

Bitte deutlich leserlich ausfüllen!

Deckblatt eines Toningenieur-Projekts

Vor- und Familienname Hannes Dieterle	Matrikelnummer 0973048
Studienrichtung Masterstudium Elektrotechnik-Toningenieur	Studienkennzahl V 066 413

Thema der Arbeit:

Gestaltung und Entwicklung eines audiosensitiven Leuchtanzuges
für eine Livemusikdarbietung

Angefertigt in der Lehrveranstaltung: Toningenieur-Projekt
(Name der Lehrveranstaltung)

Vorgelegt am: 29. 09. 2016
(Datum)

Beurteilt durch: Ao.Univ.Prof. Dipl.-Ing. Winfried Ritsch
(LeiterIn der Lehrveranstaltung)



Hannes Dieterle, BSc

Gestaltung und Entwicklung eines audiosensitiven Leuchtanzuges für eine Livemusikdarbietung

Toningenieur-Projekt

Masterstudium Elektrotechnik-Toningenieur

Universität für Musik und darstellende Kunst Graz

Institut für Elektronische Musik und Akustik (IEM)

Betreuer: Ao.Univ.Prof. Dipl.-Ing. Winfried Ritsch

Graz, September 2016

Danksagung

Ich danke Herrn Prof. Winfried Ritsch für die sehr gute Betreuung und die vielen Ideen zur Umsetzung dieses Projektes. Durch seine wertvollen Ratschläge wurde bei der Realisierung sehr viel Zeit gespart und viele Fehler bereits im Vorfeld vermieden. Sein sofortiges begeistertes Interesse an dem Thema war ein zusätzlicher Motivationsfaktor.

Weiters danke ich Prof. Marko Ciciliani und Prof. Robert Höldrich für gute Gespräche bezüglich audiovisueller Kunst an sich und für die Ratschläge bezüglich der Literatur zu dem Thema.

Außerdem möchte ich Monika Roscher für die Initiative zum Projekt danken. Diese tolle Idee hat mich sofort begeistert.

Weiterer Dank gilt Johannes Fahringer und Thomas Mayr, mit denen ich in der Zeit der Planung des Anzuges „Brainstorming“ bezüglich der technischen Umsetzung des Projektes betrieben habe.

Für Jorinda.

Zusammenfassung

Im Rahmen einer künstlerisch-technischen Projektarbeit wird für die Sängerin, Gitarristin und Bigbandleiterin *Monika Roscher* ein leuchtendes Kostüm entwickelt. Bei der künstlerischen Gestaltung wird auf die Interaktion zwischen Licht und Musik eingegangen. Der Anzug selbst wird außerdem als Instrument gestaltet, indem durch zusätzlich angebrachte Beschleunigungssensoren elektronische Klänge gesteuert werden. Eine Kernaufgaben der künstlerischen Planung ist es, die ästhetische Wirkung und die gewünschte Handhabung in der Auftrittssituation im Gesamtkonzept zu integrieren. Die Vermittlungsarbeit mit der Künstlerin bzgl. dieser Entscheidungen werden in dieser Arbeit dokumentiert.

Die technische Umsetzung richtet sich nach den ermittelten künstlerischen Anforderungen. Als leuchtende Elemente werden verschiedene LEDs und Leuchtschnüre (Electroluminescent Wires) gewählt, welche durch zwei auf dem Kostüm angebrachten Mikrocontrollern und einer Steuerschaltung aktiviert werden. Diese Vorrichtung wird durch einen externen Computer drahtlos gesteuert, während die Verbindung zum Anzug über digitalen Funk und der Rückweg über WLAN erfolgt. Mit Hilfe verschiedener Algorithmen werden Steuersignale für die leuchtenden Elemente implementiert, welche speziell auf das Musikstück angepasst werden und dabei das Audiosignal eines sich auf der Bühne befindlichen Mikrofons und der erzeugten synthetischen Klänge verarbeiten. Die Stromversorgung aller Elemente des Anzugs übernimmt ein Lithium-Polymer-Akku.

Abstract

This project thesis describes the design of an electroluminescent costume for the singer, guitar player and big band leader *Monika Roscher*. The work contains the artistic design as well as the technical implementation of the determined requirements. Concerning the artistic expression of the costume, the suit interacts with the music and even functions as a musical instrument itself by using additional acceleration sensors. Also, the aesthetic design and the strategy for the handling of the costume in a live situation are an important part of the work. Concerning these topics, the results of a meeting with the artist are documented in this thesis.

The technical implementation of the costume follows the determined artistic requirements. LEDs and EL-Wires (electroluminescent wires) are used as the luminescent elements of the suit. These elements, two microcontrollers and a control circuit are attached to the costume. Their control signals are being managed by a wirelessly connected external personal computer. While the transmission from the PC to the suit is provided by digital radio, the receiving channel is implemented using a WiFi connection. An on-stage microphone signal as well as the signal of the generated synthetic sound is transformed to specific lighting control signals by different algorithms. A rechargeable lithium polymer battery feeds all electronic elements of the costume.

Inhaltsverzeichnis

1. Einleitung	1
2. Recherche	3
2.1. Auftraggeberin	3
2.1.1. Portrait der Künstlerin Monika Roscher	3
2.1.2. Ist-Zustand des aktuellen Projekts	4
2.2. Gestalterische Recherche	5
2.2.1. Audiovisuelle Kunst	5
2.2.2. Embodiment - Live Elektronik und Bewegung	6
2.2.3. Leuchtkostüme auf der Bühne	7
3. Künstlerisches Konzept	9
3.1. Ziel des künstlerischen Ausdrucks des Kostüms	9
3.1.1. Konzept: Verstärkung der Musik	10
3.1.2. Konzept: Dialog zwischen Anzug und Musik	10
3.1.3. Erkennbarkeit einer menschlichen Gestalt	11
3.1.4. Spannungsbogen der Lichtdarbietung im Stück	11
3.2. Vermittlungsarbeit mit der Künstlerin	12
3.2.1. Inhalt des Stückes Starlight Night Crash	12
3.2.2. Umsetzung der Lichtshow mit Bezug auf den Inhalt des Stückes .	15
3.2.3. Soloteil des Stückes	16
4. Gestaltungskonzept	19
4.1. Konzepte zur Bedienung und Steuerung	19
4.2. Vorgaben der Aufführungspraxis des Stückes	20
5. Technische Realisierung	21
5.1. Realisierung der Hardware	21
5.1.1. Diskussion der verschiedenen Möglichkeiten zur Realisierung der Hardware	22
5.1.2. Energieversorgung und Leistungsbilanz der Elemente und Mikrocontroller	29
5.1.3. Verwendete Schaltung	31
5.1.4. Zusammenführung der Teile des Kostüms	37

Inhaltsverzeichnis

5.2. Realisierung der Software	40
5.2.1. Code für Arduino UNO	40
5.2.2. Code für Arduino DUE	41
5.2.3. Code für ESP8266	43
5.2.4. Patch für Max4Live	44
6. Fazit und Ausblick	51
A. Quellcodes	55
A.1. Code für Arduino UNO	55
A.2. Code für Arduino DUE	56
A.3. Code für ESP8266	60
Literaturverzeichnis	63

1. Einleitung

Nach der weltweiten Verbreitung vernetzter mobiler Geräte im Consumer-Bereich, wie die der Smartphones und Tablets, versucht sich in den letzten Jahren die Kategorie des *Wearable Computing*, der tragbaren Elektronik, weiter zu etablieren. Während diese kurz als *Wearables* bezeichneten Geräte schon länger in Form von Hörgeräten o. Ä. verbreitet sind, versuchen große Technologieunternehmen derzeit, die Branche durch vernetzte tragbare Geräte wie Smartwatches und Fitness-Tracker erneut anzutreiben¹. Im Bereich der zeitgenössischen Kunst treten Wearables beispielsweise im Kostümbild sowie im Bereich der Live-Elektronik und der Audiovisuellen Kunst auf. Leuchtende Kostüme sind dabei immer häufiger anzutreffen. Diese, meist mit LED-Technik oder mithilfe der Elektrolumineszenz realisierten Anzüge, finden bereits bei Bühnenshows im Bereich der elektronischen Tanzmusik oder bei Choreografien ihre Anwendung.

Hierbei ist die Verwendung der Elektrolumineszenz (kurz *EL*) ein immer größer werdendes Feld. Die Verfügbarkeit günstiger Hardware für diese Technik ermöglicht es sowohl Profis als auch Bastlern, Kleidung zu kreieren, welche mit EL-Schnüren (*Leuchtschnüre* bzw. *EL-Wires*) oder EL-Folien (*Leuchtfolie*) ausgestattet sind. Der Vorteil im Vergleich zur LED-Technik liegt in der Flexibilität dieser Elemente, welche zudem größere homogen leuchtende Flächen erzeugen können. Gepaart mit der Verwendung aktueller Mikrocontroller, welche ebenfalls für wenig Geld erhältlich sind, lassen sich so komplexere Leuchtkonzepte entwerfen, welche mit Hilfe verschiedener Sensoren auch auf äußere Reize reagieren können.

Für dieses Toningenieur-Projekt wird ein leuchtendes Kostüm für die Livemusikdarbietung eines Stückes im interdisziplinären Genre zwischen Jazz und elektronischer Musik der Musikerin *Monika Roscher* erstellt. Dabei soll die visuelle Komponente, erzeugt durch Leuchtschnüre und Leuchtdioden, auf die Musik adaptiv eingehen. Zusätzlich zur leuchtenden Wirkung des Anzugs werden Bewegungssensoren und Live-Elektronik eingesetzt, um das leuchtende Kostüm selbst zum Instrument werden zu lassen.

Eine wichtige Herausforderung der künstlerisch-technischen Arbeit ist hierbei, die Realisierbarkeit eines Auftritts mit dem Anzug zu ermöglichen. Dies wird v. A. in Kapitel 4 dokumentiert, welche das Gestaltungskonzept vorstellt. Weiterhin ist das Zusammenspiel zwischen Musik und Licht eine interessante Thematik, auf welche

¹Vgl. [BACHFELD; 2015], S. 17

1. Einleitung

neben der Vorstellung des Kunstprojektes im Kapitel 2 eingegangen wird. Die Vermittlungstätigkeit zwischen dem Verfasser und der Künstlerin, bei welchem auch das Ziel dieser Arbeit herausgearbeitet wird, findet sich im Kapitel 3 zum künstlerischen Konzept. Nicht zuletzt sind die technische Umsetzung und die Herstellung des Anzugs Teile der Projektarbeit, welche im Kapitel 5 beschrieben werden.

2. Recherche

Dieser erste Teil der Projektarbeit widmet sich der Recherche zu diesem Thema. Einerseits wird die Auftraggeberin des Projektes beschrieben, andererseits wird auf die gestalterische Recherche zum Projekt eingegangen.

2.1. Auftraggeberin

Diese künstlerisch-technische Arbeit ist im Zuge eines Musikprojektes initiiert worden. Die Auftraggeberin des Projektes ist die Musikerin Monika Roscher, mit deren Schaffen sich im folgenden Abschnitt 2.1.1 auseinandersetzt. Abschnitt 2.1.2 geht im Anschluss auf den Ist-Zustand des aktuellen Musikprojekts ein, für welches das Kostüm konzipiert wird.

2.1.1. Portrait der Künstlerin Monika Roscher

Die Münchner Sängerin, Gitarristin und Komponistin Monika Roscher war in den letzten Jahren neben der Veröffentlichungen zweier Musikalben unter dem Münchner Jazzlabel *Enja Records* mit einer Vielzahl von Auftritten mit ihrer selbst gegründeten 19-köpfigen Bigband (klassische Besetzung plus Live-Elektronik) in Europa unterwegs und im Rundfunk auf den Sendern *ARD*, *NDR*, *3sat*, *Arte*, *BR* und *Deutsche Welle* zu sehen und zu hören. Das Ensemble namens *Monika Roscher Bigband* wurde im Zuge eines Studienseminars an der Musikhochschule München gegründet und bestritt den ersten Liveauftritt im Sommer 2010². Die Stilistik der Musik der Preisträgerin des *Echo Jazz Newcomer National 2014* ist dabei schwer zu definieren. Die Presse bezeichnet ihr neues Werk *Of Monsters and Birds* z.B. als „Indie-Album mit dem Know How des Jazz, Elementen aus Minimal- und moderner Konzertmusik“³.

²Vgl. <http://www.monikaroscher.com/band/> [Stand: 16.09.2016]

³[KOBZINOWSKI; 2016]

2.1.2. Ist-Zustand des aktuellen Projekts

Für ein neues Nebenprojekt der Künstlerin mit einer im Vergleich zur Bigband geringeren Anzahl an Musikern wurden von der Künstlerin bereits einige neue Stücke für eine Besetzung von fünf bis neun Musikern komponiert. Neben einer Rhythmussektion und einer kleinen Bläserbesetzung bildet dabei die Live-Elektronik einen wichtigen Bestandteil des Ensembles. Dieser Part wird derzeit vom Verfasser dieser Arbeit ausgeführt, der auch beim Kompositionsvorgang mitwirkt.

Stilistik

Stilistisch ist dieses neue Projekt erneut schwer einzuordnen. Bislang existieren ca. sieben fertige Stücke, welche sich zwischen den Genres Jazz, Pop, Triphop und Electronica bewegen. Im Vergleich zur Monika Roscher Bigband sind die Lieder etwas mehr im Pop-Bereich anzusiedeln.

Musikstück *Starlight Night Crash* - das Stück für die Projektarbeit

Für das über fünf Minuten lange Stück namens *Starlight Night Crash* wurde bereits eine Demoaufnahme angefertigt, bei welchem ein experimentelles E-Gitarrensolo den Klimax des Ablaufes darstellt. Der Text handelt von einem Flugzeugabsturz über dem Ozean, welchen die Protagonistin wie durch ein Wunder überlebt, indem sie nachts unter dem Sternenhimmel gestrandet auf einer Insel aufwacht. Der Liedtext gibt den mentalen Vorgang der Protagonistin wieder, als sie versucht, das eben Geschehene nach und nach zu realisieren. Auf den Inhalt wird in Abschnitt 3.1 weiter eingegangen.

Das Stück wurde im Zuge eines Auftritts des Komponistenensembles der Musikhochschule München uraufgeführt. Dabei kam eine Besetzung mit Streichinstrumenten zum Einsatz und es wurde auf das E-Gitarrensolo verzichtet. Bei diesem Konzert verwendete Roscher während des Auftritts einige statisch leuchtende batteriebetriebene LEDs und eine statisch leuchtende, um ein Bein gewickelte EL-Schnur als Kostüm. Eine weitere LED-Kette wurde an das Mikrofonstativ angebracht (siehe Abb. 2.1).

Das Kostüm für die Uraufführung, welches Roscher als spontane Idee realisierte, war trotz der sehr guten Idee nicht in jeder Hinsicht zufriedenstellend. Die Statik der Lichter und besonders das beleuchtete Mikrofonstativ, welches vom Anzug ablenkt, bieten Verbesserungspotential. Die Anfrage Roschers an den Verfasser, eine Neuauflage des Kostüms anzufertigen, ist der Grund für den Beginn dieses Toningenieur-Projektes. Trotz der Unzulänglichkeiten dieser ersten Auflage des Anzugs, können hier bereits die ästhetischen Vorstellungen der Künstlerin herausgelesen werden.



Abbildung 2.1.: Uraufführung des Stückes *Starlight Nightcrash* am 13. März 2015 im *Gasteig*, München

2.2. Gestalterische Recherche

Der Auftrag, ein leuchtendes Kostüm zu gestalten, verlangt vor der Umsetzung nach der Recherche bzgl. bereits existierender Projekte, die ein ähnliches Konzept verfolgen. Zuvor wird in diesem Abschnitt versucht, die Disziplinen rund um das Projekt kurz einzugrenzen. Hierbei sind die beiden Begriffe der *Audiovisuellen Kunst* und des *Embodiments* zu definieren.

2.2.1. Audiovisuelle Kunst

Ein leuchtendes Kostüm bei einer Musikperformance ist dem Bereich der audiovisuellen Kunst zuzuordnen. „Audiovisuelle Kunst lässt sich an erster Stelle über die Qualität definieren, Klangliches und Visuelles zu einem eigenständigen Dritten zu verbinden.“⁴ Die Verbindung von Klang und Licht, welche die audiovisuelle Kunst beschreibt, ist dabei kein neues Phänomen. Seit Menschheitsbeginn werden Farben, Töne und Licht miteinander verbunden. Alleine die Art und die Begründung ihrer jeweiligen Zuordnung haben sich im Laufe der Zeit geändert⁵. Die Entwicklungen von Lichtorgeln, des *Live Cinemas* und schließlich des Tonfilms, sind der Kategorie der audiovisuellen Kunst zuzuordnen. Die Visualisierung von Musik ist eine der umfassendsten Unterkategorien dieses Überbegriffes neben anderen Feldern, wie beispielsweise der des *Projection Mappings* oder der *audiovisuellen Installation*. Weitere wichtige Bereiche in

⁴[FISCHER; 2014], Klappentext

⁵[JEWANSKI], S. 1

2. Recherche

diesem Zusammenhang bilden dabei die des *Live Cinema* und der *audiovisuellen Live Performance*⁶.

Während all diese Kategorien bereits kunsthistorisch untersucht sind, ist die Verwendung leuchtender Kostüme als audiovisuelle Kunstform in der Literatur noch nicht in den Fokus gerückt. Es handelt sich also um ein „unterbelichtetes“ Anwendungsfeld der audiovisuellen Kunst, auf welches aber viele Charakteristika der benachbarten Bereiche übertragbar sind. So entsteht bei der Verwendung eines leuchtenden Kostüms, welches mit der Musik verknüpft wird, im Idealfall ebenfalls ein eigenständiges Drittes, welches Licht und den Klang als neue Einheit miteinander verschränkt. Ob das Licht nun von einer Leinwand oder von dem Kostüm stammt, mag für diese neue Einheit keine große Rolle spielen. Jedoch erscheint ein Anzug als Quelle des Lichts als eine zusätzlichere Subjektivierung des Lichts anhand der darstellenden Person(en) im Vergleich zum Licht, das alleine vom Bühnenbild stammt.

2.2.2. Embodiment - Live Elektronik und Bewegung

Wie in Abschnitt 3.1 dieser Projektarbeit aufgezeigt wird, ist es bei diesem Projekt von Nöten, den Solopart des Stückes *Starlight Night Crash* anstatt eines Gitarrensolos durch den Anzug selbst via Live-Elektronik zu erzeugen. Die hierbei beste Form der Steuerung des Klanges ist mittels der Bewegungen der Trägerin zu erzielen.

In der Musik und in der Performance wird der Begriff des *Embodiments* gebraucht, welcher die Verbindung von Klang und der Bewegung des Interpreten verbindet. Der Begriff stammt eigentlich aus der Phänomenologie. Embodiment (auf deutsch *Verkörperung* oder *Verleiblichung*) ist eine These in verschiedenen Wissenschaften, die sich mit kognitiven und psychischen Prozessen beschäftigt, wobei davon ausgegangen wird, dass die Psyche einen Körper benötigt, um mit der Umwelt zu interagieren⁷. Die Theaterwissenschaftlerin Erika Fischer-Lichte verbindet den Begriff des Embodiments mit der *Präsenz* eines Performers⁸. Angewendet auf die Live-Darbietung elektronischer Musik kann somit behauptet werden, dass ein größeres Maß an Embodiment dazu beiträgt, die Präsenz des Performers und dadurch die sog. *Liveness* zu steigern.

Als ein Beispiel für das Prinzip des Embodiments in der Live-Elektronik kann das Projekt namens *Motion-Enabled Live Electronics (MELE)* angeführt werden. Bei diesem speziellen Ansatz wird der Zusammenhang der Bewegung eines Körpers und

⁶Vgl. [FISCHER; 2014], S. 6

⁷Vgl. [TSCHACHER, STORCH; 2012], S. 259

⁸[FISCHER-LICHTE; 2014], S. 7

Klangverarbeitung artikuliert, indem mittels eines hochauflösenden optischen 3D-Bewegungstrackingsystems elektronische Klänge geformt werden⁹. Ein derart genaues System erzeugt eine sehr akkurate und somit für das Publikum gut rezipierbare Kopplung des Klanges mit der Performance. Ist eine weniger präzise Kopplung notwendig, bieten auch einfachere Inertialsensoren wie Beschleunigungs- und Drehratensensoren (auf englisch *Accelerometer* und *Gyroscope*) die Möglichkeit, Bewegungen mit einer Klangsynthese oder -formung zu koppeln. Diese meist als *Microelectromechanical System* (MEMS) ausgeführten Elemente sind in vielen Consumer-Elektronikartikeln verbaut und somit günstig erhältlich. Dieser Ansatz wird bei dieser Projektarbeit für das leuchtende Kostüm verfolgt.

2.2.3. Leuchtkostüme auf der Bühne

Die Verwendung leuchtender Kleidung in der Mode sowie bei Theater-, Tanz- und Musikdarbietungen ist keine neue Idee. Nach der Verwendung fluoreszierender Materialien auf Textilien ist es spätestens seit der Entwicklung der Leuchtdiode in den 1960er Jahren möglich, batteriebetriebene leuchtende Elemente auf Kleidung anzubringen.

Heute sind Bühnenprogramme mit leuchtenden Kostümen keine Seltenheit mehr. Die Hamburger Band *Deichkind* verwendet beispielsweise für ihre Liveshow neben trapezförmigen Kopfbedeckungen mit LED-Bestückung insgesamt 200 an die Kostüme befestigte Smartphones, welche drahtlos gesteuert durch ein Netzwerkprotokoll vorgefertigte Videosequenzen abspielen (siehe Abb. 2.2 links)¹⁰.

