



# FREETAG LASER- TAGSYSTEM

# **FreeTag Lasertagsystem**

Abschlussarbeit zur Erlangung der Berufsbezeichnung

## **WERKMEISTER / IN**

*In der Fachrichtung: Mechatronik*

Klassenvorstand: **KommR Johann Fiedler**

*Schuljahr: 2017/2018  
angefertigt in der Werkmeisterschule der WIFI OÖ GmbH.  
eingereicht von:*

**Emmanuel Panholzer**

*Fachliche Betreuung durchgeführt von:  
**KommR Johann Fiedler***

Linz, am Donnerstag den 29.03.2018

*Panholzer Emmanuel*

## 1 LEBENSLAUF

### Persönliche Daten

**Name:** Emmanuel Panholzer



**Adresse:** Katzgraben Straße 90, 4203 Altenberg bei Linz

**E-Mail-Adresse:** e.panholzer@gmx.at

**Familienstand:** ledig

**Staatsangehörigkeit:** Österreich

**Geburtsdaten:** 14 Februar 1990 in Freistadt

### Berufliche Qualifikation

**seit 09/2005:** Lehre zum Produktionstechniker bei der Voest Alpine Stahl GesmbH in Linz, Abschluss der Lehre im Jänner 2009 mit Auszeichnung

**seit 02/2009:** Mitarbeiter des Kraftwerkes der Voest Alpine Stahl GesmbH in Linz, Abteilung TSS, 2009-2010 kurze Unterbrechung der Tätigkeit wegen Einberufung zum Präsenzdienst

### Schulische Ausbildung:

**1996 bis 2000:** Volksschule 2 in Freistadt

**2000 bis 2004:** Musikhauptschule Freistadt

**2004 bis 2005:** Polytechnischer Lehrgang im Fachbereich Mechatronik in Freistadt

### Berufliche Weiterbildung:

**2010:** Ausbildung zum geprüften Dampfkessel-, Dampfturbinen- und Gasturbinenwärter

**2016:** Lehrgang Wasserchemie für Mitarbeiter der Voest Alpine Stahl GesmbH

**2016:** Theoretischer Ausbildungsteil des Kraftwerker-Lehrganges der KWS

**Derzeit:** Ausbildung zum Werkmeister im Fachbereich Mechatronik

## 1.1 Persönliche Vorstellung

Werter Leser, da Sie sich die Mühe machen, die folgenden Seiten zu lesen, lassen Sie mich meine Person kurz vorstellen. Mein Name ist Panholzer Emmanuel, derzeit beschäftigt bei der Voest Alpine Stahl GesmbH, Abteilung TSS, als Anlagenbediener der Kraftwerksblöcke des Kalorischen Kraftwerkes.

Ich wurde am 14.02.1990 in Freistadt geboren wo ich auch meine schulische Ausbildung bis zum Polytechnikum absolvierte.

2005 trat ich dann als Produktionstechnikerlehrling in die Voest Alpine eine, wo ich bis heute beschäftigt bin. Zwischen 2016 und 2017 durfte ich die Ausbildung zum geprüften Kraftwerker an der KWS-Essen absolvieren.

In meiner Freizeit beschäftige ich mich viel mit Musik, Elektronik, Brett- und Videospielen und gehe aktiv dem Airsoftsport nach.

Die Idee für diese Projekt kam mir nach einem Ausflug zur Lasertaganlage Ampflwang (<http://www.hausruckpark.at/lasertaginfos>). Da eine professionelle Lasertaganlage nur relative schwer als Privatperson zu bekommen ist und die Kosten dafür schnell über 1000 € gelangen, war die Idee geboren, selbst ein System zu bauen, und um die Verbreitung zu erleichtern habe ich mich dazu entschieden, das Projekt später zu veröffentlichen und für jedermann frei zugänglich zu machen. Ich hoffe durch diesen Schritt der Verbreitung dieses Sportes zu helfen und vielleicht dem einen oder anderen auch den Einstieg in die Welt der Elektronik zu ermöglichen.

## Inhalt

|  |           |
|--|-----------|
| <b>1 Lebenslauf</b>  | <b>2</b>  |
| 1.1 Persönliche Vorstellung                                    | 3         |
| <b>2 Vorwort</b>   | <b>8</b>  |
| 2.1 Begrifflichkeiten  | 8         |
| 2.2 GNU-Lizenz   | 8         |
| 2.3 Danksagung   | 9         |
| 2.4 Was ist Lasertag?  | 10        |
| 2.5 In eigener Sache   | 10        |
| 2.6 Ein paar Gedanken zur Vermeidung politischer Debatten      | 11        |
| 2.7 Anmerkung zur rechtlichen Lage in Österreich               | 12        |
| <b>3 Kurzfassung</b>   | <b>14</b> |
| <b>4 Einleitung</b>  | <b>15</b> |
| 4.1 Sicherheitshinweise  | 15        |
| 4.2 Ziel des FreeTag-Lasertagsystems                           | 15        |
| 4.3 Die drei unterschiedlichen Teile des Projektes             | 16        |
| 4.3.1 Für die Reproduktion des Projektes                       | 17        |
| 4.4 Über den Aufbau dieses Dokumentes                          | 18        |
| <b>5 Pflichtenheft</b>   | <b>19</b> |
| 5.1 Ziel der Entwicklung                                       | 19        |
| 5.1.1 Musskriterien  | 19        |
| 5.1.2 Wunschkriterien  | 20        |
| 5.1.3 Abgrenzungskriterien                                     | 20        |
| 5.2 Produkteinsatz   | 20        |
| 5.2.1 Anwendungsbereich  | 20        |
| 5.2.2 Zielgruppe   | 21        |
| 5.2.3 Betriebsbedingungen                                      | 21        |
| 5.3 Produktumgebung  | 21        |
| 5.3.1 Software   | 21        |
| 5.3.2 Hardware   | 21        |
| 5.4 Produktdaten   | 22        |
| 5.5 Produktleistungen  | 22        |
| 5.6 Qualitätszielbestimmung                                    | 24        |
| 5.7 Entwicklungsumgebung                                       | 24        |
| 5.7.1 Software   | 24        |
| 5.7.2 Hardware   | 24        |
| <b>6 Betrieb der Freetag-Lasertaganlage</b>                    | <b>25</b> |
| 6.1 Mindestvoraussetzung zum Betrieb des Systems               | 25        |
| 6.2 Erstinbetriebnahme des Arduino Mega und dem Lasertagsystem | 25        |
| 6.2.1 Installieren der Arduino Programmiersoftware             | 26        |
| 6.2.2 Einrichten des Arduino Mega                              | 26        |

|   |           |
|---|-----------|
| 6.2.3 Hochladen des ersten Programmes und Funktionstest des Arduino Mega            | 27        |
| 6.2.4 Installieren von externen Libraries in die Arduino-Programmiersoftware        | 29        |
| 6.2.5 Installieren der FreeTag-Lasertagsoftware                                     | 30        |
| 6.2.6 Einstellen des Zählwertfehlers divisorstart, divisorzero und divisorone       | 30        |
| 6.2.7 Einstellen der Spieleparameter  | 37        |
| 6.2.8 Übertragen der Soundfiles auf die Micro-SD Karte                              | 41        |
| 6.2.9 Einlegen der SD-Karte   | 43        |
| 6.2.10 Kontrolle der Erstinbetriebnahme   | 44        |
| 6.3 Spieleparameter   | 45        |
| 6.4 Spielmodi   | 49        |
| 6.4.1 Deathmatch (Jeder gegen Jeden)  | 49        |
| 6.4.2 Team Deathmatch (Team gegen Team)   | 49        |
| 6.4.3 One Against All (Einer gegen alle)  | 50        |
| 6.4.4 King of the Hill  | 50        |
| 6.4.5 Capture the Flag (mit physischer Flagge)                                      | 50        |
| <b>7 Störungsbehebung und Wartung</b>   | <b>51</b> |
| 7.1 Der Markierer lässt sich nicht einschalten                                      | 51        |
| 7.2 Das LCD-Display zeigt nichts an   | 51        |
| 7.3 Das LCD-Display zeigt eigenartige Zeichen an                                    | 52        |
| 7.4 Die Hintergrundbeleuchtung des Displays funktioniert nicht mehr                 | 52        |
| 7.5 Das Betätigen des Abzuges hat keinen Effekt.                                    | 52        |
| 7.6 Obwohl ich auf Mitspieler ziele und den Abzug betätige, wird er nicht getroffen | 52        |
| 7.7 Die Soundwiedergabe des Markierers funktioniert nicht                           | 53        |
| 7.8 Der Markierer verströmt einen brandähnlichen Geruch                             | 53        |
| <b>8 Technische Grundlagen</b>  | <b>54</b> |
| 8.1 Übertragungsfrequenz  | 54        |
| 8.2 Übertragungsprotokoll   | 54        |
| 8.3 Zykluskorrektur divisorstart, divisorzero und divisorone                        | 58        |
| 8.4 Treiberschaltung LED  | 59        |
| 8.5 IR-Licht Bündeln  | 63        |
| 8.6 IR-Empfänger  | 66        |
| 8.7 Erstellen Eigener Sounds  | 68        |
| <b>9 Begrifflichkeiten</b>  | <b>71</b> |
| 9.1.1 Airsoft   | 71        |
| 9.1.2 Arduino   | 71        |
| 9.1.3 COM-Port  | 71        |
| 9.1.4 Datenpaket  | 71        |
| 9.1.5 Empfänger   | 72        |
| 9.1.6 GNU-Lizenz  | 72        |
| 9.1.7 Header  | 72        |
| 9.1.8 Hit-LED   | 72        |
| 9.1.9 Hoch- und Tiefpassfilter  | 72        |
| 9.1.10 Indoor   | 73        |

---

|           |   |           |
|-----------|---|-----------|
| 9.1.11    | IR-LED                                  | 74        |
| 9.1.12    | Kondensator                             | 74        |
| 9.1.13    | LED                                     | 76        |
| 9.1.14    | Libraries                               | 78        |
| 9.1.15    | Markierer, Tagger oder Phaser           | 78        |
| 9.1.16    | MilesTag 2 Protokoll                    | 79        |
| 9.1.17    | Muzzle-LED                              | 79        |
| 9.1.18    | N-Kanal MOSFET                          | 79        |
| 9.1.19    | Open Source                             | 79        |
| 9.1.20    | Outdoor                                 | 80        |
| 9.1.21    | Paintball                               | 80        |
| 9.1.22    | Physical Computing                      | 80        |
| 9.1.23    | Respawn                                 | 80        |
| 9.1.24    | Respawnzeit                             | 80        |
| 9.1.25    | Sender                                  | 80        |
| 9.1.26    | Sensorkopf                              | 80        |
| 9.1.27    | Sketch                                  | 81        |
| 9.1.28    | Spielerpaket                            | 81        |
| 9.1.29    | Trigger                                 | 81        |
| 9.1.30    | Widerstand (Bauteil)                    | 81        |
| <b>10</b> | <b>Elektropläne und Teileliste</b>      | <b>83</b> |
| 10.1      | Teileliste                              | 85        |
| 10.2      | PCB Layouts                             | 87        |
| 10.2.1    | Layout Hauptplatine Markierer           | 88        |
| 10.2.2    | Layout Platine Sensorkopf               | 88        |
| <b>11</b> | <b>Mechanischer Plan und Teileliste</b> | <b>90</b> |
| 11.1      | Mechanische Bauteilliste                | 93        |
| <b>12</b> | <b>Markierer Montiert</b>               | <b>94</b> |
| <b>13</b> | <b>Kalkulationen</b>                    | <b>95</b> |
| 13.1      | Vorkalkulation                          | 95        |
| 13.2      | Nachkalkulation                         | 95        |
| 13.2.1    | Planung Elektronik                      | 95        |
| 13.2.2    | Planung Mechanik                        | 95        |
| 13.2.3    | Fertigung Elektronik                    | 95        |
| 13.2.4    | Fertigung Mechanik                      | 95        |
| 13.2.5    | Programmierung                          | 96        |
| 13.2.6    | Tests                                   | 96        |
| 13.2.7    | Dokumentation                           | 96        |
| <b>14</b> | <b>CE-Zertifizierung</b>                | <b>97</b> |
| 14.1.1    | Bedeutung                               | 97        |
| 14.1.2    | Kennzeichnung                           | 97        |
| 14.1.3    | Zuständigkeit                           | 97        |

---

## FREETAG LASERTAGSYSTEM

---

|   |            |
|---|------------|
| 14.1.4 Vorgangsweise  | 98         |
| 14.1.5 Anbringung   | 98         |
| 14.2 6-Schritte zur CE-Kennzeichnung  | 99         |
| 14.2.1 SCHRITT 1 – Finden Sie heraus, welche Richtlinie(n) und harmonisierte Normen für Ihr Produkt gelten                    | 99         |
| 14.2.2 SCHRITT 2 – Ermitteln Sie die spezifischen Bedingungen für Ihr Produkt   | 99         |
| 14.2.3 SCHRITT 3 – Stellen Sie fest, ob eine benannte Stelle für ein Konformitätsbewertungsverfahren herangezogen werden muss | 99         |
| 14.2.4 SCHRITT 4 – Testen Sie das Produkt und überprüfen Sie seine Konformität  | 100        |
| 14.2.5 SCHRITT 5 – Stellen Sie die notwendige technische Dokumentation zusammen und halten Sie sie zur Einsicht bereit        | 100        |
| 14.2.6 SCHRITT 6 – Bringen Sie die CE-Kennzeichnung an Ihrem Produkt an und verfassen Sie die EU-Konformitätserklärung        | 100        |
| 14.3 Richtlinien und harmonisierte Normen für dieses Projekt  | 100        |
| 14.3.1 Beschränkung gefährlicher Stoffe in Elektro- und Elektronikgeräten   | 100        |
| 14.3.2 Elektromagnetische Verträglichkeit (EMV-Richtlinie)  | 102        |
| 14.4 EU Konformitätserklärung   | 104        |
| <b>15 Risikobeurteilung</b>   | <b>105</b> |
| <b>16 Eidesstattliche Erklärung</b>   | <b>108</b> |
| <b>17 Abbildungs- und Tabellenverzeichnis</b>   | <b>109</b> |
| <b>18 Datenblattverzeichnis</b>   | <b>113</b> |

## 2 VORWORT

In diesem Dokument wird, aus Gründen der Einfachheit, lediglich die männliche Form eines Wortes verwendet, es sind dabei jedoch stets beide Geschlechter, oder auch Personen, die sich keinem der beiden zugehörig fühlen, sowie Individuen oder kollektive Lebensformen, welche fähig sind, die nachfolgend beschriebene Anlage in Betrieb zu nehmen und zu verwenden, gemeint.

Sollte ich trotz aller Vorsichtsmaßnahmen jemandem zu nahe treten, etwaige Genderrichtlinien verletzen oder mich gar ihm Ton vergreifen, so möchte ich mich bereits im Vorhinein entschuldigen. Es ist nicht mein Ziel, irgendwelche Personen oder Lebensformen in diesem Dokument zu diskriminieren oder gar zu verletzen.

Alle von mir erstellten Dateien und Programme, auf welche ich in dieser Arbeit verweise, werden erst nach meiner Abschlussprüfung veröffentlicht

### 2.1 Begrifflichkeiten

Ich werde mit Beginn des Dokumentes Wörter, Ausdrücke und Bezeichnungen verwenden, die vielleicht nicht jedem Leser verständlich sind. Sollte ein Begriff nicht klar sein, so bitte ich Sie, zuerst im später folgenden Hauptkapitel Begriffe nachzuschlagen, und sollte dort keine oder nur eine unzulängliche Beschreibung gefunden werden, das Internet zu bemühen. Manche der Begriffe in diesem Dokument habe ich von anderen Themengebieten abgeleitet und sind nicht in diesem Zusammenhang in der Literatur oder dem Internet zu finden.

### 2.2 GNU-Lizenz

Das komplette Dokument ist zur freien Vervielfältigung, Abänderung und Verwendung gedacht, solange die Richtlinien der GNU-Lizenz für freiheitsgewährende Dokumentationen eingehalten werden.

Die genauen Lizenzrichtlinien finden Sie im Internet in englischer Sprache unter „<http://www.gnu.org/licenses/fdl-1.3.html>“.

Es gibt eine inoffizielle deutsche Übersetzung, welche jedoch keine Rechtsgültigkeit besitzt. Sie ist unter folgender Adresse im Internet zu finden: „[http://www.selflinux.org/selflinux/html/gfdl\\_de.html](http://www.selflinux.org/selflinux/html/gfdl_de.html)“.

Um dem werten Leser das Studieren der Lizenzrichtlinien zu ersparen eine kurze Zusammenfassung der wichtigsten Punkte die den Inhalt dieses Dokumentes und der darin Beschriebenen Anlage betreffen:

- Sämtliche Informationen in diesem Dokument, sowie die Software die zur Anlage gehört, dürfen frei verwendet und vervielfältigt werden, sofern damit keinerlei Profit erzeugt wird.
- Der Autor (also ich) der Informationen ist bei Vervielfältigung zu nennen.

### 2.3 Danksagung

Viele Menschen haben mich im Laufe dieses Projektes begleitet, mir ihr Ohr geliehen oder sind mir mit Rat und Tat zur Seite gestanden. Um meine Wertschätzung auszusprechen möchte ich ihnen hier danken und sie hier nennen, wobei die Reihenfolge der Nennung komplett zufällig geschehen ist.

Beginnen möchte ich mit Thomas Hartl, der mich zu meinem ersten Lasertagspiel nach Ampflwang einlud, wo mir die Idee für dieses Projekt kam. Weiters möchte ich Roland Zopf danken der mir bei diversen Designfragen beistand und auch beim Design des Gehäuses seine Finger im Spiel hatte.

Ich danke Alexander Pavlik, der mir des Öfteren bei den Signalreichweitentests geholfen hat, und Sebastian Eschlböck, der mir die komplexeren Befehle der Programmiersprache C beigebracht hat.

Besonderen Dank schulde ich Rene Reisinger welcher das Makierergehäuse nach meinen Skizzen entworfen und auch gedruckt hat.

Elisabeth Eibensteiner danke ich vielmals, dass sie diese Arbeit korrigiert hat, was ihr sicher viele Nerven gekostet hat.

Natürlich danke ich auch meinen Eltern, ohne die ich heute nicht hier auf diesem Planeten weilen würde. Ich danke auch den Kaffeebauern, welche meinen Kaffee geerntet haben, ohne dessen Koffeingehalt dieses Projekt nie möglich gewesen wäre.

Auch möchte ich mich bei den vielen Menschen da draußen bedanken, welche ihr Wissen in Form von Tutorials oder Ähnlichem im Internet kostenlos mit anderen teilen und ohne die ich niemals das nötige Wissen erhalten hätte, um dieses Projekt zu verwirklichen. In diesem Zusammenhang möchte ich vor allem die Personen hinter den Webseiten <http://www.pewduino.org/> und <http://www.lasertagparts.com/mtdesign.htm> hervorheben, deren Informationen mir gerade am Anfang sehr geholfen haben und ohne jene es dieses Projekt nicht gäbe.

## FREETAG LASERTAGSYSTEM

---

Ich möchte mich auch bereits jetzt schon bei jedem bedanken, der an diesem Projekt weiterarbeitet und seine Erfahrungen mit anderen teilt.

### 2.4 Was ist Lasertag?

Um dieses Projekt und seinen Sinn zu verstehen, sollte man zuerst verstehen, worum es sich bei Lasertag handelt.

Das Ziel von Lasertag ist es, ein Spiel zwischen mindestens zwei Personen zu ermöglichen. Die Gewinnparameter des Spiels können unterschiedlich sein, was jedoch bei fast jedem Spielmodus gleich ist, ist dass man gegnerische Spieler, oder sogar Mitspieler des eigenen Teams, mit dem bei Betätigung des Triggers (also des Abzugs des Markierers) via Infrarotsignal treffen kann. Das Infrarotsignal, welches vom Markierer abgegeben wird, muss dabei auf das Ziel gerichtet werden, um das Gegenüber zu treffen. Sollte das Signal beim Gegenspieler ankommen, so wird es von den Sensoren verarbeitet und ihm werden Lebenspunkte in Höhe der Schadensmenge des Senders abgezogen. Fallen die Lebenspunkte eines Spielers auf null, so wird entweder für kurze Zeit sein Markierer deaktiviert oder er scheidet gar für den Rest des Spieles aus (je nach Spielmodus und Einstellung des Systems).

Lasertag ist mit Sportarten wie Airsoft oder Paintball vergleichbar, es werden jedoch keine Kugeln oder sonstiges verschossen, sondern lediglich Infrarotsignale übertragen, was das Verletzungsrisiko und die Schmerzen während des Spieles stark reduziert. Weiter ist eine genaue Auswertung der Daten möglich, da jeder erfolgreiche Treffer gespeichert wird und später für eine Punktevergabe herangezogen werden kann.

Die Sensoren zum Erfassen eingehender Signale oder Treffer können beliebig am Körper angebracht werden, es hat sich jedoch aus spielerischer Sicht als sinnvoll erwiesen ein paar Sensoren auf den Kopf und zumindest einen Sensor auf dem Markierer anzubringen. Das System, in der Form wie es in dieser Arbeit dargestellt wird, ist nicht in der Lage zwischen den unterschiedlichen Sensorpositionen zu unterscheiden, sogenannte „Headshots“, wie sie aus Videospielen bekannt sind, sind nicht möglich. Ich habe diese Funktion bewusst nicht eingefügt, weil mir eine Schadensauswertung nach unterschiedlichen Trefferzonen bereits zu weit in Richtung Simulation und somit weg vom Spiel geht.

### 2.5 In eigener Sache

Es handelt sich hier zwar um meine Werkmeisterarbeit für den Fachbereich Mechatronik, doch möchte ich hiermit nicht nur meine Projektarbeit dokumentieren, sondern auch die Möglichkeit nutzen dieses Projekt der Öffentlichkeit zugänglich zu machen.

## FREETAG LASERTAGSYSTEM

---

Ich hoffe damit einen Start und eine Plattform geschaffen zu haben, damit sich jeder interessierte Hobbyist selbst eine Anlage bauen, modifizieren und betreiben kann. Mein Traum wäre es, dass sich unterschiedlichste Menschen dadurch begeistern lassen, sich dem Sport zu widmen, sich kreativ mit dieser Technik auseinander zu setzen und dieses Projekt weiterzuentwickeln. Wenn dann noch jeder seine Weiterentwicklungen teilt, könnte sich dies zu einem „lebendigen“ Projekt entwickeln, dessen Möglichkeiten noch lange nicht erschöpft sind.

### 2.6 Ein paar Gedanken zur Vermeidung politischer Debatten

Da es sich bei diesem Projekt um eine Lasertaganlage handelt, deren Ziel es ist, mit der Übertragung eines Infrarotsignales, eine „moderne Form“ des Räuber-und-Gendarm-Spieles zu ermöglichen, ist ein Vergleich mit Kriegsspielen naheliegend.

Die Anlage ist dazu konzipiert ein Signal zu übertragen, welches, je nach Spielmodus, dazu gedacht ist, beim Empfänger eine gewisse Anzahl seiner „Lebenspunkte“ abzuziehen. De Facto ist somit das Treffen eines Gegenspielers mit dem vom Markierer zur Verfügung gestellten Infrarotsignal und das Verarbeiten dieses Signales die Aufgabe dieser Anlage.

Das Konzept dieses Projektes lässt einen gewissen Vergleich mit anderen Sportarten wie z. B. Paintball und Airsoft zu. Weiter ist eine gewisse Nähe zu Shooterspielen nicht abzustreiten.

Ich möchte darauf hinweisen, dass ich diese Anlage geplant habe, um damit eine Variante eines Spieles zu ermöglichen, welche zwar für manchen eine Form von Kriegssimulation darstellt, und ja dieser Vergleich ist gerechtfertigt, aber es doch jedem selbst überlässt, ob er diese Form des Spieles bzw. Sportes gutheit. Persönlich möchte ich mich von jedem gewaltverherrlichten Gedankengut distanzieren, denn es ist nicht das Ziel dieses Projektes, Hass, Zwietracht oder andere Feindseligkeiten zu schüren.

Da es jedoch ein wichtiger Teil dieses Projektes ist, jegliche Information frei für jeden zu Verfügung zu stellen, kann und will ich auch nicht die Verantwortung über die Verwendung dieses Projektes übernehmen.

Sollten also Teile dieses Projektes dazu verwendet werden, um „Krieg zu spielen“ oder andere nicht gesetzeskonforme Tätigkeiten durchzuführen, so liegt die Verantwortung nicht bei diesem Projekt, sondern bei der Person, die den Unfug anstellt.

Weiters werde ich im Laufe dieser Dokumentation Wörter verwenden, welche aus anderen Gebieten entlehnt sind und je nach Kontext umstritten sein können.

## FREETAG LASERTAGSYSTEM

---

Ich werde des Öfteren den Begriff *Feuern* für das Auslösen der Signalübertragung verwenden. Mir ist bewusst, dass auf Menschen zu *feuern* als Ausdruck umstritten ist, man sollte jedoch bedenken, dass es sich hier um keine Waffe handelt, sondern um ein Spielzeug.

Das Spielzeug werde ich als Anlehnung an das Sportgerät aus dem Paintball als Markierer, Tagger oder als Phaser bezeichnen, um keine Nähe zu reellen Waffen herzustellen.

Die Form des Gehäuses ist der Form eines Sturmgewehres sehr ähnlich, dies folgt daraus, dass die Waffenindustrie ihre Produkte ergonomisch formt und diese Ergonomie sehr gut beherrscht. Da der Markierer sowie eine Schusswaffe ein ähnliches Ziel verfolgen, war es naheliegend, das Design und die Ergonomie des Markierers an die einer reellen Schusswaffe anzulehnen.

### 2.7 Anmerkung zur rechtlichen Lage in Österreich

**Ich übernehme keinerlei Haftung für die Richtigkeit der hier getroffenen Aussagen, ich möchte lediglich ein paar Hinweise liefern, um ein Lasertagspiel im gesetzlichen Rahmen zu halten.**

Laut österreichischem Waffenschutzgesetz ist das offene Führen von Waffen oder waffenähnlichen Nachbildungen verboten, d.h. sollte der Markierer aussehen wie eine Waffe, so darf er nicht erkenntlich getragen werden, was wiederum bedeutet das ein Spiel im öffentlichen Raum (z. B. auf einem Spielplatz) untersagt ist.

Dies dient dazu, Panik von Außenstehenden zu vermeiden. Man sollte daher beim Design des Phasers darauf achten, welche Form man wählt, und welche Geräusche bei betätigen des Triggers vom Gerät abgespielt werden. Leider sind die Grenzen dessen, was unter waffenähnlich fällt oder nicht, nicht klar definiert, z. B. fällt ein Spielzeug des Herstellers Hasbro, welches unter den Namen NERV vertrieben wird, nicht unter das Waffenschutzgesetz, obwohl die Form doch sehr waffenähnlich ist, die Lackierung ist jedoch sehr bunt und kindlich gestaltet, was eine Differenzierung zu einer Waffe zulässt. Außerdem verschießen die Spielzeuge nur Schaumstoffdarts und machen keine Geräusche.

Sollten das Phaserdesign und die Geräusche, die der Makierer wiedergibt, so gewählt sein, dass ein Vergleich mit einer reellen Waffe nur sehr schwer möglich ist, so sollte einem Spiel im öffentlichen Raum nichts im Wege stehen, im Zweifelsfall sollte jedoch die örtliche Polizeidienststelle befragt werden.

## FREETAG LASERTAGSYSTEM

---

Wenn das Design des Phasers einer Waffe ähnelt, und man trotzdem ein Spiel durchführen möchte, so sind einige Dinge zu beachten. Erstens muss das Spielgelände abgesperrt werden (z. B. mit Absperrband) und weiters sollten Schilder mit Warnhinweisen angebracht werden und die örtliche Polizeidirektion muss verständigt werden. Außerdem sind Spiele mit derartiger Ausrüstung, gerade falls man auch Tarnkleidung verwendet, nur als eingetragener Verein zulässig, ansonsten könnte ein solches Spiel als paramilitärische Übung interpretiert werden.

Ich möchte mit diesen Zeilen nicht den Spaß am Spiel drosseln, ich spreche jedoch aus Erfahrungen, welche ich im Bereich des Airsoftsports gemacht habe.

### 3 KURZFASSUNG

Die Aufgabe des FreeTag Lasertagsystems ist es, eine offene und modifizierbare Basis für jedermann mit technischem Verständnis zu bieten. Die komplette Dokumentation sowie der Programmcode sind vollkommen frei zugänglich, werden unter der GNU-Lizenz für freiheitsgewährende Dokumente veröffentlicht und dürfen, sofern keine kommerziellen Ziele damit verfolgt werden, verändert und verbreitet werden.

Mein Persönliches Ziel dieses Projektes war es, eine Plattform zu schaffen, um Laser- tag als Sport und Spiel für jedermann zu ermöglichen, ohne dabei über einen großen finanziellen Rahmen zu verfügen. Dies war nur dadurch möglich, dass ich mit möglichst einfachen Mitteln gearbeitet habe. Es erfordert auch einen gewissen technischen und handwerklichen Einsatz, selbst eine dieser Anlagen zu bauen und in Betrieb zu nehmen.

Der Markierer des FreeTag Lasertagsystem versendet ein Infrarotsignal, welches über eine Linse gebündelt wird und von einer anderen Anlage empfangen wird. Dabei wird vom Sender ein Signal übermittelt, welches Daten wie Teamnummer, Schadenspunkte, Spielernummer usw. überträgt. Die auswertende Einheit übernimmt bei erfolgreicher Übertragung dieses Signal, über einen Infrarotempfänger, verwertet und speichert es. Sollten, je nach gewähltem Spielmodus, die übermittelten Daten relevant sein, so werden diese auf einem Display am Markierer dargestellt. Sollten die Lebenspunkte eines Spielers auf null fallen, weil er z. B. zu oft von einem gegnerischen Spieler getroffen wurde, so wird sein Markierer gesperrt und er scheidet, für eine frei bestimmbare Zeit, aus dem Spiel aus.

Viele Parameter wie Spielername, Spielernummer, Schadenspunkte, Lebenspunkte, Anzahl der Wiedereinstiege usw. können in der Software frei eingestellt werden, um unterschiedliche Spielerlebnisse zu erzeugen.

Als Programmierplattform und Steuerungseinheit habe ich den Arduino Mega gewählt, da diese auch von Laien leicht zu bedienen ist, dabei jedoch auch sehr komplexe Aufgaben übernehmen kann, und genügend Ein- und Ausgänge sowie genügend RAM-Speicher besitzt, um die doch sehr umfangreiche Datenmenge für bis zu 64 Spieler zu meistern. Es gibt weiters sehr gute und auch viele Lehrdokumente und Videos für die Arduino Programmierumgebung, welche den Einstieg und die Weiterentwicklung des Programmcodes erleichtern.

## 4 EINLEITUNG

### 4.1 Sicherheitshinweise

Der Betrieb einer Lasertaganlage ist zwar nicht mit vielen, jedoch trotzdem mit Risiken verbunden. Ich möchte gleich darauf hinweisen, dass ich keinerlei Haftung für Schäden an der Anlage oder an Personen übernehme.

Da die Anlage mit Strom arbeitet kann es zu elektrischen Kurzschlüssen durch Materialermüdung oder falschen Umgang mit Komponenten kommen. Sollte der Markierer oder ein anderes Bauteil brandähnliche oder kokelnde Gerüche erzeugen so ist sofort die Stromzufuhr über den Schlüsselschalter zu unterbinden, die Akkumulatoren zu entfernen und die Anlage darf erst nach Überprüfung wieder in Betrieb genommen werden.

Da Infrarotlicht schädlich für das Auge sein kann, darf der Lauf nicht direkt auf das Auge gerichtet werden. Ich empfehle auch, den Abzug unter einer Distanz zum Ziel von 20 cm nicht zu betätigen.

Durch das Spiralkabel, durch das die Stirnbandsensoren mit dem Tagger verbunden sind, kann es zu Strangulationen kommen.

Generell sollte die Anlage von Kindern ferngehalten werden oder ihnen nur unter Aufsicht von Erwachsenen der Umgang damit gestattet werden.

Die Anlage ist von Wasser und Feuchtigkeit fernzuhalten um Schäden an der Elektronik zu vermeiden.

Damit das Gehör nicht geschädigt wird, sollte die Lautstärke der Soundwiedergabe so leise als möglich eingestellt werden.

### 4.2 Ziel des FreeTag-Lasertagsystems

Wie im Vorwort bereits erwähnt, ist das Ziel dieses Projektes ein Lasertagsystem für Hobbyisten zu erstellen, welches für jedermann zugänglich ist und ohne zu großen finanziellen Aufwand verwirklichbar ist. Es gibt natürlich bereits ausgereifte und sehr gute Systeme am Markt, jedes mir bekannte ist jedoch entweder sehr teuer oder durch die effektive Reichweite des Infrarotsignales für mich nicht brauchbar. Mir gefällt außerdem der Gedanke, dass durch die Veröffentlichung aller Daten das Projekt von jedem weiterentwickelt oder für seine Bedürfnisse angepasst werden kann. Solange das

## FREETAG LASERTAGSYSTEM

---

Datenübertragungsprotokoll eingehalten wird, können unterschiedliche Systeme miteinander kommunizieren. Somit ist es möglich, dass verschiedenste Abwandlungen dieses Projektes miteinander arbeiten, oder anders ausgedrückt: unterschiedlichste Personen mit unterschiedlichster Ausrüstung (welche auf diesem Projekt oder dem Datenübertragungsprotokoll basieren) können zusammen Lasertag spielen.

Da ich die Kosten und den technischen Aufwand so gering wie möglich gehalten habe, ohne dabei zu viele Kompromisse einzugehen, sollte dieser Sport/dieses Spiel für jeden zugänglich sein.

Mir war es außerdem wichtig, das Schummeln während des Spielens so schwierig wie möglich zu gestalten, um den Spielspaß so groß wie möglich zu gestalten. Dies ist dadurch möglich, dass jedes Signal, welches vom Markierer ausgegeben wird, vom Empfänger genau ausgewertet wird, und sollten die Parameter des Datenübertragungsprotokolles eingehalten werden (mehr zum Datenübertragungsprotokoll erfahren Sie im nächsten Kapitel), so wird das Signal bis zum Neustart des Systems im empfangenden System gespeichert. Die gespeicherten Daten können auch abgefragt werden, um eine Punkteauswertung durchzuführen. Diese Abfrage muss über eine externe Software abgewickelt werden, welche jedoch nicht Teil dieser Werkmeisterarbeit ist. Sobald sie jedoch verfügbar ist und funktioniert werde ich sie veröffentlichen. Die Grundsteine für eine genaue Auswertung sind jedoch bereits in der Projektsoftware enthalten.

Die Hardware und Elektronik kann frei gestaltet werden, um jedoch die von mir programmierte Software eins zu eins zu übernehmen, muss der Arduino Mega als Steuerinheit für das System gewählt werden und die Pinbelegung und die Treiberschaltung für das IR-LED mit der aus dieser Dokumentation übereinstimmen.

Da es sich hier um eine Abschlussarbeit für die Meisterschule handelt, beinhaltet diese auch einen mechanischen Teil, welche ich in Form des Markierergehäuses abliefer. Der Aufbau dieses Projektes lässt jedoch zu (ich befürworte es sogar), dass jeder das Gehäuse verwendet, das ihm am besten gefällt, und derjenige, der zuerst ein Gehäuse in Form einer Ente verwendet, wird von mir auf ein Getränk eingeladen.

Des Weiteren soll auch eine PC-Software zur Auswertung und Ausgabe der gespeicherten Signale im Markierer zur Verfügung gestellt werden. Diese wird jedoch von Marco Eibensteiner programmiert und soll nicht Teil dieses Projektes sein.

### 4.3 Die drei unterschiedlichen Teile des Projektes

Das Projekt lässt sich grob in drei Teile gliedern und zwar in den mechanischen, den elektrischen und den Programmierteil. Zum mechanischen Teil zähle ich die Fertigung des Markierergehäuses, als elektrischen Teil ist alles von der Platinenfertigung und der

## FREETAG LASERTAGSYSTEM

---

Verkabelung zu verstehen und der Programmienteil umfasst die komplette Software welche für Arduino Mega geschrieben wurde.

Für die hier geschriebene Meisterarbeit hat jeder Teil die gleiche Wichtigkeit, da sonst das Projekt als Ganzes nicht funktionieren würde.

### 4.3.1 Für die Reproduktion des Projektes

Für denjenigen der diese Arbeit liest, um selbst eine Lasertaganlage zu bauen, ist der mechanische Teil nur dann relevant, wenn er das Gehäuse nachbauen möchte, was natürlich nicht nötig ist. Die 3D-Datei mit den nötigen Daten für den 3D-Drucker ist unter <https://github.com/EmmanuelPanholzer/FreeTag-Lasertag> im Ordner Gehäuse zu finden und kann verwendet werden (natürlich nur nicht kommerziell). Ich kann nur nochmals zur Kreativität auffordern, was das Gehäuse anbelangt, ich habe zum Beispiel auch bereits einen Markierer aus einem Spielzeug der Marke Nerf gebaut.

