

# DEM ROBOTER

## DOKUMENTATION

SEMINARWOCHE SOFTWAREENTWICKLUNG

G.Zordan  
K.Troxler  
A.Schmid  
N.Börtzler

# Inhalt

1. Einleitung.....	3
1.1 SCRUM-Team .....	4
1.2 SCRUM-Fortschritt.....	4
1.2.1 Sprint 1 .....	4
1.2.2 Sprint 1 Review .....	5
1.2.3 Sprint 2 .....	5
1.2.4 Sprint 2 Review .....	5
1.2.5 Sprint 3 .....	6
1.2.6 Sprint 3 Review .....	6
1.2.7 Sprint 4 .....	6
1.2.8 Sprint 4 Review .....	6
2. Usecase.....	8
3. Sequenzdiagramm .....	10
4. Flussdiagramm Code .....	11
5. Bauanleitung Roboter.....	12
a) Einheit Achse D.....	12
b) Einheit Achsen B und C .....	13
c) Verbinden der Achsenkonstruktion Finger.....	14
d) Fundament mit Achse A.....	15
e) Verbindungselement Achse A mit Fingereinheit.....	17
f) Montage vom Verbindungsmodul mit Achse A.....	19
g) Vorkonfektion EV3.....	20
h) Montage EV3 Modul an Fundament.....	22
i) Montage des Fingers mit Fundament .....	24
6. Berechnungen .....	26

# 1. EINLEITUNG

Die Seminarwoche steht bevor und wir haben die spannende Aufgabe, ein Softwareprojekt nach der agilen Methode Scrum durchzuführen. Doch anstatt nur theoretisch zu arbeiten, wird in diesem Projekt ein Lego-Mindstorms-Roboter als praktisches Beispiel verwendet. Wir werden das Projekt in einer Gruppe bearbeiten und uns dabei an die vorgegebenen Standards halten. Dabei haben wir die Freiheit, ein beliebiges Thema für unser Projekt auszuwählen, solange es ausreichend „Komplexität“ aufweist. Die Seminarwoche ist strukturiert und bietet uns verschiedene Themen und Aktivitäten an jedem Tag, um uns in die Welt des Softwaredesigns, der Projektplanung und der agilen Entwicklung einzuführen.

Am Montag steht Softwaredesign und UML im Fokus, wobei wir lernen, wie man effektiv Software entwirft und UML-Diagramme erstellt. Es werden auch Gruppen gebildet und die Projektthemen-Suche steht auf dem Programm. Die Einführung in Projektmanagement mit GitHub hilft uns, die grundlegenden Konzepte der Versionskontrolle und Zusammenarbeit mit einem verteilten Repository zu verstehen.

Am Dienstag steht die definitive Projektwahl an. Danach beginnt die eigentliche Arbeit am Projekt mit dem ersten Sprint. Wir setzen unser Wissen aus dem Softwaredesign und der UML in die Praxis um und starten die Entwicklung unseres Lego-Mindstorms-Roboters.

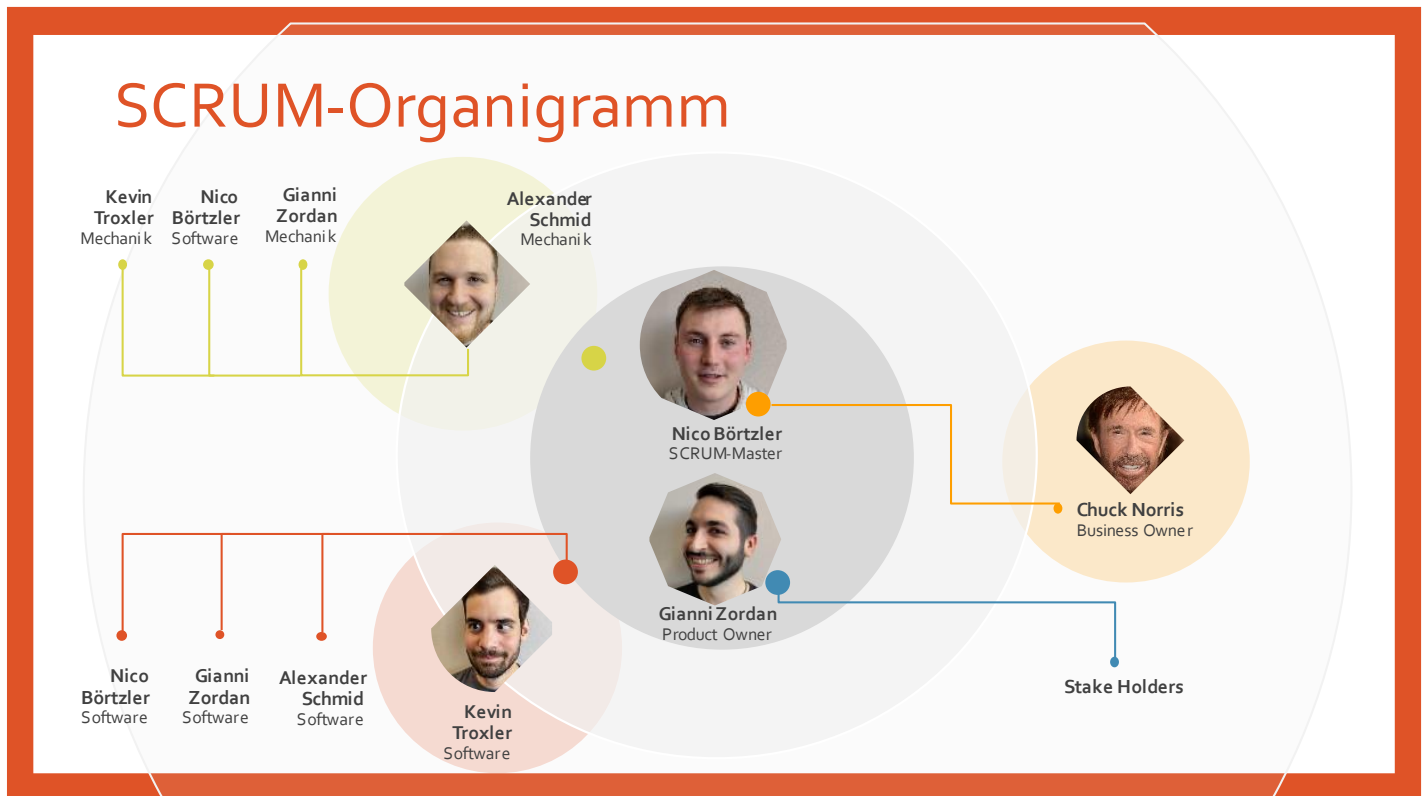
Der Mittwoch steht ganz im Zeichen der Arbeit am Projekt mit dem zweiten Sprint. Wir setzen dabei unsere Projektmanagement- und Entwicklungskenntnisse ein, um unser Roboter weiterzuentwickeln und die Funktionalitäten zu erweitern.

Am Donnerstag geht es weiter mit dem dritten Sprint, bei dem wir intensiv an unserem Projekt arbeiten und unser Roboter optimieren. Es ist auch Zeit für Feinabstimmungen und Verbesserungen, um sicherzustellen, dass der Roboter den gewünschten Anforderungen entsprechen.

Am Freitag steht der letzte Sprint an, bei dem wir unser Projekt abschliessen und unser Lego-Mindstorms-Roboter fertigstellen. Es ist Zeit für die Abgabe und Präsentation des Projektes, bei der wir unsere Arbeit vorstellen und unsere Erfahrungen in der agilen Entwicklung und im Projektmanagement teilen können.

Insgesamt bietet die Seminarwoche uns die Möglichkeit, in einem realen Projektumfeld nach Scrum zu arbeiten und praktische Erfahrungen in Softwaredesign, der agilen Entwicklung und im Projektmanagement zu sammeln. Es ist eine intensive Woche voller Herausforderungen und Lernmöglichkeiten, die uns wertvolle Einblicke in die Welt der Softwareentwicklung bietet.

## 1.1 SCRUM-Team



## 1.2 SCRUM-Fortschritt

Wir haben uns grundsätzlich an die SCUM-Methode gehalten, um das Projekt möglichst effizient durchführen zu können. Dabei haben wir nach den Sprints die Prioritäten festgelegt. Die Tasks haben wir in „Issues“ im GitHub festgehalten. Daraus haben wir die Sprint Backlogs erstellt und täglich die Sprint Review-Meetings gehalten, um immer auf dem aktuellen Stand zu bleiben.

Anschliessend an die Meetings am Abend wurde eine Retrospektive gehalten, um die nächsten Arbeiten des anfallenden Sprints allenfalls neu zu bewerten.

Das Team hat den Product Owner regelmässig über den Fortschritt informiert, während der Scrum-Master die Regeltermine überwacht und organisiert hat.

### 1.2.1 Sprint 1

Die Ziele, welche wir uns für den Sprint 1 gesetzt haben:

- Mechanischer Aufbau des Roboters
- Mechanischer Nullpunkt zum Kalibrieren des Roboters erstellen
- Testen der Mechanischen Haltefähigkeit der verschiedenen Achsen bei verschiedenen Winkeln.
- Tastaturreaster für Absolutpositionen erstellen inkl. Tabelle erstellen mit Absolutpositionen
- Roboterpositionen definieren (Nullpunkt und Wartepositionen definieren)
- Softwareprogramm eröffnen

## 1.2.2 Sprint 1 Review

Nach Abschluss des Sprints führten wir ein Sprint-Review durch, um die Fortschritte der Arbeiten zu präsentieren und alle auf den gleichen Stand zu führen.

Die Arbeiten konnten im Grossen und Ganzen zügig und erfolgreich erledigt werden. Wesentlich hatten wir Probleme beim mechanischen Aufbau des Roboters und der Haltefähigkeit der Achsen. Durch den Einbau eines 2. Motors bei der Achse BC konnte die Stabilität bedeutsam verbessert werden.

## 1.2.3 Sprint 2

Die Ziele, welche wir uns für den Sprint 2 gesetzt haben:

- Anleitung für den Aufbau des Roboters erstellen
- Initialisierung Programmieren
  - Referenzpunkt und Nullpunkt anfahren
- Formeln für die Berechnungen des Roboters erstellen
  - Formeln in der Software testen
- Koordinatenraster für die Buchstaben erstellen

## 1.2.4 Sprint 2 Review

Die Erstellung der Anleitung für den Aufbau des Roboters verlief reibungslos. Wir haben alle erforderlichen Schritte detailliert beschrieben und mit Bildern ergänzt. So kann jeder Benutzer den Roboter leicht aufbauen, ohne dass es zu Problemen kommt.

Bei der Programmierung der Initialisierung sind wir auf einige Schwierigkeiten gestossen. Der Referenzpunkt und der Nullpunkt mussten sorgfältig programmiert werden, um sicherzustellen, dass der Roboter genau an diesen Punkten anhält. Die Schwierigkeit war, das Mechanische Spiel möglichst gering zu halten. Mit verschiedenen Geschwindigkeiten und Feinabstimmungen konnten wir das Problem so erfolgreich, wie es das Spiel zulässt, lösen.

