

**STUDIENARBEIT**  
des Studiengangs Informationstechnik  
der Dualen Hochschule Baden-Württemberg Mannheim

---

**THEMA**

**Entwicklung eines Systems zur Verfolgung von Objekten mit Hilfe einer Kamera**

---

**Julius Klodt**

19. April 2022

---

Bearbeitungszeitraum	01.10.2021 - 19.04.2022
Matrikelnummer, Kurs	8431855, TINF19IT1
Ausbildungsfirma	Deutsches Zentrum für Luft- und Raumfahrt e.V.
Betreuer	Jürgen Schultheis
Unterschrift des Betreuers	

---

## **Erklärung**

Ich erkläre hiermit ehrenwörtlich:

1. dass ich meine Studienarbeit mit dem Thema *Entwicklung eines Systems zur Verfolgung von Objekten mit Hilfe einer Kamera* ohne fremde Hilfe angefertigt habe;
2. dass ich die Übernahme wörtlicher Zitate aus der Literatur sowie die Verwendung der Gedanken anderer Autoren an den entsprechenden Stellen innerhalb der Arbeit gekennzeichnet habe;
3. dass ich meine Studienarbeit bei keiner anderen Prüfung vorgelegt habe;
4. dass die eingereichte elektronische Fassung exakt mit der eingereichten schriftlichen Fassung übereinstimmt.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Mannheim, 3. April 2022

---

Julius Klodt

## **Kurzfassung**

### **Abstract**

# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>V</b>
<b>1 Einleitung</b>	<b>1</b>
<b>2 Aufgabenstellung</b>	<b>2</b>
<b>3 Methoden und Verfahren</b>	<b>3</b>
3.1 C . . . . .	3
3.2 Python . . . . .	3
3.2.1 OpenCV . . . . .	3
3.3 FreeCAD . . . . .	4
3.4 Ultimaker Cura . . . . .	4
3.5 3D-Drucker . . . . .	4
<b>4 Durchführung</b>	<b>6</b>
4.1 Hardware . . . . .	6
4.1.1 Präzise Anforderungen an die Hardware stellen . . . . .	6
4.1.2 Vorhandene Projekte sichten und Umsetzungsideen sammeln . . . . .	7
4.1.3 Konzepte erstellen und Hardware konzipieren . . . . .	9
4.1.4 Hardware konstruieren . . . . .	15
4.1.5 Benötigte Hardware einkaufen . . . . .	19
4.1.6 Hardware zusammenbauen . . . . .	19
4.2 Software . . . . .	19
4.2.1 Anforderungen an die Software stellen . . . . .	19
4.2.2 UML-Diagramm konzipieren . . . . .	19
4.2.3 Hardwareansteuerung implementieren . . . . .	19
4.2.4 Gesichtserkennung implementieren . . . . .	19
4.2.5 Algorithmus zur Verarbeitung der Position des Gesichts zu Motorbewegungen entwickeln und implementieren . . . . .	19
4.2.6 Hardwareansteuerung, Gesichtserkennung und Algorithmus zur Verarbeitung zusammenführen . . . . .	19
4.3 Testen . . . . .	19
4.3.1 Hardware auf Funktionstüchtigkeit und Grenzen testen . . . . .	19
4.3.2 Software auf Funktionstüchtigkeit und Grenzen testen . . . . .	19
4.4 Erweiterungen . . . . .	19

<b>5 Ergebnis</b>	<b>20</b>
<b>6 Reflektion</b>	<b>21</b>
<b>7 Ausblick</b>	<b>22</b>
<b>8 Danksagung</b>	<b>23</b>
<b>Literatur</b>	<b>I</b>
<b>Anhang</b>	<b>II</b>

# Abkürzungsverzeichnis

DHBW	Duale Hochschule Baden-Württemberg
IDE	Integrated Development Environment
CAD	Computer Aided Design
PLA	Polylactic acid

# Abbildungsverzeichnis

3.1	“Cubic“ - Struktur	4
3.2	“Grid“ - Struktur	4
4.1	Pan-Bewegung[9]	6
4.2	Tilt-Bewegung[10]	6
4.3	Pan-Tilt Einheit der Firma ServoCity[7]	8
4.4	3D-gedruckter Pan-Tilt-Kopf mit Hilfe von einem Differential-Getriebe[1]	8
4.5	3D-gedruckte Pan-Tilt Einheit mit Spiegelreflexkamera[4]	9
4.6	3D-Modell der Pan-Tilt Einheit von Isaac879	10
4.7	Visualisierung des grundsätzlichen Aufbaus eines innen läufigen Zahnrads.	11
4.8	Statische Basis (1.) mit Mulde an Außenseite (2.). Hauptplattform (3.) mit Mulde an Innenseite (4.). Loch für Kugeln in Hauptplattform für Befüllung der Kugeln (5.).	11
4.9	Komponenten des Kugellagers. Hauptplattform (1.), Abstandhalter-Ring (2.), Kugeln (3.), statische Basis (4.)	12
4.10	Kugeln mit Abstandhalter-Ring (1.) in Hauptplattform. Statische Basis darüber.	12
4.11	Slider Aufbau: Bewegliche Plattform (3.) fährt mit Hilfe von starrem Zahnräumen (2.) über zwei Aluminiumrohre (1.)	13
4.12	Slider Aufbau: Motor (2.) mit Zahnräder (1.) ist an der Plattform befestigt. Der Zahnrämen liegt so, dass das Zahnräder sich bei Drehung entlang des Zahnrämens bewegt.	14
4.13	Slider Aufbau Querschnitt: Plattform mit Befestigungen (4.). Stahlkugeln (3.) innerhalb der Befestigungen, um über Aluminiumrohre (2.) zu rollen. Zahnrämen (1.) mittig. Motor und Zahnräder wie in Abbildung 4.12 zur Übersichtlichkeit nicht zu sehen.	14
4.14	Box für sämtliche Elektronik auf der Hauptplattform und neben den Füßen für die Tilt-Bewegung.	15
4.15	Seitliche, auf dem Kopf stehende Ansicht eines Modells der Sony A7 RII. Länge der Kamera-Objektiv Kombination und Höhe der Sensormitte angegeben.	16

4.16 Ansicht auf das Kameramodell von unten. Breite der Auflagefläche, Breite inklusive Zylinderförmige Befestigungen (Position in Abbildung 4.15 ersichtlich) und Abstand von Außenseite zu Mitte des Gewindelochs zur Befestigung der Kamera . . . . .	16
4.17 Kamerahalterung: Auflagefläche für Kamera = 131 mm, komplette Breite für Kamera = 141 mm . . . . .	16
4.18 Kamerahalterung Ansicht von oben: Löcher für Befestigung der Objektivstütze (1., r = 1,7 mm), Loch für Befestigung der Kamera (2., r = 3,2 mm), Optionale Löcher (3., r = 3,2 mm) . . . . .	17

# **Tabellenverzeichnis**

# 1 Einleitung

In Zeiten der Covid-19-Pandemie wurden Videokonferenzen immer populärer. In Schulen, bei der Arbeit, beim Studium und selbst beim Sport werden Videokonferenzen mittlerweile genutzt. Dabei kommt es nicht selten vor, dass sich die Position einer Person verändert und nur noch schlecht im Bild zu sehen ist. Bei Verwendung mehrerer Bildschirme entsteht zusätzlich das Problem, dass je nach dem auf welchen Bildschirm geguckt wird, nur das Profil und nicht die Frontalansicht des Gesichts zu sehen ist.

