



Brandenburgische  
Technische Universität  
Cottbus - Senftenberg

# Optimierung und Erweiterung des IEEE 802.15.4a UWB-Standards für den Einsatz in Automatisierungssystemen

Von der Fakultät Fakultät 1 MINT - Mathematik, Informatik,  
Physik, Elektro- und Informationstechnik der Brandenburgischen  
Technischen Universität Cottbus - Senftenberg

zur Erlangung des akademischen Grades

**Doktor der Ingenieurwissenschaften (Dr.-Ing.)**

genehmigte Dissertation

vorgelegt von

Dipl.-Informatiker

**Dan KREISER**

geboren am 06.04.1981 in Strausberg

**Gutachter:** Prof. Dr.-Ing. Rolf Kraemer

**Gutachter:** Prof. Dr.-Ing. Heinrich Theodor Vierhaus

**Gutachter:** Prof. Dr.-Ing. Beate Meffert

**Tag der mündlichen Prüfung:** 03. Dezember 2015



# Extended Abstract

---

## **Optimization and Extension of the IEEE 802.15.4a UWB Standard for Usage in Automation Systems**

The use of radio-based systems in the automation industry will increase in the next years. Radio-based systems offer a number of advantages compared to wired systems, such as flexible network topology, mobility, and lower maintenance cost.

However, such systems need to meet high standards of safety, reliability, power consumption, speed and cost. The main requirement of wireless communication in this area is the hard real-time capability with cycle times under 1 ms for a network size of up to 127 sensor nodes and a bidirectional data transmission of up to 48 bytes per sensor node. Another challenge is the integration of these systems into existing infrastructures where coexistence with already present wireless systems is a crucial point.

The aim of this work is to empower IR-UWB to fulfill the demanding parameters of automation control based on the IEEE 802.15.4a standard while not violating regulations for UWB. These conditions have a serious impact on optimizations that can be done making this endeavor even more demanding, but ensuring at the same time that the solutions can be used in real systems.

In this work, both complex theoretical and practical implementation approaches are elaborated. Some of these approaches were simulated in Matlab, implemented in hardware (FPGA) and an ASIC has been produced. In order to fulfill the high requirements a lot of effort was put into the reduction of the main parts of the UWB communication. To achieve this, a new synchronization method as well as new data transmission method have been developed and optimized. This new scheme supports bidirectional communication with several nodes (up to 127) in parallel. A combination of Forward-Error-Correction methods (Reed-Solomon-Code and Hamming-Code) has been investigated to increase the reliability. In addition an innovative method with a small overhead for correcting errors in pulse sequences has been developed as well. Contrary to the Reed-Solomon-Code, this method makes it possible to fulfill the timing requirements.

The result of this work is an optimized transmission concept for UWB. It allows in combination with custom hardware ranging and a reliable, energy-efficient data transmission. The transmission time is less than 1 ms in a network consisting of a master and 127 slaves. If all optimizations proposed in this work are applied, the transmission time for a complete cycle can be reduced by 95% compared to the IEEE 802.15.4a standard. The energy consumption for the whole network can be reduced by 65%.

Some of these improvements have been implemented and for others it is shown how they could be implemented. The impact on the energy consumption of factors like switching time between transmitter and receiver have been analyzed.

In the course of this work, a single-chip solution has been developed which is compliant with IEEE 802.15.4a standard. It is optimized through some proprietary extensions regarding its potential use in the automation industry. This chip does not contain all proposed optimizations but is capable of reducing the transmission time of a cycle by 57%. Even with these additional features the complexity has been considerably reduced so that the used area could be reduced by 43%.

# Zusammenfassung

---

## **Optimierung und Erweiterung des IEEE 802.15.4a UWB-Standards für den Einsatz in Automatisierungssystemen**

Der Einsatz funkbasierter Steuerungs- und Überwachungssysteme wird sich in den nächsten Jahren in der Automatisierungsindustrie wesentlich erhöhen. Funkbasierte Systeme sind flexibel, das heißt skalierbar, leicht rekonfigurierbar und mobil. Außerdem sind Funksysteme kostengünstig, insbesondere bei der Wartung.

Dennoch müssen diese Systeme hohen Anforderungen genügen, wie Sicherheit, Zuverlässigkeit, Energieverbrauch, Geschwindigkeit und Wirtschaftlichkeit. Besonders wichtig für den Einsatz drahtloser Funktechnologien in Automatisierungssystemen ist die harte Echtzeitfähigkeit bei Zykluszeiten im Bereich von unter 1 ms bei Netzwerkgrößen von bis zu 127 Knoten bei einer bidirektionalen Kommunikation mit einer Nutzdatenmenge von bis zu 48 Byte je Sensorknoten. Eine weitere Herausforderung ist die Integration dieser Systeme in bestehende Infrastrukturen, wo Koexistenz mit bereits vorhandenen drahtlosen Systemen ein kritischer Faktor ist.

Ziel dieser Arbeit ist es, die IR-UWB Funktechnik basierend auf dem Standard IEEE 802.15.4a so zu verbessern, dass sie die hohen Anforderungen der Automatisierungsindustrie erfüllt. Dabei soll gewährleistet werden, dass die Abweichungen im Vergleich zum Standard möglichst gering ausfallen und dass die Vorgaben der Regulierungsbehörden nicht verletzt werden. Diese Anforderungen begrenzen die Möglichkeiten der Optimierungen stark, was die Aufgabe noch schwieriger macht, aber zugleich wird damit sichergestellt, dass die Lösungen auch in realen Systemen eingesetzt werden können.

In dieser Arbeit wurden sowohl komplexe theoretische Ansätze als auch Optimierungen der Implementierung ausgearbeitet, in Matlab simuliert und in einem FPGA implementiert. Anschließend wurde ein ASIC gefertigt.

Um die hohen zeitlichen Anforderungen zu erfüllen, wurden die einzelnen Teile des UWB-Frames optimiert. Um dies zu erreichen mussten neue Synchronisations- und Datenübertragungs-Verfahren entwickelt werden. Dieses Verfahren ermöglicht eine parallele bidirektionale Kommunikation mit einer variablen Anzahl an Sensorknoten (bis zu 127). Eine Kombination verschiedener Fehlerkorrekturverfahren (Reed-Solomon-Code und Hamming-Code) wurde untersucht, um die Zuverlässigkeit zu erhöhen. Zusätzlich wurde ein Fehlerkorrekturverfahren für ternäre Impulssequenzen mit geringem Overhead ausgearbeitet. Dieses Verfahren erlaubt eine ausreichend gute Fehlerkorrektur und erlaubt im Gegensatz zu der Fehlerkorrektur mit dem Reed-Solomon-Code das Einhalten der zeitlichen Anforderungen.

Das Hauptergebnis dieser Arbeit ist ein für IR-UWB optimiertes Übertragungskonzept, welches eine zuverlässige und energieeffiziente Datenübertragung von unter 1 ms für

einen Master und 127 Slaves ermöglicht. Teile dieses Verfahrens wurden in Hardware gefertigt und experimentell evaluiert.

Wenn alle vorgeschlagenen Optimierungen verwendet werden, kann die Zykluszeit im Vergleich zum IEEE 802.15.4a Standard, um 95% reduziert werden. Der Energieverbrauch des gesamten Netzwerkes sinkt dabei um 65%.

Einige der vorgeschlagenen Verbesserungen wurden implementiert und für die übrigen wurde gezeigt, wie sie implementiert werden können. Der Einfluss weiterer Faktoren, wie die Umschaltzeit zwischen Sender und Empfänger, und deren Auswirkungen auf den Energieverbrauch wurden untersucht.

Im Rahmen dieser Doktorarbeit entstand eine standardkonforme IR-UWB Einzelchiplösung, die um Eigenschaften erweitert wurde, die vom Standard, abweichen, aber notwendig sind um die Anforderungen der Automatisierungsindustrie zu erfüllen.

Die gefertigte Variante der Einzelchiplösung enthält nicht alle in dieser Arbeit vorgeschlagenen Optimierungen, ist aber dennoch in der Lage die Zykluszeit um 57% zu verkürzen. Obwohl dieser Chip viele zusätzliche Features bietet, wurde die Komplexität und somit auch die benötigte Chipfläche um 43% reduziert.

# Publikationen

---

Während der Anfertigung dieser Doktorarbeit und im Rahmen zweier Projekte TANDEM [1] und KUSZ [2] sind folgende Publikation entstanden.

## Publikationen als Erstautor

- **D. Kreiser, S. Olonbayar**  
*Efficient synchronization method for IR-UWB 802.15.4a non-coherent energy detection receiver*, IOTS 2010 - The 1st IEEE/ACM Internet of Things Symposium Hangzhou, PR China, December 18-20, 2010 [3]
- **D. Kreiser, S. Olonbayar**  
*Design and ASIC Implementation of a UWB-Baseband IEEE 802.15.4a for Non-Coherent Energy Detection*, Proc. Wireless Communications (WCSP 2011) [4]
- **D. Kreiser, S. Olonbayar**  
*Improvements of IEEE 802.15.4a for non-coherent energy detection receiver*, ISSSE 2012, Potsdam Germany, Oct 2012 [5]
- **D. Kreiser, M. Brzozowski, G. Panic, S. Olonbayar, T. Basmer, P. Langendörfer**  
*The all IHP sensor node: highly integrated sensor nodes using IHP components*, Proceedings of the 10th ACM international symposium on Mobility management and wireless access 2012 [6]
- **D. Kreiser, S. Olonbayar**  
*Improvements of IEEE 802.15.4a for Usage in Automation Systems*, Proc. Wireless Communications (ICEAC 2013) [7]
- **D. Kreiser, S. Olonbayar**  
*Multiple Sensor Node Synchronization for Parallel UWB Communication and Ranging*, The 2014 IEEE International Conference on Ultra-Wideband (ICUWB) [8]
- **D. Kreiser, S. Olonbayar**  
*Collision-free Full-duplex UWB Communication Based on the Standard IEEE.802.15.4a*, 2015 IEEE Wireless Communications and Networking Conference (WCNC) [9]
- **D. Kreiser, D. Martynenko, O. Klymenko, G. Fischer**  
*Simple and Efficient Localization Method for IR-UWB Systems based on Two-way Ranging*, 2015 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM) [10]

- **D. Kreiser**

*Simple and Efficient two-way Ranging Method for IEEE802.15.4a*, Proc. 1st URSI Atlantic Radio Science Conference (URSI AT-RASC) 2015 [[11](#)]



---

## Publikationen als Co-Autor

- **S. Olonbayar, D. Kreiser, R. Kraemer**  
*Performance and Design of IR-UWB Transceiver Baseband for Wireless Sensors*, Crazy Proc. ICUWB 2009, Vancouver, Sep. 2009, Canada [12]
- **S. Olonbayar, G. Fischer, D. Kreiser, R. Kraemer**  
*Automatic Gain Controlling in IR-UWB Communications Designed for Wireless Sensors*, 2010 IEEE International Conference on Ultra-Wideband (ICUWB 2010), Nanjing, Sep. 2010, PR China [13]
- **S. Olonbayar, D. Kreiser, R. Kraemer**  
*IEEE 802.15.4a IR-UWB Noncoherent receiver design and performance*, Wireless Congress 2011, Nov. 2011, Munich, Germany [14]
- **S. Olonbayar, D. Kreiser, D. Martynenko, G. Fischer, O. Klymenko, R. Kraemer**  
*Board implementation and its performance for IR-UWB IEEE. 802.15. 4a from multiple ASIC chips*, European Wireless, 2012. EW. 18th European Wireless Conference [15]
- **D. Martynenko, G. Fischer, O. Klymenko, D. Kreiser, S. Olonbayar**  
*High-band ultra-wideband transmitter for IEEE 802.15. 4a standard*, Ultra-Wideband (ICUWB), 2013 IEEE International Conference [16]
- **G. Fischer, D. Martynenko, O. Klymenko, S. Olonbayar, D. Kreiser, J. Digel, M. Masini, M. Grozing, R. Kraemer**  
*IR-UWB single-chip transceiver for high-band operation compliant to IEEE 802.15. 4a*, Ultra-Wideband (ICUWB), 2013 IEEE International Conference [17]
- **S. Olonbayar, D. Kreiser, R. Kraemer**  
*Performance and implementation of a multi-rate IR-UWB baseband transceiver for IEEE802. 15.4 a*, Ultra-Wideband (ICUWB), 2013 IEEE International Conference [18]
- **G. Fischer, M. Martynenko, O. Klymenko, S. Olonbayar, D. Kreiser**  
*Multi-Channel and Multi-Data Rate IR-UWB Single-Chip Transceiver compliant to IEEE 802.15.4a*, The 2014 IEEE International Conference on Ultra-Wideband (ICUWB) [19]
- **S. Olonbayar, D. Kreiser, R. Kraemer**  
*FPGA and ASIC Implementation and Testing of IR-UWB Baseband transceiver for IEEE802.15.4a*, The 2014 IEEE International Conference on Ultra-Wideband (ICUWB) [20]
- **D. Martynenko, D. Kreiser, O. Klymenko, G. Fischer**  
*Evaluation Platform for Verification of an IR-UWB Ranging System*, 12th

Workshop in Positioning, Navigation and Communication (WPNC'15), 2015  
[21]

- **S. Olonbayar, D. Kreiser, R. Kraemer, G. Fischer, D. Martynenko, O. Klymenko**  
*Polarity reception for IR-UWB in wireless fading channel*, European Wireless EW2015, 2015 [22]

# Inhaltsverzeichnis

<b>Extended Abstract</b>	<b>i</b>
<b>Zusammenfassung</b>	<b>iii</b>
<b>Publikationen</b>	<b>v</b>
<b>Inhaltsverzeichnis</b>	<b>ix</b>
<b>Glossar</b>	<b>xi</b>
<b>Liste der Symbole</b>	<b>xvii</b>
<b>1 Einleitung: Motivation und Aufgabenstellung</b>	<b>1</b>
1.1 Einsatz drahtloser Sensornetzwerke in der Automatisierungsindustrie	1
1.2 Aufgabenstellung dieser Arbeit . . . . .	5
1.3 Zusammenfassung . . . . .	12
<b>2 Stand des Wissens und der Technik</b>	<b>15</b>
2.1 Drahtlose Sensornetzwerke in der Automatisierungsindustrie . . . . .	15
2.2 UWB als Funktechnik für drahtlose Sensornetzwerke . . . . .	19
2.3 Abgrenzung zum Stand des Wissens: Zusammenfassung . . . . .	34
<b>3 Optimierung des UWB-Standards</b>	<b>37</b>
3.1 Prinzipielle Umsetzbarkeit der Optimierungen . . . . .	38
3.2 Optimierung der Punkt-zu-Punkt Datenübertragungen . . . . .	48
3.3 Optimierung der Punkt-zu-Multipunkt Datenübertragung . . . . .	64
3.4 Zuverlässigkeit . . . . .	97
3.5 Zusammenfassung . . . . .	110
<b>4 Umsetzung der Optimierungen in Hardware</b>	<b>113</b>
4.1 Vorgehensweise für die Implementierung . . . . .	113
4.2 Modularer Aufbau . . . . .	115
4.3 Verfahren und Methoden . . . . .	121
4.4 ASIC Implementierung . . . . .	137
4.5 FPGA Implementierung . . . . .	154
<b>5 Zusammenfassung und Ausblick</b>	<b>159</b>

---

5.1	Zusammenfassung . . . . .	159
5.2	Ausblick . . . . .	161
<b>A</b>	<b>Anhang A</b>	<b>163</b>
A.1	Eingesetzte Funkstandards in der Automatisierungsindustrie . . . . .	163
A.2	UWB Forschungsprojekte im detail . . . . .	166
A.3	Ermittlung der maximalen Datenrate . . . . .	177
A.4	Berechnung optimaler Impulssequenzen für PREAMBLE-Symbole . . . . .	178
A.5	Optimale Preamble-Sequenzen . . . . .	181
A.6	Optimale Bitraten . . . . .	191
	<b>Literaturverzeichnis</b>	<b>193</b>
	<b>Abbildungsverzeichnis</b>	<b>203</b>
	<b>Tabellenverzeichnis</b>	<b>206</b>

# Glossar

---

## A

ACE	Adaptive Communication Environment
ADC	Analogue to Digital Converter
AES	Advanced Encryption Standard
AFH	Adaptive Frequency-Hopping
ASIC	application-specific integrated circuit / anwendungsspezifische integrierte Schaltung

## B

BAN	Body Area Network
BB	Basisband
BLE	Bluetooth Low Energy
BPSK	Binary Phase-Shift Keying

## C

CARRY4	Fast Carry Logic with Look Ahead
CCA	Clear Channel Assessment
CRC	Cyclic Redundancy Check
CSMA	Carrier Sense Multiple Access
CSS	Chirp Spread Spectrum

## D

DAA	Detect And Avoid
DAC	Digital to Analogue Converter
DCM	Dual Carrier Modulation
DFP	D-Flip-Flop
DS-UWB	Direct Sequence Ultra Wideband
DSP	Digital Signal Processing

## E

ECC	Error-Correcting Code
EDGE	Enhanced Data rates for Global Evolution
EIRP	Equivalent Isotropically Radiated Power
ETSI	European Telecommunications Standards Institute

## F

FCC	Federal Communications Commission
FDS	Frequency-Domain Spreading
FEC	Forward Error Correction
FER	Frame Error Rate
FF	Flip-Flop
FHSS	Frequency Hopping Spread Spectrum
FIFO	First In First Out
FPGA	Field Programmable Gate Array
FSK	Frequency Shift Keying

## G

GPR	Ground Penetrating Radar
GSM	Groupe Spécial Mobile / Global System for Mobile Communications
GUI	Graphical User Interface

## H

HAM	Hamming-Code
HART	Highway Addressable Remote Transducer Protocol
HBC	Human Body Communications
HSDPA	High Speed Downlink Packet Access

## I

INA	In Network Aggregation
INV	Inverter
IR-UWB	Impulse Radio Ultra Wide-Band
ISA	International Federation of the National Standardizing Associations

**K**

KNX	Konnex
KNX-RF	Konnex Radio Frequency

**L**

LDC	Low Duty Cycle
LFSR	Linear Feedback Shift Register
LTE	Long-Term-Evolution
LTE-A	Long-Term-Evolution-Advanced
LUT	Look-Up Table

**M**

MAC	Media Access Control
MBOA	Multiband OFDM Alliance
MB-OFDM	Multiband Orthogonal Frequency Division Multiplexing
MUX	Multiplexer

**N**

NB	Narrowband / Schmalband
----	-------------------------

**O**

OOK	On-Off Keying
OFDM	Orthogonal Frequency Division Multiplexing

**P**

PAN	Personal Area Network
PHR	Physical Header
PHY	Physical Layer
PLL	Phase-Locked Loop
PNO	Profibus Nutzerorganisation

ppm	parts per million
PPM	Pulse-Position Modulation
PRF	Pulse Repetition Frequency / Impulsfolgefrequenz

**Q**

QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying

**R**

RADAR	Radio Detection and Ranging
RS	Reed-Solomon
RX	Receiver



**S**

SCADA	Supervisory Control And Data Acquisition
SER	Symbolfehlerrate / Symbol Error Rate
SFD	Start Frame Delimiter
SIG	Special Interest Group
SPI	Serial Peripheral Interface
SRL	Shift-Register LUT

**T**

TCP	Transport Control Protocol
TDS	Time-Domain Spreading
TER	Telegram Error Rate / Telegrammfehlerrate
TOF	Time-of-Flight
TS	Time-Slot
TX	Transmitter
TDMA	Time Division Multiple Access

**U**

UMTS	Universal Mobile Telecommunications System
UWB	Ultra Wideband / Ultra Breitband

**V**

VDI	Verein Deutscher Ingenieure
-----	-----------------------------

**W**

WAN	Wide Area Network
WBAN	Wide Body Area Network
WISA	Wireless Inteface for Sensors and Actors
WLAN	Wireless Local Area Network
WPAN	Wide Personal Area Network
WSAN	Wireless Sensor Actuator Network
WSN	Wireless Sensor Network



# Liste der Symbole

---

<b>Symbole</b>	<b>Beschreibung</b>
$T_{PS}$	Dauer eines PREAMBLE-Symbols
$T_{BIT}$	Dauer eines Bits
$T_{Bit_{PHR}}$	Dauer eines Bits im PHR-Teil
$T_{Bit_{851\text{ kb/s}}}$	Dauer eines Bits im PSDU-Teil bei einer Datenrate von 851 kb/s
$T_{Bit_{6,81\text{ Mb/s}}}$	Dauer eines Bits im PSDU-Teil bei einer Datenrate von 6,81 Mb/s
$T_{Bit_{27,24\text{ Mb/s}}}$	Dauer eines Bits im PSDU-Teil bei einer Datenrate von 27,24 Mb/s
$T_{Imp}$	Dauer eines Impulses plus der 15 leeren Impulsbreiten innerhalb eines PREAMBLE-Symbols
$T_{SYNC}$	Dauer der Synchronisationsphase
$T_{PREA}$	Dauer des PREAMBLE-Teils
$T_{DATA}$	Dauer der Datenübertragungsphase
$T_{PHR}$	Dauer des PHR-Teils
$T_{UWB-Frame}$	Dauer des UWB-Frames
$T_{TRANS}$	globale Übertragungszeit bzw. Zykluszeit
$T_{SYNC}^k$	Dauer der Synchronisation für eine Gruppe von k Sensorknoten
$T_{DATA}^k$	Dauer der Datenübertragung für eine Gruppe von k Sensorknoten
$T_{G-SYNC}$	globale Dauer der Synchronisation in einem Zyklus
$T_{G-DATA}$	globale Dauer der Datenübertragung in einem Zyklus
$T_{BROADCAST}$	Dauer einer Broadcast-Nachricht vom Master an alle Slaves

Symbole	Beschreibung
$PRFPS$	Impulswiederholrate innerhalb eines Preamble-Symbols
$PRFPREA$	Impulswiederholrate innerhalb des PREAMBLE-Teils
$PRFSYNC$	Impulswiederholrate während der Synchronisationsphase
$PRFDATA$	Impulswiederholrate während der Datenübertragungsphase
$TS$	<i>time-slot</i>
$\#TS$	Anzahl an <i>time-slots</i>
$PS$	PREAMBLE-Symbol
$\#PS$	Anzahl an PREAMBLE-Symbolen
$\#IPS$	Anzahl an Impulsen innerhalb eines PREAMBLE-Symbols
$\#IBurst$	Anzahl an Impulsen innerhalb eines Bursts
$\#IBit$	Anzahl an Impulsen innerhalb eines Bit-Signals
$\#0$	Anzahl an leeren Werten '0' innerhalb einer ternären Sequenz
$\#1$	Anzahl an Einsen '1' innerhalb einer ternären Sequenz
$\#-1$	Anzahl an minus Einsen '-1' innerhalb einer ternären Sequenz
$p^k$	ternäre Sequenz um $k$ Stellen verschoben
$F_k(a, b)$	Korrelationsfunktion zweier ternärer Sequenzen $a$ und $b$ bei einer relativen Verschiebung von $k$ Stellen bzgl. $a$ .
$F'_k(a, b)$	optimierte Korrelationsfunktion
$f(x, y)$	Gewichtungsfunktion der optimierten Korrelationsfunktion
$A_p$	Autokorrelationsgüte
$\#P_{(n,k)}$	Anzahl an möglichen Verteilungen für exakt $k$ nicht-leeren Impulsen innerhalb von $n$ Impulspositionen

Symbole	Beschreibung
$M$	Master
$S_k$	damit wird der $k$ -te Slave aus der gesamten Menge der Slaves bezeichnet
$E_k$	Fehlerwahrscheinlichkeit der Teilübertragung von Master $\rightarrow S_k \rightarrow$ Master
$E_k^m$	Fehlerwahrscheinlichkeit, dass eine Teilübertragung $E_k$ $m$ -mal in Folge fehlschlägt.
$E^{(n,m)}$	Fehlerwahrscheinlichkeit, dass bei einer $m$ -fachen kompletten Übertragung für $n$ Sensoren bei mindestens einem Sensor $S_k$ die Übertragung $m$ -mal fehlschlägt.
$T_{B \rightarrow A}$	Verzögerung von B relativ zu A
$T_P$	Verzögerung welche durch die Verarbeitungszeit (processing time) verursacht wird
$h^{(k)}$	ist die Hopping-Position für das $k$ -te Bit entsprechend dem UWB-Standard
$N_{hop}$	Anzahl an Hopping-Positionen pro Burst-Positionen tsprechend dem UWB-Standard
$TS_k$	ist der $k$ -te <i>time-slot</i> in einem Bit-Signal
$X_k$	ist der Wert des $k$ -ten Bits vom Sensorknoten X
$X_{h^{(k)}}$	ist die Hopping-Position für das $k$ -te Bit vom Sensorknoten X
$TS_{X_k}$	ist der <i>time-slot</i> für $X_k$ das $k$ -te Bit vom Sensorknoten X
$TS_{Diff}$	ist eine <i>time-slot</i> -Differenz zwischen zwei Sensorknoten
$T_D$	Signallaufzeit verursacht durch die Distanz $D$ zwischen zwei Sensorknoten



# Einleitung: Motivation und Aufgabenstellung

---

## 1.1 Einsatz drahtloser Sensornetzwerke in der Automatisierungsindustrie

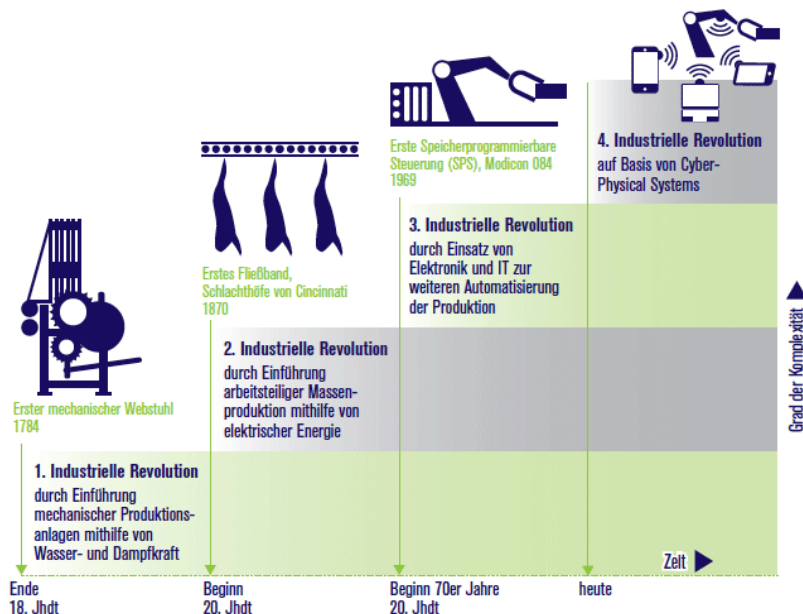
Der Einsatz von drahtlosen Sensornetzwerken (engl. *wireless sensor network*, WSN) als Ersatz für kabelgebundene Sensornetzwerke (engl. *sensor network*, SN) ist eine Folge des technischen Fortschritts und ein wichtiges Forschungsgebiet. Im Privatbereich werden drahtlose Sensornetzwerke bereits eingesetzt. Mehrere Geräte in jedem Haushalt sind vernetzt: Fernseher, Smartphone, Notebook, WLAN-Router usw. Die für diese Vernetzung am häufigsten eingesetzten Funkstandards sind:

- WLAN (engl. *Wireless Local Area Network*) [23]:  
Diese Funktechnologie wird bei lokalen Funknetzwerken eingesetzt. Außerdem können WLAN-Netzwerke über hundert Teilnehmer unterstützen. Abhängig vom genutzten WLAN können Reichweiten von 20 bis 300 m und hohe Datenraten von bis zu 120 Mbit/s erreicht werden.
- BLUETOOTH [24][25]:  
Diese Technologie wurde entwickelt, um energiesparend mit maximal sieben Geräten auf kurzer Distanz, mit Datenraten von 732 Kbit/s bis zu 2,1 Mbit/s zu kommunizieren.
- MOBILFUNK:  
Diese Funktechnologie wurde für die Kommunikation mobiler Geräte, welche die öffentlichen Mobilfunknetze nutzen, entwickelt. Mit Mobilfunknetzen können Geräte über große Entfernungen hinweg kommunizieren. Die für hohe Datenraten verwendeten Mobilfunk-Standards sind
  - UMTS (Universal Mobile Telecommunications System) [26]  
Bei diesem Standard sind Datenraten bis zu 42,2 Mbit/s erreichbar.
  - LTE (Long-Term-Evolution)[27]  
Bei diesem Standard sind Datenraten von über 100 Mbit/s erreichbar.

Eine weitere Vernetzung anderer Haushaltsgeräten wie Kühlschränke, Kaffeemaschinen, Waschmaschinen, Beleuchtung etc. ist in Zukunft vorgesehen. Allerdings gibt es Anwendungsbereiche, in denen drahtlose Systeme zurzeit nur bedingt einsetzbar

sind, zum Beispiel in der Automatisierungsindustrie.

Die Automatisierungsindustrie ist ein wichtiger Wirtschaftssektor in Deutschland und ein wachsender Markt, siehe dazu [28], [29], [30], [31]. Heutzutage sind effizientere Produktionsabläufe bei jedem Hersteller notwendig, um weiter am Markt bestehen zu können und/oder zu expandieren. Der Einsatz der Cyber-Physikalischen Systeme in der Automatisierungsindustrie ist ein Teil der 4. industriellen Revolution. Diese wird als Industrie 4.0 bezeichnet [32], siehe Abbildung 1.1.



**Abb. 1.1:** Industrielle Revolutionen bis heute, entnommen aus [32]:

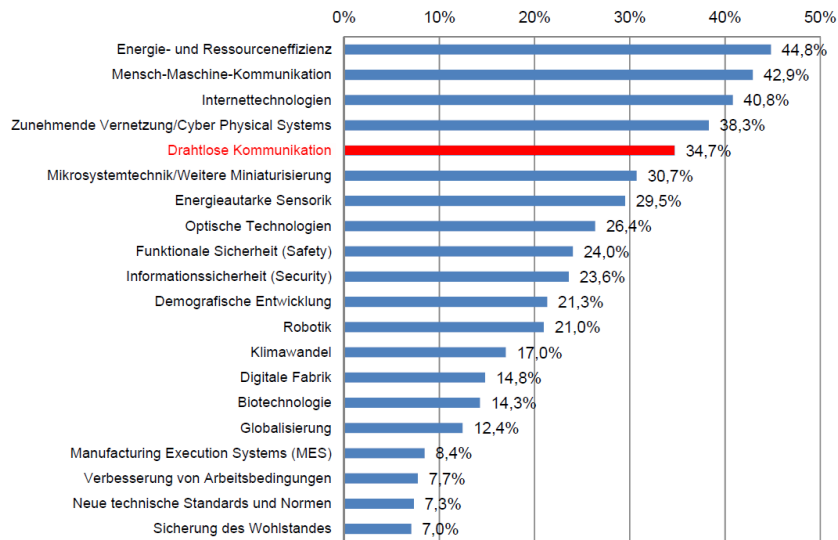
Die 4. industrielle Revolution – Industrie 4.0 – basiert auf dem Einsatz Cyber-Physikalischen Systemen in der Automatisierungsindustrie.

Das Thema "drahtlose Technologien für den Einsatz in der Automatisierung" ist nicht neu. Auf Basis einer repräsentativen Befragung von über 1150 Mitgliedern der VDI<sup>1</sup>/VDE<sup>2</sup>-Gesellschaft Mess- und Automatisierungstechnik [33] [34] kann die Entwicklung der Mess- und Automatisierungstechnik im Hinblick auf ihre Anwendungsfelder in den kommenden Jahren abgeschätzt werden. Das Ergebnis dieser Umfrage ist in Abbildung 1.2 dargestellt. Aus der Umfrage geht hervor, dass drahtlose Kommunikation ein wichtiger Faktor für die Industrie 4.0 ist.

<sup>1</sup>VDI: Verein Deutscher Ingenieure, besteht seit über 150 Jahren und hat über 152.000 Mitglieder. Mehr als 12.000 ehrenamtliche Experten bearbeiten jedes Jahr neueste Erkenntnisse zur Förderung des Technikstandorts Deutschland.

<sup>2</sup>VDE: Verband der Elektrotechnik, Elektronik und Informationstechnik. Der VDE hat 1200 Mitarbeiter. Ihm gehören 36.000 Mitglieder, davon 1300 Unternehmen und 8000 Studenten an.





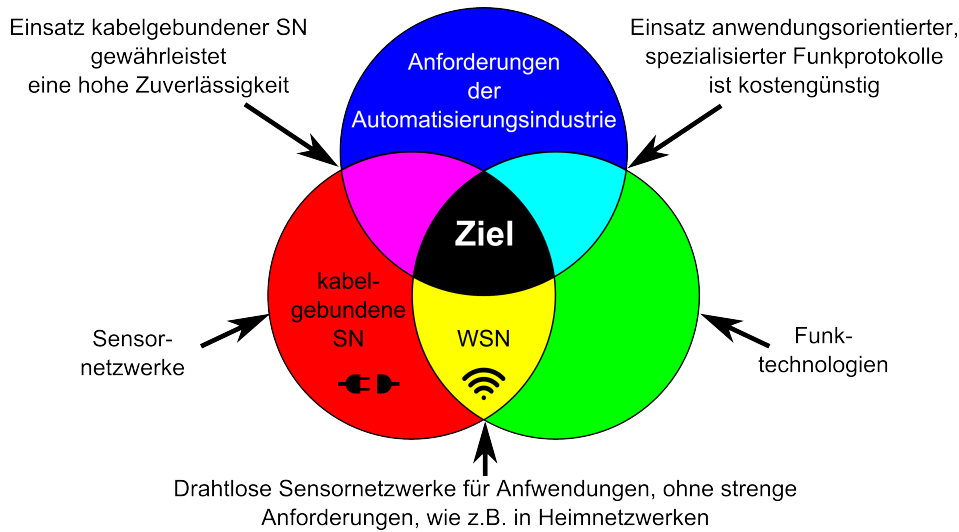
**Abb. 1.2:** Ergebnis einer Mitgliederumfrage, der Gesellschaft für Mess- und Automatisierungstechnik 2012, entnommen aus [35].  
 Nennungen zur Frage "Welche technischen bzw. sozio-ökonomischen Entwicklungen und Erfordernisse werden der Mess- und Automatisierungstechnik in den nächsten drei Jahren die größten Impulse geben?"  
 Fast 35 % der Befragten haben die drahtlose Kommunikation als größten Einflussfaktor für die Automatisierungsindustrie angegeben (siehe rote Markierung).

Die zwei Hauptanforderungen der Automatisierungsindustrie bzgl. Automatisierungssysteme sind ihre zuverlässige Funktionalität und ihre Wirtschaftlichkeit. Abbildung 1.3 zeigt, dass diese zwei Anforderungen momentan am besten mit unterschiedlichen Arten von Sensornetzwerken erfüllt werden können. Die zurzeit in der Automatisierungsindustrie eingesetzten kabelgebundenen Sensornetzwerke können eine zuverlässigere Datenübertragung als drahtlose Sensornetzwerke gewährleisten. Die Paketfehlerrate, bei kabelgebundenen Sensornetzwerke, liegt bei  $10^{-12}$ . Andererseits sind kabelgebundene Sensornetzwerke im Vergleich zu drahtlosen Sensornetzwerke teurer. Der Einsatz der drahtlosen Sensornetzwerke reduziert zwar die Kosten, aber die Zuverlässigkeit ist um mehrere Größenordnungen geringer.



**Abb. 1.3:** Zuverlässigkeit und Wirtschaftlichkeit sind die Hauptanforderungen der Automatisierungsindustrie.

Werden die Anforderungen der Automatisierungsindustrie, die heutzutage existierenden Funktechnologien und die Sensornetzwerke als Mengen dargestellt, befindet sich die notwendige Lösung in der gemeinsamen Schnittmenge, wie es in Abbildung 1.4 gezeigt ist (siehe schwarz markierte Schnittmenge).



**Abb. 1.4:** Überschneidung der Anforderungen aus der Automatisierungsindustrie mit kabelgebundenen und drahtlosen SN und Funktechnologien.

Der Teil der Sensornetzwerke, welcher sich nicht mit den Funktechnologien überschneidet stellt die kabelgebundenen Sensornetzwerke dar (siehe rot markierter Bereich).

Die gemeinsame Schnittmenge der Sensornetzwerke und Funktechnologien repräsentiert die drahtlosen Sensornetzwerke (gelb und schwarz markierte Bereiche).

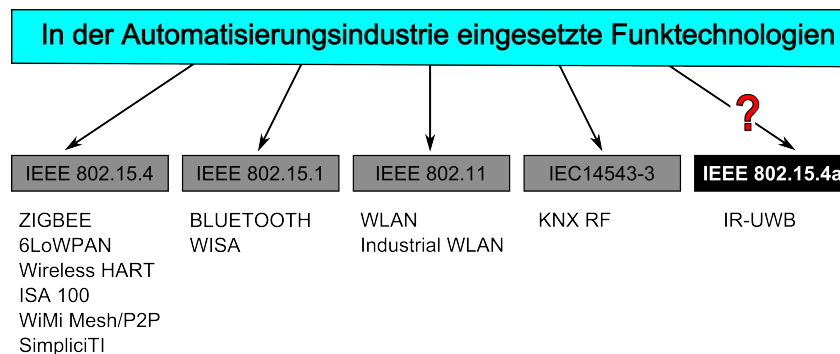
Der Einsatz von kabelgebundenen SN in der Automatisierungsindustrie gewährleistet eine zuverlässige Funktionalität der Automatisierungssysteme. Der Einsatz von WSN ist kostengünstig aufgrund von reduzierter Verkabelung und den damit verbundenen Wartungskosten.

Heutzutage werden drahtlose Sensornetzwerke mit verschiedenen Funktechnologien erfolgreich in einzelnen Bereichen der Automatisierungsindustrie bereits verwendet. Wegen nicht hinreichender Zuverlässigkeit sind diese Lösungen kein vollwertiger Ersatz für kabelgebundene Sensornetzwerke und können nur für nicht kritische Anwendungszwecke eingesetzt werden [36].

Jede Funktechnologie bzw. jeder Funkstandard weist Vor- und Nachteile für Anwendungen der Automatisierungsindustrie auf. Einige Funkstandards ermöglichen hohe Datenraten, sind extrem energiesparend, echtzeitfähig oder robust. Es gibt aber keinen Funkstandard, der alle Anforderungen der Automatisierungsindustrie gleichzeitig erfüllt. Bisher wurde kein Standard explizit für die Automatisierungsindustrie entwickelt. Alle derzeit in der Automatisierungsindustrie angewandten Standards bzw. Protokolle basieren auf existierenden Standards, welche nicht vorrangig für den Einsatz in der Automatisierung konzipiert wurden.

Abbildung 1.5 gibt eine Übersicht der am häufigsten eingesetzten Funk-Standards und Funk-Protokolle, die für die Kommunikation in SN für Automatisierungssysteme bereits verwendet werden. Die verwendeten Funktechnologien – die Standards IEEE 802.15.1, IEEE 802.15.4, IEEE 802.11 und IEC14543-3 – sind in Abbildung 1.5 grau markiert. Die IR-UWB (engl. *Impuls Radio Ultra Wideband*, IR-UWB) Funktechnik nach IEEE 802.15.4a [37] ist ein weiterer Kandidat für die Kommunikation von Sensorknoten in Automatisierungssystemen (siehe Abbildung 1.5, schwarz markierter Block). Die Einsetzbarkeit von IR-UWB für die Automatisierungsindustrie wurde bisher nicht evaluiert. Eine Beschreibung der Funkstandards aus Abbildung 1.5

befindet sich im Anhang A.1.



**Abb. 1.5:** Bereits im Einsatz befindliche Funktechnologien in der Automatisierungsindustrie. Die Einsetzbarkeit von IR-UWB für drahtlose SN wurde noch nicht untersucht.

Der Schwerpunkt dieser Arbeit ist es, die mögliche Verwendung einer impulsbasierten Ultra Breitband Funktechnologie, basierend auf dem Standard IEEE 802.15.4a [37], für die Automatisierungsindustrie zu untersuchen. Des Weiteren sollen Optimierungen des Standards vorgeschlagen werden, welche den Einsatz dieser Funktechnik ermöglichen. Diese Optimierungen sollen sich nah am vorhandenen Standard orientieren, um es Herstellern zu erleichtern, sie in bisherige Implementierungen zu integrieren und auch, um die Wahrscheinlichkeit zu erhöhen, dass diese Optimierungen in zukünftige Versionen des Standards berücksichtigt bzw. integriert werden.

Diese Arbeit konzentriert sich auf die Anforderungen der Automatisierungsindustrie bzgl. Echtzeitfähigkeit und geringen Zykluszeiten, ohne jedoch die von den Regulierungsbehörden vorgegebenen Beschränkungen für die UWB-Technologie zu verletzen.

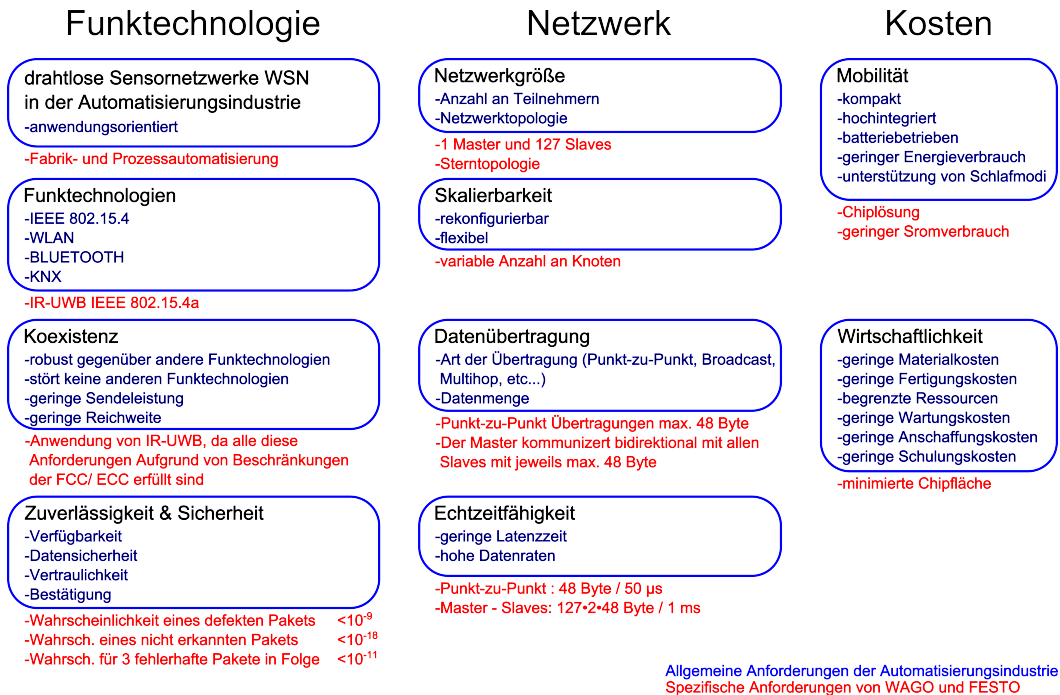
## 1.2 Aufgabenstellung dieser Arbeit

Das Ziel dieser Doktorarbeit ist es, die Optimierung des IEEE Standards 802.15.4a hinsichtlich des Einsatzes in Automatisierungssystemen durchzuführen. In diesem Abschnitt wird die Aufgabenstellung detailliert beschrieben. Dafür werden in diesem Kapitel die Anforderungen drahtloser Sensornetze für die Automatisierungsindustrie beschrieben. Die für diese Doktorarbeit relevanten Anforderungen werden genau definiert.

### 1.2.1 Anforderungen der Automatisierungsindustrie an IR-UWB

Die Anforderungen der Automatisierungsindustrie an drahtlose Kommunikationssysteme im industriellen Umfeld sind vielfältig. Die in dieser Doktorarbeit beschriebenen Untersuchungen wurden im Rahmen der Projekte TANDEM [1] und KUSZ [2] durchgeführt. Die Projektpartner aus der Automatisierungsindustrie – FESTO [38] und WAGO [39] – haben ihre Anforderungen zusammengefasst. Abbildung

1.6 stellt zusammengefasst sowohl die allgemeinen Anforderungen der Automatisierungsindustrie, als auch die spezifischen Anforderungen der Projektpartner FESTO und WAGO dar. Diese Arbeit konzentriert sich auf die Erfüllung der spezifischen Anforderungen.



**Abb. 1.6:** Anforderungen der Automatisierungsindustrie.

In den blauen Boxen sind die allgemeinen Anforderungen dargestellt.

Die spezifischen Anforderungen der Projektpartner aus der Automatisierungsindustrie sind rot markiert.

Es folgt nun eine detailliertere Beschreibung der Anforderungen aus Abbildung 1.6.

### 1.2.1.1 Einsatz von IR-UWB als Funktechnologie für die Automatisierungsindustrie

Wesentlicher Vorteil der Kommunikation mittels IR-UWB ist die Robustheit gegenüber Mehrwegeausbreitungen (engl. *multipath fading*). Diese Eigenschaft resultiert aus der Verwendung sehr kurzer Impulse. Die Mehrwegeausbreitung ist im industriellen Umfeld ein relevanter Störfaktor, aufgrund der maschinellen Umgebung mit vielen metallischen Flächen.

In einem industriellen Umfeld mit verschiedenen Funktechnologien reicht es nicht aus, robust gegenüber andere Funksysteme zu sein. Es ist auch notwendig, dass die vorhandenen Funksysteme in ihrem regulären Arbeitsverlauf nicht beeinträchtigt werden.

Die am häufigsten verwendeten Funktechnologien in der Automatisierungsindustrie sind die folgende:

- Bluetooth nach Standard IEEE 802.15.1
- WirelessHART und ZigBee nach Standard IEEE 802.15.4
- Auf dem Standard IEEE 802.11 basierende Technologien wie z.B. 2,45 GHz und 5 GHz WLAN

Die Spektralmaske für UWB-Systeme wurde so gewählt, dass eine Störung mit anderen Funksystemen verhindert wird. Bei einer UWB-Übertragung wird die gesamte Sendeleistung von wenigen Milliwatt auf einem derart großen Frequenzbereich verteilt, dass für den Funkbetrieb schmalbandiger Übertragungsverfahren keine Störungen zu erwarten sind. Störungen zwischen verschiedenen gleichzeitig operierenden UWB-Systemen werden im Standard IEEE 802.15.4a[37] mittels Zeitsprungverfahren<sup>3</sup>(engl. *time-hopping*) verhindert.

Aufgrund der guten Koexistenzeigenschaften wurde die IR-UWB Funktechnologie basierend auf dem Standard IEEE 802.15.4a von den Projektpartnern aus der Automatisierungsindustrie vorgegeben. Außer der Koexistenzeigenschaft wurden spezifische Anforderungen bzgl. Zuverlässigkeit und Sicherheit gestellt.

Bei der Zuverlässigkeit müssen mehrere Aspekte betrachtet werden: die Ausfallsicherheit, welche durch die Hardware bedingt ist; die Datensicherheit, welche durch die Signalverarbeitung und die eingesetzten Modulations-, Kodierungs- und Verschlüsselungsverfahren bedingt ist. Für Automatisierungssysteme ist der wichtigste Zuverlässigkeitsparameter die Fehlerrate bei der Datenübertragung. Bei kabelgebundenen Systemen liegt die Fehlerrate bei  $10^{-12}$ . Das bedeutet, dass die Fehlerwahrscheinlichkeit einer Übertragung bei  $10^{-12}$  liegt, d.h. dass nur eine aus  $10^{12}$  Übertragungen fehlerhaft ist. Bei drahtlosen Systemen in der Automatisierungsindustrie wird eine Zuverlässigkeit von mindestens  $10^{-9}$  gefordert.

In dem Referenzsystem des KUSZ-Projektes ist ein Fehler als ein nicht empfangenes Paket oder ein falsch empfangenes Paket definiert. Es ist gestattet, die Nachricht bis zu dreimal zu übertragen. Das Auftreten von drei nacheinander defekt empfangenen Datenpaketen muss erkannt werden und kann unter bestimmten Umständen zu einer kompletten Not-Aus bzw. Not-Stopp Situation führen. Die Vorgaben sind wie folgt:

- Die Wahrscheinlichkeit des Auftretens eines nicht erkannten Übertragungsfehlers:  $< 10^{-18}$

---

<sup>3</sup>*time-hopping* ist ein Modulationsverfahren, bei dem die einzelnen Bits bzw. Impulse eines Teilnehmers nur in kurzen Zeitabschnitten (engl. *time-slots*) gesendet werden. Aufgrund des variablen Abstandes zwischen den Impulsen werden Kollisionen erheblich verringert. Dadurch können mehrere Teilnehmer gleichzeitig aktiv sein.

- Die Wahrscheinlichkeit des Empfangens eines defekten Datenpakets:  $< 10^{-9}$
- Die Wahrscheinlichkeit, dass drei Datenpakete in Folge defekt sind:  $< 10^{-11}$

In Bezug auf Sicherheit gilt, dass ein System sicher ist, wenn es nicht unentdeckt manipuliert werden kann. Drahtlose Systeme sind schwer abschirmbar. Fremde Teilnehmer könnten in das Netzwerk eindringen und großen Schaden verursachen. Dieses Risiko wird durch das Einsetzen geeigneter Verschlüsselungsverfahren reduziert. Ein Vorteil von UWB ist die geringe Reichweite von 10 bis 30 m. Dadurch müssen fremde Teilnehmer bzw. Störer einen Abstand von mindestens 30 m zum zu korrumpierenden Knoten einhalten. Dies ist auf einem abgesperrten industriellen Gelände, wo nur bestimmte Personen zugangsberechtigt sind, als unwahrscheinlich von Projektpartner definiert. Somit wird der Aspekt der Datensicherheit im Rahmen dieser Arbeit nicht verfolgt.

### 1.2.1.2 Netzwerkanforderungen der Automatisierungsindustrie

In der Automatisierungsindustrie müssen Netzwerke flexibel sein. Produktionsverfahren sind nicht statisch und werden ständig angepasst. Neue Produktionsschritte und Anlagen können hinzugefügt oder auch aufgrund von Optimierungen entfernt werden. So ist die Anzahl an Netzwerkteilnehmern variabel.

In dieser Arbeit wird von einem Sensornetzwerk ausgegangen, welches maximal aus 128 Teilnehmern besteht, d.h. ein Master-Knoten und bis zu 127 Slave-Knoten. Die Datenkommunikation zwischen Slave und Master ist bidirektional. Es handelt sich bei diesem Netzwerk um eine Sterntopologie (siehe Abbildung 1.7). Entsprechend der Erfahrungen der industriellen Partner FESTO und WAGO bestehen 80% der Sensornetzwerke heutzutage aus maximal 10 Teilnehmern. Detaillierter ist die Verteilung wie folgt:

- 2 Teilnehmer: 25 %
- 2-5 Teilnehmer: 25 %
- 5-10 Teilnehmer: 30 %
- 11-50 Teilnehmer: 15 %
- 51-127 Teilnehmer: 5 %

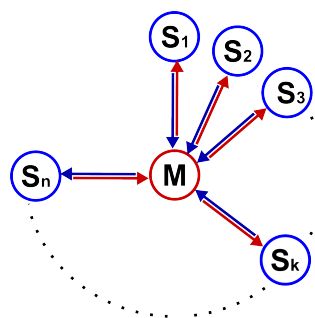


Abb. 1.7: Sterntopologie und bidirektionale Datenübertragung.

Zusätzlich zur Netzwerkgröße wurden auch sehr genaue zeitliche Anforderungen bzgl. der Datenkommunikation von den industriellen Projektpartner festgelegt, um eine harte Echtzeit gewährleisten zu können:

1. Übertragungszeit maximal  $50 \mu s$   
Diese Anforderung bedeutet: Bei einer Punkt-zu-Punkt Verbindung müssen die Nutzdaten von 3 bis 48 Byte innerhalb einer Zeitspanne von  $50 \mu s$  übertragen werden.
2. Zykluszeit maximal  $1 ms$   
Diese Anforderung bedeutet: Der Master muss in der Lage sein, alle Daten mit allen 127 Teilnehmern des Netzwerkes innerhalb  $1 ms$  auszutauschen. Das ist die Zykluszeit. Dies entspricht einer effektiven Nutzdatenrate im Worst Case<sup>4</sup> von:

$$Nutzdatenrate = \frac{2 \cdot 127 \cdot 48 \cdot 8 \text{ bits}}{1 \text{ ms}} = 97,536 \text{ Mb/s}$$

Die von den industriellen Projektpartner geforderte Nutzdatenrate ist 4 mal höher als die im UWB-Standard IEEE 802.15.4a vorgesehene maximale Datenrate von  $27,24 \text{ Mb/s}$ .

