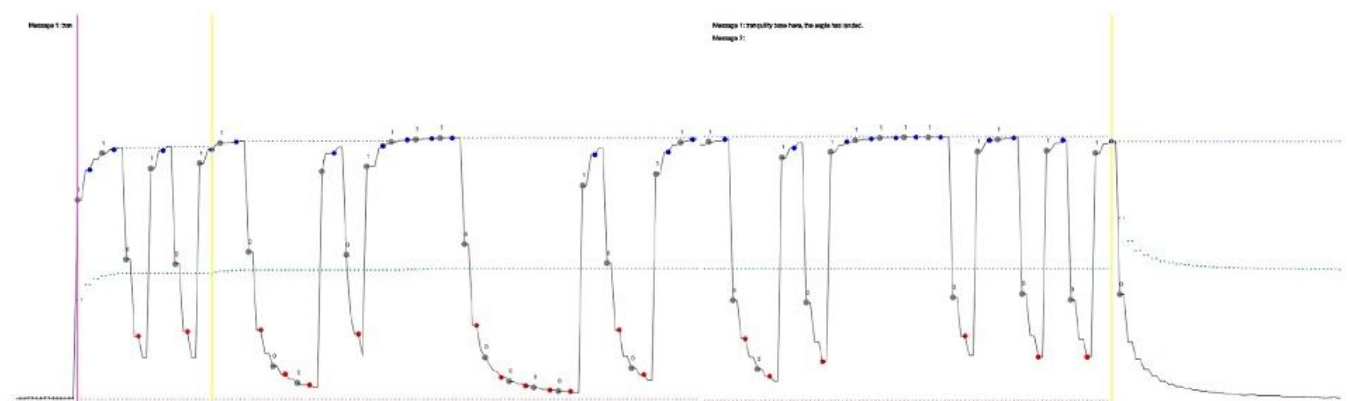
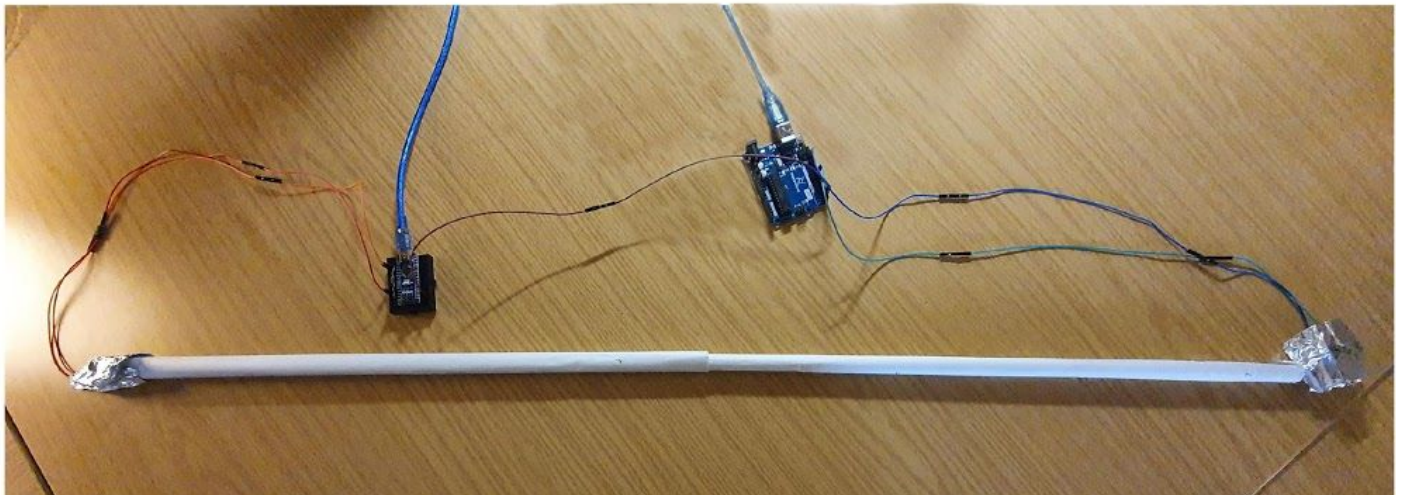


Untersuchung der Eigenschaften von Leuchtdioden zur Verbesserung der Datenübermittlungsrate eines Lichtwellenleitermodells



Inhaltsverzeichnis

Abkürzungsverzeichnis	ii
1. Einführung	1
1.1 Projektübersicht	1
1.2 Funktionsweise der Dateninterpretation	1
1.2.1 Beispielhafte Demonstration	2
1.2.2 Bestimmung von Bits	3
2. Hypothese	4
3. Theoretischer Teil	5
3.1 Funktionsweise von Leuchtdioden	5
3.1.1 Halbleiter und Leitungsarten	5
3.1.2 Dotierung und Leuchtdioden	6
3.2 Kriterien zur Auswahl von Leuchtdioden	8
3.3 Überlegungen zur Datenerfassung	8
3.3.1 Vergleichskriterien und Variablen	8
3.3.2 Parameterauswahl	9
3.3.3 Abweichungen und Umgebungsfaktoren	10
3.4 Auswertung der Daten	11
3.4.1 Allgemeine Erkenntnisse	11
3.4.2 Gesamteinfluss des Parameters Leuchtdauer	12
3.4.3 Analyse aller Leuchtdioden	12
3.4.4 Vergleich aller Leuchtdioden und Schlussfolgerungen	13
3.5 Anwendung der Ergebnisse zur Optimierung der Übertragungsrate	13
3.5.1 Verhalten beim Originalprogramm	13
3.5.2 Maßnahmen zur spezifischen Anpassung des Programms	14
4. Praktischer Teil	15
4.1 Automatisierung der Datenverarbeitung	15
4.1.1 Programmablauf	15
4.1.2 Implementierung der Google Sheets API	16
4.2 Modifizierungen an Aufbau und Code	16
5. Ergebnisse	18
Anhang	19
Literaturverzeichnis	50

Abkürzungsverzeichnis

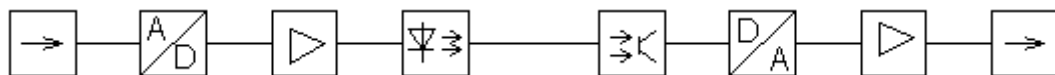
A	Ampere
BER	vom Englischen: "bit error rate", zu Deutsch: Bitfehlerhäufigkeit; Summe aller Bitfehler
bzw.	beziehungsweise
cd	Candela
d. h.	das heißt
E	Energie [J]
eV	Elektronenvolt; $1\text{eV} = 1.6022\text{E-}19\text{ J}$; aufgenommene kinetische Energie eines einfach geladenen Teilchens bei Beschleunigung mit 1 Volt
f	Frequenz [Hz]
GPIO	General Purpose Input / Output
LDR	vom Englischen: "light-dependent-resistor", zu Deutsch: Photodiode, Lichtwiderstand, Lichtsensor
LED	vom Englischen: "light-emitting diode", zu Deutsch: Leuchtdiode
LWL	Lichtwellenleiter
λ (lambda)	Wellenlänge [μm]
ms	Millisekunde
SW	Sensor-Wert; Einheit der Skala des hier verwendeten Lichtwiderstandes
T	Periodenlänge [ms]
V	Volt

1. Einführung

Ziel der Facharbeit ist die Verbesserung der Datenübermittlungsrate eines Lichtwellenleitermodells. Dazu werden die Eigenschaften von Leuchtdioden betrachtet, um diese zur Anpassung an das Modell anwenden zu können. Ein großer Bestandteil der Facharbeit wird es sein, die Dateninterpretation von Lichtwerten zu verstehen. Das dazu geschriebene Processing-Programm wird so in Verbindung mit den Erkenntnissen aus der Datenerhebung angepasst, sodass die Nachrichtenübermittlung letztlich optimal verläuft.

1.1 Projektübersicht

In Lichtwellenleitern, wie Glasfaserkabeln, werden Signale mittels einer Totalreflektion des Lichtes übertragen. Vereinfacht sieht das Prinzip des Übertragungssystems wie in der folgenden Abbildung aus:



A.a Übertragungssystem von Lichtwellenleitern

Zunächst wird das Quellsignal durch einen Analog-Digital-Wandler und eine das elektrische Signal verstärkende Treiberstufe zu einer Leuchtdiode geführt. Diese überträgt die Information in Form von Licht durch den Lichtwellenleiter bis zu einem Photoelement. Es erzeugt aus den Lichtimpulsen erneut ein elektrisches Signal, das ggf. von digital zu analog umgewandelt und verstärkt wird, bevor es zum Empfänger gelangt.

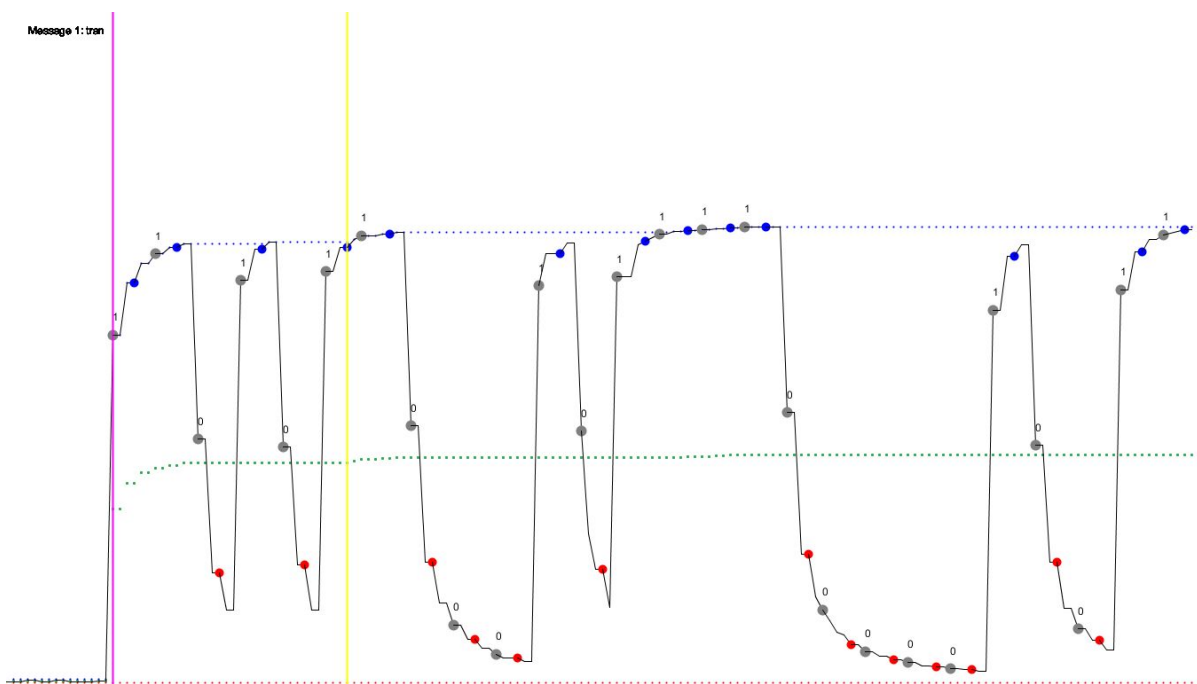
Das System lässt sich nun auf das Modell übertragen: Der Nutzer liefert das Quellsignal über die Texteingabe in einem Processing-Programm. Die Nachricht wird darin als binäre Sequenz codiert. Dies entspricht der Analog-Digital-Wandlung. Die Sequenz wird über die serielle Schnittstelle des Computers an ein Arduino-Programm übergeben, das den Microcontroller steuert. Der Arduino steuert schließlich eine LED an, welche sich in einer Röhre befindet. Mit lichtreflektierender Aluminiumfolie sowie Backpapier ausgekleidet, entsprechen diese Schichten dem Mantelglas, bzw. der äußeren Hülle eines LWL. Die ineinander gesteckten Rohre haben zusammen eine Länge von etwa einem Meter. Am anderen Ende der Röhre befindet sich ein Photowiderstand, der die rohen Lichtwerte ständig über das empfangende Arduino- an das empfangende Processing-Programm weitergibt. Letzteres ist für die Interpretation der Daten sowie die Decodierung und Ausgabe an den Endnutzer eines zweiten Computers zuständig. Der gesamte Versuchsaufbau, wie auf dem Titelblatt, ist zusammen mit Nahaufnahmen im Anhang eingeschlossen.^{[A.b] [A.b] [A.c]}

1.2 Funktionsweise der Dateninterpretation

Bevor mit der Optimierung der Datenübermittlungsrate begonnen werden kann, ist es wichtig zu verstehen, wie der Algorithmus des Empfängerprogramms funktioniert und welche Rolle Leuchtdioden darin spielen. Dazu sollen anhand einer beispielhaften Demonstration der Umfang des Programms sowie die Bestimmung von Bits mit besonderem Fokus erklärt werden. Die Nachricht "tranquility base here, the eagle has landed." wird fortan für alle Testläufe des Modells übertragen. Momentan zuverlässig ist die Übertragung unter Nutzung der Arduino-LED bei einer Periodendauer von $T_{\max} = 125 \text{ ms}$.

1.2.1 Beispielhafte Demonstration

Der Ausschnitt des erzeugten Programmfensters zeigt den Beginn der Messung. Zur besseren Veranschaulichung wurden der LDR und die LED nah beieinander platziert, wodurch der Graph hier eine deutlich höhere Amplitude als im echten Anwendungsfall besitzt. Die aufgenommenen Rohwerte des Lichtwiderstandes werden auf einer Ratioskala von null bis 1024 angegeben und auf die Höhe des Programmfensters angepasst dargestellt. Im weiteren Verlauf wird mit dieser Skala weitergearbeitet und deren Werte in SW (Sensor-Wert) angegeben. Es gilt: $1 \text{ SW} \triangleq 1.75 \text{ mcd}$, mit dem Nullpunkt beider Skalen bei null.



B.b Oszillograph zu Beginn einer Messung

Die gelben Linien zeigen Beginn und Ende der Übertragung an, während der Anfang der Nachricht durch die pinke Linie markiert ist. Die grauen Punkte entsprechen den *Cheats*, die beim Schreiben des Programms zur Fehlererkennung hinzugefügt wurden. Sie dienen zur Identifizierung kritischer Messzeitpunkte und geben den Übertragungszeitpunkt eines jeweiligen Bits an. Erfolgt die Messung zu nahe am folgenden Cheat, wird der Messpunkt statt in rot oder blau aufgrund seiner hohen Fehlerwahrscheinlichkeit pink gefärbt.

1.2.2 Bestimmung von Bits

Der Dechiffrierungsprozess der einzelnen Bits zu einer gesamten Zeichenfolge ist in seiner Gesamtheit anhand der Konsolenausgaben^[B.d] zu erkennen. Die jeweils blauen bzw. roten Punkte, welche genau eine halbe Periode nach dem Cheat, d. h. dem eigentlichen Zeitpunkt der Übertragung stehen, sind die Messpunkte für einen jeweiligen Bit. Zur Bestimmung des Bits ist nur ausschlaggebend, ob der Lichtwert über oder unter dem Durchschnittswert liegt. Grund der zeitlichen Verschiebung ist die für eine LED benötigte Zeit zum Ein- und Ausschalten abzuwarten. Für sämtliche Bitwechsel ist klar im Diagramm der Programmfenster zu erkennen, dass der Lichteinfall nicht sofortig vom Maximum zum Minimum und vice versa reguliert werden kann. Nur durch die Messverschiebung ist die unverzichtbare Reduktion der Fehlerquote möglich. Die einzelnen Bits werden während der Laufzeit zu einer Bitfolge verkettet. Diese wird sofort abgeschnitten, sobald ein Zeichen des Kodierungsbaumes^[B.e] erkannt wird. Die gesamten dechiffrierten Zeichenfolgen aller jeweilig übermittelten Nachrichten werden in der oberen linken Ecke des Programmfensters angezeigt.

2. Hypothese

Aus den vorgestellten Funktionsweisen des Programms gehen eine Mehrzahl von Eigenschaften hervor, welche Leuchtdioden zur Verbesserung der Übertragung potentiell haben sollten. Zunächst soll eine möglichst helle LED die Differenzierung von Bits über die für das Modell verhältnismäßig lange Strecke vereinfachen.

Kernpunkt der Datenerfassung wird jedoch die Analyse der Helligkeitskurven verschiedener Leuchtdioden. Es gilt zu überprüfen, ob sich die Maxima und Minima über vermehrte Wiederholungen von Bitwechseln, wie sie bei den 203 übertragenen Bits der ausgewählten Nachricht vorkommen, erhöhen. Beobachtet wird dazu, wie sich die Helligkeit zum Ende verschiedener Leuchtdauern sowie dem Ende einer weiteren Periode T_{\min} nach dem Ausschalten verhält. Dieses Verfahren wird pro Messung zweimal durchgeführt und simuliert den Ablauf zwei aufeinanderfolgender Bitwechsel. Durch nähere Betrachtung konkreter Lichtwerte bei Bitwechseln soll eine weitere wünschenswerte Eigenschaft überprüft werden: Die Fähigkeit einer LED, ihre Helligkeit möglichst schnell und mit regelmäßigen Werten zu oszillieren.

Die Ergebnisse sollen zur Auswahl einer finalen Leuchtdiode beitragen. Weiterhin können Erkenntnisse aus der Datenauswertung zur Anpassung des Interpretationsverfahrens an die ausgewählte Leuchtdiode beitragen. Durch systematisches Verkürzen der Periodendauer soll schließlich eine optimale Übertragung erzielt werden. Dabei soll die Fehlerrate unter 1,5 Prozent liegen. Angepeilt ist eine Verbesserung von 36 Prozent, was eine erfolgreiche Verminderung der Periodendauer von $T_{\max} = 125 \text{ ms}$ auf $T_{\min} = 80 \text{ ms}$, bzw. Erhöhung der Frequenz von 8 Bits/s auf 12,5 Bits/s, bedeuten würde.

3. Theoretischer Teil

3.1 Funktionsweise von Leuchtdioden

Im Rahmen der Optimierung soll das Verständnis der Funktionsweise von Leuchtdioden zur Auswahl von Leuchtdioden beitragen, im weiteren Verlauf aber auch dazu, Überlegungen zur Datenerfassung zu gestalten sowie die späterhin gemessenen Daten zu erläutern.

Alle LEDs sind Dioden, welche grundsätzlich Halbleiter sind. Daher wird im Folgenden darauf eingegangen, wie Stromfluss innerhalb von Halbleitern zustande kommt, Dioden aufgebaut sind und Licht verschiedener Wellenlängen bzw. Farben in Leuchtdioden entsteht.

