

Projektarbeit 'Bauteileautomat'

Jan Krüger	Gregor Süß	Matthias Weißhaar
Friedrichstraße 24	August-Dürr Straße 9	Strümpfelbacher Straße 184
75417 Mühlacker	76133 Karlsruhe	71384 Weinstadt
Mtkl Nr: 37032	Mtkl Nr: 36496	Mtkl Nr: 30250

20. März 2014

Inhaltsverzeichnis

1. Einführung	5
1.1. Aufgabenstellung	5
1.2. Einleitung	5
1.3. Aufgabenverteilung	5
2. Mechanik	6
2.1. Greifer	6
2.1.1. Anforderungen	6
2.1.2. Realisierung des Greifers	6
2.1.3. Antrieb	6
2.1.4. Kugellager	9
2.1.5. Antrieb und Übersetzung Drehgelenk	10
2.1.6. Kugellager Drehgelenk	10
2.2. Achsen	10
2.2.1. Anforderungen X-Y-Z-Achse	10
2.2.2. Realisierung der Achsen	11
2.2.3. Antriebsmoment der Z-Achse	12
2.2.4. Linearführung Z-Achse	14
2.2.5. Kugelumlaufspindel der Z-Achse	14
2.2.6. Berechnungen der X-Achse	15
2.2.7. Linearführung der X-Achse	17
2.2.8. Berechnungen Y-Achse	17
2.2.9. Linearführung der Y-Achse	19
2.2.10. Kugelumlaufspindel der Y-Achse	19
2.3. Gestaltung des Rahmens	19
2.4. Daten und Komponenten des Automaten	21
2.4.1. X-Y-Z Achsen	21
2.4.2. Greifarm	21
2.5. Schmierung/Instandhaltung	22
2.6. Quellenangabe	22
3. Elektronik	23
3.1. Anforderungen an die Elektronik	23
3.2. Spannungsversorgung	24
3.3. Mikrocontroller	26
3.4. Digitale Eingänge	28
3.5. UART Schnittstelle	29
3.6. Netzwerkschnittstelle	30
3.7. Motorcontroller	30
3.8. Platinenlayout	32
3.9. Errata	32
4. Software	34
4.1. Verwendete Programmzbibliotheken	34
4.2. Die State-Machine	34
4.2.1. STATE_STARTUP	35
4.2.2. STATE_ERROR	35
4.2.3. STATE_JOB_HANDLING	36

4.2.4.	STATE_MSG_HANDLING	36
4.2.5.	STATE_UART_COM	36
4.2.6.	STATE_MOT_COM	37
4.2.7.	STATE_NET_COM	37
4.2.8.	STATE_IDLE	37
4.3.	Die Schnittstellenkommunikation	37
4.3.1.	Empfangen	38
4.3.2.	Senden	38
4.3.3.	Fehler	38
4.4.	Die Ringpuffer	38
4.4.1.	getChar	38
4.4.2.	putChar	39
4.4.3.	getRemBufSpace	39
4.4.4.	putMsg	39
4.4.5.	getMsg	39
4.4.6.	insertString	39
4.4.7.	clearString	39
4.5.	Das Job-System	39
4.5.1.	decodeMsg	40
4.5.2.	validateID	41
4.5.3.	checkCRC	41
4.5.4.	validateMsg	41
4.5.5.	addJob	41
4.5.6.	callJob	42
4.5.7.	jobDone	42
4.6.	Die Motorsteuerung	42
4.6.1.	Remote_Ctrl_Motor	42
4.6.2.	Ctrl_Motor	42
4.6.3.	getBox	43
4.6.4.	putBox	43
4.6.5.	EmptyBox	43
4.6.6.	RefillBox	43
A. Anhang Mechanik		44
A.1.	Drehgelenk und Greifer	44
A.2.	Z-Achse	69
A.3.	X-Achse	76
A.4.	Y-Achse	85
B. Anhang Elektronik		94
C. Anhang Software		99
C.1.	main.c	99
C.2.	main.h	112
C.3.	circular.c	113
C.4.	inc_circular.h	116
C.5.	msg.c	117
C.6.	inc_msg.h	124
C.7.	motor.c	125
C.8.	inc_motor.h	133

C.9. def.c	134
C.10.inc_def.h	134
C.11.startup.c	138
C.12.inc_startup.h	140

1. Einführung

1.1. Aufgabenstellung

Für die Bereitstellung von elektronischen Bauteilen für die Studierenden der Elektrotechnik ist ein Ausgabeautomat zu entwickeln und zu erstellen. Der Automat ermöglicht den Studenten der Fakultät EIT zu jeder Tageszeit den Zugriff auf Elektronikbauteile (für ihre Studienprojekte und für den privaten Gebrauch). Der Automat soll viele kleine Fächer enthalten, damit aus einem reichhaltigen Sortiment an Bauteilen ausgewählt werden kann. Die Auswahl der Bauteile kann am Automaten oder vorab per Intranet erfolgen. Vor der Ausgabe der Teile ist eine Authentifizierung mit dem Studientenausweis am Automaten erforderlich. Der Bauteilebestand und die Verrechnung der entnommenen Bauteile sollen über eine Datenbank protokolliert werden.

1.2. Einleitung

Der (Zeit-)Aufwand, der für die Erfüllung der Aufgabenstellung erforderlich ist, übersteigt das zeitliche Pensum einer Projektarbeit um ein vielfaches. Daher wurde die Aufgabenstellung in einer Gruppe bearbeitet und in drei Teile aufgeteilt. Ein besonderes Augenmerk lag auf einer klaren Trennung der Bereiche, damit eine enge Zusammenarbeit zwischen den Gruppenmitgliedern erreicht werden konnte. Im Hinblick auf die Ziele der Projektarbeit (Projektgruppen die voneinander abhängig sind, Arbeiten nach Pflichten- und Lastenheften, etc) erschien diese Vorgehensweise als die beste. Der Aufbau dieses Dokuments spiegelt die klare Trennung der Aufgabenbereiche wider. Es ist in die drei Bereiche Elektronik, Software und Mechanik gegliedert, die erst einmal für sich selbst stehen und zusammen die Gesamtheit der Projektarbeit darstellen. Ziel dieses Dokuments ist es, die Arbeit zu dokumentieren und dem Leser einen Ansatzpunkt zur Weiterentwicklung des Automaten zu geben.

1.3. Aufgabenverteilung

- Jan Krüger betreut den Bereich 'Elektronik'. Dieser Bereich beinhaltet die Konzeption, Entwicklung und den Aufbau der Steuerplatine, der Verkabelung des Automaten und das Testen der Komponenten. Weiterhin gehört die Absprache der Bedürfnisse für den mechanischen Aufbau mit dem Bereich 'Software' zu seinem Aufgabengebiet.
- Gregor Süß betreut den Bereich 'Software'. Er ist zuständig für die Konzeptionierung und Programmierung des Automaten.
- Matthias Weißhaar betreut den Bereich 'Mechanik'. Dieser Bereich beinhaltet die Konzeption, Entwicklung und den Aufbau der mechanischen Komponenten, wie auch den Funktionstest derselben.

2. Mechanik

Dieses Kapitel dokumentiert die mechanische Funktionsweise und die Berechnung der zugehörigen Komponenten. Der Automat soll ca. 1000 Boxen enthalten, um eine große Auswahl zu gewährleisten. Damit dies in einer kompakten Bauweise möglich ist, werden drei unterschiedlich große Boxen verwendet. Die Boxen befinden sich in verschiedenen Schächten. Ein Schacht besteht aus jeweils zwei L-Profilen, die mit Schrauben an den Aluminiumprofilen der Rückwand befestigt werden. Die Größe der Schächte ist jederzeit änderbar und somit für jede Boxengröße variabel einstellbar. Ein Greifer, der auf X-Y-Z-Achsen verschiebar ist, fährt den ausgewählten Schacht an und greift anschließend die entsprechende Box. An der Seite des Automaten ist ein Rutsche angebracht, die zur Ausgabe der Teile dient. Der Greifer fährt diese an und kann durch ein Drehgelenk die Box um 180° drehen und das gewünschte Teil auswerfen.

2.1. Greifer

2.1.1. Anforderungen

Der Greifer hat den sicheren Transport der Boxen zwischen den Schächten und der Rutsche zu gewährleisten. Er muss ein Drehgelenk beinhalten, das sich um 180° drehen lässt. Weiter sind vom Greifer zwei verschieden große Boxen mit einem maximalen Gewicht von 1000g zu befördern. Damit die Ausgabe der Teile nicht zu lange dauert, muss die Geschwindigkeit des Greifvorgangs entsprechend hoch sein.

2.1.2. Realisierung des Greifers

Der Greifmechanismus erfolgt über zwei 20cm lange Greifarme aus Aluminium, wobei ein Greifarm fest verbaut ist und der andere über eine Trapezgewindespindel verfahrbar ist. Für den Antrieb der Spindel werden zwei Zahnräder verwendet, welche mit einem Schrittmotor verbunden sind. Damit der bewegliche Greifarm nicht verkantet, wird er über eine zusätzliche Welle geführt. Moosgummi auf dem Greifarm erhöht die Haftkraft zwischen den Boxen und den Greifarmen. Der Drehmechanismus erfolgt ebenfalls über zwei Zahnräder und einen Schrittmotor. Ein Zahnrad ist auf einer Welle befestigt, die auf einer Seite fest mit dem Greifer verbunden ist und auf der anderen Seite in einem Rillenkugellager gelagert ist. Durch das zweite Zahnrad, das durch einen Schrittmotor angetrieben ist, kann die Welle mit dem Greifer rotiert werden.

2.1.3. Antrieb

Alle relevanten Eigenschaften (Presskraft, Verfahrensgeschwindigkeit und Schrittlänge) sind abhängig vom Schrittmotor des Greifmechanismus. Für die richtige Dimensionierung des Schrittmotors sind entsprechende Berechnungen erforderlich.

- Haltekraft des Greifarms

Das Drehmoment des Motors muss den sicheren Transport einer Box mit einem Gewicht von 1000g gewährleisten. Zur Berechnung des Drehmoments wird die Presskraft benötigt, die der

dynamische Greifarm auf die Box ausübt. Hier handelt es sich um einen Erfahrungswert, da verschiedene Berechnungsvariablen (z.b. Reibungskoeffizient von Polystrol und Moosgummi) nicht vorhanden sind. Der Wert wurde daher auf $F = 50N$ ausgelegt und bewußt höher gewählt, damit die Boxen auch bei kleinen Stößen und Vibrationen nicht aus dem Greifarmen rutschen können.

Durch die Umwandlung von einer Drehbewegung in eine Längsbewegung entsteht ein Wirkungsgrad, welcher durch den Steigungswinkel des Trapezgewindes und den Gewindereibungswinkel berechnet werden kann.

Den Steigungswinkel erhält man durch:

$$\tan(\alpha) = \frac{P}{d \cdot \pi} = \frac{20mm}{4mm \cdot \pi} = \frac{5}{\pi}$$

mit P: Spindelsteigung

d: Spindeldurchmesser

α : Steigungswinkel

Für ISO-Trapezgewinde mit geschmierter Mutter und einem Reibwert von $\mu = 0.04$ erhält man den Gewindereibungswinkel mit:

$$\rho = \mu \cdot 1.07 = 0.04 \cdot 1.07 = 0.043$$

mit ρ : Gewindereibungswinkel

μ : Reibwert

Mit dem Steigungswinkel und dem Gewindereibungswinkel kann nun der Wirkungsgrad berechnet werden:

$$\eta = \frac{\tan(\alpha)}{\tan(\alpha + \rho)} = \frac{\tan\left(\frac{5}{\pi}\right)}{\tan\left(\frac{5}{\pi} + 0.043\right)} = 3.08$$

mit η : Wirkungsgrad

Das Antriebsdrehmoment der Spindel ergibt sich aus der Presskraft, der Steigung und dem Wirkungsgrad des Trapezgewindes:

$$M = \frac{F \cdot P}{1000 \cdot 2\pi \cdot \eta} = \frac{50N \cdot 4mm}{1000 \cdot 2\pi \cdot 3.08} = 0.01Nm$$

mit M: Antriebsmoment

F: Presskraft

Auf das Ergebnis müssen die Verluste durch die Lagerung und der Wirkungsgrad der Übersetzung aufgeschlagen werden. Zur Berücksichtigung dieser Verluste sollte die ausgewählte Leistung des Antriebs um 60% bis 100% über dem errechneten Wert liegen.

Mit dem Übersetzungsverhältnis der Zahnräder und den Verlusten des Greifmechanismus erhält man das geforderte Drehmoment des Motors durch:

Übersetzungsverhältnis der Zahnrräder:

$$i = \frac{z_2}{z_1} = \frac{30}{60} = 0.5$$

das Antriebsmoment des Schrittmotors:

$$M_{Motor} = M_{Spindel} \cdot i \cdot R = 0.01Nm \cdot 0.5 \cdot 1.6 = 8Nm$$

mit i: Übersetzungsverhältnis

R: Reibungsverluste

- Schrittlänge

Die Schrittlänge ist der kleinstmögliche Weg, den der Greifarm zurücklegen kann. Bei einem Schritt des Motors wird dieser Weg zurückgelegt. Alle anderen Wege ergeben sich als ein Vielfaches der Schrittlänge. Das weiche Moosgummi auf den Greifarmen und die Elastizität der Boxen erfordern eine feine Abstimmung der Presskraft der Greifarme. Für eine anwendungsgerechte Einstellung ist eine kleine Schrittlänge erforderlich.

Mit dem Übersetzungsverhältnis kann die Schrittlänge des Greifers berechnet werden:

$$S = \frac{P}{\frac{360^\circ}{\alpha}} \cdot i = \frac{4mm}{\frac{360^\circ}{1.8^\circ}} \cdot 0.5 = 0.01mm$$

mit α : Schrittwinkel des Elektromotors

P: Steigung der Trapezgewindespindel

Die geringe Schrittlänge ist ausreichend um die benötigte Presskraft einzustellen. Toleranzen des Gewindes oder der Schritte des Schrittmotors können vernachlässigt werden.

- Geschwindigkeit des Greifarms

Um einen Schacht anzufahren müssen die Greifarme schon in der richtigen Schachtgröße positioniert werden. Die Positionierung findet während der Anfahrt an die jeweiligen Schächte statt. Somit muss nur noch das tatsächliche Greifen in einem gesonderten Arbeitsgang stattfinden. Hier ist der Verfahrensweg so klein $s < 5mm$, dass die Geschwindigkeit größtenteils irrelevant ist.

2.1.4. Kugellager

Um das Kugellager richtig zu wählen und eine lange Lebensdauer zu garantieren, wird die nominelle Lebensdauer nach ISO 281 berechnet:

$$L_{10h} = \left(\frac{C}{P}\right)^p = \left(\frac{627N}{10N}\right)^3 = 246492$$

mit p: Lebensdauerexponent (bei Kugellager p=3)

C: Dynamische Tragzahl in Kn (aus Lagertabelle des Herstellers)

P: Dynamische äquivalente Belastung

L_{10} : Lebensdauer in Millionen Umdrehungen bei 10% Ausfallwahrscheinlichkeit

Die nominelle Lebensdauer gibt an, wieviele Umdrehungen 90% der Lager erreichen oder überschreiten bis die ersten Anzeichen von Werkstoffermüdung auftreten. Mit mehreren Millionen Umdrehungen bei 10% Ausfallwahrscheinlichkeit ist für unsere Lager eine hohe Lebensdauer zu erwarten. Das Lager darf nicht zu groß dimensioniert werden, da ansonsten die Belastung der Lager zu gering wäre und kein Rollen der Wälzläger sondern Gleiten stattfindet. Gleitreibung sorgt für höheren Verschleiß und verkürzt daher die Lebensdauer.

Eine Lebensdauerberechnung ist für unsere Anwendung überflüssig, da die Kugellager nur sehr kleine Kräfte aufnehmen. Zur Vervollständigung wurde diese jedoch trotzdem durchgeführt.

2.1.5. Antrieb und Übersetzung Drehgelenk

Das Drehgelenk wird ebenfalls über einen Schrittmotor angetrieben. Da hier nur eine Drehung von 180° verlangt wird und der Motor ein hohes Drehmoment benötigt, wurde bei den Zahnrädern eine Übersetzung eingebaut.

Übersetzungsverhältnis der Zähnrräder:

$$i = \frac{z_2}{z_1} = \frac{14}{70} = 0.2$$

mit z_1 : Zähnezahl₁

z_2 : Zähnezahl₂

Durch das Übersetzungsverhältnis von 0.2 kann nun ein kleinerer Schrittmotor verwendet werden, dieser benötigt jedoch für eine Umdrehung die 5-fache Schrittanzahl. Auch hier müssen wieder Verluste eingerechnet werden und der Motor muss entsprechend größer dimensioniert werden.

Das Drehmoment berechnet sich zu:

$$M = F \cdot s \cdot i \cdot R = 10N \cdot 70mm \cdot 0.2 \cdot 1.6 = 22.4Ncm$$

mit i: Übersetzungsverhältnis

R: Reibungsverluste

s: Länge(Kugellager-Greiferarm)

F: Gewichtskraft des Greifers

2.1.6. Kugellager Drehgelenk

Für die Ausgabe eines Bauteils rotiert das Kugellager 180° und anschließend wieder 180° zurück. Da nur eine Umdrehung pro Ausgabe am Kugellager notwendig ist, muss keine nominelle Lebensdauer für das Kugellager ausgerechnet werden. Es ist trotzdem wichtig, dass unser Kugellager mit einer mittleren Belastung betrieben wird. Das Verhältnis von dynamischer Tragzahl zu dynamischer äquivalenter Belastung sollte somit zwischen 8 und 15 liegen. Das Kugellager wird durch das komplette Gewicht von Greifer und Box radial belastet, welches mit 5kg ausgelegt wird. Eine axiale Belastung erfährt der Greifer nur durch die Beschleunigung der Z-Achse, welche hier gering ausfällt und somit vernachlässigt werden kann.

2.2. Achsen

2.2.1. Anforderungen X-Y-Z-Achse

Um eine hohe Auswahlmöglichkeit im Automaten zu erreichen, muss dieser möglichst viele Boxen enthalten. Deshalb ist die größtmögliche Anzahl an Schächten im Automaten unterzubringen. Die

L-Profile, welche die Schächte bilden, müssen sehr eng aneinander gereiht werden. Das Spiel zwischen den Schächten ist somit sehr gering und die Achsen müssen die jeweiligen Schächte präzise anfahren. Die Ausgabezeit des Produktes ist für den Kunden wichtig und muss berücksichtigt werden. Damit der Kunde nicht zu lange auf sein gekauftes Bauteil warten muss, sind Verfahrensgeschwindigkeit und Beschleunigung der Achsen ebenfalls zu berücksichtigen und richtig zu dimensionieren. Die Achsen müssen platzsparend im Automaten untergebracht werden und dürfen die Maße des Automaten nicht unnötig vergrößern. Da die Verfahrenswege der Achse unterschiedlich lang sind, müssen die Berechnungen einzeln stattfinden. Der Automat soll eine lange Lebensdauer erreichen.

2.2.2. Realisierung der Achsen

Für den Vorschub der Achsen wurden Kugelumlaufspindeln verwendet, die jeweils mit zwei Linearführungen ein Linearmodul bilden. Die zwei Profilschienenführungen tragen die Masse des Greifmechanismus und garantieren die Einhaltung der linearen Bewegungsrichtung. Die Kugelumlaufspindel hat eine geringe Schrittänge, welche sich im μm Bereich einreihen, was für den Bauteileautomat ausreichend ist. Durch die geringe Reibung der Kugelumlaufspindel wird ein hoher Wirkungsgrad der Motoren und geringer Verschleiß erreicht, was zu einer hohen Lebensdauer der Achsen führt. Die Positioniergenauigkeit der Kugelumlaufspindel ist hoch und entspricht den Anforderungen unseres Automaten.

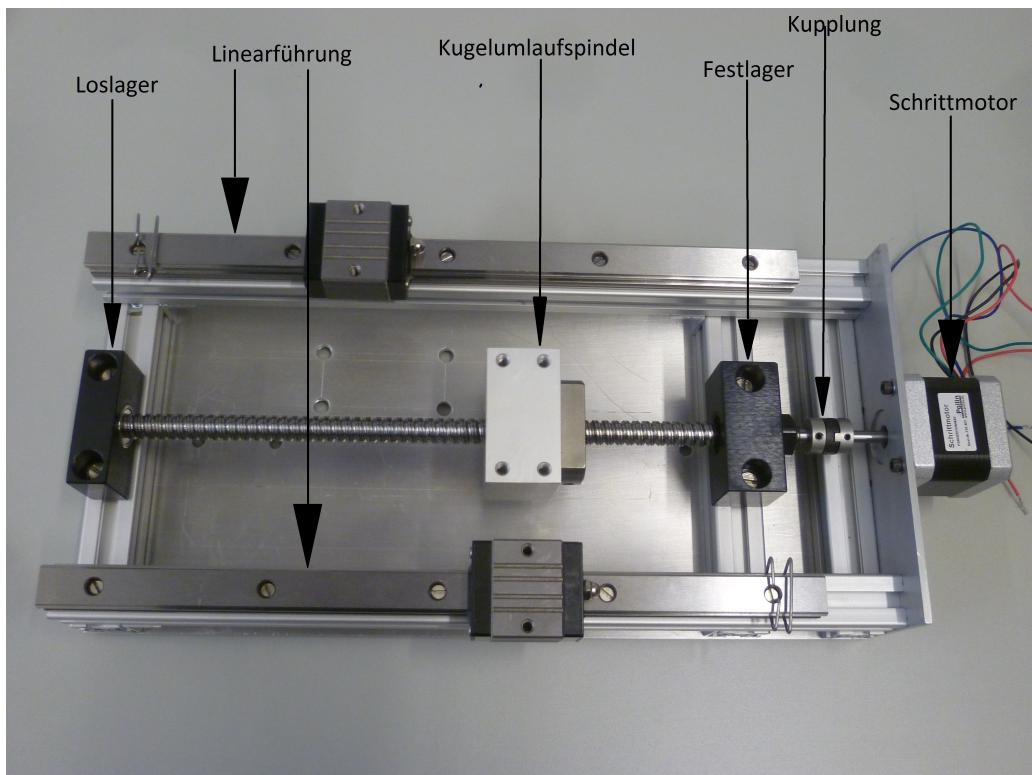


Abbildung 1: Z-Achse

Der Antrieb der Achsen erfolgt über einen Schrittmotor, welcher über eine Oldham-Kupplung mit der Welle verbunden ist. Die Kupplung überträgt das Drehmoment des Motors an die Welle und kann geringe parallele Versätze ausgleichen, die durch Fertigungsfehler entstehen können.

2.2.3. Antriebsmoment der Z-Achse

Die Bewegung der Z-Achse muss, wie der Greifmechanismus, getrennt von den anderen Bewegungen stattfinden. Die Beschleunigung und Verfahrensgeschwindigkeit darf daher nicht zu gering ausfallen. Trotz Reibung und Wirkungsgrad der Kugelumlaufspindel muss das Antriebsmoment des Schrittmotors so gewählt werden, dass eine akzeptable Verfahrenszeit erreicht werden kann.

- **Reibungskraft**

Die Reibungskraft, die in den Linearführungen auftritt, kann mit der folgenden Formel berechnet werden:

$$F_R = \mu \cdot m \cdot g + f = 0.003 \cdot 5 \text{kg} \cdot 9.81 \frac{\text{m}}{\text{s}^2} + 5 \text{N} = 0.1 \text{N} + 5 \text{N} = 5.1 \text{N}$$

mit μ : Dynamischer Reibungskoeffizient

m : Masse des Greifers

g : Erdbeschleunigung

f : Dichtungsbeständigkeit

Aus der Rechnung ist ersichtlich, dass die Masse des Greifers kaum eine Rolle auf die Reibungskraft hat. Der Großteil der Reibung entsteht durch die verschiedenen Dichtungselemente der Linearführung (z.b. Dichtungslippe), welche nicht von der Last abhängig sind.

- **Beschleunigung**

Der maximal verfahrbare Weg der Z-Achse beträgt $s = 186 \text{mm}$. Die Hälfte dieses Weges soll in einer Zeit von maximal $t = 3 \text{s}$ zurückgelegt werden. Somit ist auch gegeben, dass die Z-Achse auf der anderen Hälfte wieder gebremst werden kann.

Beschleunigung: $\ddot{x}_a = a$

Geschwindigkeit: $\dot{x}_v = a \cdot t$

Verfahrensweg: $x_s = 0.5 \cdot a \cdot t^2$

Aus dem Verfahrensweg und der Zeit ergibt sich eine konstante Beschleunigung von:

$$a = \frac{2 \cdot s}{t^2} = \frac{2 \cdot 93 \text{mm}}{9 \text{s}^2} = 21 \frac{\text{mm}}{\text{s}^2}$$

Die notwendige Kraft für die Beschleunigung erhält man aus der Masse des Greifers:

$$F_a = m \cdot a = 5\text{kg} \cdot 0.021 \frac{\text{m}}{\text{s}^2} = 0.11\text{N}$$

- **Antriebsmoment**

Mit der Reibungskraft, der notwendigen Kraft für die Beschleunigung und einem Wirkungsgrad von $\eta = 0.9$ der Kugelumlaufspindel kann nun das erforderliche Antriebsmoment berechnet werden:

$$M = \frac{(F_R + F_a) \cdot P}{2\pi \cdot \eta} = \frac{(5.1\text{N} + 0.11\text{N}) \cdot 4\text{mm}}{2\pi \cdot 0.9} = 3.69\text{Nm}\text{m}$$

mit F_R : Reibungskraft

F_a : Kraft für Beschleunigung

P: Steigung Kugelumlaufspindel

η : Wirkungsgrad Kugelumlaufspindel

Auch hier müssen noch die Verluste durch die Lagerung der Kugelumlaufspindel hinzugerechnet werden. Zusätzlich sollte noch eine Sicherheit mit eingerechnet werden. Insgesamt sollte das Antriebsmoment um das 1.5fache erhöht werden.

$$M_{\text{Schrittmotor}} = M \cdot 1.5 = 3.69\text{Nm}\text{m} \cdot 1.5 = 5.54\text{Nm}\text{m}$$

Der gewählte Schrittmotor muss somit ein Drehmoment von mindestens $M_{\text{Schrittmotor}} = 5.5\text{Nm}\text{m}$ aufweisen.

2.2.4. Linearführung Z-Achse

- **Momente**($M_A/M_B/M_C$)

Die zulässigen statischen Momente der Linearführungen dürfen nicht überschritten werden, damit ein sauberer Lauf und eine lange Lebensdauer garantiert wird. Zudem wird eine Sicherheit von 3 eingebaut, damit Mehrbelastungen durch dynamische Bewegungen nicht zu einer Überlast führen. Die statischen Momente M_B und M_C sind nicht vorhanden oder sehr gering und können vernachlässigt werden. Das kritische Moment M_A ergibt sich durch die Gewichtskraft des Greifers und der Länge des Greifers. Das Moment wird auf zwei Linearführungen aufgeteilt und muss eine Sicherheit beinhalten.

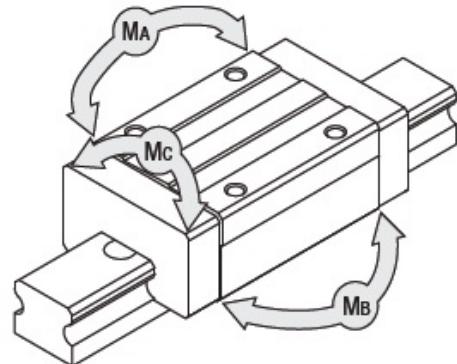


Abbildung 2: Momente

$$M_A > \frac{m \cdot g \cdot l \cdot s}{2} = \frac{5kg \cdot 9.81 \frac{m}{s^2} \cdot 0.1m \cdot 3}{2} = 7.5Nm$$

mit m: Masse

g: Erdbeschleunigung

l: Länge

s: Sicherheit

Das zulässige statische Moment der beiden Linearführungen muss mindestens 7.5Nm betragen.

- **Nennnutzungsdauer**

Die Nennnutzungsdauer gibt den Verfahrensweg an, den 90% der Linearführungen desselben Typs unter identischen Bedingungen zurücklegen können ohne dabei Schäden zu erleiden. Die Linearführungen des Bauteileautomats werden unter fast perfekten Bedingungen betrieben (kein Schmutz, keine hohen Temperaturen, niedrige Last). Der Anspruch der Lebensdauer für den Bauteileautomat wird damit weit übertragen und eine Berechnung wird überflüssig. Dies gilt auch für die X-Y-Achsen.

2.2.5. Kugelumlaufspindel der Z-Achse

- **Zulässige Axiallast**

Eine Berechnung der zulässigen Axiallast ist für die X-Z-Achsen nicht notwendig. Die Axialkräfte (Beschleunigungskraft) sind zu gering, um die zulässige Axiallast zu überschreiten.

- **Zulässige Drehzahl**

Die Zulässige Drehzahl der Kugelumlaufspindel wird durch die kritische Drehzahl festgelegt. Die kritische Drehzahl ist diejenige Drehzahl, bei der ein Kugelgewindetrieb Resonanzerscheinungen zeigt. Sie ist abhängig vom Nenndurchmesser, der Spindellänge sowie der Lagerungsart.

$$N_c = g \cdot \frac{d}{l^2} \cdot 10^7 (\text{min}^{-1}) = 15.1 \frac{\text{mm}}{\text{min}} \cdot \frac{7.8\text{mm}}{185\text{mm}^2} \cdot 10^7 = 573 \frac{1}{\text{s}}$$

mit N_c : Kritische Drehzahl

d: Spindel-Kerndurchmesser

g: Faktor für Lagerart

l: Eingespannte Spindellänge

Die Drehzahl der Kugelumlaufspindel der Z-Achse wird weit unter der zulässigen Drehzahl betrieben und muss somit nicht berücksichtigt werden. Auch die X-Y-Achsen werden weit unter den zulässigen Drehzahlen betrieben.

- **Nennnutzungsdauer**

Auch hier ist die Lebensdauer sehr hoch, da das Gewicht größtenteils von den Linearführungen getragen wird und auf unsere Kugelumlaufspindel kaum Kräfte wirken. Nur die Beschleunigungskraft wirkt axial auf die Spindel, diese ist jedoch so klein das eine Berechnung überflüssig wird. Dies gilt auch für die X-Achse.

2.2.6. Berechnungen der X-Achse

Die X-Achse ist für die horizontale Anfahrt der Schächte zuständig und muss mit ca. 1m Länge auch eine ausreichend hohe Geschwindigkeit und Beschleunigung aufweisen. Auch hier ist keine Hubkraft notwendig, da das gesamte Gewicht radial auf die Lineareinheit wirkt.

- **Beschleunigung und Geschwindigkeit**

Die Berechnung der Z-Achse zeigt, dass die Beschleunigung kaum eine Auswirkung auf das Antriebsmoment hat. Somit kann die Beschleunigung vernachlässigt werden, da der Schrittmotor bewusst höher gewählt wird, um im Betrieb Spiel zu haben und ggf. die Verfahrensgeschwindigkeit zu erhöhen.

- **Antriebsmoment**

Das Antriebsmoment der X-Achse folgt auf der gleichen Berechnung wie das Antriebsmoment der Z-Achse. Da die Masse der Z-Achse noch mitverschoben wird, ändert sich die Gewichtskraft. Zudem besitzt die Kugelumlaufspindel eine ander Steigung als die Z-Achse. Somit ergibt sich:

$$F_R = m \cdot g \cdot \mu + f = 10\text{kg} \cdot 9.81 \frac{\text{m}}{\text{s}^2} \cdot 0.005 + 5\text{N} = 5.5\text{N}$$

Das Antriebsmoment erhält man durch:

$$M = \frac{F_R \cdot P}{2\pi \cdot \eta} = \frac{5.5N \cdot 10mm}{2\pi \cdot 0.9} = 9.7Nm$$

Aufgrund von Verlusten sollte auch hier der Motor höher dimensioniert werden.

$$M_{Schriftmotor} = M \cdot 1.5 = 9.7Nm \cdot 1.5 = 14.6Nm$$

- **Durchbiegung der X-Achse**

Durch die Gewichtskraft der Z-Achse und des Greifers kann es zu einer Durchbiegung der X-Achse kommen. Die Durchbiegung muss berechnet werden und ggf. bei der Justierung des jeweiligen Schachtes berücksichtigt werden. Es wird die maximale Biegung der Aluminiumprofile berechnet, auf denen die X-Achse verläuft.

Die maximale Durchbiegung erhält man mit:

$$f = \frac{F \cdot l^3}{(48 + \frac{29 \cdot m}{l}) \cdot E \cdot I} = \frac{5kg \cdot 9.81 \frac{m}{s^2} \cdot 0.55m^3}{(48 + \frac{29 \cdot 47mm}{820mm}) \cdot 70000 \frac{N}{mm^2} \cdot 0.9cm^4} = 0.86mm$$

mit E: Elastizitätsmodul

I: Flächenmoment

l: Länge der X-Achse

F: Gewichtskraft (Greifer, Z-Achse)

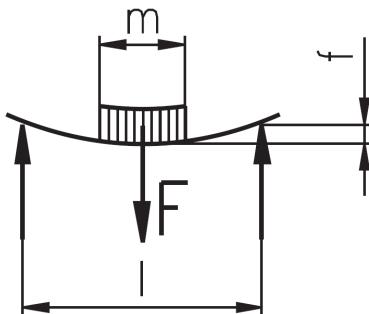


Abbildung 3: Z-Achse

Mit einer maximalen Durchbiegung von 0.86mm müssen die Anfahrtswege der Schächte individuell justiert werden und die Schritte der Schrittmotoren auf die jeweiligen Schächte angepasst werden.

2.2.7. Linearführung der X-Achse

- **Momente**($M_A/M_B/M_C$)

Die statischen Momente M_B und M_C sind nicht vorhanden oder sehr gering und können vernachlässigt werden. Das kritische Moment M_A ergibt sich durch die Gewichtskraft des Greifers, die Gewichtskraft der Z-Achse und der Breite des Greifers. Auch hier beinhaltet das Linearmodul zwei Linearführungen. Somit wird das Moment aufgeteilt und kann halbiert werden.

$$M_A > \frac{m \cdot g \cdot l \cdot s}{2} = \frac{10kg \cdot 9.81 \frac{m}{s^2} \cdot 0.5m \cdot 3}{2} = 29.5Nm$$

mit m: Masse

g: Erdbeschleunigung

l: Länge

s: Sicherheit

Das zulässige statische Moment der beiden Linearführungen muss mindestens 29.5Nm betragen.

2.2.8. Berechnungen Y-Achse

Für die Y-Achsen sind bestimmte Verfahrensgeschwindigkeiten erforderlich, da bei einer Länge von 1,7m die Anfahrtzeit des gewünschten Schachtes von Bedeutung ist. Der Schrittmotor muss das Haltemoment und das Beschleunigungsmoment aufnehmen können.

- **Beschleunigung und Geschwindigkeit**

Die Y-Achse soll die Hälfte ihrer Hublänge $s = 0.85m$ in $t = 2s$ verschieben. Das Antriebsmoment ist somit auch ausreichend um die Konstruktion in der anderen Hälfte der Hublänge wieder zum Stillstand zu bringen. Die maximale Verfahrenszeit der Y-Achse beträgt somit $t = 4s$.

Beschleunigung: $\ddot{x} = a$

Geschwindigkeit: $\dot{x} = a \cdot t$

Hublänge: $x = 0.5 \cdot a \cdot t^2$

Aus der Hublänge und Zeit ergibt sich eine konstante Beschleunigung von:

$$a = \frac{2 \cdot x}{t^2} = \frac{2 \cdot 0.85m}{(2s)^2} = 0.425 \frac{m}{s^2}$$

mit x: Hublänge

t: Zeit

- **Antriebsmoment**

Das erforderliche Antriebsmoment für den Schrittmotor ergibt sich aus dem Haltemoment und der benötigten Kraft der Beschleunigung. Da die gesamte Konstruktion durch zwei Y-Achsen angetrieben wird, teilt sich das Antriebsmoment auf die zwei Schrittmotoren auf und kann halbiert werden.

Die Kraft ergibt sich aus der zu tragenden Masse, welche auf 15kg ausgelegt wird (X-Achse, Z-Achse, Greifer, Box), und aus den Beschleunigungen.

$$F = m \cdot (a + g) = 15\text{kg} \cdot (0.425 \frac{\text{m}}{\text{s}^2} + 9.81 \frac{\text{m}}{\text{s}^2}) = 153.5\text{N}$$

Mit der Steigung der Kugelumlaufspindel, dem Wirkungsgrad und der Kraft erhält man das Antriebsmoment.

$$M = \frac{F \cdot P}{2\pi \cdot \eta} = \frac{153.5\text{N} \cdot 10\text{mm}}{2\pi \cdot 0.9} = 271.5\text{Nmm}$$

mit a: Beschleunigung

g: Erdbeschleunigung

m: Masse

P: Steigung Kugelumlaufspindel

Auch hier muss eine Sicherheit von 20% eingerechnet werden, da Verluste durch Lagerung und Reibung entstehen. Man erhält somit ein Antriebsmoment je Schrittmotor von:

$$MM_{\text{Schrittmotor}} = \frac{M \cdot \eta_{\text{Verlust}}}{2} = \frac{271.5\text{Nmm} \cdot 1.2}{2} = 162.9\text{Nmm}$$

- **Rastmoment**

Das Rastmoment definiert das maximale Drehmoment, das am stromlosen Motor anliegen kann, ohne dass er sich dreht. Dies ist besonders wichtig bei Stromausfall. Da die Kugelumlaufspindel keine Selbsthemmung besitzt würde die Y-Achse nach unten stürzen. Durch das Rastmoment $M_{\text{Rast}} = 0.068\text{Nm}$ der zwei Schrittmotoren könnte die Y-Achse abgebremst werden und nach einiger Zeit zum Stillstand kommen. Da die Masse des Greifers und der Achsen unter dem angenommenen Wert liegen wird und die Verluste nicht genau bestimmt werden können, muss dies noch im Betrieb getestet werden und möglicherweise eine zusätzliche Sicherheit bei Stromausfall eingebaut werden.

2.2.9. Linearführung der Y-Achse

- **Momente**($M_A/M_B/M_C$)

Auch hier sind die statischen Momente M_B und M_C nicht vorhanden oder sehr gering und können vernachlässigt werden. Das kritische Moment M_A ergibt sich durch die Gewichtskraft des Greifers, die Gewichtskraft der Z-Achse und X-Achse und der halben Länge der X-Achse. Durch die zwei Linearmodule mit jeweils zwei Linearführungen teilt sich das Moment auf die vier Schlitzen der Linearführung auf.

$$M_A > \frac{m \cdot g \cdot l \cdot s}{4} = \frac{15\text{kg} \cdot 9.81 \frac{\text{m}}{\text{s}^2} \cdot 0.5\text{m} \cdot 3}{4} = 55.2\text{Nm}$$

2.2.10. Kugelumlaufspindel der Y-Achse

- **Zulässige Axiallast**

Die zulässige Axiallast ist eine Maximallast, die einschließlich einer Sicherheitstoleranz vor dem Abknicken einer Gewindewelle aufgebracht werden darf. Die Y-Achse ist senkrecht im Automaten eingebaut und wird daher stark axial belastet. Somit muss die zulässige Axiallast überprüft werden und darf nicht überschritten werden.

$$P = m \cdot \frac{d^4 \cdot 10^4}{l^2} = 10 \frac{\text{kg} \cdot \text{mm}^2}{\text{s}^2} \cdot \frac{16.3^4 \text{mm}^4}{1700^2 \text{mm}^2} \cdot 10^4 \text{m} = 2.44\text{kN}$$

mit P: Zulässige Axiallast

m: Faktor basierend auf Lagerart

d: Kerndurchmesser

l: Abstand zwischen den Belastungspunkten

Jede Achse kann somit ein Axiallast von maximal 2.44kN aufnehmen. Dies ist ausreichend und wird nicht überschritten.

2.3. Gestaltung des Rahmens

Der Aufbau des Rahmens besteht aus Aluminiumprofilen. Diese bilden große Gestaltungsspielräume und einen schnellen Aufbau. Die Aluminiumprofile können nachträglich schnell demontiert werden und kleine Änderungen am Automaten sind ohne großen Aufwand zu realisieren. Die L-Profile der Schächte, welche auf die Aluminiumprofile montiert sind, können jederzeit verändert werden. Somit ist die Größe der Schächte sehr variabel. Sobald der Automat eine höhere Kapazität von großen Boxen benötigt, können die schmalen Schächte zu Breiten Schächten umgebaut werden und umgekehrt.

Der Automat ist mit Kunststoffplatten verkleidet. Um den Studenten die Funktionsweise des Automaten zu zeigen, wird an der Vorderseite ein Teil mit Plexiglas ausgefüllt. Zur Bedienung des Automaten

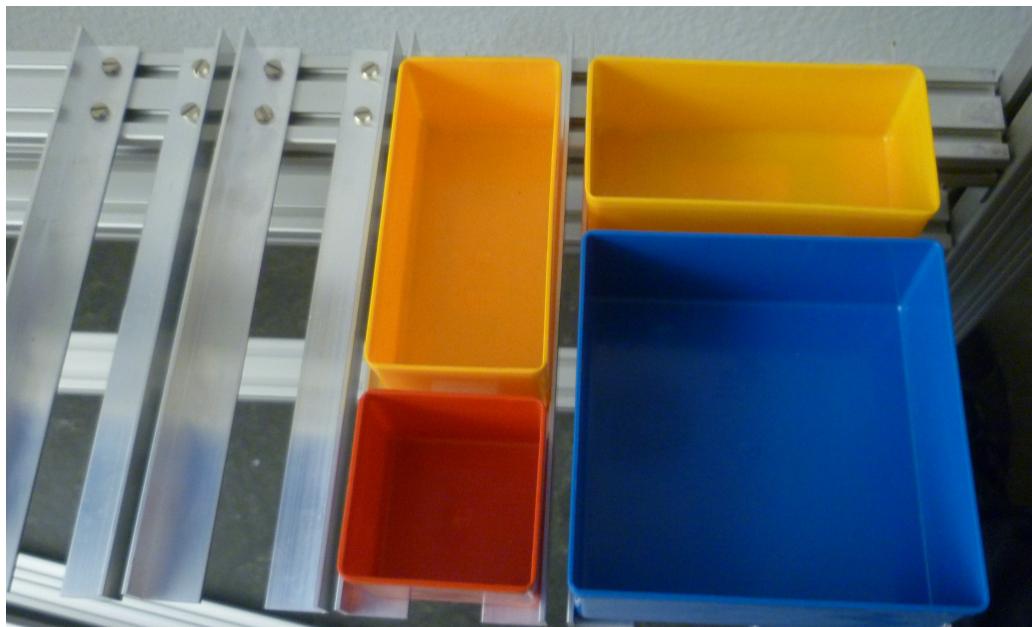


Abbildung 4: Schäfte für die Lagerung der Boxen

befindet sich an der Vorderseite ein Touchscreen. Die Bezahlung der entnommenen Teile erfolgt per Kartenlesegerät, welches noch nachträglich am Automat montiert wird. Rechts vom Automaten wird eine kleine Tür mit Schloss eingebaut. Somit kann der Automat schnell befüllt werden, ohne die Verkleidung abschrauben zu müssen.

Unter der Tür befindet sich die Ausgabe, wo die Studenten die gekauften Teile entnehmen. Die gekauften Teile werden mit einer Rutsche zum Ausgabefach befördert. Die Elektronik befindet sich am Boden des Automaten. Der Greifer kann hier nicht hinfahren, da seitlich die Motoren der Y-Achse verbaut sind. Der Platz kann daher für die Elektronik benutzt werden.

2.4. Daten und Komponenten des Automaten

Die wichtigsten Daten des Bauteileautomaten im Überblick:

- Grundmaße:** Die Grundmaße des Bauteileautomaten in Höhe, Breite und Tiefe sind ca.: 2000mmx1000mmx7500mm
- Gewicht:** Das Gesamtgewicht des Automaten ohne Inhalt der Boxen beträgt ca. 75kg.
- Kapazität:** Der Automat kann ca. 1000 rote Boxen zur selben Zeit lagern.
- Ausgabezeit:** Der Automat braucht durchschnittlich ca. 10s, um ein gewünschtes Teil zur Ausgabe zu befördern.

Die Daten der drei verschiedenen Boxen:

Farbe	Länge x Höhe x Breite	Gewicht	Werkstoff
■ rote Box	54 x 45 x 54	0.1kg	Polystyrol (PS)
■ gelbe Box	108 x 45 x 54	0.22kg	Polystyrol (PS)
■ blaue Box	108 x 45 x 108	0.35kg	Polystyrol (PS)

2.4.1. X-Y-Z Achsen

Die zugehörigen Komponenten der X-Y-Z Achsen:

Achse	Gewindetriebe	Kupplung	Linearführungen	Motor
Z-Achse	BSSC1004-280-SWC5	MCO15-5-6	SSVR24-280	PSM42BYGHW603
X-Achse	BSSC1510-835-SWC7	MCO20-6.35-10	SVRN28-820	ST5918X1008
Y-Achse	BSSC2010-1670-SWC7	MCO20-6.35-12	SVRL28-1700	ST5918L1008

Achse	Führungswagen	Loslager	Festlager
Z-Achse	BNFA1004C-30	BUV8	BSV8S
X-Achse	BNFA1510C-30	BUV12	BSV12
Y-Achse	BNFA2010C-30	BUV15	BSV15

Alle Komponenten für die Achsen sind von *MiSUMi*. Die verschiedene Maße und die technischen Eigenschaften der einzelnen Komponenten können im Anhang oder unter www.misumi-europe.de nachgeschlagen werden.

2.4.2. Greifarm

Komponenten des Greifarms:

	Festlager	Loslager	Motor	Zahnrad ₁	Zahnrad ₂
Greifmechanismus	16002	N202-E-TVP2	PSM42BYGHW603	416-010-0022	416-010-0066
Drehgelenk	16002	nicht vorhanden	PSM42BYGHW603	416-010-0014	416-010-0070

	Mutter	Trapezgewinde	Welle	Linearkugellager
Greifmechanismus	MTSPNR20	MTSRX20	PSFRM12	LMK12
Drehgelenk	nicht vorhanden	nicht vorhanden	nicht vorhanden	

Die Komponenten für den Greifarm wurden teilweise in der Elektrotechnikwerkstatt gefertigt. Verschiedene Maße und technische Eigenschaften sind aus den technischen Zeichnungen oder aus der CAD-Datei zu entnehmen. Die restlichen Komponenten wurden bei *MiSUMi* bestellt (www.misumi-europe.de), *G&G Antriebstechnik* (www.gg-antriebstechnik.de), *ekugellager* (www.ekugellager.de), *Pollin* (www.pollin.de) und *Nanotec* (www.de.nanotec.com) bestellt. Wichtige technische Größen und Maße können auf der angegebenen Homepage nachgeschlagen werden oder den angehängten Datenblättern entnommen werden.

2.5. Schmierung/Instandhaltung

Um einen ruhigen Lauf des Automaten zu garantieren, müssen die Kugelumlaufspindeln, Linearführungen und Kugellager geschmiert werden. Somit wird die Reibung minimiert und eine maximale Lebensdauer der Komponenten erreicht. Auch ohne Schmutzeinwirkung ist eine einmalige Lebensdauerschmierung erfahrungsgemäß nicht ausreichend. Es sollte regelmäßig eine Kontrolle der Schmierung stattfinden und ggf. eine Nachschmierung erfolgen.

2.6. Quellenangabe

www.misumi-europe.de
www.gg-antriebstechnik.de
www.gg-antriebstechnik.de
www.ekugellager.de
www.pollin.de
www.de.nanotec.com
www.wikipedia.de
 Tabellenbuch Metall

3. Elektronik

3.1. Anforderungen an die Elektronik

In Kapitel 2 wurde die Funktionsweise des Automaten beschrieben. Daraus ergaben sich die Anforderungen an die zu entwerfende Elektronik. In Abbildung 5 ist eine Übersicht der Komponenten abgebildet. Als Kommunikationsschnittstelle wurde das SPI¹ genutzt, da das Interface eine schnelle Datenübertragung bei einer relativ kleinen Anzahl von benötigten Pins (Im Gegensatz zu parallelen Interfaces) ermöglicht. Weiterhin haben die meisten Controller Peripherie, die entweder direkt eine SPI Schnittstelle ist oder zumindest eine SPI-kompatible Schnittstelle bereitstellt.

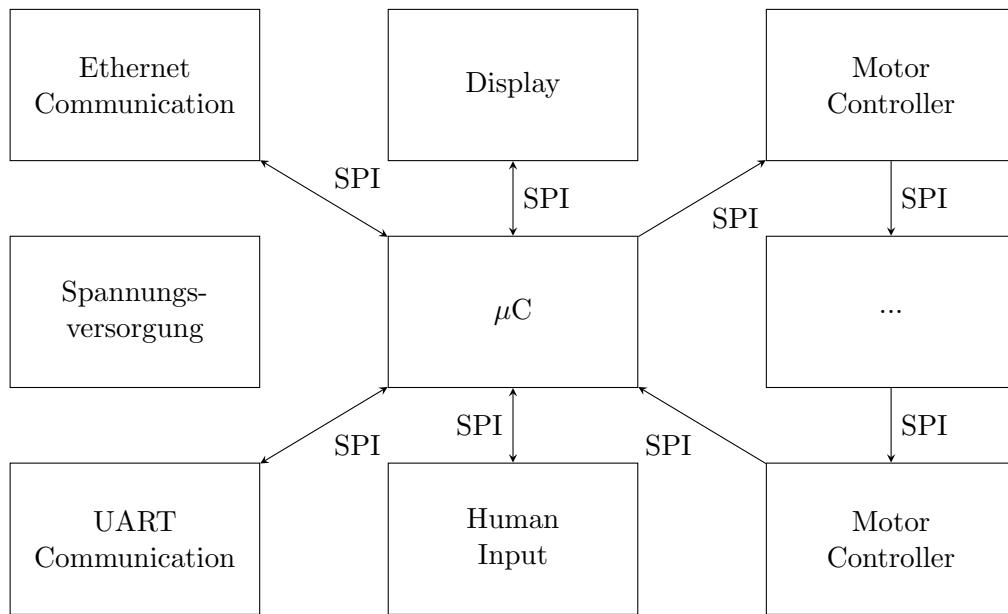


Abbildung 5: Blockdiagramm der Komponenten und deren Beziehungen zueinander

¹Serial Peripheral Interface, von Motorola entwickelt

3.2. Spannungsversorgung

Die Spannungsversorgung besteht aus zwei Teilen mit unterschiedlichen Anforderungen:

- Spannungsversorgung des Digitalteils
- Spannungsversorgung des Leistungsteils

Die Spannungsversorgung des Digitalteils muss stabile 3,3V bei ca. 500mA liefern. Dabei muss beachtet werden, dass sich durch die Verbindung des Digital- und des Leistungsteils über die Motortreiber (Siehe Kapitel 3.7) eventuell Probleme mit der Spannungsstabilität ergeben und dadurch Bitfehler entstehen können. Um universell sein zu können sollte der Digitalteil zudem mit einem weiten Eingangsspannungsbereich versorgt werden können.

Zum Betrieb der Motorenstufen ist eine zweite, sehr niederohmige Quelle nötig. Diese Quelle wird durch ein plötzliches Anfahren von einem oder mehreren Motoren gleichzeitig stark belastet und muss daher eine hohe Regelgeschwindigkeit aufweisen. Die ausgewählten Motortreiber (siehe Kapitel ??) ermöglichen einen effektiven Stromfluss von $I_{eff} = 3A$. Da der Automat über 5 Freiheitsgrade verfügt (Ein Freiheitsgrad wurde mit 2 Motoren realisiert um eine hohe Beschleunigung zu erreichen), fließt ein effektiver Gesamtstrom $I_{eff,gesamt} = 6 * 3A = 18A$. Der Spitzenstrom, der durch einen Treiber fließen kann (Treiberintern begrenzt), beträgt $I_{max} = 7A$. Die Spitzenlast beträgt also $I_{max,gesamt} = 6 * 7A = 42A$ wenn alle Treiber an ihrem Limit arbeiten. In der Praxis ist dieser Fall allerdings ausgeschlossen, da die Treiber erstens eine Limitierung des Stromes erlauben und zweitens wird ein Stillstand sofort detektiert und der entsprechende Treiber geht in den Fehlermodus. Ein Stillstand entspricht dem größten Lastfall für die Treiber, da der Strom lediglich vom ohmschen Widerstand der Motorspulen abhängt, der in der Größenordnung 1Ω bis maximal 10Ω liegt. Nicht zuletzt wurden die Motoren auch nach ihrer nominellen Stromaufnahme ausgewählt, um eine solche Überlast zu vermeiden.

Von Anfang an war eine getrennte Spannungsversorgung von Steuerelektronik und Leistungselektronik vorgesehen. Von der Lösung wurde sich mehr Flexibilität im Bezug auf die Versorgungsspannung des Leistungsteils erhofft und gleichzeitig sollte die Maßnahme die Entwicklungszeit verkürzen, indem zwei verschiedene Netzteile genutzt werden konnten. Bei einer Versorgung aus einer gemeinsamen Quelle hätte sichergestellt werden müssen, dass die Induktivitäten der Motoren bei Lastspitzen (wie sie beim Anfahren mehrerer Motoren gleichzeitig entstehen) die Versorgung des Digitalteils nicht oder nur geringfügig stören. Eine Folge eines Abfalls der Versorgungsspannung ist zum Beispiel der Brownout². Um das Eintreten eines solchen Zustand zu verhindern benötigt man entweder relativ große Kondensatoren, die die Spannung bis zur Nachregelung durch das Netzteil konstant halten oder eine gute Filterung der Spannungsversorgung für den Digitalteil.

Falls das verwendete Netzteil die Spannungsschwankungen gut ausregelt, ist es auch bei getrennten

²Als Brownout wird ein, aufgrund einer kurzzeitigen oder auch länger andauernde Verminderung der Versorgungsspannung, Zustand bezeichnet, bei der ein System nicht länger in der Lage ist, zuverlässig zu arbeiten. Aus diesem Grund besitzen die meisten modernen Mikrocontroller eine Brownout Detection, die den Controller über einen Reset wieder in einen definierten Zustand bringen.

Versorgungsanschlüssen möglich, den Leistung- und den Digitalteil aus einer Quelle zu versorgen.

Durch diese Überlegungen ergeben sich folgende Vorgaben für den Spannungswandler des Digitalteils:

- Eingangsspannungsbereich: 5-24 Volt
- Ausgangsspannung: 3,3V
- Strombelastbarkeit: 500 mA
- Restwelligkeit kleiner 5mV

Erreicht wurden diese Parameter durch eine Kombination des DC/DC Wandlers LT1376³, der die Spannung auf etwa 4-5 Volt (je nach Toleranz der Feedbackwiderstände) heruntersetzt, und einem nachgeschalteten Festspannungsregler des Typs LT1763⁴, welcher die Spannung auf die vorgegebenen 3,3 Volt heruntersetzt.

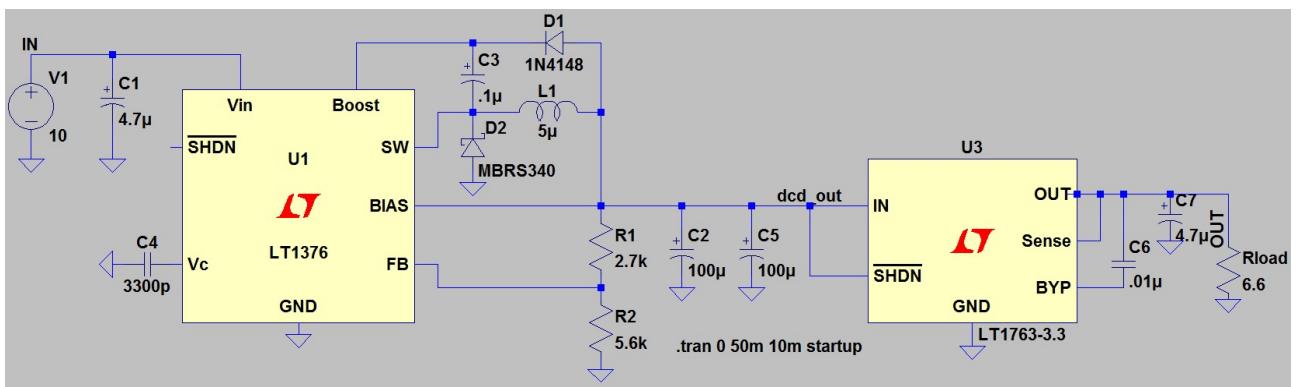


Abbildung 6: Schemazeichnung der Spannungsreglerbaugruppe mit Lastwiderstand am Ausgang.

Über zwei parallel geschaltete $4,7\mu\text{F}$ Pufferkondensatoren (In Abbildung 6 sind das C2 und C5 mit $100\mu\text{F}$) wird die Spannung über die Spule L1 und die Diode D2 herabgesetzt. D1 und C3 dienen zur Erzeugung der für den im LT1376 integrierten MOSFET nötigen Schaltspannung. Für den Spannungsteiler R1 und R2, der die Feedbackspannung erzeugt, wurden handelsübliche Werte aus der E12 Reihe gesucht, die eine Spannung von etwas über 3,8 Volt am Ausgang des DC/DC Wandlers erzeugt, welche über C2 und C5 gesiebt und stabilisiert wird. Die Ausgangsspannung des DCDC Wandlers wurde durch Addition der geplanten Ausgangsspannung des Linearreglers plus dessen maximalem Dropout berechnet. Das Ausgangsnetzwerk besteht aus dem Stützkondensator C7 und dem Bypasskondensator C6. C6 dient normalerweise dazu das Rauschen des Reglers (Größenordnung $20\mu\text{V}$) zu unterdrücken und wäre hier aufgrund der großen Spannungsrippe unnötig. Dennoch ist es eine gute Praktik ihn in das Schaltbild und die endgültige Schaltung zumindest als Option einzuplanen. Ob er im finalen Layout dann bestückt wird ist dem Entwickler überlassen.

³Datenblatt: <http://cds.linear.com/docs/en/datasheet/13756fd.pdf> (Vom 20. März 2014)

⁴Datenblatt: <http://cds.linear.com/en/datasheet/1763fg.pdf> (Vom 20. März 2014)

Um die Stromlieferfähigkeit zu überprüfen wurde die Baugruppe in LTSpice⁵ mit einem Widerstand von $6,6\Omega$ belastet und der Verlauf der Spannung über die Zeit aufgenommen. Dabei wurde eine Einschwingzeit von etwa 30ms abgewartet um der Baugruppe Zeit zu geben, sich auf die endgültige Spannung einzuschwingen.

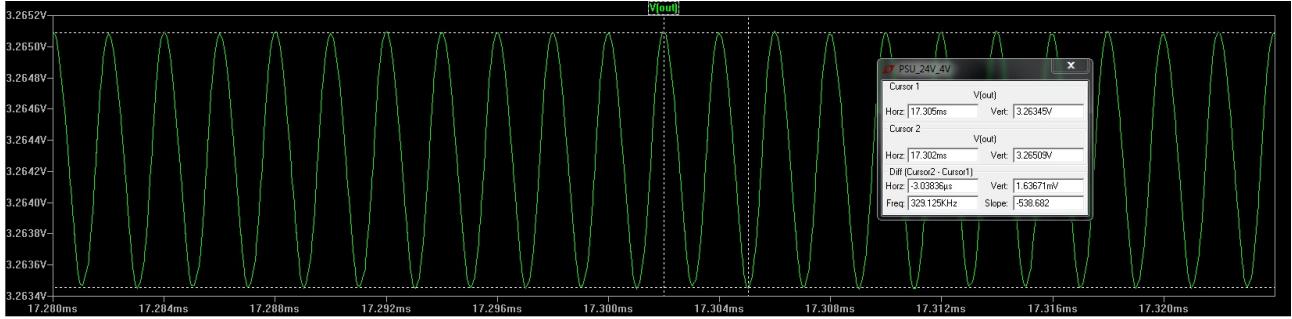


Abbildung 7: Simulation der Ausgangsspannung im eingeschwungenen Zustand über die Zeit und Messung der Restwelligkeit.

In der Simulation sieht man, dass die geregelte Spannung mit 3,26 Volt und einer Restwelligkeit von unter 2mV bei etwa 500mA stabil ist.

Der Aufbau des Spannungsreglers stellte sich allerdings als schwierig heraus. Es wurde davon ausgegangen, dass die Hochschule in der Lage ist, die meisten Bauteile zu löten. Daher wurden Widerstände, Optokoppler, etc per Hand gelötet und nur Mikrocontroller wie auch Motortreiber sollten mit Hochschulmitteln auf die Platine gelötet werden. Diese Annahme war allerdings falsch: es gestaltet sich schwer, QFN Bauteile wie den LT1763 auf die Platine zu bringen, da für QFN Bauteile mit Wärmeleitpad unter dem Chip ein Reflowofen benötigt wird. Ein Platzieren und Löten per Hand gestaltet sich aufgrund des Wärmeleitpads unterhalb des ICs und der sich daraus ergebenden Gefahr, Pins mit dem Pad zu verbinden, als schwer bis unmöglich. Daher wurde beschlossen, den LT1763 durch den bewährten Linearregler LT1117⁶ zu ersetzen, der rücklings auf eine freie Stelle der Platine geklebt wird und über Drähte mit dem Ausgang des ursprünglich geplanten Linearreglers zu verbinden. Die $4,7\mu F$ werden mit einer $10\mu F$ Kapazität gleicher Baugröße ersetzt, da der LT1117 im Gegensatz zu anderen Linearreglern einen größeren Kondensator benötigt.

3.3. Mikrocontroller

Zu Anfang der Projektarbeit war durch die relativ offen gestellte Aufgabenstellung noch nicht ganz klar, welche Rolle die Steuerelektronik spielen würde. Es gab mehrere Szenarien:

- Die Steuerelektronik übernimmt die Motoransteuerung, Sortierung der Boxen, Ein- und Ausgabefunktionen, Bestand des Warenhaushalts, das Webinterface und die Verifikation des Studentenstatus beim Kauf von Hardware.

⁵Download: <http://www.linear.com/design-tools/software/#LTspice> (Vom 20. März 2014)

⁶Datenblatt: <http://cds.linear.com/docs/en/datasheet/1117fd.pdf> (Vom 20. März 2014)

- Die Steuerelektronik übernimmt die Motoransteuerung, Sortierung der Boxen und die restliche Ein- und Ausgabefunktionen. Darüber befindet sich eine 'höhere Logik' wie zum Beispiel ein Computer, der sich um die Lagerverwaltung und die Verifikation des Studentenstatus der Käufer kümmert.
- Die Steuerelektronik übernimmt lediglich die Motoransteuerung und die Ein- und Ausgabefunktionen. Eine 'höhere Logik' übernimmt die komplette Bestandsverwaltung, die Überprüfung auf Studentenstatus und die Netzwerkfunktionalität.

