



**Hochschule
Koblenz**
University of Applied Sciences

Fachbereich Elektro- und Informationstechnik



Max-Planck-Institut
für
Radioastronomie

Bachelorarbeit

im Bachelorstudiengang Mechatronik

Automatisierung eines 4-Meter-Radioteleskops im Azimut, basierend auf dem Linux Betriebssystem Ubuntu

**von Sebastian Schweikert
Matrikel-Nr.: 521875**

Hochschulbetreuer: Prof. Dr. Ross

Institutsbetreuer: Dr. Keller

Institut: Max-Planck-Institut für Radioastronomie

Eingereicht am: 22. September 2016

Eidesstattliche Erklärung

Hiermit bestätige ich, dass die vorliegende Arbeit von mir selbständig verfasst wurde und ich keine anderen als die angegebenen Hilfsmittel – insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen – benutzt habe und die Arbeit von mir vorher nicht in einem anderen Prüfungsverfahren eingereicht wurde.

Remagen, den 22. September 2016

Sebastian Schweikert

Inhaltsverzeichnis

Erklärung	i
Abkürzungsverzeichnis	iv
Abbildungsverzeichnis	vi
Listingsverzeichnis	vii
Formelverzeichnis	viii
1 Einleitung	1
2 Rahmenbedingungen	2
2.1 Max-Planck-Institut für Radioastronomie	2
2.2 Anforderungen	3
2.3 Entwicklungsstand des Projekts bei Beginn dieser Arbeit	4
2.4 Azimut und Elevation	5
2.5 Mechanische Bedingungen	6
2.5.1 Witterungseinflüsse	6
2.5.2 Winkelauflösung Radioteleskop	7
2.5.3 Benötigte Winkelgeschwindigkeit	7
3 Entwicklung und Realisierung der Automatisierung	9
3.1 Recherche geeigneter Antriebseinheiten	9
3.1.1 Profitracker XL	9
3.1.2 Berufsbildungszentrum Euskirchen	9
3.1.3 Astropeiler Stockert e.V.	10
3.2 Auswahl eines Antriebskonzepts	12
3.2.1 Getriebe	13
3.3 Winkelmessung	16
3.3.1 Zahnräder	16
3.3.2 Schenkelfeder	20
3.3.3 Drehgeber	20
3.3.4 Referenzpunkt	22
3.3.5 Endlagenschalter	22
3.3.6 Problemstellung	22
3.4 Akquise der ausgewählten Baugruppen	24
3.4.1 Bosch Rexroth	24
3.4.2 Antriebseinheit	25
3.4.3 Kraftübertragung	26
3.4.4 Auswahl Winkelmesssystem	26
3.4.5 Referenzmessung und Endlagendetektion	27
3.5 Konstruktion der mechanischen Befestigung	28
3.5.1 CAD-Modell der mechanischen Befestigung	28
3.5.2 Aufbau mechanischer Befestigung	29
3.6 Software Konzept	31

3.7	Inbetriebnahme des Testmoduls von Bosch Rexroth und Kommunikation mit dem C-Programm	32
3.8	Inbetriebnahme Regeleinheit	35
3.9	Software Projektierung	39
3.9.1	Implementierung der Regeleinheit und das Interface im LAN unter C	41
3.9.2	Implementierung von UDP und Multicast in Python	45
3.9.3	Grafische Benutzeroberfläche	48
3.10	Inbetriebnahme des Azimutantriebs am 4-Meter-Radioteleskop	51
4	Ergebnis	52
4.1	Fazit	52
4.2	Ausblick	54
	Literaturverzeichnis	55
A	Anhang	57

Abkürzungsverzeichnis

AG	Aktiengesellschaft
BZE	Berufsbildungszentrum Euskirchen
CAD	Computer-Aided Design
EMV	Elektromagnetische Verträglichkeit
Fa	Firma
GmbH	Gesellschaft mit begrenzter Haftung
GUI	Graphical User Interface
IP	International Protection
IP	Internet Protocol
IT	Informationstechnik
JSON	JavaScript Object Notation
LabVIEW	Laboratory Virtual Instrumentation Engineering Workbench
LAN	Local Area Network
LTS	Long Term Support
NN	Normalnull
PC	Personal Computer
PDF	Portable Document Format
POSIX	Portable Operating System Interface
TCP	Transmission Control Protocol
TTL	Transistor-Transistor-Logik
UDP	User Datagram Protocol
V _{ss}	Volt _{Spitze-Spitze}
XML	Extensible Markup Language

Tabellenverzeichnis

1	Auswahl Antriebssystem	25
---	----------------------------------	----

Abbildungsverzeichnis

1	100-Meter-Radioteleskop	2
2	Konzept	3
3	(a) CAD-Modell und (b) Technische Zeichnung des 4-Meter-Radioteleskops	4
4	Astronomisches Koordinatensystem	5
5	Die Himmelskugel	5
6	Windangriffsfläche	6
7	Fotovoltaik Anlage des BZE	10
8	Antriebe der Fotovoltaikanlage	10
9	Demontierter Profitracker CL des Astropeiler Stockert e.V.	11
10	Konzept Azimutantrieb	12
11	Demontierter Elevationsantrieb	13
12	Flankenspiel	16
13	Evolventenverzahnung	17
14	Elevationsantrieb 100-Meter-Radioteleskop in Effelsberg	17
15	CAD-Modell des federverspannten Zahnrads	18
16	Schnittdarstellung federverspanntes Zahnrad mit Drehgeber	19
17	Drehfeder	20
18	TTL-Signal Drehgeber Heidenhain	21
19	Befestigung Antriebseinheit und Messsystem	28
20	Entwurf gesamter Azimutantrieb mit Abdeckung	29
21	Aufbau Azimut Antrieb mit Winkelmesssystem	30
22	Federverspanntes Zahnrad	30
23	Software Konzept	31
24	Testmodul Bosch Rexroth	32
25	Externen Drehgeber anlernen	35
26	Implementieren der digitalen Eingänge	36
27	Implementieren der Referenzfahrt	37
28	Referenzieren der Endlagen	38
29	Implementieren der mechanischen Übersetzung	38
30	Software Entwurf	40
31	GUI Parametereingabe	49
32	GUI Parameterausgabe	50

Listings

1	Parameterübergabe an Regeleinheit	42
2	Parallele Programmabarbeitung	43
3	JSON Implementierung	44
4	UDP und Import Qt-Designer	46
5	Multicast Base	47
6	Multicast Main	48
7	C-Programm	83
9	Python Multicast	88
8	Python UDP	90

Formelverzeichnis

1	Strömungswidstand	6
2	c_w Wert	6
3	Dichte Luft	7
4	Windkraft	7
5	Drehmoment Windkraft	7
6	Drehmoment Windkraft	7
7	Auflösungsvermögen Radioteleskop	7
8	Auflösungsvermögen Radioteleskop	7
9	Auflösungsvermögen Radioteleskop	7
10	Übersetzung Teleskop zu Getriebe	14
11	Getriebeübersetzung	14
12	Getriebeübersetzung zu Motor	14
13	Gesamtübersetzung	14
14	Drehzahl	14
15	Umdrehungsdauer	14
16	Übersetzung Drehgeber	21
17	Strichzahl Drehgeber	21

1 Einleitung

Das Max-Planck-Institut für Radioastronomie in Bonn beschäftigt sich mit der Erforschung des Welt- raums. Für diese Wissenschaft ist ein Radioteleskop mit einem Spiegeldurchmesser von 100 Meter im Einsatz, welches in dem Eifeldorf Effelsberg steht. Der Parabolspiegel hat die Funktion, einfallende Strahlung aus dem Weltall zu bündeln und diese mittels sehr empfindlichen Messsystemen auszuwer- ten.

Das Telekommunikationsunternehmen Deutsche Telekom AG hat dem Observatorium einen 4-Meter- Parabolspiegel zur Verfügung gestellt. Dieser Spiegel wurde ursprünglich in der Nachrichtentechnik als Kopfstation, für das ehemalige Kabelfernsehen, verwendet, welcher für die Kommunikation mit geostationären Satelliten bestimmt war. Diesen will das Radioobservatorium umbauen und für radio- astronomische Messungen verwenden. Für die ehemalige Anwendung wurde die Positionierung über zwei Gewindespindeln, in Azimut und Elevation,¹ manuell auf einen Satelliten eingestellt und in dieser Position arretiert. Für den neuen Einsatzzweck wird ein System benötigt, welches bestimmte Koordinaten für die Astronomie positionieren kann, eine hohe Winkelauflösung besitzt sowie eine Nachführung des Spiegels, um die Erdrotation zu kompensieren. Aus diesem Grund besteht für das Radioobservatorium Effelsberg ein großes Interesse ein Antriebs- kombiniert mit einem Winkelmess- system zu entwickeln, damit sich das Teleskop im Azimut sowie in der Elevation frei bewegen kann. Die Entwicklung dieses Radioteleskops hat zwei grundlegende Zielsetzungen:

Zum einen soll es als Testsystem für das 100-Meter-Radioteleskop verwendet werden. Mit welchem neue Komponenten und Geräte getestet und erprobt werden können. Zudem besteht die Möglichkeit Softwareerprobungen in kleinerem Maßstab zu testen, ohne dabei den realen Messbetrieb zu unter- brechen. Hinzu kommt die Evaluation neuer Empfänger- sowie Infrastruktur Hardware, bevor diese im großen Teleskop eingesetzt werden. Zum anderen finden radioastronomische Messungen für die studentische Ausbildung am 100-Meter-Radioteleskop statt. Mit dem 4-Meter-Teleskop würde die Möglichkeit bestehen, Studenten die radioastronomische Beobachtung zunächst an einem kleineren Modell näher zu bringen, wodurch bei dem großen System mehr Messzeit gewonnen werden kann und die Fehlerwahrscheinlichkeit der Messungen verringert wird.

Diese Arbeit wird sich mit der Problemstellung der Automatisierung eines 4-Meter-Parabolspiegels im Azimut, basierend auf dem Betriebssystem Ubuntu, beschäftigen. Es werden Konzepte zum An- trieb des Teleskops entwickelt, welche für die Umwelteinflüsse der Umgebung ausgelegt sein müssen. Beispielsweise bietet ein Parabolspiegel in dieser Dimension eine große Angriffsfläche für Windlas- ten, hinzu kommen Witterungseinflüsse wie Temperaturschwankungen und Niederschlag. Für solche Gegebenheiten muss der Antrieb und das Winkelmesssystem ausgelegt und ausreichend dimensio- niert werden. Darüber hinaus benötigt die radioastronomische Messung eine sehr präzise Winkel- auflösung. Nur wenige Bogensekunden Toleranz führen dazu, dass die Messung an einem Him- melskörper vorbeigeht. In dieser Arbeit werden die Möglichkeiten zum mechanischen Antrieb sowie der Winkelmessung aufgezeigt und daraus eine geeignete Applikation projiziert.

Infolge der Automatisierung des 4-Meter-Radioteleskops wird sich mit der Ansteuerung der Antriebs- einheit im Azimut beschäftigt. Die Vorgabe dieses Projekts besteht darin, dass dem Teleskop die Para- metervorgaben der Positionierungen sowie das Auslesen des Winkels im Bereich des LAN ermöglicht werden soll. Für diese Kommunikation ist eine Client-Server-Architektur angedacht, welche mit den Protokollen UDP und Multicast ein netzwerkbasierendes Kontrollsystem aufbaut. Ein Benutzer soll mit Hilfe des Kommunikationsdienstes UDP die Position des Radioteleskops vorgeben. Das Teleskop wird ausgerichtet und mit Hilfe eines Multicast-Routings werden die Parameter an eine Teilnehmer- gruppe im Netzwerk übermittelt.

¹Die Begrifflichkeiten Azimut und Elevation werden in Kapitel 2.4 erläutert.

2 Rahmenbedingungen

2.1 Max-Planck-Institut für Radioastronomie

Die Max-Planck-Gesellschaft zur Förderung der Wissenschaft e.V. ist eine unabhängige und gemeinnützige Forschungsorganisation, welche als Nachfolge der Kaiser-Wilhelm-Gesellschaft 1948 gegründet wurde. Im Mittelpunkt dieser Forschungseinrichtung steht die erkenntnisorientierte und anwendungs offene Grundlagenforschung in den Naturwissenschaften, der Medizin, genauso wie in den Geistes- und Sozialwissenschaften. Eine dieser Forschungseinrichtungen ist das Max-Planck-Institut für Radioastronomie, welches eine Außenstelle in dem Eifelort Effelsberg hat. Das Radioobservatorium ist im Besitz eines 100-Meter-Radioteleskops, welches bis heute eines der beiden größten voll beweglichen Radioteleskope der Erde ist (vgl. Abb. 1). Dieses wird für die Messungen im Frequenzbereich zwischen 300 MHz bis 96 GHz eingesetzt. Die Radioastronomie beschäftigt sich mit der Untersuchung von Radiowellen aus dem Kosmos. Solche Strahlungen lassen sich mit einem Radioteleskop, in Verbindung mit äußerst empfindlichen Empfangsanlagen, nachweisen. Radioastronomische Messungen ergänzen die Beobachtungen der optischen Messinstrumente. Da im Gegensatz zur optischen Beobachtung, die Radioastronomie nicht durch Wolken aus Staub im Kosmos behindert wird. Vor diesem Hintergrund war es beispielsweise möglich, die genaue Struktur unserer Milchstraße zu bestimmen oder vorher unbekannte Himmelskörper wie Pulsare zu entdecken.²



Abbildung 1: 100-Meter-Radioteleskop³

²vgl. [Max-Planck-Flyer 2014] und [Vershuur 1988, S.17]

³[Max-Planck 2007]

2.2 Anforderungen

Die Vorgabe des Instituts, an die vorliegende Arbeit, besteht darin, ein Antriebs- sowie ein Winkelmesssystem im Azimut zu entwickeln, welches für die mechanischen Belastungen bzw. die Winkelgenauigkeit radioastronomischer Messungen hinreichend dimensioniert ist. Über das interne Netzwerk des Instituts sollen astronomische Koordinaten an einen Linux Rechner übertragen und diese mittels eines C-Steuerprogramms in Azimut- und Elevationskoordinaten aufgeteilt werden. Diese Parameter sollen über Ethernet an eine Regeleinheit weitergeleitet werden. Der Regler wird das Radioteleskop positionieren und durch eine Regelung die benötigte Genauigkeit sicherstellen. Ist das Teleskop positioniert, soll der Regler die aktuelle Position zurück an das Steuerprogramm übertragen, welches anschließend die Daten an eine Teilnehmergruppe im Netzwerk sendet. Zur Veranschaulichung ist dies in Abb. 2 skizziert.

Die Rotationsgeschwindigkeit des Teleskops ist unkritisch, solange die Positionierung schneller als die Winkelgeschwindigkeit der Erdrotation⁴ ist. Angedacht ist eine Umdrehungsdauer von circa einer Minute, damit die Positionierung verschiedener Koordinaten nicht zu viel Zeit in Anspruch nimmt. Jedoch benötigt das System eine sehr hohe Positionier- sowie Wiederholgenauigkeit, um exakte Messungen durchführen zu können. Hinzu kommen die Witterungsverhältnisse: Da das Teleskop in der Natur steht, muss dieses gegen Witterung geschützt werden. Die Randbedingungen sind Regen, Schnee sowie hohe und tiefe Temperaturen. Ein weiteres Risiko besteht bei der Windlast auf das Radioteleskop. Ein Parabolspiegel dieser Größe hat eine nicht zu vernachlässigende Angriffsfläche.

In der Nähe des Aufstellorts des Radioteleskops befindet sich ein Keller. Es ist angedacht den Schaltschrank in diesen zu verlegen. Somit wären diese Bauteile vor Witterungseinflüssen geschützt und hätten konstante Umgebungsbedingungen. Für diesen Zweck müssen die Kabellängen der Versorgungs- und Datenleitungen ausgelegt werden.

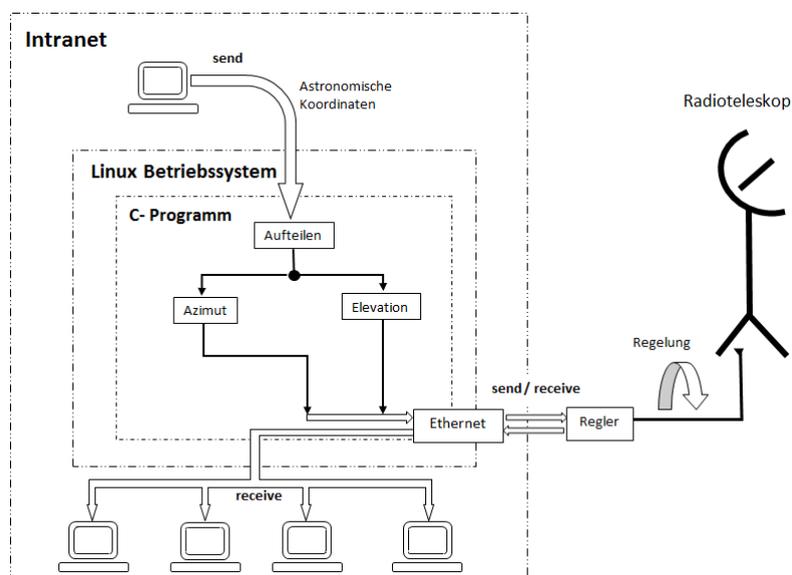


Abbildung 2: Konzept

⁴ Rotationsgeschwindigkeit der Erde: $\omega = 7,29 \cdot 10^{-5} \text{ 1/s}$ [Unsöld u. Baschek 2005, S.28]

2.3 Entwicklungsstand des Projekts bei Beginn dieser Arbeit

Vor Beginn dieser Bachelorarbeit haben sich zwei Bachelorarbeiten mit dem Umbau von einer Kopfstation zu einem Radioteleskop befasst. Zunächst ein Maschinenbaustudent der Hochschule Bonn-Rhein-Sieg. In dieser Arbeit wurde das Teleskop vermessen und mit diesen Daten ein CAD-Modell erstellt. Zudem wurde sich mit der Positionierung in Elevation und Azimut beschäftigt. Für die Elevation wurde eine Kippvorrichtung entwickelt und zu dieser ein passender Elektrozyylinder bestellt, damit sich das Teleskop, in Elevation, von 0 - 90 Grad frei bewegen kann. Für die Positionierung in Azimut wurde ein Drehlager mit Zahnkranz an dem Radioteleskop befestigt, siehe Abb. 3.

Die zweite Bachelorarbeit hat sich mit der Softwareansteuerung des Elevationsantriebs beschäftigt. Im Zuge dieses Projekts wurde ein Servoverstärker der Jenaer Antriebstechnik erworben. Dieser Verstärker dient zur Positionierung des Elevationsmotors sowie zur Kommunikation einer übergeordneten Steuerung.

An die vorangegangenen Arbeiten knüpft diese Bachelorarbeit an. Um möglichst geringe Kosten und Arbeitsaufwand zu investieren, bot es sich an ein System zu entwickeln, bei welchem die vorhandenen Bauteile verwendet werden. In dieser Arbeit sollte die azimutale Drehung des Radioteleskops mechanisch, elektrisch sowie softwaremäßig projektiert werden. Für die elektrische Ansteuerung des Azimut Antriebs steht ein 2-Achs-Servoverstärker, Ecovario 114D⁵ Jenaer Antriebstechnik, zur Verfügung. Der Name lässt erkennen, dass dieser Verstärker zwei Achsen ansteuern und regeln kann. Eine Achse wird für den Elevationsantrieb verwendet und die andere ist für den Azimutantrieb gedacht.

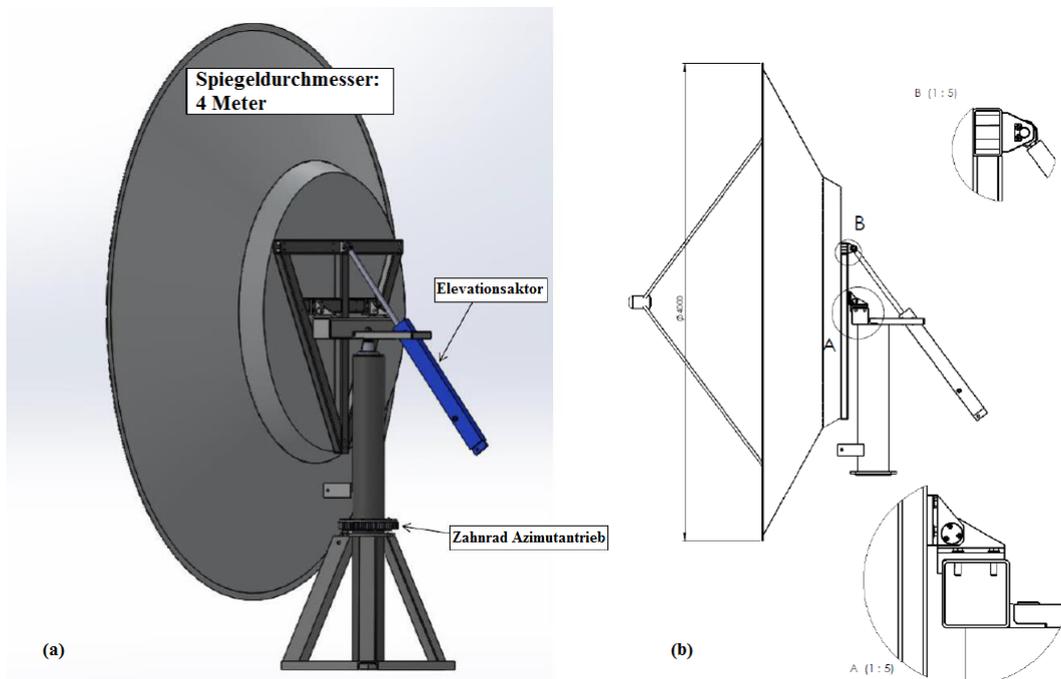


Abbildung 3: (a) CAD-Modell und (b) Technische Zeichnung des 4-Meter-Radioteleskops

⁵Datenblatt siehe Anhang

2.4 Azimut und Elevation

Zum besseren Verständnis dieser Arbeit möchte ich zunächst die Begriffe Azimut und Elevation erörtern. Will ein Beobachter von der Erdoberfläche den Kosmos betrachten, misst dieser den vertikalen und horizontalen Winkel. Für die vertikale Bezugsrichtung bietet sich das Lot, der Zenit, an. Für die horizontale Bezugsrichtung ist es von Vorteil die Nordrichtung zu verwenden, da man in diese näherungsweise, in Verlängerung der Erdachse stehend, den Polarstern findet. Somit erhalten wir zu einem Gestirn in Vertikalebene den Zenitwinkel ζ , auch Zenitdistanz genannt und den Höhenwinkel h , in unserem Fall wird dieser Elevation genannt (vgl. Abb. 4, 5). Der in Himmelsrichtung nach Norden ausgerichtete Winkel heißt Azimut α . Dieser geht von dem in die Horizontalebene projizierten Himmelsnordpol aus. Es wird rechtsläufig gezählt d.h. $90^\circ = \text{Ost}$; $180^\circ = \text{Süd}$; $270^\circ = \text{West}$. Teilweise bezieht sich der Azimut auch auf Süden. Im Zweifelsfall muss dies näher bezeichnet werden.⁶

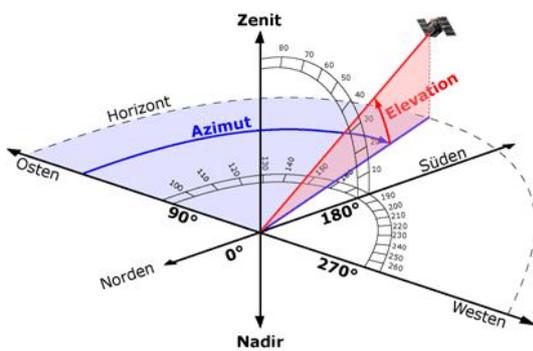


Abbildung 4: Astronomisches Koordinatensystem⁷

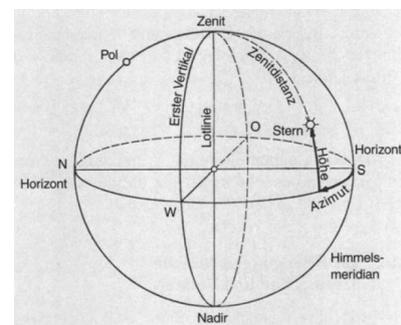


Abbildung 5: Die Himmelskugel⁸

⁶vgl. [Baumann 1994, S. 46]

⁷[INFOTRIP 2016]

⁸[Unsöld u. Baschek 2005, S. 11]

2.5 Mechanische Bedingungen

Bevor mit der Recherche für passende Bauteile begonnen werden kann, müssen die Rahmenbedingungen des Projekts geklärt sein und einige Berechnungen durchgeführt werden.

2.5.1 Witterungseinflüsse

Da das Radioteleskop im Freien betrieben werden soll, müssen die benötigten Bauteile auf diese Umwelteinflüsse ausgelegt sein. Somit ergeben sich bestimmte Randbedingungen wie beispielsweise Temperaturschwankungen, Niederschlag, Luftfeuchte und Windbelastungen. Die Betriebsbedingungen der einzelnen Bauteile können mittels Datenblätter verglichen werden. Die Windbelastung auf das Radioteleskop ist nicht zu vernachlässigen, mit einem Spiegeldurchmesser von vier Metern müssen die Bauteile auf solche Bedingungen ausgelegt sein.

Zum Auslegen der Kräfte für den Azimutantrieb wird die maximale Kraft benötigt, welche im Azimut auf die Drehachse wirkt. Näherungsweise wurde angenommen, dass der Wind seitlich den Spiegel angreift, vgl. Abb. 6. In dieser Stellung wirkt das maximale Drehmoment, in der azimutalen Drehung, auf das Radioteleskop. Die Windangriffsfläche sowie die Abstände zu der resultierenden Flächenkraft wurden aus dem CAD-Modell entnommen.

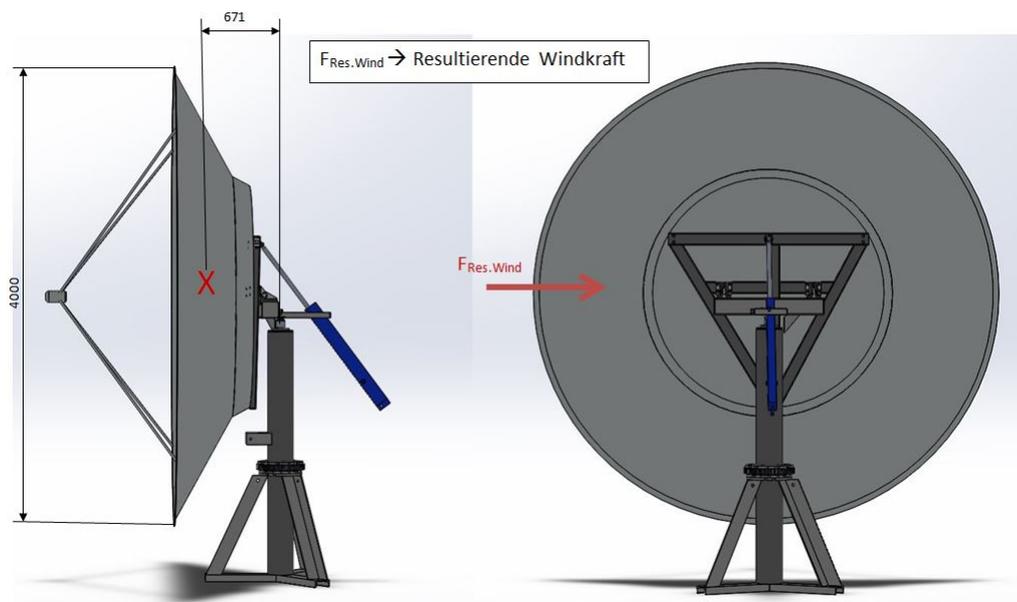


Abbildung 6: Windangriffsfläche

$$F = \frac{1}{2} \cdot c_w \cdot \rho \cdot v^2 \cdot A \quad [\text{Erbrechtu. a. 2008, S.93}] \quad (1)$$

c_w	Widerstandsbeiwert
ρ	Dichte des strömenden Stoffes
v	Strömungsgeschwindigkeit, \vec{v} senkrecht zur durchströmten Fläche A
A	Querschnittsfläche des umströmten Körpers

$$c_w = 0,34 \quad \text{Halbkugel ohne Boden} \quad [\text{Kurzweilu. a. 2008, S.88}] \quad (2)$$

$$\rho_{(20^{\circ}\text{C})} = 1,23 \text{ kg/m}^3 \text{ bei NN [Hau 2003, S.96]} \quad (3)$$

Die Windangriffsfläche des Parabolspiegels wurde aus dem CAD-Modell entnommen und entspricht einer Fläche von $A = 5,5 \text{ m}^2$. Für die maximale Windgeschwindigkeit wurde eine Windstärke von 12 Beaufort angenommen. Dies entspricht einem Orkan mit einer Windgeschwindigkeit von $v = 33 \text{ m/s}$.⁹

$$F = \frac{1}{2} \cdot 0,34 \cdot 5,5 \text{ m} \cdot 1,23 \text{ kg/m}^3 \cdot (33 \text{ m/s})^2 = 1252,4 \text{ N} \quad (4)$$

$$M = F \cdot r \text{ [Erbrechtu. a. 2008, S.86]} \quad (5)$$

$$M = 1252,4 \text{ N} \cdot 0,671 \text{ m} = 840,36 \text{ Nm} \approx 850 \text{ Nm} \quad (6)$$

2.5.2 Winkelauflösung Radioteleskop

Allein die angreifenden Kräfte zu kennen, welche auf das Teleskop wirken, reicht für eine Auslegung der Bauteile nicht aus. Zudem muss bekannt sein, welche Winkelauflösung dieses für eine radioastronomische Messung benötigt. Wenn die Winkelauflösung zu gering ist, können mit diesem System keine brauchbaren Messungen durchgeführt werden. Das Auflösungsvermögen eines Radioteleskops wird definiert als kleinster Winkelabstand zweier Sterne, die man gerade noch voneinander trennen kann. Für diesen Fall ergibt sich folgende Formel.¹⁰

$$\phi = 1,22 \cdot \frac{\lambda}{d} \cdot \frac{360^{\circ}}{2\pi} \quad (7)$$

Für d wird der Durchmesser des Radioteleskops eingesetzt. λ ist die zu empfangende Wellenlänge des Teleskops, in diesem Fall ist $\lambda = 2,2 \text{ cm}$ die kürzeste gemessene Wellenlänge. Die Konstante 1,22 ist ausgehend von der Verteilung der Strahlungsintensität bei einer kreisförmigen Apertur, welches ihr erstes Minimum bei $1,22 \cdot \lambda/d$ besitzt. Infolgedessen muss der Abstand zwischen zwei entfernten Punkten mindestens so groß sein wie der Winkel ϕ .

$$\phi = 1,22 \cdot \frac{0,022 \text{ m}}{4 \text{ m}} \cdot \frac{360^{\circ}}{2\pi} = 0,38^{\circ} \quad (8)$$

Somit ist das Auflösungsvermögen des Radioteleskops bekannt. Um in diesem Bereich auch Messungen durchführen zu können, wird ein Faktor von 10 als Sicherheit eingerechnet. Dieser Faktor wurde aus den Erfahrungen von dem 100-Meter-Teleskop empfohlen.

$$\phi = \frac{0,38^{\circ}}{10} \approx 0,04^{\circ} = 2,4' = 144'' \quad (9)$$

2.5.3 Benötigte Winkelgeschwindigkeit

Während einer radioastronomischen Beobachtung ist das Radioteleskop ständig in Bewegung, da dieses die Rotation der Erde kompensieren muss. Solange sich das Teleskop schneller drehen kann als die Erdrotation¹¹ ist die Rotationsgeschwindigkeit unkritisch. Jedoch muss vor jeder neuen Beobachtung das Teleskop zunächst in Richtung des Himmelskörpers positioniert werden. Damit dieser Vorgang nicht zu viel Zeit in Anspruch nimmt, ist es sinnvoll die Geschwindigkeit höher anzusetzen.

⁹vgl. [Hau 2003, S. 472]

¹⁰vgl. [Unsöld u. Baschek 2005, S. 130]

¹¹ Rotationsgeschwindigkeit der Erde: $\omega = 7,29 \cdot 10^{-5} \text{ 1/s}$ [Unsöld u. Baschek 2005, S.28]

Am Beispiel des 100-Meter-Radioteleskops dauert eine Umdrehung im Azimut knapp zwölf Minuten.¹² Vor dem Hintergrund, dass das 4-Meter-Teleskop um einiges kleiner ist und infolgedessen eine höhere Geschwindigkeit einfacher realisiert werden kann, ist eine Umdrehungsdauer von circa einer Minute angedacht.

¹² vgl. [Max-Planck 2016]

3 Entwicklung und Realisierung der Automatisierung

3.1 Recherche geeigneter Antriebseinheiten

3.1.1 Profitracker XL¹³

Ein passendes System bietet die Firma Egis GmbH in Offenbach an, welche die Möglichkeit bietet, den kompletten Parabolspiegel auf das System zu montieren. Dieses ist für einen Parabolspiegel von vier Meter ausgelegt und bietet die Möglichkeit, auch bei einer hohen Windlast noch zuverlässig zu arbeiten. Konstruiert wurde das Bauteil, um Satelliten zu tracken. Jedoch kann die Applikation auch für die Anwendung der Radioastronomie verändert werden. Der Winkelbereich im Azimut hat keine Begrenzung und die Elevation kann Winkel von 0° bis 90° positionieren. Die Wiederholgenauigkeit ist im Toleranzbereich. Zudem kann über ein Ethernet Interface dieses System direkt angesprochen werden. Dies kann über ein C-Programm implementiert werden. Ein Nachteil besteht darin, dass für dieses Konzept die komplette mechanische Befestigung neu konstruiert werden müsste. Da die Vorteile jedoch dominieren, wurde telefonisch Kontakt aufgenommen, um Fragen über die Applikation zu klären und ein Angebot zu erhalten. Mit einem Preis von 25.000 € würde der Rahmen dieses Projekts jedoch gesprengt, da zusätzlich alle vorhandenen Bauteile nicht mehr zu verwenden wären.

