

SEMINARARBEIT

Rahmenthema des wissenschaftspropädeutischen Seminars:

Light Amplification
by Stimulated Emission of Radiatio

Leitfach: Physik

Thema der Arbeit:

***Kommunikation mit Light Amplification
by Stimulated Emission of Radiation***

Verfasser:

Sebastian von Helmersen

Kursleiter:

Florian Soutschek

Abgabetermin:

07.11.2023

Bewertung	Note	Notenstufe in Worten	Punkte		Punkte
schriftliche Arbeit				x 3	
Abschlusspräsentation				x 1	
				Summe:	
				Summe : 2 (gerundet)	

Datum und Unterschrift des Kursleiters

Inhaltsverzeichnis

1	Einleitung	3
2	Theorie	5
2.1	Grundlagen der Datenübertragung	5
2.1.1	Aufbau eines Systems	6
2.1.2	Signale	7
2.1.3	Übertragung	16
2.2	Laser als Datenübertragungsweg	20
3	Versuch	24
3.1	Aufbau	24
3.1.1	Sender	25
3.1.2	Empfänger	29
3.1.3	Übertragungsprotokoll	34
3.2	Durchführung	35
3.3	Ergebnisse	36
4	Zusammenfassung	39
	Abkürzungsverzeichnis	41
	Abbildungsverzeichnis	43
	Tabellenverzeichnis	44
	Literaturverzeichnis	47
	Schülererklärung	48

1 | Einleitung

Getreu des Themas „Light Amplification by Stimulated Emission of Radiation (L.A.S.E.R.)“ des wissenschaftspropädeutischen Seminars Physik, befasst sich diese Seminararbeit mit diesem als innovativem Datenübertragungsweg. In Zuge dessen gilt es herauszufinden, worin Vor- und Nachteile dieser neuen Technologie liegen.

Auslöser hierfür ist die neu gewonnene Relevanz dieses Themas. So will beispielsweise *SpaceX* durch die Verwendung von L.A.S.E.R. Datenlinks mit *Terabit* Bandbreiten zwischen ihren fast viertausend funktionsfähigen Satelliten ermöglichen [Starlink, 2023]. Solch eine Umsetzung hätte revolutionäre Folgen für die Internetkommunikation und würde ganz neue Ansätze der internationalen niedrig Latenz Kommunikation ermöglichen.

Da dieses Ziel sehr ambitioniert ist und der Umfang dieser Arbeit sowohl finanziell als auch zeitlich begrenzt ist, wird sich mit einer abstrahierten Auffassung von L.A.S.E.R. Kommunikation befasst. Der Forschungsansatz besteht in der Beschreibung, Einrichtung und Analyse eines monodirektionalen L.A.S.E.R.-Datenübertragungssystems. Es handelt sich dabei also mehr um eine amateurhafte, statt einer großen kommerziellen Umsetzung. So richtet sich der genannte Forschungsansatz auch eher an Hobbyisten und berücksichtigt auch hier interessante Punkte wie Kosten und Einrichtungskomplexität.

Die Arbeit teilt sich hierbei in zwei große Hauptteile. Ersterer ist der theoretische Teil, der neben dem allgemeinen Aufbau der Datenübertragungssysteme auch auf die physikalischen Eigenschaften des L.A.S.E.R. eingeht. Der zweite Teil befasst sich mit dem Praxisteil und behandelt Aufbau, Durchführung und gesammelte Ergebnisse. Nach sepa-

rater Betrachtung der beiden Perspektiven wird in 4. *Zusammenfassung* das abschließende Fazit zu dem oben beschriebenen Forschungsansatz gezogen und eine Einordnung der Technologie in potenzielle Anwendungsbereiche geboten.

2 | Theorie

Da für den Aufbau des in *3. Versuch* beschriebenen Versuchs viel theoretischer Hintergrund benötigt wird, ist es wichtig diese voranzustellen. Dabei wird zwischen zwei großen Teilbereichen unterschieden. *2.1. Grundlagen der Datenübertragung* beschäftigt sich mit der allgemeinen Theorie der Datenübertragung in der Informations-Technologie (IT) während *2.2. Laser als Datenübertragungsweg* mehr auf die physikalischen Eigenschaften der L.A.S.E.R. eingeht.

2.1 Grundlagen der Datenübertragung

Im Jahre 1989 standen Tim Berners-Lee und seine Kollegen der Forschungseinrichtung CERN vor einem Problem: Dank der Kommerzialisierung der Computer war es ihnen möglich ihre Forschungsergebnisse noch schneller und präziser zu finalisieren. Wollten sie diese Forschungsergebnisse nun allerdings ihren Kollegen zur Verfügung stellen blieb ihnen nur der Weg über ein physisches Trägermittel wie eine Floppy Disk oder ein Blatt Papier [CERN, 2023]. Dieser Prozess war kostspielig und vor allem zeitintensiv. Das Problem war, dass jedes der Computersysteme in sich geschlossen, also isoliert, agierte. Um dieses Problem zu lösen, entwickelten Herr Berners-Lee und seine Kollegen den Vorgänger eines Netzwerkes, das sich in den folgenden Jahren zu dem weiterentwickelte, was wir heute das Internet nennen [Berners-Lee, 1989]. Diese Anekdote zeigt wie wichtig und – im Falle des Internets – weltbewegend die Verknüpfung von Computersystemen ist. Ermöglicht werden diese Netzwerke durch Daten Kommunikation. Hierbei handelt es sich um die vielfache Abfolge von Datenübertragungen zwischen mehreren Systemen. Im Folgenden wird das Konzept dieser Übertragungen, beginnend mit dem schematischen Aufbau, behandelt.

2.1.1 Aufbau eines Systems

Der allgemeine Aufbau jeder Datenübertragungseinrichtung wurde 1966 in der DIN 44302 [Deutsches Institut für Normung e.V., 1987] festgelegt. Da nicht alle Aspekte der Norm Relevanz für diese Arbeit aufweisen wird im Folgenden nicht auf all diese eingegangen. Die Norm klassifiziert Systeme in einem Datenkommunikationsnetzwerk als *Datenstationen*. Jede Datenstation ist - abhängig von der gewählten Betriebsart - in der Lage sowohl Daten zu empfangen, als auch zu senden. Folglich ist keine unabhängige Betrachtung dieser beiden Vorgänge nötig da sie in ihrer Reihenfolge der einzelnen Schritte invertiert zueinander ablaufen. Das folgende Kapitel beschränkt sich somit auf den Sendebetrieb.

Datenstation						Übertragungsleitung
DE-Einrichtung			DÜ-Einrichtung			
Eingabe- und/oder Ausgabe- und/oder Rechen- und/oder Leitwerk und/oder Speicher	Fernbetriebs-einheit	Fehler-schutz- und/oder Synchro-nisier-einheit	Prüf-einheit	Signal-umsetzer	Anschalt-einheit	
	Überwachungsteil Datenaufbereitungsteil Stationskennungsteil			Sendeteil Empfangsteil Schaltteil		

Abbildung 2.1: Schematische Darstellung der DIN 44302 [Autoren der IT, 1966]

Schematisch ist der Prozess des Sendens aus Abbildung 2.1 von links nach rechts abzulesen. In der Dateneneinrichtung (DE-Einrichtung) liegen die zu übertragenden Daten in ihrer Ursprungsform vor. Hierbei ist auch die Signalart irrelevant. So kann es sich um analoge oder digitale Daten handeln (vgl. 2.1.2. *Signale*). Es ist ihre Aufgabe, diese Daten über die Schnittstelle (DIN 66020 [Deutsches Institut für Normung e.V., 1966]) der Datenübertragungseinrichtung (DÜ-Einrichtung) zur Verfügung zu stellen. Wie in

2.1.3. Übertragung erläutert, ist für die korrekte Übertragung der Daten das Einhalten der Schrittgeschwindigkeit sehr wichtig. Ähnlich verhält es sich auch bei diesen internen Komponenten. So besitzen sowohl die DE-Einrichtung als auch die DÜ-Einrichtung eine Synchronisierereinheit die sicherstellt, dass beide mit der gleichen Schrittgeschwindigkeit arbeiten und die Übergabe der Daten an der Schnittstelle so fehlerfrei abläuft. Je nach Art der Schnittstelle kann es auch vorkommen, dass die DE-Einrichtung die Rohdaten umwandeln muss. Allerdings ist dies aufgrund der einheitlichen Verwendung digitaler Signale (vgl. *2.1.2. Signale*) zur Zeit der Veröffentlichung meist kein Problem.

Die DÜ-Einrichtung hat schließlich die Aufgabe, die im kompatiblen Format der Schnittstelle vorliegenden Daten über die Übertragungsleitung zu übertragen. Hierfür müssen die Daten erst wie in *2.1.3. Übertragung* beschrieben im Signalumsetzer zu Signalen aufbereitet werden. Abschließend verfügt die DÜ-Einrichtung über eine Anschalteinheit. Diese hat die Aufgabe die tatsächliche physische Übertragungsleitung zu bedienen und das Signal zu modulieren (vgl. *2.1.2. Signale*).

Ein Beispiel für den Aufbau eines solchen Systems ist eine Verbindung zweier Computer über ein Telefonmodem, wie es in den 1980er Jahren üblich war (vgl. Abbildung 2.2).

Der Computer ist die DE-Einrichtung und stellt alle Daten zur Verfügung. So wandelt er beispielsweise in die Tastatur eingegebenen Text in digitale Daten um und übergibt diese über die serielle Schnittstelle an das Telefon Modem. Dieses wandelt, in seiner Funktion als DÜ-Einrichtung, die Daten in analoge Signale um und überträgt diese als Analoges elektrisches Signal über die Übertragungsleitung, hier: eine Telefonleitung. Das zweite Modem empfängt dieses Signal wiederum, wandelt es in ein digitales Signal um und überträgt es über die Schnittstelle an den zweiten Computer, der die Daten dort nun abspeichert.

2.1.2 Signale

Unabdinglich für die Datenübertragung ist die Art und Weise wie diese übermittelt werden. Hierbei bedient man sich aus der Physik bekannten Informationsträgern [Beucher,

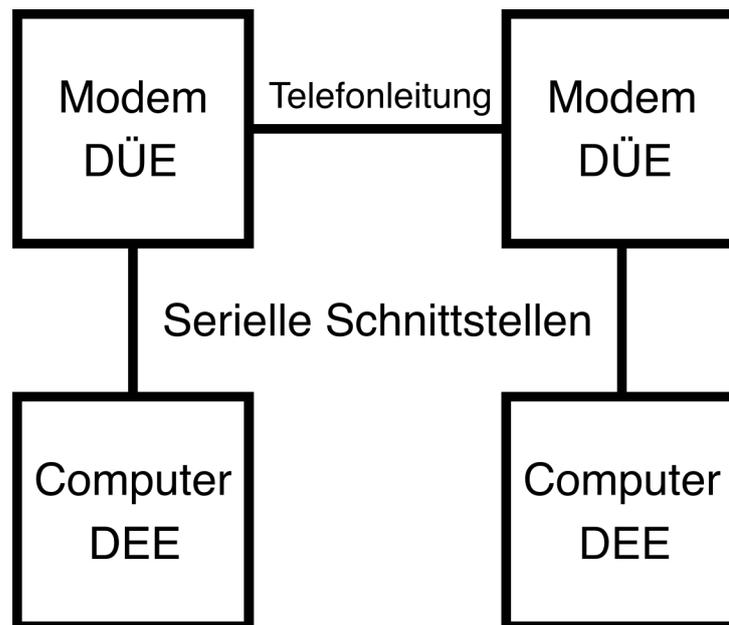


Abbildung 2.2: Blockdiagramm einer Datenübertragung zwischen zwei Computer unter Verwendung von Telefon Modem

2019, s. 23], den sogenannter Signale. Da sich anschließende Teile dieser Arbeit tiefgehend mit den Techniken der Signaltheorie beschäftigen wird, ist es wichtig eine Begriffsklärung voranzustellen.