Die Erstellung der Formeln für die Berechnungen des Roboters erwies sich als eine grosse Herausforderung. Wir mussten uns tief in die Mathematik einarbeiten und die Formeln mehrfach testen, um sicherzustellen, dass sie korrekt funktionieren. Aber nach einigen Iterationen konnten wir die Formeln erfolgreich in die Software integrieren und sie funktionieren nun einwandfrei. Die Skizze welche wir uns dazu erstellt haben, diente uns als Master für alle Arbeitsgruppen.

Schliesslich haben wir auch das Koordinatenraster für die Buchstaben erfolgreich erstellt. Wir haben anhand eines Excel-Files die Koordinaten für die einzelnen Buchstaben eingegeben. Das Koordinatenraster ermöglicht es dem Roboter, die Buchstaben präzise anzufahren. Die Schwierigkeit hier, war das genaue Ausmessen der Fix-Werte (Roboterachsen) und des Nullpunktes.

## 1.2.5 Sprint 3

Die Ziele, welche wir uns für den Sprint 3 gesetzt haben:

- Grundaufbau des Codes
- UseCase überarbeiten
- Neue Tastaturreaster
  - Grösserer Raster erstellen
- Code Zusammenfügen
  - Einzelne Codes (Berechnungen, Raster) in Hauptcode einfügen
  - Koordinaten vom neuen Raster hinterlegen
- Flussdiagramm
  - Schematische Darstellung des Codes
- Sequenzdiagramm erstellen

## 1.2.6 Sprint 3 Review

Der Grundaufbau des Programms erfolgte gut. Probleme bestanden hierbei den Code so zu gestalten, damit die inkrementalen Werte und die korrekte Richtung der Bewegung der Achsen übereinstimmten.

Der Name der einzelnen Dokumente wurde so angepasst, damit der UseCase jedes Dokuments den Namen trägt.

Wir mussten den Tastaturreaster vergrössern, weil die Toleranz in Kombination der Achsen zu gross war und dadurch keine sichere Betätigung der Taste gewährleistet werden konnte. Hierbei gab es Probleme beim Ausdrucken der richtigen Skalierung des Rasters.

Nachdem der Raster und die Koordinaten neu festgelegt wurden, konnte der Hauptcode zusammengefügt werden.

## 1.2.7 Sprint 4

Die Ziele, welche wir uns für den Sprint 4 gesetzt haben:

- General Test
- PowerPoint Präsentation fertigstellen
- Scrum-Fortschritte dokumentieren
- Grossartiges Video des Roboters erstellen

## 1.2.8 Sprint 4 Review

Der General Test war der wichtigste Schritt in diesem Sprint, um sicherzustellen, dass der Roboter bereit für den Einsatz ist. Wir haben alle Funktionen des Roboters getestet, um sicherzustellen, dass sie ordnungsgemäss funktionieren. Während des Tests haben wir einige kleinere Probleme festgestellt, wie z.B. die Treffgenauigkeit nicht 100% gewährleistet wurde. Dafür haben wir den Nullpunkt der Achse A, anstatt von der Kalibrierposition in die Nullposition zu fahren, haben wir die Nullposition direkt gesetzt. Somit summiert sich die Fehlposition nicht auf.

Wir haben alle unsere Ziele erreicht und unsere Fortschritte dokumentiert. Wir haben insgesamt viele Probleme gelöst und unser Roboterprojekt erfolgreich abgeschlossen. Wir sind stolz auf unser Team und freuen uns darauf, zukünftige Herausforderungen anzunehmen.

## 2. USECASE

### Name:

Ein LEGO Mindstorms Roboter mit 3 Achsen soll programmiert werden, um eine Texteingabe auf einer Tastatur nachzutippen.

### Akteure:

- Programmierer: Erstellt das Programm und führt die Programmierung durch
- Roboter: Führt die Aufgabe aus
- Schaumstoff-Tastatur: Eingabegerät, auf das der Roboter die Eingabe tätigt
- Mechaniker: Baut den Roboter vorschriftsgemäss auf

### Auslöser:

Es soll ein Roboter erstellt werden, der in der Lage ist, Texteingaben auf einer Schaumstoff-Tastatur nachzutippen.

### Vorbedingungen:

- Der LEGO Mindstorms Roboter wurde erfolgreich zusammengebaut und konfiguriert.
- Die Schaumstoff-Tastatur ist vor dem Roboter korrekt platziert.
- Der Programmierer hat eine geeignete Programmierumgebung eingerichtet.

### Nachbedingungen:

- Der Roboter hat den Text korrekt nachgetippt.
- Das Programm wurde erfolgreich ausgeführt.

### Erfolgs-Szenario:

- Der Programmierer schaltet den Roboter ein und stellt sicher, dass alle Systeme ordnungsgemäss funktionieren.
- Der Programmierer erstellt das Programm, um die Texteingabe auf der Tastatur zu verarbeiten und auf dem Bildschirm auszugeben.
- Der Roboter empfängt den Text über das Programm.
- Der Roboter führt die Bewegungen der 3 Achsen aus, um den Text nachzutippen.
- Das Programm endet, nachdem der Roboter den Text vollständig nachgetippt hat.

### Erweiterungen:

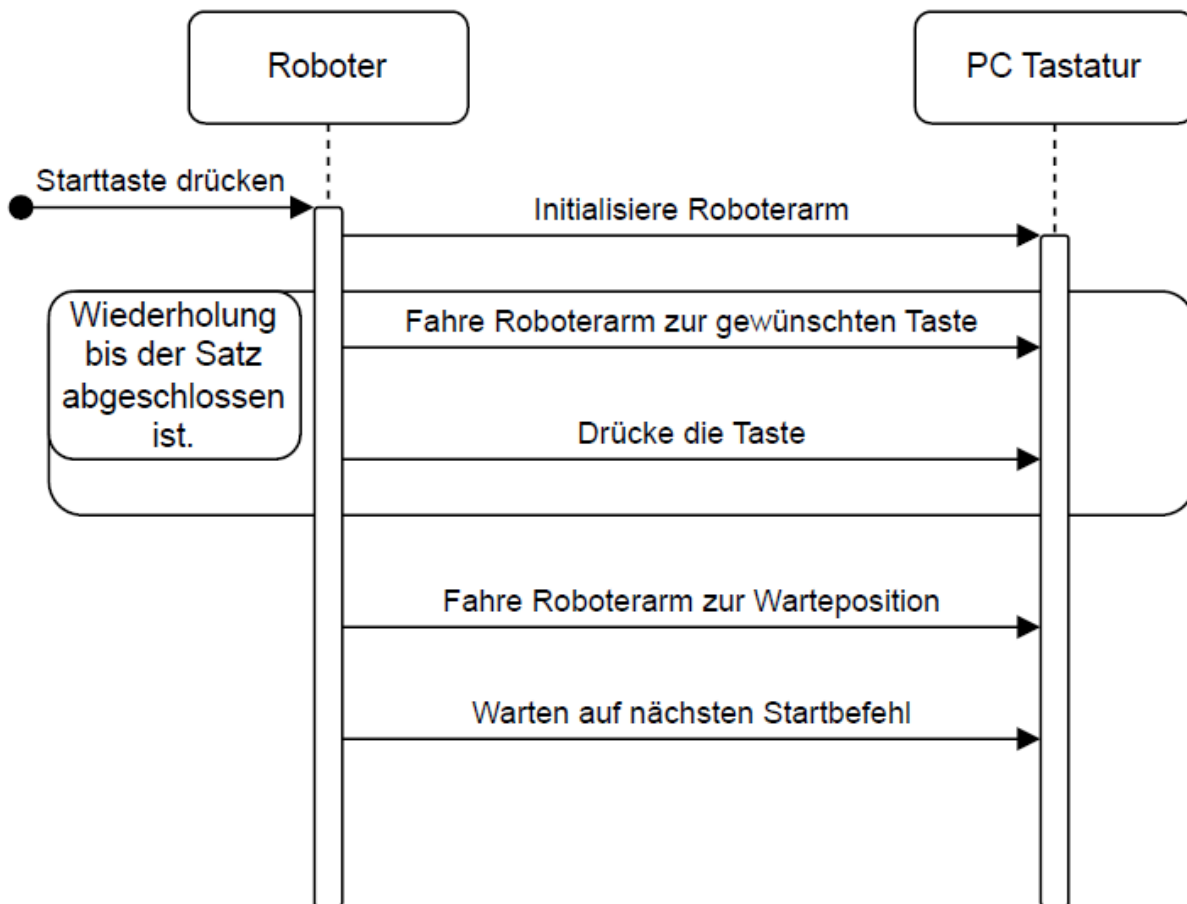
- Wenn der Roboter ein Problem während der Ausführung des Programms feststellt, sendet er eine Fehlermeldung an den Programmierer, um Unterstützung zu erhalten.
- Der Programmierer kann das Programm erweitern, um dem Roboter weitere Funktionen hinzuzufügen, wie z.B. Spezial-Tasten der Tastatur.
- Erweiterung des Programms, dass auch eine englische Tastatur zu implementiert werden kann.
- Es kann zwischen zwei verschiedenen Geschwindigkeiten gewählt werden.



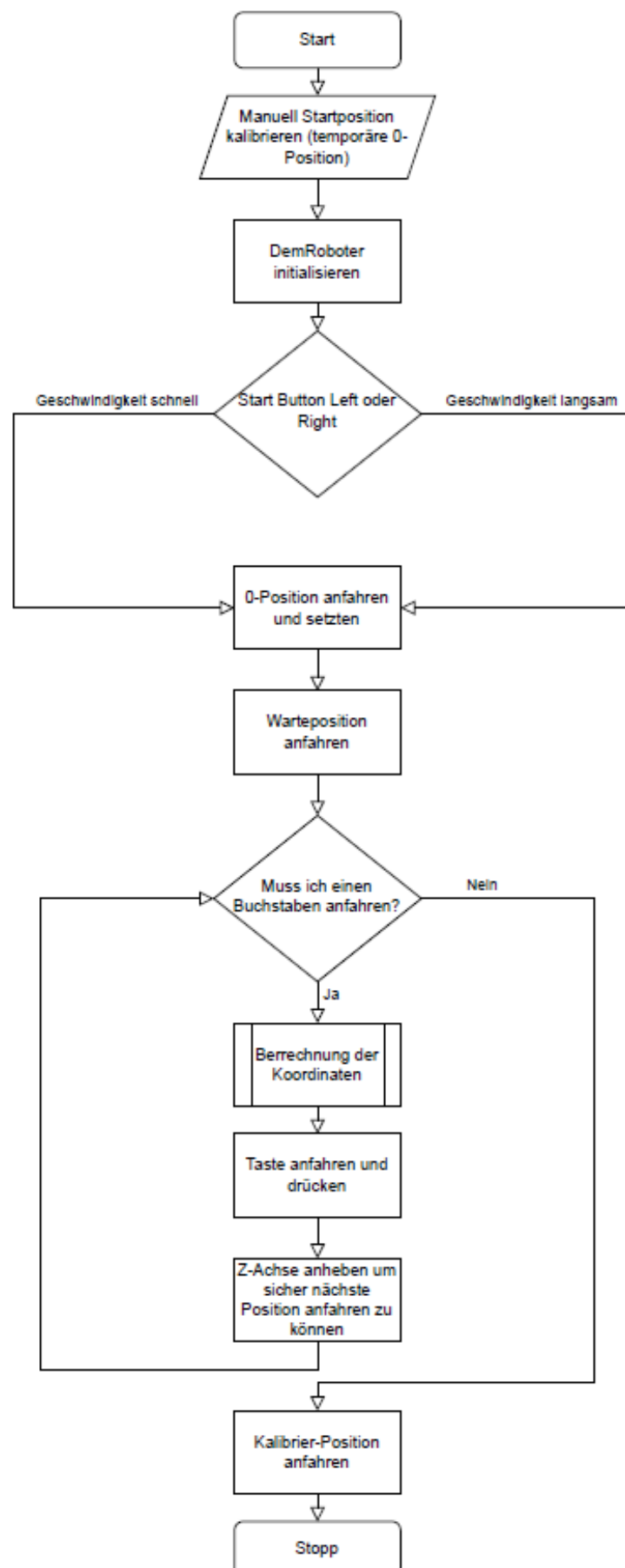
### **Nichtfunktionale Anforderungen:**

- Der Roboter soll 100% der Tasten treffen.
- Der Roboter soll ein Wort mit mindestens 4 Buchstaben schreiben können.
- Der Roboter soll für ein Wort von 4 Buchstaben nicht länger als eine Minute haben.
- Der Roboter soll mehr als ein Wort schreiben können.