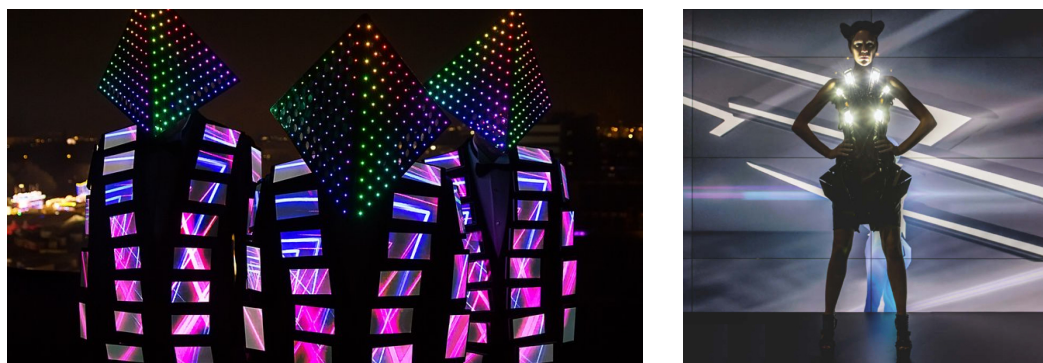


Abbildung 2.2.: **Links:** Bühnenkostüm der Hamburger Band *Deichkind* [www.zeit.de, ©Henning Besser]
Rechts: Entwurf der Modedesignerin A. Wipprecht für eine Messeshow [MAKE: 4/15]

⁹[ECKEL et al.; 2009], S. 1

¹⁰Vgl. <http://motherboard.vice.com/de/read/fragt-deichkind-smartphone-ganzkoerper-sakkos-sind-die-neuen-raver-westen-111> [Stand: 17.09.2016]

2. Recherche

Im Bereich der Mode finden sich ebenfalls mehrere Beispiele für leuchtende Kostüme. Die niederländische Modedesignerin Anouk Wipprecht verbaut beispielsweise *Arduinos* und *Edisons* in Ihre Designs um so leuchtende Kleider zu entwerfen, die z. B. von Autoscheinwerfern inspiriert sind (siehe Abb. 2.2 rechts)¹¹.

Beispiele für die Verwendung von elektrolumineszenten Materialien auf Bühnen finden sich vor allem im Bereich des Tanzes und der Choreografie. Verschiedene Tanzgruppen wie beispielsweise das *Team iLuminate* (siehe Abb. 2.3) oder die *Light Balance Electric Dance Group* führen mit ferngesteuerten Leuchtschnurkostümen Tanzeinlagen auf, welche neben einer somit erzielten „Laser-Optik“ auch optische Illusionen erzeugen können: Ähnlich wie bei dem für die Stadt Prag berühmten *Schwarzen Theater* können z. B. nur Teile der Anzüge beleuchtet werden, so dass es wirkt, als schwebten diese Leuchtschnurelemente frei im Raum.

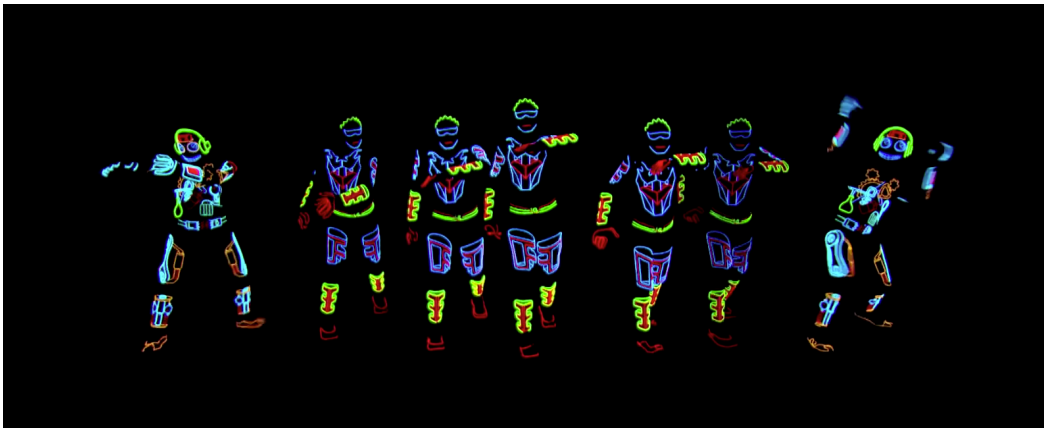


Abbildung 2.3.: Show von *iLuminate*, einer amerikanischen Firma spezialisiert auf die Fusion von Technik und Tanz [Screenshot eines Videos auf www.youtube.com]

Auch im Bereich der Populärmusik werden derartige Choreografien mit Leuchtschnurkostümen inszeniert. Beim *Eurovision Song Contest 2015* in Wien wurde eines der vorgestellten Lieder durch eine solche Einlage unterstützt. Im Bereich der ernsten Musik oder im Jazz liegen dem Verfasser dieser Arbeit noch keine Beispiele der Verwendung von Elektrolumineszenz vor. Es ist aber durchaus sehr wahrscheinlich, dass eine solche Performance in dieser Sparte bereits existiert.

¹¹[STEFFAN; 2015], S. 56ff

3. Künstlerisches Konzept

Dieses Kapitel widmet sich der Konzeption des leuchtenden Kostüms hinsichtlich künstlerischer Kriterien. Dabei ist das künstlerische Ziel des Projektes eine wichtige Überlegung für die Arbeit an sich. Die hierbei entscheidende Frage sollte also sein: „Was soll der leuchtende Anzug im Kontext der Musik ausdrücken?“ Dieser Frage geht Abschnitt 3.1 nach. Der zweite Unterpunkt dieses Kapitels dokumentiert die Vermittlungsarbeit mit der Künstlerin, die das generelle Ziel des Anzugeprojektes im Detail erweitert.

3.1. Ziel des künstlerischen Ausdrucks des Kostüms

Der Entwurf des Anzugs sollte nach künstlerischen Gesichtspunkten drei wichtige Anforderungen erfüllen: Zum einen sollte das Ergebnis für das Publikum **rezipierbar** sein. Die Verbindung des Lichts mit dem Musikstück sollten also eine schlüssige Einheit bilden. Die Frage nach dem Ausdruck des fertigen Anzugs ist hierbei entscheidend. Zum zweiten sollte das Ergebnis **reproduzierbar** sein, also kein reines Produkt des Zufalls darstellen. Zum dritten sollte das Kostüm **komponierbar** sein, es sollte also keine starre Kopplung zwischen Licht und Klang vorhanden sein, welche aus kompositorischer Sicht nicht verändert werden kann.

Während die letzten beiden Anforderungen technisch zu bewältigen sind, ist die Frage nach dem Ausdruck eine wichtige Vorabüberlegung hinsichtlich des Entwurfes. Eine wichtige Entscheidung ist hierbei, ob der Anzug eine reine Verstärkung der Musik (ähnlich einer komplexen Lichtorgel) bewirken soll und/oder ob zwischen Anzug und Musik eine Form des Dialogs möglich ist, welche den Anzug als eigenständigen Leuchtkörper zu einem Instrument werden lassen. Diese beiden Möglichkeiten, die im Verlauf des Stückes beide zur Anwendung kommen können, werden nun neben anderen wichtigen Überlegungen genauer betrachtet.

3. Künstlerisches Konzept

3.1.1. Konzept: Verstärkung der Musik

Das Stück *Starlight Night Crash* wird live auf der Bühne gespielt. Der Schall der Instrumente, die dabei erklingen und welche sowohl im Raum zu hören sind, als auch durch eine Beschallungsanlage verstärkt werden, könnte durch das System des leuchtenden Kostüms in einer abstrahierten Form in Licht umgesetzt werden. Solange der Zusammenhang zwischen Licht und Klang in einer rezipierbaren Form korreliert, übt somit das Licht in Form eines Feedbacks eine verstärkende Wirkung auf die auditiven Eindrücke aus. Das Gesamtergebnis ist ein audiovisuelles Kunstwerk und im besten Fall mehr wert, als die Summe beider Teile. Was in diesem Fall aber nicht garantiert ist, ist die Wahrnehmung des Anzuges als abstrahierte Quelle der Musik, welche von den Instrumenten gespielt wird. Durch einen wahrnehmbaren Zusammenhang könnte dieser aber trotzdem entstehen.

Das Konzept, die Musik durch das Kostüm zu verstärken, wird für das Stück nicht ausgeschlossen. Es könnte eine für das Stück dienliche Wirkung haben, um das Bild der erzählten Geschichte zu verstärken.

3.1.2. Konzept: Dialog zwischen Anzug und Musik

Mit Sicherheit wird der Anzug als eine Quelle der Musik wahrgenommen, wenn in der Musik eine Form des Dialogs zwischen dem Anzug und den restlichen Instrumentalisten entsteht. Dieser Dialog könnte zum Beispiel durch ein Frage-Antwort-Spiel zwischen Instrumenten und Anzug herbeigeführt werden. Hierbei wird durch die nicht vorhandene Korrelation zwischen dem Klang des Ensembles und den Leuchtelementen das Kostüm selbst als ein Instrument wahrgenommen. Ob das Licht des Kostüms dabei ebenfalls vertont wird ist eine eigenständige Frage. Dies muss nämlich nicht unbedingt der Fall sein.

Im Falle des bei dieser Arbeit behandelten Stückes bietet sich die künstlerische Umsetzung des Dialogs in Form eines Solos an. Da bei der Demoversion des Stückes bereits ein Gitarrensolo vorhanden ist, könnte dieses vom Anzug entweder ersetzt oder erweitert werden. Da die Solistin aber auch die Trägerin des Anzuges ist, liegt es in diesem Fall auf der Hand, das Gitarrensolo durch den Anzug zu ersetzen, da eine Gitarre einen großen Teil der Körperfläche bedecken würde und somit leuchtende Elemente auf dem Anzug verdeckt werden könnten. Die Frage, ob das Solo auch klanglich ersetzt werden sollte ist im Anbetracht des Spannungsbogens des Stückes einfach zu beantworten: Das Solo befindet sich zeitlich auf dem Klimax des Ablaufes. Ein stilles Licht-Solo wäre zwar eine interessante Variante, jedoch lange nicht so intensiv wie eine zusätzliche Vertonung des Lichtes.

3.1. Ziel des künstlerischen Ausdrucks des Kostüms

Es sollte also durch den Wunsch nach einer verstärkten Intensität des Solos eine vertonte Variante angestrebt werden. Durch das Vorhandensein eines Live-Elektronik-Musikers in der Besetzung des Ensembles liegt eine Vertonung des Lichts auf elektronischem, also computermusikbasiertem Weg auf der Hand. Der externe Laptop der Live-Elektronik ist bereits mit der Beschallungsanlage verbunden. Somit muss das Kostüm lediglich dazu in der Lage sein, das Computermusikinstrument zu steuern. Die wohl einleuchtendste Möglichkeit der Steuerung bei einem drahtlosen und batteriebetriebenen Kostüm ist durch das Prinzip des Embodiments gegeben. So könnten Bewegungssensoren den Klang des Computermusikinstrumentes ferngesteuert über Funk formen. Durch die zusätzliche Komponente der Bewegung des Kostüms neben Licht und Klang würde die Eigenständigkeit des Anzuges als Instrument weiter manifestiert werden.

3.1.3. Erkennbarkeit einer menschlichen Gestalt

Eine weitere wichtige Frage hinsichtlich des künstlerischen Ausdrucks des leuchtenden Kostümes ergibt sich in der Überlegung, ob der Anzug bei kompletter Dunkelheit die Silhouette einer menschlichen Gestalt aufweisen sollte. Dies ist vor allem dann wichtig, wenn diese Dunkelheit in einer Auftrittssituation hergestellt werden kann. Es ergibt dann einen großen Unterschied in der Wirkung, ob die leuchtenden Elemente alleine in ihrer Form Umrisse einer Person andeuten, oder ob diese nur an einigen Stellen unterstützend vorhanden sein sollten. In letzterem Fall wäre komplette Dunkelheit für die *Liveness* des Anzuges unter Umständen eher schädlich, da die Künstlerin dann nur als abstrakt leuchtende Stellen im Raum wahrgenommen werden würde und nicht als Person, welche soeben Musik darbietet.

3.1.4. Spannungsbogen der Lichtdarbietung im Stück

Bei professionellen Musikdarbietungen auf großen Bühnen werden seit der immer weiter entwickelten LED-Technik häufig sehr ausufernde Lichtshows konzipiert. Auffällig ist dabei, dass die vielen leuchtenden Elemente einer solchen Show teilweise erst im Verlauf des Konzertes voll zur Geltung kommen. Die britische Alternative-Rockband *Radiohead* verwendete bei der Tour zum Album *In Rainbows* in den Jahren 2006 bis 2009 ein Gitternetz aus vertikal ausgerichteten weißen Stangen für die Bühnenshow. Erst im Verlaufe des Konzertes wurde dabei erkennbar, dass diese mit RGB-LEDs bestückt waren, die zunächst nur weiß leuchteten um bis zum Ende der Show immer farbenfroher und eindrucksvoller zu werden. Auf diese Weise bekam die Lichtshow des ca. 2-stündigen Konzerts einen großen Spannungsbogen und bot damit zusätzlich zur Musik einige weitere Überraschungsmomente für das Publikum.

3. Künstlerisches Konzept

In diesem Sinne ist es auch ein generelles Ziel des Anzugprojektes, dass über den Verlauf des Stückes *Starlight Night Crash* ein gewisser Spannungsbogen gewahrt wird. Salopp gesagt sollte „das Pulver nicht bereits am Anfang verschossen werden“. Erst an bestimmten, gut ausgewählten Momenten im Stück, sollte der Anzug voll zur Geltung kommen.

3.2. Vermittlungsarbeit mit der Künstlerin

Bei einer künstlerisch-technischen Arbeit ist die Vermittlungsarbeit mit den dabei involvierten Künstlern eine wesentliche Aufgabe des Technikers bei der Zusammenarbeit. So widmet sich dieser Abschnitt eben dieser Aufgabe, welche bei einem mehrstündigen Gespräch des Verfassers der Arbeit mit der Künstlerin in Angriff genommen wurde.

Monika Roschers Lieder sind generell geprägt von meist aus einer subjektiven Perspektive erzählten Geschichten, welche meist auch im Liedtext behandelt werden. Bei Konzerten ihrer Bigband nutzt sie meist die Ansagen der Stücke, um auf diese Bilder hinzuweisen. Ihr Ziel ist es dabei, dem Publikum den Zugang zu diesem Bild zu erleichtern. Deswegen begibt sich die Musikerin häufig, auch durch die Verwendung von Verkleidungen, in eine Rolle um selbst besser in das Lied eintauchen zu können. Ein leuchtendes Kostüm könnte also für dieses generelle Ziel Roschers eine gewisse Unterstützung sein.

3.2.1. Inhalt des Stückes *Starlight Night Crash*

Um auf die fiktive Geschichte, die dem Stück zu Grunde liegt, auch in der Lichtshow durch den Anzug eingehen zu können, muss in der Vermittlungsarbeit auch auf den Inhalt des Liedes eingegangen werden. Dabei ist es hilfreich, einen Blick auf den Liedtext zu werfen, bevor dieser gedeutet werden kann. Zusätzlich zum Text werden im folgenden auch die Dynamikanweisungen der insgesamt acht Teile des Stückes angegeben, um gleichzeitig den Spannungsbogens des Stückes zu beschreiben.

Verse 1 (piano)

*Everywhere stars, water in the sky above
My eyes burn, my skin reflects the stars
If all those departed, where are they now? Stories of all*

Bridge 1 (piano)

*Take your turn, take your turn, turning tide, brightness
Time delay, time delay, coincidence brightness
All falls into place, all falls into place, water and sky absorb the light*

Verse 2 (forte)

*When I arrived, explosion burst the silence
Plastic melts, water in all dark shades
Destruction and crash, furious fire
Weight of the mass*

Bridge 2 (piano)

*Fight to breath, fight to breath, fight to be loving
Infinite time, mind stands still, consciousness in transformation
Uncage from the glue, free from the web, water and sky absorb the light*

Verse 3 (fortissimo)

*For all that I know, a new state has conquered me
My eyes burn, my skin reflects the stars
Swooshing water, sand and the island, Peaceful the night*

Bridge 3 (piano)

*Just one tone, just one sound, just a tone, in my head
Just one tone, just one sound, just a tone, in my head
Overwhelming deeper silence, peaceful the ocean in ancient stability*

Solo (instrumental, fortissimo)

Outro (piano)

*Just a dream, just a dream, oh a confusion i bare inside of me
Overwhelming deeper silence, peaceful the ocean in ancient stability*

3. Künstlerisches Konzept

Der Text enthält nach der Beschreibung Roschers ein Bild, welches die Musikerin mit einem Flugzeugabsturz auf offener See beschreibt. Die Protagonistin überlebt dieses Unglück wie durch ein Wunder, indem sie nachts unter dem Sternenhimmel gestrandet auf einer Insel aufwacht. Der Liedtext gibt den mentalen Vorgang der Protagonistin wieder, als sie versucht das eben Geschehene nach und nach zu realisieren.

Wichtige Elemente hierbei bilden die im Text beschriebenen Ohrgeräusche der benannten Überlebenden und außerdem mehrere „Flashbacks“ vom Absturz – also drastische Erinnerungsbruchstücke des Geschehenen, welche die Protagonistin erfährt. Diese beiden Elemente werden im Stück auch musikalisch umgesetzt so dass es im Stück einen Wechsel zwischen ruhigen und intensiveren Teilen gibt (siehe Dynamik der Teile). Die Flashbacks erreichen dabei im letzten Drittel des Stücks ihren Höhepunkt, als das Solo einsetzt. Insgesamt weist der Text keine konsistente Chronologie auf. Es handelt sich also eher um ein Fragment, welches die Irritation der Protagonistin der Geschichte weiter unterstreicht. Es folgt nun eine detaillierte Betrachtung der einzelnen Teile des Textes, wie sie aus Sicht der Urheberin in den Gesamtkontext passen.

Die erste Strophe (Verse 1) erzählt von der Strandung auf der Insel aus Sicht der Protagonistin. Es ist die erste Phase der Ruhe nach dem Absturz. Die Verunglückte betrachtet beim Aufwachen den Sternenhimmel und fragt sich, was wohl passiert sei. Der erste Zwischenteil (Bridge 1) zeigt eine erste Ahnung der Protagonistin davon, dass es wohl einen Absturz gegeben habe. Es ist also der erste Flashback, der aber noch milde ausfällt (piano). Die zweite Strophe ist eine Erzählung der Explosion, die der Protagonistin nun wieder ins Gedächtnis zurückgekommen ist. Der zweite Zwischenteil bildet eine Schlüsselstelle der Geschichte. Die Protagonistin tritt hier nach der Beschreibung Roschers in eine Art Schockzustand ein und überwindet sich dazu, von nun an den Kampf ums Überleben anzutreten. Das Wort *transformation*, bei welchem der Gesang von der Live-Elektronik aufgenommen und verzerrt wird, ist dabei eine Art Wendepunkt der Geschichte. Ab jetzt setzen die Flashbacks vom Absturz mit all ihrer Wucht ein. Dieser neue Geisteszustand wird in der dritten Strophe mit der Bezeichnung *a new state* direkt angesprochen. Die Flashbacks werden musikalisch durch das Fortissimo unterstützt. Dieser Teil wirft für die Protagonistin nach Ansicht Roschers auch die Frage auf, ob sie nun lebe oder nicht. Im dritten Zwischenteil verblassen die Flashbacks und es bleibt neben der „alten Stabilität der Natur“ ein Tinnitus im Ohr der Protagonistin. Im Soloteil kehren die intensiven Flashbacks jedoch wieder zurück und bilden den dynamischen Klimax des Stückes. Roscher empfindet hier in der Musik eine Form der Verarbeitung des Geschehenen, eine Art Austreten aus dem vorher beschriebenen Schockzustand. Das Outro des Stückes wirft unerwarteterweise erneut die Frage auf, ob die Protagonistin das Erlebte nur geträumt hat. Dies deutet erneut auf die erschütterte, aufgewühlte Verfassung der Hauptperson hin und hinterlässt der Geschichte ein offenes Ende. Es könnte aber auch den Wunsch beschreiben, dass alles

nur geträumt war. Dies wird von der Musikerin bewusst offen gelassen.

3.2.2. Umsetzung der Lichtshow mit Bezug auf den Inhalt des Stückes

Nachdem nun der Inhalt des Stückes beschrieben wurde, wie er von der Musikerin selbst gedeutet wird, steht nun die Umsetzung des Inhaltes durch das leuchtende Kostüm im Vordergrund.

Bei den Vermittlungsgesprächen ergab sich generell, dass die Künstlerin im Anzug auch in möglicher kompletter Dunkelheit als menschliche Gestalt erkannt werden kann. Es handelt sich um die Rolle einer Hauptperson, die eine Geschichte erzählt. Deswegen wird beim Design des Anzuges versucht, den kompletten Körper mit leuchtenden Elementen zu bestücken, um die menschliche Silhouette sichtbar zu machen.

Eine weitere wichtige Idee ist es, mit dem Kostüm in Form flackernder LEDs eine Sternennacht zu symbolisieren. Die oben besprochene „Transformation“ im zweiten Zwischenteil leitet dann stärkere Effekte des Anzugs ein. In Anbetracht der beiden in Abschnitt 3.1.1 und 3.1.2 vorgestellten Konzepte, ob der Anzug als Verstärkung der Musik oder im Dialog zur Musik in Erscheinung tritt, liegt somit eine Mischform der beiden Konzepte vor: Über das ganze Stück wird als Hintergrund zur Musik die Sternennacht symbolisiert. Bei den Flashbacks, die die im Text beschriebenen Explosionen ansprechen, sollte eine Verstärkung der Musik vorliegen. Im Soloteil jedoch sollte zur Musik ein Dialog geführt werden, da hier der Anzug ein eigenständiges Instrument darstellt.

Die Flashbacks können in der Lichtshow teilweise eine intensive, stroboskopartige Wirkung aufweisen. Dies kann sowohl von LEDs als auch von Leuchtschnüren ausgeführt werden. Vielleicht auch von allen Elementen gleichzeitig.

Die „Transformation“, die im zweiten Zwischenteil im Text angesprochen wird, sollte bzgl. der Lichtshow bei eben diesem Wort ein chaotisches Lauflicht von unten nach oben aufweisen, um die Zustandsänderung der Protagonistin anzudeuten. Dies könnte mit einer größeren Zahl horizontal übereinander angebrachter Leuchtschnüren ermöglicht werden. Am besten wäre es, wie oben beschrieben, wenn hier die Leuchtschnüre zum ersten Mal auftreten. So wird auch der Spannungsbogen gewahrt, den die Lichtshow aufweisen sollte.

