrechung der Abarbeitung führt.

Das SPS-Programm enthält einen Programmbaustein und 26 selbstprogrammierte Funktionsbausteine. Daneben werden weitere Funktionsbausteine und Funktionen der TwinCAT Bibliotheken aufgerufen (vgl. Abb. 3.5).

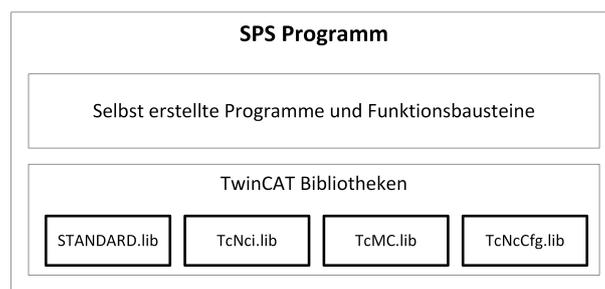


Abbildung 3.5. Aufbau des SPS Programms

Der Aufrufbaum zeigt die verwendeten Bausteine und ihre Instanziierungen. Er befindet sich im Anhang A.1.1. Der Programmbaustein MAIN wird nach Start der PLC Task als Erster aufgerufen. Er dient der übergeordneten Organisation. In ihm werden die Funktionsbausteine verwaltet. Wichtige Funktionen wie Fehlerreaktion und Startupprozedur (siehe Kapitel 3.6.1) sind direkt im Programmbaustein MAIN programmiert. Je nach Fortschritt der Startupprozedur schaltet MAIN weitere Funktionsbausteine und Funktionalitäten frei. Sicherheitsrelevante Funktionen sind immer verfügbar. Dazu gehören die Funktionsbausteine

- fbStop : anhalten der Achsen, stoppen eines NC-Programms, Rücksetzen der Start Flags
- fbReset: Rücksetzen des Interpreters, der Achsen, aller Flags und des globalen Fehlerbits
- fbKommunikation: Datenübertragung zwischen SPS und HMI
- fbFehlermanagement: Verwaltung des Fehlerpuffers, Übertragung der Fehlernummer an das HMI
- fbTuerueberwachung: anpassen der Achsen-Geschwindigkeiten bei geöffneter Schutztür, abschalten des Lasers bei öffnen der Schutztür

Außerdem Funktionsbausteine, die unabhängig der Betriebsart und des Betriebszustandes aufgerufen und ausgeführt werden können:

- fbOverride: Override der Achsen
- fbNullpunktverschiebung: Einrichtung der Werkstücknullpunkte

Der Programmbaustein MAIN verwaltet auch die Betriebsmodi. Je nachdem welcher Modus aktiv ist, wird der entsprechende Funktionsbaustein aufgerufen. Es kann jeweils nur ein Funktionsbaustein aufgerufen werden. Dadurch wird eine Überschneidung von Funktionalitäten oder Fehlbedienungen vermieden. Funktionalitäten, die in verschiedenen Funktionsbausteinen

benötigt werden, sind in einem eigenen Funktionsbaustein programmiert. Dadurch erhöht sich die Lesbarkeit des SPS – Programms. Außerdem verringert sich der Programmier- und Wartungsaufwand des Programms.

3.6.1 Spezielle Funktionsbausteine

Startup Prozedur

Die Startup Prozedur besteht aus einer Reihe von Abfragen, die gewährleisten sollen, dass sich die Maschine in einem betriebsbereiten Zustand befindet, bevor der Maschinenbediener Zugriff auf die Funktionen erhält. Sie wird nach dem Einschalten der Maschine und nach jedem Fehler ausgeführt. Wird während der Ausführung ein Fehler gefunden, so wird dieser an das Fehlermeldesystem weitergegeben. Die Prozedur wird unterbrochen, bis der Fehler vom Maschinenbediener quittiert wird. Danach wird die Prozedur erneut von Anfang an durchlaufen. Im ersten Abschnitt wird überprüft, ob in der Steuerung ein Fehler vorliegt. Dazu gehört die Überprüfung der EtherCAT Verbindung und der Zustand der TwinSAFE Baugruppen. Danach werden die Achsen auf Betriebsbereitschaft geprüft. Ist dies der Fall so wird dem Bediener eine Meldung ausgegeben, dass er die Betriebsspannung zuschalten kann. Nachdem das Zuschalten von der TwinSAFE Baugruppe bestätigt wird, werden die Achsregler freigeschaltet. Die Maschine befindet sich im betriebsbereiten Zustand. Noch kann aber nur im Handbetrieb verfahren werden. Die Bearbeitung von NC Programmen setzt eine erfolgreiche Referenzfahrt voraus. Wird die Referenzfahrt erfolgreich ausgeführt so, werden Automatikbetrieb und MDI freigeschaltet.

Referenzfahrt

Die Referenzfahrt wird bei NC Achsen mit inkrementellem Wegmesssystem benötigt, damit die Achsen richtig positioniert werden können. Das kommt daher, dass bei diesen Wegmesssystemen die aktuelle Position aus der Anzahl von Wegstücken relativ zu einem Bezugspunkt errechnet wird. Die Lage des Bezugspunktes sowie der Abstand der Achsen zu diesem Punkt gehen bei jedem Abschalten der Steuerung verloren. Daher müssen die Achsen neu referenziert werden. Die drei Indramat Servomotoren der Laserschneidemaschine besitzen einen Geber mit einer Auflösung von 3600000 Inkrementen pro Motorumdrehung. Eine Motorumdrehung entspricht, durch die Steigung des Kugelgewindetriebes, einer Strecke von 4mm. Damit entspricht jedes Inkrement einer Strecke von 1,1 nm, welche die Achse zurücklegt. Ist die Lage des Bezugspunktes bekannt, so kann die Steuerung bei jedem Impuls des Inkrementalgebers eine Strecke von 1,1 nm vom Bezugspunkt hinzu addieren oder subtrahieren und somit die aktuelle Position errechnen. Die Lage des Referenzpunktes muss nicht mit dem Maschinennullpunkt übereinstimmen. Wichtig ist nur, dass er nahe der Verfahrbereichsgrenzen der Achsen liegt. Dadurch wird sichergestellt, dass die Achse den Referenzschalter nicht schon überfahren hat, bevor die Fahrt beginnt. Bei der Referenzfahrt fährt jede Achse einzeln in eine bestimmte Richtung, die in der Steuerung hinterlegt ist. Es wird immer mit der Z-Achse begonnen und der Referenzpunkt der Z-Achse liegt in Z+ Richtung. Dadurch wird die Maschine freigefahren und die anderen Achsen können kollisionsfrei bewegt werden. Erreicht die Achse den Referenzschalter, meist ein berührungsloser Sensor, so wird die Bewegungsrichtung umgekehrt. Mit abfallendem Signal des Referenzschalters wird die Position in die Steuerung übernommen.

Dieses Vorgehen wird an den anderen Achsen wiederholt. Sind alle Achsen referenziert, so können automatisierte Bewegungen ausgeführt werden.

Ads Verbindungstimeout

Um sicherzustellen, dass HMI und SPS regelmäßig kommunizieren ist ein Mechanismus im SPS Programm eingebaut. Verbindet sich das HMI auf den Ads Port oder eine Variable in der SPS, so wird die Verbindung direkt überprüft. Ein Fehler wird daher sofort erkannt. In der SPS findet diese Überprüfung nicht statt, da die Ads Verbindung für den SPS - Betrieb nicht notwendig ist. Um die SPS zu stoppen, falls die Verbindung Unterbrochen sein sollte, befindet sich ein Kontrollbit in der stHMI_TO_PLC Structure über die das HMI und die SPS kommunizieren. Das Bit wird von einem Timer überwacht. Zu Beginn des Programmes ist das Bit nicht gesetzt. Der Timer wird gestartet. Wird das Bit durch das HMI gesetzt, so werden der Timer und das Bit durch die SPS zurückgesetzt. Wird das Bit vor Ablauf der Zeit nicht gesetzt so wird ein Fehler ausgelöst. Die Zeit des Timers beträgt 1,5 Sekunden, was drei HMI Updatezyklen entspricht.

3.6.2 Sicherheitsfunktionsbausteine

Diese Bausteine dienen dem sicheren Betrieb der Laserschneidemaschine. Sie werden aufgerufen, wenn ein fehlerhafter oder kritischer Zustand eintritt. Der Aufbau ihrer Programmstruktur unterscheidet sich insofern von dem anderer Bausteine, dass die Abarbeitung der programmierten Schritte im Fehlerfall nicht unterbrochen wird. Dadurch wird versucht die Maschine, trotz eines Fehlers, in einen sicheren Zustand zu überführen.

fbStop

Der Funktionsbaustein fbStop wird aufgerufen, wenn ein Fehler auftritt oder der NC – Stop Taster bzw. der NC – Stop Button auf dem HMI betätigt wird. Er stoppt die Ausführung des Interpreters und bringt die Achsen zu einem sicheren Halt. Außerdem wird das Laserprogramm abgebrochen und der Laser aus dem Standby-Betrieb entlassen. Um einen automatischen Neuanlauf zu verhindern werden die Flags zurückgesetzt.

fbStopAxes

fbStopAxes entzieht den Achsen ihre Reglerfreigaben. Dadurch kann eine Bewegung der Achsen nicht mehr stattfinden, da die Achsregler eingehende Bewegungsbefehle nicht mehr verarbeiten. Der Entzug der Reflerfreigabe muss auch innerhalb kurzer Zeit, nachdem der Hauptschütz abgeschaltet wird, erfolgen. Wird dies nicht gemacht, so melden die Achsregler einen Unterspannungsfehler und gehen in einen sicheren Ruhezustand, aus dem sie nur schwer zu reaktivieren sind.

fbReset

fbReset dient zum Zurücksetzen der Funktionsbausteine nach einem Fehler. Durch ihn soll die Maschine wieder in den Grundzustand gelangen um die Startupprozedur durchlaufen zu können. Dazu wird das globale Fehlerbit `m_bFehler` rückgesetzt. Danach werden Fehler der Achsen quittiert und ein Reset beim Interpretermodul und der Lasersteuerung durchgeführt. Im nächsten Schritt werden alle Funktionsbausteine in ihrer Ausführung unterbrochen und zurückgesetzt. Im letzten Schritt erfolgt das Zurücksetzen der Flags.

Türüberwachung

Die Schutztür des Bearbeitungsraumes dient dazu, den Maschinenbediener vor Verletzungen zu schützen. Bei der Laserschneidemaschine sind dies Gefahren, die von der Laserquelle ausgehen wie Laserstrahlung und Wärmeentwicklung sowie mechanische Gefahren, die sich aus der Bewegung der Achsen ergeben wie Quetschgefahr. Daher stellt die Schutztür ein Sicherheitsrelevantes Teil dar.

Die Position der Schutztür des Bearbeitungsraumes wird durch zwei Sicherheitsschalter überwacht. Durch die Überwachung wird gewährleistet, dass die Schutztür geschlossen ist bevor der Laser eingeschaltet wird. Sollte die Tür durch einen Fehler, z.B. Druckverlust durch einen beschädigten Druckluftschlauch, bei eingeschaltetem Laser aufgehen, so wird dies erkannt und die Anlage umgehend gestoppt.

Die Türüberwachung übernimmt auch die Begrenzung des Vorschubs. So werden die mechanischen Gefahren reduziert. Die Maschinenrichtlinie EN 12417 [11] schreibt vor, dass bei geöffneter Schutztür maximale Vorschübe von 2 m/min zulässig sind.

3.7 Benutzeroberfläche

Das Human Machine Interface (HMI) stellt den Dialog zwischen dem Bediener der Maschine und der Maschinensteuerung dar. Zur Benutzeroberfläche zählen alle Maschinenteile, die der Ein- und Ausgabe dienen. Auf der Benutzeroberfläche werden alle relevanten Daten angezeigt, die für die Bedienung der Maschine wichtig sind. Außerdem sind alle Bedienelemente, die der Benutzer zum Bedienen der Maschine benötigt, auf dem HMI angeordnet. Die Erstellung erfolgt nach ISO 9241 die als Richtlinie für die Gestaltung von Computerarbeitsplätzen gilt und eine Anleitung zur Erstellung einer Benutzeroberfläche gibt.

3.7.1 Bedienkonzept

Die Bedienung der Maschinensteuerung soll durch zwei Elemente realisiert werden. Ein Teil soll der reinen Eingabe dienen, während der Andere die Funktion von Ein- und Ausgabe übernimmt. Alle grundlegenden Elemente, die zur Bedienung der Maschine relevant sind sollen frei zugänglich auf der Oberfläche angebracht sein. Dazu gehören Not-Aus, Steuerung Ein/Aus, die Bedienung der Achsen im Handbetrieb, Umschalten der Betriebszustände, Bedienung der Tür, sowie der Lasersteuerung. Die Anzeige erfolgt durch ein Touch Panel. Die Touch- Funktion ermöglicht es dem Bediener Eingaben in die Steuerung zu machen. Die Anordnung der Bedienelemente wird durch zwei Bedienpositionen festgelegt, die der Maschinenbediener einnimmt.

Die erste Position (Abb. 3.6) ist stehend zwischen dem Arbeitsraum der Maschine und dem Bedienpult. Dabei hat der Bediener einen guten Einblick in den Arbeitsraum und kann dort agieren wie z.B. den Abstand zur Werkstückoberfläche mit Hilfe des Abstandsbleches einmessen.

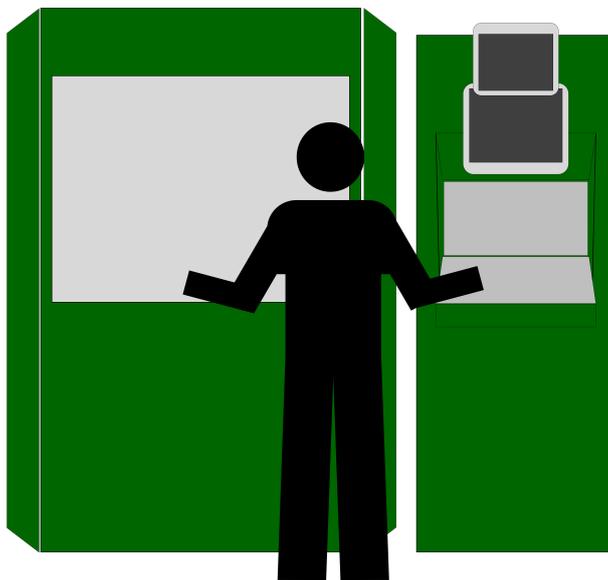


Abbildung 3.6. Maschinenbediener in Bedienposition 1

In der zweiten Bedienposition (Abb. 3.7) steht der Bediener direkt vor dem Bedienpult. Hier kann er Eingaben in die Steuerung machen wie z.B. Laserparameter einstellen oder ein NC Programm auswählen.