Eine Bestätigung (engl. *acknowledge*) des Empfangs eines Pakets, wurde von den industriellen Projektpartnern als nicht erforderlich definiert, solange die Kriterien bezüglich der Zuverlässigkeit nicht beeinträchtigt werden. Aus diesem Grund konzentriert sich diese Doktorarbeit auf die Anforderungen bzgl. Zuverlässigkeit.

### 1.2.1.3 Wirtschaftliche Anforderungen der Automatisierungsindustrie

In der Automatisierungsindustrie werden alle Eingriffe in bestehende Infrastrukturen auf ihre Wirtschaftlichkeit untersucht. Es wird gefordert, dass der Einsatz von UWB-Systemen zur Ersetzung kabelgebundener Systeme sich langfristig wirtschaftlich rentiert. Genaue Zahlen und Fristen wurden von den Projektpartnern nicht angegeben.

Anfallende Kosten für die Automatisierungsindustrie sind z.B. Anschaffungskosten, Energiekosten, Wartungskosten und Schulungskosten. Durch die Umstellung des Produktionsablaufs können auch Ausfallkosten entstehen.

Die Anwendung von drahtlosen Systemen anstatt kabelgebundener Systeme, kann die gesamten Kosten wesentlich reduzieren.

Kabel können eingespart werden und die Gestaltung des Produktionsablaufes wird flexibler. Fehler werden leichter gefunden und schneller behoben, z.B. im Falle eines Kabelbruchs, da keine neuen Kabel verlegt werden. Aufgrund der einfacheren Wartung drahtloser Systeme werden auch die Wartungskosten reduziert. Eine schnelle Behebung von Fehlern hat kürzere Produktionsunterbrechungen und

---

<sup>4</sup>Als Worst-Case wird die bidirektionale Übertragung, zwischen dem Master und 127 Slaves, von je 48 Byte Nutzdaten bezeichnet.

reduzierte Ausfallkosten zur Folge.

Systeme mit hohen Reichweiten sind kostengünstiger, da mit weniger Knoten größere Bereiche abgedeckt werden können. Dies reduziert die Anschaffungskosten. Allerdings liegen in einem industriellen Umfeld die Abstände zwischen den Maschinen im Bereich von einigen Metern. Der Abstand zum Master-Knoten oder Senke kann aber durchaus höher sein. Hier müssen geeignete Routing-Protokolle oder mehrere Master-Knoten eingesetzt werden, um größere Distanzen zu überbrücken. Außerdem gilt, je kleiner die Reichweite ist, desto weniger Störungen werden produziert. Aus Sicht der Datensicherheit sind Systeme mit geringer Reichweite günstiger. Denn um einen erfolgreichen Angriff durchzuführen, muss sich ein Angreifer näher am Sensorknoten befinden. Die Anforderungen von FESTO und WAGO bzgl. der Reichweite liegen im Bereich von 10 bis 20 Metern.

Um eine höchstmögliche Flexibilität von drahtlosen Systemen gewährleisten zu können, müssen diese kompakt und energiesparend sein. Eine kompakte Bauweise kann dadurch erreicht werden, dass der Sensorknoten hoch-integriert ist. Eine bevorzugte Lösung ist, wenn der Sensorknoten als Chip implementiert wird. Dies reduziert zugleich die Materialkosten. Eine der Aufgaben dieser Doktorarbeit ist die Entwicklung eines Chips, welcher das UWB-Basisband und das analoge Frontend beinhaltet. Die Fertigungskosten bei einem Chip können gesenkt werden, indem der Chip in einer günstigen Technologie gefertigt, und die Chipfläche minimiert wird.

In Fabrikhallen oder generell im industriellen Bereich sind Energiequellen für feste Sensorknoten ausreichend vorhanden und unproblematisch. Mobile oder schwer zugängliche Sensorknoten sind batteriebetrieben. Eine alternative Energieversorgung wie Solarenergie, Batterie oder Energy Harvesting<sup>5</sup> ist auch möglich. In Abbildung 1.8 sind verschiedene Arten von drahtloser Energieversorgung und deren mögliche Einsatzgebiete aufgelistet.

---

<sup>5</sup>Als Energy Harvesting bezeichnet man die passive Erzeugung elektrischer Energie aus der Umgebung. Mögliche Energiequellen sind die Umgebungstemperatur, Vibrationen oder Luftströmungen.



Concept	Energy Scavenging	Energy Storage	Energy Distribution
Technologies	++ Photovoltaic + Temperature gradient - Human power - Wind / air flow -- Pressure variations - Vibrations	++ Batteries + Micro-batteries - Micro-fuel cells ++ Ultra-capacitors - Micro-heat engines -- Radioactive power sources	++ Electromagnetic coupling + RF radiation - Wired power grid - Acoustic waves - Optical
Power Levels	10 $\mu$ – 15 m W/cm <sup>3</sup>	50 – 3 500 Ws/cm <sup>3</sup>	$\mu$ W – kW (m – mm)
Deficiencies	Reliability	Maintenance	Electric field radiation Power level vs. range
Industrial Applications	N/A	Portable/mobile user interface	Cell automation power supply

Abb. 1.8: Übersicht drahtloser Energieversorgungen für WSN (entnommen aus [40]).

Eine generelle Vorgabe zum Leistungsverbrauch wie  $X \mu W$  im Schlummermodus und  $Y \mu W$  pro übertragenem Bit wurde von den industriellen Projektpartner nicht definiert. Von der Industrie werden mehrere Jahre Betriebslaufzeit mit nur einer Batterie gefordert. Eines der Ziele dieser Doktorarbeit ist es den Energieverbrauch zu reduzieren.

Im Projekt KUSZ [2] waren die Anforderungen weniger kritisch.

- Spannungsversorgung: 1,8 V oder 3,3 V
- Maximale Leistungsaufnahme von 500 mW im Betrieb

Der Energieverbrauch bei drahtlosen SN hängt im Wesentlichen von drei Faktoren ab:

1. Energieverbrauch pro erfolgreich übertragenem Bit
2. Energieverbrauch im Schlummermodus
3. Hardwareimplementierung

Die Gewichtung der ersten beiden Faktoren hängt vom Anwendungsfall ab. Ist ein System vorwiegend aktiv und werden relativ viele Daten übertragen, so ist der 1. Faktor ausschlaggebend für den Gesamtverbrauch.

Ist ein System allerdings vorwiegend inaktiv und werden nur geringe Datenmengen übertragen, so ist der 2. Faktor höher zu bewerten. In Automatisierungssystemen ist dieser Fall der am häufigsten zu erwartende Fall.

Der 3. Faktor, die Hardwareimplementierung, ist nicht zu vernachlässigen, aber schwer zu ermitteln, da dieser erst nach einer erfolgreichen Implementierung bewertet werden kann. Energiesparende Implementierungstechniken werden in Kapitel 4 betrachtet.

Es gibt für ein und dieselbe Aufgabe verschiedene Herangehensweisen bei der Implementierung.

- synchrone Implementierung
- asynchrone Implementierung
- parallelisierte Verarbeitung (engl. *pipelining*)
- serielle Verarbeitung
- eine Kombination aller Varianten

Jede dieser Implementierungsvarianten hat ihre Vor- und Nachteile. Beispielsweise ist eine serielle Verarbeitung der Daten langsamer als eine parallele Verarbeitung. Eine serielle Verarbeitung benötigt aber weniger Chipfläche. Details werden im Kapitel 4 erläutert.

### 1.3 Zusammenfassung

Ziel dieser Doktorarbeit ist es, die in diesem Kapitel genannten relevanten Anforderungen der Automatisierungsindustrie zu erfüllen. Besonderer Schwerpunkt wird dabei auf Echtzeitfähigkeit und Zuverlässigkeit gelegt. Um die Echtzeitfähigkeit gewährleisten zu können, muss immer vom Worst Case ausgegangen werden, d.h. von einem Netzwerk mit 128 Teilnehmern mit einer bidirektionalen Datenübertragung von jeweils 48 Byte von Master zu Slave. Um dies zu erreichen, muss die im Standard IEEE 802.15.4a vorgesehene maximale Datenrate von 27,24 Mb/s auf mindestens 97,536 Mb/s erhöht werden.

Die Doktorarbeit ist in 5 Kapitel aufgeteilt. Die restlichen 4 Kapitel dieser Arbeit sind:

**Kapitel 2** gibt einen Überblick über den aktuellen wissenschaftlichen und technischen Stand im Bereich der drahtlosen Automatisierungstechnik und im Bereich der IR-UWB-Forschung. Außerdem wird hier eine kurze Einführung in die IR-UWB-Technik gegeben.

**Kapitel 3** untersucht die Machbarkeit bzgl. der Anforderungen aus Kapitel 2. Es werden verschiedene Übertragungskonzepte ausgearbeitet, verglichen und bewertet. Das Ziel, die Zykluszeit in einem Netzwerk von 128 Sensorknoten auf unter 1 ms zu reduzieren, wurde erreicht. Die Zykluszeit wurde um mehr als **95%** im Vergleich zum herkömmlichen IR-UWB-Standard IEEE 802.15.4a reduziert.

**Kapitel 4** gibt Einblick in die Hardwareimplementierungen einzelner Blöcke des Gesamtsystems. Es wird gezeigt wie die Hardwarekomplexität, der Energieverbrauch und die Chip-Fläche reduziert werden können. Der Energieverbrauch konnte um mehr als **65%** reduziert werden.

**Kapitel 5** fasst die wichtigsten Ergebnisse zusammen. Diese Ergebnisse stellen den wissenschaftlichen Beitrag dieser Arbeit dar. Außerdem erfolgt ein Ausblick auf mögliche Erweiterungen, basierend auf dieser Arbeit.



# Stand des Wissens und der Technik

---

## Inhaltsverzeichnis

2.1	Drahtlose Sensornetzwerke in der Automatisierungsindustrie . . . . .	<b>15</b>
2.2	UWB als Funktechnik für drahtlose Sensornetzwerke . . . . .	<b>19</b>
2.2.1	Definition von UWB . . . . .	<b>19</b>
2.2.2	IR-UWB Herkunft . . . . .	<b>19</b>
2.2.3	UWB Regulierungen . . . . .	<b>21</b>
2.2.4	IR-UWB Standardisierung . . . . .	<b>23</b>
2.2.5	Stand der Forschung im Bereich von IR-UWB . . . . .	<b>25</b>
2.2.6	Einsatz von UWB in kommerziellen Produkten . . . . .	<b>34</b>
2.3	Abgrenzung zum Stand des Wissens: Zusammenfassung . . . . .	<b>34</b>

---

In diesem Abschnitt wird der aktuelle Stand bzgl. der Anwendung von drahtlosen Sensornetzwerken (engl. *Wireless-Sensor-Network*, WSN) für die Automatisierungsindustrie vorgestellt. Dafür werden die in der Automatisierungsindustrie eingesetzten WSN einschließlich ihrer Aufteilung in Anwendungs- und Nutzklassen vorgestellt. Außerdem wird die UWB-Funktechnologie, die als Basis für die drahtlose Kommunikation der Sensorknoten in dieser Arbeit verwendet wird, dargestellt: Herkunft, Standardisierung, Stand des Wissens und der Forschung und der Einsatz in kommerziellen Produkten.

## 2.1 Drahtlose Sensornetzwerke in der Automatisierungsindustrie

Die Automatisierungsindustrie stellt hohe Anforderungen an Sensornetzwerke. Die wichtigsten Forderungen sind Zuverlässigkeit und Wirtschaftlichkeit. Kabelgebundene Sensornetzwerke erfüllen die Anforderungen bezüglich Zuverlässigkeit und Sicherheit eher als drahtlose Sensornetzwerke, da Kabelverbindungen gegenüber äußeren Einflüssen abgeschirmt sind. Drahtlose Verbindungen reagieren anfälliger auf äußere Einflüsse. Nachteile von kabelgebundenen Sensornetzwerken sind die benötigte Infrastruktur für die Verkabelung, schwierige Skalierbarkeit, keine Mobilität, Gefährdungspotenzial für Mitarbeiter, wie z.B. Stolpergefahr, eine komplizierte Fehlersuche bzw.

Fehlerbehebung im Falle einer Störung, wie z.B. Kabelbruch, die daraus resultierenden Wartungskosten und längeren Ausfallzeiten.

Da die Gewährleistung der Zuverlässigkeit immer mit zusätzlichen Maßnahmen/Kosten verbunden ist, sind in der Automatisierungsindustrie die Anforderungen bzgl. Zuverlässigkeit stark anwendungsorientiert. D.h. unterschiedliche Prozess- bzw. Fertigungsschritte können unterschiedliche Anforderungen an die Zuverlässigkeit haben. Aus diesem Grund wurden für den Einsatz von drahtlosen Sensornetzwerken in der Automatisierungsindustrie die dortigen Einsatzgebiete in zwei Kategorien unterteilt. Man unterscheidet zwischen Anwendungsklassen und Nutzklassen. Es gibt 3 Anwendungsklassen und 6 Nutzklassen. Die Anwendungsklassen sind:

- "Sicherheit" (engl. *safety*)  
Zu dieser höchsten und kritischsten Anwendungsklasse gehören zeitkritische Anwendungen, welche für Sicherheit und Prozesseingriffe zuständig sind oder mit anderen Systemen aus der Anwendungsklasse "Sicherheit" interagieren.
- "Prozesskontrolle" (engl. *control*)  
Zu dieser mittleren Anwendungsklasse gehören zeitkritische, deterministische Anwendungen, welche hohe Anforderungen an Verfügbarkeit und Zuverlässigkeit stellen.
- "Überwachung" (engl. *monitoring*)  
Hierzu gehören weniger kritische Anwendungen, welche dazu dienen, Prozesse zu überwachen oder bestimmte Messwerte anzuzeigen.

Die existierenden Nutzklassen sind von 0 bis 5 nummeriert. Je kleiner die Nummer der Nutzkategorie ist, desto höher sind die zeitlichen Anforderungen bzgl. Datenübertragung, Latenz, Antwortzeit, Zykluszeit und Echtzeitfähigkeit. In dieser Arbeit werden die Nutzklassen 0 und 1 betrachtet, d.h. die zwei Klassen mit den höchsten zeitlichen Anforderungen:

- Die Nutzkategorie 0 definiert Notfallmaßnahmen, welche sicherheitsrelevant sind; sowohl für die Fabrik, als auch für das Personal.
- Bei der Nutzkategorie 1 werden geschlossene Regelkreise mit kritischen Zeitanforderungen gesteuert. Dies beinhaltet z.B. Steuerungen für Motoren, Achsen, Durchflüsse oder Drücke.

Für diese beiden Nutzklassen werden normalerweise kabelgebundene Sensornetzwerke verwendet. Bei Nutzklassen mit höheren Nummern, von 2 bis 5 und für Monitoring-Anwendungen, wo die Anforderungen bzgl. der Zuverlässigkeit geringer sind, werden bereits drahtlose Sensornetzwerke eingesetzt (siehe Abbildung 1.4 aus Kapitel 1).

Im Bereich der Zuverlässigkeit gibt es eine weitere Klassifizierung für Echtzeitanforderungen. Man unterscheidet zwischen weichen, harten und isochronen Echtzeitanforderungen. Diese Anforderungen unterscheiden sich nach zugelassenen Zykluszeiten. Weiche Echtzeitanforderungen haben Zykluszeiten von über 10 ms. Harte Echtzeitanforderungen haben Zykluszeiten zwischen 1 ms und 10 ms. Isochrone Echtzeitanforderungen<sup>1</sup> haben Zykluszeiten zwischen 250  $\mu$ s und 1 ms mit einem Jitter kleiner als 1  $\mu$ s.

Die Menge der Nutzdaten, die zwischen Sensorknoten übertragen wird, liegt im Bereich von wenigen Bytes (48 Byte, siehe Kapitel 1.2.1). Die typischen Nutzdaten sind Status bzw. Zustandsmeldungen wie z.B. Temperatur, Drehgeschwindigkeit, aktuelle Einstellungen usw. Solche Daten können kompakt dargestellt werden.

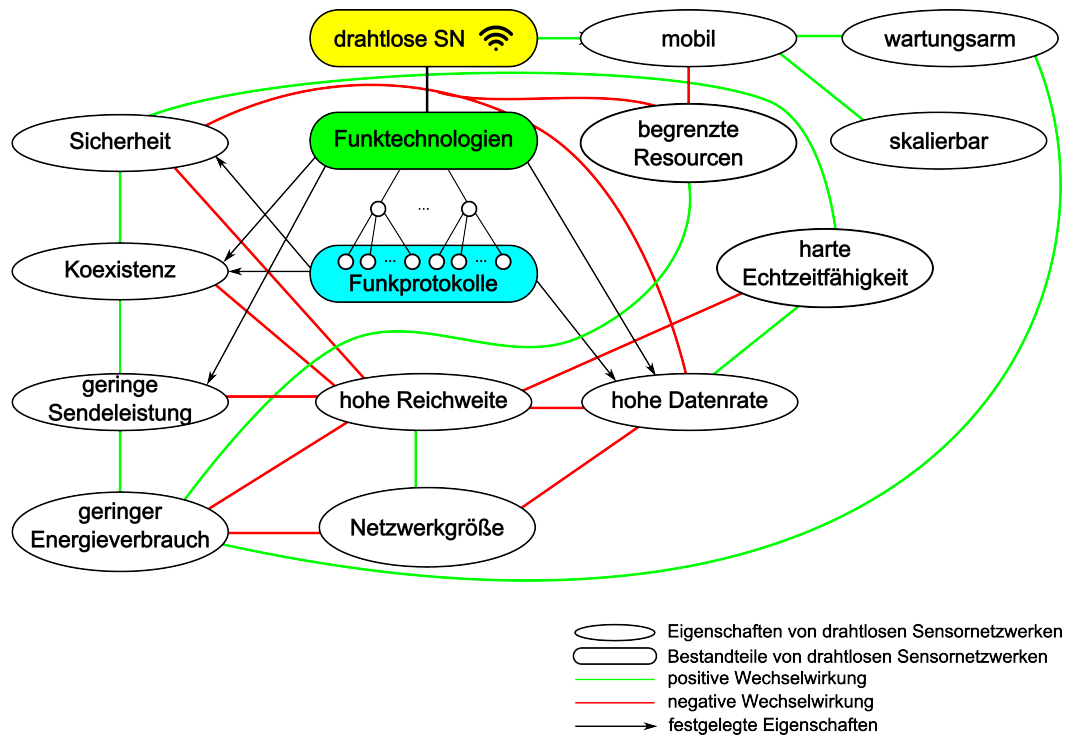
Trotz der kleinen Datenmengen werden, aufgrund der harten Echtzeitanforderungen und der möglichen Mehrfachübertragungen, hohe Datenraten in der Automatisierungsindustrie benötigt.

Bezogen auf die Nutzklassen ist das Ziel dieser Doktorarbeit, Konzepte und Lösungen für den Einsatz von WSN auf Basis der IR-UWB Funktechnologie für die Nutzklassen 0 und 1 zu evaluieren bzw. zu entwickeln. Diese Arbeit konzentriert sich auf harte Echtzeitanforderungen, unter der Annahme, dass nur geringe Datenmengen – bis zu 48 Byte – übertragen werden. Diese Ziele können durch Optimierungen verschiedener Eigenschaften der WSN erreicht werden.

In Abbildung 2.1 sind die wichtigsten Eigenschaften von WSN, welche aus den Anforderungen der Automatisierungsindustrie hergeleitet sind, dargestellt. In Abbildung 2.1 ist gezeigt, dass viele Eigenschaften sowohl direkt als auch indirekt voneinander abhängig sind. Die Optimierung einer Eigenschaft kann andere Eigenschaften des Systems verändern, d.h. positiv oder negativ beeinflussen. Aus diesem Grund ist eine Optimierung aller Eigenschaften gleichzeitig nicht möglich.

---

<sup>1</sup>Ein Echtzeitverkehr ist isochron, wenn jeder neuer Zyklus periodisch mit höchstmöglicher Genauigkeit startet.



**Abb. 2.1:** Wechselwirkungen zwischen verschiedenen Eigenschaften von drahtlosen Sensornetzwerken für die Automatisierungsindustrie.

Beispielsweise sind die Auswirkungen von nur einer Eigenschaft wie "hohe Reichweite" kompliziert und widersprüchlich. Wenn drahtlose Sensorknoten eine höhere Reichweite erreichen können, reduziert sich die Anzahl der Sensorknoten, welche die gleiche Fläche abdecken, d.h. das Netzwerk besteht aus weniger Teilnehmern. Dies reduziert die Anschaffungs- und Wartungskosten, was eine positive Auswirkung ist.

Eine hohe Reichweite kann durch Erhöhung der Sendeleistung erreicht werden. Dies führt zu einem höheren Energieverbrauch und bei batteriebetriebenen drahtlosen Sensorknoten zu einer kürzeren Lebensdauer, was wiederum mehr Wartungseinsätze erfordert und insgesamt die Kosten erhöht, was eine negative Auswirkung ist.

Außerdem kann eine höhere Sendeleistung die Koexistenzfähigkeit mit anderen Funktechnologien oder andere gleichzeitig kommunizierende drahtlose Sensorknoten beeinträchtigen. Dies hat negative Auswirkungen auf die Echtzeitfähigkeit der drahtlosen Sensorknoten. Zusätzlich hat eine hohe Reichweite Auswirkungen auf die Sicherheit: je höher die Reichweite bzw. Sendeleistung der SN, desto höher ist der Erfolg externer Angriffe durch Lauschen. Der Aufwand der Schutzmaßnahmen wie z.B. Verschlüsselung und Fehlerkorrektur erfordert leistungsstärkere und deswegen teurere Hardware, erhöht die Datenmenge und reduziert die Datenrate für Nutzdaten. Dadurch wird die Dauer der Datenübertragungsdauer und somit der Energieverbrauch erhöht, was wieder eine negative Auswirkung ist.



## 2.2 UWB als Funktechnik für drahtlose Sensornetzwerke

In diesem Abschnitt werden die Herkunft, die Grundlagen und die Regelungen der Ultra-Breitband Funktechnologie (engl. *Ultra-Wideband*, UWB) erklärt. Des Weiteren werden hier Forschungsprojekte und Forschungsarbeiten im Bereich UWB und deren Relevanz für diese Doktorarbeit dargestellt. Anschließend werden vorhandene Produkte basierend auf der UWB-Funktechnologie vorgestellt. Außerdem wird gezeigt, inwiefern sich diese Arbeit von anderen Forschungsarbeiten in diesem Bereich abgrenzt.

### 2.2.1 Definition von UWB

In der Telekommunikationstechnik werden Signale mittels ihrer Bandbreite und/oder mittels ihrer normierten Bandbreite charakterisiert.

Die Bandbreite ist die Breite des Frequenzintervalls, in welchem die Kommunikation stattfindet. Das Frequenzintervall wird durch eine untere Grenzfrequenz  $f_L$  bzw. obere Grenzfrequenz  $f_H$  definiert. Die Bandbreite  $B$  ist:

$$B = f_H - f_L$$

Die normierte Bandbreite (engl. *fractional Bandwidth*,  $B_f$ ) ist die Bandbreite, welche durch die Mittenfrequenz  $f_{mid} = \frac{f_H + f_L}{2}$  geteilt ist:

$$B_f = \frac{B}{f_{mid}} = 2 \cdot \frac{f_H - f_L}{f_H + f_L}$$

Signale werden nach ihrer normierten Bandbreite  $B_f$  wie folgt klassifiziert[41]:

- Schmalband :  $B_f < 1\%$
- Breitbandband :  $1\% \leq B_f < 20\%$
- Ultra-Breitband :  $B_f \geq 20\%$

Die UWB Funktechnologie beschreibt einen Ansatz für Nahbereichsfunkkommunikation für den kommerziellen Massenmarkt. Wichtigstes Merkmal ist die Nutzung extrem großer Frequenzbereiche mit einer normierten Bandbreite  $B_f$  von mindestens 20%. Die Definition von Ultra-Breitband wurde später erweitert, so dass auch Signale mit einer Bandbreite über 500 MHz<sup>2</sup>[42] dazugezählt werden.

### 2.2.2 IR-UWB Herkunft

Obwohl die IR-UWB Funktechnologie (engl. *Impuls Radio Ultra-Wideband*) in den letzten Jahren immer wieder als neue Zukunftstechnologie angekündigt wird, ist diese nicht neu. Tatsächlich waren die ersten Funksignale für die drahtlose Kommunikation IR-UWB-Signale. Der Erste, der die Arbeiten von Maxwell und Hertz

<sup>2</sup>Diese Definition wurde von der OFDM-Industrielobby hineinreklamiert

technisch umgesetzt und kommerziell genutzt hat, war der Italiener Guglielmo Marconi. Er war einer der Pioniere der drahtlosen Telegrafie. 1896 konnte er mit seinem entwickelten Knallfunksender (siehe Abbildung 2.2) Morsezeichen über eine Entfernung von 3 km übertragen [43]. Seine Leistungen wurden zusammen mit dem deutschen Physikprofessor Karl Ferdinand Braun 1909 mit dem Nobelpreis für Physik gewürdigt [44].

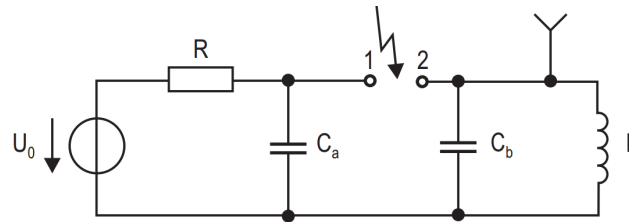


Abb. 2.2: Elektrische Schaltung eines Knallfunksender [45].

Bis Ende der 20er Jahre des 20. Jahrhunderts dominierte der Knallfunksender über die drahtlose Kommunikation. Zu dieser Zeit wurde das Potential des großen zur Verfügung stehenden Spektrums nicht erkannt und die Technik war noch nicht so weit fortgeschritten, um dieses Potential zu nutzen. Der Knallfunksender wurde von trägerbasierten Funkverfahren abgelöst, da diese sich besser für Audio- und später auch für Videoübertragungen einsetzen ließen. Des Weiteren wurden 1924 weitreichende Verbote für Funksender erlassen, um nicht die trägerbasierten Funksysteme zu stören. Damals waren Funkabstrahlungen noch nicht reguliert.

Erst in den Sechzigern Jahren des 20. Jahrhunderts wurde IR-UWB wieder zunehmend untersucht, aufgrund des technischen Fortschritts, der es ermöglichte, mit kurzen Impulsen zu arbeiten. IR-UWB wurde hauptsächlich für militärische Anwendungen im Bereich der Entfernungsmessung mittels RADAR (engl. *Radio Detection and Ranging*) eingesetzt. Als das Potential dieser Technologie auch im zivilen Bereich erkannt wurde, wurde weltweit intensiv an dieser Technologie geforscht. 1992 wurde ein erster Versuch unternommen, diese Funktechnik von der Regulierungsbehörde genehmigen zu lassen. Dies wurde aber abgelehnt, mit der Begründung, dass pulsbasierende Funksysteme ein erhebliches Interferenzpotential aufweisen, welches sich negativ auf bestehende Funksysteme auswirke. Aufgrund eines wachsenden Interesses der Industrie und der Wissenschaft wurde 2002 einer Freigabe der Benutzung durch die Regulierungsbehörde FCC (engl. *Federal Communications Commission*) [46] [47] zugestimmt. Die FCC ist eine Regulierungsbehörde, welche für die amerikanische Normung im Bereich Telekommunikation zuständig ist.

UWB ist eine Funktechnik, welche Frequenzen anderer bestehender Funksysteme nutzt. Es gibt viele Regulierungsvorschriften, die bei dieser Funktechnik zu beachten sind. Diese Vorschriften sind nicht global einheitlich, und vom Anwendungsbereich des Gerätes abhängig. Man unterscheidet die Geräte bzw. Systeme nach folgenden Anwendungsbereichen:

- Systeme, welche innerhalb von Gebäuden genutzt werden (engl. *indoor-systems*)

- Tragbare Systeme (engl. *handheld-systems*)
- Bodenradargeräte (engl. *Ground Penetrating Radars*, GPR)
- Durch-Mauern-Scanner Systeme (engl. *through-the wall imaging systems*)
- Überwachungsgeräte (engl. *monitoring systems*)
- Medizinische Geräte (engl. *medical systems*)
- Autoradarsysteme (engl. *automotive radar systems*)

Für den Massenmarkt und für Automatisierungssysteme sind die Kategorien *Indoor UWB-Systems* und *Handheld UWB-Systems* relevant. Im weiteren Teil dieser Arbeit wird auf diese Kategorien eingegangen. Des Weiteren gibt es bei den Regulierungsvorschriften starke Unterschiede zwischen den USA, Europa und weltweit. Im nächsten Abschnitt sind diese Unterschiede detaillierter beschrieben.

### 2.2.3 UWB Regulierungen

In Tabelle 2.1 sind die zugelassenen Leistungsdichten (engl. *Equivalent isotropically radiated power*, EIRP) für verschiedene Frequenzbereiche sowohl für *indoor* als auch *handheld* Geräte angegeben, die in den USA durch die FCC zugelassen sind.

**Tabelle 2.1:** EIRP-Grenzwerte der FCC für Indoor- und Handheld UWB-Geräte [43].

Frequenz, GHz	zugelassene EIRP, dBm/MHz	
	Indoor UWB-Geräte	Handheld UWB-Geräte
0-0.96	-41.3	-41.3
0.96-1.61	-75.3	-75.3
1.61-1.99	-53.3	-63.3
1.99-3.1	-51.3	-61.3
3.1-10.6	-41.3	-41.3
> 10.6	-41.3	-41.3

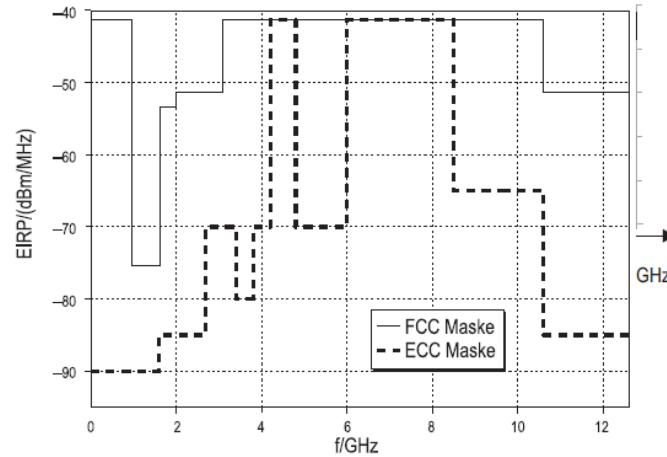
Eine Leistungsdichte von -41,3 dBm/MHz entspricht ungefähr einer Sendeleistung von 70 nW/MHz und ist für Frequenzen über 3,1 GHz und unter 0,96 GHz zugelassen (siehe Tabelle 2.1). Bei Frequenzen von 0,96 bis 1,61 GHz sind nur -75,3 dBm/MHz zugelassen. Diese Forderung begrenzt die Reichweite der Datenkommunikation, ist aber erforderlich, um andere wichtige Funktechnologien wie GSM<sup>3</sup>, GPS<sup>4</sup> oder Flugzeugsensorik, welche auch im selben Frequenzbereich arbeiten, nicht zu stören.

In Europa ist das Europäische Institut für Telekommunikationsnormen ETSI (engl. *European Telecommunications Standards Institute*) für die Regulierung von UWB zuständig. ETSI ist eine der drei großen Normungsorganisationen im Bereich

<sup>3</sup>GSM steht für (engl. *Global System for Mobile Communications*)

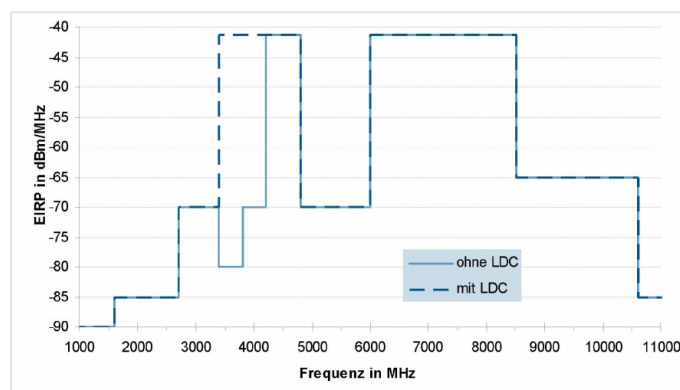
<sup>4</sup>GPS steht für (engl. *Global Positioning System*)

Telekommunikation für Europa. In Abbildung 2.3 sind die europäischen und die amerikanischen Regulierungsanforderungen zu Leistungsdichten grafisch dargestellt.



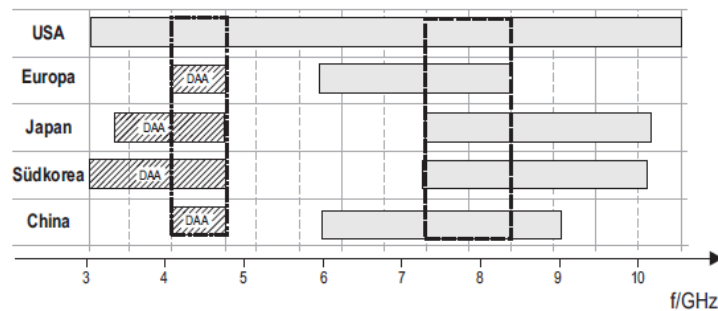
**Abb. 2.3:** Europäische und amerikanische Regulierungsmaske der Leistungsdichte für Indoor UWB-Geräte, entnommen aus [43]. Die amerikanische Regulierungsmaske erlaubt höhere Leistungsdichten als die europäische.

Wie in Abbildung 2.3 zu sehen ist, sind die europäischen Regulierungsanforderungen strenger als die amerikanischen, d.h. die zugelassene Leistungsdichte ist deutlich niedriger. Der Frequenzbereich von 4,8 bis 6 GHz wurde besonders geschützt, d.h. die zugelassene Leistungsdichte ist besonders niedrig -70 dBm/MHz, um Störungen mit anderen Systemen, welche im selben Frequenzbereich operieren, zu vermeiden. Bei der europäischen Regulierung ist es zugelassen, im Frequenzbereich von 3,4 bis 4,2 GHz mit Leistungsdichten bis zu -41,3 dBm/MHz zu operieren, wenn ein sogenannter LDC (engl. *Low Duty Cycle*) aktiv ist, siehe Abbildung 2.4. Unter LDC versteht die ETSI ein UWB-Gerät, das höchstens für eine Zeitspanne von 5 Millisekunden sendet und dann eine Pause von ca. 1 Sekunde macht. Dieses Verfahren eignet sich nur für Anwendungen mit niedrigen Datenraten.



**Abb. 2.4:** Europäische Regulierungsmaske mit und ohne LDC, entnommen aus [43]. Mit entsprechenden LDC-Verfahren, kann die Leistungsdichte im Frequenzbereich von 3,4 bis 4,2 GHz erhöht werden (siehe gestrichelte Kurve).

In Abbildung 2.5 sind die Frequenzteilungen für UWB der jeweils wichtigsten Regionen weltweit angegeben. Es gibt zwei Frequenzintervalle, von 4,2 - 4,8 GHz und von 7,5 - 8,5 GHz, welche weltweit benutzt werden können. Im ersten Frequenzintervall müssen, außer in den USA, zusätzliche Maßnahmen DDA (engl. *Detect And Avoid*) ergriffen werden, um die Koexistenz mit anderen, im selben Frequenzband operierenden, Funktechnologien sicherzustellen.

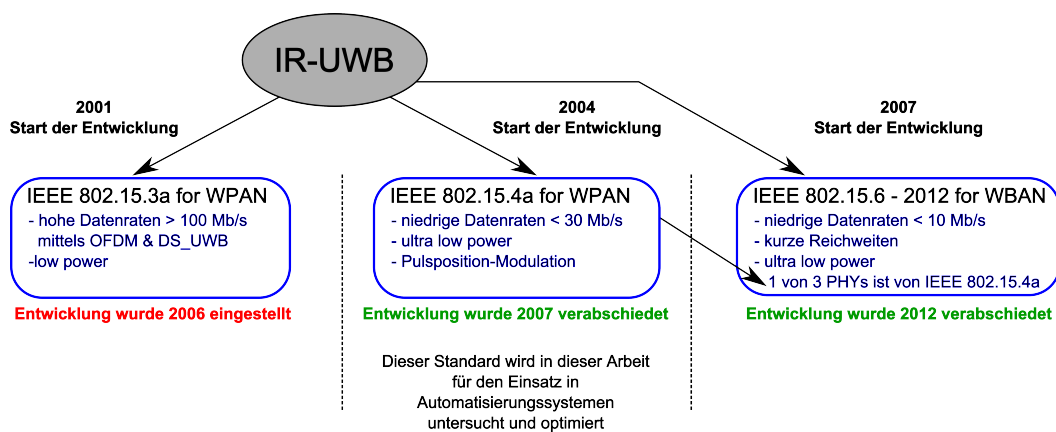


**Abb. 2.5:** Globale UWB-Frequenzteilung, entnommen aus [43]. Nur das Frequenzintervall von 7,5 GHz bis 8,5 GHz kann ohne Einschränkungen weltweit genutzt werden.

Für diese Arbeit ist das Frequenzintervall von 7,5 GHz bis 8,5 GHz relevant, da es weltweit ohne LDC oder DAA Verfahren verwendet werden kann und weil die Leistungsdichte von 41,3 MHz/dBm maximal ist.

### 2.2.4 IR-UWB Standardisierung

Die Entwicklung der verschiedenen Standards für die IR-UWB Funktechnik sind in Abbildung 2.6 schematisch dargestellt.



**Abb. 2.6:** Übersicht von IR-UWB-Standards. Es werden hier Eigenschaften wie Datenrate, Energieverbrauch, Verfügbarkeit und die Relevanz für diese Doktorarbeit dargestellt. WPAN steht für (engl. *Wireless Personal Area Network*). WBAN steht für (engl. *Wireless Body Area Network*).

In den USA formierte sich Ende 2001 eine Arbeitsgruppe (engl. *task group*), welche den Standard IEEE 802.15.3a entwickelte. Dieser Standard richtet sich an

WPAN (engl. *Wireless Personal Area Network*) Anwendungen mit hohen Datenraten und *low-power*. 2004 wurden zwei verschiedene Varianten des Standards vorgeschlagen: eine Multi-Band Variante, welche Frequenzsprünge und OFDM (engl. *Orthogonal Frequency Division Multiplexing*) beinhaltet und ein Ansatz mit DS-UWB (engl. *Direct Sequence UWB*), welcher die pulsbasierende Natur von UWB besser berücksichtigte.

Um diese beiden Ansätze besser zu vergleichen, wurde ein Standard-UWB Kanalmodell definiert. In Februar 2003 folgte daraus ein Report (IEEE 802.15.SG3a [48]), welcher die Resultate darlegt und Vorschläge zu Leistungsmessungen des im Standard definierten UWB-PHYs<sup>5</sup> macht.

Der Nachteil der Impulsfunktechnik war die Tatsache, dass diese unerprobt war. Daraufhin wurde die Multiband OFDM Alliance (MBOA) gegründet, um die OFDM-Funktechnik von WLAN (802.11a/11g) im Standard IEEE 802.15.3a durchzusetzen. Da die IEEE-Arbeitsgruppe 802.15.3a sich nicht einigen konnte, wurde Anfang 2006 die Auflösung der Arbeitsgruppe beschlossen.

Für Anwendungen, welche keine hohen Datenraten benötigen, wurde in März 2004 eine weitere Arbeitsgruppe (TG4a) gegründet, welche den Standard IEEE 802.15.4a entwickelte. Der Standard IEEE 802.15.4a spezifiziert zwei PHYs, einen UWB-PHY und einen CSS<sup>6</sup>-PHY. Der UWB-PHY ist für Frequenzen unter 1 GHz zwischen 3 und 5 GHz und zwischen 6 und 10 GHz vorgesehen. Der CSS PHY ist speziell für das 2,45 GHz ISM Band und für LR-WPAN<sup>7</sup> Anwendungen mit höherer Reichweite vorgesehen und für diese Arbeit nicht relevant.

Ein weiterer Standard für UWB-Systeme wurde in Mai 2012 von der IEEE veröffentlicht. Dieser Standard - IEEE 802.15.6-2012 für WBAN (engl. *Wireless Body Area Network*) - ist für Datenkommunikation mit kurzen Entfernungen (Körpurnah) und niedrige Datenraten (bis zu maximal 10Mb/s) vorgesehen, und stromsparend. Dieser Standard spezifiziert drei verschiedene PHYs:

1. *Narrowband* (NB)
2. *Ultra-Wideband* (UWB)
3. *Human Body Communications* (HBC)

In der Pulserzeugung unterscheidet sich der UWB-PHY des IEEE 802.15.6 nicht vom UWB-PHY des IEEE 802.15.4a. Nur die Frame-Struktur und der MAC<sup>8</sup> wurden modifiziert.

---

<sup>5</sup>PHY (engl. *Physical Layer*) ist die physikalische Schnittstelle zwischen dem MAC und dem analogen Frontend, welche hauptsächlich für das Kodieren und Dekodieren der Daten verantwortlich ist.

<sup>6</sup>CSS (engl. *Chirp Spread Spectrum*)

<sup>7</sup>LR-WPAN (engl. *Low-Rate Wireless Personal Area Network*)

<sup>8</sup>MAC steht für Medienzugriffssteuerung (engl. *Media Access Control*)

Tabelle 2.2 bietet einen erweiterten Überblick der in Abbildung 2.6 gezeigten IR-UWB-Standards. Hier wird die Verfügbarkeit der Standards und deren Eignung bzgl. der verschiedenen Anforderungen der Automatisierungsindustrie, wie z.B. Datenrate, Koexistenz usw. dargestellt.

**Tabelle 2.2:** Übersicht und Vergleich der IR-UWB Standards

Anforderungen der Automatisierungsindustrie	IR-UWB Standards		
	IEEE 802.15.3a	IEEE 802.15.4a	IEEE 802.15.6
Verfügbarkeit	x	seit 2007	seit 2012
Netzwerkgröße	+	+++	+
Netzwerk-Typ	WPAN	WPAN	WBAN
Datenrate	+++	+	+
Koexistenz	++	+++	+++
Reichweite	++	++	+
Energieverbrauch	+	+++	+++

Diese Arbeit hat als Ziel, den Einsatz der IR-UWB Funktechnologie für die Automatisierungsindustrie zu ermöglichen. Dafür soll ein existierender Standard für WPAN Netzwerke modifiziert werden, so dass mehr als 100 Sensorknoten mit hohen Datenraten kommunizieren können. Diese Aufgabe kann nur auf Basis des Standards IEEE 802.15.4a gelöst werden (siehe Tabelle 2.2). Dieser Standard unterstützt aber nur Kommunikationen mit niedrigen Datenraten und wird in dieser Arbeit dahingehend modifiziert höhere Datenraten zu unterstützen. Daher sind für diese Arbeit die Forschungsarbeiten, welche den Standard IEEE 802.15.4a nutzen, relevant.

### 2.2.5 Stand der Forschung im Bereich von IR-UWB

In den Bereichen Datenkommunikation und Lokalisierung mittels der IR-UWB-Funktechnologie wird seit 2000 wieder aktiv geforscht.

Bis 2005 existierte kein Standard für die IR-UWB-Kommunikation. Aus diesem Grund wurden viele Forschungen zwischen 2000 und 2005 auf Basis des unfertigen Standards IEEE 802.15.3a, welcher für hohe Datenraten und *low-power* konzipiert wurde, durchgeführt. Auch im Bereich der Grundlagenforschung wurde in diesem Zeitintervall gearbeitet. Einige Forschungsergebnisse, wie z.B. Kanalmodelle, Antennendesign oder Abstrahlungswerte wurden auch für spätere Projekte basierend auf dem Standard IEEE 802.15.4a verwendet. Die folgenden Schwerpunkte wurden am stärksten untersucht:

- Ermittlung der maximal erreichbaren Datenraten
- Entwicklung neuer Netzwerkarchitekturen
- Untersuchungen bzgl. der Abstrahlung
- Untersuchungen bzgl. der Koexistenzfähigkeit

- Einfluss von verschiedenen Modulationsverfahren wie PPM <sup>9</sup> und PAM <sup>10</sup>
- Ermittlung von UWB-Kanalmodellen
- Entwicklung neuer Konzepte für PHY und MAC
- Entwicklung von UWB-Lokalisierungssystemen

2005 wurde der 1. Entwurf des Standards IEEE 802.15.4a veröffentlicht. Ein besonderer Vorteil von UWB ist, dass diese Funktechnologie präzise Abstandsmessungen und Datenübertragungen ermöglicht. Forschungen über IEEE 802.15.4a konzentrieren sich auf die Bereiche der Lokalisierung und RADAR (engl. *Radio Detection and Ranging*). Ein weiterer wichtiger Forschungsschwerpunkt bleibt weiterhin die Entwicklung energiesparender effizienter UWB-Systeme. Die Orientierung des Standards auf Kommunikation mit niedrigen Datenraten und Abstandsmessung hat folgende Forschungsschwerpunkte ergeben:

- Ermittlung möglicher Einsatzgebiete
- Untersuchung alternativer Empfängerkonzepte
- Entwicklung erster Prototypen als ASIC <sup>11</sup> und FPGA <sup>12</sup>
- Entwicklung extrem energiesparender Systeme

Nach der Veröffentlichung des Standards IEEE 802.15.4a Anfang 2007 wurde in den Forschungsprojekten die Einsetzbarkeit des Standards für spezielle Anwendungsfälle geprüft und umgesetzt. Einige Forschungsprojekte hatten das Ziel, den Standard zu erweitern. Aufgrund der relativ geringen maximalen Datenrate von 27,24 Mb/s konzentriert sich die Forschung im Bereich der hohen Datenübertragung von mehr als 100 Mb/s weiterhin auf den nicht verabschiedeten Standard IEEE 802.15.3a [49].

Abbildung 2.7 gibt einen Überblick über diverse Forschungsprojekte im Bereich UWB seit 2000. Eine ausführlichere Beschreibung der Inhalte der Projekte befindet sich im Anhang A.2.

Wie in Abbildung 2.7 zu sehen ist, gab es nur wenige IR-UWB-Forschungsprojekte im Zeitintervall von 2000 bis 2004. Diese Projekte waren grundlagenforschungsorientiert. Die meisten anwendungsorientierten Forschungsprojekte liefen von 2005 bis 2013. Nur wenige Projekte wie WISE, TANDEM, KUSZ, eGo und teilweise EUWB

---

<sup>9</sup>PPM: (engl. *Pulse-Position Modulation*)

<sup>10</sup>PAM: (engl. *Pulse Amplitude Modulation*)

<sup>11</sup>ASIC: steht für anwendungsspezifische integrierte Schaltung (engl. *Application-Specific Integrated Circuit*)

<sup>12</sup>FPGA: (engl. *Field Programmable Gate Array*)



und UkoLos verwendeten den Standard IEEE 802.15.4a. Die Ergebnisse dieser Doktorarbeit, welche im Rahmen der Projekte TANDEM und KUSZ entstanden sind, wurden veröffentlicht, siehe Abschnitt "Publikationen".

Da nur wenige Projekte außer TANDEM und KUSZ ähnliche Forschungsziele verfolgt haben, können nur diese zur Evaluierung der Ergebnisse dieser Arbeit herangezogen werden.

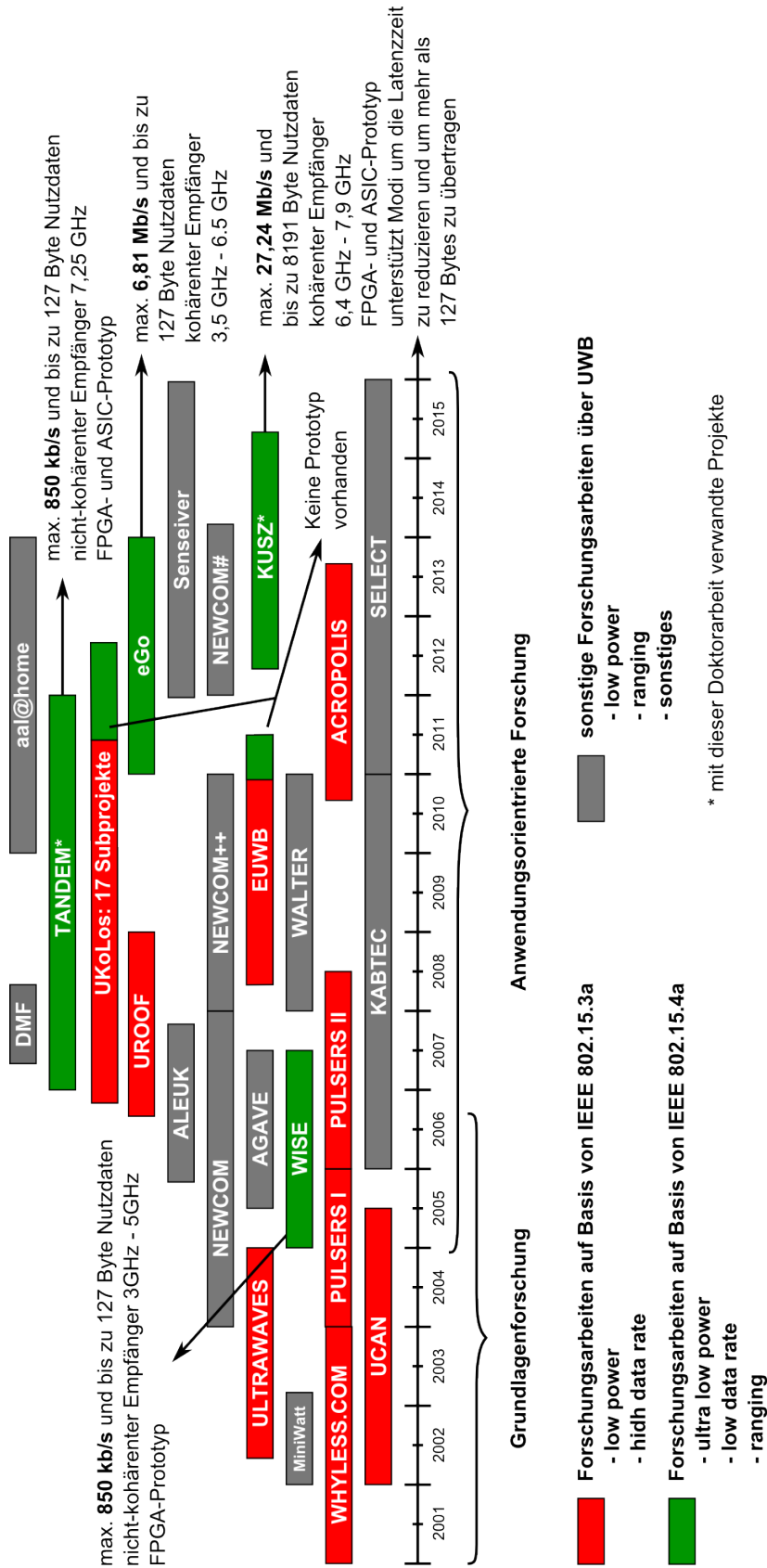
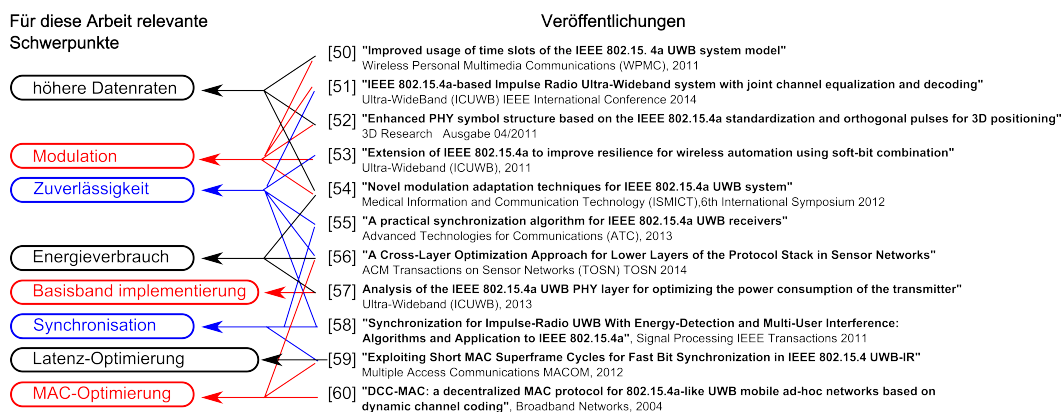


Abb. 2.7: Übersicht der Forschungsprojekte im Bereich UWB von 2000 bis 2015. Forschungsprojekte, welche den Standard IEEE 802.15.4a verwenden, siehe grün markierte Projekte, sind für diese Doktorarbeit relevant.