Dadurch ergaben sich folgende Anforderungen an den Mikrocontroller:

- Mindestens 3 Interfaces zur schnellen Anbindung von Peripherie, bevorzugt seriell
- 5V tolerante Eingänge um auch Hardware mit TTL Pegel ansteuern zu können
- Handlötbares Package. Vorzugsweise 64LQFP, maximal 100LQFP.
- Mindestens eine U(S)ART Schnittstelle
- Bewährte, konsistente und 'Royalty free' Funktionsbibliotheken bzw API
- Integrierte FPU
- Mindestens 32kB RAM
- USB Device Port für zukünftige Erweiterungen
- Genug Rechenleistung zur schnellen Auslieferung von Webseiten (Erfahrungswert etwa 50DMIPS)

Mithilfe dieser Anforderungen wurde der TM4C123A6HPM als Mikrocontroller ausgewählt. Der μ C gehört der Tiva C Serie an. Diese Serie wurde von Texas Instruments im Juli 2013 neu eingeführt und steht damit am Anfang ihres Lebenszyklus. Damit ist eine Verfügbarkeit über einen relativ langen Zeitraum gegeben. Weiterhin stellt der Mikrocontroller keine strikten Anforderungen an die Spannungs- und Taktversorgung was das Platinenlayout einfacher macht.

Ein weiteres Entscheidungskriterium waren die günstigen Evaluationsboard, die Texas Instruments zur Tiva C Serie zur Verfügung stellt. Diese wurden genutzt um den Mikrocontroller kennen zu lernen und die Firmware parallel zum Design der Elektronik entwickeln zu können.

Abbildung 10 zeigt die Grundbeschaltung des Mikrocontrollers. Da TI keine Library für das Layoutprogramm Eagle anbietet, wurde diese selbst erstellt, allerdings für den TM4C1233H6PM(I). Dieser hat jedoch das gleiche Pindiagramm wie der verwendete TM4C123A6HPM(I). In diesem Fall wurde mangels Hardware das im Chip integrierte ARM Debugwire nicht genutzt und auf JTAG als Programmier- und Debugschnittstelle gesetzt. Da der Controllertakt über eine PLL stabil gehalten wird, ist die Auswahl an Quarzen groß. Hier wurde ein 16MHz Quarz gewählt, da 16MHz Quarze noch Grundwellenquarze sind und somit relativ problemlos eingesetzt werden können, da sie im Gegensatz zu Oberwellenquarzen nur eine einfache Beschaltung benötigen.

Der RESET Pin ist LOW-aktiv, muss im Betrieb also konstant auf HIGH gehalten werden. Um

den Chip programmierbar zu machen, wird hier eine $10k\Omega$ und $1nF$ RC-Kombination eingesetzt, die den IC nach dem Programmervorgang nach lange genug im RESET Zustand hält. Erfahrungsgemäß benötigen manche Programmieradapter einen größeren Kondensator, da der RESET Pin bei hohen Programmiergeschwindigkeiten zu früh auf HIGH gesetzt wird, der Flash Speicher allerdings noch nicht fertig beschrieben ist. Wenn mit dem Programmer auf dem Evaluation Board programmiert wird, ist der $1nF$ Kondensator allerdings ausreichend.

Alle Pins, die Betriebsspannung führen, wurden mit $100nF$ Stützkondensatoren gegen GND gepuffert um Stromspitzen abfangen zu können. Um die Platine debuggen zu können wurden zwei LEDs mit eingeplant, die zur Anzeige diverser Zustände verwendet werden können. Die LEDs sind so geschalten, dass der Controller den Pin gegen GND ziehen muss damit die entsprechende LED leuchtet. Die Art der Verschaltung dient dazu, den Ausgang nicht zu stark zu belasten: Die ARM GPIO Peripherie kann (wenn keine Peripherie vom Chipsetsteller hinzugefügt wurde) maximal $8mA$ pro Pin sourcen, jedoch weitaus mehr sinken, also gegen GND ableiten.

3.4. Digitale Eingänge

Um die Stellung der Achsen ermitteln zu können ist beim Einschalten eine Referenzfahrt nötig. Die Referenz setzt sich aus mehreren Schaltern zusammen, die physikalisch am Ende jeder Achse positioniert sind und schließen, wenn der Schalter die Position $(0,0,0,0,0)$ in X,Y,Z,A,B erreicht hat. Weitere Schalter werden nötig, wenn man ermitteln möchte ob der Automat offen oder geschlossen ist. Dafür wurden 10 digitale Eingänge auf dem Board reserviert. Alle Eingänge sind galvanisch über Optokoppler vom Typ IL217AT getrennt. In Abbildung 8 ist exemplarisch für alle Eingänge dargestellt, wie die galvanische Trennung realisiert wurde. Das Ziel war, einen weiten Bereich von 0 - 24 Volt zu ermöglichen, ohne die LED durch einen zu großen Strom zu zerstören. Der Kondensator wurde hinzugefügt, um einem Prellen der Taster entgegenzuwirken. Durch den Kondensator lässt sich eine Zeitkonstante festsetzen, die der Schalter eingeschaltet sein muss um die Vorwärtsspannung der LED zu überwinden. Der $100k\Omega$ Widerstand ist aufgrund der kleinen Eingangsimpedanz (etwa $30k\Omega$) des Mikrocontrollers eigentlich unnötig, wurde aber dennoch als Option eingebaut. Der Zustand der Pins (in diesem Fall DIN0) kann nach der Initialisierung direkt durch 'DIN_0_VAL' abgefragt werden. Verantwortlich dafür ist folgendes Makro, welches 0x00 zurückgibt wenn der Pin nicht auf HIGH liegt und eine Zahl verschieden 0x00 wenn der Pin auf HIGH liegt.

```
#define DIN_0_VAL (*((volatile uint32_t *) (GPIO_PORTB_BASE + \
(GPIO_0_DATA + ( 0x20 << 2 )))) >> 5)
```

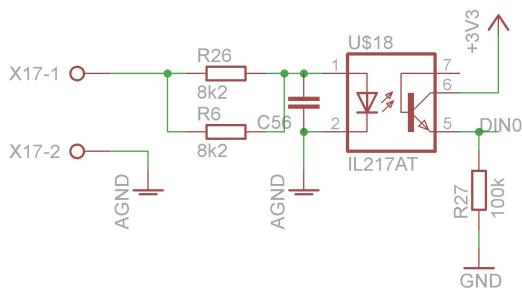


Abbildung 8: Digitaler Eingang

Um Flexibilität zu gewährleisten wurden die Eingänge der Optokoppler über Schraubklemmen nach außen geführt.

3.5. UART Schnittstelle

Der Mikrocontroller bietet zwar eine USB Device Schnittstelle, doch die Einarbeitungszeit in den von TI bereitgestellten USB-Stack oder gar die Programmierung eines eigenen Stacks würde den Umfang einer Projektarbeit weit übersteigen. Da keine großen Mengen an Daten in kurzer Zeit übertragen werden müssen bietet sich hier an, die UART Schnittstelle des Controllers zu nutzen. Der RS232 Standard schreibt allerdings vor, dass eine Verbindung über die serielle Schnittstelle mit Spannungspegeln von -12V und 12V kodiert wird, welche den Eingang des Controllers zerstören würden. Grundsätzlich gibt es zwei Lösungen dafür:

- Man nutzt einen USB<>UART Konverter, der einen TTL kompatiblen Ausgang hat.
- Man nutzt einen Levelkonverter mit Ladungspumpe um auf die vom RS232 Standard geforderten Spannungen zu kommen.

Als UART<>USB Konverter bietet sich der FT232RL an - Der Chip wurde zum Defakto-Standard in der Hobbyelektronik, da er einerseits günstig und andererseits einfach zu handhaben ist. Der FT232RL wäre für diesen Zweck hardwareseitig gut geeignet. Problematisch wird es allerdings auf der Treiberseite. Der Chip emuliert eine virtuelle COM Schnittstelle. Microsoft Windows hat den Treiber allerdings nicht im Repository oder bereits installiert. Daher muss der Treiber umständlich von der Webseite heruntergeladen werden. Bei neueren Versionen von Windows kann es auch vorkommen, dass der Treiber gar nicht erst funktioniert. Daher wurde die Schnittstelle als RS232 Schnittstelle mit einer Sub-D9 Buchse über einen Levelkonverter vom Typ MAX3232(CSE) herausgeführt. Der Levelkonverter ist ebenfalls eine bewährte Methode, eine nach dem RS232 Standard entworfene Schnittstelle bidirektional auf ein anderes Spannungslevel bei Baudraten von bis zu 1MBaud zu bringen.

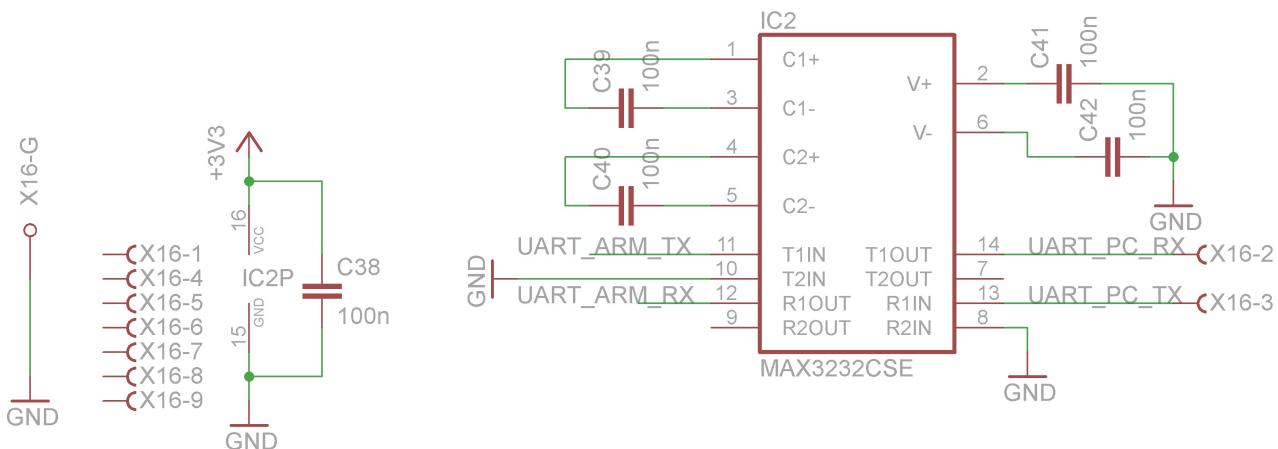


Abbildung 9: Beschaltung des MAX3232 Levelshifters.

Abbildung 9 zeigt die Beschaltung des MAX3232 und der Sub-D9 Buchse. Es sind lediglich die TXD und RXD Leitungen herausgeführt - eine Flusssteuerung ist nicht nötig, da der Mikrocontroller sei-

nen Takt von einem 16MHz Quarz ableitet und somit keinen Baudatenfehler hat, mit 9600baud übertragen wird und die Leitungslänge sehr gering ist. CTS, RTC, DTR, etc sind nicht beschaltet, da der Microcontroller keine USART besitzt und eine synchrone Übertragung aufgrund des paketbasierten Kommunikationsprotokolls nicht notwendig ist.

Die Bootstrapkondensatoren C39 bis C42 sind im Datenblatt als Elektrolytkondensatoren angegeben, Bipolarkondensatoren funktionieren allerdings genauso gut und sind in diesem Kapazitätsbereich noch kleiner als Elkos. Elektrolytkondensatoren wären notwendig, wenn es auf den Innenwiderstand (ESR - Equivalent Series Resistance) oder auf große Kapazität ankäme. Beides ist hier nicht der Fall.

3.6. Netzwerkschnittstelle

Da die Elektronik auch mit dem Intranet und vielleicht sogar mit dem Internet kommunizieren sollte, ist eine Netzwerkschnittstelle nötig gewesen. Da zum Zeitpunkt der Entwicklung der Platine noch nicht die ganze Tiva C Serie sondern nur einige wenige Exemplare ohne integrierte MAC und PHY Peripherie verfügbar waren, wurde der TCP/IP Stack über einen eigenen IC realisiert. Als Controller wurde der W5500 von Wiznet ausgewählt, da der Vorgänger im Hobbybereich als einfach ansteuerbar galt und Wiznet mit dem W5500 an den Erfolg des W5200 anknüpfen wollte. Als Alternative wurde der ENC424J600 in Betracht gezogen. Beide Stacks schneiden bei einem direkten Vergleich etwa gleich gut ab. Ausschlaggebend war die weite Verbreitung, der am Anfang stehende Produktzyklus und die Tatsache, dass es für den Wiznet Chip bereits vorgefertigte Bibliotheken gibt, die lediglich eingebunden werden müssen und somit schnell benutzbar gemacht werden können.

In Abbildung 11 ist die Ansteuerung des Stacks zu sehen. Der Chip benötigt einen 25 MHz Quarz, der relativ nahe am Chip sitzt um einerseits Störeinflüsse vom Quarz und andererseits Störeinflüsse zur Taktversorgung hin zu vermeiden. Außerdem verkleinert man den Einfluss, den die Kapazitätsbeläge der Leitungen auf den Oszillator haben. Die Widerstände R42 bis R48 sind Pulldown Widerstände für chipinterne Peripherie. Weiterhin sorgen die der Netzwerkbuchse vorgesetzten Widerstände für eine Begrenzung des Stromflusses durch die Transformatorschleifen, die zur galvanischen Trennung vorgesehen sind.

Der Wiznet Chip übernimmt sowohl die Flusskontrolle als auch die physikalische Anbindung an das Netzwerk. Die Pakete, auf denen TCP und UDP basieren, müssen allerdings vom Mikrocontroller erstellt, gelesen und interpretiert werden. Da diese Arbeit nicht den fertigen Automaten, sondern einen Entwicklungsstand beschreibt, wurde der Chip zwar beschaltet, konnte durch fehlende Software allerdings noch nicht getestet werden.

3.7. Motorcontroller

Da die Elektronik keine Motorendstufen ansteuern sollte sondern selbst als 'All in One' Lösung dienen sollte, wurden drei Konzepte in Betracht gezogen:

- Die Endstufen werden durch diskrete MOSFET Transistoren mit zugehörigen MOSFET Treiberchips realisiert. Zusätzlich zu den FETs muss noch eine Stromüberwachung- und Begrenzung in den Lastkreis eingebaut werden um bei Stillstand der Motoren einen maximalen, einstellbaren,

Strom fließen zu lassen, der das Drehmoment der Motoren aufrecht erhält, die Motoren aber nicht durch thermische Überlast zerstört.

- Die Endstufen werden durch integrierte Halb- bzw Vollbrücken mit eingebauter Stromüberwachung realisiert. Dadurch entfällt eine Strommessung und Stromregelung.
- Die Endstufen werden durch vollintegrierte Motortreiber realisiert, die über einen Bus mit der Logik verbunden sind.

Der Vorteil der ersten Lösung liegt in der Möglichkeit, durch eine geeignete Auswahl der MOSFETs eine Endstufe zu realisieren, die eine (in Grenzen) beliebig hohe Stromlieferfähigkeit besitzt und somit mit beliebig großen Motoren belastet werden kann. Der Nachteil liegt im Entwicklungsaufwand, der - insbesondere im Rahmen einer Projektarbeit - in Grenzen gehalten werden sollte. Am Ende soll ein Automat das Ziel der Arbeit sein und nicht, eine Endstufe für Schrittmotoren zu entwickeln.

Der zweite Ansatz löst das Problem durch eine integrierte Strommessung und Stromregelung. Damit lässt sich einiges an Zeit und Platinenfläche einsparen. Ein Nachteil dieser Lösung ist die beschränkte Auswahl an modernen integrierten Halb- und Vollbrücken in von Hand lötzbaren Gehäusen. Weiterhin muss der Mikrocontroller wie bei Ansatz 1 auch die Beschleunigungsrampen für die Motoren generieren.

Der dritte Ansatz stellt sich somit als die schnellste und einfachste heraus. Der Nachteil ist hier, dass die Auswahl an vollintegrierten Motortreibern mit großer Leistung noch kleiner ist als die Auswahl an Halb/Vollbrücken.

Der eingesetzte L6470⁷ von STMicroelectronics passt laut Datenblatt sehr gut auf das Anforderungsprofil, ist im PowerSOT36 Gehäuse verfügbar und lässt sich per SPI an den Mikrocontroller anbinden. Laut dem Datenblatt lässt der Treiber auch den Betrieb in einer SPI Daisychain zu wodurch sich die Anzahl der benutzten Pins am Controller auf vier (SDI, SDO, SCLK, !CS) reduziert. Zudem beinhaltet der Chip einen digitalen Signalprozessor, der nach Einstellung der Parameter sowohl die Rampen ,um Beschleunigen und Abbremsen zu können, generiert, als auch den angeschlossenen Motor überwacht, die Betriebsspannung und den Motorstrom regelt, Stillstand und Schrittverlust erkennt und eine hardwareseitige Abschaltung durch einen separaten Schaltereingang ermöglicht. Abbildung 12 zeigt die Beschaltung der Treiber mit der Ladungspumpe zur Ansteuerung der internen MOSFETs. Alle Motorcontroller können vom Mikrocontroller zurückgesetzt werden. Zusätzlich dazu kann jeder Controller über den !FLAG Pin anzeigen, wenn ein Problem oder Ausnahmezustand vorliegt. Die BUSY Pins sind nicht beschaltet, da jeder Controller den Zustand des Pins in einem Register ablegt und somit auslesbar macht. Der 'SW' Eingang liegt bei allen Controllern an einem über Optokoppler schaltbaren Eingang und stellt einen NOTAUS dar: Sobald der Eingang auf HIGH geht, stoppen die Motortreiber die Motoren mit möglichst hoher Abremsgeschwindigkeit und schalten die Ausgänge dann auf HIGH-Z wodurch die Motoren stromlos werden.

⁷Datenblatt: <http://www.st.com/web/en/resource/technical/document/datasheet/CD00255075.pdf> (Vom 20. März 2014)

3.8. Platinenlayout

Die Platine ist aufgrund der Packungsdichte vierlagig (Siehe Abbildung 13. Beim Layout wurde darauf geachtet, den Digitalteil vom Leistungsteil zu trennen, da über den Widerstand des Kupfers bei hohen Strömen eine Spannung abfällt und Logikpegel verfälschen könnte. Der Spannungsabfall ist umgekehrt proportional zur Dicke und Breite der Leiterbahn. Eine große Massefläche bietet also einen kleineren Widerstand als eine 20mil Leiterbahn.

Die Sektionen sind auf allen vier Layern gleich. Somit ist die Platine dreigeteilt: Der dritte Teil ist die Bank die die digitalen Eingänge beinhaltet. Diese sind durch Optokoppler galvanisch getrennt und benötigen somit eine eigene (wenn auch gemeinsame) Masse. Im digitalen Teil wurde auf den obersten Layer die Spannungserzeugung und - soweit möglich - auch die Spannungsversorgung für die einzelnen Bausteine gelegt. Außerdem befinden sich auf dem Top Layer noch die Signale für die Schnittstellen (LAN, RS232, DIN, JTAG) und die Taktversorgung. Der Layer darunter beherrbert die Spannungsversorgung des DCDC Wandlers und die Signale für die Motortreiber. Die beiden Bottomlayer führen die Betriebsspannung für die digitalen Eingänge, die Spannungen für den Digitalteil der Motortreiber und verstreu einigen Signale, die sonst nicht anders geroutet werden konnten.

Die beiden Top Layer im Leistungsteil wurden dazu benutzt, die Signale der Motortreiber zu verteilen, damit die große freie Fläche als Kühlkörper genutzt werden kann. Je nach Belastung ist dennoch ein Kühlkörper nötig. Der erste Bottom Layer (Gezählt wird immer von der Oberseite der Platine) führt die Spannungsversorgung der MOSFETs und die Ausgänge der Treiber zu den jeweiligen Klemmen. Der zweite Bottom Layer dient durch die Vias an den Kanten des Layers als Kühlkörper und thermische Kapazität um Lastspitzen abzufangen.

3.9. Errata

- Der Clock für die Motortreiber heißt am Mikrocontroller SPI_CLK und an den Bausteinen SPI_CLOCK. Das bedeutet, dass der SPI-Takt nicht zu den Motortreibern gelangt. In der aktuellen Revision wurde das mit einem Kabel vom Mikrocontroller zur SPI_CLOCK Leitung notdürftig behoben
- Das !MOT_RESET Signal der Motortreiber liegt auf keinem Potential. In der aktuellen Revision wurde das mit einem Kabel vom VCC Bus zur !MOT_RESET Leitung notdürftig behoben. In der Revision 2 sind die Motortreiber mit PD5 des Mikrocontrollers verbunden.
- Der Spannungsregler (IC1) von 5V auf VCC kann mit Hochschulmitteln im QFN Gehäuse nicht ohne größeren Aufwand platziert und gelötet werden. Ein Ersatz ist ein LT1117 der auf die Platine geklebt und mit Drähtchen verbunden wird
- Die Ansteuerung der Motortreiber mittels Daisychain hat sich als unnötig schwierig herausgestellt. Daher werden in Revision 2 alle Treiber parallel an den Bus angeschlossen und über den Chip Select Eingang ausgewählt oder auf einer eigenen Platine untergebracht.
- Eine gesonderte Spannungsversorgung hat sich als unnötig erwiesen. Die Idee dahinter war, die Motortreiber mit ihrer Maximalspannung zu versorgen damit die Motoren bei hoher Drehzahl

ein größtmögliches Drehmoment haben. Da der Schaltregler allerdings schon geroutet war und die Maximalspannung der Motortreiber über dem Limit des Konverters liegt, wurde, um Zeit zu sparen, eine zweite Spannungsversorgung etabliert. In Version 2 wird ein Spannungsregler eingesetzt, dessen Maximalspannung über der der Motortreiber liegt.

- Die RS232 kompatible Schnittstelle nimmt zu viel Platz auf der Platine weg. In Version 2 wird die USB Schnittstelle des ARM genutzt, um eine virtuelle serielle Schnittstellenkommunikation zu ermöglichen.
- Die Lösung zur Ethernetkommunikation in Version 1 ist unbefriedigend, da man von einem relativ kleinen Chipsetsteller abhängig ist. Weiterhin benötigt die Lösung mit Ethernetchip zu viel Platz auf der Platine. Leider war die vollständige Microcontrollerfamilie des Herstellers Texas Instruments zur Entwicklungszeit nur unter NDA einzusehen. Zum Zeitpunkt der Erstellung dieses Dokuments werden Microcontroller mit komplettem MAC und PHY Layer in der Tiva Familie gelistet. Version 2 wird die jetzige Lösung ersetzen und den TCP/IP Stack in den Mikrocontroller integrieren.
- Es sollte eine Oufferbatterie für den Microcontroller eingebaut werden um Register vor einem Stromausfall noch speichern zu können. Eventuell genügt auch ein kleiner oder auch ein großer Kondensator, mit dessen Hilfe ein Brownout verhindert werden kann.
- Die Benutzerkommunikation erfordert einen weiteren Mikrocontroller als SPI Slave. Diese Lösung ist unbefriedigend, im Hinblick auf den Zeitaufwand jedoch nicht anders machbar. Version 2 wird hinter ein Display geschraubt werden können, einen Dekoder für Tastatureingaben haben und mehr Pins am Mikrocontroller verfügbar haben.

4. Software

In diesem Kapitel wird der Ablauf und die Logik der Software erläutert, sowohl in der tatsächlichen Ausführung, als auch in der Planung für eine spätere, umfangreichere Version. Das grundlegende Konzept der Software ist eine durch Aufträge gesteuerte Zustandsmaschine. Die Maschine selbst dient lediglich der Dekodierung und Interpretation von Aufträgen, die über eine Schnittstelle übermittelt und danach in Befehle für die Motortreiber umgesetzt werden. Die Software selbst weiß nicht, welche Kiste welche Größe und welchen Inhalt hat. Aus diesem Grund muss ein externes Verwaltungsprogramm, im folgenden als "höhere Logik" bezeichnet, die Verwaltung des Boxeninhalts übernehmen.

Im Groben kann man die Software in 5 Aufgabenbereiche einteilen:

1. Die State-Machine
2. Die Schnittstellenkommunikation und Nachrichtendekodierung
3. Das Verwalten und Abarbeiten der Aufträge
4. Die Steuerung der Mechanischen Komponenten
5. Die Fehlererkennung und Behebung

Wird eine Nachricht über die Schnittstelle empfangen, wird sie in einer Interrupt-Service-Routine in einem Ringpuffer abgespeichert und später in der State-Machine dekodiert, und auf Richtigkeit überprüft. Wenn ein gültiger Auftrag empfangen wurde, wird er in Motorenbefehle umgesetzt.

Sollte dabei an einer Stelle ein Fehler erkannt werden, wird dieser im Zustand STATE_ERROR behoben.

4.1. Verwendete Programmbibliotheken

Texas Instrument liefert für den Tiva Microcontroller die Bibliothek Tivaware Peripheral Driver Library mit. Diese enthält Header-Dateien, mit API-Funktionen zur vereinfachten Ansteuerung des Microcontrollers. Diese Bibliothek beinhaltet alle Header-Dateien, die ansonsten nicht explizit erläutert werden.

Darüber hinaus bedienen wir uns einer Bibliothek namens *dSpin.obj*. Diese Bibliothek vereinfacht das Ansteuern der einzelnen Motortreiber.

Die Bibliothek wurde vom Hersteller der Motortreiber, STMicroelectronics, für Kunden erstellt und bereitgestellt. Alle Funktionen, Variablen und Definitionen in dSpin.c und inc_dSpin.h sind Teil dieser Bibliothek, die auf www.st.com kostenlos erhältlich ist.

4.2. Die State-Machine

Die State-Machine stellt die Verbindung zwischen den einzelnen Segmenten dar und verwaltet und steuert deren Aufruf.

Realisiert wurde die State-Machine mit einer switch-case Anweisung, dessen Argument ausschließlich in der State-Machine selbst verändert wird.

```

switch(SYS_STATE){
    case STATE_STARTUP: (...)

    case STATE_ERROR: (...)

    case STATE_JOB_HANDLING: (...)

    case STATE_MSG_HANDLING: (...)

    case STATE_UART_COM: (...)

    case STATE_MOT_COM: (...)

    case STATE_NET_COM: (...)

    case STATE_IDLE: (...)

}

```

4.2.1. STATE_STARTUP

Der Zustand STATE_STARTUP wird beim Systemstart aufgerufen und bereitet alle nötigen Register und Speicherbereiche für den Betrieb vor und fährt anschließend die Motoren zum Koordinatenursprung (Home-Position). Nach der Initialisierung wird der Zustand STATE_IDLE aufgerufen.

./setup_INT Mit setup_INT werden die jeweiligen Interruptquellen aktiviert und anschließend die API-Funktion IntMasterEnable aufgerufen, um Interrupts zu ermöglichen.

setup_GPIO Mit setup_GPIO werden die General-Purpose-Input\Output Ports aktiviert. Für den laufenden Betrieb ist die Aktivierung von Port A relevant, da Die UART Pins auf Port A liegen und auch als solche initialisiert werden müssen. Zusätzlich sind zum Debuggen die LEDs an den Pins 1 bis 3 von Port F als Output initialisiert.

setup_UART In setup_UART wird die UART-Peripherie aktiviert und initialisiert. Als Workaround für ein Problem das bei der Entwicklung aufgetreten ist, liegt der Transmit-Interrupt auf EOT, also End-of-Transmission. Das bedeutet, dass der Transmit-Interrupt ausgelöst wird, wenn eine Übertragung endet. Siehe auch: Init_Transmission.

setup_JOB_QUEUE In setup_JOB_QUEUE wird die Job-Warteschlange initialisiert. Vorerst wird nur eventueller "Datenmüll entfernt, also alle Elemente zu Null gesetzt. Allerdings wäre es mit physikalischen Speichermedien möglich, die Warteschlange von diesen abzurufen.

setup_HOME_POS setup_HOME_POS fährt die Motoren an den Koordinatenursprung und legt diesen als Home-Position fest. Derzeit ist setup_HOME_POS noch auskommentiert, da die Funktionalität der Motorensteuerung erst getestet werden muss (siehe: Abschnitt Elektronik).

4.2.2. STATE_ERROR

Der Zustand STATE_ERROR dient zur Bearbeitung und Behebung erkannter Fehler. Nach der Bearbeitung eines Fehlers wird außerdem eine Nachricht an die höhere Logik gesendet.

Beim Auftreten eines Fehlers wird jeweils ein entsprechendes Fehlerflag gesetzt, welche im STATE_ERROR der Reihe nach abgefragt werden. Diese Fehlerflags sind definiert als:

```
typedef struct sError {
    uint32_t SER;
    uint32_t STA;
    uint32_t ETH;
    uint32_t MSG;
    uint32_t MOT;
    uint32_t JOB;
} sErrorFlags;
sErrorFlags eState;
```

Bei der Fehlerbearbeitung wird der Wert des jeweils zur Fehlerquelle zugehörigen Elements abgefragt und mit definierten Werten abgeglichen. Gesetzt werden die Flags, indem sie mit einem entsprechenden Wert ODER-verknüpft werden, bzw mit EXOR, um sie zu löschen.

Nach der Behandlung der Fehler wird der Zustand STATE_UART_COM aufgerufen und die höhere Logik wird von der Art des Fehlers unterrichtet.

4.2.3. STATE_JOB_HANDLING

Der Zustand STATE_JOB_HANDLING übernimmt die zuvor dekodierte Nachricht und fügt sie zur Job Warteschlange hinzu, schreibt eine Bestätigung für die höhere Logik in den Sende-Ringpuffer und ruft den Zustand STATE_UART_COM auf.

4.2.4. STATE_MSG_HANDLING

Der Zustand STATE_MSG_HANDLING entnimmt die empfangene Zeichenkette aus dem Zeichenpuffer und dekodiert sie. Anschliessend wird die Nachricht auf Inhalt und Echtheit überprüft und in einen Puffer geschrieben. Dann wird der Zustand STATE_JOB_HANDLING aufgerufen.

4.2.5. STATE_UART_COM

In STATE_UART_COM werden zwei Funktionen ausgeführt: Zum einen wird eine empfangene Nachricht aus dem Empfangs-Ringpuffer in einen Zeichenpuffer geschrieben und auf dessen Überlauf geprüft, zum anderen wird überprüft, ob eine Sendeanfrage aussteht, woraufhin der Sendevorgang initialisiert wird.

Init_Transmission Aufgrund von Problemen beim Einrichten der UART-Kommunikation ist aus Zeitgründen dieser Workaround entstanden: Init_Transmission fügt "manuell" das erste Zeichen des Ringpuffers in den FIFO. Nachdem das erste Zeichen übertragen wurde, wird der EOT-Interrupt ausgelöst und die Übertragung läuft in der ISR automatisch (wie ursprünglich geplant) weiter.

4.2.6. STATE_MOT_COM

STATE_MOT_COM ist ausschliesslich für die Steuerung der Motoren zuständig. Beim Aufruf von STATE_MOT_COM wird der aktuell zu bearbeitende Job anhand des OpCodes abgehandelt. Nach Bearbeitung des Jobs wird der Job gelöscht und STATE_UART_COM wird aufgerufen, um der höheren Logik die erfolgreiche Bearbeitung des Jobs mitzuteilen.

4.2.7. STATE_NET_COM

Dieser Zustand ist für eine eventuelle spätere Anbindung an ein Netzwerk vorgesehen, wird derzeit aber noch nicht genutzt.

4.2.8. STATE_IDLE

In STATE_IDLE wird ein Job aus der Warteschlange in den aktuell zu bearbeitenden Job geschrieben, falls dieser leer ist. Anschließend wird STATE_MOT_COM aufgerufen.

Wenn die Job Warteschlange leer ist wird 'IDLE' an die höhere Logik geschickt.

4.3. Die Schnittstellenkommunikation

Um die Befehle der höheren Logik umsetzen zu können, muss der Automat sie erst einmal bekommen: diese Aufgabe übernimmt eine serielle Schnittstelle.

In der Interrupt-Service-Routine werden 3 Aufgabenbereiche bearbeitet:

- Empfangsinterrupts:

Wird eine Nachricht empfangen, so wird entweder UART_INT_RX oder UART_INT_RT ausgelöst. RX wird ausgelöst wenn der FIFO voll ist und RT wenn ein Line-Timeout entdeckt wird. Line-Timeout bedeutet, dass für einen längeren Zeitraum nichts mehr empfangen wurde.

- Sendeinterrupts:

Wenn Daten erfolgreich gesendet wurden, wird UART_INT_TX ausgelöst. Dies geschieht so oft, bis die komplette Nachricht übertragen wurde. Für den Fall dass der Interrupt noch nicht ausgelöst wurde, aber eine Sendeanfrage aussteht wird zusätzlich noch auf SER_TX_REQUEST abgefragt.

- Fehlerinterrupts:

Wenn bei der Übertragung ein Fehler auftritt, wird einer der folgenden Interrupts ausgelöst: UART_INT_OE, UART_INT_BE, UART_INT_PE, UART_INT_FE. Diese stehen, der Reihenfolge nach, für Overrun Error, Break Error, Parity Error und Framing Error.

Zu Beginn der ISR wird das Interrupt-Status Register ausgelesen, zwischengespeichert und danach zurückgesetzt, damit die ISR nicht mehrfach aufgerufen wird. Anhand des ausgelesenen Status Registers wird bestimmt, welche der drei Aktionen ausgeführt werden - es sind mehrere möglich, da sie der Reihe nach abgefragt werden.

4.3.1. Empfangen

Wenn eine Nachricht empfangen wurde, werden die Zeichen aus dem FIFO in den Ringpuffer geschrieben und auf '\n' abgefragt. Hierbei wird außerdem auch der Ringpuffer auf Überlauf überprüft. Wurde ein '\n' empfangen, so ist die Nachricht komplett und der Zustand STATE_UART_COM wird aufgerufen, sowie das Flag eState.SER mit dem definierten Wert SER_RX_MESSAGE_COMPLETE ODER-verknüpft und die Interruptquelle wird aus dem Status-Puffer gelöscht.

4.3.2. Senden

Wenn eine Nachricht gesendet werden soll, wird der Inhalt des Ringpuffers Stück für Stück in den FIFO geschrieben. Anschließend wird die Interruptquelle aus dem Status-Puffer gelöscht, und aus dem Flag eState.SER wird der Wert SER_TX_REQUEST mit EXOR gelöscht.

4.3.3. Fehler

Wird einer der Error-Interrupts ausgelöst, so wird dieser hier ausgewertet und ein entsprechendes Flag gesetzt und der Zustand STATE_ERROR wird aufgerufen.

4.4. Die Ringpuffer

Da der FIFO der seriellen Schnittstelle nur 16 Byte groß ist, müssen die Daten, die empfangen oder gesendet werden sollen, zwischengespeichert werden, bis die gesamte Nachricht übertragen wurde.

Dafür verwenden wir zwei Ringpuffer: einen für empfangene Nachrichten und einen für Nachrichten, die noch gesendet werden. Diese stehen in circular.obj.

Die Ringpuffer sind global angelegt, da sie sowohl in der main-Funktion als auch in der ISR verwendet werden, aber keine Übergabewerte zur ISR möglich sind.

In Inc_Circular.h sind die Ringpuffer definiert als

```
typedef struct Buf {  
    uint8_t buf[BUF_SIZE];  
    uint32_t rdptr;  
    uint32_t wrptr;  
    uint32_t size;  
} Circ_Buf;
```

Für die Verwaltung der Ringpuffer sind die im Folgenden erklärten Funktionen zuständig.

4.4.1. getChar

Die Funktion getChar liest ein Zeichen aus dem Ringpuffer aus, erhöht den Lesezeiger (rdptr) und gibt das Zeichen zurück. Überschreitet der Lesezeiger die Arraygröße von buf[BUF_SIZE] wird er auf 0 gesetzt. Wenn der Ringpuffer leer ist, gibt die Funktion EXIT_BUFFER_EMPTY zurück.

4.4.2. putChar

Die Funktion putChar schreibt ein Zeichen in den Ringpuffer, erhöht den Schreibzeiger (wrptr) und gibt EXIT_SUCCESS zurück. Überschreitet der Schreibzeiger die Arraygröße von buf[BUF_SIZE] wird er auf 0 gesetzt. Wenn der Ringpuffer voll ist gibt die Funktion EXIT_BUFFER_FULL zurück.

4.4.3. getRemBufSpace

Die Funktion getRemBufSpace vergleicht die relative Position von Schreib- und Lesezeiger des übergebenen Ringpuffers miteinander und ermittelt daraus den verfügbaren Speicherplatz und gibt diesen zurück.

4.4.4. putMsg

Die Funktion puMsg verwaltet das Einfügen einer Nachricht in den Ringpuffer. Bei Erfolg gibt die Funktion EXIT_SUCCESS zurück, bzw. EXIT_FAILURE wenn nicht genug Speicherplatz verfügbar ist.

Zur Überprüfung des Speicherplatz wird die Funktion getRemBufSpace aufgerufen und anschließend wird der verfügbare Platz mit dem benötigten verglichen. Bei Erfolg wird die Nachricht Zeichen für Zeichen mit putChar in den Ringpuffer geschrieben.

4.4.5. getMsg

Die Funktion getMsg verwaltet das Auslesen einer Nachricht aus dem Ringpuffer. Bei Erfolg gibt die Funktion EXIT_SUCCESS zurück.

Die einzelnen Zeichen werden mit getChar ausgelesen und auf EXIT_BUFFER_EMPTY abgefragt. Ist der Puffer leer, wird EXIT_BUFFER_EMPTY zurückgegeben. Bei erfolgreichem auslesen werden die Zeichen in einen Übergabepuffer geschrieben. Wird dieser überschritten, wird EXIT_BUFFER_SIZE zurückgegeben.

Wurden alle Zeichen erfolgreich ausgelesen und abgespeichert, wird am Ende der Nachricht ein '\n' und ein '\0' angehängt, um die spätere Dekodierung zu erleichtern.

4.4.6. insertString

Die Funktion insertString fügt eine Zeichenkette ans Ende einer anderen Zeichenkette ein. Zuerst wird die Länge der einzufügenden Zeichenkette überprüft, um einen Überschreiten des Speicherbereichs zu vermeiden. Danach wird die Zeichenkette elementweise an die andere angehängt.

4.4.7. clearString

Die Funktion clearString löscht eine übergebene Zeichenkette Elementweise.

4.5. Das Job-System

Jede empfangene Nachricht wird dekodiert und in einen Job gespeichert. Dieser Job enthält alle Informationen, die der Microcontroller benötigt, um die gewünschte Aktion auszuführen. Das Job-

System ist so angelegt, dass mit nur geringem Aufwand weitere Funktionen implementiert werden können.

Ein Job ist aufgebaut wie Folgt:

```
typedef struct Job {  
    uint32_t id;  
    uint32_t opCode;  
    uint32_t coordX;  
    uint32_t coordY;  
    uint32_t coordZ;  
    uint32_t JOINT;  
    uint32_t GRIPPER;  
    uint32_t opt1;  
    uint32_t opt2;  
    uint32_t opt3;  
    uint32_t crc;  
} sJob;
```

Die Jobs werden in einer Warteschlange und in zwei Puffern gespeichert: JobBuffer und CurrentJob. JobBuffer ist ein Puffer für die Übergabe des empfangenen Jobs von decodeMsg an addJob, und CurrentJob enthält den aktuell zu bearbeitenden Job. Nach der Bearbeitung eines Jobs wird eine Nachricht an die höhere Logik gesendet. Sollte an irgendeiner Stelle ein Fehler auftreten bzw. entdeckt werden, so wird der fehlerhafte Job gelöscht und die Art des Fehlers, sowie die Job-ID, an die höhere Logik gesendet.

Einen Job kann man in vier Teile zusammenfassen:

- Die ID ist eine eindeutige Identifikationsnummer, die jedem Job von der höheren Logik zugewiesen wird.
- Der OpCode gibt die Funktion an, die der Automat durchführen soll. Derzeit vorgesehen sind:
 - manuelle Steuerung durch den Benutzer
 - Warenausgabe
 - Waren auffüllen
 - Rekalibrierung durch den Benutzer
- Die Parameter sind je nach OpCode unterschiedlich zu interpretieren.
- Die CRC-Kennung ist wichtig um Übertragungsfehler zu erkennen.

4.5.1. decodeMsg

Die Funktion decodeMsg übernimmt die in getMsg erhaltene Zeichenkette und trennt sie, als ASCII-Zeichen interpretiert, in einzelne Bestandteile auf:

```
ID[ OpCode, X-Koordinate, Y-Koordinate, Z-Koordinate , Drehgelenk, Greiferstellung,  
Optional1, Optional 2, Optional3 ]CRC-Kennung
```

Das erste Element ist immer die Job-ID, darauf folgen, in eckigen Klammern, der OpCode und die Parameter, welche durch Kommata getrennt sind. Zusätzlich zu den Fahrparametern (X-, Y-, Z- und Greifekoordinaten) sind zusätzlich auch drei optionale Parameter vorgesehen für eventuelle zukünftige Funktionen. Zum Schluss folgt eine CRC-Kennung.

Die Elemente werden durch die Erkennung der trennenden Zeichen aufgeteilt und zwischengespeichert.

Nachdem die Elemente getrennt wurden, werden sie der Reihe nach mit atoi() in Zahlenwerte umgewandelt und überprüft:

Zuerst wird die ID überprüft, danach die CRC-Kennung und anschließend die Parameter. Bei erfolgreichen Testergebnissen gibt die Funktion EXIT_SUCCESS zurück. Wenn einer der Tests einen Fehler findet, gibt die Funktion einen entsprechenden Fehlerwert zurück.

4.5.2. validateID

Die Funktion validateID durchläuft die Job-Warteschlange und vergleicht die darin gespeicherten Job-IDs mit der empfangenen Job-ID. Dies geschieht, um zu verhindern, dass derselbe Job mehrfach ausgeführt wird. Für spätere Versionen des Automats wäre hier auch eine Sicherung gegen Missbrauch möglich, indem man alle empfangenen Job-IDs langfristig speichert und diese ebenfalls zum Vergleich verwendet.

Ebenso wäre die Überprüfung nach einem speziellen Code in der ID möglich, als eine weitere Absicherung gegen Missbrauch.

4.5.3. checkCRC

Eine zyklische Redundanzprüfung ist zwar vorgesehen, aber noch nicht implementiert. Die Funktion checkCRC ist leer und gibt lediglich ein EXIT_SUCCESS zurück.

4.5.4. validateMsg

Die Funktion validateMsg überprüft die empfangenen Parameter auf deren Inhalt, um eventuelle Schäden am Automaten zu vermeiden.

Die Funktionen testet, je nach OpCode, ob bestimmte Sicherheitskriterien eingehalten werden, damit z.B. nicht der Motor an die Wand fährt oder Ähnliches. Bei Erfolg gibt die Funktion EXIT_SUCCESS zurück, sonst EXIT_FAILURE.

4.5.5. addJob

Die Funktion addJob fügt den empfangenen und überprüften Job zur Warteschlange hinzu. Dazu durchläuft die Funktion die Warteschlange bis zum ersten leeren Element, schreibt dort den Job rein und gibt EXIT_SUCCESS zurück. Wird kein leeres Element gefunden, gibt die Funktion EXIT_FAILURE zurück.

4.5.6. callJob

Die Funktion callJob entnimmt der Warteschlange einen Job, schreibt diesen in CurrentJob und rückt die Warteschlange auf:

Zuerst wird überprüft, ob CurrentJob leer ist oder nicht. Steht noch ein Job in CurrentJob, gibt die Funktion JOB_PENDING zurück, damit dieser zuerst ausgeführt wird. Ist CurrentJob leer, wird das erste Element der Warteschlange auf Inhalt überprüft. Bei Erfolg wird dieses in CurrentJob geschrieben, bei Fehlschlag gibt die Funktion EXIT_FAILURE zurück. Nachdem das erste Element in CurrentJob übertragen wurde, werden alle anderen Elemente der Warteschlange eine Position vorgerrückt und die dadurch frei gewordenen Elemente der Warteschlange werden gelöscht.

Bei Erfolg gibt die Funktion EXIT_SUCCESS zurück.

4.5.7. jobDone

Nach der Bearbeitung, unabhängig von Erfolg oder Misserfolg, werden die jeweiligen Jobs mit jobDone gelöscht. In jobDone wird jedes Element des übergebenen Jobs zu Null gesetzt.

4.6. Die Motorsteuerung

Zur Vereinfachung der Motorensteuerung verwenden wir eine Kette von Motortreibern. Da die Motortreiber in Serie geschaltet sind, müsste man aufwendig bestimmen, welchen Motor man ansteuern möchte und den Befehl dann durch die Treiberkette schieben. Um dies zu vereinfachen, bedienen wir uns einer Bibliothek namens *dSpin.obj*. Diese Bibliothek vereinfacht das Ansteuern der einzelnen Motortreiber.

Aus Gründen, die bereits in der Einleitung erläutert sind, wurden weder die Motorsteuerbefehle noch die Funktionen in dspin.c (von uns) praktisch getestet. Darum geben sie, zum Debuggen des restlichen Codes, derzeit nur EXIT_SUCCESS zurück bevor die Befehle ausgeführt werden. Des weiteren sind die Motorsteuerungs-Funktionen in ihrer aktuellen Form lediglich als Testfunktionen gedacht und werden für den fertigen Automaten noch stark verändert werden.

4.6.1. Remote_Ctrl_Motor

Die Funktion Remote_Ctrl_Motor wird im Fernsteuerungsmodus aufgerufen. Der Fernsteuerungsmodus ermöglicht es derzeit allerdings nur, zum Schutz des Automaten, Positionen anzufahren, nicht jedoch freies Fahren. Zuerst wird die Zielposition mit der aktuellen Position verglichen, um die Fahrtrichtung zu ermitteln. Dies geschieht sowohl für die X- als auch für die Y- Koordinate. Anschließend wird gewartet bis die Motoren wieder still stehen, um die gewünschte Greiferstellung einzunehmen. Auch hier wird wieder gewartet bis die Motoren still stehen, und erst danach wird die Z-Poition angefahren. Erst wenn die Motoren wieder still stehen verlässt der Microcontroller die Funktion.

4.6.2. Ctrl_Motor

Die Funktion Ctrl_Motor ist für die automatische Steuerung des Automaten zuständig. Zu Beginn der Funktion wird ermittelt, ob und welches Sortierfeld verwendet werden soll, anhand der Greiferbreite

und Z-Koordinate des auszuführenden Jobs. Anschließend wird ermittelt ob die Ziel-Box geleert oder gefüllt werden soll.

Beide Vorgänge benutzen die gleichen Funktionen in der gleichen Reihenfolge, mit der Ausnahme dass für die Ausgabe EmptyBox verwendet wird und beim Nachfüllen RefillBox.

Wenn die Box nicht in der ersten Reihe liegt, werden alle Boxen, die davor liegen, mit getBox und putBox in den jeweiligen Sortierschacht gebracht. Anschließend wird mit getBox die gewünschte Box entnommen und entweder mit EmptyBox oder RefillBox zum Ausgabeschacht gefahren, wo die jeweilige Aktion ausgeführt wird. Danach wird die Box mit PutBox zurückgefahrene und die umgestellten Boxn werden wieder zurücksortiert, mit getBox und putBox.

4.6.3. `getBox`

Die Funktion getBox fährt eine Position an und holt eine oder mehrere Box(n) aus einem Schacht.

Zuerst wird die Fahrtrichtung in X- und Y-Richtung ermittelt und anschliessend die X- und Y-Positionen angefahren. Wenn die Motoren wieder still stehen wird die Greiferstellung angepasst. Hat der Greifer die gewünschte Stellung eingenommen, wird die Z-Position angefahren, der Greifer geschlossen, und damit die gewünschten Boxn gegriffen. Anschließend fährt der Greifer auf die Position Z=0 zurück. Wenn die Motoren wieder still stehen, wird die Funktion verlassen.

4.6.4. `putBox`

Die Funktion putBox fährt eine Position an und stellt eine Box dort ab.

Zuerst werden die X- und Y-Positionen angefahren, dann wird die Z-Position angefahren, der Greifer öffnet sich und die Position Z=0 wird angefahren.

Wie bei getBox wird zwischen jedem Schritt gewartet bis die Motoren still stehen.

4.6.5. `EmptyBox`

Die Funktion EmptyBox fährt den Ausgabeschacht an und leert dort eine Box aus.

Zuerst werden die X- und Y-Positionen angefahren, danach die Z-Position. Im Ausgabeschacht wird der Greifer um 180° gedreht, um die Box zu leeren. Danach wird der Greifer wieder zurückgedreht, und die Position Z=0 wird angefahren.

Wie bei getBox wird zwischen jedem Schritt gewartet bis die Motoren still stehen.

4.6.6. `RefillBox`

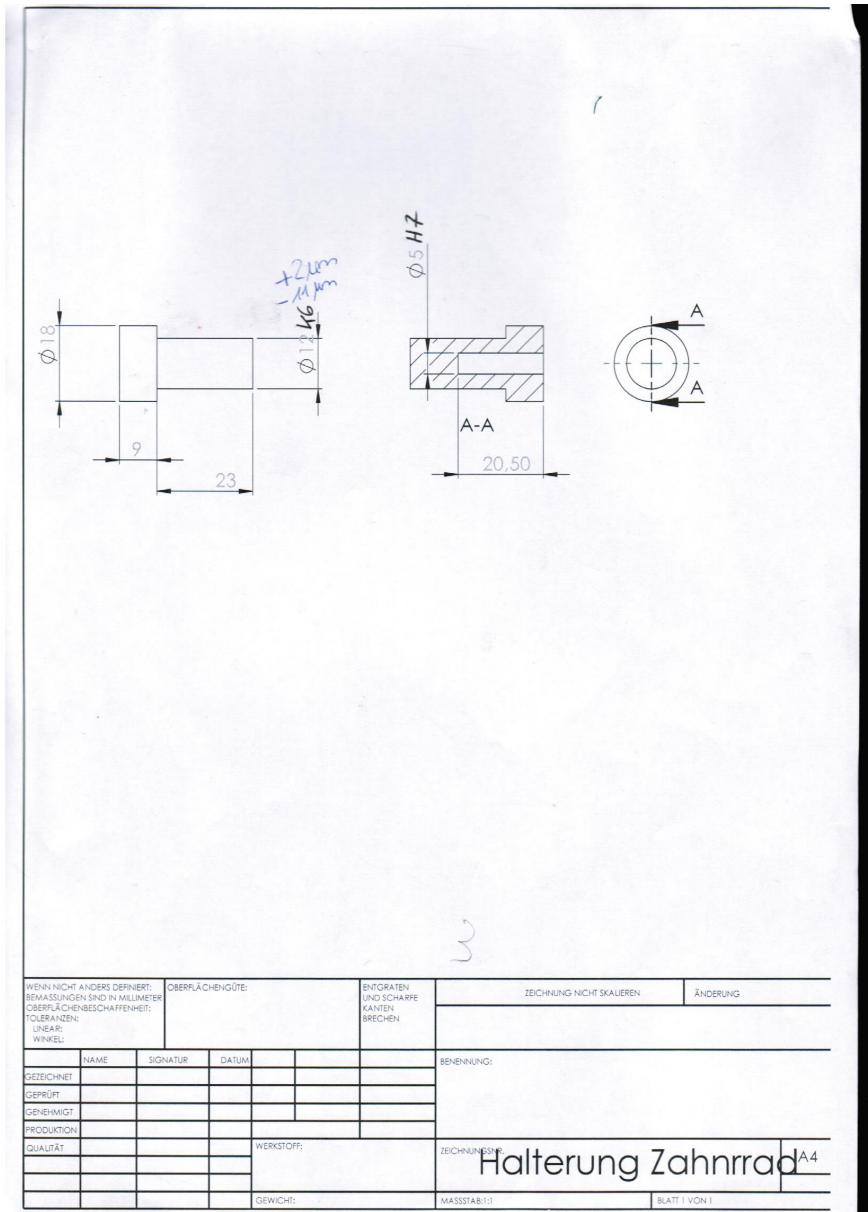
Die Funktion RefillBox fährt den Ausgabeschacht an und wartet dort bis vom Benutzer ein Zeichen kommt, dass die Box befüllt wurde.

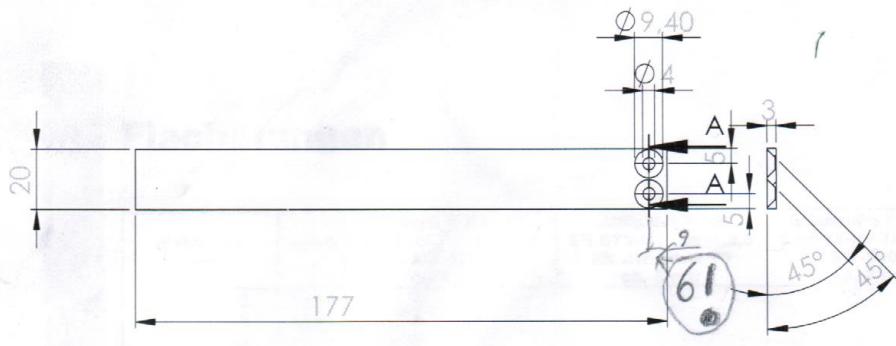
Der Ablauf von RefillBox ist fast identisch mit dem von EmptyBox, nur dass statt der Rotation, um die Box zu leeren, auf einen Tastendruck vom Benutzer gewartet wird - zur Bestätigung dass die Box befüllt wurde.

A. Anhang Mechanik

A.1. Drehgelenk und Greifer

Verschiedene Maße und wichtige Daten des Drehgelenks und des Greifers können aus den Technischen Zeichnungen und aus den Datenblättern der folgenden Seiten entnommen werden.

