

Realisierung eines Sensornetzwerkes mit CDMA *Indoor Optical Wireless*

Diplomarbeit

durchgeführt von

Andreas Brunner

Institut für Breitbandkommunikation
der technischen Universität Graz
Leiter: Univ.-Prof. Dipl.-Ing. Dr. Gernot Kubin

Eidesstattlich Erklärung

Hiermit erkläre ich, dass ich die vorliegende Diplomarbeit selbständig angefertigt habe. Es wurden nur die in der Arbeit ausdrücklich benannten Quellen und Hilfsmittel benutzt. Wörtlich oder sinngemäß übernommenes Gedankengut habe ich als solches kenntlich gemacht.

Ort, Datum

Unterschrift

Kurzfassung

Das Ziel dieser Diplomarbeit ist es ein *low cost* Sensornetzwerk zu realisieren, welches die zu übertragenden Daten mittels infrarotem Licht im Wellenlängenbereich von 850 – 900 nm kabellos überträgt. Da das Netzwerk in Innenräumen verwendet wird, werden Decke und Wände als Reflektoren für die Lichtstrahlen genutzt, um bei einer Unterbrechung der direkten Übertragungstrecke eine Aufrechterhaltung der Verbindung sicher zu stellen.

Nach einer kurzen Einleitung wird in Kapitel 2 der Aufbau eines solchen Netzwerkes erklärt. Der folgende Teil der Arbeit umfasst die gewählte Modulationsart, das Vielfachzugriffsverfahren CDMA und die Kanalcodierung. Das CDMA Verfahren ermöglicht mehreren Teilnehmern ihre Daten asynchron zu übertragen. Die Kanalcodierung beschränkt sich auf eine Paritätsbit Überprüfung. Im nächsten Kapitel werden für die Sende- und Empfängerschaltung die Bauteile dimensioniert, ebenso werden der Aufbau und die Funktionsweise beider Schaltungen erläutert. Kapitel 5 beschäftigt sich mit dem verwendeten AVR-Net-IO-Board und der Software welche für die Sensor- und Hauptstation implementiert und genutzt wurde. Danach folgen eine Leistungsbilanz (Linkbudget) sowie eine Reichweitenmessung und eine Zusammenfassung.

Abstract

The goal of this master thesis is to realize a low cost wireless sensor network transmitting the data with infrared light at a wavelength of 850 – 900 nm. The network is used for indoor applications. The ceiling and the walls can be seen as reflectors. Those reflectors ensure the connection if the line of sight link is lost.

After a short introduction the configuration of such a network is described in chapter 2. The following part of the thesis is dealing with the chosen modulation type, the multiple access method CDMA and the channel coding. The CDMA method allows the users to transmit their data asynchronous. The channel coding is limited to a parity check. The next chapter specifies the prefabricated parts for the transmit- and receive circuit, it is also given an explanation that shows how both circuit are built and work. Chapter 5 is dealing with the used AVR-Net-IO-Board and the software for the sensor- and main station. Finally the link budget is calculated, distance measurements are done followed by the conclusions.

Inhaltsverzeichnis

Kurzfassung / Abstract	1
Inhaltsverzeichnis	2
1 Einleitung und Motivation.....	4
2 Optical Wireless Indoor Communication.....	6
2.1 Aufbau des Netzwerkes.....	6
2.2 Prinzip des LSMS Sendesystems.....	8
2.3 Prinzip des <i>Angle Diversity Receiver</i> Systems.....	10
3 Modulation, CDMA und Kanalcodierung.....	13
3.1 Modulation.....	13
3.2 CDMA und Codewortgenerierung.....	14
3.3 Kanalcodierung	20
3.4 Datenübertragungsrate und Datendurchsatz	22
4 Dimensionierung der Sende- und Empfängerschaltung	24
4.1 Sendeschaltung.....	24
4.1.1 BauteilAuswahl	24
4.1.2 Dimensionierung der Sendeschaltungen	24
4.2 Empfängerschaltung.....	28
4.2.1 Ausgewählte Bauteile	29
4.2.2 Dimensionierung der Empfängerschaltung	30
5 Implementierung der Sensor- und Hauptstation	41
5.1 AVR-Net-IO-Board.....	41
5.2 Hauptstation	42
5.3 Sensorstation	46
5.3.1 Testsensor	47
5.3.2 Software der Sensorstationen	48
6 Leistungsbilanz (Linkbudget) und Reichweitenmessung	50
7 Zusammenfassung und Ausblick.....	55
Literaturverzeichnis	56
Abbildungsverzeichnis	58
Tabellenverzeichnis	59
Abkürzungsverzeichnis	60

Anhang A: Software der Haupt- und Sensorstation.....	61
A1: Software der Hauptstation	61
A2: Software der Sensorstation	66

1 Einleitung und Motivation

Das Ziel dieser Diplomarbeit ist, ein *low cost* Sensornetzwerk zu entwerfen, welches die zu übertragenden Daten mittels infrarotem Licht im Wellenlängenbereich von 850 – 900 nm kabellos überträgt.

Das Netzwerk wird von einer Hauptstation gesteuert, deren Hauptaufgabe darin besteht die Daten, welche von den Sensorstationen kontinuierlich gesendet werden, zu empfangen und zur Weiterverarbeitung bereitzustellen. Des Weiteren soll es der Hauptstation möglich sein mit steuernden Befehlen das Netzwerk zu kontrollieren, der Informationsfluss soll somit auf bidirektionalem Wege funktionieren. Die Sensorstationen sind mit einem Temperatur- und Luftfeuchtigkeitssensor ausgestattet, welcher seine Messdaten via I²C Bus an den Mikrocontroller überträgt. Von dort aus werden die Daten mittels Intensitätsmodulation zur Hauptstation gesendet.

Da das Netzwerk für Innenräume konzipiert wird, sind die Sender und Empfänger so zu realisieren, dass die Übertragung nicht abbricht, wenn die direkte Verbindung unterbrochen wird. Das bedeutet, dass die Daten via Reflexionen an der Decke und den Wänden zum Empfänger gelangen sollen. Die folgende Abbildung 1 zeigt schematisch den Aufbau einer solchen Übertragungsstrecke.

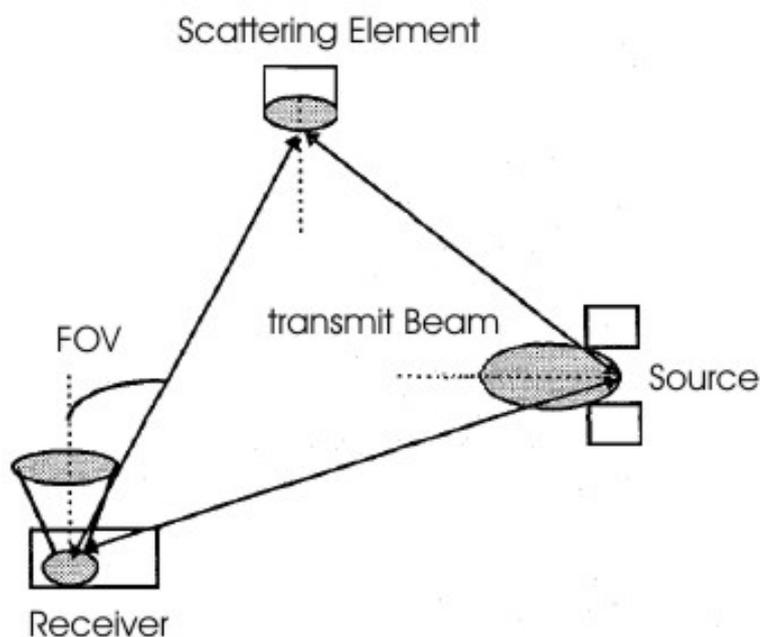


Abbildung 1: Darstellung der Übertragungsstrecke, übernommen aus [6]

Um den Vielfachzugriff auf den Übertragungskanal dieses Netzwerkes zu regeln, wird das CDMA (*Code Division Multiple Access*) Verfahren angewendet. Da sich das Netzwerk aus 2 Sensorstationen und der Hauptstation zusammensetzt, können 8 Bit lange Codewörter pro Teilnehmer verwendet werden, siehe Kapitel 3.2. Anstatt eine binäre 1

aus einem Datenwort zu senden, übertragen die Teilnehmer das ihnen zugewiesene Codewort, über das sie eindeutig identifizierbar sind. Auf Grund des CDMA Verfahrens und der *low cost* Anforderung, welche sich auf die Wahl der Mikrocontroller in Bezug auf die Taktgeschwindigkeit auswirkt, beschränkt sich die Chiprate auf 40 kbit/s. In Kapitel 5.2 wird auf das Zustandekommen der Chiprate näher eingegangen.

Die Motivation ein kabelloses Infrarot Sensornetzwerk zu entwerfen, entstammt dem Gedanken ein Netzwerk zu entwickeln, welches als Alternative zu WLANs (*Wireless Lokal Area Network*) oder Bluetooth verwendet werden kann.

Des Weiteren sollte dieses Sensornetzwerk als Grundlage für weitere Entwicklungsarbeiten auf diesem Gebiet dienen.

2 Optical Wireless Indoor Communication

Bei optischen Freiraumübertragungen, die in Innenräumen arbeiten, ergeben sich einige Vorteile gegenüber einem üblichem WLAN Netzwerk. Im Vordergrund steht die Abhörsicherheit, da die zu übertragenden Daten oder Informationen nur von einem Teilnehmer im selben Raum empfangen werden können. Anders als bei WLANs hat man eine höhere nutzbare Bandbreite, da bei optischen Übertragungen keine Gefahr besteht ein anderes Frequenzband zu stören oder zu beeinflussen. Dies ist auf die Wellennatur des Lichts und die hohe Trägerfrequenz zurückzuführen [2]. Aus diesem Grund sind optische IR (Infrarot) Verbindungen gut für *High Speed* Anwendungen geeignet. Des Weiteren gibt es bei der optischen Übertragung keine Beschränkungen der Frequenz und es ist nicht notwendig Lizenzen zu erwerben [9].

2.1 Aufbau des Netzwerkes

Es gibt verschiedene Möglichkeiten eine optische kabellose Verbindung zu konfigurieren [1]. Man kann prinzipiell zwischen zwei Arten von Verbindungen unterscheiden. Die direkte Verbindung zwischen Sender und Empfänger wird kurz LOS (*Line of Sight*) genannt. Die indirekte Verbindung über Reflexionen an der Decke und den Wänden wird als *non-LOS* bezeichnet. Des Weiteren ergeben sich durch verschiedene Abstrahlwinkel des Senders und Öffnungswinkel des Empfängers einige Möglichkeiten die Verbindung zu realisieren. Diese sind in Abbildung 2 dargestellt.

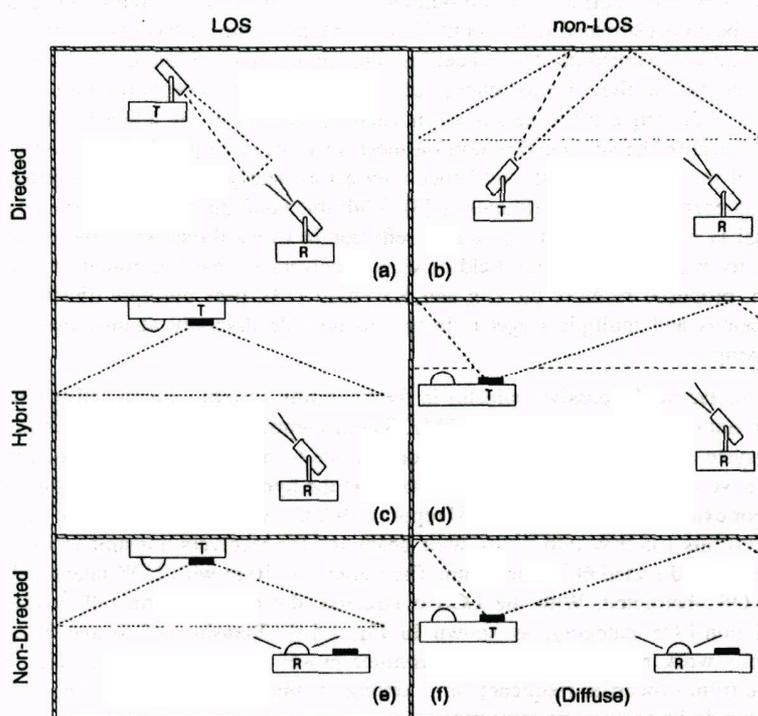


Abbildung 2: Mögliche Linkkonfigurationen, entnommen aus [1]

Bei der Frage nach der *Link* Konfiguration wird für diese Arbeit ein Hybrider Typ, siehe Abbildung 2, gewählt. Die Sendedioden der Haupt- und Sensorstationen sind gerichtete LEDs (*Light Emitting Diode*) mit einem Abstrahlwinkel von 40° . Diese werden so wohl in LOS als auch in *non*-LOS betrieben. Die Anordnungen der LEDs für die Haupt- und Sensorstationen werden in Kapitel 2.2 eingehend beschrieben.

Die PIN (*Positive Intrinsic Negative*) Photodioden sind sowohl bei der Hauptstation als auch bei den Sensorstationen mit einem Öffnungswinkel, kurz FOV (*Field of View*), von 120° gewählt worden. Jede Station ist mit nur einer Photodiode ausgestattet. In Kapitel 2.3 wird gezeigt wie ein Empfänger mit mehreren Photodioden aussehen kann. Der Vorteil eines solchen Empfängers liegt in der besseren Rundumerfassung der Lichtphotonen und einer besseren Performance. Diese Methode wird aufgrund der Tatsache das nur beschränkte finanzielle Mittel zur Verfügung stehen nicht realisiert.

Da die Aufgabe dieses Netzwerkes darin besteht, die gemessenen Daten von den Sensoren, welche überall im Raum angeordnet sein können, zu der Hauptstation zu übertragen, bietet es sich an das Netzwerk in einer Sterntopologie [3], wie Abbildung 3 darstellt, aufzubauen.

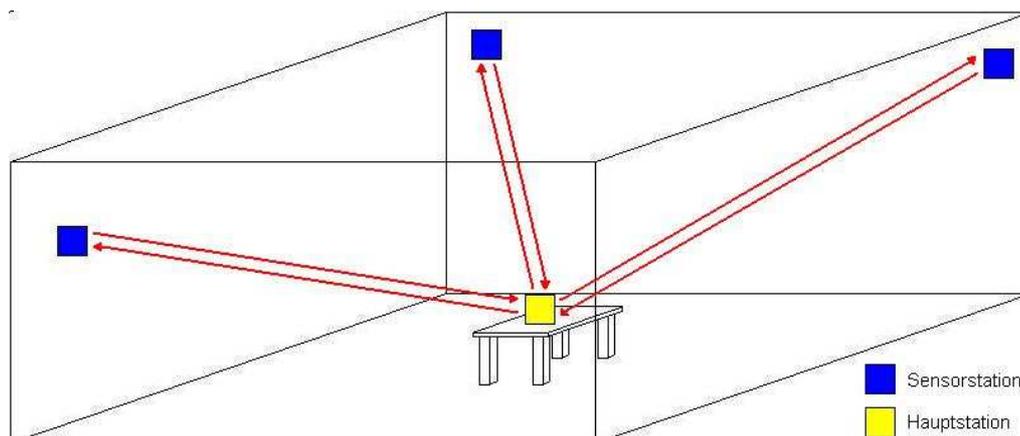


Abbildung 3: Sternförmiger Netzwerkaufbau

Die Hauptstation sollte dabei im Zentrum des Raumes positioniert werden. Sind die Teilnehmer rund um die Hauptstation verteilt, empfiehlt es sich die Photodiode mit Blick in Richtung Decke auszurichten. So können die Signale von jeder Position im Raum aus direkt oder indirekt Verbindung an die fotoempfindliche Fläche der Diode gelangen.

2.2 Prinzip des LSMS Sendesystems

Das LSMS (*Line Strip Spot-diffusing Transmitter System*) Sendesystem eignet sich gut um Signale über Reflexionen an der Decke und den Wänden zu den im Raum verteilten Empfängern zu senden. Dieses System wurde bereits vor einigen Jahren in [7] und [8] vorgestellt.

Anstatt einer LED mit einem weiten Abstrahlwinkel, wird bei diesem System auf mehrere LEDs mit einem schmalen Abstrahlwinkel zurückgegriffen. Ein Beispiel dafür ist in der folgenden Abbildung 4 dargestellt.

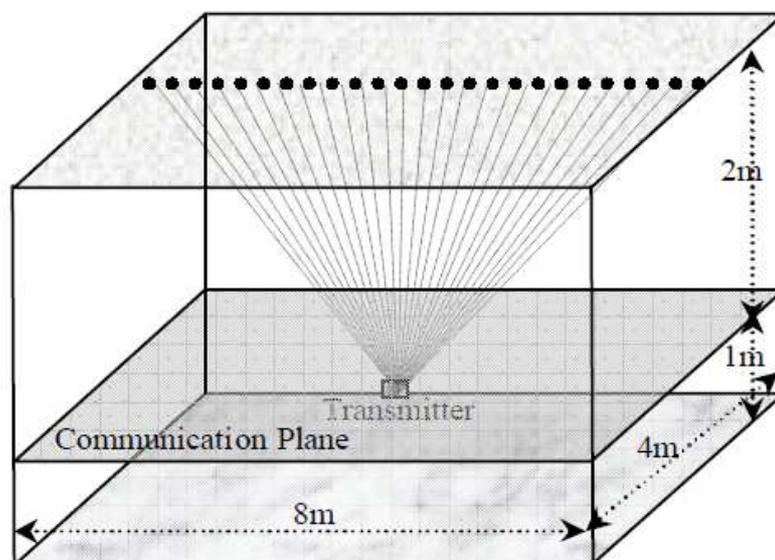


Abbildung 4: LSMS System, aus [7]

Der Vorteil eines LSMS Systems gegenüber einer weit abstrahlenden LED sind die vielen Reflexionsspot an der Decke und den Wänden, welche durch den geringeren Abstrahlwinkel eine höhere Strahlintensität in sich vereinen. Dadurch kann die Reichweite eines Übertragungssystems erhöht werden und es können bei gewissen Modulationsverfahren höhere Übertragungsraten erzielt werden.

Anders als in Abbildung 4 werden für die Sendeeinheit der Sensorstation nur 3 LEDs verwendet. Die geometrische Anordnung der LEDs und eine mögliche Positionierung im Raum werden in Abbildung 5 gezeigt. Die Sendeeinheit muss in Richtung der Hauptstation justiert werden, damit die zentral angebrachte LED eine LOS-Verbindung herstellen kann. Die zwei außen platzierten LEDs übertragen das Signal mittels einer *non-LOS*-Verbindung zum Empfänger der Hauptstation.

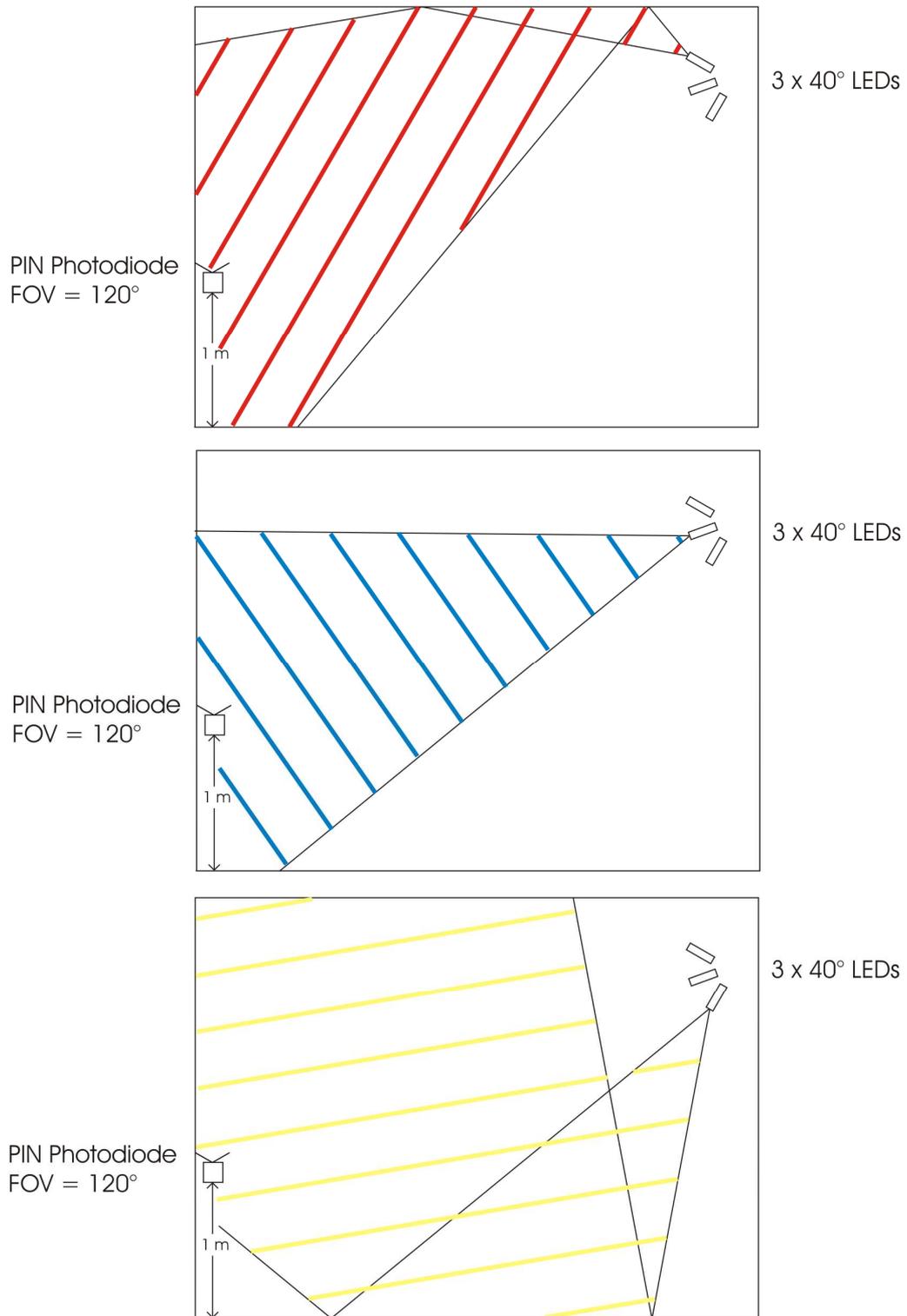


Abbildung 5: Geometrische Anordnung der LEDs für die Sensorstationen

Die Sendeeinheit der Hauptstation wird ebenfalls nach dem LSMS Prinzip realisiert, jedoch werden im Gegensatz zur Sendeeinheit der Sensorstation 9 anstatt von 3 LEDs verwendet. Der Grund dafür liegt darin, dass die Signale der Hauptstation alle verteilten Sensorstationen im Raum erreichen sollen. Da sich die Hauptstation im Zentrum des Raumes befindet, bietet es sich an die LEDs so zu positionieren, dass sie eine Art Halbkugelstrahler ergeben. In Abbildung 6 ist diese Anordnung dargestellt.

