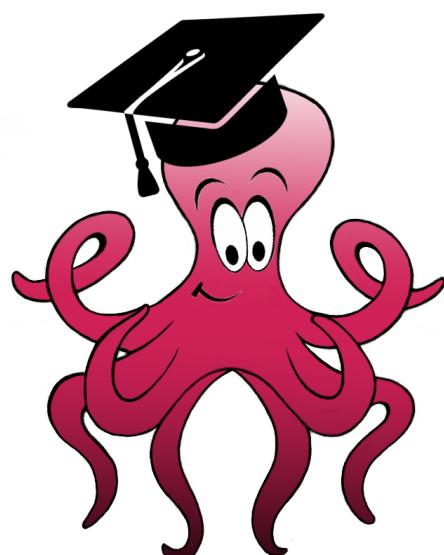


Projektmappe für den COSIMA-Wettbewerb 2018

Gestikkulläser

Die neue Technologie der Gestenerkennung

November 2018



Verfasst von

Christoph Behr
Cailing Fu
Nicole Grubert
Anna Pryadun
Daniel Wolff

Inhaltsverzeichnis

1. Einleitung	3
2. Stand der Technik	4
3. Funktionsweise und Aufbau	7
3.1. Oktokommander	8
3.2. Detektormodul	11
3.3. Software	15
4. Marketing und Kosten	17
A. Ausblick	19

1. Einleitung

Für Menschen sind Gesten schon seit frühester Zeit eine intuitive Art der Kommunikation. Bereits unsere Vorfahren haben sich auf der Jagd mittels verschiedener Gesten verständigt und bis heute nutzen wir Gesten in der zwischenmenschlichen Kommunikation, um uns z.B. mit Menschen zu verständigen, die eine andere Sprache sprechen. Da Gesten eine so grundlegende Kommunikationsmöglichkeit bilden, wurde in den letzten Jahren viel geforscht, um diese Art der Kommunikation ebenfalls auf den Bereich der Mensch-Computer-Interaktion anwenden zu können, der mit der fortschreitenden Digitalisierung unseres Lebens einen immer wichtigeren Stellenwert in unserem Alltag bekommt.

Mit dem Gestikulaser wollen wir ein neues Gestenerkennungssystem entwickeln, um statische Handgesten eines Menschen zu erkennen und diese zur Interaktion mit einem digitalen Endgerät zu nutzen. Dabei soll der Gestikulaser nicht mit einer Kamera arbeiten, wie die meisten heute verfügbaren Systeme, sondern stattdessen soll die Hand des Nutzers mit Infrarot-LEDs beleuchtet und die Gesten durch die erzeugten Reflektionsmuster erkannt werden. Eine auf diese Weise realisierte Geste-nerkennung erlaubt eine deutlich kompaktere Bauweise und hat nur einen geringen Energieverbrauch, wodurch sie für den Einsatz in eingebetteten Systemen ist bestens geeignet. Zudem kann sie auch in vollkommener Dunkelheit betrieben werden, da lediglich Lichtreflektionen detektiert werden müssen.

Durch den Entwurf eines modularen Stecksystems aus mehreren Komponenten soll es möglich sein, die für die Detektion der Lichtreflektionen benötigten Photodioden künftig in einem individuellen Muster anordnen zu können, wodurch das Gestenerkennungssystem optimal auf den jeweiligen Anwendungsfall angepasst werden kann. Im Alltag kann der Gestikulaser in unterschiedlichen Bereichen eingesetzt werden. Für den Anfang kann er für einfache Aufgaben, wie z.B. die Steuerung eines ferngesteuerten Autos eingesetzt werden. Prinzipiell sind aber auch ein Einsatz zur Steuerung einer Smart Home Einrichtung oder sonstiger digitaler Endgeräte denkbar.

2. Stand der Technik

Mit dem zunehmenden Einfluss von Computern auf die Gesellschaft wurde auch die Interaktion von Mensch und Computer ein immer wichtigerer Bestandteil unseres Alltags. Innerhalb der letzten Jahrzehnte wurde auf diesem Gebiet stetig geforscht und die verfügbaren Technologien weiterentwickelt, mit dem Ziel, die Interaktion von Mensch und Computer so natürlich wie möglich zu gestalten. Diese Entwicklung erstreckte sich von der ursprünglichen Kommunikation mittels Maus und Tastatur, über Touchscreens und Virtual Reality Systeme, bis hin zu Sprachsteuerungen von elektronischen Endgeräten. Vor diesem Hintergrund hat auch das Interesse an Gestenerkennung als zusätzlichem natürlichen Kommunikationsmittel zugenommen.

Bei Gestenerkennungssystem kann man grundsätzliche zwischen zwei verschiedenen Arten von Gestenerkennung unterscheiden: Bei der *gerätebasierten Gestenerkennung* werden die Bewegungen eines Nutzers durch Beschleunigungs- oder Positionssensoren wahrgenommen und einer Geste zugeordnet. Bei dieser Art der Gestenerkennung muss der Nutzer die Sensorik in irgendeiner Form am Körper tragen, damit die Bewegungsmuster richtig aufgezeichnet werden können. Dabei kann z.B. ein Handschuh wie z.B. der CyberGlove II des Unternehmens CyberGlove Systems (siehe auch Abbildung 2.1) oder ein einfacher Controller wie bei der Spielekonsole Wii von Nintendo (siehe Abbildung 2.2) zum Einsatz kommen. *Kamerabasierte Gestenerkennungsver-*