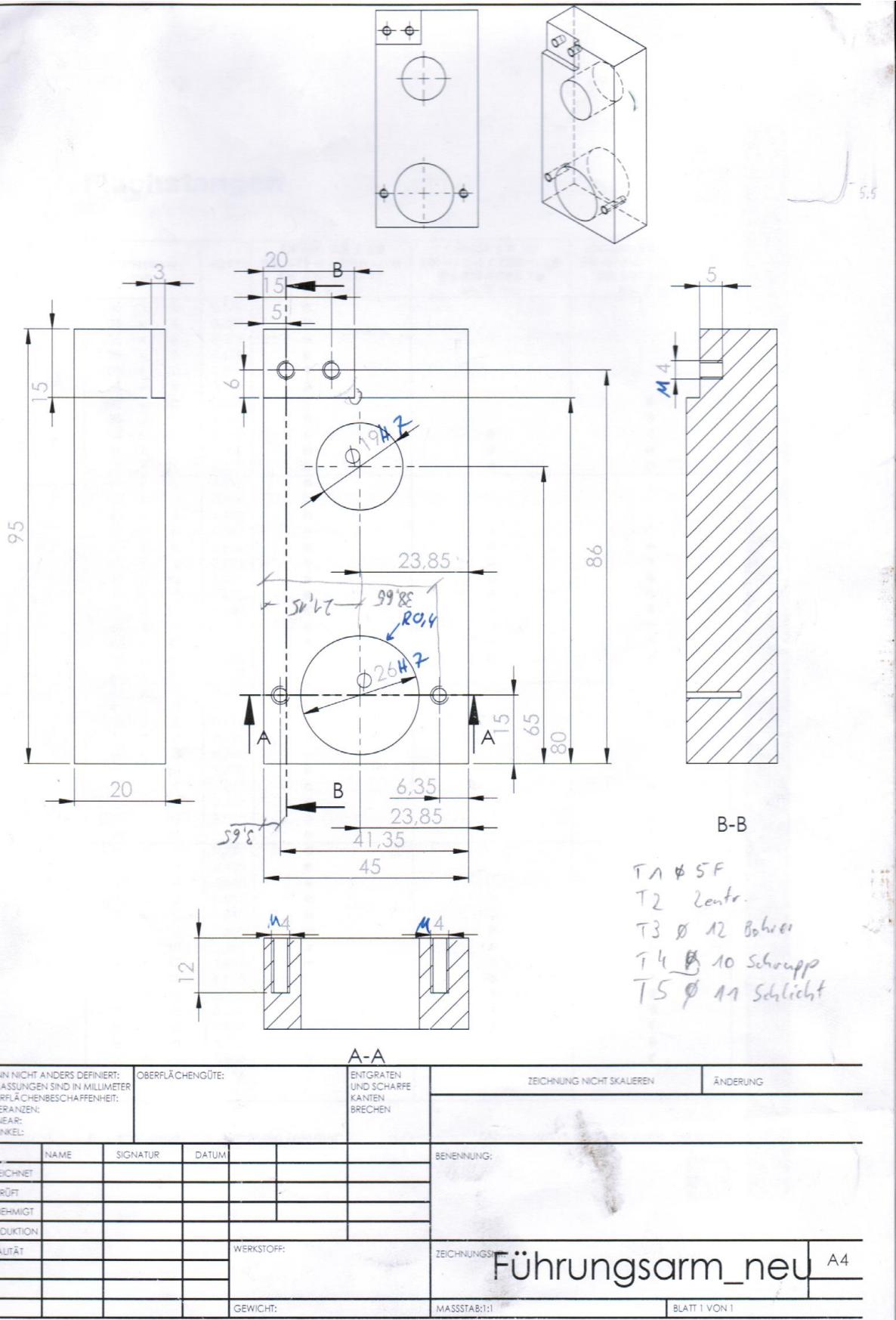


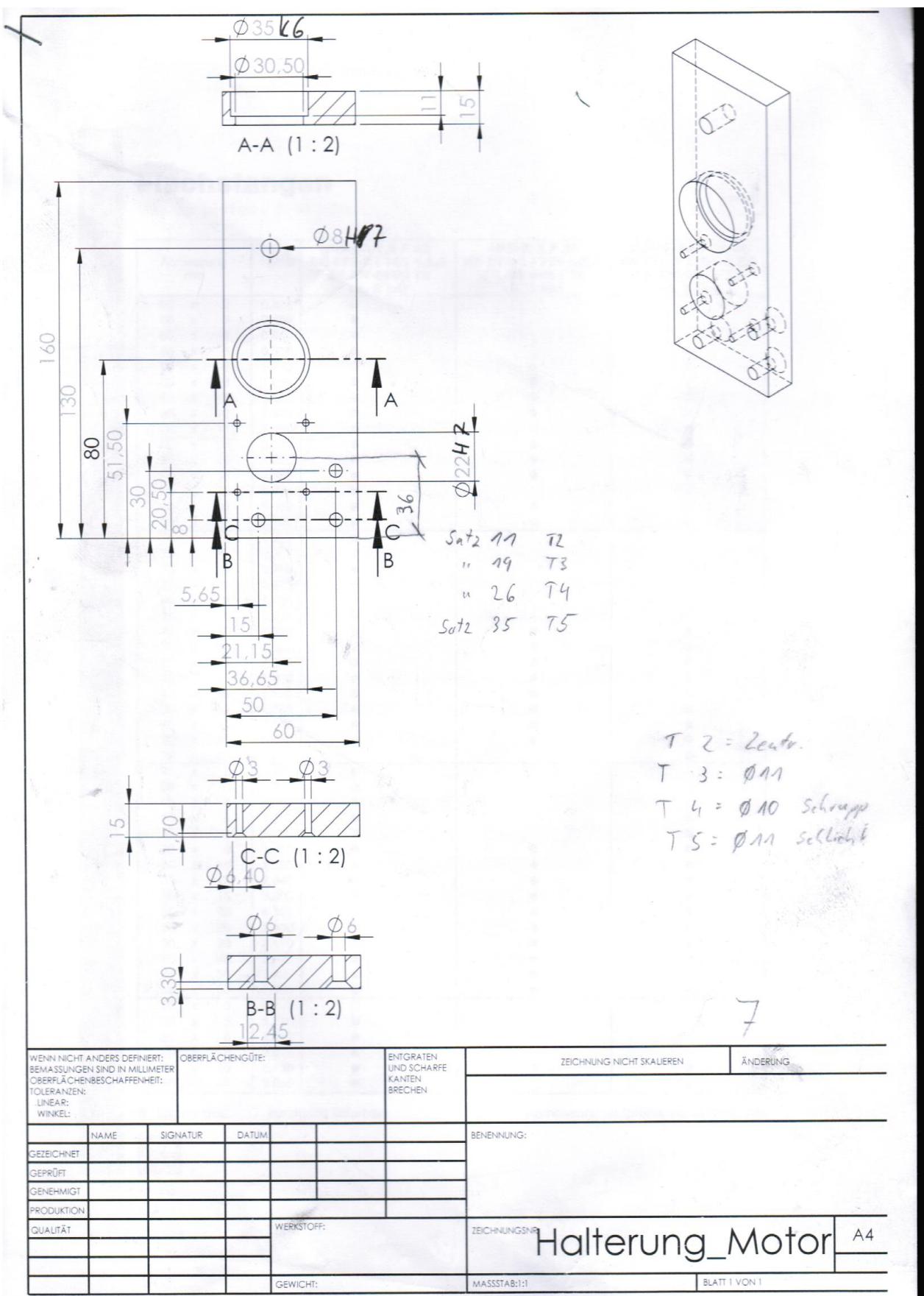


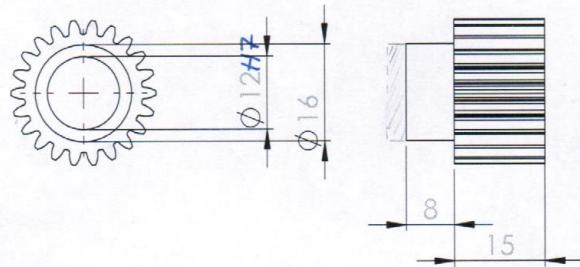
2 x Stück

2

WENN NICHT ANDERS DEFINIERT: BEMASSUNGEN SIND IN MILLIMETER OBERFLÄCHENBESCHAFFENHEIT: TOLERANZEN: LINEAR: WINKEL:		OBERFLÄCHENGÜTE:			ENTGRATEN UND SCHARFE KANTEN BRECHEN	ZEICHNUNG NICHT SKALIEREN	ÄNDERUNG
GEZEICHNET	NAME	SIGNATUR	DATUM				
GEPRÜFT							
GENEHMIGT							
PRODUKTION							
QUALITÄT				WERKSTOFF: <i>Al</i>	ZEICHNUNGSNR. Greifarm1	MASSSTAB:1:1	A4
				GEWICHT:		BLATT 1 VON 1	

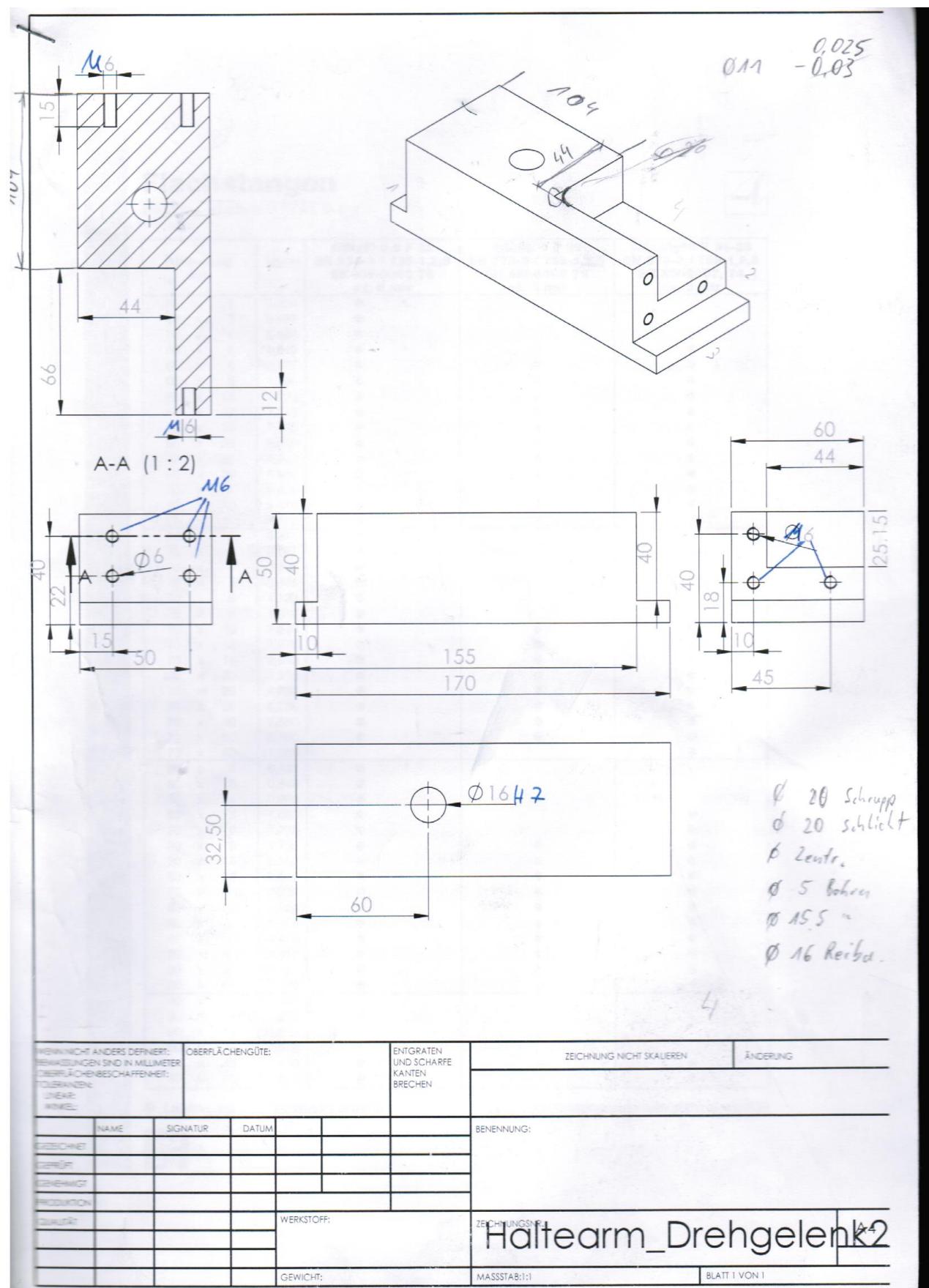


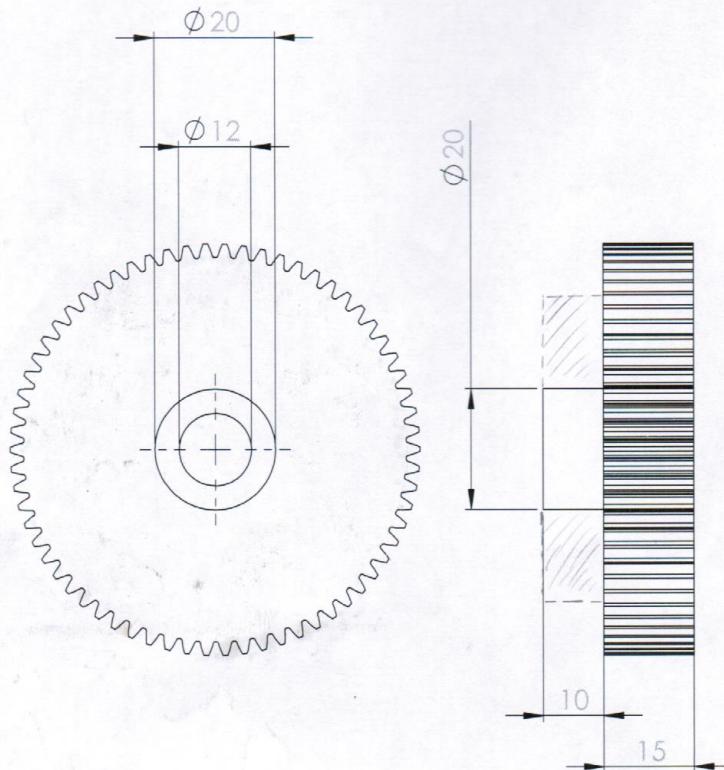




1. JHiru 2mm verkürzen
2. Bohrung Ø12 H7

WENN NICHT ANDERS DEFINIERT: BEMASSUNGEN SIND IN MILLIMETER OBERFLÄCHENBESCHAFFENHEIT: TOLERANZEN: LINEAR; WINKEL:			OBERFLÄCHENGÜTE:		ENTGRATEN UND SCHARFE KANTEN BRECHEN	ZEICHNUNG NICHT SKALIEREN	ÄNDERUNG
GEZEICHNET	NAME	SIGNATUR	DATUM				
GEPRÜFT						BENENNUNG:	
GENEHMIGT							
PRODUKTION							
QUALITÄT			WERKSTOFF:	C45		ZEICHNUNGSNR.	416-010-0022
			GEWICHT:				
						A4	



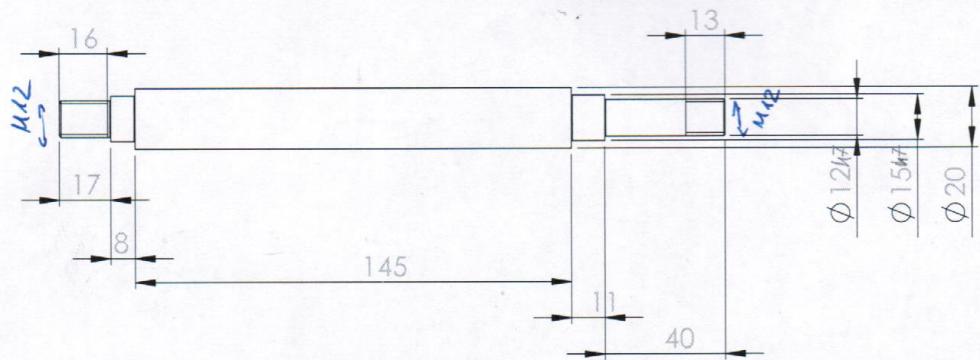


1. Stirn von Ø50 auf Ø20 verkleinern

WENN NICHT ANDERS DEFINIERT: BENASCHUNGEN SIND IN MILLIMETER OBERFLÄCHENBESCHAFFENHEIT: TOLERANZEN: LINEAR; WINDEL;		OBERFLÄCHENGÜTE:			ENTGRATEN UND SCHARFE KANTEN BRECHEN	ZEICHNUNG NICHT SKALIEREN	ÄNDERUNG
GEZEICHNET	NAME	SIGNATUR	DATUM				
GEPRÜFT							
GENEHMIGT							
PRODUKTION							
QUALITÄT				WERKSTOFF: <i>C45</i>		ZEICHNUNGSNR. <i>416-010-000200 0066</i>	A4
				GEWICHT:		MASSSTAB:1:1	BLATT 1 VON 1

1 Trapezgewinde

$L = 221 \text{ mm}$

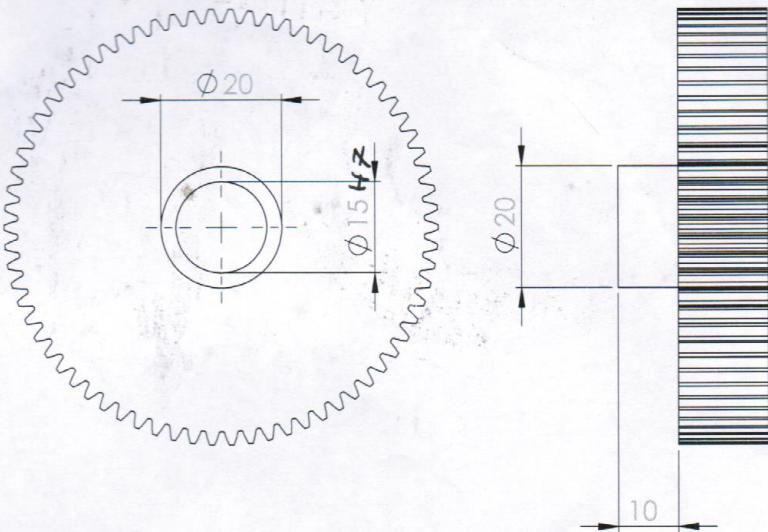


1. 2x M12 Gewinde

(bestellt)

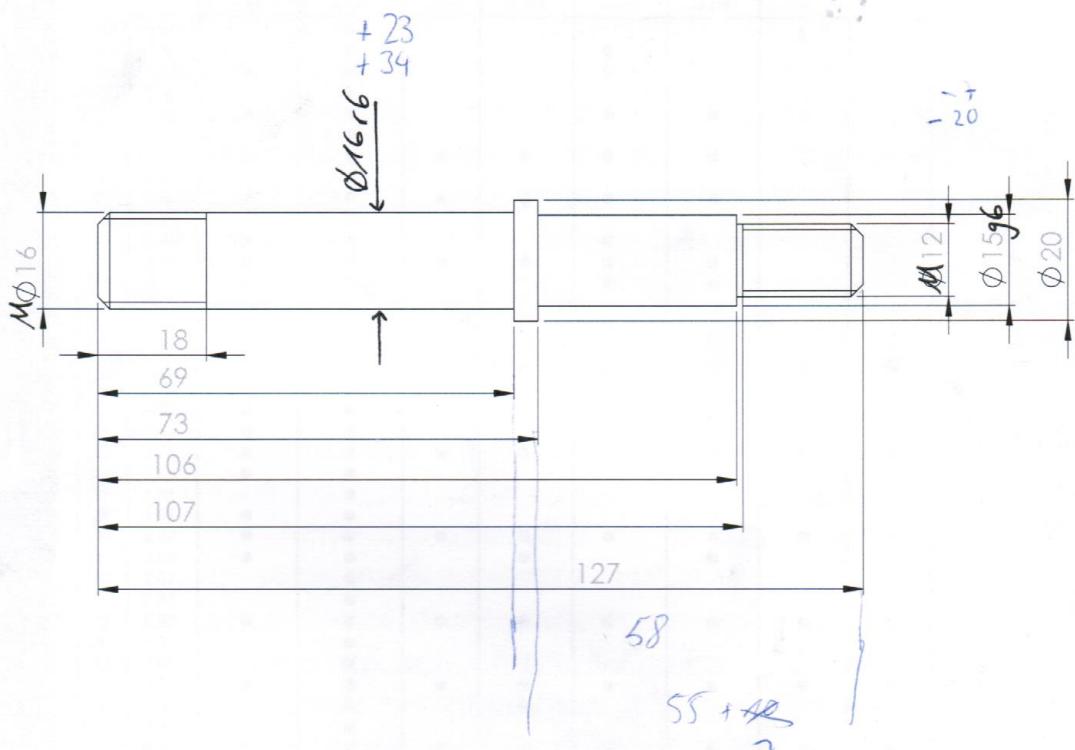
5

WENN NICHT ANDERS DEFINIERT: BEMASSTUNGEN SIND IN MILLIMETER OBERFLÄCHENBESCHAFFENHEIT: TOLERANZEN: LINEAR: WINKEL:			OBERFLÄCHENGÜTE:		ENTGRATEN UND SCHARFE KANTEN BRECHEN	ZEICHNUNG NICHT SKALIEREN	ÄNDERUNG
GEZEICHNET	NAME	SIGNATUR	DATUM				
GEPRÜFT							
GEHEMMIGT							
PRODUKTION							
QUALITÄT				WERKSTOFF: <i>CSE</i>		ZEICHNUNGSNR. Welle_bestellt	A4
				GEWICHT:		MASSSTAB:1:1	BLATT 1 VON 1

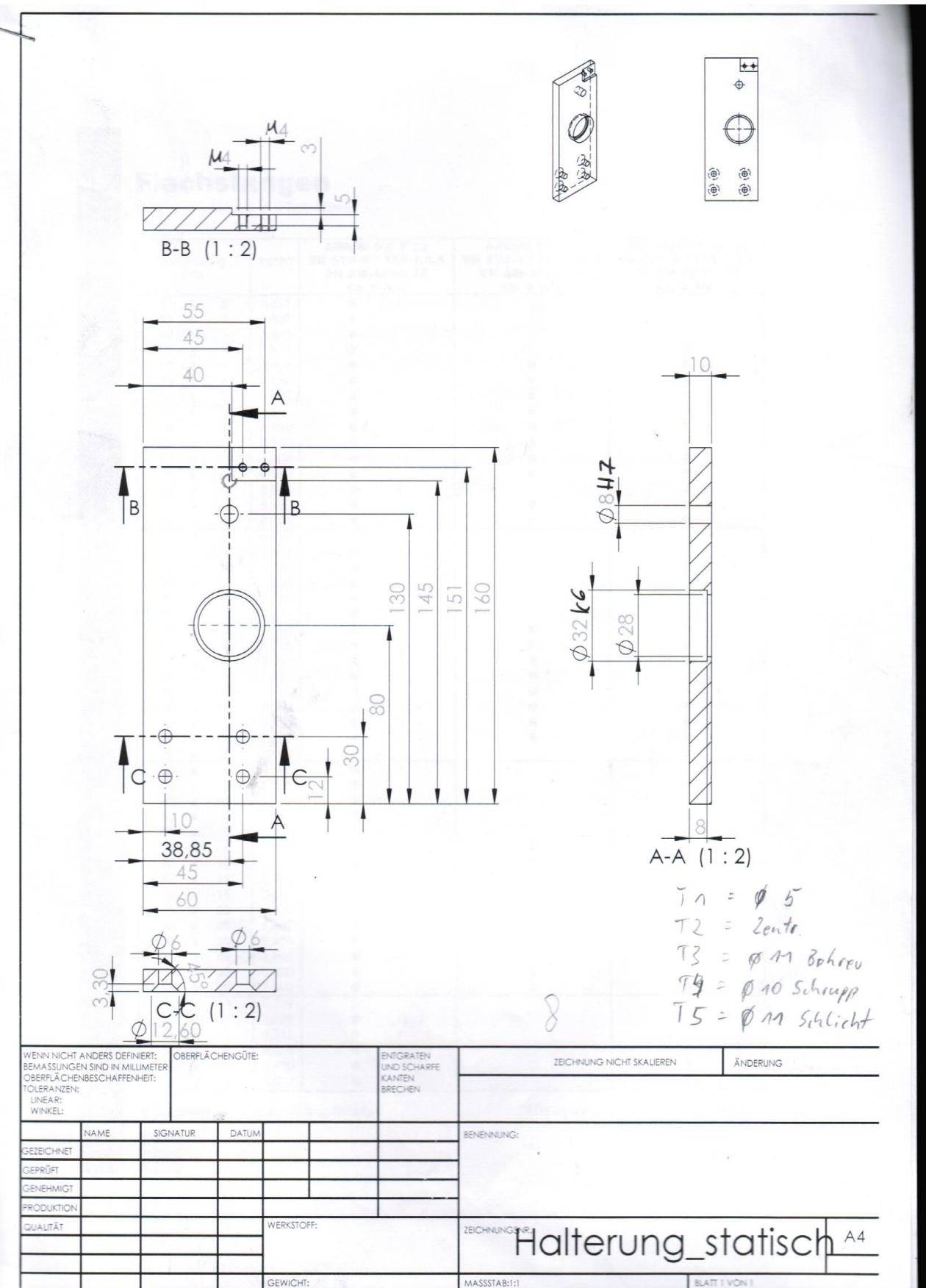


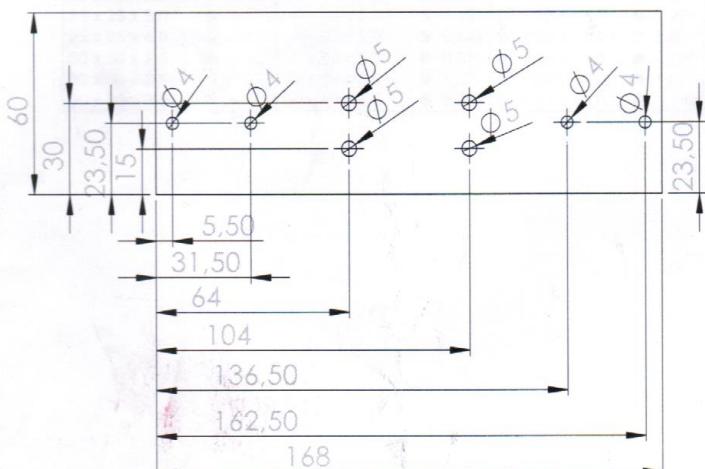
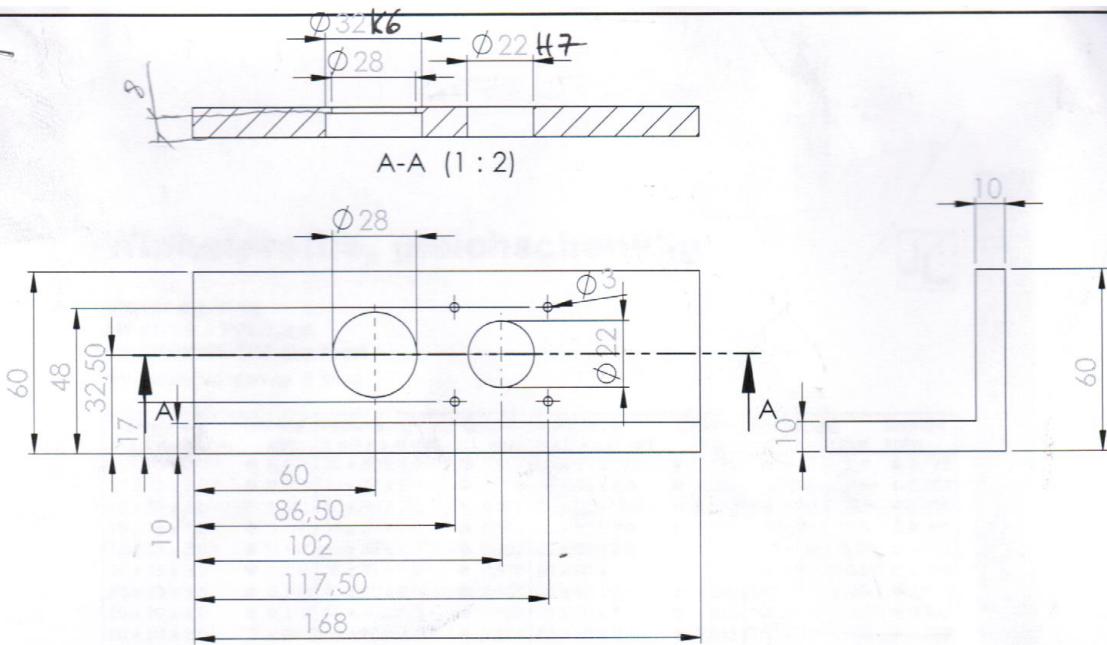
1. Stirn verkleinen auf Ø20
2. Bohrung auf Ø15 H7

WENN NICHT ANDERS DEFINIERT: BEMASSTUNGEN SIND IN MILLIMETER OBERFLÄCHENBESCHAFFENHEIT: TOLERANZEN: LINEAR; WINKEL:			OBERFLÄCHENGÜTE:		ENTGRATEN UND SCHARFE KANTEN BRECHEN	ZEICHNUNG NICHT SKALIEREN	ÄNDERUNG
GEZEICHNET	NAME	SIGNATUR	DATUM				
GEPRÜFT							
GENEHMIGT							
PRODUKTION							
QUALITÄT				WERKSTOFF:		ZEICHNUNGSNR.	416-010-0070
				GEWICHT:		MASSSTAB:1:1	BLATT 1 VON 1
							A4



WENN NICHT ANDERS DEFINIERT: BEMASSTUNGEN SIND IN MILLIMETER OBERFLÄCHENBESCHAFFENHEIT: TOLERANZEN: LINEAR: WINKEL:		OBERFLÄCHENGÜTE:		ENTGRATEN UND SCHARFE KANTEN BRECHEN	ZEICHNUNG NICHT SKALIEREN	ÄNDERUNG
GEZEICHNET	SIGNATUR	DATUM				
GEPRÜFT						
GENEHMIGT						
PRODUKTION						
QUALITÄT			WERKSTOFF:		BENENNUNG:	
					ZEICHNUNGSNR.	
					Welle_drehgelenk_neu	A4
			GEWICHT:		MASSSTAB:1:1	BLATT 1 VON 1



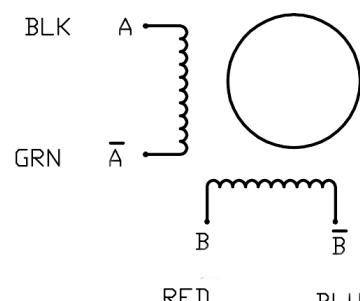
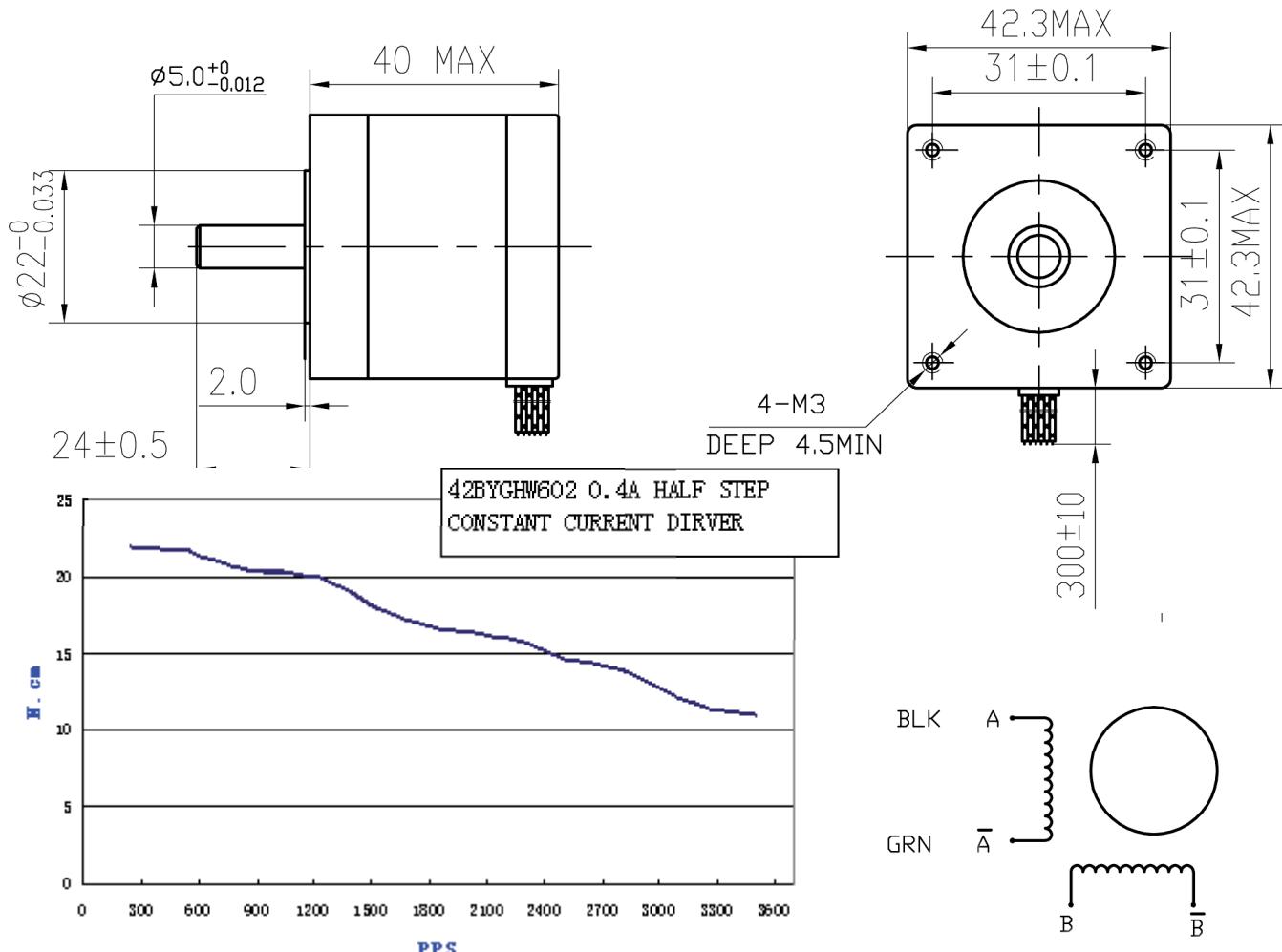


4 Senkkanten
für Senkkopf-
schrauben

WENN NICHT ANDERS DEFINIERT: BEMÄSSUNGEN SIND IN MILLIMETER OBERFLÄCHENBESCHAFFENHEIT: TOLERANZEN: LINEAR: WINKEL:		OBERFLÄCHENGÜTE:		ENTGRATEN UND SCHARFE KANTEN BRECHEN	ZEICHNUNG NICHT SKALIEREN	ÄNDERUNG
ZEICHNET	NAME	SIGNATUR	DATUM			
GEPRÜFT				BENENNUNG:		
EBENEHMIGT						
PRODUKTION						
QUALITÄT		WERKSTOFF: <i>AlMg</i>		ZEICHNUNGSNR.	Motorhalterung	A4
		GEWICHT:		MASSTAB: 1:1	BLATT 1 VON 1	

Schrittmotor PSM42BYGHW603

Best.Nr. 310 453

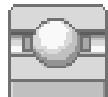


PHASE	2 PHASE
STEP ANGLE	$1.8 \pm 5\%^\circ/\text{STEP}$
VOLTAGE	12V
CURRENT	0.4 A/PHASE
RESISTANCE	$30 \pm 10\% \Omega/\text{PHASE}$
INDUCTANCE	$40 \pm 20\% \text{ mH}/\text{PHASE}$
HOLDING TORQUE	38 N.cm Min
DETENT TORQUE	2.2 N.cm Max
INSULATION CLASS	B
LEAD STYLE	AWG26 UL1007
ROTOR TORQUE	54 g.cm ²



Diese Bedienungsanleitung ist eine Publikation von Pollin Electronic GmbH, Max-Pollin-Straße 1, 85104 Pförring.
Alle Rechte einschließlich Übersetzung vorbehalten. Reproduktion jeder Art, z.B. Fotokopie, Mikroverfilmung oder die
Erfassung in elektronischen Datenverarbeitungsanlagen, bedürfen der schriftlichen Genehmigung des Herausgebers.
Nachdruck, auch auszugsweise, verboten. Diese Bedienungsanleitung entspricht dem technischen Stand bei Drucklegung.
Änderung in Technik und Ausstattung vorbehalten.

©Copyright 2011 by Pollin Electronic GmbH

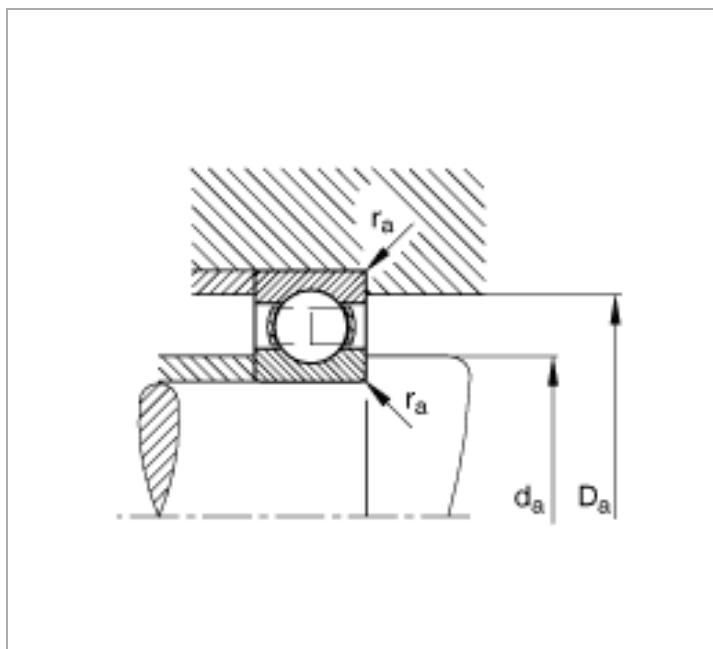
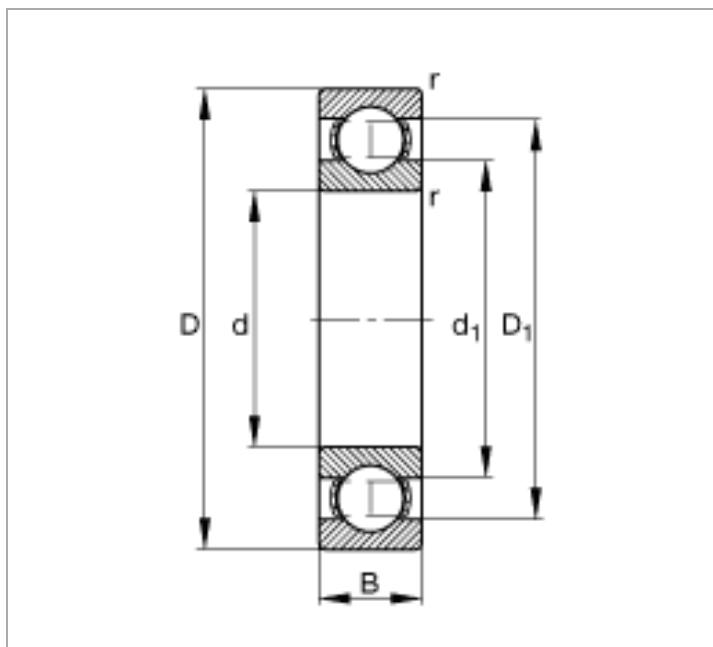


Rillenkugellager 16002 (Baureihe 160)

Hauptabmessungen nach DIN 625-1

Das vorliegende Datenblatt ist nur eine Übersicht über Maße und Tragzahlen zum gewählten Produkt. Bitte beachten Sie unbedingt alle Hinweise in diesen Übersichtsseiten. Weiterführende Informationen finden Sie für viele Produkte unter dem Menüpunkt "Beschreibung". Außerdem können Sie umfangreiches Informationsmaterial auch über die Katalogauswahl (<http://www.fag.de/content.fag.de/de/mediathek/library/library.jsp>) oder Telefon +49 (91 32) 82 - 28 97 bestellen.

d	15 mm
D	32 mm
B	8 mm
D ₁	26,9 mm
d ₁	20,5 mm
D _a	30 mm
max	
d _a	17 mm
min	
r _a	0,3 mm
max	
r _{min}	0,3 mm
m	0,027 kg Gewicht
C _r	5900 N dynamische Tragzahl, radial
C _{0r}	2850 N statische Tragzahl, radial
f ₀	13,9
n _G	33500 1/min Grenzdrehzahl
n _B	20000 1/min Bezugsdrehzahl
C _{ur}	144 N Ermüdungsgrenzbelastung, radial



Spindelmuttern für Trapezgewindetriebe

-mit Flansch · mit Kompaktflansch · mit Führung (Gewindevorbohrung oder Langloch)-



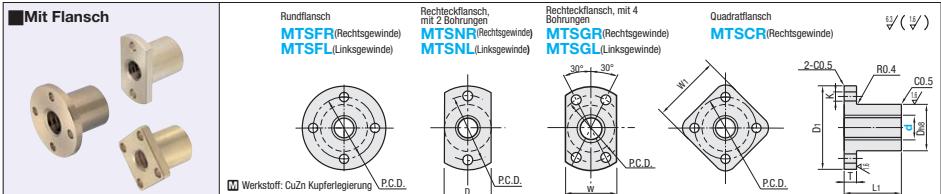
Spindelmuttern für Trapezgewindetriebe, RoHS-konform

-mit Flansch/Feingewinde/spielfrei-

CAD-Daten

Verfügbare Flanschformen: Rundflansch, Kompaktflansch mit 2 Bohrungen, Kompaktflansch mit 4 Bohrungen, Quadratflansch mit 4 Bohrungen.

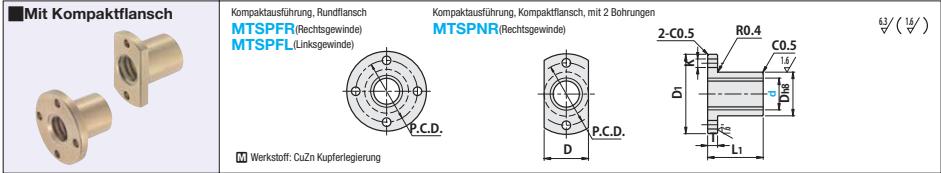
Mit Flansch



Teilenummer	Ausführung	D	Steigung P	D	L1	D1	T	Lochkreis-Ø	K	W	W1	Zulässige dynamische Axiallast (kN)	Gewicht (g)	€ Stückpreis	
									MTSFR	MTSFL	MTSNR	MTSNL	MTSGR	MTSGL	MTSCR
Rundflansch	(8)	1.5	15	20	30			22	4.3	-	-	1.47	41	33	-
Rechteckflansch, mit 2 Bohrungen	10	2	20	24	36		5	26	-	2.55	80	66	67	-	
MTSFR	12	2	20	30	44			-	3.92	120	95	96	-	-	
MTSFL	14	3	28	35	51		6	31	5.4	24	4.90	110	85	91	
Kompaktflansch, mit 4 Bohrungen	16	4	32	40	56		6	30	38	6.67	200	169	172	169	
MTSNR	18	4	32	40	56		6	34	42	9.81	260	219	220	-	
MTSGL	20	4	36	50	61	7	47	40	47	12.36	410	357	364	366	
Quadratflansch	22	4	44	56	76	8	58	9	48	58	17.95	630	538	546	548
MTSCR	25	5	56	60	84	66	-	21.08	580	490	498	-	-	-	
(*) Nur die mit gekennzeichneten Größen sind Quadratflanschführung verfügbar.	28	6	58	70	98	10	76	62	33.83	1250	1034	1044	-	-	
MTSFL	32	8	68	80	109	85	72	40.31	1631	1350	1362	-	-	-	

Kompakte Länge, kompakter Durchmesser. Platzsparend durch Feingewinde in Befestigungsbohrung

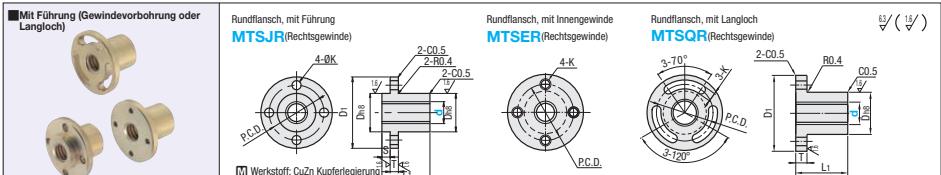
Mit Kompaktflansch



Teilenummer	Ausführung	D	Steigung P	D	L1	D1	T	Lochkreis-Ø	K	Zulässige dynamische Axiallast (kN)	Gewicht (g)	€ Stückpreis	
									MTSPFR(L)	MTSPFL	MTSPNR	MTSPFL	MTSPNR
Rundflansch	*10	2	16	19	32	4	24	3.3	-	2.02	39	-	-
MTSPFR	12	2	18	20	36		27	-	3.14	59	-	-	-
MTSPFL	14	3	20	24	38	5	31	4.3	-	3.92	73	57	-
Rechteckflansch, mit 2 Bohrungen	16	3	22	28	40		35	-	5.34	89	73	-	-
MTSPNR	20	4	26	32	44		39	5.4	-	7.85	112	94	-
(*) Alle mit angegebenen Größen 1 sind nur für Ausführung MTSPR verfügbar.	22	4	28	40	55	6	42	5.4	-	9.89	174	143	-
MTSPR	25	5	31	40	53		42	-	-	11.36	174	143	-
MTSPR	28	5	34	45	58	7	46	6.6	-	14.42	213	170	-
MTSPR	32	6	38	45	62	50	50	-	-	16.94	272	227	-

(*) Die Ausführung mit Führung und Gewindevorbohrung ist für die vertikale Befestigung auf Platten vorgesehen. Die Ausführung mit Langloch ist für Anwendungen, bei denen während der Installation Feineinstellungen erforderlich sind.

Mit Führung (Gewindevorbohrung oder Langloch)



Teilenummer	Ausführung	D	Steigung P	D	L1	D1	T	S	Lochkreis-Ø	K	Zulässige dynamische Axiallast (kN)	Gewicht (g)	€ Stückpreis			
									MTSJR	MTSER	MTSQR	MTSER	MTSQR			
Mit Führung	*14	3	22	30	44	44	5	5	33	31	5.4	M4	4.90	110	112	98
Mit Führung	*16	3	28	35	52	51	6	6	40	38	-	6.67	204	174	178	-
Mit Führung	*20	4	32	40	56	56	6	6	44	42	6.6	M5	9.81	260	264	236
Mit Führung	*22	4	36	50	60	61	7	7	48	47	-	12.36	404	414	378	-
Mit Führung	*25	5	36	44	56	-	76	8	-	14.22	344	354	318	-	-	
Mit Führung	28	5	44	56	-	76	8	-	58	9	M6	17.95	-	645	570	-
Mit Führung	32	6	44	56	-	76	8	-	-	21.08	-	595	520	-	-	

Bestellbeispiel

Teilenummer MTSR16
MTSPF125

Lieferzeit 6 Arbeits-tage

Mit Flansch 6 Arbeits-tage

außer Flanschausführung 8 Arbeits-tage

Express A

5.00 EUR/Stück

€ S.87

(*) Expressgebühr von 13,50 EUR für 3 oder mehr identische Teile.



Preis
Stückz. 1-9 10-15 16-20 21-29
Rabatt 5% 10% 18%

Bei noch größeren Bestellmengen bitte gesondert anfragen.

CAD-Daten

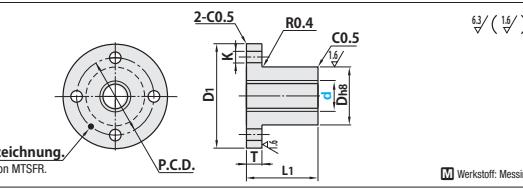
RoHS-konforme Trapezgewindetriebe-Spindelmutter mit Flansch. Lieferung erfolgt nach kürzester Durchlaufzeit.

Mit Flansch



RoHS

MTRFR (Rechtsgewinde)
MTXFR (Rechtsfeingewinde)



Werkstoff: Messing

Teilenummer	Ausführung	D	Steigung P	D	L1	D1	T	Lochkreis-Ø	K	Zulässige dynamische Axiallast (kN)	Gewicht G	€ Stückpreis
									2-C0.5	R0.4	C0.5	
MTRFR	10	2		20	24	36	5	26	4.3	2.55	80	
MTRFR	12	2		22	30	44	5	31	5.4	3.92	120	
MTRFR	14	3		28	35	51	6	38	6.67	4.90	110	
MTRFR	16	4		32	40	56	6	42	6.6	9.81	260	
MTRFR	20	4		36	50	61	7	47	6.6	14.22	350	
MTRFR	25	5		44	56	76	8	58	9	17.95	630	
MTRFR	28	6		28	35	51	6	38	6.6	21.08	580	
MTRFR	32	6		32	40	56	6	42	10.1	6.78	190	
MTXFR	16	2		28	35	51	6	38	6.6	10.1	250	
MTXFR	20	2		32	40	56	6	42	6.6	10.1	250	

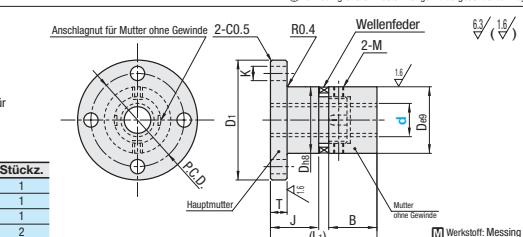
Bei noch größeren Bestellmengen bitte gesondert anfragen.

Spieldfrei



RoHS

MTBLR (Rechtsgewinde)

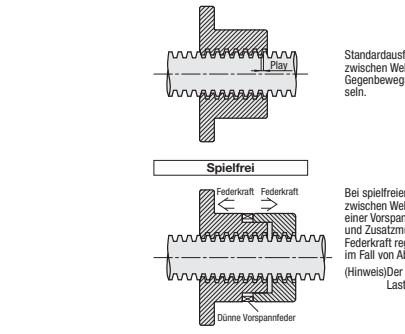


Werkstoff: Messing

Teilenummer	Ausführung	D	Steigung P	D	L1	D1	T	(L1)	J	B	Lochkreis-Ø	K	M	Zulässige dynamische Axiallast (kN)	Gewicht g	€ Stückpreis
											2-C0.5	R0.4	2-M			
MTBLR	10	2		20	36	5	34	13	15	26	4.3	3	2.60	100		
MTBLR	12	2		22	44	6	36.5	16.5	16	31	5.4	3	3.39	130		
MTBLR	16	3		28	51	6	45	21	20	38	6.6	4	6.29	230		
MTBLR	20	4		32	56	6	52	24	25	42	6.6	4	9.32	310		

Bei noch größeren Bestellmengen bitte gesondert anfragen.

Montage der spieldreien Ausführung



Bei spielfreier Ausführung kann das Spiel zwischen Welle und Nut mithilfe einer Vorspannfeder zwischen Haupt- und Zusatzmutter eliminiert werden. Die Federkraft regelt Gegenbewegungen auch im Fall von Abrieb.

(Hinweis) Der Effekt ist abhängig vom Lastzustand (Last und Richtung).

Um die Herausfallen der Stellschraube aufgrund von Vibrationen u. ä. zu vermeiden,

Um ein Herausfallen der Stellschraube aufgrund von Vibrat

ionen zu verhindern, wird die Stellschraube gelöst und aktiviert die Federkraft.

(Achtung) Die montierte Stellschraube darf nicht aus dem Außen- oder Innenbereich der Zusatzmutter herausstehen.

Um ein Herausfallen der Stellschraube aufgrund von Vibrat

ionen zu verhindern, wird die Stellschraube gelöst und aktiviert die Federkraft.

Um ein Herausfallen der Stellschraube aufgrund von Vibrat

ionen zu verhindern, wird die Stellschraube gelöst und aktiviert die Federkraft.

Testbedingungen: Probemutter: MTBLR16 Achse: MTSR16-270
Verfahrweg: 75mm

Positionierfehlervergleich unter Last (vorwärts)
(Standard VS spielfrei)



Die Wiederholgenauigkeit wird durch den spieldreien Aufbau erhöht.

(2) Vergleich Fehler Wiederholgenauigkeit (Referenzwert)



Die Wiederholgenauigkeit wird durch den spieldreien Aufbau erhöht.

Positionierfehlervergleich unter Last (vorwärts)
(Standard VS spielfrei)

Die Wiederholgenauigkeit wird durch den spieldreien Aufbau erhöht.

Positionierfehlervergleich unter Last (vorwärts)
(Standard VS spielfrei)

Die Wiederholgenauigkeit wird durch den spieldreien Aufbau erhöht.

Positionierfeh



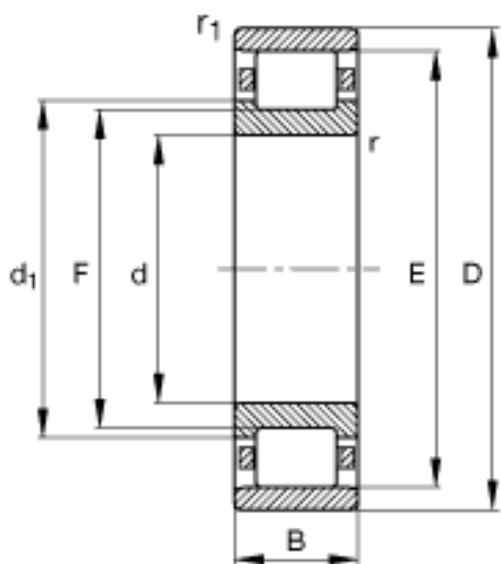
Zylinderrollenlager N202-E-TVP2 (Baureihe N2..-E)

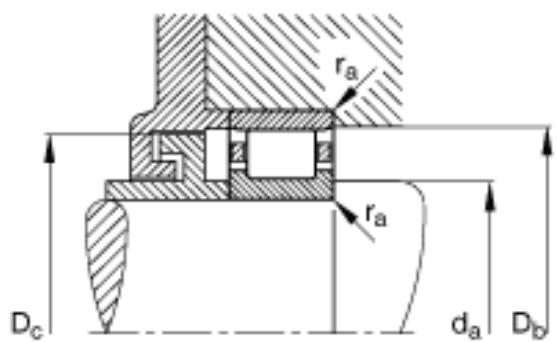
Hauptabmessungen nach DIN 5412-1, Loslager, zerlegbar,
mit Käfig

Das vorliegende Datenblatt ist nur eine Übersicht über Maße und Tragzahlen zum gewählten Produkt. Bitte beachten Sie unbedingt alle Hinweise in diesen Übersichtsseiten. Weiterführende Informationen finden Sie für viele Produkte unter dem Menüpunkt "Beschreibung". Außerdem können Sie umfangreiches Informationsmaterial auch über die Katalogauswahl (<http://www.fag.de/content.fag.de/de/mmediathek/library/library.jsp>) oder Telefon +49 (91 32) 82 - 28 97 bestellen.

d	15 mm
D	35 mm
B	11 mm
d ₁	21,6 mm
D _a	32,6 mm
max	
d _a	17,4 mm
min	
D _b	31 mm
min	
D _c	29 mm
max	
E	30,3 mm
F	19,3 mm
r ₁	0,3 mm
min	
r _{a1}	0,3 mm
max	
r _a	0,6 mm
max	
r _{min}	0,6 mm
s	0,5 mm Axiale Verschiebbarkeit aus der Mittellage
m	0,047 kg Gewicht
C _r	15100 N dynamische Tragzahl, radial
C _{0r}	10400 N statische Tragzahl, radial

nc	27000 1/min Grenzdrehzahl
nb	17600 1/min Bezugsdrehzahl
Cur	1610 N Ermüdungsgrenzbelastung, radial





Linearkugellager Standardausführung

-einfach/doppelt-

■ Auswahl der Distanzhalter und Anschläge für Linearkugellager siehe S.288

■ Einfach

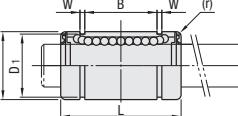


■ Doppelt

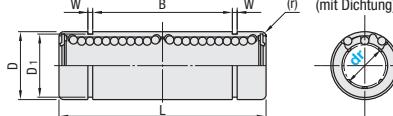


Ausführung	Außenzyliner		Stahlkugeln	Kugelfälg	Umgebungs- betriebstem.	A Zubehör
	Einfach	Doppelt	M Werkstoff	Härte	S Oberflächenbehandlung	M Werkstoff
LMU-N LMUW-N	LMUW					
LMUF LMUWF	LMUWF	1.3505/100Cr6	58HRC-		1.3505/100Cr6	Kunststoff (Duracon M90 oder gleichwertig)
LMUR LMUWR	LMUWR				1.4125/X105CrMo17	-20-80°C
LMUM LMUWM	LMUWM				1.4125/X105CrMo17	-20-110°C
SLMU SLMUW	SLMUW	1.4125/X105CrMo17	58HRC-		Kunststoff (Duracon M90 oder gleichwertig)	-20-80°C
SLMUS SLMUWS	SLMUWS				1.4112/X105CrAlA17-7	-20-120°C

Einfache Ausführung



Doppelte Ausführung



(*) Keine Dichtung bei dr=3 und 4, LMU-N, LMUF-N und LMUW-N.

(*) Eigenschaften von LTBC siehe S.278

■ Einfach (■) Für Linearkugellager werden gehärtete Wellen mit g6-Toleranz empfohlen. S.107-195

Teilenummer	D-Toleranz								L Toleranz	B Toleranz	Radialversatz (R)	Stahlkugeln Kugelreihen	Tragzahl C (dynamisch) Co (statisch)	Gewicht G	■ Stückpreis	
	Ausführung	dr	Toleranz	obere Schicht behandlung	oberflächen- behandlung	w	d1	(R)								
LMU-N (ohne Dichtung)	3	0	7	0	10	-	-	-	10	-	-	69	105	1.4		
LMU-N (ohne Dichtung)	4	-0.008	8	-0.009	-0.015	12	-0.12	-	15	8	9.6	88	127	2.0		
LMU-N (ohne Dichtung)	5	10	-	-	-	-	-	-	11	11.5	0.4	167	206	4.0		
LMU-N (ohne Dichtung)	6	12	0	0	19	11.3	-	-	15	15.3	1.1	206	265	8.5		
LMU-N (ohne Dichtung)	8	15	-0.011	-0.018	24	14.3	-	-	19	19.4	0.4	265	380	17		
LMU-N (ohne Dichtung)	10	19	-	-	29	18	-	-	20	20.4	1.3	372	544	31		
LMU-N (ohne Dichtung)	12	-0.009	21	0	0	30	-0.2	-	22	20.4	1.3	412	598	41		
LMU-N (ohne Dichtung)	13	23	-0.013	-0.021	32	22	-	-	27	23.3	1.6	510	784	46		
SLMU	16	28	-	-	37	27	-	-	30.5	30.5	1.6	775	1180	73		
SLMUS	20	32	-	-	42	27.3	-	-	30.5	30.5	1.6	882	1370	98		
SLMUS	25	40	0	-0.016	-0.025	59	37.3	-	38	38	1.6	980	1570	236		
SLMUS	30	45	-	-	64	40.8	-	-	43	43	1.5	1570	2740	262		
LMU	35	52	-	-	70	45.3	0	-0.5	49	49	1.5	1670	3140	425		
LMU	40	-0.012	60	-0.019	-0.030	80	56.3	-	57	57	1.5	2160	4020	654		
LMU	50	-0.015	80	-	100	68.8	-	-	76.5	76.5	1.6	3820	7940	1700		

kgf=Nx0.101972

■ Doppelt (■) Für Linearkugellager werden gehärtete Wellen mit g6-Toleranz empfohlen. S.107-195

Teilenummer	D-Toleranz								L Toleranz	B Toleranz	Radialversatz (R)	Stahlkugeln Kugelreihen	Tragzahl C (dynamisch) Co (statisch)	Gewicht G	■ Stückpreis	
	Ausführung	dr	Toleranz	obere Schicht behandlung	oberflächen- behandlung	w	d1	(R)								
LMUW-N (ohne Dichtung)	3	7	-	19	-	-	-	-	10	-	-	138	210	0.51	3.2	
LMUW-N (ohne Dichtung)	4	8	-0.011	-0.015	23	-	-	-	15	9.6	0.4	176	254	0.63	4.8	
LMUW-N (ohne Dichtung)	5	10	-	-	28	18.2	-	-	18	11.5	1.1	263	412	1.38	11	
LMUW-N (ohne Dichtung)	6	12	0	0	35	24.8	-	-	20	14.3	1.1	324	529	2.18	16	
LMUW-N (ohne Dichtung)	8	15	-0.013	-0.018	45	0	0.3	0	22	20	1.3	431	784	4.31	31	
LMUW-N (ohne Dichtung)	10	19	-	-	55	41.4	-0.5	-	27	20	1.3	588	1100	7.24	62	
LMUW-N (ohne Dichtung)	12	21	0	0	57	43.4	-	-	27	22	1.3	657	1200	10.9	80	
LMUW-N (ohne Dichtung)	13	23	-0.016	-0.021	61	48.8	-	-	27	22	1.3	813	1570	11.6	90	
LMUW-N (ohne Dichtung)	16	28	-	-	70	57.8	-	-	30.5	30.5	1.6	1230	2350	19.7	145	
SLMUW	20	32	0	0	80	57.8	-	-	30.5	30.5	1.6	1400	2740	26.8	180	
SLMUW	25	40	-0.019	-0.025	112	78.3	-	-	38	43	1.5	1560	3140	43.4	440	
SLMUW	30	45	-	-	123	85.3	0	-0.5	43	48.8	1.5	2490	5490	82.8	480	
SLMUW	35	52	0	0	135	94.8	-	-	43	49.1	1.5	2650	6270	110	795	
SLMUW	40	-0.015	60	-0.022	-0.030	151	116.8	-	57	57	1.5	3430	8040	147	1170	
SLMUW	50	-0.015	80	-	192	142.8	-	-	76.5	76.5	1.6	6080	15900	397	3100	

kgf=Nx0.101972

■ Die Ausführung ohne Dichtung führt durch den Gleitwiderstand (0.4-1.2N) zu ungünstigen Bewegungen aus.

■ Teilenummer

LMUR20 LMU-N00 (ohne Dichtung) (Ausführung L mit Schmiertstoff)

LMUR20L LMU-N00 (ohne Dichtung) (Ausführung G mit Schmiertstoff)

LMUR20G LMU-N00 (ohne Dichtung) (Ausführung H mit Schmiertstoff)

Alternative Schmiertstoffarten verfügbar. Lieferzeit, Preis und Leistung siehe S.272

■ Lieferzeit

■ Preis

■ Mengenrabatt (■) Auf einen Cent abgerundet S.87

■ Stückzahl 1-19 20-34 35-49 50-99

■ Rabatt € Stückpreis 5% 10% 18%

■ Lieferzeit Standardausführung Angebot

■ Beispiel

Sicherungsring- Unterlegscheiben für Linearkugellager (LBS.S.288)

Gehäuse Gehäuseplatte (LMST.S.288)

Montageplatte (LMST.S.288)

■ Hinweise für die Anwendung des Linearkugellagers

(1) Wenn assembling with linear shafts, forcing the shaft into the bushing with angular misalignment may cause the ball retainers to deform and balls to fall out. Achten Sie darauf, jeweils die Mitte aussuzentrieren und die Welle vorsichtig einzuführen.

(2) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(3) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(4) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(5) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(6) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(7) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(8) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(9) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(10) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(11) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(12) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(13) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(14) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(15) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(16) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(17) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(18) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(19) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(20) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(21) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(22) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(23) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(24) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(25) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(26) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(27) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(28) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(29) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(30) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(31) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(32) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(33) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(34) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(35) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(36) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(37) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(38) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(39) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(40) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(41) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(42) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(43) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(44) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(45) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(46) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(47) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(48) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(49) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(50) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(51) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(52) Bei noch größeren Bestellmengen bitte gesondert anfragen.

(53) Bei noch größeren Bestellmengen bitte gesondert anfragen.



Achsen und Wellen -D Toleranz h9 (kalt gezogen)/h7 (geschliffen)/g6 (geschliffen)-

Beidseitig abgesetzt mit Außengewinde

■ h9 (kalt gezogen) und h7 (geschliffen) sind neu hinzugefügt. Auswahl aus h9 (kalt gezogen), h7 (geschliffen) und g6 (geschliffen) für Ihre Anwendung.

Ausführung	Toleranz	M	Werkstoff	Oberflächenbehandlung	Toleranztabelle				
					D/P	h9 (kalt-gezogen)	h7 (geschliffen)	g6 (Maske)	
SFRMHM SFRHMS	h9	h7	1.1191/C45E	brüniert	3	-0.025	0	-0.002	
PSFRHM PSFRHMS	-	-	1.4301/X5CrNi18-10	Chemisch vernickelt	3.1-6	0	0	-0.004	
SSFRHM SSFRHMS	-	-	1.4301/X5CrNi18-10	brüniert	6.1-10	0	-0.012	-0.012	
SFRHM SFRHMS	h7	h7	1.1191/C45E	Chemisch vernickelt	10.1-18	-0.036	0.015	-0.014	
SSFRHM SSFRHMS	-	-	1.4301/X5CrNi18-10	-	18.1-30	0	0	-0.026	
(4) PSFRHM	g6	g6 (Maske)	1.1191/C45E	Chemisch vernickelt	30.1-50	-0.043	-0.018	-0.007	
SFRM PSFRMS	-	-	1.4301/X5CrNi18-10	brüniert		-0.052	-0.021	-0.026	
SSFRM SSFRMS	-	-	1.7220/34CrMo4	brüniert		0	0	-0.009	
HFRM	-	-	1.4301/X5CrNi18-10	brüniert		-0.062	0	-0.025	
			1.7220/34CrMo4	härte 30-35HRC					

■ Oberflächenrauhigkeit von Teil D für h9 (kalt gezogen) ist Ra 6.3, für h7 (geschliffen) und g6 (geschliffen) ist Ra 1.6. (▽▽▽)

■ Standardausführung

■ Rundheit, Geradheit

D	Über	Bis	Rundheit M
13	13	20	0.04
20	20	40	0.05
40	40	50	0.06

■ Nicht verfügbar für h9 (kalt gezogen).

■ Rundheit bei Durchmesser D

D	Über	Bis	Rundheit M
13	13	20	0.04
20	20	40	0.05
40	40	50	0.06

■ Nicht verfügbar für h9 (kalt gezogen).

■ Konzentrität, Rechtwinkligkeit

■ Toleranzen von L, Y und anderen Maßen

Maße	Über	Bis	Maße Toleranz
6	30	30	±0.2
30	120	120	±0.3
120	400	400	±0.5
400	800	800	±0.8

■ Nicht verfügbar für h9 (kalt gezogen).