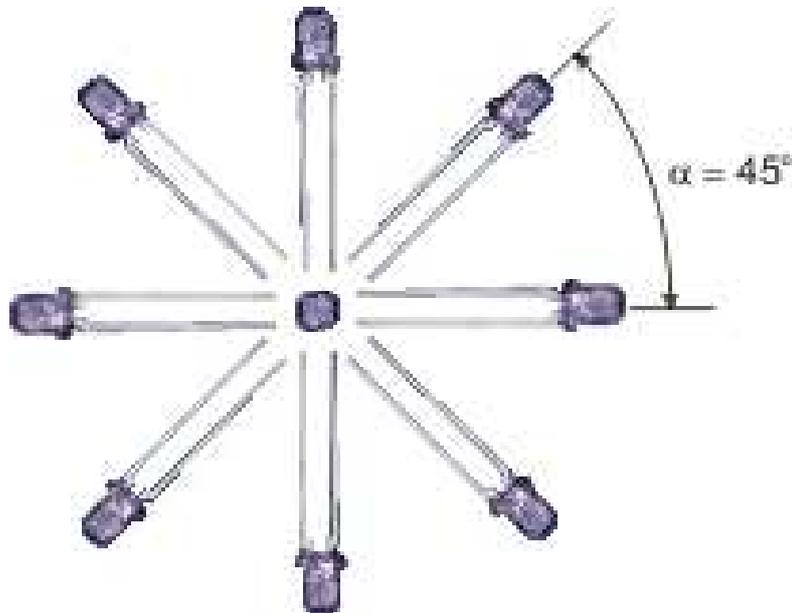


Abbildung 6: Geometrische Anordnung der LEDs für die Hauptstation

2.3 Prinzip des *Angle Diversity Receiver Systems*

Bei dem *Angle Diversity Receiver* Verfahren werden mehrere Empfänger mit einem schmalen FOV anstatt eines Empfängers mit einem weiten FOV verwendet [8]. Die Sendeeinheit überträgt OOK (*On Off Keying*) Pulse, welche intensitätsmoduliert sind. Der *Angle Diversity Receiver* funktioniert mittels direkter Signaldetektion [9]. Abbildung 7 zeigt einen konventionellen Empfänger mit einem weiten FOV. In Abbildung 8 ist der grundlegende Aufbau eines *Angle Diversity Receivers* dargestellt.

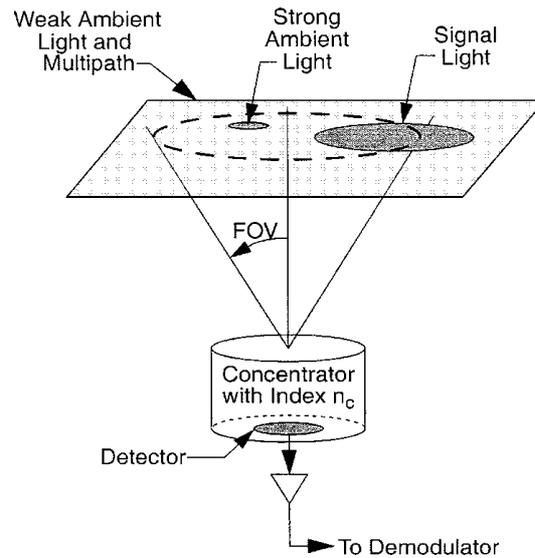


Abbildung 7: Einzelempfänger mit optischer Sammellinse, aus [8]

Das eintreffende Licht, welches von einer direkten Verbindung oder von reflektierenden Spots stammt, wird von einer Sammellinse auf die fotoempfindliche Fläche des Empfängers konzentriert. Die PIN Diode treibt proportional zu den eintreffenden Lichtphotonen einen Fotostrom, welcher in eine Spannung umgewandelt und schließlich demoduliert wird.

Die in der gegenständigen Arbeit verwendeten Empfänger besitzen anders als in Abbildung 7 und 8 keine Sammellinsen oder Filter vor der PIN Diode. Das senkt zwar die Leistung dieser Übertragung, ist aber auch bedeutend billiger.

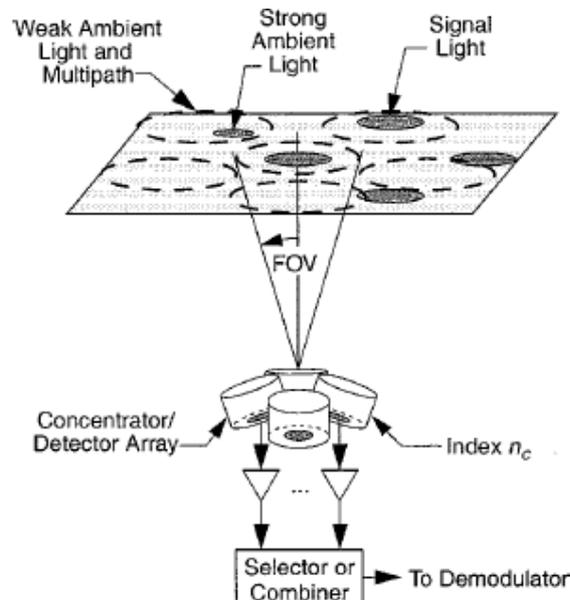


Abbildung 8: Array von schmalen FOV Empfängern, aus [8]

Die Kombination eines LSMS Systems mit einem *Angle Diversity Receiver* bietet einige Vorteile gegenüber einem konventionellen System. Durch den geringeren Öffnungs-

winkel der Empfänger trifft weniger unerwünschtes Licht in den Betrachtungsabschnitt, z.B. Sonnenlicht, Licht von Neonröhren oder Signale die bereits zwei oder mehrmals reflektiert wurden. Dies führt zu einer geringeren Verzerrung des Signals durch die Mehrwegausbreitung und einer besseren Rauschunterdrückung.

Das in Abbildung 9 zu sehende Blockschaltbild zeigt ein solches Empfängersystem, welches mit optischen Konzentratoren ausgestattet ist. Diese sollen die optische Leistung des Systems verbessern, welche aufgrund der schmalen Öffnungswinkel etwas reduziert wird.

Da fluoreszierende Lampen die optische Leistung des Systems um 2 dB verringern können, vor allem wenn sich die reflektierenden Spots in der Nähe einer solchen Lampe befinden, wird ein 16 MHz Hochpass Filter der ersten Ordnung eingebaut um dies zu verhindern [8].

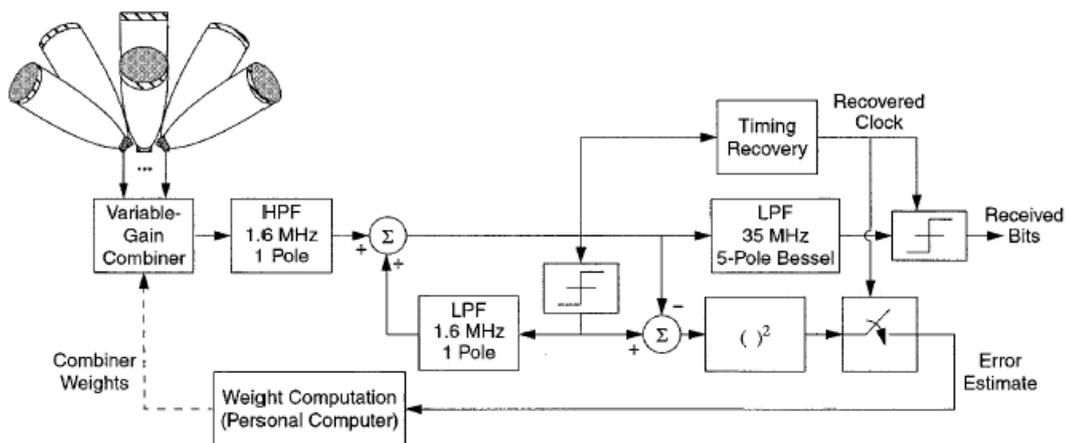


Abbildung 9: Angle Diversity Receiver System, aus [8]

3 Modulation, CDMA und Kanalcodierung

In diesem Kapitel werden die verwendete Modulationsart und die Kanalcodierung genauer betrachtet. Ebenso wird die Funktionsweise des CDMA Vielfachzugriffsverfahrens betrachtet, wobei das Hauptaugenmerk auf den Codewörtern liegt.

3.1 Modulation

Unter Modulation versteht man das Einbringen von Informationen in ein Trägersignal (d.h. die Information wird mit der Trägerfrequenz gemischt). Bei dem verwendeten Trägersignal handelt es sich um infrarotes Licht im Wellenlängenbereich von 880 nm. Die folgende Abbildung 10 zeigt die relative spektrale Emission der verwendeten Infrarot LEDs [14].

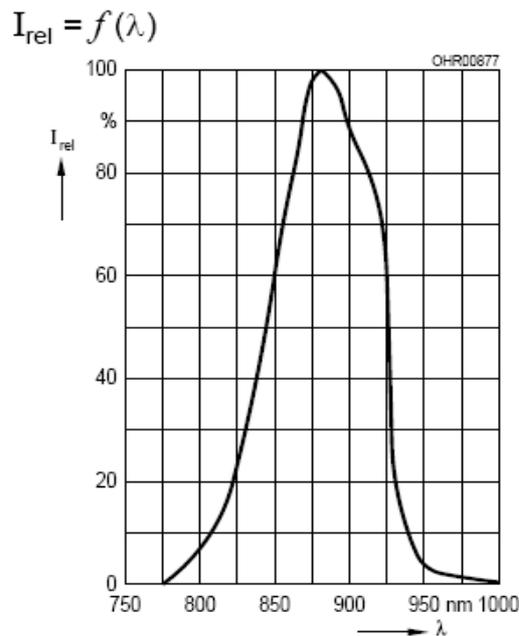


Abbildung 10: Spektrale Emissionswellenlänge, aus [14]

Aus der in Abbildung 10 dargestellten Emissionswellenlänge lässt sich die Frequenz des Trägers errechnen [2].

$$c = \lambda \cdot f \rightarrow f = \frac{c}{\lambda} = \frac{300000 \frac{km}{s}}{880 \text{ nm}} = 340,909 \text{ THz}$$

λWellenlänge in [nm], cLichtgeschwindigkeit in [km/s], fFrequenz in [Hz]

Die Trägerfrequenz befindet sich laut Berechnung bei 340,909 THz. Aufgrund der bei Licht üblichen hohen Frequenzen steht für die Übertragung ausreichend Bandbreite für die Modulation zur Verfügung. Diese beschränkt sich allerdings in dieser Arbeit, wie in Kapitel 1 erwähnt, auf 40 kHz.

Bei optischen Freiraumübertragungen die für Innenräume gedacht sind, ist die Auswahl der Modulationsverfahren beschränkt. Des Weiteren grenzt die Tatsache, dass das Netzwerk auch mittels diffusen Reflexionen funktionieren soll, die Auswahl weiter ein.

Die einfachste und günstigste Art Information in das Trägersignal einzubringen ist die OOK Modulation. Dabei wird jedes zu übertragende Bit direkt durch die Veränderung der Trägerintensität dargestellt. Bei einem „1“ Bit wird der Träger somit eingeschaltet und bei einem „0“ Bit bleibt er deaktiviert. Beispielhaft ist dieser Vorgang in Abbildung 11 dargestellt.

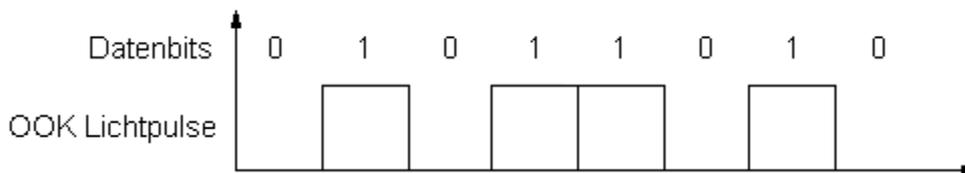


Abbildung 11: Prinzip der OOK Modulation

Die Amplitude der OOK Lichtpulse hängt vom Strom der durch die LED fließt ab. Die Stromstärke bleibt während eines Sendevorgangs konstant auf einem Level. In Abbildung 12 ist ein Beispiel abgebildet welches einen OOK-CDMA Code zeigt der dieselben Datenbits überträgt wie in Abbildung 11. Da die Pulsbreite der OOK Lichtpulse unverändert bleibt, kann man sofort erkennen dass der Datendurchsatz mit dem CDMA Verfahren geringer ist als bei direktem OOK. Mehr Details folgen im kommenden Kapitel 3.2.



Abbildung 12: Beispiel einer OOK-CDMA Datenübertragung

3.2 CDMA und Codewortgenerierung

Bei dem CDMA Vielfachzugriffsverfahren handelt es sich um eine Technik bei der jedem Teilnehmer im Netzwerk zu jedem Zeitpunkt die gesamte Bandbreite des Übertragungskanal zur Verfügung steht, siehe Abbildung 13. CDMA wird bereits seit einigen Jahren im Bereich der Mobiltelefonie aufgrund der Widerstandsfähigkeit bei Multiuser Interferenzen eingesetzt [10]. Ein weiterer Vorteil von CDMA ist, dass keine Netzwerk-

synchronisation benötigt wird. Ein großer Nachteil des CDMA Verfahrens ist ein relativ niedriger Datendurchsatz.

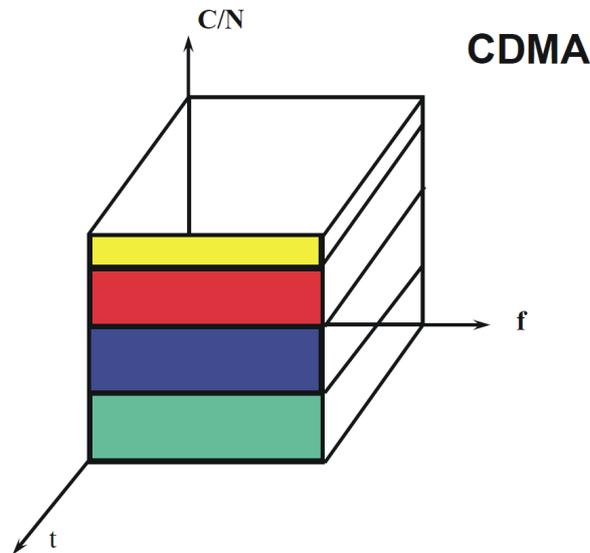


Abbildung 13: Schematische Darstellung des CDMA Verfahrens, aus [3]

Der Vielfachzugriff auf das Netzwerk erfolgt bei CDMA durch die Nutzung eines eigens generierten Codewortsets, welches aus OOCs (*Optical Orthogonal Code*) besteht. Durch diese Codewörter können die Teilnehmer über den Kanal kommunizieren und erkannt werden. Es wird somit bei CDMA keine MAC (*Medium Access Control*) Einheit benötigt. In optischen OOK-CDMA Netzwerken werden die Daten mit Hilfe der Codewörter übertragen. Jede „1“ in einem zu sendenden Datenpaket wird von dem jeweiligen Teilnehmer durch das Senden seines Codewortes übertragen. Soll eine „0“ aus dem Datenpaket übertragen werden, wird kein Codewort gesendet.

Wird CDMA für eine optische Freiraumübertragung eingesetzt ändert sich das Medium im Vergleich zur Mobiltelefonie nicht. Der Unterschied ist allerdings, dass bei der optischen Freiraumübertragung die Signale mit optischer Leistung, also mit Lichtpulsen, übertragen werden. Aus dieser Intensitätsübertragung resultiert, dass es kein negatives Signal wie bei Funkübertragungen gibt, und der Übertragungskanal somit unipolar ist [10].

Ein Codewort aus einem OOC Set besteht aus einer Sequenz von (0,1) welche N Stellen, auch Chips genannt, besitzt. Das Hamming-Gewicht w eines Codewortes bezieht sich auf die Anzahl der im Code enthaltenen „1“. Wie in [10] wird im Zuge dieser Arbeit ausschließlich mit Codes mit konstantem Hamming-Gewicht gearbeitet. Für irgendein Codewort s_1 aus dem Codeset lautet die Autokorrelationsfunktion:

$$\sum_{t=0}^{N-1} s_1(t+\tau) \cdot s_1(t) \leq k_a$$

Die Kreuzkorrelationsfunktion für ein Paar Codewörter s_1 und s_2 aus dem Codeset sieht wie folgt aus:

$$\sum_{t=0}^{N-1} s_1(t+\tau) \cdot s_2(t) \leq k_c, \text{ mit } N \dots \text{ Länge eines Codewortes (N Chips), } k_a \dots \text{ Autokorrela-}$$

tionskollisionsparameter, $k_c \dots$ Kreuzkorrelationskollisionsparameter, $s_1, s_2 \dots$ Codewörter im Codeset s.

Erfüllen die Codewörter die beiden oben angeführten Gleichungen kann von einem pseudo-orthogonalen Codeset gesprochen werden. Diese pseudo-Orthogonalität bedeutet, dass ein Codewort limitierte Interferenzen gegenüber anderen Codewörtern des Codesets aufweist.

In den meisten Code Designs wird $k_a = k_c$ gesetzt. Dieser Wert wird der maximale Kollisionsparameter k genannt. Die meisten optischen CDMA Netzwerk Designs legen k mit 1 oder 2 fest, um zu gewährleisten, dass die Interferenzen zwischen den Codewörtern gering sind. Die Parameter, um ein solches Codeset zu beschreiben, lauten also $\Phi(N, w, k)$. Die Größe eines solchen Codesets wird durch die Anzahl der Codewörter in diesem Set bestimmt. Wird ein Codeset unter den pseudo-orthogonalen Bedingungen entworfen, kann mit der folgenden Formel für die Johnson Grenze die obere Grenze für die Anzahl der Codewörter S bestimmt werden.

$$S(N, W, k) \leq \left(\frac{1}{w} \cdot \left(\frac{n-1}{w-1} \cdot \left(\frac{n-2}{w-2} \cdot \left(\dots \left(\frac{n-k}{w-k} \right) \dots \right) \right) \right) \right)$$

Es können nun mehrere (max. S) Teilnehmer, mit verschiedenen Codewörtern asynchron ihre Daten senden. Ein Empfänger der darauf abgestimmt wird ein bestimmtes Codewort zu erkennen, achtet bei den vorherbestimmten Chippositionen darauf ob eine „1“ empfangen wird. Falls die Anzahl der richtig detektierten „1“ Chips größer gleich einem bestimmten Grenzwert ist, wird der Empfänger dies als empfangenes „1“ Datenbit werten. Die Höhe des Grenzwertes wird üblicherweise mit dem Gewicht des Codes gleichgesetzt. Es kann ein Fehler auftreten, wenn durch Codewörter, welche am Kanal übertragen werden, eine Kombination der „1“ Chips entsteht, die sich mit einem vorhandenem Codewort deckt. Tritt so ein Fehler auf, wurde ein 0 Datenbit oder eine nicht vorhandene Übertragung fälschlich als „1“ Datenbit detektiert (*a false positive*). Wenn dieser Fall eintritt, kann dies zu einem Paketverlust führen. Wird ein optisches CDMA Netzwerk unter hoher Auslastung betrieben, sind solche Paketverluste verantwortlich für eine weitere Datendurchsatzreduzierung. Um solche *false positiv* zu verhindern wurde im Zuge dieser Arbeit ein *Preamble* eingeführt. Dieser *Preamble* kennzeichnet den Übertragungsstart eines Teilnehmers und dient so auch als kleine Synchronisierungshilfe, welche nötig ist, da bei zu sendenden „0“ Datenbits kein Codewort übertragen wird. Der *Preamble* besteht aus 3 aufeinander folgenden „1“ die vor den Datenbits übertragen werden. Jeder Teilnehmer sendet somit 3-mal hintereinander sein Codewort und beginnt im Anschluss mit der Übertragung der Datenbits. Wird nun

durch die Kombination von zwei oder mehreren Codewörtern ein Codewort gebildet das im Netzwerk verwendet wird, jedoch gerade nicht übertragen wird, so hat das detektierte „1“ Datenbit keine Auswirkungen, da es sehr unwahrscheinlich ist das diese Konstellation genau 3 mal in Folge vorkommt. Sollte ein *false positive* eines Teilnehmers auftreten nachdem dieser bereits sein *Preamble* gesendet hat, kommt es zur Falschinterpretierung eines „0“ Datenbits und das gesendete Datenbyte wurde nicht richtig empfangen. Um dieses Problem zu beheben führt man eine Kanalcodierung ein, siehe Kapitel 3.3.

Da dieses Sensornetzwerk nur mit 3 Teilnehmern aufgebaut wird, benötigt man ein Codeset das mindestens 3 Codewörter enthält. In [11] werden einige Methoden vorgestellt um ein OOC Set zu konstruieren. Diese Codesets sind allerdings meist für größere Netzwerke, welche auch eine höhere Auslastung haben, ausgelegt. Die Sensoren in diesem Netzwerk können jede Sekunde nur einen 8 - 14 Bit langen Messwert bereitstellen. Aus diesem Grund ist die mögliche Auslastung in diesem Netzwerk nicht sehr hoch.

Anders als in [11] beschrieben, werden in dieser Arbeit Codes mit einem Gewicht von $w = 2$ verwendet. Der Kollisionsparameter k wird standartgemäß mit 1 festgelegt. Damit lässt sich die Codewortlänge auf 8 Chips reduzieren und man erreicht trotzdem eine Codesetgröße von 3 Teilnehmern, siehe Kapitel 3.2.2. Somit wären die Parameter für dieses Codeset festgelegt $\Phi(8,2,1)$. Wird die Codelänge von $N = 8$ vergrößert, sinkt die Übertragungsrate des Netzwerkes da mit jeder weiteren Chipstelle die Korrelationsrechnung länger wird und somit mehr Rechenzeit benötigt. Auf die genaue Dauer der Korrelationsberechnung wird in Kapitel 5.3 näher eingegangen.

Die Codewörter aus dem Codeset für dieses Netzwerk sehen wie folgt aus:

Codewort 1

$$[0,0,1,0,0,1,0,0] \rightarrow \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} -1 \\ -1 \\ 1 \\ -1 \\ -1 \\ 1 \\ -1 \\ -1 \end{pmatrix}$$

Codewort 2

$$[0,0,0,0,1,0,1,0] \rightarrow \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} -1 \\ -1 \\ -1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \end{pmatrix}$$

Codewort 3

$$[0,1,0,0,0,0,0,1] \rightarrow \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} -1 \\ 1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ 1 \end{pmatrix}$$

Um überprüfen zu können ob diese Codewörter auch orthogonal untereinander sind, wird das Codewort in einen Vektor umgewandelt. Der Vorteil eines Vektors liegt darin, dass man mit dem Skalarprodukt zweier Vektoren berechnen kann, ob diese beiden Vektoren orthogonal zueinander sind.

Damit die Orthogonalität errechnet werden kann, muss man die Nullstellen im Vektor durch -1 ersetzen. Das folgende Beispiel veranschaulicht das:

$$\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix} \cdot \begin{pmatrix} -1 \\ 1 \\ 1 \\ -1 \end{pmatrix} = 1 \cdot (-1) + (-1) \cdot 1 + 1 \cdot 1 + (-1) \cdot (-1) = 0$$

Ergibt das Skalarprodukt beider Vektoren 0 sind sie orthogonal zueinander. Nun wird die Orthogonalität unter den verwendeten Codewörtern überprüft:

Codewort 1 und 2:

$$\begin{pmatrix} -1 \\ -1 \\ 1 \\ -1 \\ -1 \\ 1 \\ -1 \\ -1 \end{pmatrix} \cdot \begin{pmatrix} -1 \\ -1 \\ -1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \end{pmatrix} = (-1) \cdot (-1) + (-1) \cdot (-1) + 1 \cdot (-1) + (-1) \cdot (-1) + (-1) \cdot 1 + 1 \cdot (-1) + (-1) \cdot 1 + (-1) \cdot (-1) = 0$$

Codewort 1 und 3:

$$\begin{pmatrix} -1 \\ -1 \\ 1 \\ -1 \\ -1 \\ 1 \\ -1 \\ -1 \end{pmatrix} \cdot \begin{pmatrix} -1 \\ 1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ 1 \end{pmatrix} = (-1) \cdot (-1) + (-1) \cdot 1 + 1 \cdot (-1) + (-1) \cdot (-1) + (-1) \cdot (-1) + 1 \cdot (-1) + (-1) \cdot (-1) + (-1) \cdot 1 = 0$$

Codewort 2 und 3:

$$\begin{pmatrix} -1 \\ -1 \\ -1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \end{pmatrix} \cdot \begin{pmatrix} -1 \\ 1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ 1 \end{pmatrix} = (-1) \cdot (-1) + (-1) \cdot 1 + (-1) \cdot (-1) + (-1) \cdot (-1) + 1 \cdot (-1) + (-1) \cdot (-1) + 1 \cdot (-1) + (-1) \cdot 1 = 0$$

Alle 3 Codewörter sind orthogonal aufeinander, wie die Berechnungen gezeigt haben. Im nächsten Schritt wird die Auto- und Kreuzkorrelationsfunktion mit den Codewörtern durchgeführt um zu sehen ob diese auch erfüllt werden. Dieser Schritt wurde in Matlab umgesetzt und ausgewertet. Die Ergebnisse der Autokorrelationsfunktionen sind in der folgenden Abbildung 14 dargestellt.