Abbildung 2.1.: Der CyberGLove II von CyberGlove Systems. Er enthält verschiedene Sensoren, unter Anderem Biegesensoren für die Finger, sowie Sensoren zur Detektion der Drehung des Handgelenks. Quelle: <http://www.cyberglovesystems.com/cyberglove-ii/>



Abbildung 2.2.: Ein Controller der Nintendo Spielekonsole Wii. Die Erkennung der Bewegungen des Spielers erfolgt mit Hilfe von Positions- und Beschleunigungssensoren. Quelle: https://www.chip.de/artikel/Nintendo-Wii-Test-2_140216386.html

fahren nutzen externe Systeme, die die Bewegungen des Nutzers beobachten und Aufnahmen von dessen Bewegungsabläufen erstellen. Dabei werden meistens Kamerasysteme eingesetzt. Auf die aufgenommenen Bilddaten werden dann verschiedene Bildrekonstruktionsverfahren sowie Bildanalyseverfahren angewendet, um die Gestendaten zu extrahieren. Nachdem die Gestendaten extrahiert wurden, erfolgt ein Abgleich mit einer Datenbank, in der verschiedene Aufnahmen von verschiedenen Gesten gespeichert sind. Dieser Datenbankabgleich erlaubt dann letztendlich die Zuordnung der Geste. Ein Beispiel für ein System, das auf diese Weise arbeitet, ist die Kinect Erweiterung für die XBox-Spielekonsole von Microsoft (siehe auch Abbildung 2.3). Beide Arten von Gestenerkennungssystemen werden aktuell eingesetzt.



Abbildung 2.3.: Das kamerabasierte Gestenerkennungssystem Kinect von Microsoft. Es werden Aufnahmen des Nutzers gemacht, die nach einer Analyse mit einer Datenbank abgeglichen werden, um die Geste zu zuordnen. Quelle: <https://blogs.microsoft.com/ai/kinect-for-windows-game-on-for-commercial-use/>

Allerdings haben auch beide Verfahren gewisse Nachteile: So erfordern gerätebasierte Gestenerkennungssysteme immer Sensorik am Körper des Nutzers, was sich in

alltäglichen Situationen häufig als unkomfortabel erweist. Demgegenüber können kamerasierte Gestenerkennungssysteme beispielsweise nur solche Gesten erkennen, die in der Datenbank vorhanden sind. Zudem sind Datenbankabfragen und Vergleiche mit den darin gespeicherten Daten häufig sehr rechenaufwändig, was sich als problematisch erweisen kann, falls eine Echtzeit Erkennung der Gesten gefordert ist. Eine Anbindung an die Datenbank erfordert zudem einen ständigen Internet Zugang oder alternativ große Mengen an Speicherplatz, um die für den Abgleich benötigten Gestendaten lokal zu speichern. Insbesondere für den Einsatz in eingebetteten Systemen, wie z.B. in der Automobil-Industrie sind diese beiden Gestenerkennungsverfahren nur bedingt geeignet: Gerätebasierte Gestenerkennungssysteme benötigen die externe Sensorik am Körper des Nutzers, die mit dem eingebetteten System kommunizieren und Daten austauschen muss. Dabei ist in eingebetteten Systemen häufig die Kommunikation über externe Schnittstellen sehr zeitaufwändig. Eingebettete Systeme dürfen zudem oft nur einen geringen Energieverbrauch aufweisen. Das erschwert den Einsatz von den Kamerasytstem, die von kamerabasierten Erkennungsverfahren benötigt werden. Zudem muss die für die Bilderkennung bzw. -analyse eingesetzte Software plattformunabhängig sein, damit sie auch im Rahmen des eingebetteten Systems eingesetzt werden kann.

Genau an dieser Schnittstelle zu eingebetteten Systemen soll der Gestikulaser eingesetzt werden können. Durch die Verwendung von einfachen elektronischen Bauteilen in Kombination mit Mikrocontrollern benötigt der Gestikulaser nur wenig Energie. Die kompakte und modulare Bauweise, auf die in Abschnitt 3.1 und Abschnitt 3.1 noch genauer eingegangen wird, erlaubt den Einsatz in unterschiedlichen Größenmaßstäben, da der Gestikulaser bei Bedarf einfach erweitert werden kann. Da die für die Gestenerkennung benötigten mathematischen Modelle (siehe dazu auch Abschnitt 3.3) für jede Anwendung individuell offline antrainiert werden, benötigt das System im Live-Betrieb vergleichsweise wenig Speicherplatz, was es ebenfalls für den Einsatz in eingebetteten Systemen qualifiziert.

3. Funktionsweise und Aufbau

Der Gestikulaser gehört weder zu den gerätebasierten, noch zu den kamerabasierten Gestenerkennungssystemen. Seine Funktionsweise ist aber mit der von kamerabasierten Gestenerkennungssystemen vergleichbar. Statt jedoch den Nutzer bei seinen Bewegungen zu beobachten und davon Aufnahmen zu machen, die in einem nachfolgenden Schritt von einem Computer analysiert werden, detektiert der Gestikulaser mit Hilfe von Photodioden die infraroten Lichtsignale einer LED-Quelle, die von der Hand des Nutzers reflektiert werden, während er eine Geste macht. Die an den Photodioden gemessenen Photoströme werden dann an einen Computer weitergeleitet und dort in einer Software weiterverarbeitet, welche mit Hilfe eines zuvor antrainierten künstlichen neuronalen Netzes die tatsächliche Handgeste erkennt. Nach der Auswertung wird die erkannte Geste genutzt, um mit einem zuvor hinterlegten Befehl ein beliebiges Endgerät anzusteuern. Dieser Ablauf ist in Abbildung 3.1 dargestellt. Der