3.1.1 Halbleiter und Leitungsarten

Stromfluss entsteht durch die Bewegung freier Elektronen. Innerhalb eines Materials können sie sich aus den Umlaufbahnen ihrer Valenzschalen lösen. Sind die äußeren Elektronenschalen jedoch nicht vollständig besetzt, können anliegende Atome Bindungen eingehen, um die Edelgasregel (vgl. Liersch, S. 51) zu erfüllen, indem sie sich ein gemeinsames Valenzelektron teilen (vgl. Liersch S. 55f.). Diese Elektronenbrücke erschwert die Loslösung einzelner Elektronen, weshalb die Leitfähigkeit materialabhängig ist. Ein Atom geht über die Brücken Bindungen zu je vier weiteren Atomen ein. Da jeweils ein Elektron beider Atome an der Elektronenpaarbindung beteiligt ist, muss das Element, aus dem das Halbleitermaterial besteht, mindestens vier-wertig sein (vgl. Winzker, S. 63). Im Gegensatz zur Leitern oder Isolatoren, bei denen nahezu alle, bzw. gar keine Elektronen frei sind, befinden sich in Halbleitern nahezu alle Elektronen in Valenzschalen. Sie können sich bei Halbleitermaterialien selten loslösen, wodurch ein relativ geringer Stromfluss entsteht.

Nach dem Bändermodell beschreiben Bänder Energieniveaus von Elektronen auf ihren Umlaufbahnen um ein Atom. Im Valenzband befinden sich Elektronen in der Umlaufbahn der vollbesetzten Valenzschale. Dies ist der energieärmste Zustand, aus dem sich die Elektronen schwer lösen (vgl. Kuchling, S. 487). Beim Wechsel auf eine höhere Umlaufbahn, dem Leitungsband, verlassen sie das Valenzband. Es erfolgt die Generation eines freien Ladungsträgers, der sich relativ unbehindert bewegen kann. Möglich ist die Bewegungsfreiheit, da im Leitungsband nur sehr wenige Elektronen aufgrund der beschränkten Leitfähigkeit von Halbleitern fließen. Der Stromfluss durch Generation heißt Elektronenleitung (vgl. Winzker, S. 64). Daran schließt Winzker an, dass bei der Loslösung eines Elektrons gleichzeitig eine "Lücke" in der Valenzschale entsteht, ein sogenanntes Loch. In dieses "schlüpft" ein anderes Elektron, ohne seine Valenzschale zu verlassen. Es wird nämlich keine weitere Energie benötigt, wenn eine gleichwertige Bindung eingegangen

wird, weshalb es sich bei dem füllenden Elektron um eines desselben Atoms handelt. Analog heißt das Füllen von Löchern Rekombination von Ladungsträgern. Infolge der zuvor beschriebenen Atombindungen entsteht eine Art Kettenreaktion von Rekombinationen, wie Winzker sie beschreibt. Diese verläuft durch die kovalenten Bindungen von Atom zu Atom. Da Atome im Grundzustand elektrisch neutral sind, bleiben positiv geladene Atome nach der Generation zurück. Obwohl sich bei den Kettenreaktionen lediglich Elektronen bewegen, wird anschaulich die Bewegung eines Loches als die eines fiktiven, positiven Ladungsträgers, auch Löcherleitung genannt, beschrieben (vgl. Winzker, S. 64).

	Eigenleitung	Störstellenleitung	
		n-dotiert (Elektronenleitung)	p-dotiert (Löcherleitung)
Elemente	Gruppe IV vier Valenzelektronen: C, Si, Ge, Sn	Gruppe V fünf Valenzelektronen: N, P, As, Sb (Donatoren)	Gruppe III drei Valenzelektronen: B, Al, Ga, In (Akzeptoren)
Kristallgitter			
Bändermodell			

C.a Leitungsarten und Bändermodell

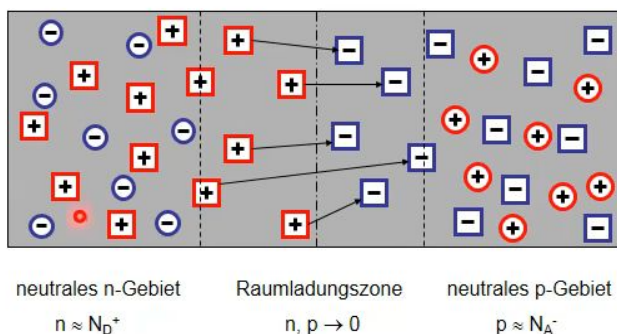
Die Vorgänge beider Leitungsarten, bzw. die der Generation und Rekombination, sowie das Bändermodell sind in der obigen Abbildung zum besseren Verständnis dargestellt. Sie sind die Ursache des Stromflusses bei Halbleitern und somit auch aller Leuchtdioden. Insgesamt herrscht im Halbleiter ein Gleichgewicht zwischen Elektronen- und Löcherleitung, womit die Gesamtzahl aller freien Ladungsträger, und infolgedessen der Stromfluss, konstant bleiben (vgl. Winzker, S. 64). Vereinfacht lässt sich unter Vernachlässigung der Selbstinduktion (vgl. production-partner.de) annehmen, dass das emittierte Licht einer Leuchtdiode proportional zum Stromfluss (vgl. elektronik-kompendium.de) ist.

3.1.2 Dotierung und Leuchtdioden

Im Gegensatz zur Eigenleitung eines reinen Halbleiters, entsteht durch das Einfügen von Fremdatomen eine Störleitung bei Dioden. Dabei werden drei- oder fünf-wertige Atome verwendet, sodass je ein Elektron zu wenig, bzw. zu viel für eine Elektronenpaarbindung vorhanden sind (vgl. Kuchling, S. 491ff.). Für Brücken ungenutzte Elektronen lösen sich mit Leichtigkeit und erzeugen eine Elektronenleitung. Drei-wertige Atome werden aufgrund der

Abgabe freier Elektronen Donatoren genannt. Entsprechend entsteht eine Löcherleitung durch das Defektelektron. Da die fünf-wertigen Atome bei der Rekombination Elektronen aufnehmen, heißen sie Akzeptoren (vgl. Winzker, S. 66f.).

Dioden sind Halbleiterbauelemente mit zwei unterschiedlich dotierten Bereichen. Der n-Leiter verfügt über zusätzliche negative Ladung durch Donatoren, weshalb er auch Überschusshalbleiter genannt wird. Der Bereich des p-Leiters verfügt über einen Elektronenunterschuss durch die Akzeptoren, weshalb er auch Mangelhalbleiter heißt (vgl. Kuchling, S. 492f.). Aufgrund der Ladungsträgerdiffusion bewegen sich Elektronen zum Mangelhalbleiter, wo sie rekombinieren. Im Normalzustand sind Atome elektrisch neutral. Lösen sich Elektronen von ihren ursprünglichen Atomen oder gehen sie Bindungen mit dem Mangelhalbleitermaterial ein, entstehen infolgedessen positive, bzw. negative Ionen (vgl. Berger, 9:00-9:30). Die Ladungstrennung erzeugt ein elektrisches Feld, das der Driftkraft entgegenwirkt, bis ein Kräftegleichgewicht herrscht und die Sperrschicht aufhört zu wachsen (vgl. Berger, 9:30-11:45). Die untere Abbildung zeigt das Verhalten von Ionen (in Kästchen) und freien Ladungsträgern (in Kreisen).



C.b Elektrisches Feld einer Diode

Wird eine äußere Spannung mit dem Minuspol an der p-Schicht und dem Pluspol an der n-Schicht angelegt, bewegen sich die Elektronen aus dem Überschusshalbleiter heraus und zum Mangelhalbleiter hin, wo sie rekombinieren und sich die Schicht vergrößert^[C.c]. Das elektrische Feld wird dadurch verstärkt, weshalb in Sperrrichtung kein Strom fließt, solange die Spannung unter der Durchbruchspannung einer Diode liegt (vgl. Kuchling, S. 494). Bei umgekehrter Polung wirkt die Spannung dem elektrischen Feld entgegen. Berger merkt dazu an: "Der Strom durch den pn-Übergang, also durch die Diode, ist ein Diffusionsstrom. Das innere elektrische Feld hindert ihn eigentlich am fließen, aber wenn [das Feld] zu klein ist, gelingt das nicht mehr." (Berger, 16:40) In diesem Moment ist die Schwellspannung erreicht, wonach die Sperrschicht abgebaut ist. Das Verhalten einer jeweiligen Diode lässt sich in Abhängigkeit von Außentemperatur und Halbleitermaterial durch eine Kennlinie beschreiben

(vgl. Winzker, S. 72f.). Sie zeigen auch für die hier genutzten LEDs, ab welcher Spannung sie zu leuchten beginnen und welche Stromstärke sie dabei haben^[D].

3.2 Kriterien zur Auswahl von Leuchtdioden

Nachdem die Funktionsweisen von Halbleitermaterialien und Dioden erklärt wurden, ist es möglich, die Erkenntnisse zur Auswahl von LEDs anzuwenden. Da ein schneller Wechsel der Lichtwerte bei Bitwechseln gefordert ist, wird angenommen, dass die Sperrschichten der Dioden möglichst schnell auf- und wieder abbaubar sein sollten. Dies wäre bei möglichst kleinen Grenzsichten der Fall, die demzufolge bei bereits kleinen Spannungen abgebaut werden würden. Die Kennlinie einer LED offenbart, dass aufgrund geringer Durchlassspannungen auch geringere Ströme fließen. Folglich ist alleinig eine sehr helle Leuchtdiode auszuwählen, entgegen der anfänglichen Hypothese, nicht zwingend vorteilhaft.

Des Weiteren wurden Leuchtdioden mit Wellenlängen ausgewählt, die über das sichtbare Lichtspektrum weit verteilt sind. Ihre Farben sind blau ($\lambda = 465 \text{ nm}$), grün ($\lambda = 570 \text{ nm}$) und rot ($\lambda = 625 \text{ nm}$). Um unnötig viele Parameter zu vermeiden, sind diese alle der Marke "Tru Components" (TC) und haben ähnliche Schwellspannungen und Durchlassströme^[D]. Sie besitzen weiterhin die gleichen Abstrahlwinkel und klare Schirme, um ihre Helligkeit voll ausschöpfen zu können.

3.3 Überlegungen zur Datenerfassung

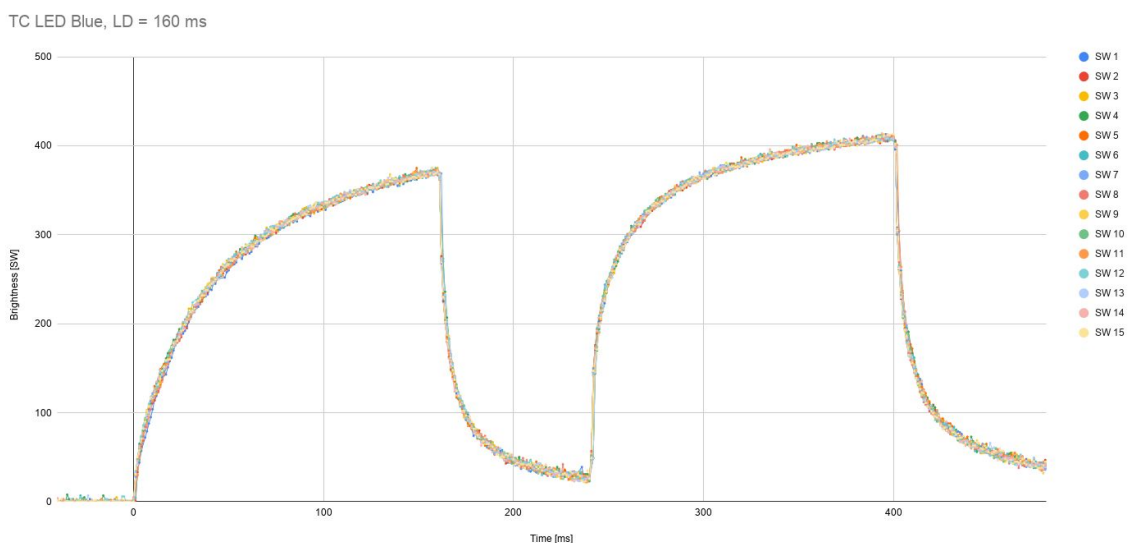
3.3.1 Vergleichskriterien und Variablen

Problematisch ist die zur Kodierung verwendete verlustfreie Komprimierung^[B,e], da dabei keine einheitliche Bytelänge existiert. Entsteht ein Bitfehler, kann somit nicht einfach unterschieden werden, ob es sich um einen längeren Byte oder eine Fehlinterpretation handelt. In diesem Fall ist die Gesamtheit oder zumindest ein Großteil der folgenden Übertragung nichtig, da sie durch die Fehlerkennung bzw. unmöglich gewordene Erkennung eines Bytes verschoben wird. Die Konsolenausgaben zeigen, dass oft nur wenige Bitfehler den gesamten Text unleserlich machen können^[B,f].

Voraussetzung für das Programm ist in diesem Rahmen daher eine nahezu fehlerfreie Übertragung. Die Fehlerquote (BER) ergibt sich aus dem Quotienten der Bitfehlerhäufigkeit und der Gesamtzahl der Bits (vgl. Breed, S. 1). Es werden nur die Fehlerquoten der jeweiligen Übertragungen betrachtet, da das Finden und ggf. Berichtigen von Übertragungsfehlern eher dem Feld eines fehlerkorrigierenden Algorithmus entspricht. Das

zuvor festgelegte Ziel einer Fehlerquote von weniger als 1,5 Prozent, hieße somit maximal drei Fehler unter den 203 übertragenen Bits. Zur Identifizierung von Übertragungsfehlern werden die korrekten Bits der bereits bekannten Nachricht als Prüfinformation mit den interpretierten Bits abgeglichen.

Das in der Hypothese festgelegte Verfahren soll zeigen, wie sich die Lichtwerte nach verlängerter Dunkelheit (Max1), nach dem Abschalten der LED (Min1) für $T_{\min} = 80$ ms, dem erneuten Einschalten danach (Max2) und schließlich erneutem Ausschalten (Min2) entwickeln. Dargelegt wird dadurch, wie sich Leuchtdioden bei rapidem Ein- und Ausschalten entgegen längeren Leuchtperioden verhalten. Die Messpunkte der Variablen sind in der exemplarischen Abbildung in genannter Reihenfolge an $t = 160$, $t = 240$, $t = 400$ und $T = 480$ der Kurve zu sehen.



F.c Helligkeitskurve TC LED Blue, LD = 160 ms

3.3.2 Parameterauswahl

Die Anzahl der Parameter ist im Rahmen dieser Facharbeit möglichst gering zu halten, da sonst eine unüberschaubare Anzahl möglicher Optionen zur Optimierung entstehen. Es werden bereits drei verschiedene Leuchtdioden untersucht, weshalb nicht mehr als ein weiterer Parameter betrachtet werden sollte.

Der Helligkeitsverlauf beim wiederholten Ein- und Ausschalten einer jeden einzelnen Leuchtdiode kann stark variieren, weshalb er anhand verschiedener Leuchtdauern zu analysieren ist. Im experimentellen Teil erfolgt keine Bitbestimmung, sondern nur die Ermittlung der als wichtig gekennzeichneten Rohwerte. Da keine Bits übermittelt werden, simulieren die Leuchtdauern Übermittlungen mehrerer kontinuierlicher Bits mit Wert 1. Die

Leuchtdauern sind Vielfache der angestrebten Leuchtdauer von 80 ms. Das Maximum beträgt dabei 480 ms, da der längste Byte sechsstellig ist, während die LED immer für $T_{\min} = 80$ ms ausgeschaltet bleibt. Zur Diskussion verschiedener Leuchtdauern werden diese fortan abgekürzt (LD) und in n Vielfachen der erstrebten Periodendauer T_{\min} angegeben ($LD = n * T_{\min}$).

3.3.3 Abweichungen und Umgebungsfaktoren

Natürlicherweise weichen die Lichtwerte bei wiederholter Datenaufnahme voneinander ab. Dies ist berücksichtigt, indem die Messung für jede Leuchtdauer pro Diode 15 mal wiederholt wird. Die Zahl kommt durch die Begrenzung der Tabellenlänge zustande, ist aber völlig ausreichend, um ein umfangreiches Bild über das Verhalten der LEDs zu geben. Zum Arbeiten mit den Variablen werden die arithmetischen Mittelwerte dieser vom Datenverarbeitungsprogramm berechnet und in einem separaten Datenblatt für jede Leuchtdiode zusammengefasst.

Die Auszüge der Demonstration zeigen, dass das Umgebungslicht keinerlei Einfluss auf die Messergebnisse hat. Die Auskleidung des Inneren sowie der Enden der Pappröhre mit Aluminiumfolie ist somit effektiv. Der LDR nimmt höchstens Werte bis 5 SW auf, wenn die LED länger abgeschaltet bleibt.