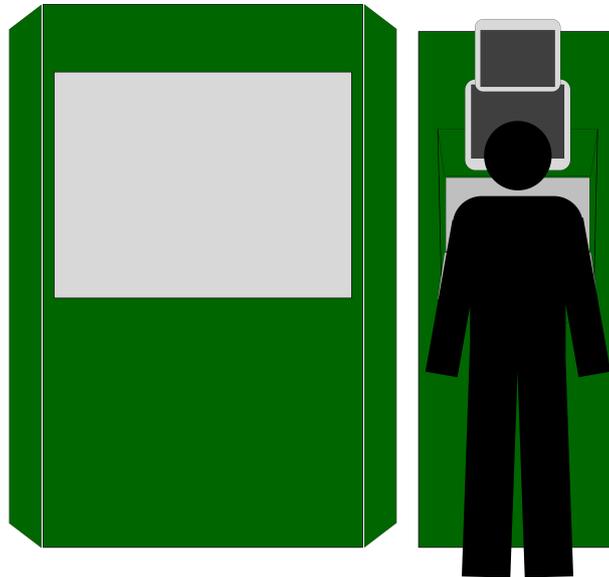


Abbildung 3.7. Maschinenbediener in Bedienposition 2

Die Anordnung der Bedien- und Anzeigeelemente erfolgt daher so, dass beide Bedienpositionen unterstützt werden. Der Panel-PC ersetzt den alten Röhrenbildschirm, der vor dem Umbau in den Schaltschrank eingebaut war. Das obere Panel A.1.9 ist leicht zur Vertikalen nach hinten geneigt, was dem Bediener die Betrachtung erleichtert. Durch die Neigung reicht die Oberfläche des Panels ca. 15cm in den Schaltschrank hinein, was Sonneneinstrahlung und Reflexionen verhindert. Außerdem weist der Bildschirm einen Betrachtungswinkel von 170° auf, was es ermöglicht seitlich auf die angezeigte Oberfläche blicken zu können.

Auf dem zweiten Panel A.1.10, welches sich unterhalb des Bildschirms befindet, sind die Tasten und die Tastatur mit Mousepad angebracht. Dieses Panel ist horizontal angebracht. Eine leichte Neigung nach unten verbessert den Zugang zu den Bedienelementen. Die Bedienelemente sind Rechtshänderkonform angeordnet. Nach ISO 9241-9 [8] sind 90% der Bevölkerung Rechtshänder.

Die Tasten, welche für die Bewegungssteuerung der Achsen verwendet werden, sind auf der linken Seite des unteren Panels angebracht. Das Verfahren der Achsen im Handbetrieb wird dadurch unterstützt, da sich der Bediener in diesem Fall in Bedienposition 1 befindet und die Tasten somit gut mit der rechten Hand erreicht werden können. Dadurch ist es dem Bediener auch möglich ca. 0,5m weiter in den Bearbeitungsraum der Maschine zu greifen. Das erleichtert die Arbeit beim Einmessen des Laserfokuses.

Die Tasten sind entlang der Bewegungsachsen angeordnet und entsprechend beschriftet. Dies ermöglicht eine intuitive Bewegungssteuerung. Eine dichte Gruppierung erlaubt das Drücken mehrerer Tasten mit einer Hand. Dadurch ist es möglich mehrere Achsen gleichzeitig zu Verfahren oder den Eilgang Taster, welcher sich in der Mitte der Anordnung befinden, zusammen mit einem Taster für eine Achsbewegung zu betätigen. Das versehentliche Betätigen mehrerer Tasten mit einem Finger wird durch einen überstehenden Rand verhindert. Darüber befinden sich der Drehpotentiometer, mit dem der Override der Achsen eingestellt werden kann, sowie

die Wahltaster der Betriebsmodi. Oberhalb ist der Not-Aus angebracht. Er ist nach DIN VDE 0113 [7] als gelb/roter Schlagschalter ausgeführt. Die Position erlaubt eine gute Erreichbarkeit von beiden Bedienpositionen. Zugleich verhindert die Platzierung eine unbeabsichtigte Betätigung, da oberhalb des Not-Aus keine weiteren Bedienelemente angeordnet sind. Unter den Tasten für die Bewegungssteuerung befinden sich die Taster für NC-Start und NC-Stop. Auch hier ist die zentrale Erreichbarkeit für die Positionierung ausschlaggebend. Elemente, die nicht so oft benötigt werden, befinden sich auf der rechten Seite des Bedienpanels. Dazu gehört der Taster um die Steuerung Ein- und Auszuschalten, ein Schalter zum Öffnen und Schließen der Schutztür sowie der Schlüsselschalter und ein Taster für Laserfreigabe bzw. Lasersteuerung Ein/Aus.

Mittig des Bildschirms, auf dem untern Panel, ist die Folientastatur mit Mousepad angeordnet. Sie dient der Eingabe von Daten in den Dialog der Maschinensteuerung. Auch sie ist für Rechtshänder konzipiert, da sich das Mousepad auf der rechten Seite der Tastatur befindet. Die Anordnung erlaubt die ergonomische Bedienung des Dialoges in der zweiten Bedienposition, bei der sich der Bediener direkt mittig vor das Bedienpult stellt.

Auf dem oberen Panel befindet sich der 15“ Touchbildschirm der zum Anzeigen des Dialoges, sowie der Eingabe dient. Rechts neben dem Bildschirm befinden sich vier Funktionstasten die jeweils einem Button im Dialog entsprechen und eine äquivalente Funktion belegen. Sie sind möglichst dicht am Bildschirmrand positioniert, um eine bessere optische Zuordnung zu erzeugen.

3.7.2 Bedienung der Benutzeroberfläche

[...]Die ISO 9241-110 behandelt die ergonomische Gestaltung von interaktiven Systemen und beschreibt Grundsätze der Dialoggestaltung, die grundsätzlich unabhängig von einer bestimmten Dialogtechnik sind, und die bei der Analyse, Gestaltung und Bewertung von interaktiven Systemen angewendet werden sollten. Dabei geht es um die Erstellung einer Benutzerschnittstelle und der Vermeidung von

- zusätzliche, unnötige Schritte, die nicht als Teil der Arbeitsaufgabe erforderlich sind;
- irreführende Information;
- unzureichende oder zu knappe Information der Benutzungsschnittstelle;
- unerwartete Antwort des interaktiven Systems;
- Einschränkungen beim Navigieren während der Benutzung;
- ineffiziente Behebung von Fehlern.

In der Norm werden dafür sieben Grundsätze beschreiben, die für die Gestaltung und Bewertung eines Benutzerdialoges wichtig sind.

- Aufgabenangemessenheit;
- Selbstbeschreibungsfähigkeit;
- Erwartungskonformität;
- Lernförderlichkeit;

- Steuerbarkeit;
- Fehlertoleranz;
- Individualisierbarkeit

[...] [10]

Die Norm dient als Grundlage zur Erstellung von Dialogen jeglicher Art. Daher können nicht alle dieser Grundsätze vollständig bei der Gestaltung einer Benutzeroberfläche für die Bedienung einer Laserschneidemaschine umgesetzt werden. Im Folgenden werden die Grundsätze genauer beschreiben und ihre Umsetzung bei der Gestaltung der Benutzeroberfläche der Laserschneidemaschine erläutert.

Aufgabenangemessenheit

[...]Ein interaktives System ist aufgabenangemessen, wenn es den Benutzer unterstützt, seine Arbeitsaufgabe zu erledigen. [...]

Auf der Benutzeroberfläche werden alle relevanten Informationen angezeigt. Um die Übersicht zu erhöhen besitzt die Oberfläche drei Designs, die je nach Betriebsmodus angezeigt werden. Die wichtigsten Informationen sind auf allen drei Benutzeroberflächen gleichermaßen zu finden und bilden damit den Grundaufbau der Benutzeroberfläche (Abb. 3.8). Dazu gehören die Anzeige der Statusicons, die Meldezeile, Positionsanzeige, Auswahl zur nullpunktorientierten Positionsanzeige und Override. Je nach Betriebsmodus kommen dazu noch weitere Anzeigen

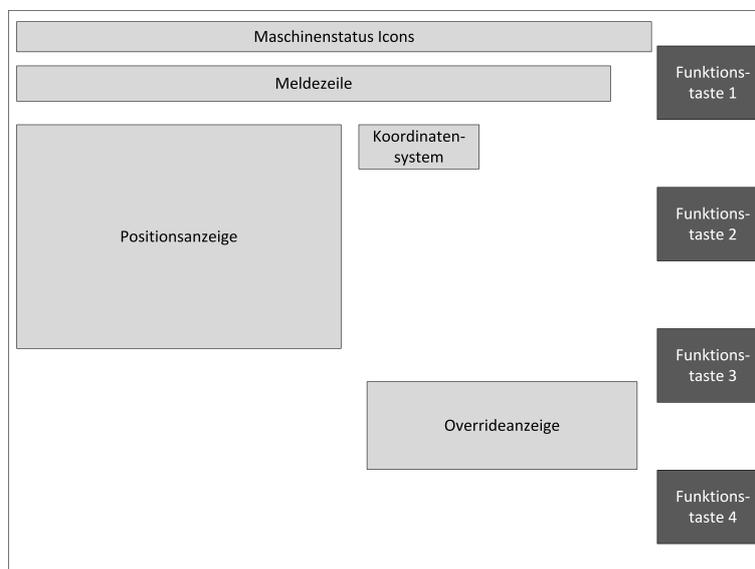


Abbildung 3.8. Grundaufbau des Benutzerdialogs

mit spezifischen Informationen. So z.B. eine Eingabezeile im MDI Modus über die die NC - Befehle in die Steuerung eingegeben werden können. Auf der Automatik Oberfläche werden das aktuell geöffnete NC – Programm sowie die gemessenen Laserleistungsdaten angezeigt. Die Anzeige der Positionen ist um die Sollposition, welche aus dem NC – Code herausgelesen

wird, und den Restweg, Strecke zwischen Ist- und Sollposition, erweitert. Zur Aufgabenangemessenheit gehört nach ISO 9241 auch die Unterstützung bei Eingaben. Dies soll am Beispiel des Dialoges Nullpunktverschiebung (Abb. 3.9) verdeutlicht werden. Der Bediener hat hier die Möglichkeit über drei Textfelder die Position des Nullpunktes des Werkstückkoordinatensystems entlang der drei Achsen zu definieren. Vor dem Textfeld für die Position befindet sich die Bezeichnung der Achse, dahinter die Einheit mm in der die Position angezeigt wird. Dem Bediener wird in den Textfeldern der Nullpunktposition zuerst die aktuelle Position angezeigt. Dadurch kann leicht errechnet werden ob die gewünschte Kontur mit der aktuellen Position als Nullpunkt geschnitten werden kann. Möchte der Bediener die aktuelle Position als Nullpunkt setzen so steht ihm dazu ein Button bereit, mit dessen Hilfe die Position genullt und zur Übernahme aktiviert wird.

The dialog box contains the following elements:

- Einzurichtendes Koordinatensystem:** A dropdown menu showing 'G 54'.
- Nullpunkt Setzen:** A dropdown menu.
- absolut:** A selected radio button.
- relativ zu:** An unselected radio button followed by a dropdown menu showing 'G 53'.
- Position X:** A text input field containing '0,000' and a button with '0'.
- Position Y:** A text input field containing '0,000' and a button with '0'.
- Position Z:** A text input field containing '0,000' and a button with '0'.
- Buttons:** 'Übernehmen' and 'Zurück'.
- Verschiebungen Table:**

	G54	G55	G56	G57
X-Achse	0,000	0,000	0,000	0,000
Y-Achse	0,000	0,000	0,000	0,000
Z-Achse	0,000	0,000	0,000	0,000

Abbildung 3.9. Dialog Nullpunktverschiebung

Selbstbeschreibungsfähigkeit

[...]Ein Dialog ist in dem Maße selbstbeschreibungsfähig, in dem für den Benutzer zu jeder Zeit offensichtlich ist, in welchem Dialog, an welcher Stelle im Dialog er sich befindet, welche Handlungen unternommen werden können und wie diese ausgeführt werden können. [...]

Der Benutzer soll dadurch unterstützt und mit nötigen Hilfen und Informationen versorgt werden. Dies wird durch eine durchgängige Beschriftung der Eingabefelder und Anzeigen erreicht. Icons auf den Buttons erlauben eine intuitive Bedienung und machen die Nutzung eines Handbuches während der Bedienung überflüssig. Werden Eingaben erwartet so ist, durch Angabe der Einheit, ersichtlich welches Eingabeformat erwartet wird.

Erwartungskonformität

[...]Ein Dialog ist erwartungskonform, wenn er den aus dem Nutzungskontext heraus vorhersehbaren Benutzerbelangen, sowie allgemein anerkannten Konventionen entspricht. [...] Dies wird erreicht, indem die Beschriftung der einzelnen Buttons und Textfelder übliche Begriffe für eine Maschinensteuerung verwenden. Die Icon für Maschinenfunktionen entsprechen der Norm ISO 55003-3 [4]. Andere Symbole und deren Funktion sind aus dem täglichen Umfeld z.B. der Windows Umgebung bekannt.



Abbildung 3.10. Icon aus der Windows Umgebung (Ordner öffnen)



Abbildung 3.11. Icon nach ISO 55003-3 (Einzelsatz)

Nach einer Eingabe erhält der Benutzer sofort eine passende Rückmeldung. So ändert sich beispielsweise der die Hintergrundfarbe eines Icons um den aktivierten Zustand anzuzeigen.