Insgesamt sollte die Lichtshow die Dynamik des Stückes unterstützen. Farblich wünscht sich die Künstlerin eher kalte, neutrale Farben also ein helles blau oder weiß. So wären weiße LEDs und hellblaue, einfarbige Leuchtschnüre eine gute Lösung.

3.2.3. Soloteil des Stückes

Was den Soloteil des Stückes angeht, wurde mit der Künstlerin vereinbart, das ursprünglich geplante E-Gitarrensolo zu ersetzen und mittels des Anzuges zu realisieren. So werden keine leuchtenden Elemente verdeckt. Höchstens eine mit Leuchtelementen präparierte Gitarre wäre eine Möglichkeit. Eine Umsetzung des Solos mittels eines Computeralgorithmus, der eine E-Gitarre simuliert wäre aber eine bessere Lösung. So kommt der Anzug selbst mehr zur Geltung, indem er als Instrument wahrgenommen wird. Eine direkte Steuerung einer Computermusiksoftware durch Bewegungssensoren auf dem Anzug bilden hierfür eine gute Lösung.

Der Sound des Solos sollte einen ähnlichen Klangcharakter wie der des ursprünglichen E-Gitarrensolos aufweisen. Wie bereits beschrieben, symbolisiert das Solo, welches im Klimax der Flashback-Szenen des Stückes vorkommt, einen Flugzeugabsturz – besser gesagt die Erinnerung daran. Auf einer Demoaufnahme wurde deswegen in einer Studiosituation ein Solo zusammengeschnitten, bei welchem lang angehaltene verzerrte Töne und Glissandi auf der E-Gitarre erzeugt wurden. Das Resultat weist einen wilden und unkontrollierbar klingenden Charakter auf, der dem *Noise*-Genre zugeordnet werden könnte.

Die lang gehalten Töne wurden dabei durch einen *Guitar-Resonator* namens *Ebow* erzeugt. Bei diesem handlichen Gerät handelt es sich um eine batteriebetriebene Spule, die ein oszillierendes magnetisches Feld auf eine sehr kurze Distanz hin emittiert. Wird diese Spule direkt über eine Gitarrensaite gehalten, wird die Saite angeregt und es entsteht ein beliebig lang gehaltener Ton. Dieser ist laut dem Hersteller¹² reich an Obertönen. Die Anregungsfrequenz des Ebows richtet sich nach der schwingenden Saite und wird mittels eines Tonabnehmers detektiert. Es handelt sich also um eine positive Rückkopplung. Durch die Platzierung des Ebows entlang der Saite können nun verschiedene harmonische Teiltöne angeregt werden. Je nachdem an welcher Stelle ein Teilton einen Schwingungsbauch aufweist, ist die Anregung dieser Frequenz wahrscheinlich.

Die Glissandi wurden auf der E-Gitarre mit der linken Hand erzeugt. Durch eine Streichbewegung über die Bünde (meist als *Slide* bezeichnet) entstehen durch eine bundweise Verkürzung oder Verlängerung der Saite diskrete Glissandi in Halbtonschritten. Da der Ebow bei der Aufnahme der Glissandi an der selben Stelle verweilte, wurden immer wieder andere Teiltöne angeregt, was zu Frequenzsprüngen innerhalb der Glissandi führte. Im Falle der Demoaufnahme sind vor allem Oktavsprünge zu hören, also ein Übergang vom ersten zum zweiten Teilton. Um immer den gleichen Teilton anzuregen,

¹²Vgl. <http://www.ebow.com> [Stand: 19.09.2016]

wäre es notwendig, das Verhältnis der Längen Bridge-Ebow zu Ebow-Finger aufrechtzuerhalten. Man müsste die Anregungsposition also dynamisch an die Fingerposition anpassen.

Dieser besondere und charakteristische Klang der springenden Glissandi, bei ständig wechselnden Teiltönen sollte nun nach Möglichkeit mittels des Computermusikinstrumentes nachgeahmt werden, da er in seiner Wirkung sehr gut zu dem gewünschten Bild der Handlung des Stückes passt. Die genaue Tonfolge des Solos ist dabei weniger wichtig und kann durchaus auch improvisiert werden, solange die Charakteristika des Klanges dem Original ähneln.

4. Gestaltungskonzept

Nachdem nun die Ergebnisse der Vermittlungsarbeit mit der Künstlerin dokumentiert und die Konzeption des leuchtenden Kostüms hinsichtlich künstlerischer Kriterien vorgestellt wurden, stellt dieses kurze Kapitel einige praktische Entscheidungen zur Umsetzung der erarbeiteten Ziele zusammen. Zunächst werden in Abschnitt 4.1 Ziele für die Bedienung und Steuerung des Anzuges zusammengetragen, bevor in Abschnitt 4.2 einige weitere Vorgaben angesprochen werden, die sich aus der Aufführungspraxis des Stückes ergeben.

4.1. Konzepte zur Bedienung und Steuerung

Die Steuerung der leuchtenden Elemente auf dem Kostüm kann in der Aufführungssituation nach zwei möglichen Konzepten erfolgen: Entweder der Anzug reagiert autark auf die Musik, oder er wird von außen gesteuert. Diese externe Steuerung könnte von einem drahtlos verbundenen Laptop erfolgen. Da das Ensemble bereits einen Musiker aufweist, der den Part der Live-Elektronik mit Hilfe eines Computers übernimmt, könnte die Steuerung hierüber realisiert werden. Dies hätte den Vorteil, dass der Anzug selbst nicht in der Lage sein muss, die Teile des Stückes eigenständig zu erkennen. Ein solcher Music-Information-Retrieval-Algorithmus könnte sich als relativ kompliziert und schwierig in der Umsetzung erweisen. Außerdem müsste der vom Anzug erzeugte Solo-Klang im Falle einer autarken Konzeption vom Mikrocontroller erzeugt und im Anschluss drahtlos zur Beschallungsanlage gesendet werden oder er würde nur die Daten der Bewegungssensoren senden, was einfacher wäre. Eine Drahtlosverbindung, welcher Art auch immer, wäre also in jedem Fall von Nöten. Da der Live-Elektronik -Musiker im Stück *Starlight Night Crash* bereits gespielte Daten über einen MIDI-Controller an den Laptop sendet, können diese auch zur Steuerung des Anzugs verwendet werden und alle Audiosignale direkt auf dem Laptop erzeugt werden, welcher bereits mit der Beschallungsanlage verbunden ist. Dies stellt sich höchstwahrscheinlich als die erfolgsversprechendere der beiden Varianten heraus.

Weitere Eingabequellen, die zur Steuerung des Lichtes beitragen können, bilden verschiedene Mikrofone, welche sich in einer Bühnensituation des Ensembles bereits für

4. Gestaltungskonzept

die Beschallungstechnik auf der Bühne befinden und auf welche der Computermusiker über einen Signalsplitter Zugang besitzt. Hierbei würden sich bei dem Stück vor allem die Bass Drum des Drum Sets anbieten sowie das Mikrofonsignal der Sängerin. Die Signale der Live Elektronik und des Solo-Klanges des Anzuges könnten bei diesem Konzept intern über den Laptop zugänglich gemacht werden.

Neben der Steuerung der leuchtenden Elemente des Anzuges ist auch dessen Bedienung durch die Trägerin zu bedenken. Der Anzug sollte in einer hektischen Auftrittssituation möglichst schnell an- und ausgezogen werden können. Im besten Fall sollte die Künstlerin dies alleine erledigen können. Außerdem muss die Elektronik von der Trägerin selbst an- und ausgeschaltet werden.

Eine drahtlose Verbindung sollte im besten Fall nach Inbetriebnahme des Kostüms von selbst erfolgen. Der Computermusiker könnte aber auch hier einen kurzen Befehl zur Verbindung in einer Livesituation durchführen.

4.2. Vorgaben der Aufführungspraxis des Stückes

Es folgen nun ein paar Vorgaben bezüglich der Aufführungspraxis des Stückes, welche eingehalten werden müssen.

Nach Wunsch Monika Roschers sollte das Kostüm am besten drahtlos realisiert werden. Unter Umständen wäre aber auch ein Kabel möglich. Die Bewegungsfreiheit der Trägerin wäre aber in diesem Fall eingeschränkter, was im besten Fall zu vermeiden ist.

Die Noten von *Starlight Night Crash* weisen sehr viele Stellen auf, die eine Fermate beinhalten. Es wird also ein Dirigat der Sängerin benötigt um die Musiker exakt zusammenspielen zu lassen. Da das Lied nach Möglichkeit in einer sehr dunklen Bühnenumgebung aufgeführt werden sollte, damit das Kostüm besser zur Geltung kommt, ist es jedoch schwierig, das Dirigat für alle Musiker sichtbar zu machen. Leuchtende Manschetten des Anzuges bilden hierfür eine Lösung. Diese könnten genau wie die anderen Leuchtschnüre auf dem Anzug platziert und in den Stellen, in denen optische Signale der Sängerin benötigt sind, statisch angeschaltet werden.

5. Technische Realisierung

Dieses Kapitel bildet neben der künstlerischen und gestalterischen Konzeption einen weiteren Hauptteil dieser Arbeit und dokumentiert die technische Realisierung des leuchtenden Kostümes und dessen Kommunikation mit dem zur Steuerung vorgesehenen Laptop. Die Abschnitte 5.1 und 5.2 behandeln dabei die Umsetzung der Hard- bzw. Software, die für das Projekt zur Anwendung kommen.

5.1. Realisierung der Hardware

Bei der Hardwareentwicklung für das Projekt handelt es sich um ein komplexes Problem. Die Elektronik auf dem Kostüm muss viele Aufgaben gleichzeitig erfüllen und dabei effizient genug gestaltet sein, um batteriebetrieben funktionieren zu können. Zum einen werden ein oder mehrere Mikrocontroller benötigt, welche die Funktionen des Anzugs koordinieren. Zum zweiten müssen diese auch mit einem über Funk verbundenen Laptop mit Hilfe verschiedener WiFi- oder Funkchips kommunizieren. Zum dritten sind neben gleichstrombetriebenen LEDs (welche mittels der Verwendung der Pulswellenmodulation, kurz *PWM*, gedimmt werden können) auch Leuchtschnüre zu versorgen, welche eine Wechselspannung mit einem wesentlich höheren Effektivwert von ca. 150 V benötigen, um zu leuchten. Zum vierten ist es eine weitere wichtige Vorgabe des Projektes, einen ausreichenden Schutz der Trägerin vor den erzeugten höheren Spannungen sicherzustellen. Galvanische Trennungen an verschiedenen Stellen der Schaltung sind dadurch ebenfalls notwendig. Die fünfte Aufgabe des Anzuges ist, die Daten zweier Accelerometer, welche an den beiden Armen des Kostüms angebracht werden, zum Laptop zu senden.

Im Folgenden werden in Abschnitt 5.1.1 zunächst verschiedene Konzepte aufgelistet und diskutiert, die zur Umsetzung des Projekts möglich wären. Danach wird in den Abschnitten 5.1.2 und 5.1.3 die verwendete Lösung betreffend der Energieversorgung und der Schaltung vorgestellt.

5. Technische Realisierung

5.1.1. Diskussion der verschiedenen Möglichkeiten zur Realisierung der Hardware

Um die eben angesprochenen Aufgaben seitens der Hardware zu erfüllen, existieren aus heutiger Sicht mehrere Möglichkeiten zum generellen Design. Vorteilhaft ist bei allen Lösungen, dass sowohl Mikrocontroller als auch verschiedene Chips, die zur drahtlosen Kommunikation dienen, sehr günstig in der Anschaffung und meist sehr gut dokumentiert sind. Neben vielen weiteren, werden im Folgenden drei verschiedene Möglichkeiten zur Realisierung der Hardware ausgewählt, welche für das Projekt in Frage kommen.

Möglichkeit 1 - Verwendung vieler ESP8266 via Websocket-Protokoll

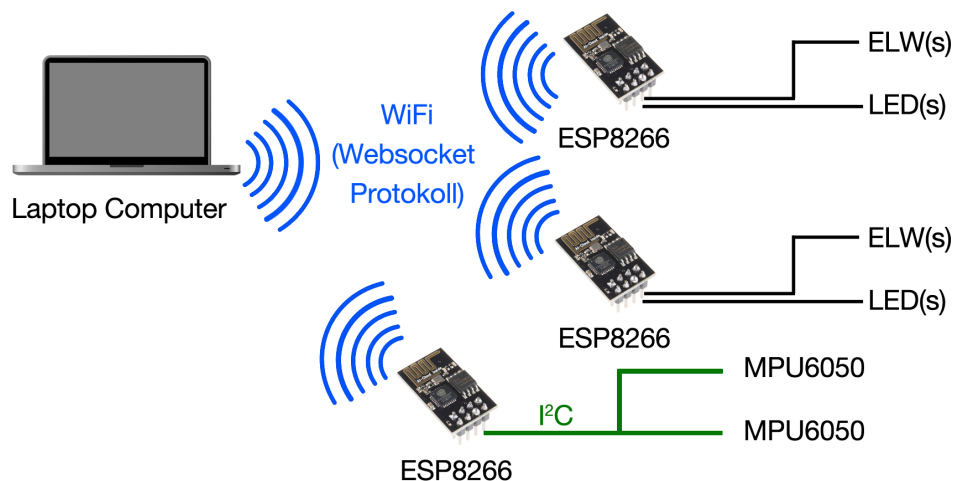


Abbildung 5.1.: Möglichkeit 1 zur Realisierung der Hardware

Die Verwendung vieler Mikrocontroller, welche allesamt über ein Websocket-Protokoll vom Laptop ferngesteuert werden, bildet die erste Möglichkeit, um die Vorgaben des leuchtenden drahtlosen Kostüms zu erfüllen. Hierzu wäre die Verwendung mehrerer Chips wie beispielsweise des *ESP8266*, ein 32-Bit-Mikrocontroller der chinesischen Firma *espressif*, eine plausible Option. Dieser Chip ist sehr energiesparend und verfügt über ein integriertes WiFi-Modul¹³. Dabei sind zur Programmierung des integrierten Prozessors mit einer maximalen Taktrate von 160 MHz bereits viele Libraries für unterschiedliche Entwicklungsumgebungen verfügbar. Der größte Vorteil dieses Chips liegt in seinem Preis. Die kleinste Ausführung mit acht zugänglichen Pins des ICs (vgl. Abb. 5.1) ist im Internet für unter zwei Euro erhältlich. Größere Ausführungen liefern eine größere

¹³Vgl. <http://espressif.com/products/hardware/esp8266ex/overview/> [Stand: 26.09.2016]

Anzahl der maximal 16 nutzbaren *GPIO*-Pins (*General Purpose Input/Output*) und sind ebenfalls günstig erhältlich. Somit wäre ein Konzept für das Kostüm denkbar, welches mehrere dieser günstigen Chips an verschiedenen Stellen des Kostüms angebracht umfasst. Die LEDs des Anzuges könnten an die verschiedenen PWM-fähigen GPIO Pins angeschlossen werden. Für die Steuerschaltung der Leuchtschnüre wären einfache digitale Ausgänge ausreichend.

Vorteilhaft wäre bei dieser ersten Möglichkeit der Umsetzung das dezentrale Design, welches u. U. das An- und Ausziehen des Anzuges erleichtern könnte, weniger lange Drahtleitungen benötigen würde und den weiteren Vorteil hätte, dass bei einer Fehlfunktion der Kommunikation nicht das ganze Kostüm ausfiele. Außerdem müsste durch die Verwendung mehrerer solcher Schaltungen nicht an allen Stellen eine galvanische Trennung der Leuchtschnüre mit dem Rest der Schaltung erfolgen. Nachteilig wäre bei diesem Konzept, dass für jede dieser unabhängigen Schaltungen eine eigene Spannungsversorgung und letztlich auch eine jeweils eigene EL-Inverterschaltung (AC-Spannungsversorgung der Leuchtschnüre) notwendig wäre, falls der Anzug konsequent dezentral gestaltet wäre. Aufgrund dieser Nachteile wurde sich im Zuge dieser Arbeit gegen dieses Konzept entschieden.

Möglichkeit 2 - Verwendung eines A20 dual core Cortex-A7 Processors

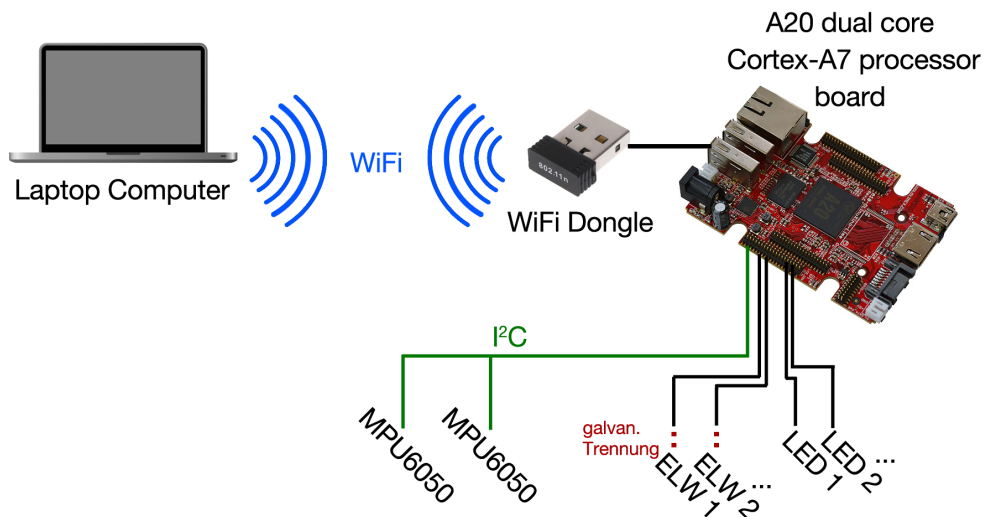


Abbildung 5.2.: Möglichkeit 2 zur Realisierung der Hardware

Die zweite Möglichkeit, einen leuchtenden Anzug umzusetzen, besteht in der Verwendung leistungstärkerer Mikrocontroller mit einer großen Anzahl von GPIOs, welche die Elemente des Anzuges zentral ansteuern. Ein Beispiel für einen hierfür passenden Mikrocontroller wäre die Verwendung eines *A20 dual core Cortex-A7 processor*. Dieser

5. Technische Realisierung

leistungsstarke Mikrocontroller liefert neben zwei CPU-cores in einer Ausführung der bulgarischen Firma *Olimex* namens *LIME* außerdem 160 GPIOs, die allesamt fähig sind, pulswellenmodulierte Ausgangssignale zu erzeugen. Zur drahtlosen Kommunikation mit dem Laptop könnte ein mit dem Mikrocontroller über USB verbundener WiFi-Dongle dienen.

Der Vorteil dieses Konzeptes (vgl. Abb. 5.2) wäre vor allem, dass durch den schnellen Prozessor eine Vielzahl an Aufgaben gleichzeitig erfüllt werden könnten. Sowohl die Kommunikation vom Laptop zum Anzug als auch die Sendung der Accelerometer-Daten der beiden Sensoren zum Laptop wären aufgrund der hohen Taktrate von 1 GHz pro CPU-Core ausreichend schnell. 160 GPIOs wären zur Ansteuerung der vielen LEDs und Leuchtschnüre mehr als ausreichend. Nachteilig ist zu diesem Konzept auch seitens des Stromverbrauchs des Prozessors nichts weiter zu bemerken. Aufgrund der Unkenntnis des Verfassers, wie ein solcher Mikrocontroller programmiert wird (meist wohl über ein *Linux*-Betriebssystem und der Programmiersprache *Python*), wurde sich zum Zeitpunkt der Entscheidung, im Zuge dieser Arbeit, gegen dieses Konzept und für die für Einsteiger zugänglichere *Arduino*-Plattform entschieden. Im Nachhinein kann nun gesagt werden, dass es sich damit höchstwahrscheinlich um eine Fehlentscheidung handelte. Dieses Konzept wäre im Endeffekt einfacher gewesen und wäre bei einem zukünftigen Projekt des Verfassers mit ähnlichen Anforderungen das Mittel der Wahl.

Möglichkeit 3 - Verwendung eines Arduino DUE

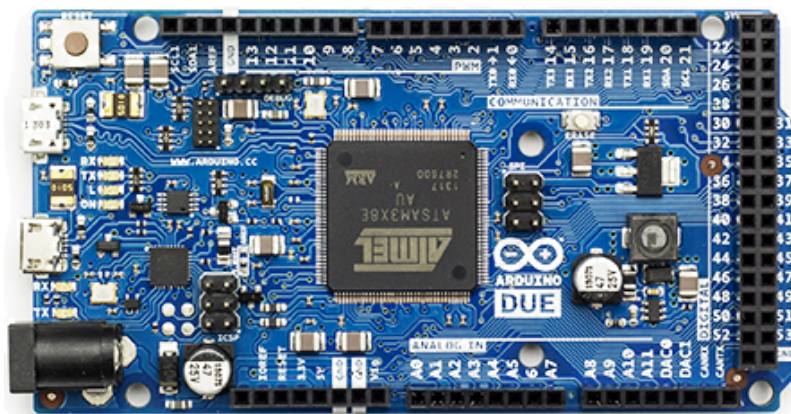


Abbildung 5.3.: Arduino-Board DUE

Die dritte Möglichkeit zur technischen Umsetzung des Projekts ist die Verwendung eines Mikrocontroller-Boards der *Arduino*-Plattform. Dieses Open-Source-Projekt kombiniert kompatible Hardware mit einer eigenen, leicht zu erlernenden Programmiersprache,

welche ein Dialekt von C++ ist. Zusätzlich steht den Nutzern eine spezielle Programmierungsumgebung mit einer großen Auswahl an vorgenerierten Libraries kostenfrei zur Verfügung¹⁴. Das bekannteste Arduino Board nennt sich *Arduino UNO*. Für dieses Board, welches auf dem Mikrocontroller *ATmega328P* basiert, ist die meiste Beispielsoftware erhältlich. Allerdings bietet es nur 16 Ein- und Ausgänge, von welchen nur sechs PWM-fähig sind und welche allesamt 5 V als logisches HIGH ausgeben. Eine Betriebsspannung von 5 V ist jedoch mit vielen Erweiterungschips (Funk, WiFi) nicht unbedingt kompatibel.

