

# Diplomarbeit

24. November 2023

## Gesamtprojekt RoboGlove - Bionische Hand

3D-Druck, Mechanik und User-Interface-Programmierung

Amir Al-Maytah    5BHEL

Betreuer: Dipl.-Ing. Christoph Diemberger

Fachlehrer: Robert Offner

Mikrokontroller-Programmierung, Testmanagement und Gesamtintegration

Fabian Schweitzer    5BHEL

Betreuer: Dipl.-Ing. Christoph Diemberger

Fachlehrer: Robert Offner

Hardwareentwicklung, PCB-Design und Projektleitung

Ladislaus Szabo    5BHEL

Betreuer: Dipl.-Ing. Christoph Diemberger

Fachlehrer: Robert Offner

Ausgeführt im Schuljahr 2023/24

## Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzen Quellen wörtlich und inhaltlich entnommenen Stellen als solche erkenntlich gemacht habe.



Amir Al-Maytah



Fabian Schweitzer



Ladislaus Szabo

## Danksagung

Wir möchten uns herzlich bei unserem Betreuer Dipl.-Ing. Christoph Diemberger bedanken, der uns bei diesem Projekt grundlegend unterstützt und motiviert hat.

Des Weiteren gilt unser Dank auch Fachlehrer Robert Offner und allen anderen Lehrpersonen, die uns in der Werkstatt betreut und geholfen haben.

Wir danken allen, die uns im Rahmen dieses Projekts zur Seite standen.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>5</b>
1.1	Ziel und Anwendungsbereich des Diplomprojekts . . . . .	5
1.2	Entwicklungsteam . . . . .	5
<b>2</b>	<b>Anforderungen</b>	<b>5</b>
2.1	Mechanik . . . . .	5
2.1.1	RoboGlove . . . . .	5
2.1.2	Bionische Hand . . . . .	5
2.2	Hardware . . . . .	5
2.2.1	RoboGlove . . . . .	5
2.2.2	Bionische Hand . . . . .	6
2.3	Software . . . . .	6
2.3.1	RoboGlove . . . . .	6
2.3.2	Bionische Hand . . . . .	6
2.3.3	User Interface . . . . .	7
<b>3</b>	<b>Beschreibung der Projektteile</b>	<b>7</b>
3.1	RoboGlove (Eingabe) . . . . .	7
3.2	Bionische Hand (Ausgabe) . . . . .	7
3.3	User Interface . . . . .	8
<b>4</b>	<b>Mechanische Entwicklung</b>	<b>8</b>
4.1	RoboGlove . . . . .	8
4.2	Bionische Hand . . . . .	8
<b>5</b>	<b>Hardwareentwicklung</b>	<b>8</b>
5.1	RoboGlove . . . . .	8
5.1.1	Überlegungen und Versuchsaufbauten . . . . .	8
5.1.2	Simulationen und Dimensionierung . . . . .	9
5.1.3	Testaufbauten und Schaltungsdesign . . . . .	10
5.1.4	Platinendesign . . . . .	10
5.2	Bionische Hand . . . . .	10
5.2.1	Überlegungen und Versuchsaufbauten . . . . .	10
5.2.2	Simulationen und Dimensionierung . . . . .	10
5.2.3	Testaufbauten und Schaltungsdesign . . . . .	10
5.2.4	Platinendesign . . . . .	10
<b>6</b>	<b>Softwareentwicklung</b>	<b>10</b>
6.1	RoboGlove . . . . .	10
6.1.1	Konzept und Überlegungen . . . . .	10
6.1.2	Realisierung . . . . .	10
6.2	Bionische Hand . . . . .	10
6.2.1	Konept und Überlegungen . . . . .	10
6.2.2	Realisierung . . . . .	10
6.3	User Interface . . . . .	10
<b>7</b>	<b>Testphase</b>	<b>10</b>

<b>8 Abschluss und Zusammenfassung</b>	<b>10</b>
<b>9 Anhang</b>	<b>10</b>

## **1 Einleitung**

### **1.1 Ziel und Anwendungsbereich des Diplomprojekts**

Das Ziel des Diplomprojekts ist es, eine bionische Hand zu bauen, die mit einem drahtlos verbundenen Handschuh gesteuert werden kann. Der Benutzer trägt den Handschuh und kann somit Objekte, mit denen er nicht in Berührung kommen darf oder will, greifen. Folglich imitiert die bionische Hand die Bewegungen des Benutzers möglichst präzise und führt somit die gewollte Arbeit des Benutzers aus. Anwendungsbereiche für diese Technologie wären zum Beispiel elektronische Unternehmen, die klinisch sauber arbeiten müssen, um ihre Produkte korrekt fertigen zu können. Ein weiteres Beispiel wäre die Anwendung bei Tätigkeiten, die mit radioaktiven, beziehungsweise giftigen Substanzen zu tun haben. Hierbei könnte aus einer sicheren Entfernung gearbeitet werden, ohne Personen zu gefährden.