Die veröffentlichten Ergebnisse der in Abbildung 2.7 gezeigten Forschungsprojekten, werden hier kurz zusammengefasst und diskutiert.

Der wichtigste Forschungsschwerpunkt dieser Arbeit ist eine hohe Datenrate zu erzielen. Alle Forschungsarbeiten im Bereich hoher Datenraten, mit mehr als 100 MB/s, basieren auf dem Standard IEEE 802.15.3a. Denn die im Standard IEEE 802.15.4a vorgesehene maximale Datenrate beträgt 27,24 MB/s. Bis jetzt konnte nur eine Verdoppelung der maximalen Datenrate erreicht werden :  $2 \cdot 27,24 \text{ MB/s} = 54,48 \text{ Mb/s}$ . Siehe dazu [60],[51] und [52] aus Abbildung 2.8.

In Abbildung 2.8 sind Veröffentlichungen aufgelistet, welche für diese Doktorarbeit relevant sind. Veröffentlichungen, welche im Rahmen der Projekte TANDEM und KUSZ entstanden sind, sind nicht aufgeführt. Es gibt einige Veröffentlichungen, [61], [62], [63], [64], [65] und [66], welche den analogen Teil eines UWB-transceivers optimieren z.B. durch pulse-shaping. Die in diesen Veröffentlichungen vorgeschlagenen Optimierungen können zusätzlich zu den Optimierungen dieser Doktorarbeit angewendet werden.



**Abb. 2.8:** Für diese Arbeit relevante Veröffentlichungen und deren Optimierungsschwerpunkte

Die Ergebnisse der in Abbildung 2.8 aufgelisteten Veröffentlichungen werden hier diskutiert. Ähnlichkeiten bzw. Unterschiede zu dieser Arbeit werden verdeutlicht. Auch wenn Konzepte oder Methoden aus den Veröffentlichungen nicht für diese Arbeit genutzt werden können, so können jedoch einige Ideen modifiziert für diese Doktorarbeit angewendet werden.

[50] "Improved usage of time slots of the IEEE 802.15.4a UWB system model", *Wireless Personal Multimedia Communications (WPMC), 2011*

In diesem Paper wurden die im Standard IEEE 802.15.4a vorgegebenen *time-slots* effizienter ausgenutzt. Es werden 2 Bursts<sup>13</sup> anstelle nur eines einzelnen Bursts innerhalb eines Datensymbols übertragen, um die Datenrate zu verdoppeln. Allerdings wurde die Pulswiederholrate (engl. *Pulse Repetition*

<sup>13</sup>Ein Burst ist eine Sequenz von Impulsen (von 1 bis 16), die für die Übertragung eines Bits verwendet wurde.

*Frequency*, PRF) nicht berücksichtigt. Ein solches 2-Burst-Verfahren ohne Reduzierung der Sendeleistung verstößt gegen die Regulierungsvorschriften.

In dieser Arbeit wurde die Idee, mehrfache Bursts in einem Datensymbol zu übertragen, auch verfolgt. Allerdings wurden hierbei die Regulierungsvorschriften und die PRF eingehalten.

- [51] ***"IEEE 802.15.4a-based Impulse Radio Ultra-Wideband system with joint channel equalization and decoding", Ultra-WideBand (ICUWB) IEEE International Conference 2014***

In diesem Artikel wurde die aktive Zeit des Empfängers durch Anpassung der PPM<sup>14</sup> wesentlich reduziert. Der zeitliche Abstand zwischen den aktiven *time-slots* des Empfängers wurde reduziert, um längere Pausen zu ermöglichen und den Energieverbrauch zu senken.

In dieser Doktorarbeit werden die *time-slots* auch derart angepasst, dass der Empfänger möglichst selten aktiv ist. Im Gegensatz zu dem Artikel [51] wurden in dieser Doktorarbeit bis zu 16 gleichzeitig sendende drahtlose Sensorknoten bei der Wahl der *time-slots* berücksichtigt.

- [52] ***"Enhanced PHY symbol structure based on the IEEE 802.15.4a standardization and orthogonal pulses for 3D positioning", 3D Research Ausgabe 04/2011***

In diesem Artikel konnte die Datenrate verdoppelt werden. Dies wurde durch Einfügen zusätzlicher Impulse in den Guard-Intervallen der Daten-Symbole erreicht. Der Nachteil dieses Ansatzes ist, dass die PRF nicht eingehalten wird und demzufolge muss die Sendeleistung reduziert werden, um die Regulierungsvorschriften nicht zu verletzen.

In dieser Doktorarbeit wurde auch die PPM angepasst, aber im Gegensatz zu [52] unter Berücksichtigung der PRF.

- [53] ***"Extension of IEEE 802.15.4a to improve resilience for wireless automation using soft-bit combination", Ultra-Wideband (ICUWB), 2011***

In dieser Veröffentlichung wurde eine zusätzliche Fehlerkorrektur beim Empfang angewendet, um die Robustheit zu erhöhen und die Anzahl an Mehrfachübertragungen zu senken. Es werden Softbits<sup>15</sup> generiert. Softbits sind

<sup>14</sup>PPM (engl. *Pulse-Position Modulation*)

<sup>15</sup>Mit Softbit wird eine Zahl bezeichnet, welche angibt mit welcher Wahrscheinlichkeit ein Bit den Wert 0 oder 1 hat.

zusätzliche Informationen der einzelnen Bits und können dazu genutzt werden, eine bessere Fehlerkorrektur zu erreichen. Die Softbits können mit den schon vorhandenen Fehlerkorrekturverfahren (Reed-Solomon und Viterbi) kombiniert werden. Zusätzlich wurde der PHY header PHR (engl. *PHY header*, PHR) erweitert, um eine wiederholte Übertragung von einer normalen Übertragung unterscheiden zu können.

Im Standard IEEE 802.15.4a ist die Möglichkeit einer Erweiterung des PHY Headers vorgesehen. In dieser Doktorarbeit wird der PHY Header auch erweitert, um zusätzliche Informationen zu übertragen. Es wurde auch versucht, verschiedene Fehlerkorrekturverfahren zu kombinieren, um bessere Latenzzeiten zu erreichen.

- [54] *"Novel modulation adaptation techniques for IEEE 802.15.4a UWB system", Medical Information and Communication Technology (ISMICT), 6th International Symposium 2012*

In diesem Artikel wurde die Datenrate von 851 kb/s auf 1,7 Mb/s verdoppelt. Dafür wurde die Impulspolarität des Bursts zur Datenübertragung anstatt der PPM verwendet.

Diese Doktorarbeit nutzt auch die Impulspolarität, um die Datenrate zu erhöhen. Gleichzeitig wird aber die Impulspolarität für eine Fehlerkorrektur genutzt.

- [55] *"A practical synchronization algorithm for IEEE 802.15.4a UWB receivers", Advanced Technologies for Communications (ATC), 2013*

Es handelt sich hierbei um ein zweistufiges Synchronisationsverfahren. Dieses Verfahren verwendet längere PREAMBLE-Symbole und ist darauf ausgelegt, die Zuverlässigkeit zu erhöhen und die Komplexität der Synchronisation zu reduzieren.

Dieses zweistufige Verfahren dient der Erhöhung der Zuverlässigkeit aber erhöht dabei die Synchronisationsdauer und beschränkt die Datenrate auf 851 kb/s. Daher ist das in [55] veröffentlichte Verfahren zur Erhöhung der Zuverlässigkeit für diese Doktorarbeit nicht verwendbar.

- [56] *"A Cross-Layer Optimization Approach for Lower Layers of the Protocol Stack in Sensor Networks" ACM Transactions on Sensor Networks (TOSN) TOSN 2014*

In diesem Artikel wurde untersucht, wie das im Standard IEEE 802.15.4a vorgesehene Reed-Solomon Fehlerkorrektur-Verfahren mit verschiedenen MAC-Ansätzen den Energieverbrauch beeinflusst. Insgesamt werden weniger Nachrichtenviederholungen benötigt, wenn das Fehlerkorrektur-Verfahren mit einem TDMA<sup>16</sup>-MAC verwendet wird. Aufgrund der zeitlichen Trennung der Übertragungen der verschiedenen Sensorknoten werden Kollisionen vermieden. So wird die Datenübertragung robuster und energiesparender.

Die Vorteile des in [56] beschriebenen Ansatzes, die Anzahl an Kollisionen zu reduzieren, werden auch in dieser Doktorarbeit verwendet. In dieser Arbeit werden die genutzten *time-slots* der verschiedenen Sensorknoten getrennt, so dass die Wahrscheinlichkeit der Kollisionen minimiert wird. Nicht vermeidbare Kollisionen werden korrigiert.

- [57] *"Analysis of the IEEE 802.15.4a UWB PHY layer for optimizing the power consumption of the transmitter", Ultra-Wideband (ICUWB), 2013*

Dieser Artikel zeigt, wie negativ sich die hohe Taktfrequenz von 499,2 MHz auf den Energieverbrauch beim Sender auswirkt. Es wurde untersucht, wie sich der Energieverbrauch reduziert, wenn einige Komponenten wie z.B. der LFSR<sup>17</sup> so implementiert werden, dass diese mit einer niedrigeren Taktfrequenzen arbeiten können. Bei einer Datenrate von 851 kb/s konnte der Energieverbrauch um 49,2 % und bei 6,81 Mb/s um 47,4 % gesenkt werden [57].

In dieser Doktorarbeit wurde die Idee, die meisten Komponenten mit einer niedrigen Taktfrequenz zu betreiben, auch verfolgt. Alle Komponenten außer Serialisierer und Parallelisierer werden mit 31,2 MHz getaktet. Serialisierer und Parallelisierer werden mit einer höheren Frequenz von 499,2 MHz getaktet.

- [58] *"Synchronization for Impulse-Radio UWB With Energy-Detection and Multi-User Interference: Algorithms and Application to IEEE 802.15.4a", Signal Processing, IEEE Transactions on (Volume:59 , Issue: 11 ), 2011*

In dieser Veröffentlichung wurde die Zuverlässigkeit der Synchronisation für Energiedetektoren verbessert. Dafür wurden optimierte PREAMBLE-Sequenzen für Energiedetektoren verwendet. Außerdem wurden die PREAMBLEe-Detektierung und die SFD<sup>18</sup>-Detektierung getrennt voneinander durchgeführt.

---

<sup>16</sup>TDMA (engl. *Time Division Multiple Access*)

<sup>17</sup>LFSR (engl. *Linear Finite Shift Register*)

<sup>18</sup>SFD (engl. *Strat Frame Delimiter*)

In dieser Doktorarbeit wurden die PREAMBLE-Sequenzen auch optimiert, jedoch für kohärente Empfänger. Die PREAMBLE-Sequenzen wurden zusätzlich gekürzt, ohne die Zuverlässigkeit zu reduzieren, um die Synchronisationsdauer zu reduzieren. Die separate Erkennung von PREAMBLE-Symbol und SFD wurde in dieser Arbeit auf eine ähnliche Weise wie in [58], aber für kohärente Empfänger implementiert.

- [59] *"Exploiting Short MAC Superframe Cycles for Fast Bit Synchronization in IEEE 802.15.4 UWB-IR", Multiple Access Communications MACOM, 2012*

Hier wurde der MAC aus dem Standard IEEE 802.15.4e mit kürzeren PREAMBLE-Symbolen aus dem Standard IEEE 802.15.4a kombiniert. Es konnte eine Zykluszeit von 2 ms für 32 Knoten erreicht werden. Die Information über die Menge der Daten, die in beiden Richtungen übertragen wurden, ist in [59] nicht angegeben.

Ein ähnlicher Ansatz, den MAC mit kürzeren PREAMBLE-Symbolen zu kombinieren wird auch in dieser Arbeit verfolgt. Es werden sowohl die Synchronisation als auch die Datenübertragung für geringe Latenzzeiten optimiert. Die in [59] erreichten Ergebnisse - 2 ms Zykluszeit bei 32 Knoten - sind für die Anforderungen dieser Arbeit nicht ausreichend. Da Sensornetze bis zu 128 Sensorknoten beinhalten sollen und die Zykluszeit auf 1 ms beschränkt ist (siehe Kapitel 1.2), müssen die Ergebnisse aus [59] mindestens 8 mal verbessert werden.

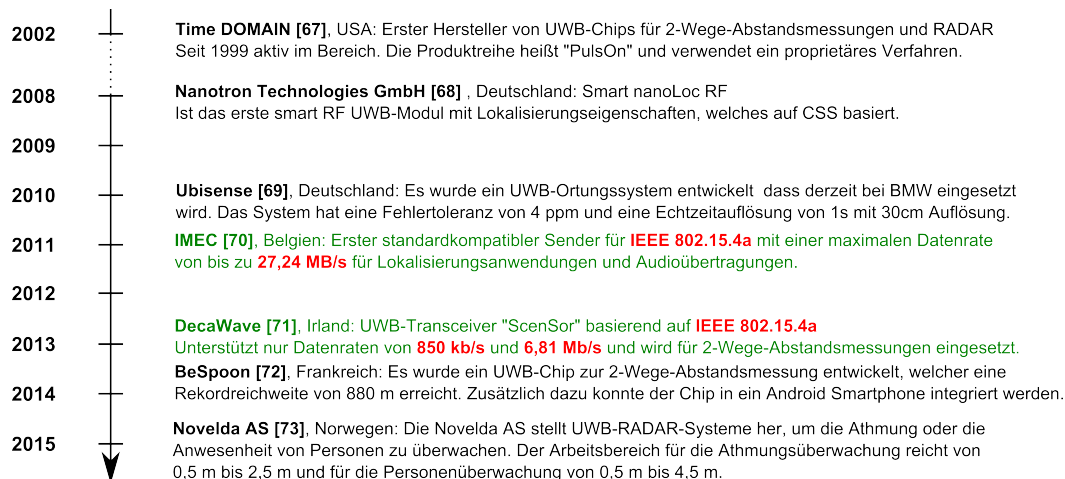
- [60] *"DCC-MAC: a decentralized MAC protocol for 802.15.4a-like UWB mobile ad-hoc networks based on dynamic channel coding", Broadband Networks, 2004*

Hier wurde ein DCC-MAC (engl. *Dynamic Channel Coding*) entwickelt, welcher im Zusammenspiel mit dem PHY (Physical Layer) in der Lage ist, die Bitrate dynamisch anzupassen. Damit wird die Robustheit der Datenübertragung verbessert, wenn mehrere Teilnehmer gleichzeitig Daten übermitteln. Diese Methode ermöglicht einen höheren Datendurchsatz, weil weniger Nachrichten wiederholt geschickt werden. Die maximale Datenrate beträgt dabei nicht mehr als 27,24 MB/s.

Diese Methode ist für größere Netzwerke nicht anwendbar. Die Datenrate wird nicht erhöht. Somit sind die Ergebnisse aus [60] für diese Arbeit nicht relevant, aber die Idee einer variablen von der Anzahl an Sensorknoten abhängigen Datenrate, wurde in dieser Arbeit in Kapitel 3 eingesetzt.

## 2.2.6 Einsatz von UWB in kommerziellen Produkten

Am Markt existieren nur wenige Produkte, welche auf dem Standard IEEE 802.15.4a basieren und die IR-UWB Impulsfunktechnik verwenden. Abbildung 2.9 zeigt eine Auflistung einiger Firmen und ihrer Produkte in diesem Bereich.



**Abb. 2.9:** Übersicht der Markteinführung kommerzieller Produkte im Bereich IR-UWB: die grün markierten Produkte sind für diese Arbeit relevant, da diese den Standard IEEE 802.15.4a verwenden. Die erreichbaren Datenraten sind rot markiert.

Für diese Arbeit sind nur die Produkte von IMEC [70] und DecaWave [71] relevant, da diese den Standard IEEE 802.15.4a verwenden. Ein UWB-Transceiver-Modul von der Firma DecaWave wird für die Evaluierung der Ergebnisse dieser Arbeit verwendet (siehe Kapitel 4.4.3).

## 2.3 Abgrenzung zum Stand des Wissens: Zusammenfassung

Die Forschungsschwerpunkte liegen bei den meisten Veröffentlichungen darin, die UWB-Funktechnik für extrem energiesparende Systeme mit geringen Datenraten oder für Abstandsmessungen zu optimieren. Nur in wenigen Projekten wurden die Möglichkeiten, höhere Datenraten und niedrige Latenzzeiten zu erreichen, untersucht. Viele Forschungsgruppen bearbeiten ähnliche Themen, aber für verschiedene Frequenzbänder und verschiedene Regionen, d.h. für verschiedene Regulierungsanforderungen. Viele der erforschten Lösungen haben den Nachteil, dass diese nicht mit dem aktuellen Standard IEEE 802.15.4a kompatibel sind und/oder die Beschränkungen der diversen Regulierungsbehörden nicht einhalten.

Diese Arbeit unterscheidet sich gegenüber den anderen dahingehend, dass der Schwerpunkt hier bei höheren Datenraten mit geringen Latenzzeiten in einem größeren Netzwerk liegt. Alle in dieser Arbeit vorgeschlagenen Optimierungen wurden unter Berücksichtigung der PRF und der Regulierungsvorschriften durchgeführt. Diese



Arbeit orientiert sich an den bestehenden Standard IEEE 802.15.4a, welcher in den nächsten Kapiteln der Doktorarbeit als UWB-Standard bezeichnet wird, aber auch nicht standardkonforme Verfahren werden untersucht. Die Optimierungen finden auf Hardware-Ebene und im Übertragungsprotokoll statt. Einige Forschungsarbeiten konzentrieren sich nur auf weniger komplexe nicht-kohärente Empfänger<sup>19</sup> (Energiedetektoren). Die in dieser Arbeit vorgestellten Optimierungen sind für kohärente Empfänger anwendbar. Die Anforderungen der Automatisierungsindustrie konnten bei unveränderter Sendeleistung für eine standardkonforme Datenübertragung erreicht werden. Das in dieser Doktorarbeit verwendete Frontend unterstützt mehrere Kanäle im Bereich von 6,4 GHz bis 7,9 GHz und kann somit weltweit verwendet werden.

---

<sup>19</sup>nicht-kohärente Empfänger sind Energiedetektoren. Sie können nur die Energie aber nicht die Polarität der empfangenen Impulse detektieren



# Optimierung des UWB-Standards

---

## Inhaltsverzeichnis

---

3.1	Prinzipielle Umsetzbarkeit der Optimierungen . . . . .	<b>38</b>
3.2	Optimierung der Punkt-zu-Punkt Datenübertragungen . . . . .	<b>48</b>
3.3	Optimierung der Punkt-zu-Multipunkt Datenübertragung . . . . .	<b>64</b>
3.4	Zuverlässigkeit . . . . .	<b>97</b>
3.5	Zusammenfassung . . . . .	<b>110</b>

---

In diesem Kapitel wird untersucht, ob die Anforderungen der Automatisierungsindustrie unter Anwendung von IR-UWB auf Basis des UWB-Standards<sup>1</sup> prinzipiell erfüllt werden können. Die Umsetzbarkeit der Optimierungen wird theoretisch untersucht und Optimierungskriterien sowie Optimierungsansätze werden dargestellt. Darauf aufbauend werden mögliche Lösungen vorgeschlagen. Die Lösungen werden theoretisch hergeleitet, berechnet und bewertet. Das Ziel dieser Arbeit ist die Erfüllung der Anforderungen der Automatisierungsindustrie mit möglichst geringen Anpassungen am UWB-Standard. Deshalb ist dieses Kapitel so aufgebaut, dass sich schrittweise einer Lösung angenähert wird, welche die Anforderungen der Automatisierungsindustrie erfüllt. Mit jedem Schritt steigt der Grad der Modifizierung des UWB-Standards.

Im Abschnitt 3.2 werden die Bestandteile eines Standard UWB-Frames einzeln untersucht und optimiert. Die Dauer der optimierten Punkt-zu-Punkt Kommunikation auf Basis des UWB-Standards wird berechnet.

Im Abschnitt 3.3 werden Optimierungen bzgl. der Geschwindigkeit der Datenübertragung bei Punkt-zu-Multipunkt Kommunikation vorgenommen.

Abschließend werden im Abschnitt 3.4 zusätzliche Optimierungen bzgl. der Zuverlässigkeit der Datenübertragung bei Punkt-zu-Multipunkt Kommunikation vorgenommen.

Die Beschreibung der teilweisen Umsetzung der hier vorgeschlagenen Optimierungen in die Praxis erfolgt im Kapitel 4.

---

<sup>1</sup>Damit ist der Standard IEEE 802.15.4a gemeint.

### 3.1 Prinzipielle Umsetzbarkeit der Optimierungen

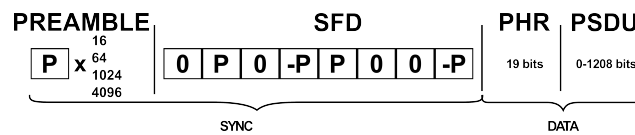
Um die Anforderungen aus Kapitel 1.2.1 zu erfüllen, müssen diese Anforderungen sowohl theoretisch als auch technisch machbar sein.

Der UWB-Standard ist nicht für Datenraten höher als 27,24 Mb/s vorgesehen. In diesem Kapitel wird untersucht, welche Anforderungen der Automatisierungsindustrie bereits mit dem UWB-Standard erfüllt sind und welche nicht erfüllt sind. Außerdem wird geprüft, ob die hier vorgeschlagenen Optimierungen die Beschränkungen seitens der Standardisierungsbehörden bzgl. UWB nicht verletzen. Dafür wird das Datenübertragungsprotokoll aus dem UWB-Standard detailliert untersucht.

#### 3.1.1 Die Struktur eines UWB-Frames

Entsprechend dem Datenübertragungsprotokoll aus dem UWB-Standard setzt sich ein

UWB-Datenpaket – ein UWB-Frame – aus der Synchronisationsphase SYNC und der Datenübertragungsphase DATA zusammen (siehe Abbildung 3.1). Die Synchronisationsphase besteht aus 2 Teilen: PREAMBLE und SFD (engl. *Start Frame Delimiter*). Die Datenübertragungsphase besteht auch aus 2 Teilen: PHR (engl. *Physical Header*) und PSDU (engl. *Physical Service Data Unit*).



**Abb. 3.1:** Struktur eines UWB-Frames.

Ein UWB-Frame besteht aus zwei Phasen: Synchronisationsphase (SYNC) und Datenübertragungsphase (DATA).

#### Synchronisationsphase

Die Synchronisationsphase ermöglicht es dem Empfänger, die richtigen Empfangseinstellungen wie z.B. den Verstärkungsfaktor (engl. *Automatic Gain Control*, AGC), einzustellen. Der PREAMBLE-Teil startet die Synchronisation und der SFD-Teil beendet die Synchronisation. Der PREAMBLE-Teil kommt vor der eigentlichen Nachricht, um eine Nachricht anzukündigen und die Synchronisation der beiden Teilnehmer zu initiieren. Der PREAMBLE-Teil besteht aus 16, 64, 1024 oder 4096 identischen PREAMBLE-Symbolen (PS) **P**. Der SFD-Teil des UWB-Frames ist eine konstante Sequenz aus 8 PREAMBLE-Symbolen, welche die Synchronisationsphase beendet und den Start der Datenübertragung ankündigt. Im SFD-Teil werden 3 unterschiedliche PS verwendet: das leere PREAMBLE-Symbol **O**, das PREAMBLE-Symbol **P** und das invertierte PREAMBLE-Symbol **-P**.

Ein PS ist eine 31-stellige ternäre Impulssequenz, d.h. eine Impulssequenz die aus positiven (bezeichnet mit '1'), negativen (bezeichnet mit '-1') oder leeren (bezeichnet

mit '0') Impulsen besteht. Abbildung 3.2 a) stellt jedes PREAMBLE-Symbol als Impulssequenz schematisch dar. Ein Teil des schematisch dargestellten PREAMBLE-Symbols **-P** (siehe grün markierten Bereich in Abbildung 3.2 a)) ist als Impulsfolge, entsprechend dem analogen Sendesignal, dargestellt (siehe Abbildung 3.2 b)). Die Dauer eines Impulses ist mittels der im UWB-Standard definierten Impulsfrequenz von 499,2 MHz definiert:

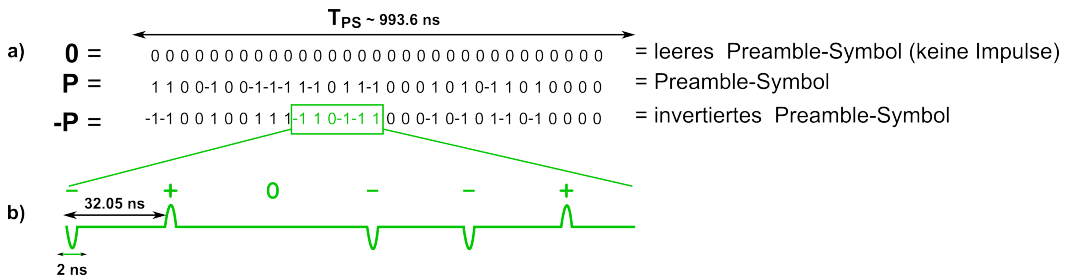
$$Impulsbreite = \frac{1}{499,2 \text{ MHz}} \approx 2 \text{ ns}$$

Der zeitliche Abstand zwischen 2 Impulsen ist im UWB-Standard auf 15 Impulsbreiten<sup>2</sup> festgelegt. Somit beträgt die Dauer eines Impulses, einschließlich der 15 leeren Impulsbreiten:

$$T_{Imp} = \frac{1 + 15}{499,2 \text{ MHz}} \approx 32 \text{ ns}$$

Daraus und aus der Länge  $n$  der Impulssequenz des PS, folgt eine PS-Dauer  $T_{PS}$  von:

$$T_{PS} = 31 \cdot T_{Imp} \approx 993,6 \text{ ns}$$



**Abb. 3.2:** Ternäre Darstellung der PREAMBLE-Symbole.  
 a) schematische Darstellung eines leeren PS **O**, des PS **P** und des invertierten PS **-P**. Die Länge der Impulssequenzen für jedes PS ist  $n = 31$ , d.h. die Sequenz besteht aus 31 Impulsen.  
 b) ein Ausschnitt aus dem invertierten PS **-P**; als analoges Sendesignal dargestellt.

Impulssequenzen für PREAMBLE-Symbole werden üblicherweise so gewählt, dass die Anzahl der Nullen  $\#0$  der Summe der minus Einsen  $\#(-1)$  und der plus Einsen  $\#1$  entspricht, daraus folgt:

$$\#0 + \#1 + \#(-1) = n$$

und

$$2 \cdot \#I_{PS} = n \tag{3.1}$$

Mit  $\#I_{PS}$  wird die Anzahl an nicht leeren Impulsen in einem PREAMBLE-Symbol und mit  $n$  wird die Länge der Impulssequenz der PREAMBLE-Symbole bezeichnet. Formel (3.1) ist kein zwingendes Kriterium. Wird es erfüllt, erleichtert

<sup>2</sup>diese 15 Impulsbreiten können auch als leere Impulse interpretiert werden, aber in dieser Arbeit ist das nicht der Fall

es das Ermitteln des optimalen Verstärkungsfaktors beim Empfänger. Aufgrund der ungeraden Länge  $n = 31$  der Impulssequenz kann dieses Kriterium nicht erreicht werden. Für  $n$  ungerade gibt es 2 Fälle, welche nur um  $\pm 1$  vom idealen Verhältnis (3.1) abweichen:

- $2 \cdot \#I_{PS} = n - 1$
- $2 \cdot \#I_{PS} = n + 1$ : Dieser Fall, mit  $n = 31$  und  $\#I_{PS} = 16$ , wird im UWB-Standard verwendet.

Anhand der Länge der Impulssequenz  $n$  des PREAMBLE-Symbols  $\mathbf{P}$  und der Anzahl an nicht leeren Impulsen  $\#I_{PS}$  kann die durchschnittliche Impulswiederholrate (engl. *Pulse Repetition Frequency, PRF*) des PREAMBLE-Symbols  $\mathbf{P}$  bestimmt werden. Die PRF für ein PREAMBLE-Symbol ist die Anzahl an nicht leeren Impulsen innerhalb des PREAMBLE-Symbols, dividiert durch die Dauer des PREAMBLE-Symbols.

$$PRF_{PS} = \frac{\#I_{PS}}{T_{PS}} = \frac{\#I_{PS}}{n \cdot T_{Imp}} \approx \frac{16}{31} \cdot \frac{1}{32 \text{ ns}} \approx 16,1 \text{ MHz} \quad (3.2)$$

Die PRF eines optimierten PREAMBLE-Symbols  $PS'$  mit anderer Impulssequenzlänge  $N$  und anderer Anzahl an nicht leeren Impulsen  $K$ , darf keine höhere PRF aufweisen als  $PRF_{PS}$ , um nicht die UWB-Regulierungen zu verletzen. Daraus folgt nun ein zweites Kriterium für die Auswahl eines optimierten PREAMBLE-Symbols  $PS'$ :

$$PRF_{PS'} \leq PRF_{PS} \Rightarrow \frac{K}{N \cdot T_{Imp}} \leq \frac{\#I_{PS}}{n \cdot T_{Imp}} \Rightarrow \frac{n}{\#I_{PS}} \cdot K \leq N$$

Mit  $n = 31$  und  $\#I_{PS} = 16$  ergibt dies folgendes Kriterium:

$$\frac{31}{16} \cdot K \leq N \quad (3.3)$$

### Datenübertragungsphase

Die Datenübertragungsphase des UWB-Frames besteht aus zwei Teilen: PHR und PSDU. Der PHR-Teil dient der Vorbereitung der Datenübertragung. Die eigentliche Datenübertragung findet im PSDU-Teil statt.

Der PHR-Teil ist eine 20-Bit lange Folge mit Informationen zum kommenden Datenstrom, wie Nachrichtenlänge und Datenrate. Beim PHR-Teil ist der Wert des ersten Bits entsprechend dem UWB-Standard immer '0'. Dadurch ist die tatsächliche Länge des PHR-Teils, d.h. ohne dieses '0'-Bit, nur 19 Bit.

Abbildung 3.3 stellt die Struktur eines Signals für die Übertragung eines Bits entsprechend dem UWB-Standard dar. Die Dauer dieses Signals  $T_{Bit}$  wird in 4 Bereichen unterteilt: Burst Position 0, Burst Position 1 und 2 Guard-Intervalle. Jeder dieser 4 Bereiche besteht aus gleich langen Zeit-Intervallen (engl. *time-slots*,

*TS*). Also insgesamt besteht ein Bit-Signal aus 32 *TS*, wobei nur einer von 32 *TS* ein *Burst*-Signal (weiter *Burst*) beinhaltet. Alle anderen *TS* sind leer. Ein *Burst* ist eine Impulsfolge, bestehend aus bis zu  $\#I_{Burst} = 16$  nicht leeren Impulsen. Die Anzahl der nicht leeren Impulse im *Burst*, und dadurch die Dauer des Bursts bzw. *TS*, ist von der gewählten Datenrate abhängig und im UWB-Standard festgelegt (siehe Tabelle 3.1).

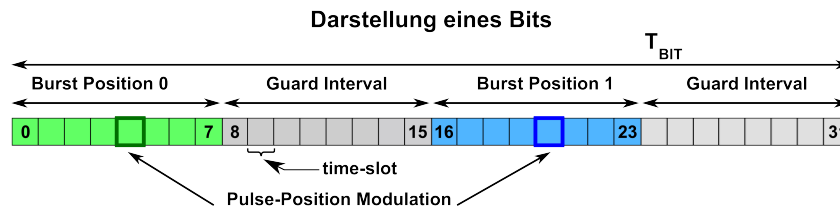
Im PHR-Teil besteht der *Burst* aus 16 Impulsen. Deshalb dauert ein *Burst* bzw. *TS*:

$$TS = \frac{\#I_{Burst}}{499,2 \text{ MHz}} \cdot 16 \approx 32 \text{ ns}$$

Die Dauer eines Bits ist dementsprechend:

$$T_{Bit_{PHR}} = 32 \cdot TS = 32 \cdot \frac{16}{499,2 \text{ MHz}} \approx 1025,64 \text{ ns}$$

Die Bit-Information wird mittels Puls-Position-Modulation (*engl. pulse-position-modulation, PPM*) kodiert übertragen. Um ein Bit mit dem Wert '0' zu übertragen, muss ein *Burst* innerhalb des Bereichs '*Burst Position 0*' gesendet werden. Um ein Bit mit dem Wert '1' zu übertragen, muss ein *Burst* innerhalb des Bereichs '*Burst Position 1*' gesendet werden.



**Abb. 3.3:** Darstellung eines Bit-Signals für die UWB-Übertragung.

Ein Bit besteht aus 32 *time-slots*.

Wenn der Wert des Bits '0' ist, wird ein *Burst* in '*Burst Position 0*' (siehe grün markierter Bereich) gesendet. Wenn der Wert des Bits '1' ist, muss ein *Burst* in '*Burst Position 1*' (siehe blau markierter Bereich) gesendet.

Die Struktur, der Bitrepräsentation ist für PHR- und PSDU-Teil gleich. Wie in Tabelle 3.1 gezeigt ist, ist die Anzahl der Impulse im *Burst* von der Datenrate abhängig. Da innerhalb eines Bit-Signals nur der *time-slot* mit einem *Burst* Impulse beinhaltet und alle anderen *time-slots* leer sind, ist die Anzahl der Impulse  $\#I_{Burst}$  gleich der Anzahl an Impulsen im Bit  $\#I_{Bit}$ . Die Anzahl der Impulse im Bit dividiert durch die Dauer des Bits bestimmt die im UWB-Standard festgelegte  $PRF_{Bit}$  von  $15,6 \text{ MHz}$ .

$$PRF_{Bit} = \frac{\#I_{Burst}}{T_{Bit}} = \frac{\#I_{Burst}}{32 \cdot TS} = \frac{\#I_{Burst}}{32 \cdot \frac{\#I_{Burst}}{499,2 \text{ MHz}}} = \frac{499,2 \text{ MHz}}{32} \approx 15,6 \text{ MHz} \quad (3.4)$$

Wie aus Formel 3.4 folgt, ist die PRF für alle Datenraten gleich. Die PRF, die Amplitude und die Breite der Impulse bestimmen die Leistung des Bit-Signals.

Diese darf die UWB-Regulierungen nicht verletzen.

Der PSDU-Teil ist die eigentliche Nachricht mit Nutzdaten und Overhead, wobei die Menge der Daten im Overhead von der Menge der Nutzdaten abhängig und im UWB-Standard festgelegt ist. Dieser Teil des UWB-Frames kann mit unterschiedlichen Nutzdatenraten, d.h. mit unterschiedlichen Geschwindigkeiten der Datenübertragung, übertragen werden: 27,24 Mb/s, 6,81 Mb/s, 851 kb/s und optional mit 110 kb/s. Die optionale Nutzdatenrate mit 110 kb/s wird für diese Arbeit nicht weiter untersucht.

Im UWB-Standard ist mit Datenrate die durchschnittliche Nutzdatenrate der PSDU gemeint, d.h. die Nutzdatenrate ist die Menge der Daten im PSDU-Teil einschließlich des Overheads dividiert durch die Dauer des ganzen PSDU-Teils. Da der Overhead variabel und nichtlinear von der Datenmenge abhängig ist, werden im UWB-Standard die durchschnittlichen Nutzdatenraten anders berechnet. Die Nutzdatenraten ergeben sich aus der tatsächlichen Bitrate multipliziert mit einem Faktor  $\alpha = \frac{\text{Overall FEC Rate}}{2}$  [37]. Der Wert des Faktors  $\alpha$  stellt den durchschnittlichen Anteil des Overheads der Nutzdaten dar. Tabelle 3.1 zeigt die Parameter eines Bit-Signals im PSDU-Teil bei verschiedenen Datenraten, entsprechend der im UWB-Standard vorgesehenen Berechnung.

**Tabelle 3.1:** Parameter eines Bits im PSDU-Teil in Abhängigkeit von der Datenrate.

Im UWB-Standard festgelegte Parameter			TS	Bit-Dauer $T_{Bit} = 32 \cdot TS$
Datenrate	$\alpha$	$\#I_{Burst}$		
851 kb/s	0,87	16	$\frac{16}{499,2 \text{ MHz}} \approx 32 \text{ ns}$	$T_{Bit_{851 \text{ kb/s}}} \approx 1025,64 \text{ ns}$
6,81 Mb/s	0,87	2	$\frac{2}{499,2 \text{ MHz}} \approx 4 \text{ ns}$	$T_{Bit_{6,81 \text{ Mb/s}}} \approx 128,21 \text{ ns}$
27,24 Mb/s	1,74	1	$\frac{1}{499,2 \text{ MHz}} \approx 2 \text{ ns}$	$T_{Bit_{27,24 \text{ Mb/s}}} \approx 64,10 \text{ ns}$

Die maximale Nutzdatenmenge im PSDU-Teil ist entsprechend dem UWB-Standard 127 Byte. Die maximale Nutzdatenmenge entsprechend den Anforderungen der Automatisierungsindustrie ist 48 Byte. Weitere Berechnungen in dieser Arbeit beziehen sich auf maximal 48 Byte.

### 3.1.2 Dauer des UWB-Frames

Tabelle 3.2 stellt die kürzeste mögliche Dauer<sup>3</sup> jeder einzelnen Phase des UWB-Frames, bzgl. des UWB-Standards, für die minimale Datenmenge von 3 Byte Nutzdaten dar.

<sup>3</sup>entspricht der höchsten Datenrate von 27,24 Mb/s mit der geringsten Anzahl an PREAMBLE-Symbolen



**Tabelle 3.2:** Kürzeste Dauer des UWB-Frames für 3 Byte Nutzdaten (minimale angeforderte Datenmenge)

UWB-FRAME	Struktur	PREAMBLE	SFD	PHR	PSDU	
					Nutzdaten	Overhead
		16 PS	8 PS	20 Bit	3 Byte	6 Byte
	Dauer	$16 \cdot T_{PS}$	$8 \cdot T_{PS}$	$20 \cdot T_{Bit_{PHR}}$	$3 \cdot 8 \cdot T_{Bit_{27,24} \text{ Mb/s}}$	$6 \cdot 8 \cdot T_{Bit_{27,24} \text{ Mb/s}}$

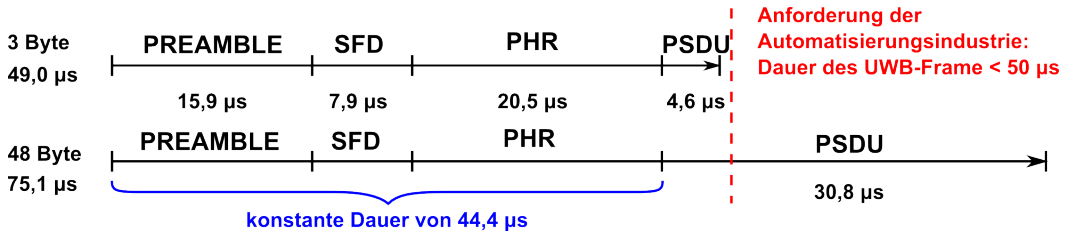
Basierend auf Tabelle 3.2 berechnet sich die Dauer eines UWB-Frames  $T_{UWB-Frame}$  für 3 Byte Nutzdaten wie folgt:

$$\begin{aligned}
 T_{UWB-Frame} &= 16 \cdot T_{PS} + 8 \cdot T_{PS} + 20 \cdot T_{Bit_{PHR}} + (3 + 6) \cdot 8 \cdot T_{Bit_{27,24} \text{ Mb/s}} \\
 &\approx 15,9 \mu s + 7,9 \mu s + 20,5 \mu s + 4,6 \mu s \\
 &\approx 49,0 \mu s
 \end{aligned}$$

Für die maximale Datenmenge von 48 Byte mit 12 Byte Overhead ergibt sich folgende Dauer eines UWB-Frames:

$$\begin{aligned}
 T_{UWB-Frame} &= 16 \cdot T_{PS} + 8 \cdot T_{PS} + 20 \cdot T_{Bit_{PHR}} + (48 + 12) \cdot 8 \cdot T_{Bit_{27,24} \text{ Mb/s}} \\
 &\approx 15,9 \mu s + 7,9 \mu s + 20,5 \mu s + 30,8 \mu s \\
 &\approx 75,1 \mu s
 \end{aligned}$$

Beide Fälle der Übertragungen von 3 Byte und 48 Byte sind in Abbildung 3.4 als Beispiel angegeben.



**Abb. 3.4:** Vergleich der bestmöglichen Übertragungszeiten für die minimale und maximale angeforderte Datenmenge – 3 und 48 Byte Nutzdaten – konform zum UWB-Standard.

Unter idealen Bedingungen, d.h. wenn die Übertragungsdauer gleich der Dauer des UWB-Frames ist, können mittels des UWB-Standards nur kurze Nachrichten bis 5 Byte mit der angeforderten Übertragungszeit von unter 50 µs übertragen werden. Im Fall, dass alle 127 Slaves mit dem Master 48 Byte in beide Richtungen austauschen, ergibt sich folgende Übertragungsdauer:  $75,1 \mu s \cdot 127 \cdot 2 \approx 19 \text{ ms}$ . Diese Übertragungsdauer erfüllt nicht die Anforderung der Automatisierungsindustrie von einer Zykluszeit unter 1 ms.

Bei diesen Berechnungen wurde eine zusätzliche Verarbeitungszeit, die für die Dekodierung der Signale notwendig ist, nicht berücksichtigt. Der Hardware-Reed-

Solomon-Dekoder braucht z.B. ungefähr  $5 \mu s$ , um das Empfangssignal zu verarbeiten. Wird diese Zeit in der Berechnung der Dauer des UWB-Frames mit einbezogen, werden die Anforderungen der Automatisierungsindustrie sogar bei der Übertragung der Datenmenge von 3 Byte nicht mehr erfüllt. Der größte Anteil an der Latenzzeit wird durch die Synchronisationsphase (PREAMBLE+SFD) und durch den PHR verursacht. Die Dauer dieser 3 UWB-Frame Teile ist für alle Datenraten identisch und beträgt  $\approx 44,3 \mu s$ .

Daher müssen die Nutzdaten (einschließlich des notwendigen Overheads) innerhalb von weniger als  $6 \mu s$  übertragen werden, wenn der UWB-Standard nicht optimiert wird. Um die Anforderungen der Automatisierungsindustrie zu erfüllen, müssen alle Teile optimiert werden.

Im nächsten Abschnitt wird untersucht, ob es theoretisch möglich ist, mit den im UWB-Standard definierten Impulsen, Impulswiederholraten und Bit-Dauer, eine Datenrate zu erreichen, welche die Anforderungen der Automatisierungsindustrie erfüllt.

### 3.1.3 Maximale Datenrate

Die minimale von der Automatisierungsindustrie geforderte Datenrate kann folgenderweise berechnet werden. Entsprechend den Anforderungen müssen 127 Sensorknoten jeweils 48 Byte in beide Richtungen mit einem Master-Knoten innerhalb von 1 ms übertragen. Dies entspricht einer Datenrate von  $(127 \cdot 2 \cdot 48 \cdot 8) \text{ b}/1\text{ms} = 97,536 \text{ Mb/s}$ . Dieses Ergebnis wurde ohne Berücksichtigung des Overheads und des PREAMBLE-, des SFD- und des PHR-Teils berechnet.

Unter derselben Annahme wird in diesem Abschnitt die maximale Datenrate berechnet, die für UWB-Signale ohne Verletzung der UWB-Regulierungen theoretisch möglich ist.

Um die Datenrate stark zu erhöhen, kann die gesamte Dauer des Bits-Signals effizienter ausgenutzt werden. Bei der im UWB-Standard festgelegten Struktur eines Bits-Signals wird nur 1 aus 32 *time-slots* mit Impulsen gefüllt – der *time-slot* mit dem *Burst*. Die anderen 31 *time-slots* sind leer (siehe Abbildung 3.3). Auf diese Weise wird nur die Datenmenge von 1 Bit übertragen. Die Impulse des *time-slots* mit dem *Burst* können aber auf allen 32 *time-slots* verteilt werden<sup>4</sup>. Solch eine Verteilung der Impulse ist im UWB-Standard nicht vorgesehen, aber verletzt keine UWB-Regulierungen. Mit einer modifizierten Impulsverteilung bleibt die Anzahl, die Breite und die Amplitude der Impulse unverändert und somit auch die Leistung.

Da die Menge der möglichen Positionen der Impulse innerhalb dieser 32 *time-slots* sehr groß ist, kann auf solche Weise eine deutlich größere Datenmenge kodiert werden, d.h. viel mehr als nur 1 Bit an Information kann in derselben Zeit,  $32 \cdot TS$ , übertragen werden. Dies erhöht die Datenrate wesentlich. Würde man zusätzlich die Polarität der Impulse des *Bursts* berücksichtigen, kann der Informationsgehalt

<sup>4</sup>Nach bestem Wissen wurde bisher kein ähnliches Verfahren veröffentlicht.

weiter gesteigert werden.

Die Anzahl der Impulse  $I_{Burst}$  zusammen mit der Anzahl der *time-slots*  $\#TS$  bestimmt die Anzahl der möglichen Impulspositionen  $n$ :

$$n = \#I_{Burst} \cdot \#TS$$

Jetzt wird berechnet, wie viele Möglichkeiten es gibt, exakt  $k$  Impulse auf  $n$  Positionen zu verteilen:

$$\#P_{(n,k)} = \binom{n}{k} \cdot 2^k \tag{3.5}$$

Die Herleitung der Formel (3.5) ist im Anhang A.3 angegeben.

Wenn die Anzahl an Impulsen pro Bit-Dauer nicht vom UWB-Standard abweichen soll, d.h.  $\#I_{Bit} = \#I_{Burst}$ , ergibt sich eine maximale Bitrate von:

$$Bitrate_{\#I_{Bit}=\#I_{Burst}} = \frac{\lfloor \log_2 (\#P_{(n,\#I)}) \rfloor}{T_{Bit}}$$

Wenn eine variable Anzahl an Impulsen pro Bit-Dauer zugelassen wird, d.h.  $\#I_{Bit} \leq \#I_{Burst}$ , erhöht sich die Anzahl der Verteilungs-Möglichkeiten wesentlich. Dadurch erhöht sich der Informationsgehalt bzw. die Bitrate. Es kann folgende maximale Bitrate erreicht werden:

$$Bitrate_{\#I_{Bit} \leq \#I_{Burst}} = \frac{\lfloor \log_2 \left( \sum_{k=0}^{\#I} \#P_{(n,k)} \right) \rfloor}{T_{Bit}} = \frac{\lfloor \log_2 \left( \sum_{k=0}^{\#I} \binom{n}{k} \cdot 2^k \right) \rfloor}{T_{Bit}}$$

In Tabelle 3.3 sind alle möglichen optimierten Bitraten für die Datenraten aus dem UWB-Standard dargestellt.

**Tabelle 3.3:** Theoretisch erreichbare Bitraten unter der Annahme, dass die Impulse des *Bursts* innerhalb des ganzen Bit-Signals verteilt sind.

Im UWB-Standard festgelegte Parameter				Bitrate	
Datenrate	$\#I_{Burst}$	$n = \#I_{Burst} \cdot \#TS$	$\#TS$	$\#I_{Bit} \leq \#I_{Burst}$	$\#I_{Bit} = \#I_{Burst}$
851 kb/s	16	512	32	155,02 Mb/s	112,13 Mb/s
6,81 Mb/s	2	64	32	101,40 Mb/s	93,60 Mb/s
27,24 Mb/s	1	32	32	93,60 Mb/s	93,60 Mb/s

Aus Tabelle 3.3 ist ersichtlich, dass insgesamt 3 verschiedene Bitraten die Anforderungen der Automatisierungsindustrie von 97,536 Mb/s theoretisch erfüllen können.

Demzufolge sind die Anforderungen der Automatisierungsindustrie, unter Berück-

sichtigung der auferlegten Restriktionen der FCC und der ETSI, prinzipiell erfüllbar. Der UWB-Standard muss dahingehend optimiert werden:

- die maximale im UWB-Standard vorgesehene PSDU-Datenrate von 27,24 Mb/s muss um mehr als 3-fache erhöht werden, um die von der Automatisierungsindustrie geforderte Datenrate von 97,536 Mb/s zu erreichen.
- die konstante Dauer der PREAMBLE-, SFD- und PHR-Teile des UWB-Frames von  $\approx 44 \mu\text{s}$  muss deutlich gekürzt werden.

Im nächsten Abschnitt wird untersucht, ob technische Beschränkungen existieren, die eine Erfüllung der Anforderungen seitens der Automatisierungsindustrie verhindern.

### 3.1.4 Technische Beschränkungen

Wie im vorigen Abschnitt gezeigt wurde, ist die Leistung des Bit-Signals von der Amplitude, Breite und der Anzahl der Impulse im Bit-Signal abhängig. Diese Parameter sind im UWB-Standard festgelegt, um die FCC- und ETSI-Regulierungen bzgl. der Nutzung des UWB-Spektrums zu erfüllen. Diese Parameter müssen in der Hardware-Implementierung der UWB-Knoten eingehalten werden. Diese Anforderung ist eine technische Beschränkung.

Eine weitere technische Anforderung ist die Fähigkeit der UWB-Knoten, die Impuls-Signale von 499,2 MHz digital zu verarbeiten, sowohl beim Sender als auch beim Empfänger.

Im Rahmen dieser Doktorarbeit als Teil der Projekte TANDEM und KUSZ, wurden standardkonforme UWB-Transceiver entwickelt und in der 0,25  $\mu\text{m}$  IHP-Technologie [74] gefertigt. Im TANDEM-Projekt wurde gezeigt, dass mit der IHP-Technologie eine Reichweite von über 15 m erreichbar ist, ohne die UWB-Regulierungen zu verletzen. Im Projekt KUSZ wurde gezeigt, dass jeder einzelne empfangene Impuls mit einer Impulsbreite von 2 ns detektiert und digital vom Basisband verarbeitet werden kann.

Die in der IHP-Technologie gefertigten UWB-Knoten erfüllen demzufolge die technischen Anforderungen bzgl. der UWB-Regulierungen, Reichweite<sup>5</sup> und der digitalen Verarbeitungsgeschwindigkeit<sup>6</sup>.

In Abschnitt 3.1.2 wurde gezeigt, dass die Anforderungen der Automatisierungsindustrie nur durch die Erhöhung der PSDU-Datenrate erreicht werden können. Da jeder einzelne

<sup>5</sup>die von der Automatisierungsindustrie geforderte Reichweite ist: 10 - 20 m.

<sup>6</sup>Im UWB-Standard ist die digitale Signalverarbeitung durch die Taktfrequenz von 499,2 MHz vorgegeben.

2 ns breite Impuls detektiert werden kann, ist die im Abschnitt 3.1.3 vorgeschlagene Änderung der Struktur des Bit-Signals zur Erhöhung der Datenrate möglich. Würde in einer solchen Bit-Struktur die Anzahl der Impulse zusätzlich erhöht, kann dadurch die PSDU-Datenrate weiter gesteigert werden.

Eine zusätzliche Erhöhung der Anzahl der Impulse bzw. der PRF ist nicht standardkonform, aber technisch möglich und entsprechend der UWB-Regulierungen erlaubt, wenn die durchschnittliche Sendeleistung nicht erhöht wird. Um mehr Impulse bei gleichbleibender durchschnittlicher Sendeleistung zu ermöglichen, muss die Amplitude und/oder die Breite der Impulse reduziert werden. Eine Reduzierung der Amplitude wirkt sich negativ auf die Reichweite aus, und wird hier daher nicht angewendet. Eine Reduzierung der Impulsbreite ist nicht mehr standardkonform und erfordert ein erhebliches Redesign des analogen Frontends. Um das gleiche Frontend sowohl für standardkonforme als auch für nicht standardkonforme Anwendungen einsetzen zu können, darf die Impulsbreite nicht verändert werden. Dies ist eine weitere technische Beschränkung.