3.1.2 Berufsbildungszentrum Euskirchen

Das BZE ist im Besitz einer Fotovoltaikanlage, mit Sonnennachführung, siehe Abb. 7. Dies schien eine gute Möglichkeit zu sein, um neue Anregungen für dieses Projekt zu sammeln.

Es stellte sich jedoch heraus, dass diese Anlage dort nicht konstruiert wurde. Diese Konstruktion hat ein externes Unternehmen entwickelt. Daher konnten einige Fragen nicht beantwortet werden. Wichtig wäre beispielsweise die Windlast gewesen und welche Motoren für diese Lasten benötigt werden. Die Anlage, wie auf Abbildung 8 zu sehen, ist in der Elevation mit einem Elektrozyylinder realisiert. Dies ist sehr ähnlich mit dem momentanen Aufbau zum Antrieb der Elevation und bestätigt die Umsetzbarkeit im realen Betrieb. Der Azimutbereich wurde mit einem Spindelantrieb angesteuert. Diese haben jedoch ein großes Spiel und sind daher für radioastronomische Messungen nicht zu verwenden. Schon bei geringem Wind konnte beobachtet werden, dass sich das System im Azimutbereich leicht bewegt. Die Sonnennachführung der Anlage wird über die Lichtintensität gesteuert. Somit hat die Fotovoltaikanlage immer den bestmöglichen Wirkungsgrad. Aus diesem Grund wird dort keine Position gemessen, sondern nur nachgeführt.

¹³Datenblatt siehe Anhang



Abbildung 7: Fotovoltaik Anlage des BZE



Abbildung 8: Antriebe der Fotovoltaikanlage

3.1.3 Astropeiler Stockert e.V.

In der Nähe von Effelsberg befindet sich der Astropeiler Stockert e.V. Dieser Verein für Radioastronomie und Amateurfunk ist im Besitz zweier Radioteleskope: Das eine hat einen Durchmesser von zehn Metern, das andere hingegen einen von 25 Metern. Zudem wurde an dem Standort ein 3-Meter-Radioteleskop aufgebaut. Dort wird das Komplettsystem Profitracker genutzt. Vor diesem Hintergrund wurde das Radioteleskop besichtigt. Es stellte sich heraus, dass die Basisausführung des Systems verwendet wird. Das System wurde für dieses Projekt jedoch bereits im Vorhinein ausgeschlossen, da der Winkelbereich und die Kräfte für ein 4-Meter-Radioteleskop nicht ausreichend sind. Zudem ist das Getriebe nicht selbsthemmend und wird bei hohen Windstärken aus der Position gedrückt. Daher hat der Verein die Antriebseinheit teilweise demontiert und nach Verbesserungsmöglichkeiten gesucht, siehe Abb. 9. Der Azimutbereich wurde mittels eines Zahnrads über einen Motor angetrieben. Es war sehr gut zu erkennen, dass zwei Zahnräder, die ineinandergreifen nicht spielfrei laufen. Wenn am Motor oder an der Antriebsspindel gedreht wurde, war das Spiel gut zu erkennen. Die Elevation wurde über eine Spindel betrieben, jedoch sind diese nicht spielfrei. Infolgedessen ist die Genauigkeit dieses Systems für den 4-Meter-Spiegel nicht geeignet. Die Winkelmesseinheit ist im Motor integriert und somit konnte das Zahnradspiel nicht kompensiert werden. Der Vorteil dieses Systems besteht darin, dass die Winkelvorgabe direkt über TCP/IP mittels C vorgegeben werden kann.

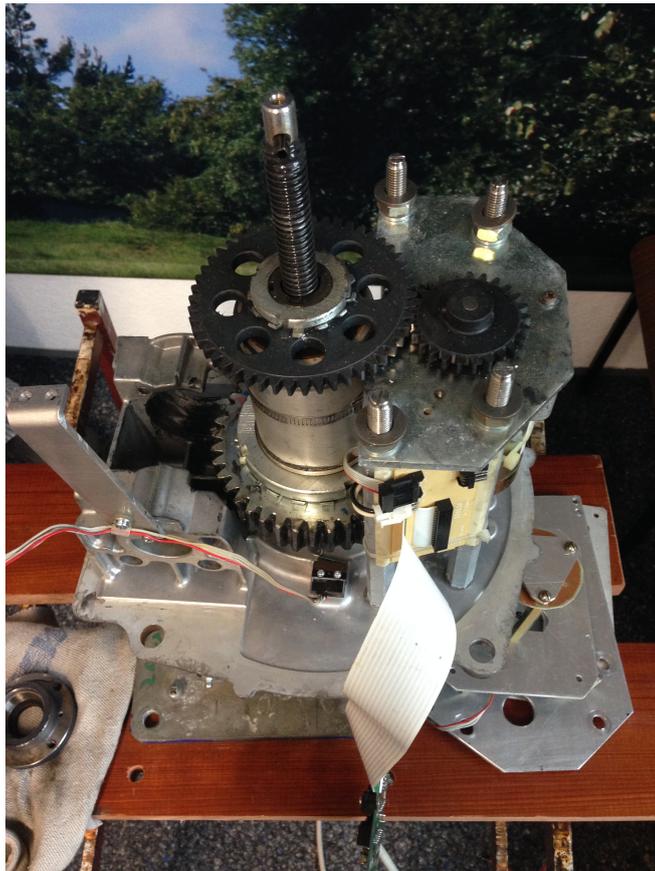


Abbildung 9: Demontierter Profitracker CL des Astropeiler Stockert e.V.

3.2 Auswahl eines Antriebskonzepts

Im Anschluss an die erste Ideenfindung, wurde zunächst der vorhandene Servoverstärker, Ecovario 114D,¹⁴ begutachtet. Dieser Verstärker bietet die Möglichkeit, zwei Achsen zu steuern und zu regeln. Aufgrund dessen können die Winkel der Elevation und des Azimuts über dieses Gerät gesteuert bzw. geregelt werden. Wie in der Aufgabenstellung gewünscht, kann die Regeleinheit über Ethernet mit dem Computer kommunizieren. Jedoch mit dem Nachteil, dass das Steuerprogramm, im Gegensatz zur Projektvorgabe, nur mittels LabVIEW realisiert werden kann. Da von dem Unternehmen Jenaer Antriebstechnik kein Interface mit C, zu Kommunikation mit ihren Regeleinheiten, zur Verfügung gestellt wird. In diesem Fall besteht das Problem zwischen dem Interface von LabVIEW und dem Steuerprogramm in dem LAN. Für die Positionsrückführung steht ein inkremental Encodereingang zur Verfügung. Dieser Encodereingang besitzt eine RS 422 Schnittstelle sowie die Möglichkeit mit einem 1 V_{ss} Signal betrieben zu werden. Zudem müssen die Bauteile für die mechanischen Bedingungen aus Kapitel 2.5 ausgelegt werden. Um das Teleskop mechanisch anzutreiben, ist an diesem ein Zahnkranz montiert. Mit dieser Grundlage wurde ein Antriebskonzept entwickelt, vgl. Abb. 10. Für die Kraftübertragung auf das Teleskop soll ein Zahnrad in den Zahnkranz eingreifen, welches zusätzlich mit einem Getriebe übersetzt werden soll. Mit Hilfe dieser Bauteile wird das zu leistende Drehmoment des Motors verringert.

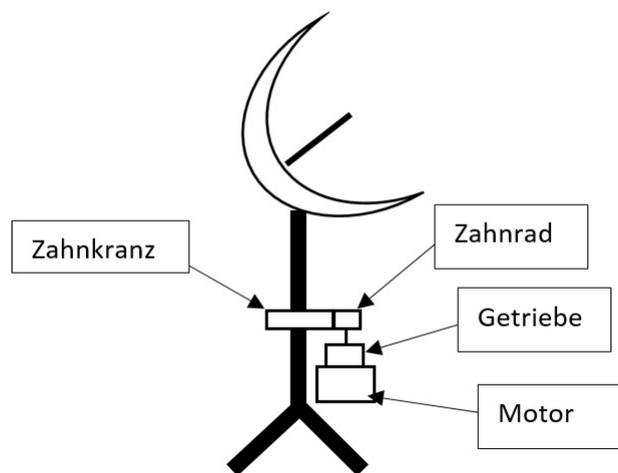


Abbildung 10: Konzept Azimutantrieb

Aufgrund der Vorgabe der Regeleinheit, wurde zunächst bei dem Unternehmen Jenaer Antriebstechnik, nach einem passenden Servomotor, recherchiert. Auf diese Weise sollten Kompatibilitätsprobleme zwischen Servoverstärker und Motor vermieden werden und eine schnelle Inbetriebnahme gewährleistet sein. Mit diesen Rahmenbedingungen wurde der Motor Ecostep, Baureihe 23s¹⁵ Jenaer Antriebstechnik, ausgewählt. Dieser ist der stärkste Motor, welcher mit dem Servoverstärker kompatibel ist. Problematisch ist jedoch, dass dieser nur ein Nenndrehmoment von 1,8 Nm zur Verfügung stellt. Aus diesem Grund ist es unabdingbar ein Getriebe zu verwenden. Dieses Getriebe benötigt eine Übersetzung, welche das Drehmoment des Motors erhöht, um die ermittelten Windlasten zu kompensieren.

¹⁴Datenblatt siehe Anhang

¹⁵Datenblatt siehe Anhang

Zunächst wurde sich mit der Inbetriebnahme des Elevationsantriebs beschäftigt. Dies bietet zum einen den Vorteil, dass dieser bereits an dem Radioteleskop befestigt werden kann und somit Referenzfahrten durchgeführt werden können. Zum anderen kann sich mit dem Servoverstärker auseinandergesetzt werden, da der Azimut- ebenso wie der Elevationsantrieb an diesem angeschlossen werden soll. Somit besteht die Möglichkeit, dass eine spätere Inbetriebnahme der Azimut Antriebseinheit schneller durchgeführt werden kann. Um den Spiegel in Betrieb zu nehmen, wurde der Aktor an dem Radioteleskop montiert. Der Servoverstärker wurde über Ethernet mit dem Computer verbunden. Mittels eines Programms der Jenaer Antriebstechnik besteht die Möglichkeit, die Zylinderstange des Aktors mit Hilfe des Computers verfahren zu können. Leider ließ sich der Aktor nicht ansteuern, aus diesem Grund wurde dieser demontiert und es fiel auf, dass dieser einen Wetterschaden hat. Der Aktor war von innen korrodiert und nicht mehr zu verwenden, siehe Abb. 11.



Abbildung 11: Demontierter Elevationsantrieb

3.2.1 Getriebe

Im Folgenden sind die benötigten Daten zur Ermittlungen eines Getriebes aufgelistet:

- Drehung Radioteleskop
 360° , in ca. einer Minute
- Windlast
 $M_W = 850 \text{ Nm}$
- Zähnezahl des Zahnkranz am 4-Meter-Radioteleskop
 $z = 80$
- Nenndrehmoment Motor¹⁶
 $M_M = 1,8 \text{ Nm}$
- Nenndrehzahl Motor
 $n = 750 \text{ min}^{-1}$

Zu dem Getriebe wird ein Zahnrad benötigt, welches die Kraftübertragung zwischen dem Zahnkranz des Teleskops und dem Getriebe herstellt. Es wurde ein Zahnrad ausgewählt, welches nur als grobe Abschätzung gedacht ist. Somit können die Dimension des Zahnrads sowie die Getriebeübersetzung überschlägig ausgelegt werden. Eine genauere Dimensionierung wird stattfinden, wenn der Motor

¹⁶Datenblatt siehe Anhang

mit passendem Getriebe gefunden ist. Daher wurde zunächst ein Zahnrad mit einer Zähnezahl von $z = 12$ festgelegt.

Mit Hilfe dieser Daten konnte die Übersetzung zwischen dem Zahnrad und des Radioteleskops berechnet werden. Die folgenden Formeln der Getriebeübersetzung wurden aus [Fischer u. a. 2008, S.259] entnommen.

$$i_z = \frac{z_{\text{Abtrieb}}}{z_{\text{Antrieb}}} = \frac{80}{12} = 6, \bar{6} \quad (10)$$

$$M_G = \frac{M_W}{i_z} = \frac{850 \text{ Nm}}{6, \bar{6}} = 127,5 \text{ Nm} \quad (11)$$

$$i_M = \frac{M_G}{M_{\text{nenn}}} = \frac{127,5 \text{ Nm}}{1,8} = 70,8 \bar{3} \quad (12)$$

$$i_{\text{ges}} = i_z \cdot i_M = 472, \bar{2} \quad (13)$$

$$n_{\text{ab}} = \frac{n_{\text{an}}}{i_{\text{ges}}} = 1,6 \text{ min}^{-1} = 0,027 \text{ s}^{-1} \quad (14)$$

$$T = \frac{1}{n_{\text{an}}} = 37 \text{ s} \quad (15)$$

Es wurde die Entscheidung getroffen, ein Planetengetriebe zu verwenden, da diese sehr spielarm sind. Ein passendes Getriebe wurde von der Firma Bosch Rexroth gefunden. Der Flansch des Motors und des Getriebes stimmen nicht überein, für diesen Fall müsste eine Platte gefertigt werden. Zudem besteht die Möglichkeit einer Übersetzung von lediglich $i = 40$. Mit diesen Werten wurden Berechnungen durchgeführt.¹⁷ Diese ergaben bei Nennbetrieb eine Umdrehungsdauer von 20 Sekunden sowie ein Nenn Drehmoment von 480 Nm. Es ist sofort zu erkennen, dass diese Konstruktion für eine Windstärke von 12 Beaufort nicht ausreichend ist.

¹⁷ siehe Anhang

Aus diesem Grund stellen sich folgende Fragen:

Muss das Teleskop bei einem Orkan betrieben werden?

Dieses Radioteleskop dient zur Schulung und Entwicklung neuer Messsysteme. Wenn der Wind zu stark wird, kann das Teleskop in eine sichere Position gefahren und dort mechanisch arretiert werden. Für die Anwendungszwecke ist es nicht von Bedeutung, dass das Teleskop bei jeder Wetterlage zur Verfügung steht. Beispielsweise werden bei dem 100-Meter-Radioteleskop, Messungen nur bis zu einer Windgeschwindigkeit von 15 m/s durchgeführt. Wenn diese Grenze überschritten wird, fährt das Teleskop in eine sichere Ausgangslage.

Welche Windstärken treten speziell am Standort Effelsberg auf?

Recherchen ergaben eine jährliche Höchstgeschwindigkeit von 18,3 m/s in Bad Münstereifel.¹⁸ Dieser Wert bezieht sich auf eine Windgeschwindigkeit, die mindestens 10 Minuten anhält. Für diesen Fall wurden die Kräfte auf das Teleskop nochmals berechnet.¹⁹ Somit würde eine Windlast von rund 260 Nm auf das Teleskop wirken. Mit diesen Randbedingungen wäre der Motor ausreichend dimensioniert. Jedoch sollte bei starkem Wind das Teleskop in eine sichere Position gefahren werden. Eine Fortsetzung des Messbetriebs sollte erst bei weniger Wind stattfinden.

Kann der Motor mit einer geringeren Drehzahl betrieben werden, um ein größeres Drehmoment zu erhalten und/ oder besteht die Möglichkeit, einen anderen Motor bzw. Getriebe zu verwenden?

In dem Datenblatt²⁰ des Motors ist die Motorkennlinie definiert. Wird die Drehzahl des Motors verringert, erhöht sich das Drehmoment. Da die Drehzahl des Radioteleskops unkritisch ist, wäre dies eine mögliche Variante. Zunächst wurde jedoch mit den Nennangaben des Motors gerechnet, um Referenzwerte zu erhalten. Wenn sich im Laufe dieser Arbeit für diesen Motor entschieden wird sowie ein Getriebe ausgewählt und ein passendes Zahnrad gefunden ist, müssten diese Bauteile nochmals explizit aufeinander ausgelegt werden. Gleichzeitig wurde die Recherche dahingehend fortgesetzt, ob die Möglichkeit besteht, einen anderen Motor oder ein Getriebe zu verwenden. Dies führte jedoch zu keinem befriedigendem Ergebnis. Es scheiterte am Servoverstärker, welcher für Motoren, die solche Drehmomente übertragen sollen, nicht ausgelegt ist.

¹⁸vgl. [Eichen 2016]

¹⁹siehe Anhang

²⁰siehe Anhang

3.3 Winkelmessung

Servomotoren, dazu zählt auch der ausgewählte Motor Ecostep 23s, besitzen einen integrierten Drehgeber. Dieser wird, wie der Motor, an der Regeleinheit angeschlossen. Über diesen kann das System die Position bzw. den Winkel des Motors ermitteln. Der Regeleinheit wird von dem PC ein Winkel vorgegeben und der Motor würde diesen anfahren. Da der Motor für den Servoverstärker ausgelegt ist, ist in diesem Fall auch der interne Drehgeber mit dem System kompatibel. Jedoch besteht der Nachteil, dass der Motor mit einem Getriebe übersetzt wird. Da Getriebe nie ganz spielfrei sind, würde die Genauigkeit des Systems beeinträchtigt werden. Das zurzeit ausgewählte Getriebe hat ein Verdrehspiel von maximal 9 Winkelminuten und wäre somit außerhalb der geforderten Toleranz des Teleskops.²¹ Auf diese Ungenauigkeit würde sich das Übersetzungsspiel von dem Zahnkranz auf das Zahnrad aufaddieren, welches die Genauigkeit der radioastronomischen Messung beeinträchtigt und daher für diesen Fall nicht verwendbar ist. Aus diesem Grund muss nach einer Messmethode geforscht werden, um das Spiel zu kompensieren.

Eine Möglichkeit besteht darin, einen externen Drehgeber zu verwenden, welcher über ein Zahnrad direkt an den Zahnkranz des Teleskops montiert wird. Dies bietet den Vorteil, dass die Ungenauigkeit des Getriebes entfallen würde, jedoch die Problematik, dass die Zahnräder nicht spielfrei zueinander laufen können.

3.3.1 Zahnräder

Für die Kraftübertragung mittels Zahnräder besteht immer der Nachteil, dass diese ohne ein Spiel verklemmen würden und somit keine Drehbewegung möglich wäre. Zudem besteht ein Problem im Zuge schwankender Temperaturen. Wäre keine Toleranz zwischen den Zahnrädern, entstehen durch Wärmeausdehnung Spannungen im Werkstoff. Diese können zu Rissen oder zur Zerstörung des Zahnrads führen. Im Gegensatz dazu entsteht bei Wärmeschrumpfung ein Spiel. In Abbildung 12 ist gut zu erkennen, dass so lange ein Zahnrad in eine Richtung bewegt wird, dieses weitgehend spielfrei läuft. Bei präzise gefertigten Zahnrädern greift der nächste Zahn in das Flankenprofil ein, bevor der Vorige den Kontakt unterbricht. Wenn sich jedoch die Laufrichtung ändert, muss zunächst das Spiel der Rückflanke überwunden werden.

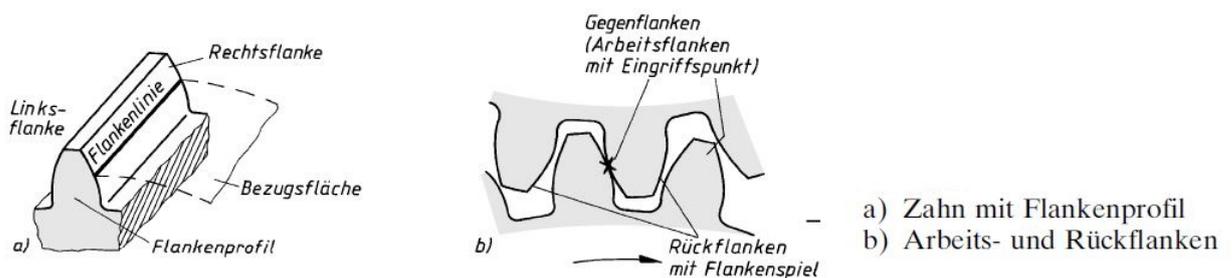


Abbildung 12: Flankenspiel²²

Daher besteht ausschließlich die Möglichkeit, die Zahnräder konstruktionstechnisch annähernd spielfrei zu halten. Spielarme Zahnräder werden gehärtet und geschliffen, um eine hohe Passgenauigkeit

²¹vgl. Kapitel 2.5.2 Winkelauflösung Radioteleskop

²²[Wittel u. a. 2015, S. 720]

zu erzielen. Zudem wird in diesem Fall eine Evolventenverzahnung verwendet. Bei dieser Verzahnung ist das Flankenprofil rund, womit beim Abrollen das Zahnrad einer Evolventen folgt. Somit haben die Zähne immer Kontakt mit dem anderen Zahnrad. Es gibt keinen Zeitpunkt an dem sich die Zahnräder nicht berühren und daher auch für diesen Fall weniger Spiel, vgl. Abb. 13. Komplettspielfrei können diese Zahnräder jedoch nicht sein.

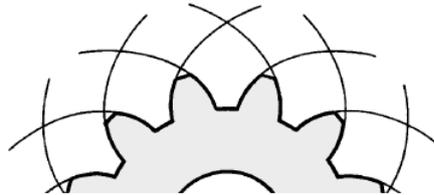


Abbildung 13: Evolventenverzahnung²³

Um eine Zahnradverbindung komplett spielfrei zu halten, bestehen nicht sehr viele Möglichkeiten. Eines dieser Verfahren wurde bei dem Elevationsantrieb am 100-Meter-Radioteleskop in Effelsberg angewandt. Dort werden zwei Motoren gleichzeitig betrieben. Der eine Motor läuft im Linkslauf und der andere dagegen im Rechtslauf. Der Motor der stärker angetrieben wird, setzt sich durch. Auf diese Weise bewegt sich die Elevation in die eine oder andere Richtung. Mit diesem Verfahren wird jeweils die linke sowie die rechte Flanke des Zahnrads mit einer Kraft beaufschlagt und es kann kein Spiel entstehen. Abbildung 14 zeigt den Elevationsantrieb des 100-Meter-Radioteleskops, die Nummern 1 und 2 sind die jeweiligen Motoren, die gegeneinander laufen.

Dieses Verfahren ist für den Anwendungsfall des 4-Meter-Radioteleskops jedoch überdimensioniert.



Abbildung 14: Elevationsantrieb 100-Meter-Radioteleskop in Effelsberg

²³[Wittel u. a. 2015, S. 728]

Eine weitere Möglichkeit das Spiel eines Zahnrads zu kompensieren, besteht darin zwei Zahnräder federverspannt zu betreiben. Dahinter steckt die gleiche Idee wie bei dem 100-Meter-Radioteleskop. Es werden zwei Zahnräder verwendet. In diese werden Nuten gefräst und eine Drehfeder montiert. Durch ein Verdrehen dieser Zahnräder wird die Feder gespannt. Diese verspannten Zahnräder werden an den Zahnkranz des Radioteleskops montiert. Somit sind immer beide Flanken des Zahnrads mit dem Zahnkranz verbunden und es besteht kein Spiel zwischen den Zahnrädern. Zudem gleicht dieses Verfahren das Problem von schwankenden Temperaturen aus. Dies wurde zur besseren Veranschaulichung und zur späteren Herstellung in einem CAD-Modell konstruiert, siehe Abb. 15.

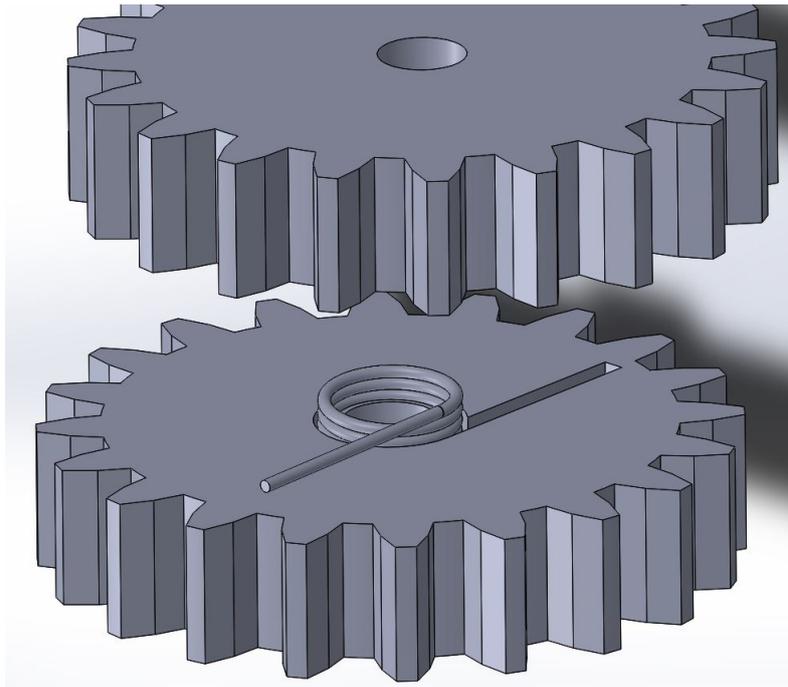


Abbildung 15: CAD-Modell des federverspannten Zahnrads

Dieses Verfahren funktioniert jedoch nicht zwischen Getriebe und Motor. Dort treten hohe Drehmomente auf, eine Drehfeder, die in dieses Zahnrad passt und zudem ein Drehmoment von mehreren hundert Newtonmetern aufnehmen kann, existiert nicht. Aus diesem Grund ist das Verfahren für den Antriebsstrang nicht anwendbar. Jedoch muss das Spiel zwischen Motor und der Kraftübertragung zum Radioteleskop gering bleiben, um eine genaue Positionierung zu gewährleisten und eine spätere Regelung des Motors effektiv betreiben zu können. Somit wird für diesen Fall ein Zahnrad mit einer Evolventenverzahnung verwendet, welches gehärtet und geschliffen ist.

Dies ist ein weiterer Grund für eine externe Winkelmessung. Der Drehgeber, welcher an dem Zahnrad angeflanscht wird, läuft nur mit dem Teleskop. Auf das Zahnrad wirkt annähernd keine Kraft. Somit ist es möglich ein federverspanntes Zahnrad zu verwenden. Die Feder in den beiden Zahnrädern muss lediglich die Kraft aufwenden, um die Flanken der Zahnräder gegeneinander zu verspannen und diese Spannung bei der Drehung des Drehgebers beizubehalten. Damit besteht die Möglichkeit, die Drehbewegung des Teleskops über den Drehgeber spielfrei zu erfassen.

Federverspannte Zahnräder in diesen Dimensionen sind keine Normteile und finden in der Industrie kaum Anwendung. Der Aufwand solche Zahnräder zu fertigen steht oftmals nicht im Verhältnis zum

gewonnen Nutzen. Aus diesen Gründen werden solche Zahnräder auf dem Markt nicht angeboten. Daher wurde ein solches Zahnrad von mir konstruiert, siehe Abb. 15. Aus dem CAD-Modell muss eine technische Zeichnung²⁴ erstellt werden. Diese wird der Werkstatt übergeben und dort gefertigt.

Ein wichtiges Kriterium für Zahnräder ist, dass diese aufeinander abgestimmt sind. Dies wird über das Modul der Zahnräder sichergestellt. Das Modul m ist das Verhältnis der Teilung p zur Zahl π . Der Teilkreisdurchmesser ergibt sich aus dem Modul multipliziert mit der Zähnezah. Das Rad und Gegenrad müssten immer das gleiche Modul haben.²⁵ Der Zahnkranz des Radioteleskops hat ein Modul von 5, somit muss das Gegenrad das gleiche Modul besitzen. Hinzu kommt die Dicke des Zahnrads, bei beiden sollten die gleichen Abmaße vorhanden sein, damit beide Zahnflanken mit der kompletten Fläche ineinander greifen können und die Kräfte der Drehbewegung auf die ganze Fläche verteilt werden. Der Zahnkranz hat eine Dicke von 45 mm, somit sollte das Gegenrad die gleichen Dimensionen haben.²⁶ Diese zwei Abmaße, Modul und Dicke, sind von dem Zahnkranz vorgegeben. Die Zähnezah für das Gegenrad wurde auf 20 festgelegt.

In das Zahnrad muss eine Passung mit der Bohrung von 15 H7 gerieben werden. Da ein Zahnrad sich frei drehen muss, das andere hingegen fest auf die Welle gepresst wird, musste eine Hülse²⁷ konstruiert werden. Die eine Seite der Hülse ist mit einer Passung von 15 f7 bemaßt und ergibt mit der Bohrung des Zahnrads 15 H7 eine Spielpassung. Daraus folgt, dass das eine Zahnrad auf der Hülse drehbar gelagert ist. Die andere Seite der Hülse ist mit einer Passung von 15 s6 ausgelegt und ergibt mit der Bohrung des zweiten Zahnrads eine Übermaßpassung. Dieses Zahnrad ist daher fest mit der Welle verbunden. Durch das Verdrehen der Zahnräder wird im Inneren eine Schenkelfeder²⁸ gespannt.

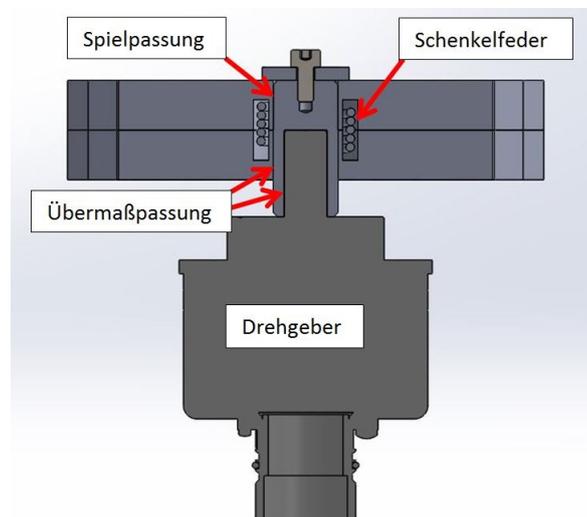


Abbildung 16: Schnittdarstellung federverspanntes Zahnrad mit Drehgeber

²⁴technische Zeichnung siehe Anhang

²⁵vgl. [Hoischen u. Hesser 2013, S. 133]

²⁶technische Zeichnung siehe Anhang

²⁷technische Zeichnung siehe Anhang

²⁸Datenblatt siehe Anhang

Zunächst war das Zahnrad so ausgelegt, dass die Flankenhöhe der beiden Zahnräder gleich sind. Nach der Übergabe der technischen Zeichnung an die Werkstatt stellte sich jedoch heraus, dass diese nicht das passende Werkzeug besitzt, um Zahnräder mit solchen Flanken zu fertigen. Aus diesem Grund wurde die Konstruktion auf die Werkzeuge angepasst. Dies ist für das Zahnrad des Drehgebers kein Nachteil, da in diesem Fall keine Kräfte übertragen werden, sondern das Zahnrad ohne Kraftübertragung mitläuft.

3.3.2 Schenkelfeder

Die Schenkel- bzw. Drehfeder ist ein Normteil und hat keine besonderen Anforderungen, die eingehalten werden müssen. Die Feder muss lediglich in die beiden Zahnräder passen und ein ausreichendes Drehmoment besitzen, um die Zahnflanken gegeneinander zu verspannen. Die Fa. Knörzer bietet eine passende Schenkelfeder an. Die Feder wurde nach Bild A²⁹ bestellt und hat ein Nenndrehmoment von 1068,14 Nmm. Wie in Abb. 17 zu sehen ist, kann das Drehmoment dieser Feder variabel eingestellt werden. Wird der Schenkel mehr oder weniger verdreht, ist das Drehmoment höher bzw. geringer. Dies bietet die Möglichkeit, die Kraft am Radioteleskop gezielt einzustellen und bei Bedarf einen Zahn mehr oder weniger zu verspannen.

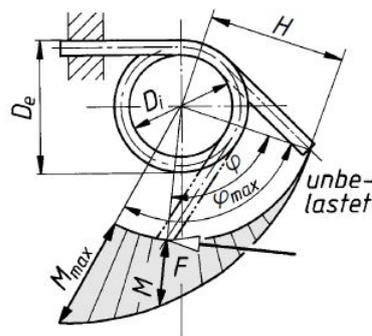


Abbildung 17: Drehfeder³⁰

3.3.3 Drehgeber

Die verwendete Regeleinheit hat einen inkremental Encodereingang, welcher eine RS 422 Schnittstelle besitzt sowie mit 1 V_{SS} betrieben werden kann. Das Unternehmen Heidenhain bietet zwei mögliche Drehgeber für dieses Projekt. Beide sind Inkrementaldrehgeber. Diese bestehen im Prinzip aus einer Scheibe mit lichtdurchlässigen und -undurchlässigen, leitenden und nichtleitenden oder reflektierenden und nichtreflektierenden Segmenten. Diese Bauteile haben in der Regel drei Spuren. Durch Zählen der Impulse, der einzelnen Spuren, wird die Position bestimmt. Zwei dieser Spuren sind um 90° phasenverschoben. Dies bewirkt, dass der Drehgeber Informationen hat, in welche Richtung er sich dreht, je nachdem ob Spur 1 oder 2 zuerst eine positive Flanke hat. Die letzte Spur ist eine Referenzmarke. Diese gibt einmal pro Umdrehung einen Impuls, somit kann der Drehgeber ermitteln, wann dieser eine vollständige Umdrehung absolviert hat. Der Nachteil des inkrementalen Drehwinkelsensors besteht darin, dass er sobald dieser ausgeschaltet wird, keine Informationen mehr über die aktuelle Position hat. In diesem Fall muss der Referenzpunkt angefahren werden.³¹ In Abb. 18 sind

²⁹ siehe Datenblatt Anhang

³⁰ [Wittel u. a. 2015, S. 331]

³¹ vgl. [Bernstein 2013, S. 303-304]

die Signale des Drehgebers dargestellt. Dies ist beispielsweise das Signal eines inkrementalen TTL Drehgebers. Spur U_{a1} und U_{a2} sind um 90° Phasenverschoben. Die Spur U_{a0} dient als Referenzspur. Dieser Geber hat zusätzlich noch eine vierte Spur $\overline{U_{aS}}$ und dient zur Erkennung von Fehlfunktionen, wie beispielsweise ein Kabelbruch oder ein Ausfall der Versorgungsleitung.