Beim elektrischen Teil empfehle ich, sich an meine Vorgaben zu halten, da ich die Software auf die Komponenten bzw. die Pinbelegung des Arduino Mega angepasst habe. Es kann natürlich auf Dinge wie die Soundwiedergabe, verzichtet werden. Die Schaltung muss mindestens aus der IR-LED Treiberschaltung, einem Empfänger, dem Trigger, dem LCD-Display und dem Reloadbutton bestehen, damit die von mir geschriebene Software funktioniert. Die Layouts für die Hauptplatine und den Empfänger sowie die dazu gehörigen Programmdateien für das Programm KiCAD sind unter <https://github.com/EmmanuelPanholzer/FreeTag-Lasertag> im Ordner PCB Layouts zu finden und können natürlich verwendet werden, um sich selbst eine Platine herzustellen oder herstellen zu lassen. Im Punkt Elektropläne und Teileliste ist der Schaltplan für die Grundausführung des Markierers zu finden, mit diesem kann man die Schaltung auch durchaus auf einer Lochrasterplatine verwirklichen, was die räumliche Anpassung an diverse Gehäuseformen ermöglicht.

Die Projektteile, welche nicht unbedingt notwendig sind, um das System nachzubauen und zu betreiben, werde ich bei den jeweiligen Punkten dieser Arbeit aufzeigen, wobei ich jedoch jedem empfehle, sich an dieses Projekt zu halten, zum einen, da es viele Möglichkeiten beinhaltet, welche ich als sinnvoll erachte, und zum anderen, da das System schon viele Tests und Modifikationen durchlaufen hat, um das bestmögliche Spielerlebnis zu erzeugen.

Ich habe auch bei der hier vorgestellten Elektronik bereits einige zusätzliche Funktionen eingeplant, welche jedoch noch nicht in der Software verwirklicht wurden, da es die Aufgabe dieses Projektes übersteigen würde. Ich werde aber im Kapitel zur Elektronik darauf hinweisen, welche Teile noch nicht in der Software implementiert sind und welche Funktionen ich mir für diese vorstellen kann.

## FREETAG LASERTAGSYSTEM

---

Bei der Beschaffung der Materialien habe ich versucht, so wenig unterschiedliche Händler wie möglich zu verwenden, um die Versandkosten so gering wie möglich zu halten. Die Elektronikkomponenten stammen von Conrad und Reichelt. Natürlich lassen sich gewisse Komponenten auch günstiger erwerben als bei den beiden Elektronikhändlern, aber bei beiden hatte ich nie Probleme und sie liefern auch kleine Mengen.

Das Montagematerial für die Linse und deren Halterung stammt aus dem Baumarkt. Das dies nicht die professionellste Lösung ist, ist mir durchaus bewusst, es ist jedoch die günstigste und wie ich finde auch einfachste (und nach dem Lackieren fällt es auch keinem mehr wirklich auf).

Dem interessierten Leser, der plant selbst eine FreeTag-Lasertaganlage zu bauen, wünsche ich viel Erfolg und Glück und hoffe, dass man sich eventuell einmal auf dem Spielfeld begegnet.

### 4.4 Über den Aufbau dieses Dokumentes

Da diese Arbeit zum einen dazu dient meine Werkmeisterarbeit schriftlich zu dokumentieren und gleichzeitig als eine Anleitung zum Nachbau einer Lasertaganlage dienen soll, möchte ich hier kurz ein paar Anmerkungen machen.

Der Leser, welchem es nur darum geht, einen FreeTag-Lasertagmarkierer nachzubauen, kann Punkte wie das Pflichtenheft, Kalkulation, CE-Zertifizierung, Risikobeurteilung und die Risikobeurteilung überspringen, sie sind Bestandteil meiner Werkmeisterarbeit, sind aber für den Bau einer Lasertaganlage nicht von Relevanz.

Ich habe bewusst den Programmcode für das Arduino nicht in dieses Dokument kopiert, da er, so wie er in der Arduinoprogrammierumgebung dargestellt wird, 99 Seiten umfasst. Anders Formatiert wirkt er jedoch sehr unübersichtlich, weswegen ich mich dagegen entschieden habe, ihn in diesem Dokument niederzuschreiben.

## 5 PFLICHTENHEFT<sup>1</sup>

Dieses Pflichtenheft stellt die Grundvoraussetzung für die Entwicklung des FreeTag-Lasertagsystemes dar. Alle hier nicht aufgeführten Funktionen des Systems sind als nicht geforderte, jedoch, wenn sie dem Spielspaß des Benutzers dienlich sind, als gewünscht anzusehen.

### 5.1 Ziel der Entwicklung

Das FreeTag-Lasertag System hat die Aufgabe, als Open-Source System für den Lasertagsport zu dienen.

#### 5.1.1 Musskriterien

Die Software des Markierers ist frei zugänglich und kann von jedem, der über die nötigen Programmierkenntnisse verfügt, verändert werden.

Das Übertragungsprotokoll, welches für die Infrarotübertragung verwendet wird, entspricht dem MilesTag 2-Protokoll.

Die Hardware, also die mechanischen und elektrischen Teile des Systems, sind möglichst kostengünstig und einfach. Die Besorgung der Komponenten ist mit möglichst geringem Aufwand zu bewerkstelligen.

Das Markierergehäuse wird mit einem 3D-Drucker hergestellt

Ein System besteht aus einem Markierer und externen IR-Sensoren.

Für das Spiel notwendige Parameter werden auf einem, in dem Markierer integrierten, LCD-Bildschirm dargestellt.

Die von anderen Spielern empfangenen Daten werden abgespeichert.

Alle Dokumente, Dateien und Programmcodes sind über das Internet frei zugänglich.

Die Reichweite des Markierers ist an die Gegebenheiten in Gebäuden und in der Natur anpassbar

---

<sup>1</sup> In Anlehnung an: [http://www.infrasoft.at/downloads/Anleitung\\_zum\\_Pflichtenheft.pdf](http://www.infrasoft.at/downloads/Anleitung_zum_Pflichtenheft.pdf) entnommen am 19.02.2018

## FREETAG LASERTAGSYSTEM

---

Beim Eingehen eines Treffers macht eine LED dies optisch sichtbar.

Die Verarbeitung von Schadenswerten von Teammitgliedern ist über die Software zu- und wegschaltbar.

### 5.1.2 Wunschkriterien

Aktionen im Spiel sollen mit Geräuschen (Sounds) unterlegt werden. Diese Sounds sollen von jedem geändert werden können. Die Lautstärke der Tonwiedergabe soll veränderbar sein.

Die Stromversorgung soll über handelsübliche Batterien oder Akkumulatoren erfolgen. Der Markierer soll über einen Schalter von der Stromversorgung getrennt werden können.

Beim Versenden eines Infrarotsignales muss eine zusätzliche LED dies optisch sichtbar machen.

Die Verwendung von imaginären Patronen und Magazinen soll implementiert werden.

Programmierung einer Auswertesoftware am PC.

Die Hardware soll Möglichkeiten zur Weiterentwicklung beinhalten

### 5.1.3 Abgrenzungskriterien

Das System darf nicht in sich geschlossen sein um die Möglichkeit der Weiterentwicklung zu bieten.

Die Software darf keine militärische Simulation zulassen.

Das System darf nicht als Spielzeug für Kinder konzipiert sein.

## 5.2 Produkteinsatz

### 5.2.1 Anwendungsbereich

Das FreeTag-Lasertagsystem ist als System für den Fun- bzw. Actionsport konzipiert und richtet sich an alle Personen die das 18. Lebensjahr vollendet haben. Es lehnt sich dabei an bereits vorhandene Lasertagsysteme an, wobei jedwede Dokumentation frei zugänglich ist. Die Spieleparameter sind relativ frei gestaltet.

### 5.2.2 Zielgruppe

Die Zielgruppe sind Personen, die bereits Kontakt mit dem Lasertagsport hatten oder aus ähnlichen Bereichen kommen, wie z.B. Airsoft, Paintball, FPS-Shooter. Speziell sollen sich auch Personen angesprochen fühlen, welche über die nötigen Fähigkeiten verfügen, um sich selbst eine Anlage nachzubauen.

### 5.2.3 Betriebsbedingungen

Dieses System soll sich in der Funktion nicht wesentlich von anderen Lasertagsystemen unterscheiden, das System soll aber von jedem verändert werden können.

Wartungsarm

Nicht für den Einsatz bei Regen und Nässe geeignet

## 5.3 Produktumgebung

Der Betrieb der Anlage setzt einen PC voraus, auf welchem es möglich ist die neueste Arduino IDE zu betreiben.

### 5.3.1 Software

Die Arduino IDE samt der mitgelieferten Programmierlibraries müssen vorhanden sein. Zusätzlich muss die WTV020SD16P-Library installiert werden. Der Programmcode, im Falle der Arduino IDE als *Sketch* bezeichnet, des FreeTag Lasertagsystems muss vorhanden sein.

### 5.3.2 Hardware

Als Hardware sind ein Markierergehäuse, welches die Elektronik samt Optik beinhaltet, und ein externer Sensor vorgesehen. Die Elektronik besteht aus:

Arduino Mega 2560;

einer Platine samt Bauteilen, um die Musskriterien zu erfüllen. Dies beinhaltet auch die Bauteile zur Audiomeldung;

einer Infrarot-LED und einer LED zur Simulation des Mündungsfeuers;

mehreren Infrarotempfängern;

einem Lautsprecher;

## FREETAG LASERTAGSYSTEM

---

einem 16 x 2 LCD-Display zur Darstellung der Spielwerte;  
drei Tastern.

Die Optik besteht im Wesentlichen aus einer optischen Linse samt zugehörigem Montagematerial.

### 5.4 Produktdaten

*Tabelle 1: Produktdaten*

|                        |   |
|------------------------|---|
| Anlagentyp:            | FreeTag Lasertagsystem Markierer + externe Sensoren |
| Baujahr                | 2018  |
| Abmessungen            | 560x245x80 mm                                       |
| Eigengewicht           | 2,4 kg  |
| Stromversorgung        | 7,2V Gleichstrom durch 6 Stück AA-Akkumulatoren     |
| Anschlussmöglichkeiten | 1x USB-B, 1x RJ10(4p4c) Buchse, 1xRJ12(6P6C) Buchse |

### 5.5 Produktleistungen

*Tabelle 2: Produktleistungen*

|                                   |         |
|-----------------------------------|---------|
| Reichweite des IR-Signales        | > 60 m  |
| Anzahl der Spieler                | Max. 64 |
| Anzahl der Teams                  | 4       |
| Anzahl der mögliche Schadenswerte | 16      |

Folgende Aktionen im Spiel sind mit Audiofiles unterlegt:

Abgeben eines Schusses bei eingestellten Spieleparameter AmmoUse von 1,2 und 3. Jeder dieser Werte besitzt ein eigenes Soundfile;

Eingehen eines Treffers;

Unterschreitung von einem Lebenspunkt;

Widereinstieg ins Spiel;

## FREETAG LASERTAGSYSTEM

---

Nachladen des virtuellen Magazins;

Entleerung des virtuellen Magazins.

Einstellbare Spielparameter:

**PlayerID:** Spielernummer

**Team:** Teamnummer

**Damage:** Schadenswert

**Health:** Lebenspunkte

**Life:** Anzahl der Leben

**Ammo:** die Anzahl an virtuellen Patronen im virtuellen Magazin

**Clips:** die Anzahl an virtuellen Magazinen, die man mitführt

**ammoUse:** Anzahl der verbrauchten Patronen pro Schuss

**respawnTime:** Zeit in Minuten, welche verstreicht bis man wieder am Spiel teilnehmen kann

**fullAuto:** true = vollautomatisches feuern ein; false = vollautomatisches feuern aus

**respawn:** true = unendliche Leben; false = unendliche Leben aus

**friendlyFire:** true = Teambeschuss ein; false = Teambeschuss aus

**clipUse:** true = beim Nachladen werden Magazine verbraucht; false = keine Magazine werden verbraucht

**shortRange:** true = kurze Reichweite der IR-LED aktiviert; false = lange Reichweite aktiv

**indoorMode:** true = Indoor Reichweite der IR-LED aktiviert; false = Outdoor Reichweite aktiv

**sound:** true = Audiowidergabe ein; false = Audiowidergabe aus

## 5.6 Qualitätszielbestimmung

Tabelle 3: Qualitätszielbestimmung

|                          | sehr wichtig | wichtig | weniger wichtig | unwichtig |
|--------------------------|--------------|---------|-----------------|-----------|
| Robustheit               |              | X       |                 |           |
| Einfachheit              |              | X       |                 |           |
| Zuverlässigkeit          |              | X       |                 |           |
| Benutzungsfreundlichkeit |              | X       |                 |           |
| Veränderbarkeit          | X            |         |                 |           |
| Reproduzierbarkeit       | X            |         |                 |           |
| Spielspaß                | X            |         |                 |           |

## 5.7 Entwicklungsumgebung

Wenn möglich werden nur Programme verwendet, welche als Freeware erhältlich sind.

### 5.7.1 Software

Arduino IDE, samt der WTV020SD16P-Library

SOMO-14D zur Umwandlung der Audio Dateien

KiCad für die PCB Layouts und den Schaltplan

MECHANISCHES TOOL einbauen

### 5.7.2 Hardware

Arduino MEGA 2560

## 6 BETRIEB DER FREETAG-LASERTAGANLAGE

### 6.1 Mindestvoraussetzung zum Betrieb des Systems

Um das FreeTag-Lasertagsystem sinnvoll nutzen zu können, müssen mindestens folgende Punkte erfüllt sein.

Man benötigt mindestens zwei Spieler samt Markierer, welche gegeneinander antreten, und ausreichend Platz (entweder Indoor oder Outdoor). Der Platz sollte, wenn möglich, abgesperrt sein.

Eine erfolgreiche Erstinbetriebnahme jedes Markierers musste zuvor durchgeführt werden -> Siehe dazu Punkt 6.2 Erstinbetriebnahme.

Es muss ein PC zur Verfügung stehen, auf welchem die aktuelle Arduino Programmierumgebung installiert ist. Es muss außerdem die folgende externe Library installiert sein: Wtv020SD16p.

Die gewünschten Sounddateien müssen richtig benannt auf die Micro SD-Karte kopiert werden.

Die Spieler müssen sich auf die Spieleparameter einigen und dementsprechend die dazugehörigen Parameter in der Software ändern und diese auf die Markierer laden.

Die genaue Vorgehensweise dazu wird später erläutert.

Die Stromversorgung jedes Phasers muss gegeben sein

Wenn die Daten, welche am LCD-Display des Markierers dargestellt werden, mit denen übereinstimmen, welche über die Arduinosoftware eingespielt wurden, so kann das Spiel beginnen.

Um das Spiel fair zu gestalten, sollte zuvor bestimmt werden ob die Phaser, die im Spiel verwendet werden, funktionieren und über annähernd dieselbe Signalübertragungsreichweite verfügen.

Die Kopfbandsensoren sollten in die vorgesehenen RJ Buchse gesteckt werden

### 6.2 Erstinbetriebnahme des Arduino Mega und dem Lasertagsystem

Zur erstmaligen Inbetriebnahme des Arduinoboards und dem FreeTag-Lasertagsystem folgen Sie bitten den Punkten 6.2.1 bis 6.2.10, diese dienen dazu die Kommunikation zwischen Arduino und dem Rechner herzustellen und eine Funktionskontrolle des Arduino durchzuführen.

### 6.2.1 Installieren der Arduino Programmiersoftware

In den folgenden Schritten werde ich auf die Installation der Programmiersoftware auf Windows-Rechnern eingehen. Für andere Systeme lesen Sie bitte die Anleitung unter <https://www.arduino.cc/en/Guide/HomePage> (in englischer Sprache)

Zuerst müssen Sie die Arduino-Programmiersoftware von  
<https://www.arduino.cc/en/Main/Software>

herunterladen. Bitte wählen Sie dazu das für Sie passende Betriebssystem aus, z.B. für Windows auf „Windows Installer“ klicken. Nach dem Herunterladen der .exe-Datei führen Sie diese bitte aus und folgen Sie der Installationsanweisung. Man sollte sich den Dateipfad merken, unter dem die Programmiersoftware auf der Festplatte installiert wird (standardmäßig unter: C:\Programme(x86)\Arduino), da später noch eine externe Library in einen Unterordner des Installationsordners entpackt werden muss. Windows wird während des Installationsvorganges mehrfach fragen: „Möchten Sie diese Gerätesoftware installieren?“ Dies bitte mit dem Button *Installieren* zulassen (siehe Abbildung 1). Nach Beendigung der Installation sollte sich eine Verknüpfung mit dem Namen „Arduino“ auf dem Desktop befinden. Mit einem Doppelklick auf dieses Symbol öffnen Sie die Programmiersoftware.



Abbildung 1: Aufforderung zur Installation der Arduino Gerätesoftware

### 6.2.2 Einrichten des Arduino Mega

Damit die Programmiersoftware mit Arduino Mega 2560 umgehen kann, muss diese eingerichtet werden. Hierzu unter dem Programmreitern Tools -> Board Arduino Mega APK wählen (siehe Abbildung 2).

## FREETAG LASERTAGSYSTEM

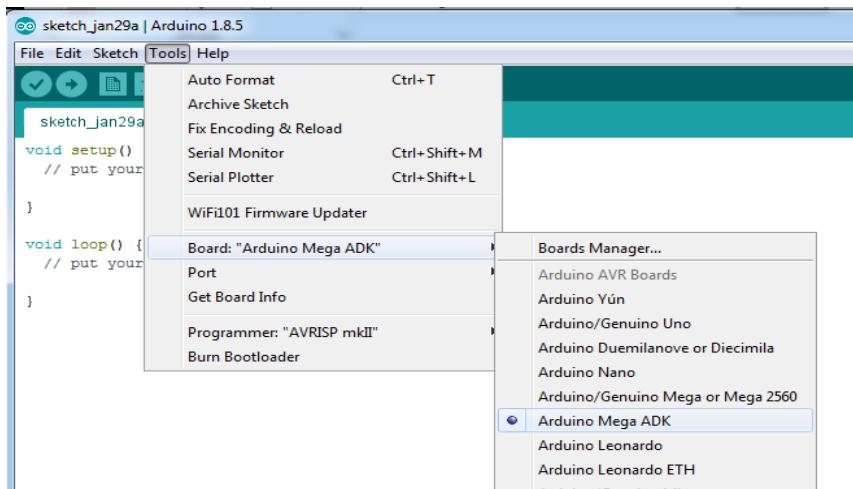


Abbildung 2: Auswahl Arduino Mega APK in der Programmierumgebung

Nun muss der Programmierumgebung noch mitgeteilt werden an welchen COM-Port sich das Arduinoboard befindet. Schließen Sie dazu bitte Arduino mit einem passenden USB-Kabel an Ihrem Rechner an. Zuerst wird Windows die Treiber für Arduino Mega installieren. Nachdem dies vollendet ist, muss man in der Programmierumgebung unter dem Reiter Tools -> Port den COM-Port wählen, neben welchem der Name des Arduinoboardes steht (z.B. COM-Port 10, siehe Abbildung 3).

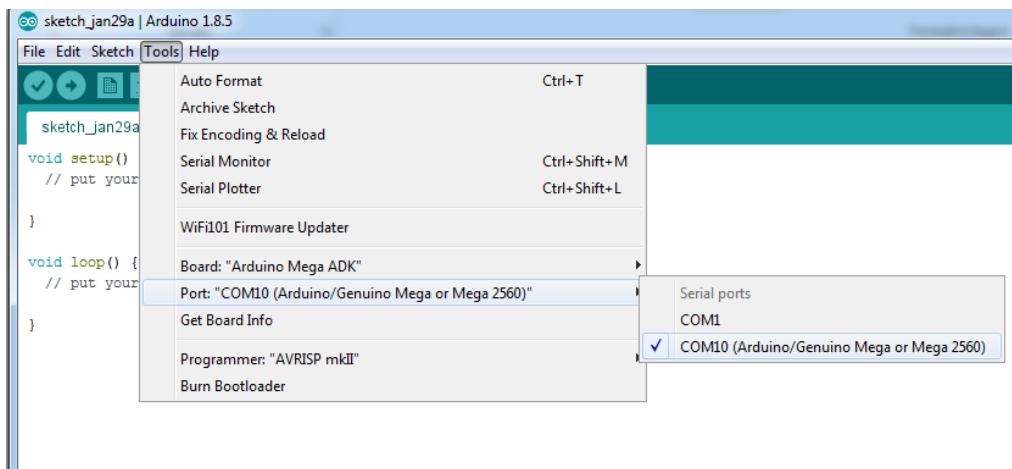


Abbildung 3: Portauswahl in der Programmierumgebung

Der Arduino und die Programmierumgebung sind nun bereit zum Datenaustausch.

### 6.2.3 Hochladen des ersten Programmes und Funktionstest des Arduino Mega

Um sicherzustellen, dass Arduino Mega funktioniert, sollten Sie ein kleines Programm, welches bereits in der Arudoinosoftware inkludiert ist, auf das Gerät laden. Dazu wählen Sie bitte unter dem Reiter File -> 01.Basics -> Blink aus (Abbildung 4).

## FREETAG LASERTAGSYSTEM

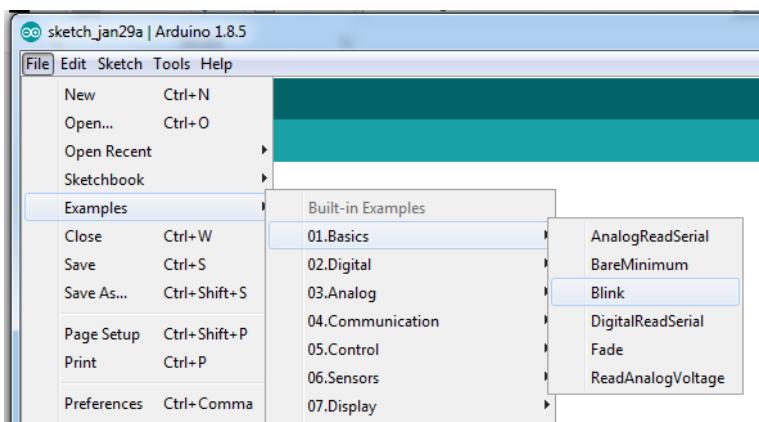


Abbildung 4: Auswahl Blink Sketch

Es öffnet sich danach ein neues Fenster, welches den Programmcode für den Blink-Sketch beinhaltet. Kontrollieren Sie in diesem Sketch noch einmal die Port- und die Boardwahl (siehe Punkt 6.2.2), danach können Sie auf den Uploadbutton links oben klicken (siehe Abbildung 5).

```
/* Blink | Arduino 1.8.5
File Edit Sketch Tools Help
Upload
Blink
/*
 * Blink
 *
 * Turns an LED on for one second, then off for one second, repeatedly.
 *
 * Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
 * it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
 * the correct LED pin independent of which board is used.
 * If you want to know what pin the on-board LED is connected to on your Arduino
 * model, check the Technical Specs of your board at:
 * https://www.arduino.cc/en/Main/Products
 *
 * modified 8 May 2014
 * by Scott Fitzgerald
 * modified 2 Sep 2016
 * by Arturo Guadalupi
 * modified 8 Sep 2016
 * by Colby Newman
 *
 * This example code is in the public domain.
 * http://www.arduino.cc/en/Tutorial/Blink
 */
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);    // turn the LED on (HIGH is the voltage level)
  delay(1000);                      // wait for a second
  digitalWrite(LED_BUILTIN, LOW);     // turn the LED off by making the voltage LOW
  delay(1000);                      // wait for a second
}

Done uploading.
Sketch uses 1462 bytes (0%) of program storage space. Maximum is 253952 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 8183 bytes for local variables. Maximum is 8192 bytes.
```

Abbildung 5: Upload des Blink Programmes

Wenn der Uploadvorgang erfolgreich war, so wird unten in der türkisen Zeile „Done Uploading“ angezeigt. Funktioniert die Arduino-Plattform ordnungsgemäß, so sollte die Onboard-LED nun in einem Takt von einer Sekunde blinken (siehe Abbildung 6).

## FREETAG LASERTAGSYSTEM

---

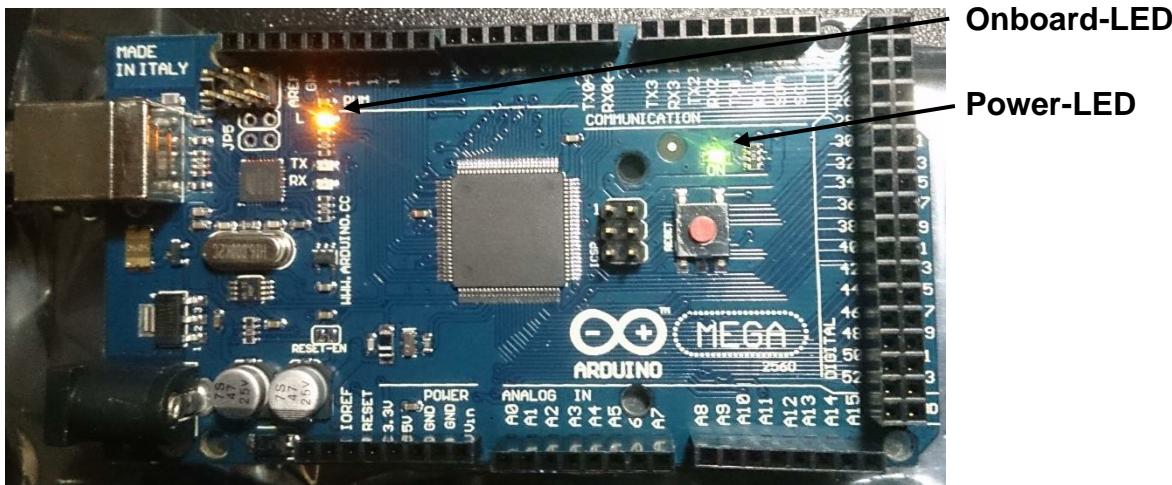


Abbildung 6: Arduino Mega mit leuchtendem Onboard-Led

### 6.2.4 Installieren von externen Libraries in die Arduino-Programmiersoftware

Die externe Programmierlibrary für das Wtv020sd16p-Soundmodul muss unter folgendem Link heruntergeladen werden:

<https://docs.google.com/file/d/0B4p82-pNB6o7V2owUnIXdIBNSkU/edit>.

Zum Installieren und Verwenden der Library muss zuerst die Programmierumgebung geschlossen werden und dann der Ordner mit dem Namen Wtv020sd16p in den Librariesordner des Arduino Installationsordners entpackt werden. Der Arduino Librariesordner ist direkt im Arduino Installationsordner zu finden (z.B. unter C:\Arduino\arduino-1.0.5-r2\libraries). Nach dem Entpacken der .zip-Datei in den richtigen Ordner steht er zur Verfügung. Das untere Bild zeigt die Ordnerstruktur der unterschiedlichen Libraries, die der Programmierumgebung zur Verfügung stehen, wenn die Wtv020sd16p.zip-Datei richtig entpackt wurde.

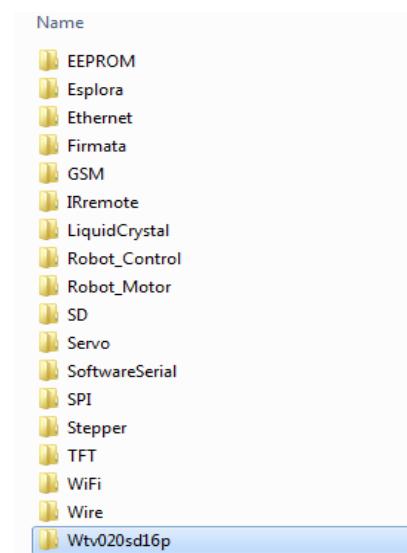


Abbildung 7: Beispiel Ordnerstruktur der Arduino Libraries

## FREETAG LASERTAGSYSTEM

---

Es sollten sich folgende Dateien in dem Wtv020sd16p-Ordner befinden:

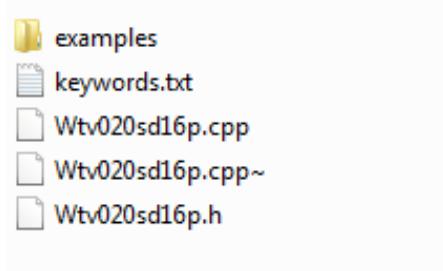


Abbildung 8: Dateien Wtv020sd16p-Ordner

### 6.2.5 Installieren der FreeTag-Lasertagsoftware

Zuerst muss die Lasertagsoftware aus dem Internet unter folgendem Link, im Ordner „Arduino Software“ heruntergeladen werden:

[https://github.com/EmmanuelPanholzer/FreeTag-Lasertag.](https://github.com/EmmanuelPanholzer/FreeTag-Lasertag)

Die Datei sollte dann in den Unterordner mit dem Namen „Sketches“ des Arduino Installationsordners entpackt werden. Wenn dies geschehen ist, kann der Programmcode, also die Software für den Markierer, entweder durch doppeltes Anklicken der Datei „FreeTag Lasertag.ino“ oder durch Verwenden der Funktion Datei -> Öffnen und Auswahl der Datei in der Arduinosoftware geöffnet werden.

### 6.2.6 Einstellen des Zählwertfehlers divisorstart, divisorzero und divisorone

Um diesen Schritt genauer zu verstehen, müsste ich zuerst erklären wie Arduino Signale wie den Header, eine binäre „1“ oder „0“ überträgt. Dazu bitte den Punkt „Zyluskorrektur divisorstart, divisorzero, divisorone“ im Kapitel Technische Grundlagen lesen.

In der Software sind jeweils eigene Korrekturwerte für jeden Übertragungswert, also Header, eins und null vorgesehen um diese Abweichung zu korrigieren, genannt *divisorstart* für den Header, *divisorzero* für die binäre „0“ und *divisorone* für die binäre „1“.

Leider arbeitet jede Arduino-Mega-Ausführung mit leicht abweichender Geschwindigkeit, weswegen diese Korrekturwerte bei jedem Markierer bzw. jedem Aduino Mega separat eingestellt werden müssen. Um die Korrekturwerte genau einzustellen benötigt man einen zweiten Markierer. Sollte dieser nicht vorhanden sein, so können vorerst die Werte *divisorstar*, *divisorzero* und *divisorone* auf 38 eingestellt werden.

## FREETAG LASERTAGSYSTEM

---

Um die Korrekturwerte genau einzustellen, muss zuerst der zweite Markierer, welcher als Empfänger dient, an einen PC angeschlossen werden. Es muss danach die Arduino Programmierumgebung gestartet werden und der COM-Port sowie der richtige Arduino-Typ eingestellt werden (siehe dazu Punkt 6.2.2).

Nachdem der Port und der Arduinotyp gewählt wurden, muss der Serielle Monitor unter Tools -> Serieller Monitor gestartet werden (siehe Abbildung 9).

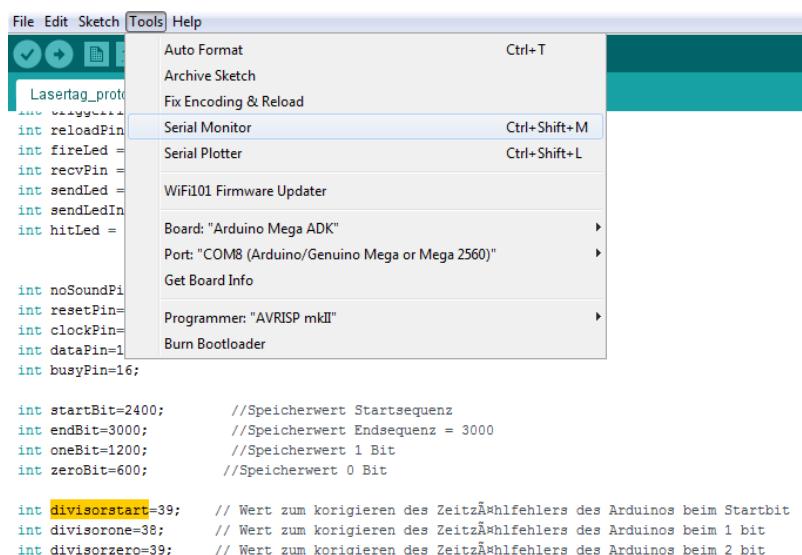


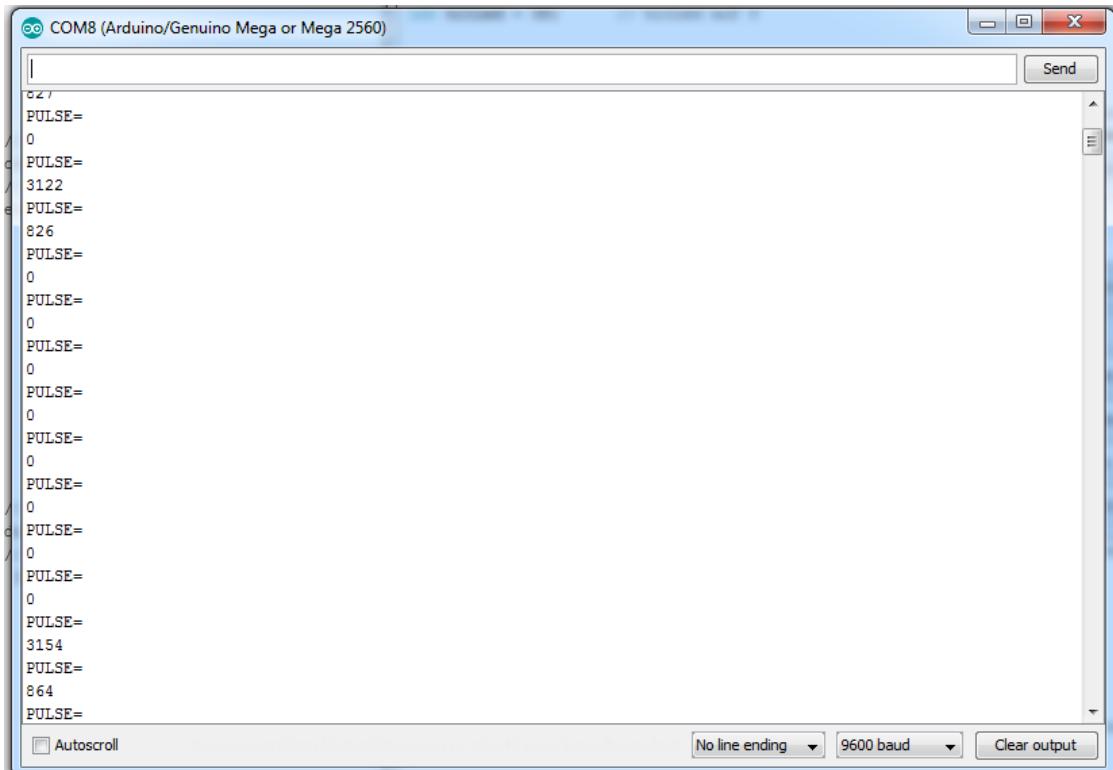
Abbildung 9: Öffnen des Seriellen Monitors

Es wird ein neues Fenster geöffnet, welches sogleich diverse Werte darstellt, welche jedoch anfangs ignoriert werden können.

Wenn man jetzt einmal auf den Empfänger feuert, so wird unter der Bezeichnung ein Zahlenwert ausgegeben. Da jedoch die Übertragung von Daten vom Arduino weiterläuft, scrollt der Serielle Monitor weiter, daher das Häkchen links unten neben „Autoscroll“ entfernen und mit dem Balken zurück zu der Zeile fahren wo unter Pulse ein Zahlenwert über null steht (siehe Abbildung 10).

Dieser Wert unter Pulse gibt an, wie lange der Impuls gedauert hat, welcher empfangen wurde. Dieser weicht leider von dem Sollwert des Übertragungsprotokolls ab.

## FREETAG LASERTAGSYSTEM



*Abbildung 10: Pulslänge vor Korrektur*

Wie hier zu sehen ist, sind mehrere Werte eingefangen worden. Die Werte um die 3100 sollten eigentlich 2400(Header) sein. Die Werte um die 800 sollten eigentlich 600 (binär „0“) sein.

Um die Korrekturwerte zu ermitteln, benutzen Sie nun folgende Formel:

$$\left( \frac{26 * \text{Sollwert Pulse}}{2 * \text{Sollwert Pulse} - \text{Gemessener Pulswert}} \right) = \text{Korrekturwert}$$

Für den obigen Fall, dass der gemessene Pulswert 3122 und der Sollwert 2400 beträgt, sieht die Rechnung wie folgt aus:

$$\left( \frac{26 * 2400}{2 * 2400 - 3122} \right) = 37,19 \approx 37$$

Der Wert 37 muss nun in der FreeTag Software als *divisorstar* eingetragen werden und die Software auf den einzustellenden Markierer hochgeladen werden. Notieren Sie sich diesen Wert oder Speichern Sie ihn in der FreeTag Software ab.