Im Rahmen einer Studienarbeit der **Duale Hochschule Baden-Württemberg (DHBW)** Mannheim ist daraus die Idee entstanden, ein System zu entwickeln, welches die genutzte Kamera automatisch so bewegt, dass die gefilmte Person immer mittig und frontal im Bild zu sehen ist.

Dabei soll sowohl Hardware als auch Software erstellt werden. Um die benötigte Präzision der Hardware zu gewährleisten kam die Idee, das System 3D zu drucken. Alle Komponenten sollen dafür modelliert und gedruckt werden. Die Software soll die Motor und Peripherie Ansteuerung beinhalten. Des Weiteren eine Gesichtserkennung und ein Algorithmus zur Weiterverarbeitung der Gesichtsposition zu Motorbewegungen.

# 2 Aufgabenstellung

Die Studienarbeit wurde in folgende Aufgaben unterteilt:

## Aufgabe 1: Hardware

- 1.1. Präzise Anforderungen an die Hardware stellen
- 1.2. Vorhandene Projekte sichten und Umsetzungsideen sammeln
- 1.3. Konzepte erstellen und Hardware konzipieren
- 1.4. Hardware konstruieren
- 1.5. Benötigte Hardware einkaufen
- 1.6. Konstruierte Teile mit 3D-Drucker drucken und nachbearbeiten
- 1.7. Hardware zusammenbauen

## Aufgabe 2: Software

- 2.1. Anforderungen an die Software stellen
- 2.2. UML-Diagramm konzipieren
- 2.3. Hardwareansteuerung implementieren
- 2.4. Gesichtserkennung implementieren
- 2.5. Algorithmus zur Verarbeitung der Position des Gesichts zu Motorbewegungen
- 2.6. Hardwareansteuerung, Gesichtserkennung und Algorithmus zur Verarbeitung zusammenführen

## Aufgabe 3: Testen

- 3.1. Hardware auf Funktionstüchtigkeit und Grenzen testen
- 3.2. Software auf Funktionstüchtigkeit und Grenzen testen

## Aufgabe 4: Erweiterungen

# 3 Methoden und Verfahren

In diesem Kapitel werden die zugrunde liegenden Theoretischen Kenntnisse, die Methodik und Verfahrensgrundlage erörtert. Zu dem wird erklärt, warum diese genutzt wurden.

## 3.1 C

Die Hardwareansteuerung erfolgt mit der Programmiersprache C, welche nativ auf dem verwendeten *Arduino nano* unterstützt wird. C wurde ursprünglich von Dennis Ritchie in den frühen 1970er Jahren an den Bell Laboratories entwickelt und basiert auf der von Ritchie und Ken Thompson entwickelten Programmiersprache B [8].

Die Möglichkeit auf Hardwarekomponenten direkt zuzugreifen, macht C sehr hardwarenah [3]. Dies bringt auch den Vorteil mit sich, dass C sehr schnell ist. Um mit zu programmieren und den fertigen Code auf den Arduino nano zu übertragen wurde die Integrated Development Environment (IDE) Arduino-Code mit der Version 1.8.13 verwendet.

## 3.2 Python

Für die Gesichtserkennung und die Weiterverarbeitung der Gesichtspositionsdaten wird die Programmiersprache Python verwendet. Python ist eine objektorientierte, höhere Programmiersprache [6] und wird häufig für Anwendungen im Bereich des Maschinellen Lernens verwendet.

Im Rahmen dieser Arbeit wird die Python Version 3.9.7 verwendet.

### 3.2.1 OpenCV

OpenCV ist eine Open-Source Bibliothek welche für digitale Bildverarbeitung genutzt wird und in den Programmiersprachen Java, C, C++ und Python verwendet werden kann [OpenCV]. Die OpenCV Bibliothek bietet unter anderem Algorithmen zur Gesichtserkennung an und wird daher für diese Arbeit mit der Version 4.5.5 genutzt.

### 3.3 FreeCAD

FreeCAD ist ein Open-Source Computer Aided Design ([CAD](#))-Programm. Es wird für die Konstruktion und Modellierung von Objekten genutzt [2]. In Rahmen dieser Arbeit wird FreeCAD mit der Versionsnummer 0.19 genutzt, um die benötigten Teile für das Verfolgungssystem als 3D-Modelle zu konstruieren.

### 3.4 Ultimaker Cura

“Ultimaker Cura ist eine Open-Source Slicer-Software zur Umwandlung von 3D Modellen in G-Code (Druckerbefehle für 3D-Drucker)“[11].

Für diese Arbeit wurde Ultimaker Cura mit der Version 4.12.1 genutzt. Das Programm bietet vielseitige Möglichkeiten an, wie ein 3D-Modell gedruckt werden soll. Zum Beispiel kann manuell eingestellt werden, wie viel Prozent des Innenraums eines 3D-Modells mit Material gefüllt werden soll. Diese Einstellung wird automatisch mit einer einstellbaren Inneren Struktur umgesetzt. Im folgenden sind zwei unterschiedliche innere Strukturen zu sehen, welche automatisch von Ultimaker Cura produziert worden sind. Beide Abbildungen zeigen die 44. Druckschicht eines Zahnrads mit 20% Füllung des Innenraums.

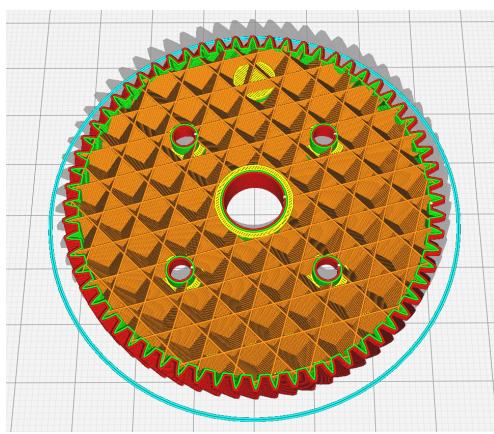


Abbildung 3.1: “Cubic“ - Struktur

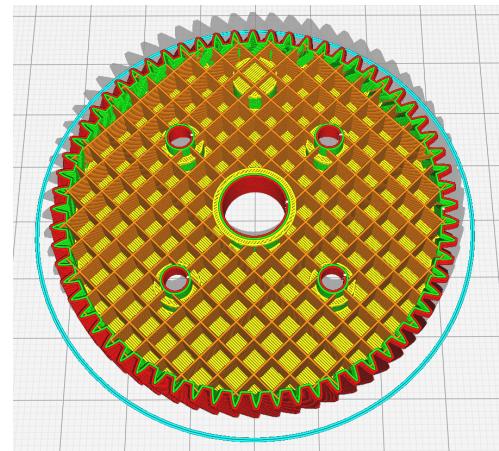


Abbildung 3.2: “Grid“ - Struktur

### 3.5 3D-Drucker

3D-Drucken ist ein additives Verfahren, in welchem ein dreidimensional beweglicher Druckerkopf, Material Schicht für Schicht übereinander lagert.