(1) Toleranz von D=h9 (kalt gezogen)/Toleranz von P,Q=h7 (2) Toleranz von D=h9 (kalt gezogen)/Toleranz von P,Q=g6

Teilenummer	0,1mm-Schritte			1mm-Schritte			P/Q wählbar	1mm-Schritte	Nur mit Schlüsselstellen	W	ε ₂ (Y) max.
	Ausführung	D	L	F/T	B	S					
(1) Teil D h9/Teil P,Q h7	6	20.0-290.0					für P<6	3 4 5		5	300
SFRHM SFRHMS	8	20.0-390.0					für P>6: B<Px3 & S<Qx3 & S<T-2	3 4 5 6		7	8 400
PSFRHM PSFRHMS	10	20.0-490.0					für P=3 oder 10: B<Px3 & S<Qx3 & S<T-2	4 5 6 8		8	500
SSFRHM SSFRHMS	12	25.0-590.0					für P=3 oder 10: B<Px3 & S<Qx3 & S<T-2	5 6 8 10		10	600
(Die ist nicht für SSFRHM verfügbar.)	15	25.0-690.0					für P=3 oder 10: B<Px3 & S<Qx3 & S<T-2	6 8 10 12		13	700
PSFRGMG SFRGMG SSFRGMG SSFRGMG	20	30.0-790.0					für P=12: B<Px3 & S<Qx3 & S<T-2	8 10 12 16 20		22	800
(Die ist nicht für SS-PSFRGMG verfügbar.)	25	30.0-790.0					für P=12: B<Px3 & S<Qx3 & S<T-2	8 10 12 16 20 24		27	15
	30	30.0-790.0					für P=12: B<Px3 & S<Qx3 & S<T-2	10 12 16 20 24 30		30	
	35	40.0-790.0					für P=12: B<Px3 & S<Qx3 & S<T-2	10 12 16 20 24 30		36	
	40	40.0-790.0					für P=12: B<Px3 & S<Qx3 & S<T-2	12 16 20 24 30		41	20

(3) Ausführung h7 (geschliffen)

Teilenummer	0,1mm-Schritte			1mm-Schritte			P/Q wählbar	1mm-Schritte	Nur mit Schlüsselstellen	W	ε ₂ (Y) max.
	Ausführung	D	L	F	B	S					
SFRHM SFRHMS	6	20.0-290.0					für P<6: B<Px3 & S<Qx3 & S<T-2	3 4 5		5	300
PSFRHM PSFRHMS	8	20.0-390.0					für P<6: B<Px3 & S<Qx3 & S<T-2	3 4 5 6		7	8 400
SSFRHM SSFRHMS	10	20.0-490.0					für P<6: B<Px3 & S<Qx3 & S<T-2	4 5 6 8		8	500
(Die ist nicht für SSFRHM verfügbar.)	12	25.0-590.0					für P<6: B<Px3 & S<Qx3 & S<T-2	5 6 8 10		10	600
PSFRGMG SFRGMG SSFRGMG SSFRGMG	15	25.0-690.0					für P<6: B<Px3 & S<Qx3 & S<T-2	6 8 10 12		13	700
(Die ist nicht für SS-PSFRGMG verfügbar.)	20	30.0-790.0					für P<6: B<Px3 & S<Qx3 & S<T-2	8 10 12 16 20		27	15 800
	25	30.0-790.0					für P<6: B<Px3 & S<Qx3 & S<T-2	8 10 12 16 20 24		30	
	30	30.0-790.0					für P<6: B<Px3 & S<Qx3 & S<T-2	10 12 16 20 24 30		36	
	35	40.0-790.0					für P<6: B<Px3 & S<Qx3 & S<T-2	12 16 20 24 30		41	20
	40	40.0-790.0					für P<6: B<Px3 & S<Qx3 & S<T-2	16 20 24 30		46	
	50	40.0-790.0					für P<6: B<Px3 & S<Qx3 & S<T-2	16 20 24 30		51	

(4) Ausführung g6 (geschliffen)

Teilenummer	0,1mm-Schritte			1mm-Schritte			P/Q wählbar	1mm-Schritte	Nur mit Schlüsselstellen	W	ε ₂ (Y) max.
	Ausführung	D	L	F/T	B	S					
SFRM SFRMS	6	20.0-290.0					für P<6: B<Px3 & S<Qx3 & S<T-2	3 4 5		5	300
PSFRM PSFRMS	8	20.0-390.0					für P<6: B<Px3 & S<Qx3 & S<T-2	3 4 5 6		7	8 400
SSFRM SSFRMS	10	20.0-490.0					für P<6: B<Px3 & S<Qx3 & S<T-2	4 5 6 8		8	500
(HFRM (nur mit markierten Größen))	12	25.0-590.0					für P<6: B<Px3 & S<Qx3 & S<T-2	5 6 8 10		10	600
PSFRGMG SFRGMG SSFRGMG SSFRGMG	15	25.0-690.0					für P<6: B<Px3 & S<Qx3 & S<T-2	6 8 10 12		11	700
(HFRM (nur mit markierten Größen))	16	25.0-690.0					für P<6: B<Px3 & S<Qx3 & S<T-2	5 6 8 10 12		13	700
PSFRGMG SFRGMG SSFRGMG SSFRGMG	17	30.0-790.0					für P<6: B<Px3 & S<Qx3 & S<T-2	5 6 8 10 12		14	800
(HFRM (nur mit markierten Größen))	18	30.0-790.0					für P<6: B<Px3 & S<Qx3 & S<T-2	5 6 8 10 12		14	
PSFRGMG SFRGMG SSFRGMG SSFRGMG	19	30.0-790.0					für P<6: B<Px3 & S<Qx3 & S<T-2	6 8 10 12 16		15	
(HFRM (nur mit markierten Größen))	20	30.0-790.0					für P<6: B<Px3 & S<Qx3 & S<T-2	8 10 12 16 20		22	
PSFRGMG SFRGMG SSFRGMG SSFRGMG	21	30.0-790.0					für P<6: B<Px3 & S<Qx3 & S<T-2	8 10 12 16 20 24		27	15 800
(HFRM (nur mit markierten Größen))	22	30.0-790.0					für P<6: B<Px3 & S<Qx3 & S<T-2	8 10 12 16 20 24 30		30	
PSFRGMG SFRGMG SSFRGMG SSFRGMG	23	30.0-790.0					für P<6: B<Px3 & S<Qx3 & S<T-2	12 16 20 24 30		36	
(HFRM (nur mit markierten Größen))	24	30.0-790.0					für P<6: B<Px3 & S<Qx3 & S<T-2	12 16 20 24 30		41	20
PSFRGMG SFRGMG SSFRGMG SSFRGMG	25	30.0-790.0					für P<6: B<Px3 & S<Qx3 & S<T-2	16 20 24 30		46	
(HFRM (nur mit markierten Größen))	26	30.0-790.0					für P<6: B<Px3 & S<Qx3 & S<T-2	16 20 24 30		51	
PSFRGMG SFRGMG SSFRGMG SSFRGMG	27	30.0-790.0					für P<6: B<Px3 & S<Qx3 & S<T-2	16 20 24 30		56	
(HFRM (nur mit markierten Größen))	28	30.0-790.0					für P<6: B<Px3 & S<Qx3 & S<T-2	16 20 24 30		61	
PSFRGMG SFRGMG SSFRGMG SSFRGMG	29	30.0-790.0					für P<6: B<Px3 & S<Qx3 & S<T-2	16 20 24 30		66	
(HFRM (nur mit markierten Größen))	30	30.0-790.0					für P<6: B<Px3 & S<Qx3 & S<T-2	16 20 24 30		71	
PSFRGMG SFRGMG SSFRGMG SSFRGMG	31	30.0-790.0					für P<6: B<Px3 & S<Qx3 & S<T-2	16 20 24 30		76	
(HFRM (nur mit markierten Größen))	32	30.0-790.0					für P<6: B<Px3 & S<Qx3 & S<T-2	16 20 24 30		81	
PSFRGMG SFRGMG SSFRGMG SSFRGMG	33	30.0-790.0					für P<6: B<Px3 & S<Qx3 & S<T-2	16 20 24 30		86	
(HFRM (nur mit markierten Größen))	34	30.0-790.0					für P<6: B<Px3 & S<Qx3 & S<T-2	16 20 24 30		91	
PSFRGMG SFRGMG SSFRGMG SSFRGMG	35	30.0-790.0					für P<6: B<Px3 & S<Qx3 & S<T-2	16 20 24 30		96	
(HFRM (nur mit markierten Größen))	36	30.0-790.0					für P<6: B<Px3 & S<Qx3 & S<T-2	16 20 24 30		101	
PSFRGMG SFRGMG SSFRGMG SSFRGMG	37	30.0-790.0					für P<6: B<Px3 & S<Qx3 & S<T-2	16 20 24 30		106	
(HFRM (nur mit markierten Größen))	38	30.0-790.0					für P<6: B<Px3 & S<Qx3 & S<T-2	16 20 24 30		111	
PSFRGMG SFRGMG SSFRGMG SSFRGMG	39	30.0-790.0					für P<6: B<Px3 & S<Qx3 & S<T-2	16 20 24 30		116	
(HFRM (nur mit markierten Größen))	40	30.0-790.0					für P<6: B<Px3 & S<Qx3 & S<T-2	16 20 24 30		121	
PSFRGMG SFRGMG SSFRGMG SSFRGMG	41	30.0-790.0					für P<6: B<Px3 & S<Qx3 & S<T-2	16 20 24 30		126	
(HFRM (nur mit markierten Größen))	42	30.0-790.0					für P<6: B<Px3 & S<Q				

Antriebstechnik

Stirnräder Modul 1

Stirnräder Modul 1,5

Stirnräder Modul 2

Stirnräder Modul 2,5

Stirnräder Modul 3

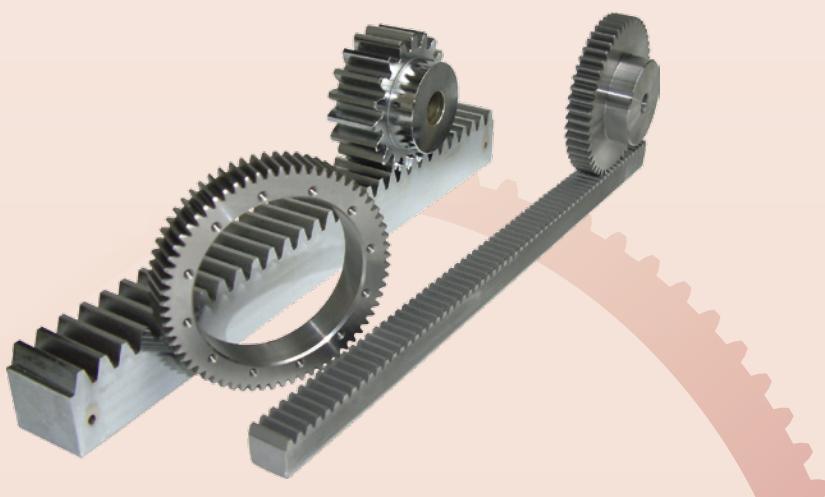
Stirnräder Modul 4

Stirnräder Modul 5

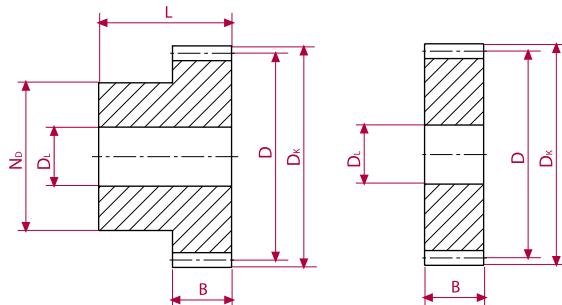
Stirnräder Modul 6

Stirnräder Modul 8

Sonderanfertigungen komplett nach Ihrer Zeichnung sowie
mechanische Bearbeitung von Standardteilen auf Anfrage.



G&G Antriebstechnik GmbH • Dieselstraße 102 • D-33334 Gütersloh
Fon (05241)40349-0 • Fax (05241)40349-22 • www.gg-antriebstechnik.de

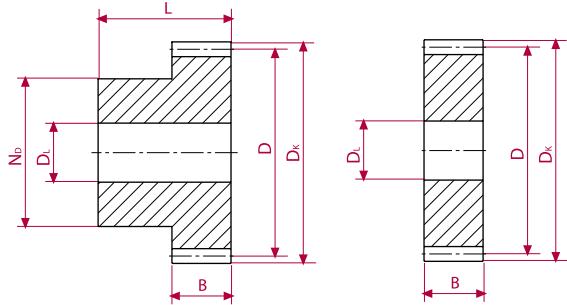


Stirnräder Modul 1

G&G Bestell-Nr.	Zz	D _k	D	N _D	D _L
416-010-0012	12	14.0	12.0	9	5
416-010-0013	13	15.0	13.0	10	5
416-010-0014	14	16.0	14.0	11	5
416-010-0015	15	17.0	15.0	12	6
416-010-0016	16	18.0	16.0	13	6
416-010-0017	17	19.0	17.0	14	8
416-010-0018	18	20.0	18.0	15	8
416-010-0019	19	21.0	19.0	15	8
416-010-0020	20	22.0	20.0	16	8
416-010-0021	21	23.0	21.0	16	8
416-010-0022	22	24.0	22.0	16	8
416-010-0023	23	25.0	23.0	18	8
416-010-0024	24	26.0	24.0	20	8
416-010-0025	25	27.0	25.0	20	8
416-010-0026	26	28.0	26.0	20	8
416-010-0027	27	29.0	27.0	20	8
416-010-0028	28	30.0	28.0	20	8
416-010-0029	29	31.0	29.0	20	8
416-010-0030	30	32.0	30.0	20	8
416-010-0031	31	33.0	31.0	25	10

geradverzahnt
20° Eingriffswinkel
Gesamtbreite L = 25 mm
Zahnbreite B = 15 mm

bis Zähnezahl 70 mit einseitiger Nabe
ab Zähnezahl 72 als Stirnrad scheibe
Material C45



Stirnräder Modul 1

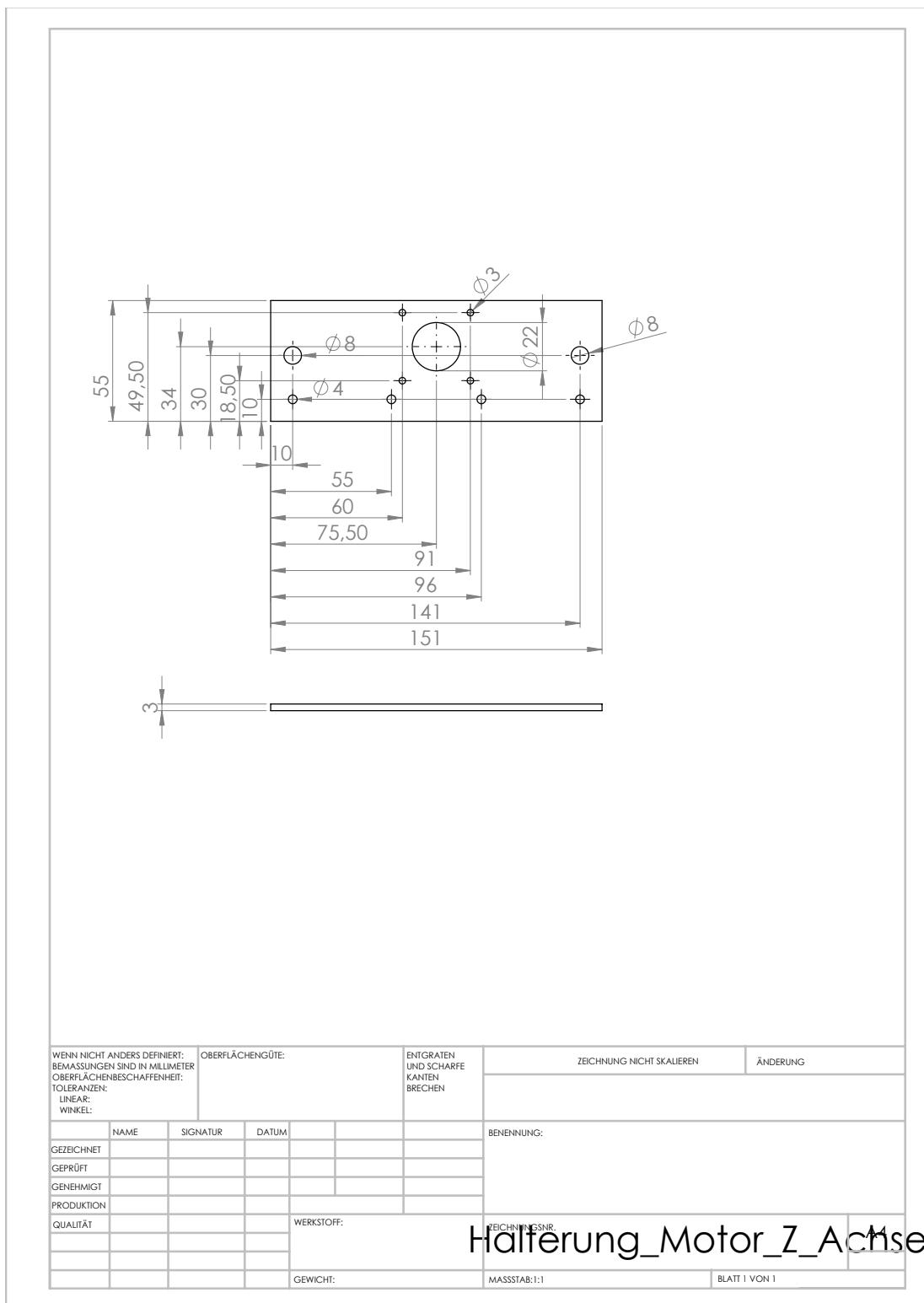
G&G Bestell-Nr.	Zz	D _b	D	N _d	D _L
416-010-0051	51	53.0	51.0	40	12
416-010-0052	52	54.0	52.0	40	12
416-010-0053	53	55.0	53.0	40	12
416-010-0054	54	56.0	54.0	40	12
416-010-0055	55	57.0	55.0	40	12
416-010-0056	56	58.0	56.0	40	12
416-010-0057	57	59.0	57.0	40	12
416-010-0058	58	60.0	58.0	40	12
416-010-0059	59	61.0	59.0	40	12
416-010-0060	60	62.0	60.0	40	12
416-010-0061	61	63.0	61.0	50	12
416-010-0062	62	64.0	62.0	50	12
416-010-0063	63	65.0	63.0	50	12
416-010-0064	64	66.0	64.0	50	12
416-010-0065	65	67.0	65.0	50	12
416-010-0066	66	68.0	66.0	50	12
416-010-0067	67	69.0	67.0	50	12
416-010-0068	68	70.0	68.0	50	12
416-010-0069	69	71.0	69.0	50	12
416-010-0070	70	72.0	70.0	50	12

geradverzahnt
20° Eingriffswinkel
Gesamtbreite L = 25 mm
Zahnbreite B = 15 mm

bis Zähnezahl 70 mit einseitiger Nabe
ab Zähnezahl 72 als Stirnradscheibe
Material C45

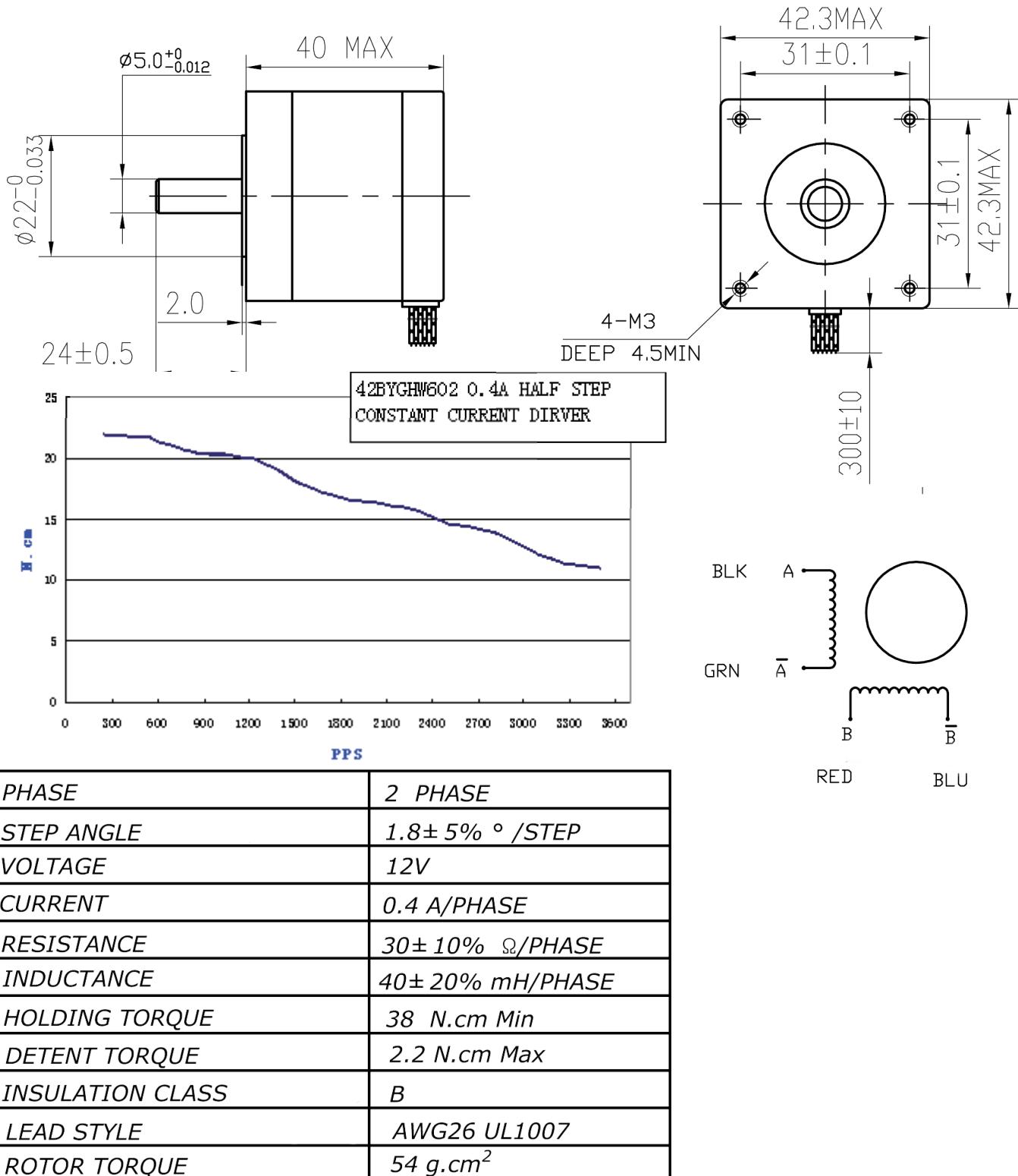
A.2. Z-Achse

Verschiedene Maße und wichtige Daten der Z-Achse können aus den Technischen Zeichnungen und aus den Datenblättern der folgenden Seiten entnommen werden.



Schrittmotor PSM42BYGHW603

Best.Nr. 310 453



Diese Bedienungsanleitung ist eine Publikation von Pollin Electronic GmbH, Max-Pollin-Straße 1, 85104 Pförring.
Alle Rechte einschließlich Übersetzung vorbehalten. Reproduktion jeder Art, z.B. Fotokopie, Mikroverfilmung oder die
Erfassung in elektronischen Datenverarbeitungsanlagen, bedürfen der schriftlichen Genehmigung des Herausgebers.
Nachdruck, auch auszugsweise, verboten. Diese Bedienungsanleitung entspricht dem technischen Stand bei Drucklegung.
Änderung in Technik und Ausstattung vorbehalten.

©Copyright 2011 by Pollin Electronic GmbH



Flanschlagergehäuse · quadratisch

-Festlager Kompaktausführung/Festlager Niedrigprofil/Festlager Befestigungsbohrungen mit geringem Abstand-



Flanschlagergehäuse · quadratisch

-Loslagerseite Kompaktausführung/Loslagerseite Flach/Loslagerseite Befestigungsbohrungen mit geringem Abstand-

CAD-Daten

• Durch seine Kompaktheit (geringere Höhe und Breite im Vergleich mit konventionellen Flanschlagergehäusen) platzsparend einsetzbar und insbesondere für kleine Geräte geeignet.

■ Flanschlagergehäuse - Festlagerseite - Kompakt



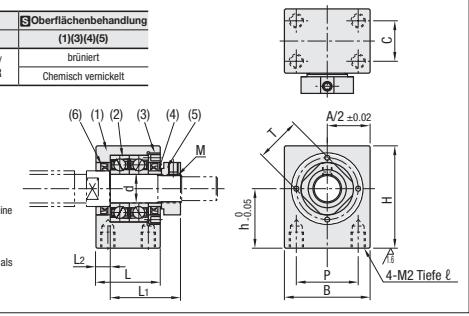
RoHS

Bauteildaten

Nr.	Bezeichnung	Stückz.
(1)	Lagergehäuse	1
(2)	Schrägkugellager Klasse 0°	1 Satz
(3)	Abdeckung	1
(4)	Hülse	1
(5)	Klemmutter (mit Druckstück)	1
(6)	Radialwellendichtring	2

Ausführung	Werkstoff	Oberflächenbehandlung
BSQ	1.1191/C45E	1.0038/S235JR brüniert
BSQM		Chemisch vernickelt

• Kompakteste Größe.



- Demontieren Sie (1), (2) und (3) nicht, da diese als eine Montageeinheit zusammengehören.
- d=8 wird mit nur einer Oldichtung (6) geliefert.

* Schrägkugellager Klasse 0° bei Nr. 8.
Bei Nr. 10S ist der Innen-Ø der Oldichtung geringer als bei der vorherigen Ausführung.

Teilenummer

Ausführung	Nr.	d	L	L1	L2	B	H	P	M2	ɛ	C	M (Fein)	T	Radialwellendichtring Verfügbarer Wellen-Ø	BSQ	BSQM
8	8	23	26	4	32	32	17	23	5	6	14	M8x1.0	14	11.5		
10S	10	27	29.5	6	36	43	25	26	6	10	17	M10x1.0	17	12		
10	10	27	29.5	6	36	43	25	26	6	10	17	M10x1.0	17	14		
12	12	27	29.5	6	36	43	25	26	6	10	17	M12x1.0	19	15		
15	15	29	38	5	46	52	30	34	6	12	19	M15x1.0	22	20		

■ Flanschlagergehäuse - Festlagerseite - Flach



RoHS

Bauteildaten

Nr.	Bezeichnung	Stückz.
(1)	Lagergehäuse	1
(2)	Schrägkugellager Klasse 5	1 Satz
(3)	Abdeckung	1
(4)	Zylinderschraube mit Innensechskant	4
(5)	Hülse	1
(6)	Klemmutter (mit Druckstück)	1
(7)	Radialwellendichtring	2

- Keine Oldichtung (7), wenn d=6.
- d=8 wird nur mit einer Oldichtung (7) geliefert.

E Demontieren Sie (1), (2) und (3), da diese als eine Montageeinheit zusammengehören.

Schrauben (4) für BSVM in Edelstahlauflösung.

Bei Nr. 8S und 10S ist der Innen-Ø der Oldichtung geringer als bei der vorherigen Ausführung.

Teilenummer

Ausführung	Nr.	d	L	L1	L2	L3	B	H	p	d1	d2	ɛ	M (Fein)	T	BSV	BSVM
6	6	20	-	22.5	3.5	42	22	11	30	5.5	9.5	12	M6x0.75	12		
8S	8	23	-	26	4	52	28	14	38	6.6	11	15	M8x1.0	14	10	
10S	10	24	6	29.5	6	70	36	18	52	9	14	19	M10x1.0	17	12	
12	12	24	6	29.5	6	70	36	18	52	9	14	19	M12x1.0	19	15	
15	15	25	6	38	5	80	41	21	60	11	17	21	M15x1.0	22	20	
20	20	42	10	52	10	95	56	28	75	11	17	29	M20x1.0	30	25	

■ Flanschlagergehäuse Festlagerseite Befestigungsbohrungen mit geringem Abstand



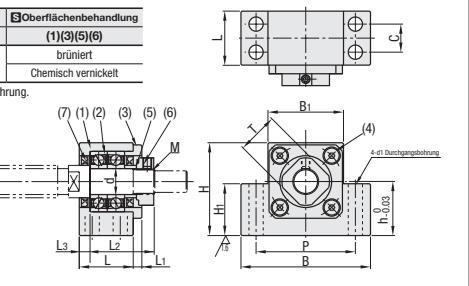
RoHS

Bauteildaten

Nr.	Bezeichnung	Stückz.
(1)	Lagergehäuse	1
(2)	Schrägkugellager Klasse 5	1 Satz
(3)	Abdeckung	1
(4)	Zylinderschraube mit Innensechskant	4
(5)	Hülse	1
(6)	Klemmutter (mit Druckstück)	1
(7)	Radialwellendichtring	2

Ausführung	Werkstoff	Oberflächenbehandlung
BSA	1.1191/C45E	1.0038/S235JR brüniert
BSAM		Chemisch vernickelt

• Auslegungsbreite für Kompatzausführung.



- Demontieren Sie (1), (2) und (3) nicht, da diese als eine Montageeinheit zusammengehören.
- Schrauben (4) für BSAM in Edelstahlauflösung.

* Bei Nr. 10S ist der Innen-Ø der Oldichtung geringer als bei der vorherigen Ausführung.

• (4) der Schrauberkopf (4) ragt um 2mm aus der Abdeckung heraus.

Teilenummer

Ausführung	Nr.	d	L	L1	L2	L3	B	H	B1	H1	P	C	d1	M (Fein)	T	BSA	BSAM
10S	10	25	4	29.5	5	60	39	22	34	21	46	13	6.6	M10x1.0	17	12	
10	12	25	4	29.5	5	60	43	25	35	24	46	13	6.6	M12x1.0	19	15	
12	15	27	6	38	6	70	48	20	47	54	15	15	6.6	M15x1.0	22	20	

Bestell-
beispiel

Teilenummer
BSM6
BSV10S
BSA12

Liefer-
zeit

6

Arbeits-
tage

8.7

Für Einzelheiten zur Präzision und zu den Lagern der Flanschlagergehäuse-Montageschnittstelle
siehe S.685

Kugelgewindetreib-Kompatibilitätsdiagramm S.685

Flanschlagergehäuse · quadratisch

-Loslagerseite Kompaktausführung/Loslagerseite Flach/Loslagerseite Befestigungsbohrungen mit geringem Abstand-

CAD-Daten

• Gleiche Maße wie bei den Festlager-Flanschlagergehäusen, aber in Höhe und Breite kompakter. Verwenden Sie die Festlager-Flanschlagergehäuse zusammen.

■ Flanschlagergehäuse Loslagerseite - Kompakt

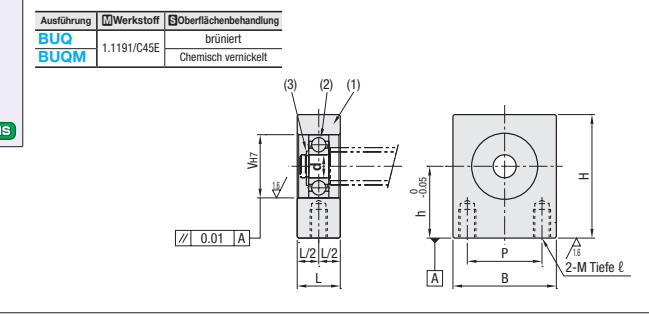


RoHS

Bauteildaten

Nr.	Bezeichnung	Stückz.
(1)	Gehäuse	1
(2)	Lager	1
(3)	Sicherungsring*	1

* Sicherungsring (3) von BUQM in Edelstahlauflösung.



Teilenummer

Ausführung	Nr.	d	L	B	H	p	V	M	ɛ	Lager Ausführung	Sicherungsring	BUQ	BUQM
8	6	15	32	32	17	23	17	5	6	B606ZZ	STWN-S6		
10	8	20	36	43	25	26	26	10	10	B608ZZ	STWN-S8		
12	10	20	40	52	30	34	32	12	12	B6000ZZ	STWN-S10		
15	15	26	46	56	28	75	47	11	17	B6024ZZ	STWN-S15		

■ Flanschlagergehäuse - Loslagerseite - Flach

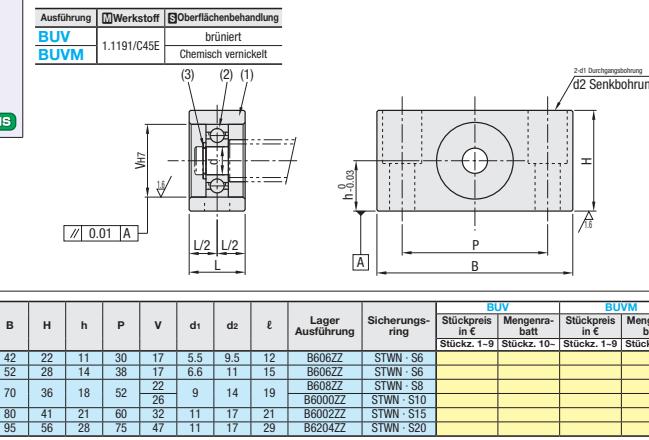


RoHS

Bauteildaten

Nr.	Bezeichnung	Stückz.
(1)	Gehäuse	1
(2)	Lager	1
(3)	Sicherungsring*	1

* Sicherungsring (3) von BUV in Edelstahlauflösung.



Teilenummer

Ausführung	Nr.	d	L	B	H	p	V	M	ɛ	Lager Ausführung	Sicherungsring	BUV	BUVM
6	6	12	42	22	11	30	17	5.5	12	B606ZZ	STWN-S6		
8	6	15	52	28	14	38	17	6.6	11	B606ZZ	STWN-S6		
10	8	20	70	36	18	52	22	9	14	B608ZZ	STWN-S8		
12	10	20	80	41	21	60	32	11	17	B6000ZZ	STWN-S10		
15	15	26	95	56	28	75	47	11	17	B6024ZZ	STWN-S15		

■ Flanschlagergehäuse Loslagerseite Befestigungsbohrungen mit geringem Abstand



RoHS

Bauteildaten

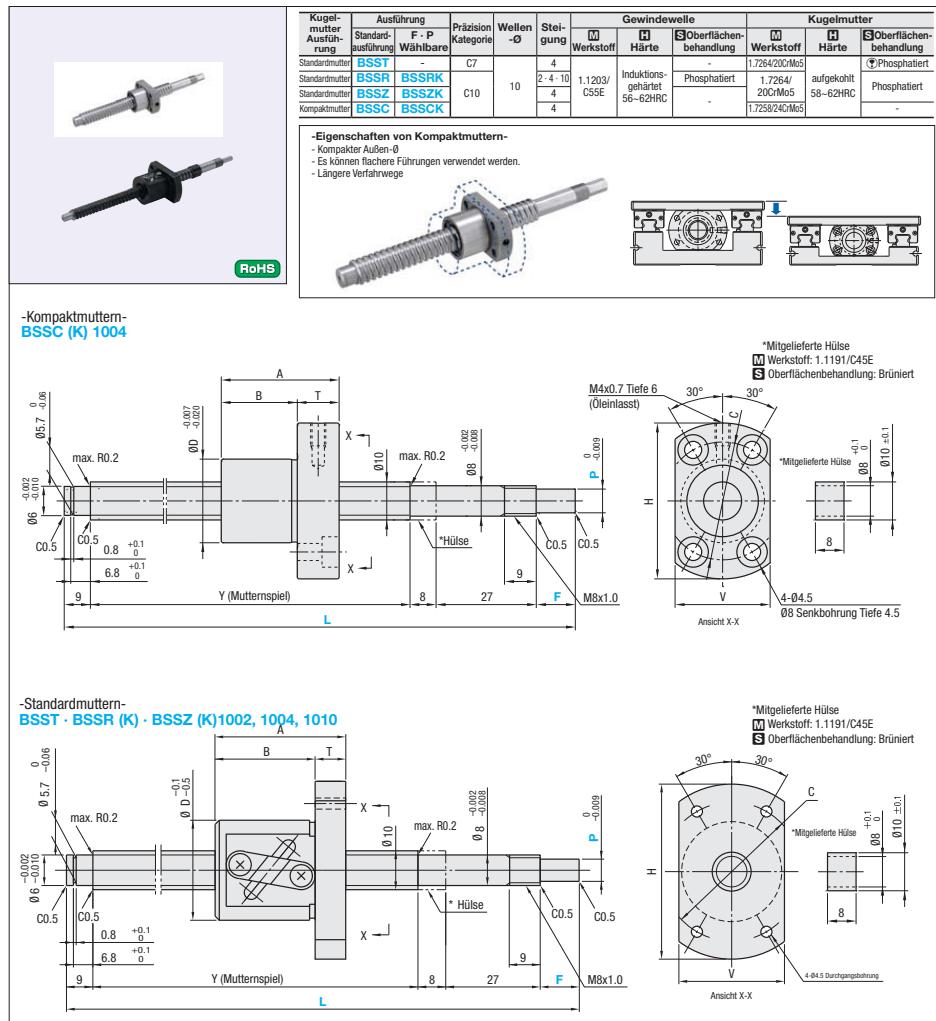
Nr.	Bezeichnung	Stückz.
(1)	Gehäuse	1
(2)	Lager	1
(3)	Sicherungsring*	1



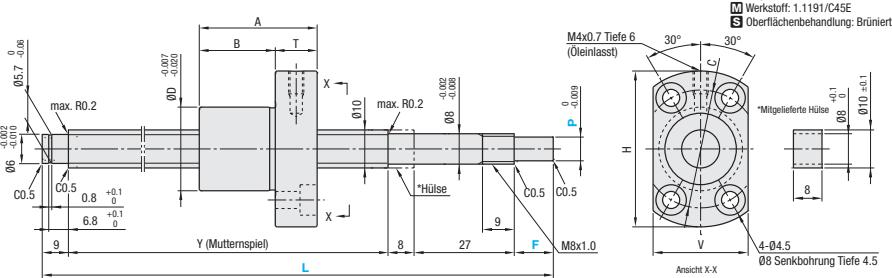
Kugelgewindetriebe gewalzt Wellen-Ø 10, Steigung 2 · 4 · 10- -Präzisionskategorie C7 · C10-

Price Reduction
Bis zu 26%

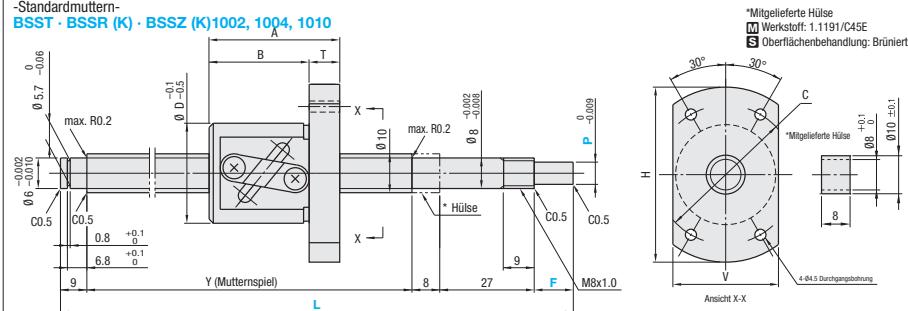
CAD-Daten



-Kompaktmuttern-
BSSC (K) 1004



-Standardmuttern-
BSS1 · BSSR (K) · BSSZ (K) 1002, 1004, 1010



Kugelmutter-Ausführung	Präzision Kategorie	Teilenummer		1mm-Schritte			Y	D	A	B	T	H	C	V	Kugel Ø	Kugel-Mitten-Ø	Kern-Ø	Anzahl Kreise	Tragzahl	Zulässiges Axialspiel	Drehrichtung			
		Ausführung	Gewinde-wellen-Ø	Steigung	L	*F	*P																	
Standardausführung Kugelmutter	C10	BSSR	02	150-585	10-18	5 · 6	L-(44+F)	10	6	L-54	23	28	22	6	43	33	27	1.5875	10.3	-8.4	1.7	3.0	max. 0.05	Rechts
		BSSRK						10	6	L-54	26	34	26	8	46	36	28	2.3812	10.6	-7.8	2.3	4.8	max. 0.03	
	C7	BSS1	10	150-600	10-18	5 · 6	L-(44+F)																	
		BSSZ						10	6	L-54	22	31	20	11	41	31	25	2	10.25	-8.4	3 Umdr. in einer Reihe	1.87	3.0	max. 0.05
	C10	BSSRK	04		10-18	5 · 6	L-(44+F)	10	6	L-54	28	34	26	8	47	36	30	2.3812	10.6	-7.8	1.5 Umdr. in einer Reihe	1.85	3.2	
		BSSZK						10	6	L-54														
Kompakt-Kugelmutter	BSSC																							
Standardausführung Kugelmutter	BSSRK																							
Standardausführung Kugelmutter	BSSR																							
Standardausführung Kugelmutter	BSSRK																							

*F und P sind nur für BSSRK - BSSZK und BSSCK konfigurierbar. P=Fx3

kgf=Nx0.101972



Preis

Kugelmutter-Ausführung	Präzision Kategorie	Teilenummer	EUR Stückpreis Stückz. 1-4		
			L150-200	L201-400	L401-600
Standardmutter	C10	BSSR1002	223,40	227,90	232,50
	C7	BSST1004	257,50	266,60	275,70
		BSSR1004	161,50	169,70	173,70
Kompaktmutter	C10	BSSZ1004	137,30	144,50	147,50
		BSSC1004	116,30	122,40	124,90
Standardmutter		BSSR1010	252,40	257,50	262,70

Bei noch größeren Bestellmengen bitte gesondert anfragen.
 ☐ Addieren Sie für den Preis von BS..._K Konfigurationen F - P)
 19,50 EUR zum Preis für die Standardausführung.
 z. B. BSSZK1004-300-F12-P6
 144,50 EUR+19,50 EUR=164,00 EUR



Teilenummer - L - F - P
BSSC1004 - 270
BSSZK1004 - 300 - F15 - P6
BSSZ1004L - 550 (Typ L mit Schmierstoff)
BSS - 550 (Typ G mit Schmierstoff)



- BSST - BSSR (K) - BSSZ (K)
8 Arbeitstage
Express A 8,00 EUR/Stk. **S.38**
 ☐ Expressgebühr von 21,60 EUR für 3 oder mehr identische Teile.

Hinweis

Die Montagefläche für den BSST-Kugelmutterflansch ist nicht oberflächenbehandelt.
 Mit Lithiumverseiftem Schmierstoff gefüllt (Alvania-Fett S2 von Showa Shell Sekiyu K.K.).

Hinweis: Lassen Sie die Muttern nicht überlaufen, um trennen Sie die Muttern nicht von den Gewindewellen.

Dies kann dazu führen, dass Kugellager herausfallen oder die Kugelrückführungsstelle beschädigt werden.

Es sind verschiedene Schmierstoffausführungen erhältlich. Zu Lieferzeit, Preisen und Spezifikationen siehe S.651 BSSC ist nicht verfügbar.

Zur Präzision der Kugelgewindetriebe siehe S.651 - 653

Einzelheiten zu Flanschlagergehäusen siehe S.685-S.694



Teilenummer - L - F - P - (FC · KC-...usw.)
BSSR1004 - 270 - SC7

Optionen	Opt.-Nr.	Spez.	Aufpreis
Keine Bearbeitung des Wellenendes (Loslagerseite)	NC	Ohne Bearbeitung des Wellenendes auf der Loslagerseite. Bestellnr.: NC	BSST - BSSR BSSZ -20,00 BSSC -7,00
Kugelmutter-Ausrichtung umgekehrt (Loslagerseite)	RLC	Umkehrung der Mutter-Ausrichtung. Bestellnr.: RLC	Ohne Aufpreis
Standard			
Geändert			
Keine Sicherungsringnut am Wellenende (Loslagerseite)	RNC	Ohne Sicherungsringnut am Wellenende auf der Loslagerseite. Bestellnr.: RNC Keine Kombination mit FC verfügbar.	Ohne Aufpreis
Andere Länge des Wellenendes (Loslagerseite)	FC	Änderung der Länge des Wellenendes auf der Loslagerseite. FC-1mm-Schrifte Bestellnr.: FC20 FC-Maf ist verkürzt.	BSST - BSSR BSSZ -7,00 BSSC -3,00

Optionen	Opt.-Nr.	Spez.	Aufpreis
Schlüsselflächen auf Festlagerseite	SZC	Mit Schlüsselflächen am Wellenende (Festlagerseite). Old Kugellager fallen heraus, wenn die Kugelmutter mit den Schlüsselflächen in Kontakt kommt.	BSST - BSSR BSSZ -7,00 BSSC -3,00
Nut am Wellenende (Festlagerseite)	KC	Mit einer Nut am Wellenende (Festlagerseite). P-S ist nicht verfügbar. KC-1mm-Schrifte Bestellnr.: KC5 KC-2-KC-Px3 KC-F-1	BSST - BSSR BSSZ -14,00 BSSC -6,00
Kugelmutter-Ausrichtung umgekehrt (Festlagerseite)	KLC	Mit einer Nut in einem kundendefinierten Bereich am Wellenende (Festlagerseite). Bestellnr.: KLC-KS2 KS-1mm-Schrifte Bestellnr.: KLC-KS2-Px3 KS-2-KC-Px3 KC-F-1	BSST - BSSR BSSZ -14,00 BSSC -6,00
Nut am Wellenende (Festlagerseite)	SC	Mit einer Spannfläche am Wellenende (Festlagerseite). SC-1mm-Schrifte Bestellnr.: SC7 SC-2-KC-Px3 KC-F-1	BSST - BSSR BSSZ -6,00 BSSC -3,00
Spannfläche am Wellenende (Festlagerseite)	SCW	Mit einer Spannfläche am Wellenende (Festlagerseite). JIS SWC-90° Winkel SGC: 120° Winkel 1mm-Schrifte Bestellnr.: JIS SWC JIS SWC-90° Winkel SGC: 120° Winkel 1mm-Schrifte Bestellnr.: JIS SWC-GC-Px3 SWC, SGC-F-1	BSST - BSSR BSSZ -je 7,00 BSSC -je 3,00

Kombination mit Flanschlagergehäusen

Teilenummer Kugelgewindetrieb	Empfohlene Flanschlagergehäuse				
	Ausführung	Gewinde-welle Außen-Ø	Steigung	Teilenummer Nr.	Form
BSSC				BSV 8S	Quadratisch, flach
	02			BUV 8	
BSST	10			BRW 8S	Rund
BSSR				BUR 6	
BSSZ				BSW 8S	Quadratische Ausführung
	04			BUN 8	
	10				

Spindelmuttern sind in mehreren Werkstoffvarianten und mit unterschiedlichen Oberflächenbehandlungen erhältlich. (S.695)

Empfohlene Artikel: Einfache Positionierung bei Flanschlagergehäusen mit Zylinderschaftbohrung,

da die Zylinderschaftbohrungen auf den Einheiten vorhanden sind.

Teilenummer Kugelgewindetrieb	Empfohlene Spindelmuttern				
	Ausführung	Gewinde-welle Außen-Ø	Steigung	Teilenummer Nr.	Seite
BSSC	04			BNFB 1004C	S.695
BSST	02			BNFM 1002R	S.695
BSSR	10			BNFR 1004R	S.695
BSSZ	04			BNFA 1010R	S.695

Spindelmuttern sind in mehreren Werkstoffvarianten und mit unterschiedlichen Oberflächenbehandlungen erhältlich. (S.695)

Empfohlene Artikel: Einfache Positionierung bei Flanschlagergehäusen mit Zylinderschaftbohrung,

da die Zylinderschaftbohrungen auf den Einheiten vorhanden sind.



Linearführungen für mittlere und hohe Traglast

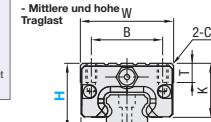
-Edelstahl- -Normales Spiel-

Merkmale: Linearführungen für mittlere und hohe Traglast, hervorragender Korrosionsschutz.



Ausführung		MX (Schmiereinheiten)	Maß L	Führungswagen Stückz.	Werkstoff Härte
Mittlere Traglast	Hohe Traglast				
Standard-Führungswagen	Standard-Führungswagen	MX	1 2 1 2 1 2	1 2 1 2 1 2	Edelstahl (Werkstoff: 1.4125/ X105CrMo7- 5HRC-)
Breites Gehäuse (Unterteil)					

Wärmebeständigkeit: -20-80°C

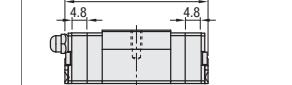


Schmiereinheiten MX Neu

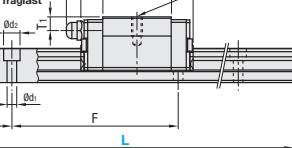
Bieten langfristigen wartungsfreien Betrieb.

Maß für Führungswagen mit MX (Schmiereinheiten)

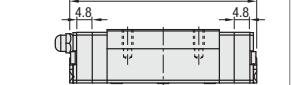
- Mittlere Traglast, L1



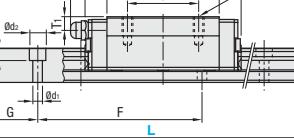
- Mittlere Traglast, L1



- Hohe Traglast, L1



- Hohe Traglast, L1



Vorsichtsmaßnahmen

Die Schienen können nicht zur Verlängerung verbunden werden.

Siehe [S.501](#)

Radiales Spiel und Genauigkeit werden nicht garantiert, wenn die Führungswagen und -schiene aus der Kombination des Originalsetzes heraus ausgetauscht werden.

Auf den Bezugstellen sind gerade Nuten eingebracht. Stellen Sie sicher, dass beim Zusammenstellen die Bezugslinien passen.

Die Schienen können nicht zur Verlängerung verbunden werden.

Laufparallelität ist der Wert, der nach Montage der Führungsschiene gemessen wird.

Andere

Für das Maß G bei der Ausführung mit konfigurierbarem Maß L, siehe [S.507](#)

- Gefüllt mit Ithiumverseiftem Schmierstoff (Alvania-Fett S2 von Showa Shell Sekiyu K.K.). Schmierlippe: gerade Ausführung für H24 und abgewinkelt für H28 und H33.

Teilenummern		Führungswagenmaße										Führungsschiennenmaße										
Ausführung	MX	H	W ±0,05	L1 Standard-ausführung	MX	B	C	SxL	K	T	CB	Schmiernippel	E	T1	W1 ±0,005	W2	Ca	Senkköpfchen	d1xd2xh	F	G	
Mittlere Traglast	Rohling ohne -MX: Mitgeliefert	1 Führungswagen(2 Führungswagen)	24	34 (52)	41	50,6 (41)	26	-	M4x7 (4,5)	25	20	7 (7)	0,85	M5xP0,8	6	5	12,5	15 (18,5)	0,5	3,5x6x4,5	60	20
			28	42 (59)	47	56,6 (49)	32	-	M5x8 (5,5)	27,6	22,5	7,5 (9)	1	M6xP0,75	13	6	15,5	20 (19,5)	0,6	6x9,5x8,5	60	20
		1 Führungswagen(2 Führungswagen)	33	48 (73)	59	68,6 (60)	35	-	M6x9 (7)	37	26,5 (10)	8 (1)	1	M6xP0,75	13	6,8	18	23 (25)	0,8	7x11x9	60	20
			24	34 (52)	57	66,6 (41)	26	26	M4x7 (4,5)	41	20	7 (7)	0,5	M5xP0,8	6	5	12,5	15 (18,5)	0,5	3,5x6x4,5	60	20
		1 Führungswagen(2 Führungswagen)	28	42 (59)	67	76,6 (49)	32	32	M5x8 (5,5)	47,6	22,5	7,5 (9)	1	M6xP0,75	13	6	15,5	20 (19,5)	0,6	6x9,5x8,5	60	20
			33	48 (73)	83	92,6 (60)	35	35	M6x9 (7)	61	26,5 (10)	8 (1)	1	M6xP0,75	13	6,8	18	23 (25)	0,8	7x11x9	60	20

Maße in () gelten für Ausführung mit breitem Führungswagen.

kfg-Nx0.101972

Tragzahl

Zulässiges statisches Moment

Gewicht

Führungswagenmaße

Führungsschiennenmaße

Wahlbar L

Führungswagen	H	C (Dynamisch) (kN)	C (Statisch) (kN)	Mz/Ma N * m	Mc N * m	Führungswagen kg	Führungsschienen kg
Mittlere Traglast	24	5,0	8,23	33	57	0,15	0,20
	28	7,2	12,1	58	135	0,20	0,25
	33	11,7	19,6	109	225	0,30	0,40
Hohe Traglast	24	8,6	14,2	69	98	0,20	0,25
	28	12,5	21,3	155	232	0,30	0,35
	33	20,2	34,5	275	393	0,45	0,60

Tragzahl

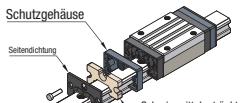
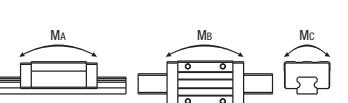
Zulässiges statisches Moment

Gewicht

Führungswagenmaße

Führungsschiennenmaße

Wahlbar L



Vorteile der Schmiereinheit MX
Langfristiger wartungsfreier Betrieb
Geringere Instandhaltungskosten
Besonders zu empfehlen an unzugänglichen
Schmiestellen
Weitere Informationen finden Sie auf [S.501](#)

Artikel für Linearführung	
Positionierungs-hilfen	Führungsschienen Distanz-blöcke
Führungswagen Abfallschutz	Befestigungs-teile Wagen/ Schienen

S.581- S.588

Price Reduction

10%



CAD-Daten



Bestellbeispiel
Teilenummer - L

SSXR28
- 575
SSXR-MX28
- 575 (Mit Schmiereinheiten)
RSXRL28
- 575 (LTBC)
SSXRL28L
- 575 (Typ L mit Schmierei)
SSXRL28G
- 575 (Typ G mit Schmierei)

Alternativ Schmiereinheitsführungen verfügbar (außer für Führungswagen mit Schmiereinheiten).

Für Lieferzeit, Preis und Leistung, siehe [S.507](#)



8 Arbeits-tage
Express A 8,00 EUR/ Stück

Teile mit LTBC-Beschichtung (Verfügbar nur für Standard-Ausführung.)

13 Arbeits-tage

Teile mit LTBC-Beschichtung (Verfügbar nur für Standard-Ausführung.)

Informations zu Merkmalen und Präzisionskategorien finden Sie auf [S.508](#)

Für Artikel mit LTBC-Beschichtung addieren Sie den Preis für die Oberflächenbehandlung auf S.508 zu dem Preis der Standard-Ausführung hinzu.



Preis Mengenrabatt (Abgerundet auf einen Cent) S.87
Stückz. 1-4 5% 5-10 10% 11-18%

Stückz.
Rabatt

€ euros
Teilenummer - L - (RLC - LLC)
SSXR33 - 520 - RLC

Optionen

8 Arbeits-tage
Express A 8,00 EUR/ Stück

Optionen Opt.-Nr. Spez. Aufpreis

Beschmitt der Führungsschienenenden
Bestellnr. L L C R L C

Zuschmitt links (LLC) RLC
H L N L
Zuschmitt rechts (RLC) RLC

* Nur bei freie wählbarer Ausführung.
** Gestaltung wird durch Bezeichnen L-L oder R-R.
*** Gilt nicht für Ausführung für hohe Traglast mit H-Wert 24 und L-Wert 100.

H	Wählbar L	€ Stückpreis							
		Mittlere Traglast				Hohe Traglast			
1 Führungswagen	2 Führungswagen	1 Führungswagen	2 Führungswagen	1 Führungswagen	2 Führungswagen	1 Führungswagen	2 Führungswagen	1 Führungswagen	2 Führungswagen
101-159									
161(192)-219									
221-279									
281-339									
341-399									
24	100								
160									
220									
280									
340									
24	340								
400									
460									
520									
580									
640									
700									
160									
220									
280									
340									
400									
460									
520									
580									
640									
641-699									
161-219									
221-279									
281-339									
341-399									
24	401-459								
461-519									
521-579									
581-639									
641-699									
161-219									
221(262)-279									
281-339									
341-399									
24	401-459								
461-519									
521-579									
581-639									
641-699									

Konfigurierbare Maße L in () sind nicht für die Ausführung für hohe Traglast mit 2 Führungswagen verfügbar.

■ MX (Schmiereinheit) Stückpreis in €

H	1 Führungswagen	2 Führungswagen
24	137,70	17,10 = 154,80
28		
33		

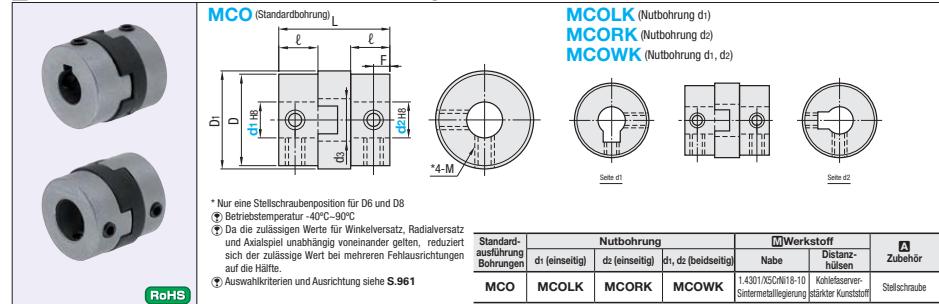
Preis für Führungsschienen mit MX (Schmiereinheit) = Stückpreis für Linearführung in € + Stückpreis für MX € (Bsp.) für SSXR-MX28-640, 137,70 + 17,10 = 154,80 EUR

Kupplungen

-Oldham-Kupplungen, Stellschraube/Distanzscheiben-

CAD-Daten

Eigenschaften: Nabe und Distanzscheibe trennbar, einfache Montage. (1)Ausführung mit Nabenklemmung siehe S.977



Teilenummer		d1, d2 wählen (d1 muss ≤ d2 sein)						Stellschraube		€ Stückpreis										
Ausführung	Nr.	(1)Die Ausführung mit Nutbohrung ist mit Ø6 oder höher frei wählbar.						D	d1	d2	L	ℓ	F	M	Anzugs- moment (Nm)	MCO	MCOLK	MCORK	MCOWK	
MCO	6 1 1.5 2	6	6.2	2.4	8.4	3	1.5	M1.6	0.15											
MCOLK	8 1 2 3	8	8.2	3.4	9.6	3.5	1.7	M2	0.3											
MCORK	10 2 2.3 4	10	10.2	4.4	10.2	3.7	1.8	M3	0.7											
MCOWK	12 3 4 5	12	12.5	4.0	14.2	5.2	2.5	M4	1.7											
MCO	15 4 5 6 6.35 7 8	14.5	15	5.0	16	5.4	2.6													
MCOLK	17 5 6 6.35 7 8	16.8	17.5	7.2	19.8	6.7	3.2													
MCORK	20 6 6.35 7 8 9.53 10 11 12	20	21	8.2	21.4	7	3.4													
MCOWK	26 6 6.35 7 8 9.53 10 11 12 14	26	27	12.0	25.6	9	4													
MCO	30 8 10 12 14	30	31	13.0	33	12	6													
MCOLK	34 10 11 12 14 15 16	34	35	13.0	34	13	5.5													
MCORK	38 10 12 14 15 16 18 20	38	41	16.0	40	15	7													
MCO	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50																			
MCOLK																				
MCORK																				
MCOWK																				
MCO	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50																			
MCOLK																				
MCORK																				
MCOWK																				

Nutemaß

	Wellenbohrungs-Ø d ₁ , d ₂	B	t	Nutmaße bxh
Referenzdurchm.	6-7,9	2	1.0	2x2
Referenzdurchm.	8-10	3	1.4	3x3
Referenzdurchm.	10,1-12	4	1.8	4x4
Referenzdurchm.	12,1-17	5	2,3	5x5
Referenzdurchm.	17,1-20	6	2,8	6x6

(1) Das zulässige Drehmoment ändert sich mit der Betriebstemperatur. S.961

Bestellbeispiel: Teilenummer - Wellenbohrungs-Ød₁ - Wellenbohrungs-Ød₂



Teilenummer - Wellenbohrungs-Ø d(LDC) - Wellenbohrungs-Ø d(RDC) - (KLH - KRL)

MCOGK00 - 8 - 10 - KRH4

8 Arbeits-tage

©Kein Expressversand.

Lieferzeit: 8 Arbeits-tage

Express A 2,00 EUR/Stk. S. 88

©Expressgebühr von 5,40 EUR für 3 oder mehr identische Teile.

-Nutbohrung:

8 Arbeits-tage ©Kein Expressversand.

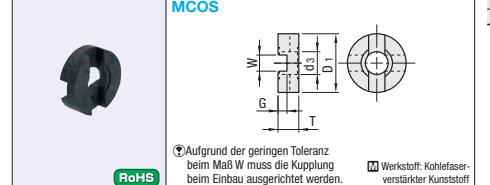
Mengenrabatt: (1)Abgerundet auf einen Cent.) S. 87

5-9 1-9 10-14 15-19 20-29

Rabatt € Stückpreis 5% 10% 18%

©Bei noch größeren Bestellmengen bitte gesondert anfragen.

Distanzscheiben (für: MCO_ ... MCOC_) S.977



Bestellbeispiel: Teilenummer MCOS15



Teilenummer 8 Arbeits-tage Express A 2,00 EUR/Stk. S. 88

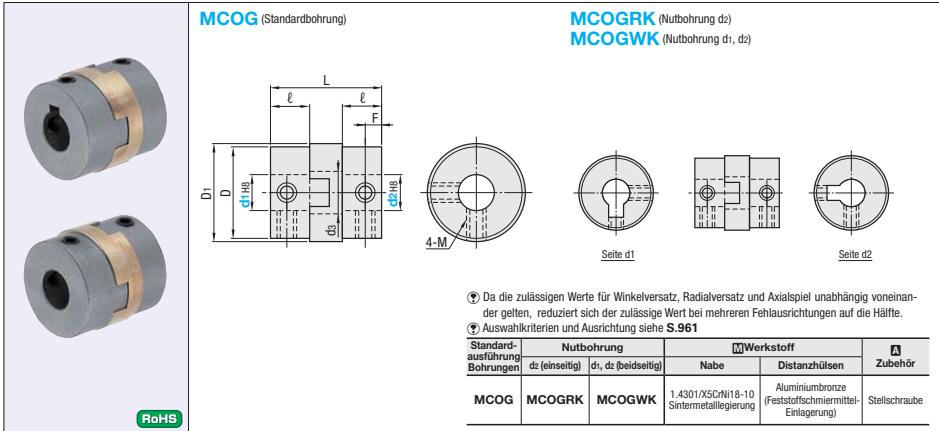
©Expressgebühr von 5,40 EUR für 3 oder mehr identische Teile.

Kupplungen

-Torsionssteife Oldham-Kupplungen, Stellschraube-

CAD-Daten

Eigenschaften: Die Distanzscheibe besteht aus Aluminiumbronze. Das zulässige Drehmoment ist doppelt so hoch wie bei der Kunststoffausführung. (1)Ausführung mit Nabenklemmung siehe S.978



Teilenummer		d1, d2 wählen (d1 muss ≤ d2 sein)						Stellschraube		€ Stückpreis									
Ausführung	Nr.	(1)Die Ausführung mit Nutbohrung ist mit Ø6 oder höher frei wählbar.						D	d1	d2	L	ℓ	F	M	Anzugs- moment (Nm)	MCOG	MCGRK	MCOGWK	
MCOG	15 4 5 6 6.35 7 8	14.5	15	7.2	16	5.4	2.6	M3	0.7										
MCGRK	17 5 6 6.35 7 8	16.8	17.5	8.2	19.8	6.7	3.2	M4	1.7										
MCOGWK	20 6 6.35 7 8 9.53 10 11 12	20	21	9	21.4	7	3.4	M4	1.7										
MCGRK	26 6 6.35 7 8 9.53 10 11 12 14	26	27	12	25.6	9	4	M5	4.0										
MCOGWK	30 8 10 11 12 14 15 16	30	31	14	33	12	6	M5	4.0										
MCGRK	34 10 11 12 14 15 16 18 20	34	35	14	34	13	5.5	M5	4.0										
MCOGWK	38 10 12 14 15 16 18 20	38	41	17	39.5	15	7	M5	4.0										
MCOG	15 3	15	3	800	8000	4x10 ⁻⁸	±0.1	15											
MCGRK	17 5	17	5	1000	7000	1x10 ⁻⁷	±0.1	25											
MCOGWK	20 7	20	7	2200	6000	2x10 ⁻⁶	±0.1	37											
MCGRK	26 10	26	10	4000	5000	6x10 ⁻⁶	±0.2	79											
MCOGWK	30 30	30	5500	5000	2.5x10 ⁻⁵	±0.3	120												
MCGRK	34 32	34	8000	4000	4x10 ⁻⁵	±0.2	180												
MCOGWK	38 50	38	11000	4000	1x10 ⁻⁴	±0.3	256												

(1) Hervorragend geeignet für hohe Drehmomente und hohe Drehzahlen.

(1) Bei Radialversatz über 0.1 ist der Distanzscheibenverschleiß proportional zu: Lastmoment, Radialversatz und Drehzahl.

