

Diplomarbeit

25. Januar 2024

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
Ladislav 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

1	Einleitung	5
1.1	Kurzfassung	5
1.2	Abstract	5
1.3	Ausgangslage	5
1.4	Untersuchungsanliegen der individuellen Themenstellungen	5
1.5	Allgemeine Zielsetzung	6
1.6	Terminplan (Milestones)	6
1.7	Geplantes Ergebnis der Prüfungskandidaten	6
1.8	Entwicklungsteam	6
2	Anforderungen der einzelnen Projektteile	7
2.1	Mechanik	7
2.1.1	Handschuh	7
2.1.2	Roboterhand	7
2.2	Elektronische Hardware	7
2.3	Handschuh	7
2.3.1	Roboterhand	7
2.4	Software	8
2.4.1	Handschuh	8
2.4.2	Roboterhand	8
2.4.3	Bionische Hand	8
2.4.4	User Interface	8
3	Detaillierte Beschreibung der Projektteile	8
3.1	Handschuh (Eingabe)	8
3.2	Roboterhand (Ausgabe)	9
3.3	User Interface	10
4	Mechanische Entwicklung	10
4.1	Handschuh	10
4.2	Roboterhand	10
5	Hardwareentwicklung	10
5.1	Handschuh	10
5.1.1	Simulationen und Versuchsaufbauten	10
5.1.2	Überlegungen und Berechnungen	11
5.1.3	Schaltplandesign	13
5.1.4	Platinendesign	13
5.2	Roboterhand	13
5.2.1	Simulationen und Versuchsaufbauten	13
5.2.2	Überlegungen und Dimensionierung	13
5.2.3	Schaltplandesign	13
5.2.4	Platinendesign	13

6 Softwareentwicklung	13
6.1 Handschuh	13
6.1.1 Konzepte und Überlegungen	13
6.1.2 Realisierung und Gliederung	13
6.2 Roboterhand	13
6.2.1 Konzepte und Überlegungen	13
6.2.2 Realisierung und Gliederung	13
6.3 User Interface	13
6.3.1 Entwicklungsumgebung	13
6.3.2 Konzepte und Überlegungen	13
6.3.3 Realisierung und Gliederung	13
7 Abschluss und Zusammenfassung	13
8 Anhang	13
9 Quellen -und Literaturverzeichnis	13
10 Abbildungsverzeichnis	13

1 Einleitung

1.1 Kurzfassung

Aufgabenstellung: Im Rahmen eines Diplomprojekts soll eine Roboterhand mittels eines Handschuhs gesteuert werden. Dazu müssen verschiedene Sensoren auf Genauigkeit getestet und anschließend die ausgewerteten Daten kabellos an die Roboterhand übertragen werden. Die Bewegungen der Finger sollen mit Motoren nachgebildet werden und die Stellungen dieser soll im User-Interface dargestellt werden.

Realisierung: Die Bewegungen der menschlichen Hand werden mit Flexsensoren am Handschuh erfasst. Die benötigten Daten werden anschließend über ein drahtloses Protokoll an die Roboterhand übertragen. Die Bewegungen der Roboterfinger werden mit Servomotoren realisiert. Als User-Interface soll eine Application dienen.

Ergebnis: Die Handbewegungen können korrekt ausgewertet und übertragen werden. Der Benutzer trägt den Handschuh und greift ein Objekt. Die Servomotoren interpretieren mittels des PCBs die empfangenen Daten und ermöglichen es der Roboterhand ebenfalls das gleiche Objekt zu greifen. Beim Öffnen der Hand muss sich die Roboterhand in ihre Ausgangsstellung zurückbewegen.

1.2 Abstract

Task: As part of a feasibility study, a robotic hand is to be controlled by means of a glove. For this purpose, various sensors must be tested for accuracy and the evaluated data then transmitted wirelessly to the robotic hand. The positions of the fingers are to be displayed in a user interface.