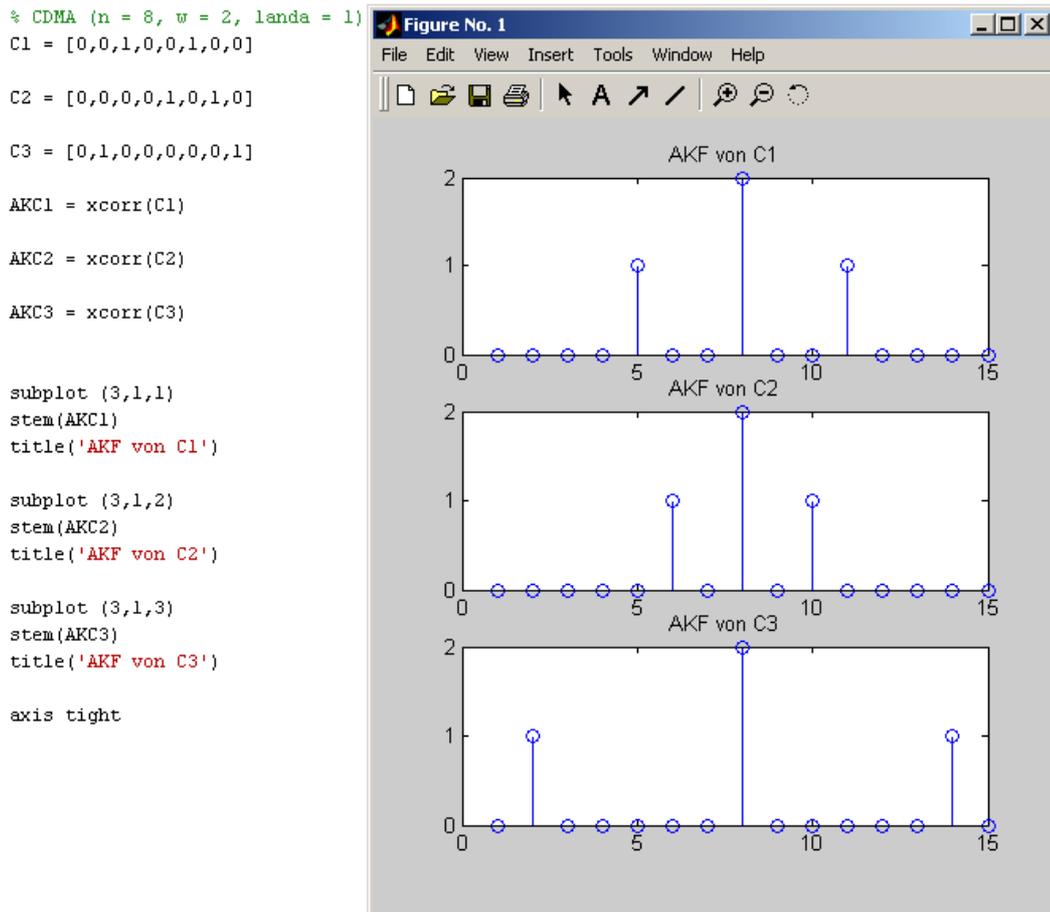


Abbildung 14: Autokorrelationsfunktion der Codewörter

In Abbildung 14 ist deutlich zu erkennen, dass sich die Codewörter genau an einer Stelle decken. Diese wird erreicht, wenn es keine Verschiebung zwischen dem Codewort und seiner Kopie gibt. Die Ergebnisse der Kreuzkorrelationsfunktionen zeigt Abbildung 15. Es ist eindeutig zu erkennen, dass die Codewörter sich untereinander nicht gleichen und jeweils nur an einer Chipstelle eine Übereinstimmung herrscht.

Das erstellte OOC Set erfüllt die Auto- und Kreuzkorrelationsfunktion und kann somit für die Übertragung benutzt werden. Es bleibt nun nur mehr zu bestimmen, wie viele Codewörter in diesem $\Phi(8,2,1)$ Codeset verwendet werden können. Dafür wird die Formel aus Kapitel 3.2 für die Berechnung der Johnson Grenze benutzt.

$$S(8,2,1) \leq \left(\frac{1}{2} \cdot \left(\frac{8-1}{2-1} \right) \right) = \frac{1}{2} \cdot \frac{7}{1} = 3,5$$

Aus der Berechnung geht hervor, dass mit diesem Codeset 3,5 Codewörter verwendet werden können. Da für diese Arbeit nur 3 Codewörter benötigt werden, reicht das erstellte Codeset vollkommen aus. Sollte die Auslastung des Netzwerkes sehr hoch werden, könnte es sein dass dieses Codeset nicht mehr ausreichend für eine fehlerfreie Übertragung geeignet ist, da die Codewörter am Kanal bei hoher Auslastung beginnen sich gegenseitig zu interferieren.

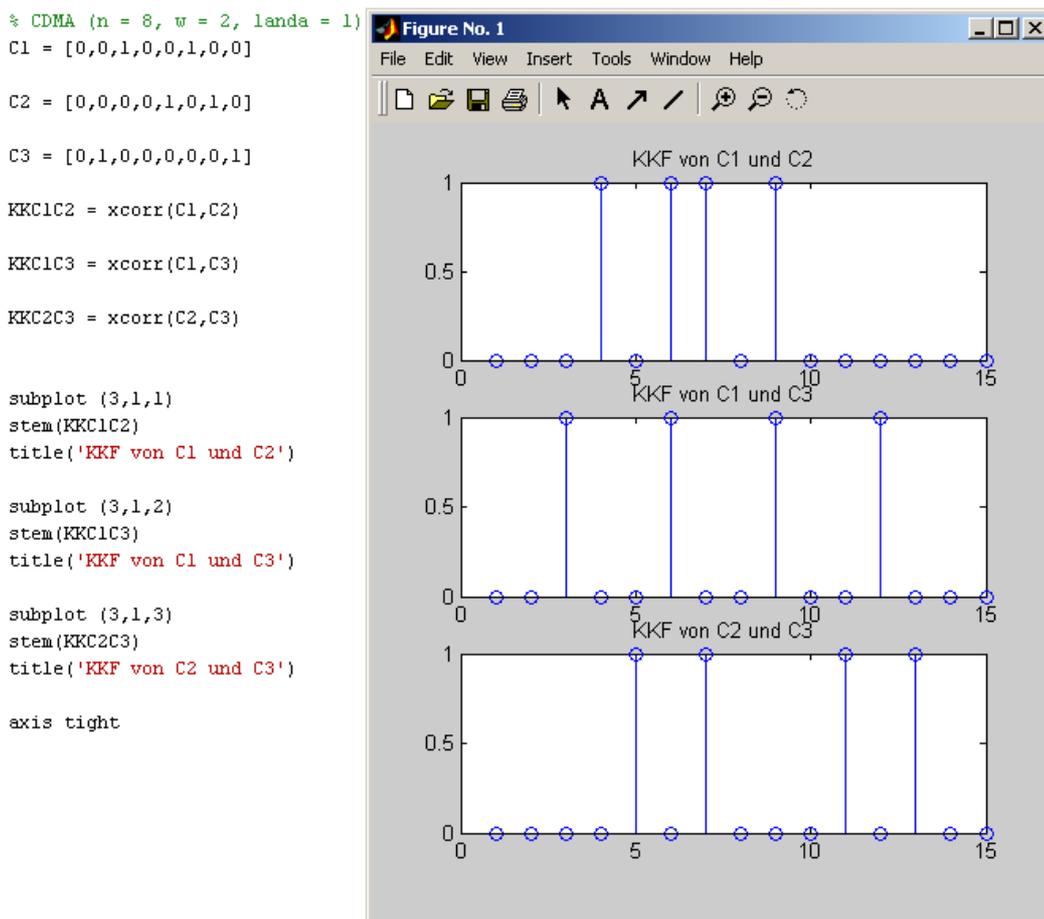


Abbildung 15: Kreuzkorrelationsfunktion der Codewörter

Als nächstes wird die spektrale Effizienz berechnet. Sie ist ein Maß dafür, wie gut das Kodierschema im Vergleich zu einem perfektem TDMA (*Time Division Multiple Access*) System ($\eta=1$) arbeitet. In diesem Netzwerk liegt die spektrale Effizienz deutlich über dem Wert anderer CDMA Funknetze. Der Grund dafür ist die Kombination aus geringer Teilnehmeranzahl und der geringen Codelänge.

Die spektrale Effizienz dieses Codesets sieht wie folgt aus: $\eta = \frac{A}{N} = \frac{3}{8} = 0,375 = 37,5\%$

Der Parameter A bezeichnet die Anzahl aktiver Teilnehmer, während N die Anzahl der Chips eines Codewortes widerspiegelt.

3.3 Kanalcodierung

Ein Übertragungskanal, der als Medium die Luft zum Übertragen der Daten nutzt, kann leicht von außen beeinflusst werden. Deswegen ist die Wahl einer Kanalcodierung in diesem Fall eine sinnvolle Option. Es wird im Zuge dieser Arbeit auf eine einfache Paritätsprüfung zurückgegriffen. Diese hat den Vorteil, dass sie sehr einfach via Software

zu implementieren ist. Das Paritätsbit ist dafür geeignet einzelne Fehler in einem Datenpaket zu erkennen. Der Fehler kann allerdings nicht korrigiert werden, da es sich hier ausschließlich um eine fehlererkennende Methode handelt [4]. Wird die Paritätsprüfung auf die gerade Anzahl der „1“ im Datenwort angewandt, so kann diese Methode nur eine ungerade Anzahl an Fehlern im Datenwort erkennen. Umgekehrt gilt dasselbe für eine ungerade Anzahl von „1“ und eine gerade Anzahl an Fehlern.

Die Paritätsprüfung wurde gewählt, weil es damit möglich ist eine, *false positive* Detektion bei dem CDMA Verfahren zu erkennen und den empfangenen Messwert nicht weiter zu verarbeiten. Die folgende Tabelle 1 zeigt ein Beispiel dieser Kodiermethode.

Wird mit dem Sensor die Luftfeuchtigkeit gemessen, sieht der gemessene Wert für 25% so aus: 01000000. Soll bei der Paritätsprüfung eine gerade Anzahl an „1“ herauskommen, so wird die Stelle des Paritätsbits in diesem Fall auf „1“ gesetzt.

	Informationsbits								Prüfbit	Gewicht
Luftfeuchtigkeit	X1	X2	X3	X4	X5	X6	X7	X8	y	
0%	0	0	0	0	0	0	0	0	0	0
25%	0	1	0	0	0	0	0	0	1	2
50%	1	0	0	0	0	0	0	0	1	2
75%	1	1	0	0	0	0	0	0	0	2
100%	1	1	1	1	1	1	1	1	0	8

Tabelle 1: Ein Beispiel zum Einsatz der Paritätsprüfung

Werden die informationstragenden Binärstellen unverändert übertragen, und die Fehlerkontrolle durch zusätzliche Prüfbits realisiert, spricht man von systematischen Codes. In jedem Codewort eines systematischen Codes lassen sich m informationstragende Bits und k redundante Bits finden. Addiert man m und k erhält man die Gesamtanzahl der Bits n . Da nach jedem Messwert eine Übertragung zur Hauptstation stattfindet, muss das *Preamble* zu den k redundanten Bits hinzuaddiert werden.

Die Gesamtanzahl der Bits eines zu sendenden 8 Bit langen Luftfeuchtigkeitsmesswertes beträgt somit:

$$n = m + k = 8 + 4 = 12$$

Die Coderate R gibt das Verhältnis von Informationsbits m zur Gesamtbitanzahl n wieder.

$$R = \frac{m}{n} = \frac{n - k}{n} = 1 - \frac{k}{n} \leq 1$$

$$R = \frac{12-4}{12} = 0,6$$

Die relative Redundanz ergibt sich somit zu:

$$r = \frac{n-H}{n} = \frac{12-ld256}{12} = \frac{4}{12} = 0,3 = 33,3\%$$

In der folgenden Tabelle 2 sind die Coderaten und die relative Redundanz für 12 und 14 Bit lange Messwerte angeführt.

Auflösung	12 Bit	14 Bit
Coderate R	$R = \frac{16-4}{16} = 0,75$	$R = \frac{18-4}{18} = 0,77$
relative Redundanz r	$r = \frac{16-ld4096}{16} = 0,25 = 25\%$	$r = \frac{18-ld16384}{18} = 0,22 = 22,2\%$

Tabelle 2: Coderate und relative Redundanz

3.4 Datenübertragungsrate und Datendurchsatz

Die Datenübertragungsrate setzt sich bei einem Netzwerk, welches mit CDMA arbeitet, aus der Division der Chiprate durch N zusammen [4]. N stellt die Anzahl der Chips pro Codewort dar. Hier bestätigt sich nun der Nachteil des CDMA Verfahrens in einer geringeren Datenübertragungsrate, als beispielsweise bei dem TDMA Verfahren. Die verwendete Chiprate beträgt 40 kchip/s und die Länge der Codewörter beträgt N = 8. Somit ergibt sich folgende Datenübertragungsrate:

$$\text{Datenübertragungsrate} = \frac{\text{Chiprate}}{\text{Codewortlänge}} = \frac{40 \text{ kchip/s}}{8} = 5 \text{ kbit/s}$$

Man kann in diesem Netzwerk also 5 kbit/s pro Teilnehmer übertragen. Da jedoch, wie schon in Kapitel 3.2 erwähnt, die verwendeten Sensoren in einer Sekunde nur einen Messwert aufnehmen und dieser aus 8 - 14 Bits besteht, würde auch eine noch geringere Datenübertragungsrate ausreichen.

Neben der Datenübertragungsrate gilt es auch noch den Datendurchsatz zu bestimmen [4]. Diese Größe bezieht sich auf die tatsächlichen Nutzdaten, welche über den Kanal übertragen werden. Das bedeutet, dass die redundanten Bits, welche die Übertragungssicherheit erhöhen sollen, in die Berechnung mit einbezogen werden, um den Datendurchsatz bestimmen zu können.

Der Datendurchsatz ändert sich mit der Wahl der Auflösung des Sensors. So erreichen die 14 Bit langen Messwerte einen höheren Datendurchsatz als die Messwerte mit 8 Bit Länge, dargestellt in Tabelle 3. Da jeder Teilnehmer vor dem Senden des Messwertes den *Preamble* überträgt und, wenn nötig, ein Paritätsbit anhängt, sinkt der Datendurchsatz im Vergleich zur Datenübertragungsrate. Der absolute und der relative Datendurchsatz lassen sich wie folgt berechnen:

$$\text{Datendurchsatz} = \frac{\text{Datenübertragungsrate} \cdot m}{n}$$

$$\text{relativer Datendurchsatz} = \frac{m}{n} \cdot 100\%$$

Auflösung	8 Bit	12 Bit	14 Bit
Absoluter Datendurchsatz	3,33 kbit/s	3,75 kbit/s	3,88 kbit/s
relativer Datendurchsatz	66,6%	75%	77,7%

Tabelle 3: Absoluter und relativer Datendurchsatz

Es gibt verschiedene Methoden um den Datendurchsatz von CDMA Netzwerken unter hoher Auslastung zu erhöhen. Einige dieser Methoden werden in [10] und [12] vorgestellt. So wird zum Beispiel die Methode der Interferenzen Abtastung und der Interferenzen Erkennung, kurz IS/ID (*Interference Sense / Interference Detection*), erklärt. Hierbei handelt es sich um ein Verfahren bei dem jeder Netzwerkteilnehmer die Daten anderer Teilnehmer, welche über den gemeinsam genutzten Kanal übertragen werden, empfängt und mittels eines Algorithmus auswertet. Ziel dieser Methode ist es festzustellen, zu welchem Zeitpunkt die eigenen Daten über den Kanal übertragen werden können, ohne dabei Interferenzen mit Codewörtern anderer Teilnehmer zu bilden. In diesem Zusammenhang wurden in [12] der *selfish sensing* Algorithmus und der *threshold based sensing* Algorithmus vorgestellt. In [10] wurden Vergleiche zwischen IS/ID und dem Aloha-CDMA Mechanismus gezogen.

4 Dimensionierung der Sende- und Empfängerschaltung

In diesem Kapitel wird der Aufbau der Sende- und Empfängerschaltung entworfen. Dafür werden in Bezug auf die verwendeten Bauteile die Funktionsblöcke erläutert.

4.1 Sendeschaltung

In Kapitel 3 wurde OOK als Modulationsart ausgewählt. Somit ist die Aufgabe der Sendeschaltungen an dem Sendestrom eine Intensitätsmodulation durchzuführen. Diese Modulation ist mit einer einfachen Transistorschaltung leicht zu realisieren. Der Transistor wird vom Mikrocontroller über einen Basiswiderstand R_1 für eine gewisse Zeit T durchgesteuert. Der Widerstand R_2 ist für die Strombegrenzung im Zweig der LEDs zuständig, siehe Abbildung 16.

4.1.1 Bauteilauswahl

Die folgenden Bauteile werden aufgrund geringer Kosten ausgewählt, der Transistor BC639 und die LED SFH487.

Der Transistor BC639 ist ein bipolarer NPN Transistor, der für die Anforderungen dieser Arbeit gut geeignet ist [21].

Bei der LED handelt es sich um die Type SFH487 von Osram [14]. Diese LED wird aufgrund mehrerer Faktoren ausgewählt. Zum einen emittiert sie Licht im Infrarotbereich (880 nm) und hat einen schmalen Öffnungswinkel von 40° , zum anderen hat diese LED geringe Anstiegs- und Abfallzeiten.

4.1.2 Dimensionierung der Sendeschaltungen

In Kapitel 2.2 wurden die Ausrichtung und die Anzahl der verwendeten LEDs festgelegt. Für die Sendeschaltung der Sensorstationen werden 3 LEDs verwendet (Abbildung 5) und für die Hauptstation 9 LEDs (Abbildung 6). Die Schaltbilder der beiden Sendeschaltungen wurden in Eagle entworfen und sind in der folgenden Abbildung 16 zu sehen. Die Sendeschaltungen wurden mit 3 LEDs pro Sendezweig ausgestattet, um Transistoren zu sparen und den Stromverbrauch zu senken. In Abbildung 17 werden die beiden realisierten Sendeschaltungen gezeigt.

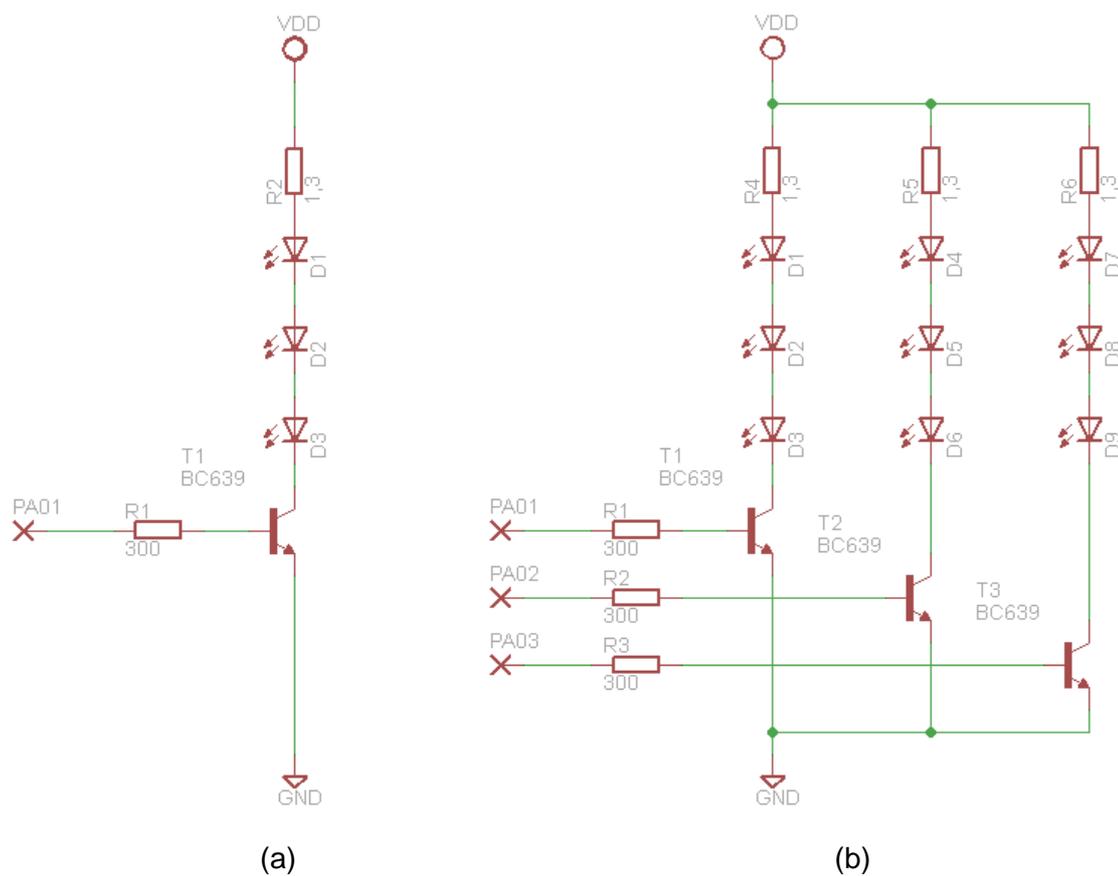


Abbildung 16: Sendeschaltung der a) Sensor- und b) Hauptstation

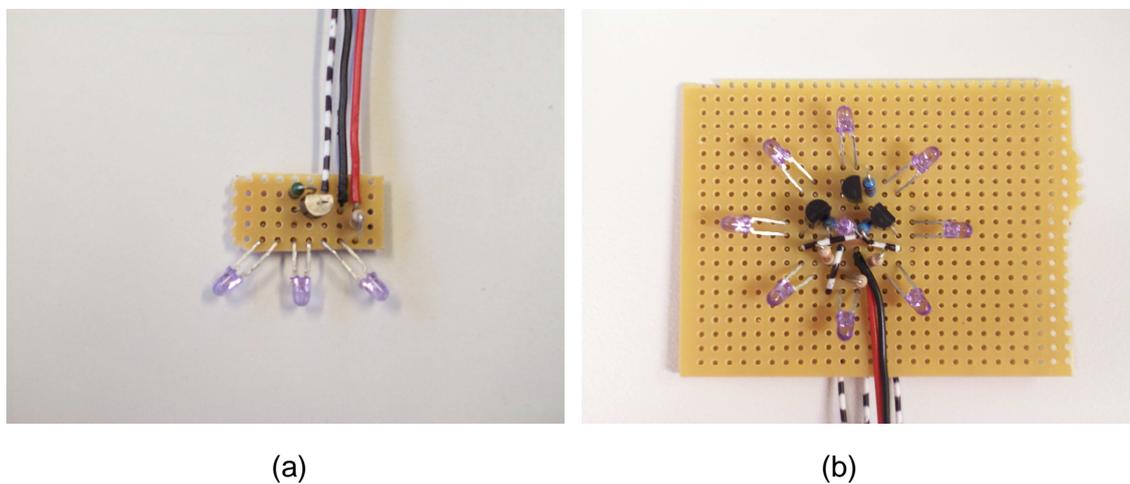


Abbildung 17: Aufbau der Sendeschaltung a) Sensor- und b) Hauptstation

Im Zuge der Aufbauarbeiten dieser Arbeit wurde ein Sendestrom von 100 mA gewählt. Aus dem Datenblatt der LEDs kann entnommen werden, dass bei einem Durchlassstrom von 100 mA eine Spannung von 1,45 V an der LED abfällt [14]. Dieser angegebene Wert konnte in Tests nicht erreicht werden. Der Spannungsabfall an den LEDs wurde bei ca. 100 mA mit 1,61V gemessen. Der Spannungsabfall am Transistor be-

trägt laut Datenblatt 50 mV [21]. Bei einer Versorgungsspannung von 5 V errechnet sich der Spannungsabfall am Vorwiderstand R_2 zu:

$$U_{R_w} = U_V - 3 \cdot U_{LED} - U_{T_{SAT}} = 5V - 3 \cdot 1,61V - 50mV = 0,12V$$