### 3. SEQUENZDIAGRAMM



## 4. FLUSSDIAGRAMM CODE



# 5. BAUANLEITUNG ROBOTER

## a) Einheit Achse D

Bezeichnung	Menge in Stück
Schwarzer Steckverbinder	16
Beige Steckverbinder	5
Graue Führung 7 Loch	4
Graue Führung 5 Loch mit Gummiband	1
Weisse Führung 15 Loch	1
Starker Motor	1
Grauer gerader Doppelverbinder	1

Verbinden Sie die Komponenten wie in den Abbildungen 1-1 und 1-2 da gestellt.

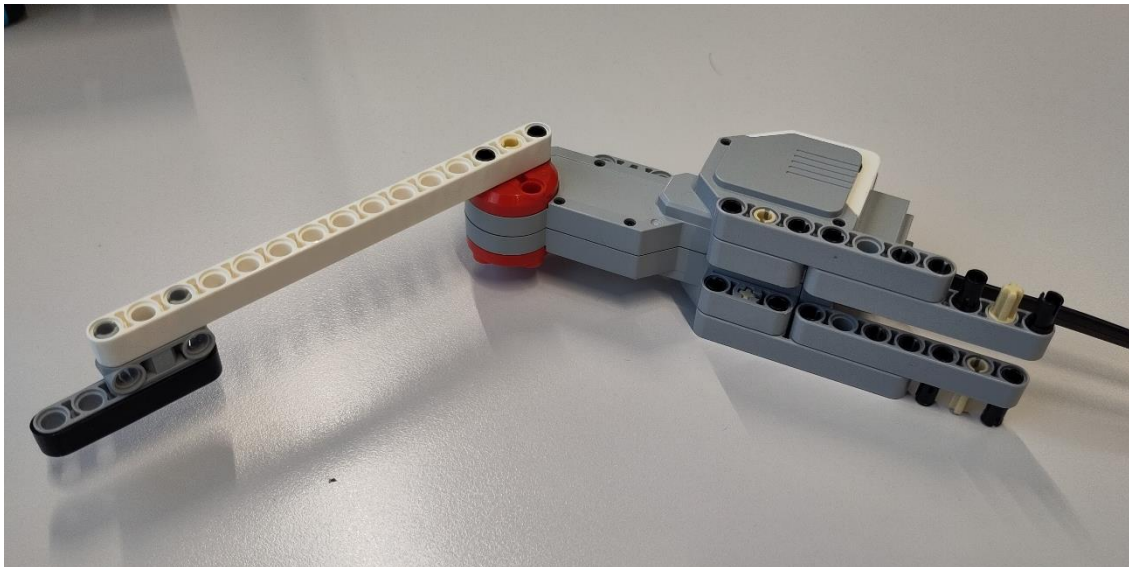


Abbildung 0-1 Achse D

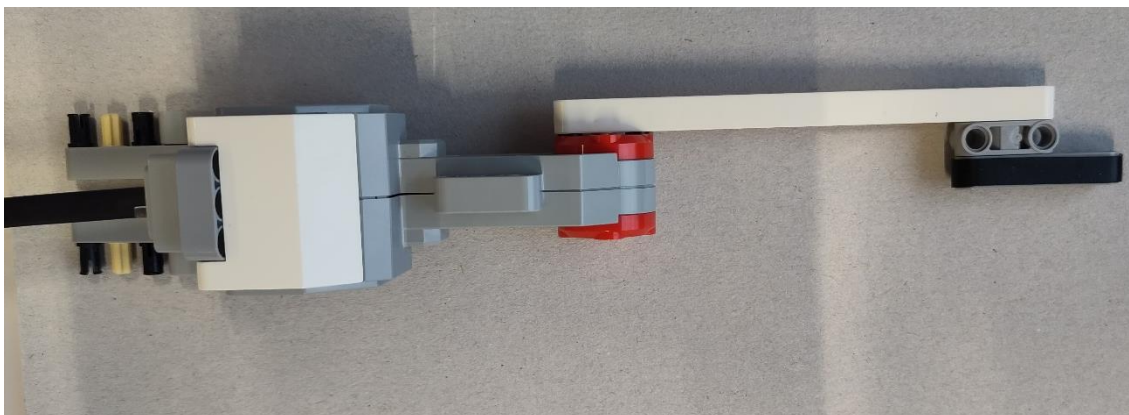


Abbildung 0-1 Achse D

## b) Einheit Achsen B und C

Bezeichnung	Menge in Stück
Schwarzer Steckverbinder	4
Blauer Doppelverbinder	10
Grauer kleiner Rahmen	2
Grauer Führung 9 Loch	1
Grauer Führung 7 Loch	1
Starker Motor	2

Verbinden Sie die Komponenten wie in den Abbildungen 2-1 und 2-2 da gestellt.

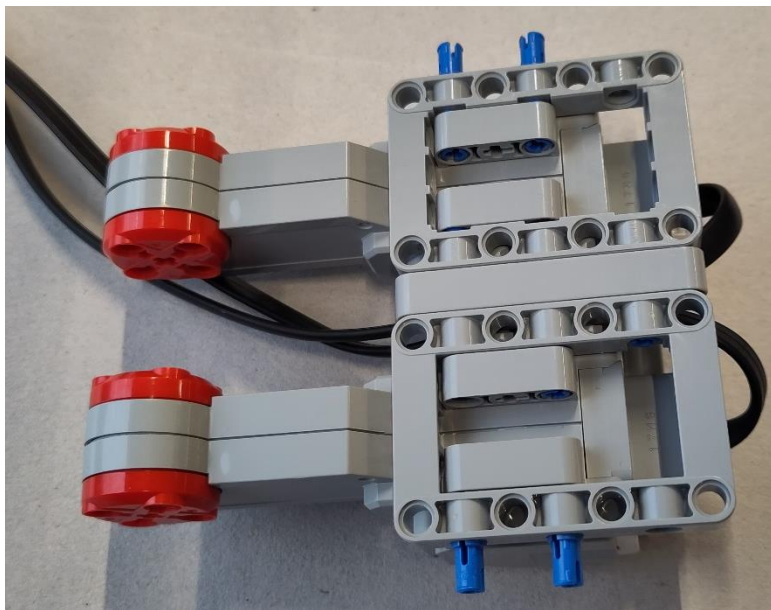
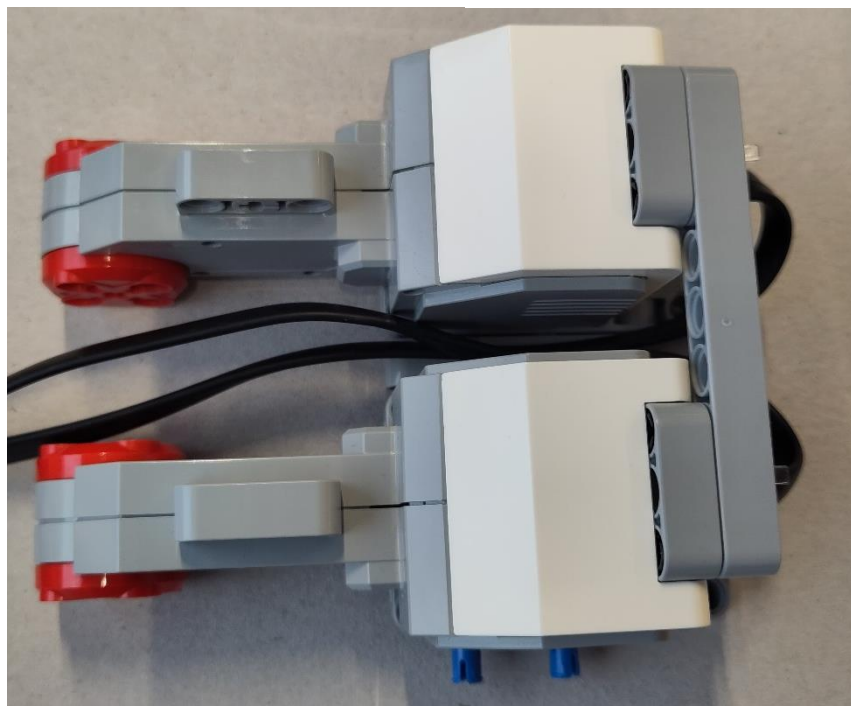


Abbildung 0-1 Achse B und C

Abbildung 0-2 Achse B und C



### c) Verbinden der Achsenkonstruktion Finger

Bezeichnung	Menge in Stück
Baugruppe Einheit Achse D	1
Baugruppe Einheit Achse B und C	1

Verbinden Sie die Komponenten wie in den Abbildungen 3-1 und 3-2 da gestellt.