Präzises, schnelles, unkompliziertes und kostengünstiges 3D-Drucken ist für den Privatanwender erst in dem letzten Jahrzehnt möglich geworden.

### *3 Methoden und Verfahren*

---

Im Rahmen dieser Arbeit wurden drei verschiedene 3D-Drucker genutzt, um selbst konstruierte Objekte mit dem Kunststoff **Polylactic acid (PLA)** zu drucken.

1. Ultimaker 2+
2. Renkforce RF2000v2
3. Creality Ender 3 V2

# 4 Durchführung

Dieses Kapitel erörtert die Durchführung der Aufgabenstellung. Die Strukturierung gleicht dem Aufbau der Aufgabenstellung.

- Hardware
- Software
- Testen
- Erweiterungen

## 4.1 Hardware

In diesem Abschnitt wird beschrieben, wie die Hardware entwickelt wurde.

### 4.1.1 Präzise Anforderungen an die Hardware stellen

Damit das System eine Person im Bild einer Kamera verfolgen kann, muss gewährleistet sein, dass das System die Kamera drehen und neigen kann. Diese Drehbewegungen werden als Pan und Tilt bezeichnet.



Abbildung 4.1: Pan-Bewegung[9]



Abbildung 4.2: Tilt-Bewegung[10]

Die Pan-Bewegung soll dabei  $360^\circ$  durchführbar sein damit das System möglichst vielseitig genutzt werden kann. Die Tilt-Bewegung soll aus der Horizontalen mindestens  $45^\circ$  nach unten und oben ausführbar sein. Sowohl die Pan, als auch die Tilt Bewegung soll eine Mindestverstellgenauigkeit von einem halben Grad aufweisen. Die Geschwindigkeit der

Pan-Bewegung soll mindestens  $180^\circ$  pro 3 Sekunden betragen. Die Geschwindigkeit der Tilt-Bewegung soll mindestens  $90^\circ$  pro 2 Sekunden betragen.

Um zu gewährleisten, dass immer die Frontalansicht einer Person zu sehen ist, muss sich die Kamera mindestens in einer Achse im Raum bewegen können. Diese Arbeit beschränkt sich dabei auf eine gerade Achse. Die Bewegung entlang der Achse soll mindestens 50 cm betragen. Die Verstellgenauigkeit soll mindestens 1 mm betragen. Die Geschwindigkeit soll mindestens 10 cm pro Sekunde betragen.

Das System soll die beschriebenen Anforderungen mit der vorhanden Kamera-Objektiv Kombination Sony A7 RII und Tamron 28-75 mm F2.8 Di III RXD erfüllen. Diese Kombination wiegt zusammen ca. 1,1 kg.

Hier eine tabellarische Übersicht der Anforderungen:

Anforderungsbeschreibung	Wert
Pan-Bewegung, Drehbereich	$\geq 360^\circ$
Pan-Bewegung, Verstellgenauigkeit	$\leq 0.5^\circ / \text{Schritt}$
Pan-Bewegung, Geschwindigkeit	$\geq 180^\circ / 3 \text{ Sekunden}$
Tilt-Bewegung, Drehbereich aus der horizontalen	$\geq \pm 45^\circ$
Tilt-Bewegung, Verstellgenauigkeit	$\leq 0.5^\circ / \text{Schritt}$
Tilt-Bewegung, Geschwindigkeit	$\geq 90^\circ / 2 \text{ Sekunden}$
Bewegung im Raum, entlang einer räumlichen Achse	$\geq 50\text{cm}$
Bewegung im Raum, Verstellgenauigkeit	$\leq 1\text{mm} / \text{Schritt}$
Bewegung im Raum, Geschwindigkeit	$\leq 10\text{cm} / \text{Sekunde}$
Gewicht der zu bewegenden Kamera	$\geq 1.1\text{kg}$

### 4.1.2 Vorhandene Projekte sichten und Umsetzungsideen sammeln

Um Ideen für das Projekt zu sammeln, wurde recherchiert was für Projekte schon vorhanden sind und welche hilfreich bei der Entwicklung der Hardware sein könnten. Bei der Recherche sind verschiedene Konzepte öfter aufgetaucht.

## 1. Einseitige Befestigung der Kamera entlang der auszuführenden Tilt-Bewegung-Achse

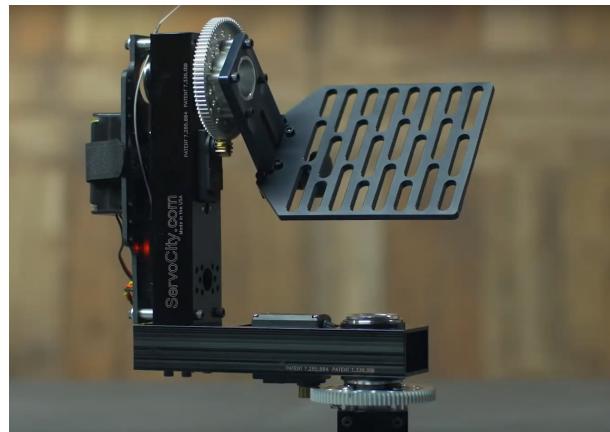


Abbildung 4.3: Pan-Tilt Einheit der Firma ServoCity[7]

In Abbildung 4.3 ist zu sehen, dass die drehende Kamera Befestigung nur an der sich drehenden Seite befestigt ist. Dadurch wird ein starker einseitiger Druck aufgebaut, wodurch es notwendig ist, mit sehr stabilen Materialien wie Metallen zu arbeiten. Gleiches ist bei der Pan-Achse ersichtlich.

## 2. 3D-gedruckter Pan-Tilt-Kopf mit Differenzial Getriebe.

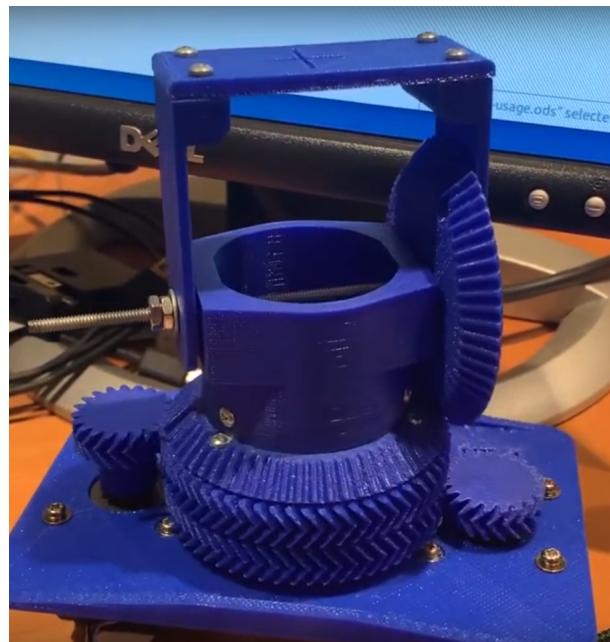


Abbildung 4.4: 3D-gedruckter Pan-Tilt-Kopf mit Hilfe von einem Differential-Getriebe[1]

Der in Abbildung 4.4 zusehende Pan-Tilt-Kopf nutzt ein Differentialgetriebe, um sowohl die Pan, als auch die Tilt Bewegung mit Motoren auszuführen, welche beide in der gleichen Achse montiert werden können.