Bei Signalen handelt es sich um ein messbares, zeitabhängiges, physikalisches Objekt [Beucher, 2019, s. 16]. In dieser Arbeit, die bezüglich der Signaltheorie einen mathematischen Ansatz verfolgt, werden Signale genauer im Kontext einer zeitabhängigen Funktion, die eine physikalische Größe abbildet, betrachtet. Daraus ergibt sich die allgemeine Definition:

$$s(t) \tag{2.1}$$

Hierbei handelt es sich um die Funktion s , die für jeden Zeitpunkt $t \in \mathbb{R}$ definiert ist [Mertins, 2020, s. 1]. Diese Eigenschaft der Signale bezeichnet man als Zeitkontinuität. Eine Differenzierung zwischen zeitkontinuierlichen und zeitdiskreten Signalen ist für diese Arbeit nicht von Relevanz. Aus diesem Grund kann bei allen Signalen pauschal von Zeitkontinuität ausgegangen werden.

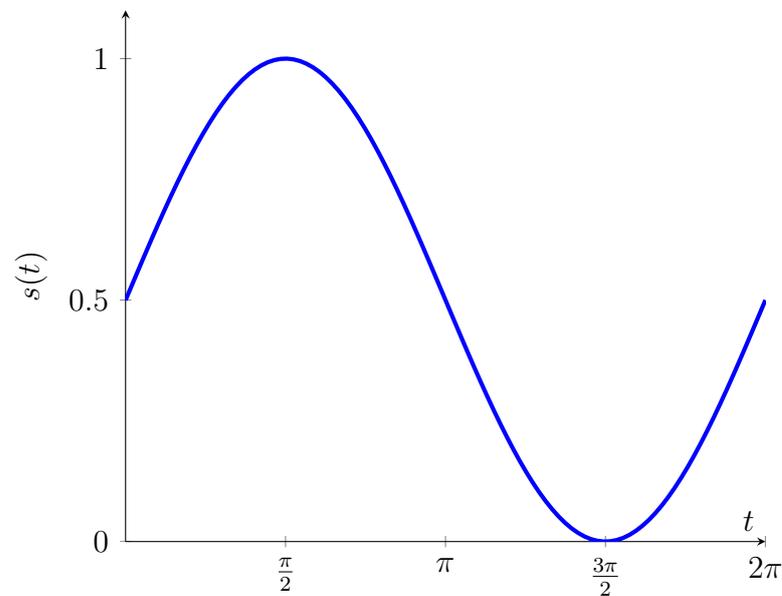


Abbildung 2.3: Graph einer typischen analogen Funktion $(\frac{1}{2} \cdot \sin(x) + \frac{1}{2})$

Signalarten

Nach dieser sehr allgemeinen Definition ist allerdings eine differenziertere Betrachtung der Signale nötig. So lassen sich Signale bezüglich ihrer Werte wie folgt unterscheiden:

1. *Wertkontinuierliche Signale*: Hierbei handelt es sich um Signale deren Werte kontinuierlich ineinander übergehen.

$$\mathfrak{W} \subseteq \mathbb{R}$$

Die Funktion aus Gleichung 2.1 kann kontinuierlich Werte aus \mathfrak{W} annehmen. Solche Signale nennt man auch *analoge Signale*. Ein berühmter Vertreter dieser Funktionen ist die Sinusfunktion (vgl. Abbildung 2.3).

2. *Wertdiskrete Signale*: Diskrete Signale haben die besondere Eigenschaft, dass sie nur vorbestimmbare Werte annehmen können.

$$\mathfrak{W} \subseteq \mathbb{R}$$

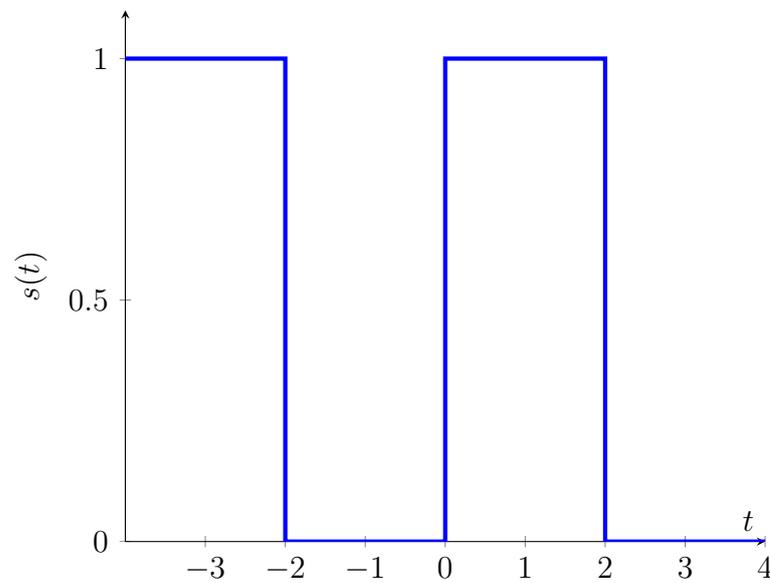


Abbildung 2.4: Graph einer typischen digitalen Funktion

Für die Menge \mathfrak{W} bedeutet das, dass sie aus endlich oder unendlich vielen reellen Werten besteht. Eine besondere Unterart dieser Signale sind *binäre wertdiskrete Signale*. Für sie gilt:

$$s(t) = \mathfrak{W} \in \{0; 1\} \quad (2.2)$$

Solche Signale nehmen entweder den Wert 0 oder 1 an. Man nennt sie auch *digitale Signale*. In der Praxis wird hier meist nicht zwischen den physikalischen Größen 0 und 1, sondern zwischen den zwei Zuständen *aus* oder *an* unterschieden. Ein *digitales Signal* ist also 0 wenn $s(t) = 0$ und 1 wenn $s(t) > 0$. Seitdem Computer diese Signalart ermöglichen hat sich ein eigener Jargon um sie entwickelt, der die Bezeichnung gewisser Zustände stark vereinfacht. Aus diesem Grund wird kurz auf die wichtigsten Begriffe eingegangen. Den Wert, den ein digitales Signal zum Zeitpunkt t haben kann (vgl. Gleichung 2.1) bezeichnet man als binary digit (Bit). Acht zusammengehörige Bit nennt man einen *Byte*. Eine Abfolge mehrere Bit/Byte nennt man Bit-/Byte-Stream. Für wertdiskrete Signale steht die Umschalt-Funktion beispielhaft (vgl. Abbildung 2.4).

Warum digital

Es ist wichtig zu verstehen, dass ganzheitlich kontinuierliche Signale ein theoretisches Konstrukt sind. In der Praxis ist - beschränkt durch die Möglichkeiten der Technik - nur das Messen von zeit- und wertdiskreten Signalen möglich [Beucher, 2019, s. 17]. Trotzdem wird, um eine möglichst hohe Genauigkeit sicherzustellen, in folgender Theorie weiterhin von zeitkontinuierlichen Signalen ausgegangen.

Wie das Statistische Bundesamt 2018 berichtete, benutzen rund 86,7% aller Haushalte einen stationären Internetanschluss [Statistisches Bundesamt, 2018]. Internetanbieter in Deutschland setzten zum Zeitpunkt dieser Arbeit weites gehend auf Digital Subscriber Line (DSL). Wie sich aus dem Namen dieser Technik schon ableiten lässt handelt es sich bei dieser Übertragungsweise um eine, die sich der digitalen Signale bedient. Diese Entscheidung ist darin begründet, dass wertdiskrete Signale gegenüber analogen klare Vorteile besitzen. Alle Übertragungswege sind störanfällig. Ob durch nicht ideal designte Schaltkreise oder andere Umwelteinflüsse muss bei der Übertragung von Signalen immer von einer Störung der einen oder anderen Art ausgegangen werden. Man spricht hierbei von *Noise* (oder auch *Interferenz*) [Sklar and Harris, 2019, s. 2f]. Noise kann statisch oder statistisch sein. Diese Tatsache erschwert vor allem die Übertragung analoger Signale. Analoge Signale übertragen Informationen durch direkte Repräsentation als physikalische Größe. Wenn durch Noise diese physikalische Größe geändert wird, ändert sich auch die repräsentierte Information um den Wert der Noise. Der Empfänger hat somit keine Möglichkeit diese Information zu erhalten. Auch wertdiskrete Signale sind von Noise betroffen. Doch durch die klare Abgrenzung der annehmbaren Werte aus \mathfrak{M} kann der Empfänger hier allerdings das empfangene, durch Noise verfälschte, Signal Interpretieren. Denn je nach Abstand des empfangenen Wertes zu denen aus \mathfrak{M} kann er unter Verwendung eines *regenerative repeaters* runden und so das Originalsignal wiederherstellen [Sklar and Harris, 2019, s. 2]. Vor allem bei digitalen Signalen ist diese Eigenschaft besonders ausgeprägt. Hier kann schlicht davon ausgegangen werden, dass die Maxima des empfangenen Signals „an-Zustände“ waren (vgl. Abbildung 2.5). Diese Methode ist trotz ihrer einfachen Um-

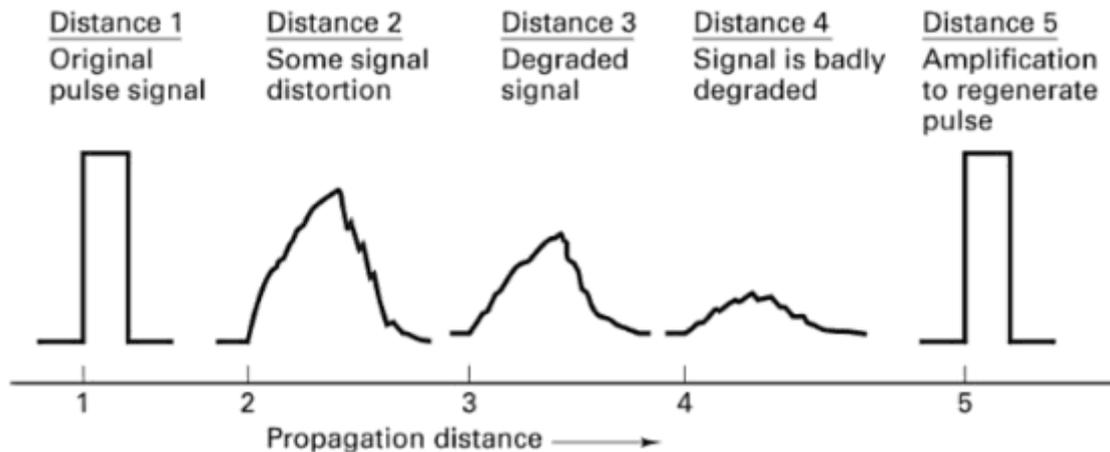


Abbildung 2.5: Beispiel Signal Regeneration eines Binären digitalen Signals [Sklar and Harris, 2019]

setzung sehr effektiv (siehe 3.1. Aufbau) und der primäre Grund für die Entscheidung für digitaler vor analoger Signale. Digitale Signale bieten darüber hinaus auch ökonomische Vorteile. Durch die hohe Flexibilität die Mikroprozessoren der Entwicklung von digitalen Schaltkreisen bieten, wird vermehrt auf diese umgestiegen. Wenn nun zwei solcher Systeme miteinander kommunizieren sollen, bietet sich eine digitale Übertragung an, da so das Formatieren der Signale erspart wird. (vgl. 2.1.3. Übertragung).