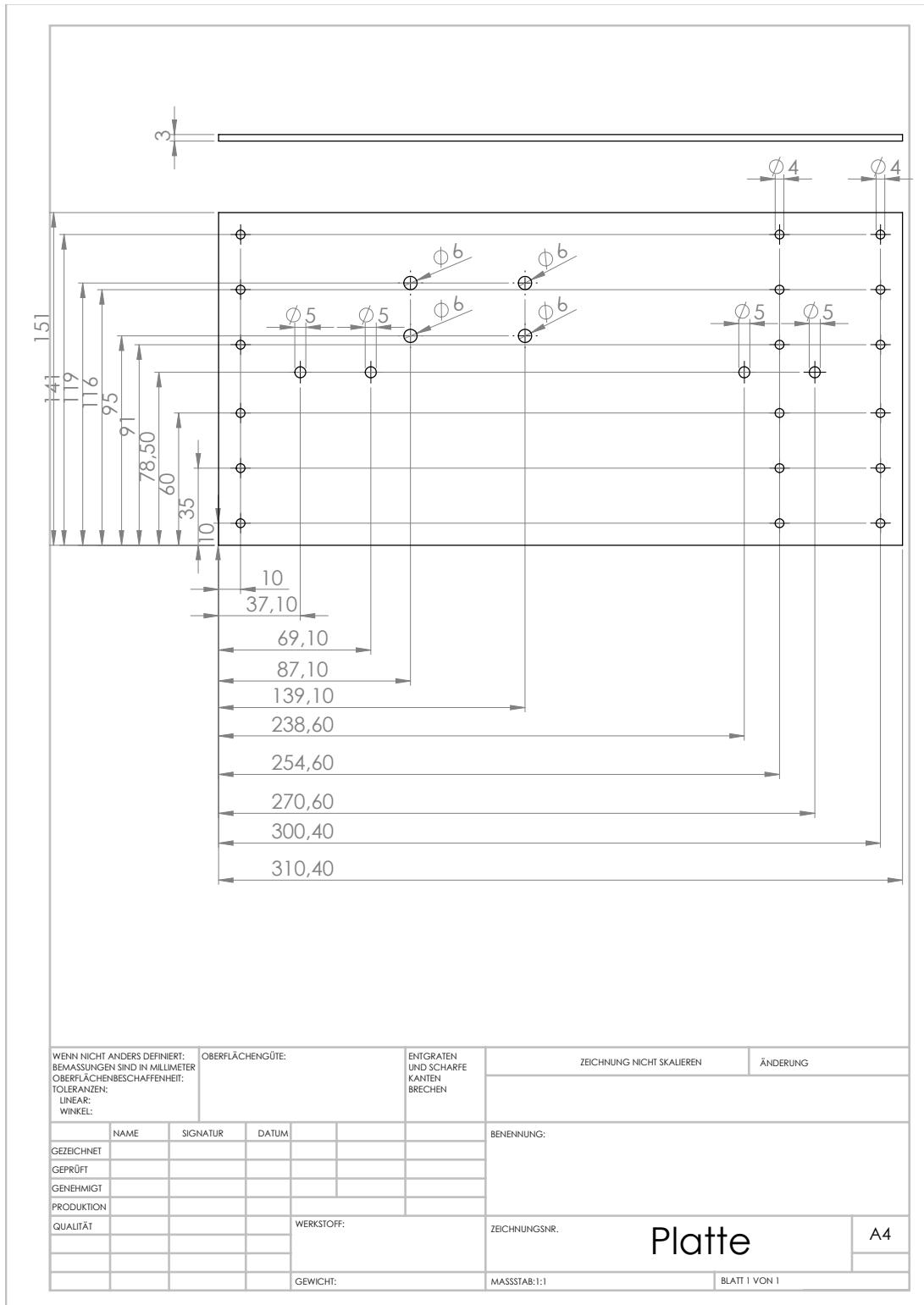
Teilenummer		Wellenbohrungs-Ø						Nutbreite		Änderung der Nutbreite (b) gemäß folgender Tabelle.							
Ausführung	Nr.	Wellenbohrungs-Ø d	Referenzdurchm.	Toleranz	Wellenbohrungs-Ø d ₁ , d ₂	Referenzdurchm.	Toleranz	Wellenbohrungs-Ø d	Referenzdurchm.	Wellenbohrungs-Ø d ₁ , d ₂	Referenzdurchm.	Toleranz	Wellenbohrungs-Ø d ₁ , d ₂	Referenzdurchm.	Toleranz		
MCGRK	15 1	15	4-8	1	14.5	7	1.8	15	1	14.5	7	1.8	15	1	14.5	7	1.8
MCGRK	17 2	17	5-8	2	16.8	8	2.2	17	2	16.8	8	2.2	17	2	16.8	8	2.2
MCGRK	20 3	20	6-12	3	19.8	9	3.2	20	3	19.8	9	3.2	20	3	19.8	9	3.2
MCGRK	26 4	26	6-15	4	24.8	10	4.2	26	4	24.8	10	4.2	26	4	24.8	10	4.2
MCGRK	30 5	30	8-14	5	28.8	12	5.2	30	5	28.8	12	5.2	30	5	28.8	12	5.2
MCGRK	34 6	34	10-16	6	32.8	14	6.2	34	6	32.8	14	6.2	34	6	32.8	14	6.2
MCGRK	38 7	38	10-20	7	36.8	16	7.2	38	7	36.8	16	7.2	38	7	36.8	16	7.2

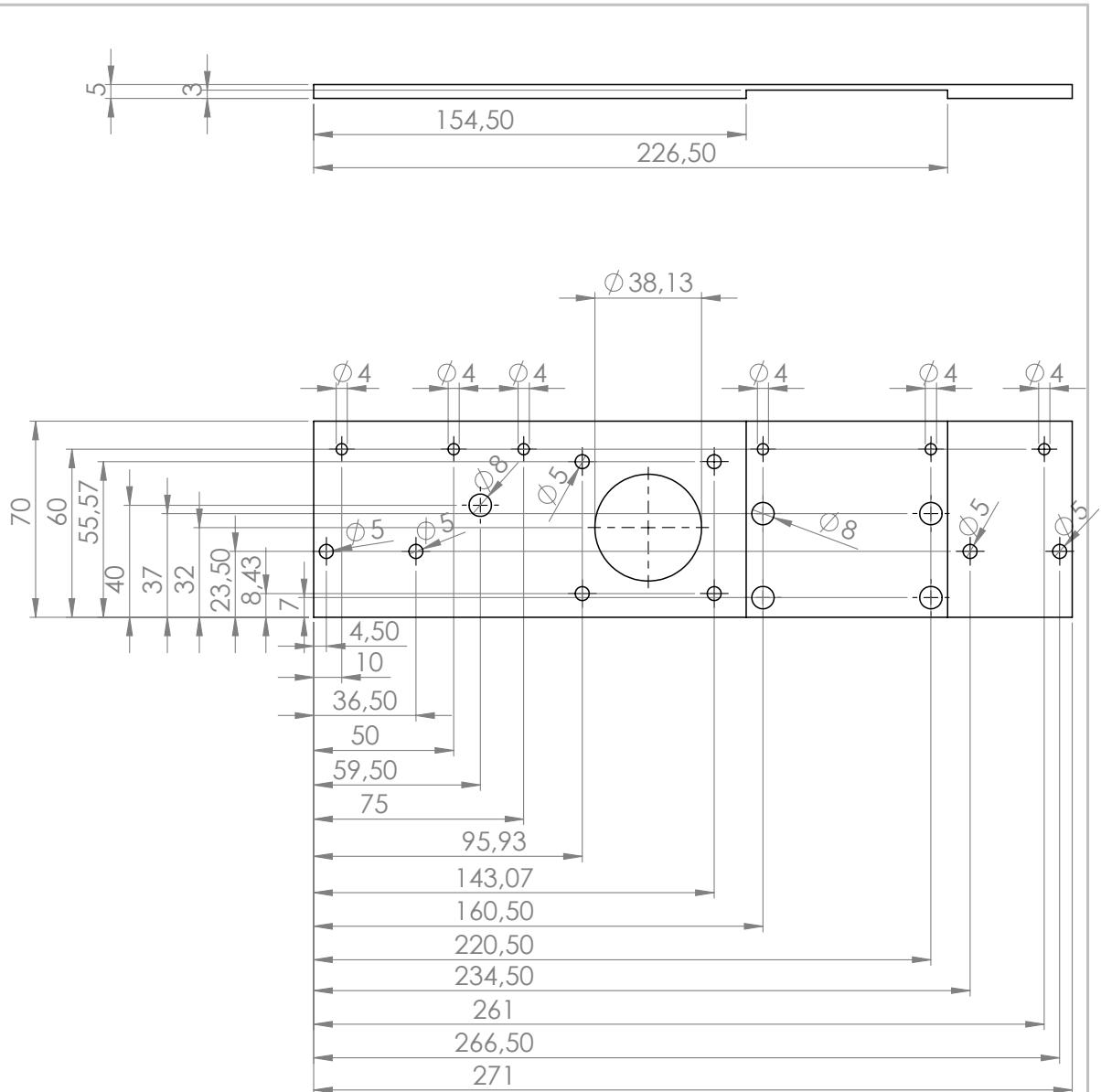
©Keine Kombination mit geänderten Wellenbohrungs-Ø (D, RDC) möglich.
 Nur für Nutbohrungen verfügbar.

Teilenummer		Wellenbohrungs-Ø						Passende Kupplungen		€ Stückpreis		
Ausführung	Nr.	D	T	ds	W	G	Passende Kupplungen	Aufpreis	Passende Kupplungen	Aufpreis	Passende Kupplungen	Aufpreis

A.3. X-Achse

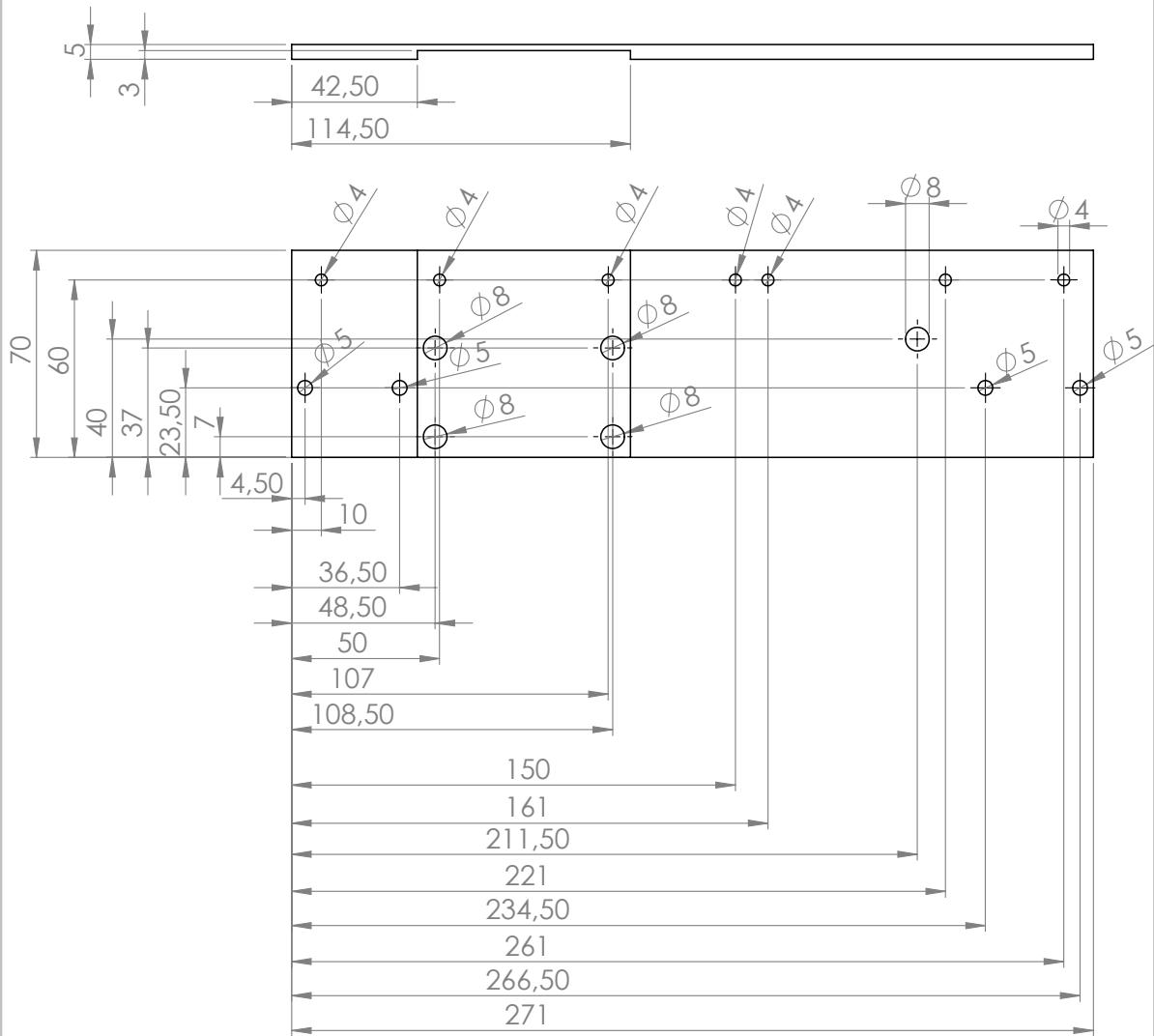
Verschiedene Maße und wichtige Daten der X-Achse können aus den Technischen Zeichnungen und aus den Datenblättern der folgenden Seiten entnommen werden.





WENN NICHT ANDERS DEFINIERT: BEMASSTUNGEN SIND IN MILLIMETER OBERFLÄCHENBESCHAFFENHEIT: TOLERANZEN: LINEAR: WINKEL:				OBERFLÄCHENGÜTE:	ENTGRATEN UND SCHARFE KANTEN BRECHEN	ZEICHNUNG NICHT SKALIEREN	ÄNDERUNG
	NAME	SIGNUR	DATUM		BENENNUNG:		
GEZEICHNET							
GEPRÜFT							
GENEHMIGT							
PRODUKTION							
QUALITÄT				WERKSTOFF:	ZEICHNUNGSNR.		A4
						BLATT 1 VON 1	
				GEWICHT:	MASSSTAB:1:1		

Befestigung_x_Achse_recht



WENN NICHT ANDERS DEFINIERT: BEMASSTUNGEN SIND IN MILLIMETER OBERFLÄCHENBESCHAFFENHEIT: TOLERANZEN: LINEAR: WINKEL:				OBERFLÄCHENGÜTE:	ENTGRATEN UND SCHARFE KANTEN BRECHEN	ZEICHNUNG NICHT SKALIEREN	ÄNDERUNG
GEZEICHNET	NAME	SIGNATUR	DATUM			BENENNUNG:	
GEPRÜFT							
GENEHMIGT							
PRODUKTION							
QUALITÄT				WERKSTOFF:		ZEICHNUNGSNR.	
						Befestigung_x_Achse	A4
				GEWICHT:		MASSSTAB:1:1	
						BLATT 1 VON 1	



Flanschlagergehäuse · quadratisch

-Festlager Kompaktausführung/Festlager Niedrigprofil/Festlager Befestigungsbohrungen mit geringem Abstand-



Flanschlagergehäuse · quadratisch

-Loslagerseite Kompaktausführung/Loslagerseite Flach/Loslagerseite Befestigungsbohrungen mit geringem Abstand-

CAD-Daten

• Durch seine Kompaktheit (geringere Höhe und Breite im Vergleich mit konventionellen Flanschlagergehäusen) platzsparend einsetzbar und insbesondere für kleine Geräte geeignet.

■ Flanschlagergehäuse - Festlagerseite - Kompakt



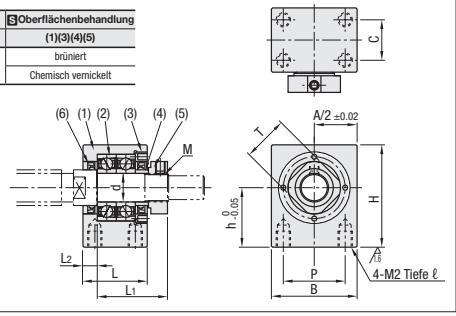
RoHS

Bauteildaten

Nr.	Bezeichnung	Stückz.
(1)	Lagergehäuse	1
(2)	Schrägkugellager Klasse 0	1 Satz
(3)	Abdeckung	1
(4)	Hülse	1
(5)	Klemmutter (mit Druckstück)	1
(6)	Radialwellendichtring	2

Ausführung	Werkstoff	Oberflächenbehandlung
BSQ	1.1191/C45E	1.0038/S235JR brüniert
BSQM		Chemisch vernickelt

• Kompakteste Größe.



• Demontieren Sie (1), (2) und (3) nicht, da diese als eine Montageeinheit zusammengehören.
• d=8 wird mit nur einer Olditung (6) geliefert.

* Schrägkugellager Klasse 0 bei Nr. 8.
Bei Nr. 10S ist der Innen-Ø der Olditung geringer als bei der vorherigen Ausführung.

Teilenummer	d	L	L1	L2	B	H	P	M2	ɛ	C	M (Fein)	T	Radialwellendichtring Verfügbarer Wellen-Ø	BSQ	BSQM
8	8	23	26	4	32	32	17	23	5	6	14	M8x1.0	14	11.5	
10S	10	27	29.5	6	36	43	25	26	6	10	17	M10x1.0	17	12	
10	10	27	29.5	6	36	43	25	26	6	10	17	M10x1.0	17	14	
12	12	27	29.5	6	36	43	25	26	6	10	17	M12x1.0	19	15	
15	15	29	38	5	46	52	30	34	6	12	19	M15x1.0	22	20	

■ Flanschlagergehäuse - Festlagerseite - Flach



RoHS

Bauteildaten

Nr.	Bezeichnung	Stückz.
(1)	Lagergehäuse	1
(2)	Schrägkugellager Klasse 5	1 Satz
(3)	Abdeckung	1
(4)	Zylinderschraube mit Innensechskant	4
(5)	Hülse	1
(6)	Klemmutter (mit Druckstück)	1
(7)	Radialwellendichtring	2

• Keine Olditung (7), wenn d=6.
• d=8 wird nur mit einer Olditung (7) geliefert.

E Demontieren Sie (1), (2) und (3), da diese als eine Montageeinheit zusammengehören.
Schrauben (4) für BSVM in Edelstahlauflaufung.

• Bei Nr. 8S und 10S ist der Innen-Ø der Olditung geringer als bei der vorherigen Ausführung.

Teilenummer	Nr.	d	L	L1	L2	L3	B	H	p	d1	d2	ɛ	M (Fein)	T	BSV	BSVM
6	6	20	-	-	22.5	3.5	42	22	11	30	5.5	9.5	12	M6x0.75	12	-
8S	8	23	-	-	26	4	52	28	14	38	6.6	11	15	M8x1.0	14	10
10S	10	24	6	29.5	6	70	36	18	52	9	14	19	17	M10x1.0	17	12
12	12	24	6	29.5	6	70	36	18	52	9	14	19	19	M12x1.0	19	15
15	15	25	6	38	5	80	41	21	60	11	17	21	21	M15x1.0	22	20
20	20	42	10	52	10	95	56	28	75	11	17	29	29	M20x1.0	30	25

■ Flanschlagergehäuse Festlagerseite Befestigungsbohrungen mit geringem Abstand



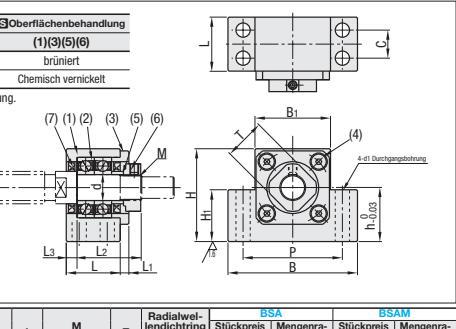
RoHS

Bauteildaten

Nr.	Bezeichnung	Stückz.
(1)	Lagergehäuse	1
(2)	Schrägkugellager Klasse 5	1 Satz
(3)	Abdeckung	1
(4)	Zylinderschraube mit Innensechskant	4
(5)	Hülse	1
(6)	Klemmutter (mit Druckstück)	1
(7)	Radialwellendichtring	2

Ausführung	Werkstoff	Oberflächenbehandlung
BSA	1.1191/C45E	1.0038/S235JR brüniert
BSAM		Chemisch vernickelt

• Auslegungsbreite für Kompaktausführung.



• Demontieren Sie (1), (2) und (3), da diese als eine Montageeinheit zusammengehören.
• Schrauben (4) für BSA in Edelstahlauflaufung.

• Bei Nr. 10S ist der Innen-Ø der Olditung geringer als bei der vorherigen Ausführung.

• (4) der Schraubenkopf (4) ragt um 2mm aus der Abdeckung heraus.

Teilenummer	Nr.	d	L	L1	L2	L3	B	H	B1	H1	P	C	d1	M (Fein)	T	BSA	BSAM
10S	10	25	4	29.5	5	60	39	22	34	21	46	13	6.6	M10x1.0	17	12	
10	12	25	4	29.5	5	60	43	25	35	24	46	13	6.6	M12x1.0	19	15	
12	15	27	6	38	6	70	48	20	47	54	15	15	6.6	M15x1.0	22	20	

Bestell-
beispiel

Teilenummer
BSM6
BSV10S
BSA12

Liefer-
zeit

6

Arbeits-
tage

8.7

Für Einzelheiten zur Präzision und zu den Lagern der Flanschlagergehäuse-Montageschnittstelle
siehe S.685
Kugelgewindetreib-Kompatibilitätsdiagramm S.685

■ Flanschlagergehäuse Loslagerseite - Kompakt

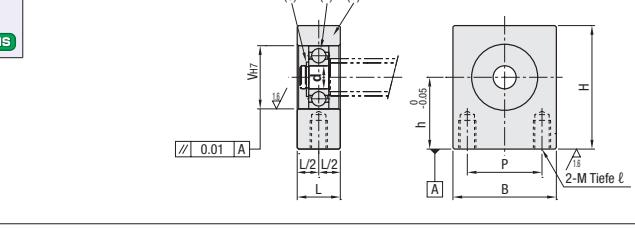


RoHS

Bauteildaten

Nr.	Bezeichnung	Stückz.
(1)	Gehäuse	1
(2)	Lager	1
(3)	Sicherungsring*	1

Ausführung	Werkstoff	Oberflächenbehandlung
BUQ	1.1191/C45E	brüniert
BUQM		Chemisch vernickelt



* Sicherungsring (3) von BUQM in Edelstahlauflaufung.

Teilenummer	Ausführung	Nr.	d	L	B	H	p	V	M	ɛ	Lager Ausführung	Sicherungsring	Stückpreis in €	Mengen- rabatt	BUQ	BUQM
8	8	6	15	32	32	17	23	17	5	6	B606ZZ	STWN-S6				
10	10	8	20	36	43	25	26	22	6	10	B608ZZ	STWN-S8				
12	12	10	20	36	43	25	26	26	6	10	B6000ZZ	STWN-S10				
15	15	15	20	46	52	30	34	32	12	12	B602ZZ	STWN-S15				

■ Flanschlagergehäuse - Loslagerseite - Flach

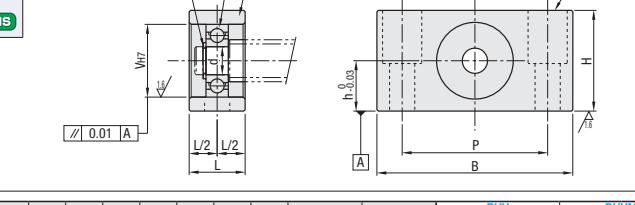


RoHS

Bauteildaten

Nr.	Bezeichnung	Stückz.
(1)	Gehäuse	1
(2)	Lager	1
(3)	Sicherungsring*	1

Ausführung	Werkstoff	Oberflächenbehandlung
BUV	1.1191/C45E	brüniert
BUVM		Chemisch vernickelt



* Sicherungsring (3) von BUV in Edelstahlauflaufung.

Teilenummer	Ausführung	Nr.	d	L	B	H	B1	H1	P	V	d1	Lager Ausführung	Sicherungsring	Stückpreis in €	Mengen- rabatt	BUV	BUVM
6	6	12	42	22	11	30	17	17	5.5	12	B606ZZ	STWN-S6					
8	8	15	52	28	14	38	17	6.6	11	15	B608ZZ	STWN-S8					
10	10	20	70	36	18	36	26	22	9	14	B6000ZZ	STWN-S10					
12	15	20	80	41	21	60	32	11	17	21	B602ZZ	STWN-S15					
20	20	95	56	28	75	47	11	17	29	29	B620ZZ	STWN-S20					

■ Flanschlagergehäuse Loslagerseite Befestigungsbohrungen mit geringem Abstand



RoHS

Bauteildaten

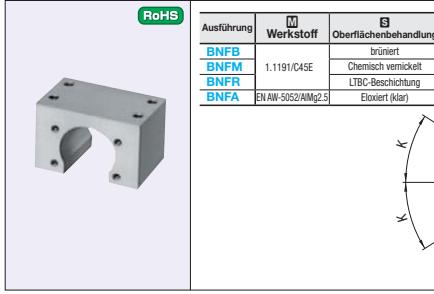
Nr.	Bezeichnung	Stückz.

<tbl_r



Halterungen für Kugelgewindetriebe

-Ausführung mit Führungswagen-



Teilenummer Ausführung	Nr.	H		Auswahl	A	B	Lochkreis-Ø	K	M1xØ	D	ε ₁	F	L	ε ₂	ε ₃	M2xØ	M4x 8	M5x 7	M5x 10	M6x 12	M8x 16	MISUMI-Kugelgewindetriebe								
		A	B																											
BNFB	801X	10	20	23	25																									
BNFM	802S	18	20	23	25																									
BNFR	802C	17	20	23	25	28	30																							
BNFA	804R	17	20	23	25	28	30																							
	1002R	20	23	25	28	30	35																							
	1002X	20	23	25	28	30	35																							
	1002C	17	20	23	25	28	30																							
	1002S	17	20	23	25	28	30																							
	1004S	17	20	23	25	28	30	35	40																					
	1004C	17	20	23	25	28	30	35	40																					
	1004R	17	20	23	25	28	30	35	40																					
	1010R	17	20	23	25	28	30	35	40																					
	1010S	17	20	23	25	28	30	35	40																					
	1202S	20	23	25	28	30	35	40																						
	1202C	20	23	25	28	30	35	40																						
	1202R	20	23	25	28	30	35	40																						
	1204S	17	20	23	25	28	30	35	40																					
	1204C	17	20	23	25	28	30	35	40																					
	1204R	17	20	23	25	28	30	35	40																					
	1205S	20	23	25	28	30	35	40	45																					
	1205C	20	23	25	28	30	35	40	45																					
	1205R	20	23	25	28	30	35	40	45																					
	1406R	20	23	25	28	30	35	40																						
	1405C	20	23	25	28	30	35	40	45																					
	1405R	20	23	25	28	30	35	40	45																					
	1405S	20	23	25	28	30	35	40	45																					
	1505C	20	23	25	28	30	35	40	45																					
	1505R	20	23	25	28	30	35	40	45																					
	1505S	20	23	25	28	30	35	40	45																					
	1510C	20	23	25	28	30	35	40	45																					
	1510R	20	23	25	28	30	35	40	45																					
	1510S	20	23	25	28	30	35	40	45																					
	1520S	20	23	25	28	30	35	40	45																					
	1520R	20	23	25	28	30	35	40	45																					
	1520C	20	23	25	28	30	35	40	45																					
	1520X	20	23	25	28	30	35	40	45																					
	2005C	25	30	35	40	45	50	55	60																					
	2005S	25	30	35	40	45	50	55	60																					
	2005R	25	30	35	40	45	50	55	60																					
	2020R	25	30	35	40	45	50	55	60																					
	2505C	25	30	35	40	45	50	55	60																					
	2505S	25	30	35	40	45	50	55	60																					
	2505R	25	30	35	40	45	50	55	60																					
	2010S	30	35	40	45	50	55	60	65																					
	2010R	30	35	40	45	50	55	60	65																					
	2010C	35	40	45	50	55	60	65	70																					
	2010X	35	40	45	50	55	60	65	70																					
	2010S	35	40	45	50	55	60	65	70																					
	2505R	35	40	45	50	55	60	65	70																					
	2505C	35	40	45	50	55	60	65	70																					
	2505X	35	40	45	50	55	60	65	70																					
	2010S	35	40	45	50	55	60	65	70																					
	2505S	35	40	45	50	55	60	65	70																					
	2505R	35	40	45	50	55	60	65	70																					
	2505X	35	40	45	50	55	60	65	70																					
	2010R	35	40	45	50	55	60	65	70																					
	2505C	35	40	45	50	55	60	65	70																					
	2505X	35	40	45	50	55	60	65	70																					
	2010C	35	40	45	50	55	60	65	70																					
	2505R	35	40	45	50	55	60	65	70																					
	2505X	35	40	45	50	55	60	65	70																					
	2010X	35	40	45	50	55	60	65	70																					
	2505S	35	40	45</td																										



Kugelgewindetriebe gewalzt Wellen-Ø 15, Steigung 5 · 10 · 20-

-Präzisionskategorie C7 · C10-

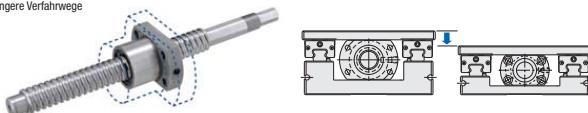
Price Reduction
bis zu 25%

CAD-Daten



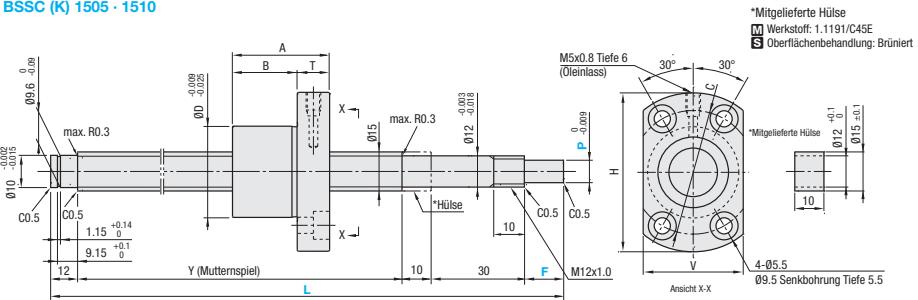
Kugelmutter	Ausführung	Standard-Konfigurationen F	Präzision Kategorie	Wellen-Ø	Steigung	Gewindewelle			Kugelmutter		
						M Werkstoff	Härte	Oberflächenbehandlung	Werkstoff	Härte	Oberflächenbehandlung
Standardmutter	BSS1	-	C7	5 · 10 · 20	15	1.1203 / G35E	-	-	1.7264/20CrMo5	-	Phosphatiert
Standardmutter	BSSR	BSSRK	C10	5 · 10 · 20	1.1203 / G35E	Induktionsgehärtet 56–62HRC	Phosphatiert	-	1.7264/20CrMo5	aufgekohlt 58–62HRC	LTBC-Verchromung
Standardmutter	BSSZ	BSSZK	-	5 · 10	-	-	-	-	1.7258/24CrMo5	-	-
Kompaktmutter	BSSC	BSSCK	-	-	-	-	-	-	-	-	-

- Eigenschaften von Kompaktmuttern-
- Kompaakter Außen-Ø
- Es können flachere Führungen verwendet werden.
- Längere Verfahrwege



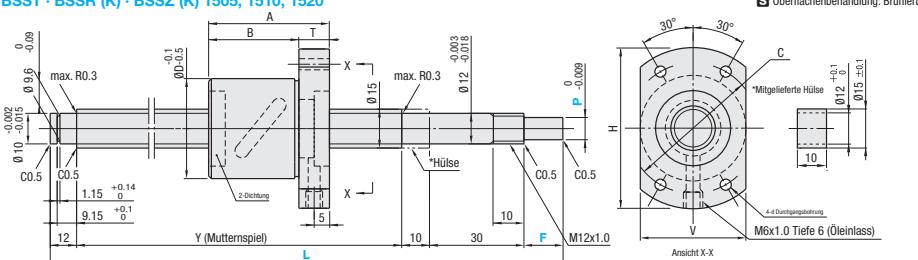
-Kompaktmuttern-

BSSC (K) 1505 · 1510



-Standardmuttern-

BSS1 - BSSR (K) · BSSZ (K) 1505, 1510, 1520



Kugelmutter-Ausführung	Präzision Kategorie	Teilenummer	1mm-Schritte			Gewindewellen-Ø	Steigung	L	F	P	Y	D	A	B	T	H	C	V	d	Kugel Ø	Kugel Mitten-Ø	Schraube Kern-Ø	Anzahl Kreise	Tragzahl O (dynamisch) / Co (statisch) kN	Zulässige Axialspiel	Drehrichtung
			Ausführung	Gewindewellen-Ø	Steigung																					
Standardausführung Kugelmutter	C7	BSS1				15	10	L-67	34	40	30	10	54	44		4.5		15.8	(125)	2,5 Unidr., in einer Reihe	5,1	10,5	max. 0,03			
		BSSR				15-30	6-10	L-(52+F)	31	33	22	11	54	42		-		15.5	(125)	3 turns, 1 row	4,14	7,06	max. 0,10			
Kompaktmutter	C10	BSSRK				15	10	L-67	34	52	42	10	57	45		5,5		15.8	(125)	2,5 Unidr., in einer Reihe	5,1	10,5	max. 0,03			
		BSSZK				15-30	6-10	L-(52+F)	31	48	37	11	54	42		-		15.5	(125)	3 turns, 1 row	4,25	7,45	max. 0,10			
Standardausführung Kugelmutter	C7	BSS1				15	10	L-67	34	59	49	10	57	45		5,5		15.8	(125)	1,5 Unidr., zwei Reihen	5,8	13,3	max. 0,03			
		BSSR				15-30	6-10	L-(52+F)	31	48	37	11	54	42		-		15.5	(125)	3 turns, 1 row	4,25	7,45	max. 0,10			
Kompaktmutter	C10	BSSZK				15	10	L-67	34	59	49	10	57	45		5,5		15.8	(125)	1,5 Unidr., zwei Reihen	5,8	13,3	max. 0,03			
Standardausführung Kugelmutter	C7	BSS1				15	10	L-67	34	59	49	10	57	45		5,5		15.8	(125)	1,5 Unidr., zwei Reihen	5,8	13,3	max. 0,03			
		BSSR				15-30	6-10	L-(52+F)	31	48	37	11	54	42		-		15.5	(125)	3 turns, 1 row	4,25	7,45	max. 0,10			
Kompaktmutter	C10	BSSZK				15	10	L-67	34	59	49	10	57	45		5,5		15.8	(125)	1,5 Unidr., zwei Reihen	5,8	13,3	max. 0,03			

*F und P sind nur für BSSRK - BSSZK und BSSCK konfigurierbar. F-Px3

kgf=Nx0.101972

Preis
euros

Kugelmutter-Ausführung	Präzision Kategorie	Teilenummer	€ Stückpreis Stückz. 1-4				
			L150-200	L201-400	L401-600	L601-800	L801-900
BSS1	C7	BSS1505					
		BSS1505					
	C10	BSS1505					
		BSS1505					
	C7	BSS1510					
		BSS1510					
	C10	BSS1510					
		BSS1510					
	C7	BSS1520					
		BSS1520					
	C10	BSS1520					
		BSS1520					

Bestell-Teilenummer - I - F - P - Liefzeit - BSST - BSSR (K) - BSSZ (K)
Bestell-Teilenummer - I - F - P - Liefzeit - BSSC (K)
Bestell-Teilenummer - I - F - P - Liefzeit - BSSC (K)

Bei noch größeren Bestellmengen bitte gesondert anfragen.
Addieren Sie für den Preis von BSS_ K (Konfigurationen F, P) 29,00 EUR zum Preis für die Standardausführung.
z. B. BSSZ1510-200-F18-P7 153,30 EUR+29,00 EUR=182,30 EUR

Hinweis

- Die Montagefläche für den BSST-Kugelmutterflansch ist nicht oberflächenbehandelt.
- Mit lithiumversteiftem Schmierstoff gefüllt (Alvania-Fett S2 von Showa Shell Sekiyu K.K.).
- Hinweise: Lassen Sie die Muttern nicht überlaufen, und trennen Sie die Muttern nicht von den Gewindewellen. Dies kann dazu führen, dass die Kugeln herausfallen oder die Kugelrückführungssteile beschädigt werden.
- Es sind verschiedene Schmierstoffausführungen erhältlich. Zu Lieferzeit, Preisen und Spezifikationen siehe S.651 BSSC ist nicht verfügbar.
- Zur Präzision der Kugelgewindetriebe siehe S.651, 653
- Einzelheiten zu Flanschlagergehäusen siehe S.685–S.694

Optionen Teilenummer - I - F - P - (FC - KC...usw.)

Optionen	Opt.-Nr.	Spez.	Aufpreis	Optionen	Opt.-Nr.	Spez.	Aufpreis
Keine Bearbeitung des Wellenendes (Loslagerseite)	NC	Ohne Bearbeitung des Wellenendes auf der Loslagerseite.	BSS1-BSSR BSS2 BSSC -25,00 -8,00	Schlüsselflächen auf Festlagerseite	SZC	Mit Schlüsselflächen am Wellenende (Festlagerseite). Die Kugellager fallen heraus, wenn die Kugelmutter mit den Schlüsselflächen in Kontakt kommt.	BSS1-BSSR BSS2 BSSC 7,00 3,00
Kugelmutter-Ausrichtung umgekehrt (Loslagerseite)	RLC	Umkehrung der Mutter-Ausrichtung.	Bestell-Nr.: RLC	Nut am Wellenende (Festlagerseite)	KC	Mit einer Nut am Wellenende (Festlagerseite) KG-1mm-Schritte Bestell-Nr.: KC10 K+S-1mm-Schritte Bestell-Nr.: KC10-K+S-1mm-Schritte K+S-1mm-Schritte Bestell-Nr.: KC10-K+S-1mm-Schritte	BSS1-BSSR BSS2 BSSC 14,00 6,00
Keine Sicherungsringnut am Wellenende (Loslagerseite)	RNC	Ohne Sicherungsringnut am Wellenende (Loslagerseite).	Bestell-Nr.: RNC	Spannfläche am Wellenende (Festlagerseite)	SC	Mit einer Spannfläche am Wellenende (Festlagerseite). SC-1mm-Schritte Bestell-Nr.: SC10 K+S-1mm-Schritte Bestell-Nr.: SC10-K+S-1mm-Schritte	BSS1-BSSR BSS2 BSSC 7,00 3,00
Andere Länge des Wellenendes (Loslagerseite)	FC	Änderung der Länge des Wellenendes auf der Loslagerseite.	Bestell-Nr.: FC20 FC-1mm-Schritte Bestell-Nr.: FC30 FC-1mm-Schritte FC-Maß ist verkürzt.	Spannfläche am Wellenende (Festlagerseite)	SGC	Mit einer Spannfläche am Wellenende (Festlagerseite). SGC-1mm-Schritte Bestell-Nr.: SGC10 K+S-1mm-Schritte Bestell-Nr.: SGC10-K+S-1mm-Schritte	BSS1-BSSR BSS2 BSSC je 7,00 je 300
Innengewinde am Wellenende (Loslagerseite)	MC	Mit einer Gewindeschlange am Wellenende (Loslagerseite). MC-1mm-Schritte Bestell-Nr.: MC20 MC-1mm-Schritte MC-Maß ist verkürzt.	Bestell-Nr.: MC20 MC-1mm-Schritte MC-Maß ist verkürzt.	2 Spannflächen am Wellenende (Festlagerseite)	SWC SGC	Mit zwei Spannflächen am Wellenende (Festlagerseite). JS 90°-90°-Winkel SGC: 120°-Winkel SGC-Maß Bestell-Nr.: JS SWC K+S-1mm-Schritte Bestell-Nr.: JS SWC K+S-1mm-Schritte Bestell-Nr.: JS SWC K+S-1mm-Schritte	BSS1-BSSR BSS2 BSSC je 7,00 je 300

Kombination mit Flanschlagergehäusen

Teilenummer Kugelgewindetrieb	Empfohlene Flanschlagergehäuse				
	Ausführung	Teilenummer	Form	Festlager	Loslager
BSV	12	Quadratische Flach	0	0	S.691
BUV	12	Ausführung Flach	0	0	S.692
BRW	12	Rund	0	0	S.693
BUR	12	Rund	0	0	S.693
BSW	12	Quadratische Ausführung	0	0	S.687
BUN	12	Quadratische Ausführung	0	0	S.689

Spanschlagergehäuse sind noch weitere Formvarianten und Oberflächenbehandlungen erhältlich. (S.685–S.694)

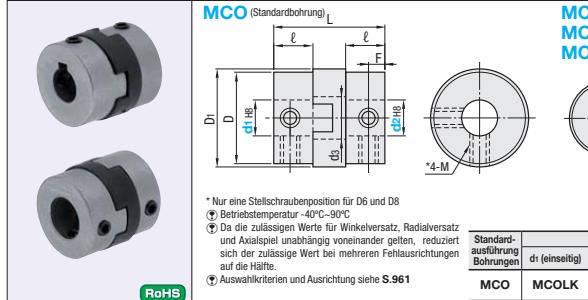
Spindelmuttern sind in mehreren Werkstoffvarianten und unterschiedlichen Oberflächenbehandlungen erhältlich. (S.695)

Kupplungen

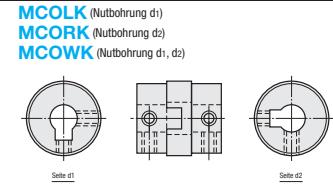
-Oldham-Kupplungen, Stellschraube/Distanzscheiben-

CAD-Daten

Eigenschaften: Nabe und Distanzscheibe trennbar, einfache Montage. (Ausführung mit Nabenklemmung siehe S.977)



* Nur eine Stellschraubenposition für D6 und D8
Betriebstemperatur -40°C...+90°C
Da die zulässigen Werte für Winkelversatz, Radialversatz und Axialspiel unabhängig voneinander gelten, reduziert sich der zulässige Wert bei mehreren Fehlausrichtungen auf die Hälfte.
Auswahlkriterien und Ausrichtung siehe S.961



Standard-ausführung	Nutbohrung			Werkstoff	Nabe	Distanz-hülsen	Zubehör
	d1 (einseitig)	d2 (einseitig)	d1, d2 (beidseitig)				
MCO	MCOLK	MCORK	MCOWK	1.4301/X5CrNi18-10 Sintermetalllegierung	Kohlefaser-verstärkter Kunststoff		Stellschraube

Teilenummer	d1, d2 wählen (d1 muss ≤ d2 sein)					Stellschraube	€ Stückpreis
	Ausführung	Nr.	D	d1	ds		
MCO	6 1 1.5 2	6	6.2	2.4	8.4	3	1.5
MCOLK	8 1 2 3	8	8.2	3.4	9.6	3.5	1.7
MCORK	10 2 2.3 4	10	10.2	4.4	10.2	3.7	1.8
MCOWK	12 3 4 5	12	12.5	4.0	14.2	5.2	2.5
	15 4 5 6 6.35 7 8	14.5	15	5.0	16	5.4	2.6
	17 5 6 6.35 7 8	16.8	17.5	7.2	19.8	6.7	3.2
	20 6 6.35 7 8 9.53 10 11 12	20	21	8.2	21.4	7	3.4
	26 6 6.35 7 8 9.53 10 11 12 14	26	27	12.0	25.6	9	4
	30 8 10 12 14	30	31	13.0	33	12	6
	34 10 11 12 14 15 16 18 20	34	35	13.0	34	13	5.5
	38 10 12 14 15 16 18 20	38	41	16.0	40	15	7

Teilenummer	Zulässiges Drehmoment (Nm)					Trägheitsmoment (kg·m²)	Zulässiges Axialspiel	Gewicht G	Nutmaß bxbh
	Zulässiger Winkelversatz (°)	Zulässiger Radialversatz (mm)	Sattige Federkonstante der Distanzscheibe (Nm/rad)	Max. Drehzahl (1/min)	Referenzdurchm. (mm)				
MCO	0.3	0.3	1.4	12000	1.0	$1.7 \cdot 10^{-5}$	± 0.25	2.5	
MCOLK	0.4	0.4	13			$2.2 \cdot 10^{-5}$	± 0.3	2.5	
MCORK	0.4	0.4	21			$3.6 \cdot 10^{-5}$	± 0.32	4	
MCOWK	0.5	0.5	44			$1.6 \cdot 10^{-7}$	± 0.35	8	
	1.2	1	90			$3.5 \cdot 10^{-7}$	± 0.45	11	
	15 1.6	1	250			$7.8 \cdot 10^{-8}$	± 0.55	18	
	20 3.2	1.5	340	8000	7.8x10 ⁻⁸			29	
	26 6	2	420	6500	6.2x10 ⁻⁸			65	
	30 15	2	1200	6500	6.2x10 ⁻⁸			100	
	34 16	2.5	2400	6000	2.5x10 ⁻⁸			100	
	38 28	2.5	3500	5800	8x10 ⁻⁸			240	

(*) Das zulässige Drehmoment ändert sich mit der Betriebstemperatur. S.961



Teilenummer - Wellbohrungs-Ød1 - Wellbohrungs-Ød2
MCO20 - 6 - 6
MCOLK20 - 8 - 12
MCORK20 - 10 - 12



Teilenummer - Wellbohrungs-Ø d(LDC) - Wellbohrungs-Ø d(RDC) - (KLH - KRL)
MCOG20 - LDC6.5 - RDC9
MCOGWK30 - 8 - 10 - KRH4

Arbeits-tage

Kein Expressversand.



Standardbohrung

8 Arbeits-tage

Express A 2,00 EUR/Stk. S. 88

Expressgebühr von 5,40 EUR für 3 oder mehr identische Teile.



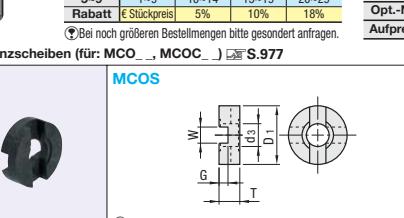
Mengenrabatt (Abgerundet auf einen Cent.) S. 87

5-9 1-9 10-14 15-19 20-29
Rabatt € Stückpreis 5% 10% 18%

Bei noch größeren Bestellmengen bitte gesondert anfragen.



Distanzscheiben (für: MCO_ ... MCOC_ ...) S. 977



Kein Expressversand.

Bestellbeispiel MCOS15



Teilenummer 8 Arbeits-tage
Express A 2,00 EUR/Stk. S. 88

Expressgebühr von 5,40 EUR für 3 oder mehr identische Teile.



Mengenrabatt (Abgerundet auf einen Cent.) S. 87

5-9 1-9 10-12 13-14 15-29
Rabatt € Stückpreis 5% 10% 18%

Bei noch größeren Bestellmengen bitte gesondert anfragen.

Kupplungen

-Torsionssteife Oldham-Kupplungen, Stellschraube-

CAD-Daten

Eigenschaften: Die Distanzscheibe besteht aus Aluminiumbronze. Das zulässige Drehmoment ist doppelt so hoch wie bei der Kunststoffausführung. (Ausführung mit Nabenklemmung siehe S.978)

Teilenummer	d1, d2 wählen (d1 muss ≤ d2 sein)					Stellschraube	€ Stückpreis
	Ausführung	Nr.	D	d1	ds		
MCOG	15 4 5 6 6.35 7 8	15	7.2	16	5.4	2.6	M3 0.7
MCOGRK	17 5 6 6.35 7 8	17	8.2	19.8	6.7	3.2	
MCOGWK	20 6 6.35 7 8 9.53 10 11 12	20	9	21.4	7	3.4	
	26 6 6.35 7 8 9.53 10 11 12 14	26	27	12	25.6	9	M4 1.7
	30 8 10 12 14	30	31	14	33	12	6
	34 10 11 12 14 15 16 18 20	34	35	14	34	13	5.5
	38 10 12 14 15 16 18 20	38	41	17	39.5	15	7

(*) Auswahlkriterien und Ausrichtung siehe S.961

Teilenummer	d1, d2 wählen (d1 muss ≤ d2 sein)					Stellschraube	€ Stückpreis
	Ausführung	Nr.	D	d1	ds		
MCOG	15 4 5 6 6.35 7 8	15	7.2	16	5.4	2.6	M3 0.7
MCOGRK	17 5 6 6.35 7 8	17	8.2	19.8	6.7	3.2	
MCOGWK	20 6 6.35 7 8 9.53 10 11 12	20	9	21.4	7	3.4	
	26 6 6.35 7 8 9.53 10 11 12 14	26	27	12	25.6	9	M4 1.7
	30 8 10 12 14	30	31	14	33	12	6
	34 10 11 12 14 15 16 18 20	34	35	14	34	13	5.5
	38 10 12 14 15 16 18 20	38	41	17	39.5	15	7

Teilenummer	d1, d2 wählen (d1 muss ≤ d2 sein)					Stellschraube	€ Stückpreis
	Ausführung	Nr.	D	d1	ds		
MCOG	15 3	15	800	8000	4×10^{-8}	± 0.1	15
MCOGRK	17 5	17	1000	7000	1×10^{-7}	± 0.1	25
MCOGWK	20 7	20	2200	6000	2×10^{-6}	± 0.1	37
	26 10	26	4000	5000	6×10^{-6}	± 0.2	79
	30 30	30	5500	5000	2.5×10^{-5}	± 0.3	120
	34 32	34	8000	4000	4×10^{-5}	± 0.2	180
	38 50	38	11000	4000	1×10^{-4}	± 0.3	256

Teilenummer	d1, d2 wählen (d1 muss ≤ d2 sein)					Stellschraube	€ Stückpreis
	Ausführung	Nr.	D	d1	ds		
MCOS	15 4 5 6.35 7 8	15	4.8	8	4.8	1	
	17 5 6 6.35 7 8	17	5.8	8	5.8	1	
	20 6 6.35 7 8 9.53 10 11 12	20	6.8	12	12	1	
	26 10 6 6.35 7 8 9.53 10 11 12 14	26	6.8	15	15	1	
	30 14 6 6.35 7 8 9.53 10 11 12 14 16	30	8.8	18	18	1	
	34 16 6 6.35 7 8 9.53 10 11 12 14 16 18	34	10.8	20	20	1	
	38 20 6 6.35 7 8 9.53 10 11 12 14 16 18 20	38	12.8	22	22	1	

Teilenummer	d1, d2 wählen (d1 muss ≤ d2 sein)					Stellschraube	€ Stückpreis
	Ausführung	Nr.	D	d1	ds		
MCOS	15 4 5 6.35 7 8	15	4.8	8	4.8	1	
	17 5 6 6.35 7 8	17	5.8	8	5.8	1	
	20 6 6.35 7 8 9.53 10 11 12	20	6.8	12	12	1	
	26 10 6 6.35 7 8 9.53 10 11 12 14	26	6.8	15	15	1	
	30 14 6 6.35 7 8 9.53 10 11 12 14 16	30	8.8	18	18	1	
	34 16 6 6.35 7 8 9.53 10 11 12 14 16 18	34	10.8	20	20	1	
	38 20 6 6.35 7 8 9.53 10 11 12 14 16 18 2						



Linearführungen für mittlere Traglast

-mit Zylinderstiftbohrung/Normales Spiel-

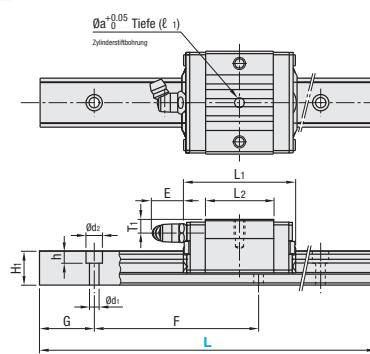
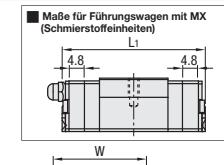
■ Merkmale: Originalspezifikationen von MISUMI mit Zylinderstiftbohrungen. Spart Zeit bei der Montage und bietet verbesserte Wiederholgenauigkeit.



■ Führungswagen und -schienen können nicht separat bestellt werden. Ausführung mit normalem Spiel bietet ebenso wie die Sätze aus Führungswagen und -schienen ein garantiertes radiales Spiel und Genauigkeit.

Ausführung Standardwert		Maß L	Werkstoff Härte
SVRN		Frei wählbar	JIS STKM (Stahllegierung einschließlich JIS SCM) 58HRC~
SV2RN		2	
SVRLN		1	
SV2RLN		2	
SSVRN		Frei wählbar	Edestahl (Werkstoff: 1.4125/ X105CrMo17) 56HRC~
SSV2RN		1	
SSV2RLN		2	

Wärmebeständigkeit: -20~80°C



Vorsichtsmaßnahmen

■ Führungswagen sind mit Kugellagern ausgerüstet, um zu verhindern, dass Kugeln herausfallen. Siehe [S.501](#)

■ Radiales Spiel und Genauigkeit werden nicht garantiert, wenn die Führungswagen und -schienen aus der Kombination des Originalsets heraus ausgetauscht werden.

■ Auf den Bezugsfächeln sind gerade Nuten eingebracht. Stellen Sie sicher, dass beim Zusammenstellen die Bezugsfächeln passen.

■ Die Schienen können nicht zur Verlängerung verbunden werden.

■ Laufparallelität ist der Wert, der nach Montage der Führungsschiene gemessen wird.

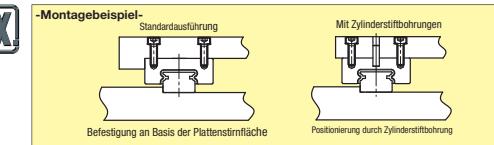
Ausführung	MX	H	Führungswagenmaße								Führungsschiene maße														
			W ± 0.005	L1	B	SxL	L2	K	T	CB	Schmiernippel	Befestigungsbohrungen	Zylinderstiftbohrungsmasse	H1	W1 ± 0.005	W2	Ca	Senkbohrungen	dixdzkh	F	G				
(1 Führungswagen) (2 Führungswagen)																									
SVRN	SV2RN		24	34	41	50.6	26	M4x7	25	20	7	0.85	M5xP0.8	6	5	17	3	4	12.5	15	9.5	0.5	3.5x6x4.5	60	20
SVRN	SV2RNL		28	42	47	56.6	32	M5x8	27.6	22.5	7.5	1	M6xP0.75	13	6	21	4	5	15.5	20	11	0.6	6x9.5x8.5	60	20
SSVRN	SSV2RN		33	48	59	68.6	35	M6x9	37	26.5	8	1	M6xP0.75	13	6.8	24	4	5	18	23	12.5	0.8	7x11x9	60	20
SSV2RN	SSV2RLN																								

$$kgf=Nx0.101972$$

H	G (Dynamisch) [kN]	Co (Statisch) [kN]	Mx/Ms [Nm]	Mc [Nm]	Führungsmaße	Referenzelle	Standards für Vorlast und Genauigkeit	
							Radielles Spiel (μm)	Maßgenauigkeit (μm)
24	5.0	8.23	33	57	0.15	1.5	H24: -4~+2	Höhe H Toleranz ±100
28	7.2	12.1	58	135	0.20	2.4	H28: -5~+2	Höhe H paarweise Änderung 20
33	11.7	19.6	109	225	0.30	3.4	H33: -6~+3	Breite Wz: Toleranz ±100



-Vorteile der Schmiereinheit MX
Langfristiger wartungsfreier Betrieb
Geringere Instandhaltungskosten
Besonders zu empfehlen an unzugänglichen Schmiersätzen.
Weitere Informationen finden Sie auf [S.498](#).



Schmiereinheiten MX Neu

Bieten langfristigen wartungsfreien Betrieb.

Bestellbeispiel	Teilenummer	- L -
	SVRN20	- 000 -
	SVRN-MX28	- 880 -
	RVRN28	- 880 -
	SVRN28L	- 880 -
	SVRN28G	- 880 -

Alternative Schmiereinheiten verfügbar (außer für Führungsschienen mit Schmiereinheiten). Für Lieferzeit, Preis und Leistung, siehe [S.507](#)

Lieferzeit	Arbeits-tage	Express A	8,00 EUR/Stück
8			

Expressgebühr von 21,60 EUR für 3 oder mehr identische Teile.

Artikel für Linearführung			
Positionierungshilfen	Führungs- schienen Distanz- blöcke	Führungswagen Abflaschutz	Befestigungs- teile Wagen/ Schienen
			S.581~ S.588

Price Reduction
10%



CAD-Daten

Optionen	Teilenummer	- L -	(TMS - TMC - RLC - LLC)
	SVRN28	- 880 -	TMC

Keine Express-Lieferung für TMS und TMC.

Optionen	Opt.-Nr.	Spez.	Aufpreis
TMS Bearbeitung d. Innengewinde + 2 Anschlagplatten	TMS	Es werden beidseitig Innengewindebohrungen an den Führungsschienen angebracht, um zu verhindern, dass die Führungswagen herausfallen.	
TMC Bearbeitung d. Innengewinde	TMC	Innengewindebohrungen an den Führungsschienen angebracht, um zu verhindern, dass die Führungswagen herausfallen.	
LLC	LLC	Beschneidung der Führungsschienenenden.	
Zuschmitt links (LLC)		Zuschmitt links (LLC)	8,00
Zuschmitt rechts (RLC)		Zuschmitt rechts (RLC)	8,00

■ Artikel mit LTBC-Beschichtung (Verfügbar nur für Standard-Ausführung.)

■ Teilen: Andem Sie den ersten Buchstaben „S“ für die Standard-Ausführung auf „R“.

■ Informationen zu Merkmalen und Präzisionskategorien finden Sie auf [S.508](#)

■ Für Artikel mit LTBC-Beschichtung addieren Sie den Preis für die Oberflächenbehandlung auf S.508 zu dem Preis der Standard-Ausführung hinzu.

■ Mengenrabatt (Auf einen Cent abgerundet, S.87)

■ Stückz.: 1~4 5 oder 6 7 oder 8 9 oder 10

■ Rabatt: 5% 10% 15% 18%

■ Markierte Größen sind nicht verfügbar für LTBC-beschichtete Ausführung.

■ Teilen: Andem Sie den ersten Buchstaben „S“ für die Standard-Ausführung auf „R“.

■ Informationen zu Merkmalen und Präzisionskategorien finden Sie auf [S.508](#)

■ Für Artikel mit LTBC-Beschichtung addieren Sie den Preis für die Oberflächenbehandlung auf S.508 zu dem Preis der Standard-Ausführung hinzu.

■ Mengenrabatt (Auf einen Cent abgerundet, S.87)

■ Stückz.: 1~4 5 oder 6 7 oder 8 9 oder 10

■ Rabatt: 5% 10% 15% 18%

■ Markierte Größen sind nicht verfügbar für LTBC-beschichtete Ausführung.

■ Teilen: Andem Sie den ersten Buchstaben „S“ für die Standard-Ausführung auf „R“.

■ Informationen zu Merkmalen und Präzisionskategorien finden Sie auf [S.508](#)

■ Für Artikel mit LTBC-Beschichtung addieren Sie den Preis für die Oberflächenbehandlung auf S.508 zu dem Preis der Standard-Ausführung hinzu.

■ Mengenrabatt (Auf einen Cent abgerundet, S.87)

■ Stückz.: 1~4 5 oder 6 7 oder 8 9 oder 10

■ Rabatt: 5% 10% 15% 18%

■ Markierte Größen sind nicht verfügbar für LTBC-beschichtete Ausführung.

■ Teilen: Andem Sie den ersten Buchstaben „S“ für die Standard-Ausführung auf „R“.

■ Informationen zu Merkmalen und Präzisionskategorien finden Sie auf [S.508](#)

■ Für Artikel mit LTBC-Beschichtung addieren Sie den Preis für die Oberflächenbehandlung auf S.508 zu dem Preis der Standard-Ausführung hinzu.

■ Mengenrabatt (Auf einen Cent abgerundet, S.87)

■ Stückz.: 1~4 5 oder 6 7 oder 8 9 oder 10

■ Rabatt: 5% 10% 15% 18%

■ Markierte Größen sind nicht verfügbar für LTBC-beschichtete Ausführung.

■ Teilen: Andem Sie den ersten Buchstaben „S“ für die Standard-Ausführung auf „R“.

■ Informationen zu Merkmalen und Präzisionskategorien finden Sie auf [S.508](#)

■ Für Artikel mit LTBC-Beschichtung addieren Sie den Preis für die Oberflächenbehandlung auf S.508 zu dem Preis der Standard-Ausführung hinzu.

■ Mengenrabatt (Auf einen Cent abgerundet, S.87)

■ Stückz.: 1~4 5 oder 6 7 oder 8 9 oder 10

■ Rabatt: 5% 10% 15% 18%

■ Markierte Größen sind nicht verfügbar für LTBC-beschichtete Ausführung.

■ Teilen: Andem Sie den ersten Buchstaben „S“ für die Standard-Ausführung auf „R“.

■ Informationen zu Merkmalen und Präzisionskategorien finden Sie auf [S.508](#)

■ Für Artikel mit LTBC-Beschichtung addieren Sie den Preis für die Oberflächenbehandlung auf S.508 zu dem Preis der Standard-Ausführung hinzu.

■ Mengenrabatt (Auf einen Cent abgerundet, S.87)

■ Stückz.: 1~4 5 oder 6 7 oder 8 9 oder 10

■ Rabatt: 5% 10% 15% 18%

■ Markierte Größen sind nicht verfügbar für LTBC-beschichtete Ausführung.

■ Teilen: Andem Sie den ersten Buchstaben „S“ für die Standard-Ausführung auf „R“.

■ Informationen zu Merkmalen und Präzisionskategorien finden Sie auf [S.508](#)

■ Für Artikel mit LTBC-Beschichtung addieren Sie den Preis für die Oberflächenbehandlung auf S.508 zu dem Preis der Standard-Ausführung hinzu.

■ Mengenrabatt (Auf einen Cent abgerundet, S.87)

■ Stückz.: 1~4 5 oder 6 7 oder 8 9 oder 10

■ Rabatt: 5% 10% 15% 18%

■ Markierte Größen sind nicht verfügbar für LTBC-beschichtete Ausführung.

■ Teilen: Andem Sie den ersten Buchstaben „S“ für die Standard-Ausführung auf „R“.

■ Informationen zu Merkmalen und Präzisionskategorien finden Sie auf [S.508](#)

■ Für Artikel mit LTBC-Beschichtung addieren Sie den Preis für die Oberflächenbehandlung auf S.508 zu dem Preis der Standard-Ausführung hinzu.

■ Mengenrabatt (Auf einen Cent abgerundet, S.87)

■ Stückz.: 1~4 5 oder 6 7 oder 8 9 oder 10

■ Rabatt: 5% 10% 15% 18%

■ Markierte Größen sind nicht verfügbar für LTBC-beschichtete Ausführung.

■ Teilen: Andem Sie den ersten Buchstaben „S“ für die Standard-Ausführung auf „R“.

■ Informationen zu Merkmalen und Präzisionskategorien finden Sie auf [S.508](#)

■ Für Artikel mit LTBC-Beschichtung addieren Sie den Preis für die Oberflächenbehandlung auf S.508 zu dem Preis der Standard-Ausführung hinzu.

■ Mengenrabatt (Auf einen Cent abgerundet, S.87)

■ Stückz.: 1~4 5 oder 6 7 oder 8 9 oder 10

■ Rabatt: 5% 10% 15% 18%

■ Markierte Größen sind nicht verfügbar für LTBC-beschichtete Ausführung.

■ Teilen: Andem Sie den ersten Buchstaben „S“ für die Standard-Ausführung auf „R“.

■ Informationen zu Merkmalen und Präzisionskategorien finden Sie auf [S.508](#)

■ Für Artikel mit LTBC-Beschichtung addieren Sie den Preis für die Oberflächenbehandlung auf S.508 zu dem Preis der Standard-Ausführung hinzu.

■ Mengenrabatt (Auf einen Cent abgerundet, S.87)

■ Stückz.: 1~4 5 oder 6 7 oder 8 9 oder 10

■ Rabatt: 5% 10% 15% 18%

■ Markierte Größen sind nicht verfügbar für LTBC-beschichtete Ausführung.

■ Teilen: Andem Sie den ersten Buchstaben „S“ für die Standard-Ausführung auf „R“.

■ Informationen zu Merkmalen und Präzisionskategorien finden Sie auf [S.508](#)

■ Für Artikel mit LTBC-Beschichtung addieren Sie den Preis für die Oberflächenbehandlung auf S.508 zu dem Preis der Standard-Ausführung hinzu.

■ Mengenrabatt (Auf einen Cent abgerundet, S.87)

■ Stückz.: 1~4 5 oder 6 7 oder 8 9 oder 10

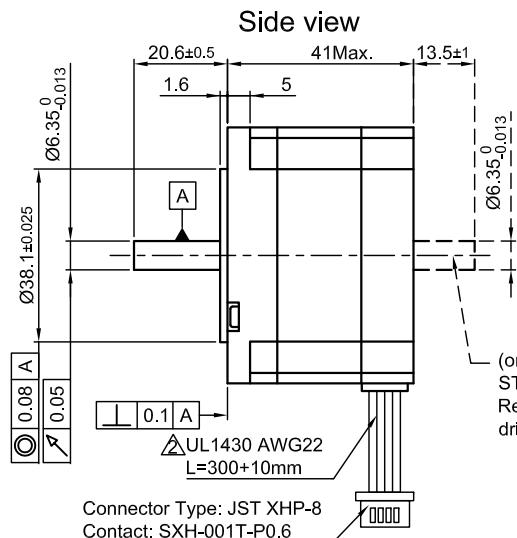
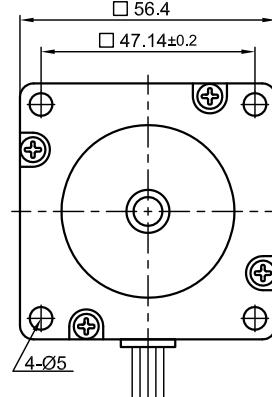
■ Rabatt: 5% 10% 15% 18%

■ Markierte Größen sind nicht verfügbar für LTBC-beschichtete Ausführung.