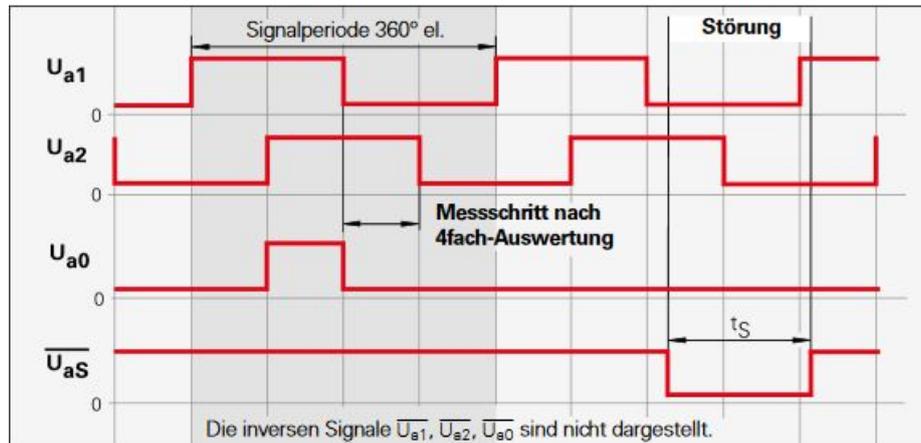


Abbildung 18: TTL-Signal Drehgeber Heidenhain³²

Der eine, ROD 420, gibt ein TTL Signal mit einem Differenzleitungstreiber nach RS 422 aus, der andere, ROD 480, hat als Ausgangssignal $1 V_{SS}$. TTL-Pegel haben als Ausgangssignal Rechtecksignale, das Ausgangssignal von $1 V_{SS}$ sind Sinuspegel.

Um die genaue Auswahl der Drehgeber zu bestimmen, muss für diese die benötigte Strichzahl berechnet werden. Je höher die Strichzahl des Encoders, umso genauer ist die Winkelauflösung. Die Übersetzung i_D ist zwischen Drehgeber und Zahnkranz. Für das Zahnrad, welches auf den Drehgeber geflanscht werden soll, wird eine Zähnezah von $z_{zd} = 20$ angenommen. Somit ist die Übersetzung $i_D = 4$, da sich in diesem Fall der Drehgeber vier mal um die eigene Achse dreht, bis sich das Radioteleskop um 360° gedreht hat. Daraus ergibt sich, dass der Drehwinkel des Teleskops in den Zähler und die Übersetzung in den Nenner fällt. Der Wert $\frac{1}{4}$ beruht auf der Tatsache, dass bei jedem Strich des Drehgebers vier Flanken registriert werden. \varnothing ist die benötigte Winkelauflösung des Radioteleskops.

\varnothing aus (9) und $z_{Abtrieb}$ aus (10)

$$i_D = \frac{z_{Abtrieb}}{z_{zd}} = \frac{80}{20} = 4 \quad (16)$$

$$z = \frac{1}{4} \cdot \frac{360^\circ}{\varnothing \cdot i_D} = \frac{1}{4} \cdot \frac{360^\circ}{0,04^\circ \cdot 4} = 562,5 \quad (17)$$

Um spätere Änderungen in der Konstruktion mit zu berücksichtigen, wurde nach einem Drehgeber mit einer Strichzahl, welche mindestens doppelt so groß ist, recherchiert. So ergab sich der Wert 2048, dies ist ein Standardwert und daher schnell lieferbar. Für beide Drehgeber, TTL sowie $1 V_{SS}$, wurde ein Angebot eingeholt.

³²Datenblatt [Heidenhain 2016]

3.3.4 Referenzpunkt

In dem Fall des 4-Meter-Radioteleskops, soll der Drehgeber über ein Zahnrad mit dem Teleskop verbunden werden. Somit besteht eine Übersetzung von $\neq 1$. Dies hat zur Folge, dass der Drehgeber mehrere Umdrehungen machen muss, damit sich das Teleskop um 360° dreht. Solange der Inkrementaldrehgeber die Anfangsposition kennt und mit Spannung versorgt ist, kann dieses Messsystem ohne Einschränkungen verwendet werden. Wird jedoch das System ausgeschaltet oder die Spannungsversorgung unterbrochen, hat dieser keine Informationen über den aktuellen Winkel. Der Drehgeber kann zwar den eigenen Referenzpunkt anfahren, jedoch weiß dieser nicht, wie viele Umdrehungen er machen muss, bis das Teleskop den minimalen bzw. maximalen Winkel erreicht hat. Daher wird für diesen Fall ein Referenzpunkt an dem Teleskop festgelegt, welcher nach jedem Neustart des Systems angefahren werden muss.

Von dem Referenzpunkt hängt ab, wie genau das ganze System ist. Da von diesem Punkt aus die Messungen des Inkrementalgebers beginnen. Wenn die Messung des Referenzpunktes zu ungenau ist, sind alle folgenden Messwerte unbrauchbar. Mögliche Messsysteme sind Gabellichtschranken, diese haben eine sehr kleine Auflösung und daher eine hohe Genauigkeit. Mit diesem Messsystem werden auch die Inkremente des Drehgebers gemessen. Bei jedem Inkrement schaltet die Lichtschranke und der Drehwinkel kann ermittelt werden.

Ein geeigneter Sensor wurde bei der Firma Dietz-Sensortechnik³³ gefunden. Dieser hat eine sehr kleine Auflösung und gute Wiederholgenauigkeit. Nach dem gleichen Prinzip der Referenzspur des Drehgebers, soll dieses Verfahren angewendet werden. Wenn sich ein Objekt durch die Lichtschranke bewegt, schaltet diese. Somit kann an dem Teleskop eine Vorrichtung befestigt werden, welche bei genau einer Umdrehung des Radioteleskops schaltet.

3.3.5 Endlagenschalter

Das Teleskop benötigt zwei Endlagen, diese dienen dazu, dass das System nicht über einen maximalen bzw. minimalen Winkel fahren kann. Wäre dies nicht der Fall, würden sich, bei Überschreiten des maximalen bzw. minimalen Winkels, die Kabel um das Teleskop drehen und irgendwann abreißen. Um dies zu verhindern müssen Endschalter eingebaut werden. Diese benötigen jedoch keine speziellen Anforderungen. Aus diesem Grund kann nach Standardsensoren gesucht werden.

3.3.6 Problemstellung

Zur Klärung des aktuellen Stands, einer Besprechung des Projekts sowie die Thematisierung der vorhandenen Probleme, wurde eine Abteilungsbesprechung einberufen. In dieser Sitzung waren die Systemgruppe, in welcher die Arbeit projiziert wird sowie die IT-Abteilung anwesend. Ziel war es mit einem breiten Fachwissen zu diskutieren. Zunächst wurde eine kurze Einführung in das Projekt gegeben. Mit diesem Grundwissen konnte eine kurze Erörterung der beiden möglichen Drehgeber vorgetragen werden. Zudem wurde das weitere Vorgehen diskutiert.³⁴

In der Besprechung wurde entschieden, dass zunächst nach alternativen Antriebseinheiten recherchiert werden soll, da der Elevationsantrieb defekt ist. Zudem birgt die Kommunikationsbereitschaft mit dem Unternehmen Jenarer Antriebstechnik Schwierigkeiten, welche kein Angebot des Azimut

³³Datenblatt siehe Anhang

³⁴Besprechungsprotokoll siehe Anhang

Motors erstellen. Vor diesem Hintergrund wurden alternativ entsprechende Komponenten bei Bosch Rexroth angefragt.

3.4 Akquise der ausgewählten Baugruppen

3.4.1 Bosch Rexroth

Bosch Rexroth bietet die Möglichkeit Applikationen nach Wunsch des Kunden auf die benötigten Projekte anzupassen. In diesem Fall wurden die vorhandenen Berechnungen aus Kapitel 2 sowie das Datenblatt des vorhandenen Servoverstärkers, Ecovario 114D, der Firma Bosch Rexroth zukommen gelassen. Die Rahmenbedingungen des Elevationsaktors sind aus den vorangegangenen Arbeiten übernommen worden. Zudem wurde eine kurze Einführung in die Problematik gegeben, damit diese sich in das Projekt einarbeiten können.

Um die ersten Eindrücke zu evaluieren, wurde eine Besprechung abgehalten, in welcher die Systeme sowie die Möglichkeiten, die das Unternehmen anbieten kann, vorgestellt wurden. Es stellte sich heraus, dass der Servoverstärker der Jenaer Antriebstechnik mit den Bosch Rexroth Motoren nicht kompatibel ist. Für die geforderte Anwendung stellt dieser eine zu geringe Leistung zur Verfügung, um die benötigten Motoren zu betreiben. Somit müsste von Bosch Rexroth zusätzlich eine Regeleinheit geordert werden. Diese Regeleinheit kann über Ethernet mit den Motoren kommunizieren, mit dem großen Vorteil eines Interface, zur TCP/IP Kommunikation mit der Regeleinheit und eines C-Programms. Bosch Rexroth bietet eine Bibliothek an, in welcher Funktionen für die Parameterübergabe implementiert sind.

In dem Angebot von Bosch Rexroth ist ein Elektrozyylinder für den Elevationsantrieb sowie ein Servomotor mit Getriebe für den Azimutantrieb enthalten. Für beide Antriebe ist jeweils eine Regeleinheit vorgesehen. Zudem besteht die Möglichkeit, an beide Achsen einen externen Drehgeber anzuschließen. Das Getriebe hat eine Lieferzeit von sieben Wochen und ist mit Hinblick auf die zeitliche Begrenzung dieses Projekts eine nicht annehmbare Zeitspanne. Die Firma Neugart bietet vergleichbare Getriebe zu den Bosch Rexroth Bauteilen an. Diese sind Präzisions-Planetengeräte, mit dem Vorteil, dass das Bauteil ein geringes Verdrehspiel besitzt und zudem eine passende Übersetzung hat. Bosch Rexroth gab die Freigabe für dieses Bauteil, welches eine Lieferzeit von drei bis vier Wochen beträgt.

Bosch Rexroth stellt zur Überbrückung der Lieferzeit ein Testmodul zur Verfügung. An diesem Testmodul können somit die ersten Softwaretests durchgeführt werden, um diese an den bestellten Bauteilen später zu übernehmen.

3.4.2 Antriebseinheit

Bevor die Antriebseinheit bestellt wird, sind nochmals die Vor- und Nachteile der drei möglichen Antriebseinheiten in Tabelle 1 aufgelistet. Somit kann sich ein besserer Überblick verschafft werden.

Motoren	Vorteile	Nachteile
Jenarer Antriebstechnik Kosten: kein Angebot erhalten!	<ul style="list-style-type: none"> • Servoverstärker und Motor gleicher Hersteller • Kompatibilität • Servoverstärker vorhanden 	<ul style="list-style-type: none"> • geringes Nenndrehmoment • kein Angebot erhalten • keine Kommunikation mit Unternehmen möglich • kein Elevationsantrieb im Sortiment • Steuerung nur mittels LabVIEW möglich
Profitracker XL Kosten: 25.000 €	<ul style="list-style-type: none"> • komplettes System für Azimut und Elevation • Steuerung über C möglich 	<ul style="list-style-type: none"> • hoher Anschaffungspreis • neuer mechanischer Aufbau notwendig • vorhandene Teile, wie Zahnkranz unbrauchbar
Bosch Rexroth Kosten: Azimut: 2895,72 € Elevation: 4804,09 €	<ul style="list-style-type: none"> • Motoren und Getriebe nach eigener Berechnung ausgelegt • Steuerung über C möglich • Elevation und Azimut aus einem System • gute Kommunikation • Hilfestellung bei Inbetriebnahme • Testmodul wird zur Verfügung gestellt 	<ul style="list-style-type: none"> • vorhandener Servoverstärker nicht kompatibel

Tabelle 1: Auswahl Antriebssystem

Nach Auflistung der Vor- und Nachteile kristallisiert sich heraus, dass die Antriebseinheit von Bosch Rexroth die beste Wahl ist. Mit den Vorteilen, dass durch Absprache mit Bosch Rexroth die bestmöglichen Komponenten für dieses Projekt ausgesucht werden konnten, alle Bauteile miteinander kompatibel sind und die Ansteuerung mit der Hochsprache C über das TCP/IP implementiert werden kann. Hinzu kommt, dass dies preislich viel günstiger ist, als das Komplettsystem, Profitracker XL.³⁵ Zudem steht zur Überbrückung der Lieferzeit ein Testmodul zur Verfügung. Dies ist eine abgespeckte Applikation, mit welcher die Inbetriebnahme realisiert werden kann. Für die Inbetriebnahme des Testmoduls stellt Bosch Rexroth einen Fachmann zur Verfügung.

Das Getriebe wird von der Fa. Neugart geordert. Um einen externen Drehgeber mit einzubinden, wurde die Regeleinheit mit einer Standardschnittstelle ausgestattet. Mit dieser könnten beide Drehgeber Varianten aus Kapitel 3.3.3 angeschlossen werden.

Die Institutsleitung bietet zunächst nur die Möglichkeit an den Azimutantrieb zu bestellen, jedoch mit der Aussicht, wenn dieser läuft den Elevationsantrieb ebenfalls zu ordern. Da sich diese Arbeit zentral um die Projektierung im Azimutbereich dreht, ist dies für das Projekt annehmbar. In

³⁵Datenblatt siehe Anhang

späteren Arbeiten müssten somit nur noch die Bauteile bestellt werden, da diese speziell für das 4-Meter-Radioteleskop ausgelegt sind. Somit kann dort die Recherche entfallen.

Im Folgenden ist eine kurze Auflistung der bestellten Bauteile für die Antriebseinheit dargestellt:

- Antriebsmotor: Synchron Servomotor IndraDyn S ³⁶
- Getriebe: Präzisions-Planetengeräte ³⁷
- Regler: INDRADRIVE HCS02
- Kompaktumrichter
- Steuereinheit

3.4.3 Kraftübertragung

Die Kraftübertragung zwischen Motor und Radioteleskop wird mittels dem Präzisionsplanetengeräte³⁸ der Fa. Neugart realisiert. Dieses Getriebe hat eine Übersetzung von $i = 64$ sowie ein geringes Verdrehspiel. An dieses Getriebe wird über eine Passfederverbindung ein Zahnrad angeflanscht. Dieses Zahnrad ist ein Standardzahnrad. Um geringes Spiel zu gewährleisten, wird dieses gehärtet, geschliffen und mit einer Evolventenverzahnung versehen. Modul und Breite des Zahnrads sind von dem Zahnkranz des Teleskops vorgegeben. Ein weiterer Punkt, welcher beachtet werden muss, ist die Übersetzung. Diese wird in der Regeleinheit implementiert. Somit kann ein Winkel vorgegeben werden und mittels der Übersetzung kann die benötigte Rotation des Motors berechnet werden. Um eine hohe Regelgüte zu erhalten, ist es sinnvoll eine Übersetzung mit einer ganzen Zahl zu verwenden. Beispielsweise wäre die Übersetzung von $i = 6,6$ nicht sinnvoll, diese müsste gerundet werden und führt somit zu vermeidbaren Abweichungen.

Daraus resultierend und aus den Dimensionen des Radioteleskops ist eine Zähnezahl von 20 geeignet. Aus diesem Grund werden mit dieser Zähnezahl und den gewünschten Bauteilen Berechnungen durchgeführt.³⁹ Infolgedessen ergibt sich ein resultierendes Drehmoment von 1561,6 Nm und einer Umdrehungsdauer von 5,12 Sekunden für das Radioteleskop. Vergleicht man diese mit den berechneten Werten aus Kapitel 2, maximales Drehmoment mit einer Windlast von 850 Nm und einer Umdrehungsdauer von circa einer Minute, ist sofort zu erkennen, dass dieses Konzept für die Anforderungen ausreichend ist. Die Umdrehungsdauer ist für die Anwendung einer radioastronomischen Messung viel zu schnell. Jedoch können die Rotationsgeschwindigkeiten über die Regeleinheit vorgegeben werden. Es muss im Verlauf der Inbetriebnahme getestet werden, welche Rotationsgeschwindigkeiten für die radioastronomische Messung geeignet sind.

3.4.4 Auswahl Winkelmesssystem

Die Regeleinheit von Bosch Rexroth ist mit beiden ausgewählten Heidenhain Drehgebern kompatibel. Vor diesem Hintergrund wurde die Entscheidung getroffen, den inkremental Drehgeber⁴⁰ ROD 420 mit TTL Pegel, zu verwenden. Ebenso wie der interne Drehgeber des Antriebsmotors, wird dieser an der Regeleinheit angeschlossen. In diesem Fall sollte die Übersetzung, aus der selben Prämisse wie

³⁶Datenblatt siehe Anhang

³⁷Datenblatt siehe Anhang

³⁸Datenblatt siehe Anhang

³⁹Berechnungen siehe Anhang

⁴⁰Datenblatt siehe Anhang

in Kapitel 3.4.3, eine natürliche Zahl sein. Somit wurde sich für ein Zahnrad mit der Zähnezah von 20 entschieden, welches auf den Drehgeber geflanscht wird. Mit dem Zahnkranz, dessen Zähnezah 80 ist, ergibt sich somit eine Übersetzung von vier. Die Genauigkeit des Drehgebers ist abhängig von der Übersetzung. Daher musste für diesen Fall die Überschlagsrechnung, aus Kapitel 3.3.3, nochmals überprüft werden. In dieser wurde bereits mit einer Zähnezah von 20 gerechnet und ergibt mit den verwendeten Bauteilen eine benötigte Strichzahl von 563. Infolgedessen ist die ausgewählte Strichzahl von 2048 hinreichend genau.

3.4.5 Referenzmessung und Endlagendetektion

Die Referenzmessung wird mit der Gabellichtschranke⁴¹ des Unternehmens Dietz-Sensortechnik realisiert. Diese Lichtschranke hat eine Gabelweite von 30 mm. Das Bauteil soll so an dem Radioteleskop befestigt werden, dass diese bei jeder vollen Umdrehung einmal schaltet und dies der Steuereinheit signalisiert. Da der verwendete Drehgeber ein Inkrementaldrehgeber ist, muss dieser bei einem Neustart des Systems den Referenzpunkt anfahren. Dies muss über die Software gelöst werden. Die Gabellichtschranke hat eine sehr genaue Auflösung sowie eine präzise Wiederholgenauigkeit. Aus diesem Grund wurde sich für dieses Bauteil entschieden. Damit die Lichtschranke schaltet, muss ein Objekt den Laserstrahl unterbrechen. Der Winkel, in dem die Lichtschranke schaltet, muss dem System genau vorgegeben sein. Dieser Punkt ist eine fixe Position, was bedeutet wenn die Lichtschranke schaltet, kann die Position genau referenziert werden, unabhängig von dem Messwert des Inkrementaldrehgebers. Es muss darauf geachtet werden, ob die Lichtschranke von rechts oder von links überfahren wird. Da das Bauteil, welches die Lichtschranke unterbricht, eine bestimmte Dicke hat, würde dies zu einem Offsetfehler führen. Diese Dicke des Bauteils muss bekannt sein und in das Steuerprogramm implementiert werden. Daher sollte die Steuerung wissen, aus welcher Richtung sich das Objekt durch die Lichtschranke bewegt.

Ein wichtiges Bauteil zur Automatisierung eines Radioteleskops sind Endlagenschalter. Diese Schalter dienen dem Zweck, dass die Endlage erkannt wird und über diese hinaus keine weitere Rotation möglich ist. Somit wird sichergestellt, dass das Teleskop sich nicht über eine bestimmte Position drehen kann und Gefahr läuft, dass die Versorgungs- und Datenkabel abgerissen werden. Der Standort Effelsberg hat passende Taster für diesen Zweck auf Lager, welche von der Fa. Honeywell sind. Dies sind Standardtaster mit einer Schutzklasse von IP 67 und somit für den Betrieb draußen geeignet.

⁴¹Datenblatt siehe Anhang

3.5 Konstruktion der mechanischen Befestigung

3.5.1 CAD-Modell der mechanischen Befestigung

Infolge der Bestellungen wurde sich mit der Befestigung dieser Komponenten beschäftigt. Die Halterung soll an den Beinen des Radioteleskop montiert werden. In dieses Bauteil wurden Nuten konstruiert. Mit Hilfe dieser sind die Komponenten frei verschiebbar und es können nach der Fertigung Feinadjustierungen vorgenommen werden.

Um das Teleskop für radioastronomische Messungen zu verwenden, muss dieses einen Winkelbereich größer als 360° besitzen. Aus diesem Grund wurde eine Platte konstruiert in welcher ein Mitnehmer in einer Führung geschoben wird und mit Hilfe dieser Technik Rotationen über 360° ermöglicht. Am Ende dieser Führung sind jeweils Endschalter angebracht. Diese sollen elektrisch an der Regeleinheit angeschlossen werden und eine weitere Drehung des Teleskops vermeiden. Da ein Inkrementalgeber verwendet wird, soll die Gabellichtschranke an der Grundplatte montiert werden. Bei jeder Umdrehung wird diese geschaltet und somit der Inkrementalgeber referenziert, siehe Abb. 19.

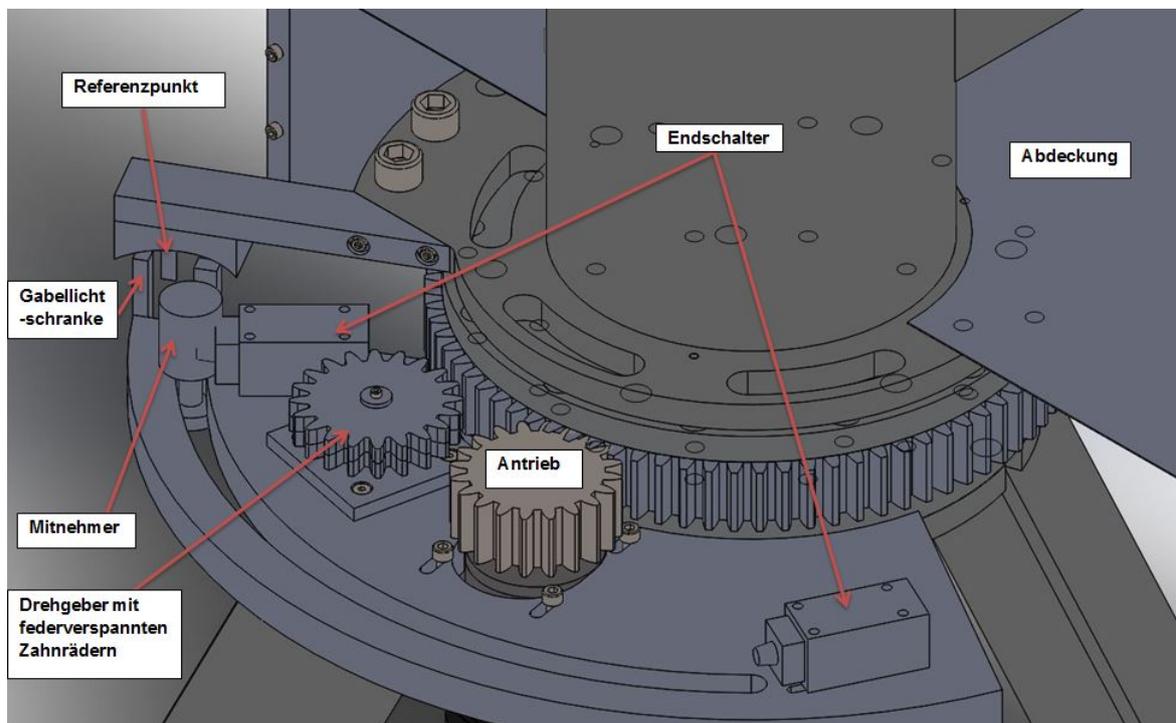


Abbildung 19: Befestigung Antriebseinheit und Messsystem

Alle Bauteile sind für die vorhandenen Witterungsverhältnisse ausgelegt. Wie jedoch bei dem vorherigen Elevationsantrieb zu sehen, sind die Angaben zur IP-Schutzklasse nicht zuverlässig. Hinzu kommt, dass die Zahnräder geschmiert werden und sich das Schmiermittel durch Niederschlag nicht lösen soll. Aus diesen Gründen wurde eine Abdeckung konstruiert, welche über die Antriebseinheit montiert werden soll und die Bauteile so gegen Umwelteinflüsse schützt. In Abbildung 20 ist das CAD-Modell zu sehen. Unter Zuhilfenahme der Dokumentationen und der CAD-Modelle des Motors, Getriebe und Drehgeber, konnte ein präzises Modell für die Halterung entworfen werden. Die Halterung kann in der eigenen Werkstatt gefertigt werden.

Nachdem die bestellten Bauteile geliefert waren, wurden an dem CAD-Modell nochmals Anpassungen vorgenommen und aus diesen, technische Zeichnungen erstellt.⁴² Diese Zeichnungen wurden an die Werkstatt übergeben, welche die Bauteile fertigt.

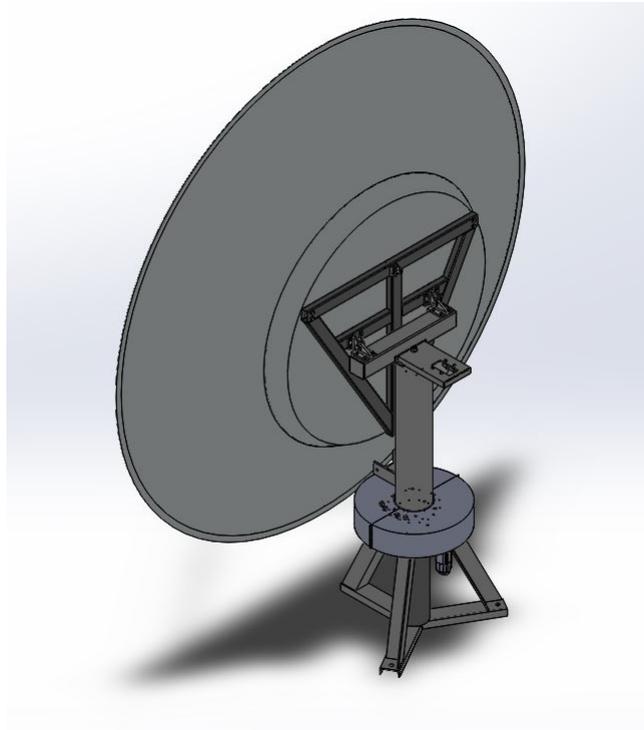


Abbildung 20: Entwurf gesamter Azimutantrieb mit Abdeckung

3.5.2 Aufbau mechanischer Befestigung

Nachdem die Bauteile unter Vorgabe der technischen Zeichnungen gefertigt wurden, konnten diese an dem Radioteleskop montiert werden, siehe Abb. 21. Die Grundplatte wird an den Beinen des Teleskops befestigt. Auf diese werden die Antriebseinheit sowie das Winkelmesssystem montiert. Wie in der Planung sind zwei Endschalter an dem Teleskop angebracht, welche die Endlagen signalisieren. Zudem wird über einen Hebel der Mitnehmer in einer Führung geschoben und ermöglicht somit Rotationen über 360° . Hinzu kommt, dass mit Hilfe des Hebels die Gabellichtschranke geschaltet wird und diese das Messsystem referenziert.

⁴²technische Zeichnungen befinden sich im Anhang



Abbildung 21: Aufbau Azimut Antrieb mit Winkelmesssystem

In Abb. 22 ist das federverspannte Zahnrad dargestellt. Wie gut zu erkennen, sind die Zahnradflanken des federverspannten Zahnrads gering versetzt. Mit Hilfe dieser Konstruktion wird das Flankenspiel vermieden und infolgedessen die Rotation des Zahnrads, ohne Spiel, auf den Drehgeber übertragen und systematische Messabweichungen vermieden.



Abbildung 22: Federverspanntes Zahnrad

3.6 Software Konzept

In Abb. 23 ist eine Übersicht des Konzepts zur Automatisierung des 4-Meter-Radioteleskops dargestellt. Die Regeleinheit von Bosch Rexroth soll dem Teleskop einen Winkel und eine Rotationsgeschwindigkeit vorgeben und mit dieser Vorgabe das System positionsgeregelt bewegen. Mit Hilfe des Drehgebers wird die aktuelle Position signalisiert. Diese Parameter werden über Ethernet an den Rechner übermittelt. Das Programm auf dem Computer gibt dem Regler eine Soll-Position vor und ließt die Ist-Werte aus. Dies geschieht über ein C-Programm. Es ist vorgesehen, dass das C-Programm das Interface zwischen Regler und Computer herstellt. Für diese Anwendung stellt Bosch Rexroth eine Bibliothek zur Verfügung. Mittels dieser Bibliothek ist es möglich die Parameter auszulesen und zu übergeben. Die Parameter zur Positionierung sowie zum Auslesen sollen mittels der Hochsprache Python an das C-Programm übergeben werden. Es ist vorgesehen eine grafische Oberfläche in Python, unter Verwendung von Qt, zu realisieren. Qt ist ein Framework zur plattformunabhängigen Programmierung von grafischen Benutzeroberflächen. Die Übertragung bzw. das Auslesen der Parameter soll über das interne Netzwerk des Instituts möglich sein. Die Planung ist dahingehend die Parameterübergabe mittels UDP zu übertragen sowie das Auslesen dieser Daten über einen Multicast an eine Teilnehmergruppe zu verschicken.

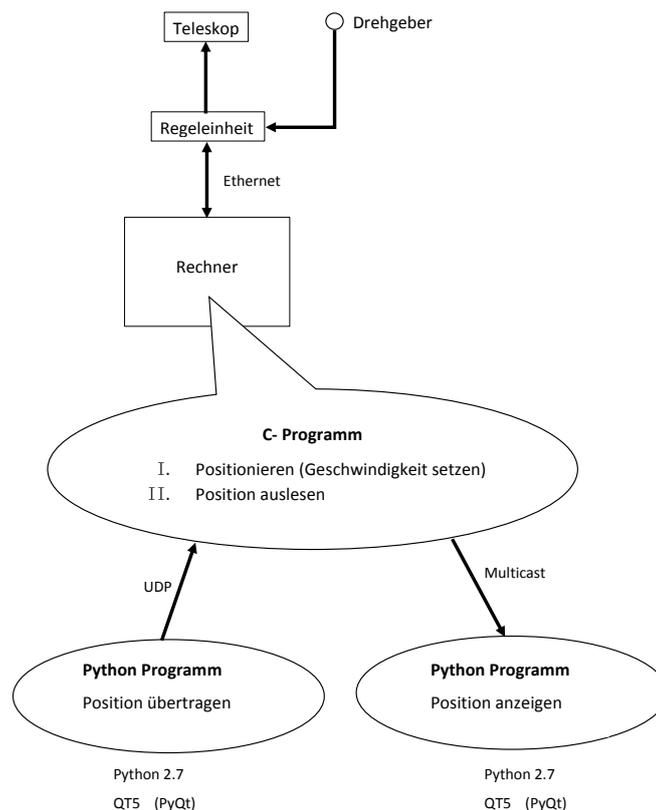


Abbildung 23: Software Konzept

3.7 Inbetriebnahme des Testmoduls von Bosch Rexroth und Kommunikation mit dem C-Programm

Zur Überbrückung der Lieferzeit stellt Bosch Rexroth ein Testmodul zur Verfügung. Mit Hilfe dieses können die Inbetriebnahme und erste Softwarearbeiten durchgeführt werden.

Für den Steuerrechner ist das Betriebssystem Ubuntu vorgegeben. Der Computer soll über Ethernet mit der Regeleinheit kommunizieren. Für diesen Fall stellt Bosch Rexroth eine C-Bibliothek zur Verfügung, welche die Kommunikation zwischen Computer und Regler herstellt.

Das Testmodul von Bosch Rexroth, siehe Abb. 24, ist mit einer kleinen Regeleinheit und einem Motor ausgestattet. Die Ansteuerung des Reglers und Motors sind analog zu dem bestellten Modul. Somit kann mit dem Testmodul die Ansteuerung realisiert und dies ohne weiteren Aufwand für die bestellten Bauteile übernommen werden.

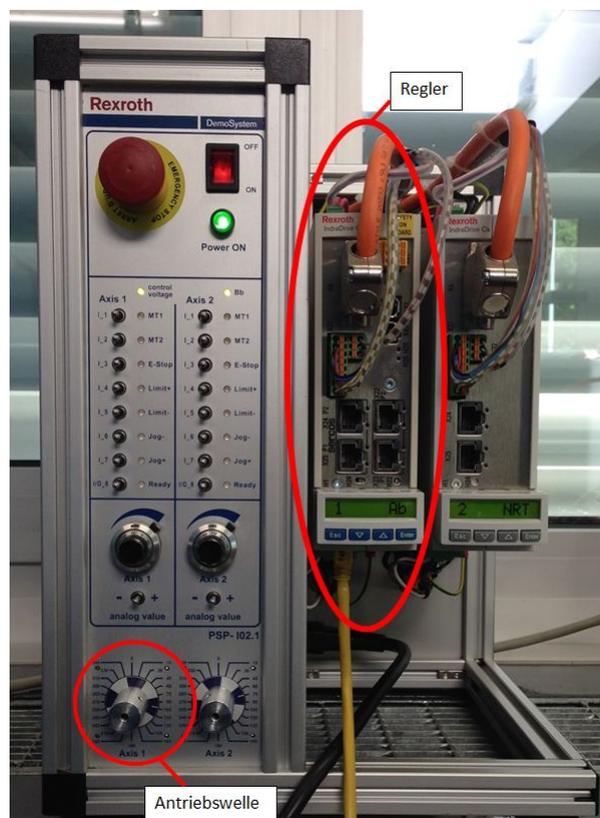


Abbildung 24: Testmodul Bosch Rexroth

Zur Inbetriebnahme des Reglers wird die Software IndraWorks zur Verfügung gestellt. Dieses Programm dient zur Administration der verbundenen Komponenten mit der Regeleinheit sowie zum Erstellen von Projekt- und Konfigurationsdaten. Im Folgenden wird die Inbetriebnahme erläutert. Der Regler wird über ein Ethernet-Kabel mit dem Computer verbunden. Diesem muss eine passende IP-Adresse vorgegeben werden. Von der Regeleinheit wird der Motor mit dem integrierten Drehgeber erkannt, welche die Motordaten sowie die Drehgeberspezifikation automatisch einstellt. Mittels dieses Programms können dem Regler, Grundeinstellungen vorgegeben werden, wie beispielsweise

einen zweiten Drehgeber zu implementieren. Zudem werden die optimalen Regelparameter ermittelt und eingestellt. Diese können bei Bedarf manuell eingestellt bzw. optimiert werden. Dieses Programm bietet eine Oszilloskop-Funktion, in welcher Prozesse aufgezeichnet und die Regelparameter an diese angepasst werden können. Dieses Werkzeug ist sinnvoll, um bei späteren Testfahrten des Teleskops Feineinstellungen vorzunehmen und die benötigte Genauigkeit für die radioastronomische Messung zu erreichen. Hinzu kommt, dass das Programm vorgeben kann, ob das System eine lineare oder rotative Bewegung vollführt. Die rotative Bewegung ist für diesen Fall die geeignetere Wahl. Infolgedessen kann dem Regler ein Winkel vorgegeben werden, welchen der Motor positioniert.