Mit dem errechneten Spannungsabfall und dem gewünschten Strom ergibt sich in weiterer Folge der Vorwiderstand R_2 zu:

$$R_2 = \frac{U_{R_2}}{I_S} = \frac{120mV}{100mA} = 1,2\Omega$$

Es wird ein 1Ω Widerstand verwendet, welcher einen tatsächlichen Wert von $1,3\Omega$ besitzt. Der tatsächlich errechnete Sendestrom beträgt somit:

$$I_S = \frac{U_{R_2}}{R_2} = \frac{120mV}{1,3\Omega} = 92,3mA$$

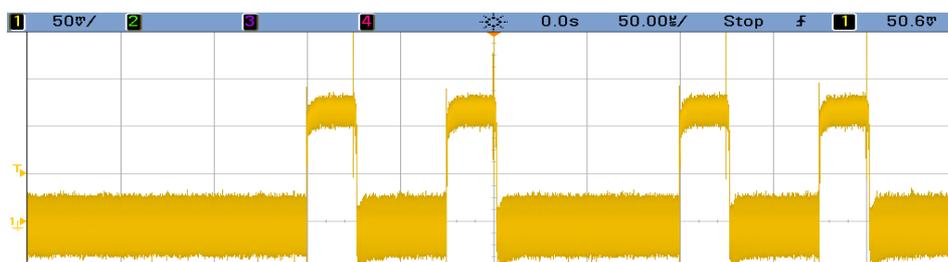


Abbildung 18: Spannungsabfall U_{R_2} am Vorwiderstand R_2

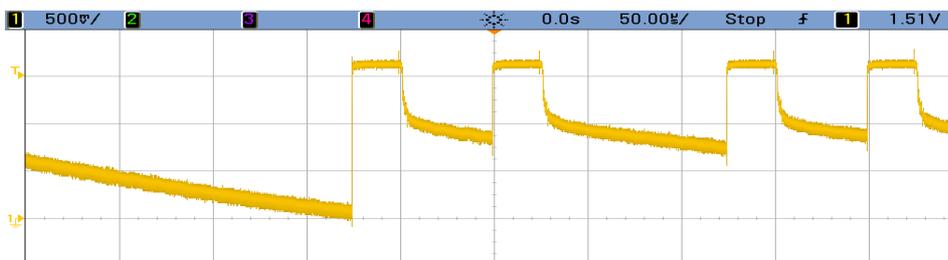


Abbildung 19: Spannungsabfall U_{LED} an der LED SFH487

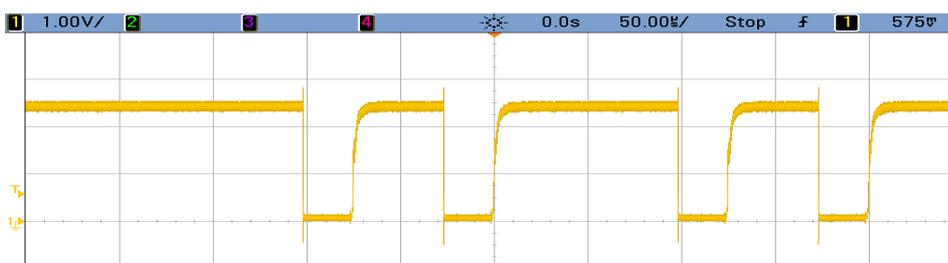


Abbildung 20: Spannungsabfall $U_{T_{SAT}}$ am Transistor BC639

In Abbildung 18 ist der gemessene Spannungsabfall am Vorwiderstand R_2 dargestellt. Die gepulste Spannung U_{R_2} erreicht einen Wert von 120 mV. Der Spannungsabfall an der LED U_{LED} ist in Abbildung 19 dargestellt und beläuft sich auf 1,61 V. Des Weiteren ist in Abbildung 20 der Spannungsverlauf $U_{T_{SAT}}$ am Transistor abgebildet, welcher gut erkennen lässt wann der Transistor durchschaltet.

Der Basiswiderstand R_1 wird mit 300Ω gewählt. Mit diesem Widerstandswert ergibt sich ein Spannungsabfall am Widerstand von 4 V. Die Höhe dieses Spannungsabfalls ist in Abbildung 22 zu sehen. An der Basis – Emitter Strecke des Transistors fallen 850 mV ab, siehe Abbildung 23. Der Basisstrom ergibt sich somit zu:

$$I_B = \frac{U_{R_1}}{R_1} = \frac{4 V}{300 \Omega} = 13,3 mA$$

Die Abbildung 21 zeigt die Spannung am Ausgang des verwendeten I/O Ports, welche 4,85 V beträgt, und sich wie zuvor erwähnt auf den Basiswiderstand mit 4 V und die Basis – Emitter Strecke mit 850 mV aufteilt.



Abbildung 21: Spannung am I/O Port



Abbildung 22: Spannungsabfall U_{R_1} am Basiswiderstand



Abbildung 23: Transistor Basis - Emitter Spannungsabfall



Abbildung 24: Pulsfolge für einen gesendeten Messwert inklusive *Preamble*

In Abbildung 24 wird eine größere *Time / Division* am Oszilloskop gewählt um einen Messwert inklusive *Preamble* und Paritätsbit darzustellen.

4.2 Empfängerschaltung

Das Designen einer Photodioden-Empfängerschaltung ist nicht ganz einfach. Es gibt einige Faktoren, welche die Leistung der Schaltung beeinflussen und somit beachtet werden müssen. So sind zum Beispiel, die Wahl der Photodiode sowie die Wahl ihrer Beschaltung ausschlaggebend. Des Weiteren muss das Rauschen, von anderen Lichtquellen unterdrückt werden. Bei der Wahl des OPVs (Operationsverstärker) gibt es auch einige Punkte zu beachten, so sollte er z.B. einen sehr geringen Biasstrom (im pA Bereich) und eine *Offset*-Spannung im μV Bereich aufweisen.

Der von der Photodiode produzierte Strom sollte von einem Transimpedanzverstärker verstärkt werden, um die daraus resultierende Spannung weiter verarbeiten zu können [5]. In [13] sind 2 Arten von Schaltungen zu finden, diese zeigen wie die Photodiode an den Transimpedanzverstärker anzuschließen ist. Diese beiden Methoden sind die Photovoltaik und die Photokonduktive Methode, siehe Abbildung 25. Die Photovoltaik Methode ist für diese Arbeit eher geeignet da eine höhere Genauigkeit, jedoch auf Kosten der Geschwindigkeit, erreicht wird.

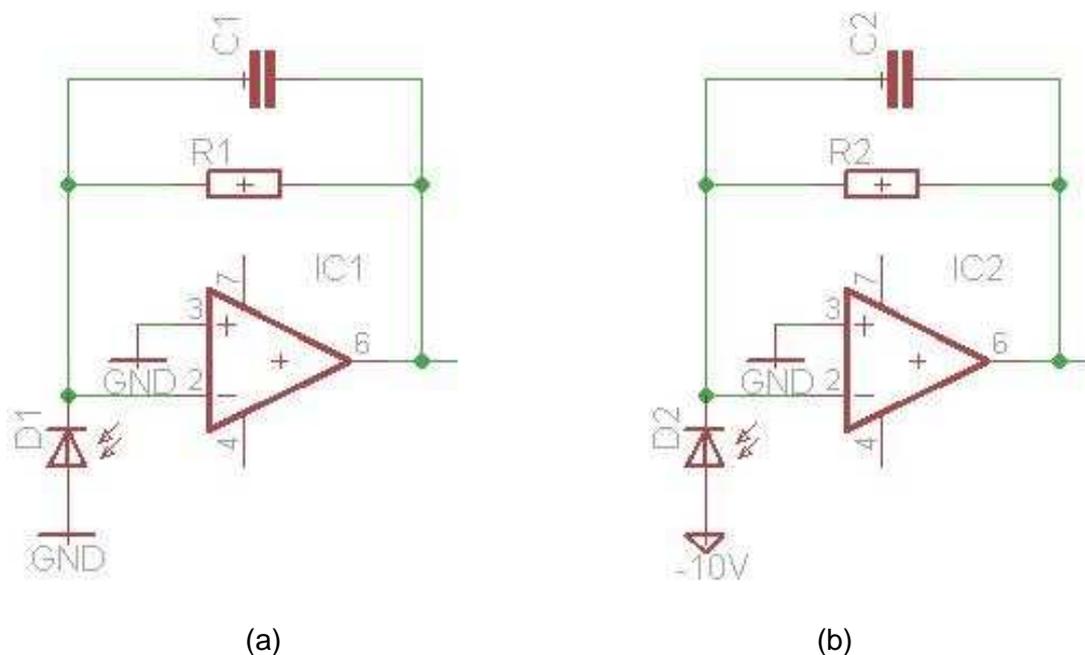


Abbildung 25: a) Photovoltaik Methode; b) Photokonduktive Methode

Bei der Photovoltaik Methode wird die Kathode der Diode auf Masse gelegt. Dadurch liegt über der Diode keine Spannung an, welche an dem internen Diodenwiderstand abfallen kann. Dies hat zur Folge, dass in diesem Fall keinen Dunkelstrom fließt. Durch das Wegfallen des Dunkelstromes wird ein kompletter Rauschterm eliminiert, wodurch es möglich wird kleinere Messwerte zu erfassen. Die Kapazität im Rückkoppelzweig des Operationsverstärkers wird aus Kompensationsgründen eingeführt. Da jede PIN Diode aus Sicht des OPVs als Kapazität verstanden wird ergeben sich ohne die Kapazität im Rückkoppelzweig Schwingungen am Ausgang des OPVs. Auf die weiteren Signalverarbeitungsschritte der gesamten Schaltung wird in Kapitel 4.2.2 näher eingegangen.

4.2.1 Ausgewählte Bauteile

Das Kriterium *low budget* gilt auch für den Aufbau der Empfängerschaltung. So fiel die Wahl der Photodiode auf die SFH205 von Osram [15]. Der Transimpedanzverstärker nach der Photodiode ist von der Firma Texas Instruments und trägt die Bezeichnung OPA380 [18].

Für die Signalverstärkung nach dem Transimpedanzverstärker wird der OPV LTC2051 von Linear Technology verwendet [17]. Der Filterbaustein namens LTC1562 ist ebenfalls von Linear Technology [16].

Der anschließende Komparator, der als invertierender Schmitt-Trigger beschalten ist, wird von Texas Instruments hergestellt. Seine Bezeichnung lautet LM311 [19]. Am Ausgang wurde ein Monoflop der Firma NXP mit der Bezeichnung 74HC123 verwendet [20].

4.2.2 Dimensionierung der Empfängerschaltung

In Abbildung 26 ist ein Blockschaltbild der Empfängerschaltung dargestellt. Diese vereinfachte Darstellung setzt sich aus der PIN Diode und dem Transimpedanzverstärker, einem nicht invertierenden Verstärker [5], einem Filterbaustein, einem invertierenden Schmitt-Trigger [5] und einem Monoflop am Ausgang zusammen [5]. Die vollständige Empfängerschaltung ist in Abbildung 27 dargestellt und wurde mit dem Programm Eagle, so wie alle anderen Schaltbilder in diesem Kapitel, entworfen.

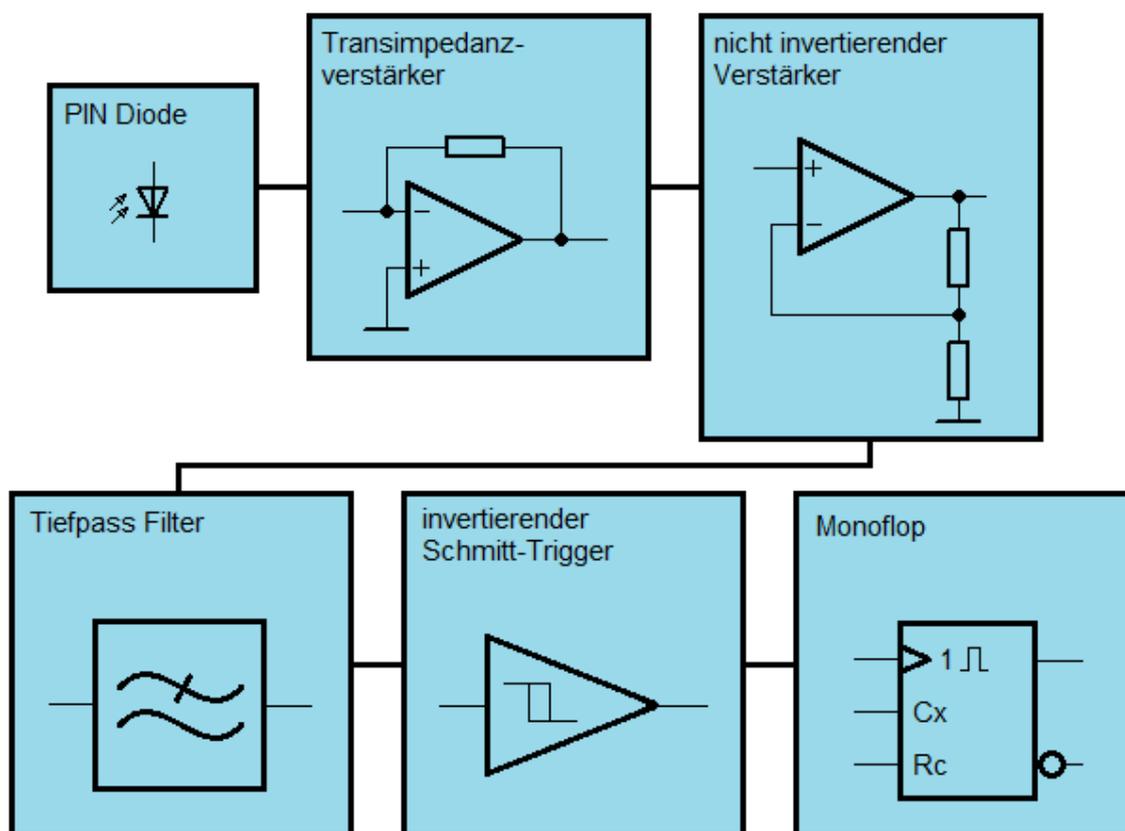


Abbildung 26: Blockschaltbild der Empfängerschaltung

Diese Funktionsblöcke werden in den kommenden Kapiteln genau erläutert. Des Weiteren wird auf die genaue Dimensionierung der einzelnen Bauteile, sowie deren Beschaltung eingegangen. In weiterer Folge wird erklärt wie die Signaldetektion in dieser Schaltung von statten geht.

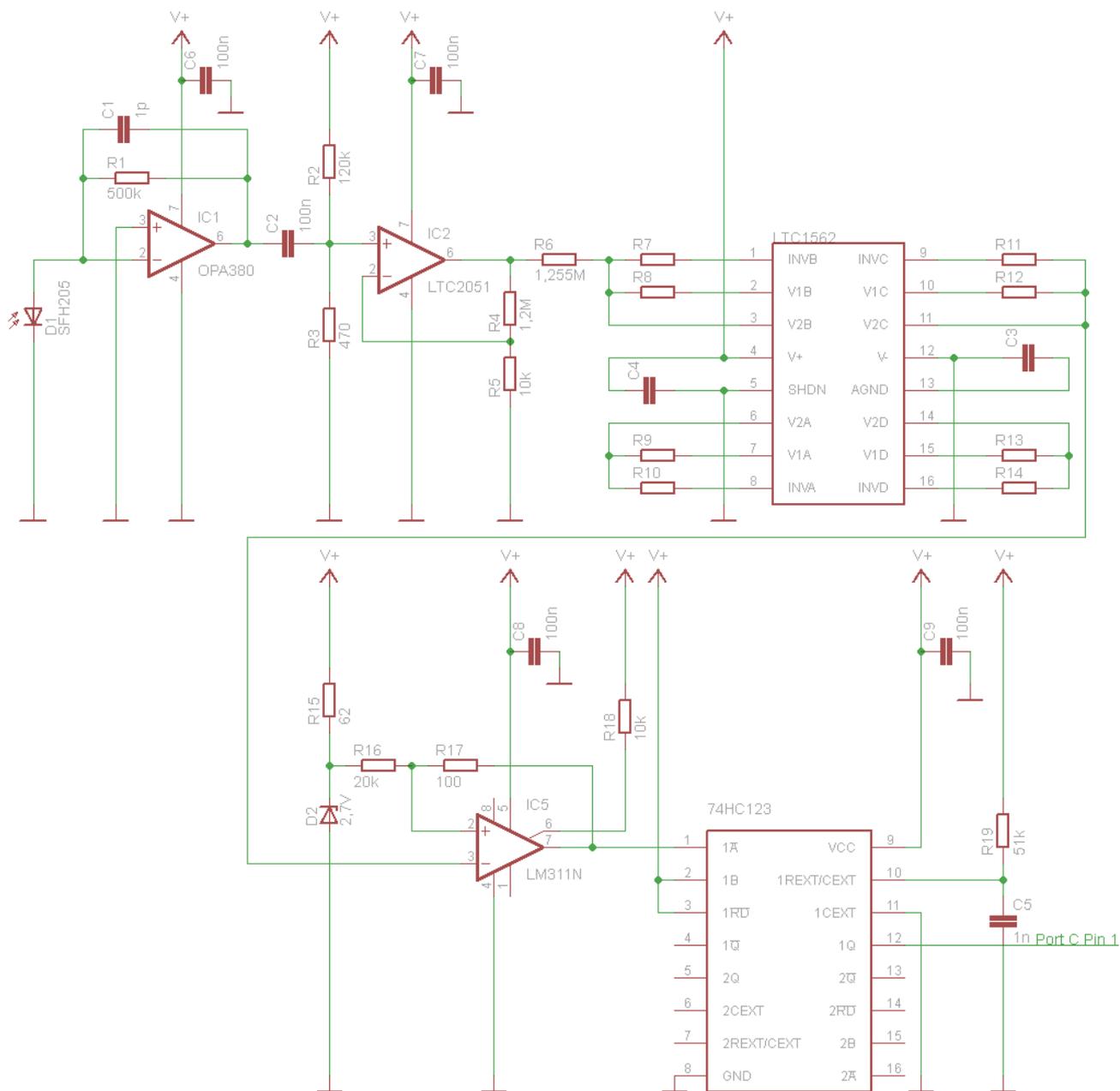


Abbildung 27: Schaltbild der Empfängerschaltung

Die realisierte Empfängerschaltung und ihr Anschluss an das AVR-Net-IO-Board sind in Abbildung 28 dargestellt.

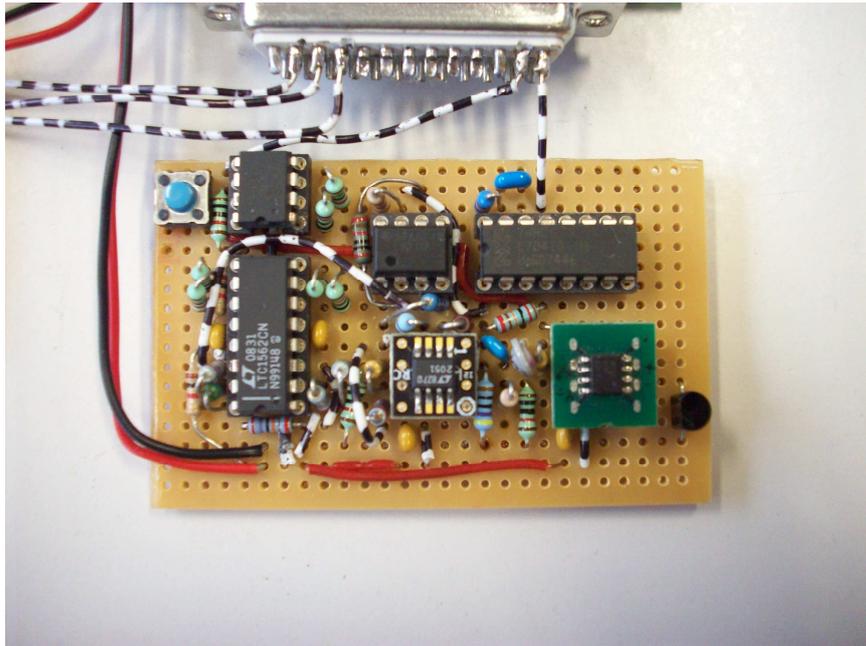


Abbildung 28: Aufbau der Empfängerschaltung

4.2.2.1 Transimpedanzverstärker

Der Transimpedanzverstärker dient zur Umwandlung des Photostromes in eine Spannung. Der Rückkoppelwiderstand R_1 wird mit $500 \text{ k}\Omega$ definiert. Um die Größe der Kapazität C_1 zu bestimmen, wird das Datenblatt des OPA380 herangezogen [18]. Ein weiterer Faktor zur Bestimmung der Kapazität im Rückkoppelzweig ist die Kapazität der PIN Diode. Die verwendete Diode SFH205 hat eine Kapazität C_{DIODE} von 72 pF [15]. Es ergibt sich laut [18] eine Kapazität zwischen $0,5 - 2 \text{ pF}$, siehe Abbildung 29 und 30. Der nächstgelegene Kapazitätswert beträgt 1 pF und wird somit verwendet.