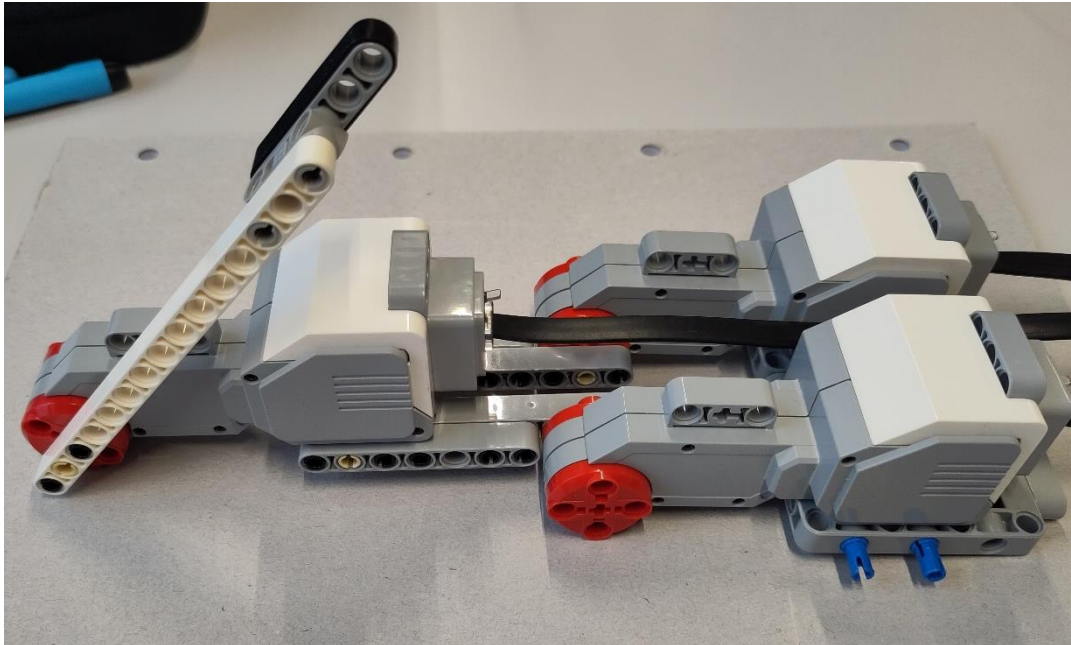


Abbildung 0-1 Achsenkonstruktion Finger

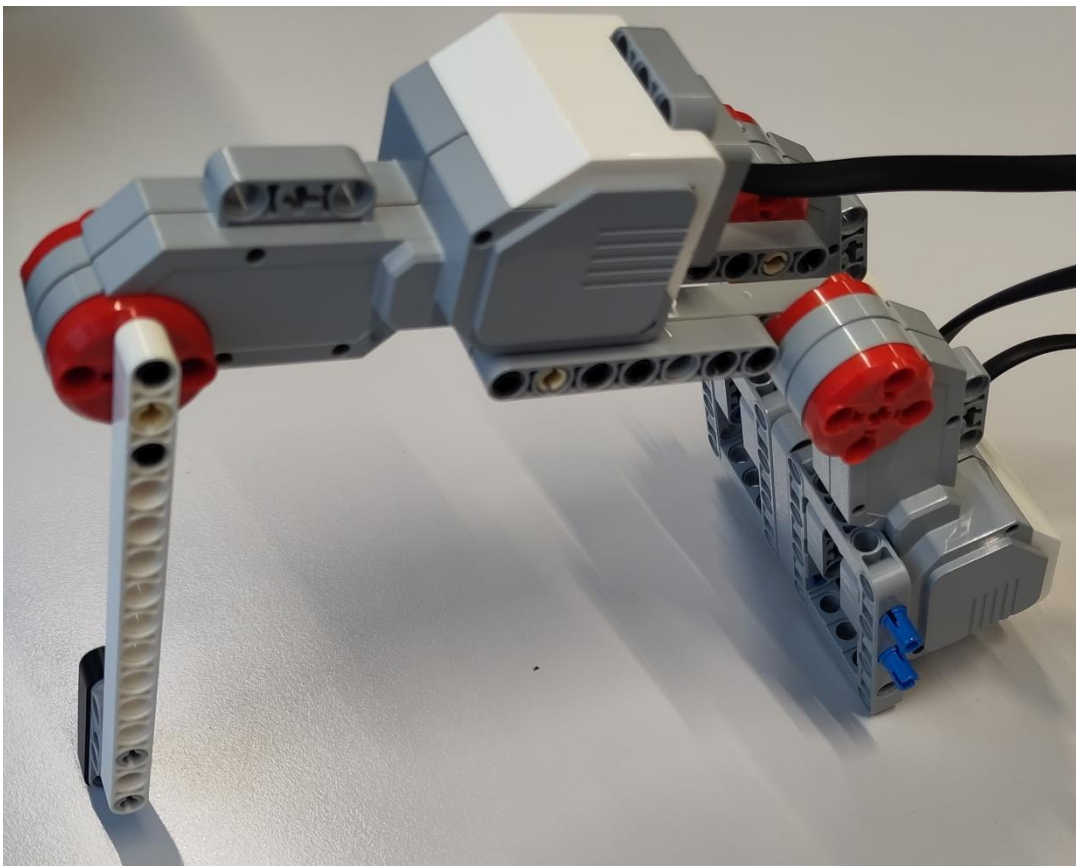


Abbildung 0-2 Achsenkonstruktion Finger



## d) Fundament mit Achse A

Bezeichnung	Menge in Stück
Schwarzer Steckverbinder	54
Schwarze Kreuz kurz	1
Schwarze T- Stück	2
Blauer Doppelverbinder	29
Roter Doppelverbinder	1
Graue Kreuz mit Anschlag	1
Weisse Führung 15 Loch	16
Weisse L-Stück	3
Graue Führung 13 Loch	14
Graue Führung 11 Loch	3
Graue Führung 9 Loch	4
Graue Führung 7 Loch	3
Graue Führung 5 Loch	7
Graue Kreuzmodul	9
Graue Doppelmodul	5
Graue Doppelhülse	5
Rote L-Stück	2
Rotes Winkelteil	1
Blaue Führung 3 Loch	2

Verbinden Sie die Komponenten wie in den Abbildungen 4-1, 4-2 und 4-3 da gestellt.