Der wesentliche Vorteil des Testmoduls ist der Test der Verbindung zwischen Regeleinheit und C-Programm, demzufolge wie der Regler mit einem C-Programm kommunizieren kann. Es stellte sich heraus, dass diese Variante Schwierigkeiten birgt, da Bosch Rexroth dieses Interface erst seit kurzem zur Verfügung stellt. Folglich zeigt Bosch Rexroth erst geringe Erfahrung mit dieser Implementierung. Der Vertreter hatte ein PDF-Dokument mitgebracht, in welchem die Vorgehensweise protokolliert ist, welche Programme installiert werden müssen und wie die Bibliothek eingebunden werden kann. Die benötigten Programme wurden installiert und die vorgegebenen Schritte durchgeführt. Dennoch kam es stets zu einem Absturz des C-Programms. Zunächst wurde die Kommunikation zwischen der Regeleinheit und dem Computer getestet, diese war gegeben. Dem Vertreter von Bosch Rexroth wurde der Fehlercode mitgegeben. Dieser sicherte zu, das Problem zu beheben.

Nach einigen Telefonaten mit Bosch Rexroth wurde ein Ansprechpartner gefunden. Dieser sitzt jedoch in Indien. Über eine Webkonferenz mit dem Programm WebEx, konnte mit diesem Konferenzen abgehalten werden. Mit Hilfe dieses Portals besteht nicht nur die Möglichkeit der Internettelefonie, sondern auch live den Bildschirm des anderen Teilnehmers zu betrachten und zudem den entfernten Computer zu steuern. Mir fiel auf, dass der Mitarbeiter aus Indien eine andere Ubuntu Version verwendet. Das vorhandene Betriebssystem ist auf der aktuellen Ubuntu Version 16.04, Bosch Rexroth verwendet jedoch die Version 14.04. Es wurde versichert, dass das System unabhängig von dem Stand der Ubuntu Version arbeitet. Davon wollte ich mich jedoch selber versichern und verblieb erstmals dabei, ein Betriebssystem mit Ubuntu 14.04 aufzusetzen und Bosch Rexroth die Kompatibilität mit Ubuntu 16.04 nochmals prüfen.

Nachdem Ubuntu 14.04 LTS, inklusive aller benötigten Programme, installiert und die Schritte zur Einstellung der Kommunikation durchgeführt wurden, wurde die Ansteuerung nochmals getestet. Mit diesem Betriebssystem konnte der anfängliche Fehler behoben werden. Nun kann der Regler in den Parametrierbetrieb schalten, jedoch nicht in Betriebszustand gehen, in welchem dieser Positionsvorgaben entgegennehmen kann. Aus diesem Grund wurde nochmals eine Webkonferenz mit dem Mitarbeiter aus Indien abgehalten. In dieser wurde bestätigt, dass das Ansprechen der Regeleinheit mit der Ubuntu Version 16.04 nicht möglich ist. Im Bezug auf den neuen Fehler konnte mir nicht weitergeholfen werden. Jedoch wurde ein Testprogramm zur Verfügung gestellt, um die Parameter aus der Regeleinheit auszulesen. Dieses Programm wurde getestet und konnte ohne Fehler ausgeführt werden. Das löst allerdings nicht das ursprüngliche Problem. Somit konnten die Parameterwerte des Reglers zwar ausgelesen, Positionen jedoch nicht vorgegeben werden.

Nach mehreren Webkonferenzen mit Bosch Rexroth Indien und einigen Änderungen im Quellcode der Bosch Rexroth Bibliothek sowie einer Vielzahl von Telefonaten mit den deutschen Mitarbeitern, konnte der Fehler behoben werden. Das Problem lag darin, dass die Regeleinheit nur mit bestimm-

ten voreingestellten Grundeinstellungen⁴³ arbeiten kann. Aus diesem Grund wurde zunächst die Regeleinheit auf Werkseinstellung zurückgesetzt und die benötigten Einstellungen der Regeleinheit übernommen. Infolgedessen konnte der Motor angesteuert sowie die aktuellen Werte ausgelesen werden. Die Änderungen in der Bosch Rexroth Bibliothek konnten zurückgesetzt werden, da mit diesen Einstellungen mit Hilfe der Standard Bibliothek gearbeitet werden kann.

⁴³siehe Anhang

3.8 Inbetriebnahme Regeleinheit

Nach der Lieferung der Antriebskomponenten, wurde das Testmodul durch diese ersetzt. Zunächst wurden der Motor und die Regeleinheit in Betrieb genommen. Jedoch bestand nicht die Möglichkeit, mittels des Programms IndraWorks den Motor an die Regeleinheit anzulernen. Es stellt sich heraus, dass der Motor aus der neusten Generation ist und die Regeleinheit, aus diesem Grund ein Firmware Update benötigt. Bosch Rexroth stellte dies zur Verfügung und somit konnte ein Update vollzogen werden. Nach diesem Update wurde der Motor von der Regeleinheit erkannt und diese ermittelt die Motor- und Drehgeberspezifikationen automatisch. Der Motor kann genauso wie das Testmodul über das C-Programm angesprochen werden.

In den Standardeinstellungen wird die Position des Motors von dem internen Drehgeber evaluiert. Wie eingangs erörtert⁴⁴ ist diese Messmethode nicht die geeignetste Wahl. Infolgedessen soll der zweite externe Drehgeber an die Regeleinheit angelernt werden. Der vorhandene Drehgeber ist ein Inkrementaldrehgeber⁴⁵ mit 5V TTL Signalen. In IndraWorks müssen diese Parameter angegeben werden, siehe Abb. 25. Zudem besteht die Möglichkeit die Verwendung des Drehgebers festzulegen. Zurzeit ist der Geber als Lageregelungsgeber implementiert. Jedoch besteht die Möglichkeit diesen als redundanten Geber zu verwenden. Dies wäre ein möglicher Verwendungszweck, jedoch müssen für diese Methode einige Referenzfahrten durchgeführt werden, damit die Messwerte beider Geber genau bekannt sind, um redundante Messungen adäquat durchführen zu können. Da zunächst die softwareseitige Inbetriebnahme Vorrang hat, wird die Verwendung Lageregelungsgeber eingestellt. Falls sich bei späteren Testfahrten herausstellt, dass der externe Drehgeber nicht die benötigte Genauigkeit ausweist, kann sich mit der redundanten Messung beschäftigt werden, um mit Hilfe von Referenzfahrten die beiden Messsysteme aufeinander anzupassen.

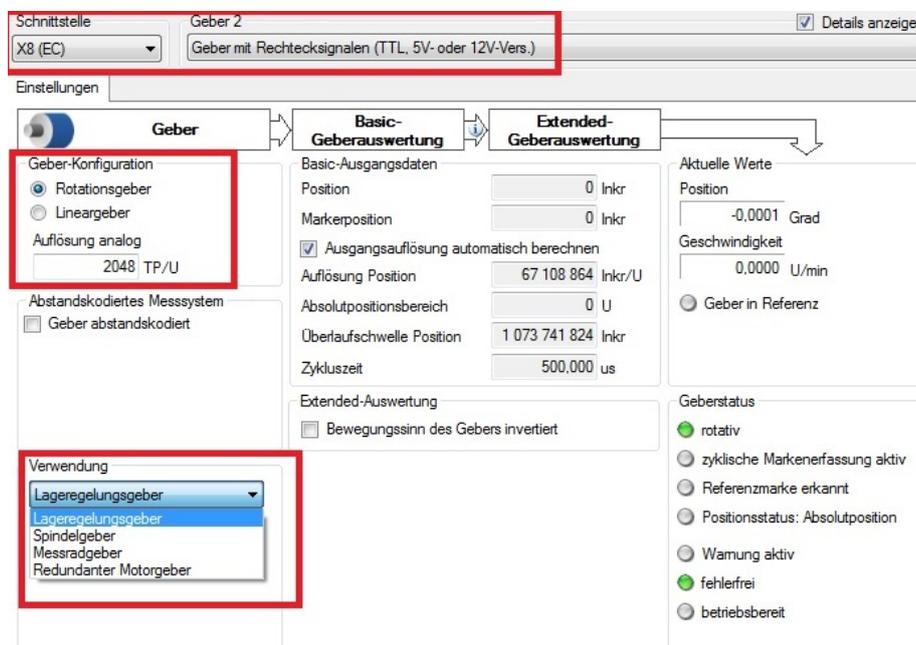


Abbildung 25: Externen Drehgeber anlernen

⁴⁴vgl. Kapitel 3.3

⁴⁵Datenblatt siehe Anhang

Infolgedessen, dass der externe Drehgeber ein Inkrementaldrehgeber ist, benötigt dieser einen Referenzpunkt. Für diesen Zweck wurde eine Gabellichtschranke⁴⁶ bestellt. Dieses Bauteil soll über die digitalen Eingänge der Regeleinheit den Referenzpunkt signalisieren. Zunächst bestand das Problem, dass die digitalen Eingänge nicht geschaltet werden konnten, wie sich herausstellte, sind diese galvanisch von den anderen Komponenten getrennt und benötigen eine zusätzliche Spannungsversorgung. Nach Beseitigen des Problems konnte die Lichtschranke in IndraWorks eingebunden werden. Das Programm bietet die Möglichkeit die digitalen Eingänge für festgelegte Zustände zu verwenden, siehe Abb. 26. Mit Hilfe dieser Funktionen kann dem Eingang, an welchem die Gabellichtschranke angeschlossen ist, die Aufgabe des Referenzschalters zugeordnet werden. Daraus resultiert, dass bei jedem Schalten der Lichtschranke der Inkrementaldrehgeber referenziert wird.

Digitale Eingänge			Status
PIN	Signal	Bit	
X31/1	S-0-0400 : Referenzschalter	0	
X31/2	P-0-0222 : Fahrbereichsgrenzscharter-Eingänge	0	
X31/3	P-0-0222 : Fahrbereichsgrenzscharter-Eingänge	0	
X31/4	S-0-0000 : Nicht zugewiesen	0	
X31/5	S-0-0000 : Nicht zugewiesen	0	
--	S-0-0000 : Nicht zugewiesen	0	
--	S-0-0000 : Nicht zugewiesen	0	
--	S-0-0000 : Nicht zugewiesen	0	
--	S-0-0000 : Nicht zugewiesen	0	
--	S-0-0000 : Nicht zugewiesen	0	
--	S-0-0000 : Nicht zugewiesen	0	

Abbildung 26: Implementieren der digitalen Eingänge

Der externe Drehgeber muss als Referenzgeber deklariert werden. Infolgedessen wird die Positionierung ausgehend von diesem Messsystem durchgeführt. Der Inkrementalgeber benötigt nach jedem Starten des Systems einen Referenzpunkt, von welchem dieser die Messungen referenzieren kann. Die Einstellung ist so gewählt, dass das Radioteleskop zunächst die Gabellichtschranke anfährt. Wurde die Initialisierungsfahrt durchgeführt, stoppt die Rotation und die Regeleinheit wartet auf Vorgaben zur Positionierung. Die Initialisierungsfahrt wird nach jedem Starten einmalig durchgeführt. Für die Beschleunigung bzw. Verzögerung der Referenzfahrt wurde $0,5 \text{ rad/s}^2$ gewählt sowie eine Winkelgeschwindigkeit von $0,5 \text{ U/min}$. Somit ist sichergestellt, dass die Initialisierungsfahrt nicht zu schnell ist und beispielsweise den Referenzschalter überfährt, ohne dass dieser schaltet. Diese Einstellungen können in der Inbetriebnahmesoftware implementiert werden, siehe Abb. 27.

⁴⁶Datenblatt siehe Anhang

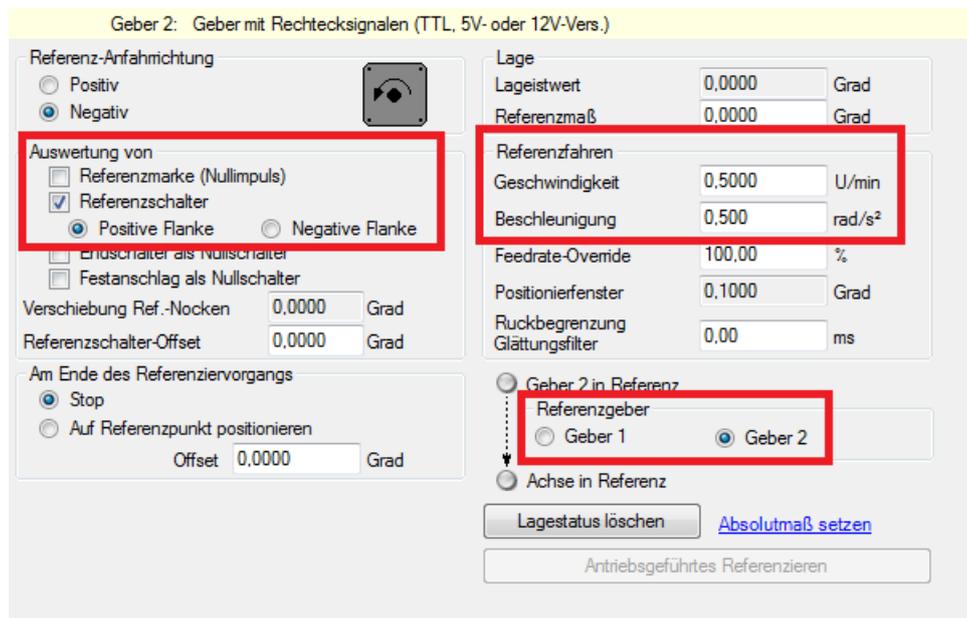


Abbildung 27: Implementieren der Referenzfahrt

Das Einbinden der Endschalter geschieht, genauso wie bei der Lichtschranke, über das Auswahlfenster. Dort werden die beiden Endschalter als Fahrbereichsgrenzschalter deklariert, einmal in der positiven Endlage und einmal in der negativen Endlage. Zudem besteht bei den Endschaltern die Möglichkeit auf das Schalten mit zwei verschiedenen Zuständen zu reagieren. Die eine Option stoppt beim Betätigen des Endschalters die Rotation und gibt eine Warnung aus. In diesem Fall kann das Teleskop zurückgefahren werden und den Messbetrieb weiter fortsetzen. Die zweite Variante stoppt das Teleskop beim Schalten der Endlage und erzeugt einen Fehler. Dieser Fehler muss explizit zurückgesetzt und mit Hilfe von IndraWorks aus diesem Zustand herausgefahren werden.

Für den Anwendungszweck dieser Arbeit ist die zweite Methode die geeignete Wahl. Da für das Teleskop zusätzlich Softwareendlagen implementiert werden, welche beim Überfahren der Grenzwerte schalten und den Rotationsvorgang stoppen. Die Endschalter sollen als redundantes System arbeiten, falls die Softwareendlage nicht schaltet oder der Benutzer diese umgeht, sind diese als Hardware Stopp zu sehen, welche die Rotation sofort beendet und ein weiteres Verfahren nicht ermöglicht. Die Fahrbereichsendlagen sind als Öffner deklariert. Mit dieser Variante ist sichergestellt, dass bei einem Unterbrechen der Spannungsversorgung der Endschalter, das Teleskop abschaltet und dieser Fehler nicht unbemerkt bleibt, siehe Abb. 28. Die Endschalter müssen somit als Öffner verdrahtet werden.

Die Softwareendlagen können genauso wie die Fahrbereichsgrenzschalter in IndraWorks eingebunden werden. In diesem Fall sind die Grenzwerte so eingestellt, dass diese vor den Hardwareendlagen den Rotationsvorgang stoppen.

Mit Hilfe dieser Mechanismen ist sichergestellt, dass das Teleskop den maximalen Rotationsbereich nicht überfahren kann und somit eine mögliche Beschädigung vermieden wird. Zudem kann der Operateur das Radioteleskop aus dem Observatorium nicht sehen und muss sich vollständig auf das Programm verlassen können. Wenn das Radioteleskop aufgebaut und dies in Betrieb genommen ist, wäre eine zusätzliche Webcam von Vorteil, dass der Operateur das System beobachten kann.

Das Einbinden dieser beiden Funktionen ist in Abb. 28 dargestellt.

Abbildung 28: Referenzieren der Endlagen

Um das Radioteleskop positionieren zu können, ist der Motor über ein Getriebe übersetzt. Diese Übersetzung⁴⁷ von 256 zwischen Motor und der Regeleinheit muss implementiert werden. Zudem benötigt das Programm das Übersetzungsverhältnis zwischen dem Drehgeber und dem Teleskop. In Kapitel 3.4.4 wurde eine Übersetzung von vier gewählt. Diese muss in IndraWorks eingebunden sein. In Abb. 29 ist die Einstellung der Übersetzung in IndraWorks dargestellt.

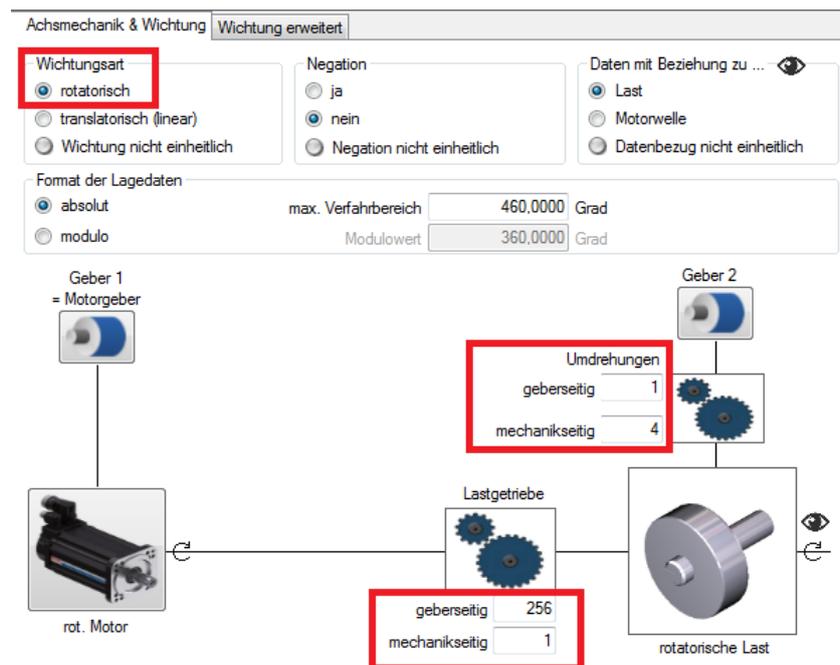


Abbildung 29: Implementieren der mechanischen Übersetzung

⁴⁷vgl. Kräfte- und Rotationsgeschwindigkeitsberechnung mit ausgewählten Bauteilen im Anhang

3.9 Software Projektierung

Der Softwareentwurf ist grafisch in Abb. 30 dargestellt. Dieser besteht aus einem C-Programm, welches in die drei Teile *Consumer*, *Regler* und *Publisher* aufgeteilt ist sowie zwei Python Programmen.

Das zum Setzen der Sollwerte erstellte Programm basiert auf Python und ist als GUI Anwendung ausgeführt. Diese werden in ein Datenpaket verpackt und mit Hilfe eines UDP Interface an das C-Programm übertragen.

Der *Consumer* des C-Programms hat die Aufgabe die Datenpakete aus dem UDP Interface zu lesen. Diese Pakete müssen in die Parameter Soll-Position und Soll-Zustand des Motors aufgeteilt werden. An den Programmteil *Regler* werden diese Parameter übergeben.

Der Programmteil *Regler* nimmt die Parameter des *Consumers* entgegen und dient zudem zur Kommunikation mit der Regeleinheit und dem Computer. Diese Kommunikation geschieht über Ethernet. Dieser Programmteil versetzt die Regeleinheit in den Zustand, dass diese die empfangenen Parameter entgegennehmen kann. Wird mit Hilfe des *Consumers* der Startbefehl vorgegeben, wird der Motor initialisiert. Ist der Motor in dem Zustand, dass dieser auf eine Winkelvorgabe wartet, soll dieser die Positionsparameter des *Consumers* auslesen und an die Regeleinheit übertragen. Infolgedessen wird das Radioteleskop positioniert. Ist die Position erreicht, werden die Parameter Position und Zustand des Motors ausgelesen und an den *Publisher* übertragen.

Der dritte Teil des C-Programms ist der *Publisher*, dieser erhält den Zustand des Motors sowie die aktuellen Positionswerte. Diese werden wie beim UDP in ein Datenpaket gepackt und über einen Multicast an eine Teilnehmergruppe übertragen.

Das zweite Python Programm dient zur Darstellung der aktuellen Parameter des Radioteleskops. Diese Parameter werden aus dem Multicast Interface gelesen und grafisch visualisiert.

Für die beiden Programmteile *Regler* und *Consumer* ist vorgesehen, diese in Pthreads zu verarbeiten. Mit Hilfe dieser Pthreads besteht die Möglichkeit einer parallelen Programmabarbeitung. Infolgedessen können die Programmteile zeitgleich und unabhängig voneinander arbeiten. Dies hat den großen Vorteil, dass der Thread *Regler* ständig mit dem Teleskop kommunizieren kann und der *Consumerthread* gleichzeitig aktuelle Parameter übermittelt.

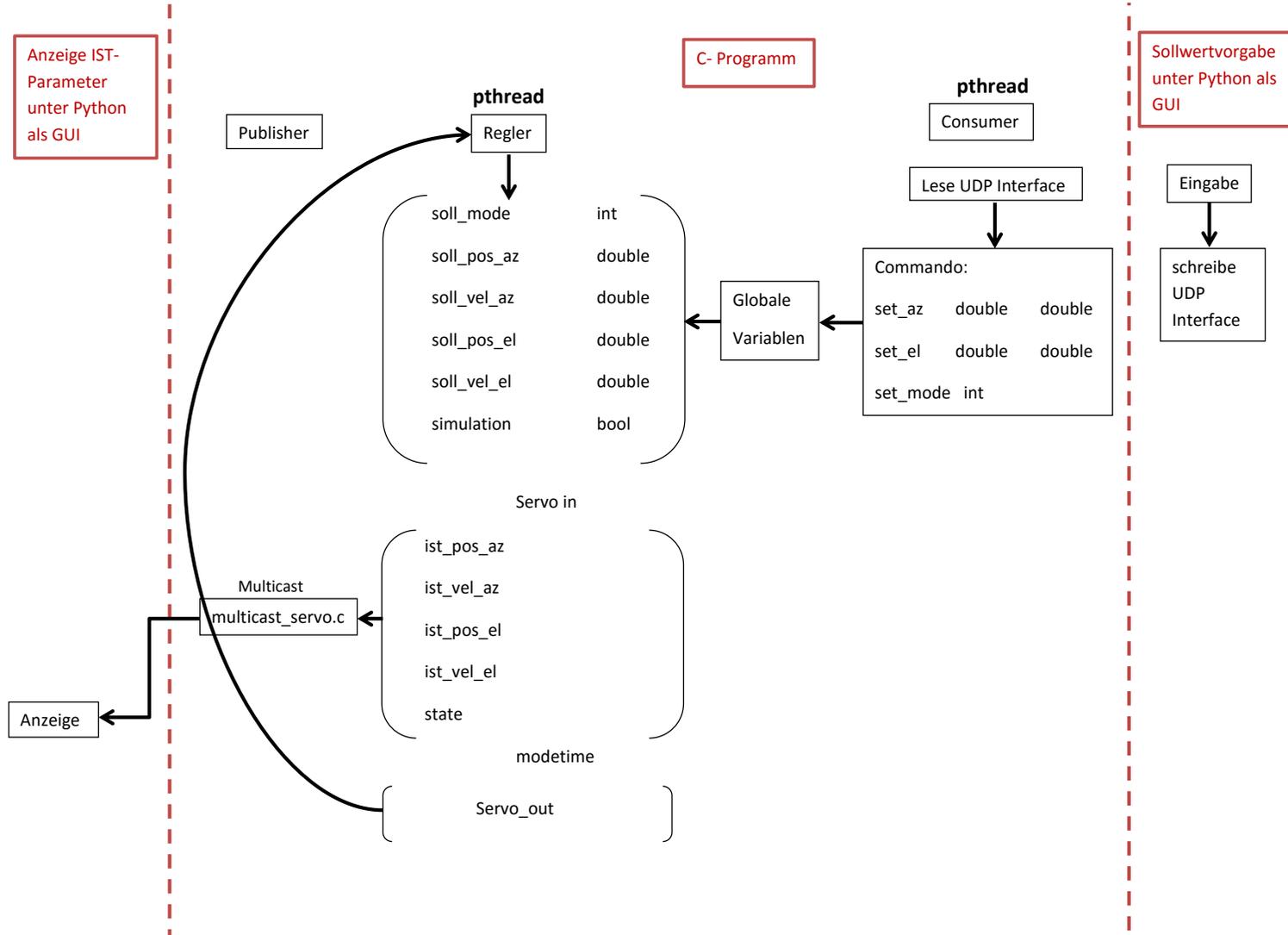


Abbildung 30: Software Entwurf

3.9.1 Implementierung der Regeleinheit und das Interface im LAN in C⁴⁸

Der Softwareentwurf aus Abb. 30 wird als Vorlage für die Programmimplementierung verwendet. Das C-Programm dient zur Kommunikation mit der Regeleinheit, zum Einlesen der vorgegeben Parameter vom UDP-Port und zum Senden der Daten über den Multicast-Port. Diese Implementierung wird im Folgenden kurz erörtert.

In Listing 1 ist die Parametervorgabe für die Regeleinheit dargestellt. Mittels der Funktion *ealMotionMovementMoveAbsolute()*, aus der Bosch Rexroth Bibliothek, können der Regeleinheit die Parameter Rotationsgeschwindigkeit, Beschleunigung, Verzögerung, Ruck und Winkel übergeben werden. Ist diese Funktion aktiv befindet sich die Regeleinheit in einer Positionsregelung.

Die Parameter der Position sowie Winkelgeschwindigkeit werden über den UDP-Port an das C-Programm übertragen. Unter zu Hilfenahme der parallelen Programmabarbeitung kann ein Thread ununterbrochen in der Positionsregelung bleiben, gleichzeitig von dem *Consumerthread*⁴⁹ die aktuellen Parameter entgegennehmen und diese sofort positionieren. Der Beschleunigungsparameter ist auf den Wert $0,8 \text{ min}^{-2}$ eingestellt, so ist sichergestellt, dass der Motor nicht zu schnell beschleunigt und schlimmstenfalls mechanische Bauteile beschädigt. Für die Verzögerung wurde der Wert von $0,1 \text{ min}^{-2}$ gewählt. Das Teleskop kann in diesem Fall langsam an die vorgegebene Position fahren und somit beispielsweise ein Überspringen vermeiden. Der Parameter Ruck wurde zunächst auf 0 gehalten. Mittels dieses Parameters kann das Anfahrverhalten des Radioteleskops beeinflusst werden. Dies ist jedoch erst sinnvoll, wenn der Motor an dem Teleskop befestigt ist und somit ein passender Wert im realen Betrieb approximiert werden kann.

Bevor das Radioteleskop die erste Positionierung absolvieren kann, wird zunächst die Bosch Rexroth Funktion *ealMotionMovementHome()* aufgerufen. Mit Hilfe dieser Funktion wird eine Referenzfahrt durchgeführt. Das Teleskop dreht sich, bis dieses ein Signal durch die Gabellichtschranke erhält. Schaltet diese Lichtschranke den digitalen Eingang der Regeleinheit, wird der Inkrementaldrehgeber referenziert. Dieser Vorgang geschieht immer bevor das Radioteleskop aus dem Stoppzustand in den Startzustand versetzt wird. Jedoch bewegt sich das Teleskop erst mit Hilfe der Funktion *ealMotionMovementMoveAbsolute()*. Für den Fall, dass der Benutzer die Vorgabe Start signalisiert, wird somit zunächst eine Referenzfahrt durchgeführt, bevor neue Parameter entgegengenommen werden. Wenn die Referenzfahrt durchgeführt ist und das Programm im Startzustand bleibt, wird die Referenzfahrt nicht nochmals abgearbeitet.

Solange der Zustand Start aktiv ist, befindet sich das Teleskop in einer Positionsregelung. In diesem Fall hält der Motor das System auf dem voreingestellten Winkel bzw. fährt eine neue Position an. Greift beispielsweise Wind an das Radioteleskop an, wird durch die Regelung das Teleskop auf der vorgegeben Position gehalten bzw. nachjustiert. Hinzu kommt, dass das System während einer radioastronomischen Messung die Erdrotation kompensieren und somit ständig positionsgeregelt bewegt werden muss. Ist das Radioteleskop positioniert, können die aktuellen Positionen mit Hilfe der Funktion *ealMotionGetActualPosition()* ausgelesen werden. Zudem wird über *ealMotionGetDiagnosisText()* der Zustand der Regeleinheit ermittelt. Diese beiden Zustände werden mit der Funktion *multicast()* an das Python Programm übergeben.

⁴⁸Um dem Umfang dieser Arbeit nicht unnötig in die Länge zu ziehen, wird nur auf die prägnanten Teile der Programmierung eingegangen. Der vollständige Programmcode ist im Anhang nachzulesen.

⁴⁹vgl. Abb. 30

Beendet der Benutzer die Freigabe des Systems, schaltet der Motor ab und die Motorbremse schließt. In diesem Zustand wird so lange geblieben, bis eine Eingabe des Benutzers das Teleskop freigibt. Ist dies der Fall beginnt der Programmablauf von vorne. Der Motor wird initialisiert, die Referenzfahrt durchgeführt usw.

```
2  if (result >= 0) {
3  //bevor Regeleinheit gestartet wird macht diese eine Referenzfahrt
4      result = ealMotionMovementHome(&handle, 0);

6  //so bleibt der Regler immer in der Regelung und es kann direkt ein Wert
      uebergeben werden
7  while(start && result >= 0){

9  // Positionsgeregelt
10     result = ealMotionMovementMoveAbsolute(&handle,0, pos, vel, acc, dec, 0);
11     result = ealMotionMovementWait(&handle, 0,100000);
12     result = ealMotionGetDiagnosisText(&handle,0,diagnosisTxt,1000, &numElements);

14 // Parameter auslesen
15     ealMotionGetActualPosition(&handle,0,&x);
16     multicast(x, diagnosisTxt);
17 }
```

Listing 1: Parameterübergabe an Regeleinheit

Die Kommunikation zwischen Programm und Regeleinheit sowie das Empfangen der UDP-Daten soll parallel ablaufen. Dies wurde mittels POSIX-Threads (Pthreads) realisiert, welche eine parallele Programmabarbeitung ermöglichen. Mit Hilfe dieser Methode kann das Radioteleskop ständig im Regelalgorithmus bleiben, gleichzeitig ankommende Parameter verarbeiten und eine Positionierung vornehmen.

In Listing 2 ist die Implementierung dargestellt.⁵⁰ Der Befehl `pthread_create` erstellt neue Threads. In diesem Fall ein Thread für die Kommunikation mit der Regeleinheit und einer zum Empfangen und Verarbeiten der Daten aus den UDP-Paketen. Die Programmabarbeitung der beiden Funktionen läuft parallel ab.

```
1 void *thread_regler(void *ch) {
3     //Programmablauf zur Kommunikation mit Regler
5 }
7 void *thread_consumer(void *ch) {
9     //Programmablauf Kommunikation UDP
11 }
13 int main() {
15     pthread_t p1, p2;
16     pthread_create(&p1, NULL, thread_regler, &ch1);
17     pthread_create(&p2, NULL, thread_consumer, &ch2);
19     return 0;
21 }
```

Listing 2: Parallele Programmabarbeitung

Damit die Werte für die Positionierung aus dem UDP entgegengenommen werden können, muss in dem Programm ein UDP-Client implementiert werden. Dieser Client soll die gesendeten Daten empfangen, verarbeiten und an die Regeleinheit weiterleiten. Um die gesendeten Parameter zu empfangen, benötigt der Client die IP-Adresse sowie den Port des Servers, auf welchen dieser sendet. Die empfangenen Daten aus dem UDP werden als String Daten übermittelt. Dieses Datenpaket wird entgegengenommen und in das richtige Format, zur Übergabe an die Regeleinheit, konvertiert.

Der Programmcode des UDP-Clients wird hier nicht nochmals explizit aufgeführt, dies ist ein Standard Programmcode, welcher aus dem Buch *core Python programming*⁵¹ entnommen und in einen C-Code adaptiert wurde.

Die aktuellen Werte des Radioteleskops werden über einen Multicast übertragen, mit dem Vorteil, dass Nachrichten gleichzeitig über eine Adresse an mehrere Teilnehmer oder an eine geschlossene Benutzergruppe übertragen werden können. Die Implementierung ist ähnlich zu der des UDP-Clients. Jedoch wird in diesem Fall der Server programmiert. Dem Multicast muss eine IP-Adresse sowie ein

⁵⁰Der Inhalt die Funktionsaufrufe wurde zur Übersichtlichkeit weggelassen, kann jedoch im Anhang nachvollzogen werden.