### **1.2 Entwicklungsteam**

Folgende Schüler des TGMs haben sich mit der Entwicklung des in Punkt 1.1 beschriebenen Projekts befasst:

Projektleiter: Ladislaus Szabo (Hardwareentwicklung, PCB-Design, Projektleitung)

Mitarbeiter: Fabian Schweitzer (Mikrokontroller-Programmierung, Testmanagement, Gesamtintegration)

Amir Al-Maytah (3D-Druck, Mechanik, Userinterface-Programmierung)

## **2 Anforderungen**

### **2.1 Mechanik**

#### **2.1.1 RoboGlove**

#### **2.1.2 Bionische Hand**

- Die Roboterhand/der Roboterarm soll mittels 3D-Druck gefertigt werden.
- Die Servomotoren der Elektronik werden mithilfe von Schnüren mit den jeweiligen Fingern verbunden.
- Die Finger sollen sich sowohl zum Handballen als auch vom Handballen weg kontrolliert bewegen können.
- Die Finger sollen sich zitterfrei bewegen können.

### **2.2 Hardware**

#### **2.2.1 RoboGlove**

Folgende Anforderungen muss der „RoboGlove“ nach Abschluss des Projekts erfüllen:

- Die Bewegung der Finger muss mit Sensoren erfasst werden.
- Die Sensordaten müssen mit einem Mikrokontroller ausgewertet werden können.
- Der zur Datenerfassung gewählte Mikrokontroller muss:

Die I2C Kommunikation unterstützen.

Die Möglichkeit für eine kabellose Kommunikation aufweisen.

Analoge und Digitale Eingänge haben.

Programmierbar sein.

- Es muss möglich sein, mindestens 30 verschiedene Sensorwerte, von minimaler bis maximaler Fingerbeugung, zu erfassen.

- Der RoboGlove muss die Möglichkeit zur Versorgung mit einem Akku oder einer Batterie und einem Netzteil aufweisen.

### 2.2.2 Bionische Hand

Folgende Anforderungen muss die „Bionic Hand“ nach Abschluss des Projekts erfüllen:

- Jeder Finger der „Bionic Hand“ soll durch einen eigenen Motor gesteuert werden.
- Jeder dieser Motoren muss individuell regelbar sein, um die Griffkraft der Hand zu kontrollieren.
- Jeder Finger soll sich zum Handballen beugen und wieder strecken lassen.
- Diese Bewegungen sollen störungsfrei und ohne zittern erfolgen.

## 2.3 Software

### 2.3.1 RoboGlove

Folgende Anforderungen müssen für die Software für die Handschuhplatine nach Abschluss des Projekts erfüllt sein:

- Der Widerstand jedes Flexsensors, der an jedem Handschuh jeweils über jedem Finger angebracht ist, soll einzeln ausgelesen werden können. Die Software soll damit feststellen, wie sehr der Flexsensor gebogen ist.

- Diese Daten sollen erfolgreich eingelesen werden und direkt an die Empfängerplatine gesendet werden.

### 2.3.2 Bionische Hand

Folgende Anforderungen müssen für die Software für die Roboterhandplatine nach Abschluss des Projekts erfüllt sein:

- Die von der Handschuhplatine gesendeten Daten sollen erfolgreich empfangen und gespeichert werden.

- Eine Fehlerkorrektur sorgt dafür, dass die empfangenen Daten richtig verarbeitet werden können.

- Die Software muss bei jedem Finger etwas anders agieren, da die Widerstandswerte der verschiedenen Flexsensoren stark variieren. Es soll bei jedem Finger eine gleich gute Steuerung der Roboterhand möglich sein.