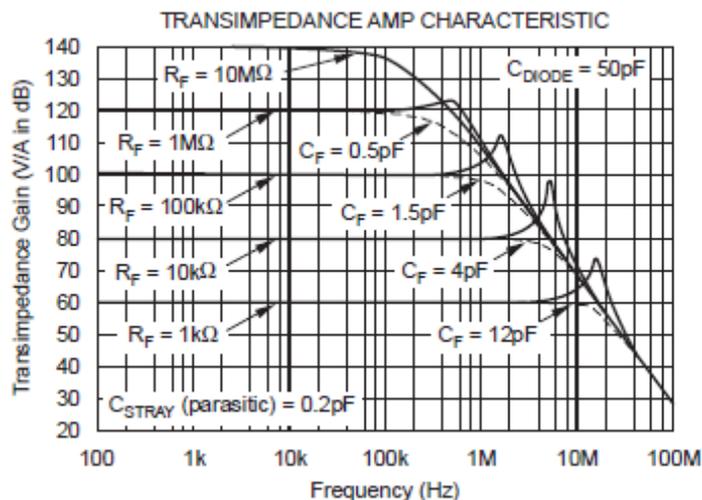


Abbildung 29: Verstärkung über die Frequenz mit C_{DIODE} von 50 pF , aus [18]

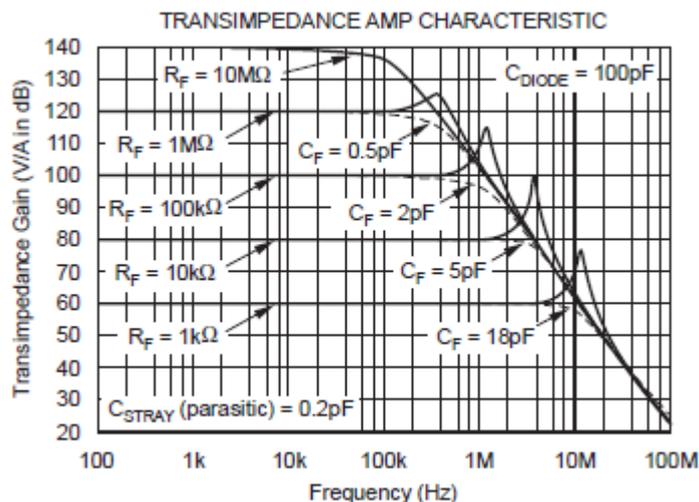


Abbildung 30: Verstärkung über die Frequenz mit C_{DIODE} von 100 pF, aus [18]

4.2.2.2 Verstärker

Nach dem Transimpedanzverstärker bringt ein Kondensator die durch die Sonneneinstrahlung erzeugte Gleichspannung auf Null. Dies muss geschehen da der folgende Verstärker mit einem fixen Verstärkungsfaktor arbeitet und die eintreffende Sonneneinstrahlung nicht konstant ist. Mit diesem Kondensator wird somit eine Übersteuerung des folgenden OPVs (LTC2051) durch Sonneneinstrahlung unterbunden. Würde man nur den Kondensator zwischen dem OPA380 und dem nicht invertierenden Verstärker schalten, würde der Biasstrom des nicht invertierenden Verstärkers nach einer gewissen Zeit den Kondensator voll aufladen und es könnte kein Signal mehr verstärkt werden. Deshalb wird mit Hilfe eines Spannungsteilers der Biasstrom zum einen gegen Masse abgeleitet und ein fixes Gleichspannungsniveau festgelegt. Zum anderen bildet der Kondensator mit dem Spannungsteiler einen Hochpass erster Ordnung. Nachdem der Spannungsteiler definiert wurde kann die Kapazität so gewählt werden, dass der Hochpass an der richtigen Stelle seine Grenzfrequenz besitzt. Für den Spannungsteiler werden die Widerstände R_2 mit 120 k Ω und R_3 mit 470 Ω gewählt:

$$U = \frac{R_3}{R_2 + R_3} \cdot U_V = \frac{470 \Omega}{118,6 \text{ k}\Omega + 470 \Omega} \cdot 5V = 19,73 \text{ mV}$$

Aufgrund der Toleranz des Widerstandes R_2 , beträgt sein gemessener Widerstandswert 118,6 k Ω . Der Widerstandswert von R_3 wurde mit den tatsächlichen 470 Ω gemessen.

Da ein Pulsintervall 25 μs dauert, dies entspricht 40 kHz, es aber beim Senden der Daten aufgrund der Codewörter zu keiner dauerhaften Wiederholung kommt, muss die Grenzfrequenz so gewählt werden, dass das Signal nicht abgeschwächt wird. Bei einer Codewortlänge von $N = 8$ ergibt sich mit der Pulsdauer von 25 μs eine Zeit von 200 μs

für ein Codewort. Diese Zeit entspricht einer Frequenz von 5 kHz in der die Datenbits der Messwerte eintreffen können. Die Grenzfrequenz f_G sollte somit unter 5 kHz liegen. Daraus resultiert die Kapazität des Hochpasses:

$$f_G = \frac{1}{2 \cdot \pi \cdot R_3 \cdot C_2} \rightarrow C_2 = \frac{1}{2 \cdot \pi \cdot R_3 \cdot f_G} = \frac{1}{2 \cdot \pi \cdot 470 \Omega \cdot 5 \text{ kHz}} = 67,72 \text{ nF}$$

Die Kapazität muss also größer als 67,72 nF sein. Daher wird eine Kapazität von 100 nF gewählt. Es ergibt sich somit die endgültige Grenzfrequenz bei:

$$f_G = \frac{1}{2 \cdot \pi \cdot R_3 \cdot C_2} = \frac{1}{2 \cdot \pi \cdot 470 \Omega \cdot 100 \text{ nF}} = 3386,27 \text{ Hz}$$

Der Verstärkungsfaktor des OPVs LTC2051 wird mit 121 definiert, wie aus folgender Berechnung hervorgeht:

$$V = 1 + \frac{R_4}{R_5} = 1 + \frac{1,2 \text{ M}\Omega}{10 \text{ k}\Omega} = 121$$

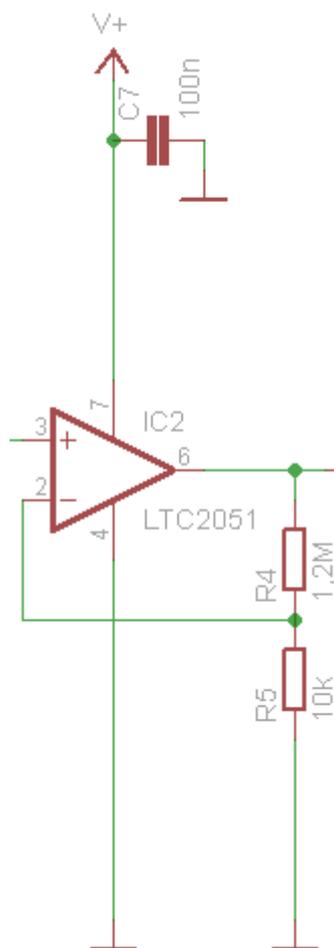


Abbildung 31: Beschaltung des Verstärkers LTC2051

Durch die Verstärkung der Signalspannung wird auch die zuvor durch den Spannungsteiler eingerichtete Gleichspannung verstärkt. Dadurch ergibt sich ein Gleichspan-

nungsniveau nach der Verstärkerstufe von 2,388 V. Dieser Spannungswert wird deshalb gewählt, da die erste Schaltschwelle des Schmitt-Triggers bei 2,5 V liegt, siehe Kapitel 4.2.2.4.

In der folgenden Abbildung 32 ist der Spannungsverlauf am Ausgang der Verstärkerstufe, nach einer Entfernung von 2 m dargestellt. Die zu erkennenden Störungen werden mit dem Filterbaustein (siehe Kapitel 4.2.2.3) entfernt.

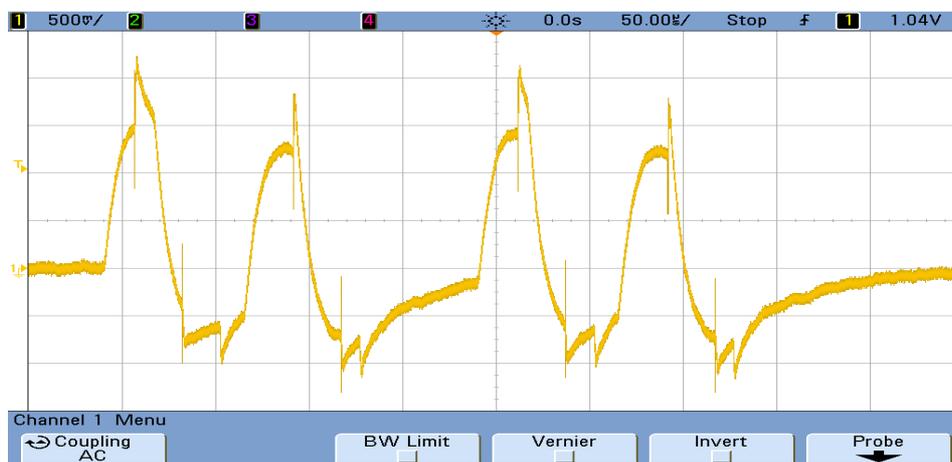


Abbildung 32: Empfangenes Signal nach der Verstärkerstufe

4.2.2.3 Filter

Der Filterbaustein wird mit Hilfe des Programmes FilterLAB von Linear Technology entworfen. Es wird ein Tiefpass Filter vierter Ordnung verwendet, da dieser Baustein unter Verwendung eines Bandpass Filters eine nicht optimale Impulsantwort liefert. Die Grenzfrequenz des Filters wird auf 20 kHz festgelegt. Sollte das Netzwerk so umgebaut werden, dass es mehr Teilnehmer nutzen können und/oder die Übertragungsgeschwindigkeit erhöht wird, müsste diese Grenzfrequenz ebenfalls erhöht werden.

In Abbildung 33 ist die Beschaltung des LTC1562 für *single supply* dargestellt. Die Eingänge A und D werden bei einem Filter vierter Ordnung nicht benötigt und werden somit mit 10 k Ω Widerständen beschalten.

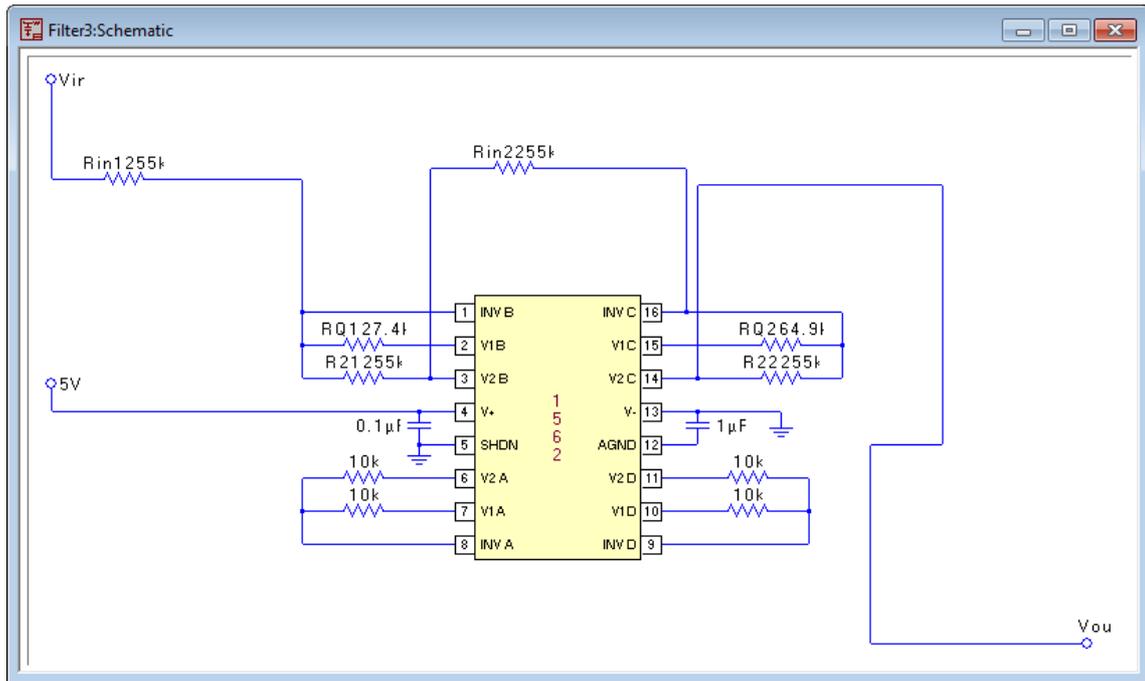


Abbildung 33: Beschaltung des Filterbausteins

Die Abbildung 34 zeigt den Amplitudenverlauf des Bodediagramms. Aus dem Bodediagramm ist ablesbar, dass hochfrequente Signale ab ca. 60 kHz mit 40 dB gedämpft werden.

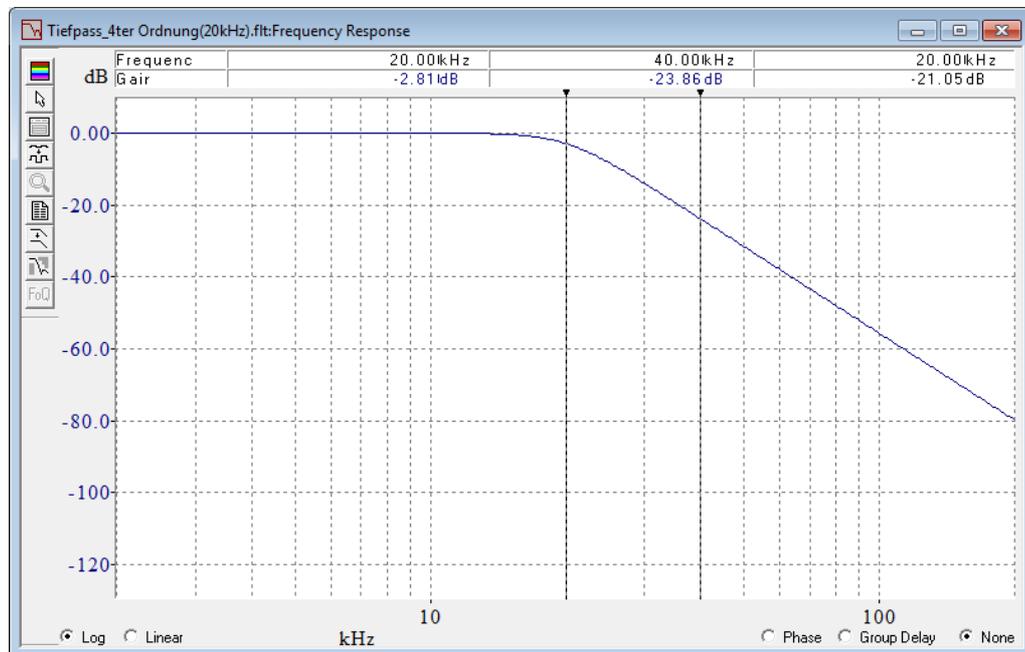


Abbildung 34: Bodediagramm des Filters

In Abbildung 35 wird die Impulsantwort des Filters dargestellt. Nach 70 μs schwingt der Ausgang des Filters noch einmal nach und übersteigt die Nulllinie nur minimal. Im Fall eines Bandpasses würde dieses Überschwingen sehr viel stärker ausfallen und man

könnte Probleme mit der nachfolgenden Schaltschwelle des Schmitt-Triggers bekommen.

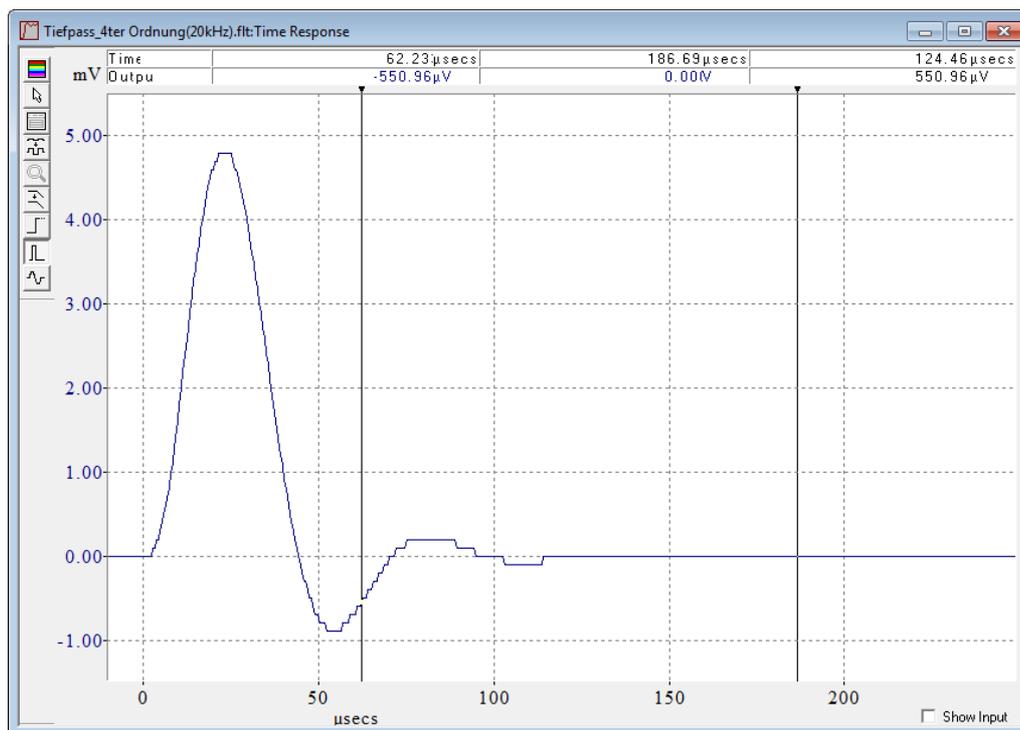


Abbildung 35: Impulsantwort des Filters

Im Vergleich mit Abbildung 32 ist aus Abbildung 36 ersichtlich, dass das Filter die Störeinflüsse sehr gut beseitigt.

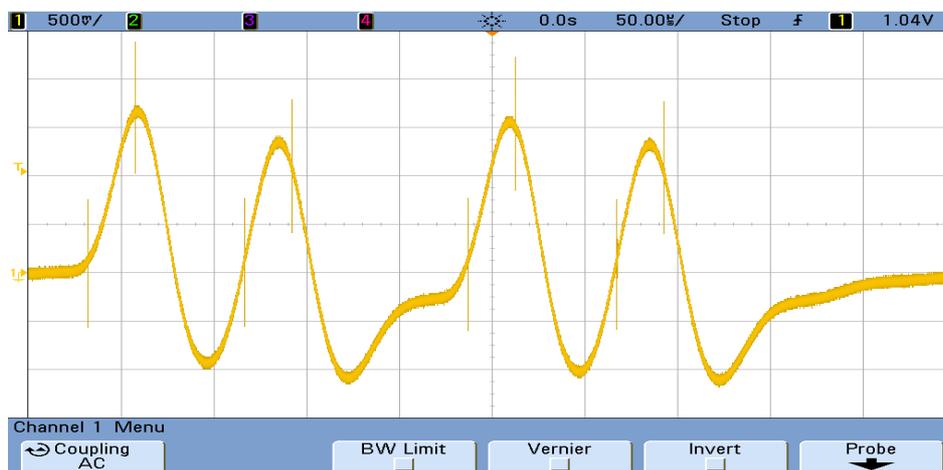


Abbildung 36: Empfangenes Signal nach dem Filter

4.2.2.4 Invertierender Schmitt-Trigger

Der invertierende Schmitt-Trigger [5], der in dieser Arbeit Verwendung findet, ist mit dem Komparator LM311 von Texas Instruments realisiert worden [19]. Die Begründung für die Verwendung eines Schmitt-Triggers anstatt eines normalen Komparators liegt

am Prellen der Spannung auf Höhe der Komparator Schwelle. Dadurch schaltet der Komparator den Ausgang in kurzer Zeit oft ein und aus, was eine Verlängerung der Pulsdauer nach dem Monoflop zur Folge hätte.

Durch das Beschalten des Komparators mit den 2 Widerständen R_{20} und R_{21} erhält man einen invertierenden Schmitt-Trigger, welcher im Gegensatz zum Komparator 2 Schaltschwellen besitzt, die voneinander versetzt sind. Diese liegen bei:

$$U_1 = \frac{R_{20} \cdot U_r + R_{21} \cdot U_v}{R_{20} + R_{21}} = \frac{20 \text{ k}\Omega \cdot 2,49 \text{ V} + 100 \Omega \cdot 5 \text{ V}}{20,1 \text{ k}\Omega} = 2,502 \text{ V}$$

$$U_2 = \frac{R_{20} \cdot U_r - R_{21} \cdot U_v}{R_{20} + R_{21}} = \frac{20 \text{ k}\Omega \cdot 2,49 \text{ V} - 100 \Omega \cdot 5 \text{ V}}{20,1 \text{ k}\Omega} = 2,452 \text{ V}$$

Das Gleichspannungsniveau vor dem Schmitt-Trigger beträgt 2,388 V, das heißt das jeder empfangene Spannungspuls, ab einer Höhe von $\sim 120 \text{ mV}$, die Schaltschwelle von U_1 übersteigt und zu einer Signaldetektion führt.

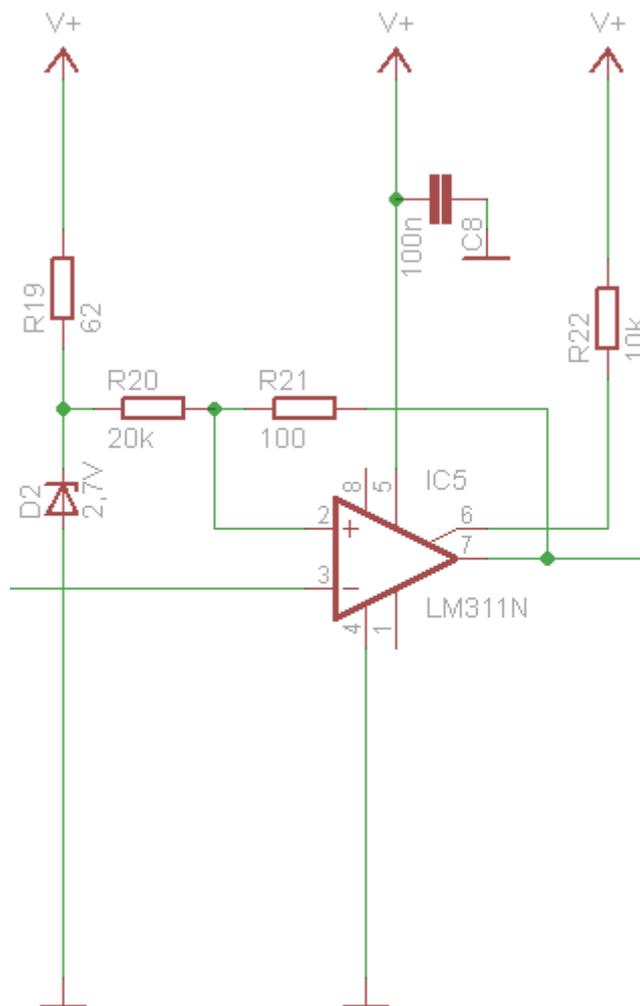


Abbildung 37: Komparator als Schmitt-Trigger beschalten

Durch den Widerstand R_{19} und durch die Zenerdiode D_2 wird die Spannung U_r auf 2,49 V konstant gehalten. Wird U_1 überschritten geht der Ausgang auf *high*, wo er so lange verweilt bis U_2 unterschritten wird. Der Widerstand R_{22} ist als reiner *Pullup*-Widerstand ausgeführt.

In Abbildung 38 ist das invertierende Verhalten des Schmitt-Triggers an Hand seiner Ausgangsspannung gut zu erkennen.