**Vorsicht:** Sobald der Korrekturwert in der FreeTag Software geändert wurde, funktioniert die obige Formel nicht mehr!

## FREETAG LASERTAGSYSTEM

---

```
prototyp | Arduino 1.8.5
File Edit Sketch Tools Help
prototyp §
boolean flagFire; // Speicherwert ob man während man die Flagge hat, Schießen kann

//Hardwareinfos
int triggerPin = 9; //Abzugschalter ist auf Pin 9
int reloadPin = 8; //Nachladeknopf ist auf Pin 8
int fireLed = 34; //Mündungsled ist auf Pin 6
int recvPin = 6; //Empfängerled ist auf Pin 13
int sendLed = 42; //IR-Led(lange Reichweite) ist auf Pin 7
int sendLedIndoor = 44; //IR-Led(kurze Reichweite) ist auf Pin A0(14)
int hitLed = 38; //hitLed auf 8

int noSoundPin=14;
int resetPin=16;
int clockPin=17;
int dataPin=18;
int busyPin=15;

int startBit=2400; //Speicherwert Startsequenz
int endBit=3000; //Speicherwert Endsequenz = 3000
int oneBit=1200; //Speicherwert 1 Bit
int zeroBit=600; //Speicherwert 0 Bit

int divisorstart=37; //Wert zum korrigieren des Zeitzählfehlers des Arduinos beim Startbit
int divisorone=26; //Wert zum korrigieren des Zeitzählfehlers des Arduinos beim 1 bit
int divisorzero=26; //Wert zum korrigieren des Zeitzählfehlers des Arduinos beim 2 bit

Done uploading.
Sketch uses 22548 bytes (8%) of program storage space. Maximum is 253952 bytes.
Global variables use 2177 bytes (26%) of dynamic memory, leaving 6015 bytes for local variables. Maximum is 8192 bytes.

57 Arduino Mega ADK on COM8
```

Abbildung 11: Divisor Start eingefügt

Um den Korrekturwert *divisorstart* zu ändern, scrollen Sie bitte bis zu dem Punkt (siehe Abbildung 11) wo Sie die Zeile „*int divisorstart*“ finden und ersetzen Sie den Wert 26 mit dem von Ihnen errechneten Wert (z.B. 37 wie im vorherigen Beispiel).

Damit es zu keinen Problemen während des Hochladens des Programmes kommt, schließen Sie bitte den Markierer, der als Empfänger dient, vom PC ab und schließen Sie stattdessen den einzustellenden an. Der COM-Port muss wieder neu ausgewählt werden.

Laden Sie nun das Programm, durch Drücken des Upload-Buttons, auf den Markierer. Wenn dies erledigt ist, schließen Sie den Markierer ab und schließen Sie den Empfänger wieder an. Ändern Sie den COM-Port auf den des aktuellen Markierers und öffnen Sie erneut den Seriellen Monitor.

Geben Sie nun wieder einen Schuss auf den Empfänger ab, und scrollen Sie zu den Zeilen, wo sich ein Zahlenwert unter der Zeile Pulse befindet (siehe Abbildung 12).

## FREETAG LASERTAGSYSTEM

---

```
PULSE=
0
PULSE=
0
PULSE=
0
PULSE=
2203
834
834
841
862
830
835
834
1594
834
834
837
1603
860
1612
Umrechnung in Binaer erfolgt
0
0
0
0
```

Abbildung 12: Pulslängen nach Korrektur Header

Die Werte, welche Sie im Seriellen Monitor zu sehen bekommen, sollten denen in Abbildung 12 ähneln. Wie Sie sehen ist der erste Wert nach „PULSE=“ 2203, dies ist der bereits korrigierte Header. Ich möchte anmerken, dass die gemessenen Impulszeiten bei jeder Messung leicht variieren und es fast unmöglich ist, die genauen Zeiten nach Protokoll einzuhalten. Man kann sich jedoch durch Versuche und erhöhen bzw. verringern des Korrekturwertes an den Idealwert herantasten. Ich würde z. B. den Korrekturwert divisorstart von 37 auf 34 verringern, damit die Arduino-Plattform noch 3 Zyklen mehr durchführt.

Werte um die 800 stellen das Binärsignal „0“ dar und sollten eigentlich 600 sein. Verwenden Sie die zuvor verwendete Formel, um den Korrekturwert „divisorzero“ zu errechnen. In meinem Falle ergibt sich der Wert 42 als Korrekturwert (siehe Rechnung):

$$\text{divisorzero} = \left( \frac{26 * 600}{2 * 600 - 830} \right) = 42,16 \approx 42$$

Werte um die 1500 stellen das Binärsignal „1“ dar und sollten eigentlich 1200 sein. Verwenden Sie die zuvor verwendete Formel um den Korrekturwert „divisorone“ zu errechnen. In meinem Fall ergibt sich der Wert 39 als Korrekturwert (siehe Rechnung):

$$\text{divisorone} = \left( \frac{26 * 1200}{2 * 1200 - 1603} \right) = 39,15 \approx 39$$

## FREETAG LASERTAGSYSTEM

---

Notieren Sie sich die errechneten Werte, geben Sie diese in der Software ein und laden Sie das Programm auf den einzustellenden Markierer.

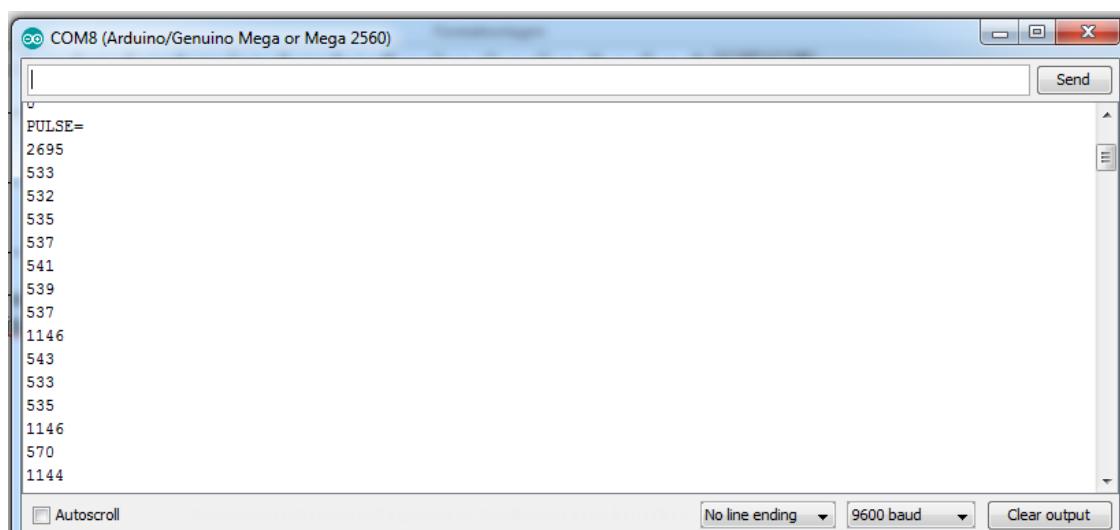
Nach dem Hochladen schließen Sie den Empfänger erneut an und öffnen Sie den Seriellen Monitor. Geben Sie einen Schuss ab und kontrollieren Sie die Zahlenwerte nach der Zeile „PULSE=“, diese sollten jetzt annähernd die Werte des Datenprotokolls betragen, also 1200 für den Header, 600 für eine binäre „0“ und 1200 für eine binäre „1“. Sollten die Werte zu stark abweichen, so verändern Sie, wie zuvor beschrieben, die Korrekturwerte bis sie annähernd an die Sollwerte gekommen sind.

Da die Werte nie genau dem Protokoll entsprechen, ist bei der Auswertung des Signals eine gewisse Toleranz zu berücksichtigen. Diese ist in der folgenden Tabelle ersichtlich:

Tabelle 4: Signaltoleranzen

| Signalname | Sollwert lt. Protokoll | Toleranzbereich         |
|------------|------------------------|-------------------------|
| Header     | 2400µ Sekunden         | 2100 bis 2700µ Sekunden |
| Binär „0“  | 600µ Sekunden          | 300 bis 900µ Sekunden   |
| Binär „1“  | 1200µ Sekunden         | 1000 bis 1500µ Sekunden |

Wenn die Korrekturwerte erfolgreich angepasst wurden, sollte die Signalausgabe am Seriellen Monitor in etwa wie in Abbildung 13 aussehen.



```
PULSE=
2695
533
532
535
537
541
539
537
1146
543
533
535
1146
570
1144
```

Abbildung 13: Pulslängen nach Korrektur divisorstart, divisorzero und divisorone

## FREETAG LASERTAGSYSTEM

---

Wie in Abbildung 13 zu sehen ist, weicht der Wert für den Header mit 2695 doch stark ab, der Korrekturwert *divisorstart* sollte somit um den Wert eins oder zwei erhöht werden. Die Werte für die binäre „1“ und „0“ sind bereits sehr nahe am Soll und müssen nicht weiter angepasst werden.

Wenn die Korrekturwerte gut eingestellt wurden, wird der Empfänger unter den Zahlenwerten nach „*Pulse=*“ weiter Zeilen übertragen, beginnend mit der dem Satz: „*Umrechnung in Binaer erfolgt*“, welche dem Binärkode des Senders entsprechen (siehe Abbildung 14). Zuerst wird der Binärkode des ersten Bytes des Spielerpaketes dargestellt, die oberste Stelle entspricht dabei der linken Position des Binärcodes. Nach der Zeile „*Byte 1 fertig*“ werden die nächsten 6 Bit des Spielerpaketes dargestellt, die oberste Stelle entspricht wieder der linken Position des Binärcodes.

Am Ende, also unter der Zeile „*EnemyID*“, wird noch der Dezimalzahlenwert der Spielernummer, der Teamnummer und des Schadenswerts angezeigt. Wenn man nun die Dezimalwerte mit den eingestellten Werten vergleicht, kann festgestellt werden ob die Datenübertragung per Infrarot ohne Probleme erfolgte.

```
10
PULSE=
2543
533
523
528
543
545
545
535
535
1150
540
535
528
1146
529
529
1146
Umrechnung in Binaer erfolgt
0
0
0
0
0
0
0
0
0
0
1
Byte 1 fertig
0
0
0
1
0
1
Byte 2 fertig
EnemyID:
1
1
10
Autoscroll
```

The screenshot shows a terminal window titled "COM8 (Arduino/Genuino Mega or Mega 2560)". The window displays a series of numbers and text indicating the transmission of a game packet. The data starts with "10" and "PULSE=". It then lists several binary values (2543, 533, 523, 528, 543, 545, 545, 535, 535, 1150, 540, 535, 528, 1146, 529, 529, 1146) followed by the text "Umrechnung in Binaer erfolgt". Below this, there are several zeros and a single "1", followed by the text "Byte 1 fertig". This is followed by another set of zeros and a single "1", followed by the text "Byte 2 fertig". Finally, the text "EnemyID:" is displayed, followed by three "1"s and the number "10". At the bottom of the window, there are buttons for "Autoscroll", "No line ending", "9600 baud", and "Clear output".

Abbildung 14: Kontrolle der Übertragung des Spielerpaketes

Die Inbetriebnahme des Markierers ist nun fast beendet, es müssen jetzt nur noch die Spielrelevanten Daten in der FreeTag Software geändert und hochgeladen werden. Wenn während des Spieles auch Sounds vom Markierer wiedergegeben werden sollen, so müssen noch die nötigen Sounddateien auf die Micro-SD Karte geladen werden, dazu jedoch später mehr.

Ich empfehle die Software mit den geänderten Korrekturwerten zusätzlich zur Originalsoftware abzuspeichern, damit diese nicht verloren gehen bzw. man stets seine eigene Software mit den passenden Werten verwenden kann. Außerdem ist es klug, die Korrekturwerte zu notieren, falls das Hochladen vor dem Spiel von einer zentralen Stelle aus passiert, damit diese auch eingetragen werden können.

### 6.2.7 Einstellen der Spieleparameter

Die Signalübertragung des Phasers sollte nun funktionieren, es müssen nun nur noch die Spieleparameter in der FreeTag-Software eingegeben werden und dann kann das Spiel beginnen.

Zuerst müssen in der Software die Namen der Spieler eingegeben werden, dies erfordert zwar einiges an Aufwand, wenn man jedoch mit Bedacht vorgeht, so sollte dieser Schritt nur einmal nötig sein, da es möglich ist 64 Spielernamen einzugeben und, sofern man öfters mit denselben Personen spielt, jeder Spielername seine eigene Spielernummer hat.

Jeder Spieler sollte jetzt seine Spielernummer (1 bis 64) und seinen Spielernamen erhalten, damit die Werte in die Software eingetragen werden können.

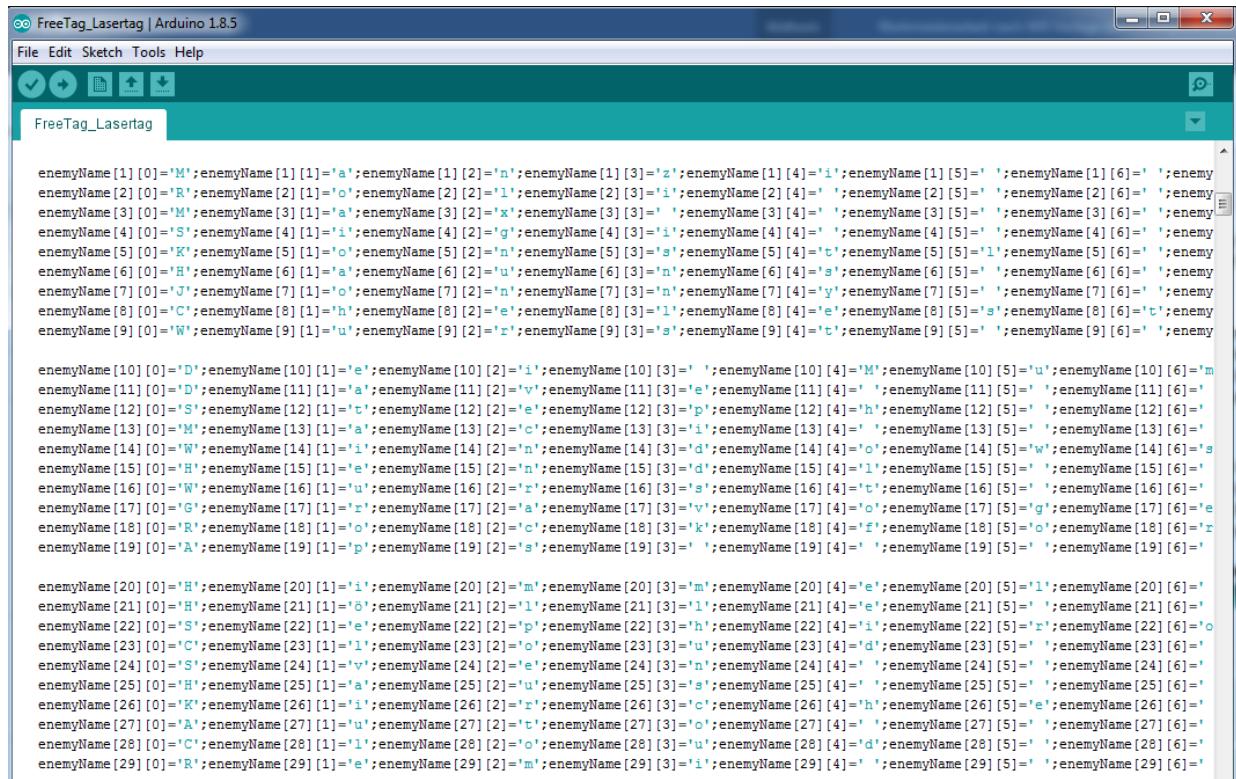
In der Software sind die Spielernamen bereits mit beliebigen Namen ausgefüllt. In der Software werden die Spielernamen als „enemyName“ bezeichnet. Um die Spielernamen zu ändern, muss in der Software soweit hinuntergescrollt werden, bis man auf die Auflistung trifft, die man in Abbildung 15 sieht. Bei der Eingabe der Namen ist auf die Formatierung zu achten und ein Spielername darf maximal elf Zeichen lang sein.

Am besten lässt sich die Eingabe an einem Beispiel zeigen. Wenn Sie zum Beispiel den Spielernamen von Spielernummer 1 ändern möchten, so müssen Sie die Zeichen in der Zeile enemyName[1][x] eingegeben, die [1] neben enemyName steht dabei für die Spielernummer, das [x] steht dabei für die Stelle des Buchstabens im Namen, wobei der erste Buchstabe die Nummer [0] hat. Möchten sie jetzt zum Beispiel den Namen des Spielers 1 in Harald umändern so würde die Programmzeile wie folgt aussehen:

```
enemyName[1][0]='H';enemyName[1][1]='a';enemyName[1][2]='r';
enemyName[1][3]='a';enemyName[1][4]='l';enemyName[1][5]='d';
enemyName[1][6]=' ';enemyName[1][7]=' ';enemyName[1][8]=' ';
enemyName[1][9]='(';enemyName[1][10]='';enemyName[1][11]='\0';
```

## FREETAG LASERTAGSYSTEM

Es ist darauf zu achten das die Buchstaben stets zwischen zwei 'Zeichen stehen und das an Position [11] steht die Zeichenfolge '**'0'** steht, ansonsten kann die Software nicht mit den Daten umgehen.



The screenshot shows the FreeTag\_Lasertag software running on an Arduino IDE version 1.8.5. The window title is "FreeTag\_Lasertag | Arduino 1.8.5". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for back, forward, file operations, and sketch tools. The main area displays a list of enemy names, each consisting of a single character followed by a double quote, then a sequence of characters, another double quote, and finally a character. The list starts with "enemyName [1] [0] = 'M';" and continues through various letters of the alphabet, including some special characters like 'W' and 'G'. The list ends with "enemyName [29] [0] = 'R';".

```
enemyName [1] [0] = 'M'; enemyName [1] [1] = 'a'; enemyName [1] [2] = 'n'; enemyName [1] [3] = 'z'; enemyName [1] [4] = 'i'; enemyName [1] [5] = ' ' ; enemyName [1] [6] = ' ' ; enemyName [2] [0] = 'R'; enemyName [2] [1] = 'o'; enemyName [2] [2] = 'l'; enemyName [2] [3] = 'i'; enemyName [2] [4] = ' ' ; enemyName [2] [5] = ' ' ; enemyName [2] [6] = ' ' ; enemyName [3] [0] = 'M'; enemyName [3] [1] = 'a'; enemyName [3] [2] = 'x'; enemyName [3] [3] = ' ' ; enemyName [3] [4] = ' ' ; enemyName [3] [5] = ' ' ; enemyName [3] [6] = ' ' ; enemyName [4] [0] = 'S'; enemyName [4] [1] = 'i'; enemyName [4] [2] = 'g'; enemyName [4] [3] = 'i'; enemyName [4] [4] = ' ' ; enemyName [4] [5] = ' ' ; enemyName [4] [6] = ' ' ; enemyName [5] [0] = 'K'; enemyName [5] [1] = 'o'; enemyName [5] [2] = 'n'; enemyName [5] [3] = 'a'; enemyName [5] [4] = 't'; enemyName [5] [5] = 'l'; enemyName [5] [6] = ' ' ; enemyName [6] [0] = 'J'; enemyName [6] [1] = 'a'; enemyName [6] [2] = 'u'; enemyName [6] [3] = 'n'; enemyName [6] [4] = 's'; enemyName [6] [5] = ' ' ; enemyName [6] [6] = ' ' ; enemyName [7] [0] = 'J'; enemyName [7] [1] = 'o'; enemyName [7] [2] = 'n'; enemyName [7] [3] = 'n'; enemyName [7] [4] = 'y'; enemyName [7] [5] = ' ' ; enemyName [7] [6] = ' ' ; enemyName [8] [0] = 'C'; enemyName [8] [1] = 'h'; enemyName [8] [2] = 'e'; enemyName [8] [3] = 'l'; enemyName [8] [4] = 'e'; enemyName [8] [5] = 'a'; enemyName [8] [6] = 't'; enemyName [9] [0] = 'W'; enemyName [9] [1] = 'u'; enemyName [9] [2] = 'r'; enemyName [9] [3] = 's'; enemyName [9] [4] = 't'; enemyName [9] [5] = ' ' ; enemyName [9] [6] = ' ' ; enemyName [10] [0] = 'D'; enemyName [10] [1] = 'e'; enemyName [10] [2] = 'i'; enemyName [10] [3] = ' ' ; enemyName [10] [4] = 'M'; enemyName [10] [5] = 'u'; enemyName [10] [6] = 'm'; enemyName [11] [0] = 'D'; enemyName [11] [1] = 'a'; enemyName [11] [2] = 'v'; enemyName [11] [3] = 'e'; enemyName [11] [4] = ' ' ; enemyName [11] [5] = ' ' ; enemyName [11] [6] = ' ' ; enemyName [12] [0] = 'S'; enemyName [12] [1] = 't'; enemyName [12] [2] = 'e'; enemyName [12] [3] = 'd'; enemyName [12] [4] = 'h'; enemyName [12] [5] = ' ' ; enemyName [12] [6] = ' ' ; enemyName [13] [0] = 'M'; enemyName [13] [1] = 'a'; enemyName [13] [2] = 'c'; enemyName [13] [3] = 'i'; enemyName [13] [4] = ' ' ; enemyName [13] [5] = ' ' ; enemyName [13] [6] = ' ' ; enemyName [14] [0] = 'W'; enemyName [14] [1] = 'i'; enemyName [14] [2] = 'n'; enemyName [14] [3] = 'd'; enemyName [14] [4] = 'o'; enemyName [14] [5] = 'w'; enemyName [14] [6] = 's'; enemyName [15] [0] = 'H'; enemyName [15] [1] = 'e'; enemyName [15] [2] = 'n'; enemyName [15] [3] = 'd'; enemyName [15] [4] = 'l'; enemyName [15] [5] = ' ' ; enemyName [15] [6] = ' ' ; enemyName [16] [0] = 'W'; enemyName [16] [1] = 'u'; enemyName [16] [2] = 'x'; enemyName [16] [3] = 's'; enemyName [16] [4] = 't'; enemyName [16] [5] = ' ' ; enemyName [16] [6] = ' ' ; enemyName [17] [0] = 'G'; enemyName [17] [1] = 'r'; enemyName [17] [2] = 'a'; enemyName [17] [3] = 'v'; enemyName [17] [4] = 'o'; enemyName [17] [5] = 'g'; enemyName [17] [6] = 'e'; enemyName [18] [0] = 'R'; enemyName [18] [1] = 'o'; enemyName [18] [2] = 'c'; enemyName [18] [3] = 'k'; enemyName [18] [4] = 'f'; enemyName [18] [5] = 'o'; enemyName [18] [6] = 'r'; enemyName [19] [0] = 'A'; enemyName [19] [1] = 'p'; enemyName [19] [2] = 's'; enemyName [19] [3] = ' ' ; enemyName [19] [4] = ' ' ; enemyName [19] [5] = ' ' ; enemyName [19] [6] = ' ' ; enemyName [20] [0] = 'H'; enemyName [20] [1] = 'i'; enemyName [20] [2] = 'm'; enemyName [20] [3] = 'm'; enemyName [20] [4] = 'e'; enemyName [20] [5] = 'l'; enemyName [20] [6] = ' ' ; enemyName [21] [0] = 'H'; enemyName [21] [1] = 'o'; enemyName [21] [2] = 'l'; enemyName [21] [3] = 'n'; enemyName [21] [4] = 'e'; enemyName [21] [5] = ' ' ; enemyName [21] [6] = ' ' ; enemyName [22] [0] = 'S'; enemyName [22] [1] = 'e'; enemyName [22] [2] = 'p'; enemyName [22] [3] = 'h'; enemyName [22] [4] = 'i'; enemyName [22] [5] = 'r'; enemyName [22] [6] = 'o'; enemyName [23] [0] = 'C'; enemyName [23] [1] = 'l'; enemyName [23] [2] = 'o'; enemyName [23] [3] = 'u'; enemyName [23] [4] = 'd'; enemyName [23] [5] = ' ' ; enemyName [23] [6] = ' ' ; enemyName [24] [0] = 'S'; enemyName [24] [1] = 'v'; enemyName [24] [2] = 'e'; enemyName [24] [3] = 'n'; enemyName [24] [4] = ' ' ; enemyName [24] [5] = ' ' ; enemyName [24] [6] = ' ' ; enemyName [25] [0] = 'H'; enemyName [25] [1] = 'a'; enemyName [25] [2] = 'u'; enemyName [25] [3] = 's'; enemyName [25] [4] = ' ' ; enemyName [25] [5] = ' ' ; enemyName [25] [6] = ' ' ; enemyName [26] [0] = 'K'; enemyName [26] [1] = 'i'; enemyName [26] [2] = 'n'; enemyName [26] [3] = 'c'; enemyName [26] [4] = 'h'; enemyName [26] [5] = 'e'; enemyName [26] [6] = ' ' ; enemyName [27] [0] = 'A'; enemyName [27] [1] = 'u'; enemyName [27] [2] = 't'; enemyName [27] [3] = 'o'; enemyName [27] [4] = ' ' ; enemyName [27] [5] = ' ' ; enemyName [27] [6] = ' ' ; enemyName [28] [0] = 'C'; enemyName [28] [1] = 'l'; enemyName [28] [2] = 'o'; enemyName [28] [3] = 'u'; enemyName [28] [4] = 'd'; enemyName [28] [5] = ' ' ; enemyName [28] [6] = ' ' ; enemyName [29] [0] = 'R'; enemyName [29] [1] = 'e'; enemyName [29] [2] = 'm'; enemyName [29] [3] = 'i'; enemyName [29] [4] = ' ' ; enemyName [29] [5] = ' ' ; enemyName [29] [6] = ' ' ;
```

Abbildung 15: Liste enemyName

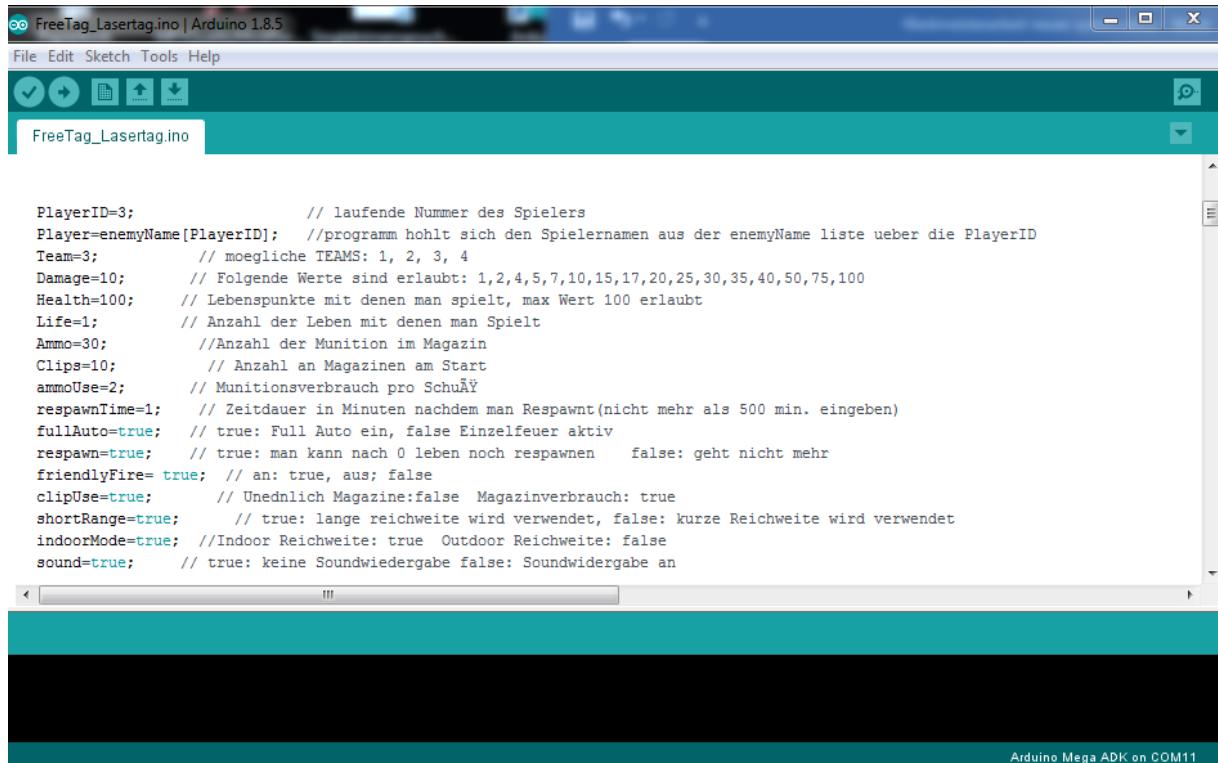
Wenn jeder Spieler seine Nummer hat und sein Spielername in die Software eingetragen wurde, so müssen nun die Tagger einzeln angeschlossen werden und die weiteren spielrelevanten Daten eingestellt werden.

Der Bereich für die Spieleinstellungen ist direkt unter der Liste der Spielernamen zu finden und sieht wie in Abbildung 16 zu sehen ist aus.

Ich werde nun der Reihe nach die einzelnen Werte aufzählen und eine kleine Erklärung dazu abliefern, für genauere Informationen zu den einzelnen Werten und Vorschläge für Spielmodi siehe bitte Punkt 6.3 Spielparameter und Punkt 6.4 Spielmodi. Manche dieser Optionen müssen mit den anderen Spielern abgesprochen werden, um ein gerechtes Spiel zu ermöglichen. Es ist auch zu empfehlen, dass vor jedem Spiel ein Unparteiischer das Hochladen der Software übernimmt damit kein Schummeln möglich ist.

## FREETAG LASERTAGSYSTEM

---



```
PlayerID=3;           // laufende Nummer des Spielers
Player=enemyName[PlayerID]; //programm hohlt sich den Spielernamen aus der enemyName liste ueber die PlayerID
Team=3;               // moegliche TEAMS: 1, 2, 3, 4
Damage=10;             // Folgende Werte sind erlaubt: 1,2,4,5,7,10,15,17,20,25,30,35,40,50,75,100
Health=100;            // Lebenspunkte mit denen man spielt, max Wert 100 erlaubt
Life=1;                // Anzahl der Leben mit denen man Spielt
Ammo=30;               //Anzahl der Munition im Magazin
Clips=10;              // Anzahl an Magazinen am Start
ammoUse=2;             // Munitionsverbrauch pro Schu Y
respawnTime=1;          // Zeitdauer in Minuten nachdem man Respawnnt(nicht mehr als 500 min. eingegeben)
fullAuto=true;          // true: Full Auto ein, false Einzelfeuer aktiv
respawn=true;           // true: man kann nach 0 leben noch respawnnen    false: geht nicht mehr
friendlyFire= true;     // an: true, aus: false
clipUse=true;           // Unednlich Magazine:false  Magazinverbrauch: true
shortRange=true;         // true: lange reichweite wird verwendet, false: kurze Reichweite wird verwendet
indoorMode=true;         //Indoor Reichweite: true  Outdoor Reichweite: false
sound=true;              // true: keine Soundwiedergabe false: Soundwiedergabe an
```

Abbildung 16: Spielparameter

**PlayerID:** Dies ist die Spielernummer, hier wird also die jeweilige Nummer des Spielers eintragen, die höchste Spielernummer ist 64.

**Team:** Hier wird die Teamnummer eingegeben, erlaubte Werte sind 1 bis 4. Spieler mit derselben Teamnummer sind im selben Team.

**Damage:** Hier wird die Schadenssumme, welcher der Markierer an den Empfänger abgibt, eingetragen, mögliche Werte sind 1, 2, 4, 5, 7, 10, 15, 17, 20, 25, 30, 35, 40, 50, 75, 100. Sollte ein anderer Wert gewählt werden, so funktioniert die Übertragung eines Schusses nicht.

**Health:** Dies sind die Lebenspunkte welche man pro Leben besitzt, dieser Wert kann frei zwischen 1 und 100 gewählt werden.

**Life:** Hier wird die Anzahl der Leben eintragen, welche man besitzt. Ein Wert zwischen 0 und 99 ist möglich.

**Ammo:** Die Anzahl an virtuellen Patronen im virtuellen Magazin. Als Wert ist 1 bis 999 möglich.

**Clips:** Die Anzahl an virtuellen Magazinen, die man mitführt, ein Wert zwischen 1 und 99 ist möglich.

**ammoUse:** Dieser Wert gibt an wie viele Patronen beim Feuern verbraucht werden, derzeit einprogrammierte Werte sind 1, 2 und 3. Sollte ein anderer Wert gewählt werden, so funktioniert die Schussabgabe nicht.

**respawnTime:** Hier wird die Zeit in Minuten eintragen, welche verstreicht bis man wieder am Spiel teilnehmen kann, nachdem man 0 Lebenspunkte unterschritten hat.

**fullAuto:** Hier muss gewählt werden ob der Markierer bei betätigtem Abzug durchgehend, bzw. bis das Magazin leer ist, ein Schusssignal abgibt: das heißt Dauerfeuert. Gibt man hier „true“ ein, so ist Dauerfeuer aktiv, gibt man „false“ ein, so muss nach jedem Schuss der Abzug entspannt werden um einen neuen Schuss abzugeben.

**respawn:** Gibt man hier „true“ ein, so kann man selbst nach Verlust aller Leben noch am Spiel teilnehmen, gibt man „false“ ein, so wird der Markierer für das Spiel deaktiviert, wenn man null Leben und null Lebenspunkte hat.

**friendlyFire:** Setzt man den Wert auf „true“, so können Mitglieder des eigenen Teams beschossen werden und erhalten auch den übertragenen Schaden. Bei „false“ sind die Teammitglieder durch Beschuss der eigenen Teammitglieder sicher.

**clipUse:** Bei „true“ werden beim Nachladen Magazine verbraucht, bei „false“ werden keine Magazine verbraucht.

**shortRange:** Hier legt man fest, ob man mit kurzer oder langer Reichweite spielt. Bei „false“ verwendet man die lange Reichweite. Setzt man hier den Wert auf „true“, so spielt man mit der kurzen Reichweite. Ich empfehle die lange Reichweite für Markierer welche nicht vollautomatisch feuern können.

**indoorMode:** Hier legt man fest, ob man die Indoorreichweite oder Outdoor Reichweite verwendet. Bei „false“ verwendet man die Outdoor Reichweite, welche ich für Outdoorspiele empfehle, da es sonst zu Reflektionen von Wänden kommen kann. Setzt man hier den Wert auf „true“, so spielt man mit kürzerer Reichweite welche für Indoorspiele gedacht ist.

**ctf und flagFire:** Diese Punkte sind für den Spielmodus *Capture the Flag* gedacht welcher noch in Entwicklung ist, diese Werte sollten auf „false“ gestellt werden, es ändert jedoch derzeit noch nichts, wenn der Wert auf „true“ gesetzt wird.

**sound:** Möchte man, dass der Markierer Töne bei gewissen Vorgängen, zum Beispiel wenn man getroffen wird, wiedergibt, so ist hier „true“ einzutragen, ansonsten „false“.

### 6.2.8 Übertragen der Soundfiles auf die Micro-SD Karte

Damit das Wtv020sd16p-Soundmodul die Sounddateien während des Spieles abspielen kann, müssen diese zuerst auf die SD-Karte geladen werden. Leider ist der Umgang mit diesem Modul etwas heikel daher hier ein paar Hinweise, um die reibungslose Funktion zu garantieren.

Bevor Sie die Micro-SD Karte entnehmen, drehen Sie mit dem Schlüsselschalter die Spannungsversorgung zum Markierer ab, und heben Sie den Arduino Mikrocontroller von der Platine ab, darunter finden Sie dann das Soundmodul.

Die Micro-SD Karte darf nicht größer als 2GB sein, zudem darf es sich nicht um eine SDHC-Karte handeln.

Die Micro-SD Karte muss in FAT-Format formatiert sein, tätigen Sie dazu im Arbeitsplatz des PCs einen Rechtsklick auf das Laufwerksymbol der Micro-SD Karte, es öffnet sich nun ein Untermenü wie in Abbildung 17 zu sehen ist.

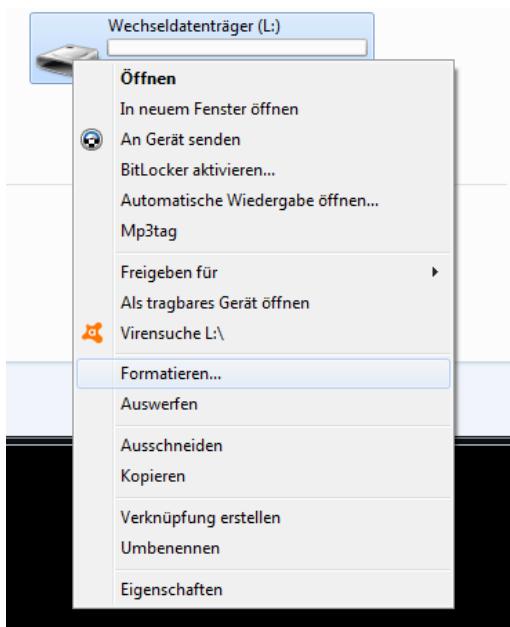


Abbildung 17:Untermenü Laufwerk

In diesem Untermenü wählen Sie bitte die Option „Formatieren“ wie sie in Abbildung 17 blau hinterlegt ist. Danach öffnet sich ein Dialog (siehe Abbildung 18), in welchem Sie die Einstellungen zum Formatieren vornehmen können.