⁵¹vgl. [J. Chun 2001, S.580]

Port vorgegeben werden, auf welchen dieser senden bzw. empfangen kann. Die Datenübertragung geschieht wie bei dem UDP über einen String. Damit die Teilnehmer der Multicastgruppe den String auslesen können, muss dieser in einem Protokoll implementiert werden. Für dieses Verfahren bietet sich die Serialisierung JSON an. JSON ist ein Datenaustauschformat, in welchem Daten als Liste von Schlüssel/ Wert-Paaren dargestellt werden.

In Listing 3 ist ein Teil des Programm-Codes zum Erstellen eines JSON Strings und die Übertragung über den Multicast aufgeführt. Zunächst muss ein JSON Objekt erstellt werden. In dieses Objekt werden drei Strings geschrieben. Der erste String, *posstring*, ist die aktuelle Position des Radioteleskops. Der zweite String ist eine Zeitmarke, welche nach dem positionieren des Radioteleskops mitgeschickt wird. Über diese Zeitmarke kann evaluiert werden ob die angezeigte Position noch aktuell ist oder eine neue Positionierung erforderlich ist. Der dritte String ist der Zustand der Regeleinheit. Die drei Strings werden in das JSON-Format gebracht. Würde der JSON String ausgegeben werden, würde nun folgendes auf der Konsole erscheinen:

```
{ "posaz": "234.000000", "time": "Tue Aug 23 11:22:56 2016", "state": "A0161 Antriebsgeführtes
Positionieren mit Achsreglersteuerwort" }
```

In diesem Fall sind die Schlüsselwörter: *posaz*, *time* und *state*. Jedem Schlüsselwort ist der nachfolgende String zugeordnet. Somit kann der Multicastempfänger das Datenpaket verarbeiten und mittels der Schlüsselwörter die benötigten Parameter auswerten. Der Vorteil dieses Formats ist, dass es ohne viel Aufwand erweitert werden kann. Beispielsweise wäre es nach der Implementierung der Elevationsachse sinnvoll, diese Parameter dem Format anzuhängen. Es müsste ein Schlüssel deklariert und die aktuelle Position in den String geschrieben werden.

```
1 // Erstellen JSON Objekt
2 json_object * jobj = json_object_new_object();

4 // Erstellen JSON Strings Position , Timestamp , Reglerzustand
5 json_object *jstring1 = json_object_new_string(posstring);
6 json_object *jstring2 = json_object_new_string(asctime(ts));
7 json_object *jstring3 = json_object_new_string(achszustand);

9 // JSON-Format erstellen
10 json_object_object_add(jobj, "posaz", jstring1);
11 json_object_object_add(jobj, "time", jstring2);
12 json_object_object_add(jobj, "state", jstring3);

14 // laenge von string
15 laenge = strlen(json_object_to_json_string(jobj));

17 // Dateneubertragung Multicast
18 sendto(socket_descriptor, json_object_to_json_string(jobj), laenge, 0, (struct
sockaddr*) &address, sizeof(address));
```

Listing 3: JSON Implementierung

3.9.2 Implementierung von UDP und Multicast in Python⁵²

Das Benutzerinterface wird unter Zuhilfenahme von Python implementiert. Zum Senden wird ein Unicast verwendet, wobei beim Empfangen ein Multicast realisiert wird. Der Unicast wird in einem UDP implementiert. Zur besseren Benutzerfreundlichkeit werden diese in grafische Oberflächen eingebunden.

Das Senden und Empfangen der Unicast- bzw. der Multicast-Parameter sind in Python die exakten Gegenstücke des im C-Programm implementierten Codes. Die IP-Adresse und der Port müssen übereinstimmen, damit eine Kommunikation aufgebaut werden kann.

Die GUI wurde mittels dem Programmierwerkzeug Qt-Designer realisiert. Qt ist ein objektorientiertes Framework zur Entwicklung plattformunabhängiger Anwendungen. Das Erstellen einer Oberfläche mittels Qt-Designer ist nicht an eine Programmiersprache gebunden und kann ohne Änderungen in einer Vielzahl von Hochsprachen verwendet werden. Die Oberflächenbeschreibung wird in einer XML-Datei gespeichert. Die grafische Benutzeroberflächen kann aus verschiedenen Elementen per Drag & Drop zusammengesetzt werden.

In Listing 4 ist ein Programmausschnitt zum Einlesen der eingegeben Parameter des Users dargestellt. Zunächst wird über den Befehl *import* die erstellte XML-Datei des Qt-Designers in das Python Programm eingebunden. Um eine Verbindung zwischen der Anwendung und der GUI-Elemente herzustellen, wird der Objektname *ui* als Prefix vorgeschoben. Qt-Anwendung sind ereignisgesteuert. In diesem Fall ist die Reihenfolge der Programmabarbeitung vom Benutzer vorgegeben und arbeitet nicht, wie bei beispielsweise in der Konsolenanwendung, die einzelnen Schritte nacheinander ab. Diese Abarbeitung regelt der Programmaufruf *QApplication(sys.argv)*. Qt verwendet eine Signal-Slot-Verbindung, diese Verbindung wird automatisch getrennt, wenn eines der kommunizierenden Objekte gelöscht wird. Mit der Funktion *QObject.connect* wird eine Signal-Slot-Verbindung eingerichtet. Mit dem Signal *clicked()* wird der Button *send* mit dem Objekt *setpushOk* verbunden. Infolgedessen ist festgelegt, dass die Parameter die in der Funktion *setpushOk()* eingegeben werden, über das Klicken des Buttons *send* übermittelt werden. Die eingegebenen Parameter werden in ein Datenpaket gepackt und über UDP an das Radioteleskop übermittelt.

⁵²Wie im C-Programm wird nur auf die prägnanten Programmcodes eingegangen. Das vollständige Programm befindet sich im Anhang.

```

1  import guidata #importieren der erstellten Qt-Designer Datei

4  class MyMuxerForm(QtGui.QMainWindow):
5  def __init__(self, parent=None):

7  #QtGui Klassen zum Erstellen und Zeichnen von grafischer Oberflaechen hinterlegt.
8  QtGui.QMainWindow.__init__(self, parent)

10 #Ui_MainWindow() Funktion aus guidata
11 self.ui = guidata.Ui_MainWindow()
12 self.ui.setupUi(self)

14 #Slots einrichten
15 QtCore.QObject.connect(self.ui.sent,QtCore.SIGNAL("clicked()"),self.setpushOk)
16 QtCore.QObject.connect(self.ui.Eingabepos,QtCore.SIGNAL('activated(QString)'),self.
    setpushOk)

18 def setpushOk(self):

20 #isChecked -> auslesen radiobutton
21 startWert = int(self.ui.Freigabe.isChecked())

23 #von float auf string da UDP String benoetigt
24 start = str(startWert)

26 #text -> auslesen von editline
27 posWert = float(self.ui.Eingabepos.text())
28 pos = str(posWert)

30 velWert = float(self.ui.Eingabevel.text())
31 vel = str(velWert)

33 #erstellt String zum Uebertragen der Daten ueber UDP
34 data = start + ";" + vel + ";" + pos
35 print data
36 udpCliSock.sendto(data, ADDR)

38 if __name__ == "__main__":

40 app = QtGui.QApplication(sys.argv)
41 myapp= MyMuxerForm()
42 myapp.show()

44 sys.exit(app.exec_())

```

Listing 4: UDP und Import Qt-Designer

Das C-Programm übermittelt über Multicast die Daten der aktuellen Azimut Position sowie eine Zeitmarke, wann das letzte Mal diese Position übertragen wurde. Dies wird in dem Code eingebunden, vor dem Hintergrund, dass in Folge der Erdrotation sich das zu messende Objekt, aus Sicht des Betrachters bewegt. Zusätzlich wird der Zustand der Regeleinheit übertragen. Diese Parameter werden für den Benutzer in der GUI visualisiert.

Das Programm wurde in drei Dateien aufgeteilt. In der ersten Datei ist der erstellte XML-Code des Qt-Designers eingebunden. Dieser XML-Code beschreibt die grafische Oberfläche. Der zweite Teil stellt die Kommunikation zu dem Multicast mit dem C-Programm her und übermittelt die Parameter an das Hauptprogramm. Zudem ist in diesem Teil eine Routine implementiert, welche überprüft, ob eine Multicast-Verbindung besteht. Ist dies nicht der Fall werden die Anfragen an den Port unterbrochen und nach 60 Sekunden nochmals eine Verbindung getestet. Dies wird solange

durchgeführt bis von dem C-Programm Parameter übermittelt werden. In diesem Fall wird die Verbindung wieder aufrechterhalten.

In Listing 5 ist der Programmteil dargestellt, welcher die Verbindung zum Multicast herstellt und die empfangenen Daten an das Hauptprogramm übergibt. Der Aufbau der Multicast-Verbindung ist nicht dargestellt, da dieser dem Verbindungsaufbau des UPD ähnelt.

```
1 # Programmteil vorher Verbindung Multicast , vollstaendiges Programm im Anhang
2
3 class Base(QtGui.QMainWindow):
4     def __init__(self):
5         QtGui.QMainWindow.__init__(self)
6
7     # Workerthread erstellen und starten
8     self.myworkerthread = QtCore.QThread()
9     self.myworker = MulticastWorker()
10    self.myworker.moveToThread(self.myworkerthread)
11
12
13 # Winkel
14 def setFullDegree(self, data_in):
15     posaz = float(data_in) # weil winkel double ist
16     return "%s" % posaz
17
18 #Zeit uebermitteln
19 def setTime(self, data_in):
20     time = data_in
21     return "%s" % time
22
23 #Zustand der Achse
24 def setstate(self, data_in):
25     state = data_in
26     return "%s" % state
```

Listing 5: Multicast Base

In Listing 6 werden die empfangenen Parameter an die GUI übergeben. Aus dem Multicast kann das JSON-Format ausgelesen werden. Die Schlüsselwörter werden in der jeweiligen Funktion aufgerufen, in Listing 5 verarbeitet und zurückgegeben.

Am Beispiel der Zeitmarke ist der Name *Zeit*, der deklarierte Block aus dem Qt-Designer. Mittels *self.setTime()* wird aus dem Programm, Listing 5, die Zeitmarke aus dem JSON-Format ausgewertet und der Funktion *setText()* übergeben. *setText()* übermittelt dem Qt-Designer die Zeitmarke und gibt den Befehl diesen Inhalt auf der Oberfläche zu visualisieren.

```

1 class Sdh(Base):
2 def __init__(self):
3     Base.__init__(self)
4     # create GUI from ui-file
5     uic.loadUi('lesemulticast.ui', self)
6     self.show()

8 # Workerthread starten
9 self.myworker.paramDataChanged[object].connect(self.updateGui, QtCore.Qt.
    QueuedConnection)
10 self.myworkerthread.started.connect(self.myworker.start)
11 self.myworkerthread.start()

14 # GUI-Aktualisierung
15 @QtCore.pyqtSlot(object)
16 def updateGui(self, dict):

19 # GUI

21 #setze Uhrzeiten
22 self.Zeit.setText(self.setTime(dict['time']))

24 # Azimut
25 self.posaz.setText(self.setFullDegree(dict['posaz']))#float

27 # lese Zustand des Reglers
28 self.state.setText(self.setstate(dict['state']))

30 # Ende GUI

```

Listing 6: Multicast Main

3.9.3 Grafische Benutzeroberfläche

Für die Benutzereingabe wurde eine GUI entwickelt, siehe Abb. 31. Die GUI wurde mit dem Programm Qt-Designer entworfen. Der Vorteil von diesem Programm besteht darin, dass dieses nicht an eine Programmiersprache gebunden ist. Qt-Designer speichert Dateien in XML-Format und kann in Python eingebunden werden. Der Benutzer hat die Möglichkeit dem Regler eine Rotationsgeschwindigkeit sowie eine Position vorzugeben. Das C-Programm wartet auf die Eingabe des Benutzers.

Um die Parameter an die Regeleinheit übergeben zu können, muss der Button *Freigabe* aktiviert sein. Rotationsgeschwindigkeit und Winkel können über die Tastatur eingegeben werden. Der Button *send* dient zur Übertragung der Daten, solange dieser nicht betätigt ist, werden keine Parameter an den Regler übermittelt. Ist der Button *Freigabe* betätigt, eine Rotationsgeschwindigkeit und ein Winkel eingegeben, kann durch Betätigung des Buttons *send* die Parameter übertragen werden. In diesem Fall sendet das Programm über das UDP, die eingegeben Parameter an das C-Programm. Der Motor löst die Bremse, macht zunächst eine Referenzfahrt und positioniert im Folgenden das Teleskop auf den vorgegeben Winkel. Das Radioteleskop befindet sich in der Regelung, solange bis der *Freigabe* Button zurückgesetzt und dies mittels *send* an den Regler übertragen wird. Bleibt der Button *Freigabe* aktiviert und werden neue Parameter eingegeben, positioniert das Radioteleskop diese, ohne eine erneute Referenzfahrt durchzuführen. Wird der *Freigabe* Button zurückgesetzt und dieses übertragen, wird die Bremse des Motors geschlossen, das System ist mechanisch arretiert und das Programm

wartet auf eine erneute Freigabe.

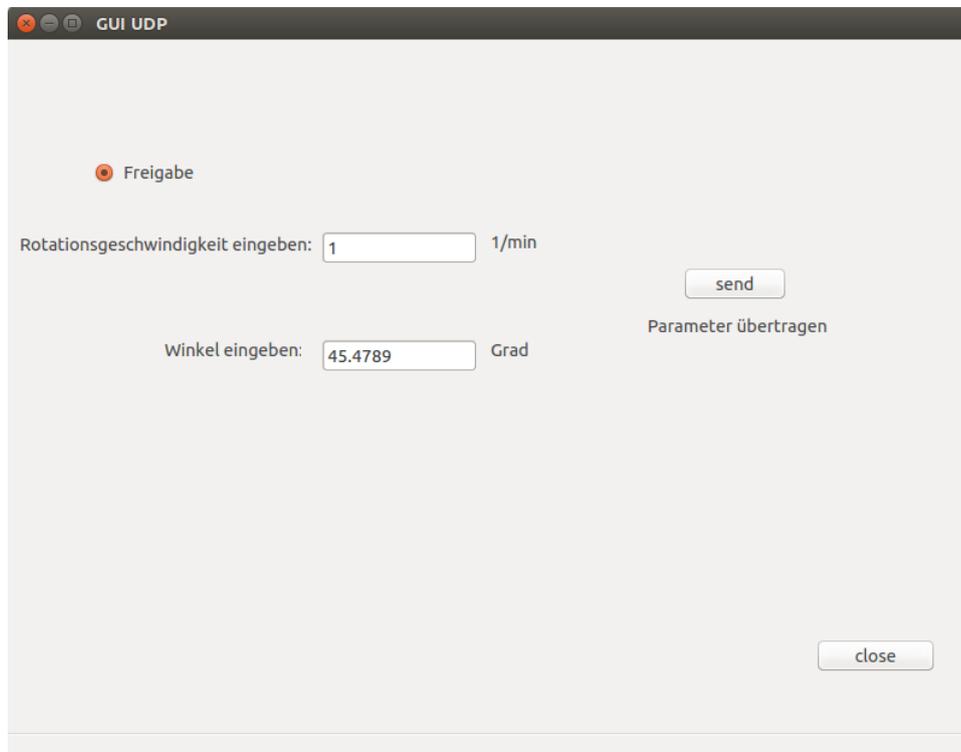


Abbildung 31: GUI Parametereingabe

Die GUI des Multicast dient allein zur Visualisierung des aktuellen Zustands des Radioteleskops bzw. der Regeleinheit. Diese Oberfläche zeigt den aktuellen Zustand der Regeleinheit an. Zudem werden die Positionen in Azimut und die Uhrzeit der Datenübertragung ausgegeben, siehe Abb. 32.

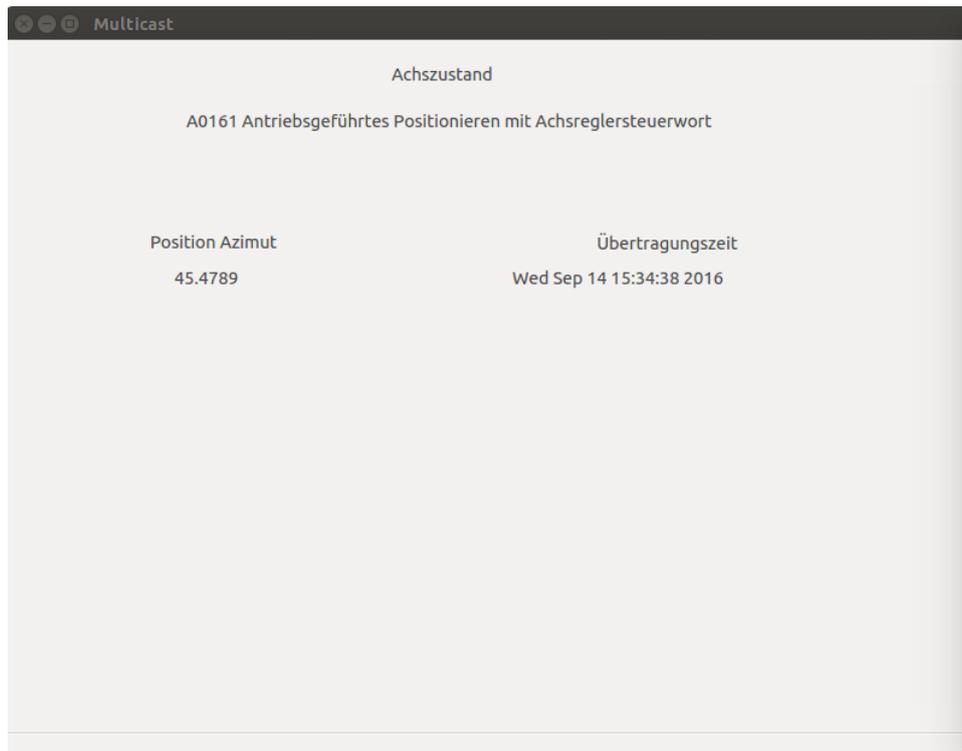


Abbildung 32: GUI Parameterausgabe

3.10 Inbetriebnahme des Azimutantriebs am 4-Meter-Radioteleskop

Die GUI ist implementiert, die mechanischen Bauteile sind gefertigt. Infolgedessen kann sich mit der Inbetriebnahme des Radioteleskops beschäftigt werden. Das System befindet sich zurzeit in der Werkshalle. Der 4-Meter-Spiegel wurde zunächst nicht montiert, da sonst die Gefahr eines Kippens des Gestells besteht. Die mechanische Befestigung ist wie in Kapitel 3.5.2 an dem Teleskop montiert. Die elektrischen Komponenten wurden angeschlossen sowie der Rechner auf welchem sich das Linux Betriebssystem befindet, mit der Regeleinheit verbunden.

Für die Inbetriebnahme konnten leider nicht die beiden vorgesehen Endschalter verwendet werden, da diese aus dem Lager des Instituts entnommen wurden. Während dem Verlauf dieser Arbeit wurden die Bauteile jedoch benötigt. Neue Komponenten sind bestellt. Aus diesem Grund konnte diese Variante nicht an dem Teleskop selbst getestet werden, lediglich versuchsweise ohne den Anbau am Radioteleskop.

Mit diesem Aufbau besteht die Möglichkeit, dass ein Benutzer innerhalb des LAN, eine Parametervorgabe dem Teleskop übermittelt. Diese Eingabe geschieht über die implementierte GUI. Nach Eingabe und Übermittlung der Parameter, hat das Teleskop zunächst eine Initialisierungsfahrt durchgeführt. Im Anschluss dieser Initialisierungsfahrt, wurde der vorgegebene Winkel positioniert. Bleibt das System im Betriebszustand, kann der Benutzer neue Parameter vorgeben und das Teleskop führt eine Positionierung, ohne eine erneute Referenzfahrt, durch. Durch Änderung der Parameter Rotationsgeschwindigkeit konnte die Positionierung schneller und langsamer durchgeführt werden. Infolge der Parametervorgabe des C-Codes, wird der Spiegel langsam auf die vorgegebene Geschwindigkeit beschleunigt und vor Erreichen des vorgegebenen Winkels langsam reduziert. Nach Abschluss jeder Positionierung werden die aktuelle Position über Multicast übertragen und die Teilnehmer der Multicastgruppe können diese Parameter auslesen.

Wenn der Benutzer mit Hilfe der grafischen Oberfläche das System ausschaltet, wird der Motor abgeschaltet und die Bremse befestigt. Somit ist das Teleskop mechanisch arretiert und kann nicht mehr bewegt werden. In diesem Zustand werden keine Parameter für die Positionierung entgegengenommen, solange keine Freigabe erfolgt. Gibt der Benutzer die Rotation frei, wird der Motor initialisiert, die Motorbremse löst sich und der Vorgang beginnt von vorne.

Im Verlauf der Testfahrten stellte sich heraus, dass der Mitnehmer, welcher Rotationen über 360° ermöglicht, teilweise verkanntet und somit unsauber in der Führung läuft. Die Idee besteht darin, den Mitnehmer mit einem Nadelrollenlager zu führen und mit dieser Methode das Verklemmen zu vermeiden.

4 Ergebnis

4.1 Fazit

In dieser Bachelorarbeit wurde sich zunächst mit den Rahmenbedingungen zur Automatisierung eines 4-Meter-Radioteleskops im Azimut auseinandergesetzt. Mit diesem Vorgehen konnte in Erfahrung gebracht werden, für welche Bedingungen ein Radioteleskop, in solchen Dimensionen, ausgelegt sein muss. Zwei Gesichtspunkte waren besonders kritisch zu evaluieren. Zum einen die Winkelgenauigkeit des Radioteleskops. Wird diese nicht erreicht, beeinträchtigt dies die Durchführbarkeit einer radioastronomischen Messung. Zum anderen benötigen die Antriebsmotoren genügend Leistung, damit diese bei hohen Windlasten ihre Position halten können. Für beide Fälle wurden Berechnungen durchgeführt, vor dem Hintergrund, dass dies die Grundlage der mechanischen Auslegung der Antriebs- sowie Winkelmesseinheit ist.

Ich habe mich für das Antriebssystem von Bosch Rexroth entschieden. Da diese, nach meinen Auslegungen und Vorstellungen, passende Bauteile angeboten haben und gemeinsam nach einer optimalen Lösung zum Einhalten der Randbedingungen geforscht wurde. Zudem bestand die Möglichkeit das Interface zwischen der Regeleinheit und einem C-Programm zu implementieren. Mittels einer guten Kommunikation, einigen Telefonaten und regem E-Mail Verkehr konnten Probleme, Überlegungen und Ideen thematisiert werden.

Die Kraftübertragung des Motors auf das Radioteleskop geschieht über ein Präzisionsplanetenge triebe. Diese besitzen ein geringes Verdrehspiel und somit eine hohe Genauigkeit. Auf dieses Getriebe wurde ein Standardzahnrad montiert, welches im Zuge der geringen Toleranzanforderungen gehärtet, geschliffen und mit einer evolventen Verzahnung versehen wurde.

Infolge der Auswahl der mechanischen Bauteile wurde die anfängliche Überschlagsrechnung nochmals verifiziert und die Kombination Motor, Getriebe und Zahnrad aufeinander angepasst. Auf diese Weise war sichergestellt, dass die Rahmenbedingungen eingehalten werden.

Die Rotationsgeschwindigkeit des Teleskops, im Azimut, war mit einer Umdrehung pro Minute vorgegeben. In Bezug auf die durchgeführten Berechnungen würde theoretisch die Möglichkeit bestehen, dass das System innerhalb von 5,12 Sekunden eine vollständige Umdrehung absolviert. Während der Inbetriebnahme wurden Winkelgeschwindigkeiten bis zu 3 min^{-1} durchgeführt. Folglich ist die Rotationsgeschwindigkeit des 4-Meter-Radioteleskops, für radioastronomische Beobachtungen, ausreichend.

Die Winkelmessung ist mit einem externen Drehgeber realisiert worden, zum Kompensieren der systematischen Messabweichungen des Motordrehgebers, in Bezug auf die Getriebeübersetzung und das Flankenspiel der Zahnradverbindung. Übertragen wird die Drehbewegung des Gebers unter Zuhilfenahme einer Zahnradverbindung. Zur Kompensation des Flankenspiels, wurde in Folge dieser Arbeit ein federverspanntes Zahnrad entwickelt und dieses in der Werkstatt gefertigt.

Zur Projektierung eines geeigneten Winkelmesssystems musste sich zunächst mit der benötigten Winkelauflösung, zur Durchführung einer radioastronomischen Messung, befasst werden. Diese Auflösung steht in Abhängigkeit zum Durchmesser des Parabolspiegels sowie der kürzesten gemessenen Wellenlänge. Es wurde sich für einen Inkrementaldrehgeber, welcher als Ausgangssignal einen TTL-Pegel zur Verfügung stellt, entschieden. Die Auflösung des Drehgebers konnte in Folge

der ermittelten Parameter ausgewählt werden und verfügt somit über die benötigte Genauigkeit, ein Radioteleskop hinreichend genau zu positionieren. Zum Referenzieren des Inkrementalgebers dient eine Gabellichtschranke als Referenzpunkt, welche nach Starten des Systems einmalig positioniert wird.

Damit ein Radioteleskop adäquate Messungen durchführen kann, benötigt dieses, im Azimut, einen Rotationsbereich größer 360° . Für diesen Fall wurde ein Mitnehmer entworfen, welcher am Ende einer Führung die Endlagen schaltet. In Bezug darauf, dass das Teleskop von dem Observatorium aus nicht zu sehen ist, wurde sich für ein redundantes Messsystem entschieden. Zu diesen Hardwareendlagen wurden zusätzliche Softwareendlagen implementiert, welche einen bestimmten Winkelbereich zum Positionieren des Teleskops eingrenzen.

Die Befestigung der Antriebs- sowie Winkelmesssysteme wurde im Verlauf dieser Arbeit konstruiert, ein CAD-Modell entwickelt und aus diesem technische Zeichnungen entworfen. Diese Bauteile wurden gefertigt und zusammen mit den Antriebs- und Winkelmesssystemen an dem Radioteleskop montiert. Damit diese Konstruktion gegen Witterungseinflüsse geschützt ist, wurde zusätzlich eine Abdeckung entworfen, welche über die Baugruppe montiert wird. Die Bauteile sind zwar gegen die vorhandenen Witterungseinflüsse ausgelegt, wie jedoch bei dem Elevationsantrieb zu sehen, ist ein zusätzlicher Schutz nicht von Nachteil. Der Schaltschrank des Radioteleskops wird in einen nahegelegenen Keller verlegt und ist somit vor Umwelteinflüssen geschützt.

Im Verlauf dieser Arbeit wurde sich mit der softwareseitigen Inbetriebnahme des Radioteleskops beschäftigt. Die Vorgabe dieses Projekts bestand darin, ein netzwerkbasierendes Kontrollsystem, zum Positionieren des Radioteleskops, zu implementieren. Die Kommunikation mit dem Radioteleskop ist innerhalb des LAN möglich. Für das Senden der Daten wurde ein Unicast verwendet, wobei das Empfangen in einem Multicast eingebunden ist. Das Interface der Unicast-Verbindung verwendet UDP. Ein Benutzer kann dem Teleskop, anhand einer GUI, Positionierungsdaten vorgeben. Die Unicast-Verbindung übermittelt die Daten an ein C-Programm, welche dort verarbeitet werden. Bosch Rexroth stellt ein Interface zur Kommunikation zwischen dem C-Programm und der Regeleinheit zur Verfügung, welches die empfangenen Daten über Ethernet an die Regeleinheit weiterleitet. Infolgedessen positioniert das Teleskop die vorgegebenen Parameter. Der Zustand des Radioteleskops wird an das C-Programm übertragen, welches die Daten mit Hilfe eines Multicastroutings an eine Teilnehmergruppe weiterleitet. Die empfangenen Daten aus dem Multicast sind zur Visualisierung in einer GUI eingebettet. Das Interface zum Übertragen und Auslesen der Parameter ist in Python implementiert.

Zudem wurde die elektrische und mechanische Inbetriebnahme, im Azimut, an dem Radioteleskop durchgeführt. Der Benutzer konnte wie vorgesehen dem Radioteleskop Parameter vorgeben. Nachdem das System erstmalig gestartet wurde, hat dieses eine Initialisierungsfahrt durchgeführt. Die Gabellichtschranke wurde angefahren und meldet den Zustand Referenzpunkt erreicht, somit wird der Inkrementaldrehgeber referenziert und positioniert anschließend den vorgegeben Winkel. Im Folgenden konnten neue Winkel und Rotationsgeschwindigkeiten vergeben werden und das Radioteleskop hat diese, ohne erneute Referenzfahrt, positioniert. Die aktuellen Parameter des Radioteleskops können innerhalb der Multicastgruppe verfolgt werden. Solange das Radioteleskop im Startzustand ist, befindet sich dieses in einer Positionsregelung, somit ist sichergestellt, dass das Teleskop die vorgegebenen Positionen hält. Im Falle einer angreifenden Windlast ist unter Zuhilfenahme der Regelung sichergestellt, dass die Position gehalten bzw. nachjustiert wird. Wird das System abgeschaltet, schließt die Motorbremse und das Radioteleskop ist mechanisch arretiert.

4.2 Ausblick

Infolgedessen, dass die Ansteuerung des Azimutantriebs nach den Vorstellungen des Instituts implementiert wurde und dieser funktionsfähig ist, wurde mir die Möglichkeit geboten, den Elevationsantrieb von Bosch Rexroth zu bestellen. Dieser Antrieb wird in den nächsten Wochen eintreffen. Da die Antriebseinheit in Elevation und Azimut von einem Hersteller ist, entfallen Kompatibilitätsprobleme der Bauteile. Der Antriebsmotor der Elevation ist derselbe wie des Azimutantriebs. Dies bietet den Vorteil, dass das Interface der Ansteuerung im Azimut übernommen werden kann und lediglich an die Elevationsachse angepasst werden muss. Der Programmcode ist so implementiert, dass eine Einbindung der Elevationsachse in das Interface vorgesehen ist.

Durch den zeitlichen Rahmen dieser Bachelorarbeit war es leider nicht mehr möglich Testfahrten zum evaluieren der Genauigkeit des Azimutantriebs durchzuführen. Unter Zuhilfenahme der Parameter aus dem Multicast-Interface, besteht die Möglichkeit, die Positionierung des Radioteleskops zu plotten. Mit Hilfe dieser Aufzeichnung kann die Genauigkeit des Antriebs bewertet werden. Zudem würde die Möglichkeit bestehen, unter Zuhilfenahme dieses Plots, die Regeleinstellungen zu optimieren. Zudem sollte das Schaltverhalten der Gabellichtschranke untersucht werden, da von dieser die Genauigkeit des Messsystems abhängt. Das Schalten der Lichtschranke ist abhängig davon, aus welcher Richtung die Lichtschranke geschaltet wird. Da das Bauteil, welches die Lichtschranke schaltet eine bestimmte Dicke besitzt, besteht ein Unterschied aus welcher Richtung die Lichtschranke durchfahren wird. Dies führt zu vermeidbaren systematischen Messabweichungen und muss im Programm-Code implementiert werden.

Ist der Elevationsantrieb implementiert, sollte sich mit der Ephemeridenrechnung auseinandergesetzt werden, da es für die astronomischen Messungen nicht ausreicht die Winkel in Azimut und Elevation zu kennen. In der Ephemeride werden die Positionswerte sich bewegender astronomischer Objekte bezogen auf ein astronomisches Koordinatensystem ermittelt. Ist diese Rechnung in dem Programm implementiert, kann die GUI mit einer Liste von signifikanten Objekten im Kosmos erweitert werden und das Radioteleskop diese autonom positionieren sowie eine Nachführung des Teleskops durchführen.

Ein weiterer wichtiger Gesichtspunkt, welcher in folgenden Arbeiten durchgeführt werden müsste, ist die EMV-Messung. Bevor das Radioteleskop die ersten Beobachtungen durchführt, sollte diese Messung durchgeführt werden, um Störungen in der eigenen Messung bzw. der Messung des 100-Meter-Radioteleskops zu vermeiden. Für den Fall, dass die Störungen der Bauteile die Messungen beeinflussen, wäre eine Abschirmung dieser sinnvoll.