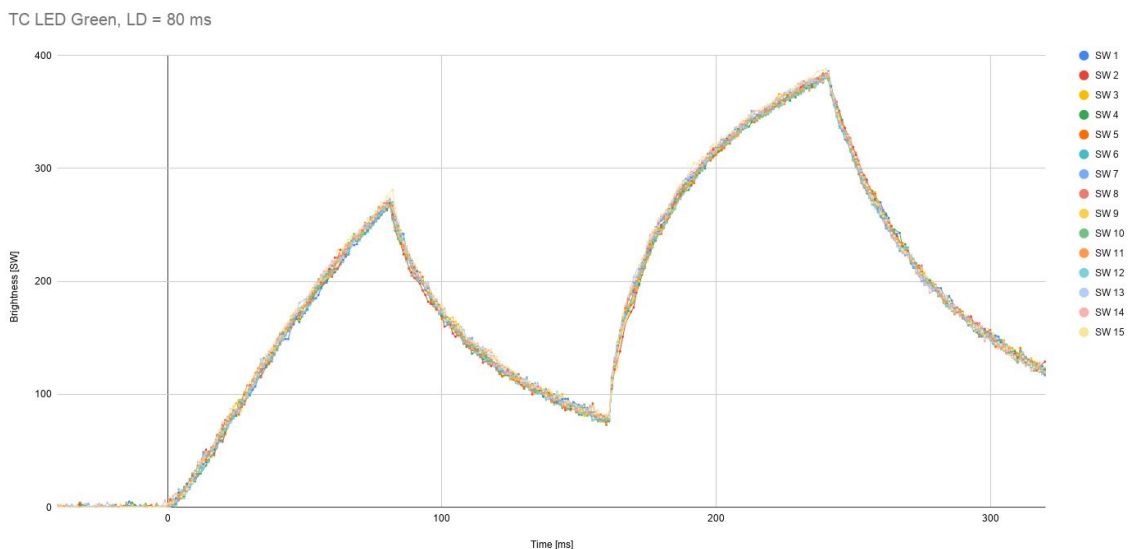
Zu den Umgebungsfaktoren zählen hier nicht nur äußere Umstände: Auch Faktoren der Programmumgebungen sind nicht zu vernachlässigen. Bei den Testläufen und während des Schreibens des Programmcodes fiel auf, dass es beim Processing-Programm, bei welchem eine akkurate zeitliche Koordinierung von höchster Wichtigkeit ist, zu Verschiebungen der periodischen Programmausführung kam. Die Problematik besteht in der Funktion "draw" der Umgebung. Sie wird mit einer Frequenz von 60 Hz, d. h. etwa einmal in 17 Millisekunden ausgeführt (vgl. processing.org), ist dabei jedoch verhältnismäßig unregelmäßig. Daher wird ständig berechnet, wann die Zeit nach der letzten Periode abläuft und dieser Zeitpunkt abgewartet, bevor der zeitabhängige Code ausgeführt wird. Nichtsdestotrotz weicht der Ausführungszeitpunkt aufgrund der äußerst geringen Frequenz, sowie der Leistungsbeschränkung des genutzten Computers ab. Die Divergenz der Periode steigt proportional zur Laufzeit des Programms. Dabei nimmt nicht die Höhe der Abweichungen über Zeit zu, sondern die Häufigkeit der Verspätungen. Daher ist die Länge der zu übertragenden Nachricht unter etwa 15 Wörtern zu halten, bevor sich die zeitlichen Verschiebungen zur Länge einer gesamten Periode summieren und ein Bit versetzt gemessen wird.

3.4 Auswertung der Daten

Besonders die zusammenfassenden Datenblätter der Variablen (Max1, Max2, Min1 und Min2) mit ihren zugehörigen Balkendiagrammen geben Aufschluss über das Verhalten der Leuchtdioden. Aber auch die Helligkeitskurven einzelner Leuchtdauern sind beispielhaft für wünschenswerte oder unzureichende Eigenschaften.

3.4.1 Allgemeine Erkenntnisse

Unabhängig der Parameter verschiedener LEDs und Leuchtdauern lassen sich globale Beobachtungen machen: Max2 liegt immer 20 SW bis 30 SW über Max1. Grund dafür ist, dass der Stromfluss einer Diode, bzw. proportional dazu die Helligkeit, nicht unmittelbar auf null sinkt. Bevor der Stromfluss in den Ausgangswert zurückfallen kann, steigt er nach Ende der Periode bereits wieder an, weshalb er bei Max2 höher ist. Analog dazu gilt dasselbe für Min1 und Min2: Bei Max2 ist die LED heller, weshalb sie auch bei Min2 mit gleicher Periodendauer noch etwas weiter aufleuchtet. Folglich sind die Minima abhängig von vorhergehenden Maxima, als auch der Dauer, in der die LED ausgeschaltet bleibt (hier konstant bei T_{\min}). Dieser Effekt ist im Graphen der grünen Leuchtdiode von "Tru Components" außerordentlich deutlich.



F.f Helligkeitskurve TC LED Green. LD = 80 ms

Dadurch bestätigt sich, wie in der Hypothese bereits vermutet, die Erhöhung von Max2 und Min2 bei wiederholten Bitwechseln. Es ist anzunehmen, dass die Erhöhung der Maxima und Minima über sehr häufige Wiederholungen von Bitwechseln unvorhersehbarer sowie

extremer werden. Dadurch lässt sich die regelmäßige Oszillation als wünschenswerte Eigenschaft, wie in der Hypothese formuliert, zur regelmäßigen Oszillation unabhängig von der Anzahl der Bitwechsel erweitern.

3.4.2 Gesamteinfluss des Parameters Leuchtdauer

Idealerweise ist genanntes Merkmal gleichfalls unabhängig aller Leuchtdauern. Stattdessen ist zu beobachten, dass die Streuung der Messkurven mit geringeren Leuchtdauern zunehmen, welche jedoch für die meisten LEDs tolerierbar bleiben. Weiterhin zeigt sich, dass sämtliche Lichtwerte mit abnehmender Leuchtdauer ebenfalls sinken. Die Zerfallsrate verhält sich dabei für jede Diode unterschiedlich und ist möglichst gering zu halten. Überraschenderweise zeigt sich, dass sämtliche LEDs verschieden auf die jeweiligen Leuchtdauern reagieren, weshalb sich keine weiteren Dioden-übergreifenden Schlüsse zu den Leuchtdauern machen lassen. Vor dem Vergleich werden sie aus diesem Grund alle einzeln betrachtet.

3.4.3 Analyse aller Leuchtdioden

Wie zu erwarten, schneidet die standardmäßige LED des Arduino schlecht ab. Der gefärbte Schirm filtert einen Großteil des Lichtspektrums ab, weshalb nur ein Höchstwert von 150 SW erreicht wird. Die Helligkeit ist, auch bei klaren, einwandfreien Unterschieden zwischen Maxima und Minima, über die Distanz von einem Meter unzureichend. Nichtsdestotrotz weist die LED ein sehr vorhersehbares Verhalten auf: Dem Diagramm der Durchschnittswerte ist zu entnehmen, dass die Beträge der Variablen allesamt linear abnehmen^[F.a].

Die TC LED in Blau liegt im Gegensatz dazu mit Höchstwerten von 445 SW in einer angemessenen Größenordnung. Ihre Streuungen sind von der Leuchtdauer unabhängig und kompensierbar. Allerdings nimmt Max2 nicht wie Max1 linear, sondern exponentiell ab. Der Effekt würde sich mit hoher Sicherheit bei rapiden, aufeinanderfolgenden Bitwechseln verstärken, weshalb auch die Leistung dieser Leuchtdiode unzureichend ist.

Keinerlei positive Merkmale weist die grüne LED derselben Marke auf. Vergleichsweise besteht ein geringerer Unterschied zwischen Maximum und Minimum sowie signifikante Unterschiede zwischen Min1 und Min2. Ähnlich wie bei den vorherigen Dioden festgestellt, sind diese Charakteristika extrem anfällig sich bei längeren Bitfolgen zu verschlechtern. Das zuvor genannte Ausschlusskriterium, nachdem sich die Erhöhung von Maxima und Minima bei Bitwechseln verstärken, ist in der vorherigen Abbildung für LD = 1T am deutlichsten. Ersichtlich gruppierte^[F.e], jedoch unverhältnismäßig weite Streuungen, machen das Verhalten zudem sprunghaft^{[F.d], [F.f]}.

Herausstechend ist die rote TC LED durch ihr extrem konstantes Verhalten: Nahezu unabhängig der Leuchtdauer gilt $\text{Max}_1 = \text{Max}_2$, $\text{Min}_1 = \text{Min}_2$ sowie $\text{Max} \approx 2.15 \cdot \text{Min}^{[F.q]}$. Letzterer Zusammenhang bedeutet, dass diese LED mit einem vergleichsmäßig großen Höchstwert bei 945 SW ihre Helligkeit in einer Periode nur auf etwa die Hälfte mindern kann. Der Graph für $\text{LD} = 6T^{[F.h]}$ zeigt, im Gegenzug die nahezu infinitesimale, benötigte Dauer, um den Maximalbetrag zu erreichen. Die Abnahme der Helligkeit über verkürzte Leuchtdauern ist aufgrund ihrer Geringfügigkeit schlecht von Messungenauigkeiten zu unterscheiden, weshalb sie für Leuchtdauern um T_{\min} als inexistent angesehen werden kann. Die Ergebnisse schienen zunächst überaus unwahrscheinlich, waren jedoch bei einer Wiederholung der gesamten Messung^[F] mit identischen Werten reproduzierbar.

3.4.4 Vergleich aller Leuchtdioden und Schlussfolgerungen

Beinahe alle Leuchtdioden weisen gewisse Ausschlusskriterien auf. Zuwider der Annahme der Hypothese, nach welcher hellere LEDs die Differenzierung von Bits vereinfachen würden, sind es die der Marke "True Components", welche bei frequenten Bitwechseln unter verhältnismäßig starkem Helligkeitsverlust leiden oder unberechenbar auf sie reagieren. Ausschließlich die rote TC LED ist in ihren Eigenschaften nahezu optimal. Die Abgrenzung ihres Verhaltens ist schwer zu begründen, da Betriebsspannungen- und Ströme sowie Materialeigenschaften alle sehr ähnlich sind^[D].

3.5 Anwendung der Ergebnisse zur Optimierung der Übertragungsrate

Die Nachrichtenübertragung gilt es nun mithilfe der roten Leuchtdiode zu optimieren. Aufgrund ihrer soeben beschriebenen Merkmale, muss der Processing-Code im Folgenden an sie angepasst werden.

3.5.1 Verhalten beim Originalprogramm

Der bisherig angewandte Algorithmus zur Bitbestimmung erweist sich als unpassend unter Nutzung der neuen LED. Grund ist der geringe Unterschied zwischen ihrem Minimal- und Durchschnittswert^[H.a]. Ihre wünschenswerten Eigenschaften bleiben entgegen den Annahmen des vorherigen Kapitels jedoch vollständig erhalten. Unabhängig von der Leuchtdauer vor einem jeweiligen Bitwechsel fällt die Helligkeit nach einer Periode auf denselben festen Wert, wie in den Messungen, mit $\text{Max} = 2.15 \cdot \text{Min}$. Für jede weitere Periode, in der die LED ausgeschaltet bleibt, fällt der Wert weiterhin schlagartig. Diese Eigenschaft ging aus der Datenerfassung zwar nicht hervor, ist jedoch nicht nachteilig. Des

Weiteren wechselt der Helligkeitswert von jedem beliebigen Minimalwert auf das feste Maximum von 945 SW.

3.5.2 Maßnahmen zur spezifischen Anpassung des Programms

Unter Ausnutzung letzterer Eigenschaft wurde der Durchschnittswert umfunktioniert zu einer festen Schwelle knapp unter dem Maximum. Da die Diode in der Lage ist rapide und konstant zwischen den Extremwerten zu oszillieren, wird nur überprüft, ob der aktuelle Wert über oder unter der Schwelle liegt. Mit dieser Verbesserung verläuft die Übermittlung bei T_{\max} mit einer Fehlerquote von null Prozent^[H.b].

Somit wird die Periodendauer auf T_{\min} verkürzt. Dabei entstehen jedoch sehr viele Bitfehler, weshalb der Programmcode eine weitere Variable, "periodDelay", erhält. Sie reguliert die Messverschiebung (vgl. Punkt 1.2.2), welche eine Konstante mit Wert $0,5T$ war. Aufgrund der schnellen Oszillation kann sie, dem tatsächlichen Übertragungszeitpunkt näher kommend, verringert werden. Der zeitlichen Verschiebung aufgrund der Leistungsbeschränkung der Processing-Umgebung über Zeit wirkt sie somit entgegen. Zahlreiche Versuche und Ideen, um diese zu kompensieren, bewirken jedoch nichts, weil die Verschiebung unvorhersehbar ist (vgl. Punkt 3.3.3). Folglich ist eine Übertragung für $T < T_{\max}$ trotz einer optimalen Leuchtdiode mit einer hohen Fehlerquote verbunden.

In einer Art zweiten Phase der Anpassung des Programmcodes wurde ein fehlerkorrigierender Algorithmus entwickelt, welcher die angestrebte Periodendauer T_{\min} anwendbar macht. Die temporären Konsolenausgaben zeigen das Vorgehen, nach dem jede Folge von Nullen durch ihre Übertragungsdauer überprüft wird^[H.c], ^[H.d]. Die Funktionsweise der Fehlerkorrektur ist im praktischen Teil weiter ausgeführt. Systematisches Verringern der Periodendauer ergibt, dass somit Übertragungen bei T_{\min} fehlerfrei funktionieren und bis zu einer optimierten Periodendauer $T_{\text{opt}} = 65$ reduzierbar sind.

4. Praktischer Teil

Der LDR und die LED wurden in eine geschlossene Umgebung direkt nebeneinander platziert, um die Einheit der Sensor-Skala in Candela zu bestimmen. Der Mittelwert der roten TC LED ergibt sich aus dem Datenblatt mit 1750 mcd. Bei eingeschalteter LED ergaben die Messungen etwa einen Wert von 1004 SW. Somit ergibt sich für die Skala: $1 \text{ SW} = 1,75 \text{ cd}$ pro $1004 \text{ SW} \approx 1,75 \text{ mcd}$.

Zur Erzeugung der Daten wurde ein neues Arduino-Programm geschrieben, dass gleichzeitig für das Emittieren und Aufnehmen des Lichtes zuständig ist. Das Hinzuziehen eines dritten, unabhängigen Microcontrollers für die Datenaufnahme, ersparte das ständig wechselnde Hochladen von alten und neuen Programmcodes.

4.1 Automatisierung der Datenverarbeitung

Schon von Beginn an war klar, dass die Messwerte weder von Hand erzeugbar, noch dokumentierbar waren. Daher musste ein separates Programm geschrieben werden, dass den dritten Arduino ansteuert sowie dessen Werte verarbeitet und verzeichnet. Dazu wurde die Programmiersprache Python verwendet, weil ihr Syntax minimalistisch und ausdrucksstark ist. Sie erleichtert den Umgang mit Daten, da Python eine objektorientierte Sprache ist und das Nutzen von Klassen ermöglicht. Zusätzlich verwaltet sie Speicher dynamisch und automatisch. Somit ließen sich Ergänzungen zur Menge der Daten von der Entwicklung des Programms bis zur Datenaufnahme problemlos skalieren. Es war effizienter diesen Prozess zu automatisieren, unter anderem indem das Programm die Datenblätter selbstständig erzeugt und auf Google Sheets hochlädt.

Das Python-Programm wurde noch vor dessen Ausarbeitung in die Regionen "Serial Communication", "Data Extraction & Evaluation" und "Google Sheets Upload" unterteilt. Der Ablauf der Datenerfassung kann diesen Bereichen folgend erklärt werden.

4.1.1 Programmablauf

Zunächst sendet das Arduino-Programm die Werte als Rohtext in Form von Bits über die serielle Schnittstelle an das Python-Programm. Dieses liest jeden Zeit- mit dessen zugehörigem Lichtwert heraus. Weiterhin ist das Verfahren für eine variable Anzahl an Wiederholungen für jede Leuchtdauer in einem Programmablauf möglich. Die Differenzierung der Werttypen erfolgt im Programm durch Linientrennung und sogenannte "Header" über jedem Datensatz.

Es wird für jede Leuchtdauer der nächstmögliche Messzeitpunkt aller jeweiligen Variablen ermittelt und das arithmetische Mittel aller Variablen berechnet. Am Ende der zweiten Region liegt in separaten Arrays (Listen) für jede Leuchtdauer sowohl eine Klasse der Koordinaten aus t und den Messungen (SW 1-15), als auch eine Klasse für die Durchschnittswerte aller Variablen.

4.1.2 Implementierung der Google Sheets API

Die finalen Dokumente und darin enthaltenen Datenblätter werden in der letzten Region des Programms automatisch erzeugt und bearbeitet. Es ist in der Lage zu unterscheiden, ob ein Dokument erstellt, gewisse Teile überschrieben oder ergänzt werden müssen. Dabei mussten zahlreiche, extensive Algorithmen entwickelt werden, die bei variablen Datenmengen eine feste Tabellenstruktur generieren und sämtliche Zellen eines einzelnen Datenblattes mit einem einzigen Upload befüllen. Letzteres ist notwendig, sodass die Upload-Beschränkungen der Google Sheets API nicht überlastet werden. Nähere Erläuterungen zur Entwicklung des Programms sind jedoch nicht zielführend für die Untersuchung der Eigenschaften von Leuchtdioden. Deshalb ist der Python-Code lediglich im Anhang zur näheren Betrachtung beigelegt^[E.a]. Die Konsolenlogs waren dabei eine fundamentale Unterstützung beim Verfassen des Codes. Sämtliche Ausgaben, die sich bei der Erstellung der Datensätze generierten, wurden festgehalten und demonstrieren die Fähigkeiten des Codes in seinem vollen Maße^[E.b - E.e].

Ausschließlich die Fehlerquoten wurden bei der Datenerhebung nicht dokumentiert, da sie als Ausschlusskriterium fungierten. Betrug die BER mehr als 1,5 Prozent, war es nicht notwendig Fehlerquoten für unzureichende Algorithmen des Processing-Programms zu verzeichnen. Dementsprechend gab es nur den finalen Processing-Code, welcher die angestrebte Fehlerquote einhielt. Das Sammeln von Werten für das eindeutig funktionierende Programm wurde durch zahlreiche, exemplarische Abbildungen abgelöst. Schließlich sind bei Laufzeit generierte Fenster- und Konsolenausgaben zum Vergleich der Wirksamkeit deutlich vertrauenswürdiger und aussagekräftiger, als ihre alleinigen Zahlen.

4.2 Modifizierungen an Aufbau und Code

Am originalen Processing-Programm mussten nur wenige Änderungen vorgenommen werden, um die Fehlerquote ermitteln zu können: Am Ende jeder Nachricht werden die interpretierten Bits mit den vorliegenden Prüfbits verglichen. Sollte das Programm durch Fehlinterpretation des terminierenden Bytes weiter laufen, bricht das Programm fortan die Dechiffrierung ab, sodass die Bits nichtsdestotrotz vergleichbar sind. Bei frühzeitiger

Beendigung der Nachricht fordert das Programm den Nutzer zum eigenständigen Suchen der Fehler auf, da das Generieren einer Fehlerquote bei unvollständiger Interpretation unsinnig ist. Die hinzugefügten Fähigkeiten des Programms sind anhand der Konsolenausgaben aller drei genannten Fälle dargestellt^[G]. Letztlich wurde die direkte Kabelverbindung zwischen dem sendenden und empfangenden Arduino aufgehoben, da ein Cheat bei bereits bekannter Nachricht nicht notwendig war.