### 3. Beidseitige Befestigung der Kamera entlang der auszuführenden Tilt-Bewegung-Achse



Abbildung 4.5: 3D-gedruckte Pan-Tilt Einheit mit Spiegelreflexkamera[4]

Abbildung 4.5 zeigt ein funktionierendes Pan-Tilt System mit einer montierten Spiegelreflexkamera. Zu sehen ist, dass im Gegensatz zu dem System auf Abbildung 4.3, die Kamera beidseitig entlang der Tilt-Achse befestigt ist. Dadurch verteilt sich das Gewicht der Kamera gleichmäßiger.

Nach Betrachtung aller Konzepte, bietet sich für die Umsetzung dieser Studienarbeit der grundsätzliche Aufbau des dritten gezeigten Systems in Abbildung 4.5 am besten an. In dem Video, aus welchem das Bild entnommen wurde, wird gezeigt, dass das System mit einer Spiegelreflexkamera tadellos funktioniert und mit Hilfe von einem billigen 3D-Drucker für Privatanwender gebaut werden kann. Somit müssen keine hochpräzisen Metallarbeiten erledigt werden, welche voraussichtlich deutlich teurer wären. Das Differential-System hat den Konstruktions-Nachteil, dass bei Abschaltung der Motoren, die Kamera aufgrund des Gewichts, unkontrolliert nach vorne fallen würde.

#### 4.1.3 Konzepte erstellen und Hardware konzipieren

Nach der Recherche wurde sich dazu entschieden, die Grundidee des Projekts aus Abbildung 4.5 als Orientierung zu nutzen. Ein bereits vorhandenes 3D-Modell des System, hat dabei geholfen, das entwickelte System und die dahinterstehenden Konzepte näher zu betrachten und zu verstehen[5].

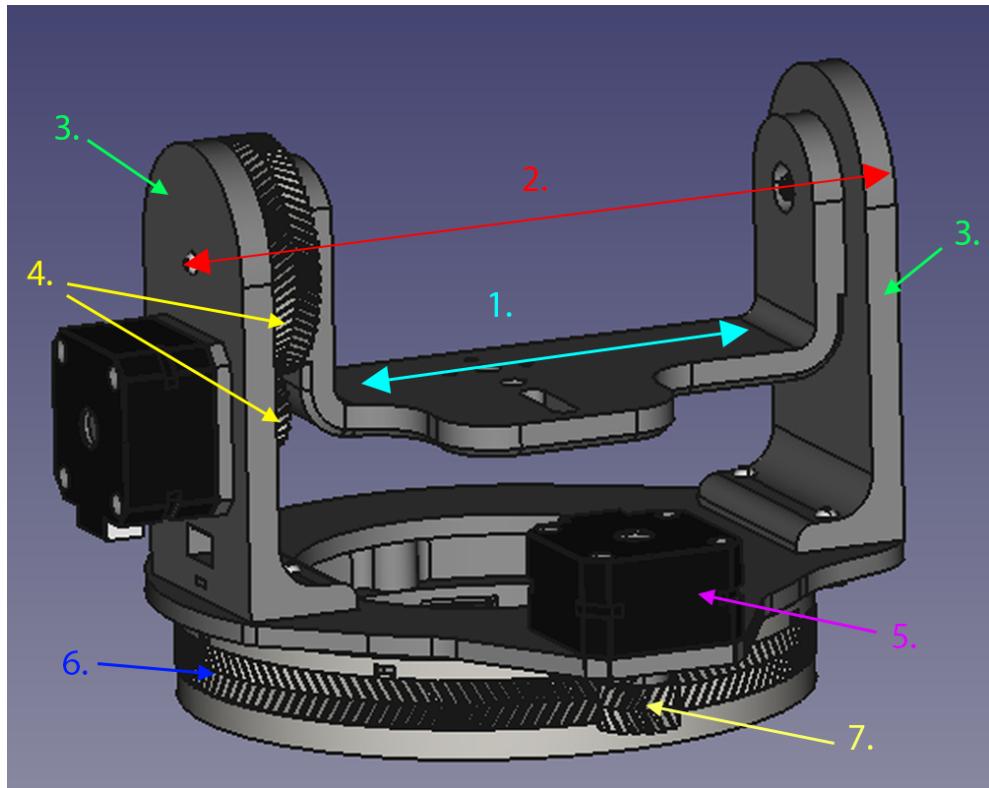


Abbildung 4.6: 3D-Modell der Pan-Tilt Einheit von Isaac879

### Tilt-Bewegung

Um die Tilt-Bewegung zu realisieren, wird das in Abbildung 4.6 gezeigte Prinzip übernommen. Zu sehen ist, dass die Kamera auf einer Fläche (4.6, 1.) unterhalb der Drehachse (4.6, 2.) montiert wird. Dadurch ergibt es sich, dass der Schwerpunkt der Kamera auf der Höhe der Drehachse ist. Des Weiteren fällt die Kamera bei Ausfall der Motoren somit immer wieder in die Grundposition zurück, wodurch keine ungewollten Effekte entstehen. Die groben Proportionen der seitlichen Befestigungsfüße (4.6, 3.) für die Tilt-Achse und die Zahnradübersetzung (4.6, 4.) werden aus dem Projekt von "Isaac879" übernommen. Beides hat sich in dem dazugehörigen Video [4] als gut umsetzbar und druckbar dargestellt. Des Weiteren werden durch d

### Pan-Bewegung

In Abbildung 4.6 ist zudem zu sehen, dass die Einheit die die Tilt-Bewegung durchführt und der frontal zu sehende Motor (4.6, 5.) , zusammen auf einer Plattform befestigt sind. Diese Plattform ist so gelagert, dass sich diese unabhängig von dem weiter unterliegendem Teil (4.6, 6.) drehen kann. Dieses Teil ist fixiert und an der außen Seite mit Zähnen bestückt. Dadurch kommt es zu einer Pan-Bewegung der ganzen oberen Plattform, sobald

der Motor (4.6, 5.) sich dreht und das Zahnrad (4.6, 7.) um die fixierte Basis (4.6, 6.) fährt.

Dieses Konzept, dass sich eine Hauptplattform um eine statische Basis dreht, wird für diese Studienarbeit übernommen. Um zu gewährleisten, dass System weniger Fehleranfällig durch Einflüsse von außen ist, wird der Motor (4.6, 5.) mit dem Zahnrad (4.6, 7.) die Pan-Bewegung nicht außerhalb der statischen Basis durchführen, sondern wie in Abbildung 4.7 zu sehen, innen-läufig.