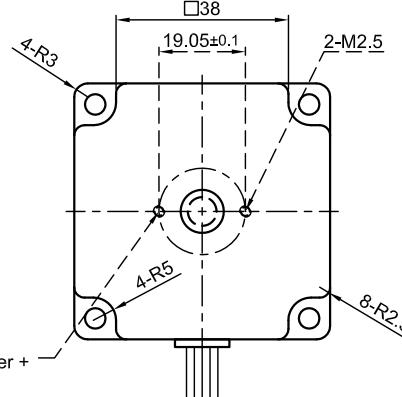
■ Teilen: Andem Sie den ersten Buchstaben „S“ für die Standard-Ausführung auf „R“.

■ Informationen zu Merkmalen und Präzisionskategorien finden Sie auf [S.508](#)

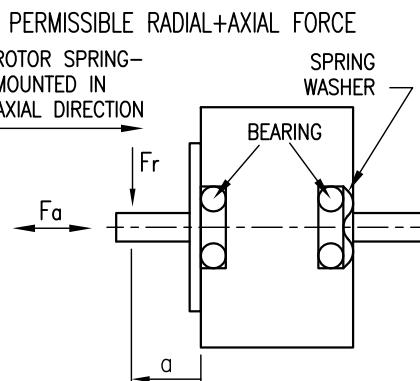
Front view and mounting



Rear view



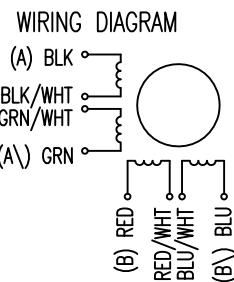
SPECIFICATION	CONNECTION	UNIPOLAR OR BIPOLAR-1 WINDING	BIPOLAR		PERMISSIBLE RADIAL+AXIAL FORCE ROTOR SPRING-MOUNTED IN AXIAL DIRECTION
			SERIAL	PARALLEL	
VOLTAGE (VDC)		5.0			
AMPS/PHASE		1.0	0.71	1.41	
RESISTANCE/PHASE (Ohms)@25°C		5.0±15%	10±15%	2.5±15%	
INDUCTANCE/PHASE (mH) @1KHz		5.4±20%	21.6±20%	5.4±20%	
HOLDING TORQUE (Nm) [lb-in]		0.38 [3.363]	0.537 [4.752]	0.537 [4.752]	
DETENT TORQUE (Nm) [lb-in]		0.0147 [0.13]			
STEP ANGLE (°) ± ACCURACY		1.8±5% (NON-ACCUM)			
BACK-EMF (V) (300 U/min)			16.13		
ROTOR INERTIA (Kg-m²) [lb-in²]		1.35x10⁻⁵ [0.046]			
WEIGHT (Kg) [lb]		0.49 [1.08]			
TEMPERATURE RISE: MAX.80°C (MOTOR STANDSTILL; FOR 2 PHASE ENERGIZED)					AXIAL-FORCE F_a (N) $F_a=10$
AMBIENT TEMPERATURE -10°~ 50°C [14°F ~ 122°F]					DISTANCE a (mm) 5 10 15 20
INSULATION RESISTANCE 100 MOhm (UNDER NORMAL TEMPERATURE AND HUMIDITY)					RADIAL-FORCE F_r (N) 130 90 70 52
INSULATION CLASS B 130° [266°F]					AXIAL RADIAL
DIELECTRIC STRENGTH 500VAC FOR 1 MIN. (BETWEEN THE MOTOR COILS AND THE MOTOR CASE)					SHAFT PLAY (mm) 0.08 0.02
AMBIENT HUMIDITY MAX. 85% (NO CONDENSATION)					AT LOAD MAX: (N) 4.5 4.5



UNIPOLAR	TYPE OF CONNECTION (EXTERN)			MOTOR		
	BIPOLAR 1WINDING	BIPOLAR SERIAL	BIPOLAR PARALLEL	CONNECTOR PIN NO. ▲	LEADS	WINDING
A — A	A —	A —	A —	1	BLK	A
COM —	A —	—	—	3	BLK/WHT	A\
A\ —	A\ —	A\ —	A\ —	2	GRN/WHT	A
B —	B —	B —	B —	4	GRN	B\
COM —	B —	—	—	5	RED	B
B\ —	B\ —	B\ —	B\ —	7	RED/WHT	B\
				6	BLU/WHT	BLU
				8	BLU	BLU

FULL STEP 2 PHASE-Ex.,
WHEN FACING MOUNTING END (X)

STEP	A	B	A\	B\	CCW
1	+	+	-	-	
2	-	+	+	-	
3	-	-	+	+	
4	+	-	-	+	CW



 **Nanotec®**
PLUG & DRIVE
ST5918X1008

REV	DESCRIPTION	DATE	APVD
2	NEW UL NO.	01.09.08	J.W.
1	PIN-ASSIGNMENT	08.01.08	J.W.

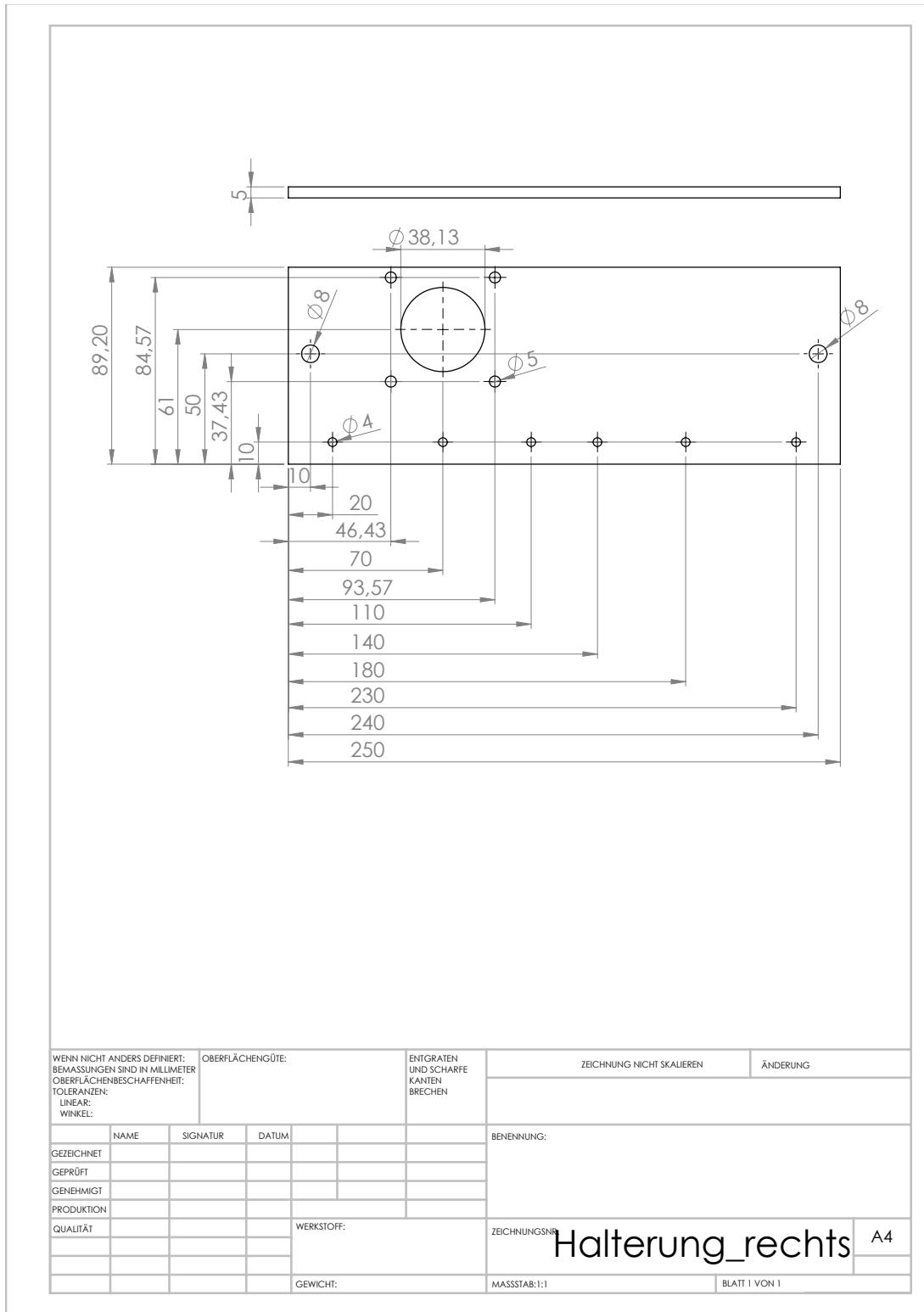
SCALE	FREE	APVD	S.Ha.	19.03.07
X	±0.5	CHKD		
1PL	±0.2			
2PL	±0.1	DRN	J.W.	22.11.06
ANGLE	±30'			

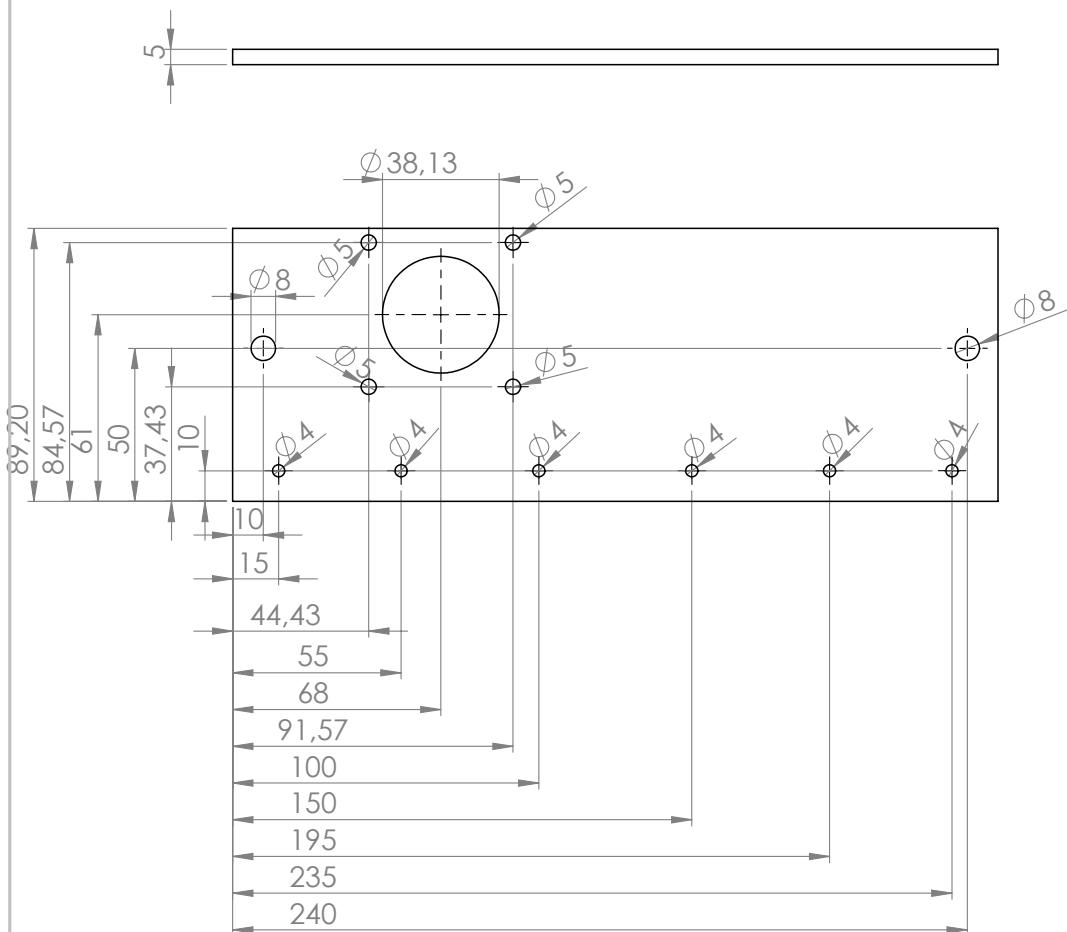
SIGNATURE	DATE

STEPPING MOTOR
DWG.NO. ST5918X1008

A.4. Y-Achse

Verschiedene Maße und wichtige Daten der Y-Achsen können aus den Technischen Zeichnungen und aus den Datenblättern der folgenden Seiten entnommen werden.





WENN NICHT ANDERS DEFINIERT: BEMASSTUNGEN SIND IN MILLIMETER OBERFLÄCHENBESCHAFFENHEIT: TOLERANZEN: LINEAR: WINKEL:				OBERFLÄCHENGÜTE:	ENTGRATEN UND SCHARFE KANTEN BRECHEN	ZEICHNUNG NICHT SKALIEREN	ÄNDERUNG
GEZEICHNET	NAME	SIGNUR	DATUM			BENENNUNG:	
GEPRÜFT							
GENEHMIGT							
PRODUKTION							
QUALITÄT				WERKSTOFF:	ZEICHNUNGSNR. Motorplatte_y.Achse^{A4}		
				GEWICHT:	MASSSTAB:1:1	BLATT 1 VON 1	



Flanschlagergehäuse · quadratisch

-Festlager Kompaktausführung/Festlager Niedrigprofil/Festlager Befestigungsbohrungen mit geringem Abstand-



Flanschlagergehäuse · quadratisch

-Loslagerseite Kompaktausführung/Loslagerseite Flach/Loslagerseite Befestigungsbohrungen mit geringem Abstand-

CAD-Daten

• Durch seine Kompaktheit (geringere Höhe und Breite im Vergleich mit konventionellen Flanschlagergehäusen) platzsparend einsetzbar und insbesondere für kleine Geräte geeignet.

■ Flanschlagergehäuse - Festlagerseite - Kompakt



RoHS

Bauteildaten

Nr.	Bezeichnung	Stückz.
(1)	Lagergehäuse	1
(2)	Schrägkugellager Klasse 0	1 Satz
(3)	Abdeckung	1
(4)	Hülse	1
(5)	Klemmutter (mit Druckstück)	1
(6)	Radialwellendichtring	2

Kompakteste Größe.

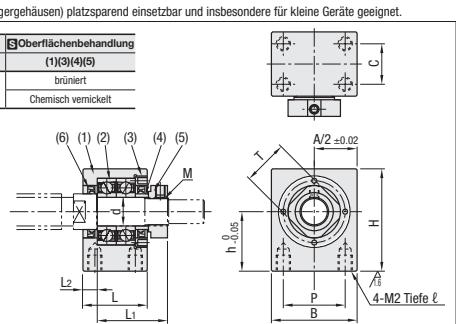
Demontieren Sie (1), (2) und (3) nicht, da diese als eine Montageeinheit zusammengehören.

d=8 wird mit nur einer Oldichtigung (6) geliefert.

*Schrägkugellager Klasse 5 bei Nr. 8.
Bei Nr. 10S ist der Innen-Ø der Oldichtigung geringer als bei der vorherigen Ausführung.

Ausführung | Werkstoff | Oberflächenbehandlung

Ausführung	Werkstoff	Oberflächenbehandlung
BSQ	1.1191/C45E	1.0038/S235JR brüniert
BSQM		Chemisch vernickelt



■ Flanschlagergehäuse - Festlagerseite - Flach



RoHS

Bauteildaten

Nr.	Bezeichnung	Stückz.
(1)	Lagergehäuse	1
(2)	Schrägkugellager Klasse 5	1 Satz
(3)	Abdeckung	1
(4)	Zylinderschraube mit Innensechskant	4
(5)	Hülse	1
(6)	Klemmutter (mit Druckstück)	1
(7)	Radialwellendichtring	2

Keine Oldichtigung (7), wenn d=6.

d=8 wird mit nur einer Oldichtigung (7) geliefert.

Demontieren Sie (1), (2) und (3) nicht, da diese als eine Montageeinheit zusammengehören.

Schrauben (4) für BSVM in Edelstahlauflaufung.

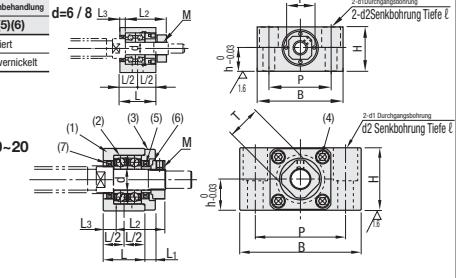
*Bei Nr. 8S und 10S ist der Innen-Ø der Oldichtigung geringer als bei der vorherigen Ausführung.

Ausführung | Werkstoff | Oberflächenbehandlung

Ausführung	Werkstoff	Oberflächenbehandlung
BSV	1.1191/C45E	1.0038/S235JR brüniert
BSVM		Chemisch vernickelt

Flachste Ausführung.

d=10-20



■ Flanschlagergehäuse - Festlagerseite - Befestigungsbohrungen mit geringem Abstand



RoHS

Bauteildaten

Nr.	Bezeichnung	Stückz.
(1)	Lagergehäuse	1
(2)	Schrägkugellager Klasse 5	1 Satz
(3)	Abdeckung	1
(4)	Zylinderschraube mit Innensechskant	4
(5)	Hülse	1
(6)	Klemmutter (mit Druckstück)	1
(7)	Radialwellendichtring	2

Auslegungsbreite für Kompatzausführung.

Demontieren Sie (1), (2) und (3) nicht, da diese als eine Montageeinheit zusammengehören.

Schrauben (4) für BSA in Edelstahlauflaufung.

*Bei Nr. 10S ist der Innen-Ø der Oldichtigung geringer als bei der vorherigen Ausführung.

(4) Der Schraubenkopf (4) ragt um 2mm aus der Abdeckung heraus.

Ausführung | Werkstoff | Oberflächenbehandlung

Ausführung	Werkstoff	Oberflächenbehandlung
BSA	1.1191/C45E	1.0038/S235JR brüniert
BSAM		Chemisch vernickelt

Auslegungsbreite für Kompatzausführung.

(7) Demontieren Sie (1), (2) und (3) nicht, da diese als eine Montageeinheit zusammengehören.

Schrauben (4) für BSAM in Edelstahlauflaufung.

*Bei Nr. 10S ist der Innen-Ø der Oldichtigung geringer als bei der vorherigen Ausführung.

(4) Der Schraubenkopf (4) ragt um 2mm aus der Abdeckung heraus.

Ausführung | Werkstoff | Oberflächenbehandlung

Ausführung	Werkstoff	Oberflächenbehandlung
BSA	1.1191/C45E	1.0038/S235JR brüniert
BSAM		Chemisch vernickelt

Auslegungsbreite für Kompatzausführung.



Teilenummer

Ausführung	Nr.	d	L	L1	L2	L3	B	H	B1	H1	P	C	d1	M (Fein)	T	Radialwellendichtring Verfügbarer Wellen-Ø	Stückpreis in €	Mengen- rabatt	Stückpreis in €	Mengen- rabatt
BSA	10S	10	25	4	29.5	5	60	39	22	34	21	46	13	6.6	M10x1.0	17	12			
BSA	10	20	25	4	29.5	6	70	36	18	52	9	19	17	10	M10x1.0	17	12			
BSA	12	24	6	29.5	6	70	36	18	52	9	14	19	15	12	M12x1.0	19	15			
BSA	15	25	6	38	5	80	41	21	60	11	17	21	15	15	M15x1.0	22	20			
BSA	20	20	42	10	52	10	95	56	28	75	11	29	25	20	M20x1.0	30	25			

Teilenummer

Bestell- beispiel	Teilenummer
BSM6	BSV10S
BSA12	BSAM

Liefer-
zeit

6 Arbeits-
tage

BT.S.87

Für Einzelheiten zur Präzision und zu den Lagern der Flanschlagergehäuse-Montageschnittstelle
siehe S.685

Kugelgewindetreib-Kompatibilitätsdiagramm S.685

CAD-Daten

Gleiche Maße wie bei den Festlager-Flanschlagergehäusen, aber in Höhe und Breite kompakter. Verwenden Sie die Festlager-Flanschlagergehäuse zusammen.

■ Flanschlagergehäuse - Loslagerseite - Kompakt

RoHS

Bauteildaten

Nr.	Bezeichnung	Stückz.
(1)	Gehäuse	1
(2)	Lager	1
(3)	Sicherungsring*	1

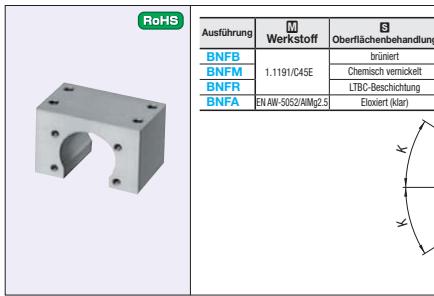
*Sicherungsring (3) von BUQM in Edelstahlauflaufung.

RoHS



Halterungen für Kugelgewindetriebe

-Ausführung mit Führungswagen-



Teilenummer Ausführung	Nr.	H		A	B	Lochkreis-Ø	K	M1xℓ	D	ℓ₁	F	L	ℓ₂	ℓ₃	M2xℓ	MISUMI-Kugelgewindetriebe				
		Auswahl	H													BSX(C3)	BSS(C5)	BSSE(C7)	BS(C10)	
BNFB	801X	10	20	23	25		30	M3x5	12,4	24	8	16	3		10	M3x5	0801	0802	-	-
BNFM	802S	18	20	23	25		40	M3x5	20,4	30	14	28	8				-	-	0802	-
BNFR	802C	17	20	23	25	28	30	M4x7	16,4	30	12						-	-	0802	-
BNFA	804R	17	20	23	25	28	30	M3x5	20,4	30	17						-	-	0804	-
	1002R	17	20	23	25	28	30	M4x7	23,4	42	16	24	6				-	-	1002	-
	1002C	17	20	23	25	28	30	M5x7	24,4	42	23	24	6				-	-	1004	-
	1004S	17	20	23	25	28	30	M4x7	22,4	40	18						-	-	1004	-
	1004C	17	20	23	25	28	30	M5x7	26,4	42	22	32	10	16			-	-	1004	-
	1004R	17	20	23	25	28	30	M4x7	22,4	42	22						-	-	1010	-
	1010R	17	20	23	25	28	30	M5x7	25,4	42	20	28	8	14			-	-	1020	-
	1010S	17	20	23	25	28	30	M4x7	24,4	42	23	27	6	15			-	-	1204	-
	1202S	20	23	25	28	30	35	M5x7	30,4	50	24	34	7	20			-	-	1204	-
	1204C	20	23	25	28	30	35	M4x7	30,4	50	24	34	7	20			-	-	1204	-
	1204R	20	23	25	28	30	35	M5x7	30,4	50	24	36	8				-	-	1210	-
	1205S	20	23	25	28	30	35	M4x7	30,4	50	24	40	10				-	-	1405	-
	1210S	20	23	25	28	30	35	M6x12	34,4	52	28	36	8				-	-	1505	-
	1406R	20	23	25	28	30	35	M4x7	31,4	52	24	38	7	24			-	-	1505	-
	1505C	20	23	25	28	30	35	M4x7	31,4	52	28	36	8				-	-	1510	-
	1505R	20	23	25	28	30	35	M5x7	31,4	52	24	38	7	24			-	-	1510	-
	1505S	20	23	25	28	30	35	M4x7	31,4	52	28	44	7	30			-	-	1520	-
	1510C	20	23	25	28	30	35	M6x12	34,4	52	28	50	10				-	-	2005	-
	1510R	20	23	25	28	30	35	M6x10	36,4	60	26	40	10				-	-	2010	-
	1520S	20	23	25	28	30	35	M6x10	36,4	60	26	50	10				-	-	2005	-
	2005C	25	28	30	35	40	45	M6x10	36,4	60	26	55	10				-	-	2005	-
	2010C	25	28	30	35	40	45	M6x10	36,4	60	26	55	10				-	-	2020	-
	2005S	25	28	30	35	40	45	M6x10	36,4	60	26	55	10				-	-	2040	-
	2005R	25	28	30	35	40	45	M6x10	36,4	60	26	55	10				-	-	2505	-
	2020R	25	28	30	35	40	45	M6x10	36,4	60	26	55	10				-	-	2010/2020	-
	2505C	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505R	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505S	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505R	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505S	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505R	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505S	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505R	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505S	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505R	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505S	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505R	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505S	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505R	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505S	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505R	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505S	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505R	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505S	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505R	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505S	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505R	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505S	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505R	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505S	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505R	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505S	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505R	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505S	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505R	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505S	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505R	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505S	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505R	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505S	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505R	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505S	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505R	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505S	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505R	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505S	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505R	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
	2505S	25	28	30	35	40	45	M6x10	42,4	64	32	40	10				-	-	2505	-
</td																				



Kugelgewindetriebe gewalzt Wellen-Ø 20, Steigung 5 · 10 · 20- -Präzisionskategorie C7 · C10-

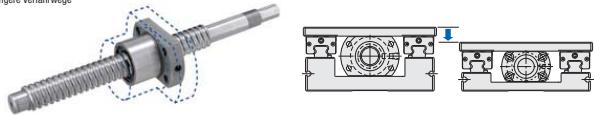
Price Reduction
Bis zu 28%

CAD-Daten



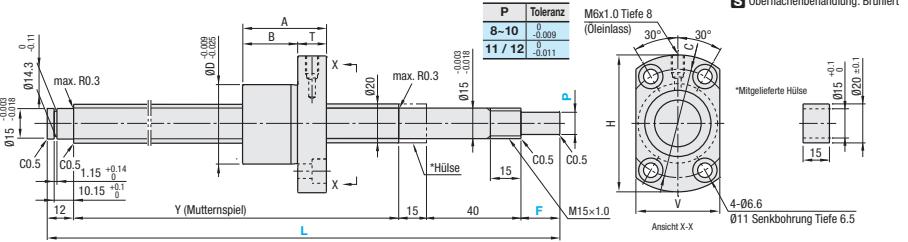
Kugelmutter-Ausführung	Standardausführung	F	P	Ausführung	Präzision Kategorie	Wellen-Ø	Steigung	Gewindewelle			Kugelmutter		
								Werkstoff	Härte	Oberflächenbehandlung	Werkstoff	Härte	Oberflächenbehandlung
Standardmutter	BSS1	BSSR	BSSZ		C7	20	5 · 10 · 20	1.1200 / C5E	Induktionsgehärtet 56-62HRC	Phosphatiert	1.726420CM6S	umgekehrt 58-62HRC	LTBC-Verchromung
Standardmutter	BSS2	BSSRK	BSSZK		C10	20	5 · 10 · 20	-	-	-	1.726420CM6S	-	-
Kompaktmutter	BSSC	BSSCK				5 · 10	-	-	-	-	1.726240CM6S	-	-

- Eigenschaften von Kompaktmuttern-
- Kompativer Außen-Ø
- Es können flachere Führungen verwendet werden.
- Langere Verfahrwege

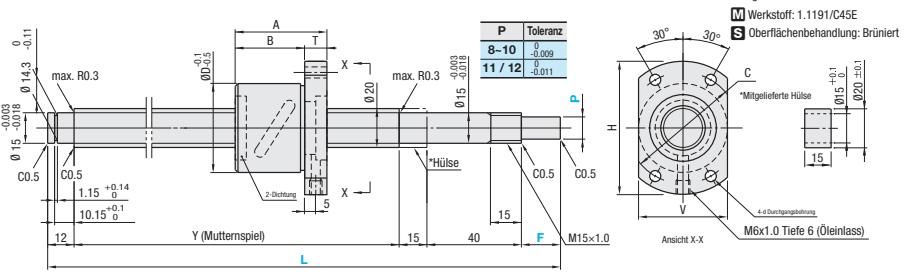


RoHS

-Kompaktmutter- BSSC (K) 2005 · 2010



-Standardmuttern- BSS1 - BSSR (K) · BSSZ (K) 2005, 2010, 2020



Kugelmutter-Ausführung	Präzision Kategorie	Ausführung	Teilenummer	1mm-Schritte								Tragzahl	Zulässiges Längenspiel	Drehrichtung								
				L	f	*P	Y	D	A	B	T	H	C	V	d							
Standard-ausführung Kugel-mutter	C7	BSS1	200-1200	05	20	12	L-87	40	40	30	10	60	50	40	4.5	20.8	(17.5)	2.5, umdr., in einer Reihe	6.2	14.7	max. 0.03	
	BSSR	BSSZ	200-2000	20-36	8-12	L-(67+F)	-	-	-	-	-	-	-	-	-	3.175	(17.25)	4 Umdr., in einer Reihe	-	-	max. 0.10	
Kompakt Kugelmutter	BSSC	BSSCK	20-36	8-12	L-(67+F)	20	12	52	57	45	12	82	67	52	6.5	4.7625	21	(16.3)	2.5, umdr., in einer Reihe	6.6	14.05	max. 0.05
Standard-ausführung Kugel-mutter	C7	BSS1	250-2000	20-36	8-12	L-(67+F)	20	12	58	55	45.5	13	62	49	38	-	20.5	(17.25)	4 Umdr., in einer Reihe	6.53	13.93	max. 0.15
Kompakt Kugelmutter	BSSC	BSSCK	20-36	8-12	L-(67+F)	20	12	39	78	68	10	62	50	39	5.5	3.175	20.8	(17.5)	2.5, umdr., in einer Reihe	6.2	14.7	max. 0.03
Standard-ausführung Kugel-mutter	C10	BSS1	250-2000	20-36	8-12	L-(67+F)	20	12	58	55	45.5	13	62	49	38	-	20.8	(17.5)	2.5, umdr., in einer Reihe	6.2	14.7	max. 0.10

* und P sind nur für BSSPK, BSSZK und BSSCK konfigurierbar. (F-Px3)

Preis
euros

Kugelmutter-Ausführung	Präzision Kategorie	Teilenummer	€ Stückpreis Stückz. 1-4					
			L200-400	L401-600	L601-800	L801-1000	L1001-1200	L1201-1500
Standardmutter	C7	BSS2005						
		BSS2005						
Kompaktmutter	C10	BSSZ2005						
		BSSZ2005						
Standardmutter	C7	BSS2010						
		BSS2010						
Kompaktmutter	C10	BSSZ2010						
		BSSZ2010						
Standardmutter	C7	BSS2020						
		BSS2020						
Kompaktmutter	C10	BSSZ2020						

Bestellbeispiel: Teilenummer - L - F - P
bsp: BSSC2005 - 400
BSSZK2020 - 320 - F30 - P12
BSS2010L - 500 (Typ L mit Schmierstoff)
BSSR2010G - 400 (Typ G mit Schmierstoff)

Lieferzeit - BSST - BSSR (K) - BSSZ (K)
8 Arbeitstage
Express A 8,00 EUR/Stk. S.88

Bei noch größeren Bestellmengen bitte gesondert anfragen.
Addieren Sie für den Preis von BS_ K (Konfigurationen F-P) 29,00 EUR zum Preis für die Standardausführung, z.B. BSSZK2010-300-F22-P8 178,30 EUR+29,00 EUR=207,30 EUR

Hinweis

Die Montagefläche für den BSST-Kugelmutterflansch ist nicht oberflächenbehandelt.

Mit lithiumverseiftem Schmierstoff gefüllt (Alvania-Fett S2 von Showa Shell Sekiyu K.K.).

Hinweise: Lassen Sie die Muttern nicht überlaufen, und trennen Sie die Muttern nicht von den Gewindewellen.

Dies kann dazu führen, dass die Kugeln herausfallen oder die Kugelrückführungsstelle beschädigt werden.

Es sind verschiedene Schmierstoffausführungen erhältlich. Zu Lieferzeit, Preisen und Spezifikationen siehe S.651 - S.653

Für Einzelheiten zu Flanschlagergehäusen siehe S.685 - S.694

Optionen Teilenummer - L - F - P - (FC · KC ... usw.)

Optionen	Opt.-Nr.	Spez.	Aufpreis
Keine Bearbeitung des Wellenendes (Loslagerseite)	NC	Ohne Bearbeitung des Wellenendes auf der Loslagerseite. Bestellnr.: NC	BSST-BSSR BSS2 -30,00 BSSC -8,00
Kugelmutter-Ausrichtung umgedreht (Loslagerseite)	RLC	Umkehrung der Mutter-Ausrichtung. Bestellnr.: RLC	Ohne Aufpreis
Keine Sicherungsringnut am Wellenende (Loslagerseite)	RNC	Ohne Sicherungsringnut am Wellenende (Loslagerseite). Bestellnr.: RNC Keine Kombination mit FC verfügbar.	Free of charge
Andere Bearbeitung des Wellenendes (Loslagerseite)	GC	Änderung der Bearbeitung auf der Loslagerseite. 0-12 · 15 G=1-mm-Schrifte Bestellnr.: GC-010-G20 G=5-mm-Durchmesser Y-Maß ist verkürzt.	BSST-BSSR BSSZ 15,00 BSSC 6,00
Andere Länge des Wellenendes (Loslagerseite)	FC	Änderung der Länge des Wellenendes auf der Loslagerseite. FC=1-mm-Schrifte FC=13-mm-F230 Y-Maß ist verkürzt.	BSST-BSSR BSSZ 7,00 BSSC 3,00
Innenrinne am Wellenende (Loslagerseite)	MC	Mit einer Rinne am Wellenende (Loslagerseite). M=1-mm-Schrifte Bestellnr.: MC20 M=1-mm-Schrifte Bestellnr.: MC15 Y-Maß ist verkürzt. M=1-mm-Schrifte Bestellnr.: MC30	BSST-BSSR BSSZ 14,00 BSSC 14,00 je 3,00

Optionen	Opt.-Nr.	Spez.	Aufpreis
Schlüssellochflächen auf Festlagerseite	SZC	Mit Schlüssellochflächen am Wellenende (Festlagerseite). Bestellnr.: SZC Die Kugellager fallen heraus, wenn die Kugelmutter mit den Schlüssellochflächen in Kontakt kommt.	BSST-BSSR BSSZ 7,00 BSSC 3,00
Nut am Wellenende (Festlagerseite)	KC	Mit einer Nut am Wellenende (Festlagerseite). KC=1-mm-Schrifte Bestellnr.: KC15 4xKCxPx3 KCx-F-1	BSST-BSSR BSSZ 14,00 BSSC 6,00
Nut am Wellenende (Festlagerseite)	KLC	Mit einer Nut in einem Kundendefinierten Bereich am Wellenende (Festlagerseite). (Maß der Festelagerdurchmesser identisch mit KC). KS=1-mm-Schrifte Bestellnr.: KLC-K7-S2 5xKSxPx3 Y-Maß ist verkürzt.	BSST-BSSR BSSZ 14,00 BSSC 6,00
Spannfläche am Wellenende (Festlagerseite)	SC	Mit einer Spannfläche am Wellenende (Festlagerseite). SC=1-mm-Schrifte Bestellnr.: SC7 5xSCxPx3 SCx-F-1	BSST-BSSR BSSZ 7,00 BSSC 3,00
2 Spannflächen am Wellenende (Festlagerseite)	SWC SGC	Mit zwei Spannflächen am Wellenende (Festlagerseite). SWC: 90°-Winkel SGC: 120°-Winkel 1mm-Schriften Bestellnr.: SWC10 5xSJ-SWC, SGC:Px3 SGC, SGC-F-1	BSST-BSSR BSSZ je 7,00 BSSC je 3,00

Kombination mit Flanschlagergehäusen

Teilenummer Kugelgewindetrieb	Empfohlene Flanschlagergehäuse						Seite
	Teilenummer	Ausführung	Nr.	Form	Festlager	Loslager	
Ausführung							S.691
BSSV	05	Quadratische Ausführung	Flach	0	0	0	S.691
BUV	15	Quadratische Ausführung	Flach	0	0	0	S.692
BRW	15	Rund	0	0	0	0	S.693
BUR	15	Rund	0	0	0	0	S.694
BSW	15	Quadratische Ausführung	Flach	0	0	0	S.687
BUN	15	Quadratische Ausführung	Flach	0	0	0	S.689

Für Flanschlagergehäuse sind noch weitere Formvarianten und Oberflächenbehandlungen erhältlich. (S.685 - S.694)

Teilenummer Kugelgewindetrieb	Empfohlene Spindelmuttern					
	Teilenummer	Ausführung	Nr.	Seite	Ausführung	Nr.
BSSC	05	Quadratische Ausführung	Flach	0	0	2005C
BSSC	10	Quadratische Ausführung	Flach	0	0	2010C
BSST	05	Quadratische Ausführung	Flach	0	0	2005R
BSSR	10	Quadratische Ausführung	Flach	0	0	2010R
BSSZ	20	Quadratische Ausführung	Flach	0	0	2020R

Spindelmuttern sind in mehreren Werkstoffvarianten und mit unterschiedlichen Oberflächenbehandlungen erhältlich. (S.695)



Linearführungen für mittlere Traglast

-Normales Spiel-

Merkmale: Die einfachste Basisvariante unter allen branchenüblichen Standardspezifikationen.



Führungswagen und -schiene können nicht separat bestellt werden. Ausführung mit normalen Spiel bietet ebenso wie die Sätze aus Führungswagen und -scheinen ein garantiertes radiales Spiel und Genauigkeit.

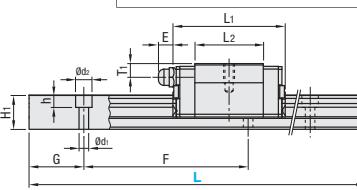
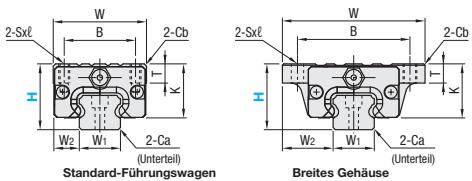
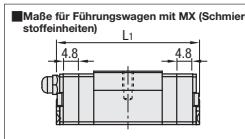
Schmiertstoffeinheiten MX Neu

Bieten langfristigen wartungsfreien Betrieb.



Standard- wert	Ausführung		MX (Schmiert- stoffeinheiten)	L	Werkstoff Härte
	1 Führungswagen	2 Führungswagen			
SVR	SVJR	SV2R	SV2RJ		Frei wählbar
SVRL	SVJRL	SV2RL	SV2RLJ		Wählbar
SVW	-	SV2W	-		Frei wählbar
SWL	-	SV2WL	-		Wählbar
SVWT	-	SV2WT	-		Frei wählbar
SWTL	-	SV2WTL	-		Wählbar

Wärmebeständigkeit: -20...80°C



Vorsichtsmaßnahmen

Führungswagen sind mit Kugelfäden ausgerüstet, um zu verhindern, dass Kugeln herausfallen.

Siehe [S.501](#)

Radiales Spiel und Genauigkeit werden nicht garantiert, wenn die Führungswagen und -scheinen aus der Kombination des Originates heraus ausgetauscht werden.

Auf den Bezugslinien sind gerade Nuten eingebracht. Stellen Sie sicher, dass beim Zusammenstellen die Bezugslinien passen.

Die Schienen können nicht zur Verlängerung verbunden werden.

Laufparallelität ist der Wert, der nach Montage der Führungsschiene gemessen wird.

Andere

Für das Maß G bei der Ausführung mit konfigurierbarem Maß L, siehe [S.507](#)

Gefüllt mit lithiumverfestigtem Schmiertstoff (Alvania-Fett S2 von Showa Shell Sekiyu K.K.).

Schmiertopf: gerade Ausführung für H24 und abgewinkelt für H28 und H33.

Teilenummer	Ausführung	MX	H	Führungswagenmaße						Führungsschiene maße														
				W ±0,05	L1 Standardausf.	MX	B	Sxℓ	L2	K	T	CB	Schmiernippel	Befestigungsbohrungen	E	T1	H1 ±0,005	W1	W2	Ca	Senkbohrungen	d1xd2xh	F	G
(I Führungs- wagen) (2 Führungs- wagen)	(I Führungs- schiene) (2 Führungs- schiene)	SVR	SV2R	24	34	41	50,6	26	M4x7	25	20	7	0,85	M5xP0,8	6	5	12,5	15	9,5	0,5	3,5x6x4,5	60	20	
				28	42	47	56,6	32	M5x8	27,6	22,5	7,5	1	M6xP0,75	13	6	15,5	20	11	0,6	6x9,5x8,5	60	20	
				33	48	59	68,6	35	M6x9	37	26,5	8	1	M6xP0,75	13	6,8	18	23	12,5	0,8	7x11x9	60	20	
		SVRL		24	52	41	50,6	41	4,5 (M5)	25	20	7	0,5	M5xP0,8	6	5	12,5	15	18,5	0,5	3,5x6x4,5	60	20	
				28	59	47	56,6	49	5,5 (M6)	27,6	22,5	9	1	M6xP0,75	13	6	15,5	20	19,5	0,6	6x9,5x8,5	60	20	
				33	73	59	68,6	60	7 (M8)	37	26,5	10	1	M6xP0,75	13	6,8	18	23	25	0,8	7x11x9	60	20	

(*) Maße in () gelten für Innengewindebohrungen der Ausführung mit breitem Führungswagen.

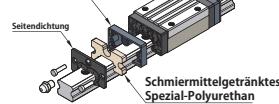
kgf=Nx0,101972

H	Tragzahl (Dynamisch) (kN)	C (Statisch) (kN)	Zulässiges statisches Moment Mx/Ms Nm + m	Mc Nm	Gewicht	Führungswagen kg		Führungsschiene kg/m	
						Befestigungs- ausführung	Breit	Breite W2 Toleranz	Höhe H Toleranz
24	5,0	8,23	33	57	0,15	0,20	1,5		
28	7,2	12,1	58	135	0,20	0,25	2,4		
33	11,7	19,6	109	225	0,30	0,40	3,4		



Schmiertstoffeinheiten MX

Schutzgehäuse



-Vorteile der Schmiertstoffeinheit MX
Langfristiger wartungsfreier Betrieb
Geringere Instandhaltungskosten
Besonders zu empfehlen an unzweckgängigen Schmierstellen.
Weitere Informationen finden Sie auf [S.498](#)

Standards für Vorlast und Genauigkeit		Ausführung mit normalem Spiel	
Radiales Spiel (µm)		Maßgenauigkeit (µm)	
H24	-4/+2	Höhe H Toleranz	±100 ±40
H28	-5/+2	Höhe H paarweise Änderung	20 15
H33	-6/+3	Breite W2 Toleranz	±100 ±20
		Breite W2 variiert paarweise	20 15
			H24, 28 H33 30
		Laufparallelität der Oberfläche C zur Oberfläche A	Siehe S.501
		Laufparallelität der Oberfläche D zur Oberfläche B	

Artikel für Linearführung			
Positionierungshilfen	Führungsschienen Distanz- blöcke	Führungswagen Abflasch- satz/Schutz	Befestigungsteile Wagen/Schienen
S.581 - S.588			

Price Reduction
Up to 12%

CAD-Daten



Teilenummer

- L

SVR28 - 880

(Mit Schmiertstoffeinheiten)

RVR28

880 (LTB)

SVR28L

880 (Typ G mit Schmiertstoff)

SVR28G

880 (Typ G mit Schmiertstoff)

Alternativ Schmiertstoffausführungen verfügbar (außer für Führungswagen mit Schmiertstoffeinheiten).

Für Lieferzeit, Preis und Leistung, siehe [S.507](#)



8 Arbeits-
tage

Express A

8,00 EUR/
Stück

S.88

Expressgebühr von 21,60 EUR für 3 oder mehr identische Teile.



Preis

europ

€ 8,87

8,00



Teilenummer

- L

SVR28 - 880

(TMS, TMC etc.)

Keine Express-Lieferung für TMS, TMC und WC.

Bestellbeispiel

Alternativ Schmiertstoffausführungen verfügbar (außer für Führungswagen mit Schmiertstoffeinheiten).

Für Lieferzeit, Preis und Leistung, siehe [S.507](#)

Expressgebühr von 21,60 EUR für 3 oder mehr identische Teile.

Bestellbeispiel

Alternativ Schmiertstoffausführungen verfügbar (außer für Führungswagen mit Schmiertstoffeinheiten).

Für Lieferzeit, Preis und Leistung, siehe [S.507](#)

Expressgebühr von 21,60 EUR für 3 oder mehr identische Teile.

Bestellbeispiel

Alternativ Schmiertstoffausführungen verfügbar (außer für Führungswagen mit Schmiertstoffeinheiten).

Für Lieferzeit, Preis und Leistung, siehe [S.507](#)

Expressgebühr von 21,60 EUR für 3 oder mehr identische Teile.

Bestellbeispiel

Alternativ Schmiertstoffausführungen verfügbar (außer für Führungswagen mit Schmiertstoffeinheiten).

Für Lieferzeit, Preis und Leistung, siehe [S.507](#)

Expressgebühr von 21,60 EUR für 3 oder mehr identische Teile.

Bestellbeispiel

Alternativ Schmiertstoffausführungen verfügbar (außer für Führungswagen mit Schmiertstoffeinheiten).

Für Lieferzeit, Preis und Leistung, siehe [S.507](#)

Expressgebühr von 21,60 EUR für 3 oder mehr identische Teile.

Bestellbeispiel

Alternativ Schmiertstoffausführungen verfügbar (außer für Führungswagen mit Schmiertstoffeinheiten).

Für Lieferzeit, Preis und Leistung, siehe [S.507](#)

Expressgebühr von 21,60 EUR für 3 oder mehr identische Teile.

Bestellbeispiel

Alternativ Schmiertstoffausführungen verfügbar (außer für Führungswagen mit Schmiertstoffeinheiten).

Für Lieferzeit, Preis und Leistung, siehe [S.507](#)

Expressgebühr von 21,60 EUR für 3 oder mehr identische Teile.

Bestellbeispiel

Alternativ Schmiertstoffausführungen verfügbar (außer für Führungswagen mit Schmiertstoffeinheiten).

Für Lieferzeit, Preis und Leistung, siehe [S.507](#)

Expressgebühr von 21,60 EUR für 3 oder mehr identische Teile.

Bestellbeispiel

Alternativ Schmiertstoffausführungen verfügbar (außer für Führungswagen mit Schmiertstoffeinheiten).

Für Lieferzeit, Preis und Leistung, siehe [S.507](#)

Expressgebühr von 21,60 EUR für 3 oder mehr identische Teile.

Bestellbeispiel

Alternativ Schmiertstoffausführungen verfügbar (außer für Führungswagen mit Schmiertstoffeinheiten).

Für Lieferzeit, Preis und Leistung, siehe [S.507](#)

Expressgebühr von 21,60 EUR für 3 oder mehr identische Teile.

Bestellbeispiel

Alternativ Schmiertstoffausführungen verfügbar (außer für Führungswagen mit Schmiertstoffeinheiten).

Für Lieferzeit, Preis und Leistung, siehe [S.507](#)

Expressgebühr von 21,60 EUR für 3 oder mehr identische Teile.

Bestellbeispiel

Alternativ Schmiertstoffausführungen verfügbar (außer für Führungswagen mit Schmiertstoffeinheiten).

Für Lieferzeit, Preis und Leistung, siehe [S.507](#)

Expressgebühr von 21,60 EUR für 3 oder mehr identische Teile.

Bestellbeispiel

Alternativ Schmiertstoffausführungen verfügbar (außer für Führungswagen mit Schmiertstoffeinheiten).

Für Lieferzeit, Preis und Leistung, siehe [S.507](#)

Expressgebühr von 21,60 EUR für 3 oder mehr identische Teile.

Bestellbeispiel

Alternativ Schmiertstoffausführungen verfügbar (außer für Führungswagen mit Schmiertstoffeinheiten).

Für Lieferzeit, Preis und Leistung, siehe [S.507](#)

Expressgebühr von 21,60 EUR für 3 oder mehr identische Teile.

Bestellbeispiel

Alternativ Schmiertstoffausführungen verfügbar (außer für Führungswagen mit Schmiertstoffeinheiten).

Für Lieferzeit, Preis und Leistung, siehe [S.507](#)

Expressgebühr von 21,60 EUR für 3 oder mehr identische Teile.

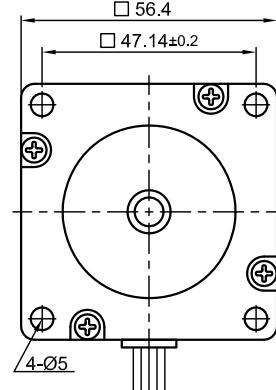
Bestellbeispiel

Alternativ Schmiertstoffausführungen verfügbar (außer für Führungswagen mit Schmiertstoffeinheiten).

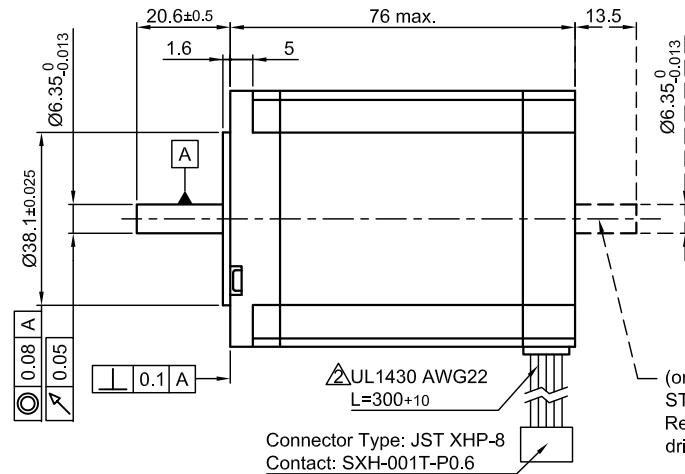
Für Lieferzeit, Preis und Leistung, siehe [S.507](#)

Expressgebühr von 21,60 EUR für 3 oder mehr identische Teile.

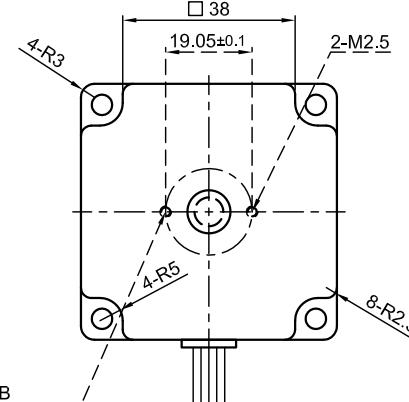
Front view and mounting



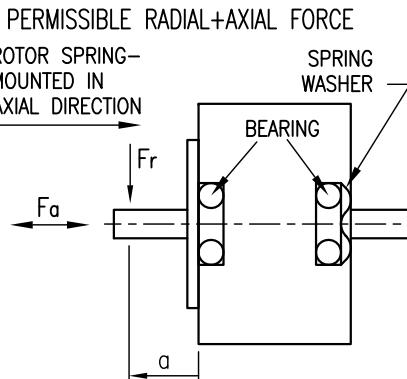
Side view



Rear view



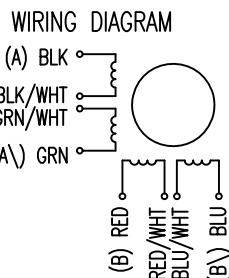
SPECIFICATION	CONNECTION	UNIPOLAR OR BIPOLAR-1 WINDING	BIPOLAR		PERMISSIBLE RADIAL+AXIAL FORCE ROTOR SPRING-MOUNTED IN AXIAL DIRECTION
			SERIAL	PARALLEL	
VOLTAGE (VDC)		8.8			
AMPS/PHASE		1.0	0.71	1.41	
RESISTANCE/PHASE (Ohms)@25°C		8.8±10%	17.6±10%	4.4±10%	
INDUCTANCE/PHASE (mH) @1KHz		19±20%	76±20%	19±20%	
HOLDING TORQUE (Nm) [lb-in]		1.32 [11.71] △	1.87 [16.52] △	1.87 [16.52] △	
DETENT TORQUE (Nm) [lb-in]		0.068 [0.602]			
STEP ANGLE (°) ± ACCURACY		1.8±5% (NON-ACCUM)			
ROTOR INERTIA (Kg-m²) [lb-in²]		4.8x10⁻⁵ [0.164]			
WEIGHT (Kg) [lb]		1.0 [2.2]			
TEMPERATURE RISE: MAX.80°C (MOTOR STANDSTILL; FOR 2 PHASE ENERGIZED)					AXIAL-FORCE F_a (N) $F_a=15$
AMBIENT TEMPERATURE -10°~ 50°C [14°F ~ 122°F]					DISTANCE a (mm) 5 10 15 20
INSULATION RESISTANCE 100 MOhm (UNDER NORMAL TEMPERATURE AND HUMIDITY)					RADIAL-FORCE F_r (N) 130 90 70 52
INSULATION CLASS B 130° [266°F]					AXIAL RADIAL
DIELECTRIC STRENGTH 500VAC FOR 1 MIN. (BETWEEN THE MOTOR COILS AND THE MOTOR CASE)					SHAFT PLAY (mm) 0.08 0.02
AMBIENT HUMIDITY MAX. 85% (NO CONDENSATION)					AT LOAD MAX: (N) 4.5 4.5



UNIPOLAR	TYPE OF CONNECTION (EXTERN)			MOTOR		
	BIPOLAR 1WINDING	BIPOLAR SERIAL	BIPOLAR PARALLEL	CONNECTOR PIN NO.△	LEADS	WINDING
A — A —	A —	A —	A —	1	BLK	A
COM —	A —	—	—	3	BLK/WHT	
A\ —	A\ —	A\ —	A\ —	2	GRN/WHT	
B — B —	B —	B —	B —	4	GRN	A\
COM —	—	—	—	5	RED	B
B\ —	B\ —	B\ —	B\ —	7	RED/WHT	
—	—	—	—	6	BLU/WHT	
—	—	—	—	8	BLU	B\

FULL STEP 2 PHASE-Ex.,
WHEN FACING MOUNTING END (X)

STEP	A	B	A\	B\	CCW	CW
1	+	+	-	-		
2	-	+	+	-		
3	-	-	+	+		
4	+	-	-	+		



Nanotec®
PLUG & DRIVE
ST5918L1008

REV	DESCRIPTION	DATE	APVD
3	NEW VALUE OF HOLD. TOR.	04.11.13.	J.D.
2	UL NO.	17.07.09.	J.W.
1	PIN-ASSIGNMENT	04.01.08	J.W.

SCALE	FREE	APVD	S.Ha.	19.03.07
X	±0.5	CHKD		
1PL	±0.2			
2PL	±0.1	DRN	J.W.	21.11.06
ANGLE	±30'			
		SIGNATURE		DATE

STEPPING MOTOR
DWG.NO. ST5918L1008

B. Anhang Elektronik

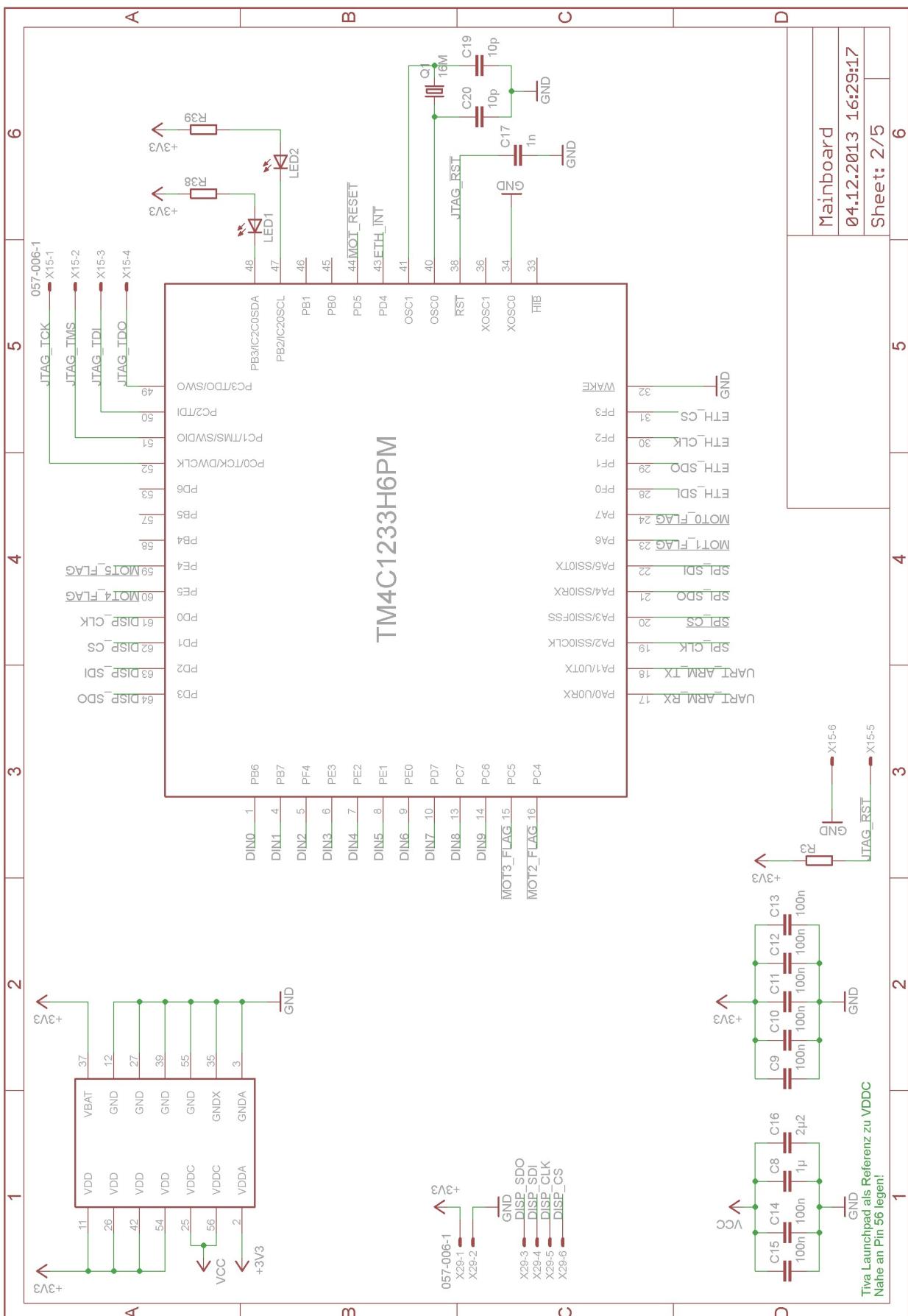


Abbildung 10: Schemazeichnung des Mikrocontrollers mit Pinbelegung, JTAG Header und Takterzeugung.