Modulation

Doch auch mit dieser eindeutigen Überlegenheit der digitalen Signale in der Computer zu Computer Kommunikation ist der Übertragungsweg an die physikalischen Gegebenheiten gebunden. So bieten viele Übertragungswege keine Möglichkeit Bits direkt zu übertragen. Um diese Einschränkung der Naturtechnik zu umgehen, bedient sich die Signaltheorie hier der Modulation um das digitale Signal zu einem analogen zu transformieren. Der folgende Abschnitt befasst sich mit den verschiedenen Methoden die sich hierzu eignen.

Die Modulation ist ein Verfahren bei dem ein sogenanntes *Carrier-Signal* in seinen physikalischen Eigenschaften abhängig von einem *Messenger-Signal* angepasst wird. Das

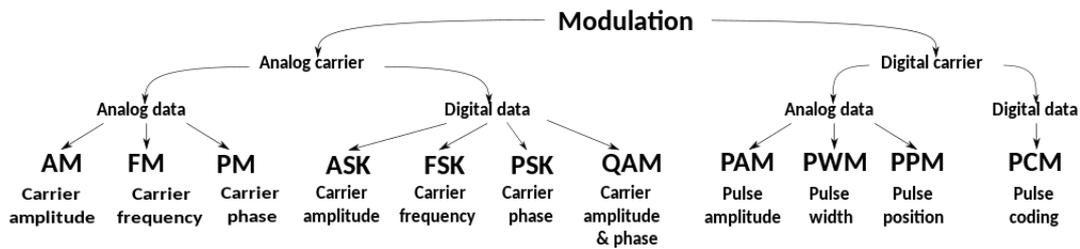


Abbildung 2.6: Übersicht der Arten der Modulation

Carrier-Signal ist dabei das Signal, das auf dem Datenübertragungsweg übertragen wird, während das *Messenger-Signal* die Daten widerspiegelt die im System (vgl. 2.1.1. *Aufbau eines Systems*) übertragen werden sollen [Proakis and Salehi, 2007, s. 95].

Welche Methode der Modulation anzuwenden ist hängt stark von der Art des *Messenger- und Carrier-Signals* ab. Die Unterteilung der verschiedenen Modulaktions-Arten sind der Abbildung 2.6 zu entnehmen. Da sich diese Arbeit auf einen Laser als Datenübertragungsweg beschränkt, wird im Folgenden von einem *analogen Carrier-Signal* ausgegangen. Da, wie oben beschrieben, digitale Daten zu bevorzugen sind, wird auch von einem *digitalen Messenger-Signal* ausgegangen. Es bleiben nun vier verschiedene Methoden den Bitstream auf das analoge Signal des Lasers zu modulieren. Jede dieser Arten bezieht sich dabei auf eine der Welleneigenschaften Amplitude, Frequenz oder Phase:

1. *Amplitudenumtastung (ASK)*:

Bei diesem Verfahren wird die Amplitude des Carrier-Signals $C(t)$ je nach Bit des Signals $s(t)$ (Gleichung 2.2) angepasst [Carson, 1922, s. 57]. Für das modulierte Signal M folgt also:

$$M(t) = s(t) \cdot C(t) \tag{2.3}$$

Durch die Einführung der Variablen a und b lässt sich die Amplitude des modulierten

Signals anpassen.

$$M(t) = (a \cdot s(t) + b) \cdot C(t) \quad (2.4)$$

$$a \in \mathbb{R} \setminus \{0\} \quad (2.5)$$

$$b \in \mathbb{R} \quad (2.6)$$

Für die Amplitude der modulierten Funktion bei $s(t) = 0$ gilt also:

$$M(t) = b \cdot C(t) \quad (2.7)$$

Die Amplitude der modulierten Funktion bei $s(t) > 0$ wird ausgedrückt durch:

$$M(t) = ab \cdot C(t) \quad (2.8)$$

Diese Art der Modulation ist sehr einfach da sie meist durch simples Ein- und Ausschalten des Carrier-Signal-erzeugenden Gerätes erreicht werden kann. Auch ist das demodulieren, also das Umwandeln des modulierten Signals in das Messenger-Signal, sehr einfach. Hierbei muss im Vergleich zu den andern Modulationsarten keine Frequenz- oder Phasen-Analyse durchgeführt werden, sondern nur überprüft werden, ob die Amplitude des empfangenen Signals größer als $b \cdot C(t)$ (Gleichung 2.7) ist.

2. Frequenzumtastung (FSK):

Diese Methode ändert in Abhängigkeit des Messenger-Signals die Frequenz des Carrier-Signals. Für das modulierte Signal M gilt:

$$M(t) = C(t \cdot (s(t) + 1)) \quad (2.9)$$

C ist die Funktion des Carrier-Signals. Dieses wird in Abhängigkeit von s (Gleichung 2.2) in seiner Frequenz moduliert. So ist im Falle eines niedrigen ($\hat{=} 0$) Bits der Ausdruck $s(t) + 1 = 1$. Die Frequenz des Carrier-Signals bleibt unverändert.

Im Falle eines aktiven Bits ($\hat{=} 1$), ist der Ausdruck $s(t) + 1 = 2$ und die Frequenz wird halbiert. Durch das Einführen einer Variable a lässt sich die Formel nun so anpassen, dass eine gleiche Frequenz bei aktivem Bit moduliert werden kann:

$$M(t) = C(t \cdot (a \cdot s(t) + 1)) \quad (2.10)$$

$$a \in \mathbb{R} \quad (2.11)$$

Für die Frequenz des modulierten Signals f_M folgt also:

$$f_M = \frac{2\pi}{a \cdot s(t) + 1} \quad (2.12)$$

3. Phasenmodulation (PSK):

Da, wie in 2.1.2. *Warum digital* geschildert, ein binäres Messenger-Signal verwendet wird befasst sich dieser Abschnitt mit Binäre Phasenmodulation (BPSK). Hierbei wird die Phase des Carrier-Signals angepasst. Allgemein gilt:

$$M(t) = C(t - (s(t) + 1) \cdot a) \quad (2.13)$$

$$a \in \mathbb{R} \quad (2.14)$$

Die Phase des Carrier-Signals wird in Abhängigkeit von $s(t)$ um a verschoben. So hat $M(t)$ bei einem niedrigen Bit keine Verschiebung und bei einem hohen Bit eine Verschiebung von a .

Um die größtmögliche Demodulierbarkeit, und somit auch die geringste Störungsanfälligkeit zu erreichen, wird die Phase des Carrier-Signals in der Regel um 180° angepasst. Es folgt [Stern and Mahmoud, 2003, s. 508]:

$$M(t) = C(t - (s(t) + 1) \cdot \pi) = -(s(t) + 1) \cdot C(t) \quad (2.15)$$

Daraus ergeben sich zwei Fälle für das Signal $M(t)$

$$M_{s_0}(t) = C(t) \quad \text{Bei } s(t) = 0 \quad (2.16)$$

$$M_{s_1}(t) = -C(t) \quad \text{Bei } s(t) = 1 \quad (2.17)$$

Diese Art der Modulation ist sowohl simpel als auch sehr zuverlässig. So wird sie beispielsweise bei den sehr weit verbreiteten Technologien Wireless Local area network (WLAN), Bluetooth und Radiofrequenz-Identifikation (RFID) angewendet.

4. *Quadratur-Amplituden-Modulation (QAM):*

Dieses Verfahren kombiniert sowohl Phasenmodulation (PSK) als auch Amplitudenumtastung (ASK), wodurch ein sehr zuverlässiges Moderationsverfahren erreicht wird. Allerdings ist die Umsetzung dadurch auch deutlich komplexer und übersteigt den Anspruch dieser Arbeit.

Welche dieser Modulationsarten verwendet werden sollte, beziehungsweise angewendet werden müssen, hängt von physikalischem Medium des Carrier-Signals ab. Hierbei ist auf [2.2. Laser als Datenübertragungsweg](#) zu verweisen.

2.1.3 Übertragung

Die Systeme aus [2.1.1. Aufbau eines Systems](#) sind nun dank [2.1.2. Signale](#) in der Lage digitale Signale zu übertragen. Es stellt sich nun die Frage, wie Daten in Signale umgewandelt werden. Bei Zahlen ist dieser Prozess recht einfach, doch je komplexer die Daten, desto weniger intuitiv scheint diese Aufgabe. Aus diesem Grund beschäftigt sich der folgende Abschnitt mit der Aufbereitung von Daten zu digitale Signale.

Das Blockdiagramm aus [Abbildung 2.7](#) zeigt diesen Prozess, der *Formatierung* genannt wird. Da diese Arbeit den Fokus auf die Grundlagen legt, wird im Folgenden nur die

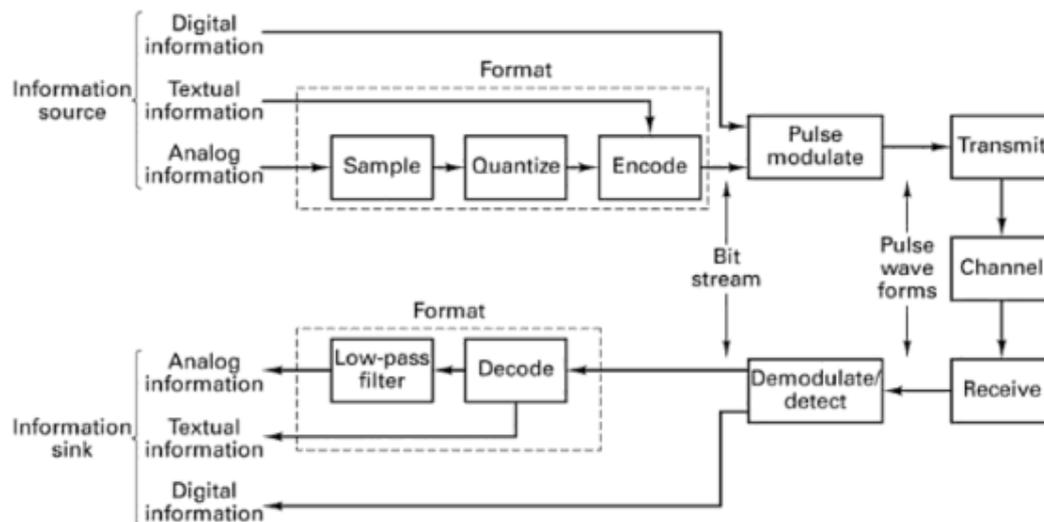


Abbildung 2.7: Schema der Formatierung und Übertragung von Signalen [Sklar and Harris, 2019]