Die Schaltung wurde so umgebaut, dass sich durch Steckverbindungen für jede Diode variable Widerstände ändern lassen würden. Um Spannung und Stromstärke konstant zu halten, wurden sie durch ein vom Arduino unabhängiges Netzteil bezogen. Die äußere Spannung konnte nur mit dem Microcontroller mithilfe eines Relais gesteuert werden^[A,e]. Bei der Recherche ergab sich jedoch, dass die GPIO-Pins des Arduinos Strom und Spannung für geringe Stromstärken bis 20 mA und Spannungen bis 5 V durch innere Widerstände automatisch regulieren können (vgl. store.arduino.cc). Obwohl das auf Langzeit unvorteilhaft für die Elektronik ist, stellt dies für kurze Betriebszeiten wie im Rahmen des LWL-Modells kein Problem dar. Des weiteren war die Schaltung via Relay extrem verzögert. Somit wurden die direkten Steckverbindungen von den GPIO-Pins zu den Instrumenten wiederhergestellt.

Unter Ausnutzung der schnellen Oszillation der roten TC LED korrigiert der neue Algorithmus jeweilige Bits mit Wert null, die durch Messverschiebungen fälschlicherweise mit Wert eins interpretiert werden. Dem konkreten Anwendungsfall entsprechend, ermittelt und kennzeichnet das Programm die Steigung des Graphen^[H,e]. Somit kann die Fallzeit vor jedem Bitwechsel von Nullen zu Einsen bestimmt werden. Die Länge des Zeitraums gibt Aufschluss darüber, wie viele Bits mit einem Wert von null darin übertragen werden müssten. Die theoretische Anzahl wird bei jedem interpretierten Bit des Wertes eins abgeglichen. Sind sie nicht gleich bedeutet das eine verspätete Messung des letzten Bits mit Wert null, womit der Bitfehler korrigierbar ist^[H,c], ^[H,d]. Auch hier ist der Algorithmus nur auf den vorliegenden Anwendungsfall, d. h. das Verhalten bei einer Röhrenlänge von einem Meter und der roten TC LED, optimiert.

5. Ergebnisse

Die Optimierung der Datenübermittlungsrate des LWL-Modells war erfolgreich. Durch die ausführliche Erklärung des Processing-Programms in der Einleitung konnten Variablen von Wichtigkeit bereits in der Hypothese definiert werden. Das festgelegte Verfahren, in dem zwei aufeinander folgende Bitwechsel zusammen mit verschiedenen Leuchtdauern simuliert wurden, bestätigte die Annahme, dass Minima abhängig von vorherigen Leuchtdauern sind. Über kürzere Zeiträume der Dunkelheit erhöhen sich die Minima, während sich die Maxima bei längeren Leuchtdauern vergrößern. Diese Beobachtung erklärt das inkonsistente Verhalten vieler Leuchtdioden, das die Nachrichtenübermittlungen bei Periodendauern $T < T_{\max}$ unmöglich machen. Glücklicherweise weist die rote TC LED unabhängig von Leuchtdauer und Periodenlänge konstante Lichtwerte zusammen mit einer extrem schnellen Oszillation auf. Obwohl die Programmumgebung die Dateninterpretation im Anwendungsfall mit T_{\min} unrealistisch machte, konnten die Merkmale der LED aus der Datenauswertung zur Entwicklung eines fehlerkorrigierenden Algorithmus ausgenutzt werden. Dieser geht zwar über den Rahmen der Facharbeit hinaus, spielt aber eine unverzichtbare Rolle in der Optimierung. Der Code wurde an die Leuchtdiode angepasst, womit ein BER $< 1,5$ Prozent gewährleistet ist. Schließlich konnte die Periodendauer unter Anwendung der Fehlerkorrektur über T_{\min} bis hin zu $T_{\text{opt}} = 65 \text{ ms}$ reduziert werden. Dies entspricht einer enormen Verbesserung von 48 Prozent, statt der zu Beginn angestrebten 36 Prozent. Die finale Frequenz beträgt bemerkenswerte 15,4 Bits/s.

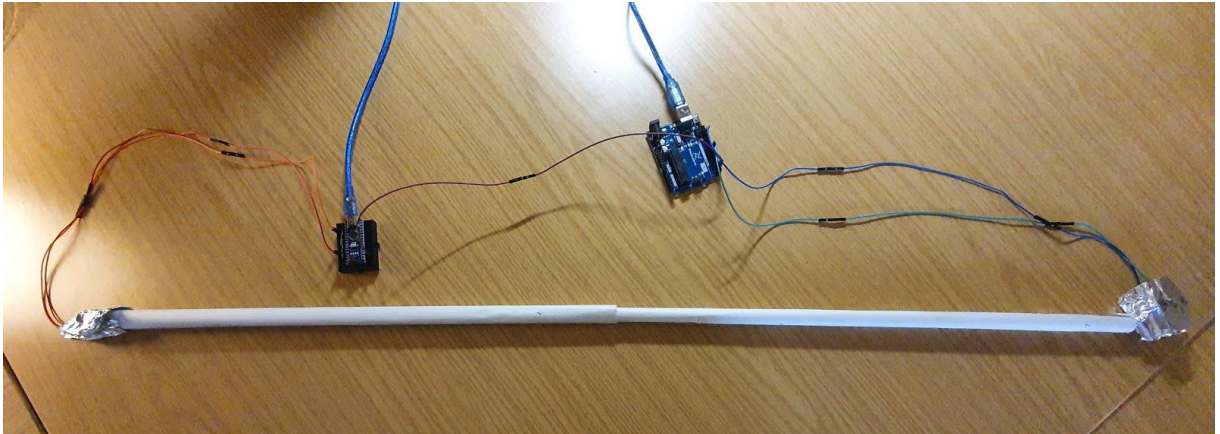
Anhang

A. Versuchsaufbauten	21
A.a Übertragungssystem Lichtwellenleiter	21
A.b Aktor Nahaufnahme mit roter TC LED	21
A.c Sensor Nahaufnahme	22
A.d Aufbau Nachrichtenübermittlung	22
A.d Aufbau bei der Datenerfassung	22
A.e Aufbau bei der Datenerfassung mit Relay	23
B. Originales Processing-Programm	23
B.a Programmcode	23
B.b Oszillograph zu Beginn der Messung	23
B.c Oszillograph am Ende der Messung	24
B.d Konsolenausgabe zu Beginn und Ende der Messung	24
B.e Codierung der verlustfreien Komprimierung am Binärbaum	25
B.f Bitfehler und Problematik der verlustfreien Übertragung	25
C. Funktionsweise von Leuchtdioden	26
C.a Leitungsarten und Bändermodell	26
C.b Elektrisches Feld einer Diode	26
C.c Verhalten einer Diode in Sperrichtung	27
C.d Elektromagnetisches Farbspektrum	27
C.e Emittierung von Licht an der LED	28
C.f Aufbau einer LED	28
D. True Components LEDs	29
D.a Datenblatt der blauen TC LED	29
D.b Datenblatt der grünen TC LED	31
D.c Datenblatt der roten TC LED	33
E. Datenerfassung- und Upload	36
E.a Programmcode	36
E.b Konsolenausgaben beim Daten-Upload für Arduino Standard LED	36
E.c Konsolenausgaben beim Daten-Upload für blaue TC LED	37
E.d Konsolenausgaben beim Daten-Upload mit Überschreibung	38
E.e Konsolenausgaben beim Daten-Upload mit Ergänzung	39
F. Datensätze	40
F.a Data-Average für Arduino Standard LED	40
F.b Data-Average für blaue TC LED	40
F.c Helligkeitskurve TC LED Blue, LD = 160 ms	41
F.d Data-Average für grüne TC LED	41
F.e Helligkeitskurve TC LED Green, LD = 400 ms	42

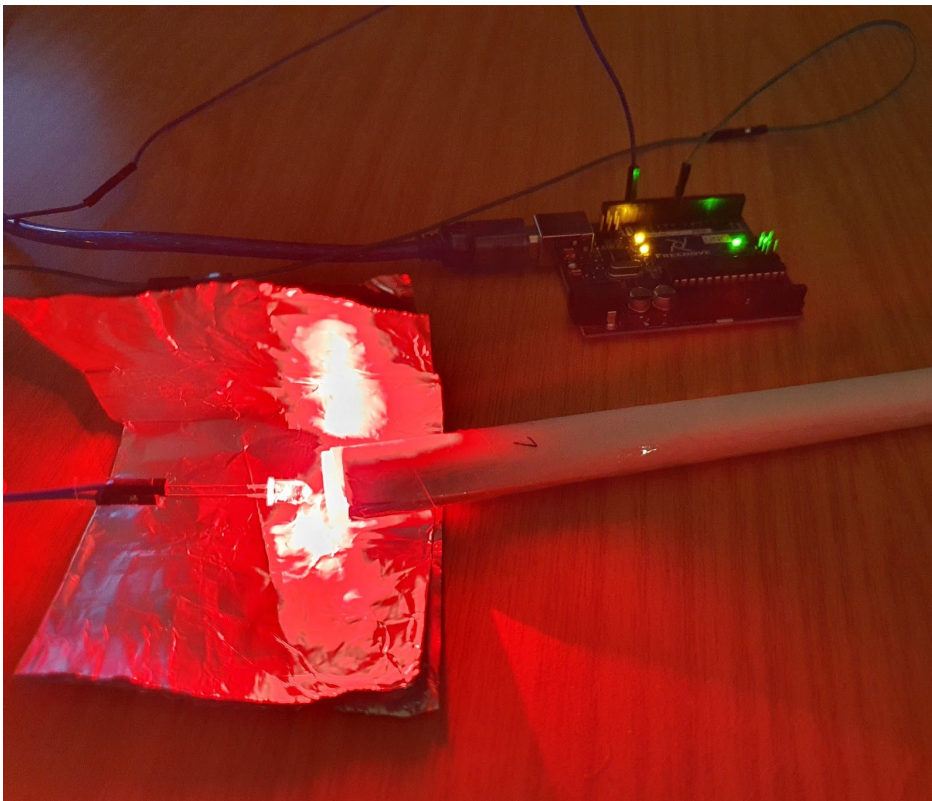
F.f Helligkeitskurve TC LED Green, LD = 80 ms	42
F.g Data-Average für rote TC LED	43
F.h Helligkeitskurve TC LED Red, LD = 480 ms	43
F.i Helligkeitskurve TC LED Red, LD = 80 ms	44
G. Fehlerquote	44
G.a Konsolenausgabe für fehlerfreie Übertragung	44
G.b Konsolenausgabe für Übertragung mit Fehler	45
G.c Konsolenausgabe für unvollständige Übertragung	45
H. Optimierung mit roter TC LED	46
H.a Problematische Übertragung vor Anpassung des Programms, T = 125 ms	46
H.b Erfolgreiche Übertragung nach Anpassung des Programms, T = 125 ms	46
H.c Konsolenausgaben bei Identifizierung eines Übertragungsfehlers	47
H.d Konsolenausgaben bei Korrektur von Übertragungsfehlern	47
H.e Am Oszillograph markierte Steigungen und Fehlerkorrekturen	48

A. Versuchsaufbauten

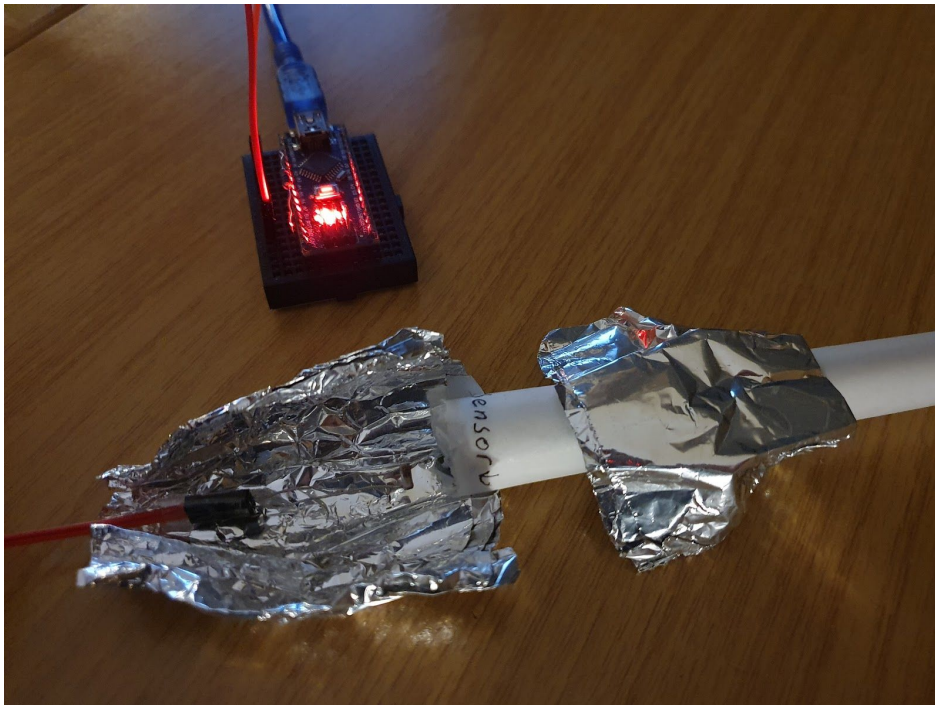
A.a Übertragungssystem Lichtwellenleiter