Realization: The movements of the human hand are recorded with Flex sensors on the glove. The required data is then transmitted to the robotic hand via Bluetooth. A website will serve as the user interface.

Result: The hand movements can be correctly evaluated and transmitted. The user wears the glove and grasps an object. The servo motors interpret the received data by means of the PCB and enable the robot hand to also grasp the same object. When the hand is opened, the robot hand must also move back to its starting position.

1.3 Ausgangslage

Es gibt Produktionsbereiche, die klinisch sauber gehalten werden müssen, da durch Menschen Kontaminationen entstehen können. Für dieses Vorhaben soll ein Prototyp einer Roboterhand, die über einen Sensorhandschuh kabellos gesteuert wird, entwickelt und aufgebaut werden. Diverse Parameter der Roboterhand sollen in einem Interface für den Benutzer dargestellt werden.

1.4 Untersuchungsanliegen der individuellen Themenstellungen

Mit dieser Diplomarbeit soll eine Roboterhand nach einem fertigen Design aufgebaut werden, die mit einem kabellos angebundenen Handschuh gesteuert wird. Zu diesem Zweck müssen verschiedene Sensoren getestet werden, die die Fingergelenksstellung messen. Deren Daten müssen ausgelesen und mittels kabelloser Schnittstelle an die Roboterhand übertragen werden (Schweitzer). Um die Fingerbewegungen an der Roboterhand nachzustellen, muss folglich eine Motoransteuerung entwickelt werden (Szabo). Um die elektronischen Bauelemente unterzubringen, muss ein entsprechendes Gehäuse, in Form einer menschlichen Hand beziehungsweise eines Arms, gefertigt werden. Zusätzlich soll ein Interface erstellt werden, in dem der Benutzer Daten des Roboterarms und des Handschuhs einsehen kann. (Al-Maytah)

1.5 Allgemeine Zielsetzung

Ziel des Projekts ist es, eine Roboterhand zu bauen, die über einen kabellos verbundenen Handschuh gesteuert werden kann. Das Ziel ist es, Daten der Fingergelenkssensoren auszulesen, zu übertragen und die Bewegungen mit der Roboterhand nachzustellen. Endziel ist es, Daten richtig zu verarbeiten und die Roboterhand entsprechend zu bewegen. Daten sollen in einem Interface dargestellt werden.

1.6 Terminplan (Milestones)

Lastenheft fertig – 10.10.2023

Kostenkalkulationen fertig – 24.10.2023

Prototypen proof of concept – 21.11.2023

Hardwaredesign fertig – 05.12.2023

PCB-Design fertig – 09.01.2024

Softwareimplementierung für Integrationstest fertig - 30.01.2024

Integration aller Komponenten – 06.02.2024

User Interface fertiggestellt – 13.02.2024

Abnahme durch die Projektbetreuer – 12.03.2024

1.7 Geplantes Ergebnis der Prüfungskandidaten

Es soll eine Erfassung der Sensordaten möglich sein. Diese sollen übertragen und empfangen werden können. Die Roboterhand soll die Finger nach der Vorgabe des Handschuhs bewegen können. Als Endergebnis soll eine halbvolle 500mL Plastikflasche umschlossen und in der Luft gehalten werden. In dem Interface, dem User-Interface, sollen wichtige Daten dargestellt und Parameter verändert werden können.

1.8 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 der einzelnen Projektteile

2.1 Mechanik

2.1.1 Handschuh

Folgende mechanische Anforderungen muss der Handschuh nach Abschluss des Projekts erfüllen:

2.1.2 Roboterhand

Folgende mechanische Anforderungen muss der Handschuh nach Abschluss des Projekts erfüllen:

- 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 Elektronische Hardware

2.3 Handschuh

Folgende elektronische Anforderungen muss der Handschuh nach Abschluss des Projekts erfüllen:

2.3.1 Roboterhand

Folgende elektronische Anforderungen muss die Roboterhand 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.
- 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.