Ein größeres Board namens *Arduino DUE* (Abb. 5.3) bietet einerseits einen wesentlich schnelleren Mikrocontroller namens *Atmel SAM3X8E ARM Cortex-M3 CPU* mit einer Taktrate von 84 MHz. Andererseits bietet dieses Board mit 54 Digitalen Ein- oder Ausgängen, von welchen zwölf PWM fähig sind (bei einer Betriebsspannung von 3,3 V), genug Kanäle zur Realisierung des Anzugprojekts.

Auch der bereits erwähnte WiFi-fähige Chip *ESP8266* kann mittels der *Arduino IDE* programmiert werden. Hierfür bestehen u. a. bereits Beispielcodes für die Verwendung des WiFi-Chips als Erweiterung eines gewöhnlichen Arduino-Boards. Außerdem sind für die Verarbeitung von OSC-Messages (*Open Sound Control*, ein Art Nachfolger von MiDi) über WiFi bereits Codebeispiele vorhanden.

Der große Vorteil der Verwendung der *Arduino*-Plattform liegt in der Einfachheit. Durch das Vorhandensein von sehr viel Beispielsoftware und Online-Tutorials ist die Verwendung dieser Plattform eine große Hilfe und ein großes Zeitersparnis für die Umsetzung dieses Projektes. In diesem Falle wäre ein Konzept mit einem *Arduino DUE* als zentrale Steuereinheit des Anzuges anzustreben. Vorteilhaft hierbei wäre, dass zur Erzeugung der Wechselspannung für die Leuchtschnüre nur eine EL-Inverterschaltung notwendig ist. Nachteilig an diesem Konzept ist eine verringerte Performance der mittels Arduino programmierten Chips sowie einer gewissen Komplexität des Aufbaus: Viele verschiedene Elemente wie Funk- und WiFi-Chips müssen hierbei mit dem *Arduino* kommunizieren. Ein weiterer Nachteil besteht in der Beschränktheit der Anzahl von Ein- und Ausgängen der *Arduino* Boards, welche im Falle des *DUE* jedoch für dieses Projekt gerade ausreichend vorhanden wären.

Zur Umsetzung der Kommunikation des Anzuges mit dem Laptop bestehen drei weitere Optionen, die nun ebenfalls diskutiert werden:

1. Kommunikation über WiFi: Die erste Option wird in Abb. 5.4 dargestellt. In diesem Fall wird ein *ESP8266*-WiFi-Chip über die serielle Schnittstelle mit dem *Arduino* verbunden. Der *ESP8266* würde hierbei die vom Laptop gesendeten OSC-Lichtdaten über die

¹⁴vgl. <https://www.arduino.cc/en/Guide/Introduction> [Stand: 26.09.2016]

5. Technische Realisierung

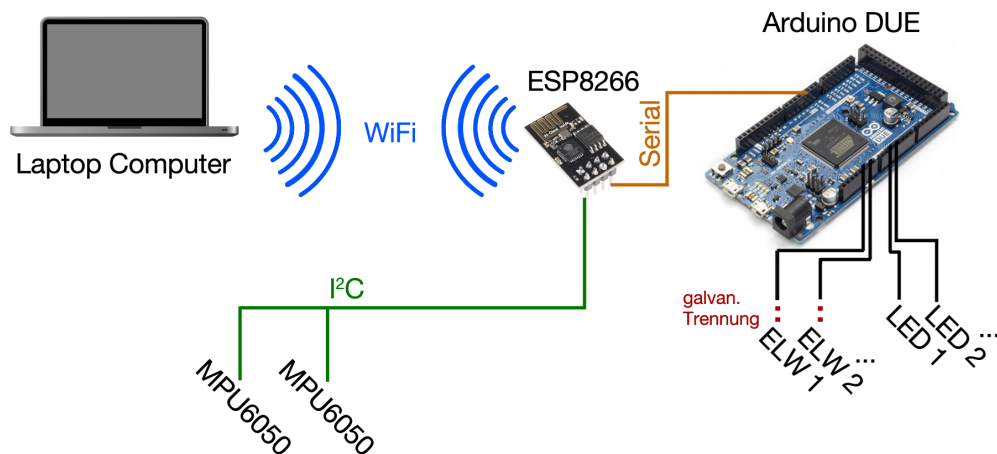


Abbildung 5.4.: Möglichkeit 3 zur Realisierung der Hardware, Kommunikation über WiFi

serielle Schnittstelle an den *DUE* übertragen, welcher diese dann als Steuerspannungen an die Leuchtschnüre und LEDs weitergibt. Der Wifi-Chip würde dabei zusätzlich die Verwaltung der beiden Accelerometer *MPU6050* übernehmen. Diese könnten über eine I²C-Schnittstelle angeschlossen werden und ihre Daten ebenfalls als OSC-Pakete an den Laptop liefern.

Diese Möglichkeit wurde im Zuge des Projekts testweise umgesetzt, jedoch konnte keine zufrieden stellende Performance der Lichter erzielt werden. Einerseits konnten keine ausreichend hohen Schaltfrequenzen der Lichter erzeugt werden. Andererseits war die Verwendung des ESP8266 als Access Point instabil. Aufgrund dessen und mangels ausreichender Debug-Kenntnisse des Verfassers bzgl. dieser Probleme wurde nach einer anderen und schnelleren Option für die Kommunikation des *DUE* mit dem Laptop gesucht.

2. Kommunikation über zwei Arduinos und zwei nRF24-Funkchips: Die zweite Möglichkeit, um die Kommunikation umzusetzen, besteht über die Verwendung zweier *Arduinos*. Dabei wird einer für den Anzug verwendet und ein zweiter als Sende- und Empfangseinheit via USB an den Laptop angeschlossen. Die Kommunikation kann dann über eine digitale ISM-Funkstrecke erfolgen (vgl. Abb. 5.5). Der Vorteil hierbei ist die Tatsache, dass eine solche Funkstrecke sich nicht mit einem Netzwerk verbinden muss, und somit eine solche Verbindung, im Falle dass sie ausfällt (z. B. durch eine übertretene Reichweite), sofort wieder funktioniert, sobald die Übertragungsstrecke wieder hergestellt ist. Außerdem ist die Performance einer solchen Verbindung in Bezug auf die Latenz und die Menge der übertragbaren Daten sehr zufriedenstellend, solange sie in eine Richtung geht. Als Funkchips kamen im Zuge dieser Projektarbeit zwei Exemplare des *nRF24L01+* des norwegischen fabless Halbleiterherstellers *Nordic*

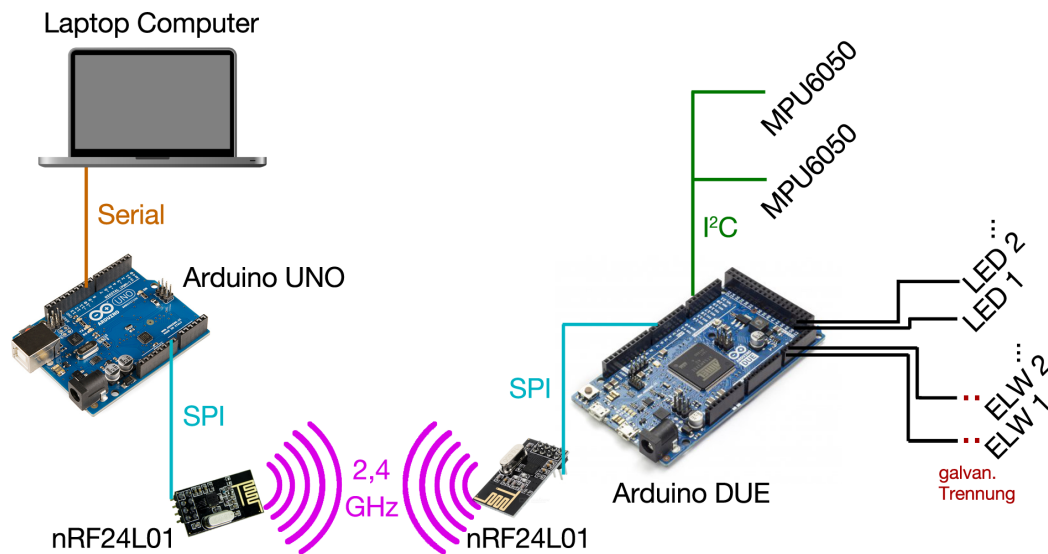


Abbildung 5.5.: Möglichkeit 3 zur Realisierung der Hardware, Kommunikation über 2,4 GHz ISM-Digitalfunk

Semiconductor zur Anwendung, der über das 2,4-GHz-ISM-Frequenzband (*Industrial, Scientific and Medical*) überträgt. Der Vorteil dieses Chips ist die Tatsache, dass er extrem stromsparend ist und somit für batteriebetriebene Anwendungen sehr gut geeignet ist¹⁵. Bei einer preisgünstigen Variante des *nRF24L01+* (ca. 1 Euro pro Board) werden neben der Antenne acht Pins des Chips auf einem kleinen Board ausgeführt, welche über eine SPI-Verbindung (*Serial Peripheral Interface*) mit dem jeweiligen *Arduino* verbunden werden können. Für die *Arduino IDE* sind zur Verwendung dieses Funkchips bereits Codebeispiele vorhanden, welche die Kommunikation zweier *Arduinos* herstellen.

Bei dem hier vorgestellten Konzept müssten die Accelerometer über den I^2C -Bus direkt am *DUE* angeschlossen werden welcher die Daten an den Funkchip weiterleiten würde. Der Nachteil dieser Variante liegt allerdings in der Tatsache, dass bei einem *Full-Duplex*-Betrieb (Senden und Empfangen) der *nRF24L01+*-Verbindung, beide *Arduinos* ständig zwischen Sender und Empfänger hin- und herschalten müssen. Dies bewirkt eine große Verringerung der Performance bei der Übertragung. Die Geschwindigkeit stellte sich nach einigen Tests im Zuge dieser Arbeit als nicht zufriedenstellend heraus, weswegen eine weitere Strategie gefunden werden musste.

3. Kommunikation über zwei Arduinos, zwei nRF24L01+-Funkchips als Hin- und einen ESP8266 Wifi-Chip als Rückweg: Um die Geschwindigkeit der Kommunikation zwischen Anzug und Laptop hoch und die Fehlerrate dabei niedrig zu halten,

¹⁵Vgl. <http://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24L01+> [Stand: 26.09.2016]

5. Technische Realisierung

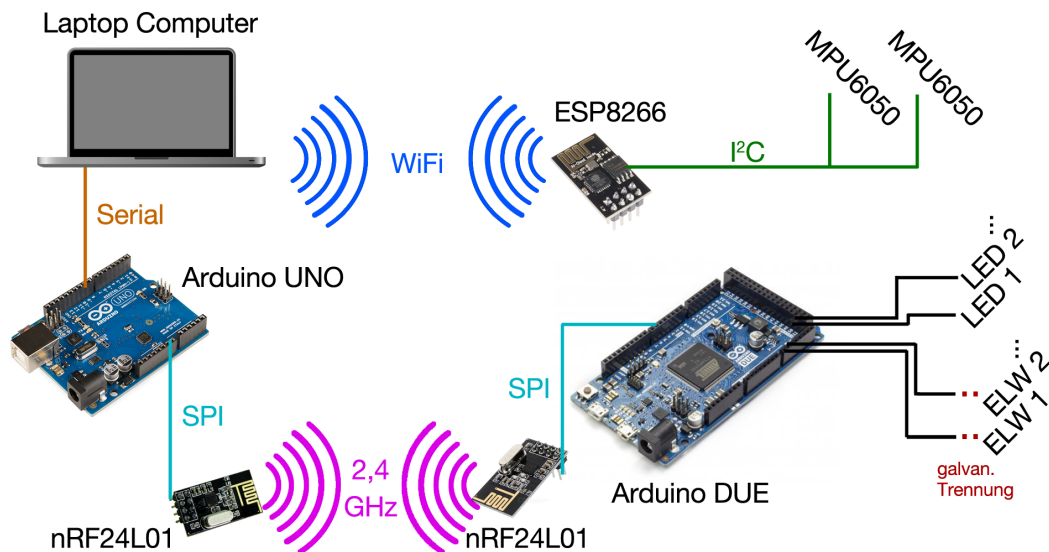


Abbildung 5.6.: Möglichkeit 3 zur Realisierung der Hardware, Kommunikation über 2,4 GHz Digitalfunk und Rückweg über WiFi

bietet sich die Verwendung mehrerer mit dem Anzug verbundener Mikrocontroller an. Auf diese Weise kann der Sendeweg von Laptop zum Anzug sich ungestört der einzigen Aufgabe widmen, die Steuersignale durch den *Arduino DUE* in an- und ausgeschaltetes Licht umzusetzen. Die beste Möglichkeit, den Rückweg der Accelerometerdaten zum Laptop zu herzustellen, wäre es dann, eine zweite Funkstrecke dafür zu etablieren. So wäre nun eine zweite ISM-Funkübertragung denkbar. Alternativ erscheint jedoch eine WiFi-Verbindung mithilfe eines *ESP8266*, welche die Accelerometerdaten als OSC-Pakete autark überträgt als eine elegante Lösung (vgl. Abb. 5.6). Da diese Übertragung im Rückweg zum Laptop in der hier vorliegenden Anwendung nicht das Ausmaß der Schnelligkeit des Hinwegs benötigt (der I²C-Bus ist in Sachen Übertragungsgeschwindigkeit ohnehin limitiert), können die oben Angesprochenen Performanceeinbrüche der OSC-Übertragung mittels des *ESP8266* verkraftet werden. Durch die geringe Größe einer solchen Ausführung des Mikrocontrollers, nimmt dieser auch keinen unnötigen zusätzlichen Raum auf dem Anzug ein.

Der Nachteil dieses Konzeptes liegt an einer relativ hohen Komplexität des Aufbaus und der Verwendung mehrerer Übertragungsstrecken in der Praxis. So muss im Anwendungsfall des leuchtenden Kostümes sowohl ein zweiter Arduino an den Laptop angeschlossen, als auch eine WiFi-Verbindung (evtl. mit Zuhilfenahme eines WiFi-Routers) eingerichtet werden. Trotz dieses zusätzlichen Aufwandes wurde sich im Zuge dieser Projektarbeit für diese Strategie entschieden, da sie die beste Performance der hin- und rückwärtigen Übertragung aufweist (abgesehen von der Verwendung eines schnelleren Mikrocontrollerboards wie des *A20 dual core Cortex-A7* Prozessors, was aber

nicht umgesetzt wurde).

5.1.2. Energieversorgung und Leistungsbilanz der Elemente und Mikrocontroller

Nachdem nun das zu verfolgende Hardwarekonzept des Anzugprojektes feststeht, kann sich dieser Abschnitt nun der Umsetzung dieser Strategie widmen.

Als erstes ist zur Umsetzung entscheidend, wie groß die elektrische Leistungsaufnahme des leuchtenden Kostüms in der Anwendung ausfällt. So kann im Vorfeld ermittelt werden, welche Dimensionen die Energieversorgung des Anzuges ungefähr aufweisen muss.

Tabelle 5.1 stellt die wesentlichen elektrischen Bauteile des Anzuges mitsamt der benötigten Spannung sowie der Strom- und Leistungsaufnahme zusammen. Dabei wurden die typischen Werte übernommen, welche für diese Bauteile in der Literatur zu finden sind. Ausgelassen wurden dabei die Leuchtschnüre und die LEDs.

Bauteil	Stromaufnahme [mA]	Leistungsaufnahme [W]	Anmerkungen
<i>Arduino DUE</i>	90	0,45	durchschnittlicher Gebrauch
<i>ESP8266</i>	170	0,561	TX 802.11b
<i>nRF24Lo1+</i>	13,5	0,045	RX+TX, max. Leistung
2x <i>MPU6050</i>	1	0,003	-
Gesamt	274,5	1,059	ohne Leuchtschnüre/LEDs

Tabelle 5.1.: Leistungsbilanz der verwendeten Bauteile ohne Leuchtschnüre und LEDs

Ohne Berücksichtigung der Lasten durch die Lichtelemente liegt also eine Leistungsaufnahme von insgesamt ca. einem Watt vor. Bei der Verwendung eines typischen Akkus mit 5 V Ausgangsspannung und einem Ausgangsstrom von 1 A, blieben also noch 4 W Leistungsaufnahme für die Lasten durch die Lichtelemente übrig. Die verwendeten hellblauen Leuchtschnüre mit 5 mm Durchmesser der Firma *EL Arts* weisen laut Datenblatt eine Leistungsaufnahme von 3,2 mW/cm auf. Der DC-DC-Wandler *IB0505S-2W*, welcher zur galvanischen Trennung der EL-Inverterschaltung verwendet wird (siehe Abschnitt 5.1.3), weist laut Angabe des Herstellers eine „Effizienz“ (gemeint ist wohl der Wirkungsgrad) von 70% auf. Außerdem beträgt seine maximale Ausgangsleistung 2 W. D.h. es können im normalen Betrieb des Anzuges nach folgender Berechnung

$$x = \frac{2 \text{ W}}{3,2 \text{ mW/cm}} = 625 \text{ cm}, \quad (5.1)$$

5. Technische Realisierung

also 6,25 m Leuchtschnur sichergestellt werden. In der Praxis zeigt sich jedoch, dass dieser Wert ohne Weiteres überschritten werden kann. Nach einigen Tests wurde ermittelt, dass beim Anschluss von 15 m Leuchtschnur keine wesentliche Verringerung der Leuchtkraft zu sehen ist. Die in Gleichung 5.1 getätigte Rechnung ist also zu streng. Womöglich ist die Leistungsaufnahme der Leuchtschnüre wohl doch geringer als es das Datenblatt angibt oder aber der DC-DC-Wandler ermöglicht eine höhere Ausgangsleistung als die angegebenen 2 W. Im Falle dass letzterer Wert korrekt ist, ergibt sich für die galvanisch getrennte EL-Inverterschaltung mit Last die insgesamt maximale Leistungsaufnahme von

$$P_{\text{EL, ges}} = P_{\text{out, DC-DC}} \cdot \frac{1}{\eta_{\text{DC-DC}}} = 2 \text{ W} \cdot \frac{1}{0,7} = 2,857 \text{ W}, \quad (5.2)$$

Betreffend der Leuchtdioden wird pro Bündel a vier LEDs ein 220Ω -Widerstand vorgeschalten. Also ergibt sich für ein LED-Bündel jeweils eine Leistungsaufnahme von

$$P_{4\text{LEDs}} = U \cdot I = \frac{U^2}{R} = \frac{(3,3 \text{ V})^2}{220 \Omega} = 0,0495 \text{ W} \quad (5.3)$$

Bei dem Anzugprojekt werden insgesamt 12 dieser Bündel verwendet. Also ist für die Leistungsaufnahme aller 48 LEDs insgesamt mit 0,594 W zu rechnen. Sind also alle LEDs in Betrieb, bleiben für die gleichzeitige Verwendung der Leuchtschnüre noch ca. 3,5 W Leistung übrig. Nach Gleichung 5.2 reicht dieser Wert also aus so dass ein 5 V Akku mit einem maximalen Ausgangsstrom von 1 A für den vollen Betrieb aller leuchtenden Elemente eingesetzt werden kann.

Um einen 5 V Akku mit 1 A Ausgangsstrom bereitzustellen, wäre es eine Möglichkeit, einen handelsüblichen Lithium-Polymer-Akku mit einer zugehörigen Ladeschaltung als Spannungsversorgung des Kostüms einzuplanen. Noch einfacher gestaltet sich jedoch die Verwendung einer sogenannten *Powerbank*. Diese Geräte dienen als handliche, tragbare Ladestationen mobiler 5-V-Geräte wie Smartphones etc. und haben bereits einen Ladecontroller und eine Ladestandsanzeige integriert. Die verwendete 5-V-Powerbank mit maximal 1 A Ausgangsstrom schaltet sich dabei von selbst ein, sobald eine Last angeschlossen wird.

5.1.3. Verwendete Schaltung

Abbildung 5.7 zeigt die gesamte verwendete Schaltung auf dem leuchtenden Anzug.