Im UWB-Standard sind verschiedene Impulswiederholfrequenzen im UWB-Frame vorgesehen. Die PRF für PREAMBLE-Symbole beträgt  $PRF_{PS} = 16,1 \text{ MHz}$  und die PRF für Datenbits beträgt  $PRF_{Bit} = 15,6 \text{ MHz}$ . Dies führt dazu, dass die PRF für das ganze UWB-FRAME  $PRF_{UWB-FRAME}$  mit der Länge der Synchronisationsphase und der Datenmenge variiert. Um sicherzustellen, dass die Sendeleistung optimal ausgenutzt wird, sollte für optimierte PREAMBLE-Symbole PS' folgendes gelten:

$$\begin{aligned}
 PRF_{PS'} &\leq PRF_{Bit} \\
 \Rightarrow \frac{K}{N \cdot T_{Imp}} &\leq 15,6 \text{ MHz} \\
 \Rightarrow \frac{K}{N \cdot \frac{16}{499,2} \text{ MHz}} &\leq \frac{499,2 \text{ MHz}}{32} \\
 \Rightarrow 2 \cdot K &\leq N
 \end{aligned} \tag{3.6}$$

Dieses Kriterium ist strenger als im UWB-Standard vorgesehen, ermöglicht es aber, die erlaubte Sendeleistung optimal auszunutzen. Eine erhöhte Sendeleistung erhöht die Reichweite und auch die Zuverlässigkeit der Kommunikation.

Zusammengefasst wird aufgrund der beschriebenen technischen Beschränkungen, bei den Optimierungen in dieser Doktorarbeit immer von folgenden Parametern ausgegangen:

- die Breite der Impulse ist entsprechend dem UWB-Standard  $\frac{1}{499,2 \text{ MHz}} \approx 2 \text{ ns}$
- die Amplitude der gesendeten Impulse ist konstant
- eine durchschnittliche PRF von  $15,6 \text{ MHz}$  für das gesamte UWB-Frame. Dies gilt insbesondere für PREAMBLE-Symbole, siehe Kriterium (3.6).

### 3.1.5 Zusammenfassung

In Abschnitt 3.1 wurde folgendes gezeigt:

- die Anforderungen der Automatisierungsindustrie können nicht mit dem aktuellen UWB-Standard erfüllt werden
- die Anforderungen der Automatisierungsindustrie können nur durch Änderung der im UWB-Standard vorgesehenen Daten-Struktur erfüllt werden
- die 0,25  $\mu\text{m}$  IHP-Technologie ist geeignet, um die UWB-Knoten zu fertigen ohne weitere Beschränkungen seitens der Technik zu berücksichtigen.

In dieser Doktorarbeit werden sowohl standardkonforme als auch nicht standardkonforme Optimierungen vorgeschlagen und implementiert, um die Übertragungszeit und den Energieverbrauch unter Einhaltung der Zuverlässigkeitsanforderungen zu reduzieren.

Eine veränderte Daten-Struktur ist mit dem UWB-Standard nicht realisierbar: dies ist eine nicht standardkonforme Optimierung. Weitere mögliche nicht standardkonforme Optimierungen, die in dieser Arbeit betrachtet werden, sind z.B. eine Anpassung des MAC-Protokolls (siehe Abschnitt 3.3) und die Veränderung der Synchronisationsphase (siehe Abschnitte 3.2 und 3.3).

Es werden auch Optimierungen vorgenommen, welche konform zum Standard sind, wie z.B. eine effizientere Hardware-Implementierung, Entfernung redundanter Komponenten im System, Entfernung von Flaschenhälsen bzw. Engpässen bei der Datenverarbeitung oder die Nutzung optimierter Algorithmen.

Im nächsten Abschnitt wird gezeigt, wie die Punkt-zu-Punkt Kommunikation des UWB-Standards mit geringen Anpassungen optimiert werden kann.

## 3.2 Optimierung der Punkt-zu-Punkt Datenübertragungen

In diesem Abschnitt wird die Übertragung der Nutzdaten bis zu einer maximalen Nutz-Datenmenge von 48 Byte optimiert. Ziel ist es, die Vorgabe von 50  $\mu\text{s}$  für die Dauer einzelner Punkt-zu-Punkt Verbindungen zu erfüllen. Dafür werden alle Teile des UWB-Frames (PREAMBLE, SFD, PHR und PSDU) einzeln optimiert.

Die Hauptidee in diesem Abschnitt ist es, alle Teile eines UWB-Frames zu kürzen, ohne dabei die Zuverlässigkeit zu verringern und die PRF zu erhöhen. Es wird untersucht, ob die Anzahl an PREAMBLE-Symbolen und die Länge der Impulssequenzen der PREAMBLE-Symbole in der Synchronisationsphase reduziert werden können.

Sowohl bei den Impulssequenzen der PREAMBLE-Symbole im PREAMBLE- und SFD-Teil als auch bei der PS-Sequenz im SFD-Teil werden ternäre Sequenzen

benutzt. Deshalb wird hier zuerst untersucht, ob und wie ternäre Sequenzen optimiert werden können.

### 3.2.1 Optimierung ternärer Sequenzen

Die im UWB-Standard verwendeten ternären Sequenzen sind Impulssequenzen und Sequenzen von PREAMBLE-Symbolen (weiter als PS-Sequenzen bezeichnet). Diese Sequenzen werden zum Synchronisieren verwendet. Eine erfolgreiche Synchronisierung kann nur dann gewährleistet werden, wenn Sequenzen zeitlich eindeutig und zuverlässig detektiert werden können, also gute Autokorrelationseigenschaften besitzen. Das bedeutet, dass solche Sequenzen sich selbst bei verschiedenen Verschiebungen unähnlich sein sollen; was auch bedeutet, sie sind nicht periodisch. Um festzustellen, ob geeignete ternäre Sequenzen mit besseren Eigenschaften existieren, d.h kürzere als im UWB-Standard und mit guten Autokorrelationseigenschaften<sup>7</sup>, wurde im Rahmen dieser Arbeit für jede passende ternäre Sequenz ihre Autokorrelationsgüte berechnet. Folgende Notationen werden verwendet:

- $p = (p_1, \dots, p_n)$  ist eine ternäre Sequenz bestehend aus  $n$  Werten mit  $p_i \in \{-1, 0, 1\}$  für  $i \in (1, \dots, n)$ ;  $n$  ist die Länge der Sequenz
- $p^j$  ist die ternäre Sequenz  $p$ , welche um  $j$  Stellen verschoben wurde. Die Verschiebung entspricht der eines Ringschieberegisters, was bedeutet, dass alle Zahlen um  $j$  Stellen nach rechts verschoben werden und die letzten  $(n - j)$  Werte werden links in gleicher Reihenfolge hinzugefügt:

$$p^j = \begin{cases} p & j = 0 \\ (p_{j+1}, \dots, p_n, p_1, \dots, p_j) & 1 \leq j < n \\ p^j \text{ mod } n & j \geq n \text{ oder } j < 0 \end{cases}$$

- $F_j(p, b)$  ist die Korrelationsfunktion zweier ternärer Sequenzen  $p = (p_1, \dots, p_n)$  und  $b = (b_1, \dots, b_n)$  bei einer Verschiebung der Sequenz  $p$  von  $j$  Stellen.

$$F_j(p, b) = \sum_{i=1}^n (p_i^j \cdot b_i)$$

Die Korrelationsfunktion  $F_j(p, b)$  gibt an wie ähnlich zwei ternäre Sequenzen  $p$  und  $b$  sind, wenn die Sequenz  $p$  um  $j$  Stellen verschoben wurde. Die Korrelationsfunktion  $F_j(p, b)$  der abgesendeten Sequenz  $p$  und der empfangenen Sequenz  $b$  berücksichtigt aber nicht die Tatsache, dass die abgesendeten und die empfangenen Sequenzen sich wegen physikalischer Einflüsse bei der Übertragung, wie z.B. Dämpfung, unterscheiden können. Das heißt,  $F_j(p, b)$  berücksichtigt keine möglichen Übertragungsfehler. Die Übertragungsfehler müssen beim Interpretieren des Empfangssignals berücksichtigt werden.

<sup>7</sup>d.h nicht schlechter als im UWB-Standard vorgesehen

Beispielsweise ist ein schwacher analoger Impuls nicht von einem nicht gesendeten Impuls unterscheidbar. Die Korrelationsfunktion muss zusätzlich gewichtet werden, um solche Fälle zu berücksichtigen. Alle Fälle die auftreten können, sind wie folgt:

Fall 1: Es wird ein leerer Impuls erwartet und auch ein leerer Impuls empfangen, d.h.

$$p_i^j = b_i = 0$$

Diese mathematische Beschreibung entspricht nicht nur dem hier genannten korrekten Fall, sondern auch einem weiteren fehlerhaften Fall, wenn ein leerer Impuls nicht von einem schwachen Impuls unterschieden werden kann. Um diese beiden Möglichkeiten, die zu dieser Beschreibung passen, zu berücksichtigen, wird diese Beschreibung als *'korrekt schwach'* bezeichnet und mit dem Koeffizienten '0' gewichtet.

Fall 2: Es wird ein nicht leerer Impuls erwartet und auch empfangen, d.h.

$$p_i^j = b_i \neq 0$$

Diese mathematische Beschreibung entspricht nur dem Fall, dass ein nicht leerer Impuls mit der richtigen Polarität empfangen wurde. Dieser Fall wird als *'korrekt stark'* bezeichnet und mit '1' gewichtet.

Fall 3: Es wird ein nicht leerer Impuls erwartet und kein Impuls empfangen, d.h.

$$p_i^j \neq 0 \text{ und } b_i = 0$$

Dies ist eine mathematische Beschreibung eines Fehlers, welcher aber auch bei einer korrekten Verschiebung auftreten kann, wenn der Impuls zu schwach ist. Daher wird diese Beschreibung als *'leichter Fehler'* bezeichnet und mit '0' gewichtet.

Fall 4: Es wird kein Impuls erwartet, aber es wird ein Impuls empfangen, d.h.

$$p_i^j = 0 \text{ und } b_i \neq 0$$

Dies ist wieder eine Beschreibung eines Fehlers. Ein Grund des Auftretens solch eines Fehlers könnte ein zu hoher Verstärkungsfaktor beim Empfänger sein. Dieser Fall wird als *'schwerer Fehler'* bezeichnet und mit '-1' gewichtet.

Fall 5: Es wird ein nicht leerer Impuls erwartet und es wird ein nicht leerer Impuls, aber mit invertierter Polarität empfangen, d.h.

$$p_i^j = -b_i \neq 0$$

Dies ist die Beschreibung eines Fehlers, welcher aufgrund einer Phasenverschiebung bzgl. des Senders, Reflektionen oder anderen Störungen auftreten



kann. Dieser Fall wird als 'schwerer Fehler' bezeichnet und mit '-1' gewichtet.

Die zuvor genannten Fälle und deren Gewichtungen sind in Tabelle 3.4 aufgelistet.

**Tabelle 3.4:** Alle Zustände für  $p_i^j$  und  $b_i$  und Gewichtungskoeffizienten

$p_i^j$	$b_i$	$p_i^j \cdot b_i$	Bezeichnung	Gewichtungskoeffizienten
0	0	0	korrekt schwach	0
0	1	0	schwerer Fehler	-1
0	-1	0	schwerer Fehler	-1
1	0	0	leichter Fehler	0
1	1	1	korrekt stark	1
1	-1	-1	schwerer Fehler	-1
-1	0	0	leichter Fehler	0
-1	1	-1	schwerer Fehler	-1
-1	-1	1	korrekt stark	1

Die Gewichtungskoeffizienten aus Tabelle 3.4 können als eine Gewichtungsfunktion  $f(p_i^k, b_i)$  folgenderweise beschrieben werden:

$$f(p_i^j, b_i) = f(x, y) = \begin{cases} -y & x = -1 \\ -y^2 & x = 0 \\ y & x = 1 \end{cases} \Rightarrow f(x, y) = x \cdot y + x^2 \cdot y^2 - y^2 \quad \forall x, y \in \{-1, 0, 1\}$$

Daraus ergibt sich die gewichtete Korrelationsfunktion  $F_j'(p, b)$  für zwei ternäre Sequenzen  $p$  und  $b$ :

$$F_j'(p, b) = \sum_{i=1}^n f(p_i^j, b_i)$$

Die Autokorrelationsfunktion ist die Korrelationsfunktion  $F_j'(p, b)$  für den Fall, dass Sequenz  $p$  gleich der Sequenz  $b$  (bezeichnet mit  $F_j'(p, p)$ ) ist. Die Autokorrelationsgüte  $A_p$  für eine ternäre Sequenz  $p$ , ist ein Maß zur Bewertung der Autokorrelationseigenschaft von  $p$ .  $A_p$  ist der Abstand zwischen dem höchsten Peak der Autokorrelationsfunktion<sup>8</sup> und dem zweithöchsten Peak der Autokorrelationsfunktion<sup>9</sup>. Daher wird  $A_p$  wie folgt berechnet:

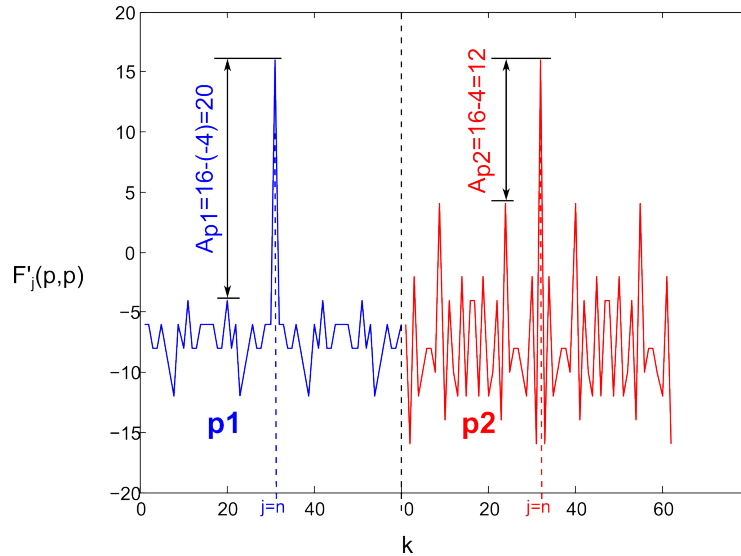
$$A_p = F_n'(p, p) - \max_{j=1}^{n-1} F_j'(p, p)$$

In Abbildung 3.5 ist die Autokorrelationsgüte für zwei gleichlange ternäre Sequenzen  $p_1$  und  $p_2$  grafisch dargestellt. Sowohl für  $p_1$  als auch  $p_2$  gilt:  $\#1 + \#(-1) = 16$ , aber die Reihenfolge der Einträge, d.h. Werte, ist zufällig gewählt. Je höher die Autokorrelationsgüte einer ternären Sequenz ist, desto höher ist die Wahrscheinlichkeit, diese Sequenz auch bei vorhandenen Fehlern korrekt zu erkennen. Aus

<sup>8</sup>Der höchste Peak der Autokorrelationsfunktion entspricht dem Fall, wenn  $p^j = p$ , wie z.B. für  $j = n$

<sup>9</sup>Der zweithöchste Peak der Autokorrelationsfunktion entspricht dem Maximalwert von  $F_j'(p, p)$  bei einer Verschiebung der Sequenz  $p$  um  $j$  Stellen, mit  $0 < j < n$ .

Abbildung 3.5 folgt, dass die Reihenfolge der nicht leeren Einträge einen Einfluss auf die Autokorrelationsgüte hat.



**Abb. 3.5:** Grafische Darstellung der Autokorrelationsgüte für zwei ternäre Sequenzen  $p1$  und  $p2$ .

Die blau markierte Sequenz ist:

$p1 = (1, 1, 0, 0, 1, 0, 0, -1, -1, -1, 1, -1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, -1, 1, 0, 1, 0, 0, 0, 0)$ .

Der höchste Peak ist  $F'_n(p1, p1) = 16$  und der zweithöchste Peak ist  $\max_{j=1}^{n-1} F'_j(p1, p1) = -4$ :

$A_{p1} = 16 - (-4) = 20$ .

Die rot markierte Sequenz ist:

$b = (0, 1, 0, -1, 0, 0, -1, 1, 0, 1, 0, 1, 0, 0, -1, 1, -1, 1, 0, 1, 0, -1, 1, 0, 1, 0, 0, 0, 0, -1, 1)$ .

Der höchste Peak ist  $F'_n(p2, p2) = 16$  und der zweithöchste Peak ist  $\max_{j=1}^{n-1} F'_j(p2, p2) = 4$ :

$A_{p2} = 16 - 4 = 12$ .

Da  $A_{p1} > A_{p2}$ , besitzt die Sequenz  $p1$  bessere Autokorrelationseigenschaften als die Sequenz  $p2$ .

Im Rahmen dieser Arbeit wurde geprüft, ob alle Impulssequenzen der PREAMBLE-Symbole aus dem UWB-Standard bzgl. ihrer  $A_p$  optimal sind. Aufgrund der Komplexität des mathematischen Beweises wurde eine *Brute-Force* Methode angewendet, d.h. hier wurden alle  $\sum_{l=1}^n 3^l$  möglichen Impulssequenzen mit einer Länge bis  $n = 31$  getestet.

Um die Autokorrelationsgüte für alle möglichen ternären Sequenzen bis einer Länge von  $n = 31$  zu berechnen, wurden die folgenden 3 Ansätze verfolgt:

1. **Matlab-Skript:** Es wurde ein Matlab-Skript geschrieben, welches alle ternären Sequenzen und deren Autokorrelationsgüten berechnet. Die Berechnung wurde abgebrochen, da sie mehrere Jahre benötigen würde.
2. **C-Code:** Es wurde ein C-Programm geschrieben, welches effizienter als das Matlab-Skript alle Berechnungen durchführen konnte. Auch dieser Ansatz wurde abgebrochen, da die Berechnung mehrere Monate benötigen würde.
3. **Hardware-Implementierung auf dem FPGA:** Es wurde für jede Sequenzlänge  $n$  mit  $n \leq 31$  ein VHDL-Code für einen Virtex5 FPGA[75] programmiert.

miert. Das FPGA wurde mit einer Taktfrequenz von 100 MHz betrieben. Die Berechnung dauerte nur einige Tage.

Im Anhang A sind ausführlichere Angaben und Beispiele zu den gefundenen ternären Sequenzen angegeben.

Die Ergebnisse der Berechnungen sind in Tabelle 3.5 dargestellt. Tabelle 3.5 zeigt die optimalen  $A_p$  für alle ternären Sequenzen bis zur Länge  $n = 31$  mit jeweils  $k$  nicht leeren Stellen an. Da die Anzahl  $k$  der nicht leeren Stellen innerhalb von  $n$  Stellen nicht größer als  $n$  sein kann, wurde  $A_p$  nur für  $k \leq n$  berechnet (siehe schwarzer Hintergrund in Tabelle 3.5).

**Tabelle 3.5:** Optimale Autokorrelationsgütern ternärer Sequenzen der Länge  $n = 31$  mit  $k$  nicht leeren Stellen. Fälle mit  $2 \cdot k > n$  sind auf weißem und Fälle mit  $2 \cdot k \leq n$ , welche das Kriterium (3.6) der technischen Beschränkungen erfüllen, auf grauem Hintergrund dargestellt.

$n \backslash k$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
1	0																															
2		4																														
3			6																													
4				4																												
5					4																											
6						4																										
7							8																									
8								8																								
9									8																							
10										8																						
11											8																					
12												8																				
13													8																			
14														8																		
15															8																	
16																8																
17																	8															
18																		8														
19																			8													
20																				8												
21																					8											
22																						8										
23																							8									
24																								8								
25																									8							
26																										8						
27																											8					
28																												8				
29																													8			
30																														8		
31																															8	

In diesem Abschnitt wurde ermittelt, welche Autokorrelationsgütern für ternäre Sequenzen einer bestimmte Länge  $n$  mit  $k$  nicht leeren Einträgen optimal sind. Diese Information wird in den nächsten Abschnitten verwendet, um die Synchronisationsphase (d.h. PREAMBLE- und SFD-Teil des UWB-Frames) zu optimieren.

### 3.2.2 Optimierung der PREAMBLE-Symbole in der Synchronisationsphase

Der PREAMBLE-Teil kann entsprechend dem UWB-Standard aus 16, 64, 1024 oder 4096 PREAMBLE-Symbolen  $P$  bestehen. Jedes PREAMBLE-Symbol  $P$  ist eine

Abfolge von Impulsen, welche mittels einer ternären Impulssequenz, bestehend aus Impulsen '0', '1' und '-1', generiert wird. Die Länge der Impulssequenz ist  $n = 31$ .

Tabelle 3.6 zeigt Impulssequenzen von PREAMBLE-Symbolen entsprechend dem UWB-Standard und ihre Autokorrelationsgüten  $A_p$ . Mit  $\#I_{PS}$  wird die Anzahl an nicht leeren Impulsen der Impulssequenz jedes PREAMBLE-Symbols bezeichnet. Es sei bemerkt, dass die PREAMBLE-Symbole aus dem UWB-Standard nicht das Kriterium  $2 \cdot \#I_{PS} \leq n$  (siehe Kriterium (3.6)) für optimierte PREAMBLE-Symbole erfüllen, denn es gilt:

$$2 \cdot \#I_{PS} = 32 > n = 31$$

**Tabelle 3.6:** Autokorrelationsgüte für Impulssequenzen der PREAMBLE-Symbole, entnommen aus dem UWB-Standard [37].

Im UWB-Standard festgelegte Impulssequenz für PREAMBLE-Symbole	$n$	$\#I_{PS}$	$A_p$
-1 0 0 0 0 1 0-1 0 1 1 1 0 1-1 0 0 0 1-1 1 1 1 0 0-1 1 0-1 0 0	31	16	24
0 1 0 1-1 0 1 0 1 0 0 0-1 1 1 0-1 1-1-1-1 0 0 1 0 0 1 1 0 0 0	31	16	24
-1 1 0 1 1 0 0 0-1 1-1 1 1 0 0 1 1 0 1 0 0-1 0 0 0 0-1 0 1 0-1	31	16	24
0 0 0 0 1-1 0 0-1 0 0-1 1 1 1 1 0 1-1 1 0 0 0 1 0-1 0 1 1 0-1	31	16	24
-1 0 1-1 0 0 1 1 1-1 1 0 0 0-1 1 0 1 1 1 0-1 0 1 0 0 0 0-1 0 0	31	16	24
1 1 0 0 1 0 0-1-1-1 1-1 0 1 1-1 0 0 0 1 0 1 0-1 1 0 1 0 0 0 0	31	16	24
1 0 0 0 0 1-1 0 1 0 1 0 0 1 0 0 0 1 0 1 1-1-1-1 0-1 1 0 0-1 1	31	16	24
0 1 0 0-1 0-1 0 1 1 0 0 0 0-1-1 1 0 0-1 1 0 1 1-1 1 1 0 1 0 0	31	16	24

Vergleicht man die  $A_p$  aus Tabelle 3.6, mit den optimierten  $A_p$  aus Tabelle 3.5 ist ersichtlich, dass die ternären Impulssequenzen der PREAMBLE-Symbole aus dem UWB-Standard für die Länge  $n = 31$  und  $k = \#I_{PS} = 16$  optimal sind. Deren Autokorrelationsgüte  $A_p = 24$  ist in diesem Fall maximal.

Gleichlange ternäre Sequenzen mit einer höheren  $A_p$ , erfüllen auch nicht das Kriterium (3.6). Außerdem gibt es keine ternären Sequenzen mit:

- einer Länge von  $n \leq 31$
- einer Autokorrelationsgüte  $A_p \geq 24$
- die das Kriterium (3.6) erfüllen, d.h  $2 \cdot \#I_{PS} \leq n$

Die Länge der ternären Sequenzen für PREAMBLE-Symbole im PREAMBLE-Teil lässt sich somit nicht kürzen, ohne die Zuverlässigkeit zu beeinträchtigen oder die PRF zu überschreiten. Es bleibt zu untersuchen, ob der PREAMBLE-Teil in Kombination mit dem SFD-Teil optimiert werden kann.

Der SFD-Teil besteht aus positiven, negativen oder leeren PREAMBLE-Symbolen. Insgesamt gibt es 8 PREAMBLE-Symbole (siehe Abbildung 3.1). Da die SFD-Sequenz ebenfalls eine ternäre Sequenz ist, kann die Tabelle 3.5 zur Optimierung genutzt werden. Einerseits reduziert sich die Länge des SFD-Teils mit der Kürzung

der PREAMBLE-Symbol-Länge. Andererseits kann die SFD-Sequenz, d.h. die Anzahl an PS im SFD-Teil selbst, reduziert werden.

Es wurde zuvor gezeigt, dass eine Optimierung des PREAMBLE-Teils die UWB-Regulierungen verletzen würde, da die  $PRF_{PREAMBLE}$  höher als die zulässige durchschnittliche PRF wäre. Aufgrund der leeren PREAMBLE-Symbole im SFD-Teil, reduziert sich jedoch die durchschnittliche PRF der Synchronisationsphase  $PRF_{SYNC}$ . Somit ist es möglich, den PREAMBLE-Teil weiter zu optimieren.

Im UWB-Standard können bis zu 4096 PREAMBLE-Symbole ausgesendet werden. In diesem Fall hat der SFD-Teil wenig Auswirkung auf die  $PRF_{SYNC}$ . Hier bei wenigen PREAMBLE-Symbolen, also bis zu 16 PREAMBLE-Symbolen ist der Einfluss größer und muss berücksichtigt werden.

Sei  $x$  die Anzahl an nicht leeren PREAMBLE-Symbolen im SFD-Teil,  $y$  die Gesamtanzahl an PREAMBLE-Symbolen des SFD-Teils und  $\#PS$  die Anzahl an PREAMBLE-Symbolen im PREAMBLE-Teil, dann berechnet sich die  $PRF_{SYNC}$  wie folgt:

$$PRF_{SYNC} = \frac{\#IPS \cdot (\#PS + x)}{n \cdot T_{IMP} \cdot (\#PS + y)} = \frac{\#IPS \cdot (\#PS + x)}{n \cdot 16 \cdot (\#PS + y)} \cdot 499,2 \text{ MHz}$$

Aus der Bedingung

$$PRF_{SYNC} \leq 15,6 \text{ MHz} = \frac{499,2 \text{ MHz}}{32}$$

folgt die Ungleichung:

$$\#IPS \leq \frac{n}{2} \cdot \frac{\#PS + y}{\#PS + x} \quad (3.7)$$

Die  $A_p$  der im UWB-Standard festgelegten SFD-Sequenz, mit  $n = 8$  und  $k = x = 4$ , ist 6, siehe Tabelle 3.5. Aus Tabelle 3.5 folgt die Existenz einer effizienteren SFD-Sequenz, welche kürzer ist und dennoch nicht die Zuverlässigkeit der Erkennung reduziert.

Eine kürzere SFD-Sequenz reduziert die Synchronisationsphase. Dafür kann aber die Länge der PREAMBLE-Symbole nicht gleichzeitig reduziert werden da auch die Anzahl an leeren PREAMBLE-Symbolen abnimmt. Idealerweise sollte die optimierte SFD-Sequenz mehr leere PREAMBLE-Symbole beinhalten und gleichzeitig kürzer sein als die originale SFD-Sequenz. Anhand der Tabelle 3.5 und der Formel (3.7) wurden die Tabellen 3.7 und 3.8 berechnet. In Tabelle 3.7 sind für SFD-Sequenzlängen von 5 bis 8 die maximal möglichen Anzahl der Impulse  $\#IPS$  je PREAMBLE-Symbol angegeben. Es wurden nur PREAMBLE-Symbole mit  $A_p \geq 24$  berücksichtigt, um mindestens die gleiche Zuverlässigkeit wie im UWB-Standard zu gewährleisten, siehe dazu Tabelle 3.6.

Für  $\#PS = 16$  konnte der PREAMBLE-Teil nicht optimiert werden. Der SFD-Teil konnte jedoch von 8 PS auf 3 PS bzw. 37,5 % reduziert werden. Diese Überlegung kann für alle  $\#PS \in \{1, \dots, 16\}$  fortgeführt werden. Die Ergebnisse sind in

Tabelle 3.8 dargestellt.

**Tabelle 3.7:** Maximal mögliche Anzahl an Impulsen  $\#IPS$  je SFD-Sequenzlänge  $y$  und Impulssequenzlänge  $n$  der PS für  $\#PS = 16$  im PREAMBLE-Teil.

Die in rot markierten Einträge stellen die jeweils optimale Anzahl an Impulsen  $\#IPS$  für eine bestimmte PREAMBLE-Symbol-Länge dar.

Der in grün markierte Eintrag ist die kürzeste Dauer der optimierten Synchronisationsphase  $T_{SYNC_{opt}}$ , welche alle Anforderungen erfüllt.

$n \backslash y$	8	7	6	5	$T_{SYNC_{opt}}$
31	18	17	17	<b>16</b>	<b>20865 ns</b>
30	18	17	<b>16</b>	–	21154 ns
29	17	<b>16</b>	–	–	21378 ns

**Tabelle 3.8:** Maximal mögliche Anzahl an Impulsen  $\#IPS$  je SFD-Sequenzlänge  $y$ , PREAMBLE-Symbol-Länge  $n$  und Anzahl an PREAMBLE-Symbolen  $\#PS$  im PREAMBLE-Teil.

$\#PS$	$n$	$\#IPS$	$y$	$T_{SYNC_{opt}}$	
16	31	16	5	20865 ns	–12, 50%
15	29	16	6	19519 ns	–18, 15%
14	29	16	6	18590 ns	–22, 04%
13	29	16	6	17660 ns	–25, 94%
12	29	16	6	16731 ns	–29, 84%
11	30	16	5	15385 ns	–35, 48%
10	30	16	5	14423 ns	–39, 52%
9	30	16	5	13423 ns	–43, 71%
8	30	16	5	12500 ns	–47, 58%
7	30	16	5	11538 ns	–51, 61%
6	30	16	5	10577 ns	–55, 64%
5	29	16	5	9295 ns	–61, 02%
4	29	16	5	8365 ns	–64, 92%
3	29	16	5	7436 ns	–68, 82%
2	29	16	5	6506 ns	–72, 72%
1	29	17	5	5577 ns	–76, 61%

Beispielsweise kann für  $\#PS = 15$  die Impulssequenzlänge der PS um 6,5% und die SFD-Sequenzlänge um 25% gekürzt werden. Insgesamt wird die Synchronisationszeit um 18,15 % reduziert, ohne die Zuverlässigkeit zu reduzieren.

Wenn die benötigten Verstärkungseinstellungen zwischen zwei Knoten annähernd bekannt sind, kann die Anzahl der PREAMBLE-Symbole im PREAMBLE-Teil  $\#PS$  noch weiter bis auf 4 reduziert werden, ohne die Zuverlässigkeit zu beeinträchtigen. Dies wurde experimentell bestätigt.

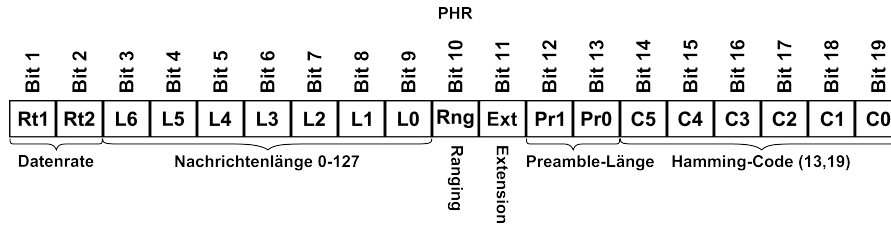
Nachdem die Synchronisationsphase optimiert wurde, folgt nun die Optimierung der Datenübertragungsphase.

### 3.2.3 Optimierung der Datenübertragungsphase

Die Datenübertragungsphase besteht aus PHR- und PSDU-Teil. Obwohl beide dieselbe Datenübertragungsstruktur aufweisen, müssen beide separat betrachtet werden. Denn die Daten im PHR-Teil sind keine Nutzdaten und werden auch anders gegen Fehler geschützt als im PSDU-Teil. Der PSDU-Teil kann nur korrekt interpretiert werden, wenn der PHR-Teil korrekt empfangen wurde.

#### PHR

Jedes Bit des PHR-Teils beinhaltet wichtige Informationen über die darauffolgenden Daten. Abbildung 3.6 zeigt die Struktur des PHR detailliert.



**Abb. 3.6:** PHR nach IEEE 802.15.4a  
 Bit 1-2: die zu verwendende Datenrate  
 Bit 3-9: die Nachrichtenlänge ohne FEC-Overhead 0-127 Byte.  
 Bit 10-11: Zusatzfunktionen wie Ranging oder proprietäre Erweiterungen  
 Bit 12-13: Anzahl an PREAMBLE-Symbolen während des PREAMBLE-Teils  
 Bit 14-19: Overhead vom FEC Hamming-Code (13,19)

Der hier benutzte FEC ist der Hamming-Code(13,19). Dieser Hamming-Code(13,19) generiert für 13 Bit Nutzdaten 6 Bit an Overhead also insgesamt 19 Bit. Der Hamming-Code(13,19) kann 2 Fehler detektieren und 1 Fehler korrigieren.

Der PHR wird standardmäßig mit 851 kb/s übertragen, unabhängig von der genutzten Datenrate. Daher benötigt der PHR 20513 ns und wirkt sich negativ auf die Übertragungsdauer aus.

Da aufgrund der Anforderungen immer mit der höchsten Datenrate von 27,24 Mb/s gearbeitet werden muss, ist es sinnvoll, auch den PHR mit dieser Datenrate zu übertragen. Ein weiterer Vorteil ist die Reduktion der Komplexität der physikalischen Schnittstelle, da anstatt 3 Datenraten – 27,24 Mb/s, 6,81 Mb/s und 851 kb/s – nur eine Datenrate von 27,24 Mb/s unterstützt werden muss. Dies führt zu einer kleineren Chipfläche sowie einem geringeren Energieverbrauch. Die Dauer des PHR-Teils bei der Datenrate 27,24 Mb/s wäre:

$$T_{PHR} = 20 \cdot T_{Bit_{27,24 \text{ Mb/s}}} = 1282 \text{ ns}$$

Dies entspricht einer Reduktion der Dauer des PHR-Teils von 93,75%.

### PSDU

Die PSDU (Physical Layer Service Data Unit) ist der Teil des UWB-Frames, welcher die Nutzdaten beinhaltet. Die Dauer der PSDU kann durch Komprimierung der Nutzdaten reduziert werden. Der Grad der Kompression ist von den Nutzdaten abhängig. Die Komprimierung (bzw. Dekomprimierung) benötigt eine zusätzliche Verarbeitungszeit, was die Latenzzeit erhöht. Aus diesem Grund, wurde die Optimierung des PSDU-Teils mittels Daten-Komprimierung im Rahmen dieser Arbeit nicht betrachtet.

Bei einer standardkonformen UWB-Übertragung werden nicht nur die Nutzdaten sondern auch der Overhead übertragen. Der Overhead besteht aus Daten, welche zusätzliche Informationen über die Nutzdaten enthalten und zur Fehlerkorrektur benötigt werden. Es werden 3 Arten von Fehlerkorrekturen im UWB-Standard eingesetzt:

- Hamming-Code: nur für PHR

- REED-Solomon-Code: nur für PSDU
- Viterbi: für PHR und PSDU

Mit dem Reed-Solomon-Verfahren aus dem UWB-Standard ist es möglich, bis zu 4 einzelne fehlerhafte Bits und maximal 24 fehlerhafte Bits in Folge zu korrigieren. Dies gilt für eine Datenmenge bzw. Datenblock von bis zu 41 Byte, daher gehört der REED-Solomon-Code auch zu den Blockcodes. Die Datenmenge der Nutzdaten wird in Blöcken von bis zu 41 Byte aufgeteilt. Für jeden Block wird ein Overhead von 6 Byte generiert.

Die minimal geforderte Menge der Nutzdaten von 3 Byte kann als 1 Block dargestellt werden und benötigt einen Overhead von 6 Byte.

Die maximal geforderte Menge der Nutzdaten von 48 Byte kann als 2 Blöcke dargestellt werden: ein Block mit 41 Byte und ein Block mit 7 Byte. Für jeden Block sind 6 Byte an Overhead vorgesehen, d.h. der gesamte Overhead für 48 Byte Nutzdaten beträgt demnach  $2 \cdot 6 \text{ Byte} = 12 \text{ Byte}$ . Dies verursacht eine zusätzliche Übertragungszeit von  $3077 \text{ ns} \approx 64,1 \text{ ns} \cdot 8 \cdot 6$  für 3 Byte Nutzdaten und  $6154 \text{ ns} \approx 64,1 \text{ ns} \cdot 8 \cdot 12$  für 48 Byte an Nutzdaten.

Da der Reed-Solomon-Code nur von der Anzahl an Blöcken und nicht direkt von der Nachrichtenlänge abhängt, ist dieser für kleinere Datenmengen ungünstig, nicht linear mit der Nachrichtenlänge skaliert, ist dieser für kurze Nachrichten nicht geeignet. Ein Kompromiss zwischen Zuverlässigkeit und der Länge des Overheads muss gefunden werden.

Eine mögliche Alternative zum REED-Solomon-Code ist es, den Hamming-Code (13,19) zu verwenden, welcher auch im PHR-Teil entsprechend des UWB-Standards benutzt werden soll. Der Vorteil ist, dass im Fall der Nutzung des Hamming-Codes keine weitere Hardware benötigt wird, da dieser schon vorhanden ist. Für Nutzdaten bis zu 13 Byte ist der Overhead des Hamming-Codes kürzer als der des Reed-Solomon-Codes, siehe Abbildung 3.7. Bei einer Datenmenge von 48 Byte kann der Hamming-Code bis zu 11 einzelne Fehler korrigieren, aber keine aufeinanderfolgenden Fehler. Da nur bei korrekter Nachrichtenlänge aus dem PHR-Datenfeld die Nutzdaten dekodiert werden können, wird die Zuverlässigkeit des Gesamtsystems durch die Erweiterung der Hamming-Kodierung vom PHR auf die PSDU nicht gemindert. Die Länge des Overheads für eine Hamming-Kodierung von  $48 \cdot 8 \text{ bit} = 384 \text{ bit}$  ist:

$$\left\lceil \frac{384}{13} \right\rceil \cdot 6 \text{ bit} = 180 \text{ bit} = 22,5 \text{ Byte}$$

Dieser Overhead des Hamming-Codes von 22,5 Byte ist in diesem Fall wesentlich größer als der vom Reed-Solomon-Code mit 12 Byte.

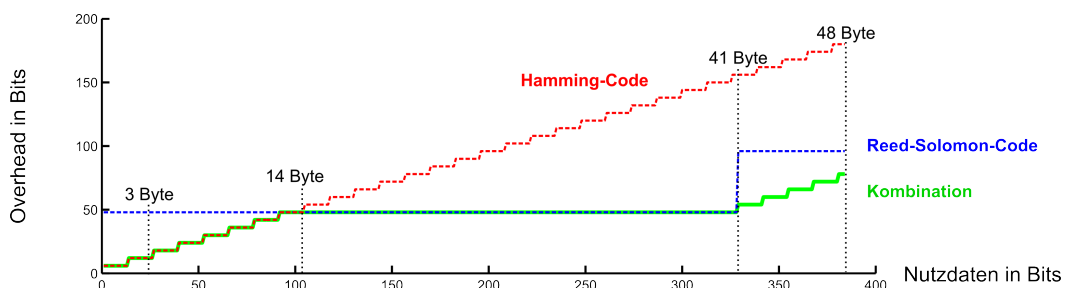
Eine Reduktion des Overheads kann durch eine Kombination beider Verfahren –



Hamming-Code und Reed-Solomon-Code – erreicht werden. Dabei werden 3 Fälle unterschieden:

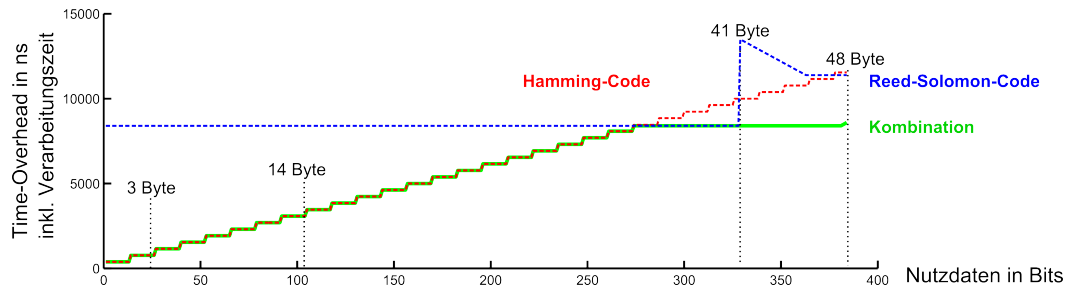
- Nachrichtenlänge kleiner als 14 Byte: Verwendung des Hamming-Codes.
- Nachrichtenlänge ab 14 bis 41 Byte: Verwendung des Reed-Solomon-Codes.
- Nachrichtenlänge ab 42 bis 48 Byte: Verwendung des Reed-Solomon-Codes auf die ersten 41 Byte und Verwendung des Hamming-Codes auf die übrigen Byte.

Die Abbildung 3.7 stellt den Hamming-Code, den Reed-Solomon-Code und deren Kombination dar. Im Fall für einen Nachrichtenlänge von 48 Byte kann somit der Overhead um bis zu 18,75% bzw. 1154 ns reduziert werden.



**Abb. 3.7:** Overhead in Bits in Abhängigkeit der verwendeten Fehlerkorrektur-Verfahren  
 In rot ist der durch den Hamming-Code verursachte Overhead in Bits dargestellt.  
 In blau ist der durch den Reed-Solomon-Code verursachte Overhead in Bits dargestellt.  
 In grün ist der durch die Kombination beider Verfahren – Hamming-Code und Reed-Solomon-Code – verursachte Overhead in Bits dargestellt.

Bisher wurde die Verarbeitungszeit der Dekodierungsverfahren bei der Dauer des UWB-Frames nicht berücksichtigt. Aufgrund der hohen Komplexität des Reed-Solomon-Dekoders benötigt dieser, für jeden Block,  $3,12 \mu s$  für die Dekodierung und weitere  $2,2 \mu s$ , um die dekodierten Daten in den RAM zu schreiben. Die gesamte Verzögerung bei der Dekodierung beläuft sich demnach auf zusätzliche  $5,32 \mu s$  je Block. Da der erste Block verarbeitet werden kann, während der zweite Block empfangen wird, ist der zeitliche Overhead des Reed-Solomon-Codes nicht monoton wachsend. Da der Hamming-Dekoder wesentlich einfacher aufgebaut ist, kann dieser quasi verzögerungsfrei arbeiten. In Abbildung 3.8 ist der zeitliche Overhead, d.h. Übertragungsdauer der zusätzlichen Overhead-Bits und die Verarbeitungszeit, für verschiedene Datenlängen und für eine Datenrate von 27,24 Mb/s angegeben.



**Abb. 3.8:** Zeitlicher Overhead in ns inkl. Verarbeitungszeit für verschiedene Fehlerkorrektur-Verfahren. In rot ist der zeitliche Overhead des Hamming-Codes dargestellt. In blau ist der zeitliche Overhead des Reed-Solomon-Codes, inkl. einer Verarbeitungszeit von  $5,32 \mu s$ , dargestellt. In grün ist der zeitliche Overhead der Kombination beider Verfahren inkl. Verarbeitungszeit dargestellt.

Wie in Abbildung 3.8 gezeigt ist, sind Einsparungen im Bereich von 0-95%, je nach Datenmenge, möglich. Im Fall von einer Nachrichtenlänge mit 48 Byte Nutzdaten ist der zeitliche Overhead wie folgt:

- Reed-Solomon-Code (48 Byte): 11482 ns
- Hamming-Code (48 Byte): 11538 ns
- Kombination von Reed-Solomon-Code und Hamming-Code (48 Byte): 8405 ns, was den Zeitlichen Overhead um 27% reduziert.

Im UWB-Standard ist ein weiteres Fehlerkorrektur-Verfahren – Viterbi bzw. Faltungscodes  $1/2$  – vorgesehen. Im Gegensatz zum Hamming-Code und Reed-Solomon-Code ist die Verwendung des Viterbi-Codes zum Empfangen der Nachricht optional, muss aber dennoch mitgesendet werden. Der Viterbi-Code wird als Ergänzung zum Reed-Solomon-Code verwendet da der Reed-Solomon-Code für Burst-Fehler optimiert ist und nur wenige einzelne Fehler (bis zu 4 Bitfehler je Block) korrigieren kann. Der Viterbi-Code kann mehrere einzelne Bit-Fehler, aber keine Burst-Fehler korrigieren. Außerdem können sich Fehler fortpflanzen, d.h. mehrere Bit-Fehler in Folge können dazu führen, dass korrekte Bits als falsch erkannt werden. Der Overhead des Viterbi-Code entspricht der Datenmenge, d.h. der Nutzdatenmenge plus der Overhead-Datenmenge (Reed-Solomon-Code und/oder Hamming-Code).

Entsprechend UWB-Standard werden in jedem Daten-Symbol 2 Bit an Informationen übertragen:

- Burst-Position (OOK<sup>10</sup>): 0 oder 1
- Burst-Polarität (BPSK<sup>11</sup>): 0 oder 1

Die Datenmenge – Nutzdaten, Reed-Solomon-Code Overhead und Hamming-Code Overhead – werden mittels der Positionierung des Bursts (OOK) kodiert und

<sup>10</sup> engl. *On-Off Keying*

<sup>11</sup> engl. *Binary Phase-Shift Keying*

der Overhead des Viterbi-Codes wird in der Polarität des Bursts (BPSK) kodiert.

Die Polarität des Bursts kann nur von kohärenten Empfängern detektiert und ausgewertet werden. Nicht-Kohärente Empfänger bzw. Energiedetektoren können nur die Burst-Position detektieren. Aus diesem Grund wird:

- die Burst-Polarität nur dazu genutzt redundante Daten zu übertragen.
- der Viterbi-Dekoder optional verwendet

Wird die Polarität des Bursts dazu genutzt Nutzdaten zu übertragen, kann die Datenrate verdoppelt werden. Aufgrund der höheren Datenrate und der nicht mehr vorhandenen Verzögerung des Viterbi-Dekoders reduziert sich die gesamte Latenzzeit. Weitere Vorteile sind eine Reduzierung der Chipfläche und eine Reduzierung des Energieverbrauchs.

Als Nachteil ergibt sich eine reduzierte Zuverlässigkeit. Dieser Nachteil kann durch den Einsatz anderer Kodierungsverfahren gemindert werden.

Im nächsten Abschnitt werden alle zuvor vorgestellten Optimierungen sowohl für die Synchronisationsphase als auch für die Datenübertragungsphase zusammengefasst.

### 3.2.4 Ergebnisse

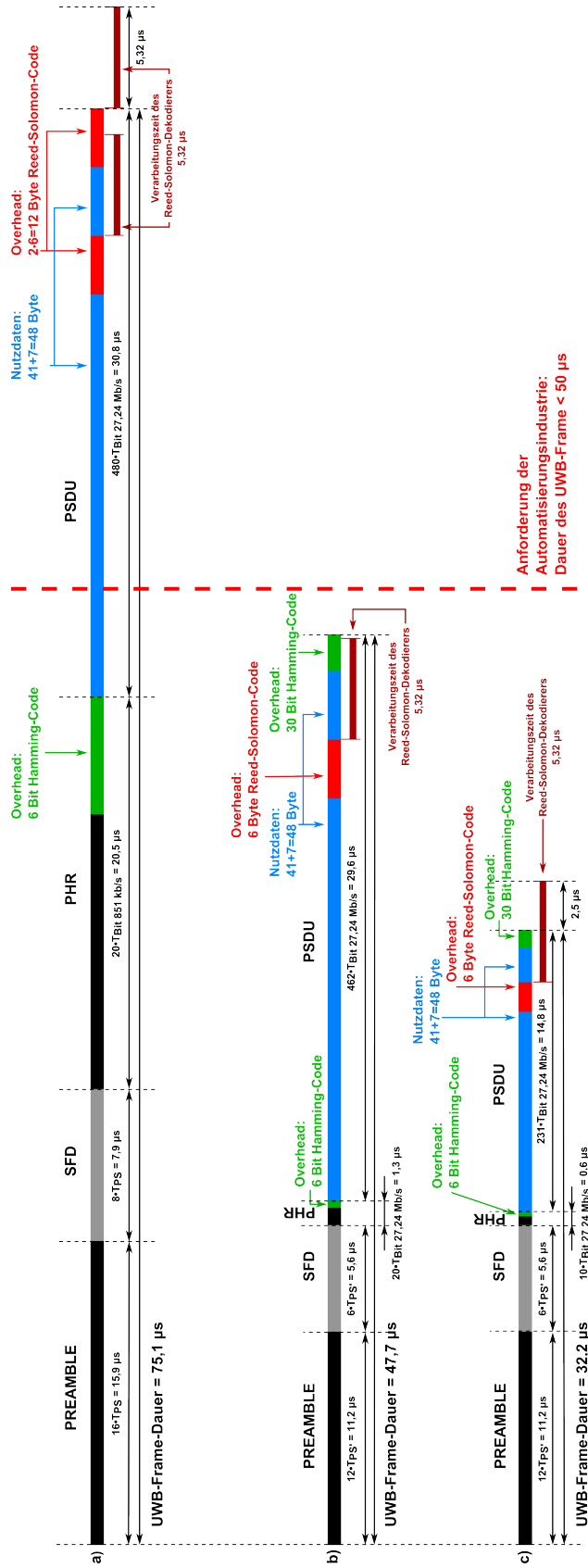
Bisher wurden die UWB-Frame-Teile separat untersucht und optimiert. Werden alle Optimierungen zusammen angewendet, kann die UWB-Frame-Dauer für die maximal angeforderten Datenmenge von 48 Byte von 75127 ns auf 32179 ns, d.h. um 57,16% gekürzt werden. Betrachtet man die UWB-Frame-Dauer inkl. Verarbeitungszeit, so konnte die Dauer von 80447 ns auf 34743 ns, d.h. um 56,81% reduziert werden. Die Dauer der Synchronisationsphase ist bei beiden Optimierungen gleich und beträgt 16731 ns.

Abbildung 3.9 stellt die Dauer des UWB-Frames unter Anwendung der Optimierungen grafisch dar. Es werden alle UWB-Frame-Teile, der Hamming-Code Overhead, der Reed-Solomon-Code Overhead, die Nutzdaten und die Verarbeitungszeit des Reed-Solomon-Codes dargestellt.

Die Reed-Solomon-Kodierung und die Hamming-Kodierung haben auf die gesamte Übertragungsdauer keinen Einfluss, wenn die Kodierungen mittels Hardware parallel zur Synchronisationsphase durchgeführt werden. Die Kodierungszeit ist erheblich kürzer als die Synchronisationsphase. Aus diesem Grund wurde diese Verarbeitungszeit hier nicht berücksichtigt.

Abbildung 3.9 zeigt, dass die zeitlichen Anforderungen der Automatisierungsindustrie bzgl. einer Punkt-zu-Punkt Verbindung erfüllt sind, d.h. eine Übertragungsdauer von unter 50  $\mu$ s kann erreicht werden.

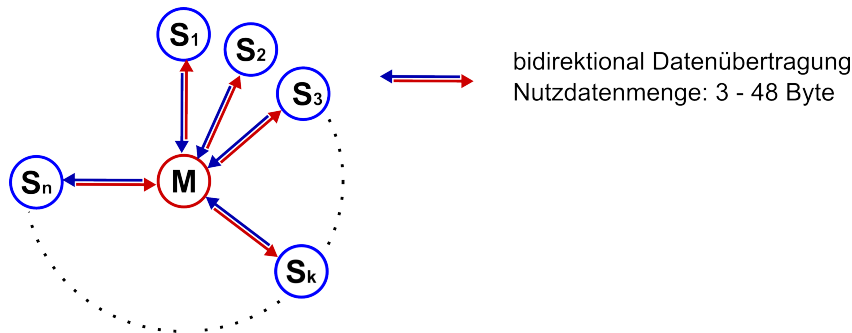
Es muss untersucht werden, ob die bisherigen Optimierungen ausreichen, um zusätzlich die zeitlichen Anforderungen für Punkt-zu-Multipunkt Verbindungen zu erfüllen. Dies geschieht im nächsten Abschnitt.