- Es muss immer wieder (alle 10ms) der Strom gemessen werden. Mit einem Timing Interrupt muss also zwischen dem Bewegen der Finger immer wieder dieser gemessen werden.

- Es sollen unterschiedliche Modi erstellt werden, je nachdem, was mit der Roboterhand gehalten werden soll. Die Griffkraft soll je nach Objekt variieren.

### 2.3.3 User Interface

## 3 Beschreibung der Projektteile

### 3.1 RoboGlove (Eingabe)

Der „RoboGlove“ ist das Hauptaugenmerk des Projekts. Er ermöglicht dem Benutzer die „Bionic Hand“ zu steuern. Dies geschieht durch Flexsensoren, die an den Fingern des „RoboGlove“ angebracht sind. Wird ein Finger gebeugt, so ändert sich der Widerstand des Sensors. Diese Änderungen werden von einem Mikrokontroller erfasst, ausgewertet und für die kabellose Übertragung an die „Bionic Hand“ vorbereitet. Damit der „RoboGlove“ funktioniert, bedarf es einer Spannungsversorgung. Diese wird in Form einer Batterie oder eines Akkus vorgesehen sein. Für einen möglichen stationären Betrieb kann der „RoboGlove“ auch mit einem Netzteil betrieben werden. Die Passform des „RoboGlove“ soll möglichst komfortabel sein, um eine möglichst lange Benützung zu ermöglichen. Dies setzt eine gut überlegte Integration der Elektronik voraus, die bei diesem Projekt auch bestmöglich beachtet wurde. Der „RoboGlove“ ist so gebaut, dass er auch von Personen ohne elektronische Ausbildung und ohne Vorwissen bedient werden kann.

**Folgende Richtlinien und Normen werden von dem Produkt „RoboGlove“ erfüllt:**

Die Software des Roboterhandschuhs ist dafür verantwortlich die sich verändernden Widerstandswerte der Flexsensoren jeweils einzeln einzulesen. Über die UART-Schnittstelle wird die Software auf den ESP32 hochgeladen. Dabei werden direkt aus dem in der Schaltung verbauten ADC-Werte, über die I2C-Schnittstelle, ausgelesen. Man erhält bereits digitale Werte, die der ESP32 dann direkt in einer festgelegten Reihenfolge an die Empfängerplatine, bei der Roboterhand, per ESP-NOW Übertragung senden kann.

### 3.2 Bionische Hand (Ausgabe)

Die „Bionic Hand“ ist die Ausgabe des Projekts und wird vom „RoboGlove“ gesteuert. Jeder Finger wird von einem Servomotor bewegt. Dieser ist durch zwei dünne Schnüre mit dem korrespondierenden Finger verbunden. Die vom „RoboGlove“ kommenden Daten werden von einem Mikrokontroller ausgewertet und in ein PWM-Signal (Pulsweitenmodulation) zur Ansteuerung der Servomotoren umgewandelt. Jeder Finger hat drei Gelenke und kann daher kontrolliert gebeugt und wieder gestreckt werden. Die komplette „Bionic Hand“ wird mittels 3D-Druck gefertigt. Silikonprofile auf den Fingern versichern das Objekte nicht mehr aus dem Griff der Hand rutschen können.

**Folgende Richtlinien und Normen werden von dem Produkt Bionische Hand erfüllt:**

Die Software des Roboterarmes ist dafür verantwortlich, dass die von der Handschuhplatine, via ESP32, gesendeten Daten erfolgreich empfangen und gespeichert werden. Die Software wird hierzu per UART-Schnittstelle auf den ESP32 hochgeladen. Die Daten der Flexsensoren, die man bereits erhalten hat, sollen dann verarbeitet werden. Anhand des Widerstandswertes und der jeweiligen Veränderung, wird dann der Winkel des Servos eingestellt. Eine Fehlerkorrektur soll sicherstellen, dass die Griffkraft passend eingestellt wird. Jeder Flexsensor hat unterschiedliche Widerstandswerte, es muss also über jeden Finger eine gezielte Software geschrieben werden, damit man schlussendlich durch das Zusammenspiel aller Finger die Griffkraft richtig einstellen kann. Falsche Messungen sollen bestmöglich korrigiert werden, sodass die Servos jeweils die richtigen Winkeleinstellungen erhalten. Der ESP32 stellt hierbei die Zentrale der Steuerung dar.