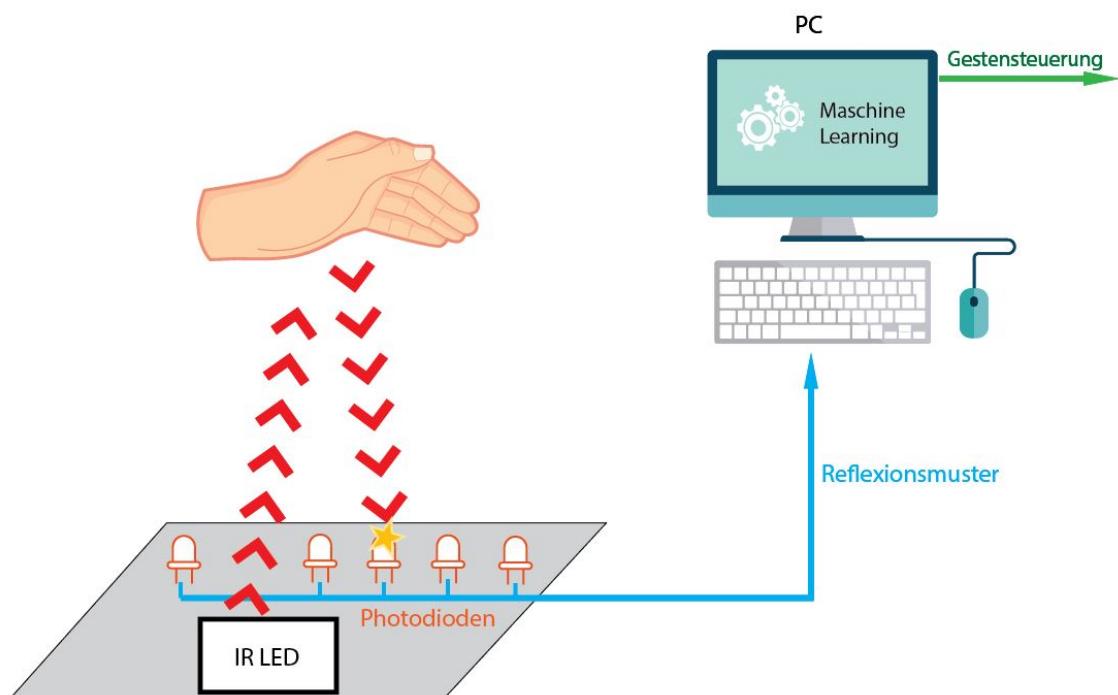


Abbildung 3.1.: Schmatische Darstellung der Gestenerkennung mit dem Gestikulaser.

Gestikulaser besteht aus mehreren Modulen, die über USB-Steckverbindungen mit einander verbunden werden können. Jedes Steckmodul besteht aus einer einzelnen kleinen Box, in welche die Elektronik integriert ist. Das Herzstück des Gestikulasers bildet der *Oktokommander* (siehe auch Abschnitt 3.1), der zugleich die Schnittstelle zu dem Computer bereitstellt, in dem die eigentliche Gestenerkennung erfolgt. An ihn können in der aktuellen ersten Version bis zu sieben *Detektormodule* (siehe auch Abschnitt 3.2) angesteckt werden. Auf jedem dieser Module befinden sich die Photodioden, um das reflektierte Licht zu messen.

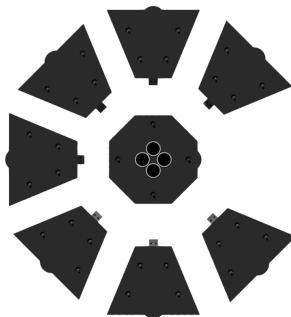


Abbildung 3.2.: Aufsicht auf ein CAD Modell des Gestikulasers. In der ersten Version können bis zu sieben Detektormodule (außen) an den Oktokommander (innen) angesteckt werden. Eine Erweiterung um weitere Detektormodule ist technisch möglich.

3.1. Oktokommander

Die zentrale Elektronik Einheit des Oktokommanders bildet ein Arduino Micro, der die übrigen Komponenten des elektrischen Schaltkreises über einen I2C Bus ansteuert. Für die Übertragung der Sensordaten ist er über ein Datenkabel mit einem Computer verbunden, wodurch zugleich die Stromversorgung des Schaltkreises sichergestellt wird. Die I2C Bus Verbindung benötigt lediglich 2 Pins auf dem Arduino, SDA für die eigentliche Datenverbindung und SCL für den vom Arduino vorgegebenen Takt. Die Ansteuerung jedes einzelnen Sensors erfolgt über eine, im System einzigartige, Hardware Adresse (I2C-Adresse). Mit den eingesetzten I2C Expandern lassen sich maximal 4 I2C Adressen erzeugen. Jeder I2C Expander verfügt dabei über 4 Pins, über die die verschiedenen Sensoren angesteuert und ausgelesen werden können. Auf diese Weise könnten maximal 16 Photodioden für die Detektion der von der Hand des Nutzers reflektierten Infrarot Strahlung der LEDs verwendet werden. Um ein Rahmenwerk zur Erkennung von möglichst vielen verschiedenen Gesten zu schaffen, ist eine große Anzahl von Photodioden nützlich. Aus diesem Grund wird