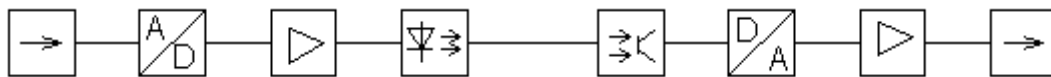
A.b Aktor Nahaufnahme mit roter TC LED



A.c Sensor Nahaufnahme

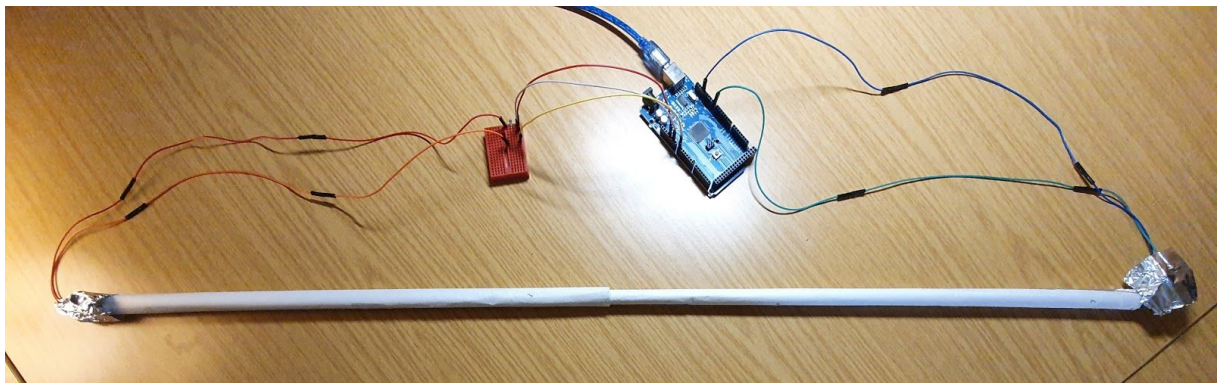


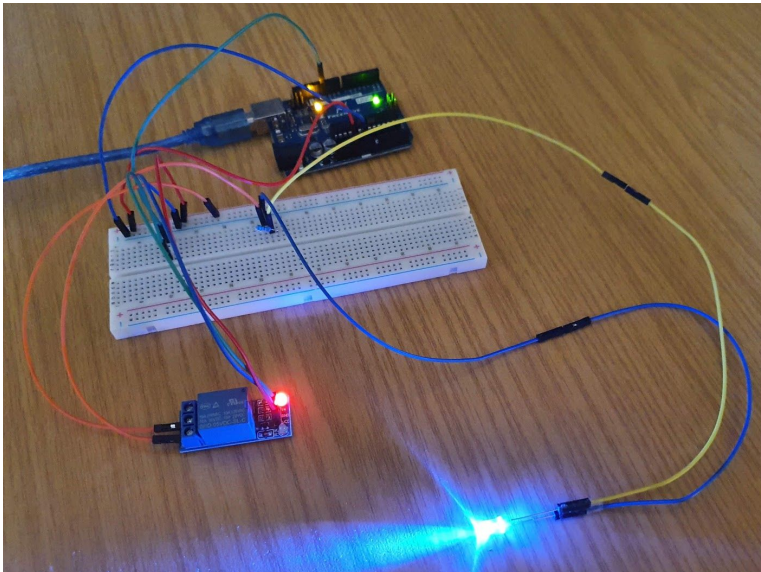
A.d Aufbau Nachrichtenübermittlung



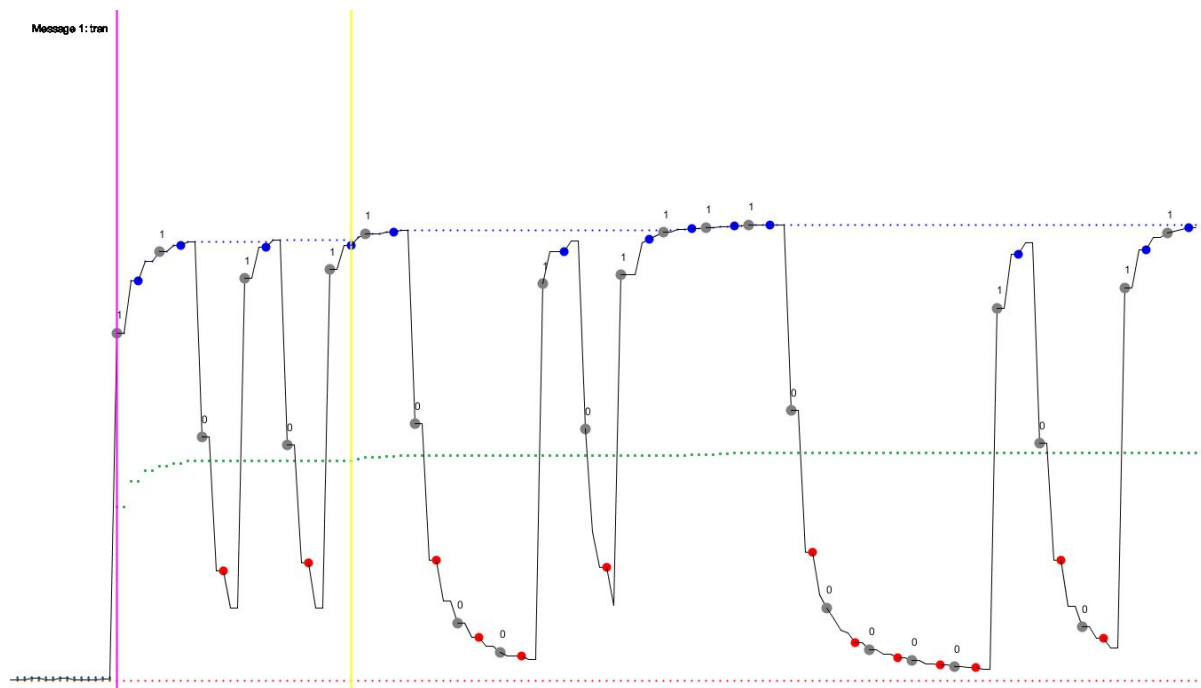
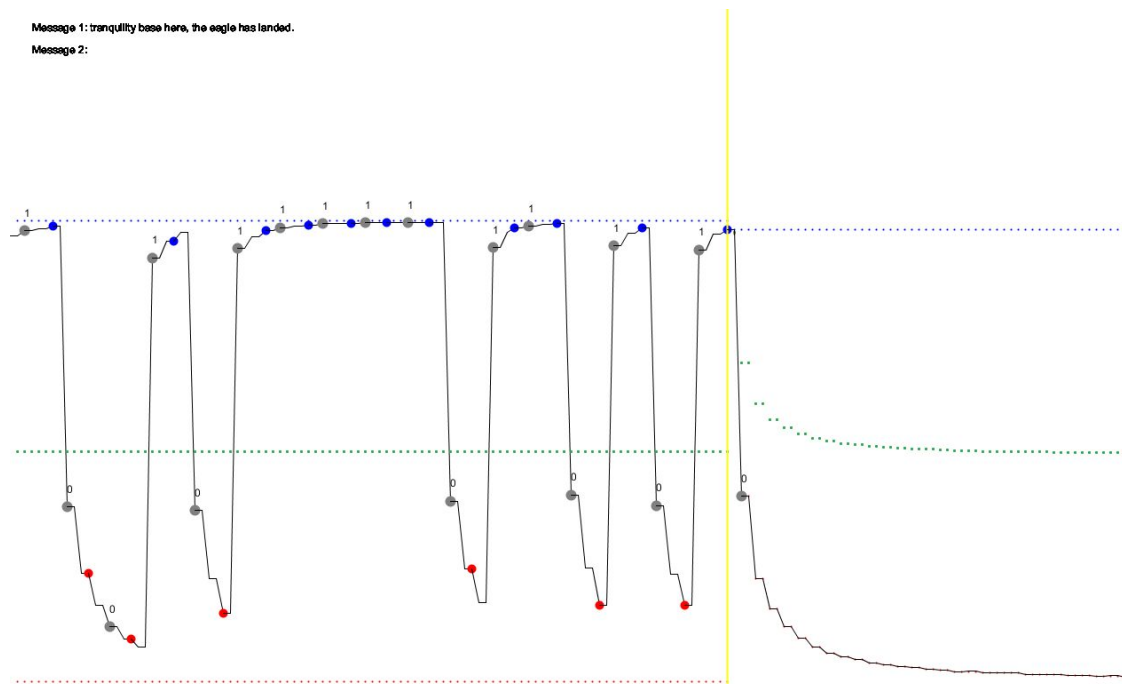
Bildquelle: elektronik-kompendium.de

A.d Aufbau bei der Datenerfassung



A.e Aufbau bei der Datenerfassung mit Relay**B. Originales Processing-Programm****B.a Programmcode**

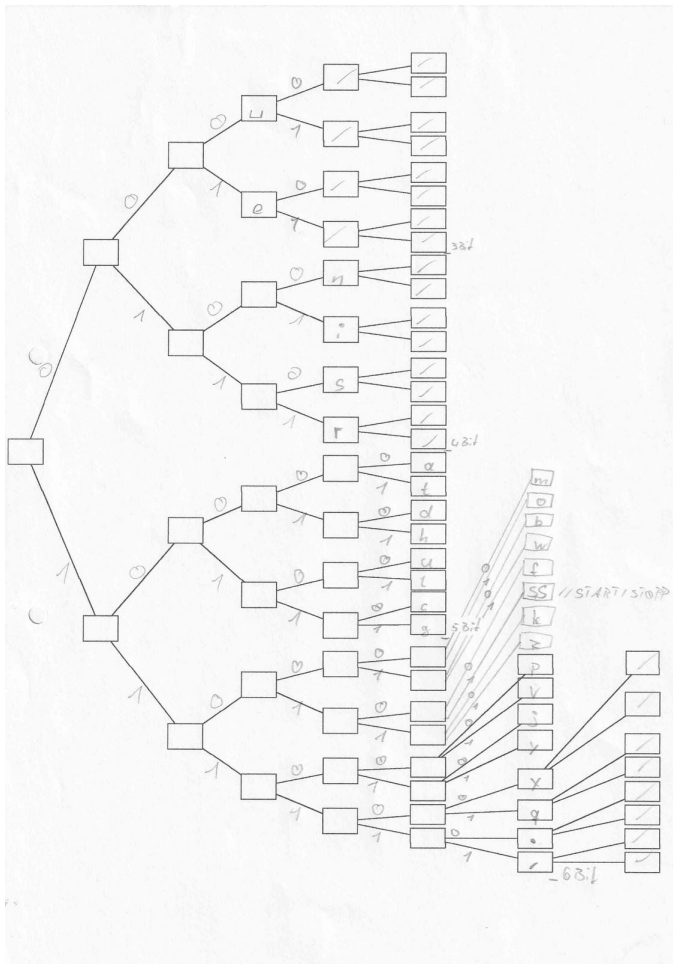
Der Programmcode ist aufgrund seiner Länge für den Anhang zu umfangreich, kann aber unter <https://github.com/dango301/Originales-Dateninterpretationsprogramm> eingesehen werden. Dort sind sämtliche sendende sowie empfangende Arduino- und Processing-Programme eingebunden.

B.b Oszillograph zu Beginn der Messung**B.c Oszillograph am Ende der Messung**

B.d Konsolenausgabe zu Beginn und Ende der Messung

1	tranquility base here, the eagle has la
11	0
110	01
1101	010
11010	0100
110101	tranquility base here, the eagle has lan
START OF MESSAGE	1
1	10
10	100
100	1001
1000	10010
10001	tranquility base here, the eagle has land
t	0
0	00
01	001
011	tranquility base here, the eagle has lande
0111	1
tr	10
1	100
10	1001
100	10010
1000	tranquility base here, the eagle has landed
10000	1
tra	11
0	111
01	1111
010	11111
0100	111110
tran	tranquility base here, the eagle has landed.
1	1
11	11
111	110
1111	1101
11110	11010
111101	110101
tranq	
1	
10	tranquility base here, the eagle has landed.
101	END OF MESSAGE
1010	
10100	
tranqu	=====

B.e Codierung der verlustfreien Komprimierung am Binärbaum



B.f Bitfehler und Problematik der der verlustfreien Übertragung

[illegible]

C. Funktionsweise von Leuchtdioden

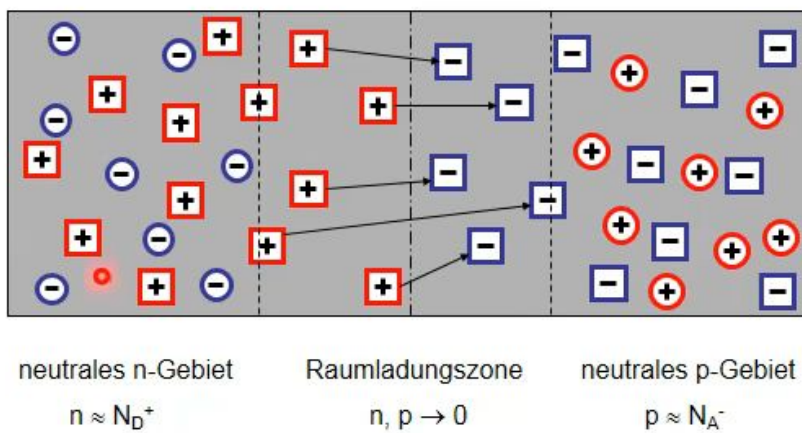
C.a Leitungsarten und Bändermodell

	Eigenleitung	Störstellenleitung	
		n-dotiert (Elektronenleitung)	p-dotiert (Löcherleitung)
Elemente	Gruppe IV vier Valenzelektronen: C, Si, Ge, Sn	Gruppe V fünf Valenzelektronen: N, P, As, Sb (Donatoren)	Gruppe III drei Valenzelektronen: B, Al, Ga, In (Akzeptoren)
Kristallgitter			
Bändermodell			

Bildquelle:

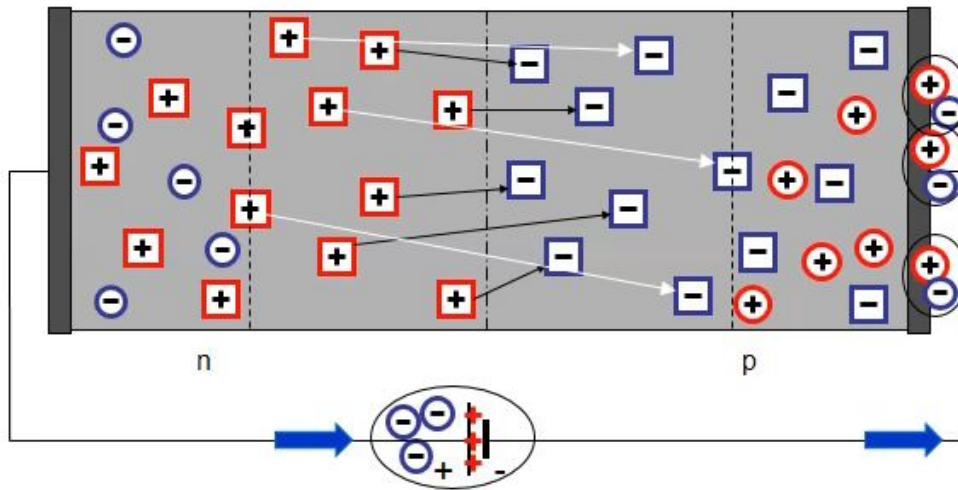
https://media.springernature.com/lw785/springer-static/image/chp%3A10.1007%2F978-3-662-49355-7_9/MediaObjects/47441_12_De_9_Fig40_HTML.gif (Stand: 15.03.2020)

C.b Elektrisches Feld einer Diode



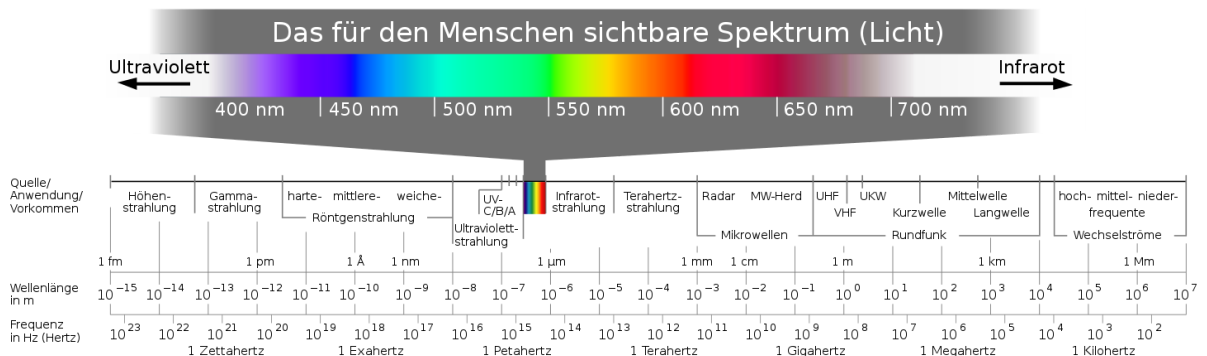
Bildquelle: Berger, 11:00

C.c Verhalten einer Diode in Sperrichtung



Bildquelle: Berger, 11:55

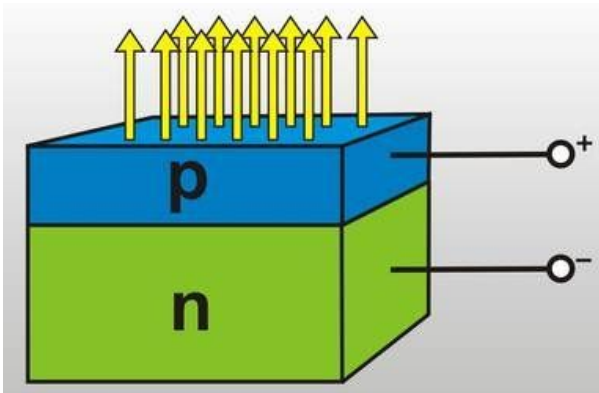
C.d Elektromagnetisches Farbspektrum



Bildquelle:

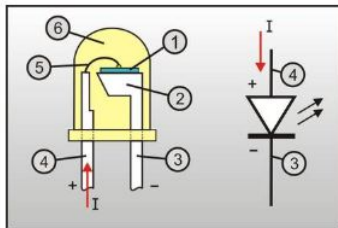
https://de.wikipedia.org/wiki/Elektromagnetisches_Spektrum#/media/Datei:Electromagnetic_spectrum_-de_c.svg (Stand: 12.02.2020)

C.e Emittierung von Licht an der LED



Bildquelle: conrad.de

C.f Aufbau einer LED



Aufbau einer Leuchtdiode am Beispiel einer bedrahteten LED.

1 Der **Halbleiterkristall (auch LED-Chip)** ist das Herzstück der LED und erzeugt Licht mittels physikalischer Prozesse.

2 Die **Reflektorwanne**, in die der LED-Chip eingebaut ist, erhöht sowohl Wirkungsgrad als auch Lichtausbeute.

3 Über die **Anode** fließt Strom in Richtung Halbleiterkristall.

4 Die **Kathode** ist mit der Reflektorwanne verbunden.

5 Ein **dünner Draht (Bond Draht)** verbindet den LED-Chip mit der Anode.

6 Eine **Kunststofflinse** umschließt den Aufbau und beeinflusst Strahlungswinkel und Lichtausbeute. Je nach Verwendungszweck ist sie oval, dreieckig, gewölbt oder abgeflacht.

Bildquelle: conrad.de

D. True Components LEDs

D.a Datenblatt der blauen TC LED

Absolute Maximum Rating (T_a=25°C)

Parameter	Symbol	Absolute Maximum Rating	Unit
Forward Pulse Current	I _{FPM}	70	mA
Forward Current	I _{FM}	30	mA
Reverse Voltage	V _R	5	V
Power Dissipation	P _D	140	mW
Operating Temperature	T _{opr}	-40~+80	°C
Storage Temperature	T _{stg}	-40~+100	°C
Soldering Heat (5s)	T _{sol}	260	°C

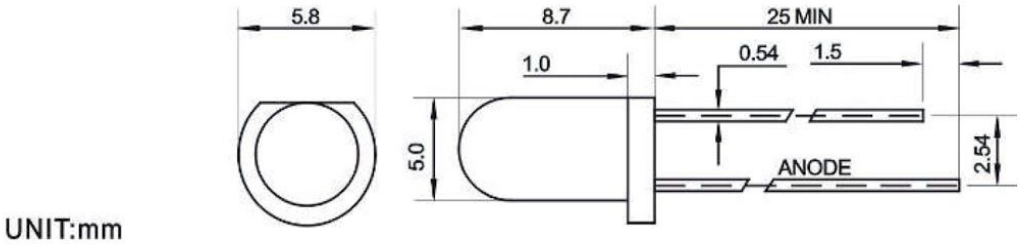
Electro-Optical Characteristics (T_a=25°C)

Parameter	Symbol	Min.	Typ.	Max.	Unit	Test Condition
Luminous Intensity	I _v	7000	---	9000	mcd	IF=20mA(Note1)
Viewing Angle	2θ _{1/2}	---	15	---	Deg	(Note 2)
Peak Emission Wavelength	λ _p	460	---	470	nm	IF=20mA
Spectral Line Half-Width	△λ	25	30	35	nm	IF=20mA
Forward Voltage	V _F	2.9	---	3.5	V	IF=20mA
Reverse Current	I _R	---	---	10	μA	VR=5V