### 3.3 User Interface

## 4 Mechanische Entwicklung

### 4.1 RoboGlove

### 4.2 Bionische Hand

## 5 Hardwareentwicklung

### 5.1 RoboGlove

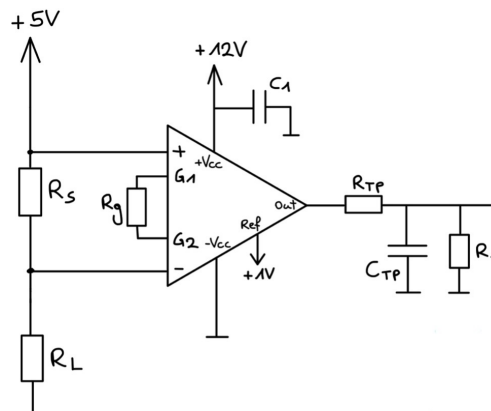
#### 5.1.1 Überlegungen und Versuchsaufbauten

Sensorik:

Es muss in irgendeiner Weise die Fingerbeugung, beziehungsweise Bewegung, des Benutzers mit dem „RoboGlove“ gemessen werden können. Dies könnte man durch das Messen des Beugungswinkels realisieren. Allerdings hat jeder Finger drei Gelenke, wodurch man diese auch bei der „Bionic Hand“ individuell steuern müsste. Die zweite Möglichkeit wäre, durch eine visuelle Aufnahme die Bewegung des „RoboGlove“ und dadurch des Benutzers aufzuzeichnen. Da dies allerdings nur in dafür vorgesehenen, mit Kameras ausgestatteten, Räumen funktionieren würde, ist dies für uns auch keine sinnvolle Möglichkeit. Schließlich haben wir uns für die Erfassung der Fingerbewegungen mittels Flexsensoren entschieden. Diese ändern den Widerstand je nach der aktuell vorherrschenden Beugung. Bei dieser Art der Bewegungserfassung muss man nicht jedes Fingergelenk einzeln steuern und braucht auch keine externen Kameras. Somit ist bei dieser Methode der Datenerfassung ein sehr flexibler Verwendungsbereich des „RoboGlove“ gewährleistet.

Auslesen der Sensoren:

Das Auslesen der Flexsensoren kann durch einen einfachen Spannungsteiler erfolgen. Dabei ist die Genauigkeit (Auflösung) allerdings nicht gegeben und ist daher nicht für unsere Anwendung geeignet. Die Lösung für dieses Genauigkeitsproblem ist eine ADC-Schaltung mit einer davor geschalteten OPV-Schaltung.



Der Flexsensor ( $R_L$ ) bezieht seine Versorgung über einen Shunt-Widerstand ( $R_S$ ). Je nach Belastung, ändert sich der Spannungsabfall an diesem (Je größer die Beugung des Sensors, desto kleiner ist der Spannungsabfall). Die Spannungsdifferenz am Shunt-Widerstand wird von einem Operationsverstärker verstärkt. Ein Tiefpassfilter ( $R_{TP}$  und  $C_{TP}$ ) ist hinter den Ausgang des OPV



geschaltet, um mögliche Spannungsstörungen (Ripple) herauszufiltern. Der zu GND geschaltete Widerstand (R1), entlastet den Eingang des angeschlossenen ADCs. Dieser Analog-Digital-Wandler digitalisiert das analoge Eingangssignal und überträgt diese an den verbauten Mikrokontroller. Da die OPV-Messschaltung verhältnismäßig viel Platz in Anspruch nimmt und diese fünfmal, für jeden Flexsensor einmal, benötigt wird, wäre es von großem Vorteil alle Sensoren mit nur einer Schaltung auszumessen. Hierfür verwenden wir einen Multiplexer, der in Bruchteilen einer Sekunde zwischen allen Biegemessstreifen durchschaltet. Dies ermöglicht das Auslesen aller Sensoren mit nur einer Verstärkerschaltung und trägt wesentlich zur möglichst kompakten Dimensionierung der später folgenden Platine bei.