Übertragung von *alpha numerischer* Text Informationen betrachtet.¹ Diese Datenart ist besonders relevant, da sie die meist verwendete ist [Sklar and Harris, 2019, s. 55]. Bei der Formatierung von alpha numerischer Text Information handelt es sich um eine Codierung [Sklar and Harris, 2019, s. 55]. Bei dieser Codierung wird jedem Zeichen der ursprünglichen Zeichenmenge ein alternatives Zeichen, einer möglicherweise völlig andersartigen Zeichenmenge zugeordnet [Claus and Schwill, 1993]. Ein gutes Beispiel einer solchen Neuweisung ist der von Samuel F. B. Morse im Jahre 1837 erfundene *Morsecode*. Hier wird der ursprünglichen Zeichenmenge, bestehend aus lateinischen Buchstaben, Ziffern, Umlaute et cetera jeweils ein neues Zeichen, aus langen und kurzen Tonsignalen und Pausen zugeordnet. So wird statt dem Buchstaben „A“ ein „• –“ (*lang kurz*) übertragen. Solch eine Codierung wird nun auch bei der digitalen Übertragung verwendet. Hierbei verwendet man alternative Zeichenmengen die Buchstaben durch Bitstreams ersetzen. Zwei berühmte Vertreter sind hierbei American Standard Code for Information Interchange (Ascii) und Unicode Transformation Format (Utf). Vor allem Utf ist hier erwähnenswert, da es dank der Verwendung im Hypertext Transfer Protocol (Http) eine der weit verbreitetsten Co-

¹Die Übertragung von sonstigen digitalen Information wird ignoriert, da diese durch simples überspringen der in 2.1.3. *Übertragung* beschriebenen Prozesse übertragen werden können

Bits					Column	0	1	2	3	4	5	6	7
b ₇	b ₆	b ₅	b ₄	b ₃	Row	0	1	2	3	4	5	6	7
0	0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	0	4	EOT	DC4	\$	4	D	T	d	t
0	1	0	1	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	0	8	BS	CAN	(8	H	X	h	x
1	0	0	1	1	9	HT	EM)	9	I	Y	i	y
1	0	1	0	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	1	11	VT	ESC	+	;	K	[k	{
1	1	0	0	0	12	FF	FS	,	<	L	\	l	
1	1	0	1	1	13	CR	GS	-	=	M]	m	}
1	1	1	0	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	1	15	SI	US	/	?	O	_	o	DEL

Abbildung 2.8: Ascii-Tabelle aller 128 Zeichen aus MIL-STD-188-100 (1972)

dierungen ist. Zusätzlich unterstützt er eine doppelt so hohe Zeichenmenge und ist zu Ascii abwärtskompatibel. Aus diesem Grund ist die Verwendung von Utf unabdinglich. Ascii ist die amerikanische Form der *ISO 646* [International Telecommunication Union, 1991]. In ihm wird eine Menge von 95 druckbaren und 33 nicht druckbaren (steuerungs-) Symbolen durch eine Kette aus 7-Bit (also ein Byte) kodiert (vgl. Abbildung 2.8). Er bildet so die Basis Zeichenmenge des Utf-Standards.

Diese Kodierung kann nun genutzt werden, um alpha numerische Text Informationen in einen Bitstream umzuwandeln und als digitales Signal zu übertragen.

Das größte Problem bei der Übertragung stellt allerdings der Fakt dar, dass die DE-Einrichtung und DÜ-Einrichtung zwei, voneinander vollständig isolierte Systeme sind. Das bedeutet, dass die Taktsignale nicht synchronisiert sind. Abhängig von der Bauart der Taktgeber² kann es so zu nicht vorhersehbaren Taktunterschieden kommen. Dieser Effekt

²Taktgeber werden benötigt um mit der Richtigen Frequenz Bits zu empfangen oder zu senden. Hierfür

ist kleiner, je niedriger die Übertragungsgeschwindigkeit ist. Bei hohen Geschwindigkeiten kann dieser Effekt allerdings zu „*Bitshifting*“ (*Bitverschiebung*) führen. Auch ist es möglich das dieser Effekt durch Software verstärkt wird. Braucht eines der beiden Systeme länger als das andere zur Verarbeitung der Signale, so kann sich auch hier ein Bit verschiebt. Daraus ergibt sich diese Bedingung für *Bitshifting*:

$$d - \Delta t \leq \int_{t_0}^t n(t) dt - (d - \Delta t) \quad (2.18)$$

f ist die Frequenz in *bit/s*, die angibt wie viele Bits pro Sekunde übertragen werden. d ist die Zeit, die das System warten muss, bis ein nächster Bit empfangen wird. Sie kann durch

$$d = 1/f \quad (2.19)$$

berechnet werden. $n(t)$ ist der oben beschriebene statistische Taktunterschied der Systeme in Abhängigkeit von t . Es wird im Folgenden davon ausgegangen, dass dieser Unterschied nur positiv ist. t ist die Zeit seit Übertragungsstart. Δt ist der ursprüngliche Zeitunterschied der Systeme, der sich durch die Übertragungszeit des ersten Bit ergibt.³

Wie Gleichung 2.18 zeigt, ist es, da Δt konstant ist, nur durch kleine t möglich diesen Effekt einzuschränken. Dieser Zustand führt folglich dazu, dass nur kleine Datenmengen übertragen werden können. Deswegen werden diese üblicherweise auf *Pakete* aufgeteilt. Dadurch wird t und somit der gesamte Taktunterschied möglichst klein gehalten. Ein weiterer Vorteil ist, dass sich Pakete durch einen *Marker-Bit* ankündigen, und sich die DE-Einrichtung und DÜ-Einrichtung so bei jedem System wieder neu synchronisieren können.

muss eine gewisse Zeit gewartet werden. Taktgeber geben in einer sehr hohen festen Frequenz ein Signal mit dessen Hilfe die richtige Zeit bestimmt werden kann.

³Da den Systemen weder Ausbreitungsgeschwindigkeit noch Länge des Übertragungsweges bekannt ist lässt sich dieser Unterschied nicht vermeiden. Bei nicht gestörter Übertragung hat dieser Fakt allerdings keine Auswirkung auf das korrekte empfangen von Bits

2.2 Laser als Datenübertragungsweg

Seit der Fertigstellung des ersten L.A.S.E.R. von Theodore H. Maiman im Jahr 1960 [MAIMAN, 1960] wurden sowohl in Industrie, als auch in privaten Haushalten immer mehr Anwendungsfälle für diese gefunden. So waren sie im Verlauf der letzten Jahre aus keinem Wohnzimmer wegzudenken und wurden millionenfach in DVD-Playern verbaut. Auch hier dienten sie, wenn auch nicht direkt, zur „Datenübertragung“. Der L.A.S.E.R. scheint somit Eigenschaften zu haben die alternativer elektromagnetischer Strahlung (EMS) überlegen sind.

Wie in *2.1.2. Signale* beschrieben hängt die Wahl der Modulation von verwendeten Datenübertragungsweg ab. Bei der Verwendung einse handelsüblichen L.A.S.E.R. bietet sich hier, aufgrund der festen Wellenlänge, nur ASK an. Diese Modulation kann wie beschrieben durch an- und ausschalten des L.A.S.E.R. erreicht werden.

Der folgende Abschnitt beschäftigt sich speziell mit dem L.A.S.E.R. als dem in *2.1.1. Aufbau eines Systems* beschriebenen Datenübertragungsweg und wiegt seine Vorteile sowie Nachteile ab.

Das Themengebiet der L.A.S.E.R. ist sehr Umfangreich und es müssen viele Aspekte beim Vergleich mit üblicher EMS betrachtet werden. Dieser Abschnitt stellt somit keinen Anspruch an Vollständigkeit, um inhaltlichen Fokus zu bewahren.

Aufgrund der Reflexionseffekte im Resonator sind L.A.S.E.R. sehr effizient. Das Betreiben einer DÜ-Einrichtung ist somit mit niedrigerem Energieaufwand möglich als bei üblichen Übertragungsarten [Ali, 2019].

Ein sehr bedeutsamer Unterschied des L.A.S.E.R. zu üblicher EMS ist die deutlich höhere Frequenz. Viele, vor allem kommerzielle, Produkte setzen heutzutage auf Ultra-High-

Frequency (UHF). Nennenswerte Vertreter wären hier vor allem WLAN und Global Positioning System (GPS). UHF hat eine maximale Frequenz von $f_{\text{uhf}} = 3\text{GHz}$. Selbst L.A.S.E.R. der Klasse 1 erreichen laut DIN 60825-1 [Deutsches Institut für Normung e.V., 2022] im Vergleich Frequenzen von bis zu $f_{\text{laser}} = 0,43\text{PHz}$. Somit sind L.A.S.E.R. in der Lage deutlich höhere Energien zu übertragen und sind für weite Distanzen geeignet. Im Kontext dieser Arbeit bedeutet dieser Fakt jedoch auch, dass L.A.S.E.R. größere Datenübertragungsgeschwindigkeiten unterstützen. Dafür ist es wichtig auf die in 2.1.2. *Signale* beschriebenen Modulationsverfahren zu blicken. Speziell anhand der ASK lässt sich dieser Zusammenhang gut erklären. Wie beschrieben sind digitale Signale nur ein theoretisches Konstrukt, da es für Signale nicht möglich ist mit einem Zeitunterschied von $\Delta t = 0$ seinen Zustand zu wechseln. Diese Tatsache beschränkt nun die Übertragungsgeschwindigkeit. Ist diese für das genutzte Carrier-Signal zu hoch, so hat dieses keine Möglichkeit seinen Zustand schnell genug zu wechseln und die Amplitude des modulierten-Signals ist zu niedrig um von der DE-Einrichtung erkannt zu werden (Abbildung 2.9 veranschaulicht dieses Prinzip). Die Frequenz einer Welle gibt an, wie schnell diese ihren Zustand ändern kann. So kann ein, wie oben erwähnter, Klasse 1 L.A.S.E.R.

$$2 \cdot f_{\text{laser}} = 86 \cdot 10^{13} \quad (2.20)$$

mal die Sekunde seinen Zustand ändern und somit auch maximal $86 \cdot 10^{13}\text{bit/s}$ übertragen.

Doch neben der technischen Spezifikationen bietet der L.A.S.E.R. auch große *Sicherheitsaspekte*. Durch die, wie nun schon oft erwähnte, sehr kleine Wellenlänge, breitet er sich im Vergleich zu anderer Strahlung kaum im Raum aus. Dies *behindert das Detektieren* durch Spektrums Analyse [Ali, 2019, s. 3]. Zusätzlich dazu breitet sich der L.A.S.E.R. linear aus und ermöglicht so das Abhören nur nach physischer Brechung dieses Strahles. Detektoren und Abhör-Einrichtungen können so nicht zentral angebracht werden.