ein zusätzlicher I2C Multiplexer in den Oktokommander eingebaut. Der Multiplexer erhöht die Anzahl der verwendbaren I2C Expandermodulen und damit letztendlich auch die Anzahl der verwendbaren Photodioden, indem er die erzeugten Adressen multiplext. Der verwendete Multiplexer kann 8 verschiedene Adressen erzeugen, die sich nicht mit den Adressen der I2C Expander überlagern und besitzt je 8 SDA und SCL Pins, an denen die I2C Expander angeschlossen werden. In der Theorie lassen sich damit bis zu 1024 Sensoren über einen Mikrocontroller ansteuern. Für den Gestikulaser beschränken wir uns auf die Verwendung eines einzelnen Multiplexers, wodurch wir bis zu 128 Photodiode zur Detektion der reflektierten Strahlung verwenden können.

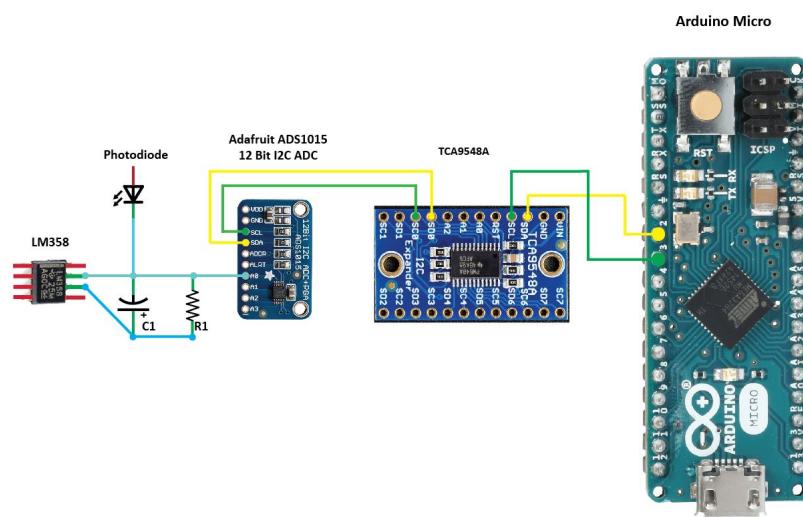


Abbildung 3.3.: Schematischer Aufbau des Oktokommanders bestehend aus einem Arduino einem I2C Multiplexer, einem I2C Expander und einer Verstärkerschaltung.

Eine schematische Übersicht über die Verschaltung der im Oktokommander untergebrachten Elektronik ist in Abbildung 3.3 dargestellt. Im Folgenden wird der Aufbau des Oktokommanders näher erläutert.

Konstruktion

Die Unterbringung der Elektronik im Oktokommander erfolgt über drei sogenannte *Stages*. Die Stages werden im Oktokommander übereinander angeordnet. Stage 1 ist in Abbildung 3.4 dargestellt. Es handelt sich dabei um eine Platine, auf der der Arduino Micro, der I2C Multiplexer, ein I2C-Expander, die Stromversorgung und sieben female USB-2A-Schnittstellen zur Verbindung mit den Detektormodulen angebracht sind. Zugleich bildet sie die unterste Elektronikeinheit des Oktokommanders.

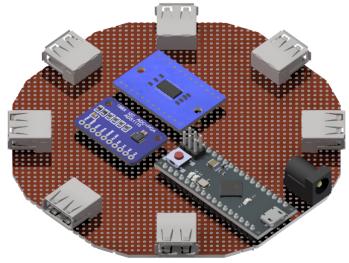


Abbildung 3.4.: Stage 1 des Oktokommanders, auf der die wichtigsten Elektronik Bauteile untergebracht sind.

Die LED-Einheit zur Lichterzeugung wird auf einer weiteren Plattform befestigt, die über der ersten Stage positioniert wird. Diese zweite Stage ist in Abbildung 3.5 dargestellt. Die LED-Einheit besteht aus vier LEDs, die zusammen mit Parabolzyllindern, die das von den LEDs emittierte Licht etwas nach oben fokussieren sollen, auf einen Kühlkörper geklebt werden. Zudem befinden sich die LED-Treiber auf dieser Stage, die eine externe Stromversorgung der LEDs durch ein Netzteil ermöglichen. Die Aufnahme der LED-Einheit besteht aus einer dünnen Aluminiumplatte für den Wärmeabtransport. Die Bohrungen in der Aluminiumplatte dienen lediglich zur Durchführung der Kabel zur unteren Stage 1.

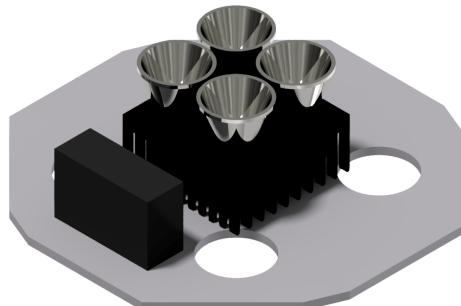


Abbildung 3.5.: Stage 2 des Oktokommanders mit den Infrarot-LEDs, die im Betrieb die Hand des Nutzers anstrahlen.

Die oberste Stage 3 trägt vier Photodioden. Das additiv aus ABS-Kunststoff gefertigte Bauteil ist in Abbildung 3.6 dargestellt.