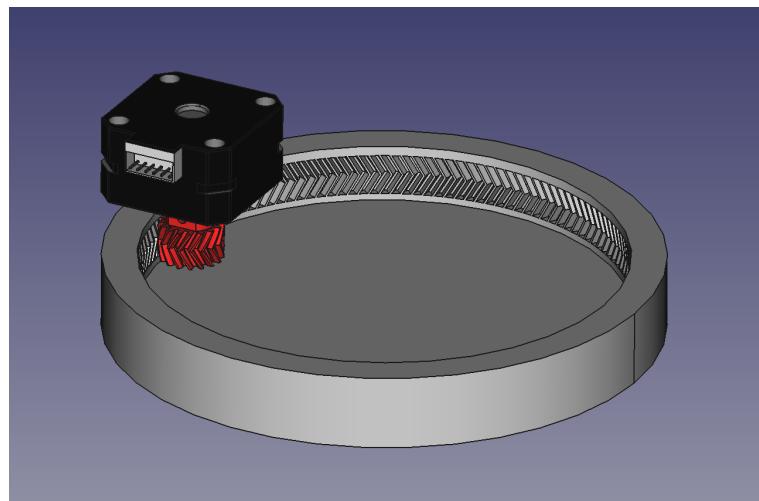


Abbildung 4.7: Visualisierung des grundsätzlichen Aufbaus eines innen läufigen Zahnrads.

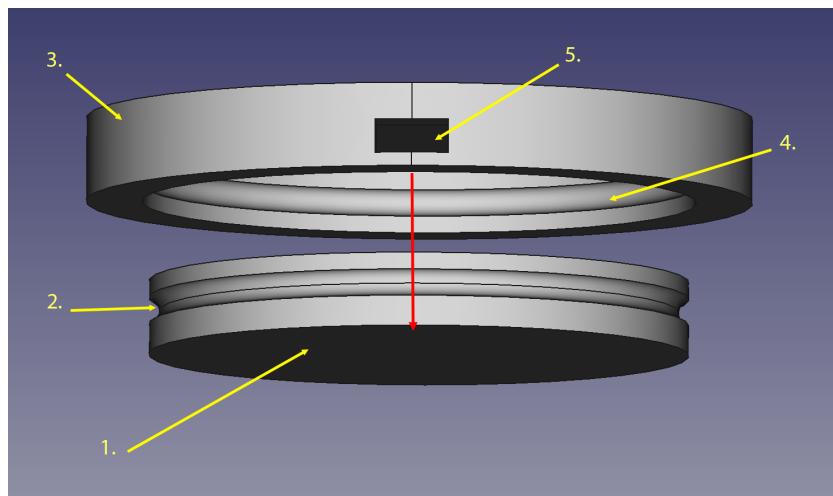


Abbildung 4.8: Statische Basis (1.) mit Mulde an Außenseite (2.). Hauptplattform (3.) mit Mulde an Innenseite (4.). Loch für Kugeln in Hauptplattform für Befüllung der Kugeln (5.).

### Kugellager

Damit die Drehbewegung der Hauptplattform um die statische Basis herum gewährleistet wird, verbindet ein Kugellager beide Teile miteinander. Für die Realisierung des Kugellager,

wird die Hauptplattform (4.8, 3.) um die Basis (4.8, 1.) gestülpt und jeweils die Außenseite der statischen Basis (4.8, 2.) und die Innenseite der Hauptplattform (4.8, 4.) erhalten eine rundliche Mulde. Nach dem übereinander Stülpen sind die Mulden nun auf einer Höhe, sodass durch ein Loch in der Hauptplattform (4.8, 5.) Kugeln in die Donut-förmige Aushöhlung eingeführt werden können.

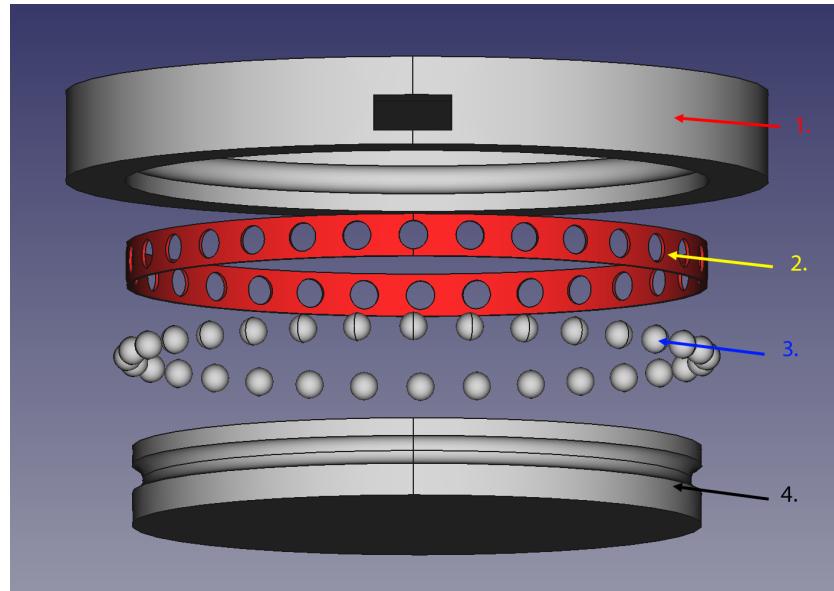


Abbildung 4.9: Komponenten des Kugellagers. Hauptplattform (1.), Abstandhalter-Ring (2.), Kugeln (3.), statische Basis (4.).

Damit nicht die ganze Mulde mit Kugeln gefüllt werden muss, kommt ein Ring (4.9, 2.) zwischen Hauptplattform (4.9, 1.) und Basis (4.9, 4.). Dieser Ring ist dafür da, die eingeführten Kugeln (4.9, 3.) an ihrem Platz zu halten.

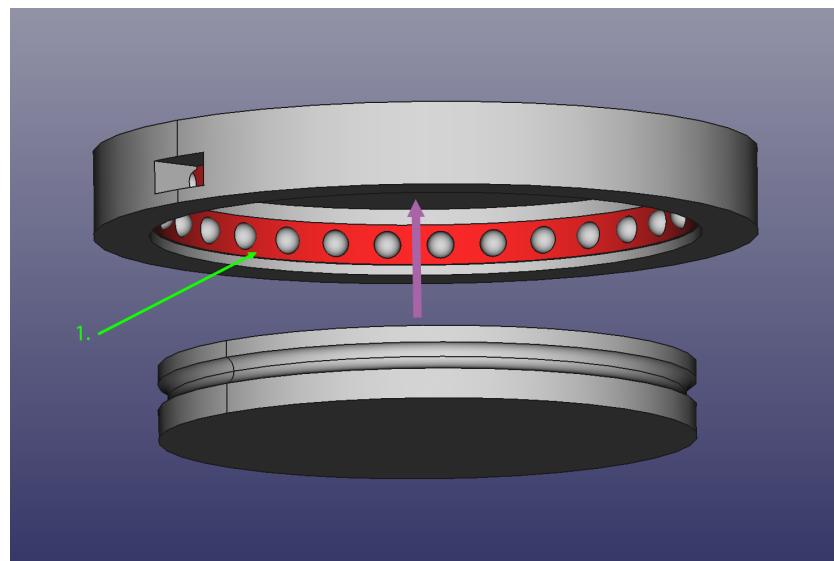


Abbildung 4.10: Kugeln mit Abstandhalter-Ring (1.) in Hauptplattform. Statische Basis drüber.