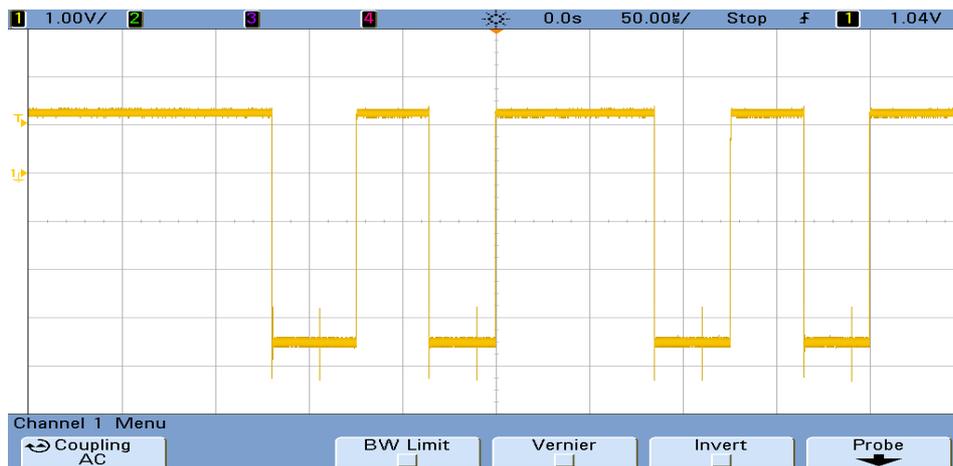


Abbildung 38: Empfangenes Signal nach dem Schmitt-Trigger

4.2.2.5 Monoflop

Das Monoflop 74HC123 von NXP bildet den Abschluss der Empfängerschaltung und dient der Signalaufbesserung nach dem Schmitt-Trigger. Die Aufgabe des Monoflops ist es, die Pulse des Schmitt-Triggers mit einer konstanten Pulsweite am Ausgang wiederzugeben. Diese Korrektur muss geschehen, weil die empfangenen und verstärkten Photostrompulse nicht immer exakt dieselbe ansteigende und fallende Flanke besitzen. Ohne das Monoflop würden Pulse mit variierender Pulsweite am Eingang des Mikrocontrollers anliegen.

Die Pulsbreite t_w kann mit der folgenden Formel aus dem Datenblatt berechnet werden [20]. Mit der gewählten Pulsweite von 25 μs und dem gewählten Widerstand R_{23} mit 51 $\text{k}\Omega$ (siehe Abbildung 39) kann man die benötigte Kapazität berechnen.

$$t_w = 0,55 \cdot R_{23} \cdot C_5 \rightarrow C_5 = \frac{t_w}{0,55 \cdot R_{23}} = \frac{25 \mu\text{s}}{0,55 \cdot 51 \text{k}\Omega} = 891,26 \text{ pF}$$

Es wird ein Kondensator mit einem Kapazitätswert von 1 nF gewählt. Aufgrund der Toleranz, welche der Widerstand und der Kondensator besitzen, können die 25 μs nicht exakt erreicht werden. Die genaue Pulsweite, welche von dem Monoflop erzeugt wird beträgt ca. 25,5 μs .

In Abbildung 40 sind die endgültigen Rechteckpulse der Empfängerschaltung dargestellt. Die leichte Pulsweitenverlängerung ist hier nicht gut zu erkennen. Da dieses Mo-

noflop einen invertierenden Eingang besitzt, wird die Signalleitung vom Ausgang des invertierenden Schmitt-Triggers mit diesem Eingang verbunden, siehe Abbildung 27.

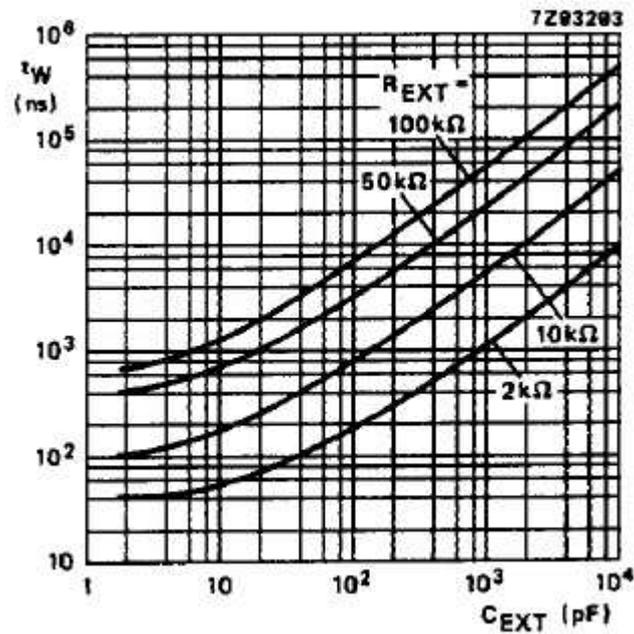


Abbildung 39: Ausgangspulsweite als Funktion der externen Kapazität, aus [21]



Abbildung 40: Empfangenes Signal nach dem Monoflop

5 Implementierung der Sensor- und Hauptstation

Diese Kapitel befasst sich mit dem Aufbau der Sensor- und Hauptstation. Alle Stationen haben als Basis das AVR-Net-IO-Board, siehe Kapitel 5.1. Es wird auf die entwickelte Software, für die jeweiligen Stationen eingegangen und des Weiteren erklärt, weshalb die Chiprate bei 2 Sensorstationen auf 40 kHz beschränkt ist.

5.1 AVR-Net-IO-Board

Das AVR-Net-IO-Board, in Abbildung 41 dargestellt, besteht aus einer Ethernet Platine, die mit dem Netzwerkcontroller ENC28J60 und einem ATmega32 Mikrocontroller ausgestattet ist.

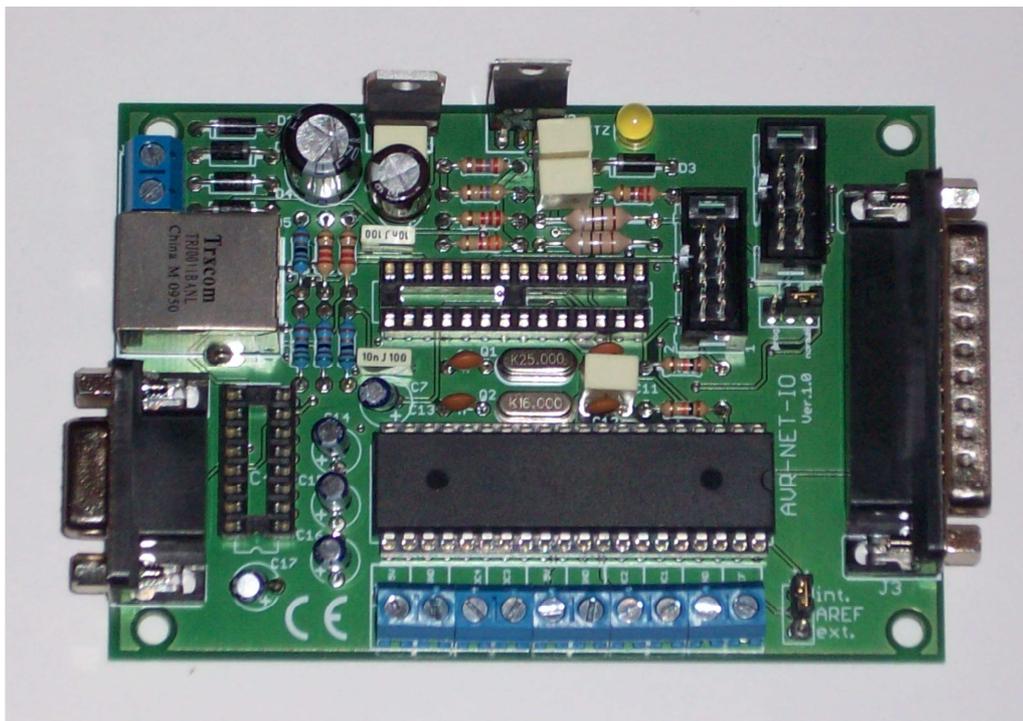


Abbildung 41: AVR-Net-IO Board

Dieses Board wird gewählt, da es mit einem Preis von 19,99 € sehr günstig ist und das benötigte ISP (*In-System-Programmer*) Programmiergerät bereits am Institut vorhanden ist.

Das AVR-Net-IO-Boards ist durch folgende Eigenschaften definiert:

- Mikrocontroller ATmega32, 16 MHz Quarz
- Netzwerkcontroller ENC28J60
- 8 digitale Ausgänge die über den 25 pol. Sub-D-Stecker abgegriffen werden

- 4 digitale Eingänge die ebenfalls den 25 pol. Sub-D-Stecker zugeführt werden
- 4 ADC-Eingänge (10 Bit) die direkt über die Anschlussklemmen eingespeist werden
- ISP-Steckerleiste (Atmel-Standardbelegung) für die Programmierung des Mikrocontroller mit geeignetem Schnittstellenadapter
- RS232-Schnittstelle (Tx, Rx) mit Pegelanpassung über MAX232
- Separate AVRef-Referenzspannungszuführung
- 10-polige Pfostenleiste (Extension-Port) für weitere Mikrocontroller-Ports
- Betriebsspannung 9 V~
- Max. Stromaufnahme ca. 190 mA

Die vorhandene Ethernet Schnittstelle sowie die serielle Schnittstelle eignen sich dafür, die empfangenen Daten der Hauptstation an einen PC weiter zu leiten. Die vorhandene Anzahl an Ein- und Ausgänge reichen für diese Arbeit vollkommen aus.

Das AVR-NET-IO-Board biete die Möglichkeit die Stromversorgung mittels einem 9 V~ Netzgerät zu sichern. Es ist allerdings auch möglich das Board mittels einer Batterie zu versorgen, falls eine Anbindung an eine Steckdose wegen räumlicher Einschränkungen nicht realisiert werden kann. Dazu muss man allerdings einen Batterie Anschluss nach dem Gleichrichter einlöten. Die gesamte Schaltung des AVR-NET-IO-Boards ist in Abbildung 42 dargestellt. Die Sende- und Empfängerschaltung werden über die 10 polige Pfostenleiste mit 5 V versorgt.

5.2 Hauptstation

Die Aufgabe der Hauptstation besteht darin, das empfangene Signal mittels einer Korrelationsfunktion dem jeweilig sendenden Teilnehmer zuzuordnen, und steuernde Befehl an die Netzwerkteilnehmer zu versenden.

Die richtig zugeordneten Daten werden im Speicher abgelegt. Es wird keine Routine zum Versenden der Daten via Ethernet oder RS232 eingebunden, da dies auch nicht Inhalt der Aufgabenstellung dieser Arbeit war.

Die steuernden Befehle reduzieren sich in dem entwickelten Programm auf einen einzigen. Dieser Befehl ist eine Start/Stopp Anweisung, die über einen Taster auf der Empfängerplatine der Hauptstation ausgeführt ist und in Abbildung 43 an der linken unteren Ecke der Empfängerplatine zu finden ist. Des Weiteren zeigt Abbildung 43 den Prototyp der Hauptstation.

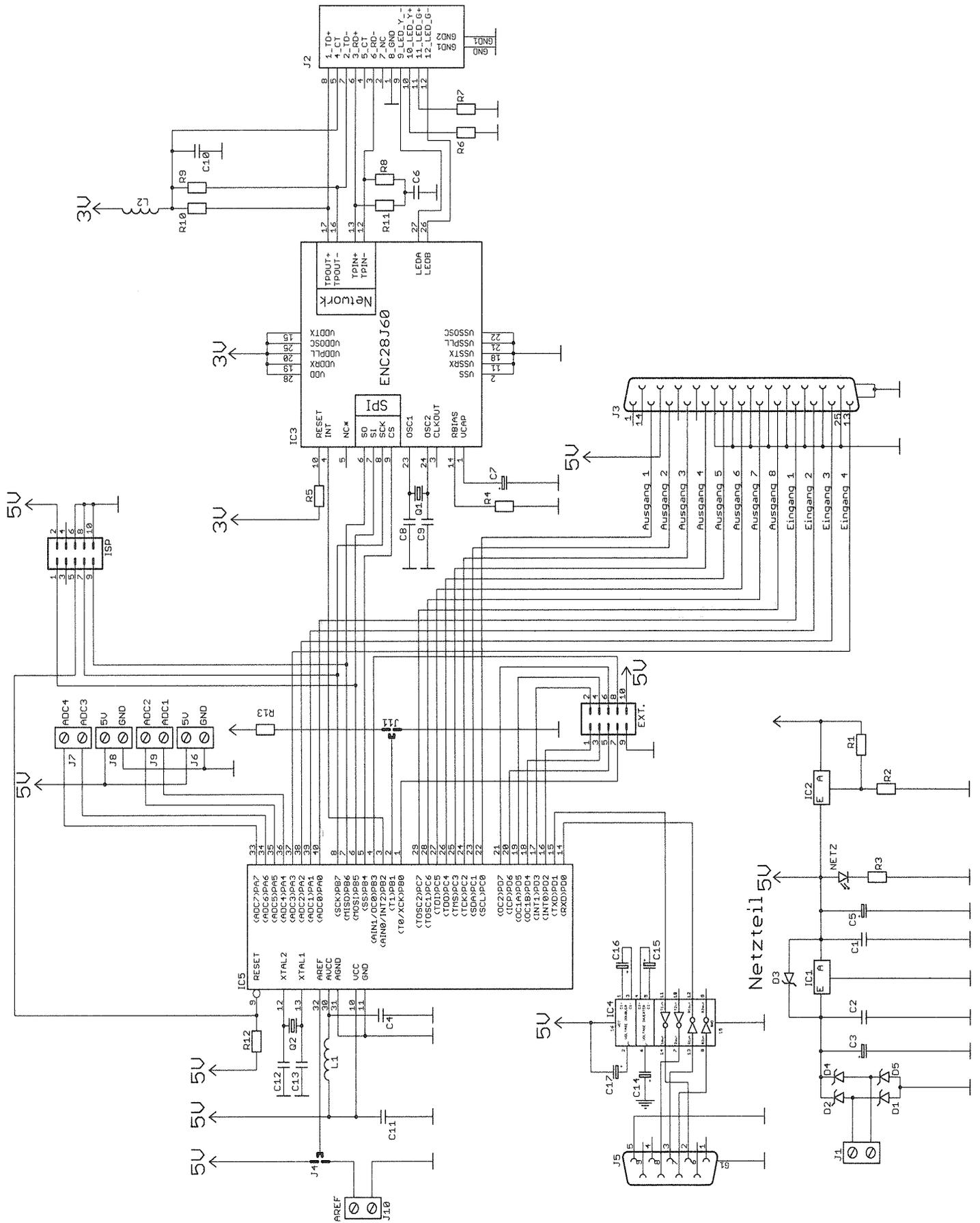


Abbildung 42: Schaltbild des AVR NET IO Board

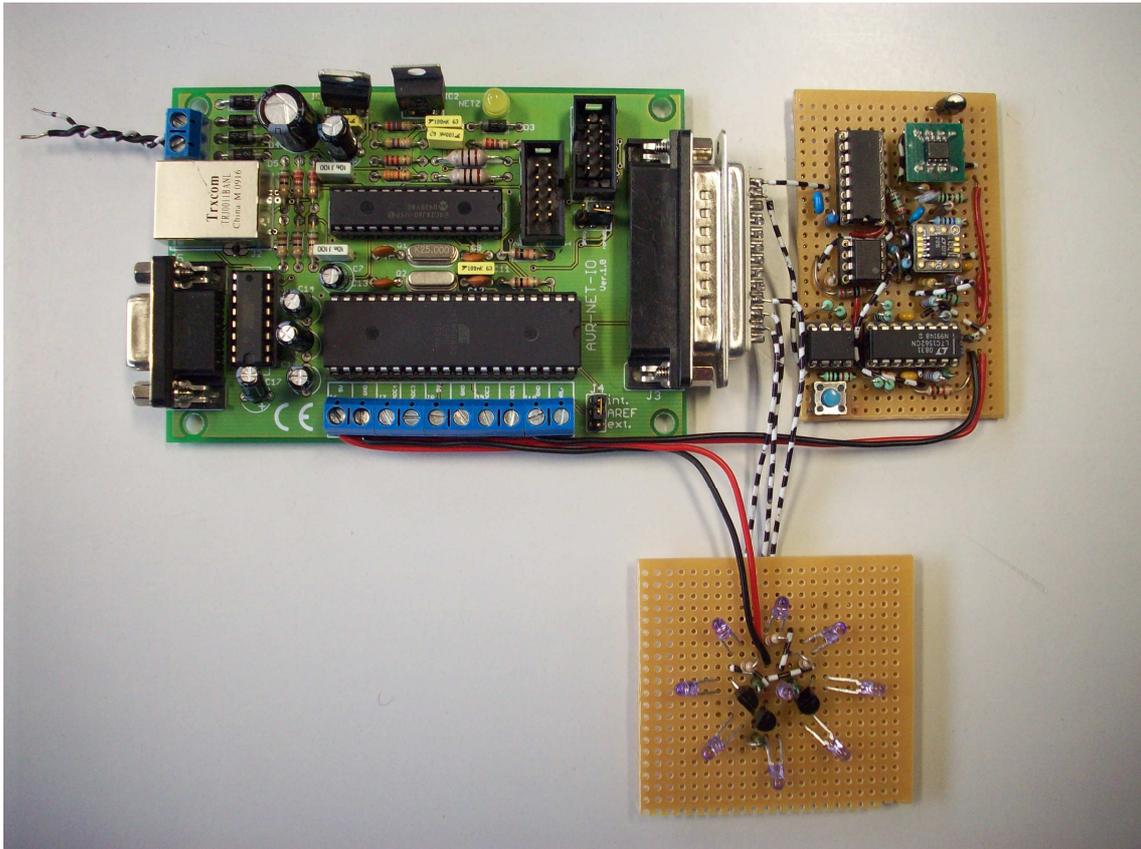


Abbildung 43: Prototyp der Hauptstation

Die Hauptstation muss nach jedem erkannten „0“ oder „1“ Chip die Korrelationsfunktion von 2 Teilnehmern berechnen (siehe Anhang A) und darauf folgend überprüfen ob das Ergebnis einem Teilnehmer zuordenbar ist. Dieser ganze Vorgang beeinflusst die Übertragungsgeschwindigkeit, da er abgeschlossen sein muss, bevor der nächste Chip empfangen wird. In Kapitel 3.4 wurde erwähnt dass die Chiprate 40 kHz beträgt. Diese Frequenz könnte mit einer entsprechenden Optimierung der Software auf bis zu 50 kHz erhöht werden. Höhere Frequenzen können mit dem AVR-NET-IO Board in dieser Ausstattung nicht erreicht werden, da mit einer Taktfrequenz von 16 MHz, was einer einzelnen Taktdauer von 62,5 ns entspricht (siehe Berechnung), der gesamte Korrelationsvorgang für 2 Teilnehmer nicht schneller abgewickelt werden kann.

$$t_{\text{Zyklus}} = \frac{1}{f_{\text{Quarz}}} = \frac{1}{16 \text{ MHz}} = 62,5 \text{ ns}$$

Wie viele Taktzyklen der Mikrocontroller für die einzelnen Operationen benötigt ist in [23] zu finden. Für das entwickelte Programm beträgt die Dauer eines Durchlaufes des Korrelationsvorganges 22 μs . Das bedeutet, dass der ganze Vorgang um 3 μs schneller als notwendig ist und könnte bei entsprechender Optimierung noch ausgenutzt werden.

Die folgende Abbildung 44 zeigt das Flussdiagramm der Software für die Hauptstation. Das gesamte Programm ist im Anhang A1 zu finden. Dort sind als Verständnishilfe der einzelnen Programmschritte Kommentare in grüner Schrift am Seitenrand platziert.

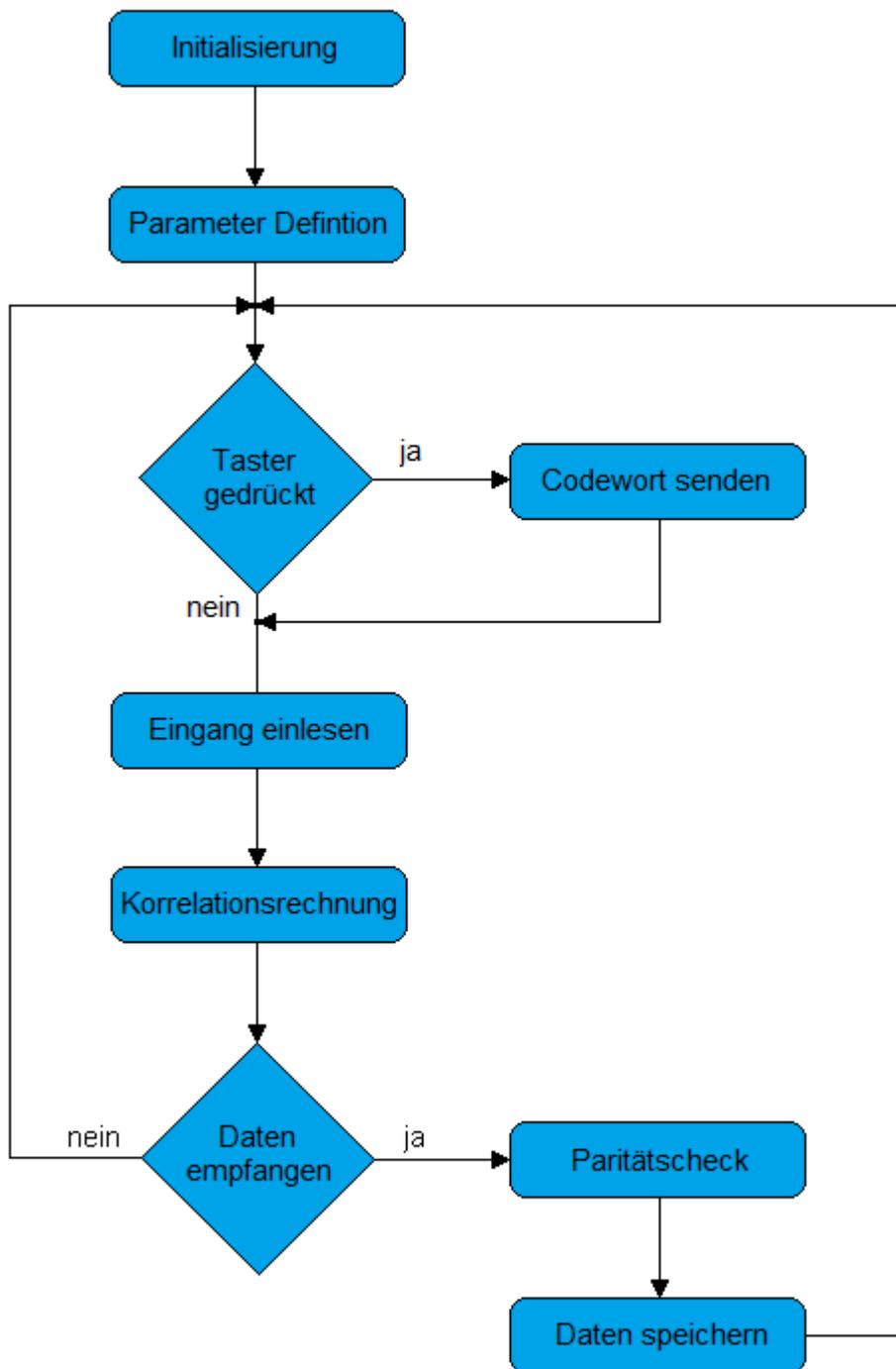


Abbildung 44: Flussdiagramm der Hauptstationssoftware

Nach der Initialisierung der einzelnen Bibliotheken werden die Parameter festgelegt. Hierbei handelt es sich um die Definition der Variablen und die Timereinstellungen. Danach beginnt die Hauptschleife mit dem Abfragen des Eingangs an dem der Taster liegt. Wird der Taster betätigt, wird das Codewort mit Hilfe eines Timers gesendet und bei den Sensorstationen als steuernder Befehl interpretiert.