Abbildung 3.12. Icon im inaktiven Zustand (Referenzfahrt)



Abbildung 3.13. Icon im aktiven Zustand (Referenzfahrt)

Ist abzusehen, dass zwischen der Eingabe des Benutzers und der Reaktion durch das System eine längere Zeit vergeht, so wird auch dieser Zustand angezeigt. Umgesetzt ist dies z.B. beim Aufbau der Verbindung zur Lasersteuerung. Nachdem der Bediener die Lasersteuerung eingeschaltet hat muss er auf den Aufbau der Profibusverbindung warten, bevor weitergearbeitet werden kann. Dieser Vorgang dauert ca. 30 Sekunden. Um dem Benutzer anzuzeigen, dass am Verbindungsaufbau gearbeitet wird beginnt der Hintergrund des Icons rot/grün zu blinken. Ist die Verbindung hergestellt wechselt der Hintergrund dauerhaft auf grün.

Lernförderlichkeit

[...]Ein Dialog ist lernförderlich, wenn er den Benutzer beim Erlernen der Nutzung des interaktiven Systems unterstützt und anleitet. Dies soll durch Erklärung zugrundeliegender Konzepte, Unterstützung beim Erlernen neuer oder der Bedienung selten genutzter Dialoge oder der Rückmeldung bei erfolgreich ausgeführten Handlungen erfolgen. Der Lernaufwand sollte insgesamt so gering wie möglich gehalten werden. [...]

Der Nutzen der Lernförderlichkeit ist bei der Bedienung der Benutzeroberfläche sehr begrenzt. Die einzelnen Schritte, die zur Ausführung eines NC –Programms benötigt werden, sind, nach einer kurzen Einarbeitungsphase, leicht umzusetzen. Es wäre denkbar eine Anleitung zur Parameterentwicklung zu implementieren (vgl. 4.2.9). Dies ginge jedoch über den Umfang dieser Arbeit weit hinaus.

Steuerbarkeit

[...]Ein Dialog ist steuerbar, wenn der Benutzer in der Lage ist, den Dialogablauf zu starten sowie seine Richtung und Geschwindigkeit zu beeinflussen, bis das Ziel erreicht ist. [...]

Dieser Teil der Norm richtet sich an die Erstellung von Formularen bei denen der Benutzer zwischen den einzelnen Eingabemasken wechseln kann um Eingaben vorzunehmen. Allerdings wird hier auch der Umgang mit großen Datenmengen und voreingestellten Werten beschrieben, was am Beispiel der Lasersteuerung gezeigt werden soll. Die Lasersteuerung (Abb. 3.14) dient zur Verwaltung der Laserschneidparameter. Diese sind in einer Datenbank nach Material sortiert gespeichert. Jedem Material sind wiederum verschiedene Materialstärken und die zugehörigen Schneidparameter zugeordnet. Um die Auswahl des richtigen Parameters zu erleichtern wird die Anzeige gefiltert. Der Benutzer kann durch zwei Klicks die Parameter für ein Material herausuchen. Es werden jeweils nur die Materialien und Materialstärken angezeigt, für die Parameter hinterlegt sind. Nach der Auswahl werden die Parameter angezeigt, sodass der Benutzer die Möglichkeit bekommt die Parameter zu überprüfen, bevor er sie durch einen Klick auf einen Button in die Steuerung übernimmt. Diese voreingestellten Werte sind für den Normalfall ausreichend. Eine Änderung der Parameter vor der Übernahme in die Steuerung ist dennoch möglich und soll einem fortgeschrittenen Bediener die Möglichkeit bieten die Parameter entsprechend anzupassen. Dadurch wird die Arbeit z.B. während der Parameterentwicklung erleichtert.

LASER Status

- Verbindung
- Externe Ansteuerung
- LASER ist ein
- LASER bereit
- Programm aktiv
- Störung
- Überwachung
- Leistungsgrenze

Laser Parameter

Leistung W

Pulsdauer ms

Frequenz Hz

Auslastung %

Radius mm

L Offset mm

Vorschub mm/min

Material

Materialstärke

Laser Parameter aus NC Programm Verwenden

Abbildung 3.14. Dialog der Lasersteuerung

Fehlertoleranz

[...]Ein Dialog ist fehlertolerant, wenn das beabsichtigte Arbeitsergebnis trotz erkennbar fehlerhafter Eingaben entweder mit keinem oder mit minimalem Korrekturaufwand seitens des Benutzers erreicht werden kann. Fehlertoleranz wird mit den Mitteln erreicht:

- Fehlererkennung und -vermeidung (Schadensbegrenzung);
- Fehlerkorrektur oder
- Fehlermanagement, um mit Fehlern umzugehen, die sich ereignen.

[...]

Hier sollen zwei Beispiele für Fehlererkennung und Fehlerkorrektur aufgeführt werden. Das Fehlermanagementsystem wird in Kapitel 3.10 näher beschrieben. Bei der Einstellung der Nullpunktverschiebung wird, nach der Eingabe einer Position in ein Textfeld, der Inhalt des Feldes geprüft. Kann die Eingabe nicht in einen Zahlentyp gewandelt werden, z.B. weil Buchstaben eingegeben werden, so wird das Feld zurückgesetzt. Ist die Eingabe in einen Zahlentyp wandelbar so wird die Anzeige auf drei Nachkommastellen gerundet. Dies vermeidet Fehler wenn der Benutzer z.B. die Zahl 5,5mm eingeben möchte und 5.5 (entspricht 5500) in das Textfeld eingibt. Die Überprüfung formatiert die Anzeige dann auf 5500,000 wodurch die Fehleingabe deutlich wird.

Ein weiteres Beispiel ist die Überwachung der eingegebenen Laserparameter. Hier werden zum einen die Grenzen der Parameter überwacht. So darf beispielsweise die Pulsleistung nicht größer 8000 Watt oder kleiner 500 Watt sein. Wird der Wert über- oder unterschritten so wird die Anzeige rot gefärbt, sodass der Fehler sichtbar wird. Des Weiteren wird die mittlere Laserleistung berechnet. Ist diese größer 200 Watt so wird sie ebenfalls rot gefärbt. Es ist dennoch möglich die Parameter in die Steuerung zu übernehmen. Bei der Parameterentwicklung sollen die Grenzen des Möglichen ausgelotet werden. Durch vom System gesteckte Grenzen wäre das nicht möglich.

Individualisierbarkeit

[...]Ein Dialog ist individualisierbar, wenn Benutzer die Mensch-System-Interaktion und die Darstellung von Informationen ändern können, um diese an ihre individuellen Fähigkeiten und Bedürfnisse anzupassen. [...]

Dieser Leitsatz wird bei der Gestaltung nur hinsichtlich der Parametrisierung umgesetzt, wodurch der Benutzer alle Laserparameter nach Belieben anpassen kann. Eine Anpassung der Oberfläche ist nicht zweckmäßig und wird daher nicht realisiert.

3.7.3 Aufbau der Benutzeroberfläche

Die grafische Gestaltung der Benutzeroberfläche erfolgt nach ISO 9241-410 „Gestaltungskriterien für physikalische Eingabegeräte“ [12]. Dabei werden Anforderungen aufgezeigt um die Effektivität, Effizienz und Benutzerzufriedenheit zu sichern.

[...]Effektivität beschreibt die Genauigkeit und Vollständigkeit, mit der Benutzer bestimmte Ziele erreichen. [ISO 9241-11:1998]

Effizienz bezeichnet den, im Verhältnis zur Genauigkeit und Vollständigkeit, eingesetzten Aufwand, mit dem Benutzer Ziele erreichen. [ISO 9241-11:1998]

Zufriedenheit beschreibt das Freisein von Beeinträchtigungen und positive Einstellungen von Benutzern gegenüber der Nutzung des Produkts. [ISO 9241-9] [...] [12]

Die Gestaltungskriterien beziehen sich dabei auf

- Angemessenheit,
- Handhabbarkeit,
- Steuerbarkeit und
- biomechanische Belastung.

Im Anhang J der Norm befinden sich die Merkmale für Berührungsbildschirme. Die einzigen Richtlinien, die in dieser Norm beschrieben werden und für diese Arbeit relevant sind, lauten

- dass eine Berührungsaktive Zone mindestens die Größe des 95. Percentil (männlich) des Zeigefingers haben muss. Nach ISO 33402-2 [9] sind dies 22mm. Für kleinere Ziele ist ein Hilfsmittel oder softwaretechnische Unterstützung bereitzustellen.
- Um jeden aktiven Bereich sollte eine inaktive Zone von 5mm sein.

Die zweite Richtlinie lässt sich leicht umsetzen. Die Erste hingegen stellt ein Problem dar, da sich z.B. die Größe des DropDown Buttons bei einer ComboBox nur relativ zur Schriftgröße ändert. Außerdem wird nur die Höhe des Buttons angepasst. Die Breite bleibt konstant. Diesem Problem kann abgeholfen werden, indem ein Button über den DropDown Button gelegt wird, sodass er diesen komplett verdeckt. Der Button weiß im „_Click“ Event der „ComboBox.DroppedDown“-Eigenschaft den Wert „true“ zu. Damit übernimmt der Button die gleiche Funktion wie der Button, welcher in der ComboBox integriert ist. Bei Textboxen kann die Schriftart, aus Platzgründen, nicht so groß gewählt werden, dass jeder Buchstabe den geforderten 22mm entspricht. Hier kann mithilfe des Touchbildschirmes nur das Textfeld aktiviert werden. Für eine Positionierung innerhalb des Feldes werden die Pfeiltasten der Tastatur, die Maus oder der Bildschirmstift verwendet.

3.8 Implementierung des Benutzerdialogs

Die Implementierung des Benutzerdialogs erfolgt in Microsoft Visual Studio 2008. Dieses Programm stellt eine Entwicklungsumgebung für das .Net Framework bereit. Die Version 2008 wird verwendet, da hierdurch die Kompatibilität zur TwinCAT Library sichergestellt wird. Außerdem wurde das Visual Studio 2008 für Entwicklungen im .Net Framework 3.5 konzipiert, welches in Windows XP, dem Betriebssystem des Panel PC, verwendet wird. Das Projekt wird in der objektorientierten Programmiersprache Visual C# erstellt. Die Objektorientierung erlaubt eine übersichtliche Strukturierung des Projekts. Die visuellen Steuerelemente (vgl. Kapitel 0.2), welche in der Entwicklungsumgebung zur Verfügung stehen, unterstützen die Erstellung des Benutzerdialogs.

Um über Ads mit den TwinCAT Komponenten kommunizieren zu können wird die Bibliothek „TwinCAT.Ads.dll“ in das Projekt eingebunden. Sie stellt die Objekte zur Kommunikation wie den TcAdsClient oder AdsStream sowie Informationen über den Verbindungszustand wie den StateInfo oder DeviceInfo.

Das Programm des Benutzerdialogs, welches für die Laserschneidemaschine erstellt wird, beinhaltet sieben WindowsForms

- frmAutomatik (vgl. Anhang A.1.2)
- frmFehlerMeldungen (vgl. Anhang A.1.7)
- frmHandbetrieb (vgl. Anhang A.1.3)
- frmHMI_LASER (vgl. Anhang A.1.8)
- frmLasersteuerung (vgl. Anhang A.1.6)
- frmMDI (vgl. Anhang A.1.4)
- frmNullpunktVerschiebung (vgl. Anhang A.1.5)

und fünf Klassen.

- Datenbank
- GlobaleVariablen
- Kommunikation
- StatusBits
- Programm

Die Dialoge, welche durch die Forms dargestellt werden entsprechen den Richtlinien aus ISO 9241 wie in den Kapiteln 3.7.2 und 3.7.3 beschrieben.

Mit dem Start des Windows Betriebssystems wird eine im Autostart abgelegte Batch-Datei ausgeführt. Diese öffnet die ausführbare Datei „LASER_Oberflaeche.exe“. Dadurch wird die Benutzeroberfläche gestartet und auf dem Panel PC angezeigt.

Programmintern wird beim Programmstart zuerst die Klasse „Programm“ aufgerufen. Diese erzeugt eine WindowsForm vom Typ „frmHMI_LASER“. Diese Form dient als Grundbaustein des Benutzerdialogs. Wird diese Form geschlossen, so wird der Dialog geschlossen. Ihr kommt auch die Verwaltungsaufgabe innerhalb des Dialogs zu. Sie erzeugt je ein Objekt vom Typ frmHandbetrieb, frmMDI und frmAutomatik. Mit den Methoden „.Show()“ und “.Hide()“ wird

gesteuert welche Form auf der Oberfläche angezeigt wird, was sich am ausgewählten Betriebsmodus orientiert.

Neben den Oberflächen werden je ein Objekt vom Typ frmFehlerMeldungen und Kommunikation instantiiert. Dies bedeutet, dass die Klasse ein private static Objekt des eigenen Typs besitzt. Bei der Instantiierung wird die Methode „GetInstance()“ aufgerufen, welche dieses Objekt zurückgibt.

Codebeispiel aus der Klasse Kommunikation.

```

1 class Kommunikation
2 {
3     /* Diese Klasse dient zur Ads Kommunikation zwischen PLC und HMI */
4
5     // Instanz von Kommunikation
6     private static Kommunikation Instance = new Kommunikation();
7
8     public static Kommunikation GetInstance()
9     {
10        // Instanz zurückgeben
11        return Instance;
12    }
13 }

```

Dadurch greifen alle Klassen, welche die Klasse Instantiieren, auf das gleiche Objekt zu. Dadurch bleiben Zustände innerhalb der Klasse für alle zugreifenden Klassen erhalten. Dadurch ist es beispielsweise möglich, die Form frmFehlerMeldungen im Handbetrieb anzeigen zu lassen, in den Automatikbetrieb zu wechseln und die Form dort zu schließen. Ohne die Instantiierung würde in jedem Betriebsmodus ein eigenes Objekt erstellt wodurch die Klassenübergreifende Handhabung nicht möglich wäre. Die Dialoge von Nullpunktverschiebung und Lasersteuerung werden direkt von den WindowsForms frmHandbetrieb, frmMDI und frmAutomatik, nach einem entsprechenden Button klick oder FN-Tastenbetätigung instantiiert und angezeigt.