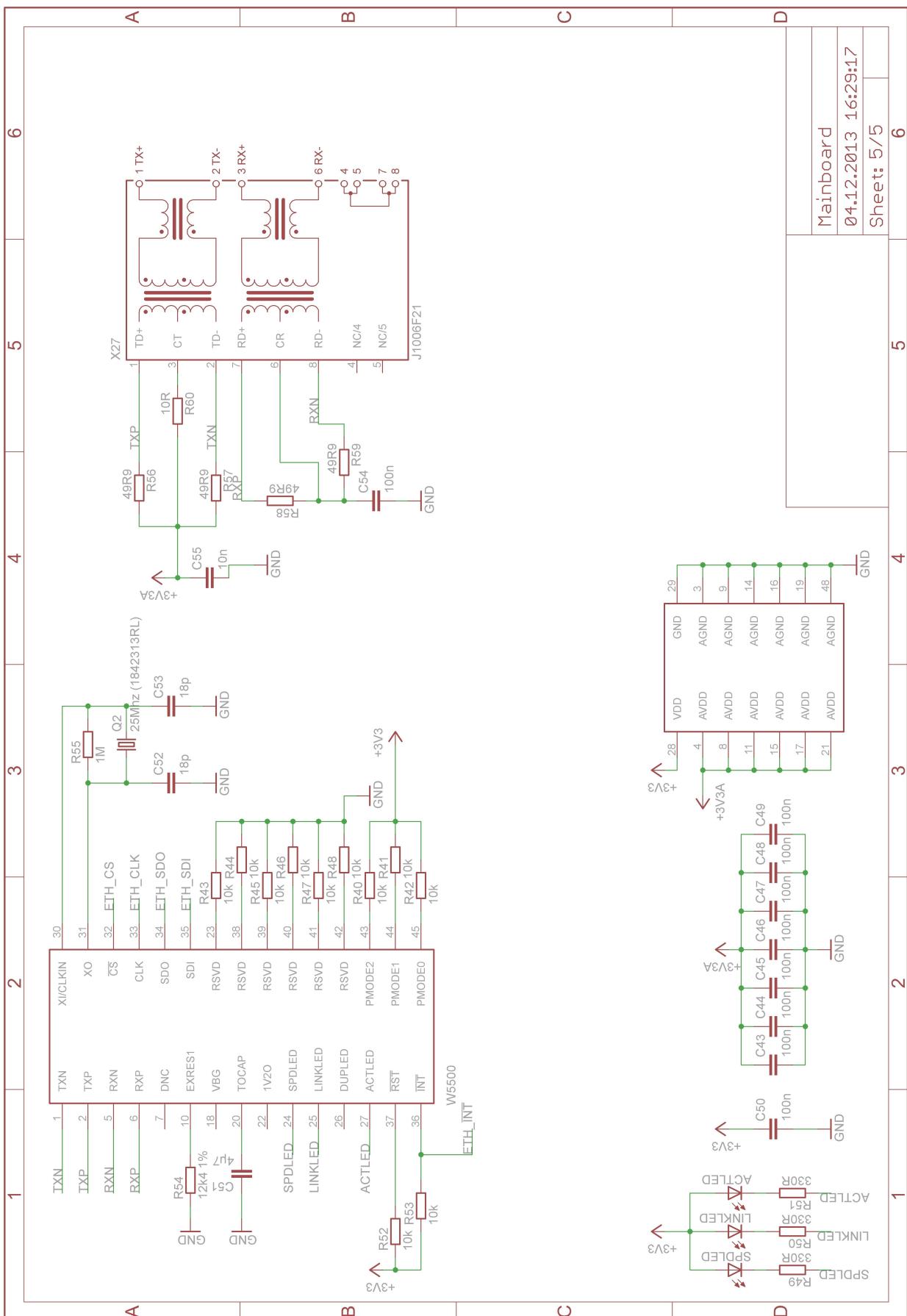


Abbildung 11: Schemazeichnung der Netzwerkanbindung mit Koppeltrafo

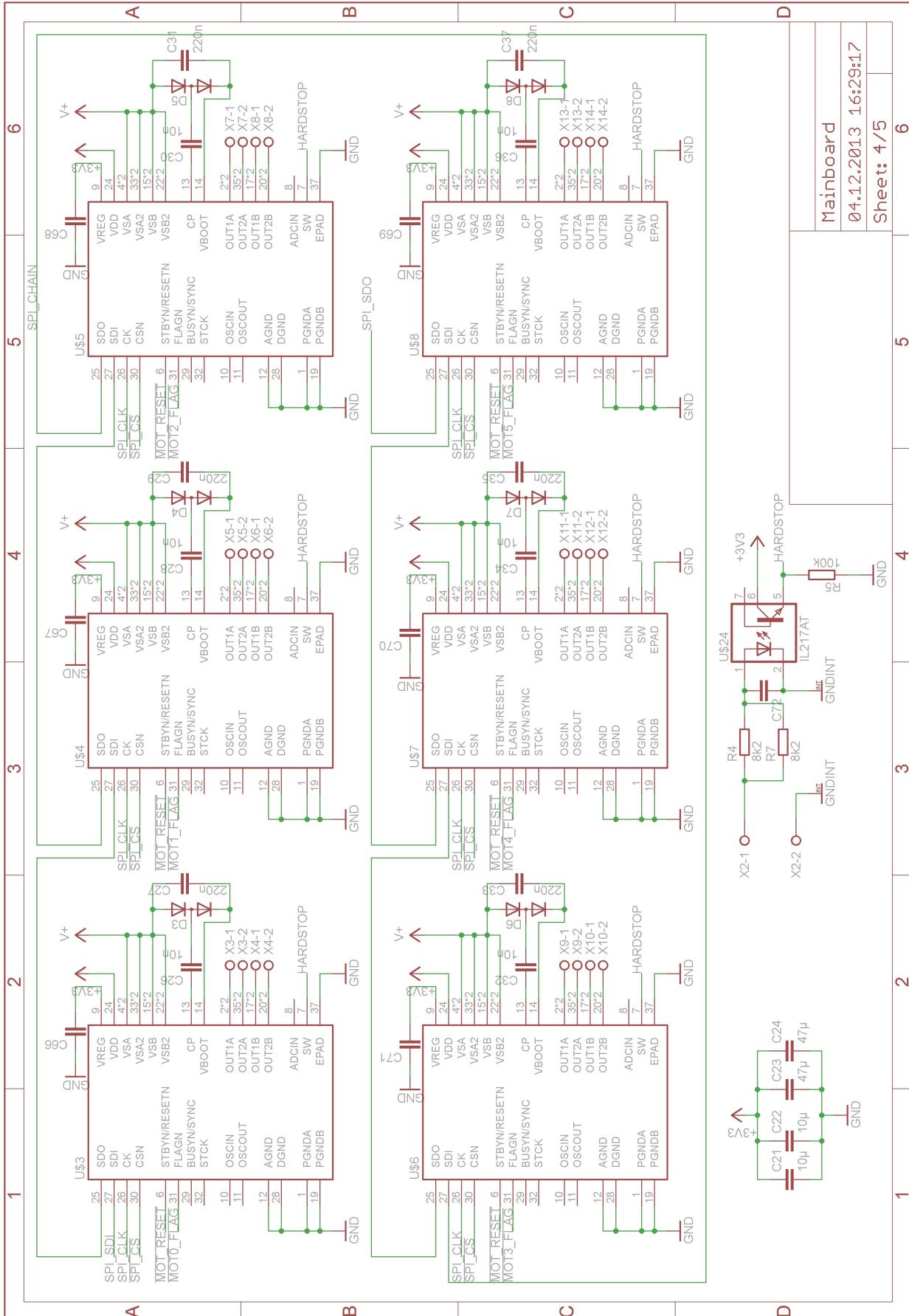


Abbildung 12: Schemazeichnung der Motortreiber in der Daisychain

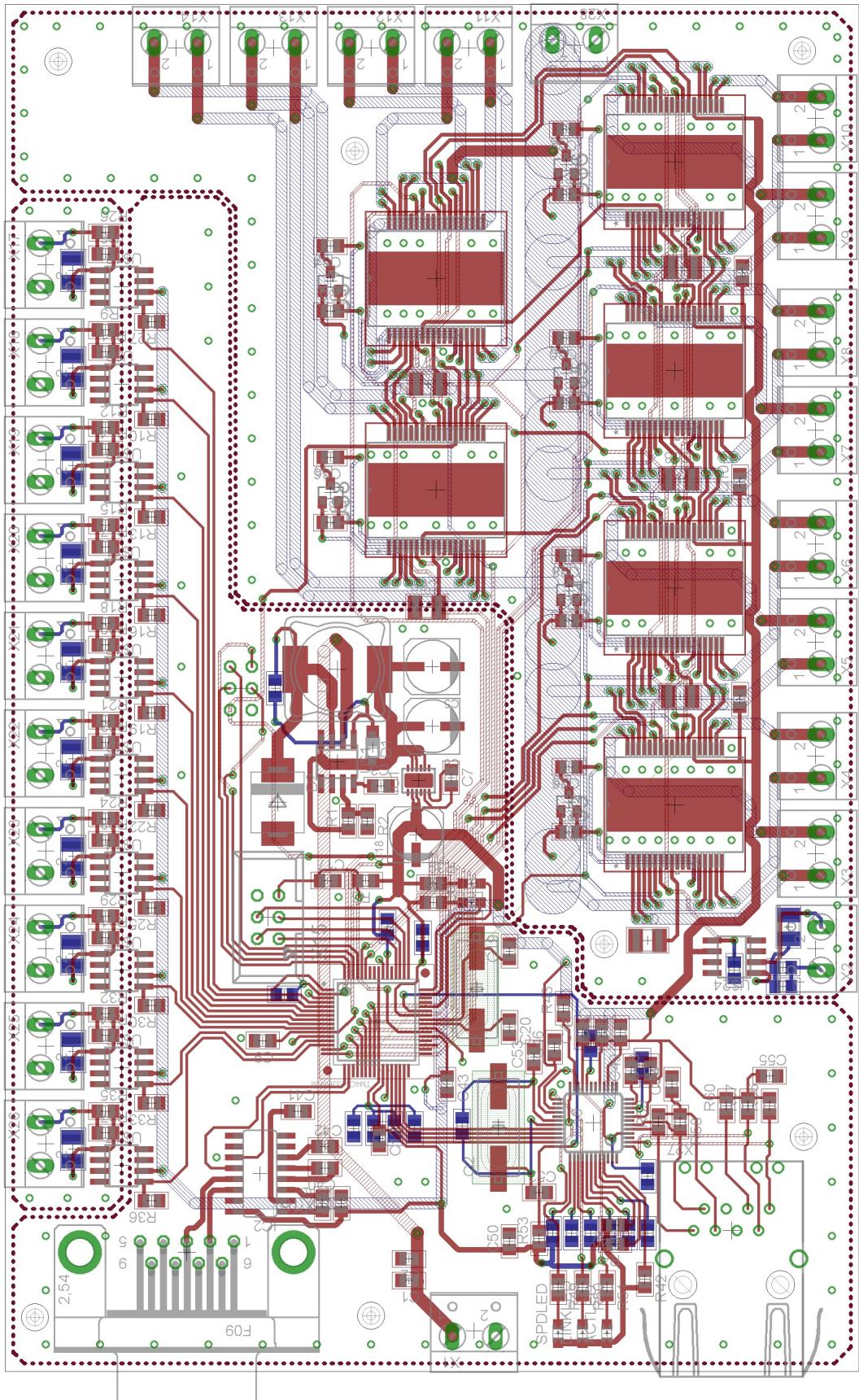


Abbildung 13: Schemazeichnung der Motortreiber in der Daisychain

C. Anhang Software

Im folgenden Kapitel ist die von uns geschriebene Software dokumentiert.

C.1. main.c

```
/*
 * main.c
 *
 * Created on: 19.10.2013
 *      Author: einball
 */
#ifndef 1 // ausblenden der #include befehle, zum verbessern des "workflows"
#define TARGET_IS_BLIZZARD_RA2
#define TM4C123GH6PM
#include <stdint.h>
#include <stdlib.h>
#include <stdbool.h>

#include <driverlib/interrupt.h>
#include <driverlib/sysctl.h>
#include <driverlib/uart.h>
#include <driverlib/ssi.h>
#include <driverlib/gpio.h>
#include <driverlib/watchdog.h>
#include <driverlib/timer.h>
#include <driverlib/rom.h>
#include <driverlib/rom_map.h>

#include <hardware/hw_nvic.h>
#include <hardware/hw_ints.h>
#include <hardware/hw_sysctl.h>
#include <hardware/hw_uart.h>
#include <hardware/hw_ssi.h>
#include <hardware/hw_gpio.h>
#include <hardware/hw_watchdog.h>
#include <hardware/hw_timer.h>
#include <hardware/hw_memmap.h>

#include "main.h"
#include "inc_def.h"           // Our own defines
#include "inc_startup.h"        // Startup Functions
#include "inc_uart.h"          // UART Handler functions
#include "inc_dspin.h"          // SPI Handler functions
#include "inc_lcd.h"            // LCD handler functions
#include "inc_msg.h"            // Protocol support
#include "inc_circular.h"        // Circular Buffer
#include "inc_motor.h"          // Motor handler functions

#endif
```

```

/* Global definierte Ringpuffer, da wir sie sowohl in der Main als auch im ISR-
   Handler brauchen (keine uebergabewerte *)
Circ_Buf Ringpuffer_RX;
Circ_Buf Ringpuffer_TX;

// //////////////////////////////////////////////// UART ISR /////////////////////////////////
// //////////////////////////////////////////////// UART ISR /////////////////////////////////
// //////////////////////////////////////////////// UART ISR /////////////////////////////////
void UARTIntHandler(void){
    uint32_t ui32Status;
    uint8_t c;

    ui32Status = UARTIntStatus(UART0_BASE, true);
    UARTIntClear(UART0_BASE, ui32Status);

    if((ui32Status & UART_INT_TX) || (eState.SER & SER_TX_REQUEST)){           ////
        auf Transmit Interrupt abfragen

        c = getChar(&Ringpuffer_TX);
        while(c != EXIT_BUFFER_EMPTY){                                         //prüfen auf
            fehlerbedingung
            UARTCharPut(UART0_BASE, c);                                       //Zeichen in
            FIFO schieben
            c = getChar(&Ringpuffer_TX);
        }
        ui32Status ^= UART_INT_TX;
        eState.SER ^= SER_TX_REQUEST;
        GOTO_STATE(STATE_IDLE);
        //Finished with transmission
    }

    if((ui32Status & UART_INT_RX) || (ui32Status & UART_INT_RT)){
        while(UARTCharsAvail(UART0_BASE)){
            c = UARTCharGet(UART0_BASE);
            if(putChar(&Ringpuffer_RX, c) == EXIT_BUFFER_FULL){

                //We have had some kind of undetected buffer overflow
                eState.SER |= SER_RX_BUFFER_OVERRUN;
                GOTO_STATE(STATE_ERROR);
            }
        }
    }
}

```

```

                //In the error state we will have to scan for
                //remaining messages
                //and flush the buffer then!

            }

            if(c == '\n'){
                //We have a valid message! Do something about it!
                eState.SER |= SER_RX_MESSAGE_COMPLETE;
                GOTO_STATE(STATE_UART_COM);
            }
        }

/* Empfang Vollstaendig - softwareflags Loeschen */
ui32Status &= !UART_INT_RX;
ui32Status &= !UART_INT_RT;
}

if(ui32Status & UART_INT_OE){
    ui32Status ^= UART_INT_OE;
    eState.SER |= SER_RX_ERROR;
}

if(ui32Status & UART_INT_BE){
    ui32Status ^= UART_INT_BE;
    eState.SER |= SER_RX_ERROR;
}

if(ui32Status & UART_INT_PE){
    ui32Status ^= UART_INT_PE;
    eState.SER |= SER_RX_ERROR;
}

if(ui32Status & UART_INT_FE){
    ui32Status ^= UART_INT_FE;
    eState.SER |= SER_RX_ERROR;
}

if(ui32Status != 0){
    GOTO_STATE(STATE_ERROR);
}
}

/* Init_Transmission Startet den Sendevorgang ueber die UART-Schnittstelle
 * schnelle Doerte Loesung, aber es funktioniert
 */
void Init_Transmission(Circ_Buf *Buf){
    uint8_t c;
    c = getChar(Buf);
    if(c != EXIT_BUFFER_EMPTY)

```

```

        UARTCharPut(UART0_BASE, c);

    /* das programm laeuft einmal "zu oft" in Init_Transmission - das nutzen wir hier
       aus, um den Idle-State aufzurufen */
    else GOTO_STATE(STATE_IDLE);
}

/* insertString dient als Wrapper-Funktion um einen String in send[128] einzufuegen
 * die if-abfrage zu Beginn ueberprueft, ab wo platz zum schreiben ist
 */
uint32_t insertString(uint8_t str[128], uint8_t buf[128]){
    uint8_t i = 0;
    uint8_t ii = 0;

    while(str[ii] != 0)
        ii++;

    for(i = 0; i < 128; i++){
        *(str + ii + i) = buf[i];
    }
    return EXIT_SUCCESS;
}

/* clearString ist eine Wrapper-Funktion dessen zweck das loeschen von send[128] ist
 */
uint32_t clearString(uint8_t str[128]){
    uint8_t i = 0;
    for(i = 0; i < 128; i++){
        str[i] = 0;
    }
    return EXIT_SUCCESS;
}

///////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////// Main
///////////////////////////////////////////////////////////////////
///////////////////////////////////////////////////////////////////
int main(void) {
    uint32_t i;
    uint32_t ii=0;
    sJob JobBuffer;

```

```

sJob CurrentJob;
uint8_t MsgPuffer[128];
uint8_t send [128];
uint16_t BufSize = 256;
sJobQueue Job_Queue;
uint32_t retVal;

GOTO_STATE(STATE_STARTUP);

while(1){

    //////////////////// State Machine ///////////////////
    switch(SYS_STATE) {

        case STATE_STARTUP:
            SysCtlClockSet(SYSCTL_SYSDIV_4|SYSCTL_USE_PLL|
                           SYSCTL_XTAL_16MHZ|SYSCTL_OSC_MAIN);
            setup_GPIO();
            setup_UART();
            setup_SPI();
            setup_INT();
            setup_HOME_POS();
            setup_JOB_QUEUE(&Job_Queue);

            jobDone(&CurrentJob);
            jobDone(&JobBuffer);

            clearString(send);
            clearString(MsgPuffer);

            Ringpuffer_RX.rdptra = BUF_SIZE - 1;
            Ringpuffer_RX.wrptra = 0;
            Ringpuffer_RX.size = BUF_SIZE;

            Ringpuffer_TX.rdptra = BUF_SIZE - 1;
            Ringpuffer_TX.wrptra = 0;
            Ringpuffer_TX.size = BUF_SIZE;

            GOTO_STATE(STATE_IDLE);
            break;

        case STATE_ERROR:
            switch (eState.SER){
                case SER_RX_TIMEOUT:

                    insertString(send, (uint8_t*)" SER_RX_TIMEOUT ");
                    insertString(send, (uint8_t*)" NACK ");
                    eState.SER |= SER_TX_REQUEST;

```

```

        GOTO_STATE(STATE_UART_COM);
        eState.SER ^= SER_RX_TIMEOUT;
        clearString(send);
        for(i=0; i < 125; i++)
            *(send+i) = 0;
        break;

case SER_RX_IDLE:
    insertString(send, (uint8_t*)" SER_RX_IDLE ");
    insertString(send, (uint8_t*)" NACK ");
    putMsg(&Ringpuffer_TX, send);
    eState.SER |= SER_TX_REQUEST;
    GOTO_STATE(STATE_UART_COM);
    eState.SER ^= SER_RX_IDLE;
    clearString(send);
    break;

case SER_RX_ERROR:
    /* Dekodieren fehlgeschlagen, NACK senden */
    insertString(send, (uint8_t*)" SER_RX_ERROR ");
    insertString(send, (uint8_t*)" NACK ");
    putMsg(&Ringpuffer_TX, send);
    eState.SER |= SER_TX_REQUEST;
    GOTO_STATE(STATE_UART_COM);
    eState.SER ^= SER_RX_ERROR;
    clearString(send);
    break;

case SER_TX_ERROR:
    insertString(send, (uint8_t*)" SER_TX_ERROR ");
    insertString(send, (uint8_t*)" NACK ");
    putMsg(&Ringpuffer_TX, send);
    eState.SER |= SER_TX_REQUEST;
    GOTO_STATE(STATE_UART_COM);
    eState.SER ^= SER_TX_ERROR;
    clearString(send);
    break;

case SER_RX_BUFFER_OVERRUN:
    insertString(send, (uint8_t*)" SER_RX_BUFFER_OVERRUN
");
    insertString(send, (uint8_t*)" NACK ");
    putMsg(&Ringpuffer_TX, send);
    eState.SER |= SER_TX_REQUEST;
    GOTO_STATE(STATE_UART_COM);
    eState.SER ^= SER_RX_BUFFER_OVERRUN;
    clearString(send);
    break;

case SER_RX_BUFFER_EMPTY:

```

```

        insertString(send, (uint8_t*)" SER_RX_BUFFER_EMPTY ")
        ;
        insertString(send, (uint8_t*)" NACK ");
        putMsg(&Ringpuffer_TX, send);
        eState.SER |= SER_TX_REQUEST;
        GOTO_STATE(STATE_UART_COM);
        eState.SER ^= SER_RX_BUFFER_EMPTY;
        clearString(send);
        break;

    default:
        break;
}

switch (eState.STA) {
default:
    break;
}

switch (eState.ETH){
default:
    break;
}

switch (eState.MSG){
case MSG_LENGTH_ERROR:
    // Nachricht zu lang, bzw. Puffer voll
    insertString(send, (uint8_t*)" MSG_LENGTH_ERROR ");
    insertString(send, (uint8_t*)" NACK ");
    putMsg(&Ringpuffer_TX, send);
    eState.SER |= SER_TX_REQUEST;
    GOTO_STATE(STATE_UART_COM);
    eState.MSG ^= MSG_LENGTH_ERROR;
    clearString(send);
    break;

case MSG_INVALID:
    // Nachrichten-inhalt fehlerhaft
    insertString(send, (uint8_t*)" MSG_INVALID ");
    insertString(send, (uint8_t*)" NACK ");
    putMsg(&Ringpuffer_TX, send);
    eState.SER |= SER_TX_REQUEST;
    GOTO_STATE(STATE_UART_COM);
    eState.MSG ^= MSG_INVALID;
    clearString(send);
    break;

case MSG_ID_ERROR:
    // Nachrichten-ID fehlerhaft
    insertString(send, (uint8_t*)" MSG_ID_ERROR ");

```

```

        insertString(send, (uint8_t*)" NACK ");
        putMsg(&Ringpuffer_TX, send);
        eState.SER |= SER_TX_REQUEST;
        GOTO_STATE(STATE_UART_COM);
        eState.MSG ^= MSG_ID_ERROR;
        clearString(send);
        break;

    case MSG_CRC_ERROR:
        // CRC-Fehler
        insertString(send, (uint8_t*)" MSG_ID_ERROR ");
        insertString(send, (uint8_t*)" NACK ");
        putMsg(&Ringpuffer_TX, send);
        eState.SER |= SER_TX_REQUEST;
        GOTO_STATE(STATE_UART_COM);
        eState.MSG ^= MSG_CRC_ERROR;
        clearString(send);
        break;

    case MSG_CODE_ERROR:
        // Fehler im Coderahmen
        insertString(send, (uint8_t*)" MSG_CODE_ERROR ");
        insertString(send, (uint8_t*)" NACK ");
        putMsg(&Ringpuffer_TX, send);
        eState.SER |= SER_TX_REQUEST;
        GOTO_STATE(STATE_UART_COM);
        eState.MSG ^= MSG_CODE_ERROR;
        clearString(send);
        break;

    case MSG_UNKNOWN_ERROR:
        // Unbekannter Fehler
        insertString(send, (uint8_t*)" MSG_UNKNOWN_ERROR ");
        insertString(send, (uint8_t*)" NACK ");
        putMsg(&Ringpuffer_TX, send);
        eState.SER |= SER_TX_REQUEST;
        GOTO_STATE(STATE_UART_COM);
        eState.MSG ^= MSG_UNKNOWN_ERROR;
        clearString(send);
        break;

    default:
        break;
}

switch (eState.MOT){
default:

    break;
}

```

```

switch (eState.JOB){

case JOB_ADD_ERROR:

    insertString(send, (uint8_t*)" JOB_ADD_ERROR ");
    insertString(send, (uint8_t*)" NACK ");
    putMsg(&Ringpuffer_TX, send);
    eState.JOB ^= JOB_ADD_ERROR;
    eState.SER |= SER_TX_REQUEST;
    GOTO_STATE(STATE_UART_COM);
    clearString(send);
    break;

case JOB_UNKNOWN_ERROR:
    insertString(send, (uint8_t*)" JOB_UNKNOWN_ERROR ");
    insertString(send, (uint8_t*)JobBuffer.id);
    insertString(send, (uint8_t*)" NACK ");
    putMsg(&Ringpuffer_TX, send);
    eState.JOB ^= JOB_UNKNOWN_ERROR;
    eState.SER |= SER_TX_REQUEST;
    GOTO_STATE(STATE_UART_COM);
    jobDone(&JobBuffer);
    clearString(send);
    break;

default:
    break;
}

break;

case STATE_JOB_HANDLING:

    if(eState.JOB & JOB_ADD_PENDING){
/* Job zur Warteschlange hinzufuegen und auf Erfolg ueberpruefen,
 * und Abarbeitung bzw. Fehlerbehandlung
 * einleiten
 */
    retVal = addJob(&Job_Queue, &JobBuffer);

    if(retVal == EXIT_SUCCESS){
        insertString(send, (uint8_t*)" JOB_ADD_SUCCESS ");
        insertString(send, (uint8_t*)JobBuffer.id);
        putMsg(&Ringpuffer_TX, send);
        clearString(send);
        jobDone(&JobBuffer);
        eState.SER |= SER_TX_REQUEST;
        GOTO_STATE(STATE_UART_COM);
    }
}

```

```

        else{
            eState.JOB |= JOB_ADD_ERROR;
            GOTO_STATE(STATE_ERROR);
        }
        eState.JOB ^= JOB_ADD_PENDING;
    }
    else GOTO_STATE(STATE_IDLE);

    break;

case STATE_MSG_HANDLING:

    if(eState.SER & SER_RX_DECODE_VALID_MSG){
        retVal = decodeMsg(&Job_Queue, &JobBuffer,
                           MsgPuffer);

        clearString(MsgPuffer);
        eState.SER ^= SER_RX_DECODE_VALID_MSG;

        switch (retVal){
        case MSG_ALL_OK:
            eState.JOB |= JOB_ADD_PENDING;
            GOTO_STATE(STATE_JOB_HANDLING);
            break;

        case MSG_INVALID:
            eState.MSG |= MSG_INVALID;
            GOTO_STATE(STATE_ERROR);
            break;

        case MSG_CRC_ERROR:
            eState.MSG |= MSG_CRC_ERROR;
            GOTO_STATE(STATE_ERROR);
            break;

        case MSG_ID_ERROR:
            eState.MSG |= MSG_ID_ERROR;
            GOTO_STATE(STATE_ERROR);
            break;

        case MSG_CODE_ERROR:
            eState.MSG |= MSG_CODE_ERROR;
            GOTO_STATE(STATE_ERROR);
            break;

        default:
            eState.MSG |= MSG_UNKNOWN_ERROR;
            GOTO_STATE(STATE_ERROR);
            break;
    }
}

```

```

        }

        else GOTO_STATE(STATE_IDLE);
        retVal=0;
        break;

case STATE_UART_COM:

    if(eState.SER & SER_RX_MESSAGE_COMPLETE){
        retVal = getMsg(&Ringpuffer_RX, MsgPuffer,
                        BufSize);

        switch(retVal){
            case EXIT_SUCCESS:
                GOTO_STATE(STATE_MSG_HANDLING);
                eState.SER |= SER_RX_DECODE_VALID_MSG
                ;
                break;

            case EXIT_BUFFER_SIZE:
                eState.SER |= MSG_LENGTH_ERROR;
                GOTO_STATE(STATE_ERROR);
                break;

            case EXIT_BUFFER_EMPTY:
                eState.SER |= SER_RX_BUFFER_EMPTY;
                GOTO_STATE(STATE_ERROR);
                break;

            default:
                eState.SER |= SER_UNDEFINED_ERROR;
                GOTO_STATE(STATE_ERROR);
                break;
        }

        eState.SER ^= SER_RX_MESSAGE_COMPLETE;           // Flag Loeschen
    }

    if(eState.SER & SER_TX_REQUEST){
        //Init_Transmission startet das Senden der
        //UART schnittstelle
        Init_Transmission(&Ringpuffer_TX);
    }
    retVal = 0;
    break;

case STATE_MOT_COM:

/* Je Nach opCode wird eine entsprechende aktion ausgefuehrt */
switch (CurrentJob.opCode){


```

```

        case (OPC_AUSGABE || OPC_NACHFUELLEN):
            /* Ein- und Ausgabe - "Normalbetrieb" */
            Ctrl_Motor(&CurrentJob);
            break;

        case OPC_FERNSTEUERN:
            /* Fuer "Manuelle" Steuerung der Motoren:
             * "DEMO"/Test -Modus */
            Remote_Ctrl_Motor(&CurrentJob);
            break;

        case OPC_REKALIBRIEREN:
            /* Fuer die aenderung der Kaestchenbreite, derzeit noch ueber
             defines geloest */
            break;

        default:
            /* Irgendwas ist schiefgelaufen: id, fehlercode, NACK senden, Job
             loeschen - Fehlersuche einbauen? */
            eState.JOB |= JOB_UNKNOWN_ERROR;
            GOTO_STATE(STATE_ERROR);
            break;
    }

    /* Nachricht an die hoherstehende Logik, dass Job Ausgefuehrt wurde
     * hierzu uebertragen wir die ID des Jobs und enden mit ACK, bzw NACK
     * letzteres ist allerdings noch nicht Implementiert fuer die
     Motorsteuerung
     */
    if(eState.JOB ^ JOB_UNKNOWN_ERROR){
        insertString(send, (uint8_t*)" JOB_DONE ");
        insertString(send, (uint8_t*)CurrentJob.id);
        insertString(send, (uint8_t*)" ACK ");
        putMsg(&Ringpuffer_TX, send);
        eState.SER |= SER_TX_REQUEST;
        clearString(send);

        GOTO_STATE(STATE_UART_COM);
    }
    else{
        insertString(send, (uint8_t*)" 
JOB_UNKNOWN_ERROR ");
        insertString(send, (uint8_t*)CurrentJob.id);
        insertString(send, (uint8_t*)" NACK");
        putMsg(&Ringpuffer_TX, send);
        eState.SER |= SER_TX_REQUEST;

        clearString(send);
    }
}

```

```

                GOTO_STATE(STATE_UART_COM);
}
jobDone(&CurrentJob);
break;

case STATE_NET_COM:
    break;

case STATE_IDLE:
    // *Queue hat irgendein schrott drin stehn -
    warum?
    retVal = callJob(&Job_Queue, &CurrentJob);

    switch (retVal){
        case EXIT_SUCCESS:
            GOTO_STATE(STATE_MOT_COM);
            break;

        case JOB_PENDING:
/* eventuelle Problembehandlung hier einfuegen ??? */
            GOTO_STATE(STATE_MOT_COM);
            break;

        /*      case EXIT_FAILURE:
            Warteschlange ist leer, ERROR-state
            aufrufen und und
            * Nachsehen, ob noch Errorflags auf
            abhandlung warten

            GOTO_STATE(STATE_ERROR);
            break; */

        default:
            ii++;
            if(ii >= 150000){

                insertString(send, (uint8_t*)
                    " IDLE ");
                putMsg(&Ringpuffer_TX, send);
                eState.SER |= SER_RX_REQUEST;
                clearString(send);
                ii = 0;
                GOTO_STATE(STATE_UART_COM);
            }
            retVal = 0;
            break;
    }
}

```

```

        //          GPIOPinWrite(GPIO_PORTF_BASE
        , GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_3,
        GPIO_PIN_1);
        //          SysCtlDelay(160000);
        //          GPIOPinWrite(GPIO_PORTF_BASE
        , GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_3,
        GPIO_PIN_2);
        //          SysCtlDelay(160000);

                break;
}
}
}

```

C.2. main.h

```

/*
 * main.h
 *
 * Created on: 28.10.2013
 *      Author: einball
 */

#ifndef MAIN_H_
#define MAIN_H_

#include <stdint.h>
#include <stdlib.h>
#include <stdbool.h>

/* System states */
typedef enum {
    STATE_STARTUP ,
    STATE_ERROR ,
    STATE_MSG_HANDLING ,
    STATE_UART_COM ,
    STATE_MOT_COM ,
    STATE_NET_COM ,
    STATE_LCD_COM ,
    STATE_JOB_HANDLING ,
    STATE_IDLE
} sysState_t;

sysState_t SYS_STATE;

```

```

/* Error States */
typedef struct sError {

    uint32_t      SER;
    uint32_t      STA;
    uint32_t      ETH;
    uint32_t      MSG;
    uint32_t      MOT;
    uint32_t      JOB;
} sErrorFlags;

sErrorFlags eState;

/* Prototypes */
void GOTO_STATE(sysState_t state);

#endif /* MAIN_H_ */

```

C.3. circular.c

```

/*
 * circular.c
 *
 * Created on: 21.10.2013
 *      Author: einball
 */

#include "inc_circular.h"

/*
 * getChar gets the char that's in the Buffer
 * and advances the read pointer. The function
 * returns EXIT_FAILURE if there are no more chars
 * to get or the character in a 32 bit signed int.
 */
uint8_t getChar(Circ_Buf *Buf){

    if((Buf->rdptr + 1) == Buf->wrptr){
        return EXIT_BUFFER_EMPTY;
    }
    if((Buf->rdptr + 1) >= Buf->size){
        if(Buf->wrptr == 0){
            return EXIT_BUFFER_EMPTY;
        }
        else{
            Buf->rdptr = 0;
            return (uint8_t)Buf->buf[Buf->rdptr];
        }
    }
}

```

```

        }
    else {
        Buf->rdptr = Buf->rdptr + 1;
        return (uint8_t)Buf->buf[Buf->rdptr];
    }
}

/*
* putChar puts a character in the circular buffer
* and advances the write pointer. If the buffer
* is full, the function returns an EXIT_FAILURE. If
* successful the function returns an EXIT_SUCCESS.
*/
int32_t putChar(Circ_Buf *Buf, uint8_t c){

    if((Buf->wrptr + 1) >= Buf->size){
        if(Buf->rdptr == 0){
            return EXIT_BUFFER_FULL;
        }
        Buf->buf[Buf->wrptr] = c;
        Buf->wrptr = 0;
        return EXIT_SUCCESS;
    }
    if(Buf->rdptr == (Buf->wrptr + 1)){
        return EXIT_FAILURE;
    }
    Buf->buf[Buf->wrptr] = c;
    Buf->wrptr = Buf->wrptr + 1;
    return EXIT_SUCCESS;
}

/*
* putMsg puts a message into the circular buffer.
* it returns EXIT_SUCCESS if the message could be
* inserted into the buffer and EXIT_FAILURE if not.
*/
int32_t putMsg(Circ_Buf *Buf, uint8_t *msg){
    int32_t MsgLength = 0, i = 0, BufSpace = 0;

    BufSpace = getRemBufSpace(Buf);

    //Message length
    while(*(*msg+MsgLength) != '\0'){
        MsgLength++;
    }

    if(BufSpace < MsgLength){
        return EXIT_FAILURE;
    }
}

```

```

    }
    else{
        while(MsgLength > 0){
            if(!putChar(Buf, *(msg + i))){
                return EXIT_FAILURE;
            }
            MsgLength--;
            i++;
        }
        return EXIT_SUCCESS;
    }
}

/*
 * The function returns the remaining space
 * that's left in the buffer Buf. It returns
 * EXIT_FAILURE if the rdptr is the same as
 * the wrptr and therefore the buffer is
 * inconsistent!
*/
int32_t getRemBufSpace(Circ_Buf *Buf){
    int32_t i = 0;

    if(Buf->rdptr > Buf->wrptr){
        i = Buf->size - Buf->wrptr + (Buf->size - Buf->rdptr) - 1;
    }
    if(Buf->rdptr < Buf->wrptr){
        i = Buf->size - Buf->wrptr + Buf->rdptr - 2;
    }
    if((Buf->rdptr + 1) >= Buf->size){
        if(Buf->wrptr == 0) return Buf->size - 1;
    }
    return i;
}

/*
 * getMsg returns a pointer to the array
 * in which the message has been placed
 * or NULL if the buffer is too short for
 * the given message.
*/
uint8_t getMsg(Circ_Buf *Buf, uint8_t *MsgBuf, uint16_t BufSize){
    int32_t c, i = 0;
    c = getChar(Buf);
    while(c != '\n'){
        if(c==EXIT_BUFFER_EMPTY){
            return EXIT_BUFFER_EMPTY;
        }
    }
}

```

```

        MsgBuf[i] = c;
        i++;
        if(i >= (BufSize - 2)) return EXIT_BUFFER_SIZE;
        c = getChar(Buf);
    }

/*
 * Since we left the while before adding the \n and the
 * string end we need to add it now!
 */
MsgBuf[i] = '\n';
MsgBuf[i+1] = '\0';
return EXIT_SUCCESS;
}

```

C.4. inc_circular.h

```

/*
 * inc_circular.h
 *
 * Created on: 21.10.2013
 * Author: einball
 */

#ifndef INC_CIRCULAR_H_
#define INC_CIRCULAR_H_

#include <stdint.h>
#include <stdlib.h>
#include "inc_def.h"

#define EXIT_BUFFER_FULL 0x02
#define EXIT_MSG_PENDING 0x03
#define EXIT_BUFFER_EMPTY 0x04
#define EXIT_MSG_NOT_VALID 0x05
#define EXIT_MSG_IS_VALID 0x06
#define EXIT_BUFFER_SIZE 0x07

/* Already defined them
 * #define EXIT_SUCCESS
 * #define EXIT_FAILURE
 */

/* Do not forget to initialize the Buffers with rdptr == wrptr! */
typedef struct Buf {
    uint8_t buf[BUF_SIZE];
    uint32_t rdptr;
    uint32_t wrptr;
    uint32_t size;
}

```

```

    } Circ_Buf;

    uint8_t getMsg(Circ_Buf *Buf, uint8_t *MsgBuf, uint16_t BufSize);
    int32_t getRemBufSpace(Circ_Buf *Buf);
    int32_t putMsg(Circ_Buf *Buf, uint8_t *msg);
    int32_t putChar(Circ_Buf *Buf, uint8_t c);
    uint8_t getChar(Circ_Buf *Buf);

#endif /* INC_CIRCULAR_H_ */

```

C.5. msg.c

```

/*
 * msg.c
 *
 * Created on: 19.10.2013
 *      Author: einball
 */

#include "inc_msg.h"

#include "inc_circular.h"

/* addJob fgt einen Job zur Warteschlange hinzu */
uint32_t addJob(sJobQueue *Queue, sJob *Job){

    uint32_t i;

    for(i = 0; i < 10; i++){
        if(Queue->Job[i].id == 0){
            Queue->Job[i] = *Job;
            return EXIT_SUCCESS;
        }
    }
    //wird kein Feld mit leerer ID gefunden so ist die Warteschlange voll:
    EXIT_FAILURE
    return EXIT_FAILURE;
}

/* callJob berprft zuerst ob der aktuelle Job leer ist, wenn nicht muss dieser erst
   bearbeitet werden (return JOB_PENDING).
* Ist der Job leer wird das erste Element der Warteschlange, falls vorhanden, in den
   Job geschrieben und die restlichen Elemente werden eins aufgerckt.
* Ist die Warteschlange leer wird ein EXIT_FAILURE zurckgegeben
*/
uint32_t callJob(sJobQueue *Queue, sJob *Job){
    uint32_t i;

```

```

if(Job->id != 0)
    /* es wartet noch ein Job auf bearbeitung -> bearbeitung einleiten */
    return JOB_PENDING;

/* Abfrage, ob Element 0 in der Warteschlange ist */
if(Queue->Job[0].id != 0){
    /* Erstes Element der Warteschlange In den Motorsteuerungs-Puffer
       laden */
    Job->coordX = Queue->Job[0].coordX;
    Job->coordY = Queue->Job[0].coordY;
    Job->coordZ = Queue->Job[0].coordZ;
    Job->JOINT = Queue->Job[0].JOINT;
    Job->GRIPPER = Queue->Job[0].GRIPPER;
    Job->crc = Queue->Job[0].crc;
    Job->opt1 = Queue->Job[0].opt1;
    Job->opt2 = Queue->Job[0].opt2;
    Job->opt3 = Queue->Job[0].opt3;

    /* Alle Elemente der Warteschlange eins "Vorrcken" und die letzten
       elemente zu Null setzen */
    for(i = 0; i < 10; i++){
        if(Queue->Job[i+1].id == 0){
            Queue->Job[i].id = 0;
            Queue->Job[i].opCode = 0;
            Queue->Job[i].coordX = 0;
            Queue->Job[i].coordY = 0;
            Queue->Job[i].coordZ = 0;
            Queue->Job[i].JOINT = 0;
            Queue->Job[i].GRIPPER = 0;
            Queue->Job[i].crc = 0;
            Queue->Job[i].opt1 = 0;
            Queue->Job[i].opt2 = 0;
            Queue->Job[i].opt3 = 0;
        }
        else Queue->Job[i] = Queue->Job[i+1];
    }
    return EXIT_SUCCESS;
}
else return EXIT_FAILURE;
}

/* jobDone leert den bergebenen Job, was normalerweise der letzte bearbeitete Job ist
 */
uint32_t jobDone(sJob *Job){

    Job->id = 0;
    Job->opCode = 0;
    Job->coordX = 0;
    Job->coordY = 0;

```

```

        Job->coordZ = 0;
        Job->JOINT = 0;
        Job->GRIPPER = 0;
        Job->crc = 0;
        Job->opt1 = 0;
        Job->opt2 = 0;
        Job->opt3 = 0;

        return EXIT_SUCCESS;
    }

/* checkCRC soll, wie der Name vermuten lässt, einen CR-Check durchführen. Dieses Feature ist allerdings noch nicht implementiert. */
uint32_t CheckCRC(uint32_t CRC){

    return EXIT_SUCCESS;
}

/* validateID überprüft die empfangene ID auf Gültigkeit.
* ähnlich wie bei der CRC Prüfung ist diese Funktion noch nicht vollständig implementiert. Derzeit geht die Funktion lediglich die Warteschlange
* durch und überprüft die IDs der anderen Jobs auf Bereinstimmung, um eine "Doppelbelegung" zu vermeiden.
*/
uint32_t validateID(sJobQueue *Queue, uint32_t ID){
    uint32_t i;
    for(i=0;i<10;i++){
        if(Queue->Job[i].id == ID){
            return EXIT_FAILURE;
        }
    }
    return EXIT_SUCCESS;
}

/* Im Gegensatz zu validateID und checkCRC dient validateMsg keinen Formellen (Schutz vor Missbrauch, bzw Betrug), sondern praktischen Zwecken:
* Die Funktion überprüft ob die empfangenen Parameter Sinn ergeben, um Schäden am Automaten vorzubeugen.
* Die Prüfung von opt1-3 ist zwar vorgesehen aber noch nicht implementiert, da opt1-3 noch nicht mit Inhalt belegt sind.
*/
uint32_t validateMsg(sJob *Job){

    /* eventuelle weitere opCodes müssen natürlich auch noch hier eingefügt werden,
     * da die Funktion sonst automatisch in den default-case geht */
    switch(Job->opCode){
        case OPC_AUSGABE:
            /* alle Koordinaten müssen größer als null sein, sonst liegt ein

```

```

        * Fehlerfall vor, da (0,0,0) die Home Position ist und die
        Kstchenbreite
        * immer >=1 ist...nur die Rotationsachse darf ausschlielich 0 sein,
        * da sonst das aufnehmen der boxen nicht klappt */
if((Job->coordX == 0) || (Job->coordY == 0) || (Job->coordZ == 0) ||
   (Job->GRIPPER == 0) || (Job->JOINT != 0))
    return EXIT_FAILURE;

/* Genauso, wie ein Parameter immer grer 0 sein muss, darf natrlich
   ein Maximum nicht berschritten werden */
if((Job->coordX > Max_Pos_X) || (Job->coordY > Max_Pos_Y) || (Job->
   coordZ > Max_Pos_Z) || (Job->GRIPPER > Max_Pos_GRIPPER))
    return EXIT_FAILURE;

/*opt1..opt3 ???*/
break;

case OPC_NACHFUELLEN:
/* beim nachfllen gelten vorerst die selben bedingungen wie beim
   Ausgeben, sie sind allerdings getrennt aufgefhrt,
   * falls spter eine Trennung erforderlich wird */
if((Job->coordX <= 0) || (Job->coordY <= 0) || (Job->coordZ <= 0) ||
   (Job->GRIPPER <= 0) || (Job->JOINT != 0))
    return EXIT_FAILURE;

if((Job->coordX > Max_Pos_X) || (Job->coordY > Max_Pos_Y) || (Job->
   coordZ > Max_Pos_Z) || (Job->GRIPPER > Max_Pos_GRIPPER))
    return EXIT_FAILURE;

/*opt1..opt3 ???*/
break;

case OPC_FERNSTEUERN:
/* auch hier gilt: grenzen mssen eingehalten werden! weitere
   bedingungen fehlen auch hier */
if((Job->coordX <= 0) || (Job->coordY <= 0) || (Job->coordZ <= 0) ||
   (Job->GRIPPER <= 0) || (Job->JOINT != 0))
    return EXIT_FAILURE;

if((Job->coordX > Max_Pos_X) || (Job->coordY > Max_Pos_Y) || (Job->
   coordZ > Max_Pos_Z) || (Job->GRIPPER > Max_Pos_GRIPPER))
    return EXIT_FAILURE;
break;

case OPC_REKALIBRIEREN:
break;

default:
/* wenn wir in den default-case fallen stimmt irgendwas mit dem
   opCode nicht

```

```

        * -> Abbruch und Job leeren */
    return EXIT_FAILURE;
}

/* kein fehler ist aufgetreten: die Empfangene Nachricht ist Valide und der
   Job kann weitergefhrt werden */
return EXIT_SUCCESS;
}

/* decodeMsg entnimmt den empfangenen Datenstrom aus dem Ringpuffer und trennt die
   einzelnen Bestandteile,
* berprft und Speichert diese als Parameter in einem Job ab, sofern alles in Ordnung
   ist.
* Dieser Job wird als JobBuffer in der Statemachine weiterverarbeitet.
*/
uint32_t decodeMsg(sJobQueue *Queue, sJob *Buf, uint8_t *MsgPointer){
    uint32_t i=0, iSection = 1;
    uint8_t *ID_Buf, *CRC_Buf, *OPCode_Buf, *X_Buf, *Y_Buf, *Z_Buf, *JOINT_Buf, *
           GRIPPER_Buf;
    /* uint32_t optArgA, optArgB, optArgC;
       uint32_t *optArgA_Buf, *optArgB_Buf, *optArgC_Buf;
    */

    /* ID_Buf hlt die ID des Jobs. Die ID muss als erstes stehen */
    ID_Buf = MsgPointer;

    while( *(MsgPointer+i) != '[' ){
        i++;
        if(*(MsgPointer+i) == '\n'){
            return MSG_CODE_ERROR;
        }
    }

    /* Der Pointer + i zeigt nun auf die ffnende Klammer.
     * Die wird kurzerhand mit der Stringende Kennung ersetzt.
     */
    *(MsgPointer+i) = '\0';
    i++;

    /* Nun muss zwingend eine Kommata separierte Liste folgen!
     * Was aber auf jeden Fall klar ist, ist dass nun der OpCode folgt!
     */
    OpCode_Buf = (MsgPointer + i);

    while( *(MsgPointer + i) != ']' ){

        i++;
        if(*(MsgPointer+i) == '\n'){


```

```

        return MSG_CODE_ERROR;
    }

    if( *(MsgPointer + i) == ' ', ' '){
        switch(iSection){
            case 1:
                *(MsgPointer + i) = '\0';
                iSection = 2;
                X_Buf = (MsgPointer + i + 1);
                break;

            case 2:
                *(MsgPointer + i) = '\0';
                iSection = 3;
                Y_Buf = (MsgPointer + i + 1);
                break;

            case 3:
                *(MsgPointer + i) = '\0';
                iSection = 4;
                Z_Buf = (MsgPointer + i + 1);
                break;

            case 4:
                *(MsgPointer + i) = '\0';
                iSection = 5;
                JOINT_Buf = (MsgPointer + i + 1);
                break;

            case 5:
                *(MsgPointer + i) = '\0';
                iSection = 6;
                GRIPPER_Buf = (MsgPointer + i + 1);
                break;

            case 6:
                *(MsgPointer + i) = '\0';
                iSection++;
                //optArgA_Buf = MsgPointer + i + 1;
                break;

            case 7:
                *(MsgPointer + i) = '\0';
                iSection++;
                //optArgB_Buf = MsgPointer + i + 1;
                break;

            case 8:
                *(MsgPointer + i) = '\0';

```

```

                iSection++;
                //optArgC_Buf = MsgPointer + i + 1;
                break;

            case 9:
                *(MsgPointer + i) = '\0';
                iSection++;
                break;

            default:
                /* Wir tun hier einfach nichts, denn
                 * alle weiteren Argumente sind uns egal:
                 * Wir haben erstmal nur 3!
                 */
                break;
        }
    }

}

*(MsgPointer + i) = '\0';

/* Nun sind wir beim CRC Code angelangt */
CRC_Buf = (MsgPointer + i + 1);
while( *(MsgPointer + i) != '\n'){
    i++;
}

*(MsgPointer + i) = '\0';

/* Nun knnen wir uns ans decoden mit atoi() usw machen */

Buf->id=atoi((char*)ID_Buf);
Buf->crc=atoi((char*)CRC_Buf);

/* Job NUR speichern, wenn ID und CR-Check erfolgreich */
if(validateID(Queue, Buf->id)==EXIT_SUCCESS){
    if(CheckCRC(Buf->crc)==EXIT_SUCCESS){
        Buf->opCode=atoi((char*)OPCode_Buf);           //String in uint32
        konvertieren
        Buf->coordX=atoi((char*)X_Buf);
        Buf->coordY=atoi((char*)Y_Buf);
        Buf->coordZ=atoi((char*)Z_Buf);
        Buf->JOINT=atoi((char*)JOINT_Buf);
        Buf->GRIPPER=atoi((char*)GRIPPER_Buf);
        // Hier einfgen:
        // optArgA
        // optArgB
    }
}

```

```

        // optArgC
        //
    }
    else{
        return MSG_CRC_ERROR;
    }
    if(validateMsg(Buf) == EXIT_FAILURE){
        /* irgendetwas stimmt nicht mit der Nachricht: Lschen und
           Fehler ausgeben */
        jobDone(Buf);
        return MSG_INVALID;
    }

}
else{
    return MSG_ID_ERROR;
}
return MSG_ALL_OK;
}

```

C.6. inc_msg.h

```

/*
 * inc_msg.h
 *
 * Created on: 19.10.2013
 * Author: einball
 */

#ifndef INC_MSG_H_
#define INC_MSG_H_

#include "main.h"

//Work in progress!

typedef struct Job {
    uint32_t id;
    uint32_t opCode;
    uint32_t coordX;
    uint32_t coordY;
    uint32_t coordZ;
    uint32_t JOINT;
    uint32_t GRIPPER;
    uint32_t opt1;
    uint32_t opt2;
    uint32_t opt3;
    uint32_t crc;
} sJob;

```

```

typedef struct JobQueue {
    sJob Job[10];
} sJobQueue;

//Message Teil
uint32_t checkMsgIntegrity();
uint32_t decodeMsg(sJobQueue *Job_Queue, sJob *Buf, uint8_t *MsgPointer);
uint32_t validateMsg(sJob *Job);

//Job Teil
uint32_t addJob(sJobQueue *Queue, sJob *Job);
uint32_t callJob(sJobQueue *Queue, sJob *Job);
uint32_t jobDone(sJob *Job);

#endif /* INC_MSG_H */

```

C.7. motor.c

```

/*
 * motor.c
 *
 * Created on: 21.11.2013
 *      Author: einball-dev
 */

#include "inc_motor.h"
#include "inc_dspin.h"

uint32_t getBox(sJob *Destination){

    /* Fahrtrichtung bestimmen und XY-Position Anfahren */

    if(((Destination->coordX + Offset_X) * Stepsize) - dSPIN_GetParam(
        MOT_X_AXIS_1, dSPIN_ABS_POS)) > 0){
        dSPIN_GoTo_DIR(MOT_X_AXIS_1, 1, (Destination->coordX + Offset_X) *
            Stepsize);
    }
    else{dSPIN_GoTo_DIR(MOT_X_AXIS_1, 0, (Destination->coordX + Offset_X) *
        Stepsize);}

    if(((Destination->coordY + Offset_Y) * Stepsize) - dSPIN_GetParam(MOT_Y_AXIS
        , dSPIN_ABS_POS)) > 0{
        dSPIN_GoTo_DIR(MOT_Y_AXIS, 1, (Destination->coordY + Offset_Y) *
            Stepsize);
    }
}

```

```

    else{dSPIN_GoTo_DIR(MOT_Y_AXIS , 0, (Destination->coordY + Offset_Y) *
        Stepsize);}

    while(1){
        if((1 & dSPIN_GetParam(MOT_X_AXIS_1, dSPIN_STATUS)) && (1 &
            dSPIN_GetParam(MOT_Y_AXIS, dSPIN_STATUS))){
            // Warten, bis X und Z Position angefahren wurden
            break;
        }
    }

    /* Greifer öffnen, Z-Position anfahren, Box greifen und rausziehen */
    dSPIN_GoTo(MOT_GRIPPER, Destination->GRIPPER * (BoxSize + Offset_GRIPPER) *
        Stepsize);
    while(1){
        if(1 & dSPIN_GetParam(MOT_GRIPPER, dSPIN_STATUS)){
            // Warten, bis Greifer geöffnet wurde
            break;
        }
    }

    dSPIN_GoTo(MOT_Z_AXIS, Destination->coordZ * BoxSize * Stepsize);
    while(1){
        if(1 & dSPIN_GetParam(MOT_Z_AXIS, dSPIN_STATUS)){
            // Warten, bis Z Position angefahren wurde
            break;
        }
    }

    dSPIN_GoTo(MOT_GRIPPER, Destination->GRIPPER * BoxSize * Stepsize);
    while(1){
        if(1 & dSPIN_GetParam(MOT_GRIPPER, dSPIN_STATUS)){
            // Warten, bis Greifer geschlossen wurde
            break;
        }
    }

    dSPIN_GoHome(MOT_Z_AXIS);
    while(1){
        if(1 & dSPIN_GetParam(MOT_Z_AXIS, dSPIN_STATUS)){
            // Warten, bis Z Position zurückgesetzt wurde
            break;
        }
    }

    return EXIT_SUCCESS;
}

uint32_t putBox(sJob *Destination){

```

```

/* Fahrtrichtung bestimmen und XY-Position Anfahren */
if(((Destination->coordX + Offset_X) * Stepsize) - dSPIN_GetParam(
    MOT_X_AXIS_1, dSPIN_ABS_POS)) >0){
    dSPIN_GoTo_DIR(MOT_X_AXIS_1, 1, (Destination->coordX + Offset_X) *
        Stepsize);
}
else{dSPIN_GoTo_DIR(MOT_X_AXIS_1, 0, (Destination->coordX + Offset_X) *
    Stepsize);}

if(((Destination->coordY + Offset_Y) * Stepsize) - dSPIN_GetParam(MOT_Y_AXIS
    , dSPIN_ABS_POS)) >0){
    dSPIN_GoTo_DIR(MOT_Y_AXIS, 1, (Destination->coordY + Offset_Y) *
        Stepsize);
}
else{dSPIN_GoTo_DIR(MOT_Y_AXIS, 0, (Destination->coordY + Offset_Y) *
    Stepsize);}

while(1){
    if((1 & dSPIN_GetParam(MOT_X_AXIS_1, dSPIN_STATUS)) && ( 1 &
        dSPIN_GetParam(MOT_Y_AXIS, dSPIN_STATUS))){
        // Warten, bis X und Y Position angefahren wurden
        break;
    }
}

/* Z-Position anfahren, Box ablegen und zurckfahren */

dSPIN_GoTo(MOT_Z_AXIS, ((Destination->coordZ + Offset_Z) * Stepsize));
while(1){
    if(1 & dSPIN_GetParam(MOT_Z_AXIS, dSPIN_STATUS)){
        // Warten, bis Z Position angefahren wurde
        break;
    }
}

dSPIN_GoTo(MOT_GRIPPER, Destination->GRIPPER * (BoxSize + Offset_GRIPPER) *
    Stepsize); // Platzhalter fr Greiferstellung
while(1){
    if(1 & dSPIN_GetParam(MOT_GRIPPER, dSPIN_STATUS)){
        // Warten, bis Greifer geffnet wurde
        break;
    }
}

dSPIN_GoHome(MOT_Z_AXIS);
while(1){

```

```

        if(1 & dSPIN_GetParam(MOT_Z_AXIS, dSPIN_STATUS)){
            // Warten, bis Z Position zurckgesetzt wurde
            break;
        }
    }

    return EXIT_SUCCESS;
}

uint32_t EmptyBox(){

    dSPIN_GoHome(MOT_X_AXIS_1);

    if((AusgabeschachtY + Offset_Y) * Stepsize - dSPIN_GetParam(MOT_Y_AXIS,
        dSPIN_ABS_POS)) > 0){
        dSPIN_GoTo_DIR(MOT_Y_AXIS, 1, (AusgabeschachtY + Offset_Y) * Stepsize
            );
    }
    else{dSPIN_GoTo_DIR(MOT_Y_AXIS, 0, (AusgabeschachtY + Offset_Y) * Stepsize);}

    while(1){
        // Warten, bis X und Y Position angefahren wurden
        if((1 & dSPIN_GetParam(MOT_X_AXIS_1, dSPIN_STATUS)) && (1 &
            dSPIN_GetParam(MOT_Y_AXIS, dSPIN_STATUS))){
            break;
        }
    }

    dSPIN_GoTo(MOT_Z_AXIS, (AusgabeschachtZ + Offset_Z) * Stepsize);
    while(1){
        // Warten, bis Z Position angefahren wurde
        if(1&dSPIN_GetParam(MOT_Z_AXIS, dSPIN_STATUS)){
            break;
        }
    }

    dSPIN_GoTo(MOT_JOINT, Halbkreis);
    while(1){
        if(1 & dSPIN_GetParam(MOT_JOINT, dSPIN_STATUS)){
            // Warten, bis fertig Rotiert
            break;
        }
    }
    dSPIN_GoHome(MOT_JOINT);
    dSPIN_GoHome(MOT_Z_AXIS);
    while(1){
        if(1 & dSPIN_GetParam(MOT_Z_AXIS, dSPIN_STATUS)){
            // Warten, bis Z Position zurckgesetzt wurde
            break;
        }
    }
}

```

```

    }

    return EXIT_SUCCESS;
}

uint32_t RefillBox(){

    dSPIN_GoHome(MOT_X_AXIS_1);
    if((Ausgabeschachty + Offset_Y) * Stepsize - dSPIN_GetParam(MOT_Y_AXIS,
        dSPIN_ABS_POS)) > 0){
        dSPIN_GoTo_DIR(MOT_Y_AXIS, 1, (Ausgabeschachty + Offset_Y) * Stepsize
            );
    }
    else{dSPIN_GoTo_DIR(MOT_Y_AXIS, 0, (Ausgabeschachty + Offset_Y) * Stepsize);}

    while(1){
        if((1 & dSPIN_GetParam(MOT_X_AXIS_1, dSPIN_STATUS)) && (1 &
            dSPIN_GetParam(MOT_Y_AXIS, dSPIN_STATUS))){
            // Warten, bis X und Y Position angefahren wurden
            break;
        }
    }

    dSPIN_GoTo(MOT_Z_AXIS, (AusgabeschachtZ + Offset_Z) * Stepsize);
    while(1){
        //muss noch auf den richtigen Port/Pin abgestimmt werden, siehe auch:
        // startup.c
        if(GPIOPinRead(GPIO_PORTF_BASE, GPIO_PIN_4))
            break;
    }

    dSPIN_GoHome(MOT_Z_AXIS);

    while(1){
        if(1 & dSPIN_GetParam(MOT_Z_AXIS, dSPIN_STATUS)){
            // Warten, bis Z Position zurckgesetzt wurde
            break;
        }
    }

    return EXIT_SUCCESS;
}

uint32_t Remote_Ctrl_Motor(sJob *Destination){

    //zum debuggen
    return EXIT_SUCCESS;
}

```

```

/* X- und Y-Positionen anfahren */
if(((Destination->coordX + Offset_X) * Stepsize) - dSPIN_GetParam(
    MOT_X_AXIS_1, dSPIN_ABS_POS)) > 0){
    dSPIN_GoTo_DIR(MOT_X_AXIS_1, 1, (Destination->coordX + Offset_X) *
        Stepsize);
}
else{dSPIN_GoTo_DIR(MOT_X_AXIS_1, 0, (Destination->coordX + Offset_X) *
    Stepsize);}

if(((Destination->coordY + Offset_Y) * Stepsize) - dSPIN_GetParam(MOT_Y_AXIS
    , dSPIN_ABS_POS)) > 0{
    dSPIN_GoTo_DIR(MOT_Y_AXIS, 1,(Destination->coordY + Offset_Y) *
        Stepsize);
}
else{dSPIN_GoTo_DIR(MOT_Y_AXIS, 0, (Destination->coordY + Offset_Y) *
    Stepsize);}

while(1){
    // Warten, bis X und Y Position angefahren wurden
    if((1 & dSPIN_GetParam(MOT_X_AXIS_1, dSPIN_STATUS)) && (1 &
        dSPIN_GetParam(MOT_Y_AXIS, dSPIN_STATUS))){
        break;
    }
}

/*Z-Position Anfahren ---- Die Trennung von X/Y und Z ist zum schutz des
Automateninhalts gedacht, siehe auch putbox, getbox, refillbox und
emptybox */
if(((Destination->coordZ + Offset_Z) * Stepsize) - dSPIN_GetParam(MOT_Z_AXIS
    , dSPIN_ABS_POS)) > 0{
    dSPIN_GoTo_DIR(MOT_Z_AXIS, 1,(Destination->coordZ + Offset_Z) *
        Stepsize);
}
else{dSPIN_GoTo_DIR(MOT_Z_AXIS, 0, (Destination->coordZ + Offset_Z) *
    Stepsize);}

while(1){
    // Warten, bis X und Y Position angefahren wurden
    if(1 & dSPIN_GetParam(MOT_Z_AXIS, dSPIN_STATUS)){
        break;
    }
}

/* Greifertyp und Rotationsachse ändern, aus dem selben grund getrennt, wie
auch bei Z-Position */

```

```

        if(((Destination->GRIPPER + Offset_GRIPPER) * Stepsize) - dSPIN_GetParam(
            MOT_GRIPPER , dSPIN_ABS_POS)) > 0){
            dSPIN_GoTo_DIR(MOT_GRIPPER , 1, (Destination->GRIPPER + Offset_GRIPPER
                ) * Stepsize);
        }
        else{dSPIN_GoTo_DIR(MOT_GRIPPER , 0, (Destination->GRIPPER + Offset_GRIPPER) *
            Stepsize);}

        if(((Destination->JOINT + Offset_JOINT) * Stepsize) - dSPIN_GetParam(
            MOT_JOINT , dSPIN_ABS_POS)) > 0){
            dSPIN_GoTo_DIR(MOT_JOINT , 1, (Destination->JOINT + Offset_JOINT) *
                Stepsize);
        }
        else{dSPIN_GoTo_DIR(MOT_JOINT , 0, (Destination->JOINT + Offset_JOINT) *
            Stepsize);}

    while(1){
        if((1 & dSPIN_GetParam(MOT_GRIPPER , dSPIN_STATUS)) && ( 1 &
            dSPIN_GetParam(MOT_JOINT , dSPIN_STATUS))){
            // Warten, bis Greifer und Drehgelenk Verstellt wurden
            break;
        }
    }

    return EXIT_SUCCESS;
}

uint32_t Ctrl_Motor(sJob *Destination){

    sJob sort;
    sJob FREE_SPACE;

    //zum debuggen
    return EXIT_SUCCESS;

/*
 * FREE_SPACE ist das Sortierfeld, in das Kisten umgelagert werden,
 * um an Hintere Kisten zu kommen, z.B.:
 * man will an Kiste Nr.3 (coordZ=3), also lagert man Kisten 1 und 2 in
 * FREE_SPACE,
 * bevor man sich Kiste 3 holt
 * Die switch anweisung berprft, welches Sortierfeld gebraucht wird
 */
    FREE_SPACE.JOINT=FREE_SPACE_JOINT;
    switch (Destination->GRIPPER){
    case(1):
        FREE_SPACE.coordX=FREE_SPACE_X_1;
        FREE_SPACE.coordY=FREE_SPACE_Y_1;

```

```

    FREE_SPACE.coordZ=FREE_SPACE_Z_1;
    FREE_SPACE.GRIPPER=Destination->GRIPPER;
    break;
case(2):
    FREE_SPACE.coordX=FREE_SPACE_X_2;
    FREE_SPACE.coordY=FREE_SPACE_Y_2;
    FREE_SPACE.coordZ=FREE_SPACE_Z_2;
    FREE_SPACE.GRIPPER=Destination->GRIPPER;
    break;
// Hier mehr Gren einfgen, nach bedarf
}

sort=*Destination;
sort.coordZ--;
/*
 * Hier Greiferstellung Anpassen
 */
switch(Destination->opCode){
case OPC_AUSGABE :
    if(Destination->coordZ>1){           //Wenn Schachtel nicht in
        erster Reihe steht
        getBox(&sort);                   //getBox fhrt an die
        richtige Position und holt die angegebene Box
        //analog dazu: putBox stellt die Box in Frage an die
        angegebene Position
        putBox(&FREE_SPACE);           //FREE_SPACE_N ist ein Leerer
        Schacht der Boxgre N
    }                                     //Die
        Boxen werden immer so weit hinten wie moglich platziert
    getBox(Destination);
    EmptyBox();                         //EmptyBox
        fhrt zum ausgabeschacht und rotiert den A-Motor
    putBox(Destination);

    if(Destination->coordZ>1){           //die
        Umsortierten Schachteln werden zurckgestellt
        getBox(&FREE_SPACE);
        putBox(&sort);
    }
    break;
case OPC_NACHFUELLEN:
    if(Destination->coordZ>1){
        getBox(&sort);
        putBox(&FREE_SPACE);
    }
    getBox(Destination);
    RefillBox();
    putBox(Destination);

    if(Destination->coordZ>1){

```

```

        getBox(&FREE_SPACE);
        putBox(&sort);
    }
    break;
}
////return EXIT_FAIL muss auch noch irgendwie festgelegt werden
//--> Error states in den o.g. Funktionen setzen
return EXIT_SUCCESS;
}

```

C.8. inc_motor.h

```

/*
 * inc_motor.h
 *
 * Created on: 21.11.2013
 * Author: einball-dev
 */

#ifndef INC_MOTOR_H_
#define INC_MOTOR_H_

#include <stdint.h>
#include <stdlib.h>
#include <stdbool.h>
#include "main.h"
#include "inc_msg.h"
#include "inc_def.h"
#include <hardware/hw_memmap.h>
#include <driverlib/gpio.h>

/*
 * Koordinaten-Offsets, fr genauere Berechnung der Fahrtwege
 * sie sind durch Variablen zu ersetzen,
 * die ber die Serielle Schnittstelle Kalibrierbar sein sollen
 */
#define Offset_X          0
#define Offset_Y          0
#define Offset_Z          0
#define Offset_JOINT      0
#define Offset_GRIPPER    0      //Variationsgre, die fr Luftspalte beim greifen/
                           //loslassen sorgen soll (-> Boxsize+/-SizeVar)
#define Stepsize           1      //schrittbreite der Motoren
#define Halbkreis          1      //soll fr eine 180 drehung stehen, muss aber
                           //mit der Schrittgre vereinbart werden
#define BoxSize            1      //Boxengre als Multiplikator, abhngig von
                           //Stepsize (-> GRIPPER*Boxsize*Stepsize = Greifertbreite)

//AusgabeschachtX = Home//
```

```

#define AusgabeschachtY 1
#define AusgabeschachtZ 1

// Steuerfunktionen

uint32_t Ctrl_Motor(sJob *Destination);
uint32_t Remote_Ctrl_Motor(sJob *Destination);
uint32_t RefillBox();
uint32_t EmptyBox();
uint32_t putBox(sJob *Destination);
uint32_t getBox(sJob *Destination);

#endif /* INC_MOTOR_H_ */