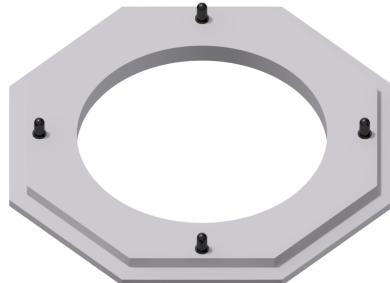


Abbildung 3.6.: Stage 3 des Oktokommanders, auf der vier Photodioden untergebracht sind, damit auch Reflektionen, die die Mitte des Gestikulators treffen, detektiert werden können.

Die drei Stages werden im Gehäuse des Oktokommanders aufgenommen. In Abbildung 3.7 ist das Gehäuse mit dem Deckel und der Elektronik des Oktokommanders dargestellt. Der Deckel und das Gehäuse des Oktokommanders werden ebenfalls additiv aus ABS-Kunststoff gefertigt.

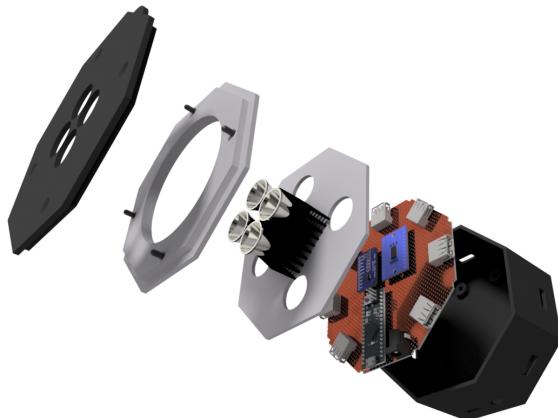


Abbildung 3.7.: Explosionsdarstellung des Oktokommanders.

3.2. Detektormodul

Die Hauptaufgabe der Detektormodule ist es, die Elektronik bereitzustellen, die zur Detektion der von der Hand des Nutzers reflektierten Infrarot-Strahlung benötigt wird. Aus diesem Grund enthält jedes Detektormodul vier Photodioden, eine Verstärkerschaltung, einen I2C Expander und einen USB-2A Stecker. Über den USB-2A-Stecker kann das Detektormodul an eine beliebige freie Stelle im Oktokommander angesteckt werden. Der USB-Stecker ist direkt mit dem I2C Expander verkabelt, der

ebenfalls mit der Verstärkerschaltung verbunden ist, die die von den Photodioden bei Lichtdetektion ausgegebenen Photoströme verstärkt und in die Datenleitung des Arduinos einspeist. Die Verstärkerschaltung wird im nächsten Abschnitt genauer erläutert.

Auslegung der Verstärkerschaltung

Die Verstärkerschaltung besteht aus zwei sogenannten Transimpedanzverstärkern. Diese setzen sich zusammen aus einem Operationsverstärker vom Typ LM358, je zwei Widerständen und Kondensatoren, sowie zwei Infrarot-Photodioden. Eine schematische Darstellung des Transimpedanzverstärkers ist in Abbildung 3.8 dargestellt.

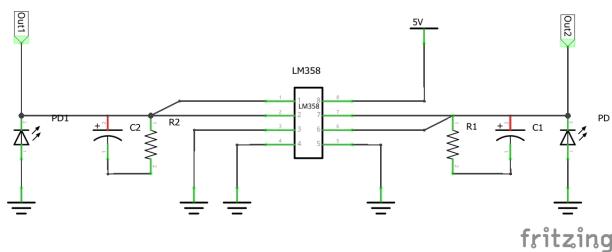


Abbildung 3.8.: Schematische Darstellung des Transimpedanzverstärkersystems im Detektormodul

Die Photodiode fungiert darin als Konstantstromquelle, die einen kleinen Strom an das System abgibt. Mit Hilfe des Widerstandes R_1 und des Kondensators C_1 kann der Operationsverstärker (LM358) eine vom Mikrocontroller verarbeitbare hohe Spannung ausgeben. Wichtig ist hier die möglichst genaue Wahl der Kapazität C_1 , ohne die das System anfangen würde zu Schwingen. Es gilt:

$$C_1 = \sqrt{\frac{C_I}{R_1 \cdot GBP}}$$

Hierbei steht GBP für *Gain Bandwidth Product*. Das ist diejenige Frequenz, bei der keine Verstärkung mehr stattfindet. Sie ist somit die Obergrenze für einen praktikablen Einsatz des Systems. Für den Operationsverstärker LM358 ist das GBP mit 1 MHz im Datenblatt angegeben. C_I ist die Summe der beiden Sperrsichtkapazitäten der Photodiode und des Operationsverstärkers. Typische Anwendungsgrößen für R_1 liegen bei etwa $1\text{ M}\Omega$. Der Arbeitsbereich des Verstärkersystems beschränkt sich durch aktive Bauelemente auf einen bestimmten Frequenzbereich. Dieser lässt sich wie folgt berechnen:

$$f_{3\text{ dB}} = \sqrt{\frac{GBP}{2\pi \cdot R_1 \cdot C_I}}$$

Für unser System liegt die Bandbreite bei etwa 73 kHz. Das ist auch die Grenzfrequenz des Verstärkersystems. Die Grenzfrequenz ist diejenige Frequenz, bei der die Verstärkung auf den $\frac{1}{\sqrt{2}}$ -fachen Wert der maximalen Verstärkung gesunken ist und bei der die Hälfte der maximalen Leistung an einen rein Ohm'schen Verbraucher abgegeben wird.