Unter Dateisysteme wählen Sie bitte FAT, und unter Größe der Zuordnungseinheit wählen Sie bitte 32 Kilobytes. Ich empfehle keine Volumebezeichnung zu vergeben.

## FREETAG LASERTAGSYSTEM

---

Drücken Sie nun auf Starten und bestätigen Sie mit OK, dass alle Daten verloren gehen. Windows wird Ihnen danach mitteilen, dass die Formatierung abgeschlossen ist, was Sie bitte mit „OK“ bestätigen.

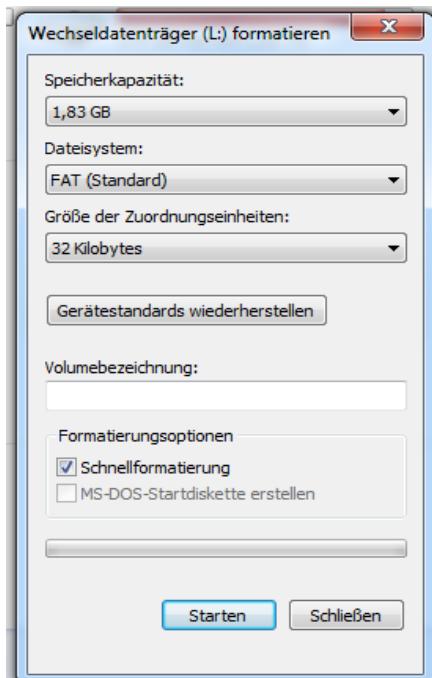


Abbildung 18: Formatierungsmenü

Danach können Sie die Sounddateien auf die SD-Karte laden. Sie können die von mir zur Verfügung gestellten Soundfiles verwenden, wobei Sie sich vorher entscheiden sollten, welchen Typ sie wählen. Unter <https://github.com/EmmanuelPanholzer/FreeTag-Lasertag> finden Sie einen Ordner mit dem Namen „sound“, welcher zwei Ordner mit der Bezeichnung Scifi und Realismus beinhaltet, in denen die nötigen Dateien bereits mit der richtigen Benennung und im richtigen Dateiformat liegen. Im Realismus-Ordner finden Sie Sounddateien, welche echten Waffen nachempfunden sind, im Scifi Ordner sind die Sounddateien Lasergewehren aus Filmen nachempfunden. Um Probleme mit Anrainern und Behörden zu vermeiden empfehle ich die Scifi Sounds zu verwenden.

Wenn Sie eigene Sounds verwenden möchten, so lesen Sie bitte den Punkt „Verwendung eigener Sounds“ im Kapitel Technische Grundlagen.

Mit dem Drehpotentiometer neben dem Verstärkerchip können Sie noch die Lautstärke der Soundwiedergabe einstellen. Um die Lautstärke zu erhöhen, einfach mit einem kleinen Schraubendreher das Potentiometer nach rechts drehen, bevor Sie jedoch das Gehäuse öffnen, bitte die Spannungsversorgung unterbrechen.

### 6.2.9 Einlegen der SD-Karte

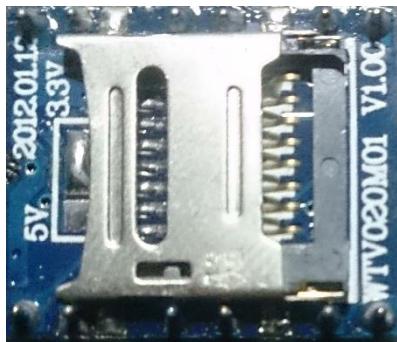


Abbildung 19: WTV020SD16P Soundchip

Zum richtigen Einlegen der SD-Karte muss zuerst die silberne Halterungskappe in Pfeilrichtung nach rechts geschoben werden. Durch das Schieben wird die Verriegelung gelöst und die Abdeckung kann nach rechts weggeklappt werden. Jetzt kann die SD-Karte eingelegt werden (siehe Abbildung 20).



Abbildung 20: Einlegen der SD-Karte

Nachdem die SD-Karte eingelegt wurde, muss die Kappe nach links zugeklappt werden und nach links geschoben werden, damit die Kappe wieder verriegelt wird. Der Soundchip ist nun zur Audiomeldung bereit.

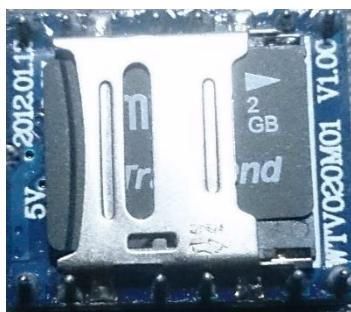


Abbildung 21: Soundchip nach einlegen der SD-Karte

### 6.2.10 Kontrolle der Erstinbetriebnahme

Bauen Sie den Phaser zusammen, legen Sie die Batterien in das Batteriefach und drehen Sie den Schlüsselschalter auf „Ein“. Überprüfen Sie, dass die Daten, welche Sie in der Software eingegeben haben, mit denen am LCD-Bildschirm übereinstimmen.

Auf der Abbildung unten ist als Beispiel der Spieler mit dem Namen „Manzi“, Team „1“, einem Leben, 999 Schuss pro Magazin und 10 Ersatzmagazine dargestellt.

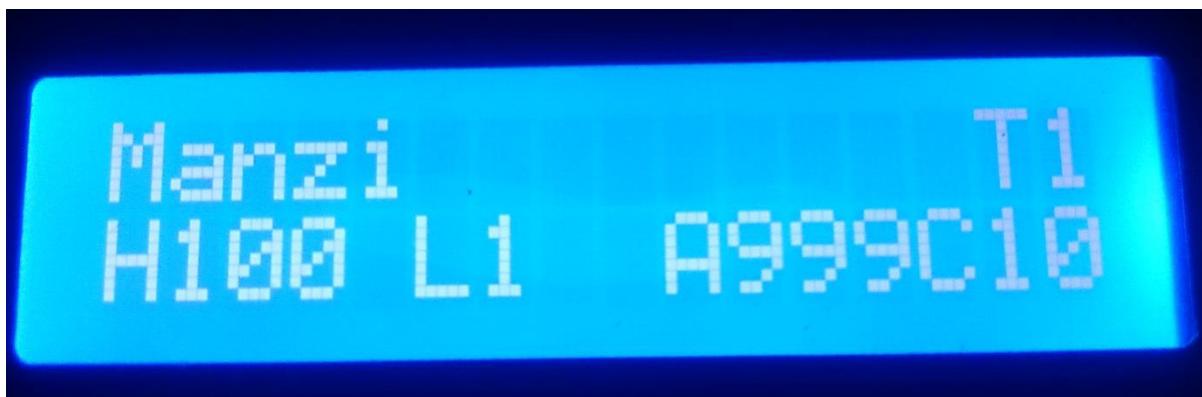


Abbildung 22: LCD-Anzeige nach Inbetriebnahme

Sollte dies der Fall sein, geben Sie einen Schuss ab und kontrollieren Sie, ob der Markierer den Munitionsverbrauch richtig berechnet und ob die Soundwiedergabe funktioniert (sofern sie diese Funktion aktiviert haben).

Feuern Sie nun auf den zweiten Markierer und kontrollieren Sie, ob diesem Markierer Lebenspunkte in Höhe Ihres Schadenswertes abgezogen werden.

Wenn dies alles funktioniert hat, feuern Sie so lange auf den anderen Tagger, bis er keine Lebenspunkte mehr hat, es sollte nun am Bildschirm des getroffenen Markierers Ihr Spielername erscheinen.

Drücken Sie während des Feuerns einmal auf den Reloadbutton und kontrollieren Sie die Soundwiedergabe und ob nach dem Nachladen das „Magazin“ wieder voll ist.

Starten Sie den zweiten Markierer neu und geben Sie nun einen Schuss auf Ihren Markierer ab und kontrollieren Sie, ob das Spielerpaket richtig empfangen wurde. Geben Sie dann so viele Schüsse ab, bis Sie die Lebenspunkte auf null sinken und kontrollieren Sie wiederum die Soundwiedergaben und die Anzeige am Bildschirm.

Starten Sie nun den Markierer neu. Die Erstinbetriebnahme ist nun abgeschlossen und Sie können nach dem Einstellen der Spieleparameter mit dem Spiel beginnen.

### 6.3 Spieleparameter

Ich möchte hier die Spieleparameter und ihre Auswirkung genauer beschreiben. Die Spieleparameter finden Sie in der Arduino-Programmierumgebung, wenn Sie die FreeTag-Software laden, relativ am Beginn des Programmes (*siehe Abbildung 16*). Die Eingabe von falschen Werten oder Parametern kann dazu führen, dass das Programm nicht richtig, oder gar nicht, funktioniert, kontrollieren Sie Ihre Eingaben daher bitte sorgfältig. Es folgen nun die Spieleparameter:

**PlayerID:** Dies ist Ihre Spielernummer. Zum einen wird diese Nummer bei Abgabe eines Schusses an den gegnerischen Tagger gesendet, zum anderen nutzt das Programm die PlayerID, um den Spielernamen aus der Liste enemy-Name zu ziehen (*siehe Abbildung 15*). Wenn zwei Spieler die selber Spielernummer verwenden so können zwar Signale zwischen den Markierern ausgetauscht werden, die Software verwirft die Daten aber, wenn sie bemerkt, dass man „sich selbst“ getroffen hat.

Als Spielernummer darf nur ein Wert zwischen 1 und 64 eingegeben werden.

**Team:** Die hier vergebene Nummer stellt die Teamzugehörigkeit des Spielers dar. Es gibt vier Teams (1 bis 4), welchen der Spieler angehören kann. Je nach verwendetem Spielmodus ergibt es Sinn, die Teamzugehörigkeit anzupassen, vor allem wenn der Freundesbeschuss (friendlyFire) aktiviert ist.

**Damage:** Hier können Sie den Schadenswert Ihres Taggers einstellen. Dieser Wert ist der Schaden, welcher dem Empfänger von seinen Lebenspunkten abgezogen wird, wenn Sie ihn treffen (sofern dies zulässig ist). Es sollte zuvor mit den Mitspielern geklärt werden, welche Kombinationen aus Schadenscode, Schussmodus, und Munitionsverbrauch erlaubt ist. Es hat z. B. nur sehr wenig Sinn einem Markierer einen hohen Schadenswert zu gewähren und ihm gleichzeitig eine hohe Magazingröße und vollautomatisches Schießen zu erlauben.

Mögliche Schadenswerte sind: 1, 2, 4, 5, 7, 10, 15, 17, 20, 25, 30, 35, 40, 50, 75, 100. Sollte ein anderer Wert gewählt werden, so funktioniert die Übertragung eines Schusses nicht.

**Health:** Dies sind die Lebenspunkte, welche man pro Leben besitzt. Anders ausgedrückt stellt dieser Wert dar, wie viel Schaden man erleiden kann bis man ein Leben verliert bzw. aus dem Spiel ausscheidet. Die Lebenspunkte können nicht unter null fallen auch wenn man rechnerisch darunterfällt. Fällt der Wert auf null Lebenspunkte und hat noch ein Leben zur Verfügung, so werden nach Verstreichen der Respawnzeit, die Lebenspunkte wieder auf den hier eingestellten Wert aufgefüllt.

Dieser Wert kann frei zwischen 1 und 100 gewählt werden.

**Life:** Der hier eingetragene Wert stellt die Anzahl der Leben dar, welche man zu Beginn des Spiels besitzt. Jedes Mal, wenn die Lebenspunkte auf 0 sinken, wird ein Leben verwendet und die Lebenspunkte werden auf den Anfangswert zurückgesetzt. Wird der Parameter „respawn“ auf false gestellt, und man besitzt kein Leben mehr, so scheidet man, wenn man auf null Lebenspunkte fällt, aus dem Spiel aus.

Erlaubt ist ein Wert zwischen 0 und 99. Wenn Sie einen höheren Wert eintragen, kann die Anzahl der Leben nicht mehr richtig am LCD-Display dargestellt werden.

**Ammo:** Hierbei handelt es sich um die Anzahl an Munition, oder anders formuliert, die Anzahl an virtuellen Patronen, welche sich im virtuellen Magazin befindet. Es ergibt Sinn, ein Vielfaches des Munitionsverbrauches pro Schuss (ammoUse) für diesen Wert zu wählen. Jedes Mal, wenn Sie einen Schuss abgeben, wird die Munition im Magazin um den Wert von ammoUse verringert bis dieser auf null fällt. Wenn zu wenig „Kugeln“ im Magazin vorhanden sind, so muss mit Betätigen des Reloadbuttons, nachgeladen werden. Beim Nachladen wird die Anzahl der Munition im Magazin auf den hier eingestellten Wert gesetzt.

Ein Wert zwischen 1 und 999 ist möglich. Wenn Sie einen höheren Wert eintragen, kann die Anzahl nicht mehr richtig am LCD-Display dargestellt werden.

**Clips:** Als Clips ist die Anzahl an virtuellen Magazinen, die man mitführt bezeichnet. Jedes Mal (sofern clipUse auf „true“ eingestellt wurde), wenn man den Reloadbutton drückt, wird dieser Wert um eins verringert und die Munition des Phasers aufgefüllt. Sinkt die Zahl der Magazine auf null, kann nicht mehr nachgeladen werden (wenn clipUse „true“ ist).

Ein Wert zwischen 0 und 99 ist möglich. Wenn Sie einen höheren Wert eintragen, kann die Anzahl nicht mehr richtig am LCD-Display dargestellt werden.

**ammoUse:** Dieser Wert gibt an, wie viele Patronen bei Betätigen des Abzuges, und somit bei Abgabe eines Schusses, verbraucht werden. Die Anzahl der Schuss, welche verbraucht werden, wird aus dem Magazin entnommen. Sollte die Anzahl der Patronen, welche sich im Magazin befinden, durch den Munitionsverbrauch unter eins sinken, dann ist die Schussabgabe nicht möglich.

Derzeit einprogrammierte Werte sind 1, 2 und 3. Sollte ein anderer Wert gewählt werden, so funktioniert die Schussabgabe nicht.

**respawnTime:** Dies ist die Zeit in Minuten, welche vergeht, bis man nach Verlust eines Lebens wieder am Spiel teilnehmen kann und die Lebenspunkte wieder aufgefüllt werden. Während dieser Zeit ist der Markierer gesperrt, kann also keine Signale abgeben und nur Signale von administrativen Geräten empfangen.

## FREETAG LASERTAGSYSTEM

---

Es kann ein Zeitraum von 1 bis 60 gewählt werden. Je länger diese Zeitspanne ist, desto stärker leidet jedoch der Spielspaß aufgrund langer „Pausenzeiten“.

**fullAuto:** Hier muss gewählt werden, ob der Markierer bei betätigtem Abzug durchgehend bzw. bis das Magazin leer ist, ein Schusssignal abgibt, das heißt dauerfeuert. Gibt man hier „true“ ein, so ist Dauerfeuer aktiv, gibt man „false“ ein, so muss nach jedem Schuss der Abzug entspannt werden, um einen neuen Schuss abzugeben. Es ergibt spielerisch Sinn, wenn Tagger höhere Schadenswerte erhalten, falls sie nur Einzelfeuer abgeben können, um mit Taggern konkurrieren zu können, welche durchgehend feuern können.

Mögliche Eingaben: „true“ Dauerfeuer ist aktiv, „false“ Dauerfeuer ist deaktiviert, wird etwas Anderes eingegeben, so lässt sich das Programm nicht mehr auf den Markierer hochladen.

**respawn:** Hier wird entschieden, ob man sozusagen über unendliche Leben verfügt. Setzt man diesen Wert auf true, so kann man, auch wenn man keine Leben mehr besitzt, nach Unterschreiten von null Lebenspunkten, wieder am Spiel teilnehmen, genauso als hätte man noch übrige Leben. Wird dieser Wert auf „false“ gesetzt, so scheidet man bei Verlust aller Leben und Lebenspunkte aus dem Spiel aus.

Mögliche Eingaben: „true“ das Spiel geht trotz des Verlusts aller Leben und Lebenspunkte weiter, „false“ der Markierer wird bei Verlust aller Leben und Lebenspunkte gesperrt. Wird etwas Anderes eingegeben, so lässt sich das Programm nicht mehr auf den Markierer hochladen.

**friendlyFire:** Setzt man den Wert auf „true“, so können Mitglieder des eigenen Teams beschossen werden und erhalten auch den übertragenen Schaden. Bei „false“ sind die Teammitglieder vor Beschuss der eigenen Teammitglieder sicher.

Wird etwas Anderes als „true“ oder „false“ eingetragen, so lässt sich das Programm nicht mehr auf den Markierer hochladen.

**clipUse:** Hier kann entschieden werden, ob beim Nachladen ein virtuelles Magazin verbraucht wird. Setzt man den Parameter auf „true“ so wird beim Nachladen die aktuelle Magazinanzahl um eins verringert. Sollte man keine Magazine mehr bei sich haben, so ist das Nachladen nicht mehr möglich. Wird dieser Wert auf „false“ gesetzt, so wird kein Magazin beim Nachladen verbraucht, man besitzt also unendlich viele Magazine.

Mögliche Eingaben: „true“ ein Magazin wird beim Nachladen verbraucht, „false“ man besitzt unendlich viele Magazine. Wird etwas Anderes als „true“ oder „false“ eingetragen, so lässt sich das Programm nicht mehr auf den Markierer hochladen.

**shortRange:** Mit diesem Parameter setzt man fest, ob der Markierer mit kurzer oder langer Reichweite arbeitet. Setzt man den Wert auf „false“, verwendet man die lange Reichweite. Die beiden Reichweiten wurden implementiert um eine spielerische Flexibilität zu ermöglichen. Es macht zum Beispiel Sinn, einen Markierer, der nicht vollautomatisch feuern kann, die lange Reichweite zu zuweisen. Setzt man hier den Wert auf „true“ so spielt man mit kurzer Reichweite. Die unterschiedlichen Reichweiten wurden bei Tageslichtverhältnissen getestet und so festgelegt, dass sie spielerisch für mich Sinn ergeben. Bei langer Reichweite kann das Signal über 60 m weit im Freien übertragen werden. Die kurze Reichweite ist so gedacht, dass das Signal im Freien ca. 50 m weit übertragen werden kann. Bei Dunkelheit würde zum Beispiel die kurze Reichweite bereits reichen, um das Signal über 60 m weit zu transportieren.

Mögliche Eingaben: „true“ lange Reichweite ist aktiviert, „false“ kurze Reichweite ist aktiviert. Wird etwas Anderes als „true“ oder „false“ eingetragen, so lässt sich das Programm nicht mehr auf den Markierer hochladen.

Die Unterschiedlichen Reichweiten können am Anfang verwirren. Die höchste Reichweite erreicht man mit shortRange=false und indoorMode=false, die kürzeste mit shortRange=true und indoorMode=true. Die kurze Outdoorreichweite (shortRange=true und indoorMode=false) ist höher als die lange Indoorreichweite (shortRange=false und indoorMode=true)

**indoorMode:** Mit diesem Parameter setzt man fest, ob der Markierer mit Indoor oder Outdoor Reichweite arbeitet. Setzt man den Wert auf „false“, verwendet man die lange Outdoor Reichweite, welche ich für Outdoorspiele empfehle, da es sonst zu Reflektionen von Wänden kommen kann. Setzt man hier den Wert auf „true“ so spielt man mit kurzer Indoor Reichweite, welche für Indoorspiele gedacht ist.

Mögliche Eingaben: „true“ Outdoor Reichweite ist aktiviert, „false“ Indoor Reichweite ist aktiviert. Wird etwas Anderes als „true“ oder „false“ eingetragen, so lässt sich das Programm nicht mehr auf den Markierer hochladen.

**ctf und flagFire:** Diese Punkte sind für den Spielmodus *Capture the Flag* gedacht, welcher noch in Entwicklung ist, diese Werte sollten auf „false“ gestellt werden, es ändert jedoch derzeit noch nichts, wenn der Wert auf „true“ gesetzt wird.

Mögliche Eingaben: „true“ oder „false“, die Eingabe hat derzeit noch keinen Effekt. Wird etwas Anderes als „true“ oder „false“ eingetragen, so lässt sich das Programm nicht mehr auf den Markierer hochladen.

**sound:** Hier wird entschieden ob man mit Soundwiedergabe vom Markierer spielen will oder nicht. Setzt man diesen Parameter auf „true“ so wird der Markierer Töne wiedergeben, sofern eine SD-Karte eingelegt ist. Wird dieser Wert auf „false“ gesetzt, so findet keine Tonwiedergabe statt.

Mögliche Eingaben: „true“ Töne werden wiedergegeben, „false“ die Tonausgabe ist deaktiviert. Wird etwas Anderes als „true“ oder „false“ eingetragen, so lässt sich das Programm nicht mehr auf den Markierer hochladen.

### 6.4 Spielmodi

Die hier vorgestellten Spielmodi sind lediglich Vorschläge und wurden von Videospielen inspiriert. Ich werde zu jedem Spielmodus eine kurze Beschreibung, sowie die nötigen Parameter dazu, angeben. Wenn ich gewisse Spieleparameter nicht anführe, dann deswegen, weil es mit dem Modus an sich nichts zu tun hat, sondern es eine Entscheidung der Spieler ist. Ob man sich bewegen darf oder nicht, wenn man sich gerade in der Respawnzeit befindet sollte zuvor geklärt werden. So lange der Markierer gesperrt ist, man sich also in der Respawnzeit befindet, leuchtet die Rote Hit-LED auf, man kann anhand der LED also erkennen ob jemand sich eventuell entgegen der Regeln bewegt. Ehrlichkeit unter den Spielern ist unbedingt notwendig, um den Spielspaß zu erhalten.

Um einen Spielmodus von vorne zu beginnen, müssen einfach nur alle Tagger neu gestartet werden.

#### 6.4.1 Deathmatch (Jeder gegen Jeden)

Jeder Spieler tritt gegen jeden anderen an, das heißt, dass man jeden Spieler treffen kann. Die Spieler, die nach einer zuvor bestimmten Zeit noch Lebenspunkte besitzen, haben gewonnen.

Spielparameter: friendlyFire = „true“, respawn = „false“

Wenn man das Spiel ohne Wertung spielen möchte, so braucht man den Wert respawn nur auf „true“ setzen und es wird nie ein Spieler ausscheiden. Wenn man plant, lange zu spielen, sollte clipUse auf „false“ gestellt werden damit einem die Munition nicht ausgeht.

#### 6.4.2 Team Deathmatch (Team gegen Team)

Ein Team tritt gegen bis zu drei Teams an. Man kann auf Zeit spielen, das heißt das Team, das am Ende der zuvor festgelegten Zeitdauer noch die meisten aktiven Spieler hat, gewinnt, oder man spielt so lange bis nur noch ein Team übrig ist. Die Spieler müssen in mindestens zwei Teams aufgeteilt werden. Es können natürlich auch 4 Teams gegeneinander antreten.

## FREETAG LASERTAGSYSTEM

---

Spieleparameter: respawn = „false“, Team = (Spieler müssen gleichmäßig auf die Teams aufgeteilt werden)

### 6.4.3 One Against All (Einer gegen alle)

Ein Spieler tritt gegen den Rest der Spieler an und versucht so lange im Spiel zu bleiben wie möglich. Am besten stoppt man die Zeit mit einer Stoppuhr bis der Spieler aus dem Spiel ausscheidet. Um das Spiel interessanter zu gestalten kann man einem gejagten Spieler einen höheren Schadenswert oder sehr viele Leben eintragen.

Spieleparameter für den Gejagten: respawn = „false“, Team = 1

Spieleparameter für die Anderen: respawn = „true“, Team = 2

### 6.4.4 King of the Hill

Für diesen Modus muss eine Stopp- bzw. Schachuhr an einen zuvor deklarierten Punkt gelegt werden. Es geht nun darum diesen Punkt so lange als möglich für sich in Anspruch zu nehmen, die Uhr darf dabei den Ort nicht verlassen. Wenn man diesen Punkt hält, so startet man die Uhr und versucht diesen Punkt für eine zuvor festgelegte Zeit zu halten. Wird man ausgeschaltet, so muss man den Ort verlassen und jeder darf versuchen die Uhr von vorne zu starten. Schafft man es so lange den Ort samt Uhr zu halten wie zuvor vereinbart wurde, so gewinnt man das Spiel. Es ergibt Sinn, die Respawnzeit etwas höher anzusetzen z. B. 5 Minuten, damit die Spieler nicht in Versuchung kommen, gleich nach ihrem Ausscheiden wieder in die Offensive zu gehen.

Das Spiel kann in Teams oder jeder gegen jeden ausführt werden, je nach Einstellung.

Spieleparameter für den Teammodus: respawn = „true“, Team = (Spieler müssen gleichmäßig auf die Teams aufgeteilt werden)

Spieleparameter für Jeder gegen Jeden: respawn = „true“, friendlyFire = „true“

### 6.4.5 Capture the Flag (mit physischer Flagge)

Die Spieler werden in zwei Teams aufgeteilt und jedes Team muss einen Ort als seine Basis deklarieren. In dieser Basis wird eine Flagge oder etwas Ähnliches platziert. Ziel ist es nun, die Flagge des gegnerischen Teams aus deren Basis zu stehlen und sie in die eigene Basis zu bringen. Die Flagge darf nur von Spielern getragen werden, welche noch am Leben sind. Wird ein Spieler ausgeschaltet während er die Flagge trägt, so muss er diese fallen lassen. Das Team das zuerst eine oder mehrere gegnerische Flaggen in ihre Basis bringt, gewinnt das Spiel.

Spieleparameter: Team = (Spieler müssen gleichmäßig auf zwei Teams aufgeteilt werden)

## 7 STÖRUNGSBEHEBUNG UND WARTUNG

In diesem Kapitel möchte ich eine grobe Übersicht über mögliche Fehler und deren Behebung geben. Es ist natürlich nicht möglich, jeden erdenklichen Fehler hier darzustellen. Sollten Sie Zweifel an der Funktion des Markierers haben, so unterbrechen Sie sofort die Spannungsversorgung durch Umdrehen des Schlüsselschalters und entfernen sie die Batterien bzw. Akkumulatoren aus dem Batteriefach.

Bei Störungen sollte ein Fachmann kontaktiert werden, damit keine Schäden an Mensch oder Gerät entstehen.

Zum Öffnen des Markierergehäuses bitte die Schrauben des Deckels lösen und den Deckel abheben.

### 7.1 Der Markierer lässt sich nicht einschalten

Stellen Sie sicher, dass Batterien bzw. Akkus im Batteriefach eingelegt sind, aufgeladen bzw. voll sind und dass die Polarität der eingelegten Akkus bzw. Batterien stimmt. Drehen Sie dann den Schlüsselschalter in die „Ein“-Position; das Licht am Schlüsselschalter sollte nun Leuchten und die LCD-Anzeige die Spielwerte zeigen.

Sollte das Licht am Schlüsselschalter nicht leuchten, so kontrollieren Sie die Akkumulatoren oder Batterien auf ihre Funktion und tauschen Sie diese gegebenenfalls aus. Sollte dies auch keine Abhilfe schaffen, so muss der Markierer vom Fachmann begutachtet und durchgemessen werden.

### 7.2 Das LCD-Display zeigt nichts an

Stellen Sie sicher, dass die Spannungsversorgung am Phaser gegeben ist (Akkus sind richtig eingelegt und der Schlüsselschalter ist in der „Ein“-Position) und kontrollieren Sie das Licht am Schlüsselschalter. Leuchtet das Licht am Schlüsselschalter, so überprüfen Sie, ob die Hintergrundbeleuchtung am LCD-Bildschirm leuchtet. Leuchtet diese, so laden sie die FreeTag-Software neu auf den Markierer und betätigen Sie den Abzug, die Muzzle-LED sollte aufleuchten und, sofern die Tonwiedergabe aktiviert wurde, sollte ein Geräusch ausgegeben werden.

Leuchtet die LED auf und wird ein Ton wiedergegeben, so funktioniert das Programm und Sie sollten die Kontrasteinstellung mit dem Potentiometer neben der Bildschirmverkabelung einstellen. Sollte dies nichts helfen, sollte der Bildschirm ausgetauscht werden.

### **7.3 Das LCD-Display zeigt eigenartige Zeichen an**

Starten Sie den Markierer neu. Sollten dies nichts helfen, so laden Sie die Software neu auf den Markierer und kontrollieren Sie die Steckverbindung und die Kabel zwischen Bildschirm und Platine. Hilft dies alles nichts, kontrollieren Sie die Software und/oder tauschen Sie das Display aus.

### **7.4 Die Hintergrundbeleuchtung des Displays funktioniert nicht mehr**

Wahrscheinlich ist der Mosfet defekt, welcher die Spannung für die Hintergrundbeleuchtung schaltet, lassen Sie den Mosfet vom Fachmann kontrollieren.

### **7.5 Das Betätigen des Abzuges hat keinen Effekt.**

Starten Sie das System neu, indem Sie den Markierer aus- und wieder einschalten. Kontrollieren Sie, ob nach Betätigen des Abzuges die Muzzle-LED aufleuchtet und ob der Munitionsverbrauch am LCD-Bildschirm dargestellt wird. Sollte beides nicht der Fall sein, so dürfte der Taster defekt sein und muss getauscht werden.

Leuchtet die Muzzle-LED auf und wird der Munitionsverbrauch am Bildschirm angezeigt, so halten sie eine Handykamera vor die Linse der IR-LED und drücken Sie den Abzug. Ist kein Licht zu erkennen so dürfte die IR-LED defekt sein und sollte getauscht werden. Ist ein Licht in der Kamera zu erkennen, so führen Sie bitte die Schritte der Erstinbetriebnahme erneut durch.

### **7.6 Obwohl ich auf Mitspieler ziele und den Abzug betätige, wird er nicht getroffen**

Halten Sie eine Handykamera vor die Linse der IR-LED und betätigen Sie mehrfach den Abzug, es sollten nun kurze Lichtimpulse in der Kamera zu sehen sein. Sollte dies nicht der Fall sein, so ist vermutlich die IR-LED defekt und muss getauscht werden.

Sollten Lichtimpulse in der Kamera zu erkennen sein, so kontrollieren Sie bitte die Einstellungen ihres Markierers in der Software oder führen Sie die Schritte der Erstinbetriebnahme erneut durch.

## 7.7 Die Soundwiedergabe des Markierers funktioniert nicht

Dies kann viele Ursachen haben. Stellen Sie zuerst sicher, dass der Spieleparameter *Sound* in der Software auf „true“ gesetzt ist. Ist dies der Fall und die Tonwiedergabe funktioniert trotzdem nicht, so kontrollieren Sie bitte, ob eine Micro SD-Karte in das Soundmodul eingelegt ist und ob Sie auch die richtigen Sounddateien (siehe Erstinbetriebnahme) darauf kopiert haben. Hilft dies alles nichts, versuchen Sie eine andere SD-Karte bzw. kontrollieren Sie, ob Sie die Lautstärke mit dem Potentiometer neben dem Verstärkerchip auch aufgedreht haben.

Wenn dies alles nichts hilft, so können entweder das Soundmodul, der Verstärkerchip oder der Lautsprecher defekt sein, lassen Sie diese von einem Fachmann kontrollieren.

## 7.8 Der Markierer verströmt einen brandähnlichen Geruch

Schalten Sie den Markierer sofort aus, entfernen Sie die Akkumulatoren bzw. die Batterien und lassen Sie die Elektronik von einem Fachmann kontrollieren.

## 8 TECHNISCHE GRUNDLAGEN

### 8.1 Übertragungsfrequenz

Das Infrarotlicht zur Übertragung wird mit einer Frequenz von annähernd 38 kHz gepulst, das heißt, dass die IR-LED für 13 µ-Sekunden mit Strom versorgt wird, und damit Licht abstrahlt, und dann die Stromzufuhr für 13 µ-Sekunden unterbrochen wird (kein Licht wird emittiert). Der so entstehende Puls dauert 26 µ-Sekunden an und wird so oft aneinander gereiht, bis die nötige Zeitspanne erreicht ist, wie es das Übertragungsprotokoll erfordert.

Die 26 µ-Sekunden Puls wurden gewählt, weil es annähern 38 kHz entspricht, welche vom IR-Empfänger benötigt werden. Um 38 kHz genau zu entsprechen müsste der Puls 26,31 µ-Sekunden dauern, leider kann das Arduinosystem im Mikrosekundenbereich nicht mit Nachkommastellen arbeiten. Mit der Pulsdauer von 26 µ-Sekunden ergibt sich eine Frequenz von 38461 Hz, welche vom IR-Empfänger gut verarbeitet werden kann.

Die Pulsdauer und die Frequenz ergeben sich aus folgender Formel:

**Frequenz f (in Hertz) = 1 / Pulszeit T (in Sekunden)**

$$f = \frac{1}{T}$$

**Pulszeit T = 1/Frequenz**

$$T = \frac{1}{f}$$

### 8.2 Übertragungsprotokoll

Damit der Empfänger versteht, welche Daten ihm der Markierer zusendet, wird ein sogenanntes Protokoll benötigt. Dies bestimmt, wie der Markierer seine Daten zu senden hat und wie diese vom Empfänger interpretiert werden müssen. Sollte das Protokoll nicht eingehalten werden, so erfolgt zwar eine Signalübertragung, sie wird aber von der Software nicht verwendet, sondern die Daten werden verworfen.

## FREETAG LASERTAGSYSTEM

---

Ich habe mich bei diesem Projekt für das Milestag 2<sup>2</sup>-Protokoll entschieden, da es sich dabei um ein Open Source Protokoll handelt, welches für nicht kommerzielle Zwecke verwendet werden darf. Außerdem findet dieses Protokoll bereits in ein paar Anlagen Anwendung, was bedeutet, dass es möglich ist mit oder gegen Tagger dieser Anlagen zu spielen.

Das Milestag 2-Protokoll definiert genau, wie das Signal aufgebaut sein muss. Das Infrarotsignal simuliert dabei eine Art Binärsignal, indem es den Binärwert „0“ als einen 600 µ-Sekunden langen Lichtimpuls festlegt. Der Binärwert „1“ muss laut Protokoll 1200 µ-Sekunden betragen. Zwischen jedem Binärwert, also zwischen jeder „0“ oder „1“ muss eine 600 µ-Sekunden lange Pause sein. Damit die Software erkennt, dass ein für sie relevantes Eingangssignal eintrifft, muss vor jeder Codeübertragung ein 2400 µ-Sekunden langer Impuls, ein so genannter Header, vorhanden sein. In Abbildung 23 ist der Aufbau eines Signales aufgezeigt.

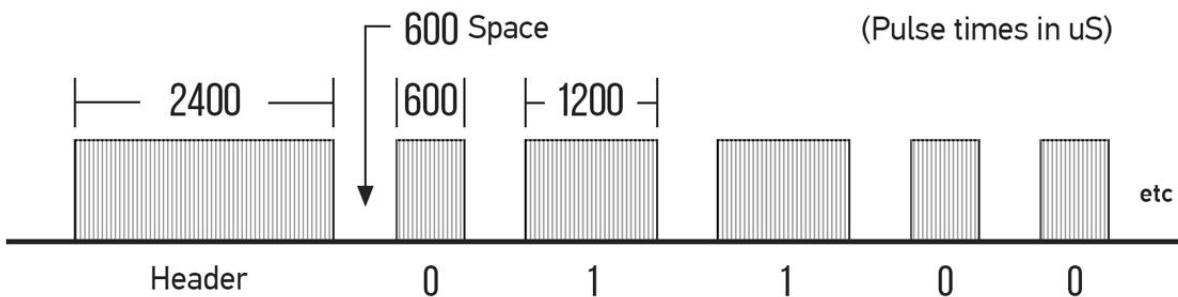


Abbildung 23: MilesTag 2 Übertragungsaufbau<sup>3</sup>

Dieses Binärsignal muss nun auch noch übersetzt werden, damit die Anlage weiß, welche Informationen gerade eingegangen sind, auch das ist in dem Protokoll festgesetzt. Dabei unterscheidet das Protokoll generell zwischen zwei Code- bzw. Datenpaketen, nämlich zwischen einem „Shot Package“ (Spielerpaket) und einem „Data Package“ (Datenpaket). Damit die Software weiß, ob es sich um ein Daten- oder Spielerpaket handelt, wird der erste Binärwert nach dem Header benötigt. Wird nach dem Header eine „1“ also ein 1200 µ-Sekunden langer Impuls gemessen, so handelt es sich um ein Datenpaket, bei einer „0“ hingegen wird ein Spielerpaket erfasst.

Das „Shot Package“ ist das Signal, welches zwischen Spieler ausgetauscht wird. Es beinhaltet die Spielernummer (PlayerID), den Schadenscode (Damage) und die Teamnummer (TeamID). Um diese Daten zu übertragen, wird nach dem Header zuerst ein 8 Bit und dann ein 6 Bit langer Binärcode übertragen.