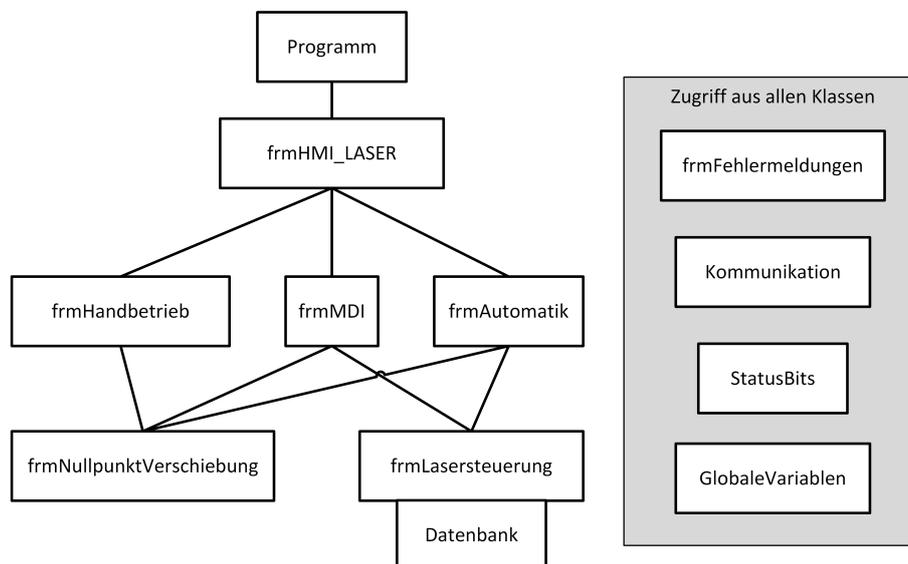


Abbildung 3.15. Klassen

Die Klasse „Kommunikation“ dient der Datenübertragung zwischen HMI und der SPS. Dieser Vorgang wird in Kapitel 3.9 beschrieben.

Die Klasse „Datenbank“ dient zur Verwaltung der Laserparameter. Die Klasse greift auf die Parameterdatenbank zu und bereitet die Schnittparameter für die Anzeige durch die WindowsForm „frmLasersteuerung“ vor. Außerdem sind in ihr Methoden zur Verwaltung der Datenbank implementiert, auf die über die Instanz zugegriffen werden kann.

Die Klasse „frmFehlerMeldungen“ dient zur Verwaltung der Meldungen und wird in Kapitel ?? beschrieben.

Die Klasse „GlobaleVariablen“ enthält Variablen, auf die von jeder Klasse im Programm zugegriffen werden kann. Dadurch wird die Verteilung der Informationen erleichtert. Die Daten, welche von der Klasse Kommunikation eingelesen werden, sind in diesen Variablen gespeichert. Die Klasse „StatusBits“ beinhaltet Bits, die den Maschinenzustand beinhalten. Der Zustand wird in der Statusleiste oberhalb der Meldezeile mit Hilfe von Icons angezeigt.

Jede WindowsForm enthält einen Timer, nach dessen Ablauf der „_Tick()“ Event ausgelöst wird. Im Eventhandler ist die Aktualisierung des Dialoges implementiert. Dabei erfolgt auch die Auswertung, ob ein FN-Taster betätigt ist. Da alle Oberflächen gleichzeitig existieren, jedoch die Belegung der FN-Taster unterschiedlich ist, wird jeweils nur die Funktionalität der Form ausgeführt, die den Fokus enthält.

3.9 Kommunikation

Die Kommunikation zwischen den einzelnen Bestandteilen der Laserschneidemaschine stellt einen wichtigen Teil dar. Nur dadurch ist das Zusammenwirken der einzelnen Komponenten möglich. Die Netzstruktur soll die Verbindungen und Kommunikationswege zwischen den einzelnen Bestandteilen aufzeigen.

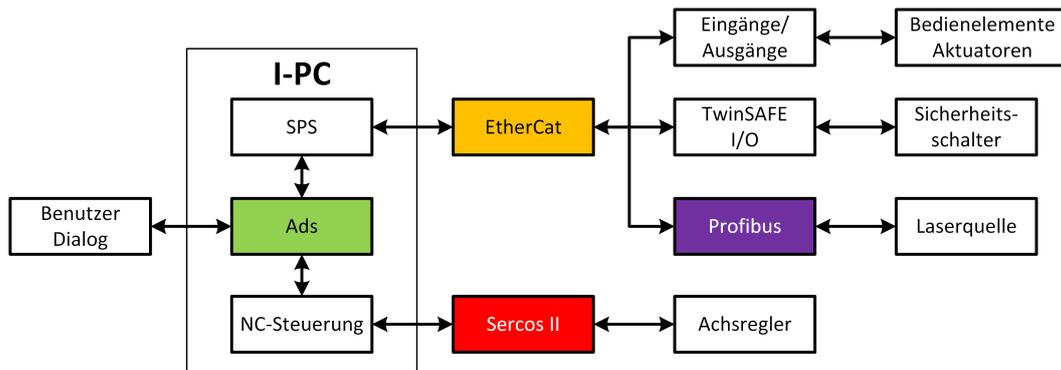


Abbildung 3.16. Netzstruktur der Laserschneidemaschine

EtherCAT

EtherCAT bildet die Verbindung zwischen der Software SPS und den Ein- und Ausgangsbaugruppen. Der Buskoppler 1100 der Firma Beckhoff stellt dabei das Bindeglied zwischen dem Bus und den Baugruppen dar. Der Buskoppler liest die für seine angehängten Baugruppen bestimmten Daten aus dem Bus aus und übermittelt gleichzeitig die Zustände und Daten seiner Baugruppen an die SPS zurück. Das Busnetz der Laserschneidemaschine ist so aufgebaut, dass auch die Profibusdaten zuerst über EtherCAT übertragen werden, bevor sie an das Profibusmodul durchgereicht und von dort weiter übertragen werden. Die EtherCAT Klemmen werden aneinandergesteckt und damit über einen Rückwandbus mit Energie und Daten versorgt.

Profibus

Über die von Trumpf bereitgestellte Software WinLas kann die Steuerung der Laserquelle konfiguriert werden. Hier werden die Laserprogramme erstellt und mit den Variablen des Profibus Interfaces verknüpft. Dazu stehen bis zu acht Ein- und Ausgangswörter zur Verfügung. Die Eingangswörter 0 und 1 sind statisch mit Steuerbefehlen belegt, mit der z.B. der Laser in Standby versetzt oder ein Laserprogramm gestartet werden kann. Die Zustände sind dabei bitweise kodiert d.h. jedes Bit beinhaltet einen Steuerbefehl. Eingangswort 1 dient der Übertragung der Programmnummer des Laserprogramms welches abgearbeitet werden soll. Die anderen Eingangswörter sind frei mit den Parametern der Laserquelle verknüpfbar. In dem Programm für die Laserschneidemaschine sind dies die Parameter für Laserleistung, Pulsdauer und Pulsfrequenz. Die Programmnummer wird von der SPS dauerhaft auf 1 gesetzt um das Laserprogramm auszuwählen. Die anderen Laserparameter werden je nach Einstellung im HMI an Material und Materialstärke angepasst.

Die Ausgangswörter sind alle statisch vorbelegt. Ausgangswort 0 und 1 besitzen den gleichen Aufbau wie die Eingangswörter. Sie liefern dadurch die direkte Rückmeldung, dass der Steuerbefehl richtig angekommen und verarbeitet worden ist. Ausgangswort 2 und 3 dienen der Übermittlung des Zustandes der Optik z.B. gewählter Strahlweg oder Stellung der Weichen. Meldungen werden in den Wörtern 4 und 5 übertragen. Ausgangswort 6 beinhaltet die mittlere Leistung der letzten Pulse, Ausgangswort 7 die gemessene Energie des letzten Pulses.

Zur Realisierung der Verbindung wird das EtherCAT Modul EL6731 verwendet. Dieses Modul stellt einen Profibus DP Master bereit, welcher im SystemManager konfiguriert werden kann. Neben den allgemeinen Einstellungen wie Baudrate und Zykluszeit wird eine Geräte-Stammdatendatei (GSD) verwendet um den Bus zu konfigurieren. In der GSD sind neben allgemeinen Angaben zu Hersteller und Gerätetyp spezielle gerätespezifische Informationen wie die Adressbereiche für den Speicherzugriff, unterstützte Verbindungsparameter und deren Bedeutung hinterlegt.

Sercos II

Über den Sercos II Bus sind die Achsregler mit der Steuerung verbunden. Hierdurch erhalten sie die Sollpositionsvorgaben von der NC Steuerung. Die Servomotoren melden ihre aktuelle Position an die Achsregler, die sie wiederum über Sercos an die Steuerung übertragen. Über die Sercoschnittstelle können die Achsregler parametrisiert und überwacht werden. Zur Einstellung der Parameter stellt TwinCAT entsprechende Dialoge bereit. Auch die Überwachung wird durch die TwinCAT Software übernommen. Wichtig ist dabei nur die richtige Verknüpfung und Umgang mit den Überwachungsdaten.

Um die Verbindung zu realisieren wird eine Beckhoff PCI Steckkarte FC7501 verwendet. Diese wird von der TwinCAT Software erkannt und installiert. Mit ihr kann ein Sercos LWL-Ring aufgebaut werden. Die Ringteilnehmer werden dabei automatisch erkannt und können mit den Achsen der NC I Task verknüpft werden. TwinCAT erlaubt die Parametrisierung des Busses, sowie eine online Überwachung der Teilnehmer und Zugriff auf die IDN Parameter.

Automation Device Specification

Die Automation Device Specification (Ads) Verbindung ist der zentrale Kanal für den Datentransfer zwischen HMI und PLC sowie HMI und NC I. Aufgrund des Aufbaus der TwinCAT Software ist es teilweise einfacher einen Parameter aus dem NC I Modul in die SPS einzulesen und von dort aus via Ads an das HMI zu senden und umgekehrt. Es besteht also eine Verbindung zwischen HMI und NC I bei der die SPS zwischengeschaltet ist.

Jede Task erhält vom SystemManager einen individuellen Port, über den sich ein Ads-Client mit der Task verbinden kann. Für den Zugriff auf eine bestimmte Variable innerhalb einer Task stehen dann verschiedene Methoden zur Verfügung. Es ist möglich die Variable über ihre Index Group und den Index Offset anzusprechen. Dies entspricht der Adresse der Variable im Speicher. Die Index Group teilt die Variablen dabei in verschiedene Kategorien ein z.B. Ein- und Ausgänge, Programm interne Variablen und Variablen zur Task Konfiguration. Der Index Offset beschreibt die Lage der Variable innerhalb der Index Group. Bei der PLC Ads Verbindung besteht die Möglichkeit direkt auf die Variablen im SPS – Programm zuzugreifen. Dazu steht eine Methode bereit, die einen Handler über den symbolischen Namen der Variable erzeugt. Auf diesen Handler kann sich der Ads-Client dann verbinden.

Die Daten werden bitweise in einen binären Datenstream gelesen. Aus diesem Stream heraus können dann über einen BinaryReader die einzelnen Werte ausgelesen werden. Es ist wichtig zu wissen, um welchen Datentyp es sich handelt und in welcher Reihenfolge sich die Variablen im Stream befinden, da es sonst zu Fehlinterpretationen des Streams kommen kann.

Zur Kommunikation zwischen PCL und HMI werden zwei Strukturen verwendet. Die Struktur `stHMI_To_PLC` beinhaltet 28 Variablen. Diese dient hauptsächlich der Übertragung von Steuervariablen um Funktionen, die in der SPS programmiert sind, aufzurufen. Über die Struktur werden auch die Daten übermittelt, die mit Hilfe der SPS in die NC-Steuerung geschrieben werden. Dazu gehören z.B. die Verschiebungen entlang der einzelnen Achsen für eine Nullpunktverschiebung. Für die Übertragung von der SPS in Richtung HMI wird die Struktur `stPLC_To_HMI` verwendet. Diese beinhaltet 44 Variablen, die zum größten Teil Daten beinhalten, die auf der Benutzeroberfläche angezeigt werden sollen. Dazu gehören Ist- und Sollwerte, Override, Fehlernummer und Flags, die den Maschinenzustand anzeigen. Neben diesen Variablen werden noch weitere übertragen, die den Zustand von Eingängen wiedergeben. Dadurch werden die Zustände der vier Funktionsbuttons an die Benutzeroberfläche weitergegeben und dort entsprechend verarbeitet.

Der Austausch zwischen HMI und SPS geschieht zyklisch im Abstand von 500 ms. Im Hauptfenster des HMI läuft dazu ein Timer, nach dessen Ablauf die Daten gelesen und geschrieben werden. Dazu wird die Klasse Kommunikation verwendet. Diese besitzt eine Lese – Methode. Die eingelesenen Daten werden im HMI einer identischen, global sichtbaren, Struktur zugewiesen. Jede Benutzeroberfläche besitzt wiederum einen UpdateTimer, nach dessen Ablauf die Daten aus der globalen Struktur gelesen werden und auf der Oberfläche angezeigt. Daten, die an die SPS übertragen werden sollen, werden ebenfalls in eine globale Struktur geschrieben. Diese Struktur wird dann zyklisch durch eine Sendemethode der Klasse Kommunikation übertragen. Die Übertragung kann auch azyklisch stattfinden, indem die Instanz der Klasse Kommunikation aufgerufen und mit deren Sendemethode gestartet wird.

In der SPS werden die Daten durch den Funktionsbaustein `fbKommunikation` verwaltet. In ihm werden die Daten aus der `stHMI_To_PLC` Struktur herausgelesen und globalen Variablen zugewiesen. Hier laufen auch alle Daten zusammen, die an das HMI übertragen werden sollen. Die Daten werden in die `stPLC_To_HMI` Struktur geschrieben und beim nächsten Updatezyklus vom HMI ausgelesen.

Die Laserparameter, die vom HMI an die Lasersteuerung übertragen werden sollen, werden zuerst über Ads an die SPS übertragen. Dort werden sie im Funktionsbaustein den Ausgangswörtern des Profibus zugewiesen und an die Lasersteuerung weitergeleitet.

Kritische Steuervariablen werden mit einem Handshake übergeben. Dies soll am Beispiel des NC-Programmstarts veranschaulicht werden (vgl. Abb. 3.17).