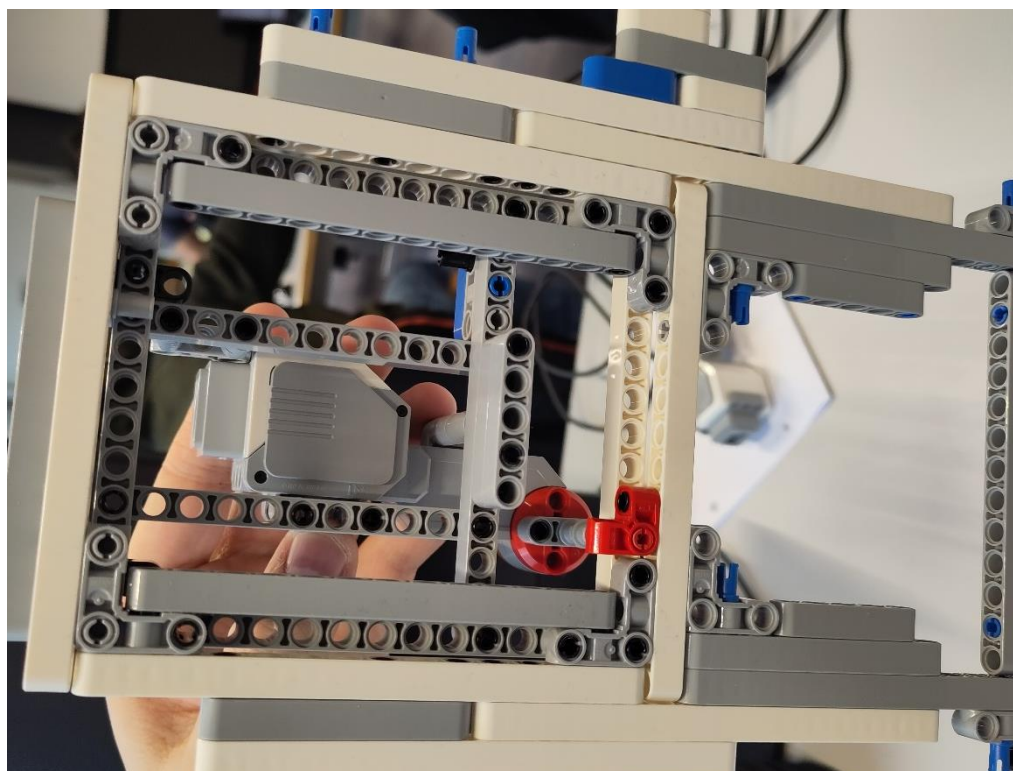


Abbildung 0-1 Fundament mit Achse A

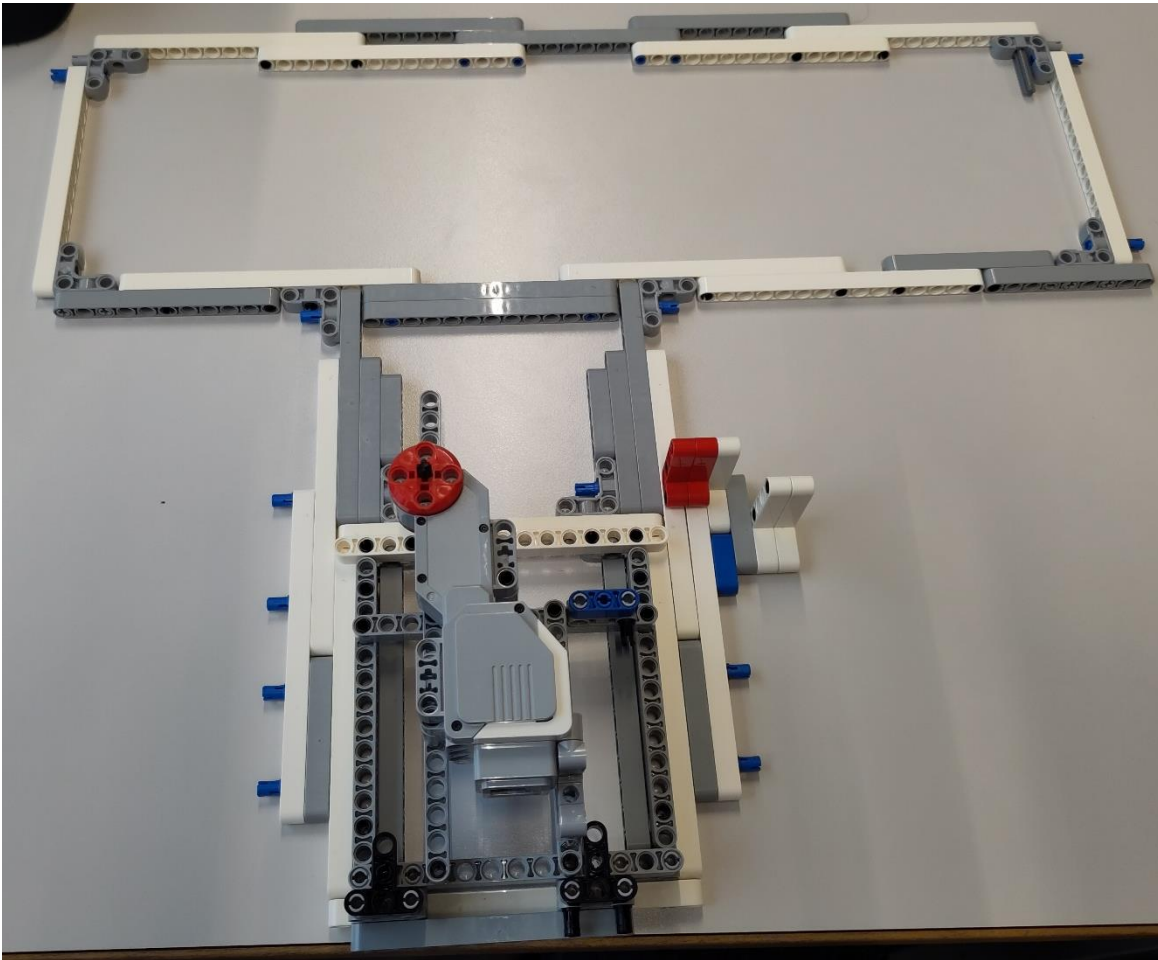


Abbildung 0-2 Fundament mit Achse A

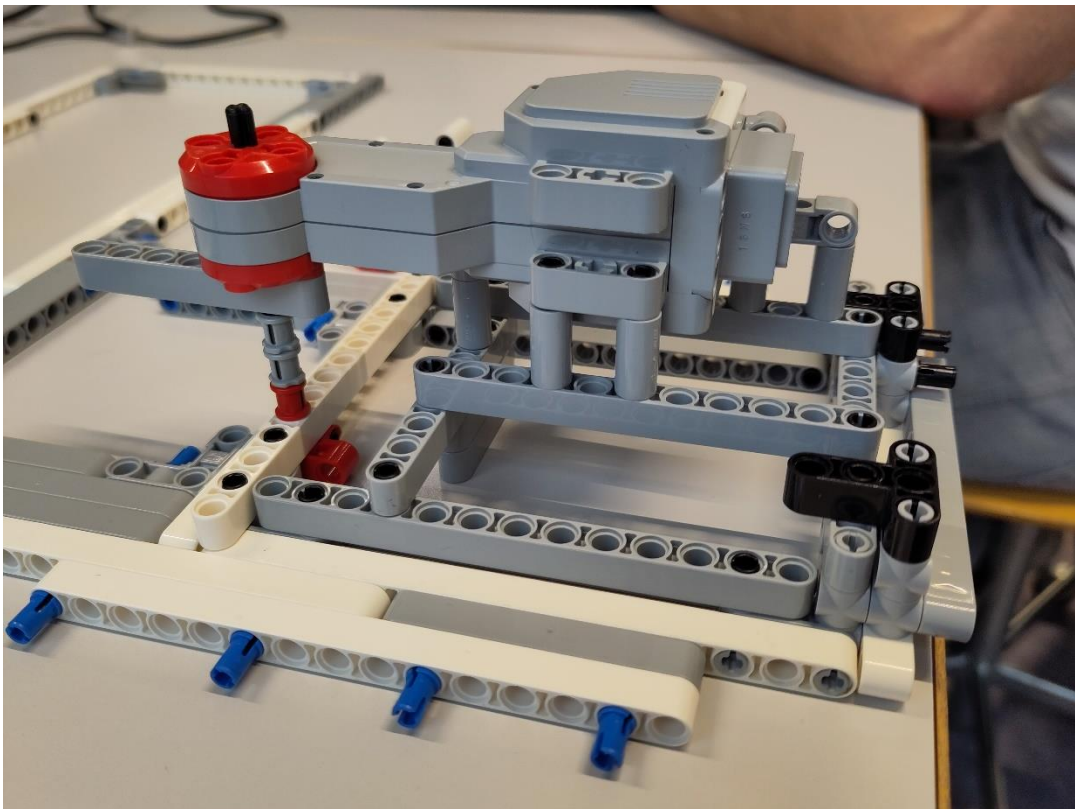


Abbildung 0-3 Fundament mit Achse A



## e) Verbindungselement Achse A mit Fingereinheit

Bezeichnung	Menge in Stück
Schwarzer Steckverbinder	12
Blaue Doppelverbinder	7
Blaue Kreuzverbinder	3
Graue Führung 7 Loch	1
Graues 3 Loch-Modul + Kreuz	2
Graue 45° Führung	2
Graues Doppelmodul	2
Graue Rahmen klein	3
Graue Dreiecksplatte	2
Weisse Führung 45°	1
Weisse L-Stück	2
Rote Führung 3 Loch	1
Rote L-Stück	2

Verbinden Sie die Komponenten wie in den Abbildungen 5-1, 5-2 und 5-3 da gestellt.

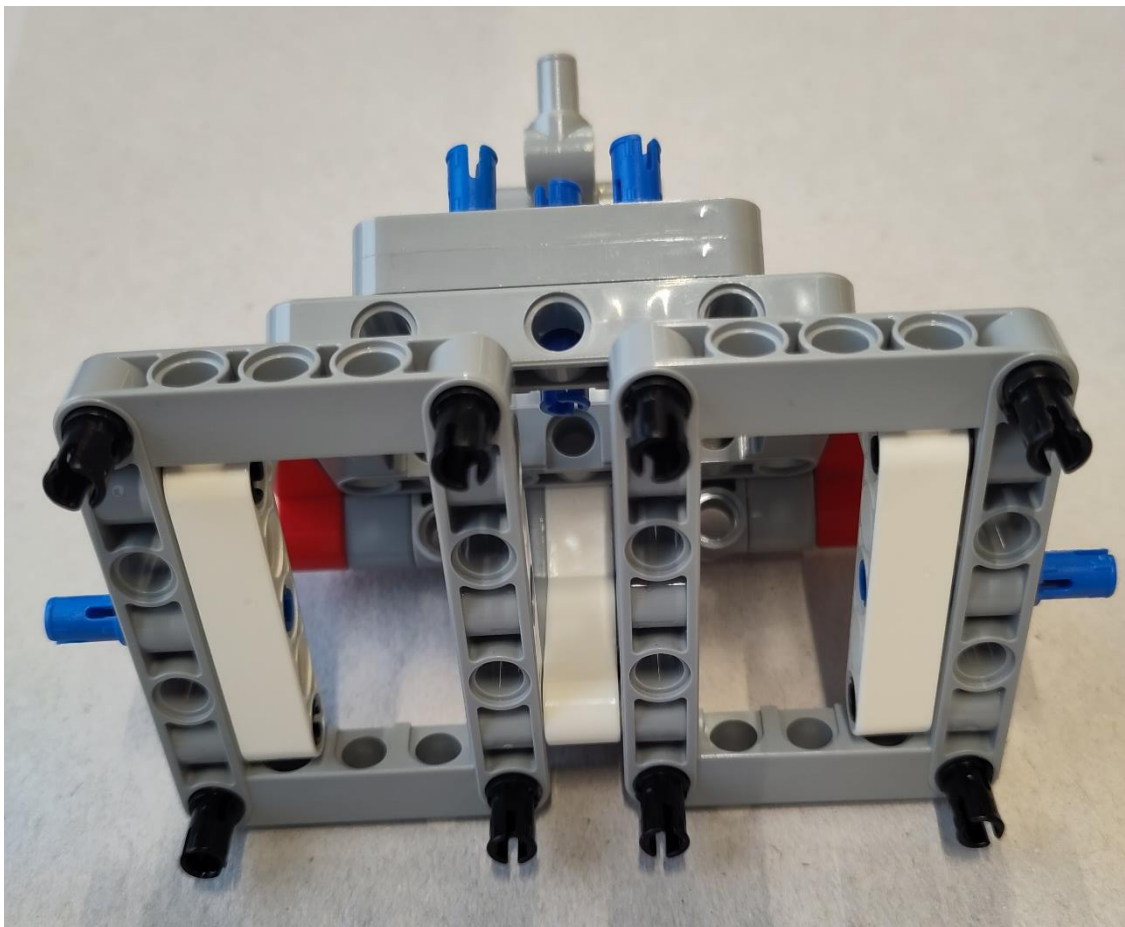


Abbildung 0-1. Verbindungselement Achse A mit Fingereinheit

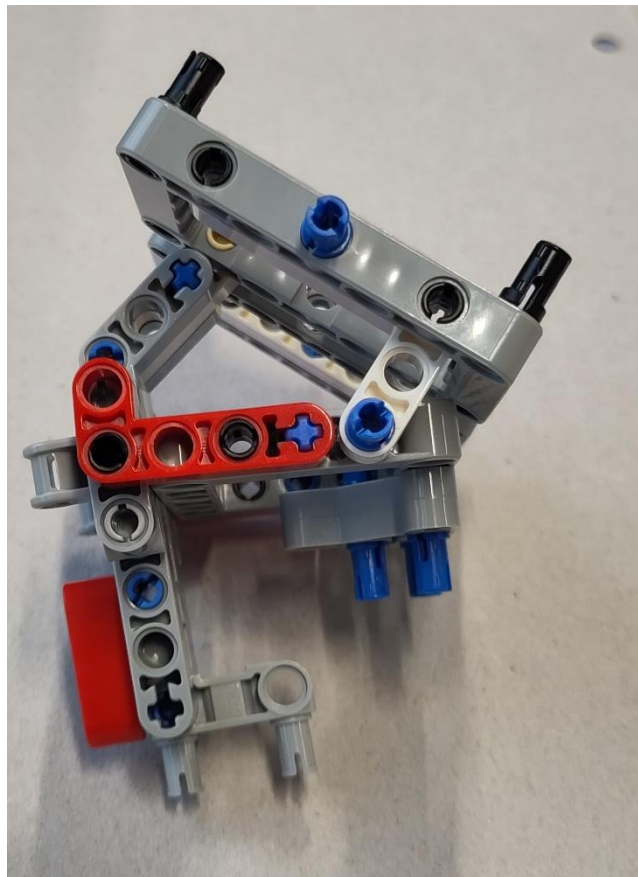


Abbildung 0-2 Verbindungselement Achse A mit Fingereinheit

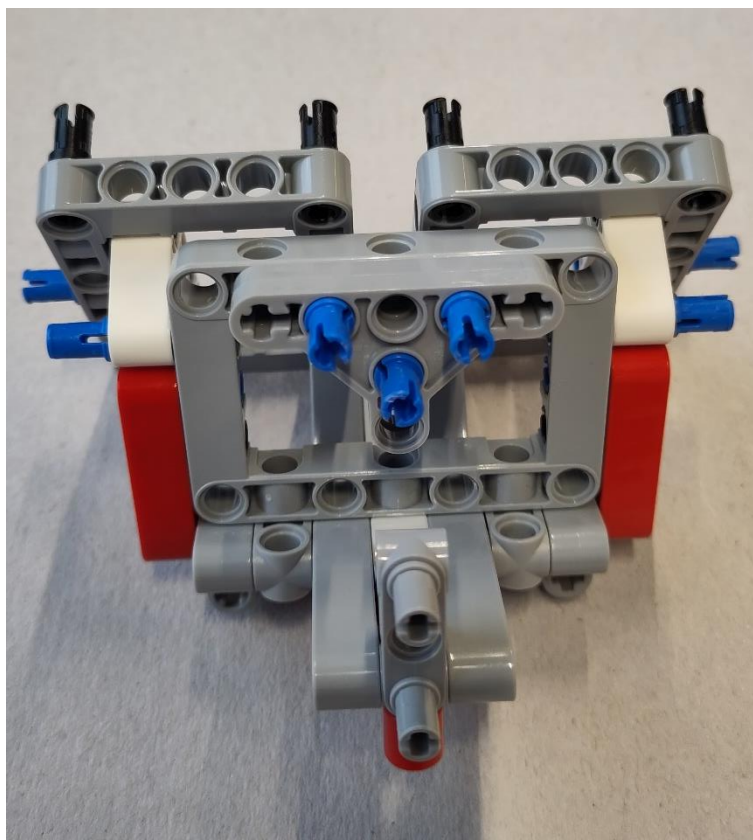


Abbildung 0-3 Verbindungselement Achse A mit Fingereinheit

## f) Montage vom Verbindungsmodul mit Achse A

Bezeichnung	Menge in Stück
Baugruppe Verbindungsmodul	1
Baugruppe Fundament mit Achse A	1

Verbinden Sie die Komponenten wie in der Abbildung 6-1 da gestellt.