Als nächstes wird das Port eingelesen an dem die Empfängerschaltung angeschlossen ist. Die so ermittelten Chip Zustände werden durch ein 8 Bit langes Register geschoben und mit jedem neuen Chip eine Korrelationsberechnung durchgeführt. Korreliert die empfangene Chipkonstellation mit einem der zu vergleichenden Codewörter 3-mal in Serie wird damit begonnen die Datenbits dieser Sensorstation zu speichern.

Nachdem ein ganzes Datenpaket empfangen und gespeichert ist, wird überprüft ob es eine Fehlinterpretation bei den Datenbits gibt. Dies geschieht mit der Paritätsprüfung, welche die vorhandenen „1“ im Datenpaket addiert und nachprüft ob die Anzahl dieser gerade ist. Fällt die Überprüfung positiv aus wird der Messwert ohne Paritätsbit endgültig abgespeichert, bei einem negativen Ergebnis, sprich einer ungeraden Anzahl an „1“ im Datenpaket, wird das ganze Datenpaket verworfen.

5.3 Sensorstation

Die Sensorstationen sollen die gemessenen Werte des Temperatur- und Luftfeuchtigkeitssensors mittels ihrer CDMA Codewörter zur Hauptstation übertragen. Dies soll erst geschehen, wenn die Stationen von der Hauptstation dazu aufgefordert werden. In Abbildung 45 wird der Prototyp der Sensorstation dargestellt.

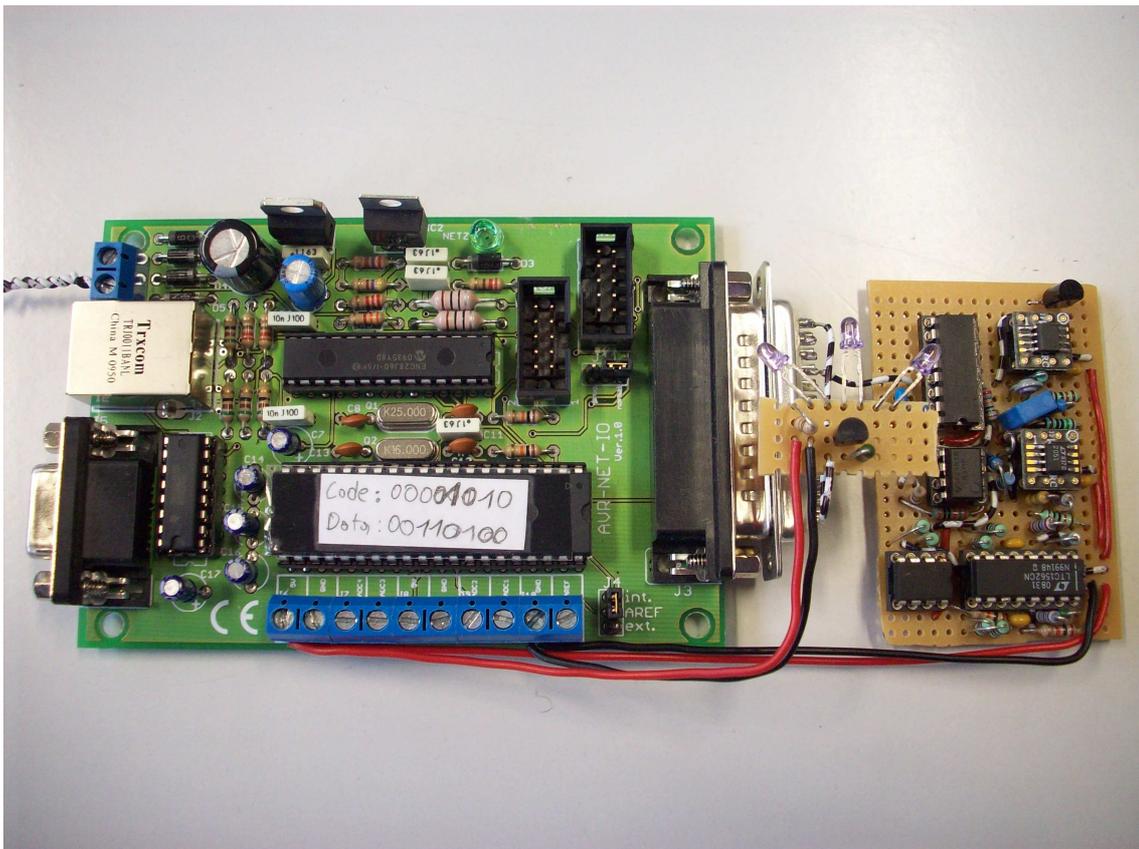


Abbildung 45: Prototyp der Sensorstation

5.3.1 Testsensor

Der Temperatur- und Luftfeuchtigkeitssensor SHT71 [22], siehe Abbildung 46, wird für diese Arbeit verwendet. Er kann die Temperatur im Bereich von -40 °C bis 120 °C und die Luftfeuchtigkeit von 0 bis 100% ermitteln. Dieser Bereich ist in Abbildung 47 festgehalten. Des Weiteren besteht die Möglichkeit die Auflösung der Messwerte zu verändern. Die Luftfeuchtigkeit kann in 8 bzw. 12 Bit und die Temperatur in 12 bzw. 14 Bit dargestellt werden.

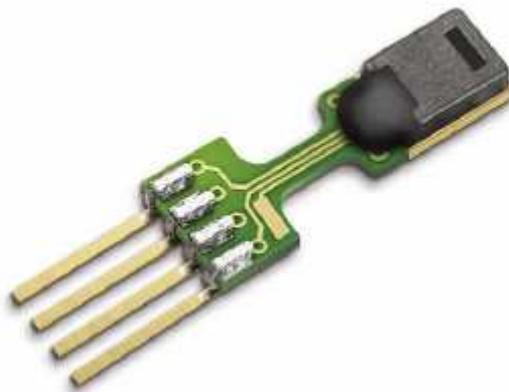


Abbildung 46: Testsensor SHT71 von Sensirion, aus [22]

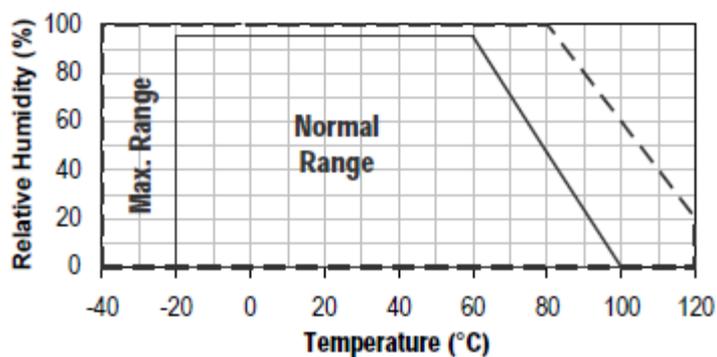


Abbildung 47: Messbereich des Sensors, aus [22]

Der Sensor wird über den I²C Bus an den Mikrocontroller angeschlossen. Die Datenleitung wird über einen *Pullup*-Widerstand mit der Betriebsspannung verbunden. Abbildung 48 zeigt wie der Sensor schematisch an den Mikrocontroller angeschlossen wird.

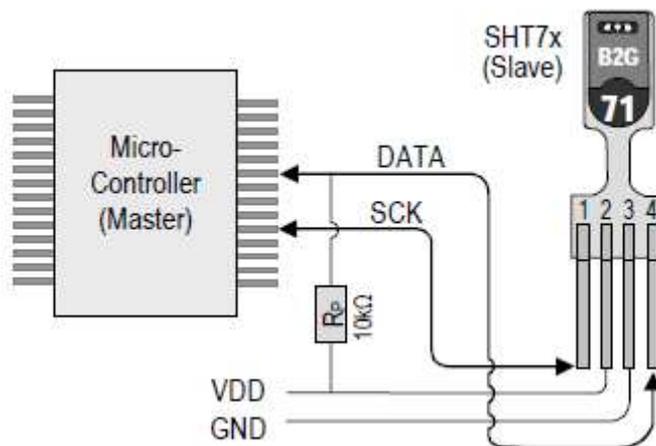


Abbildung 48: Typischer Anschluss des Sensors an den Mikrocontroller, aus [22]

5.3.2 Software der Sensorstationen

Da der Sensor zum Zeitpunkt der Testphase nicht verfügbar war wurden willkürliche Daten verwendet die der Sensor auch erzeugen kann. Des Weiteren wird das Programm vereinfacht, in dem diese Messwerte vordefiniert werden. Das dazu entwickelte Programm ist im Anhang A2 zu finden.

Der C Code startet ähnlich dem Hauptprogramm mit der Initialisierung der Bibliotheken und dem Festlegen der einzelnen Parameter, die für die Übertragung benötigt werden. Da die Sensorstation nicht ohne den dementsprechenden Befehl der Hauptstation zu messen beginnt, wird als Erstes der Eingang abgefragt, an dem die Empfängerschaltung angeschlossen ist. Es folgt die Korrelationsberechnung, die bei den Sensorstationen auf das Codewort der Hauptstation angepasst ist. Ergibt die Korrelationsberechnung eine Übereinstimmung, kann die Messung gestartet werden. Der Messwert wird nun eingelesen um das Paritätsbit bestimmen zu können. Ist dies geschehen, wird ein Datenpaket inklusive *Preamble*, Daten und Paritätsbit an die Hauptstation gesendet.

Wird der Start/Stopp Befehl empfangen während bereits eine Messung im Gange ist, so wird dies als Stopp Befehl verstanden und das Aufnehmen des Messwertes sowie das Senden der Daten wird unterbunden. In Abbildung 49 ist das Flussdiagramm der Software für die Sensorstation dargestellt.

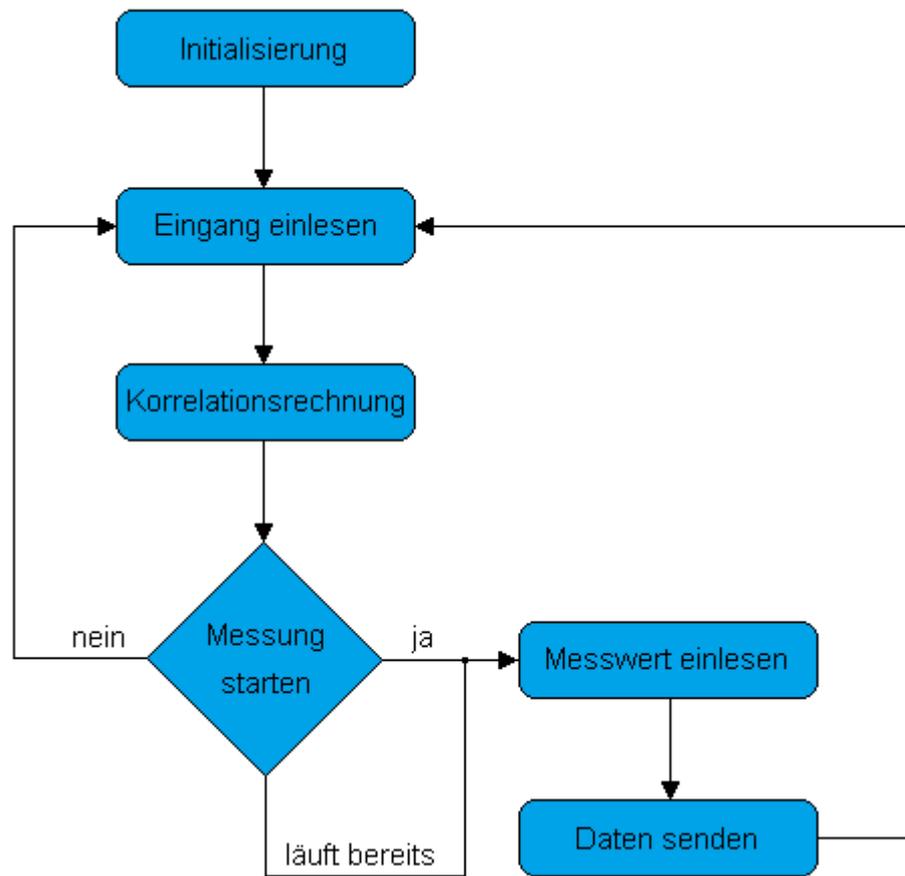


Abbildung 49: Flussdiagramm der Sensorstationssoftware

6 Leistungsbilanz (Linkbudget) und Reichweitenmessung

In diesem Kapitel wird die Leistungsbilanz berechnet und mit den tatsächlichen Messwerten verglichen. Ebenso wird ein Diagramm der gemessenen Spannung über die Entfernung angeführt und daraus die maximale Reichweite der Verbindung ermittelt. Außerdem wird gezeigt dass die Übertragung in einem Raum mit Neonröhren, die gerade erst eingeschaltet wurden, komplett zusammen bricht.

Die Leistungsbilanz einer optischen Übertragungsstrecke lässt sich wie folgt näher beschreiben. Da die gewählte LED in dieser Arbeit eine sehr kleine Abstrahlfläche besitzt, welche laut Datenblatt [14] $0,09 \text{ mm}^2$ beträgt, wird sie als punktförmiger Strahler angenommen. Mit dem Abstrahlwinkel von $\alpha = 40^\circ$ lässt sich nun die bestrahlte Fläche auf eine bestimmte Entfernung bestimmen. Die bestrahlte Fläche für ein Kugelsegment ergibt sich mit der Formel:

$$A_{KK} = 2\pi r^2 \cdot \left(1 - \cos\left(\frac{\alpha}{2}\right)\right)$$

Nachdem die bestrahlte Fläche berechnet ist, gilt es die abgestrahlte Leistung der LED zu bestimmen. Die abgestrahlte Leistung ϕ_e beträgt laut [14] bei einem Vorwärtsstrom von 100 mA 25 mW . In Kapitel 4.1.2 wurde die Berechnung des Vorwärtsstromes bereits mit $92,3 \text{ mA}$ festgehalten. Mit diesem Stromwert kann man in Abbildung 51 die Veränderung der Strahlstärke in Bezug auf den 100 mA Wert ermitteln. Die abgestrahlte Leistung verringert sich um den Faktor $0,95$ zu $23,75 \text{ mW}$. Mit der bestrahlten Fläche und der abgegebenen Leistung wird die Leistung E_e in μW pro Quadratzentimeter bestimmen.

$$E_e = \frac{\phi_e}{A_{KK}}$$

Mit der errechneten Leistung pro Quadratzentimeter E_e lässt sich der erzeugte Photostrom aus dem Datenblatt [15] der PIN Diode herauslesen, siehe Abbildung 50 [15]. Damit der Photostrom mit den gemessenen Spannungswerten verglichen werden kann, wird der Photostrom mit dem Rückkoppelwiderstand des Transimpedanzverstärkers multipliziert und somit in eine Spannung U_{Trans} umgewandelt. Als nächstes wird die Spannung U_{Trans} mit dem Verkungsfaktor 121 multipliziert. Dies ist deshalb von Nöten, da bei weiteren Entfernungen die Spannung nach dem Transimpedanzverstärker mit den vorhandenen Mitteln nicht mehr erfasst werden kann. Die kalkulierten Werte der Linkbudget Berechnung sind in Tabelle 4 aufgelistet.

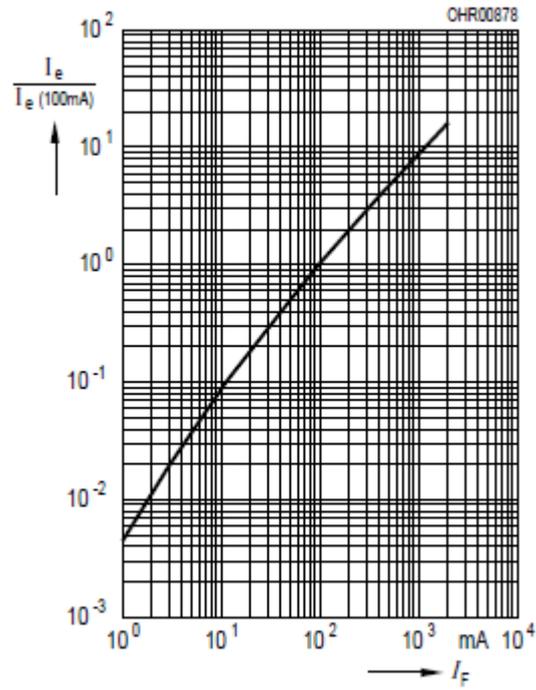


Abbildung 50: Leuchtstärke als Funktion des Vorwärtsstromes; $t_p = 20 \mu\text{s}$, aus [14]

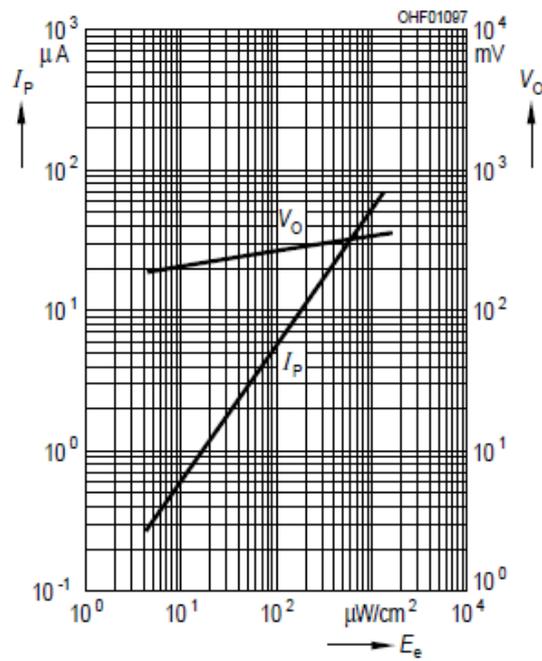
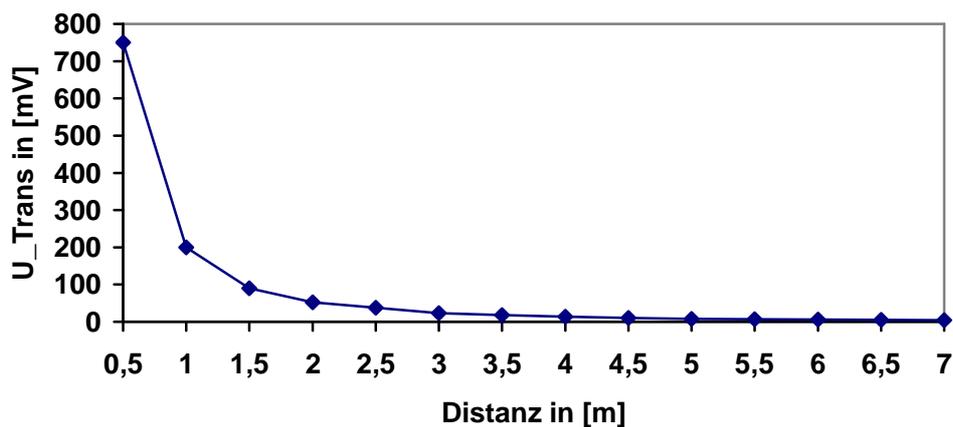


Abbildung 51: Photostrom als Funktion der Lichtleistung, aus [15]

Distanz in [m]	A_{KK} in [cm ²]	E_e in $\left[\frac{\mu W}{cm^2}\right]$	Photostrom in [nA]	U_{Trans} in [mV]
0,5	947,306	25,07	1500	750
1	3789,224	6,26	400	200
1,5	8525,754	2,785	180	90
2	15456,897	1,536	105	52,5
2,5	23682,652	1,002	75	37,5
3	34103,019	0,696	55	23
3,5	46417,998	0,511	36	18
4	60627,59	0,391	28	14
4,5	76731,793	0,309	21	10,5
5	94730,609	0,25	18	9
5,5	114624,037	0,207	14	6,5
6	136412,078	0,174	12	6
6,5	160094,73	0,148	10,5	5,25
7	185671,995	0,127	9	4,5

Tabelle 4: Berechnete Spannung U_{Trans} von 0,5 m bis 7 mAbbildung 52: U_{Trans} über eine Distanz von 7 m

In Abbildung 52 sind die in Tabelle 4 berechneten Spannungswerte über die Distanz aufgetragen und als Diagramm dargestellt.

Zum Vergleich folgen nun die gemessenen Spannungswerte für eine Entfernung von 0,5 m bis 7 m. In Abbildung 53 ist der Aufbau für diese Messung zu sehen.

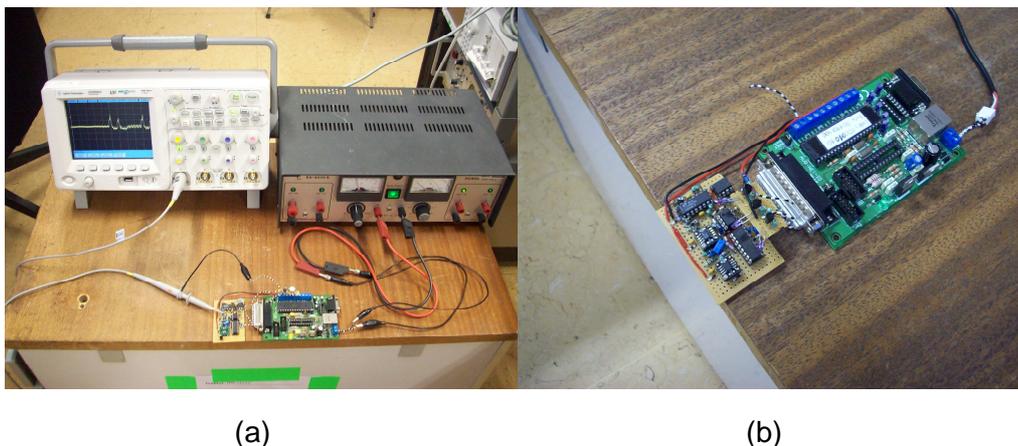


Abbildung 53: a) Hauptstation mit Oszilloskop; b) Sensorstation

Distanz in [m]	U_{Filter} in [V]
0,5	5
1	5
1,5	2,25
2	2,5
2,5	1,1
3	0,8
3,5	0,65
4	0,5
4,5	0,42
5	0,35
5,5	0,27
6	0,24
6,5	0,21
7	0,17

Tabelle 5: Gemessene Werte der Filterausgangsspannung U_{Filter}

Da der verwendete Verstärker OPV (LTC2051) am Ausgang nicht mehr als 5 V bereitstellt wenn er angesteuert ist, kann der Spannungswert bei einer Distanz von 0,5 m und 1 m die 5 V auch nicht übersteigen [17]. Das bedeutet, dass durch den gewählten Sendestrom und den fixen Verstärkungsfaktor eine Verbindung unter einem Meter

nicht richtig funktioniert. Dadurch wird das folgende Diagramm (Abbildung 54) ein wenig verfälscht. Die anderen Messpunkte stimmen gut mit dem Verlauf des Diagramms in Abbildung 52 überein. Die errechneten Spannungswerte konnten allerdings nur näherungsweise erreicht werden. Somit ergibt sich eine maximale Reichweite von 7 m, da die Höhe von U_{Filter} bei einer größeren Entfernung nicht mehr ausreicht, um die erste Schaltschwelle des Schmitt-Triggers auszulösen. Strahlt das gesendete Licht über eine Reflexion auf den Empfänger, verringert sich die maximale Reichweite auf 4 m. Dies geschieht aufgrund der Tatsache, dass der Lichtstrahl bei jeder Reflexion an Leistung verliert [1]. Der Leistungsverlust ist natürlich auch von der Materialoberfläche abhängig, an dem das Licht reflektiert wird.