Der Bediener betätigt den NC-Start Button auf der Benutzeroberfläche. Dadurch wird in der `stHMI_To_PLC` Struktur das Bit `bHMI_NC_Start_Flag` gesetzt. Dieses Flag wird in der `stHMI_To_PLC` Struktur, von der Klasse Kommunikation, an die SPS übertragen. Dort wird die Flag im Funktionsbaustein `fbKommunikation` ausgewertet und an die globale Variable `m_bNC_Start_Flag` übergeben. Dies startet die Abarbeitung des NC-Programms. Gleichzeitig wird der Zustand der Flag auch in die `stPLC_To_HMI` Struktur geschrieben und von der Klasse Kommunikation des HMI ausgelesen. Damit erhält das HMI die Rückmeldung, dass der Befehl angekommen ist. Das Bit bleibt von der SPS so lange gesetzt, bis die Abarbeitung des NC-Programms beendet ist. Dadurch kann dies zum Einen auf der Benutzeroberfläche angezeigt werden, zum Anderen ist der Steuerung dadurch bekannt, dass gerade ein Programm abgearbeitet wird und kann verhindern, dass weitere Aktionen durchgeführt werden.

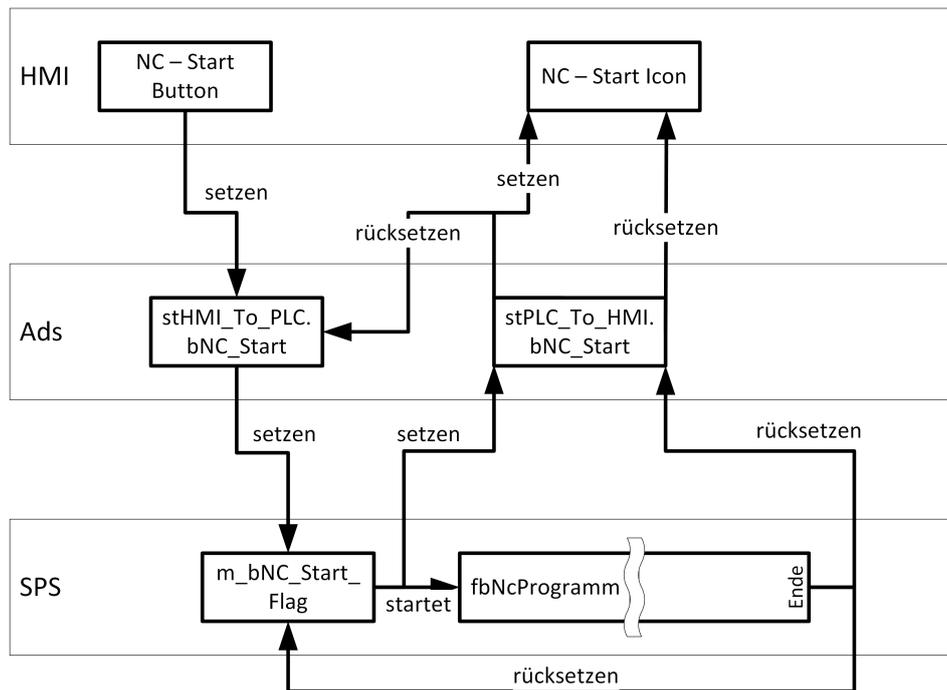


Abbildung 3.17. Vorgang zum Starten eines NC-Programms

Der direkte Zugriff auf Variablen der NC – Steuerung wird bei Werten verwendet, die nicht von der SPS eingelesen werden können oder um die SPS zu entlasten. Dabei wird die Verbindung über Index Group und Index Offset verwendet. Diese Verbindung wird verwendet, um die aktuell eingestellten Nullpunkte zurückzulesen oder die aktuelle NC – Programmzeile aus der Steuerung auszulesen. Die Nullpunkte werden in der Form Nullpunktverschiebung angezeigt und dienen dem Benutzer zur Kontrolle der Eingaben. Die aktuelle Programmzeile wird zum Einen dazu verwendet die Zeile in der NC-Code ListBox auf der Automatikoberfläche hervorzuheben zum Anderen wird die Zeile in ihre Bestandteile zerlegt. Daraus werden Sollpositionen, Vorschub und aktives Koordinatensystem abgeleitet und angezeigt.

3.10 Fehlermanagement

Die Aufgabe des Fehlermanagements liegt darin

- Fehler zu erkennen
- Fehler weiterzuleiten
- auf Fehler zu reagieren
- Fehler anzuzeigen
- Fehler zu beseitigen
- Fehler zu archivieren

Dadurch wird ein sicherer Betrieb der Maschine gewährleistet und Beschädigungen ausgeschlossen. Es wird zwischen zwei Orten unterschieden, an denen Fehler auftreten können. Zum einen ist dies das HMI zum anderen die SPS. Fehler, die in anderen Modulen des TwinCAT SystemManagers auftreten werden von der SPS erkannt und gemeldet.

Fehlermeldungen werden in drei Kategorien aufgeteilt. Hinweise, Warnungen und Fehler.

Hinweise sollen dem Maschinenbediener die Arbeit mit der Maschine erleichtern. Zum Beispiel durch das Anzeigen einer Meldung wenn die Startup Sequenz abgeschlossen wurde und die Maschine bereit zur Spannungszuschaltung ist.

Warnungen informieren den Bediener über einen Zustand, der bei weiterer Ausführung zu einem Fehler führen würde, welcher jedoch erkannt wird bevor der Fehler zustande kommt. Dies ist z.B. der Fall, wenn der Bediener im Automatikbetrieb den NC Start Taster betätigt ohne, dass vorher ein NC-Programm ausgewählt wurde. Dies hätte einen Interpreterfehler zur Folge. Wird die falsche Eingabe erkannt, so wird diese verhindert um den Fehler abzuwenden. Die Meldung weist den Benutzer auf seinen Fehler hin, sodass er diesen beheben kann.

Fehler werden angezeigt, wenn sich für den aktuellen Betriebszustand untypische Bedingungen ergeben.

3.10.1 Fehlerreaktion

Im Falle eines Fehlers werden die Antriebe gestoppt und die aufgerufenen Funktionsbausteine unterbrochen. Die SPS führt keinen Befehl aus bis der Fehler rückgesetzt wird. Dabei werden neben dem Rücksetzen des globalen Fehlerbits auch die Funktionsbausteine initialisiert. Damit wird der Grundzustand hergestellt. Nach dem Rücksetzen wird die Startup Prozedur erneut durchlaufen um die Funktionsfähigkeit der Maschine zu überprüfen.

3.10.2 Fehlermanagement SPS

Das Fehlermanagementsystem in der SPS stellt den größten und wichtigsten Teil des Systems dar, da hier alle Fehlermeldungen aus der SPS, den TwinCAT und TwinSafe Baugruppen sowie der NC und Achsensteuerung zusammenlaufen. Um eine Erkennung aller Fehler zu gewährleisten ist das Fehlermanagementsystem in alle Funktionsbausteine auf der SPS implementiert. Ziel ist es die Zeit zwischen dem Auftreten des Fehlers, seiner Erkennung und der entsprechenden Reaktion möglichst gering zu halten.

Beckhoff verwendet dabei ein System, bei dem die Fehler aus einer tieferliegenden Instanz an die jeweilig höhere weitergereicht werden [14]. Dieser Vorgang kann bei entsprechender Instanziierung sehr lange dauern, da jede Instanz ein bis zwei SPS Zyklen benötigt, um den Fehler weiterzuleiten.

Das in der Maschinensteuerung implementierte Fehlermanagementsystem setzt direkt an der untersten Ebene an. Damit liegt zwischen der Entstehung des Fehlers und der Verarbeitung lediglich ein SPS Zyklus.

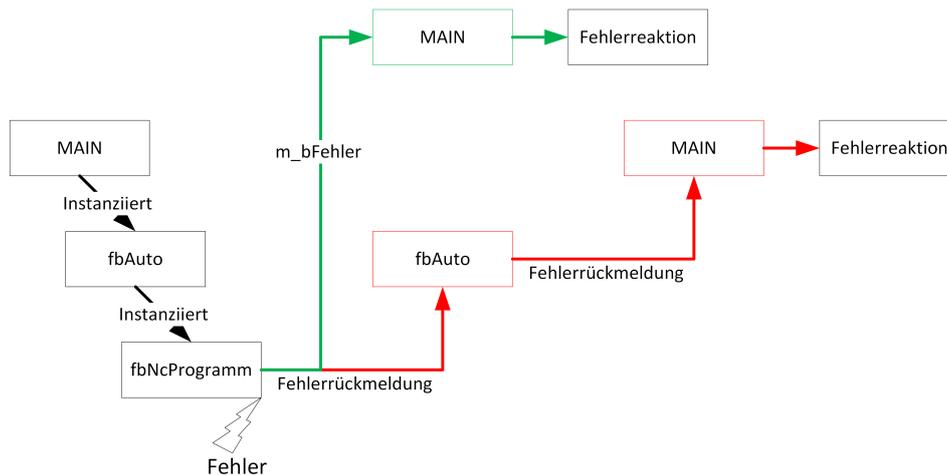


Abbildung 3.18. Fehlerrückmeldung

Dies wird durch die Verwendung von globalen Variablen möglich. Die Variablen sind im gesamten Projekt sichtbar und gültig und können somit von jedem beliebigen Punkt aus gelesen und beschreiben werden. Um den Fehler zu melden, wird ein globales Fehlerflag gesetzt, welches das Programm in jeden Zyklus überprüft. Ist dieses Flag gesetzt so weiß die Steuerung, dass ein Fehler vorliegt und kann darauf reagieren. Neben dem Flag wird eine Fehlernummer übergeben. Diese Nummer dient der Identifikation des Fehlers und der Anzeige für den Benutzer. Um alle Fehler aufzufangen, die in der Zeit zwischen dem Auftreten des ersten Fehlers und der Reaktion des Fehlermeldesystems auf diesen Fehler auftreten, wird die Fehlernummer in einen Array Puffer gespeichert. Das Array umfasst zehn Elemente. Ein Fehlerhandle sieht damit folgendermaßen aus:

```

1  (* Ein Fehler ist aufgetreten *)
2  IF Fehler THEN
3  (* globales Fehlerbit setzen *)
4  m_bFehler := TRUE;
5  (* Fehlernummer in Array schreiben *)
6  m_udiFehlerpuffer[m_iPuffernummer] := 1;
7  (* Zähler des Fehlerpuffers erhöhen *)
8  m_iPuffernummer := m_iPuffernummer +1;
9  (* in letzte Schritt des FB springen *)
10 nState := 9999;
11 END_IF

```

Der Sprung in den letzten Schritt des FB unterbricht die Ausführung des FB wodurch Folgefehler vermieden werden. Bei Sicherheitsrelevanten Vorgängen, wie z.B. dem Stoppen der Achsen, wird die Ausführung auch nach einem Fehler fortgesetzt. Hier wird versucht noch

möglichst viele nachfolgende Schritte auszuführen um die Sicherheit zu gewährleisten.

Das globale Fehlerbit der SPS wird durch einen AdsEventHandler überwacht. Der Handler führt die ihm angehängte Funktion bei jeder Änderung des Variablenzustandes aus. So wird der Zustand des globalen Fehlerbits an das HMI übertragen. Im Handler wird auch automatisch der Zeitpunkt des Zustandswechsels erfasst. Diese wird als Fehlerzeitpunkt in der Meldungsliste angezeigt. Die Fehlernummer wird über die stPLC_TO_HMI Structure übertragen. Die Verarbeitung des Zustandes und der Fehlernummer erfolgt in der WindowsForm „frmFehlermeldungen“. Die Form steuert auch das Auslesen des Fehlerarrays.

Bei Hinweisen aus der SPS wird die Hinweisnummer in ein globales Hinweisbyte geschrieben. Dieses Byte wird ebenfalls durch einen AdsEventHandler überwacht. Die Auswertung erfolgt ebenfalls durch die WindowsForm „frmFehlermeldungen“.

3.10.3 Fehlermanagement HMI

Das Fehlermanagement im HMI arbeitet, im Gegensatz zur SPS, nicht mit Fehlernummern sondern mit Strings, die die Meldungen enthalten. Dies vereinfacht und beschleunigt die Anzeige, da hier der Schritt entfällt die Meldung aus der Datenbank herauszulesen. Im HMI entstehen hauptsächlich Fehler, die mit der Ads Kommunikation zusammenhängen. Die Kommunikation wird daher über eine try- catch Anweisung überwacht. Im try-Block befinden sich dabei die auszuführenden Befehle. Schlägt die Ausführung fehl, so werden die im catch-Block enthaltenen Befehle ausgeführt. Ein Exception Objekt, welches an den catch-Block übergeben wird, liefert eine vom System erzeugte Fehlermeldung, die den Fehler im try-Block beschreibt. Diese Informationen werden an die „frmFehlermeldungen“ Klasse übergeben. Dies geschieht, ähnlich dem System auf der SPS, durch ein Puffer Array. Allerdings wird hier direkt die Fehlermeldung als String übergeben. Neben der Fehlermeldung aus dem Exception Objekt wird auch eine statische Meldung übergeben, die Aufschluss über den Entstehungsort gibt. Diese Meldung wird in der Meldungszeile angezeigt. Die Meldung aus dem Exception Objekt wird als erweiterter Hinweis in der Meldungsliste der Klasse „frmFehlermeldungen“ angezeigt. In einem zweiten Puffer Array wird der Fehlerzeitpunkt gespeichert. Er wird aus der Systemzeit, bei Aufruf des catch-Blocks ermittelt.

```

1  /* Ads Verbindung zu Port 501 herstellen (Lesen der Nullpunkte) */
2  try
3  {
4      adsClient = new TcAdsClient();
5      adsClient.Connect(501);
6  }
7  catch (Exception err)
8  {
9      /* ein Fehler ist aufgetreten */
10
11         // lokales Fehlerbit setzen
12         bFehler = true;
13         // globales Fehlerbit setzen
14         GlobaleVariablen.bHMIFehler = true;
15         // Fehlerzeitpunkt ermitteln
16         string DatumZeit = DateTime.Now.ToString() + "." +
17             DateTime.Now.Millisecond.ToString() + " ms";
18         // Fehlerzeitpunkt an Array übergeben
19         GlobaleVariablen.sHMIZeitpuffer[GlobaleVariablen.iHMIPuffernummer] = DatumZeit;
20         // statische Fehlermeldung an Array übergeben
21         GlobaleVariablen.sHMIFehlerpuffer[GlobaleVariablen.iHMIPuffernummer, 0]
22             = "Fehler beim Verbinden mit Port 501";
23         // von System erzeugte Meldung an Array übergeben
24         GlobaleVariablen.sHMIFehlerpuffer[GlobaleVariablen.iHMIPuffernummer, 1]

```

```

25         = err.Message;
26         // Zähler des Arrays erhöhen
27         GlobaleVariablen.iHMIPuffernummer += 1;
28     }

```

Hinweise aus dem HMI werden direkt als String an die Klasse „frmFehlermeldungen“ übergeben und verarbeitet.