Konstruktion

Die wesentlichen Bestandteile des Detektormoduls sind eine Platine, die die Elektronik aufnimmt, und eine weitere Stufe für die Photodioden zur Detektion des reflektierten Lichts. Auch hier werden, ähnlich wie beim Oktokommander, die Stufen übereinander angeordnet. In Abbildung 3.9 ist Stufe 1 des Detektormoduls dargestellt. Dabei handelt es sich um eine Platine mit dem male USB-Stecker, dem I2C-Expander und der Verstärkerschaltung.

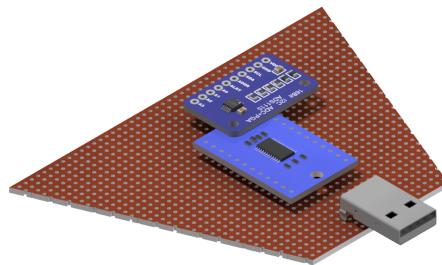


Abbildung 3.9.: Stufe 1 des Detektormoduls, bestehend aus einem I2C Expander und einer Verstärkerschaltung.

Darüber wird die zweite Stufe angeordnet. Diese additiv aus ABS-Kunststoff gefertigte Stufe erlaubt die Unterbringung der vier Photodioden und ist in Abbildung 3.10 dargestellt. Vom Aufbau her ist sie vergleichbar mit Stufe 3 des Oktokommanders.

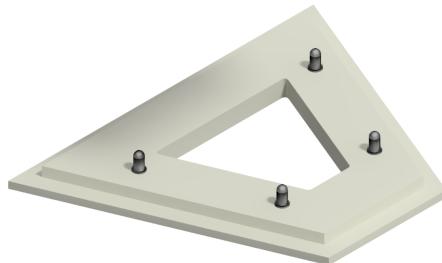


Abbildung 3.10.: Stufe 2 des Detektormoduls zur Unterbringung der Photodioden.

Das vollständige Detektormodul ist in Abbildung 3.11 dargestellt. Auch das Gehäuse und der Deckel des Detektormoduls werden additiv aus ABS-Kunststoff gefertigt.

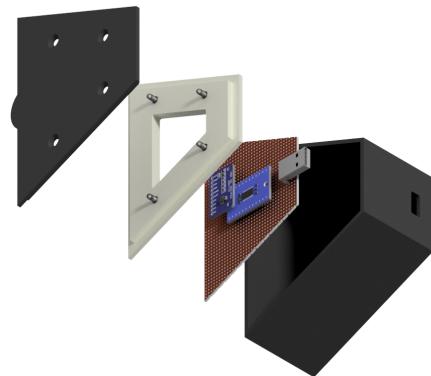


Abbildung 3.11.: Explosionsdarstellung des Detektormoduls.

Aktuell bietet der Oktokommander Ports für die Verbindung mit bis zu sieben Detektormodulen. Zusammen bilden der Oktokommander und die Detektormodule die Hardware des Gestikulasers. Der Zusammenbau der Detektormodule mit dem Oktokommander ist Abbildung 3.12 und Abbildung 3.13 dargestellt.

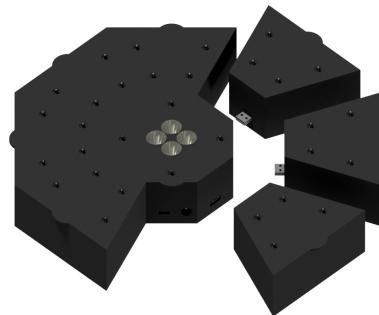


Abbildung 3.12.: Die Detektormodule können über USB Steckverbindungen mit dem Oktokommander verbunden werden.

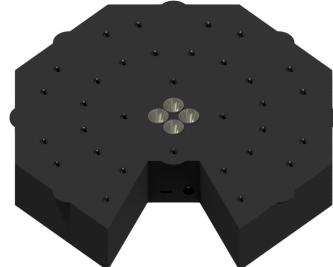


Abbildung 3.13.: CAD Modell des fertigen Gestikulasers, bestehend aus dem Oktokommander und sieben angeschlossenen Detektormodulen.

3.3. Software

Die Software-Seite des Gestikulasers ist für die Zuordnung der Gesten zuständig. Die von den Detektormodulen gemessenen Reflektionsmuster werden an den Mikrocontroller im Oktokommander und von da aus an einen Computer weitergeleitet, wo sie verarbeitet werden. Bei der dazu eingesetzten Software wurde komplett auf open-source verfügbare Programme und Bibliotheken gesetzt. Der Code auf den Microcontrollern wurde mit Hilfe der **Arduino IDE** entwickelt. Die Verarbeitung am Computer erfolgt mit Hilfe von **Python** Skripten. Für den Machine Learning Teil wurde die **TensorFlow™** Bibliothek verwendet. Die Software kann in zwei Teile unterteilt werden, die im Folgenden genauer erläutert werden.