Literatur

Baumann 1994

BAUMANN, Prof. Dr.-Ing. E.: *Vermessungskunde Lehr- und Übungsbuch für Ingenieure*. Bonn : Fred. Dümmers Verlag, 1994. – ISBN 3-427-79044-4

Bernstein 2013

BERNSTEIN, Herbert: *Messelektronik und Sensoren*. München : Springer Verlag, 2013. – ISBN 978-3-658-00549-8

Eichen 2016

EICHEN, Wetterstation: *Amateurfunk- und Wetterstation DH3PAE*. "<http://eichenwetter.homelinux.net/linuxhd/>". Version: 2016. – abgerufen am 20.05.2016

Erbrecht u. a. 2008

ERBRECHT, Prof. Dr. R. ; KÖNIG, Dr. H. ; MARTIN, Karlheinz ; PFEIL, Wolfgang ; WÖRSTENFELD, Willi: *Das große Tafelwerk*. 1. Berlin : Cornelsen, 2008. – ISBN 978-3-464-571467-0

Fischer u. a. 2008

FISCHER, Ulrich ; HEINZLER, Max ; NÄHER, Friedrich ; PAETZOLD, Heinz ; GOMERINGER, Roland ; KILGUS, Roland ; OESTERLE, Stefan ; STEPHAN, Andreas: *Tabellenbuch Metall*. 44. Haan-Gruiten : Verlag Europa Lehrmittel, 2008. – ISBN 978-3-8085-1724-6

Hau 2003

HAU, Erich: *Windkraftanlagen Grundlagen, Technik, Einsatz, Wirtschaftlichkeit*. 3. München : Springer Verlag, 2003. – ISBN 978-3-662-10949-6

Heidenhain 2016

HEIDENHAIN: *Datenblatt Drehgeber Heidenhain*. "https://www.heidenhain.de/fileadmin/pdb/media/img/349529-1G_Drehgeber_de.pdf". Version: 2016. – abgerufen am 01.06.2016

Hoischen u. Hesser 2013

HOISCHEN, Hans ; HESSER, Wilfried: *Technisches Zeichnen*. 33. Berlin : Cornelsen Verlag, 2013. – ISBN 978-3-589-24194-1

INFOTRIP 2016

INFOTRIP: *GNSS- Globale Navigationssatellitensysteme*. "http://kompendium.infotrip.de/files/wdb/GRAFIK/2700_KOMMUNIKATIONSTECHNIK/2750_GNSS/ABB_2750_02_31_EuA.gif". Version: 2016. – abgerufen am 13.05.2016

J. Chun 2001

J. CHUN, Wesley: *core Python programming*. Upper Saddle River : Prentice Hall PTR, 2001. – ISBN 0-13-026036-3

Kurzweil u. a. 2008

KURZWEIL, Peter ; FRENZEL, Bernhard ; GEBHARD, Florian: *Physik Formelsammlung*. 1. Wiesbaden : Friedr. Vieweg & Sohn Verlag, 2008. – ISBN 978-3-8348-0251-4

Max-Planck 2007

MAX-PLANCK: *Lofar*. "http://www.mpifr-bonn.mpg.de/552643/Eff_Lofar.jpg". Version: 2007. – abgerufen am 06.05.2016

Max-Planck 2016

MAX-PLANCK: *Fakten*. "<http://www.mpifr-bonn.mpg.de/effelsberg/fakten>". Version: 2016. – abgerufen am 19.08.2016

Max-Planck-Flyer 2014

MAX-PLANCK-FLYER: Max-Planck Institut für Radioastronomie. (2014)

Unsöld u. Baschek 2005

UNSÖLD, Albrecht ; BASCHEK, Bodo: *Der neue Kosmos*. 7. Leipzig : Springer Verlag, 2005. – ISBN 978-3-662-06530-3

Verschuur 1988

VERSCHUUR, Gerrit L.: *Die phantastische Welt der Radioastronomie*. 1. Basel : Birkhäuser Basel, 1988. – ISBN 978-3-0348-6665-1

Wittel u. a. 2015

WITTEL, Herbert ; MUHS, Dieter ; JANNASCH, Dieter ; VOSSIEK, Joachim: *Roloff/Matek Maschinenelemente*. 22. München : Springer Verlag, 2015. – ISBN 978-3-658-09082-1

A Anhang

➔ ECOVARIO® 114 D - 2-Achs-Servoverstärker
9-2



Ausstattung

- EtherCAT/Ethernet-Schnittstelle
- CANopen
- Profibus DP, Profinet IRT
- RS232-Schnittstelle zur Parametrierung via PC
- RS485-Schnittstelle
- Sicherheitsfunktion „STO“ gemäß EN 61800-5-2

Funktionen

- Betrieb von 2- und 3-phasigen bürstenlosen Synchronmotoren
- Betrieb von 2- und 3-phasigen Synchron-Linearmotoren
- Betrieb von bürstenbehafteten DC-Servomotoren
- Drehmoment-/Kraft-, Geschwindigkeits- und Positionsregelung
- Positionsrückführung über Inkrementalencoder: RS422, optional SINCOS, 2 Encoder pro Achse möglich
- Positionsrückführung über Absolutwertencoder: BISS®/HIPERFACE®-Schnittstelle
- Interpolierender Betrieb über EtherCAT, CANopen oder Profinet
- 2 Highspeed-Capture-Eingänge zur schnellen Ereigniserfassung

		2-phasige Motoren	3-phasige Motoren
➔ DC-Einspeisung			
Nennanschlussspannung	V_{DC}	70	70
Nennanschlussleistung	W	960	960
Nennverlustleistung	W	45	45
➔ Daten der Leistungsendstufen			
Maximaler Phaseeffektivstrom	A_{eff}	14	14
Maximaler Phasenstrom (gesamt)	A_{DC}	20	20
Nennausgangsstrom	A_{eff}	7,1	7,1
Nennausgangsspannung	V_{DC}	60	60
Maximale Ausgangsspannung	V_{DC}	70	70
Überspannungsabschaltung	V_{DC}	90	90
Zwischenkreis Kapazität	μF	660	660
Rückspeisbare Energie	Ws	0,5	0,5

➔ Steuersignale		
24-V-Einspeisung (Stromaufnahme o. Ausgänge)	V	24 ±10%
	A	0,8
10 digitale Steuersignaleingänge	V	LOW 0-7, HIGH 12-36
	mA	10 (bei 24 V)
4 digitale Steuersignalausgänge	V	24
	A	0,5

➔ Abmessungen und Gewichte		
Abmessung B x H x T	mm	42 x 279 x 167
Gewicht	kg	ca. 1,4

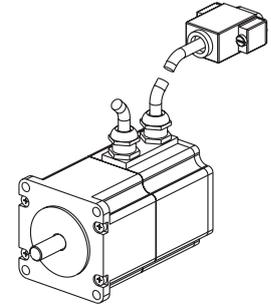
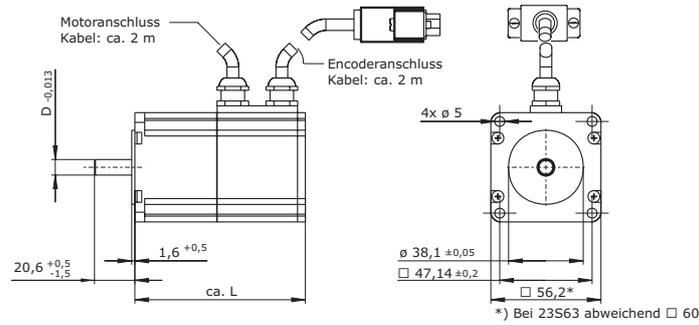
➔ Externe Absicherung		
70-V-DC-Einspeisung		max. 16 AT
24-V-DC-Einspeisung		max. 10 AF

➔ Umgebungsbedingungen	
Klimaklasse	3K3 nach EN 50178
Umgebungstemperatur im Betrieb bei Nennlast	5°C .. 40°C (Lagertemperatur: - 10 ... 70°C)
Feuchtegrad (nicht-kondensierend)	max. 95% rel. Luftfeuchtigkeit
Kühlung	Im geschlossenen Schaltschrank muss für ausreichende Umluftbewegung gesorgt werden.
Aufstellhöhe	bis 1.500 m über NN ohne Leistungseinschränkung
Einbaulage	Die technischen Daten beziehen sich auf eine vertikale Einbaulage
Schutzart	IP20, Verschmutzungsgrad 2
Angewandte Normen für CE	EMC gemäß EN61800-3, Sicherheit gemäß EN61800-5-1 bei Geräten mit Sicherheitsfunktion „STO“: Maschinenrichtlinie 2006/42/EG
Angewandte Normen für UL	UL508C

➔ Technische Änderungen vorbehalten 09/2014

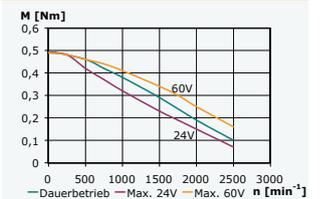

→ ECOSTEP® - Motoren / Baureihe 23S

2-2

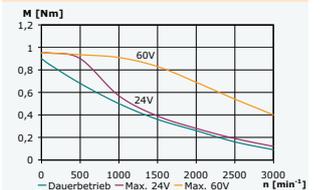


		Baureihe 23S16-0560- 805L7-52(200)	Baureihe 23S21-0560- 805L7-52(200)	Baureihe 23S31-0650- 805L7-52(200)	Baureihe 23S41-0650- 805L7-52(200)	Baureihe 23S63-0650- 805L7-52(200)
→ Bemessungswerte						
Spitzendrehmoment (ED 40%)*	[Nm]	0,50	0,95	2,00	2,80	4,0
Stillstandsmoment (ED 100%)*	[Nm]	0,50	0,90	1,70	2,20	3,8
Nennendrehmoment (ED 100%)*	[Nm]	0,30	0,52	0,70	1,38	1,8
Nennendrehzahl	[min ⁻¹]	1500	1000	1000	500	750
Spitzenstrom (pro Phase) (ED 40%)*	[A _{eff}]	5,6	5,6	8,5	8,5	6,9
Stillstandstrom (pro Phase)(ED 100%)*	[A _{eff}]	5,6	5,6	6,5	5,8	6,2
Nennstrom (pro Phase) (ED 100%)*	[A _{eff}]	4,8	3,3	2,8	3,7	3,0
Max. Zwischenkreisspannung	[V _{DC}]	60	60	90	90	90
* Montageflansch Ø 130 mm / Dicke 7,5 mm						
→ Technische Daten Motor						
Motorkonstante (bei 25 °C)	[Nm/W ^{1/2}]	0,11	0,21	0,39	0,45	0,65
Drehmomentkonstante	[Nm/A]	0,09	0,16	0,26	0,38	0,58
Spannungskonstante	[V/100min ⁻¹]	5,1	8,4	13,6	19,7	30,8
Wicklungswiderstand (bei 25 °C)	[Ω]	0,30	0,32	0,23	0,38	0,45
Wicklungsinduktivität	[mH]	0,3	0,60	0,70	1,20	2,00
Maximalstrom pro Phase	[A _{eff}]	5,6	5,6	8,5	8,5	8,0
Polpaarzahl		50	50	50	50	50
Motorträgheitsmoment	[kgm ² ·10 ⁻³]	0,014	0,026	0,046	0,069	0,150
Isolationsklasse		B, 130°C				
Umgebungstemperatur	[°C]	-20...+40				
Schutzart		IP40, optional IP65 (Wellendurchführung: IP40)				
Max. Axialbelastung	[N]	80				
Max. Radialbelastung	[N]	100				
Max. Axialbelastung Montage	[N]	150				
Masse	[kg]	0,5	0,8	1,0	1,75	1,9
Motorlänge (L)	[mm]	73	87	109	144	145
Durchmesser Welle (D)	[mm]	8	8	8	10	10

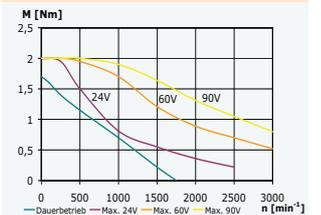
→ Kennlinie 23S16 / ECOSTEP®100



→ Kennlinie 23S21 / ECOSTEP®100



→ Kennlinie 23S31 / ECOSTEP®200

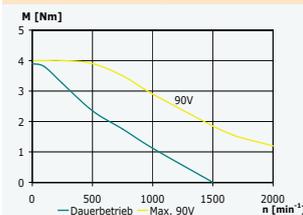


→ Technische Daten Inkrementalencoder

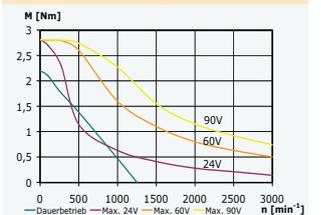
Auflösung	[Ink/U]	40.000
Betriebsspannung	[V]	5 (±10%)
Stromaufnahme	[mA]	100
Signalspezifikation		RS422

Weitere Encoder-Optionen siehe Rückseite

→ Kennlinie 23S63 / ECOVARIO®214



→ Kennlinie 23S41 / ECOSTEP®200



Technische Änderungen vorbehalten 01/2016

GTE 080 - NN2 - 040

gegeben: $i_G = 40$; $M_{Mnenn} = 1,8 \text{ Nm}$; $n_M = 750 \text{ min}^{-1}$; $i_{Zahnraeder} = 6, \bar{6}$

$$i_{ges} = i_G \cdot i_{Zahnraeder} = 40 \cdot 6, \bar{6} = 266, \bar{6} \quad (18)$$

$$i_{ges} = \frac{M_{Teleskop}}{M_{Mnenn}} \quad (19)$$

$$M_{Teleskop} = i_{ges} \cdot M_{Mnenn} = 266, \bar{6} \cdot 1,8 \text{ Nm} = 480 \text{ Nm} \quad (20)$$

$$n_{Teleskop} = \frac{n_M}{i_{ges}} = \frac{750 \text{ min}^{-1}}{266, \bar{6}} = 2,81 \text{ min}^{-1} = 0,05 \text{ s}^{-1} \quad (21)$$

$$T_{Teleskop} = 20 \text{ s} \quad (22)$$

Windgeschwindigkeiten Bad Münstereifel

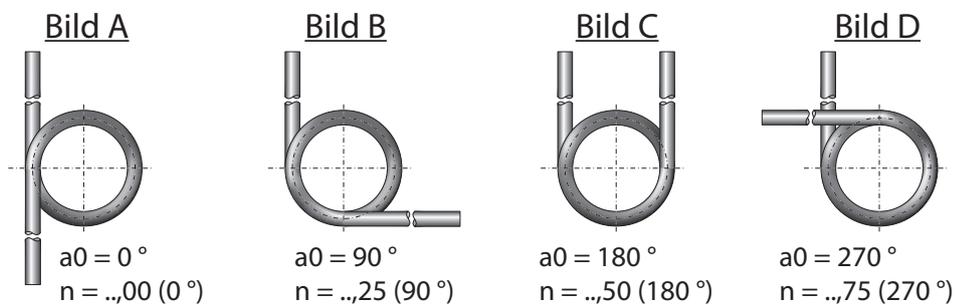
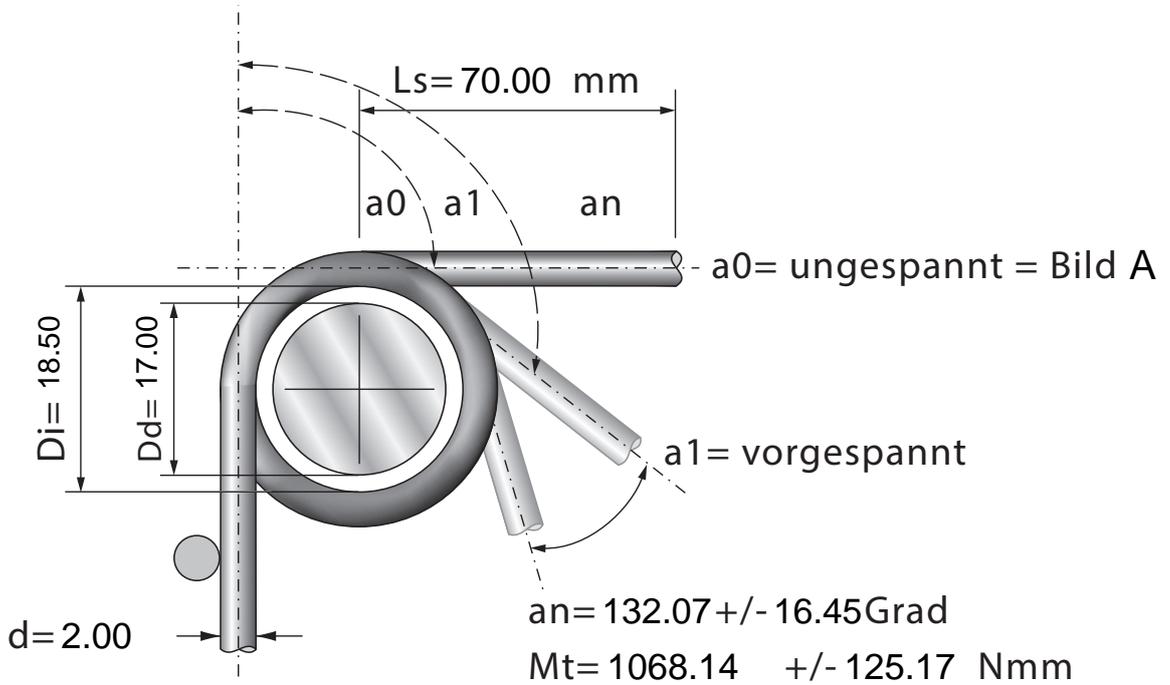
$$M = \frac{1}{2} \cdot c_w \cdot \rho \cdot v^2 \cdot A \cdot r \quad (23)$$

$$M = \frac{1}{2} \cdot 0,34 \cdot 5,5 \text{ m} \cdot 1,23 \text{ kg/m}^3 \cdot (18,3 \text{ m/s})^2 \cdot 0,671 \text{ m} = 258,43 \text{ Nm} \quad (24)$$



Artikelnummer	Bezeichnung
M20L20	Schenkelfeder

Werkstoff	1.4310
n - Windungszahl	5.00
Lk - Körperlänge (mm)	13.13
Rmr - Federmomentenrate (Nmm/°)	8.09



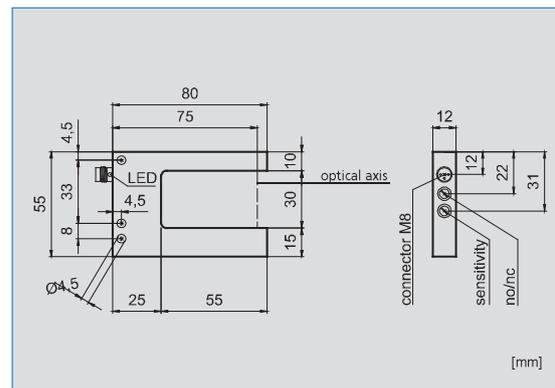
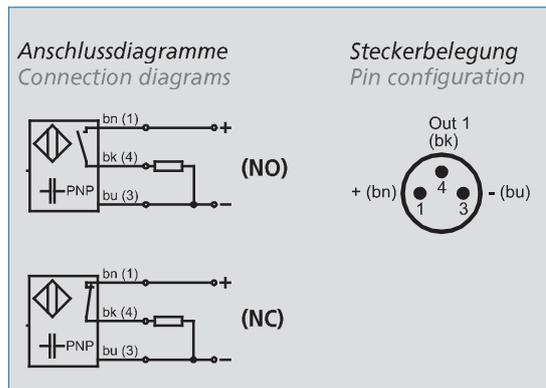
Stück	1	2	4	7	10	20	40	70	100	200	400	1000
Nettopreis/Stück	4.99	3.57	2.66	2.17	1.73	1.19	0.89	0.67	0.54	0.44	0.38	0.33

Gabellichtschranken □ Photoelectric fork sensors

Gabelweite · Fork width **30 mm**

- Laser Klasse 2 / sichtbarer Laserpunkt
- Extrem hohe Auflösung / Reproduzierbarkeit
- Empfindlichkeit einstellbar
- Schließer/Öffner umschaltbar
- Hohe Schaltfrequenz
- Schutzart IP 67

- Laser Class 2 / visible laser spot
- Extremely high resolution / reproducibility
- Sensitivity adjustable
- NO/NC selectable
- High switching frequency
- Enclosure rating IP 67



Technische Daten bei +20 °C, 24 VDC	Technical Data at +20 °C, 24 VDC	
Gabelweite	Fork width	30 mm
Auflösung	Resolution	0,03 mm
Sendelicht	Emitted light	Laser, 670 nm
Betriebsspannung	Supply voltage	10-30 VDC
Ausgangsstrom	Output current	200 mA
Kurzschlusschutz	Short-circuit protection	ja / yes
Verpolungsschutz	Polarity protection	ja / yes
Eigenstromaufnahme	Internal power consumption	< 45 mA
Isolationsspannungsfestigkeit	Insulation voltage endurance	500 V
Spannungsabfall	Voltage drop	< 2,5 V
Ansprech- / Abfallzeit	Response / Release time	0,2 ms
Schaltfrequenz	Switching frequency	2,5 kHz
Reproduzierbarkeit	Reproducibility	0,01 mm
Empfindlichkeit	Sensitivity	einstellbar / adjustable
Schalthyterese	Switching hysteresis	< 0,1 mm
Umgebungstemperatur	Ambient temperature	+2°C +45°C
Fremdlichtfestigkeit	Ambient light immunity	50 klx
Schutzart	Enclosure rating	IP 67
Gehäusematerial	Housing material	Aluminium eloxiert / Aluminum anodized
Anschlussart	Connection	Stecker M8 / plug M8
Typ	Type	
PNP, Schließer/Öffner umschaltbar	PNP, NO/NC switchable	OGSL 030 PUK-ST3
(Optionen / Options: NPN-Ausgang / NPN-output)		
passende Anschlusskabel	suitable connecting cables	M8...(3-adrig) / (3-wire)

Von:
Reinhard Keller
System-Labor
Tel. 248

Max-Planck-Institut
für Radioastronomie



Seite 1/1

Bonn, den 25.04.2016

An:

H. Hafok, E. Mann, S. Schweikert, N. Aderhold, D. Dillbohner, J. Klein, K. Grypstra, W. Schmitz, M. Esmailpour, M. Burghauve

Protokoll: Besprechung 4 m Spiegel: Winkelencoder und Antriebe

Datum, Zeit & Ort	25.4.2015	15:00h	Bibliothek Eff.
Teilnehmer	H. Hafok, E. Mann, W. Schmitz, D. Dillbohner, R. Keller (Protokoll), S. Schweikert, M. Burghauve		

Einführung:

- RK rekapituliert kurz was bisher geschah:
 - Bachelor Arbeit von Daniel Schmitt mit Linearmotor und Jenaer Servoverstärker. Ergebnis unzureichend, da keine Wiederholgenauigkeit.
 - Linearmotoren sind wg. Wetterschaden nicht mehr zu verwenden.
 - Die Fa. Jenaer Antriebstechnik zeigt unzureichende Kommunikation, weitere Hilfestellung fraglich.
- S. Schweikert trägt über Heidenhain Winkelencoder vor:
 - 2 Alternativen; RS422 oder 1Vss.
 - Kabellängen lassen auch Betrieb des Servoreglers im Heizungskeller zu.

Diskussion um weiteres Vorgehen:

- Nach Alternativen soll nochmals kurz (2-3 Tage max) gesucht werden:
 - Karte&Antriebe&Encoder aus 1 Hand (siehe Arbeit D. Schmitt).
 - Amateur-Astronomie als Quelle (-> H. Hafok).
 - Beim BZE in Euenheim gibt es ein kleines Teleskop, WS hat hierzu Kontakte und versucht einen Besuch zu organisieren.
 - Eine Lösung mit Arduino selbst programmiert und mit eigener Leistungselektronik scheint zu aufwändig.
- Zur vorhandenen Soft- und Hardware:
 - Vorhandene Software checken;
 - Spezifische Jenaer Software,
 - Treiber für LabView wurden gekauft, wo sind die (Seb)?
 - Nochmals Kontakt mit Jena aufnehmen bez. Motor mit Encoder.
- Alternativen zu Jena:
 - Bosch-Rexroth Vertreter am 4.5.2016 wg. Antrieb, Getriebe und entl. Encoder
 - Heidenhain Encoder wird bestellt, in RS422 - Ausführung.

Aufgaben

Zu erledigen	Verantwortlich	Termin
Siehe Protokoll		

Termine

Zeit	Ort	Thema
4.5.2016	Eff.	Besuch Vertreter Bosch Rexroth

R. Keller
25/04/16 17:04

intelligent & universal

3-AXIAL-ANTENNA-POSITIONER

for **Parabol-, Offset- & Yagi-Antennas**

for **GEOS & LEOS & O3Bs & MicroWave**

- Azimuth **360°** endless !
- Elevation **90°** !
- Windspeed up to **180 km/h** !
- Antennas up to **3.7m Ø** !
- Payload: **500 kg** max.
respectively: 3000 Nm max.
- Repeat accuracy: **0.1°** !
- Internal resolution: **0.01°** !
- AZ- & EL-positioning speed: **2°/sec.**
(Other speeds are to be agreed with the customer)
- Dead load: approx. 300 kg
- Temperature-range: - 25°C to +60°C.
- Auto-Focus
- Auto-Tracking
- PC-Interface
- Windsensor-Interface
- Solidstate-Relais
- Sat-Receiver-Interface
- AGC-Interface
- GPS-NMEA-Interface
- Sun-Astro-Programm
- All GEOS ,Round the world' preprogr.
- Mechanical joint plane for antennas: 60 x 60 cm.

New • **Skew-/Polarizer-Interface** New

New • **Integrated Web-Server** New

The logic of the control corresponds to the model "ProfiTracker classic", which belongs to the standard production programme of EGIS, but its capacity has been adjusted to the increased requirements.

... for
Broadcast, SNG, Science, Research and Education

... the '**PROFITRACKER XL**' of EGIS

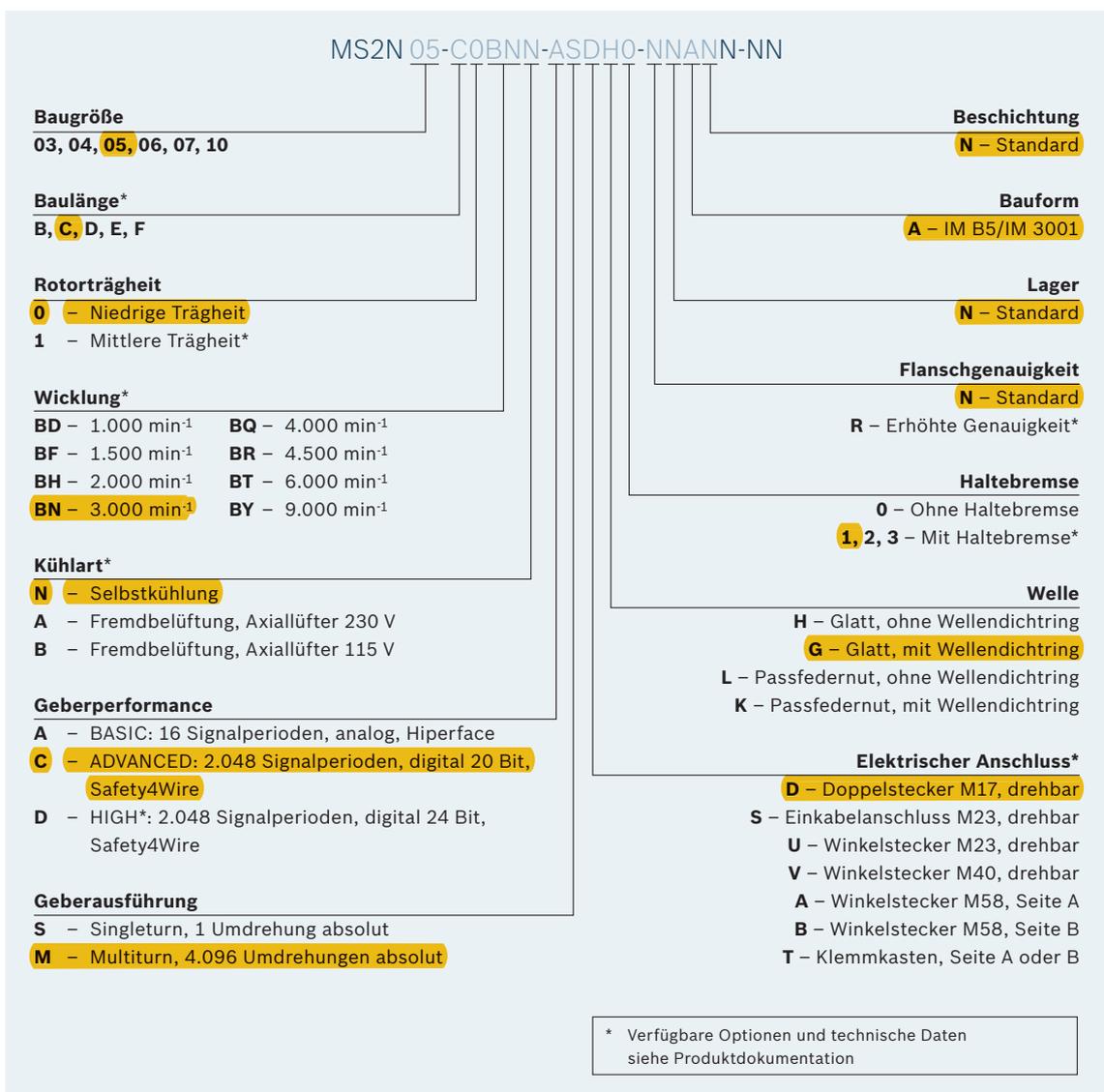


EGIS
GMBH
OFFENBACH

Technische Daten

		Drehzahl [min ⁻¹]	Drehmoment [Nm]				Strom [A]				Trägheitsmoment [kgm ²]	
Typ		n _{Max}	M _{0 60K}	M _{0 100K}	M _{0 Fan}	M _{Max}	I _{0 60K}	I _{0 100K}	I _{0 Fan}	I _{Max}	ohne Bremse	mit Bremse
03	MS2N03-B0BY	9000	0,73	0,90	-	3,75	1,30	1,60	-	7,25	0,000023	0,000030
	MS2N03-D0BY	9000	1,15	1,45	-	7,40	2,05	2,60	-	14,5	0,000037	0,000044
04	MS2N04-B0BN	6000	1,65	2,0	-	6,4	1,10	1,35	-	4,9	0,000007	0,000011
	MS2N04-B0BT	6000					2,25	2,75		9,8		
	MS2N04-C0BN	6000	2,70	3,35	-	12,6	1,85	2,30	-	9,7	0,000011	0,000016
	MS2N04-C0BT	6000					3,30	4,10		17,3		
MS2N04	MS2N04-D0BH	4000				19,0	1,65	2,05		9,7		
	MS2N04-D0BQ	6000	3,65	4,5			2,95	3,60		17,3	0,000016	0,000020
05	MS2N05-B0BN	6000				11,5	2,30	2,70		8,4		
	MS2N05-B0BT	6000	3,75	4,45			4,55	5,45		16,8	0,000017	0,000028
	MS2N05-C0BN	6000	6,10	7,15	-	22,6	3,55	4,15	-	15,1	0,000029	0,000040
	MS2N05-C0BT	6000					7,10	8,35		30,2		
	MS2N05-D0BH	4000				34,0	3,05	3,65		15,2		
MS2N05-D0BR	6000	7,90	9,35			6,05	7,20		30,3	0,000040	0,000051	
06	MS2N06-B1BN	6000	3,15	3,75		10,2	2,15	2,55		7,8	0,000048	0,000059
	MS2N06-C0BN	6000				17,3	3,75	4,50		12,8		
	MS2N06-C0BT	6000	6,0	7,1			7,50	9,00		25,5	0,000039	0,000050
	MS2N06-D1BN	6000	9,1	11,2		42,5	4,90	6,00		25,5	0,00140	0,00154
	MS2N06-D0BN	6000					6,10	7,55		25,4		
	MS2N06-D0BR	6000	9,7	11,9		34,8	7,85	9,75		32,7	0,00065	0,00079
	MS2N06-E0BH	4000				53,4	5,4	6,80		25,4		
	MS2N06-E0BR	6000	13,0	16,3			10,9	13,7		50,8	0,00089	0,00103
07	MS2N07-B1BN	6000	6,7	7,8	-	22,8	3,8	4,5	-	14,8	0,00197	0,00223
	MS2N07-C1BR	6000				46,0	6,5	8,2	9,7	29,5		
	MS2N07-C1BR	6000	11,4	14,2	16,9		9,4	11,8	14,0	42,7	0,00305	0,00331
	MS2N07-C0BN	6000				38,8	7,2	9,2	11,0	25,9		
	MS2N07-C0BQ	6000	12,8	16,0	19,0		10,1	12,9	15,4	36,4	0,00120	0,00146
	MS2N07-D1BH	4000				92,5	7,8	9,9	12,6	42,7		
	MS2N07-D1BN	6000	18,9	23,8	30,3		9,9	12,5	15,9	54,1	0,00529	0,00570
	MS2N07-D0BH	4000				79,7	8,6	11,1	14,0	36,4		
	MS2N07-D0BN	6000	22,0	28,2	35,2		12,2	15,8	19,9	51,8	0,00210	0,00251
	MS2N07-D0BR	6000					17,1	22,3	28,0	72,7		
	MS2N07-E1BH	4000				140	8,8	11,4	14,9	54,1		
	MS2N07-E1BN	6000	25,2	32,5	43,1		13,8	17,9	23,5	85,4	0,00752	0,00793
	MS2N07-E0BH	4000					10,8	14,2	18,5	51,8		
	MS2N07-E0BN	6000	29,2	38,2	49,5	120	15,1	20,0	26,2	72,7	0,00300	0,00341
MS2N07-E0BQ	6000					19,1	25,3	33,3	92,3			
10	MS2N10-C1BH	4000				86,5	11,1	12,6	16,2	40,9		
	MS2N10-C1BN	6000	27,2	30,5	38,7		16,4	18,6	24,1	60,8	0,00920	0,01067
	MS2N10-C0BH	4000				76,8	12,6	14,4	18,8	38,5		
	MS2N10-C0BN	6000	30,2	34,0	43,2		16,8	19,3	25,1	51,3	0,00480	0,00627
	MS2N10-D1BF	3000				174	13,8	16,5	22,6	60,7		
	MS2N10-D1BN	6000	45,7	54,0	73,0		27,6	32,9	45,3	122	0,01710	0,01857
	MS2N10-D0BH	4000				155	19,1	23,1	32,4	70		
	MS2N10-D0BN	6000	51,0	60,5	82,4		28,1	34,0	48,0	103	0,00810	0,00957
	MS2N10-E1BF	3000				266	16,8	20,1	29,2	81		
	MS2N10-E1BN	6000	63,2	75,0	107,5		33,5	40,1	58,4	162	0,02500	0,02770
	MS2N10-E0BH	4000				234	25,4	31,0	46,1	103		
	MS2N10-E0BN	6000	69,0	83,0	119,0		34,7	42,4	62,9	140	0,01140	0,01410
	MS2N10-F1BD	2000				360	15,5	18,6	27,7	81		
	MS2N10-F1BH	4000	78,5	93,5	138,0		30,9	37,1	55,4	162	0,03290	0,03560
MS2N10-F0BD	2000				313	16,0	19,6	29,3	70			
MS2N10-F0BH	4000	85,0	103,0	148,0		31,9	39,2	58,6	140	0,01470	0,01740	