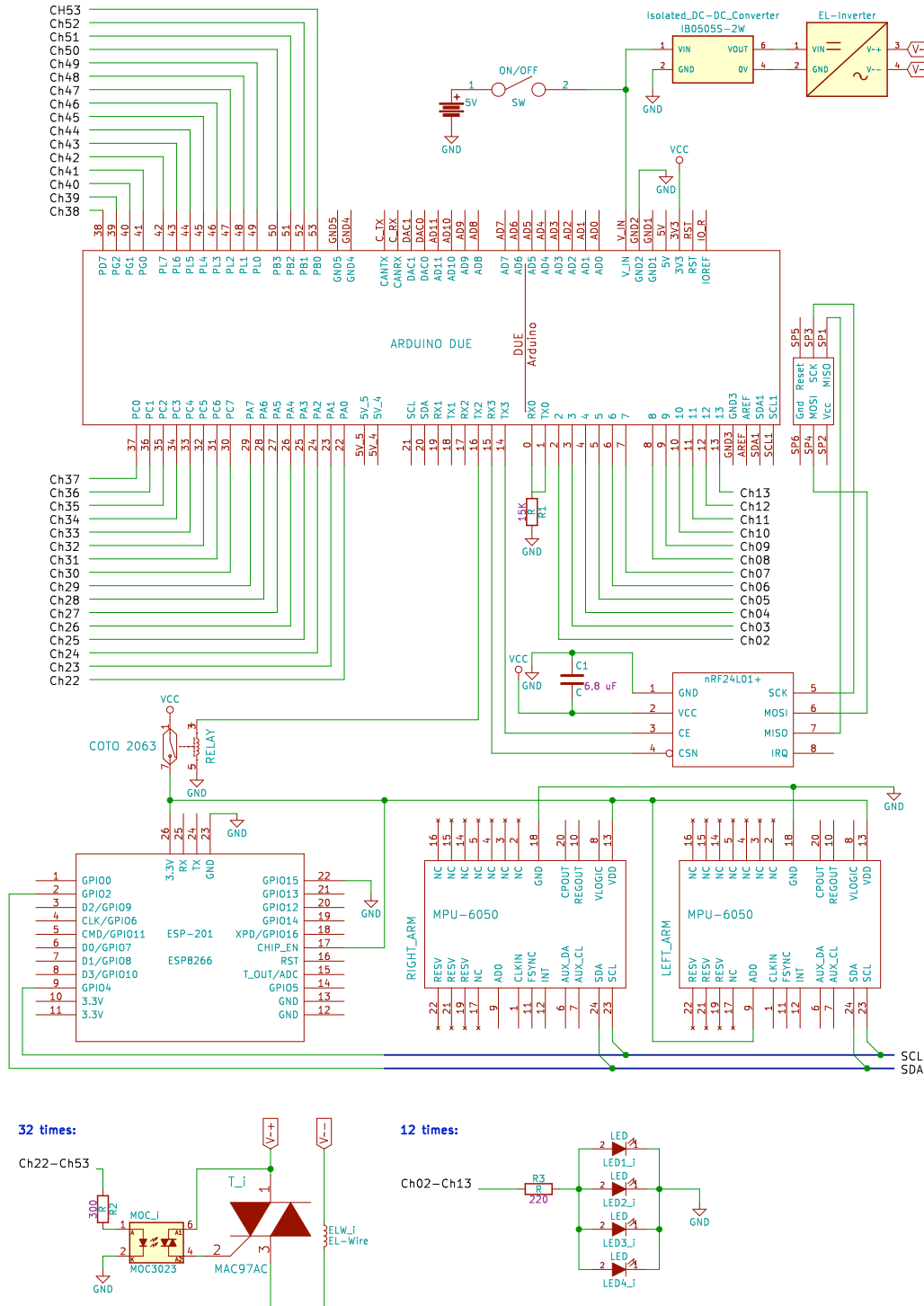


Abbildung 5.7.: Gesamte verwendete Schaltung des Anzuges

5. Technische Realisierung

Im Anschluss werden die einzelnen Komponenten dieses Schaltplans im Detail besprochen. Abb. 5.8 zeigt den physikalischen Aufbau dieser Schaltung, wie er im Zuge des Projektes umgesetzt wurde.

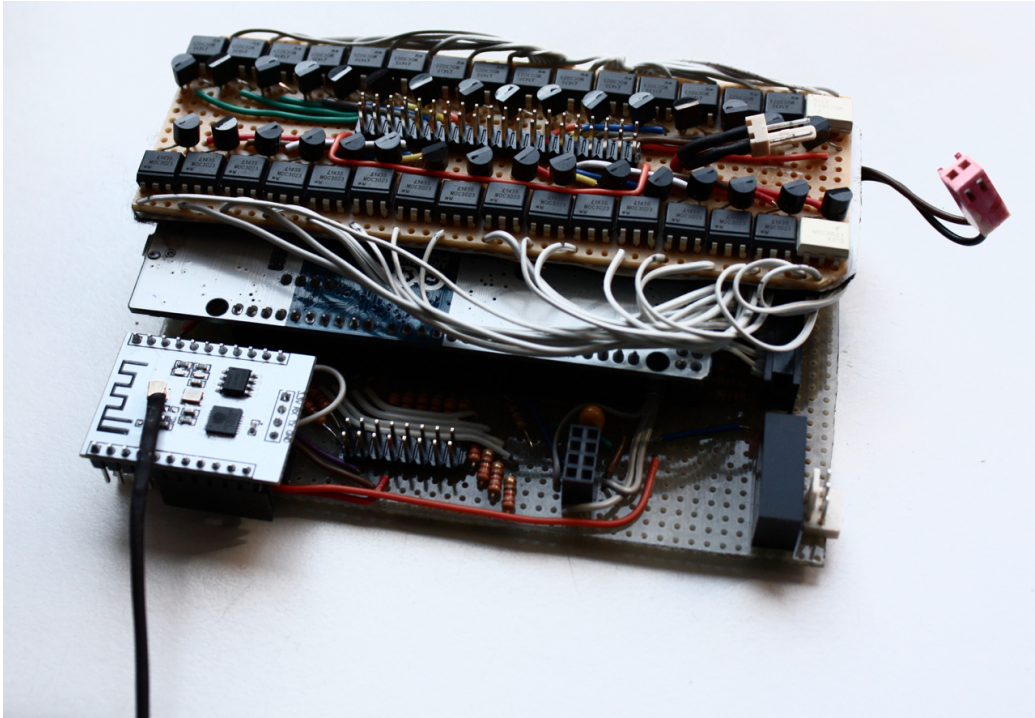


Abbildung 5.8.: Schaltung des Anzuges in der physikalischen Ausführung des Projektes ohne EL-Inverter, Accelerometer und ohne die leuchtenden Elemente

Zunächst widmet sich dieser Abschnitt der Erzeugung der Wechselspannung für die Leuchtschnüre.

EL-Inverter-Schaltung und galvanische Trennung

Für einen sogenannten EL-Inverter, also eine Schaltung, welche aus einer niedrigen Gleichspannung (in unserem Fall 5 V) eine Wechselspannung mit einem hohen Effektivwert (in unserem Fall 150 V) bereit stellt, wird generell eine sogenannte Sperrschwingerschaltung realisiert. Eine solche Schaltung dient als Impulserzeuger und generiert dabei eine hohe Impulsleistung, kurze Anstiegszeit, niedere Impedanz und einen guten Wirkungsgrad. Ein Sperrschwinger besteht häufig aus der Kombination eines Übertragers mit einem Transistor und wird in Abb. 5.9 dargestellt¹⁶.

¹⁶Vgl. [SPEISER; 1963], S. 196f

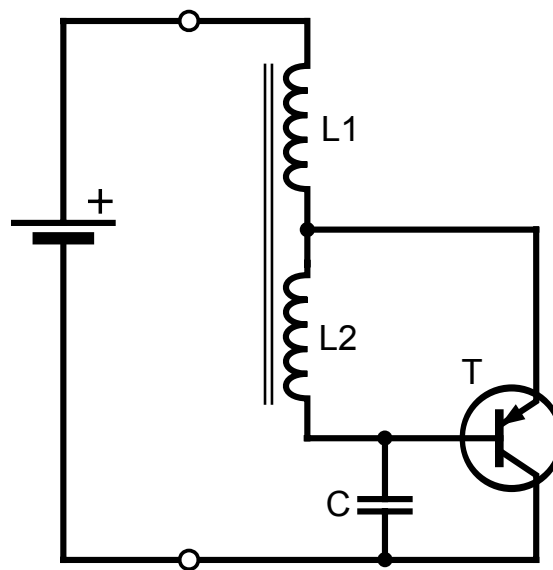
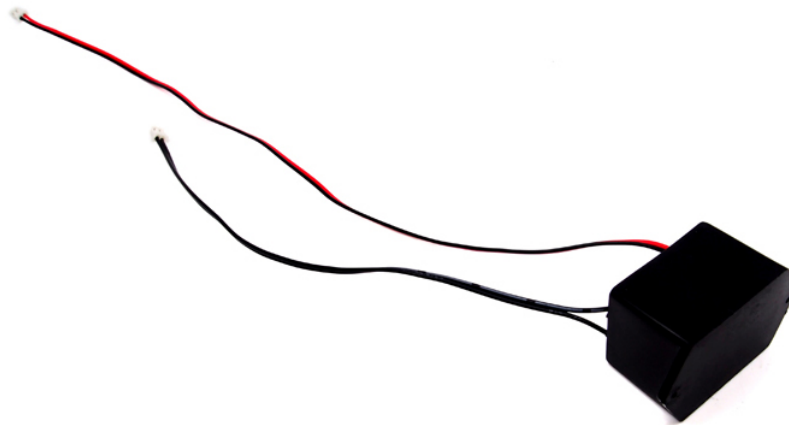


Abbildung 5.9.: Schaltung eines Sperrschwingers [DVORAK; 2011]

Die chinesische Firma *seedstudio* vertreibt eine effiziente EL-Inverterschaltung für eine Eingangsspannungsspannung von 5 V, welche nach Angaben des Herstellers für maximal 15 m Leuchtschnur ausgelegt ist¹⁷. Offiziell beträgt die Ausgangsspannung dieses Bauteils 110 V. In einer Messung mittels eines Oszilloskops wurde jedoch ein Effektivwert von 205 V ohne, und 155 V mit 15 m Leuchtschnur als Last bestimmt. Abb. 5.10 zeigt diesen EL-Inverter für den kein Datenblatt erhältlich ist.

Abbildung 5.10.: 5 V DC zu 110 V AC EL-Inverter der Firma *seedstudio*

Für das leuchtende Kostüm kommt dieser EL-Inverter zusammen mit einem vorgeschalteten DC-DC-Wandler zum Einsatz, welcher eine galvanische Trennung der Hochspan-

¹⁷Vgl. www.seedstudio.com/Robust-EL-Inverter-For-EL-Shield-p-1289.html [Stand 28.09.2016]

5. Technische Realisierung

nung zur restlichen Schaltung ermöglicht (siehe Abb. 5.7 oben rechts). Aufgrund einer Messung mit einem Multimeter ist es offensichtlich, dass die galvanische Trennung durch den Übertrager des EL-Inverters nicht gewährleistet ist, da womöglich eine gemeinsame Masse genutzt wird. Durch den zusätzlichen isolierten DC-DC-Wandler wird eine erhöhte Sicherheit für die Trägerin des Kostüms sichergestellt, welche bei den auftretenden Spannungen in der Größenordnung von EL-Invertern unbedingt gewährleistet sein sollte.

Schaltung der Leuchtschnüre und galvanische Trennung

Der Ausgang des EL-Inverters wird mittels zweier Leitungen zu einer eigenen Platine übergeben (siehe Abb. 5.11), welche diese Spannungen an die Leuchtschnüre verteilt.

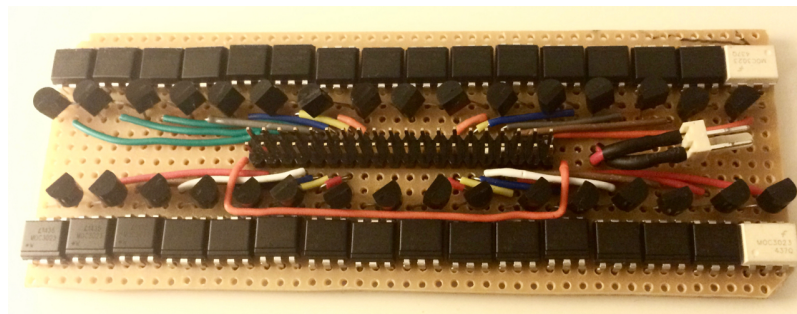


Abbildung 5.11.: Schaltung zur Verteilung der Wechselspannung an die Leuchtschnüre

Durch diese Schaltung (Abb. 5.7 unten links) werden für insgesamt 32 Kanäle für Leuchtschnüre ein Schaltmechanismus mit galvanischer Trennung bereitgestellt. Hierfür werden 32 Ausführungen des Optoisolators *MOC3023* über einen jeweiligen Vorwiderstand an die digitalen Ausgänge 22 bis 53 des *Arduino DUE* angeschlossen. Die Ausgänge der Optoisolatoren werden mit 32 *MAC97AC*-Triacs verbunden, welche die Wechselspannung des EL-Inverters auf die verschiedenen Leuchtschnüre schalten. Die Verwendung von Triacs anstatt MOSFETs vereinfacht die Schaltfunktion einer Wechselspannung, da im Falle eines Transistors bei einer solchen Anordnung keine negativen Spannungen übertragen werden könnten und ein DC-Offset bereitgestellt werden müsste.

Die separat ausgeführte Platine aus Abb. 5.11 verbindet alle ausgangsseitigen Anschlüsse an 40 Pins in zwei Reihen, sodass mithilfe einer IDE-Buchse die Verbindung zu den Leuchtschnüren erfolgen kann. Eingangsseitig werden über ein Flachbandkabel die Steuersignale des *Arduino DUE* mit dem Board verbunden. Um eine Isolation dieser Platine zum sich darunter befindlichen *Arduino DUE* zu ermöglichen, wird an der kompletten Unterseite ein Stück Filz mit einem doppelseitigen Klebeband befestigt (vgl. Abb. 5.8).

Schaltung der Bewegungssensoren und der Kommunikation zum Laptop

Die Verwaltung der Bewegungssensordaten und die Kommunikation zum Laptop wird vom *ESP8266* in der Ausführung namens *ESP-201* übernommen (vgl. Abb. 5.7 und 5.8, jeweils links unten). Der Grund für dieses spezielle Breakout-Board des *ESP8266* liegt an einem hierfür erhältlichen, speziellen Entwicklungsboard mit einem USB-zu-Serial-Interface und mehreren Jumpern zum Setzen der Pins auf VCC oder GND, durch welche es erheblich erleichtert wird, das *ESP-201*-Board zu programmieren. Außerdem verfügt dieses Board über eine externe Antenne, welche ggfs. ausgerichtet werden kann um den WiFi-Empfang des Chips zu verbessern.

Der Anzug wird in einer Auftrittssituation in einem Backstageraum angezogen und dort bereits eingeschaltet. Damit das ESP-Board erst in Betrieb genommen wird, sobald der Anzug in Reichweite ist, wird die Versorgungsspannung über ein kleines Relais, dem *COTO 2063* von Pin 16 des *Arduino* mittels eines Funksignals vom Laptop angeschalten. An der selben geschalteten Versorgungsspannung von 3,3 V sind außerdem die beiden Accelerometerchips *MPU6050* angeschlossen. Letztere werden über den I²C-Bus mit dem *ESP8266* verbunden (*SDA* und *SCL* liegen an den Pins *GPIO2* und *GPIO4*). Um die beiden Sensorchips unterscheiden zu können, benötigen die Chips unterschiedliche Adressen. In bestimmten Breakout-Boards des *MPU6050* kann die Adresse durch Verbindung des sog. *AD0*-Pins auf HIGH geändert werden. Im Zuge dieser Arbeit wurde beim Sensor des linken Armes dieser Pin mittels einer Litze direkt auf dem Board mit 3,3 V verbunden (vgl. Abb. 5.12).

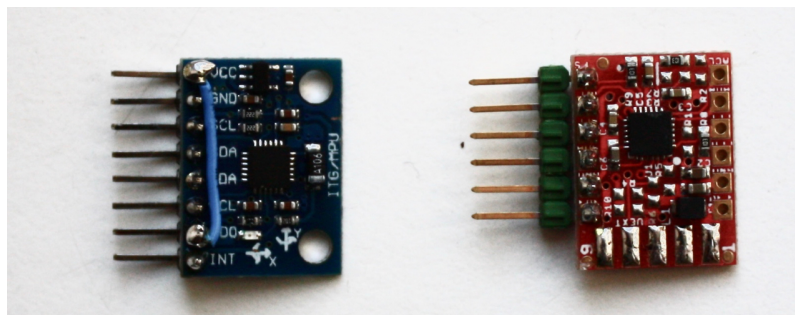


Abbildung 5.12.: Die beiden Accelerometerchips *MPU6050* mit unterschiedlichen Breakout-Boards für den linken (links) und den rechten Arm (rechts).

Die Pull-Up-Widerstände für den I²C-Bus sind in dem Breakout-Board des *MPU6050* am linken Arm intern ausgeführt, sodass diese in der Schaltung aus Abb. 5.7 nicht berücksichtigt werden müssen. Die Pins *GPIO15* und *CHIP_EN* des *ESP-201*-Boards müssen mit GND bzw. mit 3,3 V verbunden werden, damit der *ESP8266* fehlerfrei starten kann, sobald die Versorgungsspannung anliegt.

5. Technische Realisierung

Damit die Verbindung zum Laptop stabil läuft, wird zur Übertragung ein WiFi-Router hinzugezogen. Zwar kann durch den *ESP8266* auch ein Access-Point bereitgestellt werden, jedoch erweist sich dies in der Praxis als instabil und verringert die Reichweite der WiFi-Übertragung. Es wird hierfür ein verstecktes Netzwerk mit Passwort eingerichtet, bei welchem dem Laptop und dem *ESP8266* feste IP-Adressen zugeteilt werden.

Schaltung der LEDs

Für die LEDs des Anzuges werden 3 mm große Exemplare „reinweißer“, Farbe verwendet. Diese erzeugen jeweils ein Licht der Stärke 12 – 14 cd und sind für eine Spannung von 3,3 V ausgelegt. Im Vergleich zu vergleichbaren 5-mm-Exemplaren erschien der Abstrahlwinkel der kleineren Exemplare als etwas größer, weshalb sich für diese entschieden wurde. Als Vorwiderstand werden für vier parallel geschaltene LEDs ein 220 Ω -Widerstand verwendet und jeweils an einen der 12 PWM-Ausgänge des *Arduino DUE* angeschlossen. Die somit erzeugte Lichtstärke entspricht dem, was sich der Verfasser der Arbeit nach ästhetischen Maßstäben vorstellt. Außerdem können so bei einem PWM-Verhältnis von 1 (Maximum: 255), also der geringsten einstellbaren Leuchtstärke, ausreichend dunkle Leuchtstärken erzielt werden. Der Widerstandswert wurde also nach ästhetischen Maßstäben empirisch bestimmt.

Schaltung der Kommunikation des Arduino DUE

Die Kommunikation des *Arduino DUE* mit dem Laptop erfolgt wie bereits erwähnt über einen 2,4-GHz-ISM-Funkchip namens *nRF24Lo1+*. Dieser wird über die SPI-Schnittstelle des *Arduino* angeschlossen (vgl. Abb. 5.7, mittig rechts). Als *CE*- und *CSN*-Pin des *DUE* werden dabei die Pins 14 und 15 definiert. Um die Stabilität der Spannungsversorgung des Chips zu erhöhen, bietet es sich an, einen 3,3 – 10 μ F Kondensator zwischen VCC und GND zu schalten¹⁸. In der Umsetzung dieses Projektes kommt hierfür ein 6,8 μ F Keramikkondensator zum Einsatz.

Bei einer ersten Inbetriebnahme der gesamten Schaltung wurde festgestellt, dass der Chip zu nahe am EL-Inverter platziert wurde. Aus diesem Grund wird eine zusätzliche Verlängerung zwischen Sockel und Chip eingefügt, um den *nRF24Lo1+* aus dem elektromagnetischen Störfeld des EL-Inverters zu entfernen (vgl. Abb. 5.13 rechts).

¹⁸vgl. <https://arduino-info.wikispaces.com/Nrf24Lo1-2.4GHz-HowTo#lib> [Stand 28.09.2016]

Schaltung der Kommunikation des Arduino UNO

Die Sendeeinheit der Kommunikation zwischen Laptop und dem *Arduino DUE* wird durch einen über USB mit dem Rechner verbundenen *Arduino UNO* bereitgestellt. An diesem wird ebenfalls ein *nRF24Lo1+*-Chip über die SPI-Schnittstelle angeschlossen. Dabei ist der Aufbau zu dem des *DUE* in Abb. 5.7 identisch, mit der Ausnahme, dass für *CE*- und *CSN* die Pins 7 und 8 verwendet werden. Die Eingänge des *nRF24Lo1+* sind dabei 5-V-tolerant, d.h. es müssen keine zusätzlichen Spannungsteiler hinzugefügt werden, da der *Arduino UNO* wie bereits erwähnt einen auf 5 V basierten *ATmega328P* als Mikrocontroller verwendet.

5.1.4. Zusammenführung der Teile des Kostüms

Nachdem alle Teile der Schaltung *in vitro*, einzeln getestet und verbunden funktionieren, dürfen sie auch an das Kostüm angebracht nicht in ihrer Funktion gestört sein. Als Basis für den Anzug kommt ein geschlossener, schwarzer und eng anliegender Tanzanzug aus 80% Polyamid und 20% Elasthan zum Einsatz. An seiner Vorderseite befindet sich ein Reißverschluss um das An- und Ausziehen zu ermöglichen. Dadurch ist an der Rückseite Platz für die Elektronik des Kostüms. Die in Abb. 5.8 gezeigte Schaltung muss nun also an den Anzug am Rücken angebracht werden. Hierfür kommt eine handelsübliche Plastik-Brotbox zum Einsatz, welche mittels einiger angebrachter Nylonschlaufen und einem Schürsenkel an einem Stück Filz befestigt wird, welches auf den Tanzanzug genäht wird (siehe Abb. 5.13 links). Damit keine der Kontrollleuchten der Schaltung die leuchtende Wirkung des Anzuges beeinträchtigt, wird diese Box zusätzlich mit matter, schwarzer Acrylfarbe versehen.

Zusätzlich zu der bereits erwähnten Schaltung werden der EL-Inverter und die Powerbank in die Brotbox integriert. Ein An-/Ausschalter und eine Ladebuchse für die Powerbank werden ebenfalls in das Gehäuse integriert (siehe Abb. 5.13 rechts). Damit der Anzug theoretisch zu reinigen ist, müssen alle empfindlichen Elektronikteile entfernt werden können. Deshalb sind alle Anschlüsse der leuchtenden Elemente über IDE-Stecker mit der Schaltung verbunden, so dass das Rückenteil komplett entfernt werden kann. Neben dem bereits erwähnten IDE-Stecker für die Leuchtschnüre (inkl. deren Masse) befinden sich an einem zweiten Stecker (blau in Abb. 5.13 rechts) alle Niedervolt-Anschlüsse, also die I²C-Anschlüsse der *MPU6050*-Chips und die LED-Anschlüsse. Der ausgelagerte *nRF24Lo1+*-Chip wird in eine kleine schwarz angemalte Filmdose gegeben und mit einem CAT5-Multicorekabel verbunden, welches an dem 8-Pin-Sockel auf der Schaltung (vgl. Abb. 5.8 und 5.13 unten rechts) angesteckt werden kann.