3.10.4 WindowsForm „frmFehlermeldungen“

Diese Form stellt das Fehlermanagementsystem auf dem HMI dar. In ihr laufen alle Meldungen aus SPS und HMI zusammen. Die eingehenden Meldungen werden hier ausgewertet und zur Anzeige gebracht. Je nach Art der Meldung erhält sie eine Priorität, sodass gewährleistet ist, dass wichtige Meldungen, wie Fehler, immer angezeigt werden. Meldungen mit niedriger Priorität wie Hinweise oder Warnungen werden nur angezeigt, wenn keine Meldung mit höherer Priorität anliegt. Liegen mehrere Meldungen mit gleicher Priorität an, so wird die aktuellste auf der Benutzeroberfläche angezeigt. Auch die Anzeigedauer und Farbe der angezeigten Meldungstexte hängt von der Priorität ab. Fehler werden Permanent angezeigt, während Hinweise und Warnungen nach fünf Sekunden verschwinden. Dadurch soll verhindert werden, dass der Bediener dauerhaft mit Meldungen überhäuft wird. Durch die Farbe soll der Benutzer die Dringlichkeit der Meldung assoziieren. So werden Fehler in rot, Warnungen in orange und Hinweise in grün angezeigt.

Alle Meldungen werden, unabhängig von der Anzeige in der Meldezeile, in eine Liste eingetragen, die der Benutzer einsehen kann. Dadurch kann sich der Benutzer über vergangene und momentan aktuelle Meldungen informieren. Eine Zeile der Meldungsliste setzt sich aus mehreren Spalten zusammen. Die erste Spalte enthält ein Icon, welches Aufschluss über die Art der Meldung gibt. In der zweiten Spalte wird der Fehlerzeitpunkt angezeigt. Die dritte Spalte beinhaltet die Fehlernummern, welche nur bei Meldungen aus der SPS angezeigt wird. In der vierten Spalte befindet sich der Meldungstext, welcher auch in der Meldezeile angezeigt wird. In der fünften Spalte befindet sich ein erweiterter Hinweis, der weitere Informationen zur Meldung liefert.

Auswertung der Fehlernummern

Die Auswertung der Fehlernummern erfolgt im HMI durch die WindowsForm „frmFehlermeldungen“. Diese Klasse steuert die Meldungen, die in der Meldezeile der Benutzeroberflächen angezeigt wird und stellt eine Tabelle bereit in der alle Meldungen in chronologischer Reihenfolge angezeigt werden. Bei den Meldungen aus der SPS, bei denen eine Meldungsnummer übergeben wird, wird die Nummer mit einer Datenbank verglichen und die zugehörige Meldung auf der Oberfläche angezeigt. Ist kein Datensatz mit entsprechender Nummer vorhanden, so wird die Nummer auf der Oberfläche angezeigt. Dadurch hat der Benutzer die Möglichkeit sich selbst über den Fehler zu informieren.

Archivierung der Meldungen

Die Archivierung erfolgt durch die WindowsForm „frmFehlermeldungen“. Dazu wird jeden Tag, an dem die Maschine eingeschaltet wird, eine Log Datei erstellt. In dieser Datei werden

alle Meldungen in chronologischer Reihenfolge gespeichert. Wird die Maschine an einem Tag mehrfach eingeschaltet so sind die Meldungen durch einen Eintrag getrennt, der angibt wann die Maschine eingeschaltet wurde. Dadurch wird die Zuordnung der Meldungen erleichtert.

3.11 Schnittversuche

Nach dem abgeschlossenen Umbau und Neuparametrisierung der Steuerung soll nun in abschließenden Versuchen das Arbeitsergebnis überprüft werden. Neben dem erhöhten Sicherheitsaspekten und Bedienkomfort ergibt sich durch den Umbau auch ein zeitlicher Vorteil. Durch die direkte Ansteuerung des Lasers ist es möglich eine geänderte Rückzugstrategie im NC – Programm zu verwenden. Dadurch verkürzt sich die Hauptzeit des Bearbeitungsprozesses.

3.11.1 Versuch 1 Reduzierung der Hauptzeit

Mithilfe dieses Versuches soll die Zeitersparnis bei der Ausführung eines NC – Programms ermittelt werden.

Versuchsaufbau

Da sich lediglich die Nebenzeit, durch Änderung der Rückzugstrategie, verkürzt wird ein NC-Programm gewählt, welches aus vielen einzelnen kleinen Konturen besteht. Die Konturgröße beträgt 5x5mm mit einem Abstand von jeweils 2mm (vgl. Abb. 3.19). Dadurch soll der Effekt verdeutlicht werden, da sich das Verhältnis von Hauptzeit und Nebenzeit verkleinert.

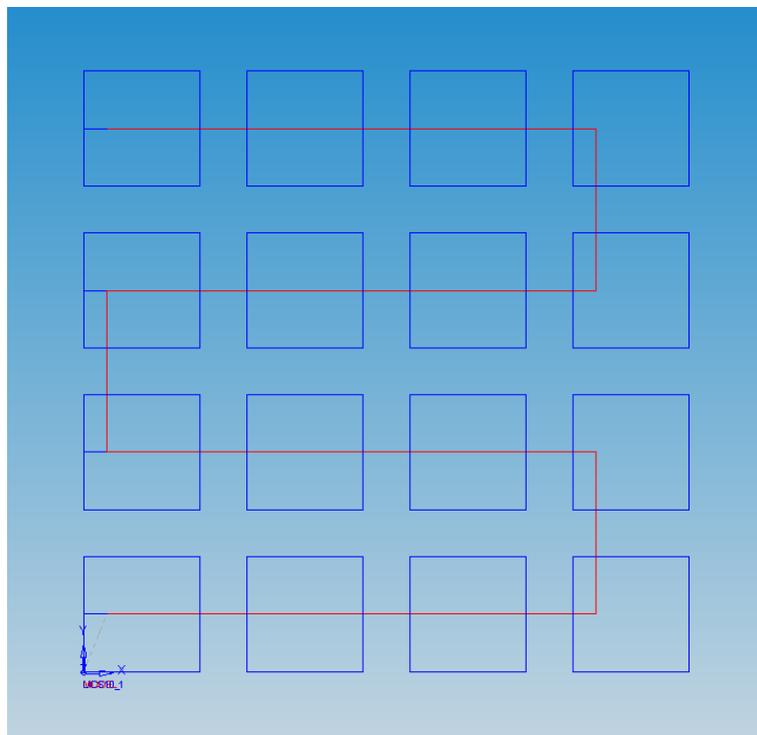


Abbildung 3.19. Gewählte Kontur, blaue Linien: Verfahrensweg in Bearbeitungsvorschub, rote Linie: Positionierbewegungen im Eilgang

Alle Bearbeitungsparameter wie Bearbeitungsvorschub, An- und Wegfahrt bleiben konstant. Auch die Laserparameter werden nicht geändert. Durch die Neuparametrisierung der Steuerung ist es möglich die Eilganggeschwindigkeit zu erhöhen. Es kann jedoch nicht mehr festgestellt werden, wie hoch die Geschwindigkeit der Maschine vor dem Umbau war. Daher bleibt dieser Einfluss ebenfalls unberücksichtigt. Als Material wird 1mm dickes Edelstahlblech verwendet.

Alte Strategie

Vor dem Umbau der Steuerung wurde der Laser durch einen separaten Laptop, der über Ethernet direkt mit der Lasersteuerung verbunden war, angesteuert. Mithilfe der Software WinLas wurde der Laser parametrisiert und Ein- bzw. Ausgeschaltet. Dadurch war es praktisch nicht möglich den Laser während der Bearbeitung exakt anzusteuern. Er wurde daher mit Programmstart eingeschaltet und nach beenden des NC- Programms ausgeschaltet. Um einzelne Konturen zu schneiden wurde in der Z-Achse so weit von der Materialoberfläche abgehoben, dass sich der Laserfokus so weit über dem Material befindet, dass er dieses nicht mehr schneidet sondern lediglich großflächig erwärmt. Die Rückzugsebene lag dabei bei 50mm. Zudem wurde 1mm über der Bearbeitungshöhe des Lasers mit einem Eintauchvorschub, der bei 30% des Bearbeitungsvorschubes lag, angefahren.

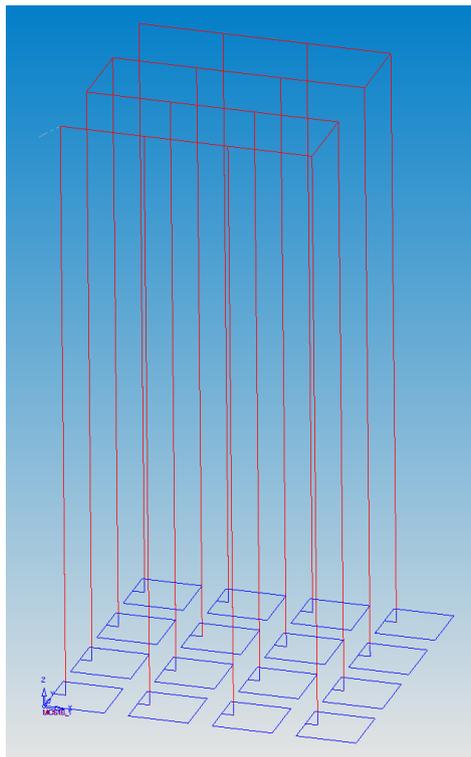


Abbildung 3.20. Alte Strategie 50mm Rückzug

Neue Strategie

Durch die direkte Ansteuerung des Lasers ist es nun möglich den Laser nach jeder Kontur auszuschalten. Werden Innenkonturen geschnitten, bei denen die Reststücke nicht durch Stege festgehalten werden sollte ein Sicherheitsabstand von 5mm programmiert werden, da sich kleine Teile in der Kontur verhaken und aufstellen können. Die Rückzugstrecke verkürzt sich dabei um 90%. Auch der Eintauchvorschub kann geändert werden. Da der Laser während der Anfahrt ausgeschaltet ist dient der Eintauchvorschub lediglich der genaueren Positionierung des Schneidkopfes über dem Werkstoff, sodass auch dieser erhöht werden kann. Es wird ein Vorschub gewählt, der dem Bearbeitungsvorschub entspricht.

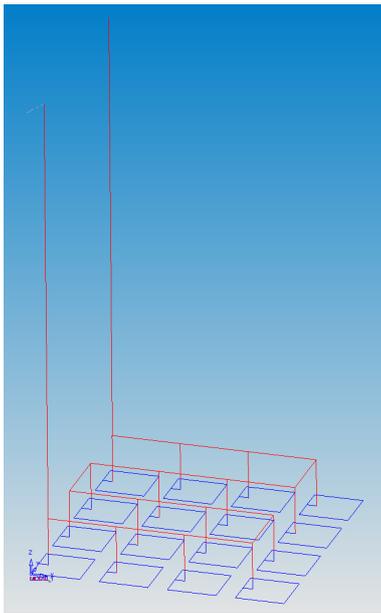


Abbildung 3.21. Neue Strategie mit 5mm Rückzug

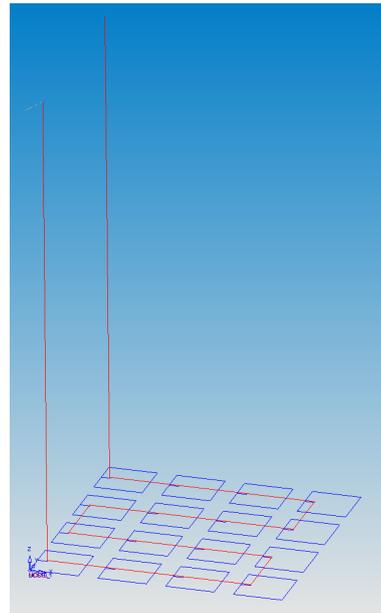


Abbildung 3.22. Neue Strategie ohne Rückzug

Versuchsauswertung

Tabelle 3.2. Auswertung von Versuch 1

	alte Strategie	neue Strategie 5mm Rückzug	neue Strategie ohne Rückzug
Gesamtstrecke [mm]	1275	617	452
Strecke im Eilgang [mm]	905	265	99
Strecke in Bearbeitung [mm]	352	352	352
Bearbeitungszeit [min:ss,ss]	1:30,16	1:12,69	1:01,53
Zeitersparnis [s]	0	17,47	28,63

Anmerkung: Die gemessenen Zeiten handelt es sich um die reine Bearbeitungszeit des NC – Programms ohne Ansteuerung des Laserstandby oder das Öffnen und Schließen der Schutztür.

Die Versuche zeigen, dass sich die Zeit um ca. 17 Sekunden verkürzt hat. Dies entspricht einer Verbesserung von 18%. Beim Schnittversuch ohne Rückzug beträgt die Verkürzung der Nebenzeit rund 29 Sekunden bzw. 32%. Diese Verbesserungen sind aufgrund der gleichbleibenden Schnittbedingungen allein auf die geänderte Rückzugstrategie zurückzuführen.

3.11.2 Versuch 2 Verbesserung der Schnittqualität

In einem zweiten Versuch soll die Verbesserung der Schnittqualität bewertet werden. Dazu wird jeweils ein Schnitt mit der alten und der neuen Rückzugstrategie gefahren. Beim Einstich entsteht ein Einbrand. Die Größe des Einbrandes wird durch die Verweilzeit beeinflusst. Durch den langen Anfahrweg und den geringen Eintauchvorschub entsteht bei der alten Anfahrstrategie ein großer Einbrand. Dieser ist stark kraterförmig, da der Fokus langsam auf die Materialoberfläche wandert und dabei viel Material abschmilzt, bevor er ein durchgängiges Loch erzeugt und das geschmolzene Material, durch den Luftstrahl, nach unten hinaus gedrückt werden kann.