Trainingsphase

Der erste Teil der Software kommt während der Trainingsphase zum Einsatz. Hier werden die von den Photodioden gemessenen Daten aus dem Microcontroller im Oktokommander zunächst ausgelesen und gemeinsam mit einem Label, das die gerade aufgenommenen Geste codiert, in eine Datei geschrieben. Die auf diese Weise gesammelten Daten werden dann im nächsten Schritt verwendet, um ein mathematisches Modell zu erstellen, das im Live-Betrieb angewendet wird, um die vom Nutzer gemachte Geste, einer der bekannten zuordnen zu können. Angenommen es stehen Daten von N Photodioden zur Verfügung, dann können diese Daten in dem Vektor $x \in \mathbb{R}^N$ zusammengefasst werden. Für den Anfang soll ein Modell entwickelt werden, welches einem beliebigen gemessenen Datensatz x eine der m vordefinierten Gesten G_1, \dots, G_m zuordnet. Dazu stellt der Nutzer vor Aufzeichnung der Daten in dem Python Skript ein, was für eine Geste G_j er als nächstes aufnehmen will. Nachdem für jede Geste ausreichend Daten aufgenommen wurden, kann als nächstes das mathematische Modell erstellt werden, um die Sensordaten zu verarbeiten. Ziel ist es, ein gegebenes Datum x_i , das den im Live-Betrieb gemessenen Photoströmen der verschiedenen Photodioden entspricht, eindeutig einer Geste G_j zuzuordnen. Damit Datum x_i der Geste G_j zugeordnet wird, muss die Bedingung

$$p(G_j|x_i) \geq p(G_k|x_i) \quad \forall k = 1, \dots, m \quad k \neq j$$

erfüllt sein. Dabei ist $p(G_j|x_i)$ die bedingte Wahrscheinlichkeit für Geste G_j gegeben die Photomessdaten x_i . Diese Bedingung lässt sich leicht überprüfen, allerdings ist die bedingte Wahrscheinlichkeit $p(G|x)$ unbekannt. Aus diesem Grund wird ein künstliches neuronales Netz mit Hilfe der zuvor aufgezeichneten Daten $(x, G) \in \mathbb{R}^N \times \mathbb{R}$ trainiert, das die unbekannte bedingte Wahrscheinlichkeit $p(G|x)$

annähert. Der Aufbau und das Training des neuronalen Netzes erfolgen mit Hilfe vordefinierter Methoden von **TensorFlow™**. Nach erfolgreich abgeschlossenem Training wird das neuronale Netz in eine Datei exportiert und kann nun im Live-Betrieb verwendet werden.

Live-Betrieb

Im Live-Betrieb wird das zuvor trainierte Modell nun verwendet, um die nun unbekannten Reflektionsmuster, die von den Photodioden gemessen und über den Mikrocontroller an den PC weitergeleitet wurden, einer bekannten Geste zuzuordnen. Nachdem die gemessenen Daten einer Geste zugeordnet wurden, wird der mit dieser Geste verknüpfte Steuerungsbefehl bestimmt und an das angeschlossene Endgerät weitergeleitet. Dieser Ablauf ist in Abbildung 3.14 dargestellt.

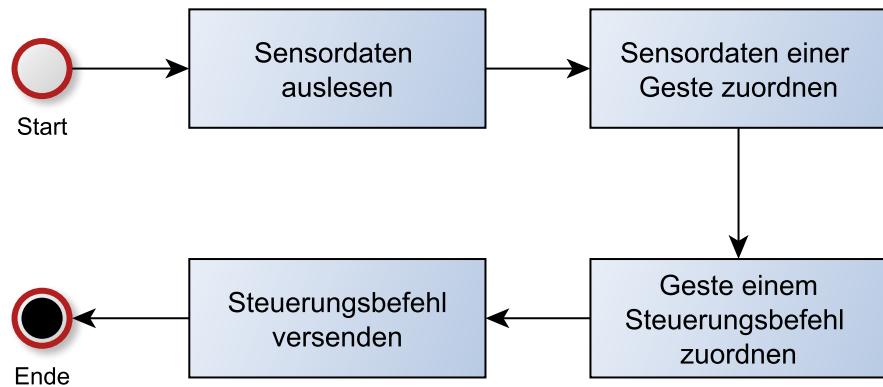


Abbildung 3.14.: Ablaufdiagramm der Software im Live-Betrieb: Die aus dem Microcontroller ausgelesenen Sensordaten werden zur Verarbeitung an das Python Skript übergeben, wo sie einer Geste und diese einem Steuerbefehl zugeordnet wird. Dieser Steuerungsbefehl wird schließlich an das anzusteuernde Endgerät weitergegeben.

4. Marketing und Kosten

Neben der Entwicklung unseres Produktes wurde im Laufe der Entwicklungszeit zusätzliche Öffentlichkeitsarbeiten geleistet. So wurde die Website <https://www.gestikulaser.de/> ins Leben gerufen, sowie der Instagram Account **laserharptos** mit über 100 Abonnenten betreut. In Zusammenhang mit diesen Marketing Maßnahmen wurden auch Sponsoren angeworben, welche Interesse an der Unterstützung unseres Projektes hatten.

Folgende Sponsoren haben uns bei diesem Projekt unterstützt:



Würth Electronik GmbH & Co. KG



Die Produktionskosten unseres Gestikulasers können wie folgt aufgegliedert werden:

Oktokommander

Produktnname	Kosten / Stück	Anzahl	Gesamtkosten	Kostenträger
Arduino Micro	19.99 €	1	19.99 €	TOS
I2C-Multiplexer TCA9548A	7.80 €	1	7.80 €	ILT
I2C ADS1015	11.20 €	1	11.20 €	ILT
Infrarot LED		4		Würth
LED Treiber		2		Würth
Infrarot Photodiode		4		ILT
OP-Verstärker	1.85 €	4	1.85 €	TOS
UBS 2 TypA -Mount		7		Würth
Passive Bauteile	—	—	1 €	TOS
			60 €	

Detektormodul

Produktnname	Kosten / Stück	Anzahl	Gesamtkosten	Kostenträger
I2C ADS1015	11.20 €	1	11.20 €	ILT
Infrarot Photodiode		4		ILT
OP-Verstärker	1.85 €	4	1.85 €	TOS
UBS 2 TypA -Plug		1		Würth
UBS 2 TypA -Mount		1		Würth
Passive Bauteile	—	—	1 €	TOS
			20 €	

Insgesamt wurden ein Oktokommander und sieben Detektormodule gebaut. Die Gesamtkosten für die Produktion belaufen sich also auf 200 €. Darüber hinaus sind Kosten von BLA € für Unterkunft und Anreise entstanden sowie T-Shirts in Wert von BLA €. Die T-Shirts wurden von Aconity3D gesponsert.