5. Technische Realisierung

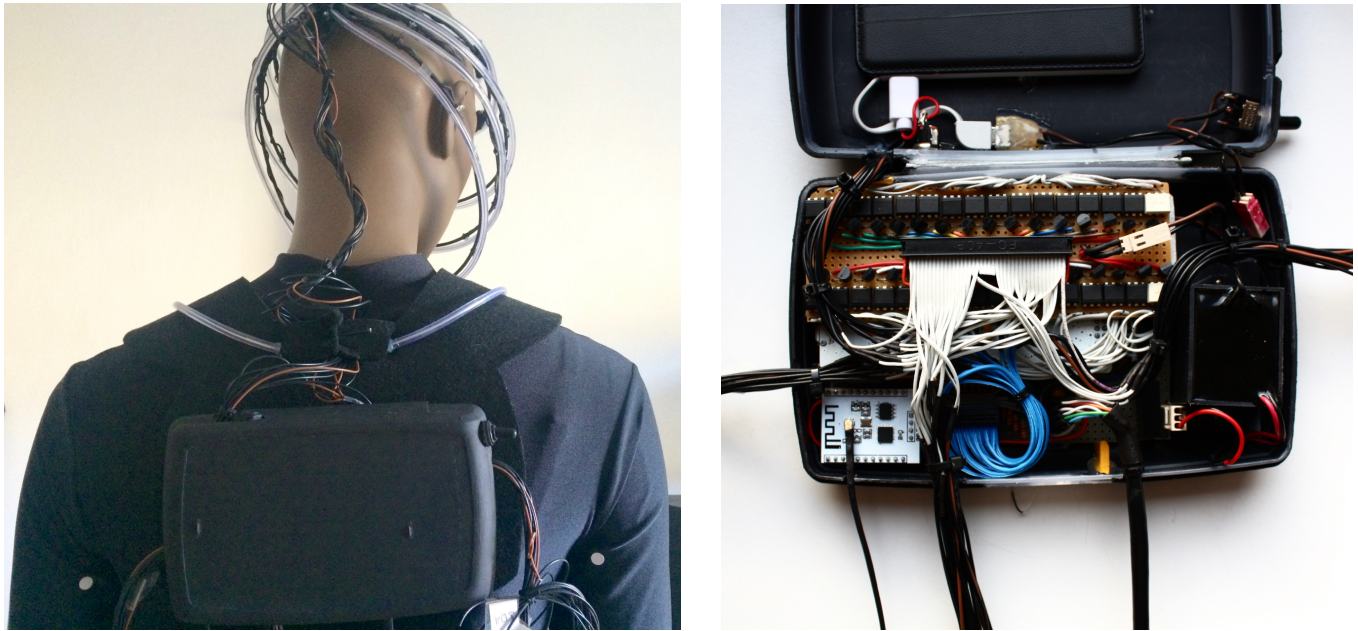


Abbildung 5.13.: **Links:** Anbringung der Schaltung am Rücken des Tanzanzuges in einer schwarz angemalten Plastik-Brotbox, **Rechts:** Innenleben des Brotbox-Gehäuses

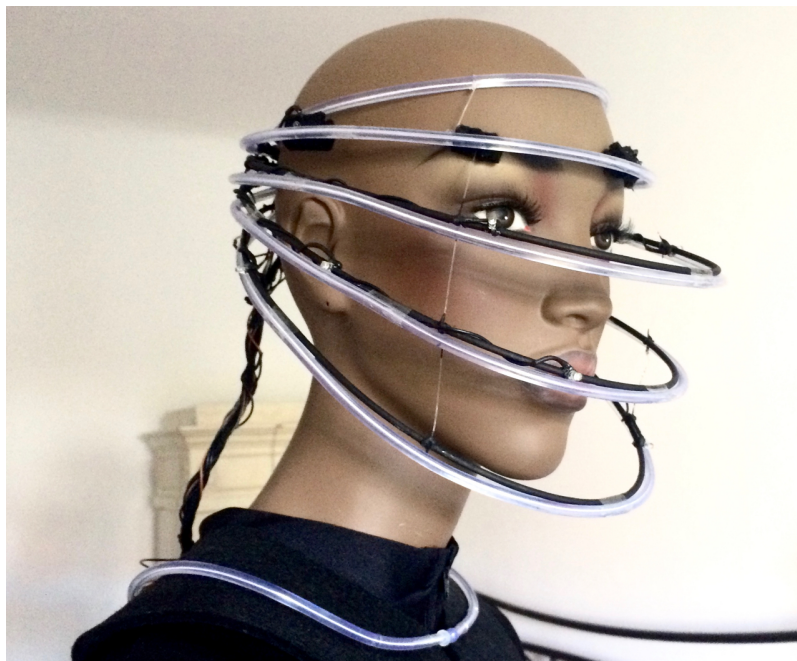


Abbildung 5.14.: Kopfteil des Anzuges mit 12 LEDs und fünf Leuchtschnur-Ringen

Für das Kopfteil werden insgesamt fünf Leuchtschnur-Ringe ausgeführt, von welchen vier an Metallringe angebracht werden. Diese vier Stahlringe dienen außerdem als

Möglichkeit zur Befestigung der 12 LEDs (genau genommen drei Kanäle a vier LEDs). Damit die Masse nicht für jede LED extra als Kabel ausgeführt werden muss, dienen die leitenden Metallringe ebenfalls als Masse der 12 LEDs am Kopf. Die Leuchtschnüre werden mit transparentem Klebeband an die Ringe befestigt und die Ringe mittels Nylonfäden miteinander verbunden (vgl. Abb. 5.14). Auf diese Weise lässt sich das ganze Konstrukt beim Transport zusammenklappen.

Die restlichen Leuchtschnüre am Anzug werden als waagrechte Ringe entlang der Rückseite des Torsos, der Arme und der Beine mittels kleiner Filzstücke befestigt. Zusätzlich werden kleine Gummizüge an jedes Ende der Leuchtschnüre befestigt, damit diese sich beim An- und Ausziehen elastisch ausdehnen können. An das andere Ende jeder Leuchtschnur werden die Zuleitungen angelötet (siehe Abb. 5.15).

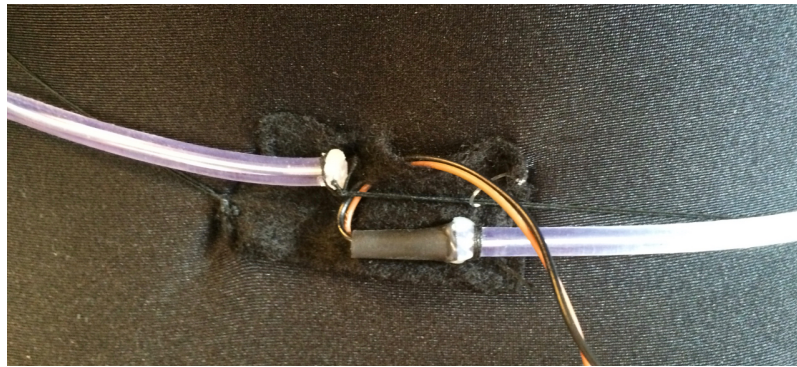


Abbildung 5.15.: Anbringung der Leuchtschnüre an den Anzug

Alle Zuleitungen der Leuchtschnüre und LEDs werden unterhalb dieser Filzstücke zum Gehäuse mit der Schaltung gelegt und dort verlötet. Dabei werden außer für die Massen (braun, Leuchtschnüre und LEDs getrennt) nur schwarze Litzen verwendet, damit die Kabel überwiegend schwarz sind. Die restlichen LEDs werden mittels kleiner zusammengeklebter Filzstücke mit Loch auf den Anzug genäht und mit den entsprechenden Kabeln (Masse und der jeweilige Kanal) verbunden.

Um zu zeigen, welcher Leuchtschnur- bzw. LED-Kanal an welcher Stelle des Körpers ist, stellt Abb. 5.16 die Verteilung dieser Kanäle schematisch dar. Die 12 LED-Kanäle werden jeweils in vier Gruppen (Kopf, Oberkörper rechts, Oberkörper links sowie unterhalb der Hüfte) a drei LED-Kanäle aufgeteilt. Die Aufzählung der Leuchtschnüre beginnt am Fußansatz mit Kanal 22, verläuft bis zum Hals bis Kanal 36, dann zum linken Arm von der Schulter bis zum Handgelenk, zum rechten Arm und dann zum Kopfteil von unten nach oben, bis Kanal 53 erreicht ist. Die Accelerometer werden an der Speiche des Unterarms an Klettbindern angebracht, damit sie einerseits im Falle einer Reinigung ebenfalls entfernt werden können und dass andererseits ihre genaue Position einstellbar bleibt.

5. Technische Realisierung

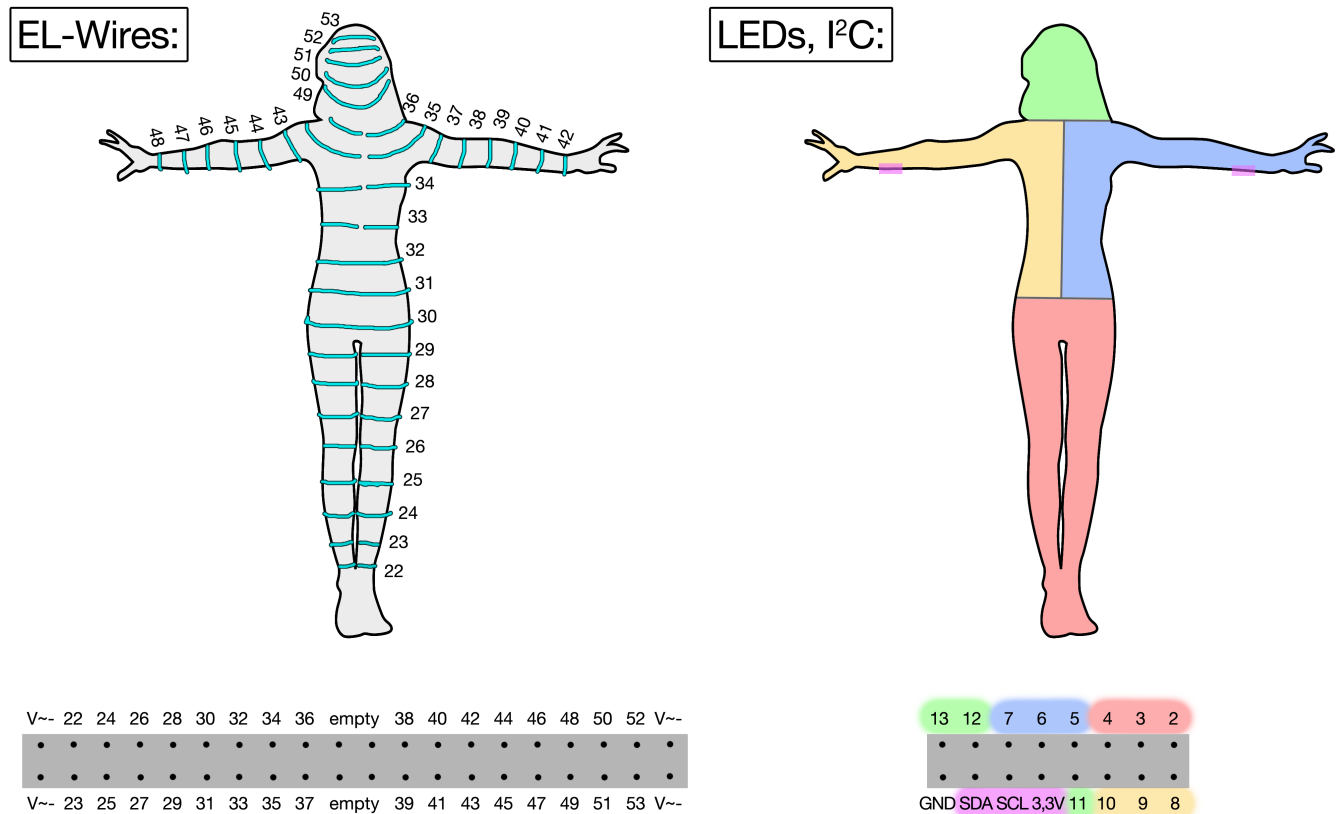


Abbildung 5.16.: Kanalbelegung der leuchtenden Elemente und der IDE-Stecker

5.2. Realisierung der Software

Nachdem die Entwicklung der Hardware des leuchtenden Kostüms nun beschrieben ist, widmet sich dieser Abschnitt der Software, die dabei zum Einsatz kommt. Hardwareseitig gibt es für das Projekt vier Instanzen, für welche eine Programmierung der Software notwendig ist. Abschnitt 5.2.1 erläutert den Code für den *Arduino UNO*, welcher als ISM-Sender am Laptop angeschlossen wird. Danach wird in Abschnitt 5.2.2 der Code des *Arduino DUE*, also des digitalen Herzstücks des Anzuges beleuchtet. In Abschnitt 5.2.3 findet sich der Code des *ESP8266*, welcher die Bewegungsdaten per WiFi an den Laptop sendet, bevor abschließend im letzten Abschnitt 5.2.4 der *Max4Live*-Patch in seiner Grundstruktur erklärt wird.

5.2.1. Code für Arduino UNO

Der Code, der für den *Arduino UNO* verwendet wird, ist vollständig in Anhang A.1 abgedruckt.

Das in der *Arduino*-Sprache (Dialekt von C++) verfasste Programm beginnt ab Zeile 18 mit dem Einfügen der Libraries für die SPI Schnittstelle und den *nRF20L01+* Funkchip. Die SPI Pins für CE und CSN werden auf Pin 7 und 8 festgelegt. Bevor die `setup`-Funktion aufgerufen wird, werden zwei Werte und zwei Adressen für die Funkstrecke initialisiert. Der Wert `inputNumber` ist dabei diejenige Zahl, welche via Digitalfunk zum *Arduino DUE* übertragen wird. In der `setup`-Funktion wird hier eine Serielle Kommunikation mit der baud rate 115200 initialisiert. Die Timeout-Zeit, die mit 3 ms festgelegt wird, ist ein fein eingestellter Wert, welcher empirisch auf den Befehl `Serial.parseInt()` abgestimmt ist, damit diese Funktion schnell genug, aber korrekt seriell empfangene Zahlen einliest. Anschließend wird im Setup die Funkverbindung initialisiert, der Verstärkungslevel auf maximal eingestellt (Zeile 44) und die Adressen für den Sende- und den Empfangsweg (nicht in Verwendung) bestimmt. Die seriellen Ausgaben danach dienen zum Debuggen.

Ab Zeile 53, in der `loop()`-Funktion, wird zunächst der Wert `inputNumber` durch die `Serial.parseInt()`-Funktion über die serielle Schnittstelle vom Laptop eingelesen. Danach wird mittels zweier `if`-Schleifen überprüft, ob diese fünfstellig ist. Falls ja wird dieser Wert mittels der `radio.write()`-Funktion an den *Arduino DUE* gesendet. Falls das Senden nicht erfolgreich war, wird über die serielle Schnittstelle eine Fehlermeldung zum Laptop gesendet. Hiermit ist der Loop beendet und die nächste Zahl kann eingelesen werden.

5.2.2. Code für Arduino DUE

Der Code des *Arduino DUE* kann in Anhang A.2 nachgelesen werden.

Hierbei werden vor der `setup()`-Funktion die selben Libraries und die selben globalen Variablen initialisiert wie für den *Arduino UNO*. Einzig die SPI-Pins für CE und CSN werden im Gegensatz zum *UNO* auf die Pins 14 und 15 gelegt. In der `setup()`-Funktion werden zunächst die LED- und Leuchtschnur-Ausgangspins des *DUE* festgelegt. Die digitalen Ausgänge ohne PWM werden dabei mittels `digitalWrite` auf den Wert LOW und diejenigen mit PWM mittels `analogWrite` auf den Wert 0 (von 255) initialisiert. Im Anschluss wird wie für den *UNO* die Funkstrecke initialisiert, mit maximaler Verstärkung und mit im Vergleich zum *UNO* vertauschten Adressen für das Senden und Empfangen. Mittels `radio.startListening()` beginnt der *DUE* auf vom *UNO* gesendete Funksignale zu „hören“. Im Anschluss werden alle leuchtenden Elemente inklusive geringer Verzögerungszeiten einmal an- und ausgeschaltet. Dies zeigt einerseits der Trägerin beim Anschalten des Anzuges, ob er ordnungsgemäß funktioniert und außerdem wird so für die Lötarbeiten eine eingebaute Testfunktion der leuchtenden Elemente bereitgestellt. Somit ist das Setup des Anzuges beendet.

5. Technische Realisierung

In der `loop()`-Funktion (ab Zeile 86) wird zunächst abgefragt, ob über Funk gesendete Daten im Puffer des *nRF20L01+* vorliegen. Falls ja, wird dieser Wert (die Variable `inputNumber`) ausgelesen. Der Rest des Codes besteht aus einem großen `if`-Statement, welches den Wert mit bestimmten Wertebereichen vergleicht (jeweils mit `if`, `else if` oder `else` ausgeführt) und je nachdem unterschiedliche Aktionen ausführt. Auf eine geschlossene Lösung anstatt der vielen `else if`-Statements wurde bewusst verzichtet, um eine bessere Wartung, Erweiterbarkeit und Lesbarkeit des Codes auf Kosten der Länge zu ermöglichen. Dabei kommt ein spezielles Aufschlüsselungssystem zum Einsatz, welches die Zahlen so verwaltet, wie es seitens des *Max4Live-Patches* (siehe Abschnitt 5.2.4) gefordert wird:

Die Zahlen werden in 1000-er Schritten den verschiedenen Kanälen zugeordnet. Zum Beispiel gehört zu Kanal 10 in Abb. 5.16 der Wertebereich zwischen 10000 und 10999. Danach wird jeweils die niedrigste Zahl des Wertebereichs abgezogen und dieses Ergebnis in die Variable `value` gespeichert. Für die digitalen Ausgänge (Leuchtschnüre) ist die letzte Stelle des Ausgangswertes entweder eine 1 oder eine 0 (Für Kanal 22 wäre der Wert beispielsweise 22001 oder 22000). Wird nun der Wert `value` ermittelt, bleibt hier entweder der Wert 1 oder der Wert 0 übrig, welche dann mittels `digitalWrite(i, value)` den Kanal auf HIGH oder LOW schaltet. Im Falle der pulswellenmodulierten LED-Kanäle bleibt nach Abzug des niedrigsten Wertes des Wertebereichs für `value` eine Zahl zwischen 0 und 255 übrig. Diese bestimmt mittels `analogWrite(i, value)` das PWM-Verhältnis des Ausganges (Beispielsweise bewirkt die Zahl 10255 für `inputNumber` eine angeschaltete LED des Kanals 10 in maximaler Leuchtkraft und 10001 die selbe LED in minimaler Leuchtkraft). Zusätzlich zu den einzelnen Kanälen aus Abb. 5.16 werden noch zusätzliche Wertebereiche eingeführt, welche alle LEDs oder Leuchtschnüre des gesamten Anzuges oder nur die des Kopfteils ansteuern. Dies geschieht innerhalb des jeweiligen `else if`-Statements in einer untergeordneten `for`-Schleife. Die Zahlen 16001 bzw. 16000 verursachen das An- bzw. Ausschalten des Relais für die Versorgungsspannung des *ESP8266* und der beiden Bewegungssensoren *MPU6050*.

Falls eine Zahl keinem Wertebereich zugeordnet werden kann, wird in Zeile 320 mittels `else` eine Fehlermeldung zum Debuggen an die Serielle Schnittstelle ausgegeben. Somit ist der Loop beendet und es wird auf die nächste eintreffende Zahl vom *UNO* gewartet. Diese `loop`-Funktion erfolgt in einer sehr hohen Geschwindigkeit, welche nur dadurch Möglich ist, dass dem *Arduino DUE* keine weiteren Aufgaben (wie z. B. der Rückweg der Funkverbindung) zugeteilt sind. Auf diese Weise kommen die vom Laptop gesendeten Daten ohne merkbare Latenz beim Anzug an und die programmierten Lichteffekte wirken sehr flüssig.

5.2.3. Code für ESP8266

Bevor auf den eben angesprochenen *Max4Live*-Patch eingegangen wird, erfolgt zunächst eine Beschreibung des Codes für den *ESP8266*, welcher die Bewegungsdaten der beiden Accelerometer *MPU6050* via WiFi an den Laptop sendet. Der Code hierzu ist in Anhang A.3 zu sehen.

Dieser ebenfalls in der *Arduino*-Sprache verfasste Code beginnt ab Zeile 21 mit dem Einbinden der Libraries für den Chip selbst, für eine WiFi Verbindung über das *User Datagram Protocol (UDP)*, einer Library für das Übertragen von OSC-Paketen und einer Library für zusätzliche Mathematikfunktionen. Im Anschluss werden globale Variablen, wie der SSID und dem Passwort des Netzwerkes (im Anhang aus Sicherheitsgründen als Sternchen dargestellt), der IP-Adresse des Laptops (statisch) sowie der Werte des Ausgangs- und des Eingangsports initialisiert. Nach der Festlegung der I²C-Pins, der Libraries für die Accelerometer und des I²C Busses (Zeile 46 bis 53), werden mittels des Befehls *MPU6050* zwei Objekte *accelgyro_r* und *accelgyro_l* für die Accelerometer des rechten und des linken Arms initialisiert. Bevor die *setup()*-Funktion ausgeführt wird, werden noch zwölf Variablen für die Bewegungsdaten definiert und die Werte mittels *#define OUTPUT_READABLE_ACCELGYRO* als lesbare Zahlen anstatt schneller zu verarbeitenden, aber unleserlichen Binaries auszugeben.

Ab Zeile 80 wird die *setup()*-Funktion begonnen. Hier wird zunächst die serielle Verbindung zum Debuggen gestartet. In Zeile 90 findet sich der Befehl zur Verbindung an das WiFi-Netzwerk mittels Übergabe der SSID und des Passworts. Eine *while*-Schleife hindert das Programm daran, weiter fortzuschreiten, bevor das WiFi noch nicht verbunden ist. In Zeile 103 wird das UDP-Protokoll und in den Zeilen 112 bis 132 die Verbindung zu den Accelerometern über den I²C-Bus aktiviert. Somit ist das Setup des *ESP8266* abgeschlossen.