**Abb. 3.9:** Vergleich zwischen einer Punkt-zu-Punkt Übertragung entsprechend dem UWB-Standard und zweier optimierten Punkt-zu-Punkt Übertragungen. Es wird bei jedem Beispiel die maximale angeforderte Nutzdatenmenge von 48 Byte übertragen. Die Reed-Solomon-Dekodierung läuft parallel zur Übertragung, wird aber nicht immer vor Ende der Übertragung abgeschlossen. Daher wird hier diese Verarbeitungszeit mitberücksichtigt.  
 a) ist dem UWB-Standard entsprechende kürzeste Übertragung für 48 Byte Nutzdaten. Die Übertragungsdauer inkl. Verarbeitungszeit ist gleich 84,4 μs.  
 b) ist eine optimierte Übertragung wobei alle Optimierungen, bis auf das Entfernen des Viterbi-Dekoders, angewendet wurden. Die Übertragungsdauer inkl. Verarbeitungszeit ist gleich 47,7 μs.  
 c) ist eine optimierte Übertragung wie in b) wobei der Viterbi-Decoder entfernt wurde, was die Datenrate verdoppelt. Die Übertragungsdauer inkl. Verarbeitungszeit ist gleich 34,7 μs.

### 3.3 Optimierung der Punkt-zu-Multipunkt Datenübertragung

In diesem Abschnitt wird untersucht, ob die zeitlichen Anforderungen der Automatisierungsindustrie bzgl. Punkt-zu-Multipunkt Verbindungen erfüllt werden können. Die zeitliche Anforderung einer Zykluszeit von unter 1 ms gilt für bidirektionale Datenübertragungen von maximal 48 Byte zwischen einem Master und 127 Slaves innerhalb eines Netzwerkes mit Sterntopologie (siehe Abbildung 3.10). Dazu wird zuerst untersucht, welche Übertragungszeiten mit dem aktuellen UWB-Standard erreicht werden können. Daraufhin werden die aus Abschnitt 3.2 erzielten Verbesserungen angewendet und es wird geprüft, ob die Anforderungen der Automatisierungsindustrie damit erfüllt werden können.



**Abb. 3.10:** Bidirektionale Kommunikation zwischen einem Master und  $n$  Slaves mit  $n \leq 127$ . Es werden zwischen Master und jedem Slave maximal 48 Byte an Nutzdaten in beide Richtungen übertragen.

Die Punkt-zu-Multipunkt Übertragung aus Abbildung 3.10 kann als mehrfache Punkt-zu-Punkt Kommunikation dargestellt werden. Wird dabei die kürzeste Punkt-zu-Punkt Kommunikation inkl. Verarbeitungszeit des UWB-Standards verwendet (siehe Abbildung 3.9), so ergibt sich eine globale Übertragungszeit  $T_{TRANS}$  in Höhe von:

$$T_{TRANS} = 127 \cdot 2 \cdot T_{UWB-FRAME} = 127 \cdot 2 \cdot 80447 \text{ ns} = 20,43 \text{ ms}$$

Wird die in Abschnitt 3.2 vorgestellte optimierte Variante einer Punkt-zu-Punkt Verbindung genutzt (siehe Abbildung 3.9), so ergibt sich eine Übertragungszeit in Höhe von:

$$T_{TRANS} = 127 \cdot 2 \cdot T_{UWB-FRAME_{Opt}} = 127 \cdot 2 \cdot 34743 \text{ ns} = 8,82 \text{ ms}$$

Somit ist die Zykluszeitanforderung der Automatisierungsindustrie von bis zu 1 ms nicht erfüllt.

In den nächsten Abschnitten dieses Kapitels wird gezeigt, wie die Anforderungen bzgl. der Zykluszeit für das gesamte Netzwerk erfüllt werden können. Zuerst wird in Abschnitt 3.3.1 der UWB-Standard um eine Broadcast-Übertragung erweitert, um

die globale Synchronisationszeit im Netzwerk zu reduzieren. In Abschnitt 3.3.2 wird eine Vollduplex-Übertragung für 2 Sensorknoten entwickelt, um die Datenrate zu steigern. Zuerst geschieht dies mit einer geringen Anpassung der im UWB-Standard verwendeten Datenstruktur. Daraufhin wurde die Vollduplex-Übertragung für Multicast (es senden/empfangen mehrere Sensorknoten gleichzeitig zum/vom Master) erweitert. Dies erforderte allerdings starke Modifizierungen, sowohl bei der Synchronisationsphase als auch bei der Datenübertragungsphase. Abschließend wurden im Abschnitt 3.4 Fehlerkorrekturmaßnahmen hinzugefügt um die Zuverlässigkeit zu erhöhen.

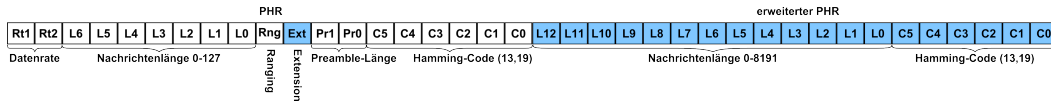
### 3.3.1 Broadcast-Übertragung

Eine Möglichkeit die Dauer der Datenübertragung zu reduzieren, ist die Verwendung von *Broadcast*-Nachrichten<sup>12</sup>. Da nach einem typischen Szenario in der Automatisierungsindustrie ein Master mit allen Slaves innerhalb eines Zyklusses kommunizieren soll, kann der Master seine Nachrichten mittels *Broadcast* als ein großes Datenpaket an alle Slaves übermitteln und nur für die Rückantwort auf herkömmliche Punkt-zu-Punkt-Übertragungen zurückgreifen. Um solch eine Broadcast-Nachricht zu verschicken, muss der Master in der Lage sein, Nachrichten der Länge von bis zu  $(48 \text{ Byte} + FEC) \cdot 127$  in einem UWB-Frame zu übertragen. In diesem Fall kann für die Fehlerkorrektur der Reed-Solomon-Code auf die gesamte Nachrichtenübertragung angewendet werden, welches zu einer besseren Ausnutzung des Overheads führt. Demzufolge beträgt die Datenmenge eines UWB-Frames für Broadcast-Nachrichten, Nutzdaten einschließlich Overhead, im Worst Case:

$$\text{Nutzdaten} + \text{Overhead} = 48 \cdot 127 + [(48 \cdot 127/41)] \cdot 6 = 6096 + 888 = 6984 \text{ Byte}$$

Der UWB-Standard unterstützt nur Nachrichtenlängen für Nutzdaten bis zu 127 Byte. Außerdem kann die erforderliche Nachrichtenlänge der Nutzdaten von 6096 Byte nicht mehr in den PHR-Teil hineinkodiert werden, da nur 7 Bit für die Länge der Nachricht zur Verfügung stehen. Der UWB-Standard ermöglicht aber für solche Fälle eine Erweiterung des PHR-Teils. Dazu muss das 10-te Bit im PHR-Teil, welches mit Ext bezeichnet ist, auf '1' gesetzt werden. Damit wird dem Empfänger mitgeteilt, dass es sich bei dieser Nachricht nicht um eine nicht standardkonforme Nachricht handelt und das der PHR-Teil anders interpretiert werden muss. Hier werden dem PHR-Teil weitere 19 Bit angehängt. Diese 19 Bit bestehen aus 13 Bit für die Länge der Nachricht (d.h. eine bis zu  $2^{13} - 1 = 8191$  Byte lange Nachricht kann gesendet werden) und weitere 6 Bit für den durch den Hamming-Code verursachten Overhead. Abbildung 3.11 zeigt wie der durch den Hamming-Code geschützte PHR-Teil erweitert werden kann, um eine über 127 Byte lange Nachricht zu übertragen.

<sup>12</sup>Damit werden Nachrichten bezeichnet, welche ausgehend von einem Teilnehmer des Netzwerkes an alle Teilnehmer des Netzwerkes gesendet werden.



**Abb. 3.11:** Erweiterte PHR für Broadcast-Nachrichten mit einer Nachrichtenlänge die als eine 13 Bit lange binäre Zahl dargestellt werden kann.

Bei solch langen Nachrichten müssen weitere Faktoren, wie *clock-drift*<sup>13</sup>, berücksichtigt werden. Im UWB-Standard wird ein maximaler *clock-drift* von 20 ppm<sup>14</sup> vorausgesetzt. Das heißt: innerhalb von 1000000 Taktzyklen, dürfen zwei Uhren oder zwei UWB-Sensorknoten maximal um  $\pm 20$  Taktzyklen (oder um  $\pm 0,002\%$ ) voneinander abweichen. Bei einer Punkt-zu-Punkt-Verbindung mit einer Nachrichtlänge von 48 Byte inkl. FEC und PHR, also  $19 + (48 + 12) \cdot 8 = 632$  Bit mit 32 *time-slots* pro Bit, entspricht dies einer Abweichung von:

$$\text{clock} - \text{drift} = 632 \cdot 32 \cdot TS \cdot \pm 0,00002\% = \pm 0,4 \cdot TS$$

Solch eine geringe Abweichung kann leicht kompensiert werden, wenn eine mögliche Abweichung von  $\pm 1 \cdot TS$  bei der Dekodierung der Daten berücksichtigt wird. Dies wurde bei der in dieser Doktorarbeit verwendeten Hardware implementiert.

Bei einer Punkt-zu-Multipunkt-Verbindung mit einer Nachrichtenlänge von 48 Byte inkl. FEC und erweiterter PHR für 127 Slaves, also  $19 + 19 + 6984 \cdot 8 = 55910$  Bit mit 32 *time-slots* pro Bit, entspricht dies einer Abweichung von:

$$\text{clock} - \text{drift} = 55910 \cdot 32 \cdot TS \cdot \pm 0,00002\% = \pm 35,78 \cdot TS$$

Diese Abweichung ist wesentlich größer, kann aber aufgrund der im Standard verwendeten vorhersagbaren Hopping-Positionen bzw. Burst-Positionen erkannt und kompensiert werden.

Eine Broadcast-Nachricht, inklusive Synchronisationsphase, erweiterter PHR und ohne Viterbi-Code<sup>15</sup>, mit 48 Byte Nutzdaten, dauert:

$$\begin{aligned} T_{BROADCAST} &= T_{SYNC} + T_{BROADCAST-DATA} \\ &= 16731 \text{ ns} + (19 + 19 + 6984 \cdot 8) \cdot 32,05 \text{ ns} \\ &\approx 1,8 \text{ ms} \end{aligned}$$

Daraufhin folgen maximal 127 Punkt-zu-Punkt-Übertragungen, für die Rückantworten der Slaves, wobei jede Rückantwort bis zu 34743 ns (siehe Abschnitt 3.2.4)

<sup>13</sup>damit wird der Fehler zwischen zwei Uhren bezeichnet

<sup>14</sup>engl. *parts per million*

<sup>15</sup>Wenn kein Viterbi-Code eingesetzt wird, werden nur 19 Bit im PHR übertragen, da das erste Null-Bit entfällt.



dauern kann. Demzufolge dauert ein kompletter Zyklus:

$$\begin{aligned}
 T_{TRANS} &= T_{BROADCAST} + T_{UWB-FRAME_{Opt}} \cdot 127 \\
 &\approx 1,8 \text{ ms} + 34743 \text{ ns} \cdot 127 \\
 &\approx 6,21 \text{ ms}
 \end{aligned} \tag{3.8}$$

Aus dem Vergleich der Ergebnisse aus den Gleichungen (3.3) und (3.8) folgt, dass die Zykluszeit um weitere 29,59%, bzgl. der schon zuvor optimierten Übertragung, reduziert werden konnte. Diese Reduzierung kommt hauptsächlich daher, dass in der Richtung Master zu Slaves nur einmal eine Synchronisation erfolgt, anstelle von 127 Synchronisationen. Diese Optimierung ist nicht ausreichend, denn allein für die Synchronisation wird folgende globale Synchronisationszeit innerhalb eines Zyklusses benötigt:

$$T_{G-SYNC} = T_{SYNC} \cdot (1 + 127) = 16731 \text{ ns} \cdot 128 \approx 2,14 \text{ ms}$$

Es muss ein anderer Ansatz gefunden werden, welcher es ermöglicht, sowohl die Datenübertragungsphase als auch die Synchronisationsphase zu reduzieren. Des Weiteren müssen auch Effekte wie *clock-drift* bei längeren Nachrichten berücksichtigt werden. Die Broadcast-Übertragung reduziert die globale Synchronisationszeit stark aber nicht die globale Datenübertragungszeit. Deshalb wird im nächsten Abschnitt die Möglichkeit der Nutzung von Vollduplex-Kommunikation untersucht. Damit wird sowohl die globale Synchronisationszeit als auch die globale Datenübertragungszeit reduziert.

### 3.3.2 Vollduplex-Kommunikation

Eine Vollduplex-Kommunikation wird vom UWB-Standard nicht unterstützt, sondern nur nur Halbduplex<sup>16</sup>-Kommunikationen, um Impuls-Kollisionen zu vermeiden.

Die Übertragungszeit einer Vollduplex<sup>17</sup>-Kommunikation kann halbiert werden und kann dazu beitragen die Anforderungen der Automatisierungsindustrie zu erfüllen.

Eine Vollduplex-Kommunikation hat auch Nachteile. Es wird mehr Energie benötigt, da Sender und Empfänger gleichzeitig aktiviert sind. Außerdem wird auch mehr Chipfläche für die Speicherlogik benötigt, da die Sendedaten und Empfangsdaten getrennt gespeichert und verarbeitet werden müssen. Dafür werden entweder verschiedene Speicher genutzt oder es wird ein größerer Speicher mit Mehrfachzugriff verwendet.

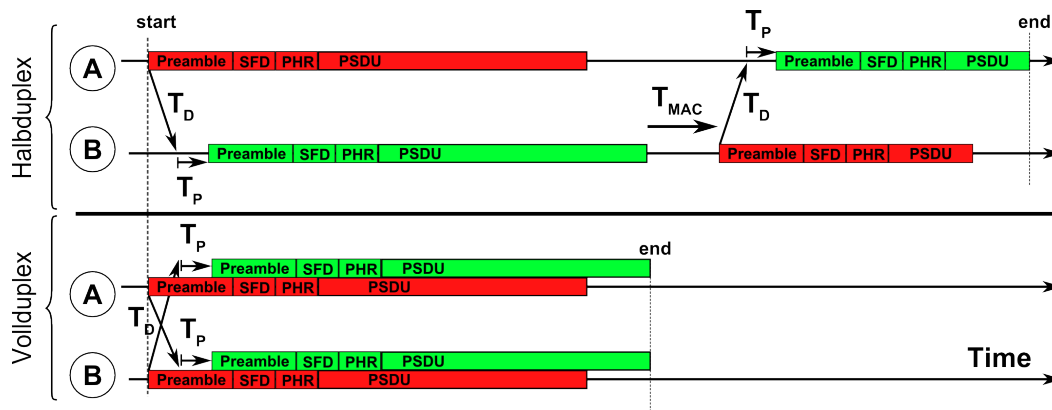
Es muss untersucht werden, wie sich die Zuverlässigkeit im Vollduplex-Betrieb ändert. Es muss geklärt werden, ob vermehrt Störungen durch Kollisionen auftreten

<sup>16</sup>Hierbei können auf einem Kanal Daten abwechselnd in beide Richtungen übertragen werden, d.h. gleichzeitig.

<sup>17</sup>Daten können in beide Richtungen gleichzeitig übertragen werden.

und welche Auswirkungen unterschiedliche Nachrichtenlängen und verschiedene Netzwerkgrößen haben.

Es gibt zwei verschiedene Möglichkeiten für 2 Sensorknoten  $A$  und  $B$  Nachrichten bidirektional auszutauschen. Abbildung 3.12 zeigt die Übertragungsstruktur und die Dauer der Übertragung für das Halbduplex- und das Vollduplex-Verfahren. Der Datenaustausch mit dem Vollduplex-Verfahren ist wesentlich kürzer, da Sender und Empfänger gleichzeitig aktiv sind.



**Abb. 3.12:** Vergleich zwischen Halbduplex und Vollduplex.

Die von einem Sensorknoten ausgesendeten Signale sind rot, die empfangenen Signale sind grün markiert.  $T_D$  ist die Verzögerung verursacht durch die Signallaufzeit. Bei einer Entfernung von 20 m entspricht dies einer Verzögerung von bis zu  $\approx 70$  ns.

$T_P$  ist die Verzögerung, die durch die Signalverarbeitungszeit im Basisband verursacht wurde. Diese Zeit ist konstant.

$T_{MAC}$  ist die durch den MAC verursachte Verzögerung der Antwort.

Fall 1: Eine Halbduplex-Kommunikation entsprechend dem UWB-Standard.  $A$  initiiert eine Halbduplex-Übertragung zu  $B$ . Nachdem die Übertragung abgeschlossen ist, initiiert  $B$  nach einer Zeit von  $T_{MAC}$  (abhängig vom MAC-Protokoll) eine Halbduplex-Übertragung zu  $A$ .

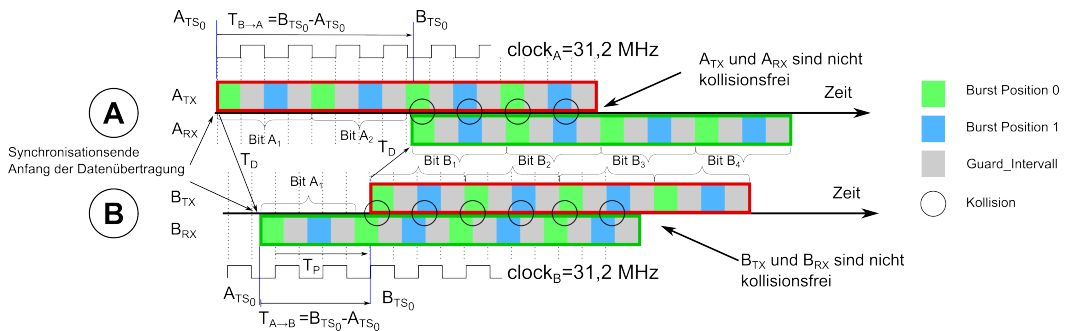
Fall 2: Vollduplex-Kommunikation.  $A$  und  $B$  synchronisieren sich und übertragen ihre Daten gleichzeitig. Durch die gleichzeitige Übertragung können Kollisionen auftreten. Diese Art der Übertragung wird im UWB-Standard nicht unterstützt. Eine Vollduplex-Kommunikation kann mittels zwei unterschiedlicher Hardwareansätze implementiert werden.

- 1) Durch Anwendung von zwei Antennen.  
In diesem Fall werden sowohl alle Impulse des eigenen Senders als auch die des Kommunikationspartners empfangen. Die empfangenen Impulse des eigenen Senders müssen ignoriert oder unterdrückt werden.
- 2) Durch Anwendung von schaltbaren Antennen.  
In diesem Fall werden Sender und Empfänger abwechselnd ein bzw. ausgeschaltet. Der Sender hat gegenüber dem Empfänger eine höhere Pri-

orität. Wenn der Empfänger zu einem bestimmten *time-slot* Daten erwartet und der Sender aber gleichzeitig Daten senden muss, dann werden die Daten gesendet und der Empfänger bekommt keine Daten für diesen *time-slot*.

### 3.3.2.1 Kollisionen und relative Verzögerungen

Bei einer Vollduplex-Kommunikation werden die übertragenen Nachrichten von  $A$  nach  $B$  ( $A \rightarrow B$ ) und von  $B$  nach  $A$  ( $B \rightarrow A$ ) relativ zueinander verzögert. Abbildung 3.13 zeigt die Abhängigkeit zwischen Kollisionen und Verzögerungszeiten.



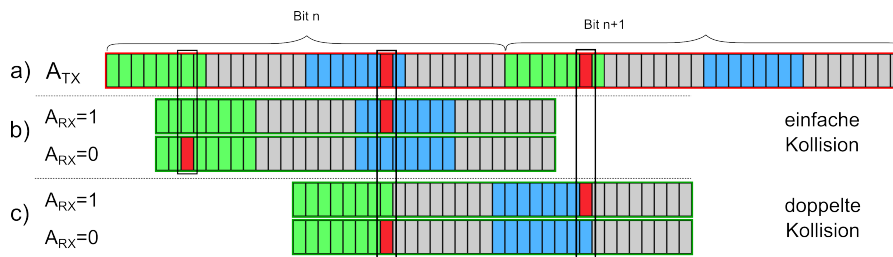
**Abb. 3.13:** Relative Verzögerungen zwischen  $A$  und  $B$  bei einer Datenrate von 27,24 Mb/s. Sowohl bei  $A$  als auch bei  $B$  treten Kollisionen auf, da sich deren Burst-Intervalle (in grün und blau markiert wie in Abbildung 3.3 auf Seite 41) teilweise überlappen. Die ausgesendeten UWB-Frames sind rot und die empfangenen UWB-Frames sind grün umrandet.  $T_{B \rightarrow A}$  ist die Verzögerung des Sendesignals von  $B$  relativ zu  $A$ . Diese ist als Zeitabstand in *time-slots* zwischen dem ersten *time-slot* des Sendesignals von  $A$  ( $A_{TS_0}$ ) und des ersten *time-slots* des Sendesignals von  $B$  ( $B_{TS_0}$ ) beim Sensorknoten  $A$  definiert.  $T_{A \rightarrow B}$  ist die Verzögerung des Sendesignals von  $A$  relativ zu  $B$ . Diese ist als Zeitabstand in *time-slots* zwischen dem ersten *time-slot* des Sendesignals von  $A$  ( $A_{TS_0}$ ) und des ersten *time-slots* des Sendesignals von  $B$  ( $B_{TS_0}$ ) beim Sensorknoten  $B$  definiert.  $T_P$  ist die Verzögerung, welche durch die Verarbeitungszeit (engl. *processing time*) im Basisband verursacht wird. Diese Verzögerung ist für beide Sensorknoten identisch und beträgt zwei Taktzyklen  $\approx 2 \cdot 32 \text{ ns} = 64 \text{ ns}$ .  $T_D$  ist die Laufzeit des UWB-Signals und ist nur vom Abstand  $D$  zwischen den beiden Sensorknoten abhängig. Daher ist die Laufzeit von  $A \rightarrow B$  gleich der Laufzeit von  $B \rightarrow A$ .

Es können zwei Typen von Kollisionen auftreten. Diese sind sowohl für Varianten mit schaltbaren Antennen als auch mit mehrfachen Antennen identisch.

1. **Einfache Kollisionen:** Während einer Übertragung kommt es zu einer Überlappung zweier Bursts. Sensor  $A$  sendet einen Burst in einem *time-slot*, aber gleichzeitig wird ein Burst vom Sensor  $B$  im selben *time-slot* erwartet. Dies hat zur Folge, dass der Empfänger von  $A$  nicht eindeutig bestimmen kann, ob der empfangene Burst von  $A$  oder von  $B$  stammt. Solch eine Uneindeutigkeit kann mithilfe des zweiten entgegengesetzten *time-slots* behoben werden. Ist ein valider Burst im entgegengesetzten *time-slot* vorhanden, so gab es keine Überlappung. Ist kein zweiter Burst vorhanden so gab es eine Überlappung oder der Burst wurde nicht detektiert.
2. **Doppelte Kollisionen:** Während einer Übertragung kommt es innerhalb

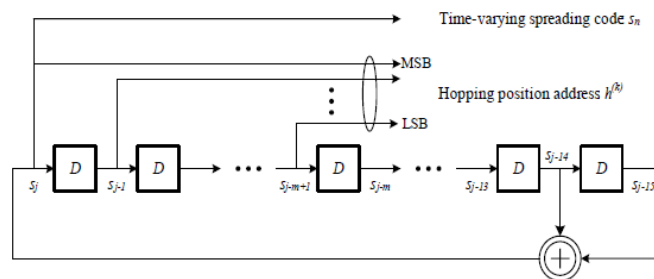
eines Datensymbols zur Überlappungen beider möglichen Burst-Positionen. Sensor  $A$  sendet zwei Burst nacheinander mit einem Abstand von exakt 16 *time-slots* genau dann, wenn auch ein Burst vom Sensor  $B$  in einem von beiden *time-slots* erwartet wird. Da in diesem Fall in den jeweils entgegengesetzten *time-slots* Bursts empfangen wurden, welche aber auch von  $A$  stammen könnten, ist diese Situation nicht eindeutig und auch nicht auflösbar.

Beide Typen von Kollisionen sind in Abbildung 3.14 dargestellt sind.



**Abb. 3.14:** Typen von Kollisionen  
 a) gesendete Bursts von  $A$   
 b) und c) empfangene Bursts bei  $A$ , welche von  $B$  gesendet wurden

Kollisionen können nur auftreten, wenn mehrere Sensorknoten gleichzeitig aktiv sind. Deshalb sind spezielle Hopping-Sequenzen im UWB-Standard vorgesehen, um Kollisionen zu vermeiden, und werden mittels eines linearem rückgekoppelten Schieberegister (*engl. Linear Feedback Shift Register, LFSR*) realisiert.



**Abb. 3.15:** Aufbau und Funktionsweise des LFSR entsprechend dem UWB-Standard, entnommen aus [37]

Das LFSR generiert pseudozufällige Sequenzen, welche die Kollisionswahrscheinlichkeit reduzieren. Der LFSR wird mit einem Wert initialisiert der sich aus dem verwendeten PREAMBLE-Symbol berechnen lässt. Dazu werden in der Impulssequenz des PREAMBLE-Symbols alle 0 entfernt und alle -1 in 0 umgewandelt. Die ersten 15 Bit des daraus resultierenden Vektors werden dann als Initialwert genommen. Mit diesem Verfahren können sowohl Master als auch der Slave die gleichen Hopping-Positionen berechnen.

Die allgemeine LFSR-Formel aus dem UWB-Standard lautet wie folgt:

$$h^{(k)} = 2^0_{S_{kN_{cpb}}} + 2^1_{S_{1+kN_{cpb}}} + \dots + 2^{m-1}_{S_{m-1+kN_{cpb}}}$$

$h^{(k)}$  ist die Hopping-Position für das  $k$ -te Bit.

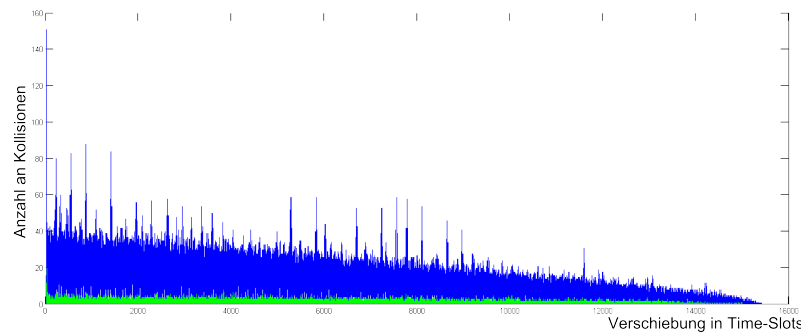
$N_{hop} = 8$  Anzahl an Hopping-Positionen pro Burst-Position

$m = \log_2(N_{hop}) = 3$

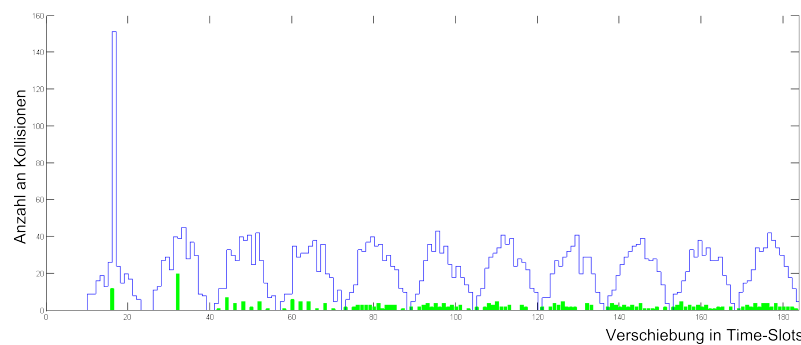
Basierend auf diesem LFSR wurde die Kollisionshäufigkeit für verschiedene Nachricht-  
enlängen abhängig von der Verzögerungszeit berechnet, siehe Abbildung 3.16. Die  
Abbildung 3.17 ist eine vergrößerte Darstellung von Abbildung 3.16.

Es wurden Simulationen für den Worst Case durchgeführt, d.h. wenn sowohl  $A$   
als auch  $B$  eine Nachricht der Länge 48 Byte (zzgl. PHR und FEC) übertragen.  
Die Übertragungszeit der PSDU beträgt daher 34743 ns<sup>18</sup> (siehe Abschnitt 3.2.4).

Abbildung 3.17 zeigt, dass die Anzahl an Kollisionen mit steigender Verzögerung  
abnimmt.



**Abb. 3.16:** Durchschnittliche Kollisionsverteilung: Häufigkeit der Kollisionen in Abhängigkeit der relativen  
Verschiebung in *time-slots*.  
Kollisionen vom Typ1 sind blau und Kollisionen vom Typ2 sind grün markiert.



**Abb. 3.17:** Vergrößerung der Kollisionsverteilung aus Abbildung 3.16.

Weiterhin verdeutlicht Abbildung 3.16, dass es einen Ausschlag bei *time-slot* 16  
für Kollisionen des Typ 1 und einen Ausschlag bei *time-slot* 16 für Kollisionen des  
Typ 2 gibt. Dieses Verhalten ist erklärbar und nachvollziehbar. Da jedem Datenbit  
jeweils ein *time-slot*-Paar ( $TS_k$  und  $TS_{k+16}$  für  $k \in \{1, \dots, 8\}$ ) zugewiesen wird, liegt  
die Wahrscheinlichkeit einer Kollision bei einer Verzögerung von exakt 16 *time-slots*  
bei 25%. Dieser Fall ist in Abbildung 3.18 a) dargestellt.

<sup>18</sup>Ein *time-slot* hat eine Länge von  $\approx 2$  ns bei einer Datenrate von 27,24 MB/s.

Der zweite Ausschlag entspricht einer Verzögerung von exakt 32 *time-slots*, da sich bei dieser Verzögerung immer alle  $A_{TS_{k+16}}$  von  $A$  und alle  $B_{TS_k}$  überlappen, dadurch ist die Wahrscheinlichkeit für Kollisionen des Typ 2 wesentlich größer. Dieser Fall ist in Abbildung 3.18 b) dargestellt.

Diese beiden Fälle können nur eintreten wenn:

$$T_{B \rightarrow A} \leq 32 \cdot TS \approx 32 \cdot 2 \text{ ns} = 64 \text{ ns}$$

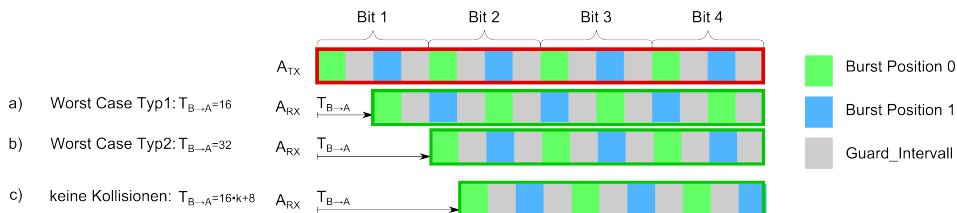
Da aufgrund der Synchronisierungsverarbeitung von  $B$  gilt  $T_P = 64 \text{ ns}$  und weil  $T_{B \rightarrow A} > T_P$  folgt:

$$T_{B \rightarrow A} > 64 \text{ ns}$$

Somit kann keiner der beiden Fällen eintreten.

Kollisionen des Typ 1 treten nicht zufällig sondern periodisch auf. Außerdem treten bei Verzögerungen um jeweils  $(k \cdot 16 + 8)$  *time-slots* keine Kollisionen sowohl vom Typ 1 und Typ 2 auf. Dies kann folgenderweise erklärt werden. Bei diesen speziellen Verzögerungen befinden sich die *time-slot*-Paare von  $A$  in den Guard-Intervallen von  $B$  und gleichzeitig befinden sich die *time-slot*-Paare von  $B$  in den Guard-Intervallen von  $A$ . Demzufolge können keine Kollisionen auftreten. Dieser Fall ist in Abbildung 3.18 c) dargestellt.

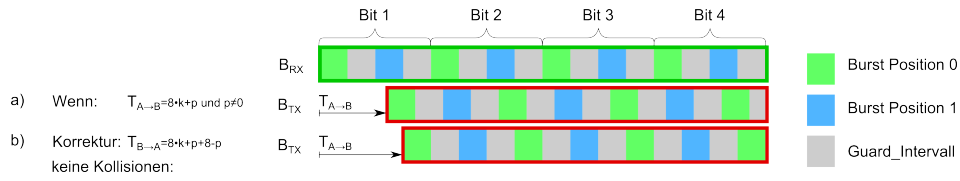
Durchschnittlich gibt es 12 Kollisionen des Typ 1 und 1,2 Kollisionen des Typ 2 pro Nachricht. Bei Nachrichten, welche um  $k \cdot 16$  *time-slots* verzögert sind, gibt es durchschnittlich 23 Kollisionen des Typ 1 und 2 Kollisionen des Typ 2.



**Abb. 3.18:** Auswirkungen der verschiedenen Verzögerungen von  $B$  relativ zu  $A$ . Die ausgesendeten UWB-Frames sind rot und die empfangenen UWB-Frames sind grün umrandet.  
a) stellt den Worst Case für Kollisionen des Typ 1 dar.  
b) stellt den Worst Case für Kollisionen des Typ 2 dar.  
c) stellt den Fall ohne Kollisionen dar.

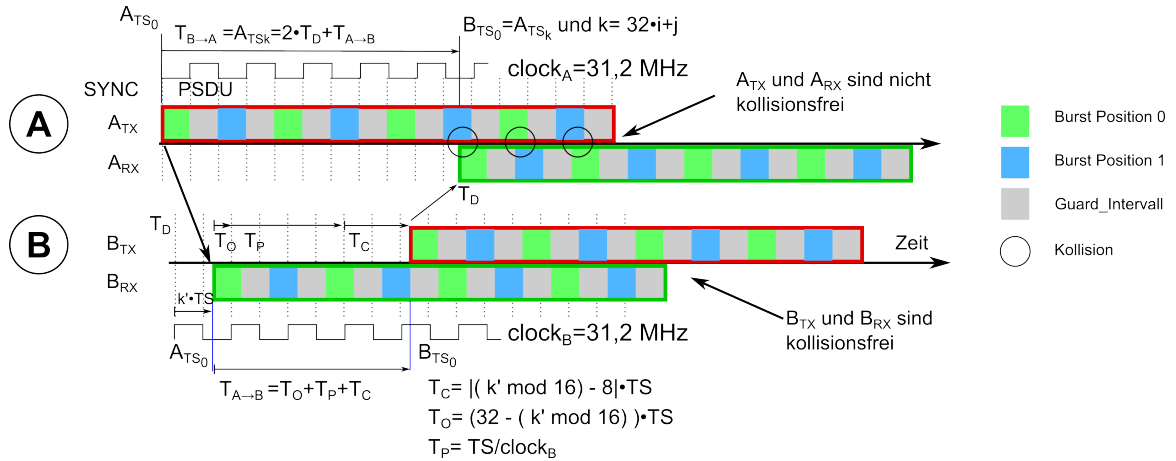
### 3.3.2.2 Kollisionsvermeidung

Um die geforderten Zuverlässigkeitsanforderungen bei einer Vollduplex-Kommunikation erfüllen zu können, müssen Kollisionen vermieden werden. In Abbildung 3.19 werden Kollisionen seitens  $B$  durch das Verzögern des Sendesignals  $B_{TX}$  verhindert. Abbildung 3.19 a) stellt das Sendesignal vom Sensorknoten  $B$ , mit Überlappungen der Burst-Positionen von  $A$  und  $B$ , dar und Abbildung 3.19 b) stellt das korrigierte Sendesignal von  $B$ , ohne Überlappungen der Burst-Positionen von  $A$  und  $B$ , dar.



**Abb. 3.19:** Auswirkungen der verschiedenen Verzögerungen von  $A$  relativ zu  $B$ . Die ausgesendeten UWB-Frames sind rot und die empfangenen UWB-Frames sind grün umrandet. a) ist das originale Sendesignal von  $B_{TX}$ . b) ist das korrigierte Sendesignal von  $B_{TX}$ . Das Sendesignal wurde derart verzögert, dass keine Überlappungen auftreten.

Abbildung 3.20 zeigt wie sich die Kollisionen im Vergleich zu Abbildung 3.13 beim Sensorknoten  $B$  verhindern lassen.  $B_{TX}$  wird um  $|(k' \bmod 16) - 8| \text{ time-slots}$  verzögert. Die Variable  $k'$  ist die relative Verzögerung des Empfangssignals  $B_{TX}$  mit  $clock_B$ . Mit  $T_O$  wird der Offset bezeichnet, welcher dadurch entsteht, dass die Verarbeitung der Daten erst mit der nächsten steigenden Flanke von  $clock_B$  beginnt.



**Abb. 3.20:** Kollisionsunterdrückung relativ zu  $B$

Aus Abbildungen 3.19 und 3.20 folgt, dass für jedes  $T_{A \rightarrow B}$  mehrere  $T_{A \rightarrow B_{Korr}}$  existieren mit  $T_{A \rightarrow B_{Korr}} = T_{A \rightarrow B} + T_C + l \cdot 16 \cdot TS$  mit  $l \in \mathbb{N}$ , für welche keine Kollisionen mit  $A$  relativ zu  $B$  auftreten.

Es bleibt zu untersuchen, ob es möglich ist, für eine gegebene Verzögerung  $T_{A \rightarrow B_{Korr}}$ , Hopping-Sequenzen für  $B$  zu berechnen bei welcher  $A_{TX}$  und  $A_{RX}$  kollisionsfrei sind.

Die Antwort ist "Ja". Es ist sogar möglich einen Mindestabstand von 3 *time-slots*, zwischen 2 Bursts zu garantieren. Damit ist eine deutliche Abgrenzung beider Signale sichergestellt. Es folgt nun der Mathematische Beweis.

**Beweis:**

Für diesen Beweis werden folgende Notationen eingeführt:

- $X_k \in \{0, 1\}$  ist der Wert des  $k$ -ten Bits vom Sensorknoten X.
- $0 \leq X_{h^{(k)}} \leq 7$  ist die Hopping-Position für das  $k$ -te Bit vom Sensorknoten X.
- $TS_{X_k}$  ist der *time-slot* für  $X_k$  das  $k$ -te Bit vom Sensorknoten X.

Seien folgende Annahmen gegeben:

- Sei  $T_{B \rightarrow A} = i \cdot 32 + j$  und  $j \in \{-8, \dots, 8\}$  für ein bestimmtes  $T_{A \rightarrow B_{Korr}}$
- Sei  $0 \leq B_{h^{(k)}} = (A_{h^{(k+i)}} + 4 - j)_{\text{mod } 8} \leq 7$  die Hopping-Position für  $B_k$  das  $k$ -te Bit von  $B$ .

Dann gilt:

$$0 \leq A_{h^{(k)}} = h^{(k)} \leq 7 \text{ die Hopping-Position für } A_k, \text{ das } k\text{-te Bit von A.}$$

$$\text{Der } \textit{time-slot} \text{ für } A_{k+i} \text{ ist } TS_{A_{k+i}} = (k+i) \cdot 32 + A_{h^{(k+i)}} + 16 \cdot A_{k+i}$$

$$\text{Der } \textit{time-slot} \text{ für } B_k \text{ ist } TS_{B_k} = k \cdot 32 + B_{h^{(k)}} + 16 \cdot B_k + T_{B \rightarrow A} = (k+i) \cdot 32 + j + (A_{h^{(k+i)}} + 4 - j)_{\text{mod } 8} + 16 \cdot B_k$$

Die *time-slot*-Differenz ist  $TS_{Diff} = TS_{A_{k+i}} - TS_{B_k}$

$$TS_{Diff} = (k+i) \cdot 32 + A_{h^{(k+i)}} + 16 \cdot A_{k+i} - (k+i) \cdot 32 - j - |A_{h^{(k+i)}} + 4 - j|_{\text{mod } 8} - 16 \cdot B_k$$

- Fall 1: Für  $A_k = B_k$  ist die *time-slot*-Differenz  
 $TS_{Diff} = A_{h^{(k+i)}} - j - |A_{h^{(k+i)}} + 4 - j|_{\text{mod } 8}$
- Fall 2: Für  $A_k \neq B_k = 0$  ist die *time-slot*-Differenz  
 $TS_{Diff} = A_{h^{(k+i)}} - j - |A_{h^{(k+i)}} + 4 - j|_{\text{mod } 8} + 16$
- Fall 3: Für  $A_k \neq B_k = 1$  ist die *time-slot*-Differenz  
 $TS_{Diff} = A_{h^{(k+i)}} - j - |A_{h^{(k+i)}} + 4 - j|_{\text{mod } 8} - 16$

Alle drei Formeln können wie folgt zusammengefasst werden:

$$\exists l \in \{-1, 0, 1\}, \text{ so dass } TS_{Diff} = A_{h^{(k+i)}} - j - |A_{h^{(k+i)}} + 4 - j|_{\text{mod } 8} + l \cdot 16$$

Eine Kollision zwischen  $B_k$  und  $A_{k+i}$  findet nur dann statt, wenn  $TS_{Diff} = 0$ .

$$\begin{aligned} TS_{Diff} = 0 &\Leftrightarrow \exists l \in \{-1, 0, 1\}, \text{ so dass } A_{h^{(k+i)}} - j - (A_{h^{(k+i)}} + 4 - j)_{\text{mod } 8} + l \cdot 16 = 0 \\ &\Leftrightarrow \exists l \in \{-1, 0, 1\} \text{ und } \exists m \in \mathbb{Z}, \text{ so dass } A_{h^{(k+i)}} - j - (A_{h^{(k+i)}} + 4 - j) + m \cdot 8 + l \cdot 16 = 0 \\ &\Leftrightarrow \exists l \in \{-1, 0, 1\} \text{ und } \exists m \in \mathbb{Z}, \text{ so dass } 4 + m \cdot 8 + l \cdot 16 = 0 \end{aligned}$$

Das ist immer falsch, daher gilt:  $TS_{Diff} \neq 0$



Weiterhin folgt daraus, dass:  $|TS_{Diff}| \geq 4$

Somit ist auch bewiesen, dass sich zwischen den beiden *time-slots*  $TS_{A_{k+i}}$  und  $TS_{B_k}$  mindestens 3 freie *time-slots* befinden. Siehe hierzu das Beispiel von Abbildung 3.21, welches den Worst Case bezüglich des Mindestabstandsproblems darstellt.

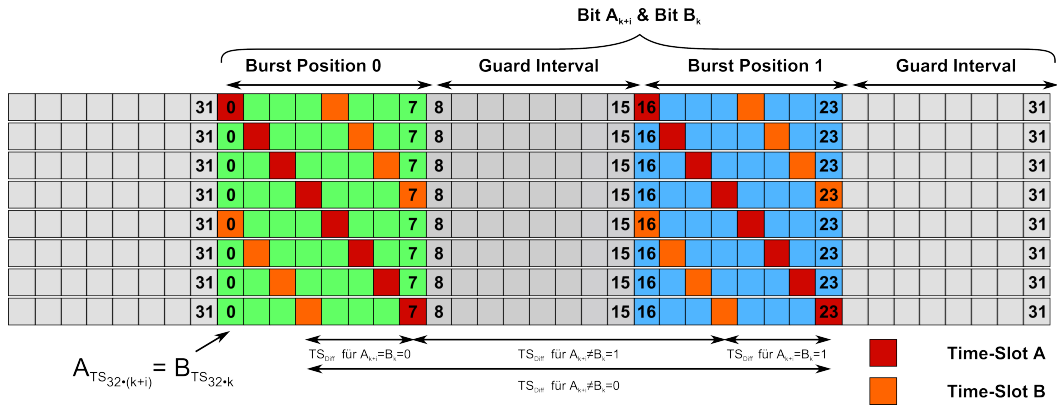


Abb. 3.21: Alle mögliche Hopping-Positionen für  $j = 0$

Eine Kollision zwischen benachbarten Bits  $A_{k+i-1}$  oder  $A_{k+i+1}$  mit  $B_k$  kann nur dann ausgeschlossen werden wenn:

$$\begin{aligned}
 & \exists t \in \{0, 1, \dots, 8\}, \text{ so dass } 32 \cdot (k+i) - t \leq TS_{A_{k+i}} - TS_{Diff} \leq 32 \cdot (k+i+1) - 1 - t \\
 \Leftrightarrow & \exists t \in \{0, 1, \dots, 8\}, \text{ so dass } -t \leq TS_{B_k} - 32 \cdot (k+i) \leq 31 - t \\
 \Leftrightarrow & \exists t \in \{0, 1, \dots, 8\}, \text{ so dass } -t \leq B_{h^{(k)}} + 16 \cdot B_k + j \leq 31 - t \\
 \Leftrightarrow & \exists t \in \{0, 1, \dots, 8\}, \text{ so dass } 0 \leq B_{h^{(k)}} + 16 \cdot B_k + j + t \leq 31
 \end{aligned}$$

Sei  $t = -j$  für  $j < 0$  und  $t = 8 - j$  für  $j \geq 0$  dann gilt:  $j + t = 0$  und daraus folgt  $0 \leq B_{h^{(k)}} + 16 \cdot B_k \leq 31$ . Diese Aussage ist immer wahr und somit ist auch bewiesen, dass es keine Überschneidungen zwischen  $A_{k+i-1}$  oder  $A_{k+i+1}$  mit  $B_k$  geben kann. In Abbildung 3.22 sind alle möglichen Verzögerungen durch  $j$  dargestellt.





5. Sobald  $A$  das PREAMBLE-Symbol  $PS_{B0}$  bzw.  $PS_{B1}$  zuverlässig erkannt hat, wird die AGC gesperrt.
6. Sobald  $A$  das PREAMBLE-Symbol  $PS_{B1}$  erkannt hat, wird ein anderes PREAMBLE-Symbol  $PS_{A1}$  um exakt  $T_{PS_{B1}} \rightarrow T_{PS_{A0}} = k \cdot TS$  verzögert übertragen.
7. Nachdem  $A$   $PS_{A1}$  ausgesendet hat, fängt  $A$  nach  $T_{PS} - k \cdot TS$  an seine Daten zu übertragen.
8. Nachdem  $B$   $PS_{A1}$  empfangen hat, fängt  $B$  nach dem letzten übertragenen PREAMBLE-Symbol  $PS_{B1}$  an seine Daten zu übertragen. Falls  $8 < (k \bmod 32) < 24$ , werden die Daten von  $B$  um 16 *time-slots* verzögert, um eine kollisionsfreie Übertragung zu gewährleisten.
9.  $A$  und  $B$  übertragen nun Ihre jeweiligen Daten gleichzeitig.  $A$  und  $B$  kennen den Wert für  $T_{PS_{B1}} \rightarrow T_{PS_{A0}} = k \cdot TS$  und sind somit beide in der Lage, die Hopping-Positionen des jeweils anderen zu berechnen. Die Synchronisation ist nun abgeschlossen.

#### Ranging:

Ein interessanter Nebeneffekt bei dieser Synchronisation ist, dass jeder Teilnehmer den Abstand zum anderen ermitteln kann, denn:  $T_{PS_{B1}} \rightarrow T_{PS_{A0}} = k \cdot TS = 2 \cdot T_D + T_P + T_{Korr}$

$$\Rightarrow T_D = \frac{k \cdot TS - T_P - T_{Korr}}{2} \Rightarrow D = \frac{k \cdot TS - T_P - T_{Korr}}{2} \cdot c$$

#### PREAMBLE-Symbole:

Aufgrund des Abstandes  $D$  bzw. der Signallaufzeit zwischen den Sensorknoten, müssen die PREAMBLE-Symbole eine gewisse Mindestdauer aufweisen damit das vorgestellte Synchronisationsverfahren funktioniert. Es gilt folgende Beschränkung für  $T_{PS}$ :

$$T_{PS} > 2 \cdot T_D + 2 \cdot T_P + 16 \cdot TS$$

$$\text{Sei } D \leq 30 \text{ m} \Rightarrow T_D \leq 100 \text{ ns} = 50 \cdot TS$$

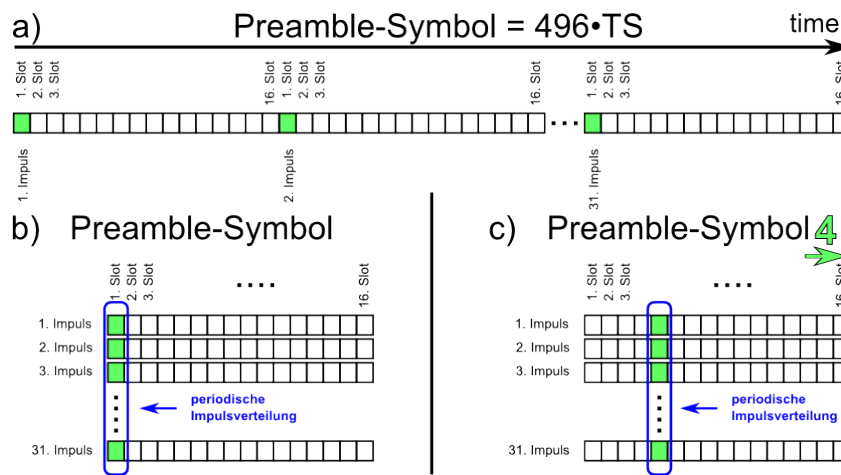
$$\text{Sei } T_P = 2 \cdot \text{Taktzyklen} = 32 \cdot TS$$

Daraus folgt:

$$T_{PS} > 2 \cdot 50 \cdot TS + 2 \cdot 32 \cdot TS + 16 \cdot TS = 180 \cdot TS$$

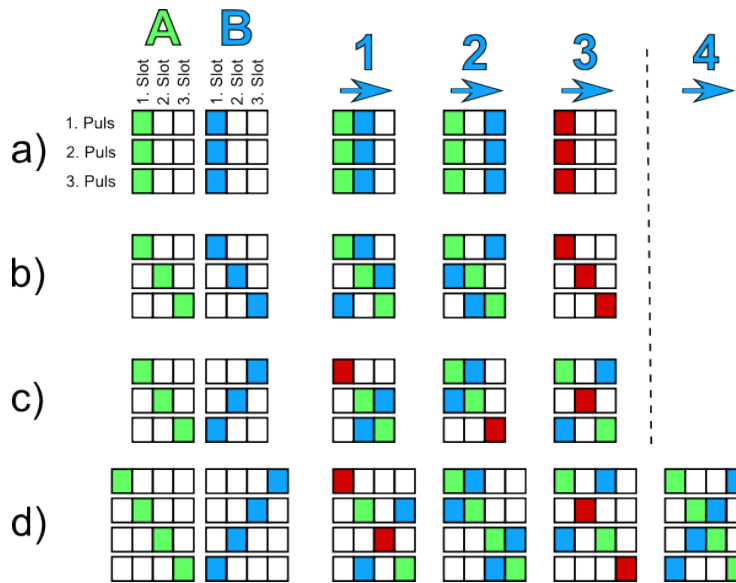
Weiterhin müssen die PREAMBLE-Symbole  $PS_{A0}$ ,  $PS_{A1}$ ,  $PS_{B0}$  und  $PS_{B1}$  so gewählt werden, dass diese gute Autokorrelationseigenschaften besitzen, robust gegenüber Kollisionen bei gleichzeitiger Übertragung sind und die UWB-Regulierungen nicht verletzen.

Derzeit besitzen die im UWB-Standard genutzten PREAMBLE-Symbole zwar sehr gute Autokorrelationseigenschaften, sind aber aufgrund ihrer Periodizität schlecht für diese Synchronisation geeignet. Die Periodizität entsteht dadurch, dass alle  $32 \text{ ns} = 16 \cdot TS$ , mit  $TS = 2 \text{ ns}$ , ein Impuls (+1,-1 oder 0) gesendet wird. Wenn  $A$  und  $B$  PREAMBLE-Symbole gleichzeitig aussenden, und diese sich exakt überlappen oder um ein vielfaches von 16  $TS$  zueinander verschoben sind, dann kollidieren alle Impulspositionen von  $A$  und  $B$ . Daher ist es sinnvoll, die PS-Impulse variable auf dem Impulsverteilungsintervall von 16 *time-slots* zu verteilen. Werden allerdings Für  $A$  und  $B$  identische Impulsverteilungen gewählt, so wird das Problem nicht behoben. Eine gegenläufige Impulsverteilung von  $A$  und  $B$  kann dieses Problem beheben. Abbildung 3.24 zeigt eine andere Darstellungsform des PREAMBLE-Symbols aus dem UWB-Standard, welche einen besseren Überblick der Impulsverteilungsfunktion ermöglicht.