---

<sup>2</sup> Quelle: <http://www.lasertagparts.com/mtformat-2.htm> Entnommen am 01.02.2018

<sup>3</sup> Quelle: <https://laserwar.com/support/dataproto> Entnommen am 01.02.2018

## FREETAG LASERTAGSYSTEM

---

Bei den ersten 8 Bit handelt es sich um die „0“, welche anzeigen, dass es sich um Spielerpaket handelt. Die weiteren 7 Bit enthalten die Spielernummer in Binärcode (was eine maximale Spieleranzahl von 127 ergibt, 1111111 in Binär = 127 in Dezimal). Die weiteren 6 Bits beinhalten zuerst die Teamnummer und dann den Schadenscode. Um den Aufbau zu verdeutlichen, folgen eine kleine grafische Darstellung, eine Tabelle mit den Umrechnungen des Binärcodes in die Teamnummer sowie dieselbe Umrechnung des Schadenscodes.

Aufbau Spielerpacket:

[Header] – [8bits] – [6bits]

[Header] – [„0“und Spielernummer] – [2Bits Teamnummer, 4 Bits Schadenscode]

Tabelle 5: Übersetzung Spielerpaket<sup>4</sup>

| Teamnummer | Schadenscode |           |              |           |              |
|------------|--------------|-----------|--------------|-----------|--------------|
| Binärcode  | Teamnummer   | Binärcode | Schadenscode | Binärcode | Schadenscode |
| 00         | 1            | 0000      | 1            | 1000      | 20           |
| 01         | 2            | 0001      | 2            | 1001      | 25           |
| 10         | 3            | 0010      | 4            | 1010      | 30           |
| 11         | 4            | 0011      | 5            | 1011      | 35           |
|            |              | 0100      | 7            | 1100      | 40           |
|            |              | 0101      | 10           | 1101      | 50           |
|            |              | 0110      | 15           | 1110      | 75           |
|            |              | 0111      | 17           | 1111      | 100          |

Der Binärcode für Spieler 1 von Team 1 mit einem Schadenscode von 50 würde wie folgt aussehen:

[Header] – [„0“Shotpackage, 0000001 Spielernummer] – [00 Teamnummer, 1101 Schadenscode]

Beim Datenpaket bzw. dem Data Package handelt es sich um Informationen die normalerweise von einem administrativen Gerät, wie z. B. dem Markierer eines Schiedsrichters oder einer Einheit, welche einem Lebenspunkte zurückgibt, versendet werden. Solche administrativen Geräte sind zwar nicht Teil dieses Projektes, teilweise wurden aber die Funktionen bzw. die Auswertung dieser Datenpakete bereits in der FreeTag-Software implementiert. Bevor ich auf die bereits implementierten Befehle, welche über ein Datenpaket ausgelöst werden können, eingehe, möchte ich den generellen Aufbau und die Funktion nach dem Milestag 2 Protokolls erklären.

Das Datenpaket unterscheidet sich im Grunde durch die Anzahl der versendeten Bits und dem ersten Bit nach dem Header vom Spielerpaket. Beim Datenpaket folgt nach dem Header eine binäre 1 (also ein 1200 µ-Sekunden langer Puls), dadurch weiß die

---

<sup>4</sup> In Anlehnung an MilesTag 2 Protokoll siehe: <http://www.lasertagparts.com/mtformat-2.htm>

## FREETAG LASERTAGSYSTEM

---

Software des Empfängers, dass es sich um ein Datenpaket handelt. Die binäre 1 ist auch bereits Bestandteil der ersten 8 Bit (entspricht einem Byte), welche übertragen werden. Insgesamt folgen nach dem Header drei 8 Bit große Blöcke, was drei Byte entspricht. Der erste Block dient dazu, bekannt zu geben um welchen Befehl es sich handelt, beim zweiten Block wird der Befehl genauer beschrieben (z. B. wird ihm ein Zahlenwert zugeordnet), beim dritten Block handelt es sich um eine Art Abschluss für die Übertragung. Dieser Abschluss hat den Dezimalwert 232, die Software weiß somit, dass das Signal beendet ist, und kann weiter die Richtigkeit der Übertragung kontrollieren. Sollte der Dezimalwert am Ende nicht dem Wert 232 entsprechen so wird das Datenpaket von der Software verworfen. Im Folgenden möchte ich nochmals auf den Aufbau des Paketes eingehen und mit einer Tabelle verdeutlichen, die die vom MilesTag 2 Protokoll vorgegebenen Befehle beinhaltet, wobei ich jene nicht erwähne, die ich für unwichtig halte. Für genaue Informationen zum Datenprotokoll besuchen sie bitte:

<http://www.lasertagparts.com/mtformat-2.htm> (in englischer Sprache)

Aufbau des Datenpaketes:

[Header] – [8bits] – [8bits] – [8bits]

[Header] – [Art des Befehls] – [dem Befehl zugehöriger Wert o. andere Angabe] – [Endbyte mit dem Binärwert 232]

Tabelle 6: Übersetzung Datenprotokoll<sup>5</sup>

| Wert Byte 1 | Befehl Byte 1              | Wert Byte 2 | Erklärung  |
|-------------|----------------------------|-------------|--|
| 128         | Füge Lebenspunkte hinzu    | 1-100       | Füge Spieler 1 bis 100 Lebenspunkte hinzu              |
| 129         | Füge Schüsse hinzu         | 1-100       | Füge dem Markierer 1 bis 100 mögliche Schüsse hinzu    |
| 131         | Admin Befehl               | 0           | Ausscheiden aus dem Spiel durch Schiedsrichter         |
| 131         | Admin Befehl               | 1           | Spiel pausieren bzw. fortsetzen                        |
| 131         | Admin Befehl               | 2           | Spiel starten  |
| 131         | Admin Befehl               | 4           | Ausgeschalteten Spieler wiederbeleben                  |
| 131         | Admin Befehl               | 6           | Dem Spieler die vollen möglichen Schüsse hinzufügen    |
| 131         | Admin Befehl               | 7           | Spiel beenden  |
| 131         | Admin Befehl               | 13          | Dem Spieler das volle Leben vom Spielbeginn hinzufügen |
| 138         | Magazine auf Startwert     | 0-15        | Munitionsbox ID (0-15)                                 |
| 139         | Fülle Lebenspunkte auf 100 | 0-15        | Heilungsbox ID (0-15)                                  |
| 140         | Flagge aufheben (CTF)      | 0-15        | Flaggenstation (0-15)                                  |

Der Dezimalcode für das Ausscheiden aus dem Spiel durch einen Schiedsrichterentscheid würde wie folgt aussehen:

[Header] – [Byte 1] – [Byte 2] – [Byte 3]

---

<sup>5</sup> In Anlehnung an MilesTag 2 Protokoll siehe: <http://www.lasertagparts.com/mtformat-2.htm>

[Header] – [Admin Befehl] – [Wert] – [Endbyte]  
 [Header] – [131] – [0] – [232]

Der Binärkode würde nach obigen Beispiel wie folgt aussehen:

[Header] – [Byte 1] – [Byte 2] – [Byte 3]  
 [Header] – [Admin Befehl] – [Wert] – [Endbyte]  
 [Header] – [10000011] – [00000000] – [11101000]

Es folgt nun eine Tabelle mit den bereits implementierten Befehlen der FreeTag-Software. Ich möchte darauf hinweisen, dass der Wert 9 in Byte 2 nicht im Originalprotokoll vorkommt, ich habe jedoch geplant ihn als einen Teil der Auswertung der Spielerdaten über Infrarot mit einer administrativen Auswerteeinheit zu verwenden.

*Tabelle 7: Implementierte Befehle via Datenpakete*

| Wert Byte 1 | Befehl Byte 1    | Wert Byte 2 | Erklärung  |
|-------------|------------------|-------------|--|
| 131         | Admin Befehl     | 0           | Ausscheiden aus dem Spiel durch Schiedsrichter         |
| 131         | Admin Befehl     | 1           | Spiel pausieren bzw. fortsetzen                        |
| 131         | Admin Befehl     | 2           | Spiel starten  |
| 131         | Admin Befehl     | 4           | Ausgeschalteten Spieler wiederbeleben                  |
| 131         | Admin Befehl     | 6           | Dem Spieler die vollen mögliche Schüsse hinzufügen     |
| 131         | Admin Befehl     | 7           | Spiel beenden  |
| 131         | Admin Befehl     | 9           | Auswerteeinheit bestätigt den Dateneingang             |
| 131         | Admin Befehl     | 13          | Dem Spieler das volle Leben vom Spielbeginn hinzufügen |
| 140         | Flagge Aufnehmen | 0-15        | Spieler erhält Flagge mit Nummer 0-15                  |

### 8.3 Zykluskorrektur divisorstart, divisorzero und divisorone

Um eine Frequenz von 38 kHz zu erzeugen, müsste zuerst die IR-LED 13 µ-Sekunden aufleuchten und dann für 13 µ-Sekunden erlöschen. Um nun so einen 2400 µ-Sekunden langen Impuls zu übertragen, müsste die Arduino-Plattform 92 Mal, die IR-LED für 13 µ-Sekunden aufleuchten und dann für 13 µ-Sekunden erlöschen, lassen. Um die Anzahl der Zyklen, welche das Gerät abarbeiten müsste, um einen Impuls mit 38 kHz zu erzeugen, sieht wie folgt aus:

$$\frac{\text{Zeitdauer } T \text{ des Impulses}}{\text{Zeitdauer } T \text{ eines Pulses}} = \text{Anzahl der Zyklen}$$

So würde z. B. für den 2400 µ-Sekunden langen Header-Impuls die Rechnung wie folgt aussehen:

$$\frac{\text{Zeitdauer } T \text{ des Impulses}}{\text{Zeitdauer } T \text{ eines Pulses}} = \frac{2400 \mu \text{ Sekunden}}{26 \mu \text{ Sekunden}} \approx 92$$

Leider benötigt das Arduino Mega auch Zeit für die Ausführung der Befehle, also zum Ein- bzw. Ausschalten der IR-LED, daher erhöht sich die Zeitdauer welche er für die Erzeugung eines 38-kHz-Pulses benötigt. In der FreeTag-Software sieht die Formel zur Errechnung der Zykluszeit daher wie folgt aus:

$$\frac{\text{Zeitdauer } T \text{ des Impulses}}{\text{divisor des Bits}} = \text{Zykluszahl}$$

In der Software sind jeweils eigene Werte für jeden Übertragswert, also Header, eins und null, vorgesehen, um diese Abweichung zu korrigieren, genannt *divisorstart* für den Header, *divisorzero* für die binäre „0“ und *divisorone* für die binäre „1“.

Leider arbeitet jeder mit leicht abweichender Geschwindigkeit, weswegen diese Korrekturwerte bei jedem Markierer bzw. jeder Arduino-Ausführung separat eingestellt werden müssen. Wie die Korrekturwerte genau eingestellt werden müssen ist in Punkt [5.2.6 Einstellen des Zählwertfehlers](#) ersichtlich.

Dadurch, dass die Arduinoplattform Zeit zum Schalten der IR-LED benötigt, verlängert sich die Pulszeit, was wiederum heißt, dass die Impulse nicht mit genau 38 kHz übertragen werden, dies kompensieren jedoch die Empfänger, da sie über ein weites Empfangsspektrum verfügen (*siehe Datenplatt TSOP4838*).

Es wäre natürlich auch möglich die Zeiten, in welchen die IR-LED zu- oder abgeschaltet wird zu verringern, für mich hat sich jedoch in Versuchen gezeigt, dass die Methode mit dem verwenden der Korrekturwerte einfacher ist und zu guten Ergebnissen führt.

## 8.4 Treiberschaltung LED

Theoretisch wäre es möglich, die LEDs direkt an einen Ausgang der Arduino-Plattform anzuschließen, um die LED damit zu steuern und mit Strom zu versorgen. Leider ist die Stromstärke, welche der Arduino Mikrocontroller zur Verfügung stellen kann, mit 500 mA begrenzt. Daher ist es nötig, die LEDs von der Hauptspannungsversorgung zu versorgen und den Arduino Mikrocontroller lediglich als Steuereinheit zu verwenden.

## FREETAG LASERTAGSYSTEM

---

Um dies zu verwirklichen wird eine Art Schalter benötigt, welchen Arduino steuern kann. Ich habe mich hierzu für einen N-Kanal Mosfet des Typs IRLZ 34N<sup>6</sup> entschieden, da er Ströme bis 27 A bei einer Spannung von 55 V sehr schnell durchschalten kann. Der Mosfet lässt sich dabei im weitesten Sinne mit einem Schütz vergleichen. Mit einem geringen Strom am Gateanschluss des Mosfets wird ein großer Strom zwischen Drain- und Source-Anschluss des Mosfets gesteuert.

Zur Berechnung des Vorwiderstandes einer LED wird folgende Formel herangezogen:

$$R = \frac{U_{Ges} - U_F}{I_F}$$

$R$ ...Vorwiderstand der LED

$U_{Ges}$ ...Gesamtspannung der Versorgung

$U_F$ ...Spannungsabfall an der LED

$I_F$ ...zulässiger Betriebsstrom der LED

Die Werte  $U_F$  und  $I_F$  sollten dabei aus dem jeweiligen Datenblatt der LED entnommen werden.

---

<sup>6</sup> Für die Genauen Leistungsdaten bitte das Datenblatt unter: <http://cdn-reichelt.de/documents/datenblatt/H100/IRLZ34N-IR.pdf>

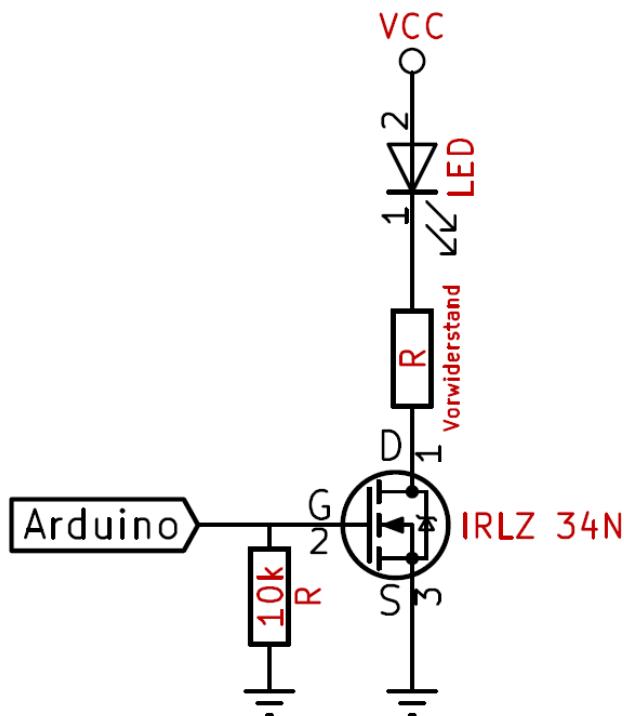


Abbildung 24: LED Treiberschaltung

Ob der Widerstand vor oder nach der LED geschaltet wird ist egal, es hat sich jedoch die Bezeichnung „Vorwiderstand“ für den strombegrenzenden Widerstand eingebürgert. Die hier verwendete Bezeichnung VCC ist gleich  $U_{Ges}$  bzw. der Versorgungsspannung der LED.

In diesem Projekt werden drei Arten von LEDs verbaut. Zum einen die IR-LED Typ TSAL 6100 zur Informationsübertragung, dann mehrere rote LEDs, welche dazu dienen anzuleuchten, wenn man getroffen wurde, weiters als Hit-Led bezeichnet und eine orange LED, welche bei Betätigung des Triggers aufleuchtet und unterhalb der IR-LED installiert ist, im weiteren Verlauf als Muzzle-LED bezeichnet. Für die genauen Bezeichnungen und die Links zu den Datenblättern der Muzzle- und Hit-LED bitte im Punkt Datenblattverzeichnis nachlesen.

Aus den jeweiligen Datenblättern der LEDs ergeben sich die Werte zur Berechnung des Vorwiderstandes. In der unten aufgeführten Tabelle sind die Daten für die Berechnung samt dem errechneten Vorwiderstand.

Die IR-LED ist hierbei eine Ausnahme in diesem Projekt, da sie bewusst mit einem zu hohen Strom  $I_F$  betrieben wird, um eine höhere Lichtausbeute und somit höhere Reichweite zu erzielen

## FREETAG LASERTAGSYSTEM

---

Tabelle 8: LED Vorwiderstände

| LED            | Versorgungsspannung<br>VCC ( $U_{Ges}$ ) | $U_F$ LED <sup>7</sup> | $I_F$ LED <sup>8</sup> | Vorwiderstand<br>berechnet |
|----------------|--|------------------------|------------------------|----------------------------|
| IR-LED         | 7,2 V                                    | 2,2 V                  | 1,51A                  | 3,3 Ω                      |
| Hit-LED        | 5 V                                      | 2V                     | 20mA                   | 150 Ω                      |
| Muzzle-<br>LED | 7,2 V                                    | 2,1 V                  | 20mA                   | 255 Ω                      |

Da Widerstände nicht in jedem beliebigen Wert verkauft werden, sollte der nächsthöhere Wert gewählt werden, in diesem Fall sind aber alle Werte erhältlich und werden auch verbaut.

---

<sup>7</sup> Siehe Datenplatt, Link dazu im Datenplattverzeichnis,  $U_F = V_F$  aus dem Datenplatt

<sup>8</sup> Siehe Datenplatt, Link dazu im Datenplattverzeichnis

## 8.5 IR-Licht Bündeln

Das Licht, welches von der IR-LED abgegeben wird, verliert durch seine kegelförmige Ausbreitung schnell an Kraft und kann nicht mehr vom Empfänger registriert werden, daher ist es nötig das Licht zu bündeln bzw. zu fokussieren. Infrarotes Licht kann dabei genauso gebündelt werden wie das für uns sichtbare Licht und zwar mit einer Optischen Linse.

Bei der Linse handelt es sich um eine sogenannte Sammellinse, diese hat die Aufgabe, Lichtstrahlen durch Brechung auf ihrer Oberfläche in Richtung ihrer Mittelachse umzulenken. Treffen zum Beispiel zwei parallel verlaufende Lichtstrahlen auf eine Sammellinse, so werden diese zueinander gelenkt (siehe Abbildung 25). Durch diese Umlenkung treffen sich diese Strahlen an einem gewissen Punkt, den Abstand zwischen Linse und diesem Punkt nennt man Brennweite. Nun kann man nicht nur Licht auf einen Punkt bündeln, sondern Lichtstrahlen auch so ablenken, dass Sie parallel zueinander verlaufen.

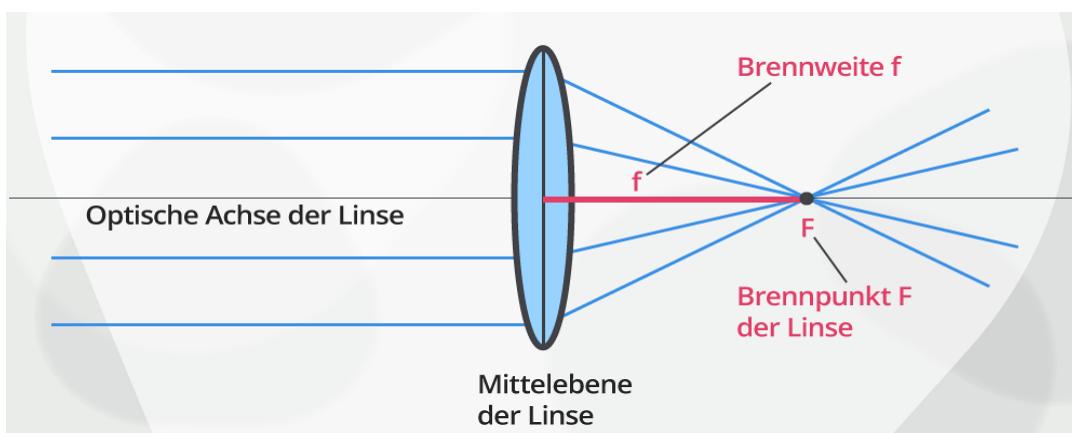


Abbildung 25: Funktionsweise Sammellinse<sup>9</sup>

Anhand der Grafik, in welcher die Lichtstrahlen als blaue Linien dargestellt werden, kann man erkennen, wie Licht gebündelt wird. Wenn man sich nun vorstellt, dass am Brennpunkt F die IR-LED sitzt und Licht in ihrem Abstrahlwinkel abgibt, kann man sich die Aufgabe der Linse veranschaulichen.

Wenn man nun versuchen möchte, das Licht der IR-LED zu parallelisieren, so sind drei Faktoren zu berücksichtigen, zum einen der Abstrahlwinkel der LED, dann noch die Brennweite und der Durchmesser der Linse. Der Abstrahlwinkel der LED kann aus dem zugehörigen Datenblatt entnommen werden. Wie in der folgenden Grafik ersicht-

---

<sup>9</sup> Entnommen von: <https://www.br.de/alphalernen/faecher/physik/2-optische-linsen-fotografie100.html>  
am 03.03.2018

## FREETAG LASERTAGSYSTEM

---

lich ist, lässt sich aus der Brennweite und dem halben Abstrahlwinkel ein Dreieck konstruieren, bei dem die Gegenkathete des halben Abstrahlwinkels dem halben Linsendurchmesser entspricht.

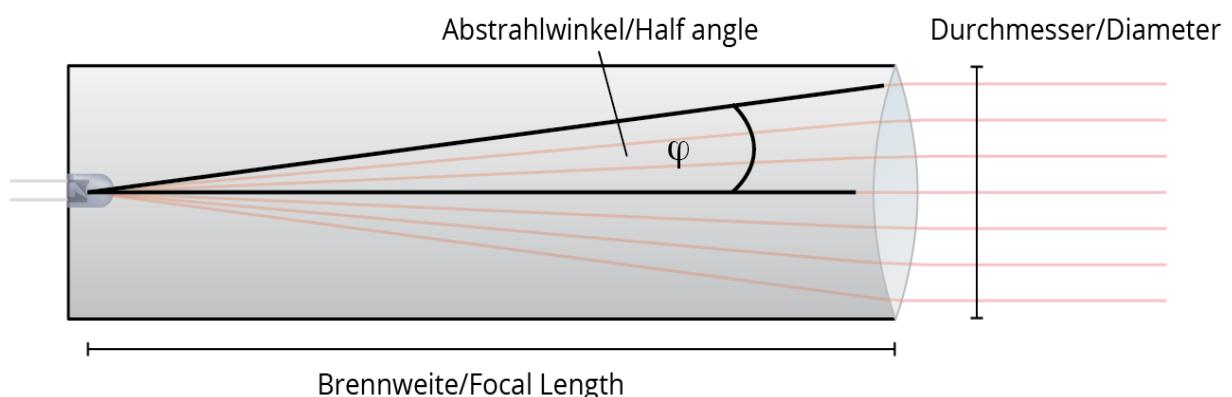


Abbildung 26: Abstrahlwinkel und Linse IR-LED<sup>10</sup>

Wenn man nun auf der Suche nach einer passenden Linse ist, so muss zuerst der halbe Abstrahlwinkel  $\varphi$  der IR-LED in Betracht gezogen werden. Dieser beträgt bei der TSAL6100-LED 10°. Nun sollte man sich Gedanken über den Durchmesser der Linse machen, welche man verwenden möchte. Optische Linsen bekommt man in allen möglichen Durchmessern mit vielen unterschiedlichen Brennweiten. Man sollte sich auch Gedanken darüber machen, wie die Linse fixiert werden soll. Meiner Erfahrung nach hat sich gezeigt das nicht beschichtete Glaslinsen mit großem Durchmessern das Licht am besten bündeln, wobei jedoch zu große Linsen sehr schwer sind und die Linsenhalterung dementsprechend groß ausfällt.

Aus all diesen Überlegungen sollte jetzt ein Durchmesser für die Linse gewählt werden. Im Internet findet man diverse Shops für optische Linsen, aus denen man sich eine Liste an möglichen Kandidaten erstellen sollte, nun muss rechnerisch ermittelt werden welche Linsen wirklich passen. Am besten setzt man den Durchmesser der Linse, welche man in Betracht zieht, in die unten folgende Formel ein, und vergleicht das Ergebnis mit der Brennweite der Linse.

$$f = \frac{r}{\tan(\varphi)}$$

r...Radius oder der halbe Durchmesser der Linse

f...Brennweite der Linse

$\tan(\varphi)$ ... Tangens des halben Abstrahlwinkels

---

<sup>10</sup> Entnommen von: [http://www.pewduino.org/wp-content/uploads/2013/04/optic\\_lens\\_assembly.png](http://www.pewduino.org/wp-content/uploads/2013/04/optic_lens_assembly.png) am 03.03.2018

## FREETAG LASERTAGSYSTEM

---

Es wird nur selten passieren, dass eine Linse genau den passenden Durchmesser und die passende Brennweite hat, aber es ist durchaus möglich, eine Linse zu finden, deren Brennweite nicht weiter als 5 mm von dem errechneten, idealen Brennweite abweicht.

Meine Erfahrung hat auch gezeigt, dass eine Linse, deren Brennweite höher ist als die errechnete Brennweite, besser funktioniert. Dies lässt sich dadurch erklären, dass die Umlenkung des Lichtes Richtung optische Mitte, weniger stark ausfällt, je höher der Brennwert ist. Das heißt zwar, dass das Licht sich weiter leicht kegelförmig ausbreitet, dies geschieht jedoch nur in einem geringen Ausmaß.

Würde man das Licht jedoch zu stark in Richtung optischer Mitte der Linse ablenken, so würde das Licht zuerst auf den Brennpunkt der Linse zusammengeführt werden und danach würde es sich stark kegelförmig ausbreiten (siehe dazu Abbildung 25), was dazu führt, dass die gespeicherte Energie des Lichtes sich über eine immer größere werdende Fläche verteilt und somit stark abnimmt, bis sie nicht mehr messbar ist.

Bei diesem Projekt habe ich eine bikonvexe Glassammellinse vom Versandhändler Sureplus Shed verwendet (<https://www.surplusshed.com/><sup>11</sup>), welche unter der Artikelbezeichnung L 10971 geführt wird. Der Durchmesser der Linse beträgt 42,4 mm bei einer Brennweite von 125 mm.

Setzt man den halben Durchmesser nun in die vorher aufgeführte Formel, so ergibt sich eine optimale Brennweite von 120,23 mm, was sehr nahe an der Brennweite der Linse dran ist und vollkommen für die Zwecke dieses Projektes ausreicht, zudem ist die Brennweite der Linse auch noch höher, als die errechnete Brennweite, was, wie oben erklärt, dem Ganzen zugutekommt.

Um die Linse zu montieren, habe ich eine Verbindungs muf f für Aufputzinstallationsrohre an einer Seite auf die Größe der Linse gedreht und mit Silikonkleber fixiert. Den Abstand zwischen Linse und IR-LED wurde dann mit einem Aufputzinstallationsrohr, einer weiteren Verbindungs muf f und einem Endstück hergestellt. Der Abstand zwischen LED und Linse beträgt dabei 121 mm.

---

<sup>11</sup> Entnommen am 04.03.2018

## 8.6 IR-Empfänger<sup>12</sup>

Als Empfänger für die IR-Signale kommen mehrere Empfänger des Types TSOP4838 von Vishay zum Einsatz. Diese haben die Eigenschaft, dass sie, wenn sie mit IR-Licht, welches mit einer Frequenz von annähernd 38 kHz und einer Wellenlänge von 940 nm auf sie trifft, den Stromfluss zwischen VSS (Pin 3) und OUT (Pin 1) unterbrechen. Die Software erkennt diese Unterbrechung, da der OUT-Pin des Sensors mit der Arduino-Plattform verbunden ist und überprüft, ob Spannung am Eingangspin anliegt.

Wenn die Stromzufuhr zum Arduinoeingangspin unterbrochen wird, also das Signal „LOW“ ist. So erfasst die Arduino-Plattform die Zeit, wie lange das Signal unterbrochen ist und verwendet diese zur Auswertung der Signale.

Um in einem möglichst großen Bereich Signale empfangen zu können sind pro Sensorkopf, drei Stück des IR-Empfängers verbaut, welche um 120 Grad zueinander versetzt sind. Die Grundschaltung für den Sensorkopf stammt dabei von Vishay selbst (siehe Datenblatt) und wurde von mir lediglich um 2 zusätzliche Sensoren erweitert. Der Kondensator und der Widerstand dienen dazu, die Sensoren vor Spannungs- oder Stromschwankungen zu schützen.

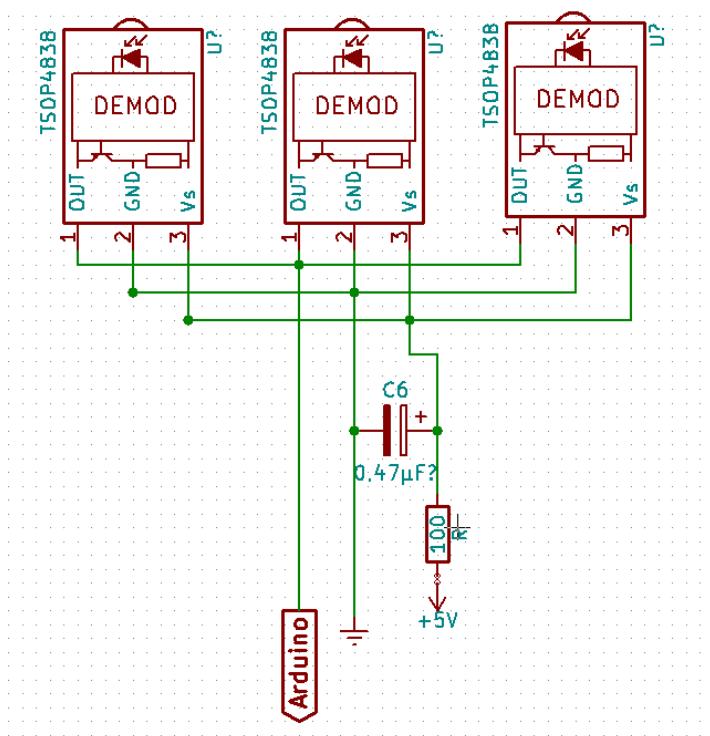


Abbildung 27: Schaltung IR-Sensoren

---

<sup>12</sup> In Anlehnung an: <https://www.vishay.com/docs/82459/tsop48.pdf> Entnommen am 02.02.2018

## FREETAG LASERTAGSYSTEM

---

In jedem Sensorkopf ist zusätzlich noch eine LED (Hit-LED) zum Anzeigen von Treffern montiert. Die Schaltung in einem Sensorkopf sieht dann wie folgt aus:

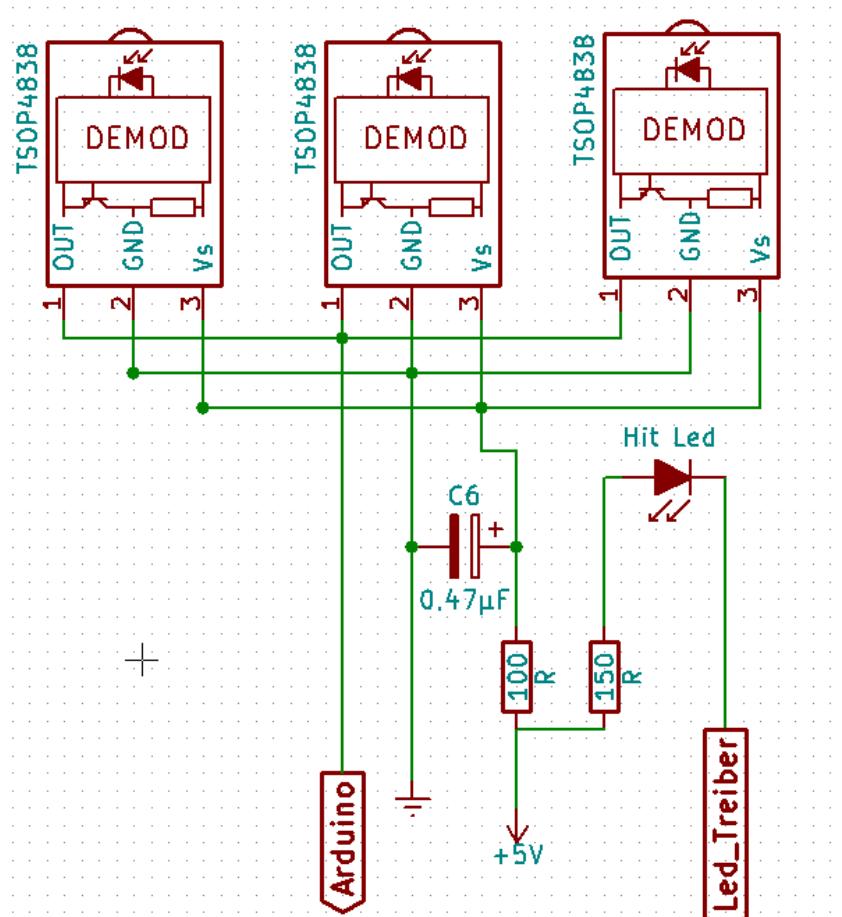


Abbildung 28: Schaltung IR-Sensorkopf inkl. Hit-LED



Abbildung 29: Schaltung verbaut in dem Sensorkopfgehäuse

## 8.7 Erstellen Eigener Sounds

Um eigene Sounds für den Phaser zu verwenden, müssen Sie zuerst die nötigen Sounddateien im .mp3- oder .wav-Format besitzen. Falls Sie diese noch nicht haben, besteht die Möglichkeit sich welche im Internet zu suchen oder selbst zu erstellen. Im Internet findet man unter dem Begriff „Royalty Free Audio Files“ große Datenbanken an Sounddateien, welche frei verwendet werden dürfen. Um Audiodateien selbst zu erstellen oder zu schneiden, empfehle ich das Programm Audacity (<http://www.audacity.de/downloads/>), welches kostenlos aus dem Internet heruntergeladen werden kann.

Der Markierer gibt die Sounddateien nach dem folgenden Schema wieder, bitte bedenken Sie dies und benennen Sie Ihre Sounddateien dahingehend um.

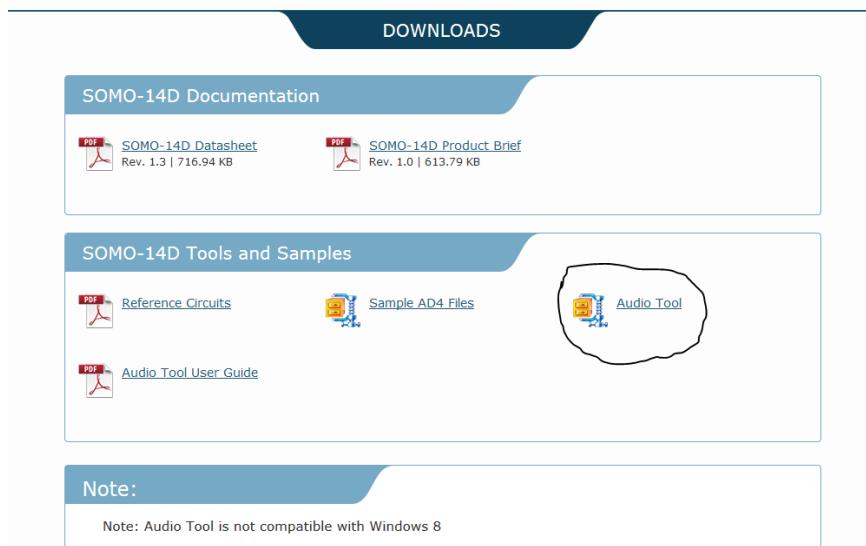
Tabelle 9: Soundwiedergabeprotokoll

| Aktion                        | Name der Sounddatei die wiedergegeben wird |
|-------------------------------|--|
| Abgabe Schuss (AmmoUse = 1)   | 1  |
| Abgabe Schuss (AmmoUse = 2)   | 2  |
| Abgabe Schuss (AmmoUse = 3)   | 3  |
| Magazin wird nachgeladen      | 4  |
| Man wird getroffen            | 5  |
| Die Lebenspunkte fallen auf 0 | 6  |
| Respawnzeit ist vorüber       | 7  |
| Magazin ist leer              | 8  |

## FREETAG LASERTAGSYSTEM

---

Wenn Sie die Audiodateien nach dem obigen Schema umbenannt haben, laden Sie sich bitte das Convertortool SOMO AD4 von [https://www.4dsystems.com.au/product/SOMO\\_14D/](https://www.4dsystems.com.au/product/SOMO_14D/) herunter. Das Programm ist als Audio Tool durch runterscrollen auf der Internetseite zu finden (siehe Abbildung 30).



*Abbildung 30: Download 4ad Converter*

Nach dem Herunterladen entpacken Sie das Programm und installieren Sie es auf Ihrem Rechner. Wenn die Installation erfolgreich war, sollte sich unter „Start“ - „Alle Programme“ ein Programm mit dem Titel „Usb Recorder“ befinden. Öffnen Sie dieses. Folgendes Bild sollte vor Ihnen erscheinen (*Abbildung 31 nächste Seite*).