In Abbildung 4.10 ist zu sehen, wie die Kugeln mit dem Abstandhalte-Ring (4.10, 1.) in der Hauptplattform liegen. Diese Abbildung dient nur der Illustration, da die Hauptplattform inklusive Kugeln nicht mehr über die Basis gestülpt werden könnte. Die Verbindung der beiden Teile entsteht ausschließlich durch die Einführung von Kugeln durch das Loch in der Hauptplattform. Dies muss nach fertigem Einfüllen geschlossen werden.

## Slider

Die Bewegung des System entlang einer Achse im Raum, wird mit einem sogenannten *Slider* durchgeführt. Der Slider wurde ohne Anregung anderer Projekte entwickelt. Besonderer Fokus wurde darauf gelegt, dass die Bewegung im Raum gleichmäßig und die Länge variabel ist. Um eine möglichst gleichmäßige und ruckelfreie Bewegung zu gewährleisten, wurde sich dazu entschieden, das Pan-Tilt-System mit Hilfe eines Zahnriemens über zwei Aluminiumrohre zu ziehen.

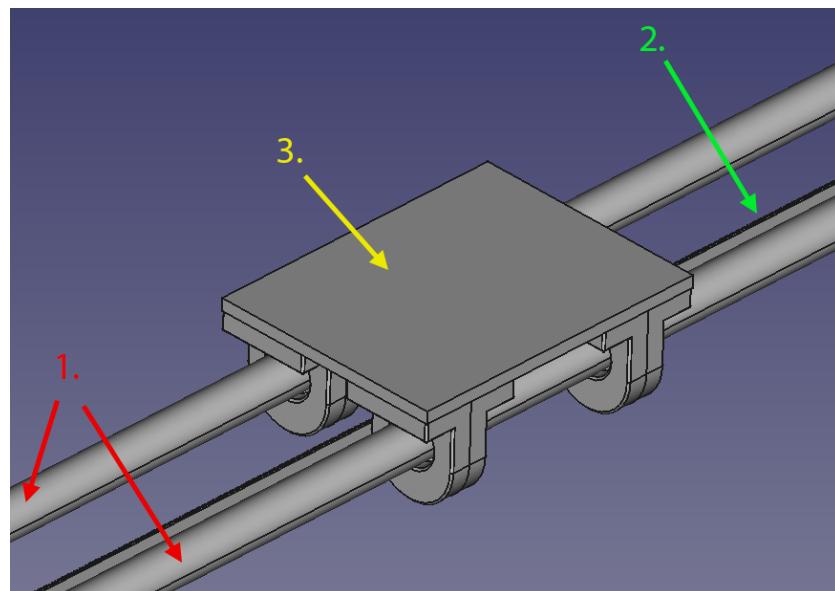


Abbildung 4.11: Slider Aufbau: Bewegliche Plattform (3.) fährt mit Hilfe von starrem Zahnriemen (2.) über zwei Aluminiumrohre (1.)

In Abbildung 4.11 ist zu sehen, wie eine Plattform (4.11, 3.) an zwei parallele Aluminiumrohre (4.11, 1.) befestigt ist. Das Pan-Tilt-System wird auf die gezeigte Plattform montiert. Die Realisierung der Bewegung der Plattform entlang der Aluminiumrohre wird in Abbildung 4.12 gezeigt.

Der Motor welcher für die Bewegung des Sliders zuständig ist, bewegt sich immer mit dem restlichen Pan-Tilt-System. Daher kann die Länge der Aluminiumrohre und des starren Zahnriemens beliebig verändert werden.

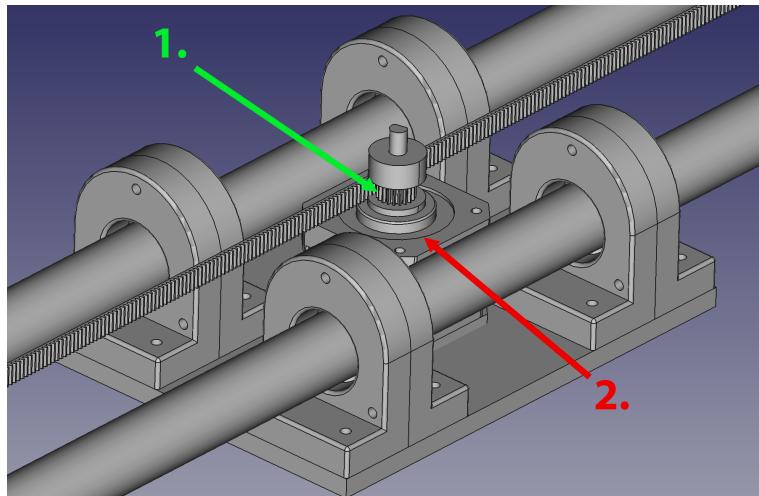


Abbildung 4.12: Slider Aufbau: Motor (2.) mit Zahnrad (1.) ist an der Plattform befestigt.  
Der Zahnriemen liegt so, dass das Zahnrad sich bei Drehung entlang des  
Zahnriemens bewegt.

Um die Bewegung möglichst gleichmäßig und ruckelfrei zu gestalten, werden die Befestigungen der Plattform an den Aluminiumrohren keinen direkten Kontakt zu den Aluminiumrohren haben. In die Befestigungen werden rund um die Aluminiumrohre Stahlkugeln platziert, welche sich zwar um sich selbst beliebig drehen können, allerdings nicht die Position verlassen können.

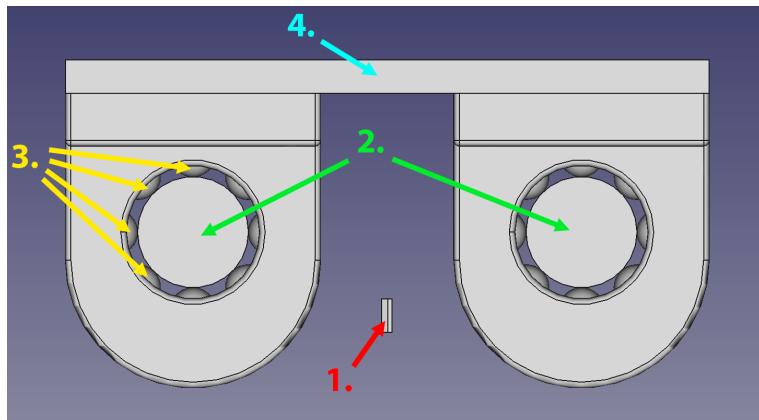


Abbildung 4.13: Slider Aufbau Querschnitt: Plattform mit Befestigungen (4.). Stahlkugeln (3.) innerhalb der Befestigungen, um über Aluminiumrohre (2.) zu rollen.  
Zahnriemen (1.) mittig. Motor und Zahnrad wie in Abbildung 4.12 zur  
Übersichtlichkeit nicht zu sehen.