**Abb. 3.24:** Unterschiedliche Darstellungen eines PREAMBLE-Symbols aus IEEE 802.15.4a. Die Impulspositionen sind grün markiert. a) zeigt die übliche zeitliche Darstellung eines PREAMBLE-Symbols mit 31 Impulsen und einer Impulsintervalllänge von 16 *time-slots* an. b) zeigt das gleiche PREAMBLE-Symbol wie in a) nur dass die Impulsintervalle untereinander angeordnet sind. Dies ermöglicht eine kompaktere Darstellung und die Periodizität der Impulsverteilung ist besser sichtbar. c) zeigt das gleiche PREAMBLE-Symbol wie in b), welches um 4 *time-slots* nach rechts verschoben ist.

In Abbildung 3.25 wird die gleiche Darstellungsform eines PREAMBLE-Symbols wie in Abbildung 3.24 genutzt. Zur Veranschaulichung des Kollisionsproblems, sind einfachere PREAMBLE-Symbole, mit Impulsen und Verteilungsintervalllänge von 3 bis 4, dargestellt.



**Abb. 3.25:** Vergleich verschiedener Impulsverteilungen für  $A$  und  $B$ , und deren Auswirkungen auf Kollisionen bei verschiedenen Verzögerungen von  $B$ . Die Impulspositionen für  $A$  sind grün markiert. Die Impulspositionen für  $B$  sind blau markiert. Kollisionen zwischen  $A$  und  $B$  sind rot markiert.

- a) zeigt eine identische Verteilung für  $A$  und  $B$ , ähnlich wie im UWB-Standard.  
 b) zeigt eine identische Verteilung für  $A$  und  $B$ , aber jeder Impuls befindet sich in einem anderen *time-slot*.  
 c) zeigt zwei Verteilungen wie in b), welche aber gegenläufig sind für  $n = 3$  ungerade.  
 d) zeigt zwei Verteilungen wie in b), welche aber gegenläufig sind für  $n = 4$  gerade.

#### Erklärung zu Abbildung 3.25:

a) zeigt eine identische Verteilung, ähnlich wie im UWB-Standard für  $A$  und  $B$ . Hierbei ist die Länge des Impulsintervalls  $n = 3$ . Dies lässt nur 3 mögliche Verschiebungen zwischen  $A$  und  $B$  zu. Eine Überlappung erfolgt nur in einem von  $n$  Fällen, aber dieser Fall ist gravierend, da es eine Überlappung mit allen Impulsen des PREAMBLE-Symbols gibt.

In b) wird eine identische Verteilung für  $A$  und  $B$ , mit der besonderen Eigenschaft, dass für jeden PS-Impuls ein anderer *time-slot* festgelegt wurde, ausgewählt. Wiederum ist  $n = 3$  und wiederum kann eine Überlappung aller Impulspositionen auftreten.

In c) wird für  $A$  die gleiche Verteilung wie in b) ausgewählt und für  $B$  wird eine gegenläufige Verteilung zu  $A$  gewählt. In diesem Fall für  $n$  ungerade, erfolgt bei jeder Verschiebung exakt nur eine Impuls-Überlappung. Je größer  $n$  gewählt wird, desto geringer wird der Einfluss dieser einen Überlappung auf das Korrelationsergebnis.

d) ist analog zu c), nur dass  $n$  gerade ist. Bei einem geraden  $n$  überlappen sich maximal exakt 2 Impulse. Interessanterweise gibt es bei geraden Verschiebungen keine Überlappungen.

Da die relativen Verschiebungen zwischen den PREAMBLE-Symbolen von  $A$  und  $B$  zufällig sind, kann eine gerade Verschiebung nicht garantiert werden. Somit

ist die in c) dargestellte Impulsverteilung, für  $n$  ungerade und nur einer Überlappung, am robustesten.

In Abbildung 3.25 wurden nur PREAMBLE-Symbole mit Impulsintervalllänge = Impulspositionen =  $n$  berücksichtigt. Dies ist auch der Optimalfall. Denn für Impulsintervalllängen  $< n$ , gibt es mindestens zwei Impulse mit gleicher Impulsposition. Daher kann mehr als eine Überlappung auftreten. Wenn die Impulsintervalllänge  $> n$  ist, dann erhöht das die Dauer eines PREAMBLE-Symbols ohne die Anzahl an Überlappungen im Worst Case zu reduzieren. Daraus folgt die optimale PS-Dauer von:

$$T_{PS} = \#I_{PS} \cdot \text{Impulsintervalllänge} \cdot TS = n^2 \cdot TS$$

Aufgrund der minimalen PS-Dauer von  $\min T_{PS} = 180 \cdot TS$  folgt  $n \geq \lceil \sqrt{180} \rceil = 14$ , und da  $n$  ungerade sein soll folgt daraus wiederum:

$$n \geq 15 + 2 \cdot k \text{ mit } k \in \mathbb{N}$$

#### Impulssequenz eines PREAMBLE-Symbols:

Es muss nun eine Impulssequenz der Länge  $n$  gefunden werden, bei welcher die Autokorrelationsgüte ausreichend gut ist. Als Korrelationssequenz wird auch wie im UWB-Standard eine ternäre Sequenz genommen, da diese einige Vorteile bringt:

1. gut geeignet, um die AGC einzustellen.
2. gute Korrelationseigenschaften vorhanden
3. aufgrund der Eigenschaft  $\#0 > 0$  gibt es nicht belegte Impulspositionen und damit sinkt die Kollisionswahrscheinlichkeit von 100 % für einen Fehler auf  $\left(1 - \left(1 - \frac{\#0}{n}\right)^2\right)$
4. Einhaltung der Anforderungen bzgl. der erlaubten PRF

Des Weiteren muss die  $PRF_{SYNC}$  von 15,6 MHz eingehalten werden.

Sei  $X$  die Anzahl an PREAMBLE-Symbolen, für die Synchronisation zwischen  $A$  und  $B$ .

Sei  $I = n$  die Anzahl an *time-slots* der Impulsverteilung.

Sei  $P = \#0 + \#1 + \# - 1 = n$  die Länge der Impulssequenz.

Sei  $\#_I = \#1 + \# - 1$  die Anzahl an Impulsen innerhalb eines PREAMBLE-Symbols.

Für  $A$  ist die Synchronisationsdauer  $T_{SYNCA}$ :

$T_{SYNCA} = (X + 2) \cdot T_{PS}$  (inkl. Pause von  $1 \cdot T_{PS}$ ) für  $(X+1)$  gesendete PS.

$$\text{Daraus folgt: } PRF_{SYNCA} = \frac{A \cdot \#_I \cdot (X+1)}{P \cdot I \cdot (X+2) \cdot TS} \leq 15,6 \text{ MHz} = \frac{1}{32 \cdot TS}$$

$$\Rightarrow A_{\#I} \leq \frac{n^2 \cdot (X+2)}{32 \cdot (X+1)}$$

Für  $B$  ist die Synchronisationsdauer  $T_{SYNCB}$ :

$T_{SYNCB} = (X+4) \cdot T_{PS} + T_D + T_P + T_{Korr}$  (inkl. Pause von  $1 \cdot T_{PS} + T_D + T_P + T_{Korr}$ )  
und für  $(X+3)$  gesendete PS.

$$\Rightarrow T_{SYNCB} \leq (X+4) \cdot T_{PS} + 32 \cdot TS + 16 \cdot TS(X+4) \cdot T_{PS} + 48 \cdot TS$$

$$\text{Daraus folgt: } PRF_{SYNCB} = \frac{B_{\#I} \cdot (X+3)}{P \cdot T \cdot (X+4) \cdot TS + 48 \cdot TS} \leq 15,6 \text{ MHz} = \frac{1}{32 \cdot TS}$$

$$\Rightarrow B_{\#I} \leq \frac{n^2 \cdot (X+4) + 48}{32 \cdot (X+3)}$$

$$A_{\#I} - B_{\#I} = \frac{n^2 \cdot (X+2)}{32 \cdot (X+1)} - \frac{n^2 \cdot (X+4) + 48}{32 \cdot (X+3)}$$

Da  $\#I$  sowohl für  $A$  als auch für  $B$  genutzt werden muss, folgt:

$$\#I = \min(A_{\#I}, B_{\#I}) \quad (3.9)$$

Es bleibt zu ermitteln, wann  $A_{\#I} \leq B_{\#I}$  oder  $A_{\#I} \geq B_{\#I}$  ist.

$$\text{sgn}(A_{\#I} - B_{\#I}) = \text{sgn}(n^2 \cdot (X+2) \cdot (X+3) - n^2 \cdot (X+4) \cdot (X+1) - 48 \cdot (X+1))$$

$$= \text{sgn}(n^2 \cdot (X^2 + 5 \cdot X + 6) - n^2 \cdot (X^2 + 5 \cdot X + 4) - 48 \cdot X - 48)$$

$$= \text{sgn}(2 \cdot n^2 + 48 \cdot X - 48) \quad (3.10)$$

Aus den Formeln (3.9) und (3.10) folgt:

- wenn  $2 \cdot n^2 + 48 \cdot X - 48 \leq 0 \Rightarrow \#I = B_{\#I}$
- wenn  $2 \cdot n^2 + 48 \cdot X - 48 \geq 0 \Rightarrow \#I = A_{\#I}$

Tabelle 3.9 zeigt die maximal mögliche Anzahl an Impulsen, je nachdem welches  $n$  (Impulsintervalllänge) bzw. welches  $X$  (Anzahl an PS für die Synchronisation) gewählt wurde. Tabelle 3.10 zeigt die jeweils mögliche Autokorrelationsgüte an.

**Tabelle 3.9:** Anzahl an Impulsen  $\#I$  in einem PREAMBLE-Symbol in Abhängigkeit von  $n$  und  $X$

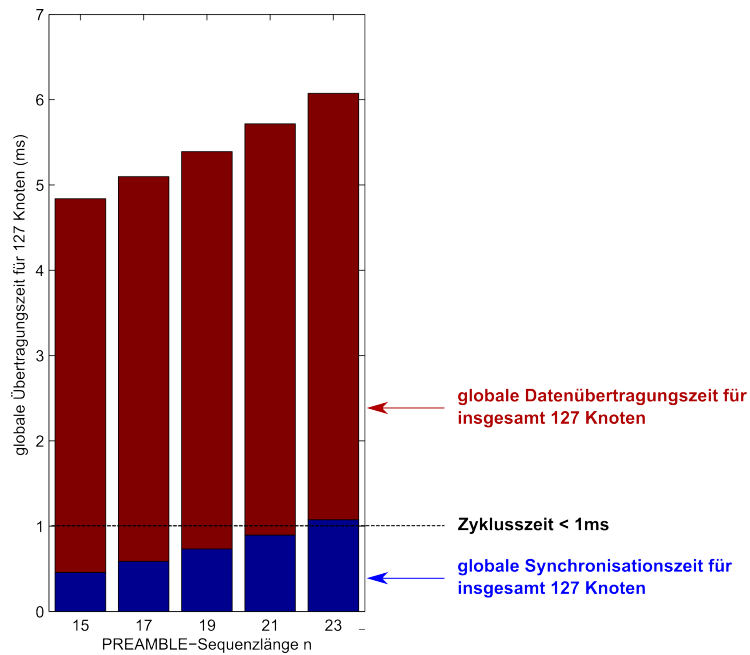
$n$	$X$															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
15	9	8	8	8	8	7	7	7	7	7	7	7	7	7	7	7
17	11	11	10	10	10	10	10	9	9	9	9	9	9	9	9	9
19	14	13	13	13	12	12	12	12	12	12	12	12	12	12	11	11
21	17	16	16	15	15	15	15	15	15	14	14	14	14	14	14	14
23	21	20	19	19	18	18	18	18	18	17	17	17	17	17	17	17



**Tabelle 3.10:** maximale  $A_p$  der PREAMBLE-Symbole in Abhängigkeit von  $n$  und  $X$ 

$n$	$X$															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
15	14	12	12	12	12	10	10	10	10	10	10	10	10	10	10	10
17	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14
19	18	18	18	18	16	16	16	16	16	16	16	16	16	16	16	16
21	20	20	20	20	20	20	20	20	20	18	18	18	18	18	18	18
23	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22

Abbildung 3.26 stellt die minimale erreichbare Zykluszeit, für verschiedene  $n$  und konstanter Anzahl an PREAMBLE-Symbolen  $X = 8$  für die Synchronisationsphase, dar.



**Abb. 3.26:** Globale Übertragungszeit  $T_{TRANS}$  für verschiedene PS-Sequenzlängen  $n$  und für  $X = 8$ . In Blau ist die globale Synchronisationszeit  $T_{SYNC}$  und in Rot ist die globale Datenübertragungszeit  $T_{DATA}$  für 127 Knoten angegeben.

Die globale Übertragungszeit  $T_{TRANS}$  berechnet sich anhand der globalen Synchronisationszeit  $T_{G-SYNC}$  und der globalen Datenübertragungszeit  $T_{G-DATA}$ , wie folgt:

$$T_{TRANS} = T_{G-SYNC} + T_{G-DATA} = 127 \cdot (n^2 \cdot X \cdot TS) + 127 \cdot 30896 \text{ ns}$$

Beispielsweise gilt für  $n = 21$  und  $X = 8$ :

$$T_{G-SYNC} = 127 \cdot (21 \cdot 21 \cdot 8 \cdot 2 \text{ ns}) = 896112 \text{ ns}$$

$$T_{G-DATA} = 127 \cdot 30896 \text{ ns} = 3923792 \text{ ns}$$

$$T_{TRANS} = T_{G-SYNC} + T_{G-DATA} = 4,82 \text{ ms}$$

Bzgl. der Synchronisationszeit ist die Anforderung einer unter 1 ms, für  $n \leq 21$  bereits erreicht. Um die Anforderungen von einer Zykluszeit  $\leq 1$  ms erfüllen zu können, muss die Datenrate erhöht werden.

Eine Möglichkeit die Datenrate weiter zu erhöhen, ist die Burst-Positionen effizienter auszunutzen. Da bewiesen wurde, dass zwischen zwei Impulsen während einer Vollduplex-Übertragung mindestens 3 *time-slots* dazwischen liegen, so ist es möglich, diese Slots zu benutzen, um mit einem Bit mehr Informationen zu übertragen. Wenn die 3 *time-slots* links von  $A_k$  bzw.  $B_k$  ausgenutzt werden, können damit insgesamt 4 Bit an Daten übertragen werden. Allerdings erhöht sich damit die Burst-Fehlerwahrscheinlichkeit. Denn jedes Mal wenn ein Impuls verloren geht oder falsch erkannt wurde, entspricht dies einem Fehler von 4 Bit in Folge. Daher können die FEC-Overhead Verbesserungen hier nicht angewendet werden und es muss wieder auf den Reed-Solomon-Code zurückgegriffen werden. In diesem Fall dauert eine komplette Übertragung:

$$T_{G-DATA} = ((20 + (48 + 6 + 6) \cdot 8) \cdot 64/4) \cdot 127 \text{ ns} = 8000 \cdot 127 \text{ ns} = 1016000 \text{ ns}$$

$$T_{TRANS} = T_{G-SYNC} + T_{G-DATA} = 896112 \text{ ns} + 1016000 \text{ ns} \approx 1,88 \text{ ms}$$

Wird komplett auf die Hopping-Sequenzen verzichtet und festlegt, dass sowohl  $A$  als auch  $B$  alle 16 verfügbaren *time-slots* für die Datenübertragung nutzen können, so werden pro Impuls 5 Bit übertragen und insgesamt verringert sich die Übertragungszeit auf:

$$T_{G-DATA} = ((20 + (48 + 6 + 6) \cdot 8) \cdot 64/5) \cdot 127 \text{ ns} = 6400 \cdot 127 \text{ ns} = 812800 \text{ ns}$$

$$T_{TRANS} = T_{G-SYNC} + T_{G-DATA} = 896112 \text{ ns} + 812800 \text{ ns} \approx 1,71 \text{ ms}$$

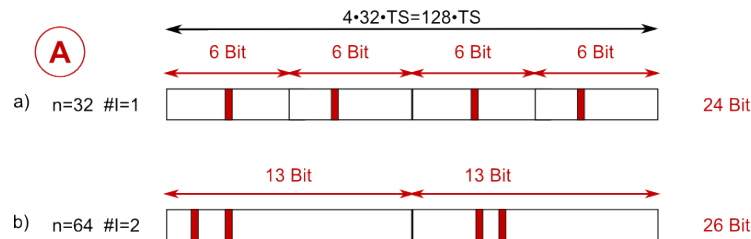
Um die Datenrate weiter zu erhöhen, muss die Symboldauer sowie Impulsanzahl vergrößert werden. In Abbildung 3.27 ist ein Beispiel für ein Sensorknoten gegeben. In Abbildung 3.27 a) ist die Symboldauer  $n = 32$  *time-slots*. In solch einem Intervall ist es erlaubt nur einen Impuls auszusenden um den Durchschnitt von 1 Impuls alle  $32 \cdot TS = 64$  ns bzw. die erlaubte PRF nicht zu überschreiten. Da dieser Impuls 32 Positionen einnehmen und positiv bzw. negativ sein kann, ist es möglich maximal 6 Bit an Daten zu übertragen.

$$\lfloor \log_2(\#P_{(32,1)}) \rfloor = \left\lfloor \log_2 \left( \frac{2 \cdot 32}{1!} \right) \right\rfloor = 6 \text{ Bit}$$

In Abbildung 3.27 b) ist zu erkennen wie sich die Bitrate erhöht, wenn die Symboldauer vergrößert wird und mehr Impulse pro Symboldauer für die Datenübertragung genutzt werden. In diesem Beispiel werden anstatt 4 Symbole je 32 *time-slots* und einem Impuls, 2 Symbole je 64 *time-slots* und 2 Impulsen übertragen. Obwohl sich die Gesamtanzahl an ausgesendeten Impulsen nicht verändert, und sich somit auch der Durchschnitt von 1 Impuls alle 32 *time-slots* nicht ändert, erhöht sich jedoch die effektive Bitrate auf

$$\lceil \log_2(\#P_{(n,1)} + \#P_{(n,2)}) \rceil = \left\lceil \log_2 \left( \frac{2 \cdot n}{1!} + \frac{2 \cdot n \cdot 2 \cdot (n-1)}{2!} \right) \right\rceil = 13 \text{ Bit}$$

Der Nachteil einer solchen Erweiterung ist jedoch dass die Zuverlässigkeit stark abnimmt, da bereits ein Impuls-Fehler einem Burst-Fehler von 13 Bit entspricht.



**Abb. 3.27:** Verschiedene Bitraten mit 4 Impulsen innerhalb von 128 TS.

In a) werden die Bits mit einem Impuls  $\#I = 1$  innerhalb von einem Intervall der Länge  $n = 32$  kodiert. Je Intervall werden 6 Bit übertragen.

In b) werden die Bits mit zwei Impulsen  $\#I = 2$  innerhalb von einem Intervall der Länge  $n = 64$  kodiert. Je Intervall werden 13 Bit übertragen.

Bei einem Sensorknoten wird nur ein *time-slot* von 32 genutzt, dies ist eine sehr ineffiziente Ausnutzung der Übertragungszeit. Um die Übertragungszeit besser auszunutzen, wurden zwei Sensorknoten dazu gebracht, ihre Daten gleichzeitig innerhalb einer Symboldauer zu übertragen. Auch in diesem Fall wirkt sich eine Verlängerung der Symboldauer mit gleichzeitiger Erhöhung der Anzahl an Impulsen positiv auf die Bitrate aus, siehe dazu Abbildung 3.28. In Abbildung 3.28 c) überlappen sich die Symbolintervalle, aber da jeder Sensorknoten nur in einer der 2 Hälften des Symbolintervalls  $n' = n/2$  dessen Impulse überträgt, kommt es zu keinen Kollisionen. Da die Impulse der jeweiligen Sensorknoten nur noch die Hälfte aller Positionen zur Verfügung haben, reduziert sich die jeweilige Bitrate der Sensorknoten um 2 Bit, aber die Globale Bitrate ist fast doppelt so hoch. Da die Restriktion, 1 Impuls alle 32 *time-slots*, nur für einen einzelnen Sensorknoten gelten, wird diese nicht durch die Kombination zweier Sensorknoten verletzt. Sowohl für A als auch für B liegt der Durchschnitt der gesendeten Impulse immer noch bei 1 zu 32.

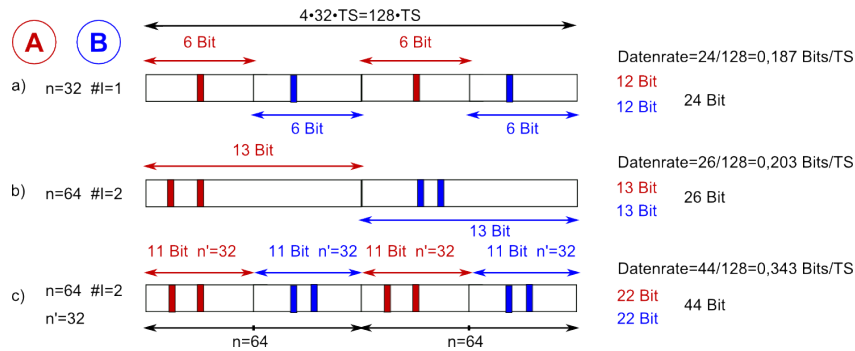


Abb. 3.28: Verschiedene Bitraten für verschiedene Intervalllängen für zwei Sensorknoten A und B.

a) und b) stellen Übertragungen der Impulse von A und B in getrennten Intervallen dar.  
 c) stellt eine Übertragung gemeinsam genutzter Intervalle für A und B dar.

Dieses Konzept lässt sich fast beliebig erweitern. In Abbildung 3.29 ist solch eine Erweiterung für drei Sensorknoten dargestellt.

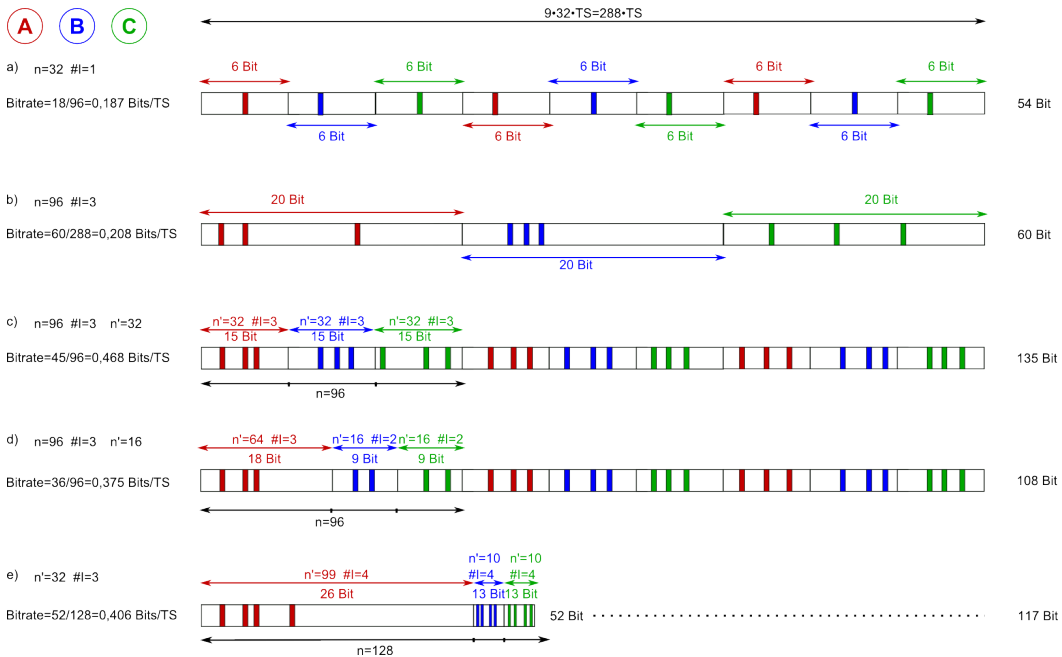


Abb. 3.29: Verschiedene Bitraten für verschiedene Intervalllängen für drei Sensorknoten A, B und C

a) und b) stellen Übertragungen der Impulse von A, B und C in getrennten Intervallen dar.  
 c) stellt eine Übertragung gemeinsam genutzter Intervalle für A, B und C dar.  
 d) und e) stellen Übertragungen mit gemeinsam genutzter Intervalle für A, B und C dar wobei  $\text{Bitrate}_A = \text{Bitrate}_B + \text{Bitrate}_C$  gilt.

Diese Erweiterung erlaubt zwar eine bessere Ausnutzung der leeren *time-slots* und ermöglicht einen beeindruckenden Anstieg der Bitrate (siehe Fall c)), aber es taucht ein weiteres Problem auf, welches bei zwei Sensorknoten nicht vorhanden war. Bei zwei Sensorknoten ist die übertragene Datenmenge von A gleich der Datenmenge von B. Dies gewährleistet eine optimale Übertragung im Worst Case von je 48 Byte sowohl von  $A \rightarrow B$  als auch von  $B \rightarrow A$ . Allerdings gilt dies nicht mehr für 3

Sensorknoten. In diesem Fall muss  $A$  immer länger als die beiden anderen aktiv sein. Im Worst Case haben  $B$  und  $C$  Ihre Datenübertragung von  $B \rightarrow A$  und  $C \rightarrow A$  bereits abgeschlossen während  $A$  erst die Hälfte der Daten  $A \rightarrow B$  und  $A \rightarrow C$  übertragen hat. Dieses Verhältnis verschlechtert sich mit steigender Anzahl an Knoten, welche zur selben Zeit mit  $A$  kommunizieren. Um eine optimale Auslastung auch im Worst Case zu erreichen, sollte die Bitrate von  $A$  der Summe aller Bitraten der restlichen Sensorknoten entsprechen. Außerdem sollten die Bitraten der anderen Sensorknoten identisch sein. In Abbildung 3.29 d) und e) sind zwei Beispiele für solch eine Verteilung für 3 Sensorknoten dargestellt.

### Verallgemeinerung:

Es wird nun untersucht, ob das zuvor vorgestellte Konzept für beliebige Sensorknoten erweiterbar ist. Außerdem werden optimale Parameter ermittelt.

Hier einige Annahmen und Notationen:

- Seien  $\{S_1, \dots, S_m\}$  die Anzahl an Sensorknoten und  $M$  der Master-knoten
- Sei  $k_M$  die Anzahl an Impulsen für  $M$
- Sei  $k_S$  die Anzahl an Impulsen für jeden Sensorknoten
- Sei  $n_M = k_M \cdot 32$  die Symboldauer für  $M$
- Sei  $n_S = k_S \cdot 32$  die Symboldauer für die Sensorknoten
- Sei  $n'_M$  bzw.  $n'_S$  das Impulsintervall für den Master bzw. der Sensorknoten
- Sei  $B_M$  die Anzahl an übertragbaren Bits im Impulsintervall  $n'_M$
- Sei  $B_S$  die Anzahl an übertragbaren Bits im Impulsintervall  $n'_S$

Um eine Impulsverteilung wie in Abbildung 3.29 für beliebig viele Knoten zu erhalten, müssen folgende Bedingungen Bitrate von  $M \geq m \cdot (\text{Bitrate von } S)$ , erfüllt sein. Die globale Bitrate  $GB$  ist die Anzahl aller übertragenen nutzbaren Bits sowohl des Masters  $B_M$  als auch der Sensorknoten  $B_S$  geteilt durch die globale Symboldauer.

- Mengen an übertragenen Bits pro Symbol ist  $BpS = \text{minimum} \left( 2 \cdot m \cdot \left\lfloor \frac{B_M}{m} \right\rfloor, 2 \cdot m \cdot B_S \right)$
- Globale Symboldauer ist gleich  $\text{minimum}(n'_M + m \cdot n'_S, n_M, n_S)$

Ziel ist es eine möglichst hohe globale Bitrate  $GB$  zu erreichen.

$$GB = \frac{\text{minimum} \left( 2 \cdot m \cdot \left\lfloor \frac{B_M}{m} \right\rfloor, 2 \cdot m \cdot B_S \right)}{\text{minimum}(n'_M + m \cdot n'_S, n_M, n_S)}$$

Die Komplette Übertragungszeit  $T_{TRANS}$  besteht aus der Übertragungszeit der Daten für  $k$  Knoten  $T_{DATA}^k$  plus der Synchronisationszeit für  $k$  Knoten  $T_{SYNC}^k$ .

Beide Zeiten sind von der Anzahl an beteiligten Knoten  $k$  abhängig. Da 127 eine Primzahl ist bleibt, immer ein Rest übrig, wenn versucht wird sich immer mit  $k$  Sensoren zu verbinden, um die ganze Nachricht an alle Sensorknoten zu übertragen. Folglich gilt  $127 = k \cdot y + k'$  für  $k > 0$  und  $0 \leq k' < k$ . Daraus kann folgende Übertragungszeit für eine komplette Nachricht von  $x$  Bits abgeleitet werden:

$$T_{TRANS} = (T_{DATA}^k + T_{SYNC}^k) \cdot y + (T_{DATA}^{k'} + T_{SYNC}^{k'})$$

$$T_{TRANS} = \left( \left\lceil \frac{2 \cdot x \cdot k}{BpS} \right\rceil + T_{SYNC}^k \right) \cdot y + \left( \left\lceil \frac{2 \cdot x \cdot k'}{BpS} \right\rceil + T_{SYNC}^{k'} \right)$$

Um die beste Bitrate ermitteln zu können, reicht eine Abschätzung von  $T_{TRANS}$  aus, welche sich nur auf  $k$  Sensorknoten bezieht. Außerdem kann  $T_{SYNC}^k$  vorerst mit  $k \cdot T_{PS} = k \cdot 7056 \text{ ns}$  approximiert werden. Daraus folgt nun folgende Abschätzung:

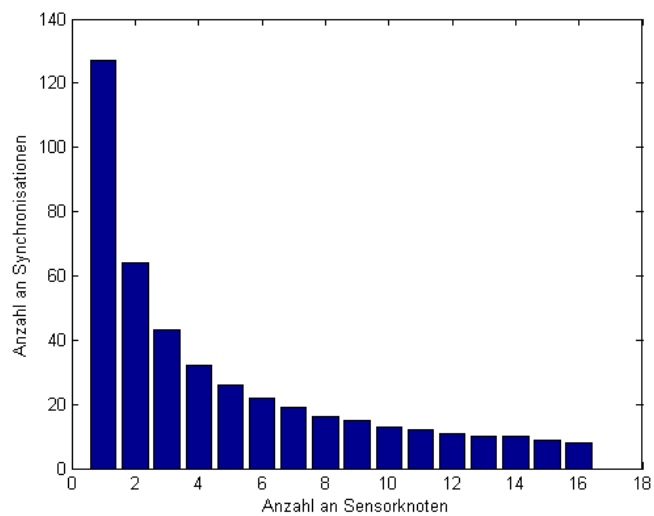
$$T_{TRANS} = \left( \left\lceil \frac{2 \cdot x \cdot k}{BpS} \right\rceil + k \cdot 7056 \text{ ns} \right) \cdot \left\lceil \frac{127}{k} \right\rceil$$

Da schon ein Impulsfehler ausreicht, um einen Burst-Fehler zu generieren ist es umso wichtiger so wenig wie möglich Daten zu übertragen, daher wurden für die folgenden Untersuchungen nur Nachrichten ohne FEC betrachtet. Dies ermöglicht es, die Nachricht von 500 Bits auf 384 zu reduzieren. Es wurden jeweils bis zu 16 Sensorknoten mit einer maximalen Impulsanzahl von 16 Impulsen untersucht.

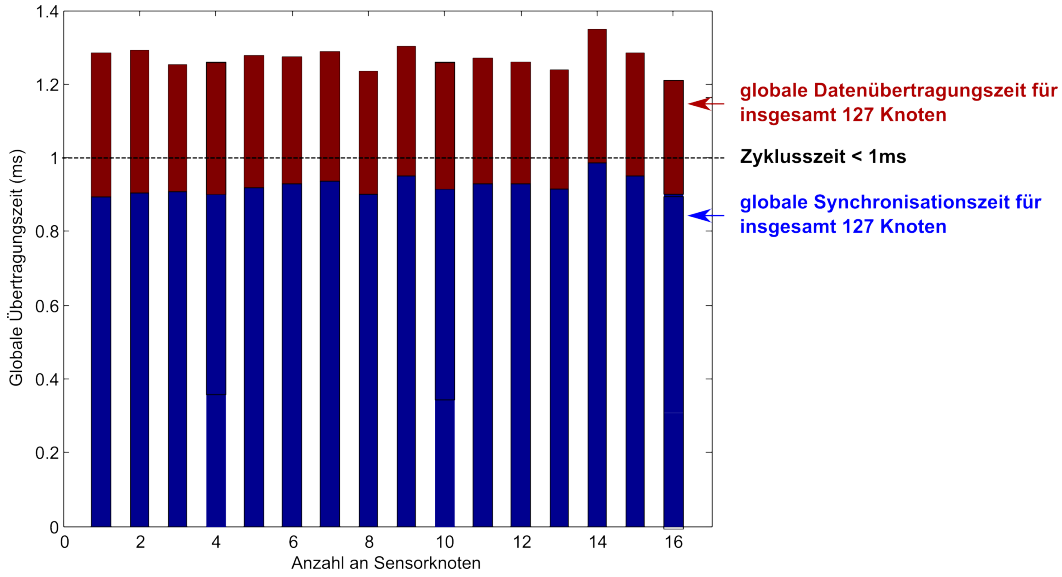
Tabelle 3.11 zeigt die bestmöglichen Übertragungszeiten je nach Anzahl der Sensorknoten. Abbildung 3.30 zeigt die Anzahl an notwendigen Synchronisationsvorgängen in Abhängigkeit von der Anzahl an genutzten Sensorknoten. In Abbildung 3.31 ist die globale Übertragungszeit angegeben. Die Übertragungszeit ist in zwei Teile geteilt. Der rote Teil ist die globale Datenübertragungszeit und der blaue Teil ist die globale Synchronisationszeit. Insgesamt geht daraus hervor, dass mit steigender Anzahl an Sensorknoten die globale Datenübertragungszeit leicht abnimmt, aber die Synchronisationszeit gleich bleibt, obwohl die Anzahl an Synchronisationsvorgängen abnimmt.

**Tabelle 3.11:** Vergleich des Impulsverhältnisses während der Synchronisation für verschiedene Sequenzlängen der PREAMBLE-Symbole

Knoten	TRANS (ms)	GB (%)	$n'_M$	$n'_S$	$k_M$	$k_S$	BpS	Symbollänge
1	0,3901	52,3438	252	252	16	16	268	512
2	0,3871	57,0313	433	31	16	14	292	512
3	0,3441	57,4219	453	17	16	12	294	512
4	0,3560	56,7050	474	12	16	11	296	522
5	0,3584	56,6406	414	17	16	7	290	512
6	0,3441	56,2500	396	16	16	6	288	512
7	0,3502	57,4219	453	8	16	8	294	512
8	0,3333	56,3147	427	7	15	7	272	483
9	0,3494	56,2500	407	7	15	6	270	480
10	0,3441	54,6875	330	12	16	4	280	512
11	0,3379	55,8594	378	10	16	4	286	512
12	0,3277	56,2500	396	9	16	4	288	512
13	0,3226	55,8594	378	7	16	4	286	512
14	0,3594	54,6875	330	10	16	3	280	512
15	0,3316	56,0166	407	5	15	4	270	482
16	0,3082	56,2500	396	5	16	4	288	512



**Abb. 3.30:** Anzahl an Synchronisationen abhängig von der Anzahl an genutzten Sensorknoten



**Abb. 3.31:**  $T_{TRANS}$  für verschiedene Sensorknoten  
Die globale Datenübertragungszeit  $T_{DATA}$  ist in rot und die globale Synchronisationszeit  $T_{SYNC}$  ist in blau dargestellt.

Der Hauptanteil der globalen Übertragungszeit wird durch die Synchronisation verursacht. Bisher konnte die globale Datenübertragung für alle 127 Sensorknoten auf unter 0,4 ms gesenkt werden, aber das Ziel einer globalen Übertragungszeit von  $< 1 \text{ ms}$  konnte nicht erreicht werden. Es ist daher notwendig, ein Synchronisationsverfahren zu entwickeln, welches für mehrere Sensorknoten geeignet ist und wobei die globale Synchronisationsdauer nicht direkt proportional zur Anzahl der Sensorknoten ist. Mit folgender Bedingung ist sichergestellt, dass die globale Synchronisationszeit mit der Anzahl an Sensorknoten abnimmt.

$$T_{SYNC}^k < k \cdot T_{SYNC}^1 \text{ mit } k > 1$$

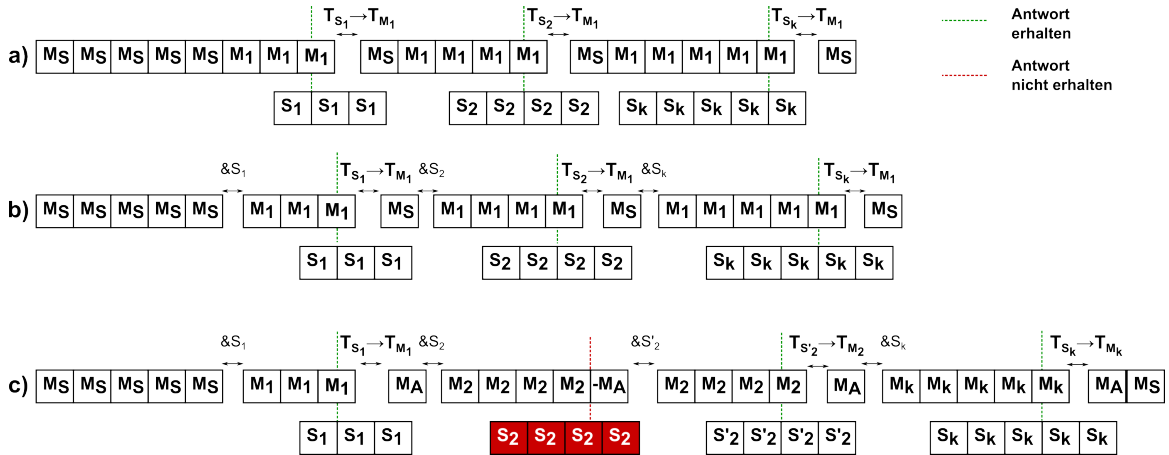
Des Weiteren sollte die Synchronisationsphase möglichst kurz und dennoch robust sein. Ein weiterer Punkt, welcher untersucht werden muss, ist wie sich die Anzahl an Impulsen auf die Zuverlässigkeit auswirkt. Es könnte notwendig sein, die Anzahl an Impulsen, auf Kosten der Übertragungszeit, zu begrenzen, um die Zuverlässigkeit zu erhöhen.

### Multi-Sensor Synchronisation

In Abbildung 3.32 werden 3 Varianten einer einfachen Synchronisation zwischen einem Master M und  $k$  Sensorknoten dargestellt. Alle 3 Varianten zeigen den Synchronisationsverlauf aus Sicht des Masters. Der Grundgedanke ist bei allen 3 Varianten gleich. Zuerst schickt der Master eine vordefinierte PREAMBLE  $M_S$ , um den Anfang eines neuen UWB-Frames zu signalisieren. Zusätzlich dient das PREAMBLE-Symbol  $M_S$  dazu, allen zuhörenden Sensorknoten die Möglichkeit zu geben, ihre jeweilige AGC einzustellen und sich auf den Master zu synchronisieren.



Ab hier unterscheiden sich dann die Varianten.



**Abb. 3.32:** 3 verschiedene Synchronisationsverfahren zwischen einem Master und mehreren Sensorknoten  
 a) Einfaches Synchronisationsverfahren  
 b) Synchronisationsverfahren mit Adressierung  
 c) Synchronisationsverfahren mit Adressierung und *acknowledgement*

**Fall a)**

Nach einer gewissen Zeit wird eine anderes PREAMBLE-Symbol  $M_1$  ausgesendet und richtet sich an  $S_1$ . Es wird erwartet, dass  $S_1$  auf jedes PREAMBLE-Symbol  $M_1$  mit einem anderen PREAMBLE-Symbol  $S_1$  antwortet. Diese Antwort kann vom Master erst als solche erkannt werden, wenn dieser die AGC passend zu  $S_1$  eingestellt hat, erst dann wird erneut ein PREAMBLE-Symbol  $M_S$  ausgesendet, welche als *acknowledge* dient und gleichzeitig  $S_1$  über dessen relativen Verzögerung zum Master bezüglich der PREAMBLE-Symbole  $M_1$   $T_{S_1 \rightarrow M_1}$  informiert. Diese Verzögerung ist als Abstand zwischen dem letzten PREAMBLE-Symbol  $M_1$  und dem aktuellen PREAMBLE-Symbol  $M_S$  kodiert. Dieses Verfahren wird solange ausgeführt, bis alle  $k$  Knoten mit dem Master synchronisiert sind und erst dann wird mit der tatsächlichen Datenübertragung begonnen. Weiterhin gilt für eine Verarbeitungszeit von 2 Taktzyklen je 32 ns und einer maximalen Distanz  $D_{max}$  zwischen Master und Sensorknoten von 30m folgendes:

$$T_{S_k \rightarrow M_1} \leq 2 \cdot T_{D_{max}} + T_P \leq 2 \cdot 50 \cdot TS + 2 \cdot 32 \cdot TS = 164 \cdot TS$$

Damit ergibt sich ein maximaler Overhead von zusätzlichen  $k \cdot 164 \cdot TS$

Diese Art der Synchronisation hat den Nachteil, dass die Reihenfolge mit welcher die Sensorknoten angesteuert, werden zuvor festgelegt sein muss und dies macht das ganze System unflexibel. Um diesen Zustand zu verbessern, wurde in Variante b) eine Art von Adressierung hinzugefügt.

**Fall b)**

Der Unterschied zu a) ist, dass hier bei jedem Wechsel von  $M_0$  zu  $M_1$  ein zusätzlicher

Abstand hinzugefügt wurde, welcher die Adresse des zu antwortenden  $S_k$  beinhaltet. Dieser zusätzliche zeitliche Overhead ist relativ zu der gesamten Synchronisationsdauer sehr gering. Der maximale zeitliche Overhead entspricht  $\sum_{k=1}^k (127 - k) \cdot TS$ .

Auch hier gibt es weiterhin Probleme die noch nicht berücksichtigt wurden. Es ist vorstellbar, dass ein Sensorknoten nicht auf die Anfrage des Masters antwortet oder dass der Master nicht in der Lage ist, die Antwort eines Sensorknoten zu detektieren. In solch einem Fall wäre die Synchronisation blockiert und es werden keine Nachrichten mehr übertragen. Ein weiterer Nachteil ist, dass die Sensorknoten eventuell auf ein PREAMBLE-Symbol  $M_1$  antworten, welches nicht für diese bestimmt war, weil der Anfang der Synchronisationsnachricht nicht empfangen wurde. Außerdem ist hier wie auch in a) die Anzahl an Knoten fest definiert und kann nicht variabel den aktuellen Empfangsbedingungen angepasst werden.

### Fall c)

Der Unterschied zu b) ist, dass hier jeder Sensorknoten  $S_k$  eine eigene Sequenz zugewiesen bekommt auf welche er antworten soll. Damit ist sichergestellt, dass auch jeder Sensorknoten  $S_k$  weiß, an welcher Stelle er im Datenstrom seine Bits zu übertragen hat. Des Weiteren wurde das *acknowledge*  $M_S$  durch  $M_A$  ersetzt und ein *not acknowledge*  $-M_A$  hinzugefügt. Mit dem *not acknowledge* wird signalisiert, dass der Master die Antwort des Sensorknotens nicht erkannt hat. Dies kann aufgrund von schlechten Empfangsbedingungen seitens des Masters und/oder des Sensorknotens geschehen. Dies ermöglicht dem Master eine sehr flexible Handhabung, wenn ein Sensorknoten  $S_k$  nicht antwortet. Es gibt 3 mögliche Szenarien in solch einem Fall:

1. Es kann versucht werden den Knoten  $S_k$  wiederholt anzusprechen bis dieser reagiert.
2. Der Knoten wird durch einen anderen Sensorknoten  $S'_k$  ersetzt und der Knoten  $S_k$  wird bei der nächsten Übertragung nochmal angesprochen.
3. Der Knoten  $S_k$  wird komplett ignoriert und es wird mit  $S_{k+1}$  fortgefahren.

Ein weiterer Fall, welcher denkbar wäre ist, dass die Synchronisation zwar erfolgreich war aber der Master die Antwort für zu unzuverlässig hält, aufgrund von Fehlern in der Korrelation. Dies kann geschehen, wenn mehrere Impulse zeitversetzt aufgrund eines oder mehrerer Echos ankommen. Der Master ist dann eventuell nicht mehr in der Lage die Daten dieses Sensorknoten korrekt zu empfangen und es besteht die Möglichkeit, dass diese Echosignale auch den Empfang der anderen Sensorknoten stören, weil die Echos bis in deren Zeitslots reichen. In solchen Fällen muss der Master für diesen Knoten dann auf eine Einzelverbindung (Voll duplex oder Simplex) ausweichen.

Um die Synchronisation abzuschließen wird nach dem letzten *acknowledge* nochmal ein  $M_S$  gesendet und dann erst startet die Datenübertragung. Ein weiterer Vorteil des Falles c) ist, dass  $k$  variabel gewählt werden kann. Beispielsweise könnte in einem

Netzwerk von 9 Knoten eine Nachricht zuerst an 5 Knoten und dann eine Nachricht an die restlichen vier Knoten geschickt werden. Somit wäre die Übertragungszeit optimal ausgenutzt.

### Synchronisationsdauer:

Die Synchronisationsdauer ist abhängig von der Anzahl an Knoten und von der Zeit, die der Master benötigt, um seine AGC korrekt einzustellen. Es ist daher sinnvoll alle Knoten (maximal 127) nach bestimmten Kriterien in Gruppen aufzuteilen. Mögliche Kriterien wären zum Beispiel der Standort und der AGC-Wert. Es ist sinnvoll Sensorknoten, welche dicht beieinander sind, zusammen zu gruppieren da aufgrund der kurzen Laufzeit der Signale eine Überlappung unwahrscheinlicher ist; außerdem sollten bei gleicher Sendeleistung auch der AGC-Wert beim Master für diese Knoten gleich oder ähnlich sein. Damit sollte es für den Master möglich sein, den korrekten AGC-Wert mit nur einer Änderung von  $\pm 1$  zu finden. Idealerweise sollte der AGC-Wert konstant bleiben. Außerdem sollten sich die Empfangsbedingungen innerhalb von 1 ms nicht drastisch verändern und mit Hilfe einer Datenbank, welche für alle Sensorknoten die letzten erfolgreichen AGC-Werte gespeichert hat, sollte es für den Master möglich sein, sich schnell an verschiedene Sensorknoten anzupassen.

Bei gleicher Annahme, wie zuvor für den Fall einer Vollduplex-Übertragung zweier Sensorknoten  $A$  und  $B$ , dass alle anderen Sensorknoten für die Synchronisation zum Master 8 PREAMBLE-Symbole benötigen entspricht dies folgender Synchronisationszeit.

$$T_{SYNC}^k = T_{MS} \cdot (8 + 1) + k \cdot 5 \cdot T_{M_k} + k \cdot T_{M_A} + \sum_{i=1}^k T_{S_k \rightarrow M_k} + \sum_{i=1}^k \&S_k \cdot 2 \text{ ns}$$

$$T_{SYNC}^k = (9 + 6 \cdot k) \cdot T_{PS} + \sum_{i=1}^k T_{S_k \rightarrow M_k} + \sum_{i=1}^k \&S_k$$

Da die Werte  $T_{S_k \rightarrow M_k}$  und  $\&S_k$  variabel sind ist auch  $T_{SYNC}^k$  variabel.

Um zu überprüfen, ob die Bedingung

$$T_{SYNC}^k < k \cdot T_{SYNC}^1 \text{ mit } k > 1$$

eingehalten wird, muss die obere und untere Schranke für  $T_{SYNC}^k$  berechnet werden.

### Obere Schranke:

Die einzelnen Variablen können wie folgt abgeschätzt werden.

$$\begin{aligned} \sum_{i=1}^k T_{S_k \rightarrow M_k} &\leq \sum_{i=1}^k (2 \cdot T_{D_{max}} + T_P) = \sum_{i=1}^k (2 \cdot 50 \cdot TS + 2 \cdot 32 \cdot TS) \\ &= k \cdot 164 \cdot TS = k \cdot 328 \text{ ns} \end{aligned}$$

Da mindesten jeder Sensorknoten in einem Übertragungszyklus vorkommt und somit auch jede Adresse, so kann eine durchschnittliche Adresslänge von  $127/2 \approx 64$  angenommen werden ohne das Resultat zu verfälschen. Demzufolge ergibt sich folgende Abschätzung.

$$\sum_{i=1}^k \&S_k \cdot 2 \text{ ns} \leq 128 \cdot k \text{ ns}$$

Daraus folgt nun:

$$T_{SYNC}^k \leq (9 + 6 \cdot k) \cdot T_{PS} + k \cdot 328 \text{ ns} + k \cdot 64 \text{ ns} \leq (9 + 6 \cdot k) \cdot T_{PS} + k \cdot 456 \text{ ns}$$

**Untere Schranke:**

$$\sum_{i=1}^k T_{S_k \rightarrow M_k} = \sum_{i=1}^k (2 \cdot T_D + T_P) \geq \sum_{i=1}^k (2 \cdot 1 + 32) = 34 \cdot k \text{ ns}$$

$$\sum_{i=1}^k \&S_k \geq \sum_{i=1}^k i = \frac{k \cdot (k+1)}{2}$$

$$\Rightarrow T_{SYNC}^k \geq (9 + 6 \cdot k) \cdot n^2 \text{ ns} + 34 \cdot k \text{ ns} + \frac{k \cdot (k+1)}{2} \cdot 2 \text{ ns}$$

Ausgehend von der oberen sowie unteren Schranke gilt folgendes:

$$\begin{aligned} T_{SYNC}^k - k \cdot T_{SYNC}^1 &\leq (9 + 6 \cdot k) \cdot T_{PS} + k \cdot 456 \text{ ns} - k \cdot ((9 + 6 \cdot k) \cdot T_{PS} + 36 \text{ ns}) \\ &\leq (9 - 9 \cdot k) \cdot T_{PS} + 420 \cdot k \text{ ns} \\ &\leq (9 - 9 \cdot k) \cdot n^2 \text{ ns} + 420 \cdot k \text{ ns} \end{aligned}$$

Für eine Länge  $n > 7$  des PREAMBLE-Symbols und  $k > 1$ , gilt somit immer:

$$(9 - 9 \cdot k) \cdot n^2 \text{ ns} + 420 \cdot k \text{ ns} < 0$$

Daraus folgt:

$$T_{SYNC}^k - k \cdot T_{SYNC}^1 < 0 \Rightarrow T_{SYNC}^k < k \cdot T_{SYNC}^1$$

**Globale Übertragungszeit:**

Wird die obere Schranke in der Übertragungszeitformel eingesetzt und wird angenommen, dass  $127 = k \cdot y + k'$  für  $k > 0$  und  $0 \leq k' < k$ , so folgt:

$$T_{TRANS} = \left( \left\lceil \frac{2 \cdot x \cdot k}{BpS} \right\rceil + (9 + 6 \cdot k) \cdot T_{PS} + k \cdot 456 \text{ ns} \right) \cdot y + \left( \left\lceil \frac{2 \cdot x \cdot k'}{BpS} \right\rceil + (9 + 6 \cdot k') \cdot T_{PS} + k' \cdot 456 \text{ ns} \right)$$

Für eine Abschätzung der Übertragungszeit reicht es, eine obere Schranke für  $T_{TRANS}$  zu definieren.

$$T_{TRANS} \leq \left( \left\lceil \frac{2 \cdot x \cdot k}{BpS} \right\rceil + (9 + 6 \cdot k) \cdot T_{PS} + k \cdot 456 \text{ ns} \right) \cdot (y + 1)$$

In Abbildung 3.33 ist die globale Übertragungsdauer für verschiedene Längen der Impulssequenz  $n = \{11, 13, 15, 17, 19, 21\}$  dargestellt. Aus Abbildung 3.33 ist er-

sichtlich, dass eine Übertragung  $< 1 \text{ ms}$  möglich ist, wenn die Länge der Impulssequenz  $n \leq 19$  ist.