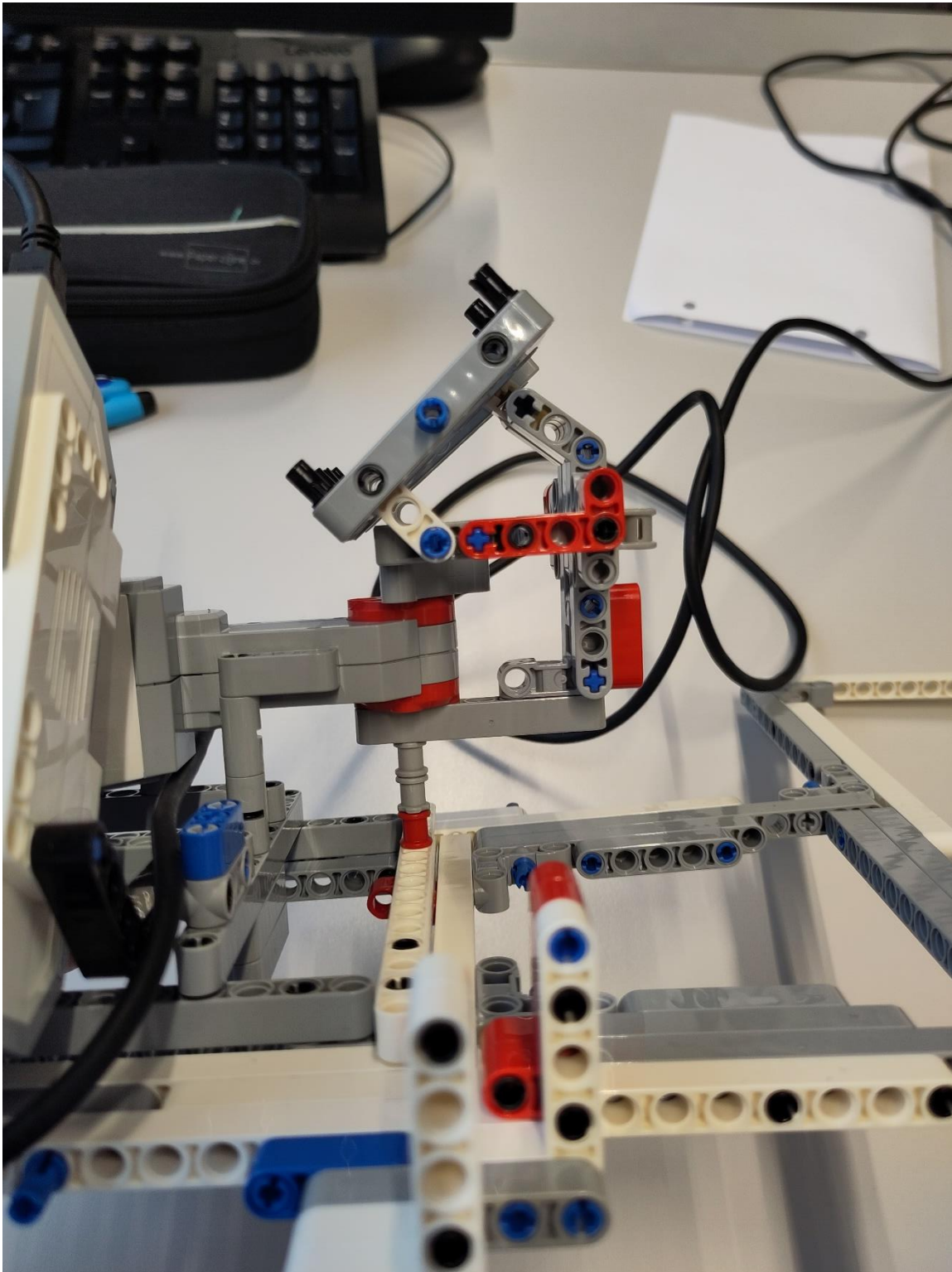


Abbildung 0-1 Montage vom Verbindungsmodul



## g) Vorkonfektion EV3

Bezeichnung	Menge in Stück
EV3 Modul	1
Graues 3 Loch-Modul + Kreuz	1
Graues Endstück	1
Blauer Kreuzstecker	3
Blaues Führung 3 Loch	1
Weisses 45° + 45° Modul	1
Schwarzes 45° Modul	2
Beiges Kreuz	1

Verbinden Sie die Komponenten wie in den Abbildungen 7-1 und 7-2 da gestellt.

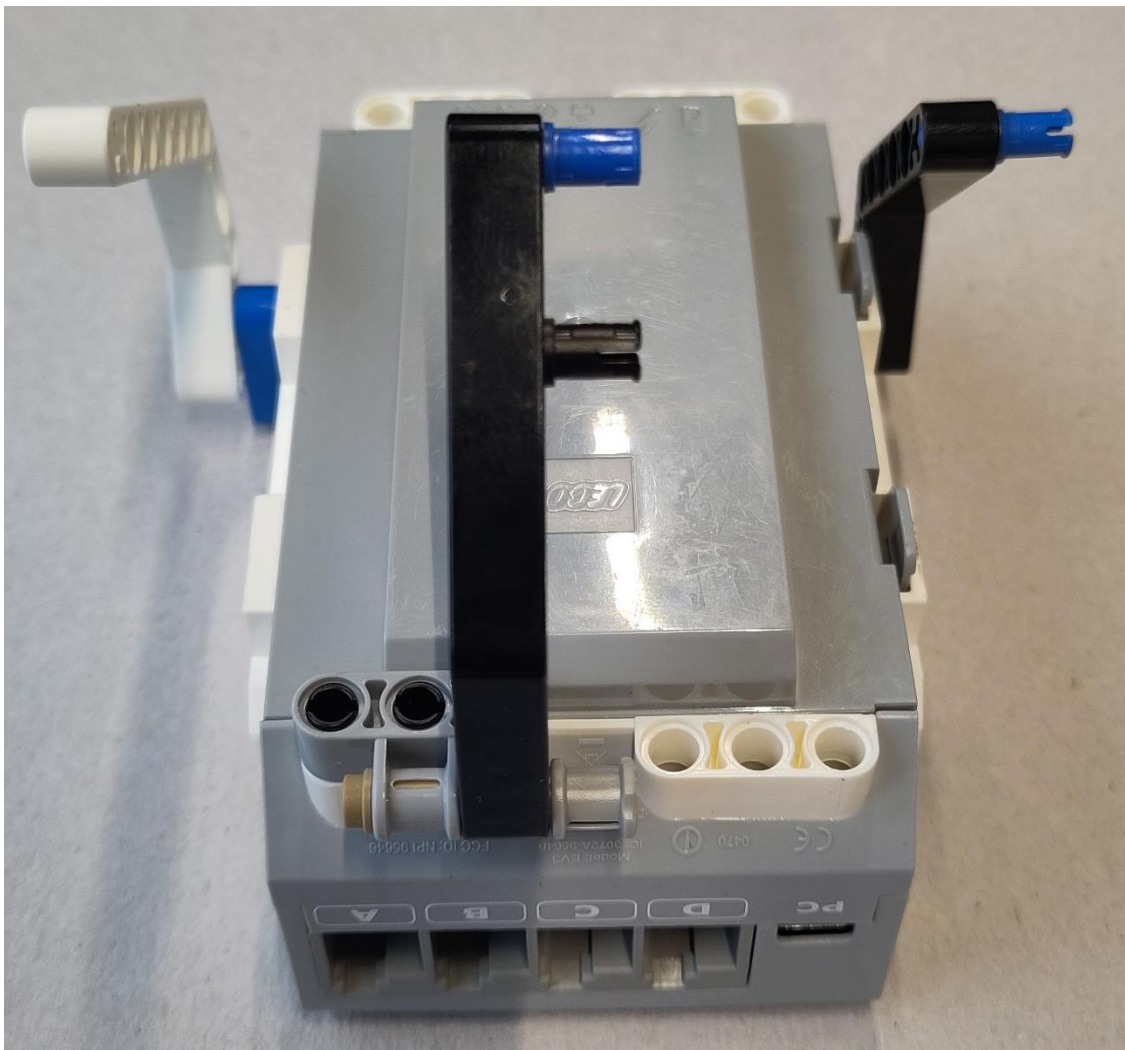


Abbildung 0-1 Vorkonfektion EV3

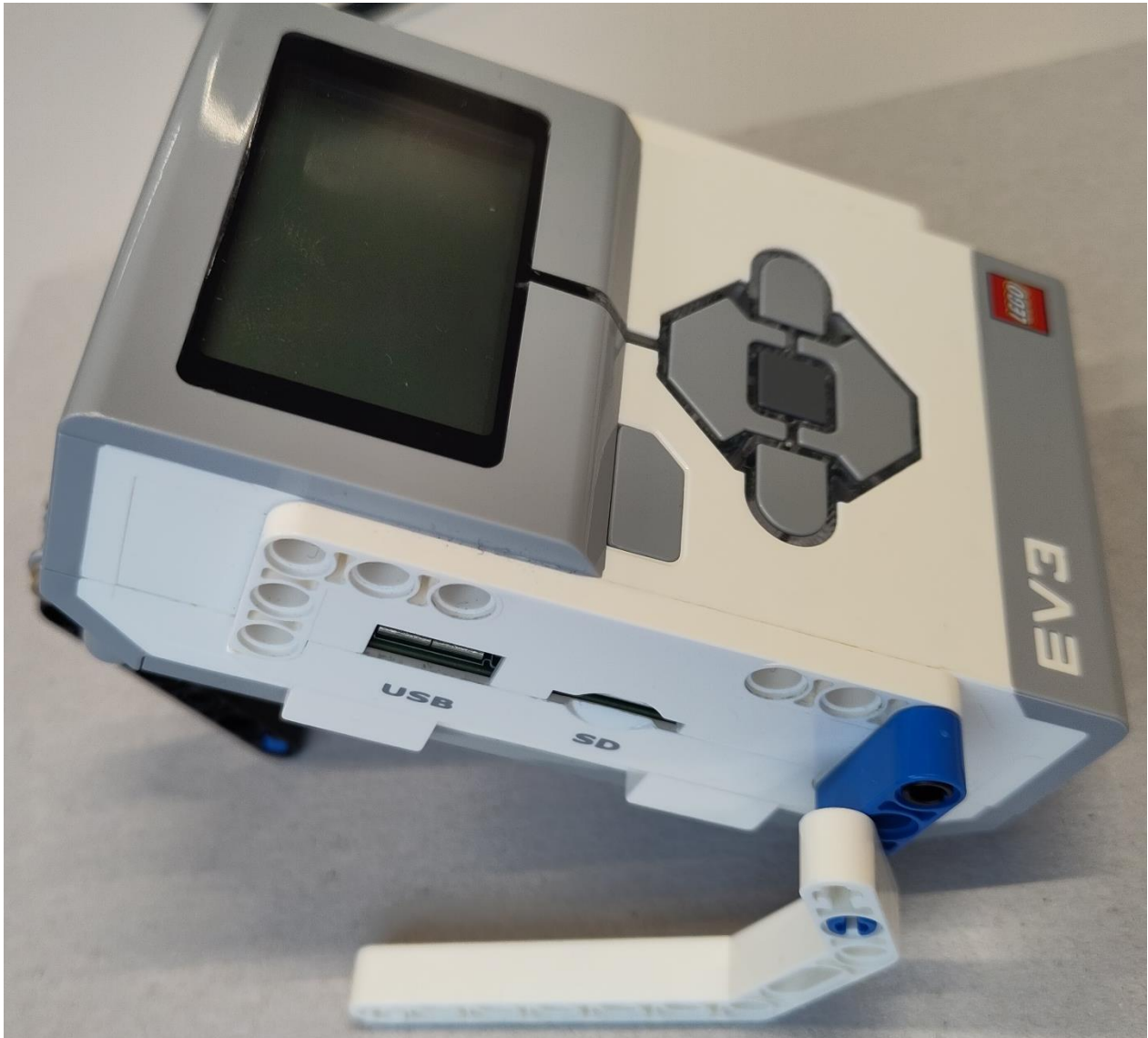


Abbildung 0-2 Vorkonfektion EV3

## h) Montage EV3 Modul an Fundament

Bezeichnung	Menge in Stück
Baugruppe EV3 2Vorkonfektion	1
Baugruppe Fundament	1

Verbinden Sie die Komponenten wie in den Abbildungen 8-1 und 8-2 da gestellt.