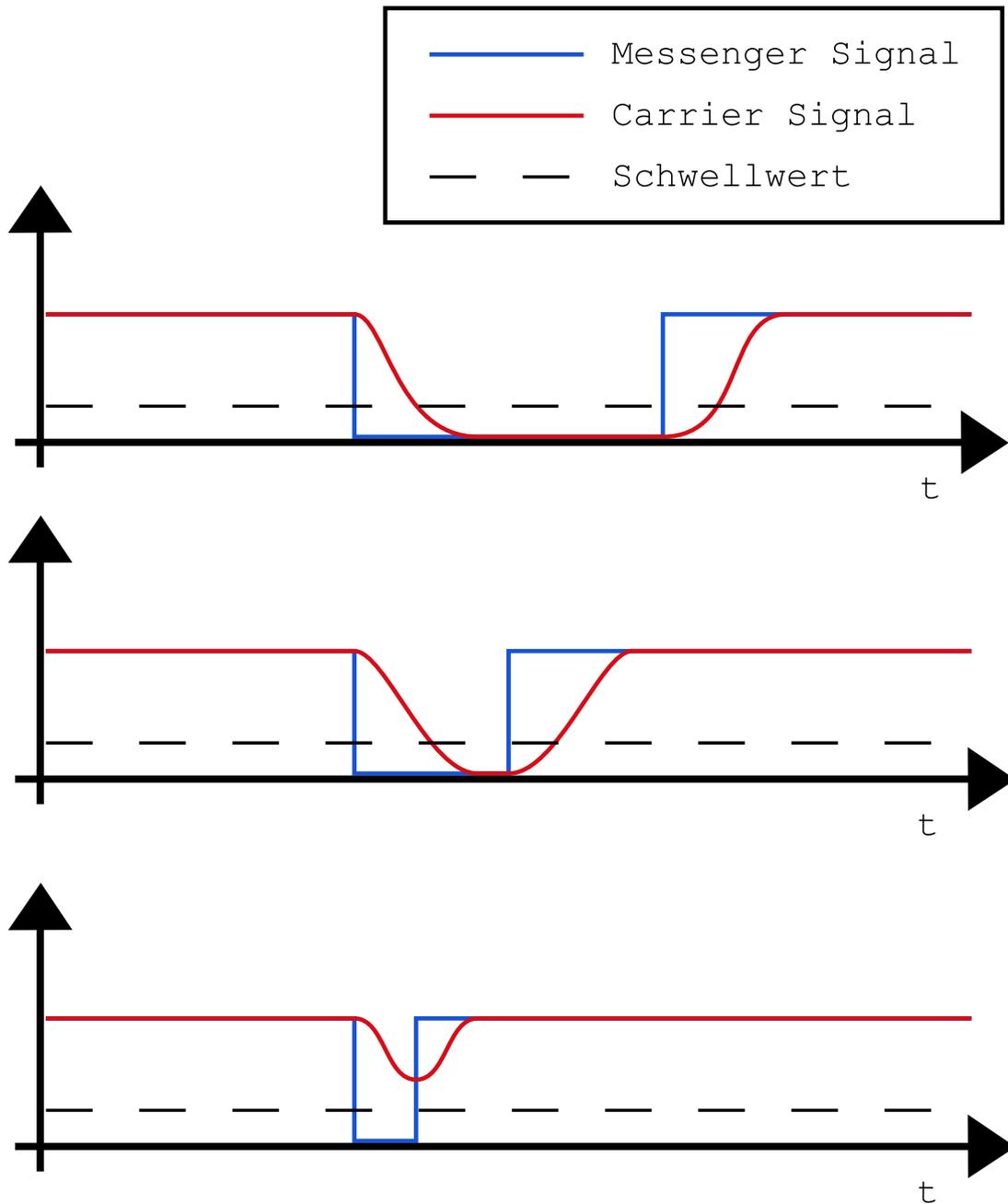


Abbildung 2.9: Daten Übertragungsgeschwindigkeit (DTR) Limitationen bei ASK-Modulation am Beispiel eines Carrier-Signals bei verschiedenen Geschwindigkeiten

Die geringe räumliche Ausbreitung bietet auch den Vorteil der *Stabilität*. Durch die beschränkte Auffindbarkeit der L.A.S.E.R. ist es erschwert diese zu finden und die Übertragung absichtlich zu unterbrechen [Ali, 2019, s. 3].

3 | Versuch

Wie in *1. Einleitung* beschrieben ist das Ziel dieser Arbeit das Untersuchen der L.A.S.E.R.-Datenübertragung anhand eines selbst durchgeführten Experiments. Dieses Kapitel orientiert sich an den in *2. Theorie* erläuterten Prinzipien und beschreibt unter deren Verwendung das praktische Experiment dieser Arbeit. Dabei ist zu erwähnen, dass der gesamte Versuchsaufbau, sowie auch die Datenerhebung und Analyse ohne fremde Hilfe oder professionelle Ausrüstung durchgeführt wurden. Aus diesem Grund darf nicht davon ausgegangen werden, dass die hier dargestellten Daten und Schlussfolgerungen ausnahmslos empirisch sind. Dennoch können die Ergebnisse ein abstraktes Bild dieser Technologie zeichnen und ermöglichen auch eine sinnvolle Einordnung.

3.1 Aufbau

Folgende grundlegende Beschreibung trifft sowohl auf den Sender, als auch auf den Empfänger zu. Dies liegt daran, dass es sich bei beiden um Datenstationen handelt, die sich nur in ihrer Betriebsart unterscheiden. (vgl. *2.1.1. Aufbau eines Systems*).

Die Hardware beider Stationen basiert auf dem *RP2040*, einem Mikrochip der britischen Raspberry Pi Foundation. Er verfügt über 2 133MHz Kerne basierend auf der ARMv6-M-Architektur. Darüber hinaus besitzt er über 20 General Purpose Inputs/Outputs (GPIOs) und einen 4-Kanal 12-Bit Analog-Digital-Umsetzer (ADC) [Raspberry Pi Ltd, 2023]. Diese Input/Output (I/O)-Flexibilität, hohe Prozessortaktung und die verhältnismäßig niedrigen Anschaffungskosten machen ihn zu einer logischen Wahl. Zusätzlich wird auch ein *Infineon CYW43439* verwendet der Wireless Verbindungen über WLAN $2,4\text{GHz}$ und

Bluetooth 5.2 ermöglicht [Cypress, 2023].

Die gesamte Software der Stationen (vgl. <https://github.com/sebivh/wsemCode>) wurde in *MicroPython* geschrieben [George, 2023]. Dabei wird auf Kern 1 ein simpler Http-Webserver ausgeführt, während Kern 2 die Aufgaben des Empfangens oder Sendens übernimmt. Dadurch wird sichergestellt, dass beide Prozesse unabhängig voneinander laufen und Fehler des einen Prozesses den anderen nicht beeinflussen. So sollen Fehler des Webserver, durch zum Beispiel fehlende Http-Header Informationen, nicht verhindern, dass die Einheit ohne Probleme weiter empfangen/senden kann.

Die folgenden Abschnitte beschäftigen sich mit dem spezifischen Aufbau der einzelnen Datenstationen.

3.1.1 Sender

Der Sender (Abbildung 3.1) hat die Aufgabe die Daten zu formatieren und den L.A.S.E.R. so anzusteuern, dass diese übertragen werden.

Dieser ist, wie in Abbildung 3.2 zu sehen, über Pin 19 an den *RP2040* angeschlossen. Bei dem L.A.S.E.R. handelt es sich, aufgrund der niedrigen Anschaffungskosten, um einen $650nm$ L.A.S.E.R. mit einer maximalen Ausgangsleistung von $5mW$ bei $5V \cdot 45mA \approx 0,23V$. Da die GPIO des *RP2040* bei einer Spannung von $3,3V$ maximal $16mA$ bereitstellen können, wird der L.A.S.E.R. hier allerdings mit einer geringeren Leistung von

$$3,3V \cdot 16mA \approx 5,3mW \tag{3.1}$$

betrieben. Das entspricht nur etwa 23% der Maximalleistung. Zusätzlich wurde an Pin 17 eine LED angebracht, die den Status des L.A.S.E.R. widerspiegelt.

Im *RP2040* lassen sich DE-Einrichtung und DÜ-Einrichtung vereinen. Erstere ist hier repräsentiert durch den Http-Webserver. Eine Umwandlung der Daten ist so nicht nötig, da



Abbildung 3.1: Bild des Senders

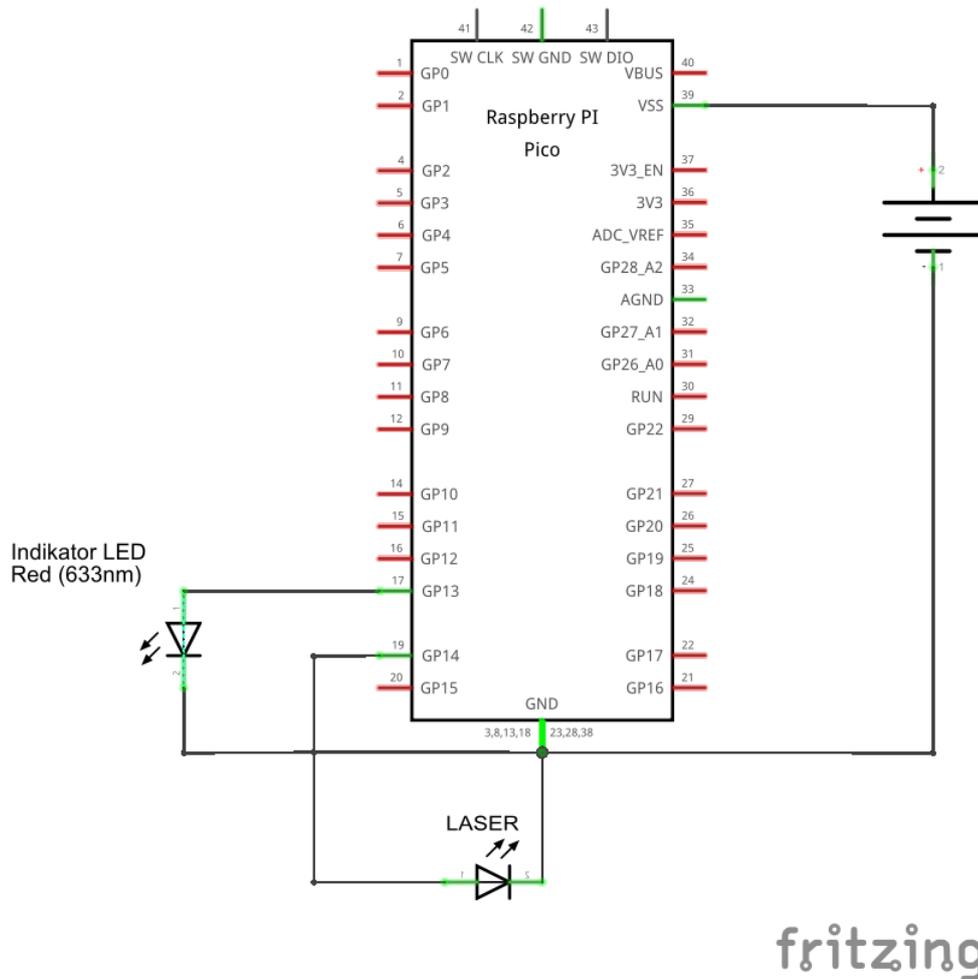


Abbildung 3.2: Schaltplan des Senders mit L.A.S.E.R. und Indikator Leuchtdiode (LED)

die Daten schon, dank des Empfanges über den Http-Websocket, in Form des benötigten *Bytearrays*, also eines Bitstreams, bereits vorliegt. Da beide Einheiten auf dem gleichen Chip laufen ist ebenfalls das Synchronisieren der Schnittstelle nicht nötig. Nach der Empfangen der Daten beginnt die DÜ-Einrichtung direkt mit der Übertragung (Code 3.1.1). Dies ist möglich, da keine weitere Aufbereitung nötig ist.

```

1 def send(received_bytes):
2     global laser, time, onboard_led, indicator_led, is_emitting,
   send_thread, time_remaining
3
4     # Ueber Empfangene Bytes Iterieren
5     for byte in received_bytes:
6
7         # Einen Bit=1 senden um neues Packet anzukundigen
8         laser.value(1)
9         indicator_led.value(1)
10        time.sleep(1/dtr)
11
12        # Ueber Byte Iterieren mit Iterations Variable i
13        for i in range(8):
14
15            # Ist bit an stelle i 1
16            bit = (byte & 1<<(7 - i)) != 0
17
18            # Sed Laser
19            laser.value(bit)
20            indicator_led.value(bit)
21
22            time.sleep(1/dtr)
23
24        laser.off()
25        indicator_led.off()
26        time.sleep(1/dtr)
27
28    return

```

Code 3.1: Die Sende-Funktion der DÜ-Einrichtung des Senders mit eingefügten Kommentaren

Hierbei handelt es sich um einen iterativen Prozess. So wird in Zeile 14 eine Variable *i* eingeführt, die auch *Iterationsvariable* genannt wird. Sie wird bei jedem Schritt um 1

erhöht und verschiebt den einzigen Bit der Ganzzahl 1, der *an* ist ($int(1) \hat{=} 00000001$). Der Ausdruck \ll verschiebt Bits nach „links“. Das bedeutet, dass im ersten Fall $i = 0$ aus 00000001 der Byte 10000000 wird. Über die Boolesche Operation $\&$ („UND,“) können so einzelne Bit isoliert und gesendet werden.

$$b \& 0 = 0 \tag{3.2}$$

$$b \& 1 = b \tag{3.3}$$

$$b_7b_6b_5b_4b_3b_2b_1b_0 \& 00010000 = 000b_40000 \tag{3.4}$$

$$b \in \{1; 2\} \tag{3.5}$$

Das Senden, also die Anschalteinheit, ist in Zeile 19 zu sehen. Hier, wird je nach Bit, der L.A.S.E.R. an- oder ausgeschaltet. Darauf hin wird, in Abhängigkeit der DTR, die richtige Sekunden Anzahl, bis der nächste Bit gesendet werden muss (Gleichung 2.19) gewartet. Dieser Prozess läuft iterativ so lange ab, bis alle Bytes gesendet wurden. Es gibt noch zusätzliche „Start“- und „Marker“-Bits die gesendet werden. Diese werden in *3.1.3. Übertragungsprotokoll* behandelt.