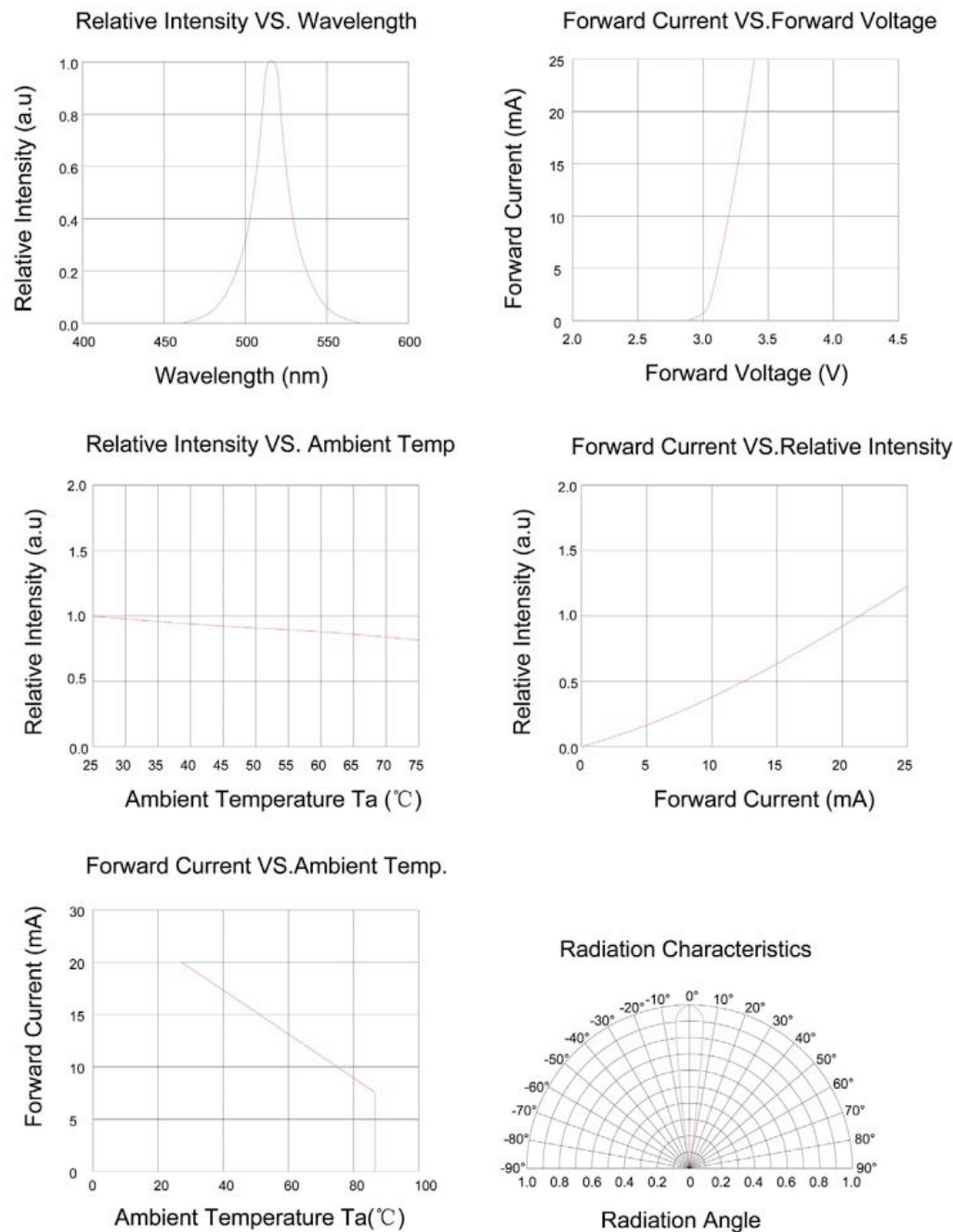
Device Selection Guide

	Chip		Lens Color
	Material	Emitted Color	
	InGaN	Blue	Water clear

Package Dimensions



Typical Electro-Optical Characteristics Curves

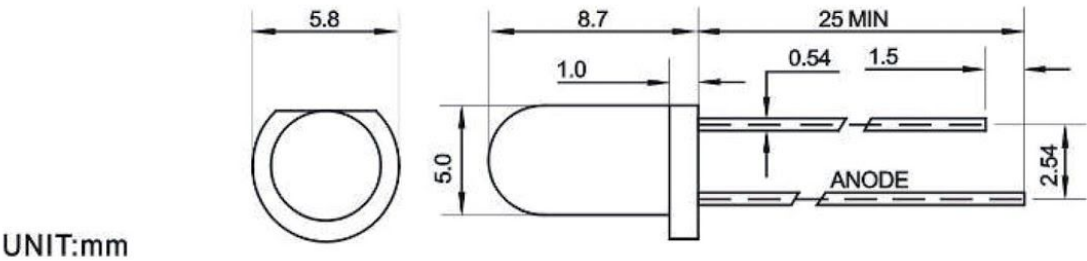


D.b Datenblatt der grünen TC LED

Device Selection Guide

	Chip		Lens Color
	Material	Emitted Color	
	AlGaInP	Green	Water clear

Package Dimensions



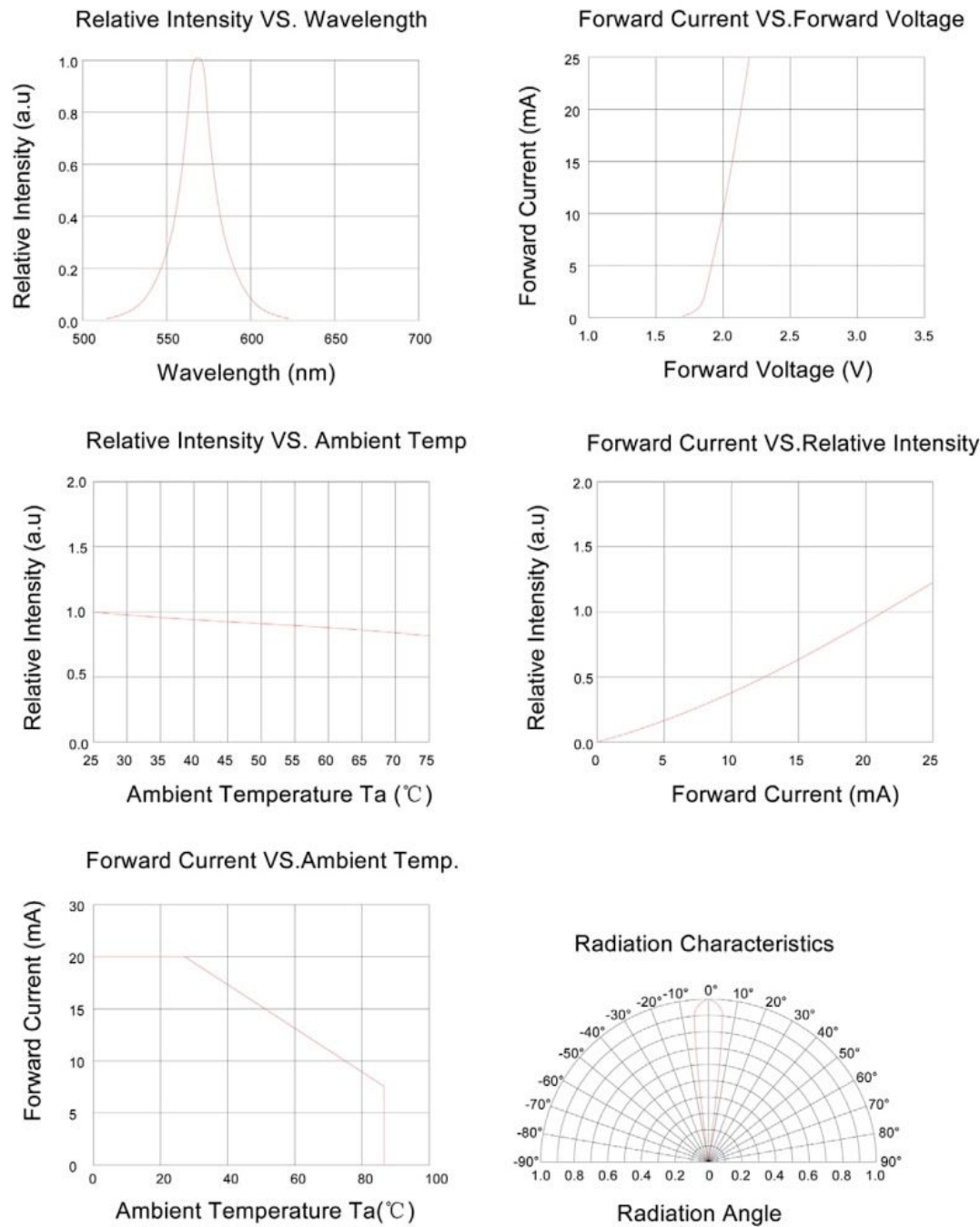
Absolute Maximum Rating (T_a=25°C)

Parameter	Symbol	Absolute Maximum Rating	Unit
Forward Pulse Current	I _{FPM}	100	mA
Forward Current	I _{FM}	30	mA
Reverse Voltage	V _R	5	V
Power Dissipation	P _D	90	mW
Operating Temperature	T _{opr}	-40~+80	°C
Storage Temperature	T _{stg}	-40~+100	°C
Soldering Heat (5s)	T _{sol}	260	°C

Electro-Optical Characteristics (T_a=25°C)

Parameter	Symbol	Min.	Typ.	Max.	Unit	Test Condition
Luminous Intensity	I _v	1000	---	2000	mcd	IF=20mA(Note1)
Viewing Angle	2θ _{1/2}	---	15	---	Deg	(Note 2)
Peak Emission Wavelength	λ _p	565	570	575	nm	IF=20mA
Spectral Line Half-Width	Δλ	15	20	25	nm	IF=20mA
Forward Voltage	V _F	1.9	---	2.3	V	IF=20mA
Reverse Current	I _R	---	---	10	μA	VR=5V

Typical Electro-Optical Characteristics Curves

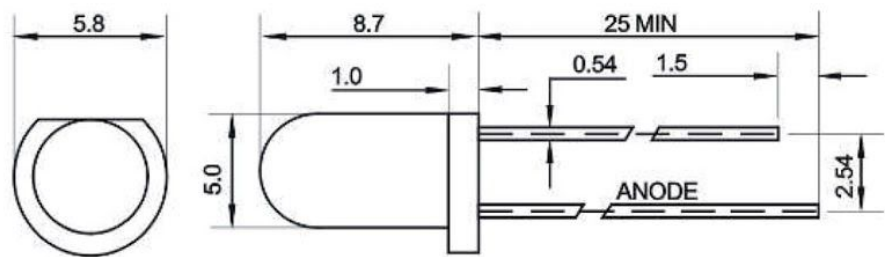


D.c Datenblatt der roten TC LED

Device Selection Guide

	Chip		Lens Color
	Material	Emitted Color	
	AlGaInP	Red	Water clear

Package Dimensions



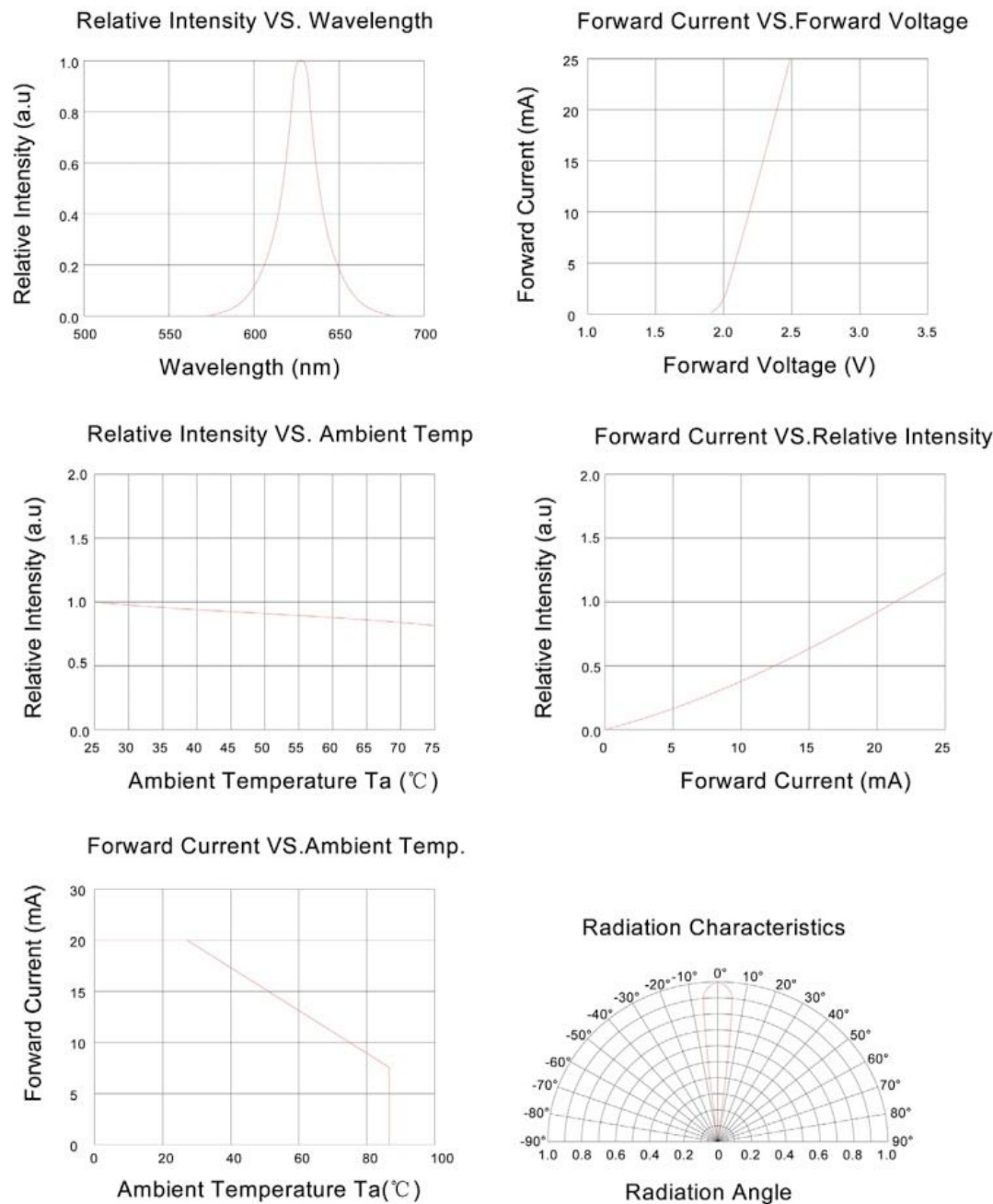
Absolute Maximum Rating (T_a=25°C)

Parameter	Symbol	Absolute Maximum Rating	Unit
Peak Forward Current (1/10 Duty Cycle, 0.1ms Pulse Width)	I _{FPM}	100	mA
Forward Current	I _{FM}	25	mA
Reverse Voltage	V _R	5	V
Power Dissipation	P _D	150	mW
Operating Temperature	T _{opr}	-30~+80	°C
Storage Temperature	T _{stg}	-40~+85	°C
Soldering Heat (3s)	T _{sol}	260	°C

Electro-Optical Characteristics (T_a=25°C)

Parameter	Symbol	Min.	Typ.	Max.	Unit	Test Condition
Luminous Intensity	I _v	15000	---	20000	mcd	IF=20mA(Note1)
Viewing Angle	2θ _{1/2}	---	15	---	Deg	(Note 2)
Peak Emission Wavelength	λ _p	620	---	630	nm	IF=20mA
Spectral Line Half-Width	Δλ	15	20	25	nm	IF=20mA
Forward Voltage	V _F	1.9	---	2.5	V	IF=20mA
Reverse Current	I _R	---	---	10	μA	VR=5V

Typical Electro-Optical Characteristics Curves



E. Datenerfassung- und Upload

E.a Programmcode

Der Programmcode ist aufgrund seiner Länge für den Anhang zu umfangreich, kann aber unter <https://github.com/dango301/Programme-der-Datenerfassung> eingesehen werden. Dort sind die Python- und Arduino-Programme eingebunden.

E.b Konsolenausgaben beim Daten-Upload für Arduino Standard LED

```
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\Paust\Code\Schule\Glasfaserkabel\Speed-Testing\Data_Processing> & C:\
pload.py

Connection with Port 'COM8' successfully established.
DATA RECEIVED after 15.983333333333333 minutes.
DATA PROCESSED

Do you want to proceed with the title 'Arduino Standard LED'? (y/n) y
Spreadsheet 'Arduino Standard LED' was created.

DATA UPLOAD STARTED
Worksheet 'Data Averages' was created.
Worksheet 'Data Averages' was updated.

Worksheet 'Leuchtdauer: 480.0ms' was created.
Worksheet 'Leuchtdauer: 480.0ms' was updated.

Worksheet 'Leuchtdauer: 400.0ms' was created.
Worksheet 'Leuchtdauer: 400.0ms' was updated.

Worksheet 'Leuchtdauer: 320.0ms' was created.
Worksheet 'Leuchtdauer: 320.0ms' was updated.

Worksheet 'Leuchtdauer: 240.0ms' was created.
Worksheet 'Leuchtdauer: 240.0ms' was updated.

Worksheet 'Leuchtdauer: 160.0ms' was created.
Worksheet 'Leuchtdauer: 160.0ms' was updated.

Worksheet 'Leuchtdauer: 80.0ms' was created.
Worksheet 'Leuchtdauer: 80.0ms' was updated.

PROGRAM FULLY EXECUTED after 18.966666666666665 minutes.
PS C:\Users\Paust\Code\Schule\Glasfaserkabel\Speed-Testing\Data_Processing> □
```

E.c Konsolenausgaben beim Daten-Upload für blaue TC LED

```
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\Paust\Code\Schule\Glasfaserkabel\Speed-Testing\Data_Processing> & C:/
pload.py

Connection with Port 'COM8' successfully established.
DATA RECEIVED after 17.816666666666666 minutes.
DATA PROCESSED

Do you want to proceed with the title 'TC LED Blue'? (y/n) y
Spreadsheet 'TC LED Blue' was created.

DATA UPLOAD STARTED
Worksheet 'Data Averages' was created.
Worksheet 'Data Averages' was updated.

Worksheet 'Leuchtdauer: 480.0ms' was created.
Worksheet 'Leuchtdauer: 480.0ms' was updated.

Worksheet 'Leuchtdauer: 400.0ms' was created.
Worksheet 'Leuchtdauer: 400.0ms' was updated.

Worksheet 'Leuchtdauer: 320.0ms' was created.
Worksheet 'Leuchtdauer: 320.0ms' was updated.

Worksheet 'Leuchtdauer: 240.0ms' was created.
Worksheet 'Leuchtdauer: 240.0ms' was updated.

Worksheet 'Leuchtdauer: 160.0ms' was created.
Worksheet 'Leuchtdauer: 160.0ms' was updated.

Worksheet 'Leuchtdauer: 80.0ms' was created.
Worksheet 'Leuchtdauer: 80.0ms' was updated.

PROGRAM FULLY EXECUTED after 23.366666666666667 minutes.
PS C:\Users\Paust\Code\Schule\Glasfaserkabel\Speed-Testing\Data_Processing> █
```


E.d Konsolenausgaben beim Daten-Upload mit Überschreibung

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell

Copyright (C) Microsoft Corporation. Alle Rechte vorbehalten.

Lernen Sie das neue plattformübergreifende PowerShell kennen - <https://aka.ms/powershell>

PS C:\Users\Paust\Code\Schule\Glasfaserkabel\Speed-Testing\Data_Processing> & C:\Users\Paust\Code\Schule\Glasfaserkabel\Speed-Testing\Data_Processing\upload.py

Connection with Port 'COM8' successfully established.

DATA RECEIVED after 14.216666666666667 minutes.

DATA PROCESSED

Do you want to proceed with the title 'Arduino Standard LED'? (y/n) n

Enter your new Title for the Document: TC LED Red [2]

Spreadsheet 'TC LED Red [2]' was opened.

DATA UPLOAD STARTED

Row for Duration 480.0 was overwritten in Worksheet 'Data Averages'.

Row for Duration 400.0 was overwritten in Worksheet 'Data Averages'.

Row for Duration 320.0 was overwritten in Worksheet 'Data Averages'.

Row for Duration 240.0 was overwritten in Worksheet 'Data Averages'.

Row for Duration 160.0 was overwritten in Worksheet 'Data Averages'.

Row for Duration 80.0 was overwritten in Worksheet 'Data Averages'.

Worksheet 'Data Averages' was updated.

Worksheet 'Leuchtdauer: 480.0ms' was overwritten.

Worksheet 'Leuchtdauer: 480.0ms' was updated.

Worksheet 'Leuchtdauer: 400.0ms' was overwritten.

Worksheet 'Leuchtdauer: 400.0ms' was updated.

Worksheet 'Leuchtdauer: 320.0ms' was overwritten.

Worksheet 'Leuchtdauer: 320.0ms' was updated.

Worksheet 'Leuchtdauer: 240.0ms' was overwritten.