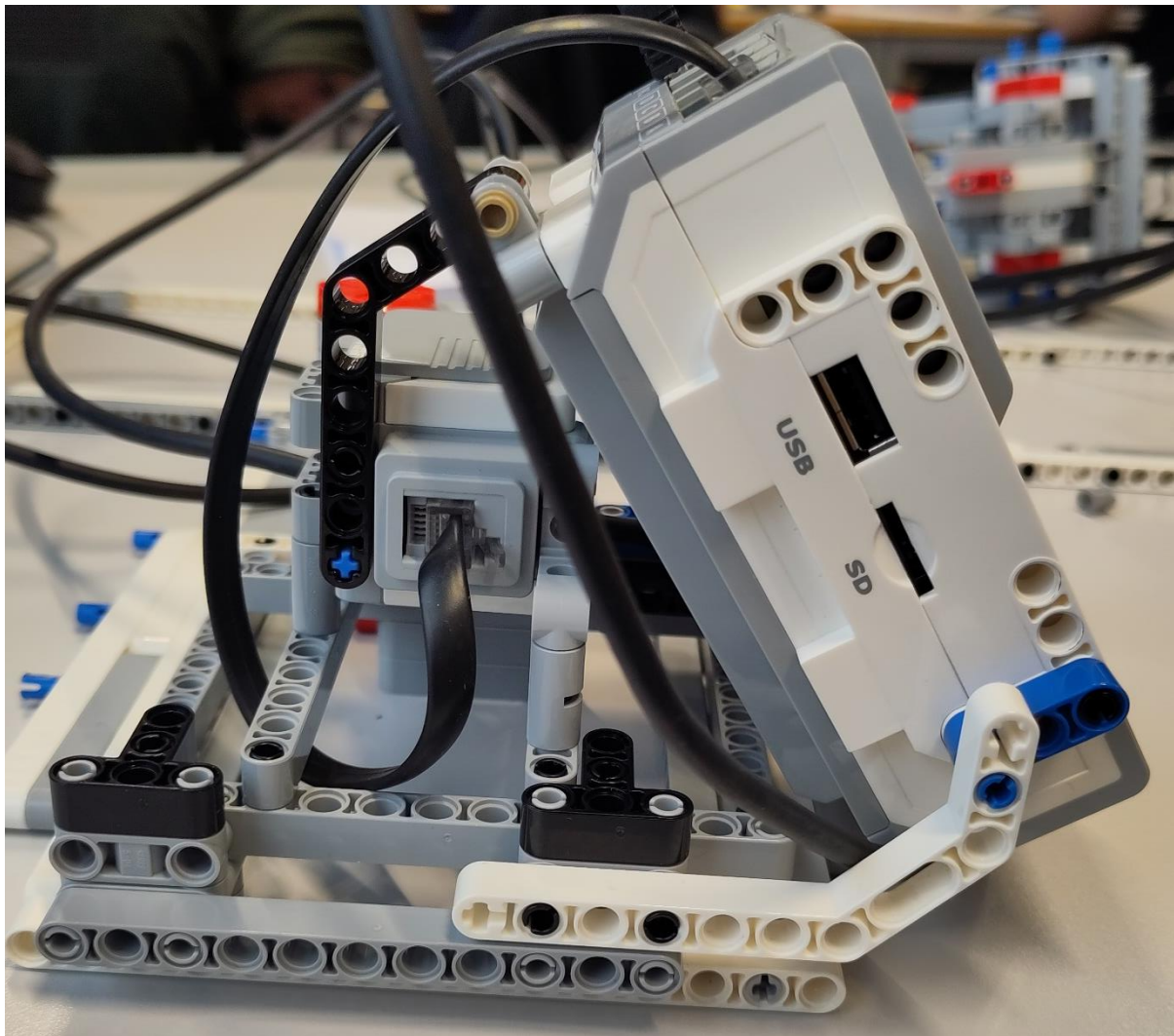
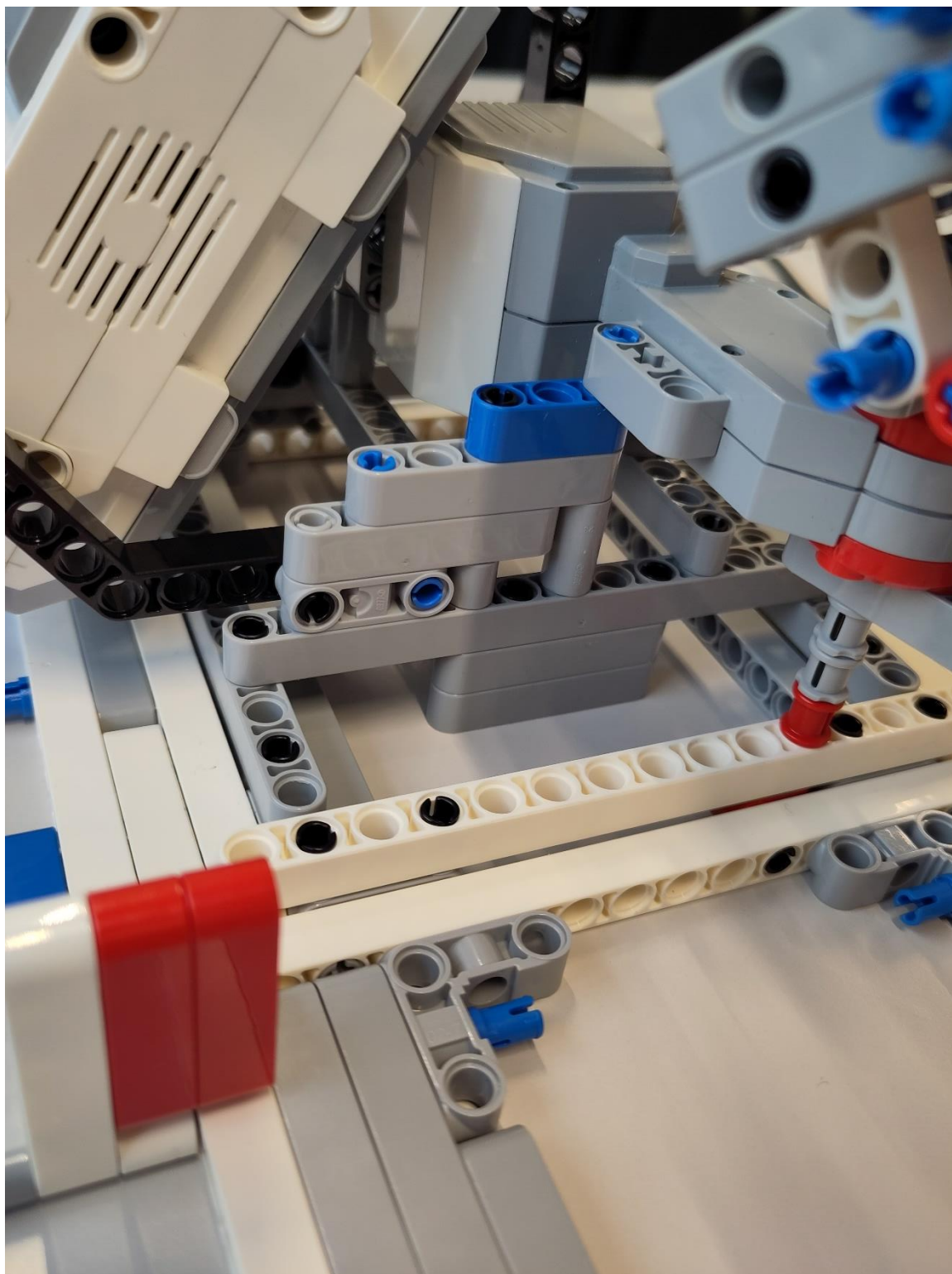


Abbildung 0-1 Montage EV3 Modul an Fundament





*Abbildung 0-2 Montage EV3 Modul an Fundament*

## i) Montage des Fingers mit Fundament

Bezeichnung	Menge in Stück
Baugruppe Fingereinheit	1
Baugruppe Fundament	1

Verbinden Sie die Komponenten wie in den Abbildungen 9-1 und 9-2 da gestellt.