Klicken Sie unten auf den Reiter „Local file“ (1), danach auf die drei Punkte in der Mitte oben (2) und dann auf Open Folder (3). Suchen Sie jetzt den Ordner, in dem Sie die Dateien gespeichert haben, welche sie umwandeln wollen und drücken sie auf „OK“. Es werden nun alle Dateien des Ordners in das Programm geladen. Sollten Sie keine Auflistung der Sounddateien im Programm sehen, so wählen Sie die richtige Dateiart unten aus (4).

Überprüfen Sie noch einmal ob die richtigen Dateien in das Programm geladen wurden und drücken Sie dann auf Encode (5). Das Programm erstellt nun die Sounddateien im .4ad-Format in dem Ordner, wo die Grunddateien zu finden sind.

## FREETAG LASERTAGSYSTEM

---

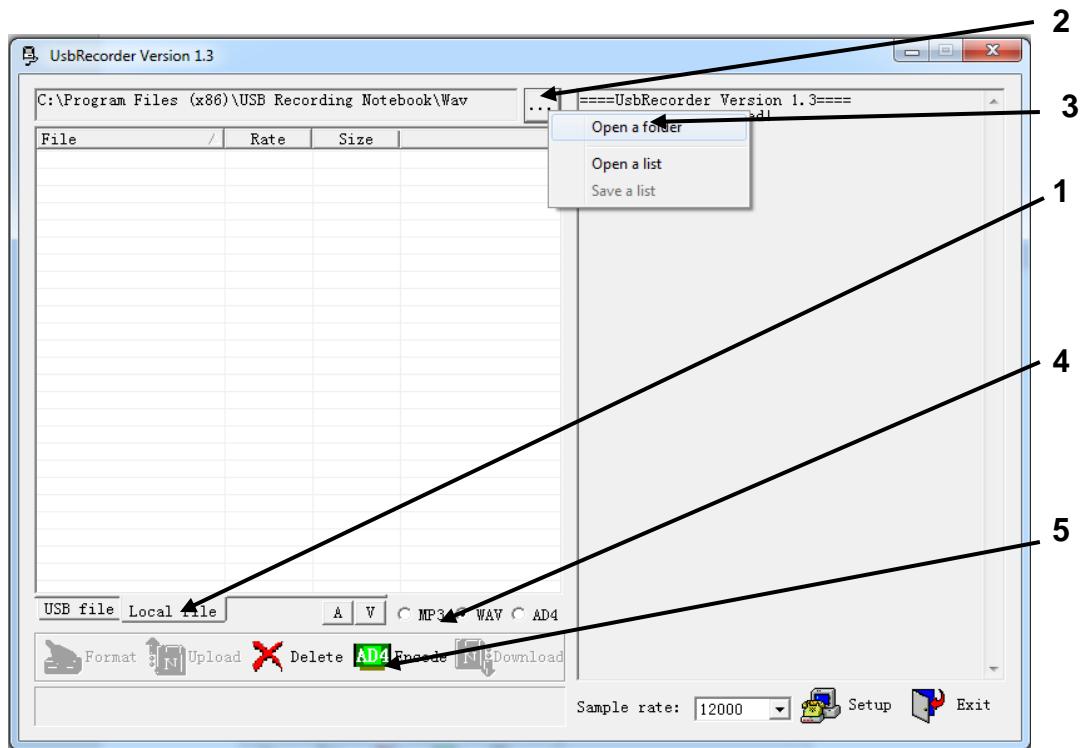


Abbildung 31:USB Recordertool

Kopieren Sie die .4ad-Dateien nun auf die SD-Karte und setzen Sie diese in das Soundmodul ein. Nehmen Sie den Markierer nun in Betrieb und testen Sie, ob die Soundwiedergabe reibungslos funktioniert.

## 9 BEGRIFFLICHKEITEN

### 9.1.1 Airsoft

Eine Sportart welche mit Waffenrepliken durchgeführt wird, welche Runde Kunststoff oder Keramikkugeln mit einem Durchmesser von 6 mm verschießen. Die Kugeln werden dabei mit Luftdruck beschleunigt, der von einer Feder erzeugt wird. Meist wird dieser Sport auch in Tarnkleidung durchgeführt. Treffer müssen vom Getroffenen deklariert werden, was häufig zu Problemen führt, da die Kugeln oft nicht spürbar sind, oder einfach geschummelt wird. Außerdem ist eine Schutzbrille von Nöten, um Verletzungen der Augen durch die kleinen Kugeln zu vermeiden.

### 9.1.2 Arduino

Beim Arduino handelt es sich um eine Physical-Computing Plattform welche aus einem Mikrocontroller samt Ein- und Ausgängen welche über den Controller steuerbar sind. Das Arduinoboard wurde entwickelt um auch nicht versierten Personen einen möglichst einfachen Zugang in die Welt der Mikrocontroller zu ermöglichen. Die Programmierumgebung nutzt eine Abwandlung der Programmiersprache C++. Es gibt mittlerweile eine Menge unterschiedliche Varianten der Plattform, welche sich in Rechenleistung, Größe, Anzahl der Ein- und Ausgänge und weiteren Parametern unterscheiden. Die Mikrocontrollerboards sind quelloffen, wodurch es auch einige Kopien am Markt gibt, die jedoch gleichwertig sind. Für weiter Informationen zum Arduino-Projekt empfehle ich folgende Links:

<https://de.wikipedia.org/wiki/Arduino> (Plattform)

<https://www.arduino.cc/en/Guide/Introduction> (in englischer Sprache)

### 9.1.3 COM-Port

Eigentlich ist der COM-Port eine veraltete Schnittstelle, welche kaum noch bei PCs zu finden ist. Der Arduino fungiert hier als USB-COM-Port-Adapter um die Daten auf den Mikrocontroller des Arduinoboards zu senden.

### 9.1.4 Datenpaket

Das Datenpaket ist die Summe an Informationen, welche von Administrativen Geräten an den Markierer gesendet werden. Für genauere Infos siehe Punkt 8.2

### 9.1.5 Empfänger

Als Empfänger werden alle Geräte bezeichnet, welche Signale empfangen, egal ob via Infrarot oder anders. Außerdem kann damit im speziellen das IR-Empfängerbauteil TSOP 4838 bzw. die Empfängerbaugruppe (z. B. vom Stirnbandsensor) gemeint sein.

### 9.1.6 GNU-Lizenz

Wie im Punkt 2.2 beschrieben handelt es sich hierbei um einen Lizenzvertrag, welcher die Veröffentlichung und Verbreitung von freien Dokumenten ermöglicht, solange damit kein gewerblicher Nutzen erzielt wird. Der Ersteller und Besitzer der Dokumente darf diese natürlich auch gewerblich nutzen.

### 9.1.7 Header

Als Header wird, in diesem Projekt, der erste Teil der Datenübertragung via Infrarot bezeichnet. Dieser Impuls dauert 2400 µ-Sekunden an und zeigt der Software bzw. der Arduino-Plattform den Eingang eines relevanten Signales an.

### 9.1.8 Hit-LED

In diesem Projekt wird die rote LED, welche zum Anzeigen von Treffern dient, als Hit-LED bezeichnet. Sie dient außerdem dazu, erkenntlich zu machen, wenn ein Spieler komplett ausgeschieden ist (dauerleuchten) oder die virtuell Flagge trägt (blitzen der Hit-LED)

### 9.1.9 Hoch- und Tiefpassfilter

Hoch- und Tiefpassfilter sind Schaltungen, welche nur gewisse Frequenzen unge-dämpft passieren lassen. Ein Hochpassfilter lässt dabei hohe Frequenzen passieren, ein Tiefpassfilter tiefe Frequenzen.

In diesem Projekt werden Hoch- und Tiefpassfilter bei der Audioschaltung zur Soundwiedergabe verwendet.

Ein einfacher Hochpassfilter erster Ordnung (*siehe Abbildung 32*) besteht aus einem Kondensator und einem Widerstand, man kann diese Konstellation auch RC-Glied nennen.

Bei niedrigen Frequenzen stellt der Kondensator einen großen Widerstand dar, wodurch der Strom stark begrenzt wird. Bei hohen Frequenzen ist der Widerstand,

## FREETAG LASERTAGSYSTEM

---

welcher der Kondensator dem Strom entgegenbringt, gering und der Strom kann bei nahe ungehindert passieren.

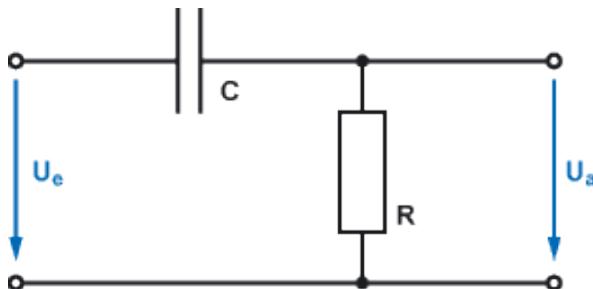


Abbildung 32: Hochpassfilter erster Ordnung<sup>13</sup>

Ein Tiefpassfilter erster Ordnung besteht ebenfalls aus einem RC-Glied, es werden jedoch die Position von Widerstand und Kondensator, im Vergleich zu einem Hochpassfilter, vertauscht (siehe Abbildung 33).

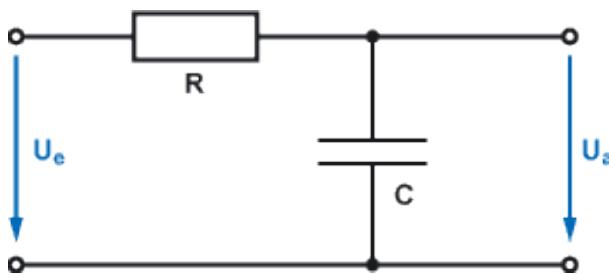


Abbildung 33: Tiefpassfilter erster Ordnung<sup>14</sup>

Bei hohen Frequenzen stellt der Kondensator nur einen geringen Widerstand dar und der Strom wird Richtung  $U_e$  abgeleitet. Bei tiefen Frequenzen ist der Widerstand, den der Kondensator darstellt, sehr hoch und der Strom fließt in Richtung  $U_a$ .

### 9.1.10 Indoor

Indoor ist Englisch und bedeutet frei übersetzt „innen“ bzw. „in einem Raum“. Im sportlichen Bereich verwendet man den Begriff Indoor, wenn man den Sport in einer Halle oder ähnlichem durchführt.

---

<sup>13</sup> Quelle: <https://www.elektronik-kompendium.de/sites/slt/schalt/02061711.gif> entnommen am 09.02.2018

<sup>14</sup> Quelle: <https://www.elektronik-kompendium.de/sites/slt/schalt/02061721.gif> entnommen am 09.02.2018

### 9.1.11 IR-LED

IR steht für Infrarot, ein Lichtspektrum, welches vom menschlichen Auge nicht wahrgenommen werden kann. Infrarotes Licht ist im Lichtspektrum rechts neben dem roten Licht angesiedelt, hat also eine größere Wellenlänge. Bei einer Wellenlänge von 780 nm bis 1 mm spricht man von Infrarotlicht. Die in diesem Projekt verwendete IR-LED emittiert Licht mit einer Wellenlänge von 940nm.

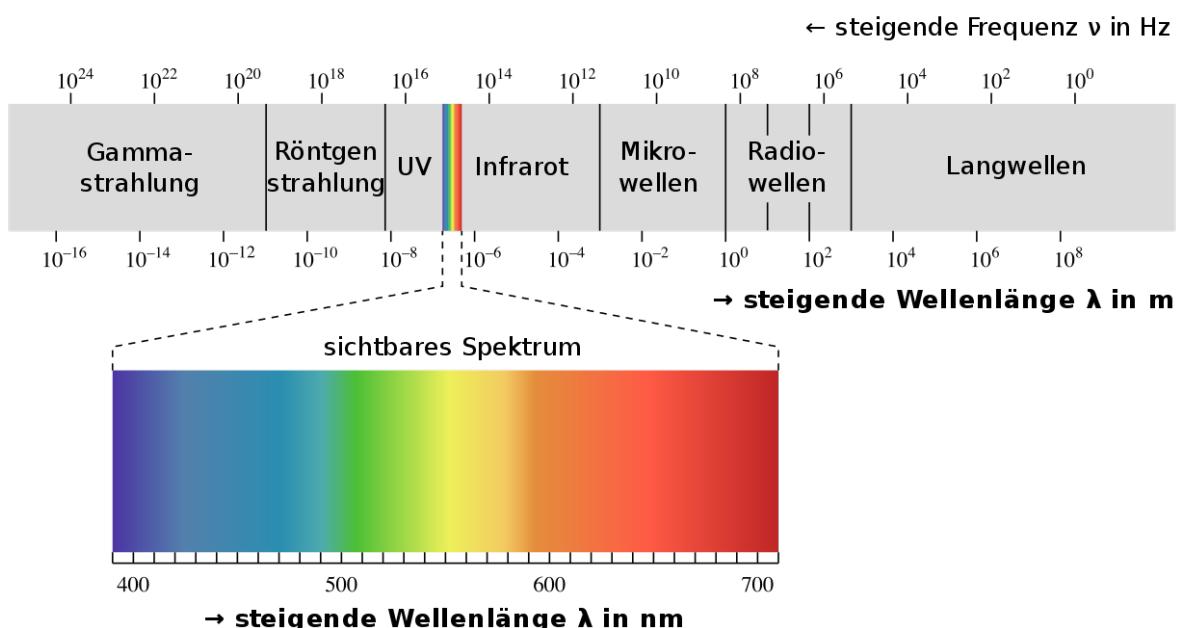


Abbildung 34: Lichtspektrum mit Wellenlängen und Frequenz<sup>15</sup>

### 9.1.12 Kondensator

Ein Kondensator ist ein passives Bauteil, welches Ladungen bzw. elektrische Energie unter Zuhilfenahme des elektrischen Feldes, speichert, welches bei Potentialunterschied zwischen zwei Leitern entsteht.

Die einfachste Form eines Kondensators besteht aus zwei Leiterplatten, welche durch einen Isolator, genannt Dielektrikum, voneinander getrennt sind (siehe Abbildung 34).

---

<sup>15</sup> Quelle: <http://denkwerkstatt-physik.de/denkwerkstatt-physik/awk/w-Licht.html> Entnommen am 02.02.2018

## FREETAG LASERTAGSYSTEM

---

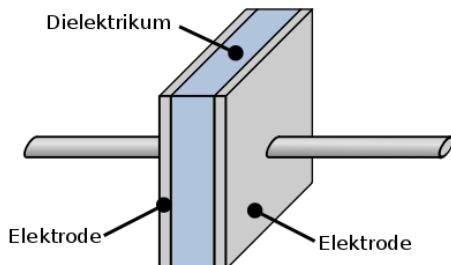


Abbildung 35: Prinzipdarstellung Kondensator<sup>16</sup>

Die Speicherkapazität C eines Kondensators wird dabei in Farad angegeben, was wiederum der Ladungsmenge Q (in Coulomb) pro Volt entspricht. Vereinfacht gesagt gibt die Kapazität Auskunft darüber, wie viele Elektronen pro Volt gespeichert werden können, d. h. wiederum, dass die Höhe der gespeicherten Ladungsmenge mit Erhöhung der Spannung zunimmt. Die Formel zur Berechnung der Ladungsmengen lautet wie folgt:

$$\begin{aligned}Q &= C * U \\I * t &= C * U \\C &= \frac{I * t}{U} \\1F &= \frac{1A * 1\text{ sek.}}{1V}\end{aligned}$$

Es gibt gepolte Kondensatoren (Elektrolytkondensatoren) sowie ungepolte Kondensatoren (Keramik und Folienkondensatoren). In diesem Projekt werden Kondensatoren zum Glätten von Spannungen und als Hoch- sowie Tiefpassfilter verwendet. Meistens werden polarisierte Elektrolytkondensatoren zum Glätten der Spannungen eingesetzt.



Abbildung 36: Schaltzeichen Kondensatoren<sup>17</sup>

<sup>16</sup> Quelle: [https://de.wikipedia.org/wiki/Kondensator\\_\(Elektrotechnik\)#/media/File:Plate\\_Capacitor\\_DE.svg](https://de.wikipedia.org/wiki/Kondensator_(Elektrotechnik)#/media/File:Plate_Capacitor_DE.svg) entnommen am 08.02.2018

<sup>17</sup> Quelle: <http://rn-wissen.de/wiki/images/archive/5/57/20081017144656%21Kondensator-Schaltzeichen.jpg> Entnommen am 08.02.2018

An Wechselspannung lädt und entlädt sich ein Kondensator permanent, bringt dem Strom jedoch einen gewissen Widerstand entgegen, welchen man kapazitiven Blindwiderstand nennt. Er ist mit dem ohm'schen Widerstand soweit ident, als dass er den Strom begrenzt, er führt jedoch auch zu einer Phasenverschiebung zwischen Strom und Spannung.

Je höher die Frequenz der Wechselspannung, desto geringer ist der Blindwiderstand, der durch den Kondensator verursacht wird. Das Phänomen, dass der Blindwiderstand abhängig von der Frequenz ist, macht man sich bei Hoch- und Tiefpassfilter zu nutzen.

### 9.1.13 LED<sup>18</sup>

LED ist die Kurzform für Light-Emitting-Diode oder zu Deutsch Leuchtdiode. Eine Leuchtdiode funktioniert wie eine normale Diode, gibt jedoch Lichtwellen ab, wenn sie in Durchlassrichtung von Strom durchflossen wird. Welche Wellenlänge von der LED abgegeben wird, hängt dabei von den Mischkristallen ab, aus welchen der Halbleiter besteht.

Eine herkömmliche LED besitzt zwei Anschlussbeine, meist ein kürzeres und ein längeres. Das längere Beinchen ist im Normalfall die Anode, also der Teil welcher an den Pluspol der Versorgungsspannung angeschlossen werden muss, sofern sie in Durchfluss Richtung betrieben werden soll.

Das kürzere Beinchen ist die Kathode, welche an den Minuspol angeschlossen werden muss, wenn die LED in Durchflussrichtung betrieben werden soll. Die Kathode ist auch an der Abflachung am LED-Gehäuse zu erkennen.

Damit der maximal zulässige Strom einer LED nicht überschritten wird, benötigt diese einen Vorwiderstand. Meist liegt Betriebsstrom einer LED bei ca. 20 mA. Für die genaue Berechnung des Vorwiderstandes siehe Punkt *Treiberschaltung LED*. Die nächste Abbildung zeigt den Aufbau einer LED und deren Schaltzeichen. Die folgende Tabelle zeigt die Halbleiterwerkstoffe, welche je nach Farbe der LED zum Einsatz kommen und welche Durchflussspannungen ca. zu erwarten sind. Die genau Betriebswerte einer LED sind im zugehörigen Datenblatt zu finden.

---

<sup>18</sup> In Anlehnung an: <https://de.wikipedia.org/wiki/Leuchtdiode> entnommen am 08.02.2018

## FREETAG LASERTAGSYSTEM

---

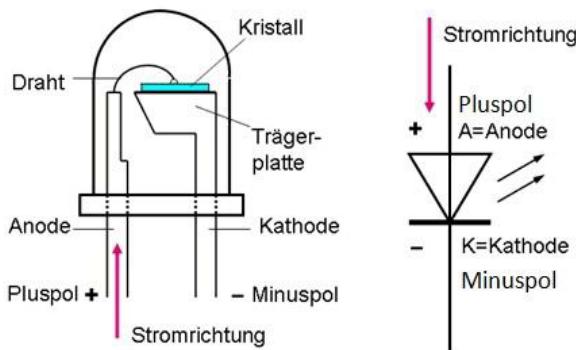


Abbildung 37: LED Darstellung und Schaltsymbol<sup>19</sup>

Tabelle 10: Halbleitermaterialien LED

### Halbleitermaterial der LEDs verschiedener Farben

| Farbe    | Wellenlänge<br>$\lambda$ in nm | Werkstoff   | Spannung<br>$U_F$ |
|----------|--------------------------------|---|-------------------|
| Infrarot | $000 > \lambda > 760$          | <u>Galliumarsenid (GaAs)</u><br><u>Aluminiumgalliumarsenid (AlGaAs)</u>   | 1,2-1,8V          |
| Rot      | $610 < \lambda < 760$          | Aluminiumgalliumarsenid (AlGaAs)<br><u>Galliumarsenidphosphid (GaAsP)</u><br><u>Aluminiumgalliumindiumphosphid (Al-GaInP)</u><br><u>Galliumphosphid (GaP)</u> | 1,6-2,2V          |
| Orange   | $590 < \lambda < 610$          | Galliumarsenidphosphid (GaAsP)<br>Aluminiumgalliumindiumphosphid (Al-GaInP)<br><u>Galliumphosphid (GaP)</u>   | 1,9-2,5V          |
| Gelb     | $570 < \lambda < 590$          | Galliumarsenidphosphid (GaAsP)<br>Aluminiumgalliumindiumphosphid (Al-GaInP)<br><u>Galliumphosphid (GaP)</u>   | 1,9-2,5V          |

<sup>19</sup> Entnommen von: [http://www.fremo-hemsbach.de/LED\\_Symbol.jpg](http://www.fremo-hemsbach.de/LED_Symbol.jpg) am 08.02.2018

## FREETAG LASERTAGSYSTEM

---

|                     |                       |  |          |
|---------------------|-----------------------|--|----------|
| <b>Grün</b>         | $500 < \lambda < 570$ | <u>Indiumgalliumnitrid (InGaN)</u> / <u>Galliumnitrid (GaN)</u><br><u>Galliumphosphid (GaP)</u><br><u>Aluminiumgalliumindiumphosphid (Al-GaInP)</u><br><u>Aluminiumgalliumphosphid (AlGaP)</u><br><u>Zinkoxid (ZnO)</u> , in Entwicklung | 1,9-2,5V |
| <b>Blau</b>         | $450 < \lambda < 500$ | <u>Zinkselenid (ZnSe)</u><br>Indiumgalliumnitrid (InGaN)<br><u>Siliziumkarbid (SiC)</u><br><u>Silizium (Si)</u> als Träger, in Entwicklung<br>Zinkoxid (ZnO), in Entwicklung   | 2,7-3,5V |
| <b>Violett</b>      | $400 < \lambda < 450$ | Indiumgalliumnitrid (InGaN)  | 2,7-3,5V |
| <b>Ultraviolett</b> | $230 < \lambda < 400$ | <u>Aluminiumnitrid (AlN)</u><br><u>Aluminiumgalliumnitrid (AlGaN)</u><br><u>Aluminiumgalliumindiumnitrid (AlGaInN)</u> <sup>[1]</sup><br><u>Diamant (C)</u><br>Hexagonales <u>Bornitrid (BN)</u> <sup>[2]</sup>                          | 3,1-4,5V |

### 9.1.14 Libraries

Libraries, zu Deutsch Bibliotheken, sind eine Ansammlung von Programmcodezeilen, welche die Kommunikation und Verwendung von diversen Bauteilen vereinfacht. Meist werden häufig genutzte Funktionen, welche mehrere Zeilen Programmcode benötigen würden, zusammengefasst und ihnen wird ein einfacher Befehl zugeteilt. Die Arduino-IDE beinhaltet diverse Libraries und es können unzählige weitere im Internet gefunden werden.

Um eine externe Library zu installieren siehe Punkt 6.2.4.

### 9.1.15 Markierer, Tagger oder Phaser

Die Einheit, welche nach Betätigen des Triggers das Infrarotsignal absendet wird in dieser Arbeit als Markierer, Tagger oder Phaser bezeichnet. Markierer bezeichnet eigentlich das Sportgerät, welches die Kugeln beim Paintball verschießt. Die Bezeichnung Phaser stammt noch von früher und wurde aufgrund von Ähnlichkeiten zu gewissen Waffen aus Science-Fiction Filmen gewählt. Tagger kommt vom englischen „to tag“ was übersetzt *markieren* bedeutet.

### 9.1.16 MilesTag 2 Protokoll

Bei diesem Protokoll handelt es sich im weitesten Sinne um ein Regelwerk, welches klarstellt, wie Sender und Empfänger miteinander kommunizieren müssen, damit die Signalübertragung funktioniert. Dieses Protokoll darf für nicht kommerzielle Zwecke verwendet werden. Für genauere Infos siehe Punkt 8.2.

### 9.1.17 Muzzle-LED

Als Muzzle-LED bezeichne ich in dieser Arbeit die orangefarbene LED, welche nach Betätigung des Triggers aufleuchtet und somit eine Art Mündungsfeuer simuliert bzw. den Markierer kurz sichtbar macht, wenn ein Schuss abgegeben wird.

### 9.1.18 N-Kanal MOSFET<sup>20</sup>

MOSFET ist die englische Abkürzung für „metal-oxide-semiconductor field-effect transistor“ zu Deutsch *Metall-Oxid-Halbleiter-Feldeffekttransistor*. MOSFETs können selbstsperrend bzw. selbstleitend sein. Legt man bei einem N-Kanal MOSFET ein Potential bzw. eine Steuerspannung an den Gateanschluss, so beginnt der MOSFET zwischen Drain- und Source-Anschluss zu leiten (beim selbstsperrenden) oder er sperrt (beim selbstleitenden). Den selbstleitenden Typ nennt man auch Verarmungstyp, den selbstsperrenden Anreicherungstyp. Für genauere Infos empfehle ich folgenden Artikel:

<https://de.wikipedia.org/wiki/Metall-Oxid-Halbleiter-Feldeffekttransistor>

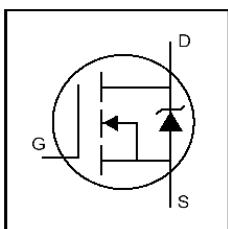


Abbildung 38: Schaltzeichen N-Kanal Mosfet<sup>21</sup>:

### 9.1.19 Open Source

Als Open Source werden Programme bezeichnet, welche über einen offenen Programm- bzw. Sourcecode verfügen und somit von jedem verändert werden kann.

---

<sup>20</sup> In Anlehnung an Wikipedia Beitrag Link: <https://de.wikipedia.org/wiki/Metall-Oxid-Halbleiter-Feldeffekttransistor>

<sup>21</sup> Quelle: <http://www.sprut.de/electronic/switch/nkanal/nkanal.gif> Entnommen am 02.02.2018

### 9.1.20      **Outdoor**

Outdoor kann frei als „draußen“ oder „an der frischen Luft“ übersetzt werden. Im sportlichen Bereich meint man meist, dass das Gelände nicht überdacht ist.

### 9.1.21      **Paintball**

Paintball ist eine Trend- bzw Actionsportart, bei welcher mit Farbstoff gefüllte Gelatinekugeln verschossen werden, welche bei Aufprall zerplatzen und somit dank der Farbe das Objekt „markieren“. Die Kugeln werden mit Druckluft aus einer Druckluftflasche beschleunigt. Im Gegensatz zu Airsoft werden meist keine waffenähnlichen Markierer verwendet und die Spieler tragen Trikots welche an Eishockeytrikots erinnern. Je nach Distanz schmerzen die Paintballkugeln beim Aufprall.

### 9.1.22      **Physical Computing**

Darunter versteht man die Verwendung von Hardware und Software, um mit der Umwelt zu agieren.

### 9.1.23      **Respawn**

Als Respawn bezeichnet man den Wiedereinstieg ins Spiel, nachdem die Lebenspunkte auf 0 gesunken sind. Beim Respawn werden die Lebenspunkte wieder auf den Startwert gesetzt und je nach Einstellung in der Software wird dem Spieler ein Leben abgezogen.

### 9.1.24      **Respawnzeit**

Die Respawnzeit ist der Zeitraum, welcher zwischen dem Unterschreiten von 0 Lebenspunkten und dem erneuten Einstieg in das Spielgeschehen verstreicht. Diese Zeitspanne kann in Minuten in der Software eingestellt werden.

### 9.1.25      **Sender**

In diesem Projekt sind alle Einheiten, im speziellen aber Tagger, gemeint welche Infrarotsignale abgeben.

### 9.1.26      **Sensorkopf**

Als Sensorkopf bezeichne ich die Empfängerschaltung, welche in dem Sensorgehäuse verbaut wurde siehe Abbildung 29

### 9.1.27 Sketch

Als „Sketch“ wird der Programmcode bezeichnet der von der Arduino-Programmierumgebung verwendet wird, also die Software welche man auf die Arduino-Plattform hochlädt.

### 9.1.28 Spielerpaket

Das Spielerpaket ist die Summe der Infrarotsignale, welche ein Markierer beim Betätigen des Abzuges abgegeben wird. Es setzt sich aus dem Header, der Spielernummer, der Teamnummer und dem Schadenscode zusammen. Für genauere Infos siehe Punkt 8.2.

### 9.1.29 Trigger

Ist eine Nachbildung des Abzuges einer reellen Waffe. Wird dieser betätigt, so wird ein Infrarotsignal vom Markierer abgegeben.

### 9.1.30 Widerstand (Bauteil)

Bei einem Widerstand handelt es sich um ein Bauelement, welches dazu dient einen ohm'schen Widerstand in einer elektrischen Schaltung zu erzeugen. In diesem Projekt werden Widerstände zum Begrenzen des elektrischen Stromes und zur Aufteilung der Spannungen verwendet.

Widerstände können aus unterschiedlichen Materialien bestehen, meist werden in der Elektronik jedoch Kohle- und Metallschichtwiderstände verwendet.

Wichtige Kennwerte sind neben dem Widerstandswert auch noch die maximale Verlustleistung sowie die Fertigungstoleranz. Bei kleinen Widerständen werden diese Werte nicht als Ziffern, sondern als Farbcodierung aufgedruckt (*siehe Abbildung 39 und 40*). Die maximale Leistung, die an einem Widerstand abfallen darf, muss aus dem Datenblatt entnommen werden. Als Faustregel kann man jedoch sagen, dass Kohleschichtwiderstände mit einer Leistung von 0,25W und Metallschichtwiderstände mit einer Leistung von 0,5W belastet werden können. Die Leistung, welche über einen Widerstand geleitet wird, kann mit folgender Formel berechnet werden:

$$P = \frac{U^2}{R}$$

P...Leistungsverlust am Widerstand in Watt

U...Spannungsabfall am Widerstand in Volt

R...Widerstandswert in Ohm

## FREETAG LASERTAGSYSTEM

---

Farbkodierung von Widerständen mit 4 Ringen

| Farbe   |   | Widerstandswert in $\Omega$ |                    |                            | Toleranz     |
|---------|---|-----------------------------|--------------------|----------------------------|--------------|
|         |   | 1. Ring<br>(Zehner)         | 2. Ring<br>(Einer) | 3. Ring<br>(Multiplikator) |              |
| „keine“ | X | —                           | —                  | —                          | $\pm 20\%$   |
| silber  |   | —                           | —                  | $10^{-2} = 0,01$           | $\pm 10\%$   |
| gold    |   | —                           | —                  | $10^{-1} = 0,1$            | $\pm 5\%$    |
| schwarz |   | —                           | 0                  | $10^0 = 1$                 | —            |
| braun   |   | 1                           | 1                  | $10^1 = 10$                | $\pm 1\%$    |
| rot     |   | 2                           | 2                  | $10^2 = 100$               | $\pm 2\%$    |
| orange  |   | 3                           | 3                  | $10^3 = 1.000$             | —            |
| gelb    |   | 4                           | 4                  | $10^4 = 10.000$            | —            |
| grün    |   | 5                           | 5                  | $10^5 = 100.000$           | $\pm 0,5\%$  |
| blau    |   | 6                           | 6                  | $10^6 = 1.000.000$         | $\pm 0,25\%$ |
| violett |   | 7                           | 7                  | $10^7 = 10.000.000$        | $\pm 0,1\%$  |
| grau    |   | 8                           | 8                  | $10^8 = 100.000.000$       | $\pm 0,05\%$ |
| weiß    |   | 9                           | 9                  | $10^9 = 1.000.000.000$     | —            |

Abbildung 39: Farocode Widerstand 1<sup>22</sup>

Farbkodierung von Widerständen mit 5 oder 6 Ringen

| Farbe   | 1. Ring<br>(Hunderter) | 2. Ring<br>(Zehner) | 3. Ring<br>(Einer) | 4. Ring<br>(Multiplikator) | 5. Ring<br>(Toleranz) | 6. Ring<br>(Temp.-Koeffizient) |
|---------|------------------------|---------------------|--------------------|----------------------------|-----------------------|--------------------------------|
| silber  |                        |                     |                    | $10^{-2}$                  |                       |                                |
| gold    |                        |                     |                    | $10^{-1}$                  |                       |                                |
| schwarz |                        | 0                   | 0                  | $10^0$                     |                       | $200 \cdot 10^{-6} K^{-1}$     |
| braun   | 1                      | 1                   | 1                  | $10^1$                     | $\pm 1\%$             | $100 \cdot 10^{-6} K^{-1}$     |
| rot     | 2                      | 2                   | 2                  | $10^2$                     | $\pm 2\%$             | $50 \cdot 10^{-6} K^{-1}$      |
| orange  | 3                      | 3                   | 3                  | $10^3$                     |                       | $15 \cdot 10^{-6} K^{-1}$      |
| gelb    | 4                      | 4                   | 4                  | $10^4$                     |                       | $25 \cdot 10^{-6} K^{-1}$      |
| grün    | 5                      | 5                   | 5                  | $10^5$                     | $\pm 0,5\%$           |                                |
| blau    | 6                      | 6                   | 6                  | $10^6$                     | $\pm 0,25\%$          | $10 \cdot 10^{-6} K^{-1}$      |
| violett | 7                      | 7                   | 7                  |                            | $\pm 0,1\%$           | $5 \cdot 10^{-6} K^{-1}$       |
| grau    | 8                      | 8                   | 8                  |                            | $\pm 0,05\%$          |                                |
| weiß    | 9                      | 9                   | 9                  |                            |                       |                                |

Abbildung 40: Farocode Widerstand 2<sup>23</sup>

<sup>22</sup> Quelle: [https://de.wikipedia.org/wiki/Widerstand\\_\(Bauelement\)](https://de.wikipedia.org/wiki/Widerstand_(Bauelement)) entnommen am 08.02.2018

<sup>23</sup> Quelle: [https://de.wikipedia.org/wiki/Widerstand\\_\(Bauelement\)](https://de.wikipedia.org/wiki/Widerstand_(Bauelement)) entnommen am 08.02.2018