Worksheet 'Leuchtdauer: 240.0ms' was updated.

Worksheet 'Leuchtdauer: 160.0ms' was overwritten.

Worksheet 'Leuchtdauer: 160.0ms' was updated.

Worksheet 'Leuchtdauer: 80.0ms' was overwritten.

Worksheet 'Leuchtdauer: 80.0ms' was updated.

PROGRAM FULLY EXECUTED after 41.6 minutes.

PS C:\Users\Paust\Code\Schule\Glasfaserkabel\Speed-Testing\Data_Processing> █

E.e Konsolenausgaben beim Daten-Upload mit Ergänzung

```
PS C:\Users\Paust\Code\Schule\Glasfaserkabel\Speed-Testing\Data_Processing> & C:\Users\Paust\Code\Schule\Glasfaserkabel\Speed-Testing\Data_Processing\upload.py

Connection with Port 'COM8' successfully established.
DATA RECEIVED after 4.783333333333333 minutes.
DATA PROCESSED

Do you want to proceed with the title 'Arduino LED'? (y/n) y
Spreadsheet 'Arduino LED' was opened.

DATA UPLOAD STARTED
Row for Duration 120.0 was added to Worksheet 'Data Averages'.
Row for Duration 100.0 was added to Worksheet 'Data Averages'.
Row for Duration 90.0 was added to Worksheet 'Data Averages'.
Row for Duration 85.0 was added to Worksheet 'Data Averages'.
Row for Duration 80.0 was added to Worksheet 'Data Averages'.
Row for Duration 75.0 was added to Worksheet 'Data Averages'.
Row for Duration 70.0 was added to Worksheet 'Data Averages'.
Row for Duration 65.0 was added to Worksheet 'Data Averages'.
Row for Duration 60.0 was added to Worksheet 'Data Averages'.
Worksheet 'Data Averages' was updated.

Worksheet 'Leuchtdauer: 120.0ms' was created.
Worksheet 'Leuchtdauer: 120.0ms' was updated.

Worksheet 'Leuchtdauer: 100.0ms' was created.
Worksheet 'Leuchtdauer: 100.0ms' was updated.

Worksheet 'Leuchtdauer: 90.0ms' was created.
Worksheet 'Leuchtdauer: 90.0ms' was updated.

Worksheet 'Leuchtdauer: 85.0ms' was created.
Worksheet 'Leuchtdauer: 85.0ms' was updated.

Worksheet 'Leuchtdauer: 80.0ms' was created.
Worksheet 'Leuchtdauer: 80.0ms' was updated.

Worksheet 'Leuchtdauer: 75.0ms' was created.
Worksheet 'Leuchtdauer: 75.0ms' was updated.

Worksheet 'Leuchtdauer: 70.0ms' was created.
Worksheet 'Leuchtdauer: 70.0ms' was updated.

Worksheet 'Leuchtdauer: 65.0ms' was created.
Worksheet 'Leuchtdauer: 65.0ms' was updated.

Worksheet 'Leuchtdauer: 60.0ms' was created.
Worksheet 'Leuchtdauer: 60.0ms' was updated.

PROGRAM FULLY EXECUTED after 5.783333333333333 minutes.
PS C:\Users\Paust\Code\Schule\Glasfaserkabel\Speed-Testing\Data_Processing> 
```

F. Datensätze

Es ist nicht möglich die gesamten Datensätze aufgrund ihrer Vielzahl und Größe einzubinden. Daher werden hier nur die Balkendiagramme der Durchschnittswerte aller jeweiligen Leuchtdioden sowie im Fließtext ausgewählte Helligkeitskurven aufgeführt. Daher sind die numerische Werte in den Tabellen, zusammen mit ihren Graphen, öffentlich zur Betrachtung unter freigegeben.

Arduino Standard LED:

<https://docs.google.com/spreadsheets/d/19x5HEz9P4Ueiv4LCMNBh6H-Px7PK9qUuWwhAO77HOA/edit?usp=sharing>

TC LED Blue:

<https://docs.google.com/spreadsheets/d/1YNaNFaw-8JelCuI7LRExAZRBekJntvZgldP9HqouJZA/edit?usp=sharing>

TC LED Green:

<https://docs.google.com/spreadsheets/d/18egQrmvfD8haDFTcDsmkCjueUQchOEDEtnh7UoKOyt8/edit?usp=sharing>

TC LED Red:

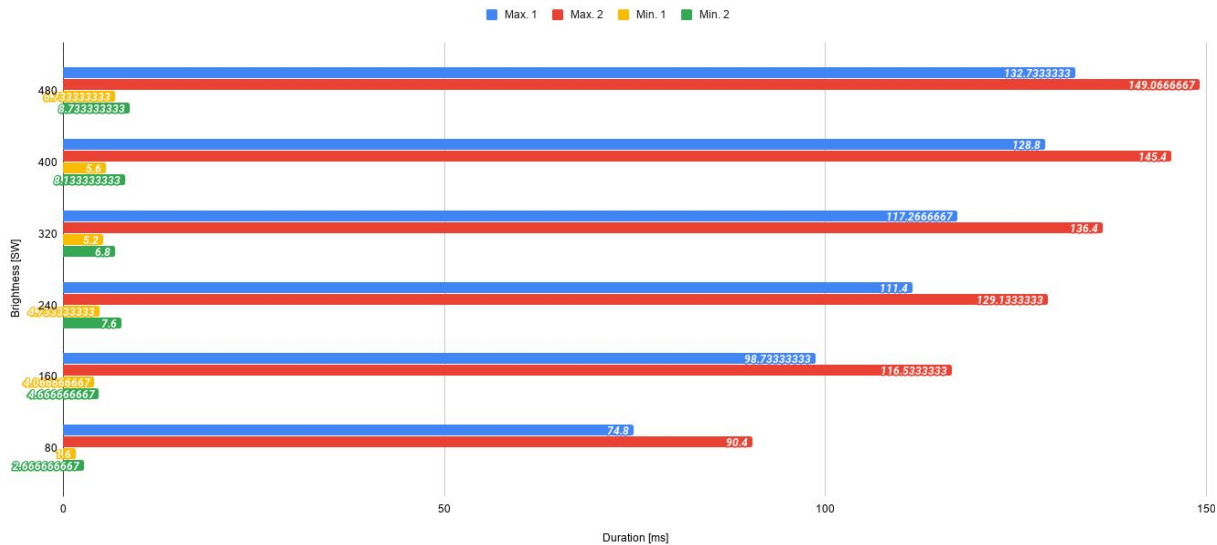
<https://docs.google.com/spreadsheets/d/17sHtoM12D5H6DvVnA3jSZMGxUAHOcgQldPJzFkLlywl/edit?usp=sharing>

TC LED Red [2] (Zweiter Datensatz zur Überprüfung):

<https://docs.google.com/spreadsheets/d/1t4sEGXE7xh8xc0RqN9qFM-wzzV9BufAv3I-rUg5fDdl/edit?usp=sharing>

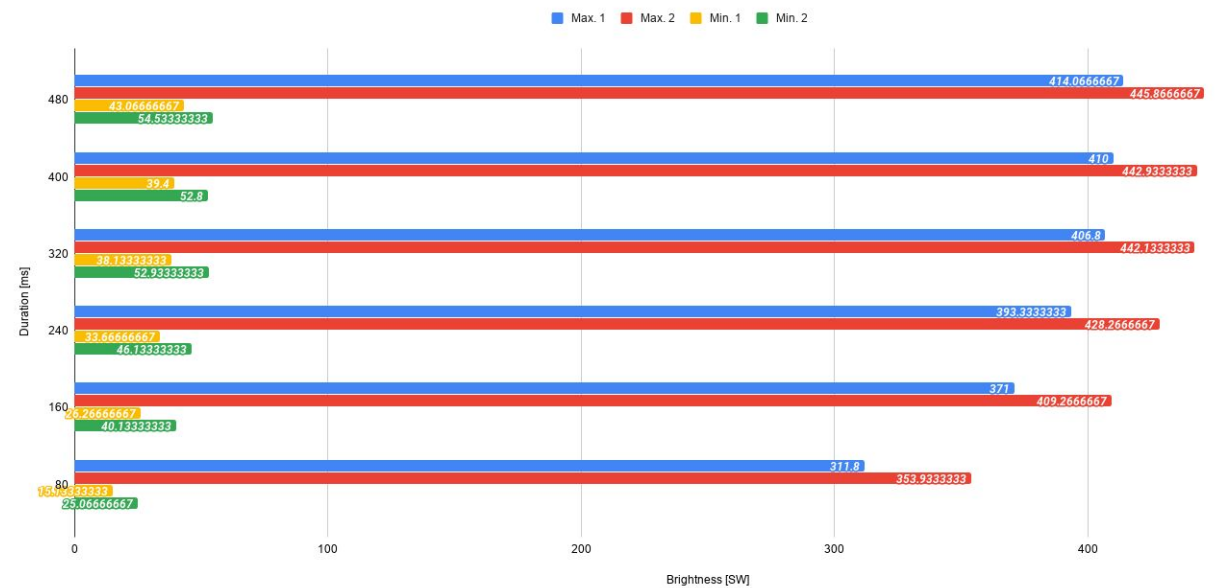
F.a Data-Average für Arduino Standard LED

Arduino Standard LED, Data-Average

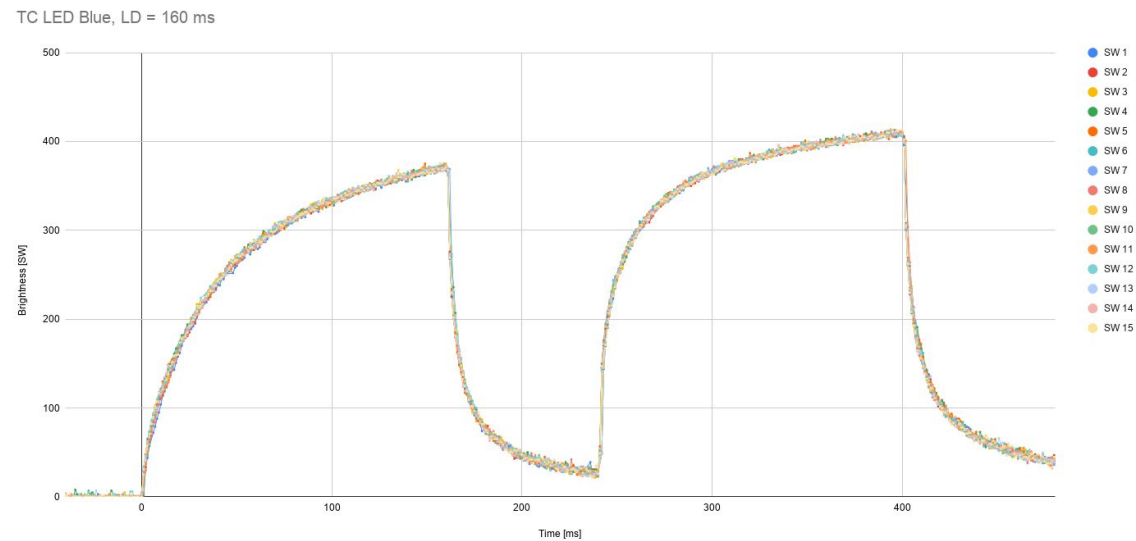


F.b Data-Average für blaue TC LED

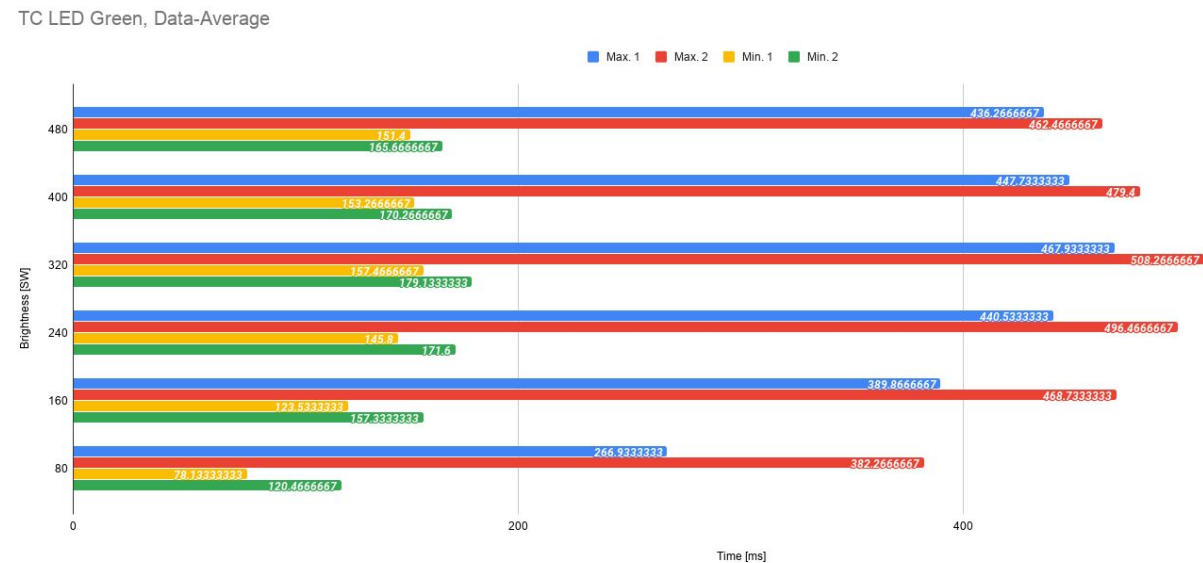
TC LED Blue, Data-Average



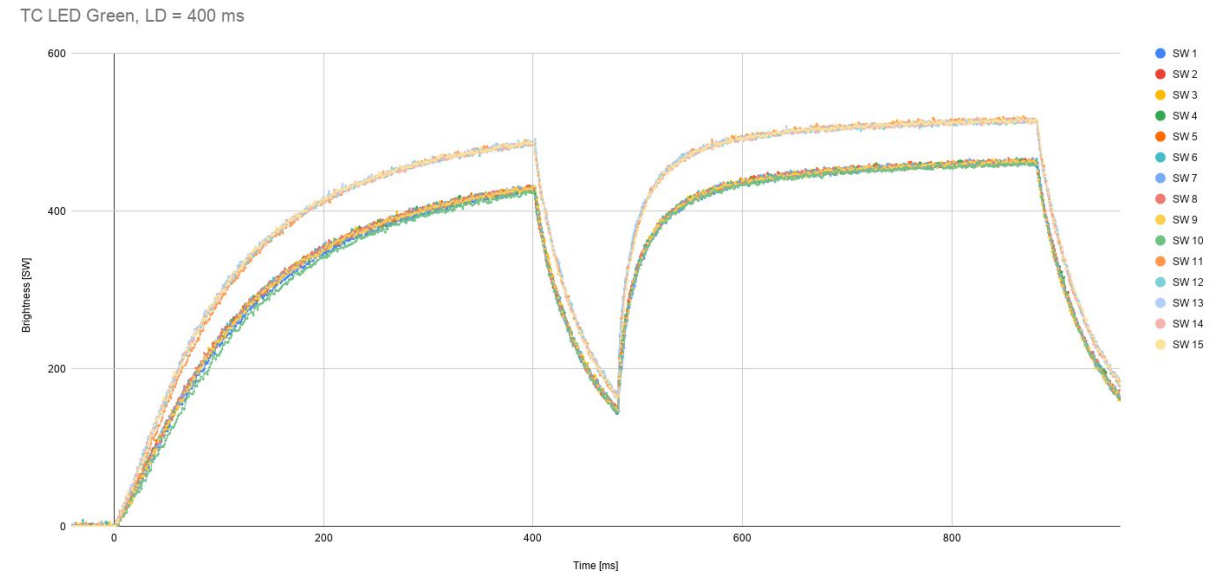
F.c Helligkeitskurve TC LED Blue, LD = 160 ms



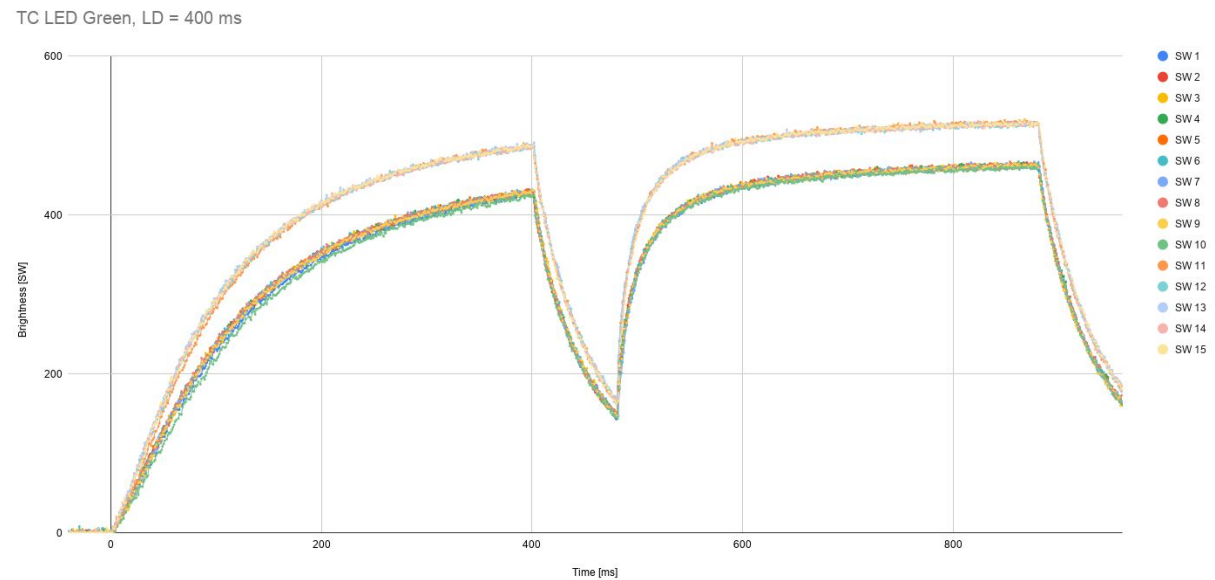
F.d Data-Average für grüne TC LED



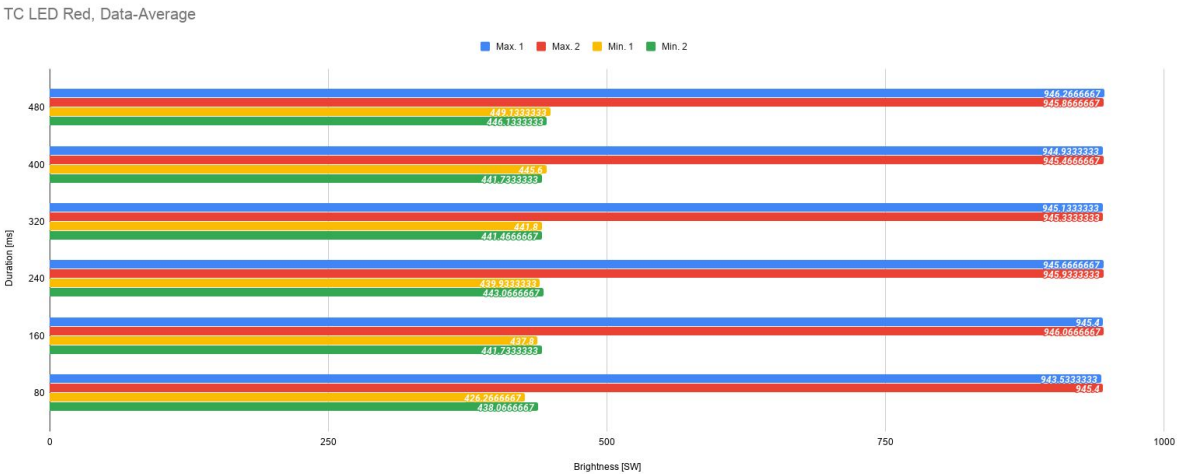
F.e Helligkeitskurve TC LED Green, LD = 400 ms