In der *loop()*-Funktion werden zunächst die Accelerometerdaten für den jeweiligen *MPU6050* am linken und rechten Arm ausgelesen und in die Variablen *axr*, *ayr*, *azr* sowie *axl*, *ayl*, *azl* gespeichert. Diese werden im Anschluss (Zeile 142 bis 144) in jeweils einen *Roll*- und einen *Pitch*-Wert für den rechten Arm und in einen *Pitch*-Wert für den linken Arm umgerechnet. Der Rest der *loop()*-Funktion (ab Zeile 146) dient dazu, diese berechneten Werte als OSC-Messages über das UDP-Protokoll an die angegebene IP-Adresse und den angegebenen Port zu senden. Am Ende der Schleife findet sich ein Delay mit einer Millisekunde, welche die Stabilität des Programmes erhöht, da somit keine der Funktionen (Empfangen und Senden der Daten) in der ausgeführten Reihenfolge durcheinander kommt.

5. Technische Realisierung

5.2.4. Patch für Max4Live

Als letztes widmet sich dieser Abschnitt und auch das Kapitel zur technischen Realisierung des Projektes dem *Max4Live*-Patch, welcher mit der grafischen Multimediamprogrammiersprache *MAX/MSP* für die digitale Audioworkstation *Ableton Live* programmiert wird.

Die Beschreibung des kompletten Patches würde dabei den Rahmen dieser Arbeit sprengen. Außerdem befindet sich dieser zum Zeitpunkt der Anfertigung dieser Arbeit noch im Betastadium. Die Programmierung dieses Patches wird sich im Zuge der nächsten Proben mit dem Ensemble noch stetig verändert. Deswegen wird im Anschluss nur kurz anhand einiger Beispiele auf die generelle Funktion des für *Ableton* programmierten Softwareinstruments zur Steuerung des Anzuges und zur Umwandlung der Bewegungsdaten in einen Solo-Sound eingegangen.

Max4Live ist eine Angepasste Version der Programmiersprache *MAX/MSP* für *Ableton Live*. Hierbei können in den Audiospuren von *Live* entweder *Max*-Instrumente, -Audioeffekte oder -MiDieffekte erstellt werden. Im Zuge dieser Arbeit wird ein sogenanntes *Max*-Instrument erstellt, welches auf eine MiDi-Spur gelegt werden kann und somit Daten von einem Masterkeyboard in Klang verwandelt. Abbildung 5.17 zeigt das GUI dieses Instruments, welches die wichtigsten Funktionen und Anzeigen für den Livebetrieb umfasst.

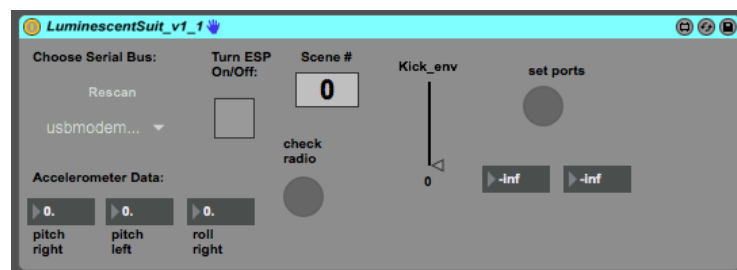


Abbildung 5.17.: GUI des *Max-Instruments* in *Ableton Live* zur Steuerung des Anzuges und des Solo-Sounds

Das Instrument wird dabei vom Masterkeyboard gesteuert. Mittels zweier definierter Tasten (Inkrement: Gis und Dekrement: Fis) können dabei die Szenen 0 bis 9 umgeschaltet werden. Diese beziehen sich auf die Verschiedenen in Abschnitt 3.2.1 vorgestellten acht Teile des Stückes plus jeweils einer Szene für vor und nach dem Stück als Intro und Outro. Diese Szenen werden auf dem GUI angezeigt (*Scene #*). Weitere Anzeigen und Schaltflächen sind dabei die Anzeigen der Accelerometerdaten (*Accelerometer Data*, unten links), externer Signalquellen aus *Ableton Live* (*Kick_env* und die Zahlen rechts daneben), der Auswahlreiter für den Seriellen Bus des über USB angeschlossenen ISM-Senders

(oben links) und zweier Schaltflächen zur redundanten Aktivierung des *ESP8266* (*Turn ESP ON/OFF*) sowie zum Testen der ISM-Verbindung (*check radio*).

Hauptpatch

Der verwendete Hauptpatch ist in Abb. 5.18 zu sehen.

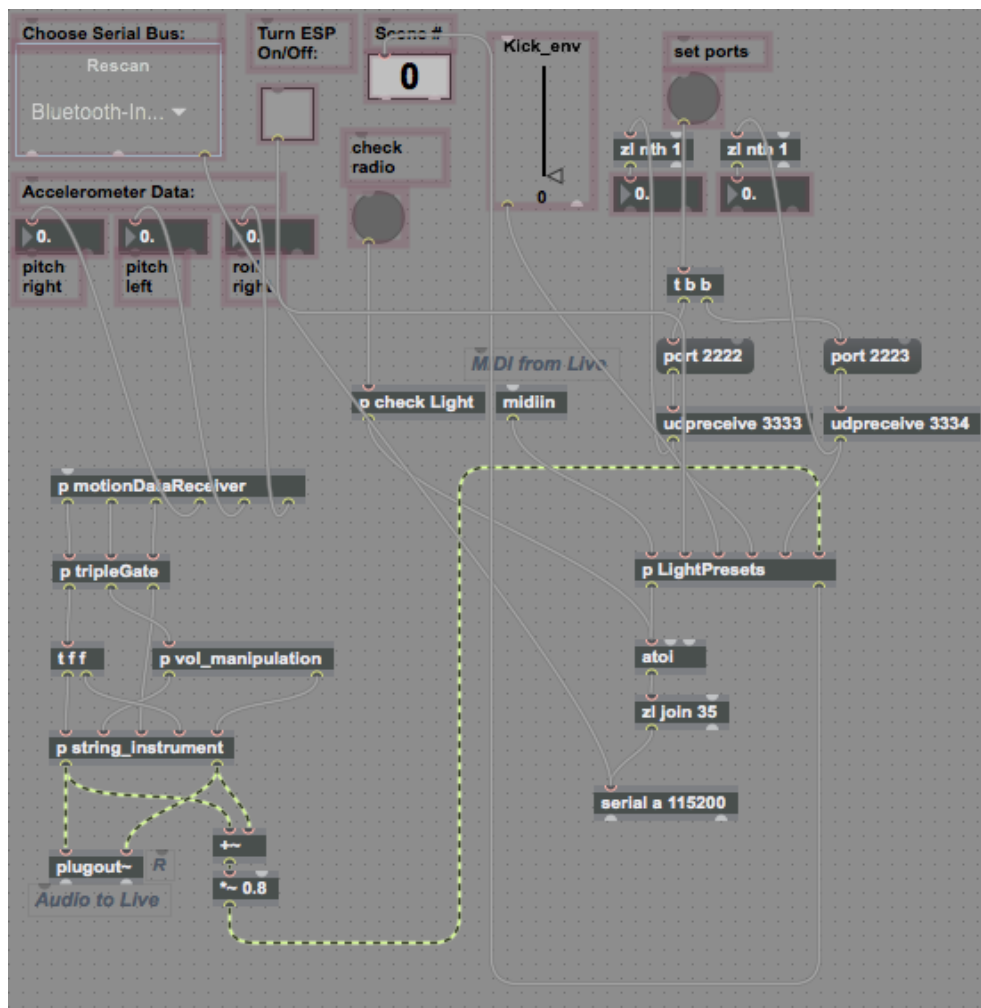


Abbildung 5.18.: Hauptpatch des *Max-Instruments*

Im oberen Viertel sind dabei die bereits beschriebenen Elemente des GUIs zu erkennen (rot). Die Objekte *midiin* und *plugout~* bilden den globalen Ein- und Ausgang des *Max-Instruments* aus der Sicht von *Ableton Live*.

5. Technische Realisierung

Senden der Licht- und Steuerdaten

Damit ein Befehl zur Aktivierung eines leuchtenden Elements auf dem Anzug an die Sendeeinheit des *Arduino UNO* erfolgen kann, wird mittels des Objektes *serial a 11520* im Hauptpatch ein serieller Ausgang zur Verfügung gestellt. Die beiden vorgeschalteten Objekte *atoi* und *zl join 35* bewirken eine Umwandlung von Integer-Numbers aus MAX in ASCII-Charaktere für die Serielle Verbindung und mittels *zl join 35* wird ein Leerzeichen (ASCII-Code 35) an den String in Form einer Liste angehängt. Dieser String wird im Code des *Arduino UNO* mittels *parseInt()* durch das Leerzeichen als Stoppsignal als zusammenhängende ganze Zahl erkannt. Mittels des Objektes *serial a 11520* wird der Bus *a* als Standard definiert. Über den Reiter des GUIs (oben links) kann dieser dann nach einem Scan aller verfügbaren Busse entsprechend verändert werden.

Einteilung der Lichtshowpresets in Szenen

In Abbildung 5.19 wird die Erzeugung von Zahlen dargestellt, welche als Steuerdaten über die serielle Schnittstelle weitergegeben werden. Dies geschieht anhand eines Beispiels von einer Szene des Stücks. Die Beschreibung aller Szenen würden den Rahmen dieser Arbeit sprengen.

Die fünf Eingänge dieser Szenen-Subpatches liefern verschiedene Daten, so wie sie im oberen Rand der Abbildung beschriftet sind. Dabei müssen nicht alle verfügbaren Daten von einer solchen Szene genutzt werden. In diesem Beispiel werden nur die Werte für die Szenennummer und die MiDi-Daten des Masterkeyboards verwendet. Ein *gate*-Objekt (links unten) lässt nur Lichtsteuerdaten an den Ausgang durch, wenn die entsprechende Szene aktiv ist. Die Taste 48 (C3) und 50 (D3) sind dabei in diesem Beispiel unterschiedlich belegt und werden mittels zweier *if*-Funktionen in *bang*-Messages verwandelt. Links erzeugt die *bang*-Message ein 65 Millisekunden langes gleichzeitiges Aufblinken aller Leuchtschnüre. Dies wird mit einem *toggle*-Objekt erzeugt (Werte 0 oder 1), welche zum Wert 54000, der Wertebereich im Code des *Arduino DUE* für „alle Leuchtschnüre“, hinzuaddiert wird. Ist Toggle aktiviert, ergibt sich der Wert 54001, wenn nicht, dann kommt 54000 am *DUE* an. Rechts sehen wir eine Funktion zur wiederholten Ansteuerung zufälliger 12 LED-Kanäle, die einen zufälligen PWM-Wert zwischen 0 und 255 aufweisen. Dies geschieht, solange die Taste D3 gedrückt wird, alle 15 Millisekunden.

In Abb. 5.20 ist der Subpatch *LightPresets* zu sehen, welcher die Szenen verwaltet.

Mittels *midiSetScene* wird dabei aus den MiDi-Daten die Szenennummer erzeugt (mithilfe eines Counters). Alle weiteren Daten neben MiDi (Inlet 1), wie der Einhüllenden der

5.2. Realisierung der Software



Abbildung 5.19.: Beispiel einer Szene (Szene 5 für Verse 3) der Lichtshow

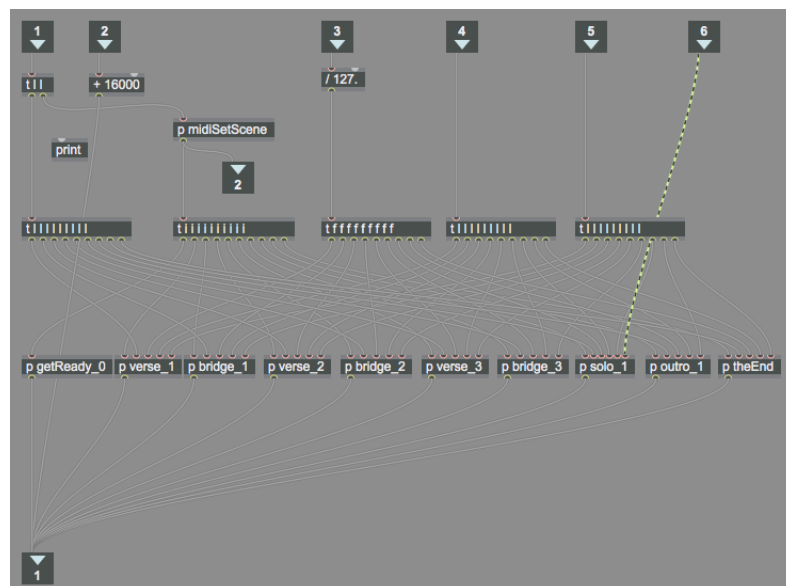


Abbildung 5.20.: Subpatch *LightPresets* zur Verwaltung der Szenen

5. Technische Realisierung

Bass Drum (Inlet 3), der FFT des Sounddesigns (Inlet 4) sowie der FFT des Gesangsmikrofones (Inlet 5), welche aus dem *Ableton*-Set stammen werden mittels *trigger*-Objekten auf die Szenen verteilt. Das Audiosignal des Solosounds (Inlet 6) wird direkt zur Solo-Szene weitergegeben. Der Ausgang des hier beschriebenen Subpatches *LightPresets* wird im Hauptpatch, wie oben erwähnt, in das *atoi*-, das *zl join 35* und letztlich in das *serial a 11520*-Objekt geleitet.

Empfangen der Bewegungsdaten

Die Accelerometerdaten werden wie in Abschnitt 5.2.3 beschrieben vom *ESP8266* via UDP an den Port 9999 gesendet. Der Subpatch *motionDataReceiver* empfängt diese Daten mittels dem *MAX*-Objekt *udpreceive 9999* (Abb. 5.21).



Abbildung 5.21.: Subpatch *motionDataReceiver* zum Empfangen und Bearbeiten der Bewegungsdaten

Im Anschluss werden in dem Subpatch die OSC-Messages mittels *OSC-route*-Befehl in die beiden Pitch-Werte und den Roll-Wert aufgetrennt. Diese drei Werte werden nach einer Aufbereitung an die Ausgänge 1 bis 3 weitergegeben. Die Ausgänge 4 bis 6 dienen der Visualisierung im GUI.

Solo-Sound und Klangbearbeitung

Die Bewegungsdaten landen nach einem weiteren speziell angefertigten *gate*-Objekt sowie einiger Skalierungsfunktionen (siehe Hauptpatch in Abb. 5.18) in einer weiteren Unterfunktion namens *string_instrument*, welche die Generierung des Solosounds vollführt. Dieser Subpatch wird in Abb. 5.22 dargestellt. Er enthält zwei Unterfunktionen mit einem *Karplus-Strong-Algorithmus*, welche den Klang des Solos erzeugen.

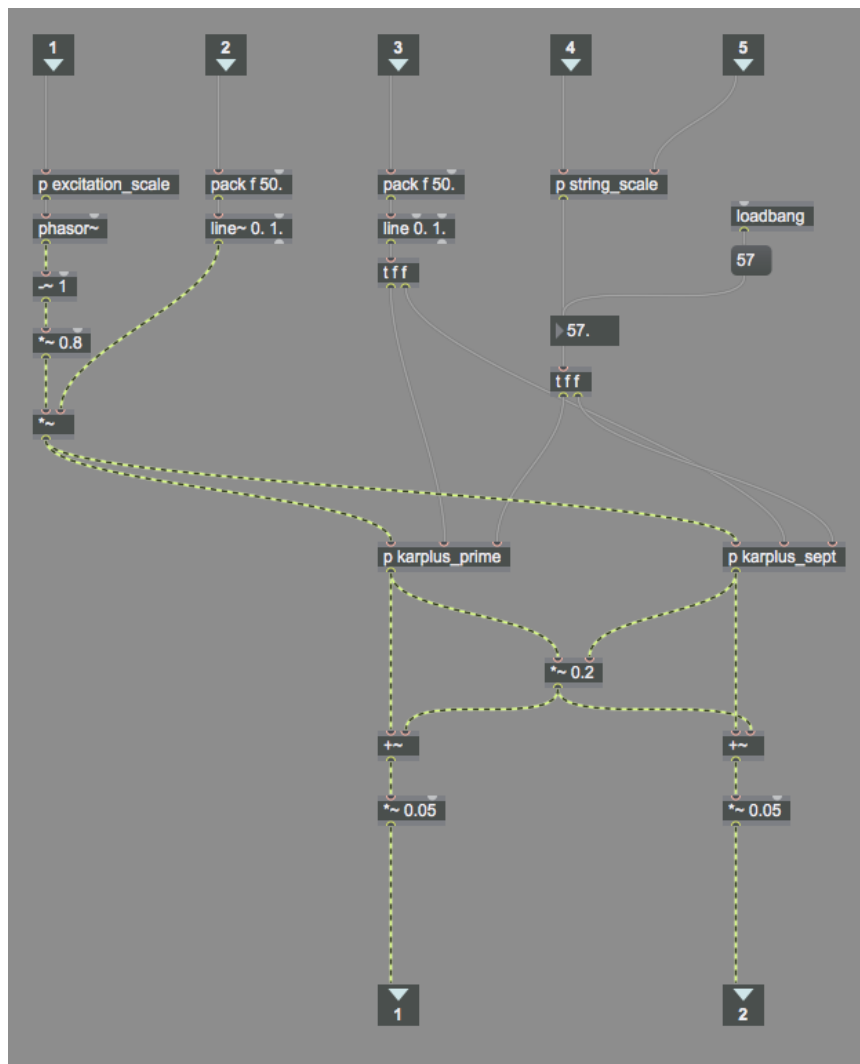


Abbildung 5.22.: Subpatch *string_instrument* zur Erzeugung des Solo-Sounds

Auf die komplette Beschreibung des Solosounds wird im Zuge dieser Arbeit verzichtet, da er bereits in einer Seminararbeit am *Institut für Elektronische Musik* (IEM) in Graz namens *Physical Modeling von Saiten, welche durch die Bewegungsdaten eines Smartphones gesteuert werden* beschrieben wird. Der für das Anzugprojekt verwendete Sound ist

5. Technische Realisierung

exakt der selbe. Einzig die Steuerung durch das Smartphone wird hier durch die beiden Bewegungssensoren an den Armen des Anzuges ersetzt. Ein weiterer Unterschied ist, dass für die Seminararbeit die Open-Source-Programmiersprache *Pure Data* anstatt *Max4Live* verwendet wurde. Die Portierung des Codes wurde dabei nach den hierfür gängigen Methoden umgesetzt.

6. Fazit und Ausblick

In dieser Projektarbeit wird die Entwicklung eines leuchtenden Kostüms für eine Live-Darbietung beschrieben. Diese künstlerisch-technische Arbeit umfasst dabei viele sehr unterschiedliche Aufgaben von der Recherche über die Vermittlungsarbeit mit der Künstlerin bis hin zur technischen Umsetzung des Projektes. Abschließend kann gesagt werden, dass das Projekt insgesamt enorm viel Arbeit mit sich brachte. Nicht nur die in diesem Dokument beschriebenen Vorgänge sondern auch die handwerklichen Maßnahmen waren dabei sehr zeitintensiv. Insgesamt wurden zur Fertigstellung dieser Arbeit weit mehr Stunden benötigt, als für ein Toningenieur-Projekt nach dem Curriculum vorgesehen sind. Die Tatsache, dass seitens des Verfassers ein großes persönliches Interesse zur Fertigstellung des Kostüms vorhanden war, wurde zu einem wichtigen Motivationsfaktor. Außerdem brachten die vielen unterschiedlichen Aufgaben einen großen Lerneffekt mit sich.

Für die weitere Entwicklung des Anzugprojektes ist zunächst ein erster Auftritt am 8.10. 2016 in München geplant. Anhand der Erfahrungen, die sich bei diesem Auftritt und der zugehörigen Probe ergeben, werden neue Ziele für die weiteren Programmier- und Werkarbeiten bezüglich der Korrektur des leuchtenden Kostüms erarbeitet. Interessant ist dabei, wie sich die Funktion des Anzugs auf einer Bühne verhält. Die beiden 2,4 GHz Funkstrecken (WiFi und ISM) des Anzuges könnten beispielsweise durch HF-Funksignalen von Publikum, Rundfunk o.Ä. gestört werden. Außerdem ist interessant, wie sich das An- und Ausziehen des Kostüms in einer Auftrittssituation gestalten lässt.

Als ein weiteres Fazit zur Projektarbeit kann festgemacht werden, dass die Verwendung der *Arduino-Plattform* eine gute Wahl zum Einstieg in die Welt der Mikrocontroller war. Andere Boards und Prozessoren, wie z. B. der *A20 dual core Cortex-A7*, wären für den Verfasser in Zukunft das Mittel der Wahl, da sie z. T. schneller sind und mehr GPIOs besitzen, welche die Upgradefähigkeit des Anzuges erhöhen würden.

Anhang

Anhang A.