3.1.2 Empfänger

Der Empfänger (Abbildung 3.3) hat die Aufgabe Daten entgegenzunehmen und wieder in ihre Ursprungsform zurück zu formatieren. Dabei greift er auf den verbauten ADC zurück. Dieser wird, wie in Abbildung 3.4 abgebildet, über Pin 31 an eine Foto-Diode (LED) angeschlossen und der ADC ermöglicht so das Ablesen der angelegten Spannung, die je nach Lichteinfallintensität steigt. Genauer, kann er einen Spannungsbereich von $[0V ; 3.3V]$ lesen. Weil der ADC mit einer Auflösung von 12-Bit arbeitet, kann er die Spannung auf

$$\frac{3.3V}{4096} = 0.8mV \tag{3.6}$$

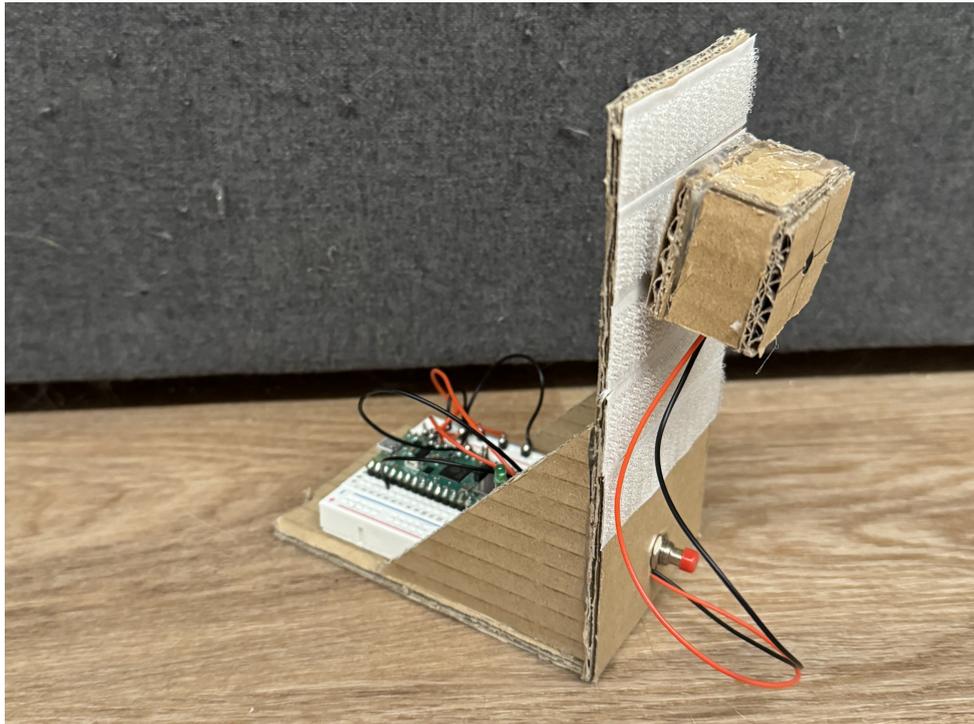
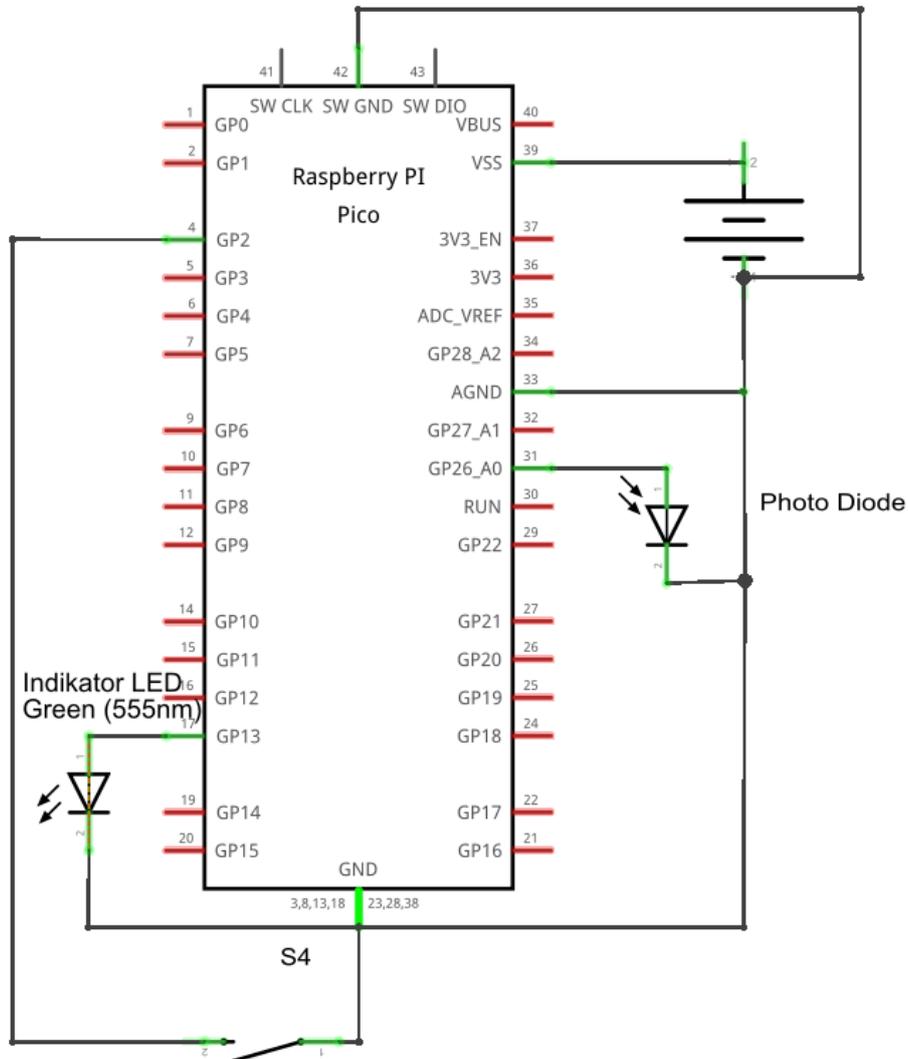


Abbildung 3.3: Bild des Empfängers

genau bestimmen. Da die Spannungen, die durch den L.A.S.E.R. indiziert werden, $\geq 1mV$ sind, reicht diese Auflösung völlig aus. Zusätzlich zu der an Pin 17 angebrachten Indikator LED sitzt an Pin 4 ein Knopf. Dieser ist dazu gedacht den Empfänger zurückzusetzen.

Auch in dieser Datenstation sind DÜ-Einrichtung und DE-Einrichtung im RP2040 vereint. Hier ist der ADC die Anschalteinheit, die das Signal empfängt. Anders als bei 3.1.2. *Empfänger* hat der Signalumsetzer hier eine wichtige Aufgabe. Durch die Modulation (vgl. 2.1.2. *Signale*) des Datensignals auf den L.A.S.E.R. wird hier nur ein analoges Signal empfangen. Um dieses umzuwandeln wird die in 2.1.2. *Signale* beschriebene Regeneration verwendet. Da nicht davon ausgegangen werden kann, dass keine Störungen induziert werden, kann nicht schlicht zwischen $0V$ und $> 0V$ unterschieden werden. Deswegen wird vor Empfangen bei Start der DÜ-Einrichtung ein Schwellwert U_t festgelegt, der sich von Messungen des Umgebungslichtes ableitet (vgl. <https://github.com/sebivh/wsemCode/blob/master/Receiver/boot.py#L46-L78>).



fritzing

Abbildung 3.4: Schaltplan des Empfängers mit Foto-Diode und Indikator LED

$$U_t = 110\% \cdot r \cdot C_{adc} \quad (3.7)$$

$$C_{adc} = \frac{3.3V}{65536} \quad (3.8)$$

$$(3.9)$$

Der Rohwert r ist der höchste aller in einer Minute gemessenen Werte des Umgebungslichtes der ADC und wird mit der Konstante C_{adc} multipliziert, um die Messung in die Einheit Volt umzurechnen.¹ Zusätzlich wird er mit dem Faktor 110% multipliziert, um die empfangenen Daten klar von der Umgebungsstrahlung abzugrenzen.

```

1 def receiveMessage(receiver_adc):
2     global END_SYMBOLE, dtr, ADC_CONV_FACTOR, SENSOR_THRESHOLD,
   onboard_led, button
3
4     raw_message = bytearray()
5     received_byte = 0
6
7     # Wenn der letzte empfangene Byte der End-Byte war dann stop
8     while received_byte != END_SYMBOLE:
9
10        # Warten auf den Packet Marker-Bit
11        while True:
12            reading = receiver_adc.read_u16() * ADC_CONV_FACTOR
13            if( reading > SENSOR_THRESHOLD):
14                break
15            # Ueberpruefen auf dem Knopf fuer das zurueckschalten
16            if button.value() == 1:
17                return raw_message
18            time.sleep(1/(4*dtr))
19
20        # Auf ersten Bit des Packetes Warten
21        time.sleep(1/dtr)
22
23        received_byte = 0

```

¹Obwohl der Hardware ADC nur eine Auflösung von 4096 hat, arbeitet *MicroPython* mit einer Auflösung von 65536 um möglichst viele Systeme zu unterstützen. Daher wird auch hier der Wert 65536 verwendet.

```

24     # Beginnt Byte zu lesen
25     for i in range(8):
26         # Ueberpruefen auf dem Knopf fuer das zurueckschalten
27         if button.value() == 1:
28             return raw_message
29
30
31         volt_read = reciver_adc.read_u16() * ADC_CONV_FACTOR
32         bit = volt_read > SENSOR_THRESHOLD
33
34         recived_byte += bit << (7 - i)
35
36         # Auf naechsten Bit warten
37         time.sleep(1/dtr)
38
39         # Ganzen Byte Gelesen, zu bytearray hinzufuegen
40         raw_message.append(recived_byte)
41
42     raw_message = raw_message[:-1]
43
44     return raw_message

```

Code 3.2: Die Empfang-Funktion des Empfängers mit eingefügten Kommentaren

Mit diesem Schwellwert kann in Zeile 32 (Code 3.1.2) der übertragenen Bit regeneriert werden. Hierbei wird die Bedingung

$$r \cdot C_{adc} > U_t \quad (3.10)$$

überprüft. Im Anschluss wird in Zeile 34, ähnlich des Prinzips aus 3.1.1. *Sender*, der Bit an die richtige Stelle des Bytes verschoben und numerisch addiert. Auch dieser Prozess wird iterativ so lange durchgeführt bis der „End“-Byte empfangen wurde (3.1.3. *Übertragungsprotokoll*).