## FREETAG LASERTAGSYSTEM

### 10 ELEKTROPLÄNE UND TEILELISTE

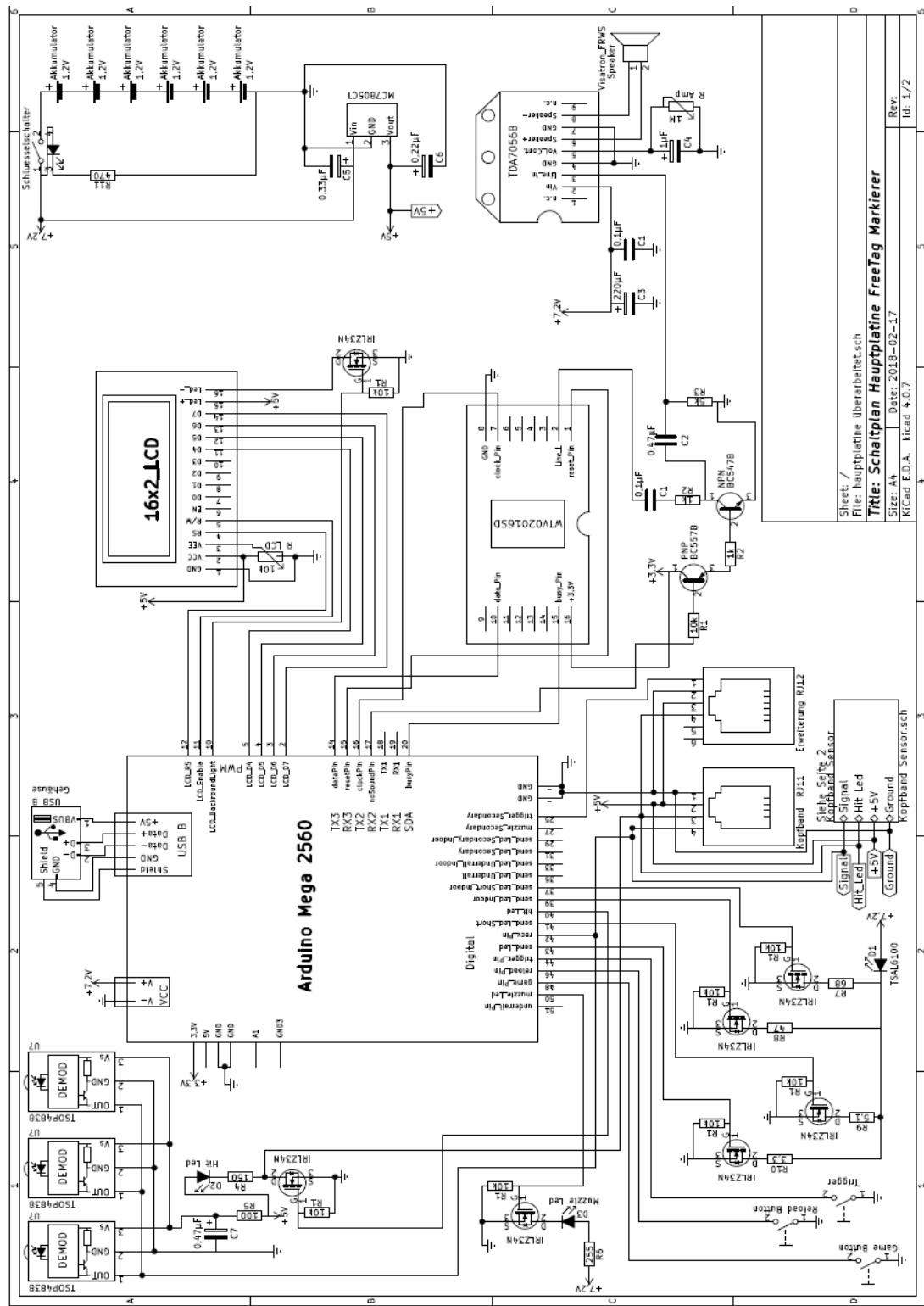


Abbildung 41: Schaltplan Hauptplatine

## FREETAG LASERTAGSYSTEM

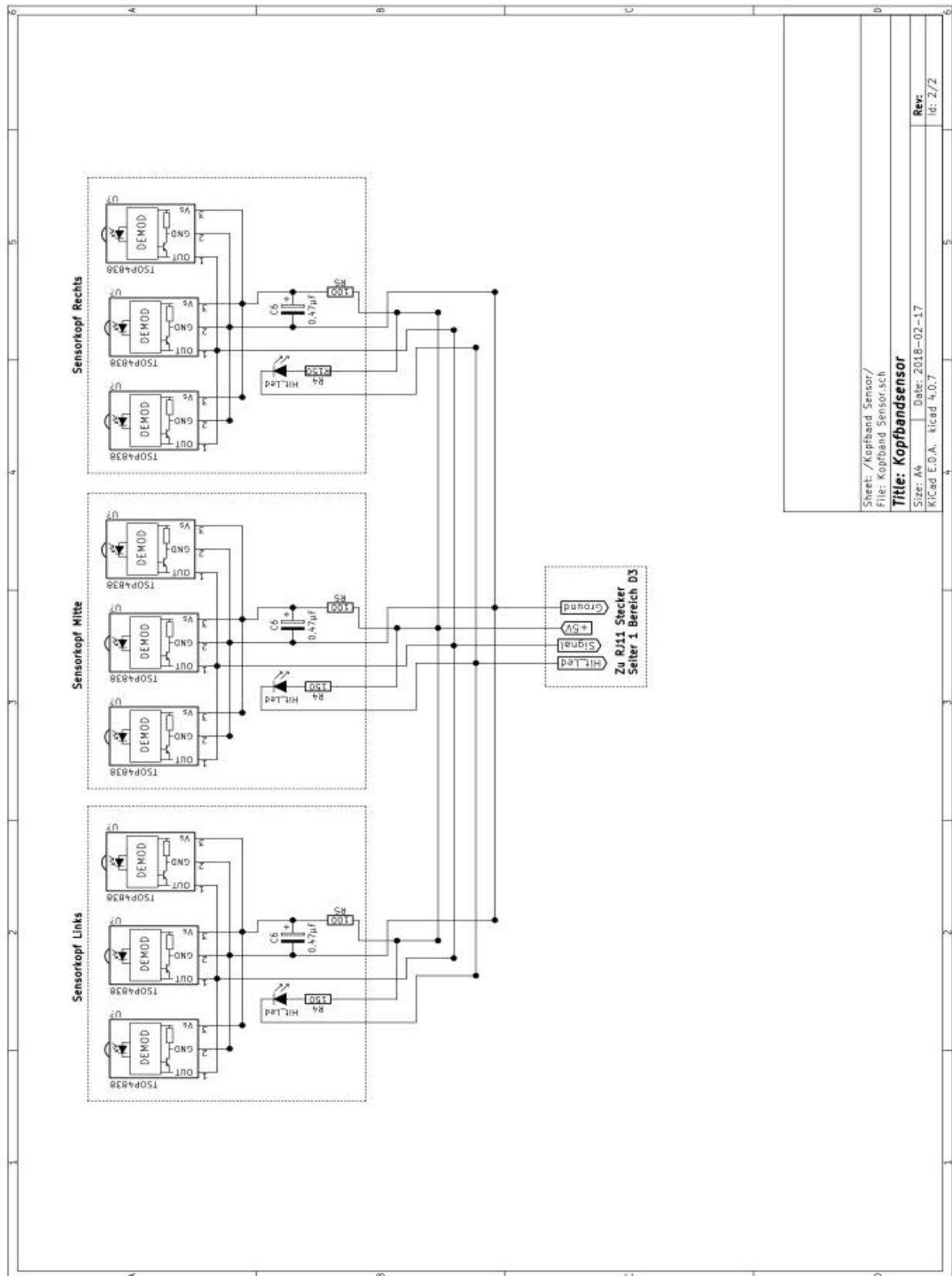


Abbildung 42: Schaltplan Kopfbandsensoren

## 10.1 Teileliste

Tabelle 11: Teileliste Elektronik

| <b>Nummer</b> | <b>Bauteilbezeichnung</b>             | <b>Wert</b> | <b>Schaltungen</b>               | <b>Anzahl</b> |
|---------------|---------------------------------------|-------------|----------------------------------|---------------|
| R1            | Metallschichtwiderstand<br>0,6 W 1 %  | 10 kΩ       | Led Treiber/No Sound             | 8             |
| R2            | Metallschichtwiderstand<br>0,6 W 1 %  | 1 kΩ        | No Sound/Audio-Verstärker        | 2             |
| R3            | Metallschichtwiderstand<br>0,6 W 1 %  | 5 kΩ        | Audio-Verstärker                 | 1             |
| R4            | Metallschichtwiderstand<br>0,6 W 1 %  | 150 Ω       | Vorwiderstand Hit-LED            | 4             |
| R5            | Metallschichtwiderstand<br>0,6 W 1 %  | 100 Ω       | Sensorkopf                       | 4             |
| R6            | Metallschichtwiderstand<br>0,6 W 1 %  | 255 Ω       | R Muzzle-LED                     | 1             |
| R7            | Metallschicht Widerstand<br>1 W 5 %   | 67 Ω        | Send LED kurz Indoor             | 1             |
| R8            | Metallschicht Widerstand<br>1 W 5 %   | 47 Ω        | Send LED Indoor                  | 1             |
| R9            | 5 W Drahtwiderstand                   | 5,1 Ω       | Send LED kurz Outdoor            | 1             |
| R10           | 7 W Drahtwiderstand                   | 3,3 Ω       | Send LED Outdoor                 | 1             |
| R11           | Metallschichtwiderstand<br>0,6 W 1 %  | 470 Ω       | R Schalter LED                   | 1             |
| RLcd          | Cermet Trimmer                        | 10 kΩ       | LCD Kontrast                     | 1             |
| Ramp          | Einstellpotentiometer 9<br>mm liegend | 1 MΩ        | Lautstärke Audio-Verstär-<br>ker | 1             |
| C1            | Kondensator Kerko 20 %                | 0,1 µF      | Audio-Verstärker/<br>WTV020SD16p | 2             |
| C2            | Vielschichtk. 470 nF RM5              | 0,47<br>µF  | Audio-Verstärker                 | 1             |
| C3            | Elektrolytkondensator                 | 220<br>µF   | Audio-Verstärker                 | 1             |
| C4            | Elektrolytkondensator 1,5<br>mm       | 1 µF        | Audio-Verstärker                 | 1             |
| C5            | ELKO RAD. 105 °C 0,33<br>µF 50 V      | 0,33<br>µF  | 5 V Spannungsregler              | 1             |
| C6            | ELKO RAD. 105 °C 0,22<br>µF 50 V      | 0,22<br>µF  | 5 V Spannungsregler              | 1             |
| C7            | Subminiatur-Elko 63 V<br>20 %         | 0,47<br>µF  | Sensorkopf                       | 4             |
| D1            | IR Diode TSAL 6100                    |             | Send LED                         | 1             |

## FREETAG LASERTAGSYSTEM

---

|    |  |   |    |
|----|--|---|----|
| D2 | LED 5 mm super-rot                         | Hit-LED   | 4  |
| D3 | LED 5 mm super-gelb                        | Muzzle-LED  | 1  |
|    |  |   |    |
|    | TSOP 4838                                  | IR Empfänger  | 12 |
|    | TDA7056B                                   | Audio Verstärker                                      | 1  |
|    | WTV020SD16P                                | Soundmodul  | 1  |
|    | Arduino Mega 2560                          | Mikrocontroller                                       | 1  |
|    | 16x2 LCD Modul                             | 16x2 LCD Bildschirm                                   | 1  |
|    | Visatron FRWS 5-8                          | Audio-Verstärker                                      | 1  |
|    | MC7805CT                                   | Spannungswandler +5 V                                 | 1  |
|    | IRLZ34N                                    | LED-Treiberschaltung                                  | 7  |
|    | PNP BC557B                                 | Audio-Schaltung                                       | 1  |
|    | NPN 547B                                   | Audio-Schaltung                                       | 1  |
|    |  |   |    |
|    | Printstecker gerade 2 polig PSS 254/2G     | USB-Verbindung  | 1  |
|    | Kupplungs-Leergehäuse 2 polig PSK 254/2W   | USB-Verbindung  | 1  |
|    | Printstecker gerade 3 polig PSS 254/3G     | USB-Verbindung  | 1  |
|    | Kupplungs-Leergehäuse 3 polig PSK 254/3W   | USB-Verbindung  | 1  |
|    | Printstecker gerade 4 polig PSS 254/4G     | Verbindung Sensorkopf zu Hauptplatine                 | 1  |
|    | Kupplungs-Leergehäuse 4 polig PSK 254/4W   | Verbindung Sensorkopf zu Hauptplatine                 | 1  |
|    | Printstecker gerade 6 polig PSS 254/6G     | Verbindung IR-LED und Muzzle-LED zu Hauptplatine      | 1  |
|    | Kupplungs-Leergehäuse 6 polig PSK 254/6W   | Verbindung IR-LED und Muzzle-LED zu Hauptplatine      | 1  |
|    | Printstecker gerade 8 polig PSS 254/8G     | Verbindung Taster zu Hauptplatine                     | 1  |
|    | Kupplungs-Leergehäuse 8 polig PSK 254/8W   | Verbindung Taster zu Hauptplatine                     | 1  |
|    | Printstecker gerade 10 polig PSS 254/10G   | 16x2 LCD zu Platine                                   | 1  |
|    | Kupplungs-Leergehäuse 10 polig PSK 254/10W | 16x2 LCD zu Platine                                   | 1  |
|    | Crimpkontakte für psk 254 20 Stück         | Kupplungs-Leergehäuse PSK                             | 2  |
|    | Tamiya Steckverbindung Buchse              | Spannungsversorgung Schlüsselschalter zu Hauptplatine | 1  |

## FREETAG LASERTAGSYSTEM

---

|                                      |  |   |   |
|--------------------------------------|--|---|---|
| Tamiya Steckverbindung<br>Stecker    |  | Spannungsversorgung<br>Schlüsselschalter zu Haupt-<br>platine | 1 |
|                                      |  |   |   |
| Hauptschaltungsplatine               |  |   | 1 |
| Platine Sensorkopf                   |  |   | 4 |
|                                      |  |   |   |
| Panasonic AV 32543 AT                |  | Trigger-Schalter  | 1 |
| Elewind 22 mm illum. key<br>schwitzh |  | Schlüsselschalter   | 1 |
| Elewind button (PM191F-<br>10/A)     |  | Game Button   | 1 |
| Elewind 19 mm (PM191B-<br>10/J/A)    |  | Reload Button   | 1 |
| RJ 12 Einbaubuchse                   |  |   | 1 |
| RJ 10 Einbaubuchse                   |  |   | 1 |
| Telefonkabel Spiral 4 m              |  | Verbindung Markierer Kopf-<br>bandsensoren                    | 1 |

## 10.2 PCB Layouts

Die hier gezeigten Layouts sind lediglich Screenshots und dienen nicht als Vorlage, die Layoutdateien können als .pdf, Gerber-file und als Arbeitsdatei für KiCad von <https://github.com/EmmanuelPanholzer/FreeTag-Lasertag> im Ordner PCB- Layout heruntergeladen werden.

Bei dem Layout für die Sensorköpfe ist darauf zu achten, dass es sich um eine zweischichtige Platine handelt, man sollte daher bei der Herstellung der Platine speziell darauf achten, dass die Ober- und Unterseite der Platine zusammenpassen.

Am PCB-Layout der Hauptplatine ist zu erkennen, dass viele Leiterbahnen zwar bereits vorgesehen sind, aber nicht im Schaltplan eingezeichnet sind. Dies röhrt daher, dass ich bereits einige Ideen für spätere Erweiterungen und Weiterentwicklungen für dieses Projekt auf der Hauptplatine verwirklicht habe. Ich habe diese Teile jedoch nicht in den Hauptschaltplan eingezeichnet, weil ich die Werte der Bauteile erst festlegen muss und die dementsprechenden Funktionen noch nicht in die Software eingeflossen sind.

Ich hätte für dieses Projekt natürlich auch eine Platine mit den Grundfunktionen, welche den Anforderungen dieses Projektes entsprechen, entwerfen können, ich hätte dann jedoch für die Weiterentwicklung der Schaltung eine neue Platine herstellen müssen, was mich bei der weiteren Arbeit nur behindert hätte.

## FREETAG LASERTAGSYSTEM

### 10.2.1 Layout Hauptplatine Markierer

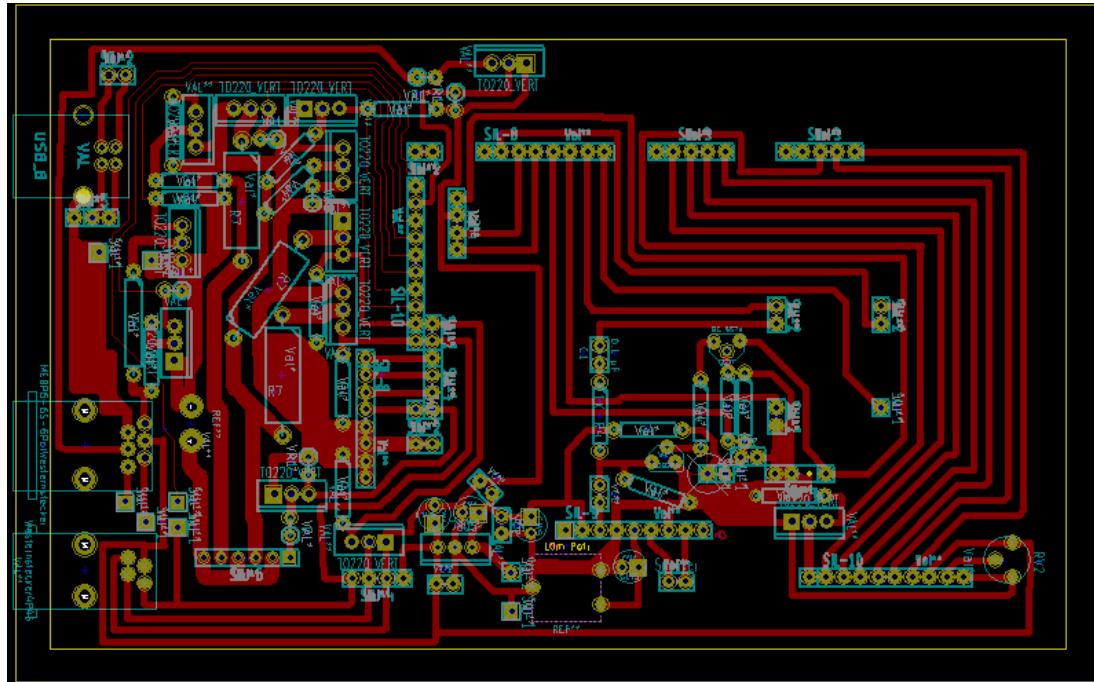


Abbildung 43: PCB Layout Hauptplatine

### 10.2.2 Layout Platine Sensorkopf

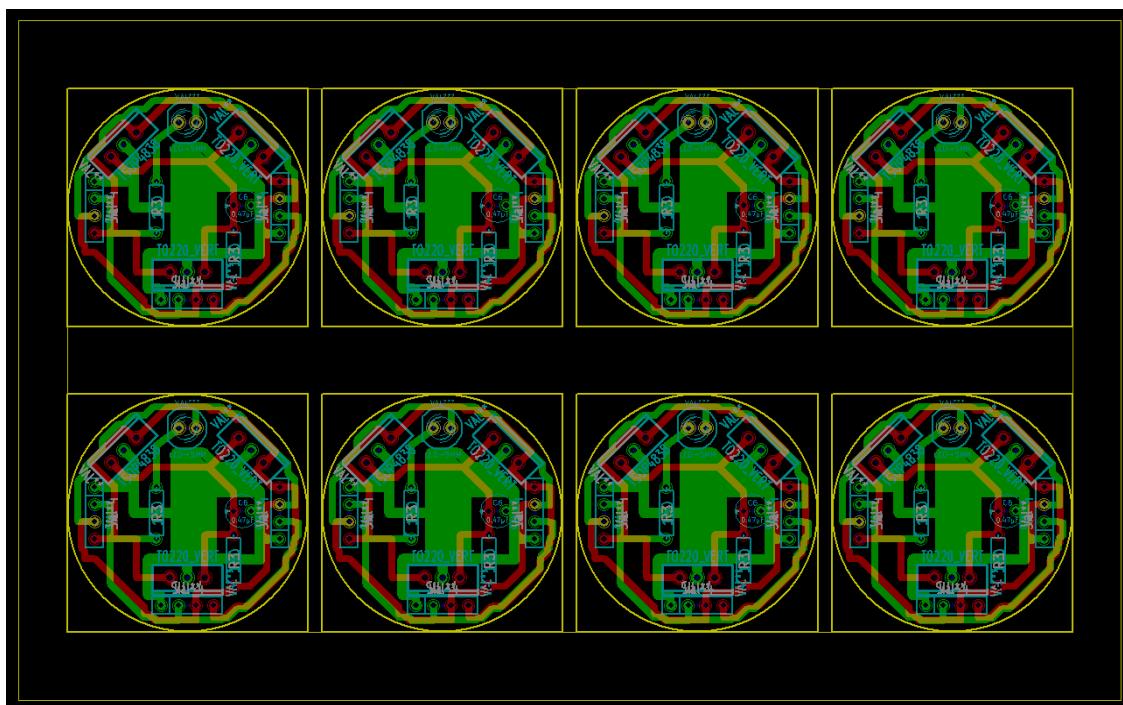


Abbildung 44: PCB Layout Hauptplatine

## FREETAG LASERTAGSYSTEM

---

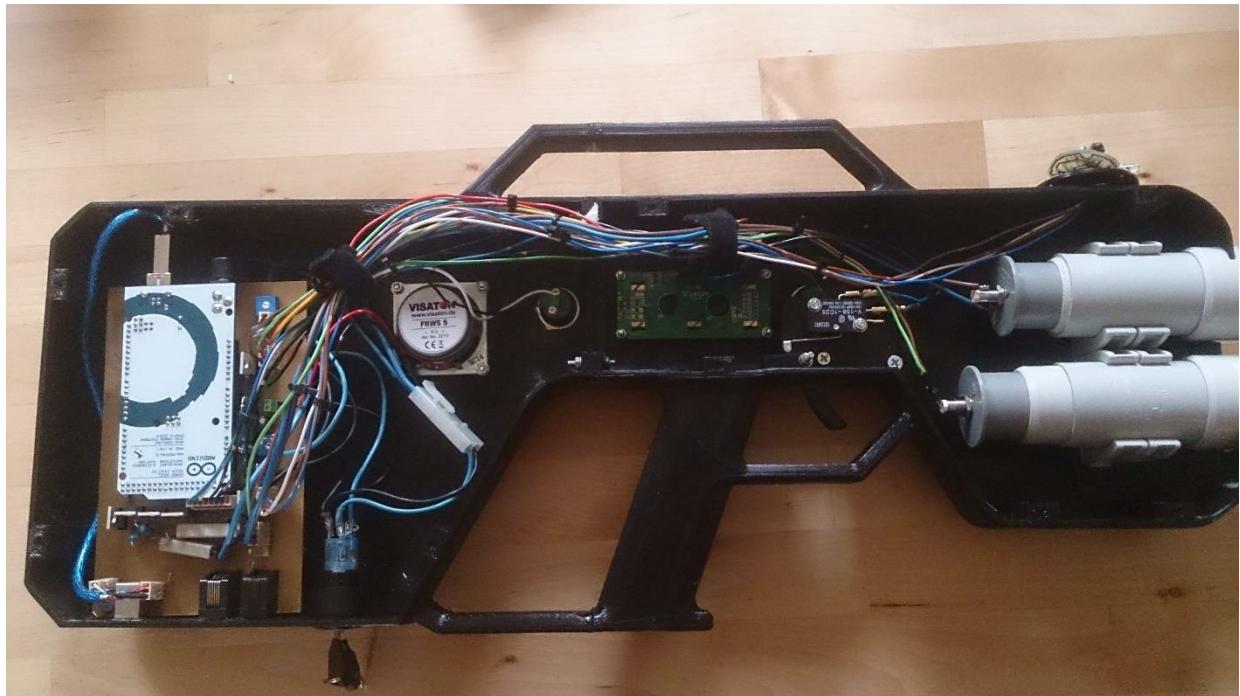


Abbildung 45: Markierer verkabelt

## 11 MECHANISCHER PLAN UND TEILELISTE

Der folgende Plan dient nur zur Veranschaulichung. Die für den 3D-Druck notwendigen STL-Dateien können unter <https://github.com/EmmanuelPanholzer/FreeTag-Lasertag> im Ordner Gehäuse heruntergeladen werden. Das Gehäuse wurde dabei in vier Teile geteilt um mit einem 3D-Drucker mit einer Druckbettgröße von 300x300x300mm gedruckt werden zu können. Das Design des Gehäuses stammt von mir, konstruiert und gedruckt wurde es von Rene Reisinger wofür ihn mein großer Dank gebührt.

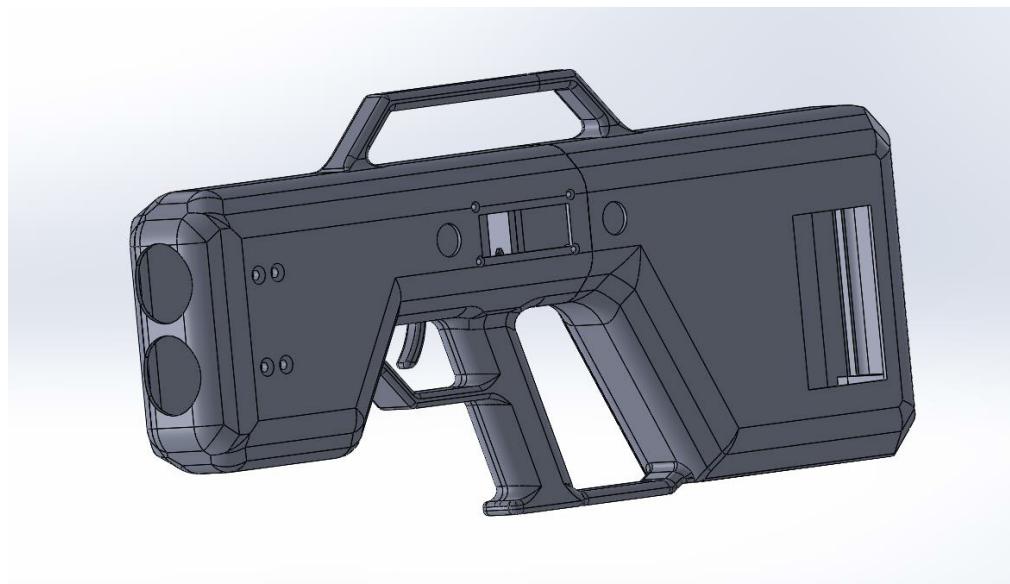


Abbildung 46: 3D-Darstellung des Gehäuses

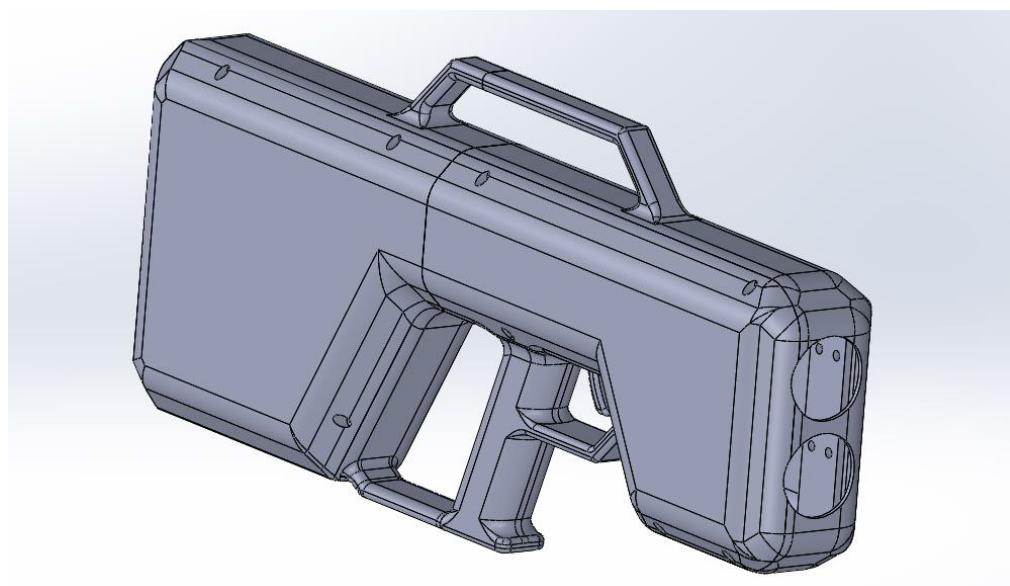


Abbildung 47: 3D-Darstellung Gehäuse 2

## FREETAG LASERTAGSYSTEM

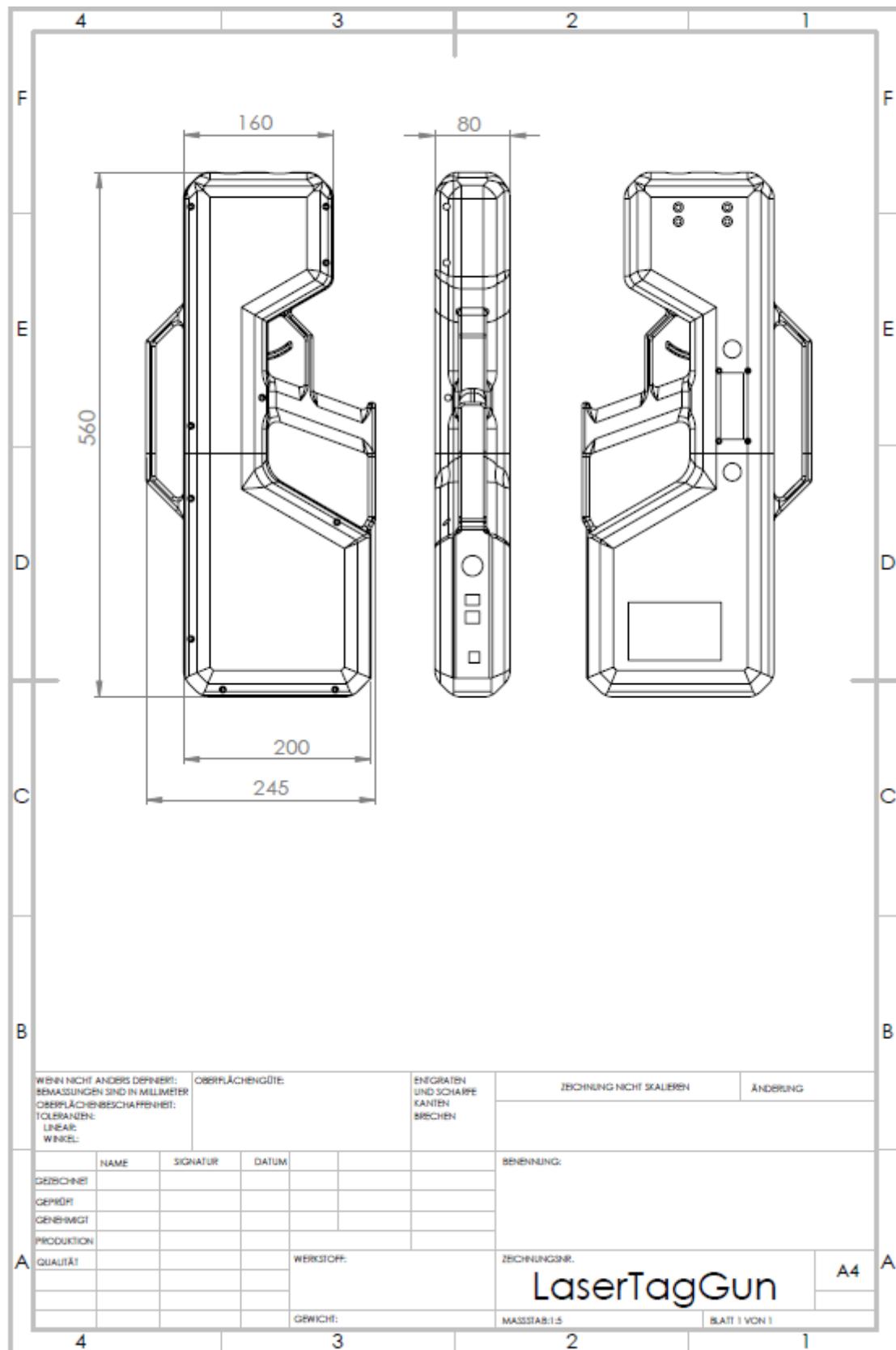


Abbildung 48: Mechanische Skizze Gehäuse

## FREETAG LASERTAGSYSTEM

---



Abbildung 49: Gehäuse nach dem Verkleben außen



Abbildung 50: Gehäuse nach dem Verkleben innen

## 11.1 Mechanische Bauteilliste

Tabelle 12: Mechanische Bauteilliste

| Bezeichnung                    | Verwendung   | Stückzahl |
|--------------------------------|--|-----------|
| Gehäusedeckel links            | Gehäusedeckel linke Seite  | 1         |
| Gehäusedeckel rechts           | Gehäusedeckel rechte Seite   | 1         |
| Gehäuseunterteil links         | Gehäuseunterteil linke Seite   | 1         |
| Gehäuseunterteil rechts        | Gehäuseunterteil rechte Seite  | 1         |
| Abzugsgrundplatte              | Montageplatte für Abzug  | 1         |
| Abzug                          | Abzug  | 1         |
| Senkkopfschraube M3x10 DIN 965 | Verbindung Gehäuse, Tastermontage, Sensorskopfmontage                | 20        |
| Senkkopfschraube M3x20 DIN 965 | Montage Abzug  | 1         |
| Sechskantmutter M3 DIN 934     | Verbindung Gehäuse, Tastermontage, Sensorskopfmontage, Montage Abzug | 22        |
| Beilagscheibe d=3,2mm          | Montage Abzug  | 2         |
| Senkkopfschraube M5x20 DIN 965 | Montage Rohrhalterungen  | 4         |
| Sechskantmutter M5 DIN 934     | Montage Klemmschellen  | 4         |
| Bohrschoraube 3,5x13 DIN7504   | Montage Abzugsplatte, Montage Platine                                | 4         |
| Klemmschelle M40               | Montage IR-Optik, Montage Muzzle-LED                                 | 4         |
| Steckmuffe M40                 | IR-Optik, Muzzle-LED   | 2         |
| Steckmuffe M32                 | IR-Optik, Muzzle-LED   | 2         |
| Aufputzrohr M40 100mm          | IR-Optik, Muzzle-LED   | 2         |
| HT Muffenstopfen DN32          | IR-Optik Halterung, Muzzle-LED Halterung                             | 2         |

## FREETAG LASERTAGSYSTEM

---

### 12 MARKIERER MONTIERT



## 13 KALKULATIONEN

### 13.1 Vorkalkulation

|  |                      |
|--|----------------------|
| Mechanische Bauteile (Markierergehäuse, Linsenhalterung) | 200 €                |
| Elektrische Bauteile (Arduino, Kondensatoren, LEDs usw.) | 200 €                |
| Gemeinkosten (Strom, Büromaterialien, Fertigungskosten)  | 100 €                |
| Stundenlohn Arbeiter von 25 € bei 300 Stunden            | 7500 €               |
| <b>Summe</b>   | <b><u>8000 €</u></b> |

### 13.2 Nachkalkulation

#### 13.2.1 Planung Elektronik

|   |                  |
|---|------------------|
| Gemeinkosten (Computernutzung, Schreibwaren usw.) | 30 €             |
| Arbeiterlohn bei 25 €/h                           | 60 h      1500 € |

#### 13.2.2

#### Planung Mechanik

|   |                 |
|---|-----------------|
| Gemeinkosten (Computernutzung, Schreibwaren usw.) | 15 €            |
| Arbeiterlohn bei 25 €/h                           | 28 h      700 € |

#### 13.2.3

#### Fertigung Elektronik

|   |                 |
|---|-----------------|
| Elektrische Bauteile (inkl. Versandkosten)  | 120 €           |
| Gemeinkosten (Ätzchemikalien, Lötzinn usw.) | 30 €            |
| Arbeiterlohn bei 25 €/h                     | 12 h      300 € |

#### 13.2.4

#### Fertigung Mechanik

|  |                |
|--|----------------|
| Mechanische Bauteile                     | 100 €          |
| Gemeinkosten (Lack, Schleifpapier, usw.) | 200 €          |
| Arbeiterlohn bei 25 €/h                  | 8 h      200 € |

## FREETAG LASERTAGSYSTEM

---

### 13.2.5 Programmierung

|  |        |
|--|--------|
| Gemeinkosten (Computernutzung, Schreibwaren, Litaratur usw.) | 60 €   |
| Arbeiterlohn bei 25 €/h                                      | 80 h   |
|  | 2000 € |

### 13.2.6 Tests

|                         |     |       |
|-------------------------|-----|-------|
| Arbeiterlohn bei 25 €/h | 4 h | 100 € |
|-------------------------|-----|-------|

### 13.2.7 Dokumentation

|  |       |
|--|-------|
| Gemeinkosten (Computernützung, Schreibwaren, usw...) | 100 € |
|--|-------|

|                         |      |        |
|-------------------------|------|--------|
| Arbeiterlohn bei 25 €/h | 80 h | 2000 € |
|-------------------------|------|--------|

|              |                      |
|--------------|----------------------|
| <b>Summe</b> | <b><u>5925 €</u></b> |
|--------------|----------------------|

## 14 CE-ZERTIFIZIERUNG<sup>24</sup>

### 14.1.1 Bedeutung

Die CE-Kennzeichnung wurde 1985 eingeführt um die Gleichheit der vor allem sicherheitstechnischen Produktanforderungen in Europa aller Hersteller sicherzustellen.

Der Hersteller bringt durch die CE-Kennzeichnung an seinen Produkten den zuständigen Behörden in Eigenverantwortung zum Ausdruck, dass sein Produkt den einschlägigen Rechtsvorschriften und technischen Spezifikationen entspricht.

Das CE-Kennzeichen ist als solches als Marktzulassungszeichen und nicht als Herkunfts-, Qualitäts-, Güte- oder Normkennzeichen anzusehen.

### 14.1.2 Kennzeichnung

Alle Produkte die für den europäischen Markt bestimmt sind, sind verpflichtend mit einem CE-Kennzeichen zu versehen.

Zumindest sofern diese in mindestens eine CE-Richtlinie fallen.

Würden hier mehr CE-Richtlinien auf ein Produkt fallen, sind sämtliche einzuhalten.

Sollte ein Produkt in keine CE-Richtlinie fallen, ist das Anbringen des CE-Kennzeichens unzulässig.

### 14.1.3 Zuständigkeit

Grundsätzlich ist in erster Linie der Hersteller zuständig, des Weiteren kann dies auch sein Bevollmächtigter in der EU bzw. im EWR (mit Norwegen, Island, Liechtenstein) zutreffen. In manchen Fällen ist auch der Erstimporteur zuständig, der in die EU importiert (je nach Richtlinie). Der Hersteller muss dafür nicht im EU-Raum ansässig sein. Falls eine notifizierte Stelle einzuschalten ist, muss diese meist ihren Sitz innerhalb des EU/EWR-Raumes haben.

Für die Richtigkeit der CE-Kennzeichnung ist der jeweilige Inverkehrbringer in einem Mitgliedstaat verantwortlich und somit unmittelbar haftbar.