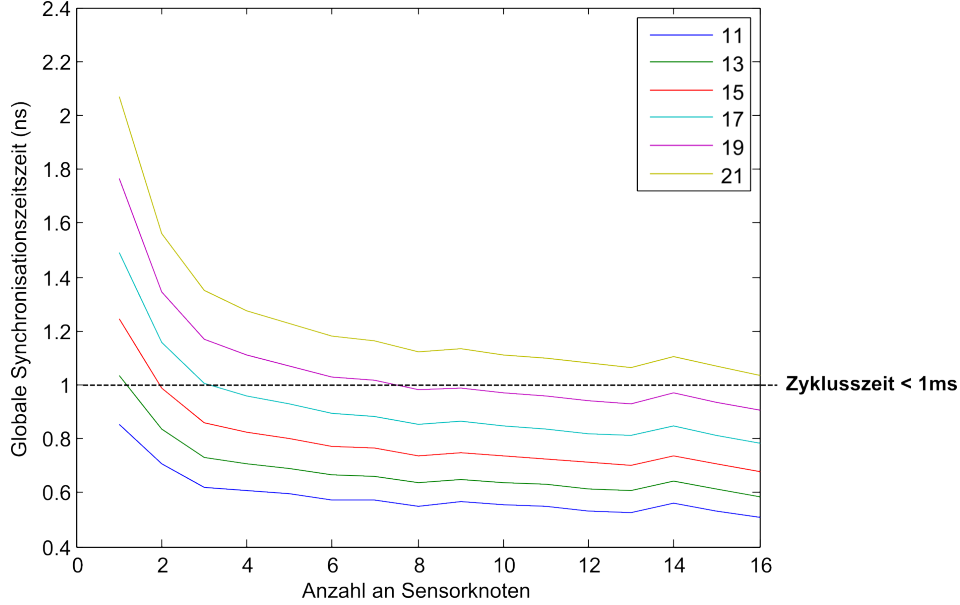


Abb. 3.33: Globale Übertragungsdauer  $T_{TRANS}$  bei verschiedenen Längen der Impulssequenz.

### UWB-Regulierungen:

Es bleibt zu berechnen, ob die Synchronisation mit diesen PREAMBLE-Symbolen die UWB-Regulierungen nicht verletzen. Wie schon zuvor muss die optimale Anzahl an Impulsen gefunden werden, um eine PRF von 15,6 MHz nicht zu überschreiten. Dann ist die Anzahl an Impulsen in  $T_{SYNC}^k$  gleich  $(9 + 6 \cdot k) \cdot \#I$ . Daher muss folgende Bedingung erfüllt sein:

$$\frac{(9 + 6 \cdot k) \cdot \#I}{T_{SYNC}^k} \leq 15,6 \text{ MHz} = \frac{1}{32 \cdot TS}$$

Da  $T_{SYNC}^k \geq (9 + 6 \cdot k) \cdot n^2 \cdot 2 \text{ ns} + 34 \cdot k \text{ ns} + \frac{k \cdot (k+1)}{2} \cdot 2 \text{ ns}$  folgt:

$$\frac{(9+6 \cdot k) \cdot \#I \cdot TS}{T_{SYNC}^k} \leq \frac{(9+6 \cdot k) \cdot \#I}{(9+6 \cdot k) \cdot n^2 + 17 \cdot k + \frac{k \cdot (k+1)}{2}}$$

Daraus folgt:

$$\text{Wenn } \frac{(9+6 \cdot k) \cdot \#I}{(9+6 \cdot k) \cdot n^2 + 17 \cdot k + \frac{k \cdot (k+1)}{2}} \leq \frac{1}{32} \Rightarrow \frac{(9+6 \cdot k) \cdot \#I}{T_{SYNC}^k} \leq \frac{1}{32 \cdot TS}$$

Sei nun die Folge  $PRF_k$  wie folgt definiert:

$$PRF_k = \frac{(9 + 6 \cdot k) \cdot \#I}{(9 + 6 \cdot k) \cdot n^2 + 17 \cdot k + \frac{k \cdot (k+1)}{2}}$$

und seien

$$\begin{aligned} a &= (9 + 6 \cdot k) \cdot \#_I \\ b &= (9 + 6 \cdot k) \cdot n^2 + 17 \cdot k + \frac{k \cdot (k+1)}{2} \\ c &= 6 \cdot \#_I \\ d &= 6 \cdot n^2 + 17 + k + 1 \end{aligned}$$

dann gilt:  $PRF_k = \frac{a}{b}$  und  $I_{k+1} = \frac{a+c}{b+d}$

Daraus folgt:

$$PRF_k - PRF_{k+1} = \frac{a}{b} - \frac{a+c}{b+d} = \frac{a \cdot (b+d) - (a+c) \cdot b}{a \cdot (b+d)} = \frac{a \cdot d - c \cdot b}{a \cdot (b+d)}$$

Da außerdem gilt:  $a \cdot d - c \cdot b = 9 \cdot (17 + k + 1) + \frac{k \cdot (k+1)}{2} > 0$  und  $a \cdot (b+d) > 0$  gilt auch  $PRF_k - PRF_{k+1} > 0$  und somit ist bewiesen, dass  $\max(PRF_k) = PRF_1$  ist.

Demzufolge reicht es aus die optimale Anzahl an Impulsen  $\#_I$  für  $PRF_1$  zu berechnen

$$PRF_1 \leq \frac{1}{32} \Rightarrow \frac{15 \cdot \#_I}{15 \cdot n^2 + 18} \leq \frac{1}{32} \Rightarrow \#_I \leq \left\lfloor \frac{n^2}{32} + \frac{19}{480} \right\rfloor$$

Tabelle 3.12 zeigt wie das Impulsverhältnis mit der Länge der PS variiert.

**Tabelle 3.12:** Vergleich des Impulsverhältnisses während der Synchronisation für verschiedene Sequenzlängen der PREAMBLE

n	$A_p$	$\#_I$	PRF
15	10	7	15,47 MHz
17	14	9	15,51 MHz
19	16	11	15,19 MHz
21	18	13	14,70 MHz
23	20	16	15,09 MHz
25	24	19	15,17 MHz

Der beste Kompromiss zwischen Reichweite und Zuverlässigkeit ist eine PREAMBLE-Symbollänge von  $n = 17$  mit  $\#_I = 9$ . Daraus resultiert eine Übertragungsdauer von unter 1 ms für 127 Sensorknoten mit jeweils 48 Byte an Daten in beide Richtungen, wenn  $k \geq 4$ , siehe Abbildung 3.33. Das Ziel einer Übertragung innerhalb 1 ms ist somit erreicht.

Obwohl das Ziel einer Übertragungszeit unter 1 ms erreicht wurde, so wurde noch nicht gezeigt, ob die Vorgaben bzgl. Zuverlässigkeit erfüllt sind. Dies wird im nächsten Abschnitt untersucht.

### 3.4 Zuverlässigkeit

Bisher wurde hauptsächlich die Latenzzeit reduziert und es wurde ein möglichst robustes Synchronisationsverfahren entwickelt. Diese Änderungen hatten aber zur Folge, dass die vorgesehenen Mechanismen zur Fehlerkorrektur aus dem UWB-Standard (Reed-Solomon, Hamming-Code und Faltungskodierer) entfernt wurden. Demzufolge müssen andere Maßnahmen ergriffen werden, um eine Fehlerkorrektur zu ermöglichen. In diesem Abschnitt wird die Datenübertragung so modifiziert, dass Fehler erkannt und korrigiert werden können. Dabei wird darauf geachtet, dass die Anforderung bzgl. Datenübertragungsrate weiterhin erfüllt werden.

Zur besseren Übersicht der Impulssequenzen, alle positiven Impulse durch '+', alle negativen Impulse durch '-' und alle leeren Impulse durch '.' dargestellt.

#### Zuverlässigkeitsanforderungen der Automatisierungsindustrie:

Es muss untersucht werden, wie hoch die Zuverlässigkeit jeder einzelnen Punkt-zu-Punkt Verbindung sein muss, um die geforderte Anforderung der Automatisierungsindustrie von  $10^{-9}$  zu gewährleisten.

Sei  $X$  die Wahrscheinlichkeit, dass ein UWB-Frame fehlerhaft zwischen zwei UWB-Knoten (Master  $\rightarrow$  Slave bzw. Slave  $\rightarrow$  Master) übermittelt wurde. Daraus ergeben sich folgende Resultate:

- Fehlerwahrscheinlichkeit  $E_k$  der Teilübertragung vom Master  $\rightarrow S_k \rightarrow$  Master ist  

$$E_k = 1 - (1 - X)(1 - X) = 1 - (1 - X)^2$$
- Fehlerwahrscheinlichkeit, dass eine Teilübertragung  $E_k$   $m$ -mal in Folge fehlschlägt ist:  

$$E_k^m = (1 - (1 - X)^2)^m$$
- Fehlerwahrscheinlichkeit  $E^{(n,m)}$ , dass bei einer  $m$ -fachen kompletten Übertragung für  $n$  Sensoren bei mindestens einem Sensor  $S_k$  die Übertragung  $m$ -mal fehlschlägt ist:  

$$E^{(n,m)} = 1 - \prod_{k=1}^n (1 - E_k^m) = 1 - (1 - (1 - X)^2)^{n \cdot m} \Rightarrow X = 1 - \sqrt[1 - \sqrt[n \cdot m]{1 - E^{(n,m)}}]{1 - E^{(n,m)}}$$
- für  $E^{(n,m)} = 10^{-9}$ ,  $m = 3$  und  $n = 127$  erhalten wir  

$$E_k = \sqrt[3 \cdot 127]{1 - E^{(127,3)}} \approx 10^{-11}$$

$$X = 1 - \sqrt{1 - E_k}$$

$$X \approx 10^{-6}$$

Dies bedeutet also, dass die Wahrscheinlichkeit eines Fehlers bei einer einzelnen Transmission bei  $10^{-6}$  liegen muss.

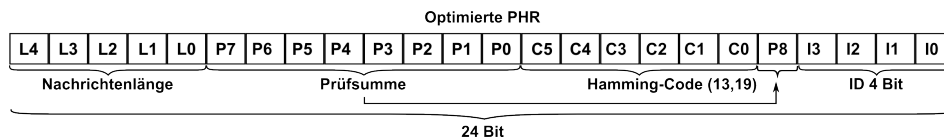
### 3.4.1 Fehlererkennung

Um Fehler zu erkennen werden für gewöhnlich Prüfsummen wie z.B. zyklische Redundanzprüfung CRC (Cyclic Redundancy Check) eingesetzt. Diese Prüfsummen sind auch sehr zuverlässig was das Erkennen von einzelnen bzw. gebündelten Bit-Fehler angeht. Es gibt auch Prüfsummen wie z.B. den Hamming-Code, welche auch eine Fehlerkorrektur ermöglichen. In diesem Fall ist der Hamming-Code allerdings ungeeignet, denn es können nur gebündelte Bit-Fehlern auftreten. Ein weiterer Nachteil von ECCs (engl. *Error-Correcting Code*), welche in der Lage sind eine geringe Anzahl an Fehler zu korrigieren, ist der große Overhead. Dieser Overhead erhöht die Datenübertragungszeit erheblich. Außerdem muss zwischen globaler Fehlererkennung und lokaler Fehlererkennung differenziert werden. Globale Fehlererkennung kann mittels einfacher Methoden wie Prüfsummen Paritybit usw. realisiert werden aber diese geben keine genaue Auskunft über die Art des Fehlers und der Fehlerhäufigkeit.

Es ist wichtig eine Prüfsummenmethode zu wählen, welche es zuverlässig ermöglicht einen Fehler zu erkennen auch wenn die Prüfsumme selbst fehlerhaft übermittelt wird. Des Weiteren sollte diese Prüfsumme möglichst einfach zu berechnen und kurz sein, um die Komplexität der Hardware sowie den Energieverbrauch und die Verarbeitungszeit nicht zu erhöhen. Eine einfache Prüfsumme wäre die binäre Quersumme der gesamten Nachricht oder anders ausgedrückt die Anzahl an Bits mit dem Wert 1. Da am Anfang einer Übertragung die Nachrichtenlänge  $> 0$  übertragen wird ist mindestens ein Bit ungleich 0 und somit ist auch die Quersumme aller Bits ungleich 0.

Üblicherweise werden Prüfsummen erst am Ende der Nachricht übertragen. Da aber in diesem Fall mit mehreren Knoten und mit verschiedenen Nachrichtenlängen gleichzeitig kommuniziert wird, ist es besser die Prüfsumme vor den eigentlichen Nutzdaten in den PHR zu integrieren.

Die Prüfsumme besteht aus 9 Bit, da die Länge der Nachricht kleiner als 512 Bit = 64 Byte ist.



**Abb. 3.34:** Optimierte PHR inkl. Nachrichtenlänge, Prüfsumme und ID. Die ersten 13 Bit des PHR-Teils wird durch einen Hamming-Code(13,19) geschützt.

Die ID zeigt an wie oft diese Nachricht wiederholt wurde. Der PHR hat demzufolge eine Länge von  $24 \text{ Bit} = 3 \text{ Byte}$ . Der Worst Case wäre also eine Gesamtlänge von  $(48 + 3) \cdot 8 = 408 \text{ Bit}$

Die Länge und die Prüfsumme ist durch den Hamming-Code zusätzlich ab-



gesichert. Das höchste Bit der Prüfsumme konnte nicht durch den vorhandenen Hamming-Code abgedeckt werden, da dieser auf 13 Bit beschränkt ist. Dies ist aber kein großer Nachteil. Denn das 9-te Bit der Prüfsumme ist nur in den seltensten Fällen gesetzt. Dies kann nur bei einer Nachrichtenlänge von mindestens 32 Byte geschehen und auch nur dann, wenn nur Einsen übertragen wurden. Bei einer Maximallänge von 48 Byte kann dies nur geschehen, wenn das Verhältnis "1 zu 0" gleich "67% zu 33 %" ist.

Die Wahrscheinlichkeit, dass eine Prüfsumme falsch ist und dennoch als korrekt interpretiert wird, ist sehr gering. Dies ist nur möglich wenn sowohl die Daten als auch die Prüfsumme falsch sind und die Anzahl an Bits dennoch mit der Prüfsumme übereinstimmen.

Es können 3 Arten von Fehler in einem Datensymbol mit  $k$  Impulsen auftreten.

1. **Fehlende Impulse:** Aufgrund eines schwachen Signals und einer zu hoch eingestellten AGC kann es dazu kommen, dass Impulse nicht detektiert werden.
2. **Zusätzliche Impulse:** Aufgrund eines schwachen Signals und einer zu niedrig eingestellten AGC kann es dazu kommen, dass weitere Impulse detektiert werden.
3. **Vertauschte Impulse:** Theoretisch könnte es dazu kommen, dass gleichzeitig einige Impulse fehlen und andere dazukommen, so dass dann immer noch  $k$  Impulse beim Empfänger detektiert werden. Dieser Fall sollte eigentlich nicht auftreten da sich die Übertragungsintervalle der verschiedenen Sensorknoten nicht überlappen sollten. Aber falls noch andere UWB-Sensorknoten in Reichweite aktiv sind, könnte dies geschehen.

In allen 3 Fällen ist es nicht möglich, ein fehlerhaftes Datensymbol ohne zusätzliche Informationen zu erkennen. Außerdem haben schon geringe Fehler in der Impulssequenz große Auswirkungen auf der dazugehörigen Bitsequenz.

Der Vorteil der Bit-Ebene ist die Einfachheit der Verarbeitung und der geringere Speicherbedarf. Der Nachteil allerdings ist, dass es nur Burst-Fehler geben kann, da ein Fehler in der Impulssequenz nicht zu einem sondern zu mehreren Fehlern in der Bitsequenz führen kann. Folgendes Beispiel verdeutlicht dies.

Beispiel-Impulssequenz={ . + . - . + . }  $\Leftrightarrow$  {100101100}

Ein fehlender Impuls  $i_3$  führt zu folgender Sequenz { . . + . . . . + . }  $\Leftrightarrow$  {011110110}

Die originale Bitsequenz unterscheidet sich von der fehlerhaften Sequenz an 6 Stellen. Somit ist auch eine Mehrfachübertragung nicht geeignet, um diese Sequenz zu korrigieren. Dies liegt daran, dass von  $x$  übertragenen Sequenzen nicht unterschieden werden kann, welche davon korrekt sind. Da sich die Bitsequenzen in jedem Fehlerfall stark unterscheiden, ist auch eine Kombination der Bits nicht geeignet,

um die Originalsequenz wiederherzustellen. Die einzige Möglichkeit die richtige Bitsequenz zu finden ist die Nachricht als korrekt anzusehen, welche am häufigsten empfangen wurde. Dies setzt allerdings voraus, dass die Nachricht mindestens zweimal richtig empfangen wurde. Dies ist eine sehr starke Anforderung.

Eine Abschwächung der Anforderungen könnte erreicht werden, wenn es möglich wäre eine falsche Impulssequenz von einer richtigen Impulssequenz unterscheiden zu können.

Eine Möglichkeit einen Fehler zu erkennen, wäre beispielsweise immer eine Konstante Anzahl an Impulsen für die Datenübertragung zu nutzen. Dies hat allerdings den Nachteil, dass die Bitrate pro Datensymbol sinkt. Der Unterschied zwischen der Bitrate mit einer variablen Anzahl an Impulsen (bis zu  $k$  Impulsen)  $BM_v$  und der Bitrate mit konstanter Anzahl an Impulsen (genau  $k$  Impulse)  $BM_c$  ist in Tabelle 3.13 dargestellt.

$$BM_v = \lfloor \log_2(\#P_{(n,1)} + \#P_{(n,2)} + \dots + \#P_{(n,k)}) \rfloor$$

$$BM_c = \lfloor \log_2(\#P_{(n,k)}) \rfloor = \left\lfloor \log_2 \left( \frac{\prod_{i=0}^{k-1} 2 \cdot (n-i)}{k!} \right) \right\rfloor = \left\lfloor \log_2 \left( \frac{n!}{k! \cdot (n-k)!} \cdot 2^k \right) \right\rfloor$$

**Tabelle 3.13:** Unterschiedliche Übertragungszeiten für  $BM_v$  und  $BM_c$

Knoten	$BM_v$ TRANS (ms)	$BM_c$ TRANS (ms)	Differenz $BM_v - BM_c$
1	0,3901	0,5202	0,1301
2	0,3871	0,4524	0,0654
3	0,3441	0,4731	0,1290
4	0,3492	0,4540	0,1048
5	0,3584	0,4608	0,1024
6	0,3441	0,4731	0,1290
7	0,3502	0,4553	0,1051
8	0,3378	0,4460	0,1081
9	0,3441	0,4588	0,1146
10	0,3441	0,4300	0,0860
11	0,3379	0,4393	0,1014
12	0,3277	0,4403	0,1127
13	0,3226	0,4424	0,1198
14	0,3594	0,4674	0,1080
15	0,3316	0,4505	0,0983
16	0,3082	0,4150	0,1068

Im Schnitt verlängert sich die Datenübertragung um rund 100  $\mu s$ . Dies ermöglicht immer noch eine komplette Datenübertragung innerhalb von 1 ms.

### 3.4.2 Fehlererkorrektur

Aufgrund der zu erwartenden Burst-Fehler sind herkömmliche FEC ungeeignet da diese sehr viel Overhead generieren, welcher immer übertragen werden muss. Außerdem ist nicht sichergestellt, dass diese FEC dann auch wirklich alle Fehler korrigieren können und somit eine erneute Übertragung überflüssig machen.

Aufgrund der Fähigkeit mit mehreren Sensoren gleichzeitig kommunizieren zu können ist das Wiederholen der Nachricht eine sinnvolle Alternative. Außerdem ist es möglich dank der PPM mehrere fehlerhaft übertragene Nachrichten zu kombinieren und zu korrigieren. Dieses Verfahren ähnelt dem aus [76] [77] wobei hier Nachrichten mehrfach durch Weiterleitung anderer Sensorknoten empfangen und kombiniert werden.

Diese Kombination der Informationen kann sowohl auf Bit-Ebene als auch tiefer auf Impulsebene realisiert werden. Tabelle 3.14 zeigt wie eine Korrektur der Impulssequenz durch Mehrfachübertragung erfolgen kann.

Um den Unterschied zwischen Bit-Ebene und Impulsebene zu verstehen ist es notwendig die Binären Daten in Impulssequenzen umzuwandeln und vice versa. Diese Umwandlung ist im Falle von Impulssequenzen mit genau  $k$  Impulsen nicht trivial.

#### Umwandlungsalgorithmus: Bitsequenz zu Impulssequenz

Notationen

- Sei  $S_{m,l}$  die Anzahl an Möglichkeiten  $l$  Impulse auf  $m$  Stellen zu verteilen ohne Berücksichtigung der Impulspolaritäten.

$$S_{m,l} = \begin{cases} \frac{m!}{l!(m-l)!} & \text{wenn } m \geq l \\ 0 & \text{wenn } m < l \end{cases}$$

- Sei  $D$  die Dezimale Darstellung der binären Daten.
- Sei  $D_k = D \bmod 2^k$  und der Rest  $R = D - D_k \cdot 2^k$ .
- Sei  $I = (i_n, i_{n-1}, \dots, i_1)$  die ternäre Impulssequenz mit  $k$  Impulsen und  $n$  Stellen mit  $i_l = \pm 1$  oder  $0$  für  $1 \leq l \leq n$ .

Für jede Bitsequenz existieren  $k$  Werte  $x_1, x_2, \dots, x_k$ , so dass  $x_k > \dots > x_1 \geq 1$  und für alle  $x_l$  mit  $1 \leq l \leq k$  gilt  $I(x_l) = \pm 1$  und

Für alle  $x \neq x_l$  mit  $1 \leq l \leq k$  gilt  $I(x) = 0$

Dann gilt folgende Rekursion für  $0 \leq l \leq k$ :

$$x_l = i + 1 \text{ wenn } S_{i+1,l} > D_l \geq S_{i,l} \text{ für } 0 \leq i \leq n$$

$$D_{k-1} = D_k - S_{x_{k-1},k}$$

Die bisher berechneten  $x_l$  geben nur die Positionen der Impulse an, die Polarität wird mittels der binären Darstellung  $R$  von  $r$  berechnet. Die binäre Darstellung von  $r$  ist:

$$R = (r_k, \dots, r_1)$$

Für alle  $l \leq k$  gilt:

$$I(x_l) = \begin{cases} -1 & \text{wenn } r_k = 0 \\ +1 & \text{wenn } r_k = 1 \end{cases}$$

### Umwandlungsalgorithmus: Impulssequenz zu Bitsequenz

Um eine Impulssequenz mit  $k$  Impulsen und  $n$  Stellen in eine Bitsequenz der Länge  $BM_c$  umzuwandeln wird wie folgt vorgegangen.

Man entferne alle  $i_l = 0$  mit  $1 \leq l \leq n$  aus  $I = (i_n, i_{n-1}, \dots, i_1)$  unter Beibehaltung der Reihenfolge der Indizes. Das Resultat ist dann  $(I(x_k), I(x_{k-1}), \dots, I(x_1))$ . Daraus kann die binäre Darstellung  $r = (r_k, \dots, r_1)$  des Restes  $R$  wie folgt berechnet werden:

$$r(l) = \begin{cases} 0 & \text{wenn } x_l = -1 \\ 1 & \text{wenn } x_l = +1 \end{cases}$$

$$D_k = \sum_{l=1}^k S(x_l - 1, l)$$

Damit kann nun  $D$  wie folgt berechnet werden:

$$(D = R + D_k \cdot 2^k)$$

Die gesuchte Bitsequenz ist nun die binäre Darstellung von  $D$ .

### Beispiel:

Angenommen die Bitsequenz "110110010" = 434 soll in eine Impulssequenz der Länge  $n = 10$  mit  $k = 3$  Impulsen umgewandelt werden, dann müssen folgende Schritte ausgeführt werden.

1.  $D_k$  berechnen  $D_k = D_3 = D \bmod 2^3 = 869 \bmod 8 = 54$
2.  $R$  berechnen  $R = D - D_3 \cdot 2^3 = 869 - 108 \cdot 8 = 2$
3. Alle benötigten  $S_{m,l}$  berechnen
4. Alle  $x_l$  berechnen

Tabelle 3.14:  $S_{m,l}$  für Impulskodierung

$S_l$	m=10	m=9	m=8	m=7	m=6	m=5	m=4	m=3	m=2	m=1	m=0
l=3	120	84	56	35	20	10	4	1	0	—	—
l=2	45	36	28	21	15	10	6	3	1	0	—
l=1	10	9	8	7	6	5	4	3	2	1	0

$$D_3 = 108 \text{ und } S_{8,3} = 56 > D_3 = 54 \geq S_{7,3} = 35 \\ \Rightarrow x_3 = 8$$

$$D_2 = D_3 - 35 = 19 \text{ und } S_{7,2} = 21 > D_2 = 19 \geq S_{6,2} = 15 \\ \Rightarrow x_2 = 7$$

$$D_1 = D_2 - 15 = 4 \text{ und } S_{5,1} = 4 > D_1 = 4 \geq S_{4,1} = 3 \\ \Rightarrow x_1 = 5$$

5. Die Impulssequenz berechnen

$$R = 2 \Rightarrow r = "010" \Rightarrow i_8 = -1, i_7 = +1 \text{ und } i_5 = -1$$

$$I = "...+.-...."$$

Um eine Impulssequenz  $I = "...+.-...."$  in eine Bitsequenz umzuwandeln wird wie folgt vorgegangen.

1. Berechne alle  $x_l$  und  $r$  aus  $I$ :  
 $i_8 = -1, i_7 = +1, i_5 = -1 \Rightarrow (x_3, x_2, x_1) = (8, 7, 5) \text{ und } r = (r_3, r_2, r_1) = (0, 1, 0)$
2. Berechne  $D_3 = \sum_{l=1}^3 S(x_{l-1}, l) = 4 + 15 + 35 = 54$
3. Berechne  $D = D_3 \cdot 2^3 + R = 54 \cdot 8 + 2 = 434 = "110110010"$

### Impulsfehlerkorrektur

Zwar müssen bei einer Impulsfehlerkorrektur mehr Daten abgespeichert werden als bei einer Bitsequenzfehlerkorrektur, allerdings überwiegen die Vorteile deutlich. Die Tabelle 3.15 zeigt anhand eines Beispiels wie Impulsfehler korrigiert werden können. Es werden Impulssequenzen der Länge 16 mit 5 Impulsen verwendet. Für die Konvertierung in Bitsequenzen werden die Impulse von links nach rechts berücksichtigt. Dies hat zur Folge, dass bei mehr als 5 Impulsen, nur die ersten 5 von links berücksichtigt werden. Bei weniger als 5 Impulsen, wird angenommen, dass der Wert der Fehlenden Impulse 0 entspricht.

Abbildung 3.35 verdeutlicht wie durch eine Mehrfachübertragung verschiedenartige Fehler korrigiert werden können. Es sei bemerkt, dass auch dauerhaft falsch

empfangene Impulssequenzen korrigiert werden können. Allerdings ist bisher keine Korrektur bei nur einer Übertragung möglich.

Wie in Tabelle 3.15 gut erkennbar ist, haben schon geringe Impulsfehler große Auswirkungen auf die Bitsequenz. Dies ist einerseits ein Nachteil, aber andererseits kann dies in Kombination mit der Prüfsumme dazu verwendet werden die Impulssequenz zu korrigieren, wenn nur ein Fehler aufgetreten ist.



**Abb. 3.35:** Fehlerkorrektur durch Mehrfachübertragung. Die Fehlerhaften Stellen in der Übertragung sind rot markiert. Die Korrigierten Stellen in der Übertragung sind grün markiert.

**Tabelle 3.15:** Impulsfehlerkorrektur

	Impulssequenz	Bitsequenz	Dezimal
Original	..+.-.+--.....	01101100011010100	55508
Impulsauslöschung			
1. Übertragung	..+...+--.....	01011100011111000	47352
2. Übertragung	..+.-.+--.....	01101010111110100	54772
Rekonstruiert	..+.-.+--.....	01101100011010100	55508
Es werden mindestens 2 Übertragungen benötigt, wenn Impulse an verschiedenen Stellen ausgelöscht werden			
Impulsecho			
1. Übertragung	..+.-.+--.....+..	01101100011010100	55508
2. Übertragung	-.+.-.+--.....	111101010111101010	125674
Rekonstruiert	..+.-.+--.....	01101100011010100	55508
Es werden mindestens 2 Übertragungen benötigt, wenn Impulse an verschiedenen Stellen hinzugefügt werden			
Kombination			
1. Übertragung	..+...+--.....+..	01011011111011010	47066
2. Übertragung	..+.-...-.....	01100111111010000	53200
3. Übertragung	+...-.+--.....-	11010111101010100	110420
Rekonstruiert	..+.-.+--.....	01101100011010100	55508
Es werden mindestens 3 Übertragungen benötigt, um eine eindeutige Entscheidung treffen zu können.			

Um die Fehlerkorrektur besser zu verstehen folgt ein Beispiel für eine empfangene Impulssequenz  $I = ..+.-.+--.....+..$  mit  $k = 5$  und  $n = 16$ :

Annahmem:

- Die ganze Nachricht inklusive Prüfsumme wurde bis auf die Impulssequenz  $I$  korrekt empfangen.

- Bei der fehlerhaften Impulssequenz  $I$  handelt es sich nicht um die Prüfsumme selbst.
- sei  $Q$  die Quersumme der gesamten Nachricht
- sei  $Q'$  die Quersumme der gesamten Nachricht ohne der fehlerhaften Impulssequenz

Zunächst fällt auf, dass die Impulssequenz  $I$   $k + 1$  Impulse beinhaltet und somit fehlerhaft ist. Wahrscheinlich ist ein Impuls dazugekommen und die anderen 4 sind richtig.

Es kommen daher 6 Impulssequenzen in Frage, siehe Tabelle 3.16. Wenn eine dieser Impulssequenzen bzw. Bitsequenzen eine Quersumme von  $Q - Q'$  aufweist, so ist diese wahrscheinlich die richtige.

**Tabelle 3.16:** Impulsfehlerkorrektur mittels Prüfsumme

	Impulssequenz	Bitsequenz	Quersumme
1.	....+-.+..+..	01000101000001001	5
2.	..+...+-.+..+..	10010000101011001	7
3.	..+.-.-.-.-.+..	10100010011110001	8
4.	..+.-.+.-.-.+..	10100100101110101	9
5.	..+.-.+-.+..+..	10100110000010101	7
6.	..+.-.+-.+..+..	10100110001110100	8

Wenn  $Q - Q' = 5$  oder  $9$  dann ist die Entscheidung für eine bestimmte Impulssequenz eindeutig.

Wenn  $Q - Q' = 7$  oder  $8$  dann ist die Entscheidung für eine bestimmte Impulssequenz zweideutig. Eine weitere Übertragung wird benötigt, um eine Entscheidung treffen zu können.

Sollte die Quersumme  $Q - Q'$  nicht in der Tabelle 3.16 auftauchen, so ist die Nachricht in mehr als nur einer Stelle Fehlerhaft. Diese zusätzliche Information kann nützlich sein bei der Rekonstruktion der Nachricht aus mehreren Übertragungen.

Dieses Verfahren ist natürlich für mehr als nur eine Impulssequenz gleichzeitig anwendbar, aber mit steigender Anzahl an Impulssequenzen steigt auch die Anzahl an nicht eindeutigen Möglichkeiten, welche die Rekonstruktion erschweren. Dennoch liefern diese nicht eindeutigen Möglichkeiten relevante Zusatzinformation bei Mehrfachübertragungen.

Ein weiterer interessanter Nebeneffekt bei diesen Impulssequenzen ist, dass bei der Umwandlung von Impulssequenz zu Bitsequenz die Polaritäten der einzelnen Impulse nur einen Einfluss auf die  $k$  niederwertigsten Bits haben. Demzufolge wird die Zuverlässigkeit für alle Bits  $\geq k$  nicht durch eine falsche Phasendetektion beeinflusst.

### 3.4.3 Fehlerrate Slave → Master

Sei die Impulsfehlerrate  $i_f = 10^{-x}$  dann ist die Symbolfehlerrate für Länge  $n$  gleich  $SER = (1 - 10^x)^n$  und die Frame-Error-Rate für eine Nachrichtenlänge  $L=408$  und Bitrate  $B$  ist dann

$$FER = 1 - ((1 - 10^x)^n)^{L/B}$$

Da Impulse rechts vom linken Impuls keine Fehler verursachen, ist die tatsächliche Symbolfehlerrate geringer.

Sei  $\#_{seq}$  die Anzahl an gültigen Impulssequenzen mit  $i_k = \dots = i_1 = 0$  dann ist die Symbol-Error-Rate (SER)

$$SER' = \prod_{i=1}^n (1 - 10^x \cdot \frac{\#_{seq}}{2^B})$$

Daraus ergibt sich eine Frame-Error-Rate von:

$$FER' = 1 - (\prod_{i=1}^n (1 - 10^x \cdot \frac{\#_{seq}}{2^B}))^{L/B}$$

Auf Abbildung 3.36 kann folgender Einfluss der verschiedenen Parameter auf die SER beobachtet werden:

- wenn die Anzahl an Impulsen  $k \nearrow$ , dann  $FER \searrow$
- wenn die Bitrate  $B \nearrow$ , dann  $FER \nearrow$
- wenn die Bitrate  $B \nearrow$ , dann  $(FER' - FER)$
- wenn die Impulsfehlerrate  $i_f \nearrow$ , dann  $FER \nearrow$



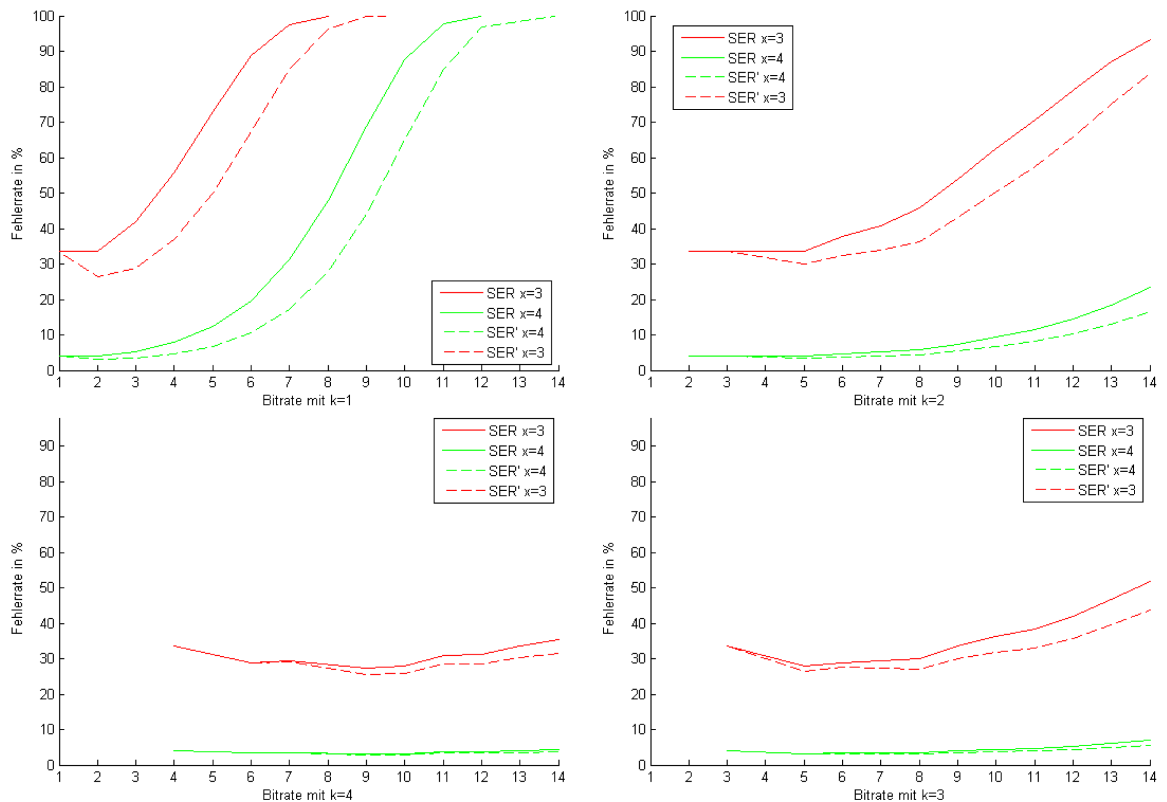


Abb. 3.36: Symbolfehlerrate für verschiedene Impulsfehlerraten und Anzahl an Impulsen

**Beispiel:** Master mit 2 Sensorknoten und 3 Impulsen. In diesem Fall überträgt der Master pro Symbol 18 Bit, somit müssen beide Slaves ihrerseits pro Symbol 9 Bit übertragen. Dafür gibt es 2 Möglichkeiten:

Fall 1:  $(k = 2, n = 17) \Rightarrow FER' = 43\%$  für  $x = 3$  /  $FER' = 5\%$  für  $x = 4$

Fall 2:  $(k = 3, n = 9) \Rightarrow FER' = 30\%$  für  $x = 3$  /  $FER' = 3\%$  für  $x = 4$

Im ersten Fall müssen für  $x=4$ , 7 Frames und für  $x=3$ , müssen 100 Frames wiederholt werden. Im zweiten Fall müssen für  $x=4$ , 4 Frames und für  $x=3$ , müssen 56 Frames wiederholt werden.

Bei diesen Berechnungen sind noch keine Fehlerkorrekturmaßnahmen berücksichtigt worden. Bestenfalls könnte man die Framewiederholungen im ersten Fall auf 55, für  $x=3$  und im zweiten Fall auf 39, für  $x=3$  senken.

### 3.4.4 Fehlerrate Master $\rightarrow$ Slave

Im Unterschied zur Übertragung Slave  $\rightarrow$  Master, kann nicht von einer wiederholbaren Impulssequenz Master  $\rightarrow$  Slave ausgegangen werden. Solch ein Fall würde nur dann auftreten, wenn die Übertragung für jeden involvierten Slave fehlerhaft ist. Demzufolge ist die Impulssequenz bei jeder Wiederholung unterschiedlich. Tabelle

3.17 stellt die Bitfehlerrate des Masters bei  $n = 16$  und  $k = 3$  dar. IE steht für 'Impulsecho', IA für 'Impulsauslöschung' und K für 'Kombination'.

**Tabelle 3.17:** Fehlerrate Master→ Slave abhängig von der Impulsfehlerrate für einzelne Datensymbole

$x$	Fehler	11	10	9	8	7	6	5	4	3	2	1	0
3	IE	0,209	0,305	0,37	0,35	0,39	0,45	0,49	0,52	0,49	0,28	0,37	0,539
	IA	0,047	0,071	0,08	0,09	0,11	0,12	0,14	0,14	0,14	0,05	0,09	0,127
	K	0,128	0,188	0,21	0,25	0,26	0,32	0,33	0,35	0,33	0,15	0,26	0,38
4	IE	0,027	0,026	0,04	0,04	0,04	0,05	0,05	0,06	0,05	0,03	0,05	0,06
	IA	0,005	0,003	0,01	0,01	0,01	0,02	0,01	0,01	0,01	0,01	0,01	0,009
	K	0,016	0,021	0,02	0,03	0,03	0,03	0,04	0,03	0,03	0,02	0,02	0,034

Wie zu erwarten, sinkt die Fehlerwahrscheinlichkeit tendenziell je höherwertiger das Bit ist. Außerdem fällt auf, dass der Worst Case immer bei dem Fehlertyp Impulsauslöschung auftritt. Daher wird in den folgenden Berechnungen auch nur dieser Fehlertyp berücksichtigt.

Bei 12 Bit können mehrere Fälle bezüglich der Übertragung Master Slave auftreten.

Es können 2,3,4 Slaves gleichzeitig angesprochen werden. Die Fehlerwahrscheinlichkeiten für jeden Slave sind in Tabelle 3.18 angegeben.

Die Symbolfehlerrate für einen Sensorknoten berechnet sich wie folgt:

Sei  $B_S$  die Bitrate vom Slave. Für jeden Sensorknoten  $S_k$  existiert eine Bitfolge  $b_1, b_2, \dots, b_{B_S}$ . Die Symbolfehlerrate von  $S_k$  ist:

$$SER = \prod_{i=1}^l (1 - b_i)$$

Die FER für  $S_k$  ist:

$$FER = \left( \prod_{i=1}^l (1 - b_i) \right)^{L/B_S}$$

Tabelle 3.18 zeigt wie die Symbolfehlerrate für jeden Sensorknoten  $S_k$  von der Anzahl an Sensorknoten und der Impulsfehlerrate abhängt. Außerdem wird die durchschnittliche FER und die durchschnittliche Anzahl an Wiederholungen  $M \rightarrow S_k$  dargestellt.

Die Wiederholungen für 127 Slaves werden mittels FER wie folgt rekursiv berechnet:

Sei  $W_0 = 127 \cdot FER$  und  $W_{k+1} = W_k \cdot FER$

Sei  $l$  der kleinste Index wo gilt  $W_l < 1$  dann ist die Anzahl an Wiederholungen

$$\left\lceil \sum_{k=1}^{l-1} W_k \right\rceil$$

**Tabelle 3.18:** Fehlerrate Master→ Slave abhängig von der Impulsfehlerrate für einzelne Datensymbole

x	SER				$\overline{SER}$	$\overline{FER}$	W
	$S_1$	$S_2$	$S_3$	$S_4$			
3	0,046619	—	—	—	0,046619	0,80273	513
	0,020504	0,026661	—	—	0,023583	0,80266	513
	0,012284	0,018254	0,016815	—	0,015784	0,80266	513
	0,008785	0,011823	0,014915	0,011924	0,011862	0,80267	513
4	0,005178	—	—	—	0,005178	0,1618	24
	0,002248	0,002936	—	—	0,002592	0,1618	24
	0,001349	0,001999	0,001839	—	0,001729	0,1618	24
	0,000899	0,001349	0,001599	0,001339	0,001297	0,1618	24

Die Tabelle 3.18 stellt die verschiedenen SER je Sensorknoten dar. Außerdem wurden auch die durchschnittliche SER sowie FER und die Anzahl an notwendigen Wiederholungen angegeben.

Wie man sehen kann sinkt mit steigender Anzahl an eingesetzten Sensorknoten die einzelne SER sowie auch die durchschnittliche  $\overline{SER}$ . Tendenziell folgt aus  $i < j$ , dass  $SER_{S_i} > SER_{S_j}$ . Die  $\overline{SER}$  ist invers proportional zur Anzahl der Sensorknoten. Da die  $\overline{SER}$  um den gleichen Faktor abnimmt, wie auch die Framelänge zunimmt, bleibt die  $\overline{FER}$  unverändert.

### 3.4.5 Optimale Konfiguration

Um die optimale Konfiguration zu finden, müssen zuerst alle Möglichkeiten Konfigurationen für eine Übertragung *Master* → *Slaves* ermittelt werden. Sofern es Konfigurationen gibt, welche alle Daten innerhalb 1 ms übertragen können, muss überprüft werden, bis zu welcher maximalen Impulsfehlerrate dies noch bewerkstelligt werden kann. Die Konfiguration, welche alle Anforderungen mit maximaler Fehlerrate erfüllt, ist die optimale. Tabelle 3.19 listet alle bestmöglichen Konfigurationen auf, welche eine Zykluszeit von unterhalb 1 ermöglichen.

Für die Berechnung der Übertragungszeit wurde eine Nachrichtenlänge von 408 Bits angenommen. Je nach Länge der PREAMBLE-Symbole treten andere Fehleraten auf. Es hat sich herausgestellt, dass in den Fällen ( $n = 11, n = 13, n = 15, n = 17$ ) immer die gleiche Konfiguration optimal ist. Und zwar sind das 16 Sensorknoten mit einer Master-Bitrate von 96 und einer Slave-Bitrate von 6.

**Tabelle 3.19:** Vergleich des Impulsverhältnisses während der Synchronisation für verschiedene Sequenzlängen der PREAMBLE

PREAMBLE	Sensorknoten	Master			Slaves			$\overline{FER}$
		$n'_M$	$k_M$	BR	$n'_s$	$k'_S$	BR	
$n = 17$	16	357	14	96	5	3	6	3,15%
$n = 15$	16	357	14	96	5	3	6	13,83%
$n = 13$	16	357	14	96	5	3	6	24,11%
$n = 11$	16	357	14	96	5	3	6	31,04%

Es konnte bewiesen werden, dass eine bidirektionale Übertragung von 48 Byte innerhalb einer Millisekunde für 127 Knoten gewährleistet werden kann, wenn die

$\overline{FER}$  nicht 24,11 % übersteigt. Dieser Wert ist natürlich vom Umfeld und Aufbau des WSN abhängig.

Um eine möglichst optimale Übertragung zu gewährleisten, sollten folgende Punkte eingehalten werden.

1. Slaves mit gleicher Gain-Einstellung für den Master sollten bevorzugt gruppiert werden, da dies eine schnellere Synchronisation begünstigt.
2. Je zuverlässiger die Verbindung zwischen Master und einem Sensorknoten ist, desto höher sollte der Wert für  $k$  bei  $S_k$  gewählt werden.
3. Slaves mit gleicher Nachrichtenlänge sollten gruppiert werden, um eine optimal Auslastung zu gewährleisten

### 3.5 Zusammenfassung

In dieser Arbeit wurde hauptsächlich nur der Worst Case betrachtet, da dieser Fall aber äußerst selten auftritt ist mit einer sehr viel höheren Zuverlässigkeit im realen Einsatz zu rechnen. Normalerweise werden nur sehr kurze Nachrichten übertragen und Netzwerke mit 127 Sensorknoten sind eher die Ausnahme als die Regel, siehe Abbildung 1.2.1.2.

In diesem Kapitel wurden viele verschiedene Verbesserungsmöglichkeiten mit unterschiedlichen Herangehensweisen untersucht. In der Tabelle 3.20 sind die wichtigsten Ergebnisse zusammengefasst dargestellt. Nur die Optimierungen mittels Vollduplex und Multicast erfüllen die Anforderungen der Automatisierungsindustrie bzgl. der Datenübertragungsrate. Nichts desto trotz sind die anderen Verbesserungen auch sinnvoll und könnten dazu beitragen den aktuellen UWB-Standard zu verbessern.

Tabelle 3.20: Zusammenfassung der erzielten Ergebnisse

	Punkt zu Punkt Ziel < 50 $\mu$ s	Punkt zu Multipunkt Ziel < 1 ms	Kurzbeschreibung der Verbesserungsmaßnahmen
IEEE 802.15.4a	80,4 $\mu$ s	20,43 ms	—
Optimierte Frame-Länge	32,2 $\mu$ s	8,82 ms	<ul style="list-style-type: none"> <li>• reduzierte Preamble-Sequenz-Längen</li> <li>• reduzierter FEC-Overhead</li> <li>• erhöhte Datenrate des PHR</li> </ul>
Broadcast	—	6,2 ms	<ul style="list-style-type: none"> <li>• reduzierte Anzahl an Synchronisationen</li> <li>• modifizierter PHR</li> </ul>
Vollduplex	—	4,8 ms - 1,7 ms	<ul style="list-style-type: none"> <li>• modifiziertes Synchronisationsverfahren</li> <li>• modifiziertes Datenübertragungsverfahren</li> <li>• Kollisionsvermeidung</li> </ul>
Vollduplex mit Multicast	—	bis zu 0,6 ms	<ul style="list-style-type: none"> <li>• Erweiterung des Synchronisationsverfahrens</li> <li>• Erweiterung des Datenübertragungsverfahrens</li> </ul>
Vollduplex mit Multicast und Fehlerkorrektur	—	< 1 ms abhängig von der Fehlerrate	<ul style="list-style-type: none"> <li>• Fehlerkorrekturverfahren mit geringem Overhead</li> <li>• modifiziertes Datenübertragungsverfahren</li> <li>• intelligente Gruppierung von Sensorknoten</li> </ul>

Im nächsten Kapitel wird unter anderem gezeigt wie einige der hier vorgestellten Optimierungen implementiert wurden.



# Umsetzung der Optimierungen in Hardware

---

## Inhaltsverzeichnis

4.1	Vorgehensweise für die Implementierung . . . . .	<b>113</b>
4.2	Modularer Aufbau . . . . .	<b>115</b>
4.3	Verfahren und Methoden . . . . .	<b>121</b>
4.4	ASIC Implementierung . . . . .	<b>137</b>
4.5	FPGA Implementierung . . . . .	<b>154</b>

---

Im Rahmen dieser Arbeit entstand ein UWB-Transceiver als Einzelchip. In diesem Kapitel werden die wichtigsten Implementierungen, die im Rahmen dieser Arbeit durchgeführt wurden, vorgestellt. Es werden auch mögliche zukünftige Implementierungsansätze vorgestellt, welche für das Ziel dieser Arbeit relevant sind.

## 4.1 Vorgehensweise für die Implementierung

Vor der Implementierung des UWB-Transceivers in Hardware, wurden sowohl Sender als auch Empfänger in Matlab simuliert. Zuerst wurde der Sender und dann der Empfänger entwickelt. Nachdem der Empfänger in der Lage war, die Daten korrekt zu empfangen und zu dekodieren, wurde die Simulation komplexer gestaltet, indem verschiedene UWB Kanalmodelle [78] und verschiedene Signalrauschverhältnisse (*engl. Signal to Noise Ratio, SNR*) hinzugefügt wurden.

Die Hardwareimplementierung wurde anfangs auf einem FPGA<sup>1</sup> realisiert und dann für eine ASIC<sup>2</sup>-Implementierung für die IHP-Technologie 0.25  $\mu\text{m}$  BICMOS angepasst. Es hat sich nach den ersten Portierungen herausgestellt, dass eine Portierung *FPGA*  $\rightarrow$  *ASIC* schwieriger ist als eine Portierung *ASIC*  $\rightarrow$  *FPGA*. Dies liegt daran, dass im Synopsys-Compiler<sup>3</sup> nicht alle für das FPGA verwendete Funktionen und Methoden unterstützt werden. Als ein Beispiel kann variable Indexierung genannt werden. Dieser Fall tritt auf, wenn der Index eines Arrays eine Variable ist. Dies ist beim FPGA zulässig aber nicht beim Synopsys-Compiler. Um den Aufwand der Portierungen zu reduzieren und um möglichst den gleichen

---

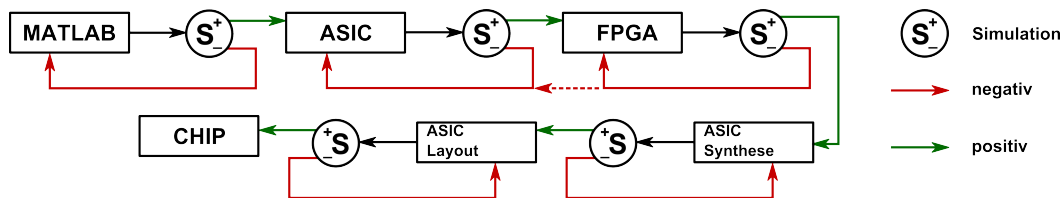
<sup>1</sup>Es wurde ein Virtex 5 von der Firma Avnet verwendet

<sup>2</sup>Anwendungsspezifische Integrierte Schaltung

<sup>3</sup>Der Synopsys-Compiler übersetzt vhdl-Code in synthetisierbaren Verilog-Code.

VHDL-Code für FPGA und ASIC nutzen zu können, wurde zuerst eine ASIC Implementierung entworfen. Dies ist besonders für Testmessungen wichtig, um ein identisches Verhalten vom FPGA und ASIC garantieren zu können.

In Abbildung 4.1 wurden alle Arbeitsschritte und deren Reihenfolge dargestellt. Nach jedem Arbeitsschritt wurde eine Simulation durchgeführt und bewertet. Bei einem positiven Simulationsergebnis wurde der nächste Arbeitsschritt bearbeitet. Bei einem negativen Simulationsergebnis wurde der letzte Arbeitsschritt solange überarbeitet, bis das Simulationsergebnis positiv war. Erst wenn alle Schritte erfolgreich absolviert wurden, wurde der Chip gefertigt. Auch wenn alle Schritte erfolgreich simuliert wurden, ist dies noch keine Garantie dafür, dass der fertige Chip wie erwartet funktioniert. Es kann vorkommen, dass bei den Simulation nicht alle Fehler entdeckt werden.



**Abb. 4.1:** Ablaufplan der Implementierung  
Positive Ergebnisse der Funktionstests sind grün markiert.  
Negative Ergebnisse der Funktionstests sind rot markiert.