## **Elektronik-Management**

Das Verarbeiten des Live-Bilds der Kamera erfolgt mit einem Raspberry Pi. Dieser ist in der Lage *Python*-Programme auszuführen und es kann für die Gesichtserkennung die bereits vorhandene Bibliothek *OpenCV* genutzt werden. Des Weiteren verfügt dieser über

USB-B Schnittstellen, welche für den Eingang des Live-Bildes benötigt werden. Da der Raspberry Pi allerdings nicht für präzise Schrittmotor Ansteuerungen geeignet ist, wird zusätzlich ein Arduino nano verwendet. Über serielle Schnittstellen des Raspberry Pis und des Arduino nanos können beide Einheiten bidirektional miteinander kommunizieren. Die Schrittmotoren arbeiten mit einer größeren Leitung als der Arduino nano bieten kann. Daher werden Motor-Treiber benötigt, welche mit Hilfe einer weiteren externen Stromquelle den Eingangsstrom des Arduino nanos verstärken. Um den Arduino nano und den Raspberry Pi nicht nur in einer Konsole über einen Computer zu steuern, werden ebenfalls ein kleines Display und ein Joystick in das System integriert. Darüber können die einzelnen Motoren angesteuert werden und das Gesichts-Tracking aktiviert und deaktiviert werden.

Um die Elektronikteile möglichst kompakt in das System zu integrieren, wird eine Box auf die Hauptplattform montiert, welche die ganze Elektronik beinhaltet.

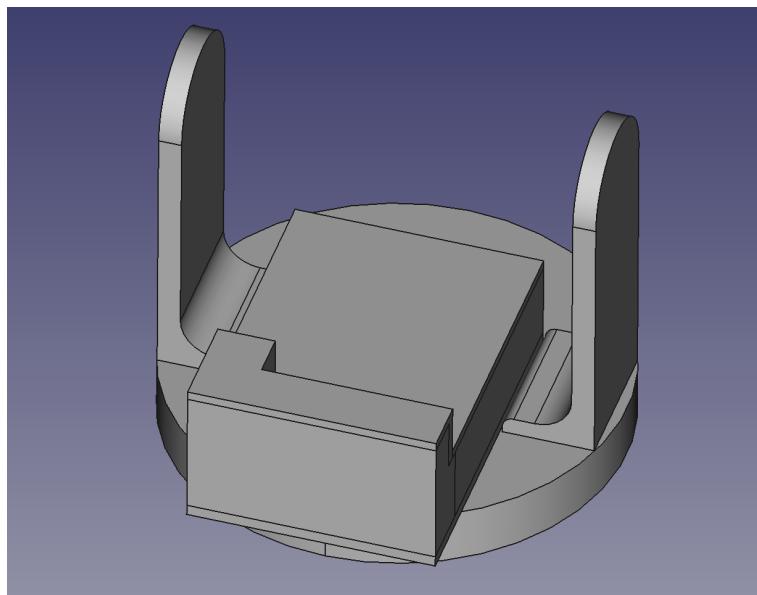


Abbildung 4.14: Box für sämtliche Elektronik auf der Hauptplattform und neben den Füßen für die Tilt-Bewegung.

### 4.1.4 Hardware konstruieren

#### Kamerahalterung

Die Größe des ganzen Systems hängt maßgeblich von der Größe der Kamera ab. Daher wird als erstes das Teil konstruiert, auf welches die Kamera montiert wird. Im Folgenden wird dieses Teil als *Kamerahalterung* benannt. Um die Maße der Kamerahalterung festzulegen, werden die ausschlaggebenden Maße der Kamera ermittelt.

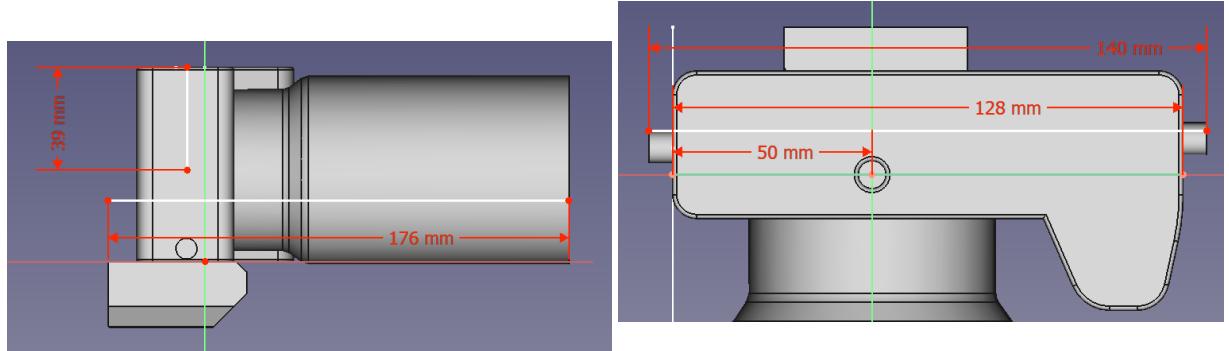


Abbildung 4.15: Seitliche, auf dem Kopf stehende Ansicht eines Modells der Sony A7 RII. Länge der Kamera-Objektiv Kombination und Höhe der Sensormitte angegeben.

Abbildung 4.16: Ansicht auf das Kameramodell von unten. Breite der Auflagefläche, Breite inklusive Zylinderförmige Befestigungen (Position in Abbildung 4.15 ersichtlich) und Abstand von Außenseite zu Mitte des Gewindelochs zur Befestigung der Kamera

Wie in Abbildung 4.16 illustriert, besitzt die Kamera mit einer 128 mm breite Auflagefläche. Daher muss die Auflagefläche der Kamerahalterung ebenfalls mindestens diese Breite aufweisen. Um einen kleinen Spielraum zu setzen, wird die Breite mit 131 mm definiert. In Abbildung 4.16 ist hinzu die Breite der Kamera inklusive zylinderförmige Halterungen ersichtlich. Die Halterungen sind nur oberhalb der Kamera vorhanden und in Abbildung 4.15 als Kreis erkenntlich. Diese haben zur Folge, dass die Breite oberhalb der Auflagefläche größer sein muss.