Nachdem die Daten nun empfangen und regeneriert sind, werden sie über die Schnittstelle wieder an die DE-Einrichtung übergeben (Zeile 44 in Code 3.1.2). Hier können die Daten nun wieder zurück formatiert und in ihre ursprüngliche Textform gebracht werden. Da die

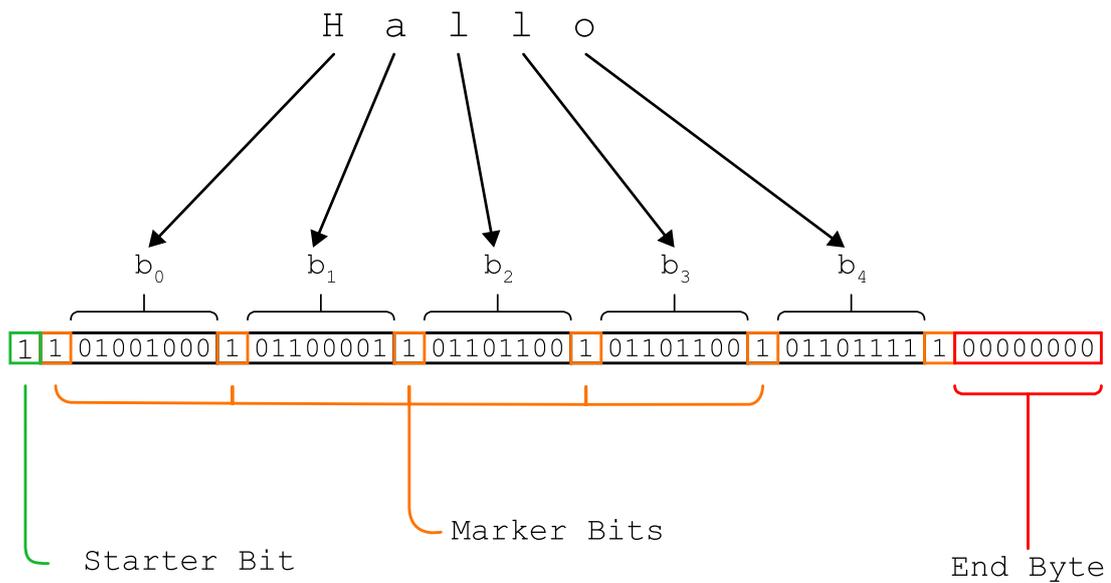


Abbildung 3.5: Schema des Datenübertragungsprotokolls unter Verwendung des Wortes „Hallo“ als Beispiel Datensatz

Daten ursprünglich in Utf formatiert waren, wird dieser Codec auch hier verwendet. Im Anschluss werden die Daten über den bereits erwähnten Http-Webserver zur Verfügung gestellt und gespeichert.

3.1.3 Übertragungsprotokoll

Damit die Signale, die die DÜ-Einrichtung überträgt, richtig interpretiert werden können, muss sich auf ein gewisses *Übertragungsprotokoll* geeinigt werden. Dieses Protokoll wird Hauptinhalt dieses Abschnittes. Als Schema dient Abbildung 3.5. Jede Übertragung beginnt mit einem „Starter“-Bit. Dieser dient dazu die in 3.1.2. *Empfänger* beschriebene Empfänger Funktion zu starten. Daraufhin folgen die einzelnen Pakete, die die Daten beinhalten. Ein Paket besitzt die Größe eines Bytes und wird durch einen hohen Marker-Bit angekündigt. In diesem Fall besitzt das Wort „Hallo“ fünf Zeichen und somit auch fünf Pakete. Am Ende der Übertragung steht ein festgelegter Byte der anzeigt, dass die Übertragung beendet ist. Bei diesem handelt es sich um 00000100 oder dezimal 4.

Mit diesem, sehr einfachen und nicht gegen Fehler abgesicherten, Protokoll lassen sich



Abbildung 3.6: Beispiel des Versuch-Aufbaus bei der Distanz $1m$

erste Daten übertragen.

3.2 Durchführung

Bei der Durchführung wird zwischen drei Versuchsreihen unterschieden. Diese unterscheiden sich dabei in ihrer Distanz zwischen Sender und Empfänger. Dabei handelt es sich um $1m$ (Abbildung 3.6), $2m$ und $10m$.

Alle Übertragungen wurden bei gleichen äußeren Lichtbedingungen durchgeführt. Jede dieser Versuchsreihen setzt sich aus sechs Übertragungen zusammen, die drei Texte mit zwei verschiedenen DTR übertragen. Bei diesen Texten handelt es sich erstens um das erste Kapitel von Goethes Faust, zweitens um den Liedtext von Peter Fox, Alles Neu und drittens das Zitat „Es gibt eine Theorie, die besagt, wenn jemals irgendwer genau herausfindet, wozu das Universum da ist und warum es da ist, dann verschwindet es auf der Stelle und wird durch noch etwas Bizarrereres und Unbegreiflicheres ersetzt. - Es gibt eine andere Theorie, nach der das schon passiert ist.“ aus Douglas Adams, Per Anhalter durch die Galaxis. Vor jeder Übertragung wird der Schwellwert U_t (vgl. Gleichung 3.7) neu bestimmt. Im Anschluss wird die Übertragung durchgeführt und die Ergebnisse, in Form der rohen empfangenen Bytes, gespeichert.

#	Distanz (m)	Text	DTR (bit/s)	BER	Korrekt ü. Byte
1	1m	1	32bit/s	0,390	122
2	1m	2	32bit/s	0,392	179
3	1m	3	32bit/s	0,103	238
4	1m	1	64bit/s	0,438	90
5	1m	2	64bit/s	0,437	128
6	1m	3	64bit/s	0,205	179
7	2m	1	32bit/s	0,389	117
8	2m	2	32bit/s	0,386	175
9	2m	3	32bit/s	0,106	230
10	2m	1	64bit/s	0,444	79
11	2m	2	64bit/s	0,654	110
12	2m	3	64bit/s	0,211	177
13	10m	1	32bit/s	0,430	100
14	10m	2	32bit/s	0,450	79
15	10m	3	32bit/s	0,324	90
	10m	1	64bit/s	/	/
16	10m	2	64bit/s	0,446	119
17	10m	3	64bit/s	0,340	87

Tabelle 3.1: Ergebnis Tabelle mit bit Error Ratio und Anzahl der korrekt übertragenen Byte

3.3 Ergebnisse

Für die Auswertung der Ergebnisse wird sich der analytischen Methode der bit Error Ratio (BER) bedient. Diese wird durch

$$ber = \frac{n_{error}}{n_{total}} \quad (3.11)$$

berechnet. Um falsch übertragene Bits nun zu verhindern ist eine möglichst niedrige BER anzustreben. Alle auf diese Weise berechneten Werte sind der Tabelle 3.1 zu entnehmen. Im Folgenden sollen die Daten auf ihre Auffälligkeiten analysiert werden.

Bei der Analyse der Daten lassen sich mehrerer Muster erkennen. Im Allgemeinen zeigt sich, dass je kürzer die zu übertragenden Daten sind, desto kleiner ist die BER. Es scheint dem Empfänger, selbst durch Verwenden der Pakete, nicht möglich zu sein, sich effektiv

zu synchronisieren. Dies liegt vermutlich an dem in 2.1.3. *Übertragung* beschriebenen Effektes des Taktunterschieds. Im Weiteren ist zu erkennen, dass bei höherer DTR die BER steigt. Dies ist von Nr. 4 zu Nr. 6 zu erkennen, bei der dieser Wert um $\approx 10\%$ -Punkte anwächst. Allgemein ist bei Dopplung der Geschwindigkeit eine Steigerung von ungefähr 5%- bis 10%-Punkten zu erkennen. Bei Erhöhung des Abstands von $1m$ auf $2m$ ist eine Veränderung der BER zu erkennen. Tatsächlich ist von Nr. 1–3 auf Nr. 7–9 sogar eine Senkung von 1%-Punkt zu erkennen, was allerdings auf Messfehler zurückzuführen ist. Bei Erweiterung des Abstandes von $1m$, beziehungsweise $2m$, auf $10m$ zeigen sich Steigerungen der BER. Dennoch muss beachtet werden, dass sich dieser Effekt auf 4%- bis 5%-Punkte beschränkt und somit einen proportional deutlich geringeren Einfluss, als beispielsweise die Erhöhung der DTR, hat.

Auffällig ist die bei Nr. 11 zu erkennende Anomalie. Diese fällt mit einer sehr hohen BER aus der Reihe Nr. 10–12. Diese Messung lässt auf eines der Hauptprobleme des Versuches schließen.

Durch die sehr simple Implementierung und die ineffiziente Messapparate ist dieser sehr unzuverlässig. Viele Übertragungen schlagen fehl, weil der anfänglich berechnete Schwellwert U_t (Gleichung 3.7) zu hoch oder zu niedrig angesetzt wurde. Auch kann es durch den ADC vereinzelt zu Fehlmessungen kommen, die von Code 3.1.2 als hoher Bit interpretiert werden und so zu falschen Zeitpunkten den Empfang eines neuen Paketes auslösen. Dieser Messfehler führt zu großen Taktunterschieden und verfälscht die gesamte nachfolgende Übertragung. So war bei $10m$ Abstand, Text 1 und $64^{bit}/s$ gar keine Messung möglich. Im Allgemeinen fluktuieren Messungen des ADC sehr stark abhängig von der DTR. Aus diesem Grund limitieren sich die Testreihen auch auf $32^{bit}/s$ und $64^{bit}/s$. Bei höheren Werten sind keine sinnvollen Messungen mehr möglich.

Doch nicht nur die reinen Messergebnisse, sondern auch die Erfahrungen bei der Einrichtung dieses Versuches haben in diesem Versuchsaufbau Relevanz. Während sich die

Zusammenstellung und -bau der Einheiten dank der Flexibilität des *RP2040* als mehr als einfach herausstellte, erwies sich die Programmierung als Herausforderung. Das Projekt enthält insgesamt ca. 900 Zeilen *MicroPython* Code und ca. 600 Zeilen *Hypertext Markup Language (HTML)*. Problematisch war hierbei jedoch nicht das Schreiben der einzelnen Funktionen, sondern das Fehlertesten dieser. Dadurch das es sich beim *RP2040* um einen Micro-Chip handelt, und nicht um eine Central Processing Unit (CPU), existiert kein Betriebssystem zwischen Soft- und Hardware, das das Erstellen von Programmen vereinfacht. Das bedeutet, dass diese nur auf dieser speziellen Hardware laufen und somit nicht direkt am Computer ausgeführt und getestet werden. Dieser Umstand machte das Suchen von Fehlern sehr zeitintensiv und wenig effizient. Auch die Tatsache, dass sich der Versuch in zwei isolierte Einheiten unterteilt, machte den Prozess der Fehlersuche nicht einfacher. Bei Beobachtung eines Fehlers gab es so immer zwei mögliche Auslöser des Fehlers, was die Suche deutlich verkomplizierte. Das Ansteuern des L.A.S.E.R. stellte allerdings keine Probleme dar. Dieser konnte durch simples An- und Abschalten der Versorgungsspannung an- beziehungsweise ausgeschaltet werden.