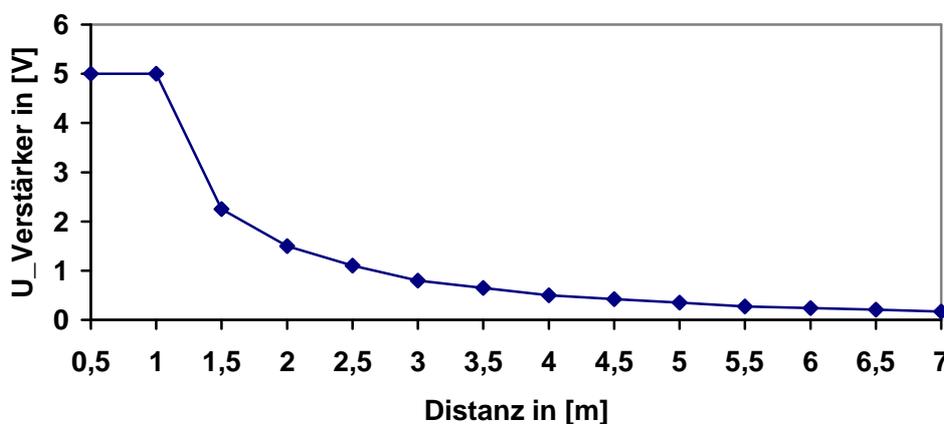


Abbildung 54: U_{Filter} über eine Distanz von 7 m

Das erstellte Netzwerk arbeitet mit 3 Teilnehmern innerhalb einer Reichweite von 1 - 7 m einwandfrei. Es wurden allerdings zwei Probleme festgestellt. Das erste Problem tritt ein, wenn der Empfänger starker direkter Sonneneinstrahlung ausgesetzt ist. In diesem Fall ist die Störung durch das Sonnenlicht zu groß als das sie durch das Filter behoben werden kann. Beim zweiten Problem handelt es sich um Neonröhren. Sie strahlen nach dem Einschalten Licht mit einem breiten Spektrum ab. Das bedeutet, dass Neonröhren Licht im Wellenlängenbereich der verwendeten PIN Diode abstrahlen. Dieses Licht stört das Empfangene Signal in der Form, dass die Übertragung ebenfalls nicht mehr detektiert werden kann.

7 Zusammenfassung und Ausblick

Die gesteckten Ziele für diese Diplomarbeit können mit den folgenden Angaben zur Gänze erfüllt und umgesetzt werden. So wird ein günstiges, kabelloses optisches Infrarot Sensornetzwerk entwickelt, welches für 3 Teilnehmer ausgelegt ist. Das Netzwerk wird von einer Hauptstation, mit einem Start/Stopp Befehl gesteuert, während die 2 Sensorstationen auf diesen Befehl hin ihre gemessenen Daten an die Hauptstation übertragen. Das gesamte System wird also mit einem bidirektionalen Übertragungsweg realisiert.

Die Teilnehmer stellen den Zugriff auf das Medium mittels CDMA sicher. Hierbei wird mit 8 Chips langen OOC Codewörter gearbeitet. Die Parameter für das verwendete OOC Set werden mit $\Phi(8,2,1)$ notiert. Die Datenübertragungsrate beträgt bei einer Chiprate von 40 kchip/s (mit 8 Chips langen Codewörtern) 5 kbit/s. Des Weiteren kann mit Hilfe einer Paritätsbitprüfung ein Fehler in der Übertragung erkannt werden und der falsch empfangene Messwert wird daraufhin aussortiert.

Um die Funktion des Netzwerkes testen und überprüfen zu können wird die Software für die Haupt- und Sensorstationen in der Programmiersprache C mit Hilfe des Programmes AVR Studio 4 implementiert. Es konnte eine Reichweite für die direkte Übertragung (LOS) bis zu 7 m und bei einer Reflexion bis zu 6 m erreicht werden.

Es besteht die Möglichkeit das Netzwerk in jeglicher Richtung auszubauen und weiterzuentwickeln. Mit einem schnelleren Mikrocontroller könnte beispielsweise die Datenübertragungsrate (und die Chiprate) gesteigert werden. Eine schwierigere Aufgabe ist es, die Teilnehmerzahl zu erhöhen, da dadurch die Codewortlänge steigt und somit die Korrelationsrechnung mehr Zeit in Anspruch nimmt. Ebenso müsste ein neues OOC Set erstellt werden, das den Ansprüchen eines größeren Teilnehmerfeldes gewachsen ist.

Eine höhere Reichweite könnte erzielt werden, wenn der Empfänger als *Angle Diversity Receiver* neu designt wird. Ein Punkt, der bei dieser Arbeit aufgrund von Kostengründen nicht beachtet wurde, ist die Verbesserung der Optik. Mit entsprechenden Filtern oder Linsen könnte die Reichweite ebenfalls erhöht werden. Ein weiterer Ansatzpunkt wäre die Verbesserung der Software der Stationen. Durch das Optimieren der Software könnte die Datenübertragungsrate gesteigert werden.

Literaturverzeichnis

- [1] J. R. Barry, "*Wireless Infrared Communications*", Georgia Institute of Technology, 1994
- [2] P. Fasser, E. Leitgeb, "Optische Nachrichtentechnik", 2007
- [3] O. Koudelka, "Kommunikationsnetze", Institut für Kommunikationsnetze und Satellitenkommunikation
- [4] W. Riedler, "Informationstheorie und Codierung", 1. Auflage WS1999/2000
- [5] U. Tietze und Ch. Schenk, "Halbleiterschaltungstechnik", 13. Auflage, 2009
- [6] N. Kalyandrug, S. Krishna "*IR communications for indoor wireless – an analysis*" (1997) *International Conference on Information, Communications and Signal Processing*
- [7] F.E. Alsaadi and J.M.H. Elmirghani, "*MC-CDMA Indoor Optical Wireless System*", 2007
- [8] J.B.Carruthers and J.M. Kahn, "*Angle Diversity for Nondirected Wireless Infrared Communication*", IEEE, 2000
- [9] A. Ramli, S. M. Idrus and A. S. M. Supa'at, "*Optical Wireless Front-End Receiver Design*", 2008
- [10] P. Kamath, J.D. Touch and J.A. Bannister, "*The Need for Media Access Control in Optical CDMA Networks*", 2004
- [11] F.R.K. Chung, J. A. Salehi and V.K. Wie, "*Optical Orthogonal Codes: Design, Analysis, and Applications*", 1989
- [12] P. Kamath, J.D. Touch and J.A. Bannister, "*Algorithms for Interference Sensing in Optical CDMA Networks*", 2004
- [13] P. Rako, "*Photodiode Amplifiers – Changing Light to Electricity*", 2004 National Semiconductor
- [14] Osram SFH487 GaAIAs-IR-Lumineszenzdiode (880nm)
- [15] Osram SFH205 FA Silizium-PIN-Fotodiode mit Tageslichtsperrfilter
- [16] Linear Technology LTC1562-2, *Very low noise, low distortion, active RC quad unicersal filter*
- [17] Linear Technology LTC2051, *Dual Zero-Drift Operational Amplifier*
- [18] Texas Instruments OPA380, *Precision, High-Speed Transimpedance Amplifier*
- [19] Texas Instruments LM311, *Differential Comparator with Strobe*
- [20] NXP 74HC123, *Dual retriggerable monostable Multivibrator with reset*

- [21] ON Semiconductor BC639, *NPN Silicon High Current Transistor*
- [22] Sensirion SHT71, *Humidity and Temperature Sensor*
- [23] Atmel ATmega32, *8 Bit AVR Microcontroller with 32 KBytes*

Abbildungsverzeichnis

Abbildung 1: Darstellung der Übertragungsstrecke, übernommen aus [6]	4
Abbildung 2: Mögliche Linkkonfigurationen, entnommen aus [1].....	6
Abbildung 3: Sternförmiger Netzwerkaufbau.....	7
Abbildung 4: LSMS System, aus [7].....	8
Abbildung 5: Geometrische Anordnung der LEDs für die Sensorstationen	9
Abbildung 6: Geometrische Anordnung der LEDs für die Hauptstation	10
Abbildung 7: Einzelempfänger mit optischer Sammellinse, aus [8]	11
Abbildung 8: Array von schmalen FOV Empfängern, aus [8]	11
Abbildung 9: Angle Diversity Receiver System, aus [8].....	12
Abbildung 10: Spektrale Emissionswellenlänge, aus [14]	13
Abbildung 11: Prinzip der OOK Modulation.....	14
Abbildung 12: Beispiel einer OOK-CDMA Datenübertragung.....	14
Abbildung 13: Schematische Darstellung des CDMA Verfahrens, aus [3].....	15
Abbildung 14: Autokorrelationsfunktion der Codewörter	19
Abbildung 15: Kreuzkorrelationsfunktion der Codewörter	20
Abbildung 16: Sendeschaltung der a) Sensor- und b) Hauptstation.....	25
Abbildung 17: Aufbau der Sendeschaltung a) Sensor- und b) Hauptstation.....	25
Abbildung 18: Spannungsabfall U_{R_2} am Vorwiderstand R_2	26
Abbildung 19: Spannungsabfall U_{LED} an der LED SFH487	26
Abbildung 20: Spannungsabfall $U_{T_{SAT}}$ am Transistor BC639.....	26
Abbildung 21: Spannung am I/O Port	27
Abbildung 22: Spannungsabfall U_{R_1} am Basiswiderstand.....	27
Abbildung 23: Transistor Basis - Emitter Spannungsabfall.....	27
Abbildung 24: Pulsfolge für einen gesendeten Messwert inklusive <i>Preamble</i>	28
Abbildung 25: a) Photovoltaik Methode; b) Photokonduktive Methode.....	29
Abbildung 26: Blockschaltbild der Empfängerschaltung.....	30
Abbildung 27: Schaltbild der Empfängerschaltung.....	31
Abbildung 28: Aufbau der Empfängerschaltung	32
Abbildung 29: Verstärkung über die Frequenz mit C_{DIODE} von 50 pF, aus [18]	32
Abbildung 30: Verstärkung über die Frequenz mit C_{DIODE} von 100 pF, aus [18]	33
Abbildung 31: Beschaltung des Verstärkers LTC2051	34
Abbildung 32: Empfangenes Signal nach der Verstärkerstufe	35
Abbildung 33: Beschaltung des Filterbausteins	36
Abbildung 34: Bodediagramm des Filters	36
Abbildung 35: Impulsantwort des Filters	37
Abbildung 36: Empfangenes Signal nach dem Filter.....	37
Abbildung 37: Komparator als Schmitt-Trigger beschalten	38
Abbildung 38: Empfangenes Signal nach dem Schmitt-Trigger	39
Abbildung 39: Ausgangspulsweite als Funktion der externen Kapazität, aus [21]	40

Abbildung 40: Empfangenes Signal nach dem Monoflop	40
Abbildung 41: AVR-Net-IO Board	41
Abbildung 42: Schaltbild des AVR NET IO Board	43
Abbildung 43: Prototyp der Hauptstation	44
Abbildung 44: Flussdiagramm der Hauptstationssoftware	45
Abbildung 45: Prototyp der Sensorstation.....	46
Abbildung 46: Testsensor SHT71 von Sensirion, aus [22]	47
Abbildung 47: Messbereich des Sensors, aus [22]	47
Abbildung 48: Typischer Anschluss des Sensors an den Mikrocontroller, aus [22]	48
Abbildung 49: Flussdiagramm der Sensorstationssoftware.....	49
Abbildung 50: Leuchtstärke als Funktion des Vorwärtsstromes; $t_p = 20 \mu\text{s}$, aus [14]..	51
Abbildung 51: Photostrom als Funktion der Lichtleistung, aus [15]	51
Abbildung 52: U_{Trans} über eine Distanz von 7 m	52
Abbildung 53: a) Hauptstation mit Oszilloskop; b) Sensorstation	53
Abbildung 54: U_{Filter} über eine Distanz von 7 m	54

Tabellenverzeichnis

Tabelle 1: Ein Beispiel zum Einsatz der Paritätsprüfung	21
Tabelle 2: Coderate und relative Redundanz.....	22
Tabelle 3: Absoluter und relativer Datendurchsatz.....	23
Tabelle 4: Berechnete Spannung U_{Trans} von 0,5 m bis 7 m	52
Tabelle 5: Gemessene Werte der Filterausgangsspannung U_{Filter}	53

Abkürzungsverzeichnis

CDMA	<i>Code Division Multiple Access</i>
TDMA	<i>Time Division Multiple Access</i>
FOV	<i>Field of View</i>
OOK	<i>On Off Keying</i>
IR	<i>Infrarot</i>
OOC	<i>Optical Orthogonal Code</i>
MAC	<i>Medium Access Control</i>
OPV	<i>Operationsverstärker</i>
ISP	<i>In-System-Programmer</i>
LED	<i>Light Emitting Diode</i>
PIN	<i>Positive Intrinsic Negative</i>
LSMS	<i>Line strip spot-diffusing transmitter system</i>
IS/ID	<i>Interference sense / Interference detection</i>

Anhang A: Software der Haupt- und Sensorstation

Im Anhang A sind die beiden Quellcodes der Haupt- und Sensorstation niedergeschrieben. Für die Implementierung wird das AVR Studio 4 verwendet und die Programmiersprache C herangezogen.

A1: Software der Hauptstation

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include <stdint.h>
#include <stdlib.h>
#ifndef F_CPU
#define F_CPU 16000000UL
#endif
#include <util/delay.h>

int autokorrelation();

uint8_t x, y, zerobit_1, zerobit_2, parity, databitcount_1, databitcount_2, data1[9], data2[9];
uint8_t haupt_code[8], datainput [8], codewords [2] [8], summe_1, summe_2, burst_1, burst_2;
uint8_t daten_1, daten_2, detecthigh, detectlow, parity;
volatile unsigned int j;

void long_delay(uint16_t ms)           //Verzögerung von 1 Mikrosekunde
{for(; ms>0; ms--) _delay_us(1);}

int main(void)
{DDRA = 0x00;
DDRC = 0xFF;
codewords[0][0]=0; codewords[0][1]=0;           //Codewörter der Sensorstationen
codewords[0][2]=1; codewords[0][3]=0;
codewords[0][4]=0; codewords[0][5]=1;
codewords[0][6]=0; codewords[0][7]=0;
codewords[1][0]=0; codewords[1][1]=1;
codewords[1][2]=0; codewords[1][3]=1;
codewords[1][4]=0; codewords[1][5]=0;
```

```

codewords[1][6]=0; codewords[1][7]=0;
haupt_code[0]=0; haupt_code[1]=1;           //Codewort der Hauptstation
haupt_code[2]=0; haupt_code[3]=0;
haupt_code[4]=0; haupt_code[5]=0;
haupt_code[6]=0; haupt_code[7]=1;
OCR0 = 0b00110010;
TIMSK = (1<<OCIE0);
while(1)                                     //Hauptschleife
{if (PINA & 0b00000100)                       //Abfrage ob der Taster gedrückt wurde
    {while (1)
        {if (!(PINA & 0b00000100))
            {TCCR0 = (1<<WGM01) | (1<<CS01);
            sei();
            break;}}}
if (PINA & 0b00001000)                       //Abfrage des Eingangs: High or Low
    {detectlow = 0;
    detecthigh++;
    if (detecthigh >= 5)
        {detecthigh = 0;
        if (daten_1 == 1)
            {zerobit_1++;}
        if (daten_2 == 1)
            {zerobit_2++;}
        for (x = 6; x >= 1; x--)              //Rotation der empfangenen Chips
            {datainput[x+1] = datainput[x];}
        datainput[1] = datainput[0];
        datainput[0] = 1;
        autokorrelation();}
    }else
    {detecthigh = 0;
    detectlow++;
    if (detectlow >= 5)
        {detectlow = 0;
        if (daten_1 == 1)
            {zerobit_1++;}
        if (daten_2 == 1)
            {zerobit_2++;}
        for (x = 6 ; x >= 1; x--)            //Rotation der empfangenen Chips
            {datainput[x+1] = datainput[x];}
        datainput[1] = datainput[0];
        datainput [0] = 0;
        autokorrelation();}}

```

```

if (paritycheck_1 == 1 && burst_1 == 0 && burst_2 == 0) //Paritycheck wenn
                                                    //keine Daten
                                                    //empfangen werden

    {for (x = 0; x <= 8; x++) //Berechnung des Paritätsbits
      {parity = parity + data1[x];}
    if (parity%2 == 0)
      {for (x = 0; x<=7; x++)
        {if (data1[x] == 1) //Speichern der Daten
          {Save_Data1[y] |= 1 << x;}
         else
          {Save_Data1[y] &= 0 << x;}}}
    parity = 0;}
if (paritycheck_2 == 1 && burst_1 == 0 && burst_2 == 0)
    {for (x = 0; x <= 8; x++)
      {parity = parity + data2[x];}
    if (parity%2 == 0)
      {for (x = 0; x<=7; x++)
        {if (data2[x] == 1) //Speichern der Daten
          {Save_Data2[y] |= 1 << x;}
         else
          {Save_Data2[y] &= 0 << x;}}}
    parity = 0;}}

int autokorrelation(void) //Autokorrelationsberechnung
{uint8_t ergebnis1[8], ergebnis2[8];
for (x=0; x <= 7; x++)
    {ergebnis1[x] = codewords[0][x] & datainput[x];
    summe_1 = ergebnis1[x] + summe_1;
    ergebnis2[x] = codewords[1][x] & datainput[x];
    summe_2 = ergebnis2[x] + summe_2;}
if (summe_1 == 2)
    {if (daten_1 == 1) //Sensorstation 1 hat ein Datenbit "1" gesendet
      {zerobit_1 = 0;
      data1[databitcount_1] = 1;
      databitcount_1++;
      if (databitcount_1 == 9)
        {burst_1 = 0;
        daten_1 = 0;
        databitcount_1 = 0;
        paritycheck_1 = 1;}
      }else //Abwarten der 3 Bursts
      {burst_1++;
      summe_1 = 0;

```

```

        if (burst_1 == 3)
            {daten_1 = 1;}}
if (summe_2 == 2)
    {if (daten_2 == 1)           //Sensorstation 2 hat ein Datenbit "1" gesendet
        {zerobit_2 = 0;
        data2[databitcount_2] = 1;
        databitcount_2++;
        if (databitcount_2 == 9)
            {burst_2 = 0;
            daten_2 = 0;
            databitcount_2 = 0;
            paritycheck_2 = 1;}
        }else                   //Abwarten der 3 Bursts
            {burst_2++;
            summe_2 = 0;
            if (burst_2 == 3)
                {daten_2 = 1;}}}
if (zerobit_1 == 8)           //Sensorstation 1 hat ein Datenbit "0" gesendet
    {zerobit_1 = 0;
    data1[databitcount_1] = 0;
    databitcount_1++;
    if (databitcount_1 == 9)
        {burst_1 = 0;
        daten_1 = 0;
        databitcount_1 = 0;}}
if (zerobit_2 == 8)           //Sensorstation 2 hat ein Datenbit "0" gesendet
    {zerobit_2 = 0;
    data2[databitcount_2] = 0;
    databitcount_2++;
    if (databitcount_2 == 9)
        {burst_2 = 0;
        daten_2 = 0;
        databitcount_2 = 0;
        paritycheck_2 = 1;}}
summe_2 = 0;
summe_1 = 0;
return 0;}

ISR (TIMER0_COMP_vect)       //Interrupt Vektor Routine des Timer 0
{if (haupt_code[j]==1)       //Senden des Codewortes
    {PORTC = 0b00000111;
    long_delay(28);
    PORTC = 0x00;}}

```

```
j++;  
if (j >= 8)  
    {j = 0;  
    TCCR0 = (0<<WGM01) | (0<<CS01);  
    cli();}
```

A2: Software der Sensorstation

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include <stdint.h>
#include <stdlib.h>
#ifndef F_CPU
#define F_CPU 16000000UL
#endif
#include <util/delay.h>

int autokorrelation();
int sent_data();
int transmit();

uint8_t x, i, n, j, modus;
uint8_t inputdata[8], codeword[8], summe, code[8];
uint8_t detecthigh, detectlow, parity;
uint8_t data[8], outputdata[9];
uint8_t burst;

void long_delay(uint16_t ms)           //Verzögerung von 1 Mikrosekunde
{for(; ms>0; ms--) _delay_us(1);}

int main(void)
{DDRA = 0x08;
 data[0] = 1; data[1] = 1;           //vordefinierter Messwert
 data[2] = 0; data[3] = 1;
 data[4] = 0; data[5] = 1;
 data[6] = 1; data[7] = 0;
 codeword[0]=1; codeword[1]=0;      //Codewort der Hauptstation
 codeword[2]=0; codeword[3]=0;
 codeword[4]=0; codeword[5]=0;
 codeword[6]=1; codeword[7]=0;
 code[0] = 0; code[1] = 0;          //Codewort dieser Sensorstation
 code[2] = 1; code[3] = 0;
 code[4] = 0; code[5] = 1;
 code[6] = 0; code[7] = 0;

 OCR0 = 0b00110010;                //Timer Einstellungen
 TIMSK = (1<<OCIE0) | (1<<TOIE2);
```

```

sei();
for (x = 0; x <= 7; x++)           //Berechnung des Paritätsbits
    {outputdata[x] = data[x];
    parity = parity + data[x];}
if (parity%2 == 0)
    {outputdata[8] = 0;}
else
    {outputdata[8] = 1;}

while(1)
{if (PINA & 0b00000001)           //Abfrage des Eingangs: High or Low
    {detectlow = 0;
    detecthigh++;
    if (detecthigh >= 11)
        {detecthigh = 0;
        for (x = 6; x >= 1; x--)   //Rotation der empfangenen Chips
            {inputdata[x+1] = inputdata[x];}
        inputdata[1] = inputdata[0];
        inputdata[0] = 1;
        autokorrelation();}
    }else
    {detecthigh = 0;
    detectlow++;
    if (detectlow >= 11)
        {detectlow = 0;
        for (x = 6 ; x >= 1; x--)  //Rotation der empfangenen Chips
            {inputdata[x+1] = inputdata[x];}
        inputdata[1] = inputdata[0];
        inputdata[0] = 0;
        autokorrelation();}}}}

int autokorrelation(void)         //Autokorrelationsberechnung
{uint8_t ergebnis[8];
for (x=0; x <= 7; x++)
    {ergebnis[x] = inputdata[x] & codeword[x];
    summe = ergebnis[x] + summe;}
if (summe == 2)
    {if (modus == 1)               //Messung stoppen
        {modus = 0;
        TCCR0 = (0<<WGM01) | (0<<CS01);
        TCCR2 = (0<<CS22) | (0<<CS20);
        cli();
        }else                       //Messung starten

```

```
        {modus = 1;
          TCCR2 = (1<<CS22) | (1<<CS20);
          sei();}}
summe = 0;
return 0;}

int sent_data(void)                //Zusammenstellung des zu sendenten Pakets
{if (burst == 0)
    {transmit();}
else if (outputdata[i-3] == 1)
    {transmit();}
  n++;
  if (n == 8)
    {n = 0;
     i++;}
  if (i == 3)
    {burst = 1;}
  if (i == 12)
    {i = 0;
     TCCR0 = (0<<WGM01) | (0<<CS01);}
return 0;}

int transmit(void)                //Sendepuls von 12 Mikrosekunden wird erzeugt
{if (code[n] == 1)
    {PORTA = 0x08;
     long_delay(10);
     PORTA = 0x00;}
return 0;}

ISR (TIMER0_COMP_vect)           //Interrupt Vektor Routine des Timer 0
{sent_data();}

ISR (TIMER2_OVF_vect)           //Interrupt Vektor Routine des Timer 2
{j++;
if (j == 61)
    {TCCR0 = (1<<WGM01) | (1<<CS01);
     burst = 0;}}
```