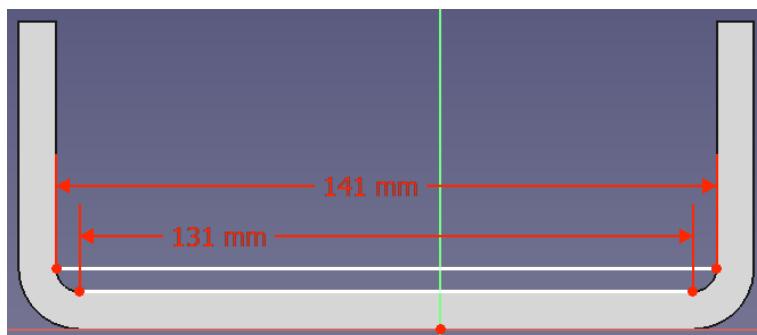


Abbildung 4.17: Kamerahalterung: Auflagefläche für Kamera = 131 mm, komplette Breite für Kamera = 141 mm

Als nächstes wird die Form der Auflagefläche bestimmt. Zu beachten ist dabei, dass das Befestigungsgewinde der Kamera nicht mittig ist. Zudem hat die Kamera-Objektiv Kombination durch das Objektiv einen stark nach vorne verlagerten Schwerpunkt. Um das Gewicht besser zu verteilen, wird durch eine separate Objektivstütze das Gewicht des Objektives direkt auf die Auflagefläche verlagert.

Um den Schwerpunkt der Kamera-Objektiv Kombination auf die Tilt-Drehachse (??, 4.) zu verlagern, werden die Löcher zur Befestigung der Objektivstütze vor die Drehachse

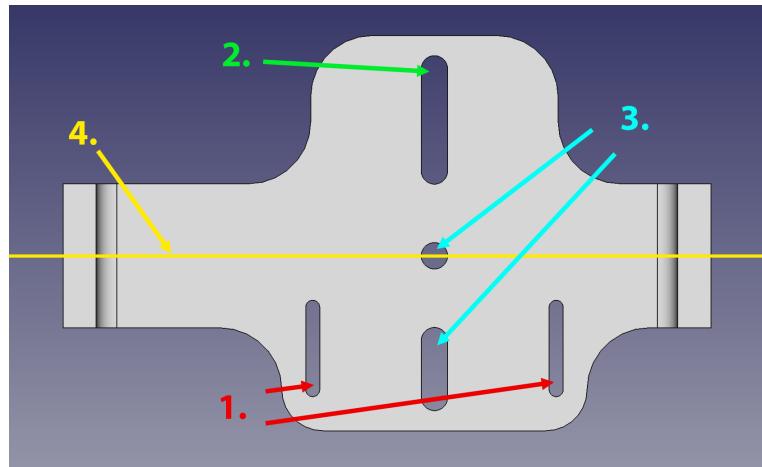


Abbildung 4.18: Kamerahalterung Ansicht von oben: Löcher für Befestigung der Objektivstütze (1.,  $r = 1,7$  mm), Loch für Befestigung der Kamera (2.,  $r = 3,2$  mm), Optionale Löcher (3.,  $r = 3,2$  mm)

(??, 1.) und das Loch für die Befestigung der Kamera etwas hinter die Drehachse (??, 2.) verlagert. Die “Löcher“ sind dabei so geformt, dass die letztendliche Position der Kamera und der Objektivstütze für Feineinstellungen variiert werden können. Um zu gewährleisten, dass andere Objekte auf die Kamerahalterung montiert werden können, werden zusätzliche Löcher integriert (??, 3.).

Die Kamerahalterung wird an einer Seite an einem Fuß gehalten und auf der anderen Seite mit einem Zahnrad verbunden welches mit dem anderen Fuß verbunden ist (siehe Abbildung 4.6).



**Objektivstütze**

**Zahnrad an Kamerahalterung für Tilt-Bewegung**

**Zahnrad an Motor für Tilt-Bewegung**

**Rechter Fuß**

**Linker Fuß**

**Hauptplattform**

**Statische Basis**

**Zahnrad an Motor für Pan-Bewegung**

**Abstandhalterung für Kugellager**

**Slider Plattform**

**Befestigungen an Aluminiumrohren**

**Motorbefestigung**

**Stabilisierungsplatte für Umlenkrollen**

**Füße zur Befestigung der Aluminiumrohre und des Zahnriemens**

**Elektronikbox**

#### **4.1.5 Benötigte Hardware einkaufen**

#### **4.1.6 Hardware zusammenbauen**

### **4.2 Software**

#### **4.2.1 Anforderungen an die Software stellen**

#### **4.2.2 UML-Diagramm konzipieren**

---

#### **4.2.3 Hardwareansteuerung implementieren**

# 5 Ergebnis

Referenzierung zu Aufgabenstellung.

→ Was habe ich geschafft, was habe ich nicht geschafft?

Als (Not-)Lösung: Tabelle der Aufgaben mit Spalte “konnte erledigt“ werden

# **6 Reflektion**

Was habe ICH gelernt (ich erlaubt)

Hier auch Gedanken zu Recycling des 3D-Mülls

# **7 Ausblick**

Ausblick mit Erweiterungsvorschlägen

# **8 Danksagung**

Hier kommt die Danksagung hin

# Literatur

- [1] Eyal Abraham. *Differential pan-tilt*. Mai 2020. URL: <https://www.youtube.com/watch?v=LF1JT0Pmddc> (besucht am 12.03.2022).
- [2] FreeCAD. *FreeCAD - Offizielle Website*. URL: <https://www.freecadweb.org/> (besucht am 07.03.2022).
- [3] Tobias Häberlein. *Technische Informatik - Ein Tutorium der Maschinenprogrammierung und Rechnertechnik*. Vieweg+Teubner, 2011. ISBN: 978-3-8348-1372-5.
- [4] Isaac879. *3D Printed DSLR Camera Pan Tilt Mount (Arduino/Stepper Driven) 2020*. Mai 2020. URL: <https://www.youtube.com/watch?v=uJ07mv4-0PY> (besucht am 12.03.2022).
- [5] Isaac879. *Pan-Tilt-Mount*, GitHub. März 2021. URL: <https://github.com/isaac879/Pan-Tilt-Mount> (besucht am 16.03.2022).
- [6] Python. *Python - Offizielle Website*. URL: <https://www.python.org/about/> (besucht am 06.03.2022).
- [7] ServoCity. *ServoCity Products: PT2645S Pan and Tilt System*. Juli 2017. URL: <https://www.youtube.com/watch?v=EfovZizgfh8> (besucht am 12.03.2022).
- [8] Wikipedia. *C-Programmiersprache*. URL: [https://de.wikipedia.org/wiki/C\\_\(Programmiersprache\)](https://de.wikipedia.org/wiki/C_(Programmiersprache)) (besucht am 06.03.2022).
- [9] Wikipedia. *Pan*. URL: [https://en.wikipedia.org/wiki/Panning\\_\(camera\)](https://en.wikipedia.org/wiki/Panning_(camera)) (besucht am 03.03.2022).
- [10] Wikipedia. *Tilt*. URL: [https://en.wikipedia.org/wiki/Tilt\\_\(camera\)](https://en.wikipedia.org/wiki/Tilt_(camera)) (besucht am 02.03.2022).
- [11] Wikipedia. *Ultimaker Cura - Wikipedia Artikel*. URL: <https://www.freecadweb.org/> (besucht am 07.03.2022).

# **Anhang**