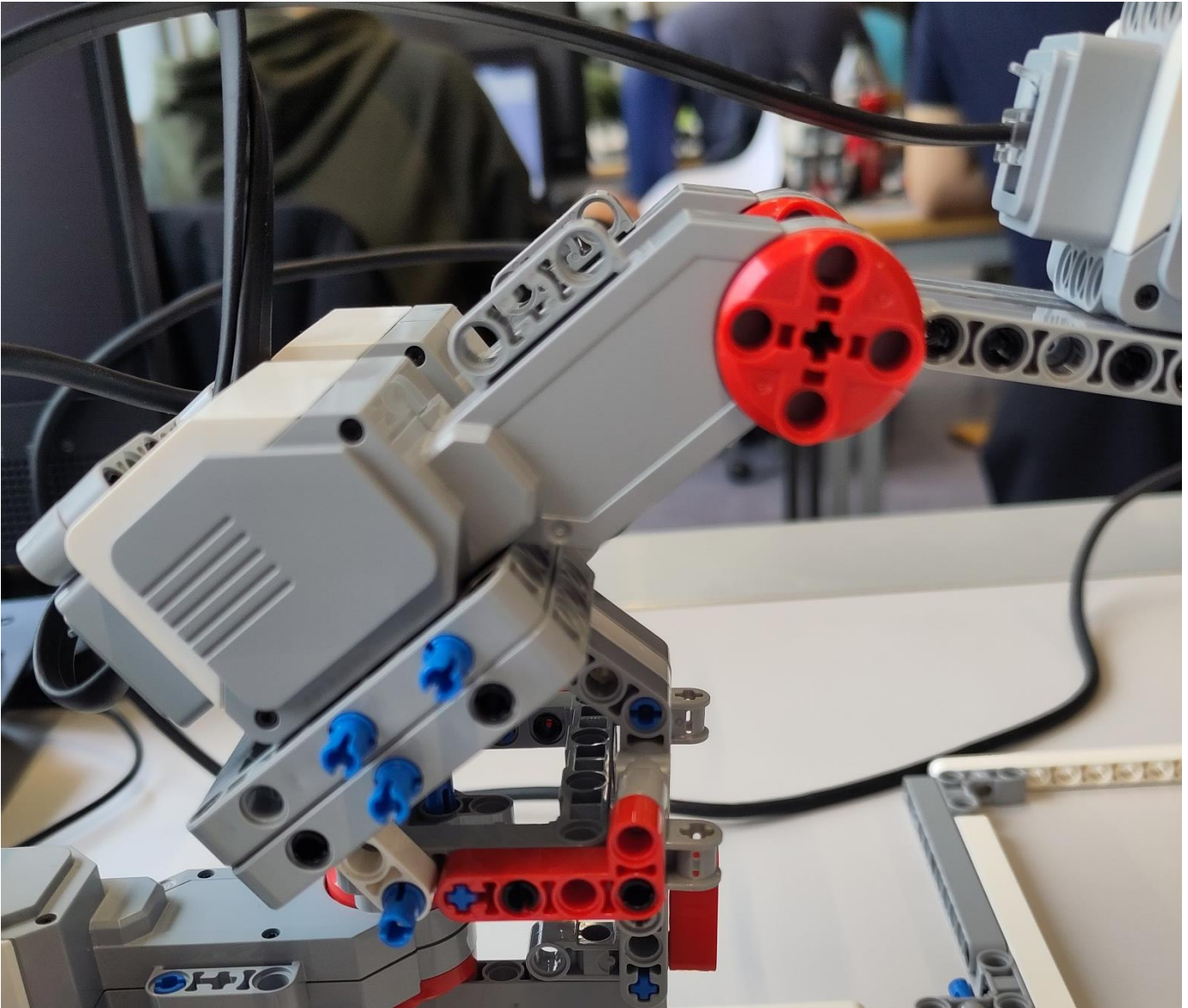


Abbildung 0-1 Montage des Fingers mit Fundament



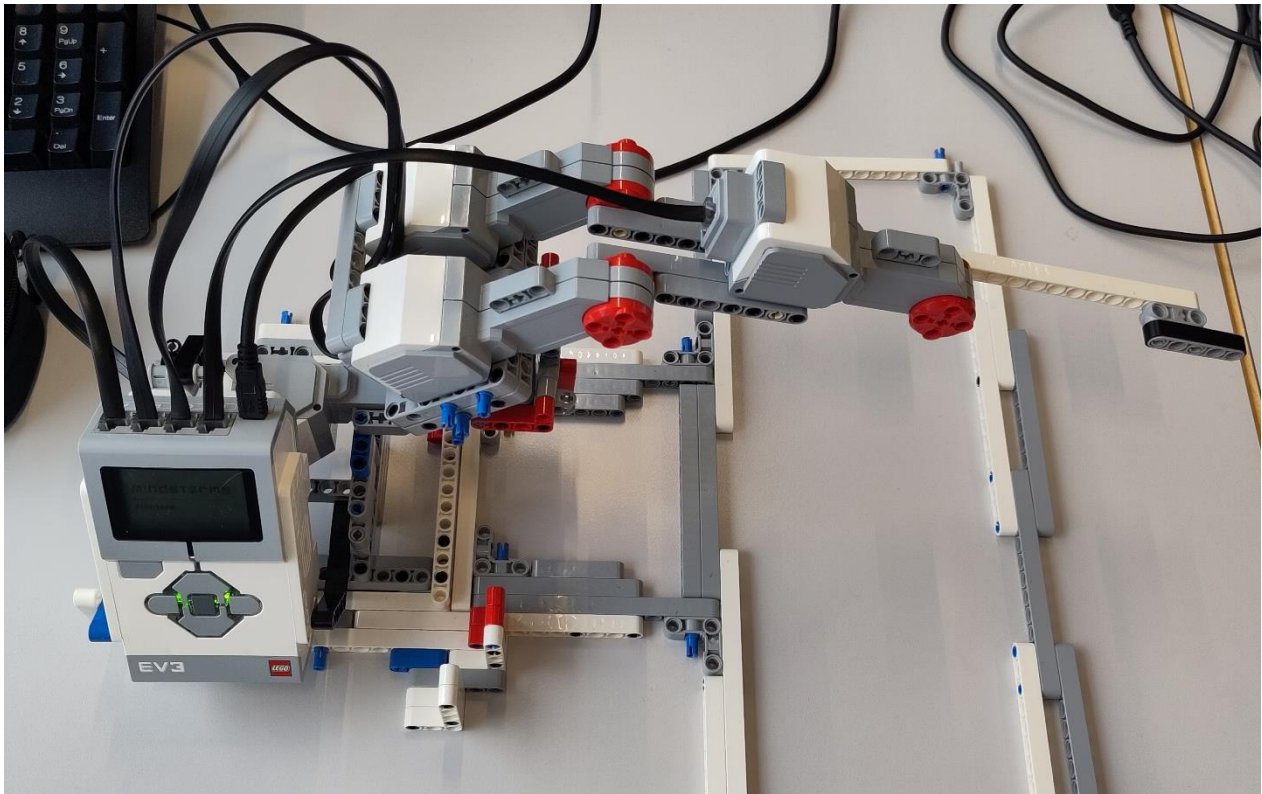
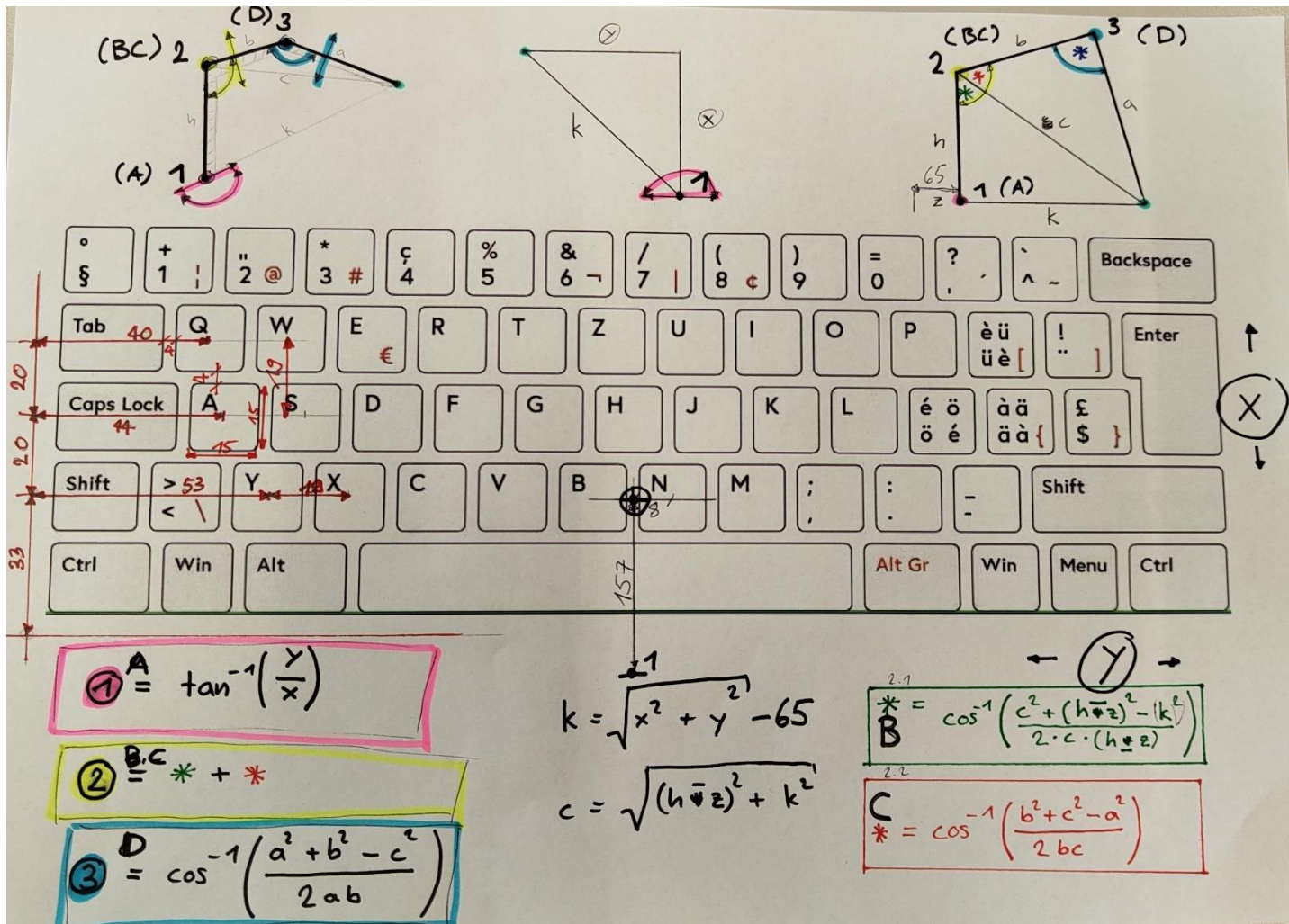


Abbildung 0-2 Montage des Fingers mit Fundament

Bitte beachten Sie, dass nach der Hauptmontage der Komponenten, die Verbindungsleitungen vom Controller zu den Motoren sauber verlegt werden sollen. Bei falscher Verlegung ist eine Beschädigung der Leitung möglich.

## 6. BERECHNUNGEN

Die Skizze welche im Brainstorming entstanden ist. Formeln zu den Berechnungen, welche im Code implementiert werden müssen:



The diagram shows a robotic arm with three joints: (A) at the base, (BC) at the elbow, and (D) at the end effector. The arm segments are labeled with lengths  $a$ ,  $b$ , and  $c$ . A coordinate system is established with origin at joint (A), where the x-axis is horizontal and the y-axis is vertical. The end effector (D) is at coordinates  $(x, y)$ . A keyboard layout is shown below the diagram, with dimensions (33, 20, 20, 40, 44, 53, 157) indicating the positions of the keys. A circled 'X' is marked on the right side of the keyboard.

Handwritten formulas for calculations:

- $\textcircled{1} A = \tan^{-1}\left(\frac{y}{x}\right)$
- $\textcircled{2} \underline{B, C} = * + *$
- $\textcircled{3} D = \cos^{-1}\left(\frac{a^2 + b^2 - c^2}{2ab}\right)$

Intermediate calculations for link lengths:

$$k = \sqrt{x^2 + y^2} - 65$$

$$c = \sqrt{(h - z)^2 + k^2}$$

Formulas for joint angles (B and C):

$$\textcircled{2.1} B = \cos^{-1}\left(\frac{c^2 + (h - z)^2 - k^2}{2 \cdot c \cdot (h - z)}\right)$$

$$\textcircled{2.2} C = \cos^{-1}\left(\frac{b^2 + c^2 - a^2}{2bc}\right)$$