Typenschlüssel



Getriebebezeichnung PLN 115-64	
Getriebestufen	2
Übersetzung ^[a]	64
Abtriebsdrehmoment T_{2N} ^{[c][e]}	150.0 Nm
Max. Abtriebsmoment ^{[c][e][g]}	240.0 Nm
Trägheitsmoment ^[i]	1.959 kgcm ²
Max. Radialbelastung 20.000 h	6000 N
Max. Axialbelastung 20.000 h	8000 N
Max. Radialbelastung 30.000 h	5400 N
Max. Axialbelastung 30.000 h	7000 N
Gewicht	9.5 kg
Max. Antriebsdrehzahl	8500 min ⁻¹
Max. mittlere Antriebsdrehzahl bei 50% T_{2N} und S1	3500 min ⁻¹
Max. mittlere Antriebsdrehzahl bei 100% T_{2N} und S1	3500 min ⁻¹
Wirkungsgrad bei Volllast	95%
Verdrehflankenspiel	< 5 arcmin
Verdrehsteifigkeit	26 Nm/arcmin
Betriebstemperatur min. ^[d]	-25° C
Betriebstemperatur max. ^[d]	90° C
Schutzart	IP 65
Schmierung	Lebensdauer-Schmierung
Lebensdauer	30.000 h
Laufgeräusch ^[h]	71 db(A)
Einbaulage	beliebig
Motorflansch- genauigkeit	DIN 42955-R
Not-Aus Moment ^[f]	2 - faches T_{2N}
Baugröße	
Gesamtlänge L1	241 mm
Gehäuselänge L2	101.5 mm

Kräfte- und Rotationsgeschwindigkeitsberechnung mit ausgewählten Bauteilengegeben:

- Getriebeübersetzung

$$i_G = 64$$

- Zahnkranz Teleskop

$$z_T = 80$$

- Zahnrad

$$z_z = 20$$

- Nenndrehmoment Motor

$$M_{0\ 60K} = 6,1\ \text{Nm}$$

- Nenndrehzahl

$$n_M = 3000\ \text{min}^{-1}$$

gesucht:

- Drehmoment am Teleskop

$$M_T$$

- Drehzahl und Umdrehungsdauer

$$n_T\ \text{und}\ T$$

$$i_T = \frac{z_T}{z_z} = \frac{80}{20} = 4 \quad (25)$$

$$i_{\text{ges}} = i_T \cdot i_G = 256 \quad (26)$$

$$M_T = i_{\text{ges}} \cdot M_{060K} = 1561,6\ \text{Nm} \quad (27)$$

$$n_T = \frac{n_M}{i_{\text{ges}}} = 11,72\ \text{min}^{-1} = 0,12\ \text{s}^{-1} \quad (28)$$

$$T = 5,12\ \text{s} \quad (29)$$

	Inkremental		
	ROD 420	ROD 430	ROD 480
Schnittstelle	□ HTL	□ HTL	~ 1 V _{SS} ¹⁾
Strichzahlen*	50 100 150 200 250 360 500 512 720		–
	1000 1024 1250 1500 1800 2000 2048 2500 3600 4096 5000		
Referenzmarke	eine		
Grenzfrequenz –3 dB	–		≥ 180 kHz
Ausgangsfrequenz	≤ 300 kHz		–
Flankenabstand a	≥ 0,39 μs		–
Systemgenauigkeit	1/20 der Teilungsperiode		
Elektrischer Anschluss*	<ul style="list-style-type: none"> • Flanschdose M23, radial und axial • Kabel 1 m/5 m, mit oder ohne Kupplung M23 		
Spannungsversorgung	DC 5 V ± 0,5 V	DC 10 V bis 30 V	DC 5 V ± 0,5 V
Stromaufnahme ohne Last	≤ 120 mA	≤ 150 mA	≤ 120 mA
Welle	Vollwelle D = 10 mm		
Mech. zul. Drehzahl n	≤ 16 000 min ⁻¹		
Anlaufdrehmoment	≤ 0,01 Nm (bei 20 °C)		
Trägheitsmoment Rotor	≤ 2,3 · 10 ⁻⁶ kgm ²		
Belastbarkeit der Welle ²⁾	<i>axial</i> : ≤ 40 N; <i>radial</i> : ≤ 60 N am Wellenende		
Vibration 55 bis 2000 Hz	≤ 300 m/s ² (EN 60 068-2-6)		
Schock 6 ms	≤ 2000 m/s ² (EN 60 068-2-27)		
Max. Arbeitstemperatur ³⁾	100 °C (80 °C bei ROD 480 mit 4096 bzw. 5000 Strichen)		
Min. Arbeitstemperatur	<i>Flanschdose oder Kabel fest verlegt</i> : –40 °C <i>Kabel bewegt</i> : –10 °C		
Schutzart EN 60529	IP67 am Gehäuse; IP64 am Welleneingang (IP66 auf Anfrage)		
Masse	ca. 0,3 kg		
Gültig für ID	376840-xx	376834-xx	376880-xx

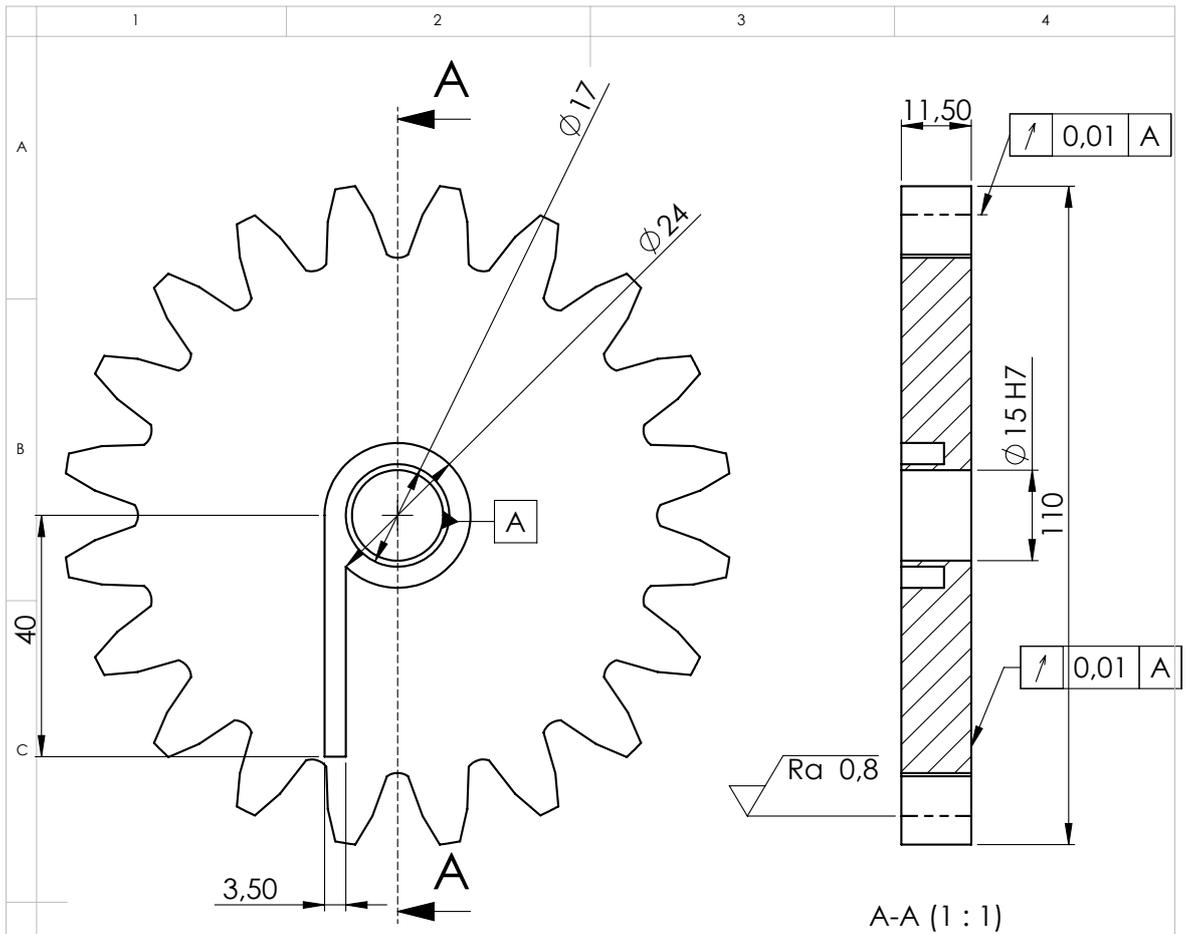
fett: Diese Ausführung ist als Vorzugstyp schnell lieferbar.

* bei Bestellung bitte auswählen

¹⁾ eingeschränkte Toleranzen: Signalgröße 0,8 bis 1,2 V_{SS}

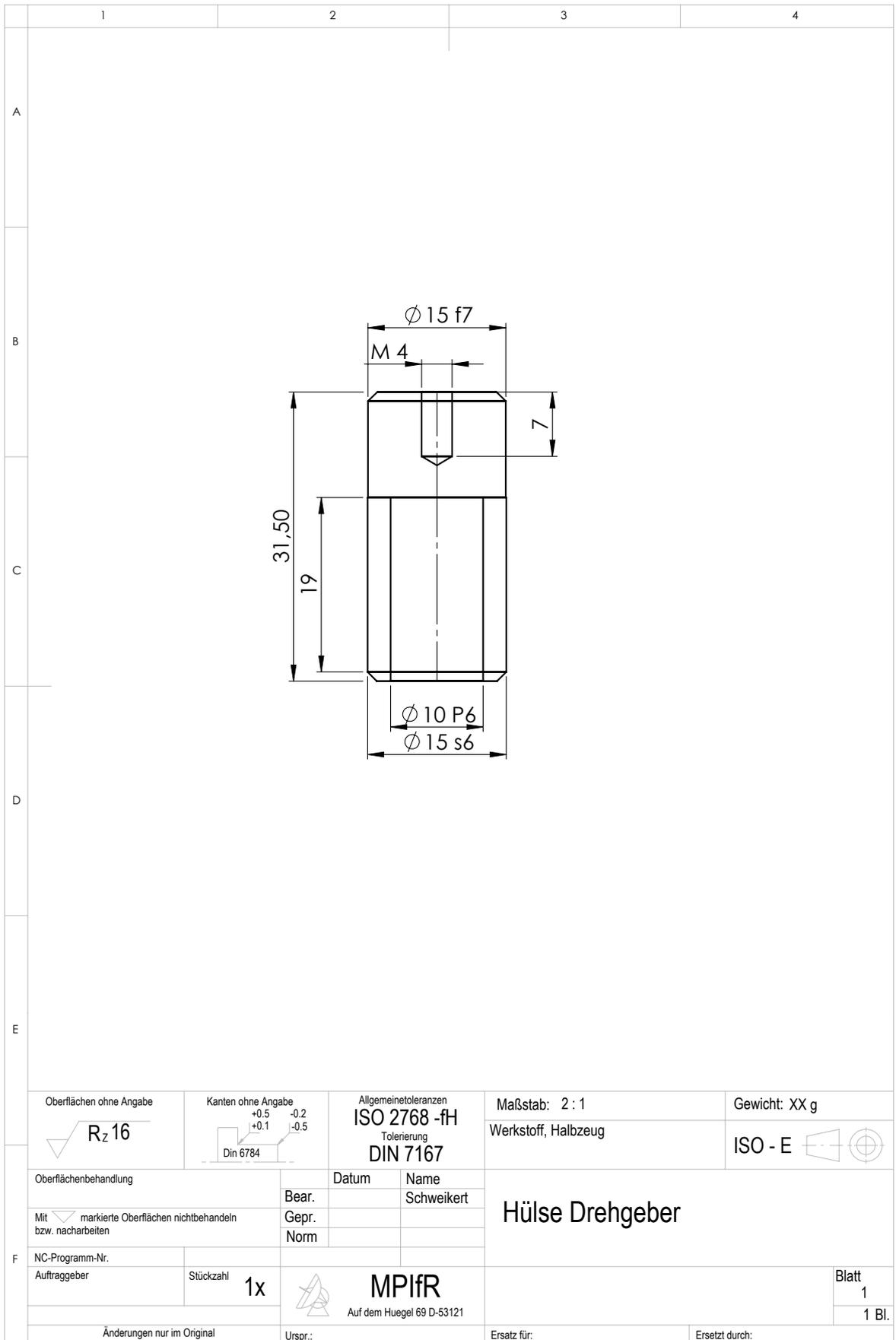
²⁾ siehe auch *Mechanische Geräteausführungen und Anbau*

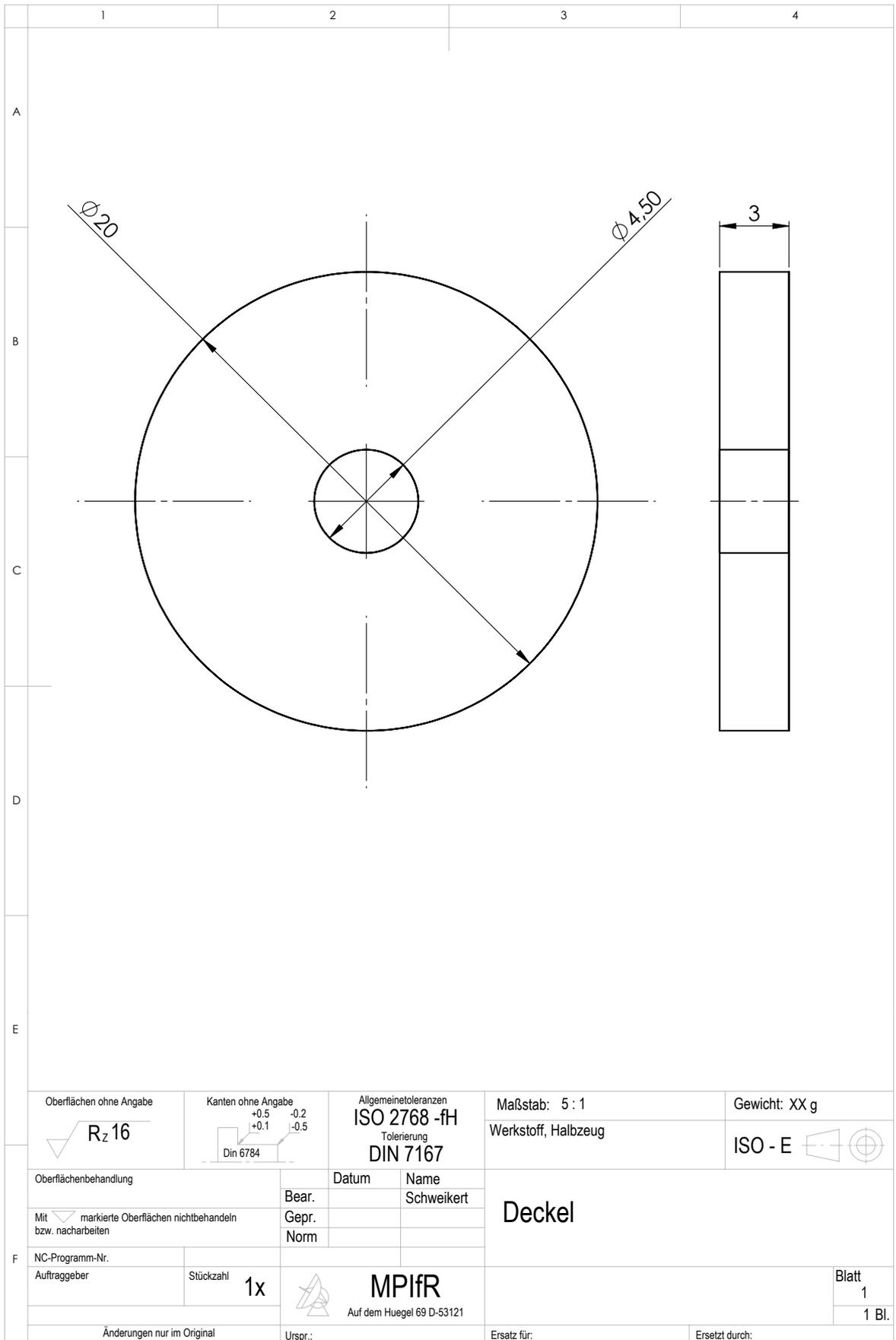
³⁾ Zusammenhang zwischen Arbeitstemperatur und Drehzahl bzw. Versorgungsspannung siehe *Allgemeine mechanische Hinweise*

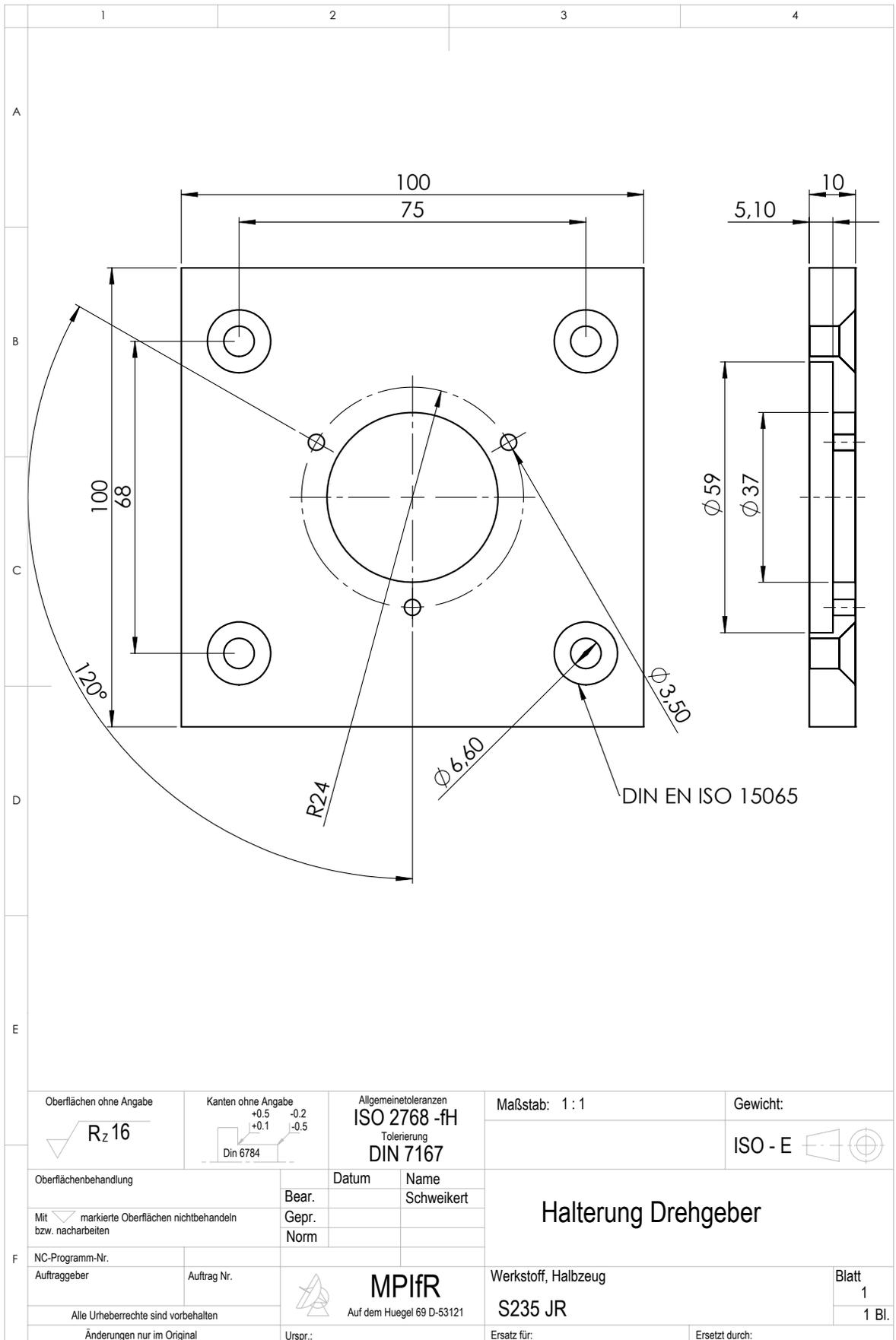


	Stirnrad	außenverzahnt
D	Zähnezahl	20
	Bezugsprofil	DIN 867
	Schrägungswinkel	0°
	Flankenaustrichtung	-
	Verzahnungsqualität	8 e 26 DIN 3967
E	Gegenrad	Sachnummer 83099270
		Zähnezahl z 80

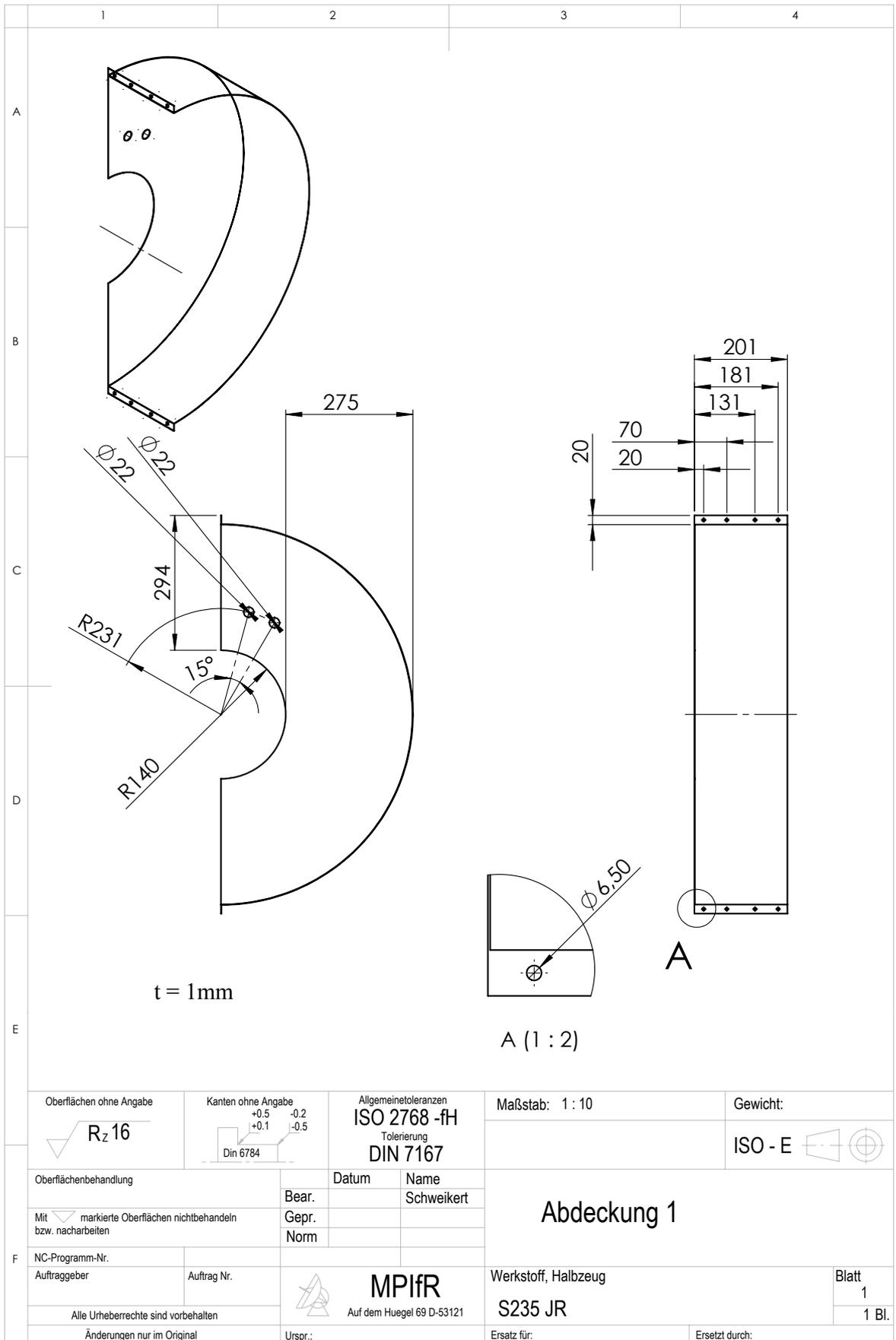
Oberflächen ohne Angabe 		Kanten ohne Angabe 		Allgemeintoleranzen ISO 2768 -fH Tolerierung DIN 7167		Maßstab: 1 : 1		Gewicht:	
Oberflächenbehandlung		Datum		Name		Werkstoff, Halbzeug Aluminium		ISO - E	
Mit <input type="checkbox"/> markierte Oberflächen nichtbehandeln bzw. nacharbeiten		Bear.		Schweikert		Zahnrad			
NC-Programm-Nr.		Gepr.		Norm					
Auftraggeber		Stückzahl 2x				Auf dem Huegel 69 D-53121		Blatt 1 1 Bl.	
Änderungen nur im Original		Urspr.:		Ersatz für:		Ersetzt durch:			

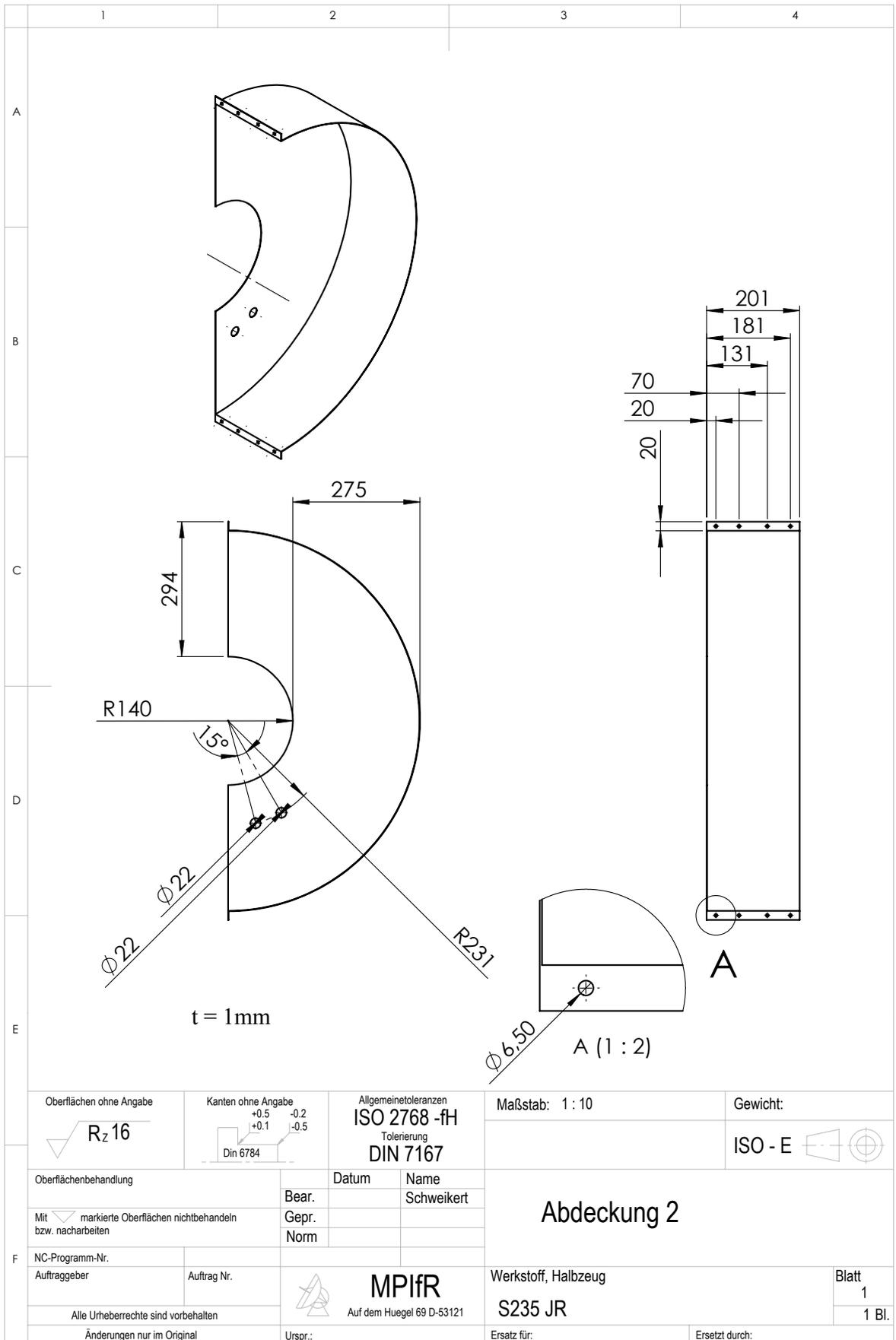


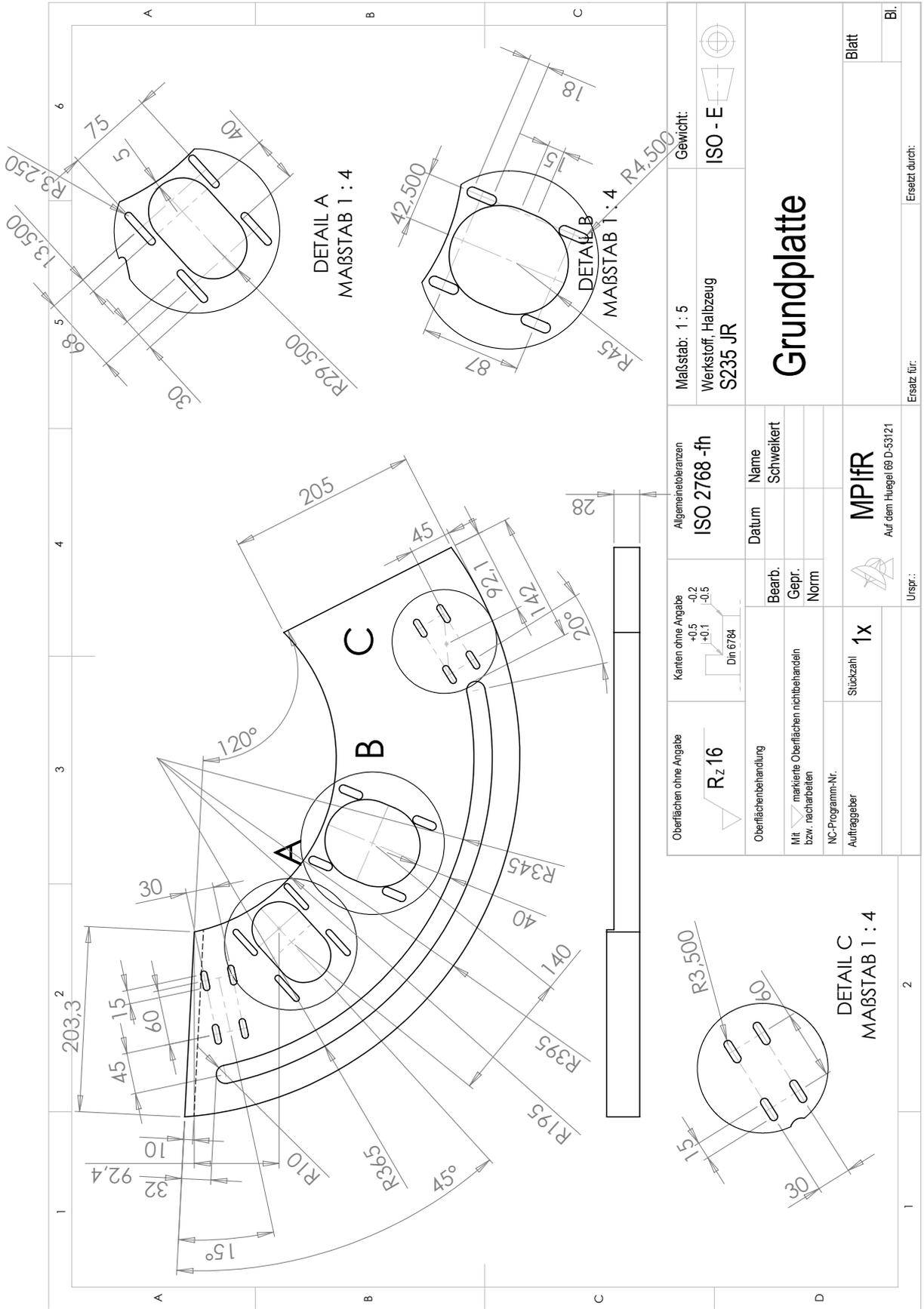




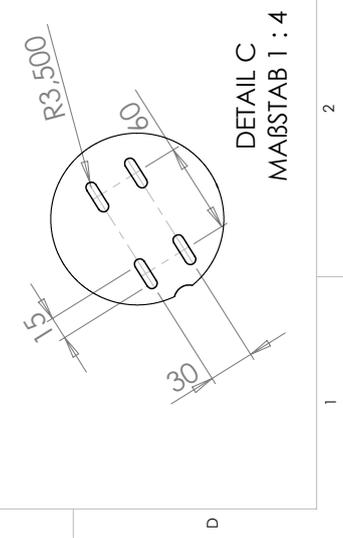
Oberflächen ohne Angabe 		Kanten ohne Angabe 		Allgemeintoleranzen ISO 2768 -fH Tolerierung DIN 7167		Maßstab: 1 : 1		Gewicht:	
Oberflächenbehandlung Mit markierte Oberflächen nichtbehandeln bzw. nacharbeiten		Datum Name		Bear. Gepr. Norm		Schweikert		ISO - E	
NC-Programm-Nr.		Auftrag Nr.		Werkstoff, Halbzeug S235 JR		Blatt 1		1 Bl.	
Änderungen nur im Original		Urspr.:		Ersatz für:		Ersetzt durch:		Alle Urheberrechte sind vorbehalten MPIfR Auf dem Huegel 69 D-53121	