---

<sup>24</sup> Das komplette Kapitel samt aller Unterkapitel wurden erstellt in Anlehnung an: <https://www.wko.at/service/innovation-technologie-digitalisierung/ce-kennzeichnung-normen.html> Entnommen am 07.02.2018

#### 14.1.4 Vorgangsweise

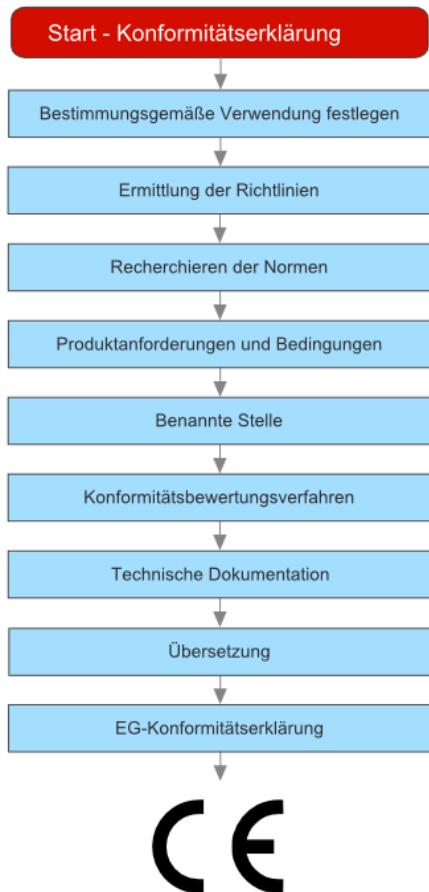


Abbildung 51: CE-Vorgangsweise<sup>25</sup>

#### 14.1.5 Anbringung

Das CE-Kennzeichen ist prinzipiell auf dem Produkt selbst, oder auf einem Schild auf dem Produkt anzubringen.

Sollte dies nicht möglich oder sinnvoll sein, ist das CE-Kennzeichen auf der Verpackung und den Begleitunterlagen anzubringen.

---

<sup>25</sup> <http://www.maschinen-sicherheit.net/01-bilder/konformitaetserklaerung-erstellen-02.PNG> entnommen am 07.02.2018

## FREETAG LASERTAGSYSTEM

---

Das CE-Kennzeichen muss gut lesbar, dauerhaft und sichtbar angebracht werden.  
Die Proportionen sind bei einer Vergrößerung oder Verkleinerung beizubehalten.  
Die Mindestgröße beträgt 5 mm.

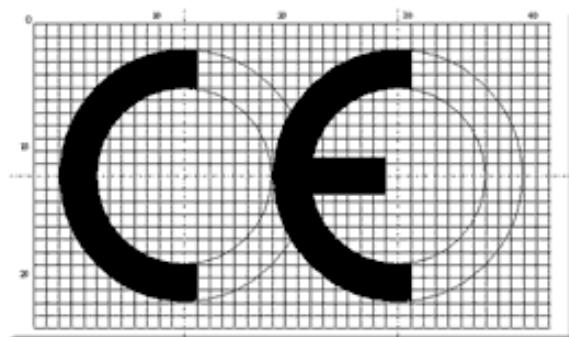


Abbildung 52: CE-Zeichen<sup>26</sup>

### 14.2.6-Schritte zur CE-Kennzeichnung

#### 14.2.1 SCHRITT 1 – Finden Sie heraus, welche Richtlinie(n) und harmonisierte Normen für Ihr Produkt gelten

Finden Sie aus den über 20 Richtlinien die für Sie entsprechenden heraus.  
Die harmonisierten Normen werden in Bezug auf die entsprechenden Richtlinien festgelegt.

#### 14.2.2 SCHRITT 2 – Ermitteln Sie die spezifischen Bedingungen für Ihr Produkt

Sollte ein Produkt allen harmonisierten Normen entsprechen, wird dem Hersteller die Vermutung der Konformität bescheinigt.

#### 14.2.3 SCHRITT 3 – Stellen Sie fest, ob eine benannte Stelle für ein Konformitätsbewertungsverfahren herangezogen werden muss

Dies stellen Sie aufgrund der Inhalte ihrer Richtlinien fest.

---

<sup>26</sup> Entnommen von: [https://ec.europa.eu/growth/single-market/ce-marking\\_en](https://ec.europa.eu/growth/single-market/ce-marking_en) am 07.02.2018

#### **14.2.4 SCHRITT 4 – Testen Sie das Produkt und überprüfen Sie seine Konformität**

Hier ist der Hersteller dafür verantwortlich, eine Risikobewertung durchzuführen.

#### **14.2.5 SCHRITT 5 – Stellen Sie die notwendige technische Dokumentation zusammen und halten Sie sie zur Einsicht bereit**

Sämtliche Dokumente müssen nun der zuständigen Behörde auf Anfrage vorgelegt werden.

#### **14.2.6 SCHRITT 6 – Bringen Sie die CE-Kennzeichnung an Ihrem Produkt an und verfassen Sie die EU-Konformitätserklärung**

Der Hersteller kennzeichnet nun das Produkt mit einem CE-Kennzeichen und hat eine Konformitätserklärung zu verfassen, zu unterzeichnen und somit zu bestätigen.

### **14.3 Richtlinien und harmonisierte Normen für dieses Projekt**

#### **14.3.1 Beschränkung gefährlicher Stoffe in Elektro- und Elektronikgeräten**

|                   |   |
|-------------------|---|
| <b>Richtlinie</b> | <b>2011/65/EU</b><br>des Europäischen Parlaments und des Rates vom 8.6.2011 zur Beschränkung der Verwendung bestimmter gefährlicher Stoffe in Elektro- und Elektronikgeräten (ABl. Nr. L 174 vom 1.7.2011) - Neufassung<br><br>zuletzt geändert durch die delegierte Richtlinie 2014/76/EU (ABl. Nr. L 148 vom 20.5.2014) |
| <b>gilt für</b>   | folgende Elektro- und Elektronikgeräte für den Betrieb mit Wechselstrom von höchstens 1000 V bzw. Gleichstrom bis höchstens 1500 V:<br><br>Haushaltsgroßgeräte<br><br>Haushaltskleingeräte<br><br>IT- und Telekommunikationsgeräte<br><br>Geräte der Unterhaltungselektronik  |

## FREETAG LASERTAGSYSTEM

---

|                       |   |
|-----------------------|---|
|                       | <p>Beleuchtungskörper<br/>elektrische und elektronische Werkzeuge<br/>Spielzeug sowie Sport- und Freizeitgeräte<br/><u>medizinische Geräte - ab 22.7.2014</u><br/><u>In-vitro-Diagnostika - ab 22.7.2016</u><br/>Überwachungs- und Kontrollinstrumente (nicht industriell) - <i>ab 22.7.2014</i><br/>Überwachungs- und Kontrollinstrumente für industrielle und gewerbliche Zwecke - <i>ab 22.7.2017</i><br/>automatische Ausgabegeräte<br/>sonstige Elektro- und Elektronikgeräte - <i>ab 22.7.2019</i></p>  |
| <b>gilt nicht für</b> | <p>Geräte zum Schutz der wesentlichen Sicherheitsinteressen von Mitgliedstaaten oder Geräte für den Einsatz im Weltraum<br/>Geräte, die speziell als Teil eines anderen, von dieser Richtlinie ausgenommenen oder nicht in den Geltungsbereich dieser Richtlinie fallenden Gerätetyps konzipiert sind und als ein solches Teil installiert werden sollen, die ihre Funktion nur als Teil dieses Geräts erfüllen können und die nur durch gleiche, speziell konzipierte Geräte ersetzt werden können<br/>ortsfeste industrielle Großwerkzeuge<br/>ortsfeste Großanlagen<br/>Verkehrsmittel zur Personen- oder Güterbeförderung mit Ausnahme von elektrischen Zweirad-Fahrzeugen, die nicht typgenehmigt sind<br/>bewegliche Maschinen, die nicht für den Straßenverkehr bestimmt sind und ausschließlich zur professionellen Nutzung zur Verfügung gestellt werden<br/><u>aktive implantierbare medizinische Geräte</u><br/>Photovoltaikmodule, die von Fachpersonal entworfen, zusammengesetzt und installiert wurden</p> |

## FREETAG LASERTAGSYSTEM

|                  |  |
|------------------|--|
|                  | Geräte, die ausschließlich zu Zwecken der Forschung und Entwicklung entworfen wurden und nur auf zwischenbetrieblicher Ebene bereitgestellt werden |
| <b>gilt seit</b> | 22.7.2012  |

Ab dem jeweiligen in der Richtlinie angeführten Stichtag dürfen in Verkehr gebrachte Elektro- und Elektronikprodukte einschließlich Kabeln und Ersatzteilen folgende Stoffe nicht enthalten:

|                     |  |
|---------------------|--|
| Blei (0,1 %)        | Sechswertiges Chrom (0,1 %)                |
| Quecksilber (0,1 %) | Polybromierte Biphenyle (PBB) (0,1 %)      |
| Cadmium (0,01 %)    | Polybromierte Diphenylether (PBDE) (0,1 %) |

In homogenen Werkstoffen (nicht durch mechanische Vorgänge zerlegbar) werden die in Gewichtsprozent angeführten Konzentrationshöchstwerte toleriert.

### Ausnahmen:

Von diesen Bestimmungen ausgenommen sind die in den Anhängen III und IV der Richtlinie genannten Verwendungen.

### 14.3.2 Elektromagnetische Verträglichkeit (EMV-Richtlinie)

|                   |  |
|-------------------|--|
| <b>Richtlinie</b> | <b>2004/108/EG</b><br>Richtlinie des Europäischen Parlaments und des Rates vom 15.12.2004 zur Angleichung der Rechtsvorschriften der Mitgliedstaaten über die elektromagnetische Verträglichkeit und zur Aufhebung der Richtlinie 89/336/EWG (ABl. Nr. L 390 vom 31.12.2004) |
| <b>gilt für</b>   | Geräte und ortsfeste Anlagen, die elektromagnetische Störungen verursachen können oder deren Betrieb durch diese Störungen beeinträchtigt werden kann und für Endnutzer bestimmt sind.   |

## FREETAG LASERTAGSYSTEM

---

|                                 |   |
|---------------------------------|---|
|                                 | <p>Als Gerät gelten auch Bauteile und Baugruppen, die vom Endnutzer eingebaut werden können, sowie bewegliche Anlagen.</p> <p>Die Richtlinie legt die Schutzanforderungen auf diesem Gebiet sowie die entsprechenden Kontrollmaßnahmen fest.</p>  |
| <b>gilt u.a.<br/>nicht für:</b> | <p>Funkanlagen und Telekommunikationsendeinrichtungen,</p> <p>nicht im Handel erhältliche Amateurfunkgeräte,</p> <p>luftfahrttechnische Erzeugnisse, Teile, Ausrüstungen,</p> <p>Geräte, die nicht für den Endnutzer im Handel sind,</p> <p>Betriebsmittel, die keine elektromagnetischen Störungen verursachen (Kabel, Batterien, Taschenlampen, ...).</p> |
| <b>Gilt:</b>                    | ab 21.7.2007 bis 19.4.2016 (Aufhebung durch <u>2014/30/EU</u> )   |

## 14.4 EU Konformitätserklärung

EU-KONFORMITÄTSERKLÄRUNG (Nr. ...)

Produktmodell/Produkt (Artikel-, Chargen-, Typen- oder Seriennummer): ...

*FreeTag Lasertag Markierer*

Name und Anschrift des Herstellers oder seines Bevollmächtigten: ...

*Emmanuel Panholzer, Katzgraben Straße 90, 4203 Altenberg bei Linz*

Die alleinige Verantwortung für die Ausstellung dieser Konformitätserklärung trägt der Hersteller.

Gegenstand der Erklärung: ...

*FreeTag Lasertagmarkierer*

Der oben beschriebene Gegenstand der Erklärung erfüllt die einschlägigen Harmonisierungsrechtsvorschriften der Union:

Beschränkung gefährlicher Stoffe in Elektro- und Elektronikgeräten(Richtlinie 2011/65/EU)

Elektromagnetische Verträglichkeit (EMV-Richtlinie)(Richtlinie 2004/108/EG)

Angabe der einschlägigen harmonisierten Normen, die zugrunde gelegt wurden, oder Angabe der anderen technischen Spezifikationen, in Bezug auf die die Konformität erklärt wird: ...

Linz am: ...

Panholzer Emmanuel

## 15 RISIKOBEURTEILUNG

# Risikobeurteilung nach EN ISO 14212-1

Produktnam e: FreeTag  
Lasertag-  
markierer  
Dokumentspra-  
che: Deutsch  
Name: Panholzer  
Emmanuel  
Datum: 02.03.2018

### Grenzen des Gerätes

#### Verwendungsgrenzen

Zur bestimmungsgemäßen Verwendung gehören auch die beschriebenen Betriebs- und Instandhaltungsbedingungen.

#### Räumliche Grenzen

Das Gerät wurde ausschließlich für den Betrieb im Trockenen entwickelt.

#### Zeitliche Grenzen

Die Lebensdauer des Gerätes endet vorzeitig mit deren irreparablen Unfunktionalität.  
Das Gerät muss dann verschrottet werden.

### Abkürzungen

| Lebensphase des Arbeitsmittels                     | Schadensausmaß (S)                          | Wahrscheinlichkeit (W)    | Schutzmaßnahme und Lösungsprinzip |
|--|---|---------------------------|-----------------------------------|
| Normalbetrieb Prüfen (NP)                          | Tod (5)                                     | häufig (5)                | K: Konstruktiv                    |
| Rüsten (R)   | schwerer bleibender Gesundheitsschaden (4)  | Gelegentlich (4)          | S: Schutzausrüstung               |
| Montage/Demontage (M)                              | leichter bleibender Gesundheitsschaden (3)  | selten (3)                | W: Warnhinweis                    |
| Inbetriebnahme (I)                                 | heilbare Verletzung mit Arbeitsausfall (2)  | unwahrscheinlich (2)      |                                   |
| Instandhaltung (Wartung, Reparatur, Reinigung) (W) | heilbare Verletzung ohne Arbeitsausfall (1) | sehr unwahrscheinlich (1) |                                   |

# FREETAG LASERTAGSYSTEM

|  |     |              |  |                  |         |   |         |   |         |   |         |
|--|-----|--------------|--|------------------|---------|---|---------|---|---------|---|---------|
| Ursprung   |     | Beschreibung |  | Risiko [ja/nein] |         | Sicherheitsmaßnahmen [Achseung > 9 iii] |         | Sicherheitsmaßnahmen [Achseung > 9 iii] |         | Sicherheitsmaßnahmen [Achseung > 9 iii] |         |
|  |     | S (1-5)      | A (1-5)  | S (1-5)          | A (1-5) | S (1-5)                                 | A (1-5) | S (1-5)                                 | A (1-5) | S (1-5)                                 | A (1-5) |
| und Lösungsprinzip   |     |              |  |                  |         |   |         |   |         |   |         |
| 1   Mechanische Gefährdungen   |     |              |  |                  |         |   |         |   |         |   |         |
| 1.1   Beschleunigung/Abbremsung  | +/- | Ja           |  |                  |         |   |         |   |         |   |         |
| 1.1.1   Quetschen  |     | Ja           |  |                  |         |   |         |   |         |   |         |
| 1.2   Spitzte Teile  |     | Ja           |  |                  |         |   |         |   |         |   |         |
| 1.2.1   Stoßen   |     | Ja           |  |                  |         |   |         |   |         |   |         |
| 1.3   Annäherung eines sich bewegten Teils an einen feststehenden Teil |     | NP; P; M; W. | Durch unachtsames Hantern mit dem Gerät kann es zu Verletzungen an einem Selbst oder an Mitmenschen kommen | 1                | 2       | S: Tragen von Handschuhen               | 1       | 1                                       |         |   |         |
| 1.4   schneidende Teile  |     | Ja           |  |                  |         |   |         |   |         |   |         |
| 1.5   elastische Elemente  |     | Nein         |  |                  |         |   |         |   |         |   |         |
| 1.6   Herabfallende Gegenstände  |     | Nein         |  |                  |         |   |         |   |         |   |         |
| 1.6.1   Quetschen  |     | Ja           |  |                  |         |   |         |   |         |   |         |
| 1.7   Schwerkraft (gespeicherte Energie)                               |     | Ja           |  |                  |         |   |         |   |         |   |         |
| 1.8   Höhe gegenüber Boden   |     | Ja           |  |                  |         |   |         |   |         |   |         |
| 1.9   Hochdruck  |     | Nein         |  |                  |         |   |         |   |         |   |         |
| 1.10   Beweglichkeit des Gerätes                                       |     | Ja           |  |                  |         |   |         |   |         |   |         |
| 1.11   Sich bewegende Teile  |     | Ja           |  |                  |         |   |         |   |         |   |         |
| 1.12   Rotierende Teile  |     | Ja           |  |                  |         |   |         |   |         |   |         |
| 1.13   Rauhe rutschige Oberfläche                                      |     | Nein         |  |                  |         |   |         |   |         |   |         |
| 1.14   Schärfte Kanten   |     | Ja           |  |                  |         |   |         |   |         |   |         |
| 1.15   Standfestigkeit, -sicherheit                                    |     | Nein         |  |                  |         |   |         |   |         |   |         |
| 1.16   Vakuum  |     | Nein         |  |                  |         |   |         |   |         |   |         |
| 2   Elektrische Gefährdungen   |     | Ja           |  |                  |         |   |         |   |         |   |         |
| 2.1   Lichtbögen   |     | Nein         |  |                  |         |   |         |   |         |   |         |
| 2.2   Elektromagnetische Vorgänge                                      |     | Nein         |  |                  |         |   |         |   |         |   |         |
| 2.3   Elektrostatische Vorgänge  |     | Nein         |  |                  |         |   |         |   |         |   |         |
| 2.4   Spannungsführende Teile  |     | Ja           |  |                  |         |   |         |   |         |   |         |
| 2.4.1   Elektrischer Schlag  |     | Ja           |  |                  |         |   |         |   |         |   |         |
| 2.5   Unzureichender Abstand zu unter Hochspannung stehenden Teilen    |     | Nein         |  |                  |         |   |         |   |         |   |         |
| 2.6   Überlast   |     | Nein         |  |                  |         |   |         |   |         |   |         |
| 2.7   Teile, die im Fehlerzustand spannungsführend geworden sind       |     | Ja           |  |                  |         |   |         |   |         |   |         |
| 2.8   Kurzschluss  |     | Ja           |  |                  |         |   |         |   |         |   |         |
| 3   Thermische Gefährdungen  |     | Nein         |  |                  |         |   |         |   |         |   |         |
| 3.1   Explosion  |     | Nein         |  |                  |         |   |         |   |         |   |         |
| 3.2   Flamme   |     | Nein         |  |                  |         |   |         |   |         |   |         |
| 3.3   Objekte oder Materialien mit hoher oder niedriger Temperatur     |     | Nein         |  |                  |         |   |         |   |         |   |         |

## FREETAG LASERTAGSYSTEM

| Beschreibung  |      |  | Ursprung   |   |   |
|---|------|--|--|---|---|
| Gefährdung zu treffen   |      |  | [Ja/Nein]  |   |   |
| Schutzmaßnahme und Lösungsprinzip                                       |      |  | [Achtung > 9 iii]  |   |   |
| <b>4 Lärmgefährdungen</b>   | Ja   |  |  |   |   |
| 4.1 Hohe Lautstärke   | Ja   |  | K: Die Lautstärke bei den Tonwiedergabe kann am Markierer eingestellt werden | 1 | 1 |
|   | Nein |  |  |   |   |
| <b>5 Schwingungsgefährdungen</b>  |      |  |  |   |   |
|   |      |  |  |   |   |
| <b>6 Strahlungsgefährdungen</b>   |      |  |  |   |   |
|   |      |  |  |   |   |
| <b>7 Material-/Substanzgefährdungen</b>                                 |      |  |  |   |   |
| <b>8 Ergonomische Gefährdung</b>  |      |  |  |   |   |
| <b>9 Gefährdung im Zusammenhang mit der Einsatzumgebung des Gerätes</b> |      |  |  |   |   |
| <b>9.1 Staub und Nebel</b>  |      |  |  |   |   |
| <b>9.2 Elektromagnetische Störungen</b>                                 |      |  |  |   |   |
| <b>9.3 Blitzschlag</b>  |      |  |  |   |   |
| <b>9.4 Feuchtigkeit</b>   |      |  |  |   |   |
| <b>9.5 Verunreinigung</b>   |      |  |  |   |   |
| <b>9.6 Schnee</b>   |      |  |  |   |   |
| <b>9.7 Temperatur</b>   |      |  |  |   |   |
| <b>9.8 Wasser</b>   |      |  |  |   |   |
| <b>9.9 Wind</b>   |      |  |  |   |   |
| <b>10 Kombination von Gefährdungen</b>                                  |      |  |  |   |   |

## 16 EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die Abschlussarbeit der Werkmeisterschule Mechatronik mit dem Titel

„*FreeTag Lasertagsystem*“

selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und alle den benutzten Quellen wörtliche oder sinngemäß entnommene Stellen als solche kenntlich gemacht habe.



Emmanuel Panholzer

Linz, am Montag, 26. November 2018

## 17 ABBILDUNGS- UND TABELLENVERZEICHNIS

|  |    |
|--|----|
| Abbildung 1: Aufforderung zur Installation der Arduino Gerätesoftware .....            | 26 |
| Abbildung 2: Auswahl Arduino Mega APK in der Programmierumgebung.....                  | 27 |
| Abbildung 3:Portauswahl in der Programmierumgebung.....                                | 27 |
| Abbildung 4: Auswahl Blink Sketch.....   | 28 |
| Abbildung 5: Upload des Blink Programmes.....  | 28 |
| Abbildung 6:Arduino Mega mit leuchtendem Onboard-Led .....                             | 29 |
| Abbildung 7: Beispiel Ordnerstruktur der Arduino Libraries .....                       | 29 |
| Abbildung 8: Dateien Wtv020sd16p-Ordner .....  | 30 |
| Abbildung 9: Öffnen des Seriellen Monitors.....  | 31 |
| Abbildung 10: Pulslänge vor Korrektur.....   | 32 |
| Abbildung 11:Divisor Start eingefügt.....  | 33 |
| Abbildung 12: Pulslängen nach Korrektur Header .....                                   | 34 |
| Abbildung 13: Pulslängen nach Korrektur divisorstart, divisorzero und divisorone ..... | 35 |
| Abbildung 14: Kontrolle der Übertragung des Spielerpaketes .....                       | 36 |
| Abbildung 15: Liste enemyName .....  | 38 |
| Abbildung 16:Spieldatenparameter.....  | 39 |
| Abbildung 17:Untermenü Laufwerk.....   | 41 |
| Abbildung 18: Formatierungsmenü .....  | 42 |
| Abbildung 19:WTV020SD16P Soundchip.....  | 43 |

---

## FREETAG LASERTAGSYSTEM

---

|   |    |
|---|----|
| Abbildung 20: Einlegen der SD-Karte .....                       | 43 |
| Abbildung 21: Soundchip nach einlegen der SD-Karte .....        | 43 |
| Abbildung 22: LCD-Anzeige nach Inbetriebnahme .....             | 44 |
| Abbildung 23:MilesTag 2 Übertragungsaufbau .....                | 55 |
| Abbildung 24: LED Treiberschaltung .....                        | 61 |
| Abbildung 25: Funktionsweise Sammellinse.....                   | 63 |
| Abbildung 26: Abstrahlwinkel und Linse IR-LED.....              | 64 |
| Abbildung 27: Schaltung IR-Sensoren.....                        | 66 |
| Abbildung 28: Schaltung IR-Sensorkopf inkl. Hit-LED .....       | 67 |
| Abbildung 29: Schaltung verbaut in dem Sensorkopfgehäuse.....   | 67 |
| Abbildung 30:Download 4ad Converter .....                       | 69 |
| Abbildung 31:USB Recordertool .....                             | 70 |
| Abbildung 32: Hochpassfilter erster Ordnung .....               | 73 |
| Abbildung 33:Tiefpassfilter erster Ordnung.....                 | 73 |
| Abbildung 34: Lichtspektrum mit Wellenlängen und Frequenz ..... | 74 |
| Abbildung 35: Prinzipdarstellung Kondensator .....              | 75 |
| Abbildung 36: Schaltzeichen Kondensatoren .....                 | 75 |
| Abbildung 37: LED Darstellung und Schaltsymbol.....             | 77 |
| Abbildung 38: Schaltzeichen N-Kanal Mosfet: .....               | 79 |
| Abbildung 39: Farocode Widerstand 1.....                        | 82 |

## FREETAG LASERTAGSYSTEM

---

|  |    |
|--|----|
| Abbildung 40: Farbcode Widerstand 2.....             | 82 |
| Abbildung 41: Schaltplan Hauptplatine .....          | 83 |
| Abbildung 42:Schaltplan Kopfbandsensoren .....       | 84 |
| Abbildung 43: PCB Layout Hauptplatine.....           | 88 |
| Abbildung 44: PCB Layout Hauptplatine.....           | 88 |
| Abbildung 45: Markierer verkabelt .....              | 89 |
| Abbildung 46: 3D-Darstellung des Gehäuses .....      | 90 |
| Abbildung 47: 3D-Darstellung Gehäuse 2.....          | 90 |
| Abbildung 48: Mechanische Skizze Gehäuse.....        | 91 |
| Abbildung 49: Gehäuse nach dem Verkleben außen.....  | 92 |
| Abbildung 50: Gehäuse nach dem Verkleben innen ..... | 92 |
| Abbildung 51: CE-Vorgangsweise .....                 | 98 |
| Abbildung 52: CE-Zeichen .....                       | 99 |
| <br>Tabelle 1: Produktdaten.....                     | 22 |
| Tabelle 2: Produktleistungen .....                   | 22 |
| Tabelle 3: Qualitätszielbestimmung .....             | 24 |
| Tabelle 4: Signaltoleranzen .....                    | 35 |
| Tabelle 5: Übersetzung Spielerpaket.....             | 56 |
| Tabelle 6: Übersetzung Datenprotokoll.....           | 57 |

## FREETAG LASERTAGSYSTEM

---

|  |    |
|--|----|
| Tabelle 7: Implementierte Befehle via Datenpakete..... | 58 |
| Tabelle 8: LED Vorwiderstände .....                    | 62 |
| Tabelle 9: Soundwiedergabeprotokoll.....               | 68 |
| Tabelle 10: Halbleitermaterialien LED.....             | 77 |
| Tabelle 11: Teileliste Elektronik .....                | 85 |
| Tabelle 12: Mechanische Bauteilliste.....              | 93 |

## 18 DATENBLATTVERZEICHNIS

| Bauteilbezeichnung | Typenbezeichnung  | URL Datenblatt  |
|--------------------|---|---|
| Gelbe Muzzle-LED   | LL-583UYC2C-Y2-4DA  | <a href="http://cdn-reichelt.de/documents/datenblatt/A500/LL-583UYC2C-Y2-4DA.pdf">http://cdn-reichelt.de/documents/datenblatt/A500/LL-583UYC2C-Y2-4DA.pdf</a>   |
| Hit-LED Rot        | LL-583VC2C-V1-4DA   | <a href="http://cdn-reichelt.de/documents/datenblatt/A500/LL-583VC2C-V1-4DA.pdf">http://cdn-reichelt.de/documents/datenblatt/A500/LL-583VC2C-V1-4DA.pdf</a>   |
| IR-LED             | Vishey TSAL 6100  | <a href="https://www.vishay.com/docs/81009/tsal6100.pdf">https://www.vishay.com/docs/81009/tsal6100.pdf</a>   |
| R1, R2, R3, R4, R5 | Metallschichtwiderstand                                       | <a href="http://www.produktinfo.conrad.com/datenblaetter/1400000-1499999/001417604-da-01-en-METALL WID 1 0207 0 6W 36R.pdf">http://www.produktinfo.conrad.com/datenblaetter/1400000-1499999/001417604-da-01-en-METALL WID 1 0207 0 6W 36R.pdf</a>                           |
| R7, R8             | Metallschichtwiderstand                                       | <a href="http://www.produktinfo.conrad.com/datenblaetter/1500000-1599999/001585427-da-01-en-METALLSCHICHT WIDERSTAND MOR01SJ0680A10.pdf">http://www.produktinfo.conrad.com/datenblaetter/1500000-1599999/001585427-da-01-en-METALLSCHICHT WIDERSTAND MOR01SJ0680A10.pdf</a> |
| R9, R10, R11       | Drahtwiderstand 5W u. 7W                                      | <a href="http://cdn-reichelt.de/documents/datenblatt/B400/KH SERIES.pdf">http://cdn-reichelt.de/documents/datenblatt/B400/KH SERIES.pdf</a>   |
| C1                 | Kondensator Kerko 20 % 0,1 µF                                 | <a href="http://www.produktinfo.conrad.com/datenblaetter/1400000-1499999/001420294-da-01-en-MLCC Z5U 100NF 20 50V RM2 54.pdf">http://www.produktinfo.conrad.com/datenblaetter/1400000-1499999/001420294-da-01-en-MLCC Z5U 100NF 20 50V RM2 54.pdf</a>                       |
| C2                 | Vielschichtk. 470 nF RM5                                      | <a href="http://www.produktinfo.conrad.com/datenblaetter/1500000-1599999/001578739-da-01-en-VIELSCHICHTK 470NF RM5 GEG.pdf">http://www.produktinfo.conrad.com/datenblaetter/1500000-1599999/001578739-da-01-en-VIELSCHICHTK 470NF RM5 GEG.pdf</a>                           |
| C3                 | Elektrolyt-Kondensator Radial bedrahtet 220 µF                | <a href="http://www.produktinfo.conrad.com/datenblaetter/1300000-1399999/001328830-da-01-en-ELKO 6 3X11MM RM2 5 220 F 16V 20 .pdf">http://www.produktinfo.conrad.com/datenblaetter/1300000-1399999/001328830-da-01-en-ELKO 6 3X11MM RM2 5 220 F 16V 20 .pdf</a>             |
| C4                 | Elektrolyt-Kondensator radial bedrahtet 1.5 mm 1 µF 50 V 20 % | <a href="http://www.produktinfo.conrad.com/datenblaetter/425000-449999/445724-da-01-en-ELKO SNAP IN 105 C1uF 50V4X5RM1 5.pdf">http://www.produktinfo.conrad.com/datenblaetter/425000-449999/445724-da-01-en-ELKO SNAP IN 105 C1uF 50V4X5RM1 5.pdf</a>                       |

## FREETAG LASERTAGSYSTEM

---

|                           |   |   |
|---------------------------|---|---|
| C5, C6, C7                | ELKO RAD. 105 °C 50 V   | <a href="http://www.produktinfo.conrad.com/datenblaetter/425000-449999/444928-datasheet/01-en-RA-DIAL_ELKO_VY_4_7UF_50_V.pdf">http://www.produktinfo.conrad.com/datenblaetter/425000-449999/444928-datasheet/01-en-RA-DIAL_ELKO_VY_4_7UF_50_V.pdf</a>         |
| IR Empfänger              | TSOP 4838   | <a href="https://www.vishay.com/docs/82459/tso-p48.pdf">https://www.vishay.com/docs/82459/tso-p48.pdf</a>   |
| Audio-Verstärker          | TDA7056B  | <a href="https://www.nxp.com/docs/en/datasheet/TDA7056B.pdf">https://www.nxp.com/docs/en/datasheet/TDA7056B.pdf</a>   |
| Sound-Modul               | WTV020SD16P   | <a href="http://www.elechouse.com/elechouse/images/product/MP3%20Sound%20Mini%20SD%20Card%20Module/WTV020%20datasheet%20V1.8.pdf">http://www.elechouse.com/elechouse/images/product/MP3%20Sound%20Mini%20SD%20Card%20Module/WTV020%20datasheet%20V1.8.pdf</a> |
| Arduino Mega              | Arduino Mega 2560   | <a href="https://store.arduino.cc/arduino-mega-2560-rev3">https://store.arduino.cc/arduino-mega-2560-rev3</a>   |
| 16x2 LCD                  | 16x2 LCD Modul  | <a href="https://www.sparkfun.com/datasheets/LCD/ADM1602K-NSW-FBS-3.3v.pdf">https://www.sparkfun.com/datasheets/LCD/ADM1602K-NSW-FBS-3.3v.pdf</a>   |
| 547B Transistor           | NPN Transistor 547B   | <a href="https://www.farnell.com/datasheets/410427.pdf">https://www.farnell.com/datasheets/410427.pdf</a>   |
| 557B Transistor           | PNP Transistor 557B   | <a href="http://cdn-reichelt.de/documents/datenblatt/A100/BC556_BC557_BC558-CDIL.pdf">http://cdn-reichelt.de/documents/datenblatt/A100/BC556_BC557_BC558-CDIL.pdf</a>   |
| N-Kanal Mosfet            | IRLZ34N   | <a href="http://www.irf.com/product-info/datasheets/data/irlz34n.pdf">http://www.irf.com/product-info/datasheets/data/irlz34n.pdf</a>   |
| Visatron Lautsprecher     | FRWS 5-8  | <a href="http://www.visaton.de/de/produkte/chassiszubehoer/breitband-systeme/frws-5-8-ohm">http://www.visaton.de/de/produkte/chassiszubehoer/breitband-systeme/frws-5-8-ohm</a>   |
| Kontrast Poti LCD         | Cermet Trimmer 10 kΩ  | <a href="http://cdn-reichelt.de/documents/datenblatt/B400/76-10_76-40%23SPE.pdf">http://cdn-reichelt.de/documents/datenblatt/B400/76-10_76-40%23SPE.pdf</a>   |
| Laustärke Poti Verstärker | ACP 9-L 1M  | <a href="http://cdn-reichelt.de/documents/datenblatt/B400/DS_ACP_CA9.pdf">http://cdn-reichelt.de/documents/datenblatt/B400/DS_ACP_CA9.pdf</a>   |
| Printstecker PSS          | Printstecker PSS  | <a href="http://cdn-reichelt.de/documents/datenblatt/C151/PSS254%23IMP.pdf">http://cdn-reichelt.de/documents/datenblatt/C151/PSS254%23IMP.pdf</a>   |
| Kupplungs-Leergehäuse PSK | Kupplungs-Leergehäuse PSK                                     | <a href="http://cdn-reichelt.de/documents/datenblatt/C100/PSK_254-5W_DB.pdf">http://cdn-reichelt.de/documents/datenblatt/C100/PSK_254-5W_DB.pdf</a>   |
| Crimpkontakte für psk 254 | Crimpkontakte für psk 254                                     | <a href="http://cdn-reichelt.de/documents/datenblatt/C151/PSK_KONTAKTE.pdf">http://cdn-reichelt.de/documents/datenblatt/C151/PSK_KONTAKTE.pdf</a>   |
| LED Halterung             | EBF I-5 :: Einbaufassung für 5 mm LEDs, Innenreflektor, chrom | <a href="http://cdn-reichelt.de/documents/datenblatt/A500/SMZ1089.pdf">http://cdn-reichelt.de/documents/datenblatt/A500/SMZ1089.pdf</a>   |

## FREETAG LASERTAGSYSTEM

---

|                       |                                      |   |
|-----------------------|--------------------------------------|---|
| Schlüsselschalter     | Elewind 22 mm illuminated key switch | <a href="https://www.aliexpress.com/item/ELEWIND-22mm-Round-illuminated-key-lock-switch-PB223PY-11Y-21A-G-2-8V/1383910180.html?spm=a2g0s.9042311.0.0.5PX5CL">https://www.aliexpress.com/item/ELEWIND-22mm-Round-illuminated-key-lock-switch-PB223PY-11Y-21A-G-2-8V/1383910180.html?spm=a2g0s.9042311.0.0.5PX5CL</a>       |
| Taster Trigger        | Panasonic AV 32543 AT                | <a href="http://cdn-reichelt.de/documents/datenblatt/C200/AVX3_DB_EN.pdf">http://cdn-reichelt.de/documents/datenblatt/C200/AVX3_DB_EN.pdf</a>   |
| Reload Button         | ELEWIND 19 mm (PM191B-10/J/A)        | <a href="https://www.aliexpress.com/item/ELEWIND-19mm-Pin-terminal-Black-surface-push-button-switch-PM191B-10-J-A/1863466407.html?spm=a2g0s.9042311.0.0.ZOK4Ks">https://www.aliexpress.com/item/ELEWIND-19mm-Pin-terminal-Black-surface-push-button-switch-PM191B-10-J-A/1863466407.html?spm=a2g0s.9042311.0.0.ZOK4Ks</a> |
| Game Button           | ELEWIND black button (PM191F-10/A)   | <a href="https://www.aliexpress.com/item/ELEWIND-Black-aluminum-anodized-button-switch-PM191F-10-A/1863441470.html?spm=a2g0s.9042311.0.0.ZOK4Ks">https://www.aliexpress.com/item/ELEWIND-Black-aluminum-anodized-button-switch-PM191F-10-A/1863441470.html?spm=a2g0s.9042311.0.0.ZOK4Ks</a>                               |
| Spannungswandler +5 V | MC7805CT                             | <a href="http://www.produktinfo.conrad.com/datenblaetter/175000-199999/175030-datasheet-01-en-Spannungsregler_MC7805CT.pdf">http://www.produktinfo.conrad.com/datenblaetter/175000-199999/175030-datasheet-01-en-Spannungsregler_MC7805CT.pdf</a>   |