Quellcodes

A.1. Code für Arduino UNO

```
1  /**-----
2  // Luminescent_Suit_UNO.ino
3  // This Arduino Sketch lets an Arduino UNO receive a serial signal in form of a code number
4  // and transmit this number as an ISM radio signal via the nRF24L01+ chip.
5  //
6  // 09/2016 by Hannes Dieterle for a project thesis at Institute of Electronic Music and
7  // Acoustics (IEM) at University of Performing Arts in Graz, Austria
8  // hannes.dieterle@gmail.com
9  //
10 // This program is free software: you can redistribute it and/or modify it under the terms of
11 // the GNU General Public License as published by the Free Software Foundation, either version 3
12 // of the License, or (at your option) any later version.
13 //
14 // Based on the following code examples:
15 // - "GettingStarted" example sketch for nRF24L01+ radios, Updated: Dec 2014 by TMRh20
16 //
17 //----- */
18 #include <SPI.h>
19 #include "RF24.h"
20
21 // Hardware configuration: Set up nRF24L01 radio on SPI bus plus pins 7 & 8
22 RF24 radio(7, 8);
23
24 unsigned int value = 0;
25 unsigned long inputNumber = 0;
26
27 byte addresses[][6] = {"1Node", "2Node"};
28
29 void setup() {
30
31   Serial.begin(115200);
32   // This is important for Serial.parseInt. It defines how long the function waits for
33   // incoming Serial Bytes:
34   Serial.setTimeout(3);
35
36   // Initialize radio communication
37   radio.begin();
38
39   // Tradeoff: power supply related issues vs. transmission range:
40   // (RF24_PA_MIN=-18dBm, RF24_PA_LOW=-12dBm, RF24_PA_MED=-6dBm, and RF24_PA_HIGH=0dBm)
41   //radio.setPALevel(RF24_PA_MIN);
42   //radio.setPALevel(RF24_PA_LOW);
43   //radio.setPALevel(RF24_PA_HIGH);
44   radio.setPALevel(RF24_PA_MAX); // (default)
45
46   radio.openWritingPipe(addresses[0]);
47   radio.openReadingPipe(1, addresses[1]);
48
49   Serial.println( F("*** Remote radio transmission device for luminescent suit ***") );
50   Serial.println( F("Ready for transmission") );
51 }
52
53 void loop() {
```

Anhang A. Quellcodes

```
55  inputNumber = Serial.parseInt();
57  if ( inputNumber < 10000 && inputNumber > 0) {
59      Serial.println("Input Error: Serial transmission failed.");
61  }
63  if ( inputNumber >= 10000 ) {
65      if (!radio.write( &inputNumber, sizeof(unsigned long) )) {
67          Serial.println(F("Radio transmission failed."));
69      }
71  }
73  } // Loop
```

A.2. Code für Arduino DUE

```
1  /***-----
2  // Luminescent_Suit_DUE.ino
3  // This Arduino Sketch lets an Arduino DUE receive an ISM radio signal in form of a code number
4  // and transfer it into specific physical outputs which run EL wire and LEDs.
5  //
6  // 09/2016 by Hannes Dieterle for a project thesis at Institute of Electronic Music and
7  // Acoustics (IEM) at University of Performing Arts in Graz, Austria
8  // hannes.dieterle@gmail.com
9  //
10 // This program is free software: you can redistribute it and/or modify it under the terms of
11 // the GNU General Public License as published by the Free Software Foundation, either version 3
12 // of the License, or (at your option) any later version.
13 //
14 // Based on the following code examples:
15 // - "GettingStarted" example sketch for nRF24L01+ radios, Updated: Dec 2014 by TMRh20
16 //
17 //----- */
18 #include <SPI.h>
19 #include "RF24.h"
20
21 // Hardware configuration: Set up nRF24L01 radio on SPI bus plus pins 14 & 15
22 RF24 radio(14, 15);
23
24 unsigned long inputNumber = 0;
25 unsigned int value = 0;
26
27 byte addresses[][6] = {"1Node","2Node"};
28
29 void setup() {
30
31     pinMode(16, OUTPUT); // Initialize ESP8266-VCC-Pin
32     digitalWrite(16, LOW);
33
34     for ( int i = 2 ; i < 14 ; i++ ){ // Initialize PWM outputs
35         pinMode(i, OUTPUT);
36         analogWrite(i, 0);
37     }
38     for ( int i = 22 ; i < 54 ; i++ ){ // Initialize non-PWM outputs
39         pinMode(i, OUTPUT);
40         digitalWrite(i, LOW);
41     }
42
43     Serial.begin(115200);
44
45     // Initialize radio communication
46     radio.begin();
47
48     // Tradeoff: power supply related issues vs. transmission range:
49     // (RF24_PA_MIN=-18dBm, RF24_PA_LOW=-12dBm, RF24_PA_MED=-6dBm, and RF24_PA_HIGH=0dBm)
50     //radio.setPALevel(RF24_PA_MIN);
51     //radio.setPALevel(RF24_PA_LOW);
52     //radio.setPALevel(RF24_PA_HIGH);
53     radio.setPALevel(RF24_PA_MAX); // (default)
54
55     radio.openWritingPipe(addresses[1]);
56     radio.openReadingPipe(1,addresses[0]);
```

```

57  radio.startListening();
59
61  Serial.println ("Suit ready");
63
65  for ( int i = 2 ; i < 14 ; i++ ){
67      analogWrite(i, 255);
69      delay(200);
71  }
73
75  delay(500);
77
79  for ( int i = 2 ; i < 14 ; i++ ){
81      analogWrite(i, 0);
83      delay(200);
85  }
87
89  delay(600);
91
93  for ( int i = 22 ; i < 54 ; i++ ){
95      digitalWrite(i, HIGH);
97      delay(45);
99      digitalWrite(i, LOW);
101  }
103
105  void loop() {
107
109      if( radio.available()){
111          //Serial.println("Radio available, value:");
113          while (radio.available()) {
115              radio.read( &inputNumber, sizeof(unsigned long) );
117              //Serial.println(inputNumber);
119          }
121
123          if ( inputNumber == 0 ) {
125              // do nothing
127          }
129
131          // LED Channels -----
133          else if ( inputNumber < 63000 && inputNumber >= 62000) {
135              value = inputNumber - 62000;
137              analogWrite(2, value);
139          }
141          else if ( inputNumber < 64000 && inputNumber >= 63000) {
143              value = inputNumber - 63000;
145              analogWrite(3, value);
147          }
149          else if ( inputNumber < 65000 && inputNumber >= 64000) {
151              value = inputNumber - 64000;
153              analogWrite(4, value);
155          }
157          else if ( inputNumber < 66000 && inputNumber >= 65000) {
159              value = inputNumber - 65000;
161              analogWrite(5, value);
163          }
165          else if ( inputNumber < 67000 && inputNumber >= 66000) {
167              value = inputNumber - 66000;
169              analogWrite(6, value);
171          }
173          else if ( inputNumber < 68000 && inputNumber >= 67000) {
175              value = inputNumber - 67000;
177              analogWrite(7, value);
179          }
181          else if ( inputNumber < 69000 && inputNumber >= 68000) {
183              value = inputNumber - 68000;
185              analogWrite(8, value);
187          }
189          else if ( inputNumber < 70000 && inputNumber >= 69000) {
191              value = inputNumber - 69000;
193              analogWrite(9, value);
195          }
197          else if ( inputNumber < 11000 && inputNumber >= 10000) {
199              value = inputNumber - 10000;
201              analogWrite(10, value);
203          }
205          else if ( inputNumber < 12000 && inputNumber >= 11000) {
207              value = inputNumber - 11000;
209              analogWrite(11, value);
211          }
213      }
215      // While there is data ready
216      // Get the payload

```

Anhang A. Quellcodes

```
141     }
142     else if ( inputNumber < 13000 && inputNumber >= 12000) {
143         value = inputNumber - 12000;
144         analogWrite(12, value);
145     }
146     else if ( inputNumber < 14000 && inputNumber >= 13000) {
147         value = inputNumber - 13000;
148         analogWrite(13, value);
149     }
150     // All LEDs
151     else if ( inputNumber < 15000 && inputNumber >= 14000) {
152         value = inputNumber - 14000;
153         for ( int i = 2 ; i < 14 ; i++ ){
154             analogWrite(i, value);
155         }
156     }
157     // All LEDs of head
158     else if ( inputNumber < 16000 && inputNumber >= 15000) {
159         value = inputNumber - 15000;
160         for ( int i = 11 ; i < 14 ; i++ ){
161             analogWrite(i, value);
162         }
163     }
164     // ESP8266: ON/OFF -----
165     else if ( inputNumber < 17000 && inputNumber >= 16000) {
166         value = inputNumber - 16000;
167         digitalWrite(16, value);
168     }
169     // EL Wire Channels -----
170     else if ( inputNumber < 23000 && inputNumber >= 22000) {
171         value = inputNumber - 22000;
172         digitalWrite(22, value);
173     }
174     else if ( inputNumber < 24000 && inputNumber >= 23000) {
175         value = inputNumber - 23000;
176         digitalWrite(23, value);
177     }
178     else if ( inputNumber < 25000 && inputNumber >= 24000) {
179         value = inputNumber - 24000;
180         digitalWrite(24, value);
181     }
182     else if ( inputNumber < 26000 && inputNumber >= 25000) {
183         value = inputNumber - 25000;
184         digitalWrite(25, value);
185     }
186     else if ( inputNumber < 27000 && inputNumber >= 26000) {
187         value = inputNumber - 26000;
188         digitalWrite(26, value);
189     }
190     else if ( inputNumber < 28000 && inputNumber >= 27000) {
191         value = inputNumber - 27000;
192         digitalWrite(27, value);
193     }
194     else if ( inputNumber < 29000 && inputNumber >= 28000) {
195         value = inputNumber - 28000;
196         digitalWrite(28, value);
197     }
198     else if ( inputNumber < 30000 && inputNumber >= 29000) {
199         value = inputNumber - 29000;
200         digitalWrite(29, value);
201     }
202     else if ( inputNumber < 31000 && inputNumber >= 30000) {
203         value = inputNumber - 30000;
204         digitalWrite(30, value);
205     }
206     else if ( inputNumber < 32000 && inputNumber >= 31000) {
207         value = inputNumber - 31000;
208         digitalWrite(31, value);
209     }
210     else if ( inputNumber < 33000 && inputNumber >= 32000) {
211         value = inputNumber - 32000;
212         digitalWrite(32, value);
213     }
214     else if ( inputNumber < 34000 && inputNumber >= 33000) {
215         value = inputNumber - 33000;
216         digitalWrite(33, value);
217     }
218     else if ( inputNumber < 35000 && inputNumber >= 34000) {
```



```

223     value = inputNumber - 34000;
224     digitalWrite(34, value);
225 }
226 else if ( inputNumber < 36000 && inputNumber >= 35000) {
227     value = inputNumber - 35000;
228     digitalWrite(35, value);
229 }
230 else if ( inputNumber < 37000 && inputNumber >= 36000) {
231     value = inputNumber - 36000;
232     digitalWrite(36, value);
233 }
234 else if ( inputNumber < 38000 && inputNumber >= 37000) {
235     value = inputNumber - 37000;
236     digitalWrite(37, value);
237 }
238 else if ( inputNumber < 39000 && inputNumber >= 38000) {
239     value = inputNumber - 38000;
240     digitalWrite(38, value);
241 }
242 else if ( inputNumber < 40000 && inputNumber >= 39000) {
243     value = inputNumber - 39000;
244     digitalWrite(39, value);
245 }
246 else if ( inputNumber < 41000 && inputNumber >= 40000) {
247     value = inputNumber - 40000;
248     digitalWrite(40, value);
249 }
250 else if ( inputNumber < 42000 && inputNumber >= 41000) {
251     value = inputNumber - 41000;
252     digitalWrite(41, value);
253 }
254 else if ( inputNumber < 43000 && inputNumber >= 42000) {
255     value = inputNumber - 42000;
256     digitalWrite(42, value);
257 }
258 else if ( inputNumber < 44000 && inputNumber >= 43000) {
259     value = inputNumber - 43000;
260     digitalWrite(43, value);
261 }
262 else if ( inputNumber < 45000 && inputNumber >= 44000) {
263     value = inputNumber - 44000;
264     digitalWrite(44, value);
265 }
266 else if ( inputNumber < 46000 && inputNumber >= 45000) {
267     value = inputNumber - 45000;
268     digitalWrite(45, value);
269 }
270 else if ( inputNumber < 47000 && inputNumber >= 46000) {
271     value = inputNumber - 46000;
272     digitalWrite(46, value);
273 }
274 else if ( inputNumber < 48000 && inputNumber >= 47000) {
275     value = inputNumber - 47000;
276     digitalWrite(47, value);
277 }
278 else if ( inputNumber < 49000 && inputNumber >= 48000) {
279     value = inputNumber - 48000;
280     digitalWrite(48, value);
281 }
282 else if ( inputNumber < 50000 && inputNumber >= 49000) {
283     value = inputNumber - 49000;
284     digitalWrite(49, value);
285 }
286 else if ( inputNumber < 51000 && inputNumber >= 50000) {
287     value = inputNumber - 50000;
288     digitalWrite(50, value);
289 }
290 else if ( inputNumber < 52000 && inputNumber >= 51000) {
291     value = inputNumber - 51000;
292     digitalWrite(51, value);
293 }
294 else if ( inputNumber < 53000 && inputNumber >= 52000) {
295     value = inputNumber - 52000;
296     digitalWrite(52, value);
297 }
298 else if ( inputNumber < 54000 && inputNumber >= 53000) {
299     value = inputNumber - 53000;
300     digitalWrite(53, value);
301 }
302 }
303 // All EL-Wires
304 else if ( inputNumber < 55000 && inputNumber >= 54000) {
305     value = inputNumber - 54000;

```

Anhang A. Quellcodes

```
307     for ( int i = 22 ; i < 54 ; i++ ){
        digitalWrite(i, value);
    }
309 }

311 // All EL-Wires of head
    else if ( inputNumber < 56000 && inputNumber >= 55000) {
313     value = inputNumber - 55000;
        for ( int i = 49 ; i < 54 ; i++ ){
315         digitalWrite(i, value);
        }
317     }

319 // Non defined numbers
    else {
321     Serial.print(inputNumber);
        Serial.println(F(": Input Error"));
323     }
325 } // Loop
```

A.3. Code für ESP8266

```
1  //-----
2  // 2MPU_via_OSC_ESP8266.ino
3  // This Arduino Sketch lets an ESP8266 WiFi enabled microcontroller send two pitch and one roll
4  // values of two MPU6050 accelerometer units to a specified IP address in a specified Wifi
5  // network via OSC.
6  //
7  // 09/2016 by Hannes Dieterle for a project thesis at Institute of Electronic Music and
8  // Acoustics (IEM) at University of Performing Arts in Graz, Austria
9  // hannes.dieterle@gmail.com
10 //
11 // This program is free software: you can redistribute it and/or modify it under the terms of
12 // the GNU General Public License as published by the Free Software Foundation, either version 3
13 // of the License, or (at your option) any later version.
14 //
15 // Based on the following code examples:
16 // - "ESP8266sendMessage" Open Sound Control (OSC) library for the ESP8266 (public domain)
17 // - "MPU6050_raw.ino": I2C device class (I2Cdev) demonstration Arduino sketch for MPU6050 class
18 //   10/7/2011 by Jeff Rowberg <jeff@rowberg.net>
19 //
20 //----- */
21 #include <ESP8266WiFi.h>
22 #include <WiFiUdp.h>
23 #include <OSCMessage.h>
24 #include <math.h>
25
26 char ssid[] = "*****"; // your network SSID (name)
27 char pass[] = "*****"; // your network password
28
29 // A UDP instance to let us send and receive packets over UDP
30 WiFiUDP Udp;
31 // remote IP (Computer hosting Max4Live), always 192.168.0.100
32 const IPAddress outIp(192,168,0,100);
33 // remote port (here's where we send the packets to Max4Live)
34 const unsigned int outPort = 9999;
35 // local port to listen for UDP packets (here's where we send the packets from Max4Live):
36 const unsigned int localPort = 8888;
37
38 // This identifies ESP8266 pin GPIO2 as a I2C SDA wire
39 #define I2C_MASTER_SDA_GPIO 2
40 // This identifies ESP8266 pin GPIO4 as a I2C SCL wire
41 #define I2C_MASTER_SCL_GPIO 4
42
43 // I2Cdev and MPU6050 must be installed as libraries, or else the .cpp/.h files
44 // for both classes must be in the include path of your project
45 #include "I2Cdev.h"
46 #include "MPU6050.h"
47
48 // Arduino Wire library is required if I2Cdev I2CDEV_ARDUINO_WIRE implementation
49 // is used in I2Cdev.h
50 #if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
```

```

#include "Wire.h"
53 #endif

55 // class default I2C address is 0x68
// first MPU with ADO low (default)
57 MPU6050 accelgyro_r;
// second MPU with ADO high
59 MPU6050 accelgyro_l(0x69); // <-- use for ADO high

61 int16_t axr, ayr, azr;
int16_t gxr, gyr, gzr;
63
65 int16_t axl, ayl, azl;
int16_t gx1, gy1, gz1;

67 // uncomment "OUTPUT_READABLE_ACCELGYRO" if you want to see a tab-separated
// list of the accel X/Y/Z and then gyro X/Y/Z values in decimal. Easy to read,
69 // not so easy to parse, and slow(er) over UART.
#define OUTPUT_READABLE_ACCELGYRO

71 // uncomment "OUTPUT_BINARY_ACCELGYRO" to send all 6 axes of data as 16-bit
73 // binary, one right after the other. This is very fast (as fast as possible
// without compression or data loss), and easy to parse, but impossible to read
75 // for a human.
// #define OUTPUT_BINARY_ACCELGYRO
77

79 void setup() {
81
83     // starting Serial communication
    Serial.begin(115200);

85     // Connect to WiFi network
    Serial.println();
87     Serial.println();
    Serial.print("Connecting to ");
89     Serial.println(ssid);
    WiFi.begin(ssid, pass);

91     while (WiFi.status() != WL_CONNECTED) {
93         delay(500);
        Serial.print(".");
95     }
    Serial.println("");

97     Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());

101     Serial.println("Starting UDP");
    Udp.begin(localPort);
103     Serial.print("Local port: ");
    Serial.println(Udp.localPort());

105     Serial.println("Sending to IP address:");
    Serial.println(outIp);

109

111     // join I2C bus (I2Cdev library doesn't do this automatically)
    #if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
        Wire.begin(I2C_MASTER_SDA_GPIO, I2C_MASTER_SCL_GPIO);
113     #elif I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE
        Fastwire::setup(400, true);
115     #endif

117     // initialize first (right) MPU6050 device
    Serial.println("Initializing MPU6050 on right arm");
    accelgyro_r.initialize();
121

    // verify connection
123     Serial.println("Testing device connection...");
    Serial.println(accelgyro_r.testConnection() ? "MPU6050 connection successful" : "MPU6050 connection failed");
125

    // initialize second (left) MPU6050 device
127     Serial.println("Initializing MPU6050 on left arm");
    accelgyro_l.initialize();

129     // verify connection
131     Serial.println("Testing device connection...");
    Serial.println(accelgyro_l.testConnection() ? "MPU6050 connection successful" : "MPU6050 connection failed");
133
}

```

Anhang A. Quellcodes

```
135 void loop() {
137     // access acceleration data by the two MPU6050 units
139     accelgyro_r.getAcceleration(&axr, &ayr, &azr);
    accelgyro_l.getAcceleration(&axl, &ayl, &azl);
141
    float roll = atan2( ayr , azr) * 180/PI;
143     float pitch_r = atan2( axr , sqrt( ayr*ayr + azr*azr) ) * 180/PI;
    float pitch_l = atan2( axl , sqrt( ayl*ayl + azl*azl) ) * 180/PI;
145
    OSCMessage msgr("/roll_r");
147     msgr.add(roll);
    Udp.beginPacket(outIp, outPort);
149     msgr.send(Udp);
    //Serial.println(roll);
151     Udp.endPacket();
    msgr.empty();
153
    OSCMessage msgpr("/pitch_r");
155     msgpr.add(pitch_r);
    Udp.beginPacket(outIp, outPort);
157     msgpr.send(Udp);
    Udp.endPacket();
159     msgpr.empty();

161     OSCMessage msgpl("/pitch_l");
    msgpl.add(pitch_l);
163     Udp.beginPacket(outIp, outPort);
    msgpl.send(Udp);
165     Udp.endPacket();
    msgpl.empty();
167
169     delay(1); // delay seems to increase the stability of the OSC sending process
}
```

Literaturverzeichnis

- [BACHFELD; 2015] BACHFELD, Daniel: *Wearables - Einfach tragbar*, Coverstory im MAKE: Magazin 04/2015, Maker Media GmbH, Hannover, 2015
- [DVORAK; 2011] DVORAK, Walter: *Schaltung Elektrisches Uhrwerk mit Sperrschwinger*, wikimedia.org, licensed under the Creative Commons Attribution 3.0 Unported, 2011, Online im WWW unter URL [Stand 28.09.2016]:
https://upload.wikimedia.org/wikipedia/commons/7/73/Schaltung_Elektrisches_Uhrwerk_mit_Sperrschwinger.svg
- [ECKEL et al.; 2009] ECKEL, Gerhard, PIRRÒ, David und SHARMA, Gerriet K.: *Motion-enabled live electronics*, Proceedings of the 6th Sound and Music Computing Conference, pages 36-41, 2009
- [FISCHER; 2014] FISCHER, Eva: *Audiovisuelle Kunst. Entwicklung eines Begriffes - VJing, audiovisuelle Live Performance und Installation im Kontext kunsthistorischer und zeitgenössischer Entwicklungen*, AV Akademikerverlag, Saarbrücken, 2014
- [FISCHER-LICHTE; 2014] FISCHER-LICHTE, Erika: *Culture as Performance: Theatre History As Cultural History*, ACTAS/ PROCEEDINGS 7 - HISTÓRIA DO TEATRO E NOVAS TECNOLOGIAS, 2014, Online im WWW unter URL [Stand 17.09.2016]:
http://ww3.fl.ul.pt/centros_invst/teatro/pagina/Publicacoes/Actas/erika_def.pdf
- [JEWANSKI] JEWANSKI, Jörg: *Eine kurze Geschichte der Farblichtmusik*, künstlerisch-wissenschaftliches Projekt *Farbe - Licht - Musik* der Zürcher Hochschule der Künste (ZHdK) und der ETH Zürich, Online im WWW unter URL [Stand 16.09.2016]:
<http://www.farblicht.ch/images/farblichtmusik/>
- [KOBZINOWSKI; 2016] KOBZINOWSKI, Jan: *Monika Roscher*, Coverstory im Jazzthetik Magazin Juli/August 2016, Verlag Christine Stephan, Münster, 2016
- [SPEISER; 1963] SPEISER, Ambros P.: *Impulsschaltungen - Erzeugung und Verarbeitung von Impulsen mit Röhren und Transistoren*, Springer-Verlag OHG., Berlin/Göttingen/Heidelberg, 1963
- [STEFFAN; 2015] STEFFAN, Philip: *Kleidung mit Launen*, Artikel im MAKE: Magazin 04/2015, Maker Media GmbH, Hannover, 2015

Literaturverzeichnis

[TSCHACHER, STORCH; 2012] TSCHACHER, Wolfgang und STORCH, Maja: *Die Bedeutung von Embodiment für Psychologie und Psychotherapie*, Magazin *Psychotherapie* 17. Jahrg. 2012, Bd. 17, Heft 2, CIP-Medien, München, 2012