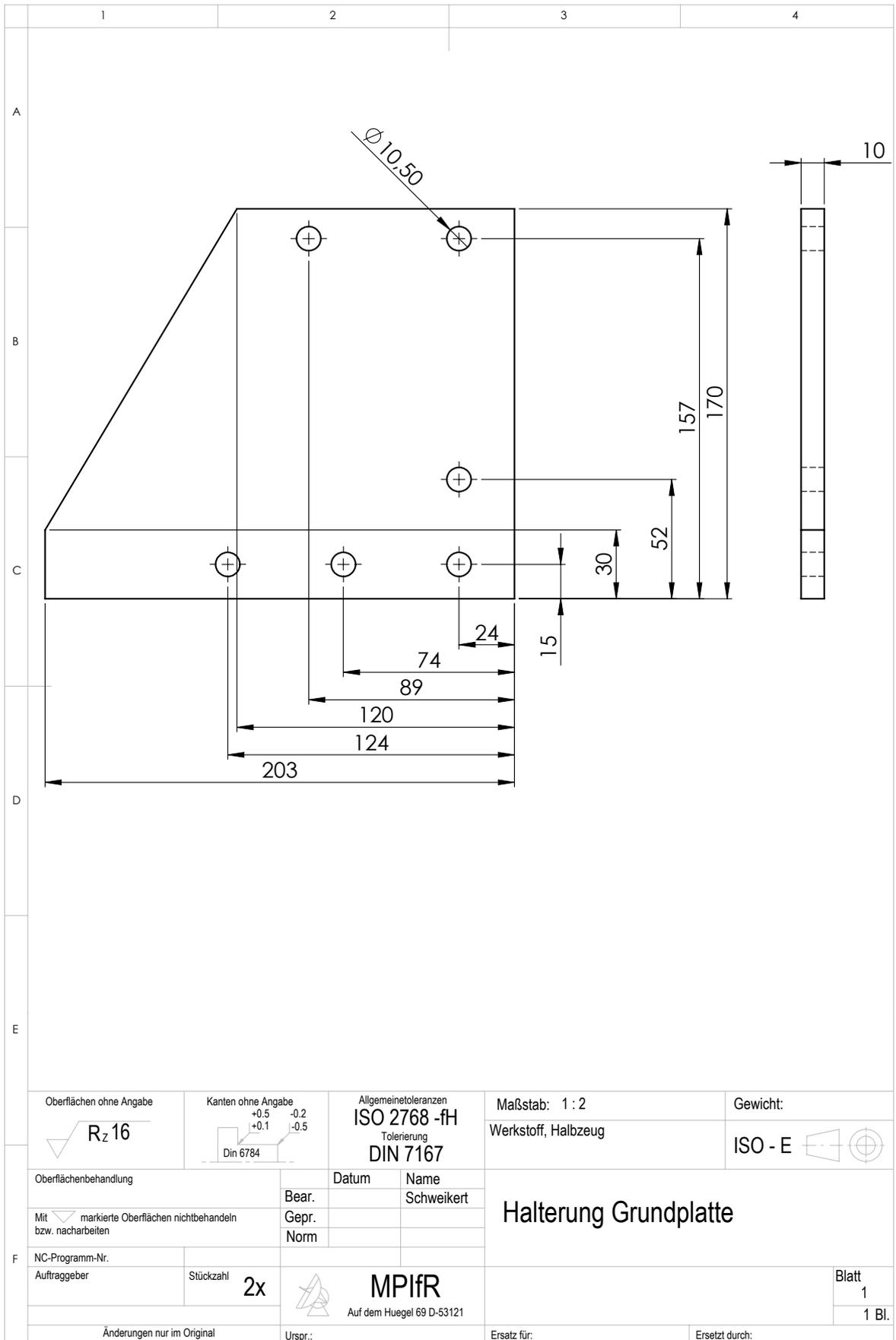




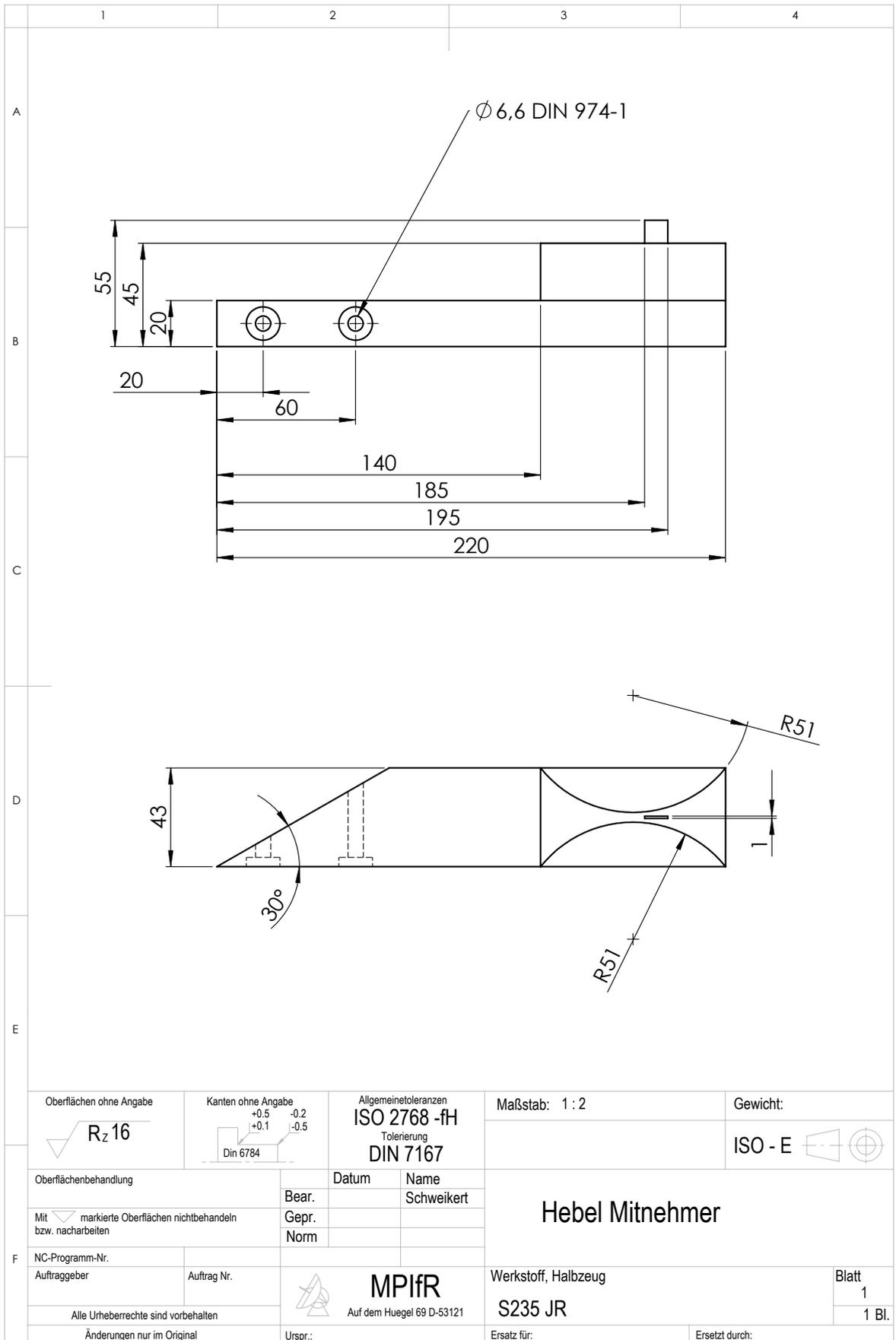


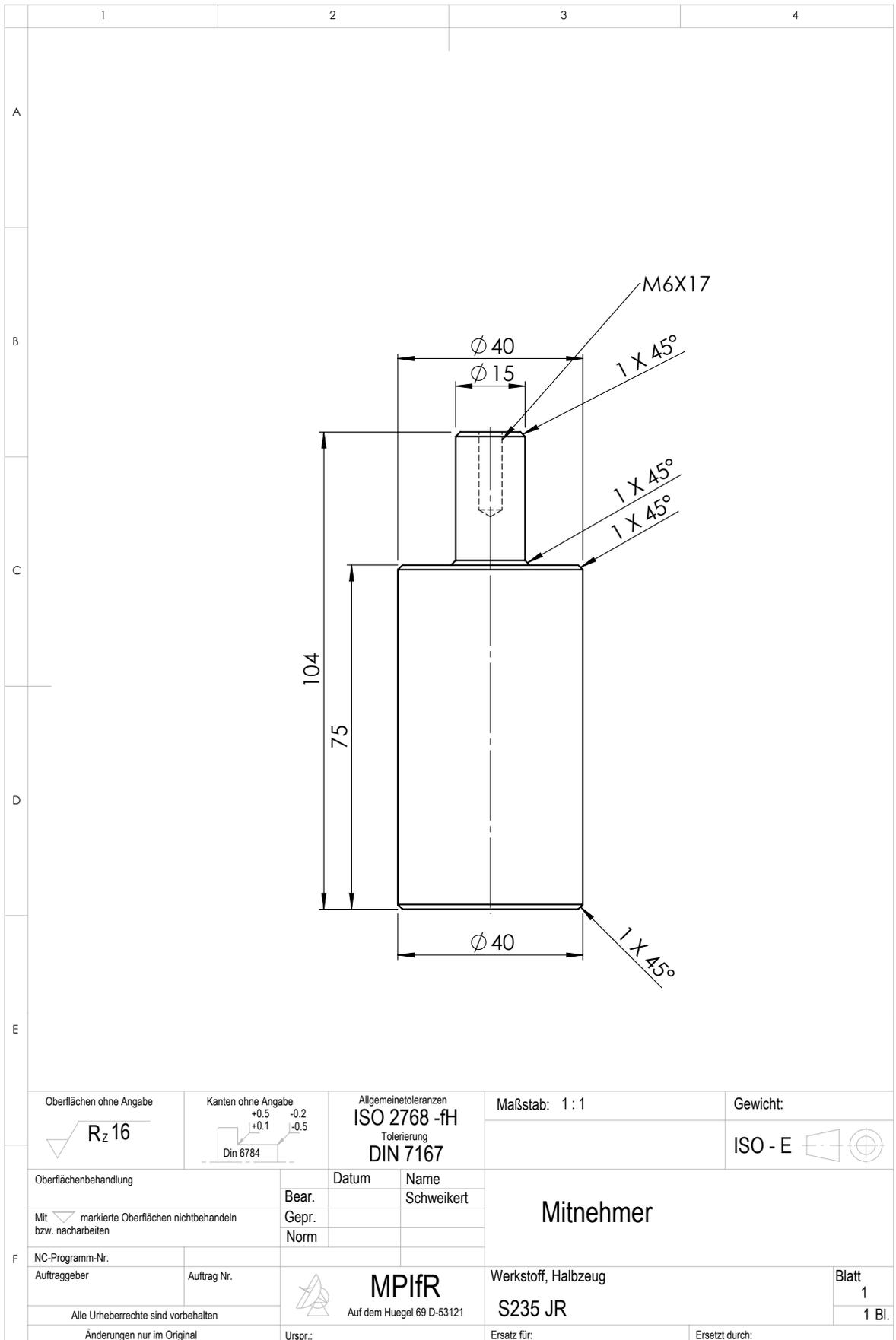
Oberflächen ohne Angabe Rz 16		Kanten ohne Angabe +0.5 -0.2 -0.5 DIN 6784		Allgemeinbezeichnungen ISO 2768 -fh		Maßstab: 1 : 5 Werkstoff, Halbzeug S235 JR		Gewicht: ISO - E	
Oberflächenbehandlung		Bearb. Gepr. Norm		Datum		Name Schweißkett		<h1>Grundplatte</h1>	
Mit bzw. nacharbeiten		markierte Oberflächen nicht behandeln							
NC-Programm-Nr. Auftraggeber		Stückzahl 1x							
		Uspr.: MPIFR Auf dem Hugel 69 D-53121		Ersatz für:		Ersetzt durch:		Blatt Bl.	

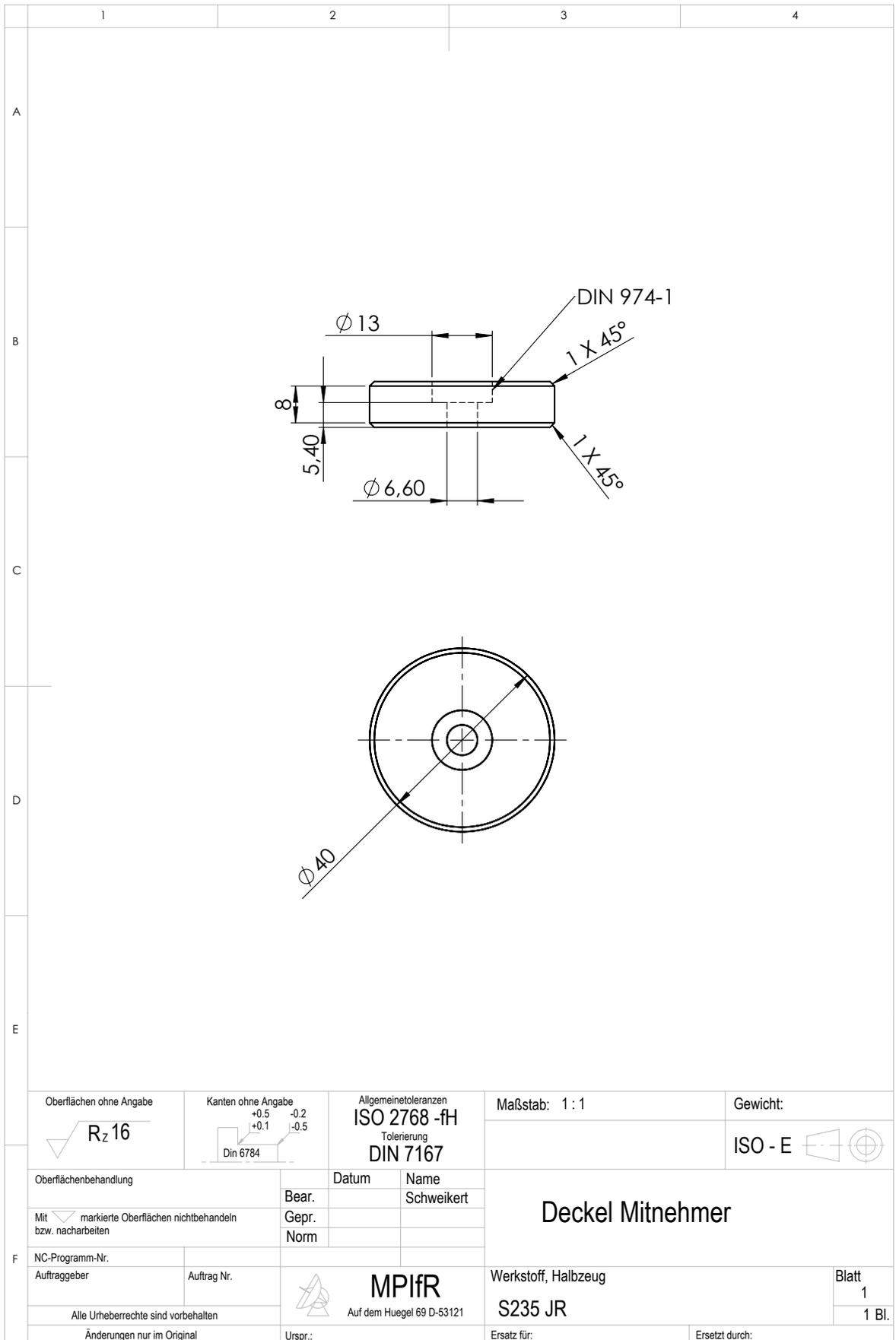


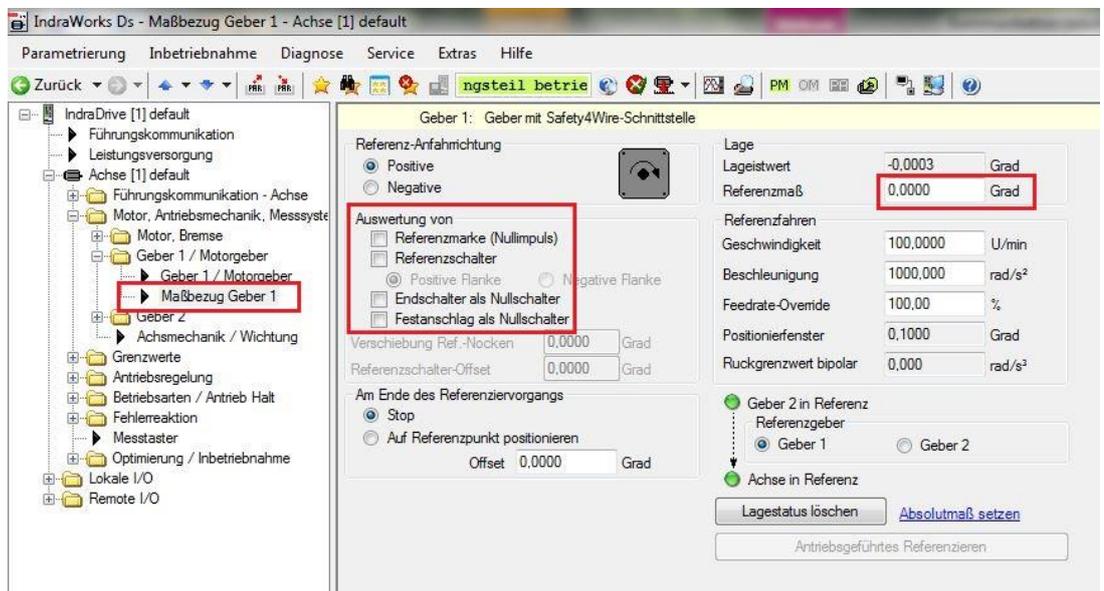
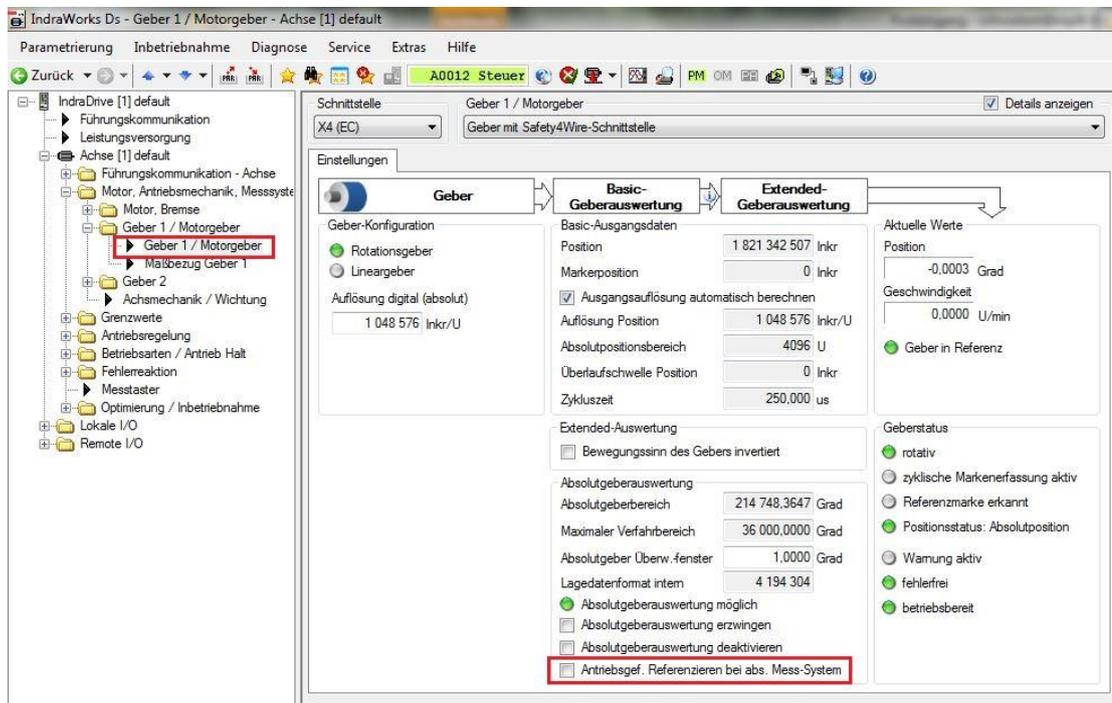


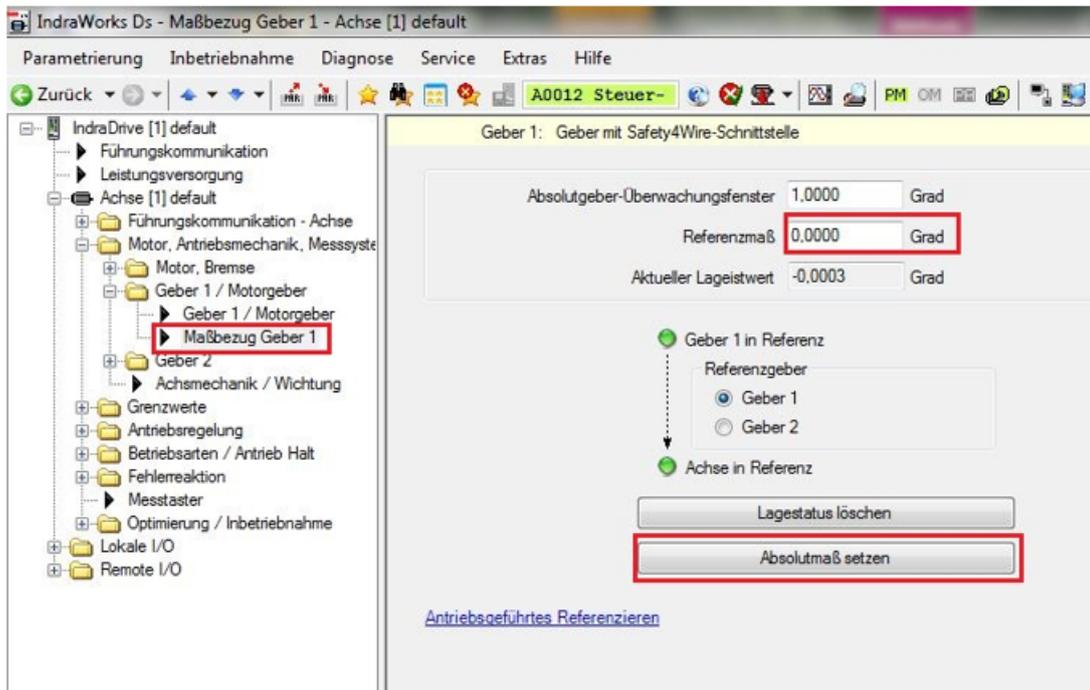
Oberflächen ohne Angabe 		Kanten ohne Angabe 		Allgemeintoleranzen ISO 2768 -fH Tolerierung DIN 7167		Maßstab: 1 : 2 Werkstoff, Halbzeug		Gewicht: ISO - E			
Oberflächenbehandlung Mit markierte Oberflächen nichtbehandeln bzw. nacharbeiten			Datum Name		Halterung Grundplatte						
			Bear. Gepr. Norm							Schweikert	
NC-Programm-Nr. Auftraggeber			Stückzahl 2x							MPIfR Auf dem Huegel 69 D-53121	
Änderungen nur im Original				Urspr.:		Ersatz für:		Ersetzt durch:			











```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <pthread.h>
4 #include <ealApiLib.h>
5 #include <ealMotionLib.h>
6 #include <stdio.h>
7 #include <sys/socket.h>
8 #include <netinet/in.h>
9 #include <string.h>
10 #include <sys/types.h>
11 #include <sys/socket.h>
12 #include <netinet/in.h>
13 #include <arpa/inet.h>
14 #include <netdb.h>
15 #include <unistd.h>
16 #include <stdlib.h>
17 #include <string.h>
18 #include <json/json.h>
19 #include <time.h>

22 //Globale Variablen, koennen in thread 1 und 2 verwendet werden
23 double pos;
24 int vel;
25 double acc = 0.8; // Beschleunigung Motor
26 double dec = 0.1; // Verzoeigerung Motor
27 bool start;

30 int multicast(double posout, char achszustand[1000])
31 {
32     int one = 1;
33     int socket_descriptor;
34     struct sockaddr_in address;
35     char posstring[256];
36     time_t t;

```

```

37 | struct tm *ts; // Zeituebertragung
38 | t = time(NULL);
39 | ts = localtime(&t); //ts = aktuelle Zeit

41 | sprintf(posstring, "%f", posout); // von int auf string

43 | // Erstellt JSON Objekt
44 | json_object * jobj = json_object_new_object();

47 | // Erstellen JSON Strings Position , Timestamp , Reglerzustand
48 | json_object *jstring1 = json_object_new_string(posstring);
49 | json_object *jstring2 = json_object_new_string(asctime(ts));
50 | json_object *jstring3 = json_object_new_string(achszustand);

53 | // JSON Format erstellen
54 | json_object_object_add(jobj, "posaz", jstring1);
55 | json_object_object_add(jobj, "time", jstring2);
56 | json_object_object_add(jobj, "state", jstring3);

60 | // Socket erstellen
61 | socket_descriptor = socket(AF_INET, SOCK_DGRAM, 0);

63 | // address-Variable mit 0 fuellen
64 | memset(&address, 0, sizeof(address));

66 | // Socket konfigurieren mit Multicast-Adresse und Port
67 | address.sin_family = AF_INET;
68 | address.sin_addr.s_addr = inet_addr("224.1.1.1");
69 | address.sin_port = htons(11421);

71 | // IP_MULTICAST_TTL setzen (TTL=1 —> Pakete werden nicht geroutet)
72 | setsockopt(socket_descriptor, IPPROTO_IP, IP_MULTICAST_TTL, &one, sizeof(one));

74 | int laenge = 0;
75 | laenge = strlen(json_object_to_json_string(jobj)); // laenge von string
76 | sendto(socket_descriptor, json_object_to_json_string(jobj), laenge, 0, (struct
    sockaddr*) &address, sizeof(address));
77 | // printf("Sent: %s\n", json_object_to_json_string(jobj));
78 | sleep(1);

80 | return 0;
81 | }

86 | void *thread_regler(void *ch) {

88 | while (1) {
89 |     if (start == 1) {

91 | // printf("\n\t-> Thread regler mit ID:%ld gestartet\n", pthread_self()); //
    Anzeigen der Thread ID

93 | // Bibliothekspfad EAL.dll einbinden
94 | ealSetLibraryPath("/home/schweikert/4mSpiegel/eal_sdk_1_1_15_0/EAL_SDK_01VRS/01
    VRS/eal4C/bin");

96 | // Schaltet vor Programmaussfuehrung zunaechst in PM, Parametriermodus
97 | EALHandle handle = ealCreateDeviceHandle(false);

99 | //IP Adresse an Regeleinheit anpassen
100 | int result = ealApiConnectDefault(&handle, "192.168.0.1");

102 | if (result >= 0) {

```

```

104     result = ealMotionMovementPower(&handle, 0, false);
106     if (result >= 0) {
108         // Parametriermodus
109         EalAxisCondition condition = 8;
110         result = ealMotionSetCondition(&handle, 0, condition);
112     if (result >= 0) {
113         puts("Erfolgreich in PM geschaltet");
115         //Wichtig: Initialisierung benoetigt DCP Modus
116         result = ealApiInitializeDefault(&handle, 0);
118     if (result >= 0) {
119         puts("Motor Initialisierung abgeschlossen");
120         condition = 0;
121         result = ealMotionSetCondition(&handle, 0, condition);
123     if (result >= 0) {
124         puts("in Parametriermodus geschaltet");
125         result = ealMotionMovementPower(&handle, 0, true);
127     if (result >= 0) {
128         puts("Motor bereit");
129         result = ealMotionMovementHome(&handle, 0); // Referenzfahrt
131         //so bleibt der Regler immer in der Regelung und es kann direkt ein Wert
           uebergeben werden
132     while(start && result >= 0){
134         // Positionsgeregelt
135         result = ealMotionMovementMoveAbsolute(&handle,0, pos, vel, acc, dec, 0);
136         result = ealMotionMovementWait(&handle, 0,100000);
138     double x;
139     char diagnosisTxt[1000];
140     unsigned long numElements;
142     result = ealMotionGetDiagnosisText (&handle,0,diagnosisTxt,1000, &numElements);
144     ealMotionGetActualPosition(&handle,0,&x);
145     multicast(x, diagnosisTxt); // geht nur mit zwei Netzwerkkarten
146     }}}}
147     ealApiDisconnect (&handle);
148     }}}}
149     return 0;
150 }
153 void *thread_consumer(void *ch)
154 {
156     int i;
157     int n=0;
158     int k=0;
159     int j=0;
160     char cstart[128];
161     char cpos[128];
162     char cvel[128];
163     int udpSocket, nBytes;
164     char buffer[1024];
165     struct sockaddr_in serverAddr, clientAddr;
166     struct sockaddr_storage serverStorage;
167     socklen_t addr_size, client_addr_size;
169     //printf("\n\t-> Thread consumer mit ID:%ld gestartet\n", pthread_self()); //
           Anzeigen der Thread ID

```

```

173 // Erstellt UDP Socket
174 udpSocket = socket(PF_INET, SOCK_DGRAM, 0);

176 // Einstellen der Adresskonfiguration
177 serverAddr.sin_family = AF_INET;
178 serverAddr.sin_port = htons(7891);
179 serverAddr.sin_addr.s_addr = inet_addr("127.0.0.1");
180 memset(serverAddr.sin_zero, '\0', sizeof serverAddr.sin_zero);

182 // bind socket
183 bind(udpSocket, (struct sockaddr *) &serverAddr, sizeof(serverAddr));

185 addr_size = sizeof serverStorage;

187 while(1){
188 // Empfangen der UDP Daten
189 nBytes = recvfrom(udpSocket,buffer,1024,0,(struct sockaddr *)&serverStorage, &
    addr_size);

191 for(i = 0; buffer[i] !='\0'; i++)
192 {
193     cstart[i] = '\0'; // problem: wenn Schleife kuerzer als vorher ist, werden die
    // alten gespeicherten char Werte noch an den neuen String angehangen
194     cpos[i] = '\0'; // mit der Loesung werden einfach die Werte aus dem String
    // geloescht
195     cvel[i] = '\0';
196 }

199 for(i = 0; buffer[i] !='\0'; i++)
200 {
201     if (buffer[i] == ';')
202     {
203         j++;
204     }
205     if ((j == 0)&&(buffer[i] != ';'))
206     {
207         cstart[i] = buffer [i];
208     }

210     if ((j == 1)&&(buffer[i] != ';'))
211     {
212         cvel[n] = buffer [i];
213         n++;
214     }

216     if ((j == 2)&&(buffer[i] != ';')) // weil sonst ; mit in String geschrieben wird
217     {
218         cpos[k] = buffer [i];
219         k++;
220     }
221 }

223 j=0;
224 k=0;
225 n=0;

227 vel = atoi(cvel);
228 pos = atof(cpos); // char in double umwandeln;
229 start = atoi(cstart);

231 sleep(1);

233 // Senden der Daten an Client
234 sendto(udpSocket,buffer,nBytes,0,(struct sockaddr *)&serverStorage,addr_size);
235 printf("Start: %d\n", start);

```

```
236 printf("eingabe Position ist: %lf\n",pos);
237 printf("eingabe Geschwindigkeit ist: %d\n",vel);

239 for(i = 0; buffer[i] !='\0'; i++)
240 {
241     buffer[i] = '\0'; // string buffer muss auch geleert werden
242 }
243 }

245 return 0;
246 }

250 int main() {
251     char ch1, ch2;
252     pthread_t p1, p2;
253     pthread_create(&p1, NULL, thread_regler, &ch1);
254     pthread_create(&p2, NULL, thread_consumer, &ch2);
255     pthread_join(p1, NULL);
256     pthread_join(p2, NULL);

258     return 0;
259 }
```

Listing 7: C-Programm


```
67 | time.sleep(60.)
68 | except:
69 | eprint()
70 | # will try to connect again in a minute
71 | print("unknown error")
72 | print("will try again in a minute")
73 | time.sleep(60.)
74 | finally:
75 | print("closing socket")
76 | self.sock.close()

81 | class Base(QtGui.QMainWindow):
82 | def __init__(self):
83 | QtGui.QMainWindow.__init__(self)

85 | # Workerthread erstellen und starten
86 | self.myworkerthread = QtCore.QThread()
87 | self.myworker = MulticastWorker()
88 | self.myworker.moveToThread(self.myworkerthread)

92 | def setFullDegree(self, data_in):
93 | posaz = float(data_in) # weil Winkel double ist
94 | return "%s" % posaz
```

Listing 9: Python Multicast

```

1  import socket
2  import time
3  import threading
4  import numpy

6  from string import atoi
7  from socket import *
8  from thread import *

10 # Daten wieder empfangen
11 """
12 def recv(udpCliSock):
13 while 1:
14 data, addr = udpCliSock.recvfrom(BUFSIZE)
15 if not data: break
16 print data
17 """
18 udpCliSock = socket(AF_INET, SOCK_DGRAM)

20 HOST = "127.0.0.1"
21 PORT = 7891
22 BUFSIZE = 8192
23 ADDR = (HOST, PORT)

25 testWert=0.0

28 class MyMuxerForm(QtGui.QMainWindow):
29 def __init__(self, parent=None):

31 QtGui.QMainWindow.__init__(self, parent)
32 self.ui = guidesigner.Ui_MainWindow() #Ui_MainWindow() Funktion aus GUI testlese
33 self.ui.setupUi(self)
34 #Slots einrichten
35 QtCore.QObject.connect(self.ui.sent,QtCore.SIGNAL("clicked()"),self.setpushOk)
36 QtCore.QObject.connect(self.ui.Eingabepos,QtCore.SIGNAL('activated(QString)'),self.
    setpushOk)

39 def setpushOk(self):
40 startWert = int(self.ui.Freigabe.isChecked()) # isChecked -> auslesen radiobutton
41 start = str(startWert) #von float auf string da udp string braucht

43 posWert = float(self.ui.Eingabepos.text()) #text -> auslesen von editline
44 pos = str(posWert)

46 velWert = float(self.ui.Eingabevel.text())
47 vel = str(velWert)

49 data = start + ";" + vel + ";" + pos
50 print data
51 udpCliSock.sendto(data, ADDR)

53 if __name__ == "__main__":

55 app = QtGui.QApplication(sys.argv)
56 myapp= MyMuxerForm()
57 myapp.show()

59 sys.exit(app.exec_())

```

Listing 8: Python UDP