Bei der neuen Anfahrstrategie befindet sich die Fokusslage des Laserstrahls schon in Bearbeitungstiefe, wenn der Laser eingeschaltet wird. Dadurch wird das Material schneller aufgeschmolzen und ein durchgängiges Loch entsteht. Außerdem ist die Verweilzeit am Einstichpunkt kürzer. Der Einbrand wird dadurch deutlich reduziert.

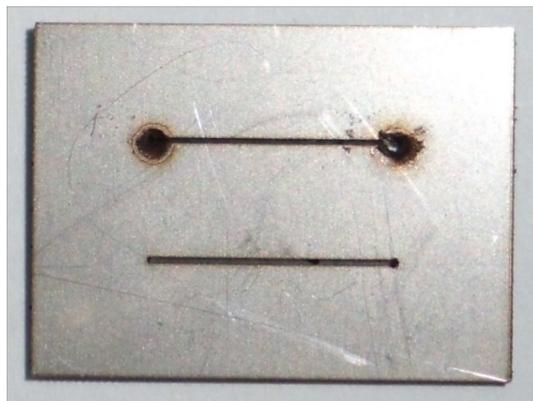


Abbildung 3.23. Vergleichsschnitte nach alter Strategie (oben) und Neuer (unten)

Der obere Schnitt in Abb. 3.23 zeigt Einstich und Rückfahrt nach der alten Strategie, der untere nach der Neuen. Wie man deutlich erkennen kann fällt der Einbrand beim Schnitt nach der alten Strategie deutlich größer aus im Vergleich zur unteren neuen Strategie. Während die Schnittbreite bei beiden Schnitten 0,2mm beträgt, ist der Einstich im oberen Schnitt mit einem Durchmesser von 0,93mm mehr als vier mal so groß. Auch beim Rückzug sieht man einen deutlichen Einbrand mit einem Durchmesser von 0,82mm. Der Einbrand am Einstich ist größer, da hier der letzte Millimeter mit Eintauchvorschub gefahren wird, während der gesamte Rückzug mit Eilgang stattfindet. Beim unteren Schnitt kann am Einstich keine Vergrößerung des Schnittes gemessen werden. Der Einbrand beim Rückzug ist mit 0,28mm im Durchmesser nur wenig größer wie der angrenzende Schnitt. Dadurch kann ein Einstich und Rückzug direkt an der Kontur nahezu ohne Qualitätsverlust stattfinden. Dieses Vorgehen wird in Versuch 3 angewandt.

3.11.3 Versuch 3 Alternative Bearbeitungsstrategie

In einem dritten Versuch soll eine neue Bearbeitungsstrategie untersucht werden, die nicht Konturorientiert sondern Richtungsorientiert arbeitet. Die Kontur soll mit möglichst wenigen Richtungswechseln geschnitten werden. Dadurch sind kontinuierliche Achsbewegungen möglich. Anfahr- und Abbremsvorgänge an den Richtungswechseln werden reduziert. Diese Bearbeitungsstrategie setzt den Einstich an der Kontur voraus und war daher vor dem Umbau nicht realisierbar. Es wird die Kontur aus dem ersten Versuch gefahren. Die Form und Anordnung der Schnitte ist bei dieser Strategie entscheidend. Bei runden Konturen kann dabei kein Vorteil entstehen, da sie aus einem permanenten Richtungswechsel bestehen. Die Bearbeitung erfolgt zeilenweise, wobei der Laser nach einem Schnitt, am Steg zwischen zwei Konturen, kurz ausgeschaltet wird und an der nächsten Kontur ohne Anfahrt direkt auf der Kontur einsticht und den Schnitt ausführt. Am Ende der Ziele wird schneidend auf die obere Hälfte der Kontur gewechselt und dort die Schnitte entlang der X – Achse ausgeführt. Nach den Schnitten entlang der X – Achse werden die Schnitte entlang der Y – Achse ausgeführt. Dadurch werden die einzelnen Schnitte zur Kontur verbunden und die Reststücke fallen heraus. Die Abfolge

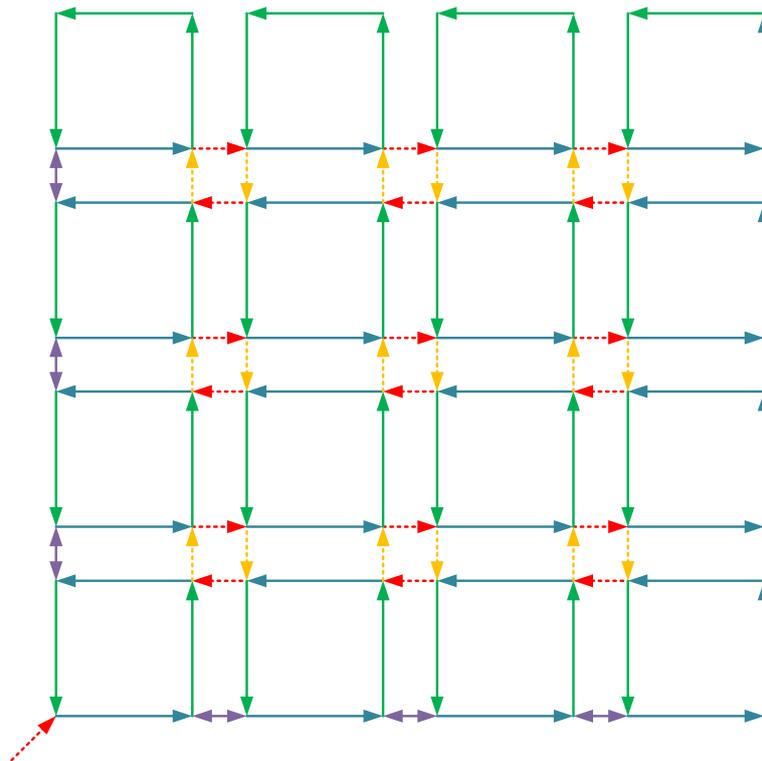


Abbildung 3.24. Alternative Bearbeitungsstrategie

der Schnitte ist in Abb. 3.24 dargestellt. Sie beginnt in der linken unteren Ecke. Zuerst werden die blau markierten Schnitte nach rechts ausgeführt. Die lila markierten Strecken werden im Eilgang gefahren. Am Ende der Ziele wechselt die Bearbeitung eine Zeile höher und wird nach links fortgesetzt. Es werden die rot markierten Strecken im Eilgang gefahren. Nach den blau und rot markierten Strecken wird mit der vertikalen Bearbeitung fortgefahren. Dabei werden die grünen Strecken schneidend und die orangen im Eilgang durchgeführt. Während der Bear-

beitung werden lediglich die sechs, lila markierten, Stecken mit einer Gesamtlänge von 12mm doppelt gefahren.

Bei dieser Strategie wird kein Rückzug verwendet. Die Gefahr, dass der Laser an herausgeschnittene Teile stößt ist gering, da diese bis zum abtrennenden Schnitt gehalten werden. Der Laserkopf bewegt sich danach von der Kontur weg und wird auch zu keinem späteren Zeitpunkt über die Kontur fahren. Zudem würde ein Rückzug, bedingt durch die vielen einzelnen Schnitte, eine hohe Nebenzeit generieren. Berechnung weg Die Strategie wird mit dem Schnitt aus Versuch 1 ohne Rückzug verglichen.

Tabelle 3.3. Auswertung von Versuch 3

		neue Strategie	alternative Strategie	
		ohne Rückzug	ohne Rückzug	5mm Rückzug
Gesamtstrecke	[mm]	452	416	896
Strecke im Eilgang	[mm]	99	96	576
Strecke in Bearbeitung	[mm]	352	320	320
Bearbeitungszeit	[min:ss,ss]	1:01,53	1:06,92	1:40,72
Zeitersparnis	[s]	0	-5,39	-39,19

Fazit

Versuche 1 und 2 haben gezeigt, dass durch den Umbau eine erhebliche Verbesserung an Qualität und Reduzierung der Nebenzeit erreicht wird. Die in Versuch 3 dargestellte Strategie eignet sich jedoch nicht für die Laserbearbeitung. Die Anforderungen an die Kontur sind zu speziell und eine Verbesserung gegenüber Versuch 1 konnte nicht erzielt werden.

4 Zusammenfassung und Ausblick

4.1 Zusammenfassung

In dieser Arbeit wurde die Realisierung einer Maschinensteuerung beschrieben. Die verwendete Software Steuerung TwinCAT der Firma Beckhoff stellt dabei das Grundgerüst dar. Mit ihr werden die Schnittstellen und Module parametrisiert sowie die TwinSAFE Baugruppen programmiert.

Die neu erstellte Benutzeroberfläche bietet dem Benutzer eine komfortable Bedienung. Das offene Bedienkonzept ist auf die Parameterentwicklung ausgelegt und soll dem Bediener einen flexiblen Einsatz gewährleisten. Weitere Maschinenfunktionen die umgesetzt wurden sollen den die Arbeit unterstützen.

Die umgesetzten Maschinenfunktionen sollen die Arbeit noch zusätzlich unterstützen.

Die Programmierung der SPS erfolgt mit dem TwinCAT Modul PLC Control. Für die Umsetzung wurden Funktionsbausteine aus der Beckhoff Bibliothek verwendet, die in selbst programmierten Bausteinen verknüpft und zu funktionalen Gruppen zusammengefügt wurden. Für die Steuerung der Laserquelle wurden eigene Funktionsbausteine angelegt.

Neben der SPS Programmierung stellt die Erstellung der Benutzeroberfläche einen weiteren Kern dieser Arbeit dar. Aufgrund des offenen Bedienkonzepts wurden hier viele Funktionalitäten hinterlegt. Außerdem wurde auf eine ergonomische und logische Anordnung der Bedienelemente geachtet. Hierdurch wird eine intuitive und komfortable Bedienung für den Maschinenbediener ermöglicht.

4.2 Ausblick

Hier sollen weitere Aufgaben beschreiben werden, die durch diese Arbeit ermöglicht werden. Diese Erweiterungen sind für die Funktion der Maschine nicht relevant, erleichtern jedoch die Bedienung und steigern den Benutzerkomfort.

Sie lassen sich in zwei Bereiche aufteilen. Ein Bereich stellt die Weiterentwicklung der Steuerung dar. Während dieser Arbeit sind viele Aspekte aufgetaucht, die in modernen Steuerungen Anwendung finden, jedoch in diesem Projekt nicht realisiert werden konnten. Dazu gehören:

- Vorschau
- Mustern
- Flaschenhalserkennung
- Fehlererkennung im NC Code
- Anpassung des Vorschubs
- Dynamische Laserleistung bei Kurvenfahrt
- Satzvorlauf

Der zweite Bereich bezieht sich auf die Forschung und Lehre an der Maschine.

- Parameterentwicklung
- Parameterentwicklungsprogramm

Auf die einzelnen Punkte soll nun genauer eingegangen werden um den jeweiligen Hintergrund und Inhalt der Idee darzustellen.

4.2.1 Vorschau

Nach dem Öffnen des NC-Programms wird die programmierte Kontur angezeigt. Dadurch ist leichter zu erkennen ob das richtige Programm geöffnet wurde und ob sich Fehler in der Kontur befinden. Wird die Nullpunktverschiebung in die Darstellung mit einbezogen so kann die Lage der Kontur im Bearbeitungsraum überprüft werden. Weiter denkbar ist die Anzeige während der Bearbeitung so zu aktualisieren, dass der Stand der Bearbeitung sichtbar ist. Der Bediener hätte dann einen besseren Überblick über den Bearbeitungsfortschritt.

4.2.2 Mustern

In der Musterfunktion soll es möglich sein ein aufgerufenes NC-Programm mehrfach im Bearbeitungsraum zu platzieren. Im einfachen Fall kann dies durch Angabe der Anzahl von Kopien entlang der X- und Y- Achse sowie deren Abstand zueinander realisiert werden. Die Steuerung könnte dann anhand der absolut größten Abmaße in X- und Y- Richtung, in Kombination mit dem Abstand, eine Nullpunktverschiebung nach jeder Kontur durchführen und das NC-Programm erneut ablaufen lassen. Eine umfangreichere Arbeit könnte die Konturen automatisch, nach Eingabe der Anzahl und eines Mindestabstandes, auf dem Blech anordnen.

Das Programm würde die Konturen so drehen und zueinander ausrichten, dass möglichst wenig Verschnitt entsteht. Eine Weiterentwicklung davon wäre die automatische Platzierung von verschiedenen Konturen mit beliebiger Anzahl. Auch hier wieder unter dem Gesichtspunkt der Reduzierung von Verschnitt.

4.2.3 Flaschenhalserkennung

Die Flaschenhalserkennung verhindert Konturverletzungen während der Bearbeitung. Dabei wird eine bestimmte Anzahl von NC-Sätzen vorausgelesen und überprüft, ob es zu einer Verletzung der Kontur durch den Laserstrahl kommt. Voraussetzung dafür ist die Verwendung der Werkzeugradiuskorrektur, auf deren Grundlage die Berechnung stattfindet. Die Flaschenhalserkennung ist bei kleinen Strukturen besonders wichtig, da es hier eher zu Konturverletzungen kommt.

4.2.4 Fehlererkennung im NC-Programm

Nach Öffnen des NC-Programms werden die Sätze auf Fehler untersucht. Der Benutzer bekommt die Fehler angezeigt und hat die Möglichkeit diese vor Programmstart zu beseitigen. Die Fehlererkennung kann von einer schlichten Prüfung der Syntax bis hin zur Interpolation der Bahn reichen. Dadurch werden Interpreter- und Interpolatorfehler während der Abarbeitung des NC-Programms verhindert.