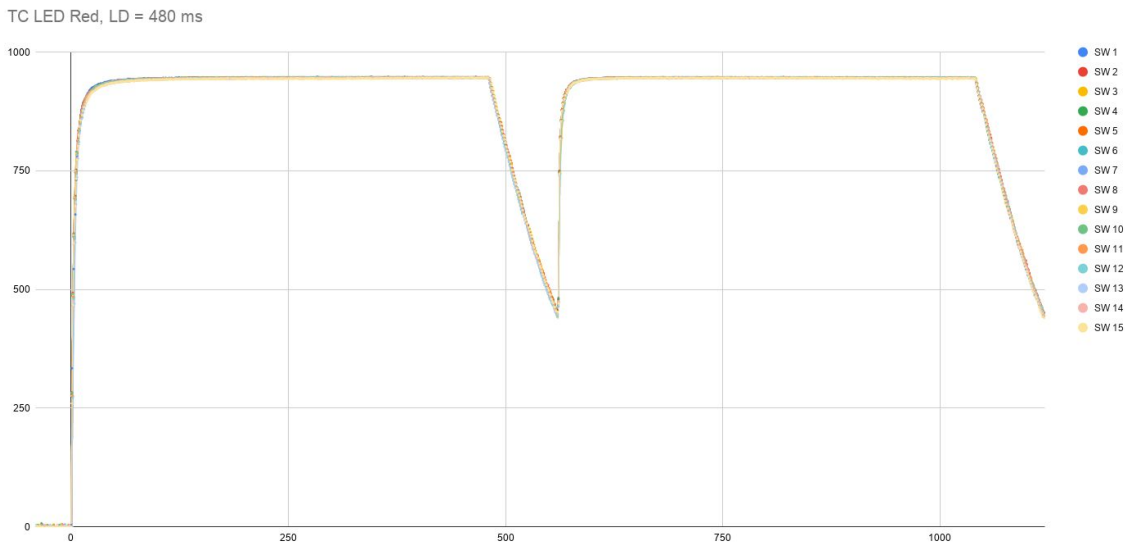
F.f Helligkeitskurve TC LED Green, LD = 80 ms



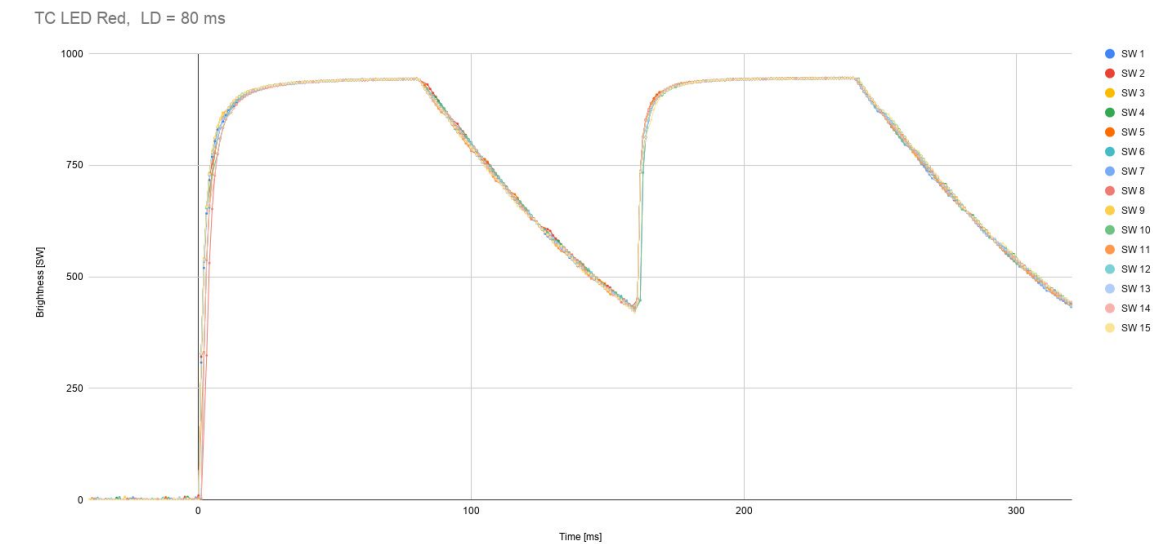
F.g Data-Average für rote TC LED



F.h Helligkeitskurve TC LED Red, LD = 480 ms



F.i Helligkeitskurve TC LED Red, LD = 80 ms



G. Fehlerquote

G.a Konsolenausgabe für fehlerfreie Übertragung

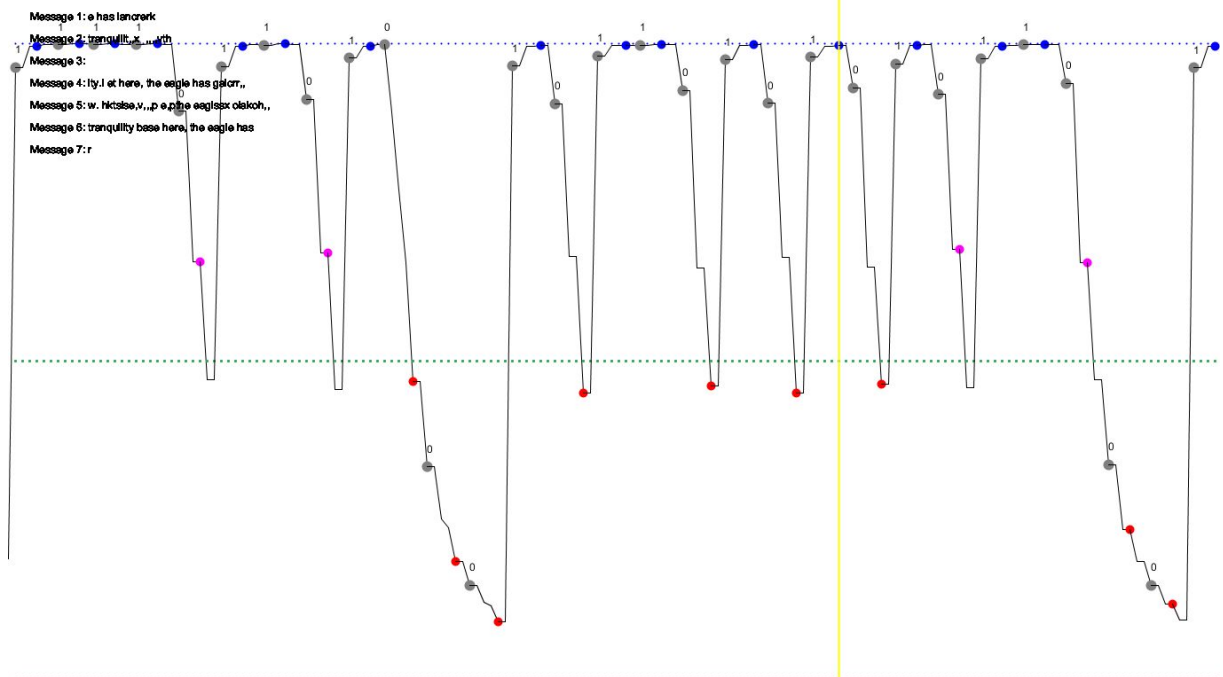
```
tranquility base here, the eagle has landed.
1
11
111
1111
11111
111110
tranquility base here, the eagle has landed.
1
11
110
1101
11010
110101

tranquility base here, the eagle has landed.
END OF MESSAGE

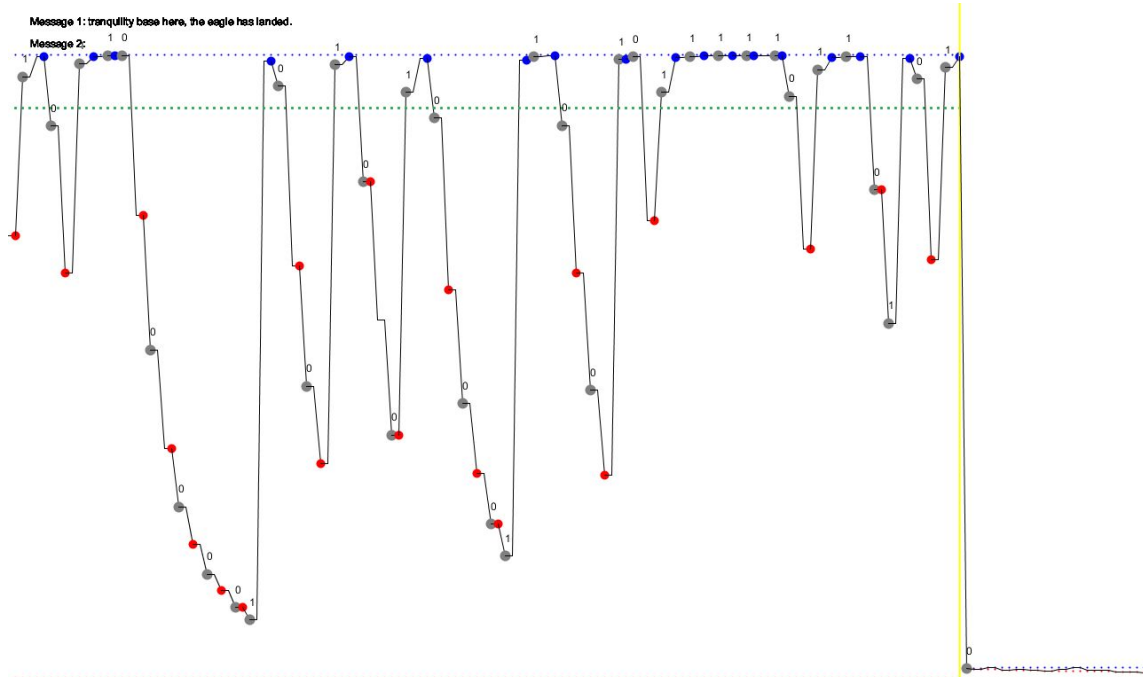
Fehlerrate: 0
110101100010111100001001111010100010110101011000111011000110010000011000100110010111001111110001000110011001000001100010111010100100010011100000110000101011000001001001000110010101
110101100010111100001001111010100010110101011000111011000110010000011000100110010111110001000110011001000001100010111010100100010011100000110000101011000001001001000110010101
=====
```


H. Optimierung mit roter TC LED

H.a Problematische Übertragung vor Anpassung des Programms, $T = 125\text{ ms}$



H.b Erfolgreiche Übertragung nach Anpassung des Programms, $T = 125\text{ ms}$



H.c Konsolenausgaben bei Identifizierung eines Übertragungsfehlers

```

1
10
100
1000
10000
tra
0
Negative Bits transmitted for 366.0 ms.
Negative Bits transmitted for 5 Periodes.
01
Negative Bits transmitted for 17.0 ms.
Negative Bits transmitted for 0 Periodes.
010
Negative Bits transmitted for 132.0 ms.
Negative Bits transmitted for 2 Periodes.
1
11
111
1111
11110
Negative Bits transmitted for 68.0 ms.
Negative Bits transmitted for 1 Periodes.
111101
tranq
1
10
Negative Bits transmitted for 67.0 ms.
Negative Bits transmitted for 1 Periodes.
101
Negative Bits transmitted for 16.0 ms.
Negative Bits transmitted for 0 Periodes.
1011
10110
tranqc
0
Negative Bits transmitted for 199.0 ms.
Negative Bits transmitted for 3 Periodes.

```

H.d Konsolenausgaben bei Korrektur von Übertragungsfehlern

```

1
11
Negative Bits transmitted for 55.0 ms.
Negative Bits transmitted for 1 T.
CAUGHT!
110
1101
Negative Bits transmitted for 18.0 ms.
Negative Bits transmitted for 0 T.
Negative Bits transmitted for 34.0 ms.
Negative Bits transmitted for 1 T.
CAUGHT!
11010
110101
START OF MESSAGE

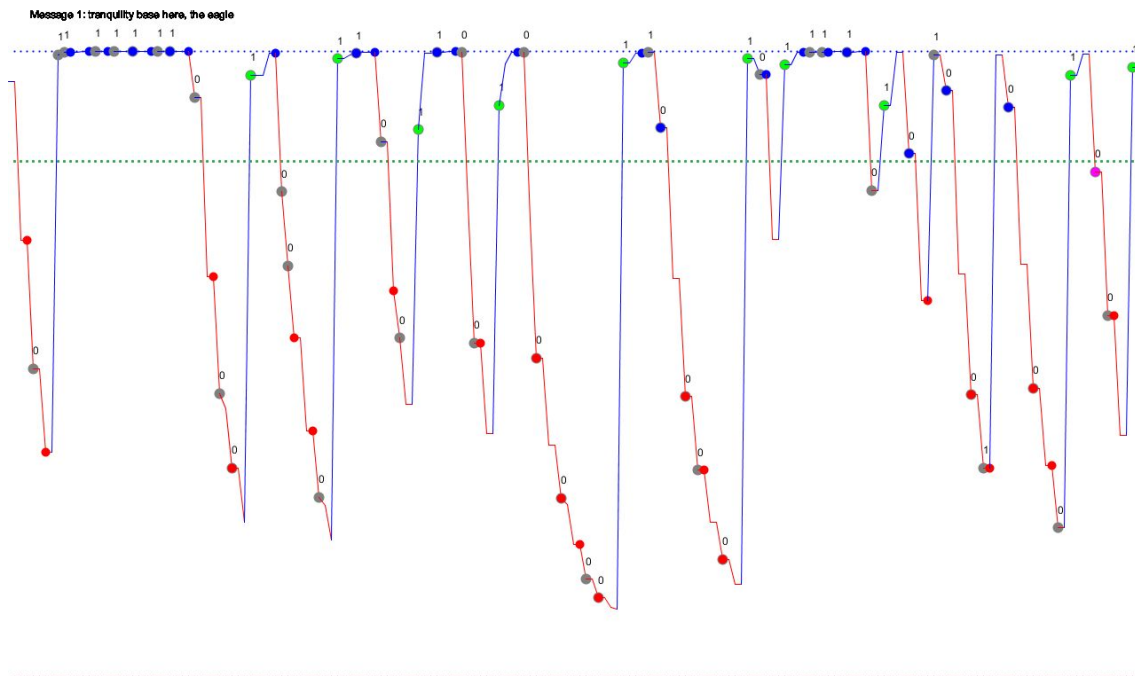
```

```

0
00
Negative Bits transmitted for 199.0 ms.
Negative Bits transmitted for 4 T.
CAUGHT!
000
tranquility base here, the eagle has

```

H.e Am Oszillograph markierte Steigungen und Fehlerkorrekturen



H.f Optimierter Processing-Programmcode

Der Programmcode ist aufgrund seiner Länge für den Anhang zu umfangreich, kann aber unter <https://github.com/dango301/Optimiertes-Dateninterpretationsprogramm> eingesehen werden. Dort sind die Änderungen am Processing-Programm eingebunden.

Literaturverzeichnis

Fachliteratur

- | | |
|--------------------------|---|
| Breed, Garry | Bit Error Rate: Fundamental Concepts and Measurement Issues,
http://highfrequencyelectronics.com/Jan03/HFE0103_Tutorial.pdf (Stand: 20.03.2020), Summit TechnicalMedia, LLC, 2003 |
| Kuchling, Horst: | Taschenbuch der Physik, 16. Auflage, München / Wien: Fachbuchverlag Leipzig im Carl Hanser Verlag, 1999 |
| Liersch, Manfred: | Chemie 1 kurz & klar, 3. Auflage, Donauwörth: Auer Verlag GmbH, 2001 |
| Winzker, Marco | Elektronik für Entscheider, 1. Auflage, Wiesbaden: Friedr. Vieweg & Sohn Verlag, 2008 |

Internetadressen

- | | |
|---|---|
| ARDUINO UNO REV3 Tech Specs | https://store.arduino.cc/arduino-uno-rev3 (Stand: 18.03.2020) |
| Glasfaser / Lichtwellenleiter (LWL) | https://www.elektronik-kompodium.de/sites/kom/0301282.htm (Stand: 29.02.2020) |
| Grundlagen der Halbleiter-Bauelemente
(Präsentation von Michael Berger an der Fachhochschule Westküste) | https://youtu.be/Q9LrPK5W8I0 (Stand: 20.03.2020) |
| LED » Die Leuchtdiode kurz erklärt
(Produktseite der genutzten LEDs) | https://www.conrad.de/de/ratgeber/technik-einfach-erklart/led-leuchtdiode.html (Stand: 23.03.2020) |
| LED - Leuchtdioden | https://www.elektronik-kompodium.de/sites/bau/0201111.htm (Stand: 29.02.2020) |
| Praktische Veranstaltungskunde: „Weil Induktivitäten sich verspäten“ | https://www.production-partner.de/basics/praktische-veranstaltungskunde-weil-induktivitaeten-sich-verspaeten (Stand: 19.03.2020) |
| Processing: frameRate() | https://processing.org/reference/frameRate_.html (Stand: 23.03.2020) |
| Python | http://www.institut.de/blog/glossar/python (Stand: 17.03.2020) |
| Skalen, Skalenniveaus | https://matheguru.com/stochastik/skalen-skalenniveaus.html (Stand: 17.03.2020) |