```

C.9. def.c

```

/*
 * def.c
 *
 * Created on: 19.10.2013
 *      Author: einball
 */

#include "main.h"

void GOTO_STATE(sysState_t state){
    SYS_STATE = state;
}

```

C.10. inc_def.h

```

/*
 * defines.h
 *
 * Created on: 18.10.2013
 *      Author: einball
 */

#ifndef DEFINES_H_
#define DEFINES_H_

/* Success of fail */
#define EXIT_SUCCESS 1
#define EXIT_FAILURE 0

/* Motorzuordnung zur SPI Chain */

```

```

#define MOT_X_AXIS_1 0x00000000
#define MOT_X_AXIS_2 0x00000001
#define MOT_Y_AXIS 0x00000002
#define MOT_Z_AXIS 0x00000003
#define MOT_JOINT 0x00000004
#define MOT_GRIPPER 0x00000005

/* Definition der opCodes */

#define OPC_AUSGABE 0x00000000
#define OPC_NACHFUELLEN 0x00000001
#define OPC_FERNSTEUERN 0x00000002
#define OPC_REKALIBRIEREN 0x00000004

/* Maximale Koordinaten, begrenzt durch Automatengre, X und Z wei ich grad nicht
auswendig, ist noch zu klren */

#define Max_Pos_X 300
#define Max_Pos_Y 300
#define Max_Pos_Z 3
#define Max_Pos_GRIPPER 3 //maximale GRIPPER stellung -> hngt direkt mit
boxsize zusammen

//X und Y sind noch festzulegen, Z ist immer der letztmögliche Platz

#define FREE_SPACE_JOINT 0

#define FREE_SPACE_X_1 1
#define FREE_SPACE_Y_1 1
#define FREE_SPACE_Z_1 3

#define FREE_SPACE_X_2 1
#define FREE_SPACE_Y_2 2
#define FREE_SPACE_Z_2 3

/* Lnge der Puffer vom Typ Circ_Buf */
#define BUF_SIZE 1024

/* Lnge der Puffer, in denen die Message bearbeitet wird */
#define MSG_BUF_SIZE 64

/* Job Error states that can be passed to and returned from various functions */

#define JOB_ALL_OK 0x00000000
#define JOB_RECEIVE_ERROR 0x00000001
#define JOB_ADD_ERROR 0x00000002
#define JOB_ADD_SUCCESS 0x00000004
#define JOB_ADD_PENDING 0x00000008
#define JOB_PENDING 0x00000010
#define JOB_UNKNOWN_ERROR 0x00000020

```

```

#define      JOB_UNDEFINED8          0x00000040
#define      JOB_UNDEFINED9          0x00000080

/* UART Error states that can be passed to and returned from various functions */
#define      SER_ALL_OK              0x00000000
#define      SER_RX_MESSAGE_COMPLETE 0x00000001
#define      SER_RX_TIMEOUT          0x00000002
#define      SER_RX_IDLE             0x00000004
#define      SER_TX_TRANSFER_COMPLETE 0x00000008
#define      SER_TX_IDLE             0x00000010
#define      SER_RX_ERROR            0x00000020
#define      SER_TX_ERROR            0x00000040
#define      SER_RX_BUFFER_OVERRUN   0x00000080
#define      SER_RX_DECODE_VALID_MSG 0x00000100
#define      SER_UNDEFINED_ERROR     0x00000200
#define      SER_TX_REQUEST          0x00000400
#define      SER_RX_BUFFER_EMPTY     0x00000800

/* (Soft-)SPI Error states that can be passed to and from various functions */
#define      SPI_ALL_OK              0x00000000
#define      SPI_UNDEFINED2           0x00000001
#define      SPI_UNDEFINED3           0x00000002
#define      SPI_UNDEFINED4           0x00000004
#define      SPI_UNDEFINED5           0x00000008
#define      SPI_UNDEFINED6           0x00000010
#define      SPI_UNDEFINED7           0x00000020
#define      SPI_UNDEFINED8           0x00000040
#define      SPI_UNDEFINED9           0x00000080

/* Stateemachine Error states that can be passed to and returned from various
functions */
#define      STA_ALL_OK              0x00000000
#define      STA_UNDEFINED2           0x00000001
#define      STA_UNDEFINED3           0x00000002
#define      STA_UNDEFINED4           0x00000004
#define      STA_UNDEFINED5           0x00000008
#define      STA_UNDEFINED6           0x00000010
#define      STA_UNDEFINED7           0x00000020
#define      STA_UNDEFINED8           0x00000040
#define      STA_UNDEFINED9           0x00000080

/* Ethernet Error states that can be passed to and returned from various functions */
#define      ETH_ALL_OK              0x00000000
#define      ETH_UNDEFINED2           0x00000001
#define      ETH_UNDEFINED3           0x00000002
#define      ETH_UNDEFINED4           0x00000004
#define      ETH_UNDEFINED5           0x00000008

```

```

#define      ETH_UNDEFINED6          0x00000010
#define      ETH_UNDEFINED7          0x00000020
#define      ETH_UNDEFINED8          0x00000040
#define      ETH_UNDEFINED9          0x00000080

/* Message system Error states that can be passed to and returned from various
functions */
#define      MSG_ALL_OK              0x00000000
#define      MSG_LENGTH_ERROR        0x00000001
#define      MSG_INVALID             0x00000002
#define      MSG_CRC_ERROR           0x00000004
#define      MSG_ID_ERROR            0x00000008
#define      MSG_UNKNOWN_ERROR       0x00000010
#define      MSG_CODE_ERROR          0x00000020
#define      MSG_UNDEFINEDo          0x00000040
#define      MSG_UNDEFINEDp          0x00000080

/* Digital Input macros */
#define      DIN_0_VAL              (*((volatile
    uint32_t *) (GPIO_PORTB_BASE + (GPIO_0_DATA + ( 0x20 << 2 )))) >> 5)
#define      DIN_1_VAL              (*((volatile
    uint32_t *) (GPIO_PORTB_BASE + (GPIO_0_DATA + ( 0x80 << 2 )))) >> 7)
#define      DIN_2_VAL              (*((volatile
    uint32_t *) (GPIO_PORTF_BASE + (GPIO_0_DATA + ( 0x10 << 2 )))) >> 4)
#define      DIN_3_VAL              (*((volatile
    uint32_t *) (GPIO_PORTE_BASE + (GPIO_0_DATA + ( 0x08 << 2 )))) >> 3)
#define      DIN_4_VAL              (*((volatile
    uint32_t *) (GPIO_PORTE_BASE + (GPIO_0_DATA + ( 0x04 << 2 )))) >> 2)
#define      DIN_5_VAL              (*((volatile
    uint32_t *) (GPIO_PORTE_BASE + (GPIO_0_DATA + ( 0x02 << 2 )))) >> 1)
#define      DIN_6_VAL              (*((volatile
    uint32_t *) (GPIO_PORTE_BASE + (GPIO_0_DATA + ( 0x01 << 2 )))) >> 0)
#define      DIN_7_VAL              (*((volatile
    uint32_t *) (GPIO_PORTD_BASE + (GPIO_0_DATA + ( 0x80 << 2 )))) >> 7)
#define      DIN_8_VAL              (*((volatile
    uint32_t *) (GPIO_PORTC_BASE + (GPIO_0_DATA + ( 0x80 << 2 )))) >> 7)
#define      DIN_9_VAL              (*((volatile
    uint32_t *) (GPIO_PORTC_BASE + (GPIO_0_DATA + ( 0x40 << 2 )))) >> 6)

/* Motor fault macros */

#define      MOT_0_FLAG              ((~(*(
volatile uint32_t *) (GPIO_PORTA_BASE + (GPIO_0_DATA + ( 0x80 << 2 )))) >> 7)) & 0
x01)

```

```

#define MOT_1_FLAG ((~(*((volatile uint32_t *) (GPIO_PORTA_BASE + (GPIO_O_DATA + (0x40 << 2)))) >> 6)) & 0x01)
#define MOT_2_FLAG ((~(*((volatile uint32_t *) (GPIO_PORTC_BASE + (GPIO_O_DATA + (0x10 << 2)))) >> 4)) & 0x01)
#define MOT_3_FLAG ((~(*((volatile uint32_t *) (GPIO_PORTC_BASE + (GPIO_O_DATA + (0x20 << 2)))) >> 5)) & 0x01)
#define MOT_4_FLAG ((~(*((volatile uint32_t *) (GPIO_PORTE_BASE + (GPIO_O_DATA + (0x20 << 2)))) >> 5)) & 0x01)
#define MOT_5_FLAG ((~(*((volatile uint32_t *) (GPIO_PORTA_BASE + (GPIO_O_DATA + (0x10 << 2)))) >> 4)) & 0x01)

#endif /* DEFINES_H_ */

```

C.11. startup.c

```

/*
 * startup.c
 *
 * Created on: 21.10.2013
 * Author: einball
 */

#include "inc_startup.h"

void setup_INT(){
    IntEnable(INT_UART0);
    UARTIntEnable(UART0_BASE, UART_INT_RX | UART_INT_RT);
    UARTIntEnable(UART0_BASE, UART_INT_OE | UART_INT_BE);
    UARTIntEnable(UART0_BASE, UART_INT_PE | UART_INT_BE);
    UARTIntEnable(UART0_BASE, UART_INT_TX);

    IntMasterEnable();
}

void setup_GPIO(){

    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
    GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3);
        //LEDs als Output deklarieren
    //der genaue input Port/Pin muss noch angepasst werden
    GPIOPinTypeGPIOInput(GPIO_PORTF_BASE, GPIO_PIN_4);
}

```

```

    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOA); // Auf Port A liegen die UART
    Pins
}

void setup_UART(){
    SysCtlPeripheralEnable(SYSCTL_PERIPH_UART0); //Enable UART Ports
    UARTConfigSetExpClk(UART0_BASE, SysCtlClockGet(), 9600,
                        (UART_CONFIG_WLEN_8 | UART_CONFIG_STOP_ONE |
                        UART_CONFIG_PAR_NONE));
    GPIOPinTypeUART(GPIO_PORTA_BASE, GPIO_PIN_0 | GPIO_PIN_1); //Rx und Tx
    deklarieren
    UARTTxIntModeSet(UART0_BASE, UART_TXINT_MODE_EOT); // Tx Interrupt wird
    mit End-of-Transmission ausgelöst: Teil der schnellen Drte Lsung, siehe:
    Init_Transmission
}

void setup_SPI(){

}

void setup_JOB_QUEUE(sJobQueue *Queue){
    uint32_t i;
/* Die Job-warteschlange wird hier mit 0 initialisiert, allerdings geht so natrlich
   auch beim Systemneustart alles verloren */
    for(i = 0; i<10; i++){
        Queue->Job[i].id = 0;
        Queue->Job[i].opCode = 0;
        Queue->Job[i].coordX = 0;
        Queue->Job[i].coordY = 0;
        Queue->Job[i].coordZ = 0;
        Queue->Job[i].JOINT = 0;
        Queue->Job[i].GRIPPER = 0;
        Queue->Job[i].crc = 0;
        Queue->Job[i].opt1 = 0;
        Queue->Job[i].opt2 = 0;
        Queue->Job[i].opt3 = 0;
    }
}

void setup_HOME_POS(){
/*
 * GoUntil(MOT_JOINT,0,0,MIN_SPEED)
 * GoUntil(MOT_GRIPPER,0,1,MIN_SPEED)
 * GoUntil(MOT_Z_AXIS,0,0,MIN_SPEED),
 * GoUntil(MOT_Y_AXIS,0,0,MIN_SPEED)
 * GoUntil(MOT_X_AXIS,0,0,MIN_SPEED)
*/
}

```

C.12. inc_startup.h

```
/*
 * inc_startup.h
 *
 * Created on: 21.10.2013
 *      Author: einball
 */

#ifndef INC_STARTUP_H_
#define INC_STARTUP_H_

#define TARGET_IS_BLIZZARD_RA2
#define TM4C1233H6PM

#include <stdint.h>
#include <stdbool.h>

#include <driverlib/interrupt.h>
#include <driverlib/sysctl.h>
#include <driverlib/uart.h>
#include <driverlib/ssi.h>
#include <driverlib/gpio.h>
#include <driverlib/watchdog.h>
#include <driverlib/timer.h>
#include <driverlib/rom.h>
#include <driverlib/rom_map.h>

#include <hardware/hw_ints.h>
#include <hardware/hw_sysctl.h>
#include <hardware/hw_uart.h>
#include <hardware/hw_ssi.h>
#include <hardware/hw_gpio.h>
#include <hardware/hw_watchdog.h>
#include <hardware/hw_timer.h>
#include <hardware/hw_memmap.h>

#include <inc_msg.h>

void setup_GPIO();
void setup_UART();
void setup_SPI();
void setup_INT();
void setup_HOME_POS();
void setup_JOB_QUEUE(sJobQueue *Queue);

#endif /* INC_STARTUP_H_ */
```

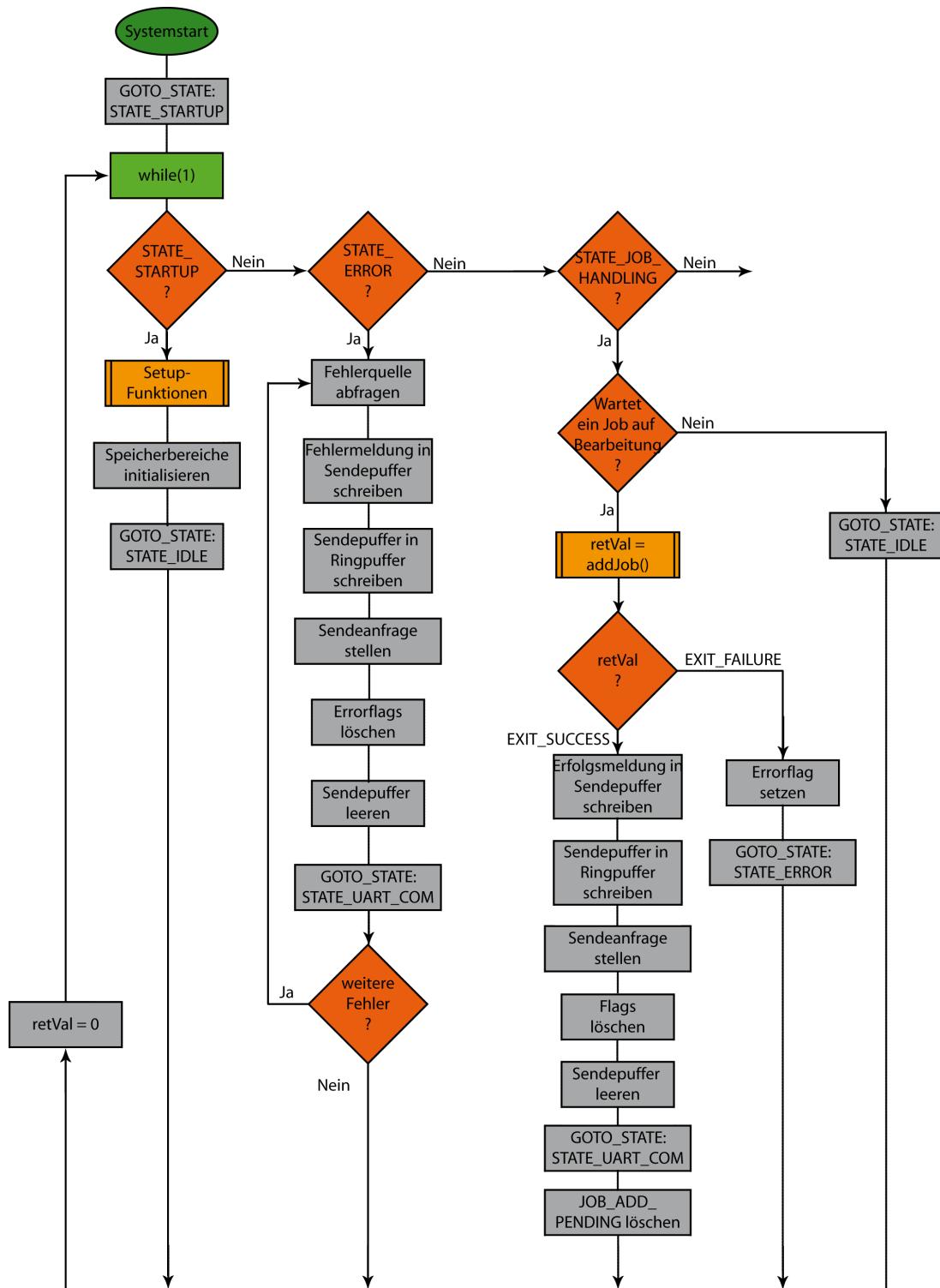


Abbildung 14: Ablaufdiagramm der main, teil 1

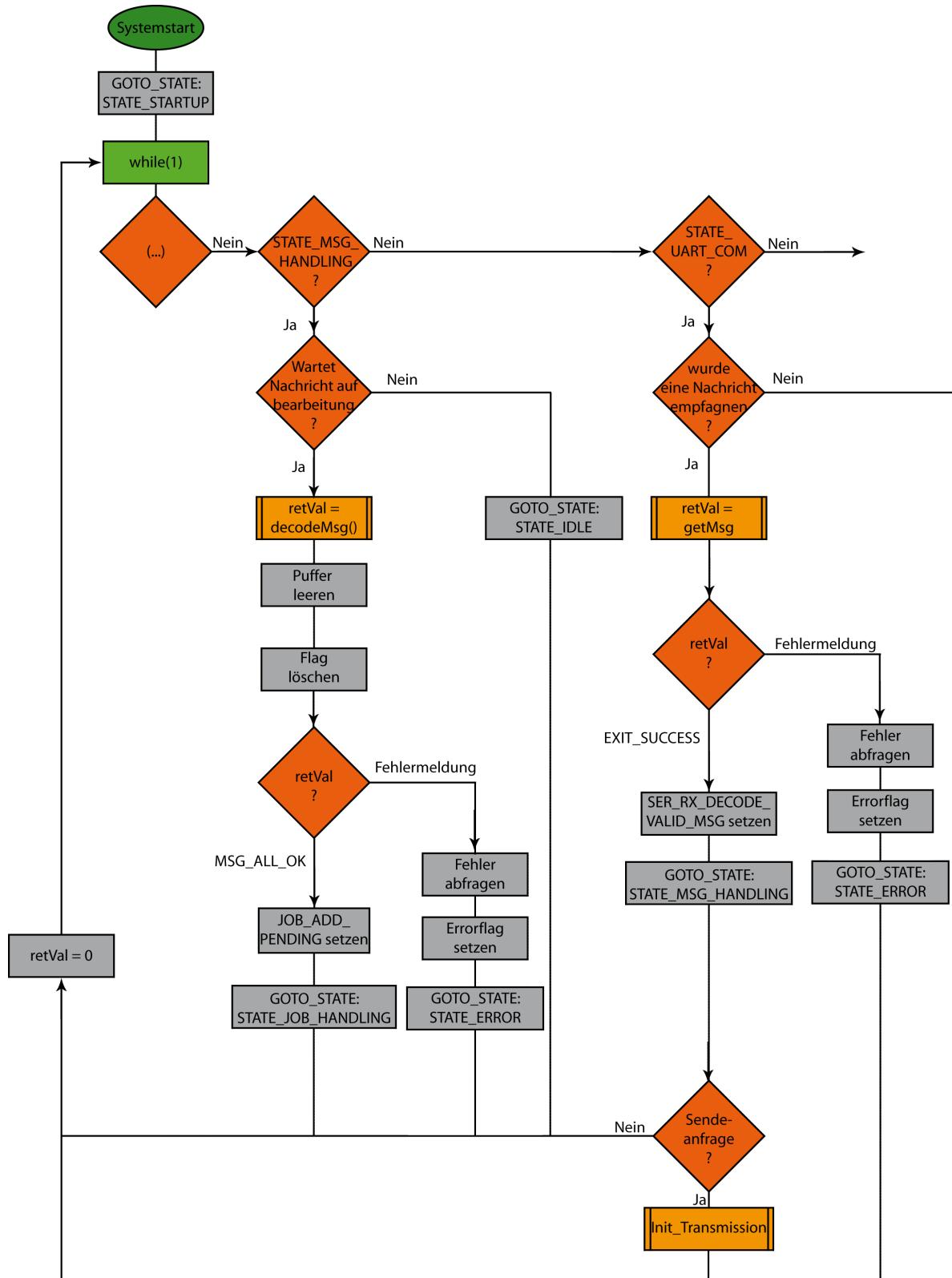


Abbildung 15: Ablaufdiagramm der main, teil 2

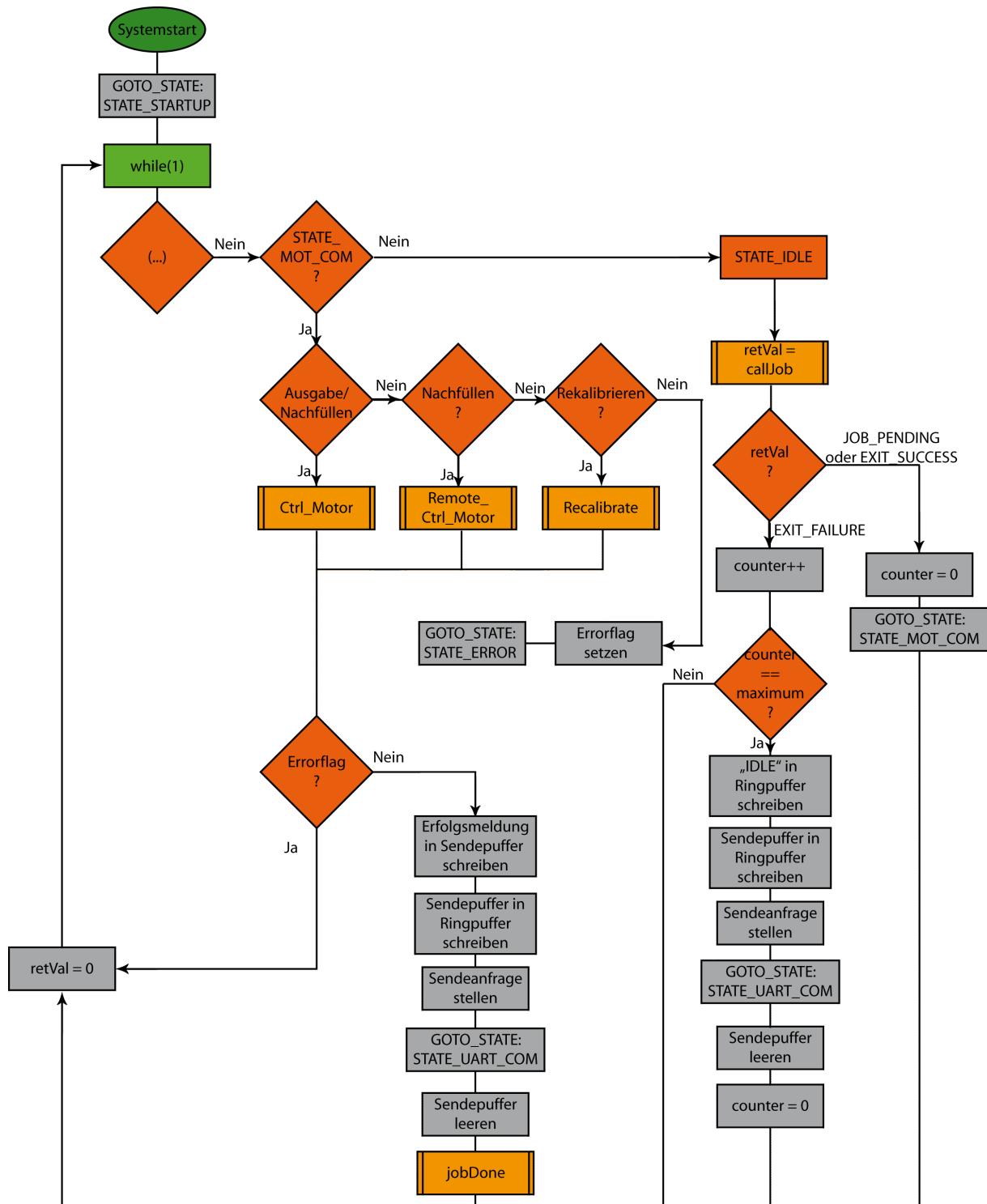


Abbildung 16: Ablaufdiagramm der main, teil 3

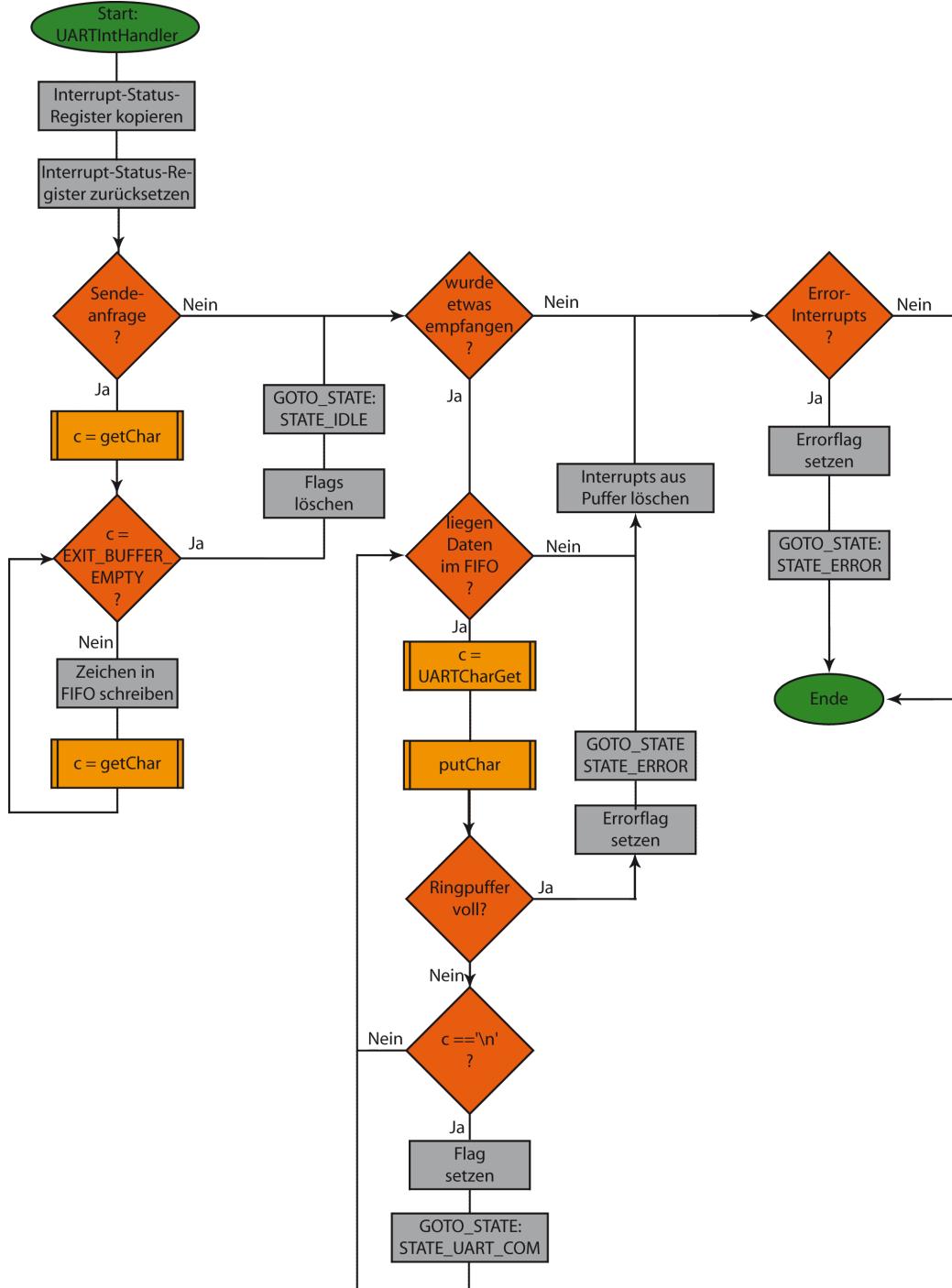


Abbildung 17: Ablaufdiagramm der Interrupt Service Routine für die UART-Schnittstelle

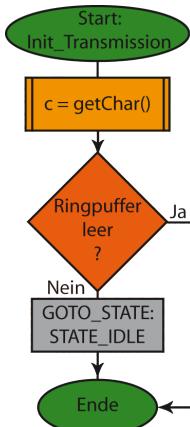


Abbildung 18: Ablaufdiagramm der Funktion `Init_Transmission`

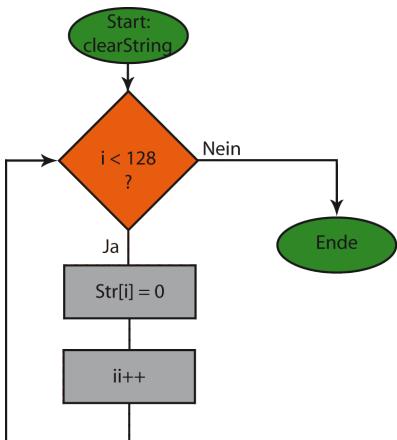


Abbildung 19: Ablaufdiagramm der Funktion `clearString`

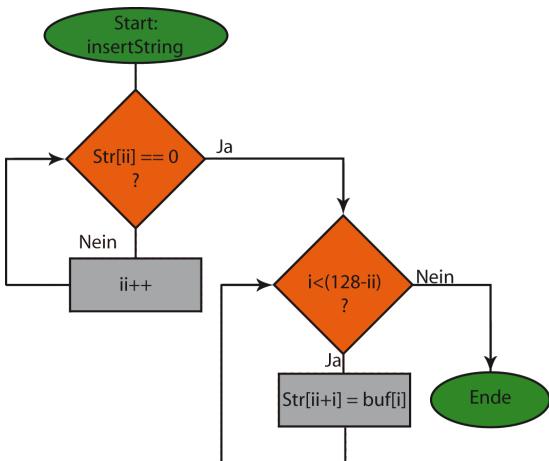


Abbildung 20: Ablaufdiagramm der Funktion `insertString`

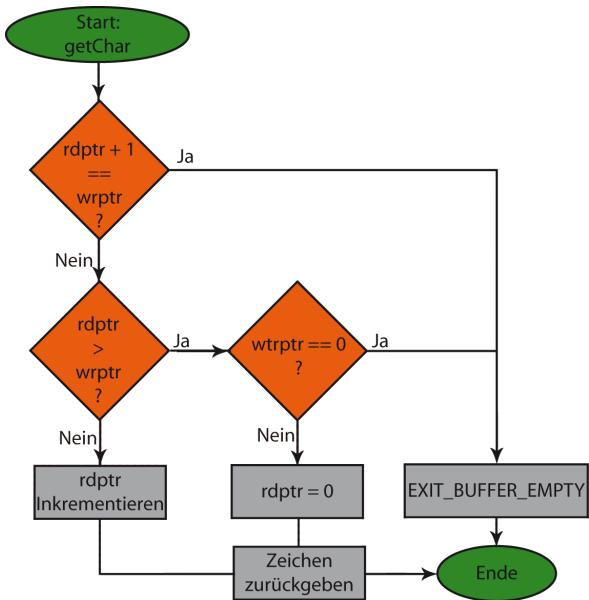


Abbildung 21: Ablaufdiagramm der Funktion `getChar`

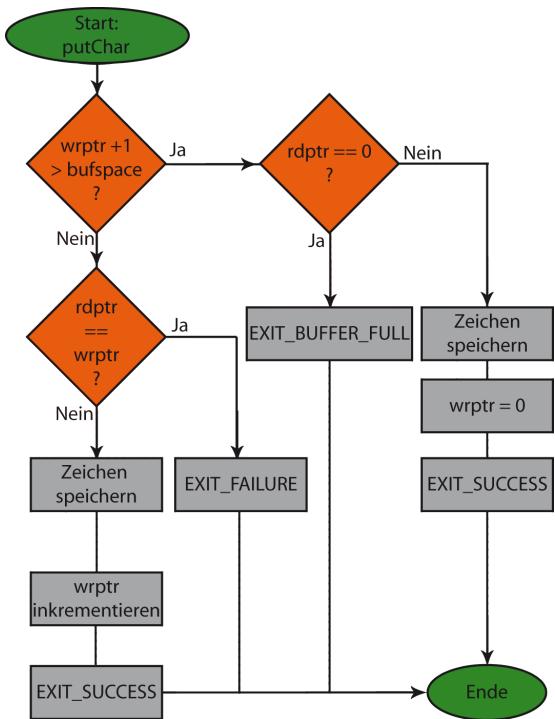


Abbildung 22: Ablaufdiagramm der Funktion `putChar`

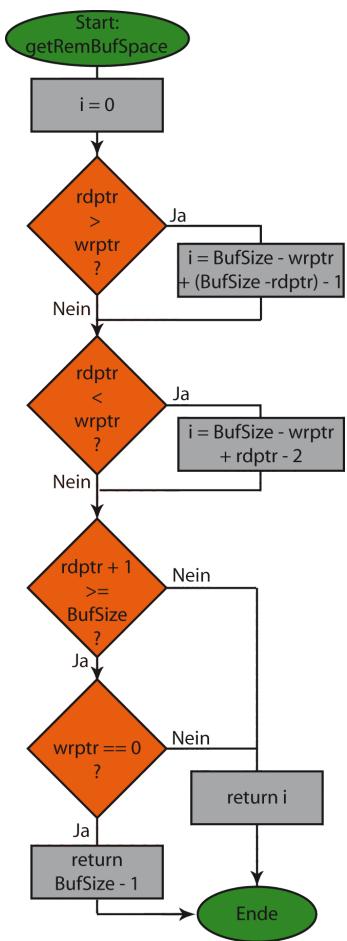


Abbildung 23: Ablaufdiagramm der Funktion getRemBufSpace

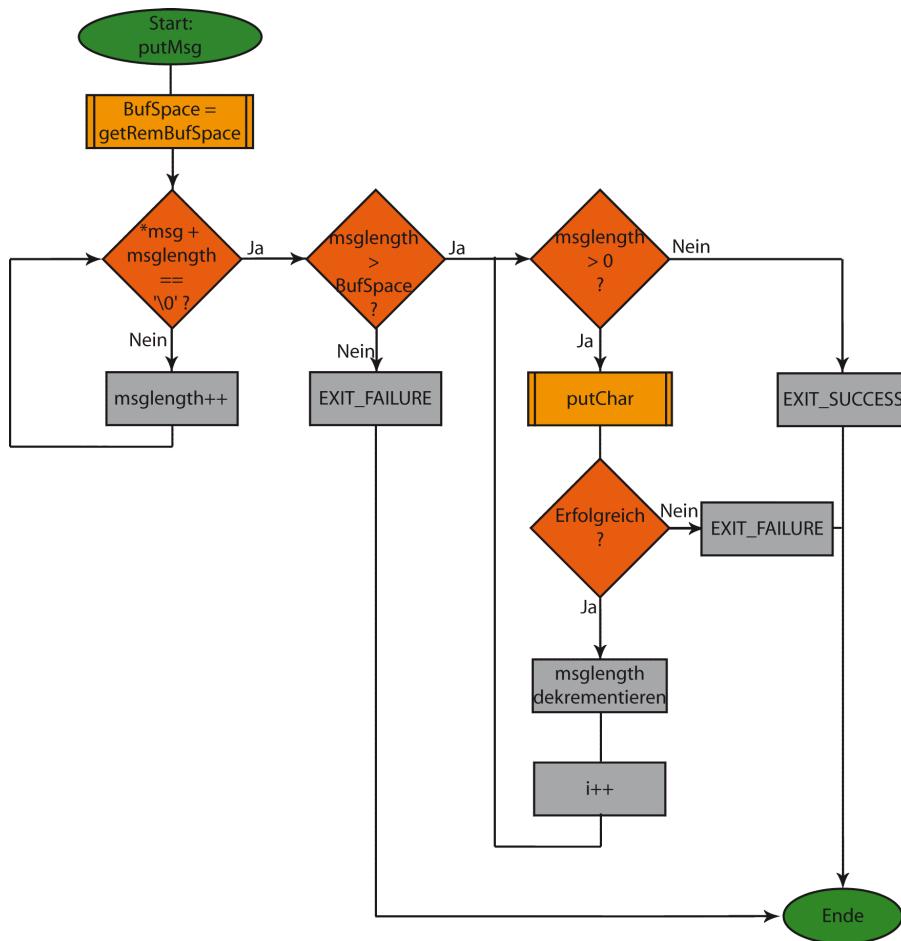


Abbildung 24: Ablaufdiagramm der Funktion putMsg

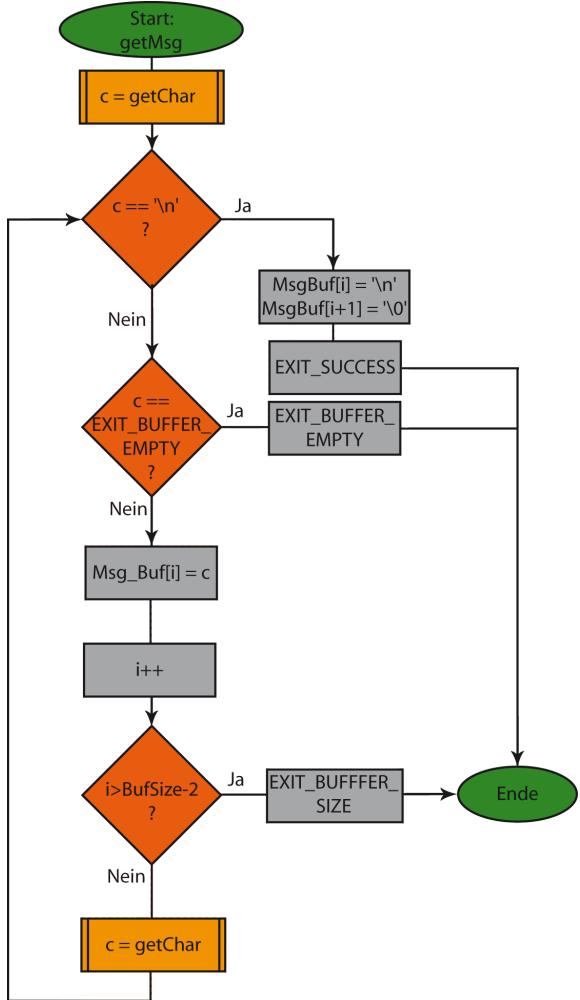


Abbildung 25: Ablaufdiagramm der Funktion `getMsg`

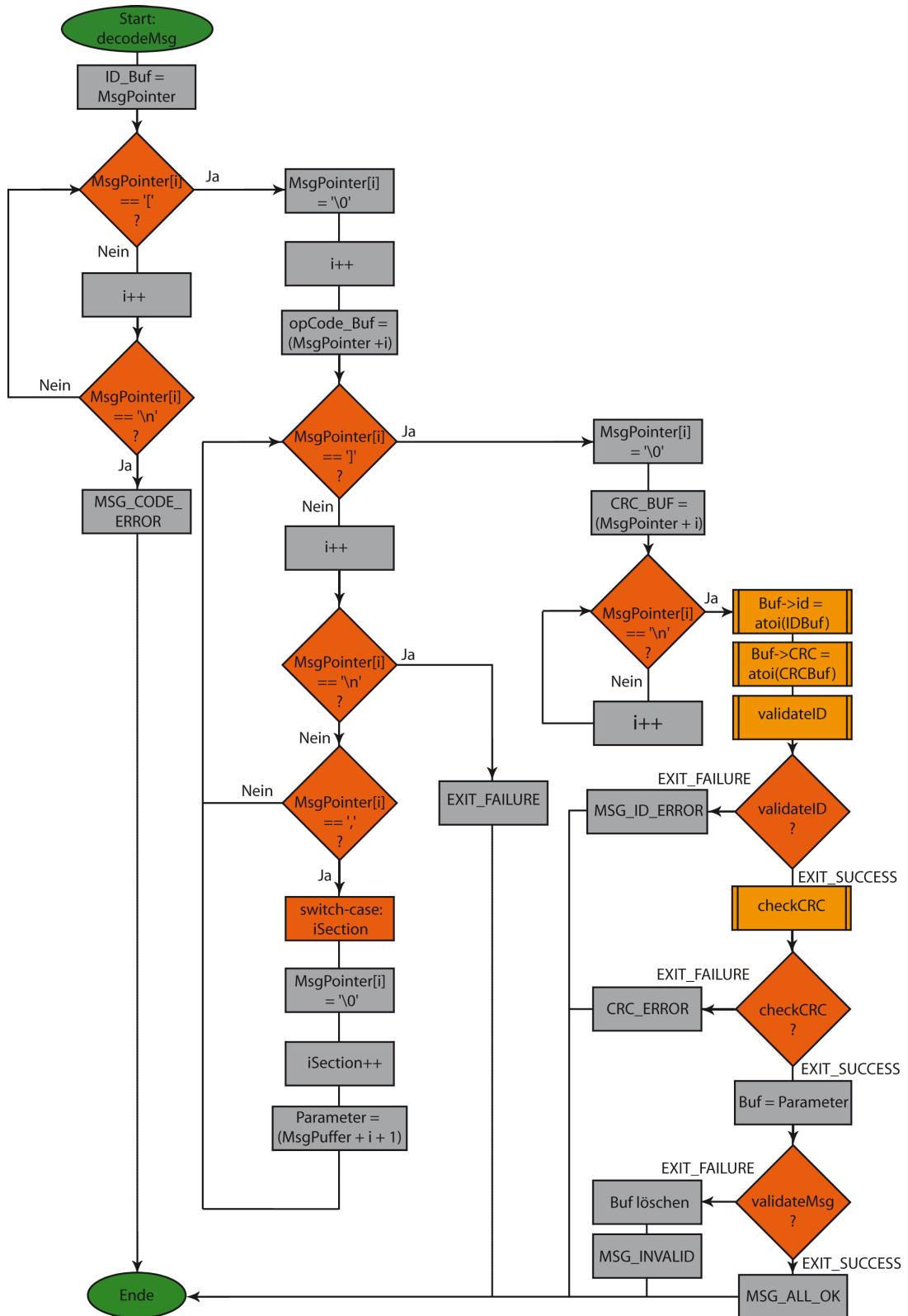


Abbildung 26: Ablaufdiagramm der Funktion decodeMsg

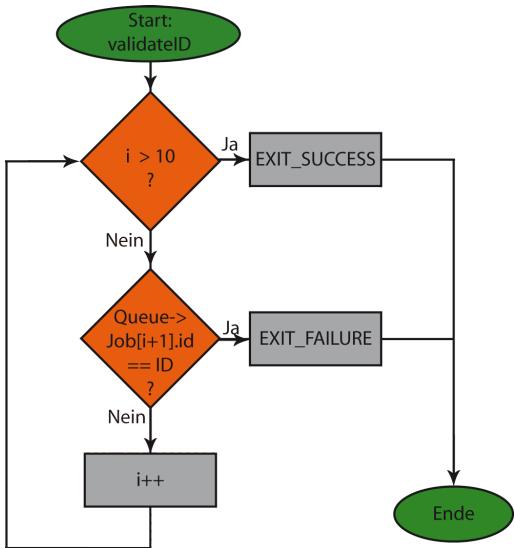


Abbildung 27: Ablaufdiagramm der Funktion validateID

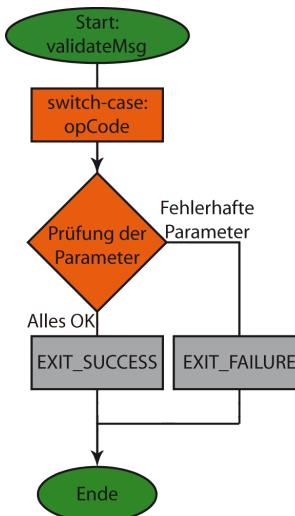


Abbildung 28: Ablaufdiagramm der Funktion validateMsg

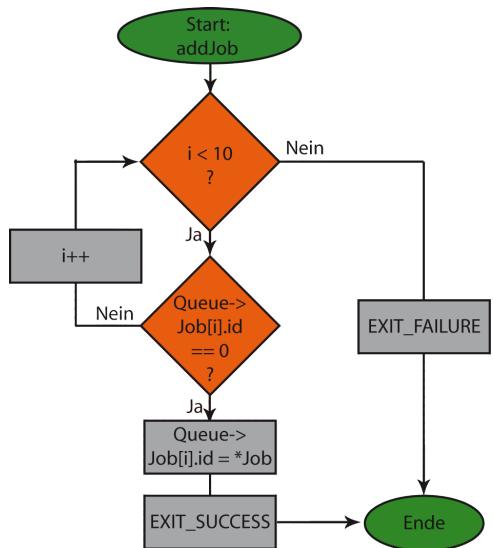


Abbildung 29: Ablaufdiagramm der Funktion addJob

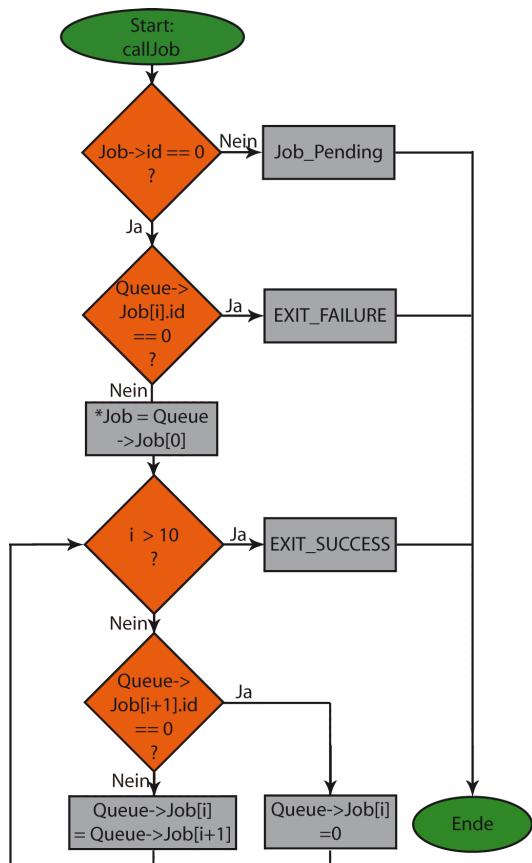


Abbildung 30: Ablaufdiagramm der Funktion callJob

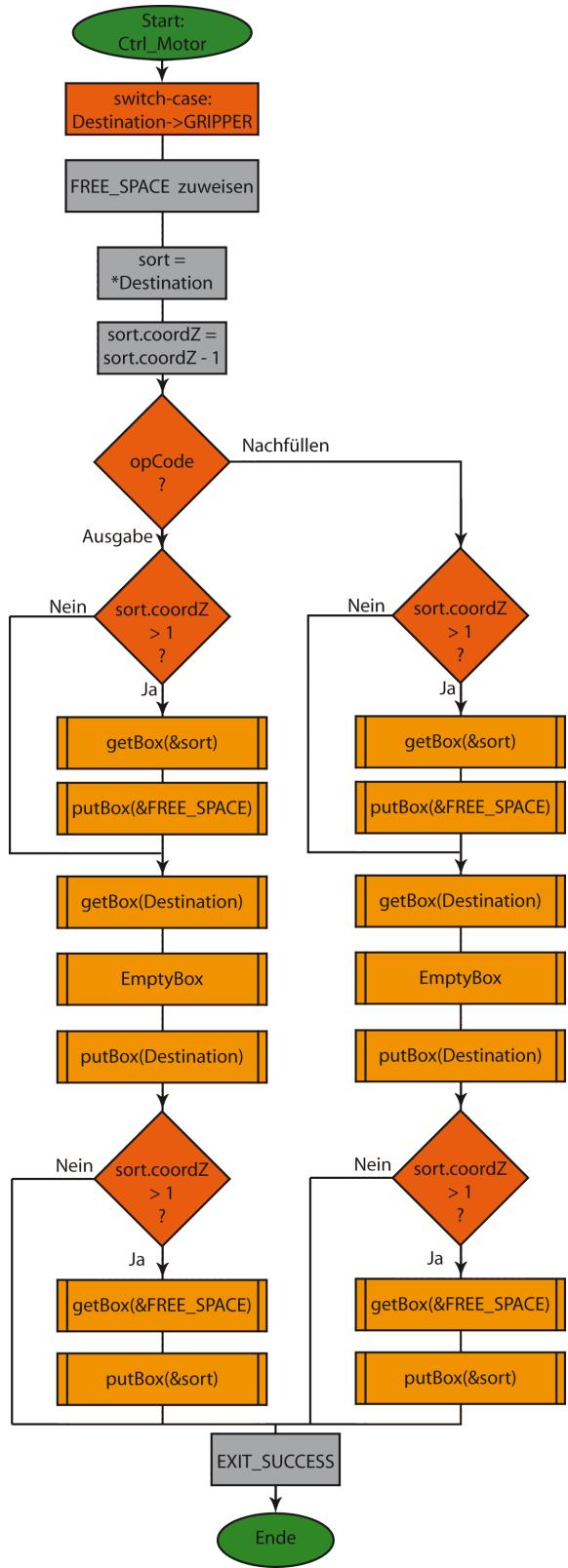


Abbildung 31: Ablaufdiagramm der Funktion Ctrl_Motor

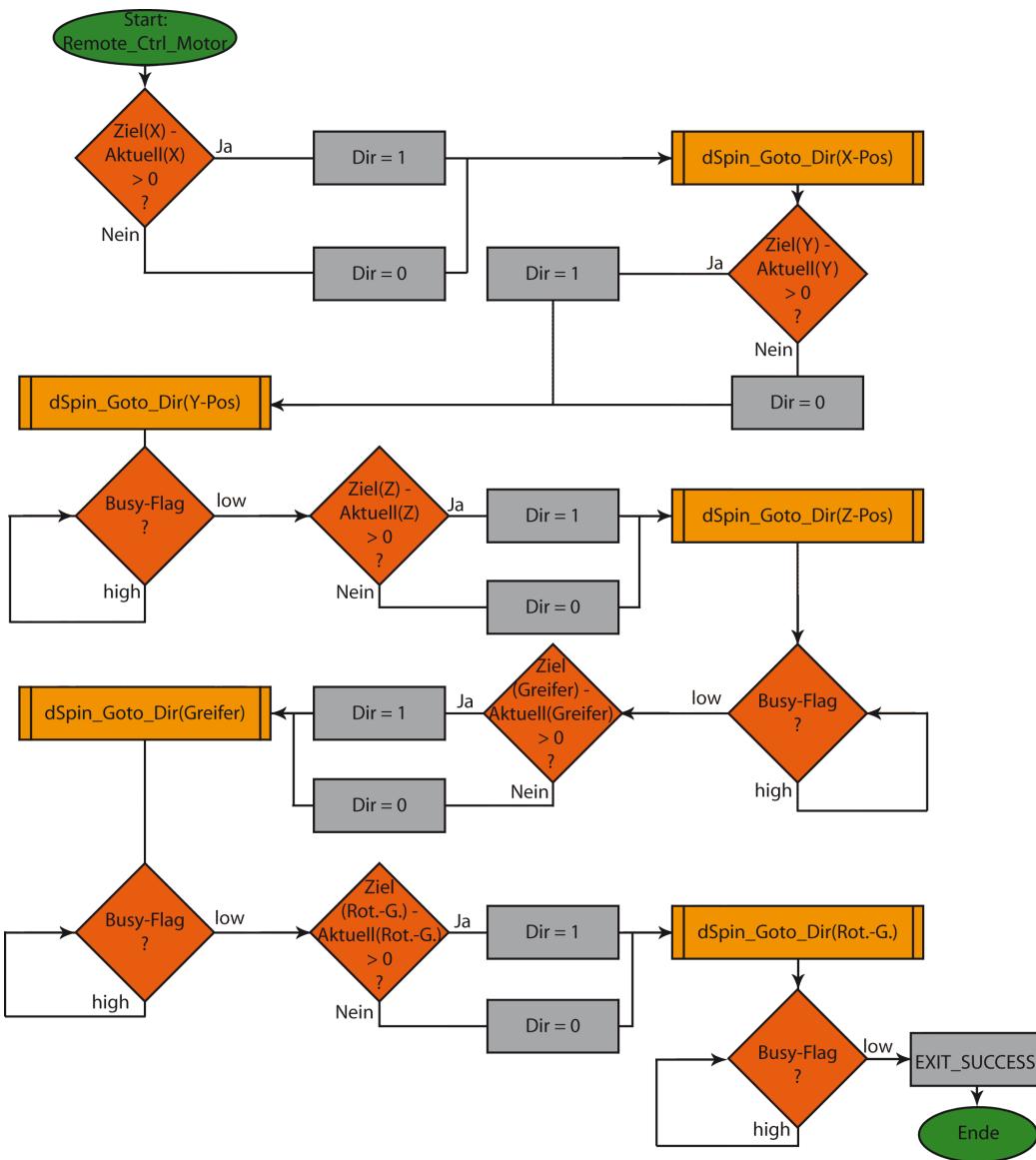


Abbildung 32: Ablaufdiagramm der Funktion Remote_Ctrl_Motor

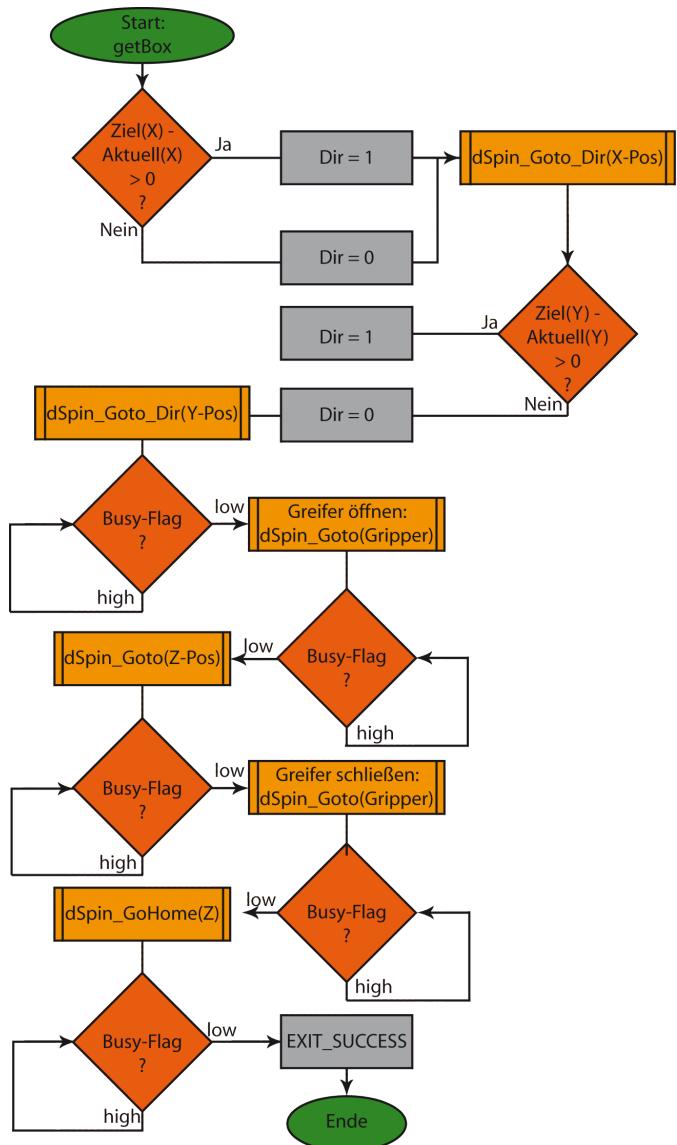


Abbildung 33: Ablaufdiagramm der Funktion getBox

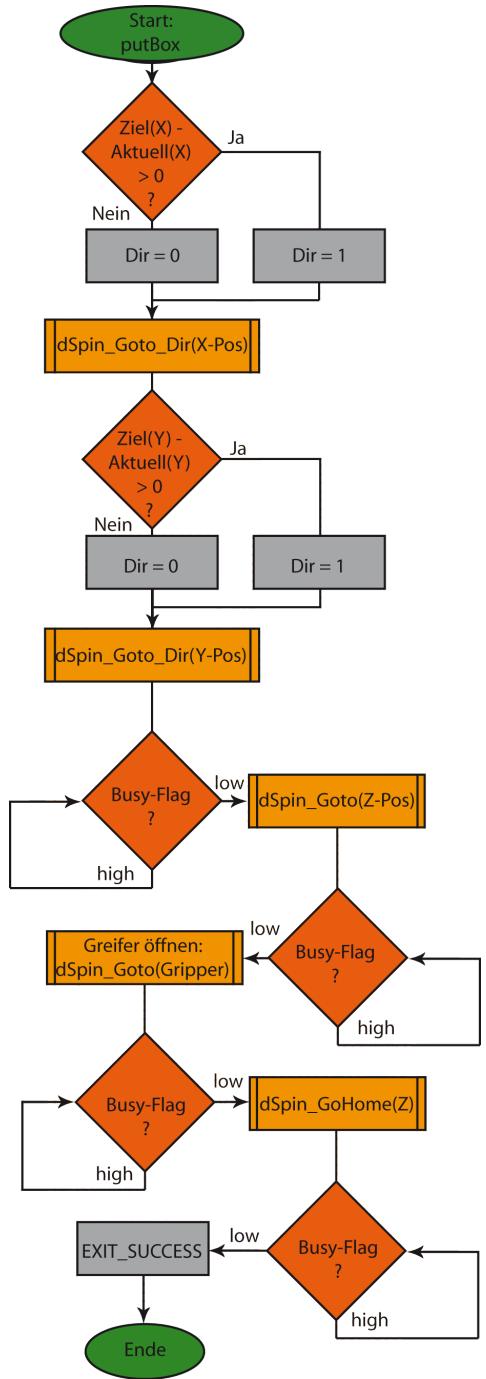


Abbildung 34: Ablaufdiagramm der Funktion putBox

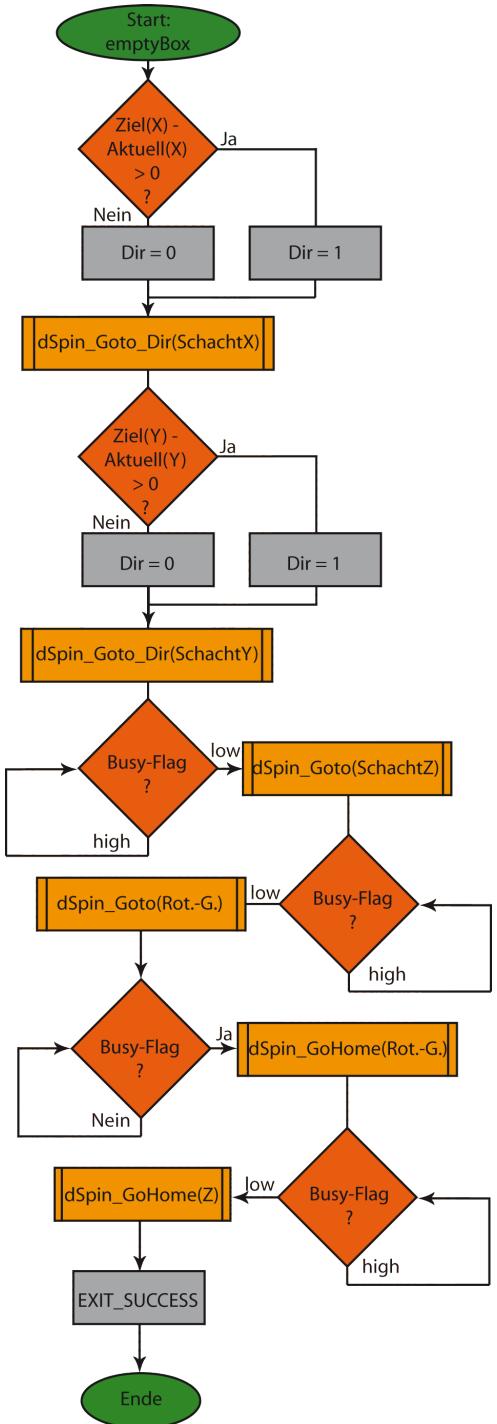


Abbildung 35: Ablaufdiagramm der Funktion EmptyBox

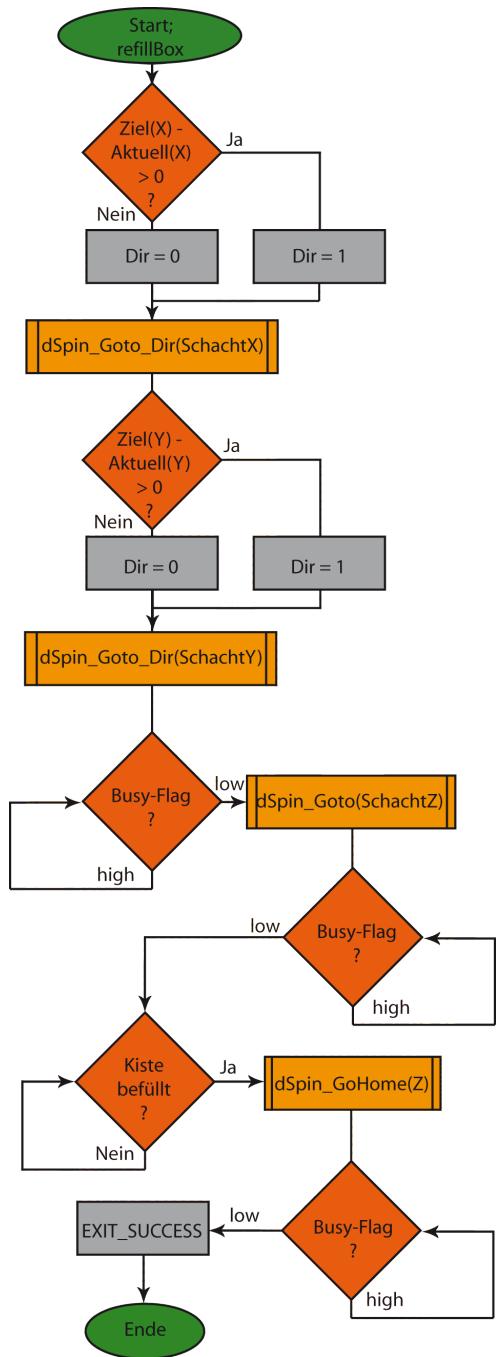


Abbildung 36: Ablaufdiagramm der Funktion RefillBox