A. Ausblick

An dieser Stelle möchten wir einen Ausblick geben, wie der von uns entwickelte Gestikulaser weiterentwickelt werden könnte.

Hier wäre zunächst einmal eine Verfeinerung der aktuell durchführbaren Gestenerkennung denkbar, die nicht nur die Stellung der Hand, sondern auch die Krümmung der einzelnen Finger berücksichtigt. Zu diesem Zweck wurde bereits ein Prototyp für einen Sensorhandschuh entwickelt.

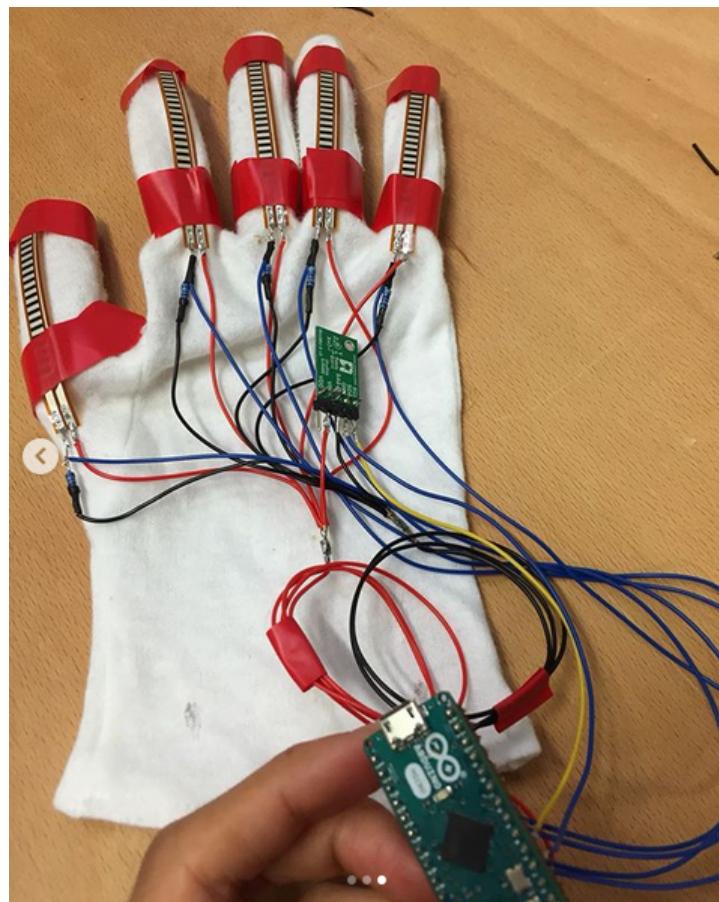


Abbildung A.1.: Aktueller Prototyp des Sensorhandschuhs. Es sind Biegesensoren für alle Finger, ein Gyroskop und ein Beschleunigungssensor im Einsatz. Bei der Verarbeitung der Sensordaten kommt ein Arduino Micro zum Einsatz.

Dieser könnte während der Trainingsphase vom Nutzer dazu verwendet werden, feinere Gesten aufzunehmen, wobei neben den von den Detektormodulen detektierten Photoströmen zusätzlich die Daten der auf dem Sensorhandschuh befindlichen Module gespeichert werden. Aktuell kommen hier Biegesensoren für jeden Finger, sowie ein Gyroskop und ein Beschleunigungssensor zum Einsatz. Diese verfeinerten Gestendaten stellen nun natürlich eine neue Herausforderung an das Modell, das zur Auswertung der Photoströme im Live-Betrieb eingesetzt wird: Statt wie vorher den eingehenden Photoströmen nur das Klassenlabel einer Geste zuzuordnen, muss es nun in der Lage sein, auf Basis der gemessenen Photoströme die wahrscheinliche Lage der Hand, sowie die Krümmung der Finger vorauszusagen. Dieses Problem ist mathematisch deutlich schwieriger zu lösen, was nicht zuletzt an der deutlich höheren Anzahl an Parametern liegt, die bestimmt werden müssen, um eine ausreichende Aussagekraft zu gewährleisten. Aufgrund der höheren Anzahl an Parametern werden zudem auch mehr Daten benötigt, um das Modell zu trainieren.

Ebenfalls denkbar wäre eine Erweiterung auf dynamische Gesten, bei denen der Nutzer die Hand bewegt. In diesem Fall müssen die Daten der Photodioden im Live-Betrieb in Form Zeitreihe aufgenommen werden und das neuronale Netzmodell muss diese Zeitreihe mit einer Geste verknüpfen. Auch hier ergibt sich ein mathematisch deutlich komplexeres Problem, das ein aufwändigeres Training und viele Trainingsdaten erfordert. Die Erweiterung um dynamische Gesten könnte ebenfalls mit der Verwendung des Sensorhandschuhs kombiniert werden.