Mikrokontroller:

Der zur Auswertung der Messdaten zuständige Mikrokontroller muss, die in Punkt 2.2.1 angeführten Anforderungen, erfüllen. Hierzu haben wir uns für den ESP32 entschieden, da dieser mehr als genug Rechenleistung bietet, mehrere drahtlose Übertragungsmöglichkeiten unterstützt und die Möglichkeit einer Verbindung mit dem Internet bietet. Der kleine Formfaktor, trotz den vielen IO-Ports, ist nur ein weiteres Argument für die Wahl dieses Mikrokontrollers. Um den ESP32 programmieren und steuern zu können, sind zwei Taster, einer für das RESET und einer für den UPLOAD, notwendig. Außerdem ist ein Mikro-USB-Anschluss notwendig, um den in einer Entwicklungsumgebung erstellten Code hochzuladen. Da man die USB-Buchse nicht direkt mit dem ESP32 verbinden kann, ist zusätzlich noch ein Bustreiber notwendig, der ein am Mikrokontroller vorhandenes Interface unterstützt.

### 5.1.2 Simulationen und Dimensionierung

**Bild der Spannungs-Simulation hier einfügen!!**

Der Widerstand der Flexsensoren ändert sich bei zunehmender Beugung von 25kOHM auf 125kOHM.

Hier das Messprotokoll der Flexsensoren einfügen!!

Die Flexsensoren werden mit +5V Gleichspannung versorgt. Zwischen der Versorgung und den Sensoren ist ein Shunt-Widerstand (RS) geschaltet, an dem je nach Beugung der Sensoren eine gewisse Spannung abfällt. Dies ist der Fall, da bei einem geringeren Widerstand des Flexsensors ein größerer Strom über den Widerstand RS fließt und somit der Spannungsabfall größer ist. Bei großen (niederohmigen) Lasten ist es sehr wichtig dabei die Verlustleistung am Shunt zu beachten, allerdings ist dies bei Flexsensoren kein Thema, da diese nur einen sehr kleinen Strom benötigen.

Um nun die korrekte Verstärkung des OPVs und den korrekten Shunt-Widerstand zu wählen ist es wichtig zu wissen, dass wir einen 10-Bit ADC mit +3.3V Versorgungsspannung verwenden. Mit dieser Information haben wir nun den Shunt und den Widerstand Rg dimensioniert. Die Ausgangsspannung des Verstärkers bei maximaler Last des Flexsensors sollte knapp unter +3.3V sein, um die Auflösung des ADCs maximal auszunützen. Zuerst haben wir die Verstärkung  $G=78$  gewählt, da bei einem höheren Gain Störungen auftreten könnten. Dafür wurde aus dem Datenblatt des OPVs der Widerstand Rg mit 1.1kOhm gewählt. Mit dieser Information haben wir dann den Shunt Rs mit 240OHM dimensioniert. Da bei größeren Widerständen des Flexsensors der Spannungsabfall allerdings so gering ist, dass der Operationsverstärker diesen nicht mehr exakt verstärken kann, ist es notwendig eine +1V Referenzspannung am OPV anzulegen. Durch diese kann auch bei sehr kleinen Spannungsabfällen ein exakt verstärktes Ausgangssignal generiert werden.

**Dimensionierung Spannungsteiler für +1V hier einfügen!!**

Durch den Tiefpass am Ausgang werden mögliche Spannungsripple der Versorgung gefiltert. Durch den Widerstand Rtp wird ein kleiner Spannungsabfall erzeugt, der bei der Messung zwar berücksichtigt wird, allerdings kein Problem darstellt. Die Grenzfrequenz des Filters wurde folgender

Maßen berechnet:

Berechnung fg hier einfügen!!

Durch den zu GND geschalteten Widerstand R1 mit 10kOHM, wird er Eingang des danach kommenden ADCs entlastet.

### **5.1.3 Testaufbauten und Schaltungsdesign**

### **5.1.4 Platinendesign**

## **5.2 Bionische Hand**

### **5.2.1 Überlegungen und Versuchsaufbauten**

### **5.2.2 Simulationen und Dimensionierung**

### **5.2.3 Testaufbauten und Schaltungsdesign**

### **5.2.4 Platinendesign**

## **6 Softwareentwicklung**

### **6.1 RoboGlove**

#### **6.1.1 Konzept und Überlegungen**

#### **6.1.2 Realisierung**

### **6.2 Bionische Hand**

#### **6.2.1 Konept und Überlegungen**

#### **6.2.2 Realisierung**

### **6.3 User Interface**

## **7 Testphase**

## **8 Abschluss und Zusammenfassung**

## **9 Anhang**