2.4 Software

2.4.1 Handschuh

Folgende Software Anforderungen muss der Handschuh nach Abschluss des Projekts erfüllen:

2.4.2 Roboterhand

Folgende Software Anforderungen muss der Handschuh nach Abschluss des Projekts erfüllen:

- 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.

2.4.3 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.

2.4.4 User Interface

3 Detaillierte Beschreibung der Projektteile

3.1 Handschuh (Eingabe)

Der Handschuh ermöglicht dem Benutzer die Roboterhand zu steuern. Dies geschieht durch Flexsensoren, die an den Fingern des Handschuhs 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 Roboterhand vorbereitet. Damit dem Handschuh funktioniert, bedarf es einer Spannungsversorgung. Diese wird in Form einer Batterie oder eines Akkus vorgesehen sein. Für einen möglichen stationären Betrieb kann der Handschuh auch mit einem Netzteil betrieben werden. Die Passform des Handschuhs 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 Handschuh ist so gebaut, dass er auch von Personen ohne elektronische Ausbildung und ohne Vorwissen bedient werden kann.

Folgende Richtlinien und Normen werden vom Handschuh 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 Roboterhand (Ausgabe)

Die Roboterhand ist die Ausgabe des Projekts und wird vom Roboterarm gesteuert. Jeder Finger wird von einem Servomotor bewegt. Dieser ist durch zwei dünne Schnüre mit dem korrespondierenden Finger verbunden. Die vom Handschuh 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 Roboterhand 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 der Roboterhand 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 Handschuh

4.2 Roboterhand

5 Hardwareentwicklung

5.1 Handschuh

5.1.1 Simulationen 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.

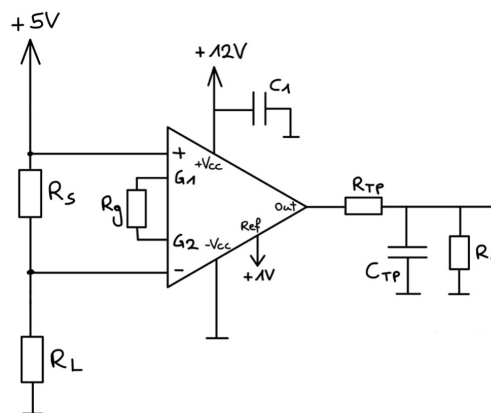


Abbildung 1: Messschaltung Spannungsmessung

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 (Rtp und Ctp) 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 Überlegungen und Berechnungen

Bild der Spannungs-Simulation hier einfügen!!

Der Widerstand der Flexsensoren ändert sich bei zunehmender Beugung von $25\text{k}\Omega$ auf $125\text{k}\Omega$.

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 $1.1\text{k}\Omega$ gewählt. Mit dieser Information haben wir dann den Shunt Rs mit 240Ω 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 R_{tp} 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 f_g hier einfügen!!

Durch den zu GND geschalteten Widerstand R_1 mit 10kOHM, wird er Eingang des danach kommenden ADCs entlastet.

5.1.3 Schaltplandesign

5.1.4 Platinendesign

5.2 Roboterhand

5.2.1 Simulationen und Versuchsaufbauten

5.2.2 Überlegungen und Dimensionierung

5.2.3 Schaltplandesign

5.2.4 Platinendesign

6 Softwareentwicklung

6.1 Handschuh

6.1.1 Konzepte und Überlegungen

6.1.2 Realisierung und Gliederung

6.2 Roboterhand

6.2.1 Konzepte und Überlegungen

6.2.2 Realisierung und Gliederung

6.3 User Interface

6.3.1 Entwicklungsumgebung

6.3.2 Konzepte und Überlegungen

6.3.3 Realisierung und Gliederung

7 Abschluss und Zusammenfassung

8 Anhang

9 Quellen -und Literaturverzeichnis

10 Abbildungsverzeichnis