Trotz dieser Probleme bei der Einrichtung des Versuches und Schwierigkeiten bei der Erhebung der Messdaten, liefert der Versuch brauchbare Ergebnisse. Es ist, bei langsamer DTR und kleiner Datenmenge so durchaus möglich erfolgreiche Übertragungen zu tätigen.

4 | Zusammenfassung

Auf Grundlage der in *2. Theorie* und *3. Versuch* erlangten Erkenntnisse verfolgt dieses Kapitel den Ansatz diese beide in Einklang zu bringen und eine Art Fazit zu bieten.

Während *2. Theorie* die großen theoretischen Vorteile des L.A.S.E.R., wie die große Reichweite, aufzeigt, erweist sich die tatsächliche Umsetzung in *3. Versuch* als eher schwierig. Im Vergleich zu üblichen Datenübertragungswegen wie einem Kabel hat L.A.S.E.R., oder EMS im allgemeinen, große Probleme mit Noise. Dies zeigt sich auch in Tabelle 3.1. Dennoch lassen sich auch hier die großen Vorteile des L.A.S.E.R. ablesen. Nachdem sowohl die theoretischen (*2. Theorie*) als auch praktischen (*3. Versuch*) Aspekte des L.A.S.E.R. behandelt wurden, werden diese in Bezug zu dem in *1. Einleitung* beschriebenen Forschungsansatz gesetzt.

Durch den zusammenfassenden theoretischen Überblick der benötigten Theorie in 2 kann *3. Versuch* nahtlos an diese anschließen und, wichtiger, darauf aufbauen. Es wird beschrieben wie die Theorie umgesetzt wird und zwei funktionsfähige Übertragungs-Einheiten gebaut werden. Diese werden dann auch in einem Versuchsaufbau mit verschiedenen Versuchsparametern getestet und die Ergebnisse auf Auffälligkeiten analysiert. Auch wenn sich hier herausstellte, dass die Einheiten sehr fehleranfällig sind und so meist an die Hälfte aller Bits falsch übertragen, ist es ihnen dennoch möglich erfolgreich kleine Datenmengen mit geringer Geschwindigkeit zu übertragen. Obwohl der Versuch keine fehlerfreien Ergebnisse liefert, lässt sich die Technologie anhand diesem dennoch einordnen. Der L.A.S.E.R. als Datenübertragungsweg eignet sich vor allem in Fällen, bei denen die Daten über sehr weite Strecken gesendet werden sollen. Wie auch in *3.1. Ergebnis Tabelle mit bit Error*

Ratio und Anzahl der korrekt übertragenen Byte zu erkennen, hat der L.A.S.E.R. wenig Probleme, wenn die Distanz erhöht wird (vgl. 2.2. *Laser als Datenübertragungsweg*). Diese Tatsache, kombiniert mit dem Fakt, dass der L.A.S.E.R. kein Medium zur Ausbreitung benötigt, lässt auf einen sehr logischen Verwendungszweck schließen. L.A.S.E.R. findet, wie in 1. *Einleitung* erwähnt, hauptsächlich in der Satelliten-Technologie Verwendung. Hier ermöglichen sie eine effiziente, latenzniedrige Übertragung von Daten im Terabit Bereich.

Aus Sicht eines Amateurs hat diese Art der Datenübertragung den Vorteil, das auf diese Weise Daten über weite Distanzen möglichst billig übertragen werden können. Allerdings muss hier immer abgewägt werden wie wichtig Geschwindigkeit und Zuverlässigkeit für Übertragung sind.

Abschließend ist zu sagen das der L.A.S.E.R. ein wirklich Innovativer und neuartiger Datenübertragungsweg ist die Möglichkeiten der weltweiten, und womöglich auch, Intraplanetaren-Vernetzung für immer verändern werden.

Abkürzungsverzeichnis

ADC	Analog-Digital-Umsetzer. 24, 29, 30, 32, 37
Ascii	American Standard Code for Information Interchange. 17, 18, 43
ASK	Amplitudenumtastung. 13, 16, 20–22, 43
BER	bit Error Ratio. 36, 37, 39, 44
Bit	binary digit. 10, 12, 14, 15, 18, 19, 24, 29, 33, 34, 36, 39
BPSK	Binäre Phasenmodulation. 15
CPU	Central Processing Unit. 38
DE-Einrichtung	Dateneneinrichtung. 6, 7, 18, 19, 21, 25, 30, 33
DSL	Digital Subscriber Line. 11
DTR	Daten Übertragungsgeschwindigkeit. 22, 29, 35–38, 43
DÜ-Einrichtung	Datenübertragungseinrichtung. 6, 7, 18–20, 25, 28, 30, 34
EMS	elektromagnetischer Strahlung. 20, 39
FSK	Frequenzumtastung. 14
GPIO	General Purpose Inputs/Output. 24, 25
GPS	Global Positioning System. 21
HTML	Hypertext Markup Language. 38

Http	Hypertext Transfer Protocol. 17, 25, 28, 34
I/O	Input/Output. 24
IT	Informations-Technologie. 5
L.A.S.E.R.	Light Amplification by Stimulated Emission of Radiation. 3, 5, 20, 21, 23–25, 27, 29, 30, 38–40, 43
LED	Leuchtdiode. 25, 27, 29–31, 43
PSK	Phasenmodulation. 15, 16
QAM	Quadratur-Amplituden-Modulation. 16
RFID	Radiofrequenz-Identifikation. 16
UHF	Ultra-High-Frequency. 20, 21
Utf	Unicode Transformation Format. 17, 18, 34
WLan	Wireless Local area network. 16, 21

Abbildungsverzeichnis

2.1	Schematische Darstellung der DIN 44302 [Autoren der IT, 1966]	6
2.2	Blockdiagramm einer Datenübertragung zwischen zwei Computer unter Verwendung von Telefon Modem	8
2.3	Graph einer typischen analogen Funktion ($\frac{1}{2} \cdot \sin(x) + \frac{1}{2}$)	9
2.4	Graph einer typischen digitalen Funktion	10
2.5	Beispiel Signal Regeneration eines Binären digitalen Signals [Sklar and Harris, 2019]	12
2.6	Übersicht der Arten der Modulation: [Bakni, 2021]	13
2.7	Schema der Formatierung und Übertragung von Signalen [Sklar and Harris, 2019]	17
2.8	Ascii-Tabelle aller 128 Zeichen aus MIL-STD-188-100 (1972)	18
2.9	DTR Limitationen bei ASK-Modulation am Beispiel eines Carrier-Signals bei verschiedenen Geschwindigkeiten	22
3.1	Bild des Senders	26
3.2	Schaltplan des Senders mit L.A.S.E.R. und Indikator LED	27
3.3	Bild des Empfängers	30
3.4	Schaltplan des Empfängers mit Foto-Diode und Indikator LED	31
3.5	Schema des Datenübertragungsprotokolls unter Verwendung des Wortes „Hallo“ als Beispiel Datensatz	34
3.6	Beispiel des Versuch-Aufbaus bei der Distanz $1m$	35

Tabellenverzeichnis

3.1	Ergebnis Tabelle mit bit Error Ratio und Anzahl der korrekt übertragenen Byte	36
-----	--	----

Literaturverzeichnis

- Sardar H Ali. Advantages and limits of free space optics. 9(3):1–6, 2019. doi: 10.5121/ijassn.2019.9301.
- Autoren der IT. DIN 44302 datenübertragung - begriffe. 8(1-6):244–246, 1966. doi: 10.1524/itit.1966.8.16.244.
- Michel Bakni. Modulation Übersichts bild. CC BY-SA 4.0 <<https://creativecommons.org/licenses/by-sa/4.0>>, via Wikimedia Commons, 2021. URL https://commons.wikimedia.org/wiki/File:Modulation_categorization.svg. Accessed: 2023-11-4.
- Tim Berners-Lee. Proposal for an information management system, 1989. URL <http://info.cern.ch/Proposal.html>. Accessed: 2023-11-4.
- Ottmar Beucher. *Signale und Systeme: Theorie, Simulation, Anwendung*. Springer Berlin Heidelberg, 2019. doi: 10.1007/978-3-662-58044-8.
- John R Carson. Notes on the theory of modulation. *Proceedings of the institute of radio engineers*, 10(1):57–64, 1922.
- CERN. The birth of the web, 2023. URL <https://home.web.cern.ch/science/computing/birth-web#:~:text=TimBerners-Lee,aBritish,andinstitutesaroundtheworld>. Accessed: 2023-11-4.
- Volker Claus and Andreas Schwill. *Duden Informatik*. Dudenverlag, 1993. ISBN 3411052325.
- Cypress. *CYW43439*. Infineon Technologies Company, 2023. URL https://www.infineon.com/dgdl/Infineon-CYW43439-DataSheet-v05_00-EN.pdf?fileId=8ac78c8c8929aa4d01893ee30e391f7a. Accessed: 2023-11-4.

- Deutsches Institut für Normung e.V. Din 66020: Funktionelle anforderungen an die schnittstellen zwischen datenendeinrichtung und datenübertragungseinrichtungen, 1966.
- Deutsches Institut für Normung e.V. Din 44302: Informationsverarbeitung; datenübertragung, datenübermittlung; begriffe, 1987.
- Deutsches Institut für Normung e.V. Din en 60825-1: Sicherheit von lasereinrichtungen - teil 1: Klassifizierung von anlagen und anforderungen, 2022.
- Damien P. George. Micropython, 2023. URL <https://micropython.org/>. Accessed: 2023-11-4.
- International Telecommunication Union. Iso 646: Informationstechnik; iso 7-bit codierter zeichensatz für informationsaustausch, 1991. URL <https://www.itu.int/rec/T-REC-T.50-199209-I/en>. Accessed: 2023-11-4.
- T. H. MAIMAN. Stimulated optical radiation in ruby. 187(4736):493–494, 1960. doi: 10.1038/187493a0.
- Alfred Mertins. *Signaltheorie*. Springer Fachmedien Wiesbaden, 2020. doi: 10.1007/978-3-658-29648-3.
- John Proakis and Masoud Salehi. *Digital Communications*. McGraw-Hill Science/Engineering/Math, 2007. ISBN 9780072957167.
- Raspberry Pi Ltd. *RP2040 Datasheet*. 2023. URL <https://datasheets.raspberrypi.com/rp2040/rp2040-datasheet.pdf>. Accessed: 2023-11-4.
- Bernard Sklar and Fredric J. Harris. *Digital Communications Fundamentals and Applications*. Pearson Higher Education and Professional Group, 2019. ISBN 9780134588568.
- Starlink. Starlink, 2023. URL <https://www.starlink.com>. Accessed: 2023-11-4.
- Statistisches Bundesamt. Daten aus der evs zur ausstattung privater haushalte mit informations- und kommunikationstechnik in deutschland, 2018. URL <https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Einkommen-Konsum-Lebensbedingungen/Ausstattung-Gebrauchsgueter/Tabellen/a-evs-infotechnik-d.html>. Accessed: 2023-11-4.

Harold P. E. Stern and Samy A. Mahmoud. *Communication Systems*. Prentice Hall, 2003.
ISBN 9780130402684.

Schülererklärung

Ich habe diese Seminararbeit ohne fremde Hilfe angefertigt und nur die im Literaturverzeichnis angeführten Quellen und Hilfsmittel benutzt.

Ort, Datum: _____ Unterschrift: _____