4.2.5 Anpassung des Vorschubs

Bisher wird der Vorschub bei der Programmierung der Kontur angegeben und in das NC-Programm übernommen. Soll die gleiche Kontur nun in zwei verschiedenen Materialien geschnitten werden so muss der Vorschub von Hand angepasst werden, z.B. durch die Suchen und Ersetzen Funktion des Editors. Die automatische Anpassung des Vorschubs an Material und Materialstärke würde den Bedienkomfort steigern. Die Anpassung könnte mit der Laserparametereinstellung verknüpft werden. In der Datenbank der Laserparameter ist dafür schon ein Datensatz mit den Vorschubgeschwindigkeiten angelegt.

4.2.6 Dynamische Laserleistung bei Kurvenfahrt

Bisherige Schnitte haben gezeigt, dass es bei Ecken und kleinen Radien zu einem erhöhten Energieeintrag kommt, der sich durch die Bildung von Einbränden negativ auf das Schnittergebnis auswirkt. Denkbar wäre es die Laserleistung an die Vorschubgeschwindigkeit zu koppeln und damit einen geringeren Energieeintrag in den Ecken zu erzeugen.

4.2.7 Satzvorlauf

Der Satzvorlauf würde es ermöglichen ein unterbrochenes NC-Programm an einem beliebigen NC-Satz fortzusetzen. Dadurch ist es nicht mehr nötig das Programm immer von Anfang an laufen zu lassen. Dadurch wird die Schnittqualität verbessert, da Konturen dann nicht mehr zweimal geschnitten werden.

4.2.8 Parameterentwicklung

Momentan steht dem Anwender nur ein kleiner Datensatz von Materialien und Materialstärken zur Verfügung. Versuche haben gezeigt, dass ein Teil dieser Parameter fehlerhaft und nicht für die Bearbeitung geeignet sind. Um die Produktivität und die Schnittqualität zu steigern ist es notwendig Schneidparameter zu entwickeln. Dabei könnten neben den metallischen Werkstoffen auch Parameter für Kunststoffe und Keramiken entwickelt werden. Dadurch entsteht ein Wissen über den Einfluss der einzelnen Parameter des Laserschneidens, was der Lehre an der Hochschule zugutekommt. Außerdem könnte es einen interessanten Vergleich zu anderen thermischen Abtragsverfahren darstellen.

4.2.9 Parameterentwicklungsprogramm

Das Parameterentwicklungsprogramm stellt eine Erweiterung der händischen Parameterentwicklung dar. Dabei soll ein Programm automatisch Schnitte ausführen, die von Nutzer lediglich auf ihre Qualität bewertet werden müssen. Das Programm, in dem eine Art statistische Prozessplanung hinterlegt ist, wählt dann selbstständig die geeignetsten Parameter aus. Dies würde die Parameterentwicklung gerade bei unbekanntem Material beschleunigen. Als Vorarbeit zu diesem Projekt ist die Verwendung der R-Parameter zur Laserparameteransteuerung in der Steuerung eingerichtet. Dadurch können durch ein externes Programm die Schnitte, samt der dafür vorgesehenen Parameter, in ein NC-Programm geschrieben werden. Der Benutzer kann dann alle Schnitte nacheinander laufen lassen ohne die Maschine anhalten zu müssen um die Schneidparameter zu ändern.

Literaturverzeichnis

- [1] EtherCAT Technologie Group: *Homepage*. <http://www.ethercat.org/en/technology.html>
- [2] PROFIBUS Nutzerorganisation e.V. (PNO): *PROFIBUS Systembeschreibung – Technologie und Anwendung*. <http://www.profibus.com/nc/download/technical-descriptions-books/downloads/profibus-technology-and-application-system-description/display/>
- [3] sercos international e.V.: *Sercos II - Technische Kurzbeschreibung*. http://www.sercos.de/sites/default/files/sercos2_kurzbeschreibung_de.pdf
- [4] Norm ISO 55003-3 03 1981. *Numerisch gesteuerte Werkzeugmaschinen - Teil 3: Bildzeichen*
- [5] Norm DIN 66025-2 09 1988. *Programmaufbau für numerisch gesteuerte Arbeitsmaschinen - Teil 2: Wegbedingungen und Zusatzfunktionen*
- [6] Norm IEC 61131-3 08 1994. *Speicherprogrammierbare Steuerungen - Teil 3: SPS-Programmiersprachen*
- [7] Norm DIN VDE 0113 11 1998. *Elektrische Ausrüstung von Maschinen - Teil 1: Allgemeine Anforderungen*
- [8] Norm ISO 9241-9 02 2000. *Ergonomie der Mensch-System-Interaktion - Teil 9: Anforderungen an Eingabemittel ausgenommen Tastaturen*
- [9] Norm ISO 33402-2 12 2005. *Körpermaße des Menschen - Teil 2: Werte*
- [10] Norm ISO 9241-110 04 2006. *Ergonomie der Mensch-System-Interaktion - Teil 110: Grundsätze der Dialoggestaltung*
- [11] Norm EN 12417 03 2006. *Werkzeugmaschinen - Sicherheit - Bearbeitungszentren*
- [12] Norm ISO 9241-410 07 2012. *Ergonomie der Mensch-System-Interaktion - Teil 410: Gestaltungskriterien für physikalische Eingabegeräte*
- [13] BECKHOFF: *Homepage*. <http://beckhoff.de/>
- [14] BECKHOFF: *Informationssystem*. <http://infosys.beckhoff.de/>
- [15] FRITZ, M.: *SPS 1 (SIMATIC S7) - Grundlagen*
- [16] KG, Trumpf Laser GmbH + C.: *Handbuch; Lasergeräte HL 62P/LCB; HL 124P/LCB; HL 204P/LCB*
- [17] MICROSOFT: *MSDN*. msdn.microsoft.com
- [18] WIKIPEDIA: *Nd:YAG-Laser*. <http://de.wikipedia.org/wiki/Nd:YAG-Laser>

A Anhang

A.1 Bildanhang

A.1.1 Aufrufbaum

Anmerkung: Bei den farblich markierten Funktionsbausteinen handelt es sich um jene, die im Rahmen dieser Arbeit erstellt wurden.

A.1.3 Handbetrieb Dialog

The interface displays the following elements:

- Status Bar:** Includes icons for PLC, TC, and a red stop symbol. Below them, the text "Meldungen" and "Override 0.0%" is shown.
- Achspositionen (Axis Positions):** Three input fields for X, Y, and Z axes, each containing the value "0,000 mm".
- Koordinatensystem (Coordinate System):** A dropdown menu currently set to "absolut".
- Override:** A slider control labeled "Override" with a value of "0,0" and a scale from "0%" to "100%".
- Tastbetrieb (Button Operation):** A row of buttons with values: "10,0", "1,0", "0,1", "0,01", "0,001", and "INF".
- Function Keys (FN1-FN4):** Located at the bottom, these include:
 - FN1: Referenzfahrt (Reference Travel)
 - FN2: NPV (New Position Value)
 - FN3: Reset
 - FN4: Einstellungen (Settings)

A.1.4 MDI Dialog

The MDI dialog interface consists of the following elements:

- Status Indicators:** FLC, TC, a lightning bolt icon, and NC.
- Override:** A large display showing **Override 0.0%**.
- Achspositionen (Axis Positions):**
 - X - Achse: 0,000 mm
 - Y - Achse: 0,000 mm
 - Z - Achse: 0,000 mm
- Koordinatensystem (Coordinate System):** A dropdown menu.
- Override Slider:** A slider ranging from 0% to 100%, with a current value of 0,0.
- Start Button:** A green diamond button with a right-pointing arrow.
- Automatik Section:** Contains 'Start', 'Stop' (red circle with slash), and 'Reset' (two parallel lines) buttons.
- Function Keys (FN1-FN4):**
 - FN1: Empty button.
 - FN2: NPV icon (North-South arrow).
 - FN3: Laserparameter icon (yellow triangle with laser beam).
 - FN4: Einstellungen icon (gears).
- Program Counter:** A large empty box for program data, with G54, G01, and G00 listed below it.

A.1.5 Nullpunktverschiebung Dialog

Einzurichtendes Koordinatensystem

Nullpunkt Setzen absolut
 relativ zu

Position X

Position Y

Position Z

Verschiebungen

	G54	G55	G56	G57
▶ X-Achse	0,000	0,000	0,000	0,000
Y-Achse	0,000	0,000	0,000	0,000
Z-Achse	0,000	0,000	0,000	0,000

A.1.6 Lasersteuerung Dialog

LASER Status

-  Verbindung
-  Externe Ansteuerung
-  LASER ist ein
-  LASER bereit
-  Programm aktiv
-  Störung
-  Überwachung
-  Leistungsgrenze

Laser Parameter

Leistung	<input type="text" value="3500"/>	W	Material	<input type="text" value="Aluminium"/>
Pulsdauer	<input type="text" value="1,000"/>	ms	Materialstärke	<input type="text" value="1,0"/>
Frequenz	<input type="text" value="51,0"/>	Hz	<input type="button" value="Übernehmen"/>	
Auslastung	<input type="text" value="89,3"/>	%	<input type="checkbox"/> Laser Parameter aus NC Programm Verwenden	
Radius	<input type="text" value="0,3"/>	mm		
L Offset	<input type="text" value="0"/>	mm		
Vorschub	<input type="text" value="500"/>	mm/min		

A.1.7 Fehlermeldungen Dialog

Meldungen		Datum & Zeit	Fehlernummer	Fehlerart	Meldungstext	Hinweis
		11.07.2013 11:49:36		Hinweis	Reset	Reset durch Benutzer
		11.07.2013 11:47:35.882 ms		Fehler	Fehler beim Aktualisieren von VLC	Die Anforderung kann nicht verarbeitet werden, da der ...
		11.07.2013 11:45:44		Hinweis	Reset	Reset durch Benutzer
		11.07.2013 11:45:35.867 ms		Fehler	Fehler beim Auslesen der NC Zeilen	Ads-Error 0x6 : Target port could not be found.
		11.07.2013 11:45:35.663 ms		Fehler	Fehler beim Starten der SPS	Ads-Error 0x6 : Target port could not be found.
		11.07.2013 11:45:34.804 ms		Fehler	Fehler beim Fehlermanagement	Ads-Error 0x6 : Target port could not be found.
		11.07.2013 11:45:33.632 ms		Fehler	Fehler beim Verbindung herstellen	Ads-Error 0x6 : Target port could not be found.
		11.07.2013 11:45:34.788 ms	00	Hinweis	HMI gestartet	Die Benutzeroberfläche der Laserschneidemaschine wu...

A.1.8 HMI_Laser Dialog

Sie sind dabei die Benutzeroberfläche zu schließen.

Möchten Sie

die Maschine Herunterfahren

oder

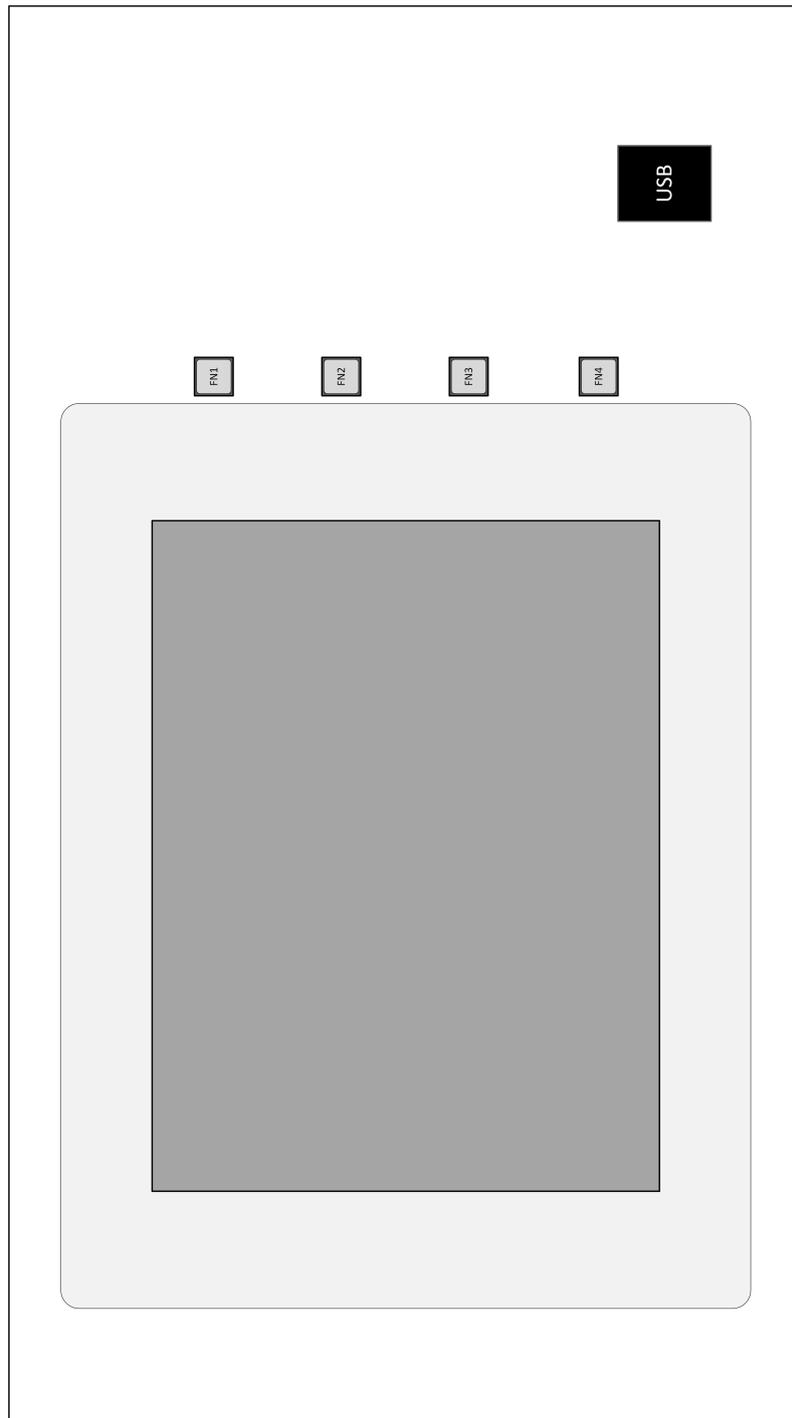
zu Windows zurückkehren

Benutzer:

Passwort:

Beenden Abbrechen

A.1.9 Oberes Panel



A.1.10 Unteres Panel