Die Funktionalität der Blöcke in Abbildung 4.1 ist wie folgt:

- MATLAB: Hier werden Algorithmen und Kodierungen erarbeitet und simuliert.
- ASIC: die Algorithmen werden in VHDL umgesetzt und an die Hardware angepasst unter Berücksichtigung von Latenz, Energieverbrauch, Taktrate und Fläche.
- FPGA: der ASIC-Code wird auf dem FPGA getestet und die Resultate werden mit dem des ASICs verglichen und wenn nötig korrigiert.
- ASIC Synthese: Das Design wird mittels Synopsys-Compiler synthetisiert. Bei der Synthese können Fehler auftreten, die vorher nicht sichtbar waren und das Design muss dann wiederum korrigiert werden.
- ASIC Layout: hier muss noch einmal überprüft werden, ob sich das Design beim Best-Case, Typical-Case und Worst-Case korrekt verhält. Falls das Design aus mehreren unabhängigen Blöcken besteht, so können auch bei der Verdrahtung dieser Blöcke Fehler auftauchen.
- CHIP: Nachdem der Chip gefertigt wurde, wird dieser getestet und die so gewonnenen Erkenntnisse fließen in das Matlab-Modell zurück und der Design-Flow startet von neuem.



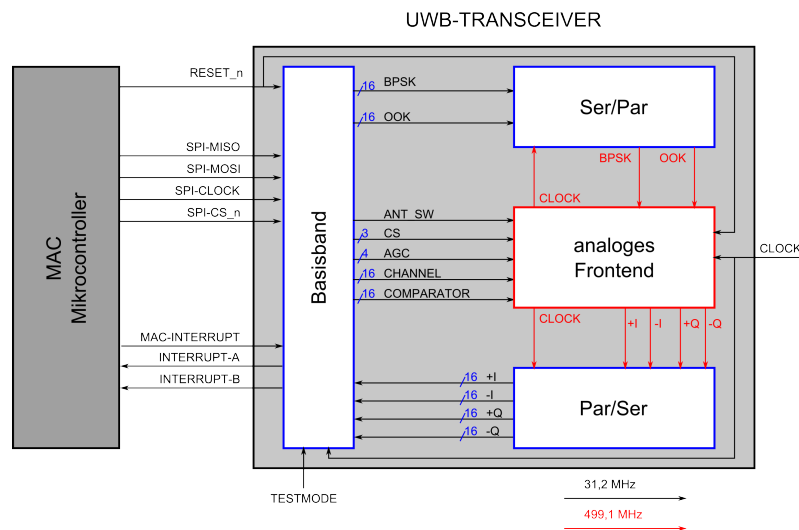
## 4.2 Modularer Aufbau

Zu Beginn der Implementierungsarbeiten wurde beschlossen, den Basisband-Entwurf möglichst modular zu gestalten. Dies hat mehrere Gründe:

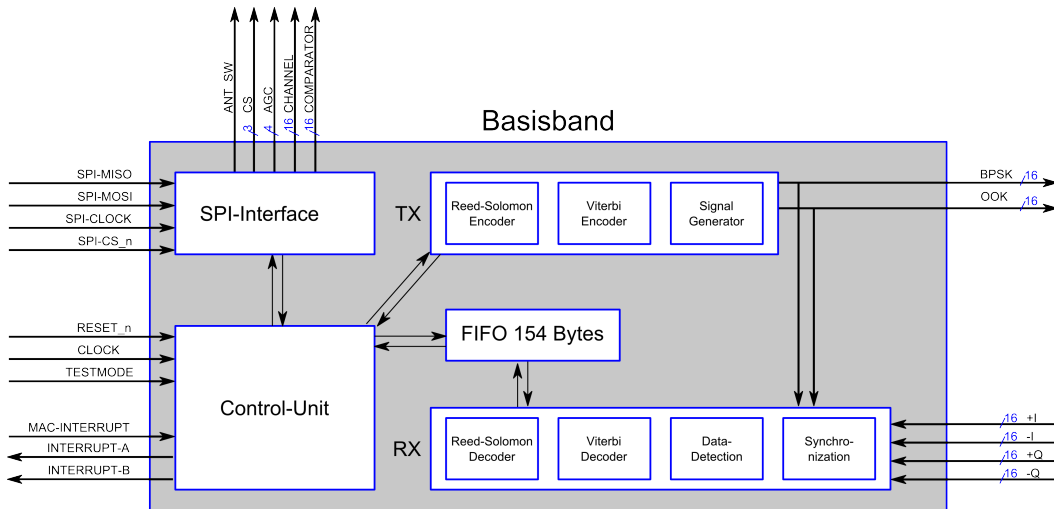
- eine bessere Übersicht
- einfachere Wartung
- erleichterter Einbau von Energiesparmechanismen wie *clock-gating*.
- Möglichkeit jeden Block einzeln zu synthetisieren und zu optimieren
- einfacher Austausch der Blöcke

Abbildung 4.2 zeigt ein Blockschaltbild des kompletten Chips und dessen Schnittstellen. Die wichtigsten Blöcke neben dem Basisband sind:

- **SER/PAR:** Serialisiert den parallelen Dateneingang von 2 x 16 Bit bei 31,2 MHz auf 2 x 1 Bit bei 499,2 MHz.
- **PAR/SER:** Parallelisiert den seriellen Dateneingang von 4 x 1 Bit bei 499,2 MHz auf 4 x 16 Bit bei 31,2 MHz.
- **Analoges Frontend:** Ist zuständig für das Aussenden bzw. Empfangen der analogen UWB-Signale entsprechend des Standards IEEE 802.15.4a.



**Abb. 4.2:** Blockschaltbild des UWB-Transceivers. In blau sind die digitalen und in rot sind die analogen Blöcke dargestellt.



**Abb. 4.3:** Darstellung des UWB-Basisbands aus Abbildung 4.2 als Blockschaltbild mit den wichtigsten Komponenten.

Abbildung 4.3 zeigt die Hauptbestandteile des Basisbands, diese sind:

- **SPI-Interface:** Schnittstelle zwischen Basisband und Mikrocontroller, welche auch genutzt wird, um das analoge Frontend zu konfigurieren.
- **Reed-Solomon Encoder:** Dies ist ein komplexer Block, welcher redundante Daten zur Fehlerkorrektur erzeugt. Der VHDL-Code dieses Blockes wurde mittels Matlab generiert. Dieser Block ist optional und kann abgeschaltet werden.
- **Viterbi Encoder:** Es handelt sich um ein Faltungskodierer der beim Senden immer verwendet wird.
- **Signal Generator:** Damit werden die Daten als Bitstream mit passender Datenrate und Modulation erzeugt.
- **Synchronization:** Dieser Block korreliert das Empfangssignal und detektiert die PREAMBLE-Symbole im PREAMBLE- und SFD-Teil unter Berücksichtigung der Phasendrehung des IQ-Signals.
- **Data-Detection:** Hier wird das Empfangssignal in einem Bitstream umgewandelt.
- **Viterbi Decoder:** Dieser Block ermöglicht eine Fehlerkorrektur des Datenstroms und ist abschaltbar.
- **Reed-Solomon Decoder:** Dieser Block ermöglicht eine Fehlerkorrektur des Datenstroms und ist abschaltbar. Der VHDL-Code dieses Blockes wurde mittels Matlab generiert.

- **Control-Unit:** Dieser Block aktiviert/deaktiviert die einzelnen Blöcke und steuert die Interrupt-Signale.

Im Folgenden werden die einzelnen digitalen Komponenten des UWB-Transceivers – also Basisband, Ser/Par und Par/Ser – detailliert beschrieben. Es werden Gründe für die Verwendung und das Verhalten der einzelnen Komponenten genannt.

#### 4.2.1 SPI-Interface

Aufgrund des genutzten Chip-Gehäuses, welches nur 32 Pins zur Verfügung stellt, wurde ein Interface mit einer geringen Anzahl an Pins benötigt. Aus diesem Grund wurde die Datenübertragungsschnittstelle SPI (engl. *Serial Peripheral Interface*) mit 4 Pins ausgewählt. Außerdem wurde SPI aufgrund dessen Einfachheit und Kompatibilität zu anderen Mikrocontrollern gewählt. Es wurde sich gegen I2C entschieden weil dafür ein zusätzlicher Pull-up Widerstand benötigt würde. Des Weiteren können diese Pins auch für andere Zwecke verwendet werden (siehe Abschnitt 4.3.4 Seite 123). Zusätzlich zum SPI gibt es drei weitere Interrupts '*interrupt-a*', '*interrupt-b*', und '*mac-interrupt*'. Diese Interrupts werden zur Steuerung des FIFOs (engl. *First In First Out*) und diverser Testmodi eingesetzt.

#### 4.2.2 FIFO

Das Basisband beinhaltet einen FIFO (engl. *First In First Out*), welcher es ermöglicht, 154 Bytes zu speichern. Diese Zahl wurde so gewählt, dass jede komplette standardkonforme Nachricht zwischengespeichert werden kann. Es werden zwei Byte für die Länge benötigt, maximal 127 Byte für die Nutzdaten und schließlich maximal  $4 \cdot 6 = 24$  Byte für den Reed-Solomon-Overhead. Des Weiteren können mit Hilfe der Interrupts wesentlich größere Datenmengen übertragen werden. Der FIFO wird sowohl für Sende- als auch für die Empfangsdaten verwendet. Zu Testzwecken ist auch eine parallele Nutzung möglich.

Um größere Datenpakete zu schicken, ist es notwendig den PHR-Teil zu erweitern. Dies geschieht, indem das reservierte Bit in der Standard-PHR "extended header" auf 1 gesetzt wird, darauf folgt dann die Länge 13 Bit und zusätzlich 6 Bit für den Hamming-Code, welcher die Integrität der Länge sicherstellen soll. Es wird der gleiche Hamming-Code benutzt, welcher auch für die normale PHR verwendet wird. Nachdem die erweiterte PHR übertragen wurde, folgen die Daten inklusive des Reed-Solomon-Code Overheads.

Da bis zu  $2^{13} = 8192$  Byte übertragen werden können und diese nicht in den FIFO passen, müssen die Daten *on-the-fly* kodiert und übertragen werden. Dies geht nur, wenn der MAC bzw. Mikrocontroller die Daten nach und nach in den FIFO schreibt. Dazu muss der Mikrocontroller feststellen können, wann der FIFO voll bzw. leer ist. Dies geschieht mittels zweier Interrupts '*interrupt-a*' und '*interrupt-b*' vom Basisband zum Mikrocontroller.

**Interruptverhalten im Sendemodus:**

- *interrupt – a* signalisiert, dass der FIFO fast leer ist, was bedeutet, dass der MAC neue Daten in den FIFO übertragen muss. Falls dies nicht rechtzeitig geschieht, wird der Sendestrom unterbrochen und führt zu einer fehlerhaften Übertragung. Die untere Schwelle für diesen Interrupt kann über SPI passend zum genutzten Mikrocontroller eingestellt werden.
- *interrupt – b* signalisiert, dass die Übertragung abgeschlossen ist und der UWB-Transceiver bereit für weitere Aufgaben ist oder abgeschaltet werden kann.

Beim Empfang überlanger Nachrichten, werden die gleichen Interrupts verwendet. Im Gegensatz zum Senden, muss der MAC bzw. Mikrocontroller anders auf die Interrupt-Signale reagieren. In diesem Fall muss dieser vom FIFO Daten auslesen und nicht hineinschreiben.

#### Interruptverhalten im Empfängermodus:

- *interrupt – a* signalisiert, dass der FIFO fast leer ist und dies bedeutet, dass der MAC keine neuen Daten vom FIFO auslesen kann. Falls der MAC dennoch versucht Daten auszulesen, obwohl der FIFO leer ist, so werden falsche Daten ausgelesen und es kann passieren, dass neue Daten seitens des Empfängers nicht korrekt im FIFO gespeichert werden können. Beides führt zu einer fehlerhaft empfangenen Nachricht. Die untere Schwelle für diesen Interrupt kann über SPI passend zum genutzten Mikrocontroller eingestellt werden.
- *interrupt – b* signalisiert, dass der Empfang abgeschlossen ist. Der UWB-Transceiver ist entweder für weitere Aufgaben bereit oder kann abgeschaltet werden.

Die Schwellen für die Interrupts müssen so gewählt werden, dass der Mikrocontroller genug Zeit hat darauf zu reagieren.

#### 4.2.3 Ser/Par

Laut dem Standard IEEE 802.15.4a, muss der PHY bzw. das Basisband für das analoge Frontend zwei digitale Signale mit einer Datenrate von 499,2 MHz zur Verfügung stellen. Diese zwei Signale sind **OOK** (*engl. On Off Keying*) und **BPSK** (*engl. Binary Phase Shift*) und weisen einen Informationsgehalt von 1.5 Bit auf. Die Wahrheitstabelle 4.1 zeigt, welche Impulse erzeugt werden können.

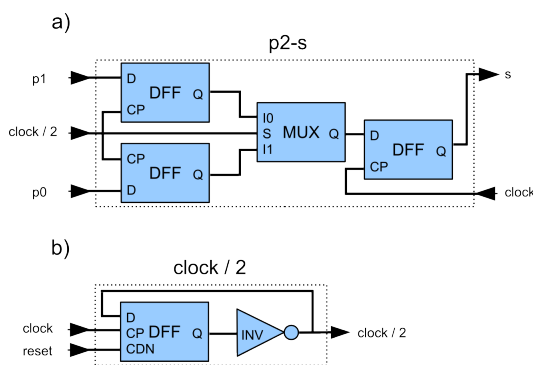
**Tabelle 4.1:** Wahrheitstabelle für OOK und BPSK

OOK	BPSK	Resultat
0	0	. kein Impuls / leerer Impuls
1	0	+ positiver Impuls
1	1	- negativer Impuls
0	1	<b>Dieser Fall tritt nicht auf</b>

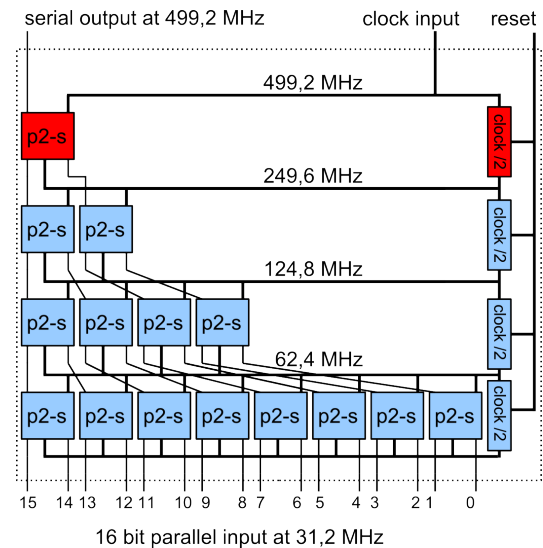
Eine wichtige Tatsache ist, dass digitale Schaltungen bei hohen Frequenzen von mehreren hundert Megahertz in der  $0.25 \mu$  IHP BICMOS Technologie nicht trivial sind und zu vielen Problemen führen können.

Da der Referenztakt für die PLL (engl. *phase-locked loop*) im analogen Frontend 31,2 MHz beträgt und weil dies exakt  $\frac{499,2}{16}$  MHz entspricht, kann das Basisband auch mit der niedrigeren Frequenz betrieben werden, um Energie zu sparen und mittels eines

16-zu-1 Serialisierers sowohl OOK als auch BPSK zu erzeugen. Der Vorteil ist, dass nur der Serialisierer bei 499,2 MHz betrieben werden muss, was den besonders kritischen Teil auf ein Minimum reduziert. Der Serialisierer wurde nicht automatisch generiert. Sowohl die Auswahl der Zellen, die Verdrahtungen als auch das Layout wurden manuell durchgeführt (siehe auch [4]). Abbildung 4.4 zeigt die Kernkomponenten aus denen der Serialisierer besteht. Abbildung 4.5 zeigt den Aufbau des Serialisierers aus den Kernkomponenten.



**Abb. 4.4:** Kernkomponenten des Serialisierers  
p2-s steht für (engl. *parallel to serial*) hier werden jeweils immer zwei einzelne Bits serialisiert.



**Abb. 4.5:** Aufbau des Serialisierers.  
Komponenten mit Bestandteilen, welche Signale mit 499,2 MHz Signalen verarbeiten, sind rot markiert.

Aufgrund des stufenweisen Aufbaus des Serialisierers wird nur eine geringe Anzahl an Gattern mit der Taktfrequenz von 499,2 MHz betrieben. Die betroffenen Komponenten sind in Abbildung 4.5 rot markiert. Insgesamt werden drei Gatter mit 499,2 MHz betrieben:  
1 Flipflop in  $p2 - s^4$ , 1 Flipflop und 1 Inverter im *clock-divider*<sup>5</sup>.

Nach der Synthese wurde der Serialisierer bis 680 MHz erfolgreich digital simuliert. Nach dem Layout wurde dieser Block nochmal bis zu 550 MHz erfolgreich analog

<sup>4</sup>p2 - s steht für (engl.) *parallel to serial*, d.h. von zwei Bit parallel zu einem Bit seriell.

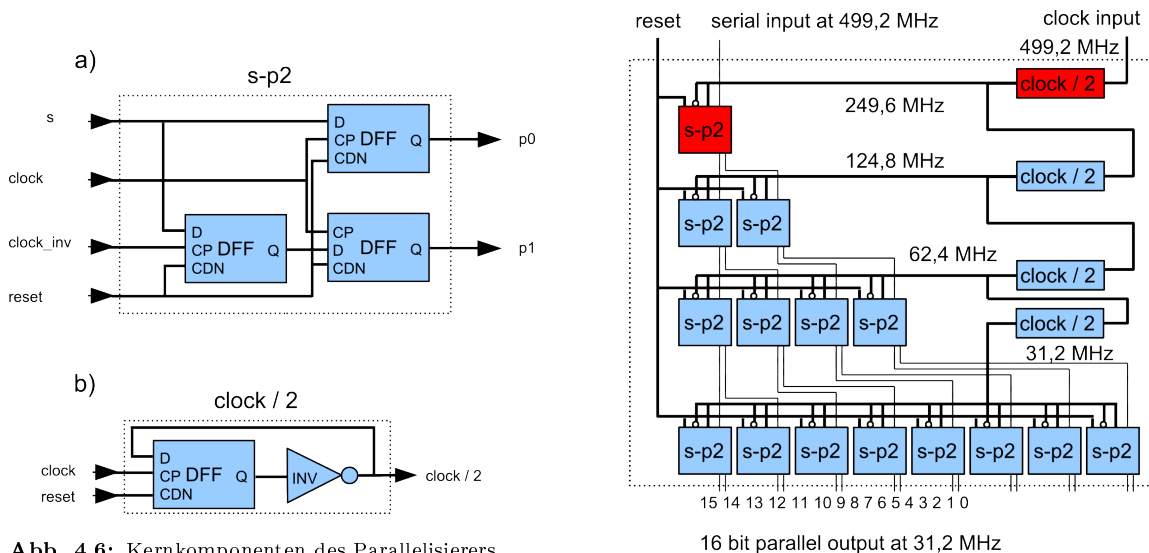
<sup>5</sup>Ein *clock-divider* ist ein Taktfrequenzteiler, welcher in diesem Fall die Taktfrequenz halbiert.

simuliert. Der Unterschied kommt daher, dass die analoge Simulation genauer ist, weil noch zusätzliche parasitäre Störungen berücksichtigt werden. Nach der Fertigung des Chips haben Messungen gezeigt, dass dieser Block zuverlässig funktioniert.

#### 4.2.4 Par/Ser

Ähnlich zum Serialisierer beim Sender, wird im Empfänger ein Parallelisierer eingesetzt, um das IQ-Empfangssignal bei einer Taktfrequenz von 499,2 MHz in einen Datenstrom mit 31,2 MHz umzuwandeln. Zu diesem Zweck wurde, wie beim Serialisierer, alles manuell entwickelt und ein stufenweiser Ansatz verfolgt, um möglichst wenige Gatter mit 499,2 MHz betreiben zu müssen. Hier werden nur im *clock-divider* zwei Gatter mit 499,2 MHz verwendet. Weitere zwei Gatter werden im Block  $s-p2^6$  mit 249,6 MHz betrieben. Dennoch ist das Eingangssignal  $s$  mit der Taktfrequenz von 499,2 MHz gesteuert.

Abbildung 4.6 zeigt die Kernkomponenten, aus denen der Parallelisierer besteht. Die betroffenen Komponenten sind in Abbildung 4.7 rot dargestellt.



**Abb. 4.6:** Kernkomponenten des Parallelisierers.  $s-p2$  steht für (*engl. serial to parallel*) von einem Bit seriell zu zwei Bit parallel.

**Abb. 4.7:** Aufbau des Parallelisierers. Komponenten mit Bestandteilen, welche Signale mit 499,2 MHz Signalen verarbeiten, sind rot markiert.

Nach der Synthese wurde der Parallelisierer erfolgreich bis 680 MHz digital simuliert. Nach dem Layout wurde dieser Block nochmal erfolgreich bis zu 550 MHz analog simuliert. Nach der Fertigung des Chips haben Messungen gezeigt, dass dieser Block zuverlässig funktioniert.

<sup>6</sup> $s-p2$  steht für (*engl. serial to parallel*), d.h. von einem Bit seriell zu zwei Bit parallel.

### 4.3 Verfahren und Methoden

In diesem Abschnitt werden einige implementierte Methoden bzw. Verfahren näher erläutert.

#### 4.3.1 Getriggerte Datenübertragung

Um eine höhere Genauigkeit beim Übertragungszeitpunkt zu erreichen, wurde ein weiterer Interrupt '*mac - interrupt*' implementiert, vom MAC zum Basisband, welcher es ermöglicht, die Datenübertragung ohne Verzögerung zu starten. Über SPI würde der Startbefehl 8 SPI-Taktzyklen (1 bis 33 MHz abhängig vom genutzten SPI-Master im Mikrocontroller) und zwei Taktzyklen (Referenztakt des UWB-Transceivers 31,2 MHz) benötigen, bevor dieser eine Aktion auslöst.

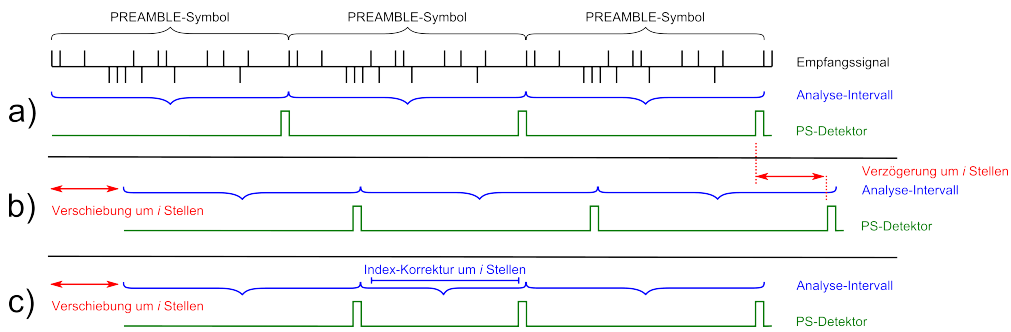
Der '*mac - interrupt*' ermöglicht es, unabhängig vom verwendeten SPI-Master, innerhalb eines Taktzyklusses einen schnellen und gleichbleibenden Übertragungszeitpunkt zu gewährleisten. Dies ist besonders für Echtzeit-Systeme wichtig: es ermöglicht auch dem MAC kürzere Guard-Intervalle für Datenübertragungen mit mehreren Knoten in einem Sensornetzwerk zu nutzen.

#### 4.3.2 Synchronisation

Die Synchronisation war eine der schwierigsten Herausforderungen. Die Korrelation an sich ist nicht kompliziert, aber die Implementierung erforderte mehr Aufwand, da andere Parameter eine wichtige Rolle spielen. Der wichtigste Punkt ist die benötigte Detektierungszeit. Dies ist besonders wichtig, da bei jeder Taktflanke 64 Bit ausgewertet werden müssen. Wenn die Detektierung im Worst-Case  $X$ -Taktzyklen benötigt, so müssen  $X \cdot 64$  Bit zwischengespeichert werden. Wird z.B. die Korrelation an allen 31 möglichen Positionen der PREAMBLE-Symbole durchgeführt, um danach das Maximum der Korrelation zu ermitteln und um festzustellen, ob und wann ein PREAMBLE-Symbol empfangen wurde, so können im Worst-Case 31 Taktzyklen benötigt werden. Dies führt dazu, dass  $31 \cdot 64 = 1984$  Bit = 248 Byte zwischengespeichert werden müssen. Dies ist größer als der verwendete FIFO. Da dieser Zwischenspeicher wie ein Schieberegister mit den aktuellen Werten gefüllt werden muss, werden bei jedem Takt 1984 Flip-Flops geschaltet. Was wiederum nicht nur die Fläche, sondern auch den Energieverbrauch erhöht. Des Weiteren müssen bei jedem Takt diverse Operationen ausgeführt werden, welche auch Einflüsse auf das Timing haben und die Synthese negativ beeinträchtigen. Daher war es wichtig, den Zwischenspeicher so klein wie möglich zu gestalten und die Anzahl an Operationen zu minimieren.

Für die Synchronisation wird der Datenstrom mehrfach aufgeteilt. Der Parallelisierer liefert jeweils für  $+I, -I, +Q$  und  $-Q$ , 16 Datenströme. Da bei einem dem UWB-Standard entsprechendem PREAMBLE-Symbol zwischen den Impulsen immer 15 *time-slots* je 2 ns liegen, befinden sich alle Impulse in einem der 16 Datenströme. Ziel ist es, den richtigen Datenstrom zu finden und die exakte relative Verschiebung zum PREAMBLE-Symbol zu ermitteln. Um dies zu bewerk-

stelligen, werden alle 16 Datenströme mittels Korrelation analysiert. Nachdem mehrfach PREAMBLE-Symbole detektiert wurden, wird die relative Verschiebung um  $i$  Stellen, dazu ausgenutzt das Zeitfenster der Analyse (Analyse-Intervall) an die PREAMBLE-Symbole anzupassen, um die Latenz zu reduzieren. Abbildung 4.8 c) zeigt wie eine solche Anpassung geschieht. Um den Best-Case, der in Abbildung 4.8 a) dargestellt ist, zu erreichen, wird das Analyse-Intervall um  $i$  Stellen gekürzt, was exakt der relativen Verschiebung entspricht. Ohne diese Korrektur entspricht dies dem dargestellten Fall in Abbildung 4.8 b).



**Abb. 4.8:** Korrektur des Analyse-Intervalls für eine verzögerungsfreie PS-Detektion. a) zeigt die Ausgabe des PS-Detektors, wenn es keine Verschiebung bzgl. der PS gibt. b) zeigt die Ausgabe des PS-Detektors, wenn es eine Verschiebung um  $i$  Stellen bzgl. der PS gibt. c) zeigt die Ausgabe des PS-Detektors und die Indexkorrektur, wenn es eine Verschiebung um  $i$  Stellen bzgl. der PS gibt.

### 4.3.3 CCA Clear Channel Assessment

Es wurden zwei Methoden implementiert, um zu erkennen, ob ein Kanal belegt ist. Es wurde ein Energiedetektor und ein PREAMBLE-Symbol-Detektor implementiert, welche parallel genutzt werden können.

Der Energiedetektor ist besonders gut geeignet, um die PRF zu ermitteln. Eine PRF von 15 Impulsen innerhalb von 992 ns ist charakteristisch für ein PREAMBLE-Symbol. Eine PRF von 16 Impulsen innerhalb von 1024, 128 oder 64 ns ist charakteristisch für eine UWB-Datenübertragung und ist ein Indiz dafür, dass gerade zwei UWB-Sensorknoten den Kanal zur Datenübertragung verwenden. Der PREAMBLE-Symbol-Detektor kann dazu verwendet werden, bestimmte PREAMBLE-Symbole zu erkennen und ist ein sicheres Indiz für eine UWB-Synchronisierung. Falls erkannt wird, dass der Kanal schon von anderen Sensorknoten verwendet wird, so kann auf einem anderen Kanal ausgewichen werden.

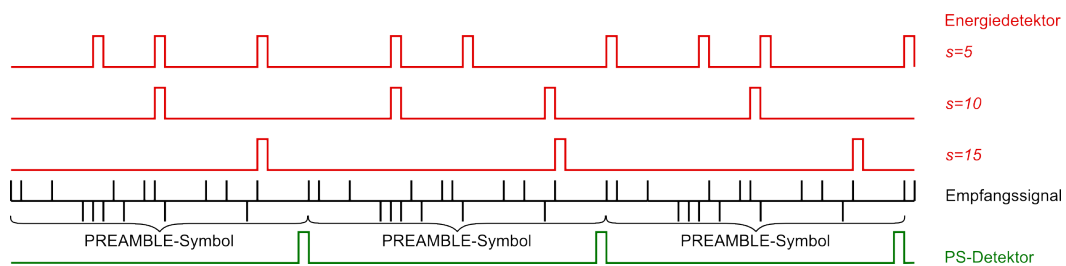
#### Energiedetektor

Es werden alle empfangenen Impulse summiert und bei Erreichen eines definierten Schwellenwertes, welcher über SPI eingestellt werden kann, wird ein Interrupt bei 'interrupt-a' ausgelöst. Nach dem Auslösen des Interrupts wird die Summe zurückgesetzt und neu gestartet. In Abbildung 4.9 sind drei Kurven für drei verschiedene Schwellenwerte  $s$  dargestellt.



### PREAMBLE-Symbol-Detektor

Hier wird das Empfangssignal nach PREAMBLE-Symbolen gescannt. Für jedes detektierte PREAMBLE-Symbol wird ein Interrupt bei *'interrupt - b'* ausgelöst. In Abbildung 4.9 ist solch ein Signal dargestellt. Da ein PREAMBLE-Symbol aus 16 Impulsen besteht, entspricht der PREAMBLE-Symbol-Detektor dem Energiedetektor mit Schwelle 15 und einer Verzögerung die von dem PREAMBLE-Symbol abhängig ist, sofern alle Impulse vorhanden sind und keine zusätzlichen Impulse empfangen wurden.



**Abb. 4.9:** CCA Energie- und PREAMBLE-Symbol-Detektor.

Im Gegensatz zum PS-Detektor hängt die Ausgabe des Energiedetektors vom eingestellten Schwellenwert  $s$  ab. Der Energiedetektor erzeugt immer ein Interrupt sobald  $s$  Impulse detektiert wurden.

#### 4.3.4 Testbarkeit

Es wurden 6 Testmodi und erweiterte Konfigurationsmöglichkeiten eingebaut, um verschiedene Teile des Chips zu testen.

##### On-Chip Basistest

In diesem Mode kann der Chip, ohne einen Mikrocontroller zu benutzen, dazu gebracht werden, spezielle Sequenzen zu senden, welche am Oszilloskop überprüft werden können. Damit ist es möglich Chips, schon vor dem Auflöten auf dem Board zu testen und wenn nötig auszusortieren. Die Auswahl der Sequenz geschieht mittels der 4 Eingänge (siehe Abbildung 4.2) *'testmode'*, *'spi - mosi'*, *'spi - cs\_n'*, *'spi - clk'* und *'mac - interrupt'*. Der Eingang *'testmode'* wird auf '1' gesetzt und *'spi - mosi'*, *'spi - cs\_n'*, *'spi - clk'* und *'mac - interrupt'* werden zu einem Signal *select* zusammengefasst. Tabelle 4.2 zeigt, welche Impulssequenzen damit generiert werden können.

Tabelle 4.2: On-Chip Testmodes

<i>select</i>	Impulssequenz
0000	.....
0001	+.....
0010	-.....
0011	.....
0100	zufällige Sequenz
0101	----++++----++-
0110	+ . + . + . + . - . - . - . - .
0111	++ . . ++ . . - . - . - . - .
1000	+++++++-----
1001	++++----++++----
1010	++--++--++--++--
1011	+ - + - + - + - + - + - + -
1100	+++ . . . . +++ . . . .
1101	++-- . . . . --++ . . . .
1110	+ - + - . . . . - + - + . . . .
1111	+ . - . . +++ . . - . - . - . - .

### Loop-Backs

Ein Loop-Back ist eine Rückkopplung bzw. eine Schleife. Dies bedeutet, dass an gewissen Schlüsselpositionen im Chip Signale umgeleitet werden, um bestimmte Funktionen zu testen. Es wurden insgesamt drei Loop-Backs implementiert. Es wurde festgestellt, dass diese drei Loop-Backs sehr hilfreich waren, um Fehler zu detektieren oder einzugrenzen. Die Loop-Backs verbinden an verschiedenen Stellen die Signale von TX zu RX. Anders ausgedrückt, sind Sender und Empfänger gleichzeitig aktiviert und der Ausgang vom Sender wird direkt in den Empfänger eingespeist. Deshalb ist es auch möglich sowohl Sender als auch Empfänger ohne externe Geräte testen zu können. Eigentlich ist es im UWB-Standard nicht vorgesehen, dass Sender und Empfänger gleichzeitig aktiviert sind.

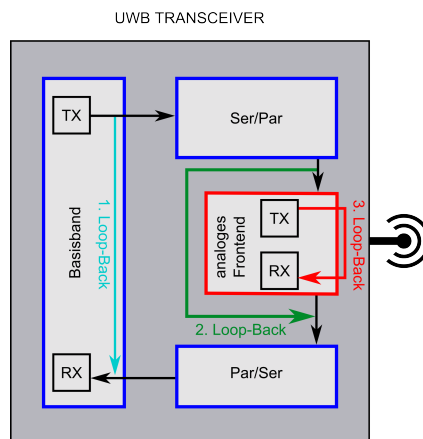
Im ersten Loop-Back ist nur das Basisband aktiviert. TX und RX laufen parallel und werden innerhalb des Basisbands verbunden. Dieser Test zeigt, ob das Basisband richtig funktioniert und ob auch alle Datenraten korrekt generiert und dekodiert werden können.

Im zweiten Loop-Back werden zusätzlich zum Basisband auch noch der Serialisierer, der Parallelisierer und die PLL im analogen Frontend aktiviert. Die PLL wird nur benötigt, um den hohen Takt von 499,2 MHz bereitzustellen. Diesmal wird der Ausgang des Serialisierers mit dem Eingang des Parallelisierers verbunden. Ein erfolgreicher Test beweist, dass die zeitkritischen Blöcke Serialisierer und Parallelisierer korrekt funktionieren.

Im dritten Loop-Back werden alle Komponenten aktiviert und der *antenna-switch* wird zur Dämpfung des Signals auf Senden eingestellt. Da das Sendersignal

nicht komplett abgeschirmt werden kann, empfängt der Empfänger dennoch dieses Signal in gedämpfter Form. Dieser Loop-Back ermöglicht das Testen des analogen Frontends. Allerdings treten hier aufgrund der Synchronizität des Empfängers und des Senders keine Phasenverschiebungsprobleme bzgl. der IQ-Pfade auf.

Abbildung 4.10 stellt alle drei Loop-Backs dar.



**Abb. 4.10:** Alle implementierten Loop-Backs.

1. Loop-Back: vom Ausgang vom Basisband TX zum Eingang vom Basisband RX.
2. Loop-Back: vom Ausgang des Serialisierers zum Eingang des Parallelisierers.
3. Loop-Back: Sowohl Sender als auch Empfänger sind im analogen frontend aktiviert.

### Parallelisierer

Weitere Testmöglichkeiten wurden speziell für den Parallelisierer implementiert. Über SPI ist es möglich, eine Impulssequenz, bestehend aus 16 beliebigen Impulsen zu definieren, welche in einer Endlosschleife gesendet wird. Parallel dazu kann der FIFO die Rohdaten des Parallelisierers speichern. Allerdings können aufgrund der großen Datenmengen von  $64 \text{ Bit}/32 \text{ ns} = 8 \text{ Byte}/32 \text{ ns}$  nur die empfangenen Daten der ersten  $\approx 600 \mu\text{s}$  im FIFO gespeichert werden. Dieser Test kann sowohl mit dem 2. als auch dem 3. Loop-Back realisiert werden. Besonders der 3. Loop-Back ist gut geeignet, um den analogen Empfänger zu testen.

Mit der Auswertung der Ausgangsdaten des Parallelisierers ist es möglich, die Bandbreite oder die Auswirkung schneller Phasenwechsel zu beobachten.

### GUI & Interface

Aufgrund des modularen Aufbaus können viele Komponenten separat eingeschaltet bzw. ausgeschaltet werden. Diese Vielfalt an Einstellungsmöglichkeiten ermöglicht es, den Chip ausführlich zu testen. Um den Chip einzustellen und das Interface zu testen, war es notwendig, eine GUI (Graphical User Interface) zu entwickeln. Die GUI wurde mittels Labview von National Instruments [79] erzeugt. Die Labview-Gui wird auf dem PC gestartet und mittels eines USB-zu-SPI Adapters von Totalphase [80] an den

Chip angeschlossen. Abbildung 4.11 zeigt die realisierte GUI.

Einstellbare Parameter sind:

- Verstärkung bzw. Gain
- Auswahl zwischen 4 Kanälen
- Ausschalten einzelner Blöcke im Frontend
- *antenna-switch* TX/RX
- Impulssequenz des PREAMBLE-Symbols
- Anzahl an PREAMBLE-Symbolen
- PREAMBLE-Symbol Korrelationsschwelle
- SFD Korrelationsschwelle
- untere Schwelle des FIFOs
- obere Schwelle des FIFOs
- Nachricht
  - standardkonform 1-127 Byte
  - nicht standardkonform bis 8191 Byte
- 5 Übertragungsmodi
- Ein/Ausschalten des Viterbi-Decoders
- Ein/Ausschalten des Reed-Solomon-Codes
- Ein/Ausschalten des CCA
- Einstellung der CCA-Energieschwelle
- Auswahl von drei Loop-Backs
- SPI-Test

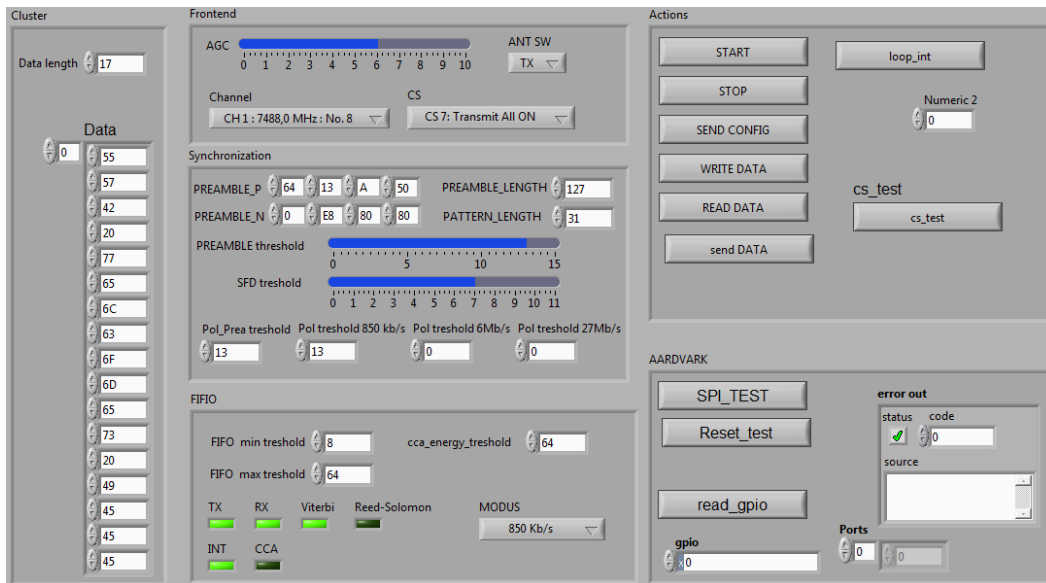


Abb. 4.11: Darstellung der entwickelten Labview GUI mit vielen Einstellungsmöglichkeiten.

### 4.3.5 Energiesparmechanismen

#### 4.3.5.1 Clock-Gating

Der Hauptanteil des Energieverbrauchs bei der CMOS-Technik ist von der Schalthäufigkeit der Gatter abhängig. Daher gilt: je höher die Taktfrequenz, desto höher die Verlustleistung.

Eine Möglichkeit die Schalthäufigkeit der Gatter eines Blockes zu reduzieren, ist die Verwendung eines Enable-Signals, welches den Block bei Bedarf deaktivieren kann. Allerdings wird beim Layoutvorgang ein Taktnetz generiert, welches bei komplexen Systemen schnell sehr große Ausmaße erreichen kann, siehe Abbildung 4.12.

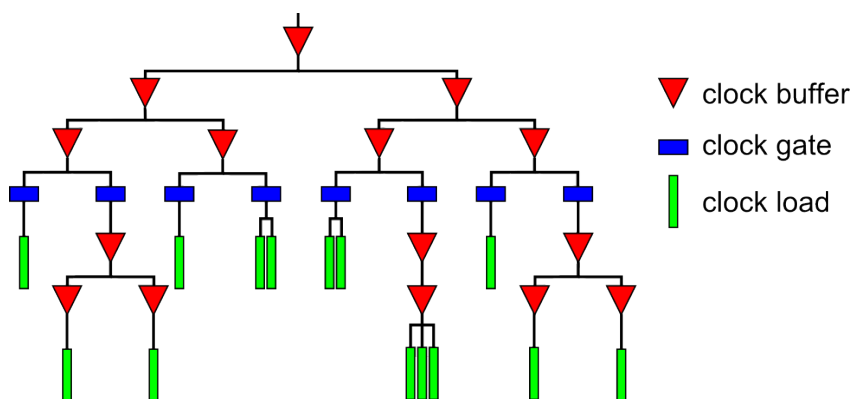


Abb. 4.12: Beispiel eines *clock-trees*

Ein Taktnetz besteht aus mehreren tausend Invertern. All diese Inverter schalten mit jedem Takt und sind daher für den hohen Leistungsverbrauch verantwortlich.

Der Leistungsverbrauch durch das Taktnetz kann mehr als 40% des gesamten Leistungsverbrauchs ausmachen [81]. Das Enable-Signal hat aber keinen Einfluss auf das Taktnetz, daher muss eine andere Lösung gefunden werden.

*Clock-gating* ist eine übliche Methode, um den Energieverbrauch des Taktnetzes zu reduzieren. Dabei werden an geeigneten Stellen im *clock-tree* Gatter platziert, welche die Taktdurchleitung unterbinden. Auch hier ist ein modularer Aufbau der Hardware hilfreich, um solche Stellen zu finden. Wichtig ist, dass der Schaltvorgang im *clock-gate* ohne Erzeugung eines *glitch* zu erfolgen hat.

#### 4.3.5.2 Power-Gating

*Power-gating* ist eine tiefer gehende Technik als *clock-gating*, um Energie zu sparen. Beim *power-gating* wird der ganze Block abgeschaltet und es fließt kein Strom mehr. Allerdings hat diese Methode einige Nachteile. Solche Designs führen zu erhöhten Latenzzeiten, da die Sicherstellung einer korrekten Umschaltung in die verschiedenen *Power-Modi* mehr Zeit benötigt. Es muss auch abgewogen werden, ob der eingesparte Leistungsverbrauch den Aufwand (zusätzliche Logik und schwierigeres Layout) rechtfertigt und ob der zusätzliche Energieverbrauch für das Umschalten der *Power-Modi* nicht zu hoch ist. *Power-gates* können via Software oder Hardware gesteuert werden.

Das einfachste *power-gating* ist ein externer Schalter, welcher die Versorgungsspannung abschaltet. Diese Methode ist gut geeignet, wenn der Block für längere Zeit ausgeschaltet werden soll. Für andere Fälle, wenn schnell und in kürzeren Intervallen umgeschaltet werden soll, sind interne *power-gatings* besser geeignet.

Komplizierter wird es, wenn die Blöcke zwar abgeschaltet werden sollen, aber deren in Flipflops bzw. Registern gespeicherten Informationen beibehalten werden sollen. In solchen Fällen muss sichergestellt werden, dass diese Informationen vor dem Abschalten zwischengespeichert werden und nach dem Einschalten wieder zurückgeladen werden. Der Unterschied zwischen einem herkömmlichen Design und einem Design mit *power-gating* ist in Abbildung 4.13 dargestellt.

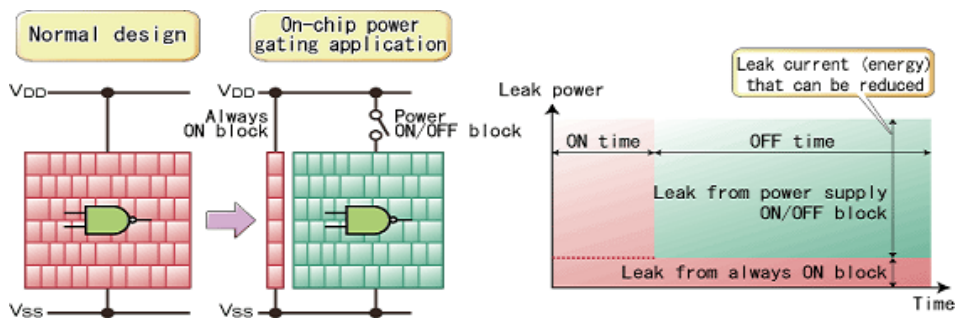


Abb. 4.13: Vergleich eines *power-gating* Designs mit einem herkömmlichen Design, entnommen aus [82].

### 4.3.5.3 Analoge Sleep-modi

Das analoge Frontend ist auch modular gestaltet und viele Blöcke lassen sich einzeln ausschalten, um Energie zu sparen. Die Blöcke lassen sich nur komplett ein/auschalten, was zu langen Schaltzeiten von mehreren Mikrosekunden führen kann. Es ist aber möglich, feinere Sleep-Modi einzuführen, welche es ermöglichen, fast den kompletten Transmitter innerhalb von 9 ns, und den Empfänger innerhalb von 30 ns ein- und auszuschalten, eventuell sind auch noch kürzere Zeiten möglich.

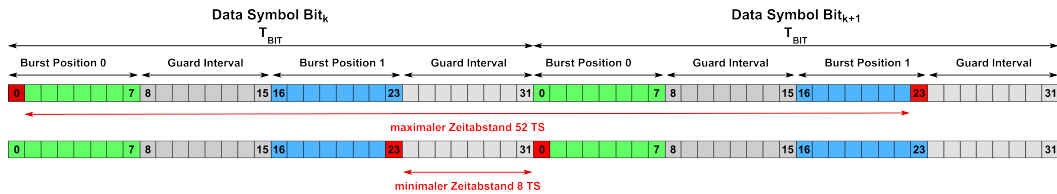
Wichtige Fragen müssen noch geklärt werden:

1. Wie können diese Umschaltzeiten genutzt werden, um während einer Übertragung Energie zu sparen?
2. Wie muss das Basisband erweitert werden, um dies zu unterstützen?

Im Rest des Abschnittes folgt eine Analyse darüber, wie viel Energie in Abhängigkeit der Umschaltzeit bei einer standardkonformen bzw. einer optimierten Übertragung eingespart werden kann. Danach wird eine effiziente Umsetzung in Hardware dargestellt.

### Standardübertragung

Die Positionsbestimmung der Burst Positionen ist bei einer Standardkonformen Übertragung nicht zufällig, sondern wird mittels eines LFSRs und des Datenbits bestimmt (siehe Abbildung 4.14).



**Abb. 4.14:** Darstellung des minimalen und maximalen zeitlichen Abstands zwischen zwei Bursts.

Der LFSR legt den *time-slot* von 1 bis 8 fest und das Datenbit legt fest, ob der Burst im linken oder rechten Teil des Datensymbols auftaucht. Demzufolge ist die endgültige Position des  $k$ -ten Bit  $Bit_k$  durch folgende Formel gegeben:

$$pos_{Bit_k} = LFSR_{Bit_k} + 16 \cdot Bit_k$$

Der Abstand zwischen zwei aufeinanderfolgenden Bits beim Sender ist:

$$Abstand_{TX} = pos_{Bit_{k+1}} + 32 - pos_{Bit_k} = LFSR_{Bit_{k+1}} + 16 \cdot Bit_{k+1} - LFSR_{Bit_k} - 16 \cdot Bit_k$$

Da aber der Empfänger nicht im Vorfeld wissen kann, welchen Wert das  $(k+1)$ -ten Bit hat, muss vom Worst-Case ausgegangen werden.  $Bit_{k+1} = 0$

Demzufolge ist der Abstand zwischen zwei aufeinanderfolgenden Bits beim Empfänger:

$$Abstand_{RX} = LFSR_{Bit_{k+1}} - LFSR_{Bit_k} - 16 \cdot Bit_k$$

Abbildung 4.15 zeigt, wie oft welcher Abstand für TX (in rot markiert) und RX (in blau markiert) durchschnittlich vorhergesagt werden kann. Aus dieser Kurve ergeben sich durchschnittliche Abstände von 32,08 *time-slots* für TX und 24,07 *time-slots* für RX. Dies gilt für alle Datenraten.

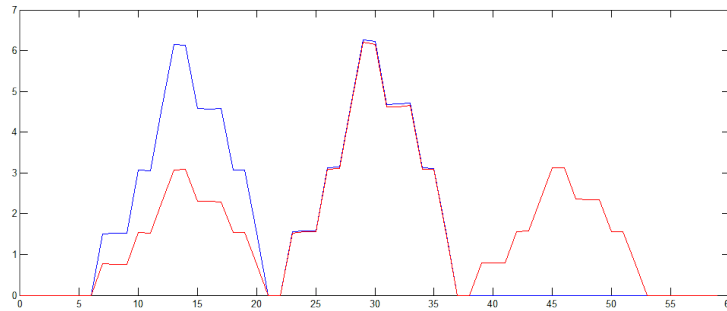


Abb. 4.15: Burst-Position-Vorhersage

Bei diesen Werten wurde aber noch nicht die Umschaltzeit berücksichtigt. Es lohnt sich, nur den Empfänger bzw. Transmitter zwischenzeitlich zu deaktivieren, wenn:

$$Umschaltzeit = Einschaltzeit + Ausschaltzeit \geq \text{Zeitabstand}$$

Abbildung 4.16 zeigt den durchschnittlichen Zeitabstand zwischen zwei Bursts. Außerdem sind auch die Durchschnittswerte für die drei Datenraten 851 kb/s, 6,81 Mb/s und 27,24 Mb/s für eine Umschaltzeit von 39 ns abgebildet. Diese Werte wurden wie folgt ermittelt:

$$Umschaltzeit = Einschaltzeit + Ausschaltzeit = 9 \text{ ns} + 30 \text{ ns} = 39 \text{ ns}$$

Für 851 kb/s gilt:  $TS=32 \text{ ns}$  ;  $Umschaltzeit \approx \lceil \frac{39}{32} \rceil = 2 \cdot TS$ .

Für 6,81 Mb/s gilt:  $TS=4 \text{ ns}$  ;  $Umschaltzeit \approx \lceil \frac{39}{4} \rceil = 10 \cdot TS$ .

Für 27,24 Mb/s gilt:  $TS=2 \text{ ns}$  ;  $Umschaltzeit \approx \lceil \frac{39}{2} \rceil = 20 \cdot TS$ .



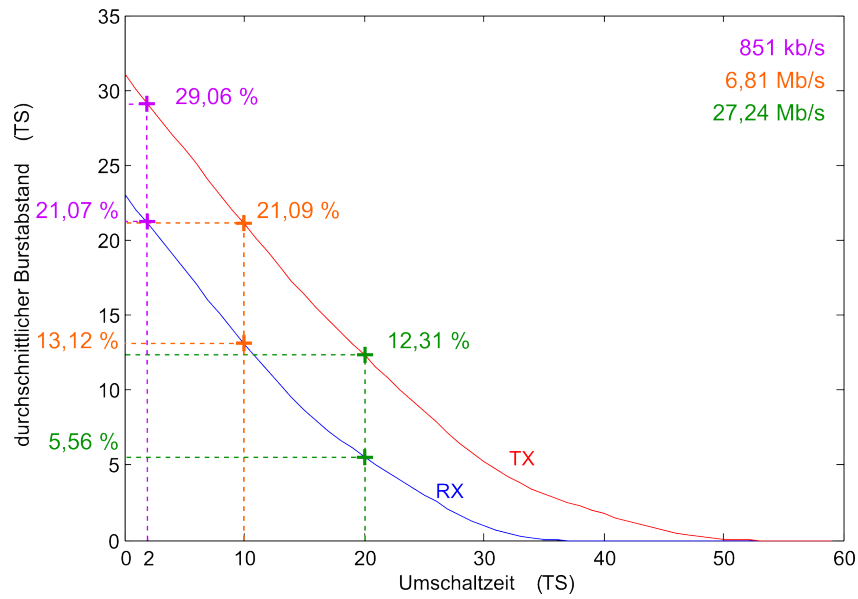


Abb. 4.16: Durchschnittlicher Burstabstand abhängig von der Umschaltzeit und der verwendeten Datenrate.

Wie in Abbildung 4.16 erkennbar ist, ist die Energieersparnis nicht nur von der Umschaltzeit, sondern auch von der Datenrate stark abhängig. Abbildung 4.17 zeigt die zu erwartende Energieersparnis abhängig von der Umschaltzeit.

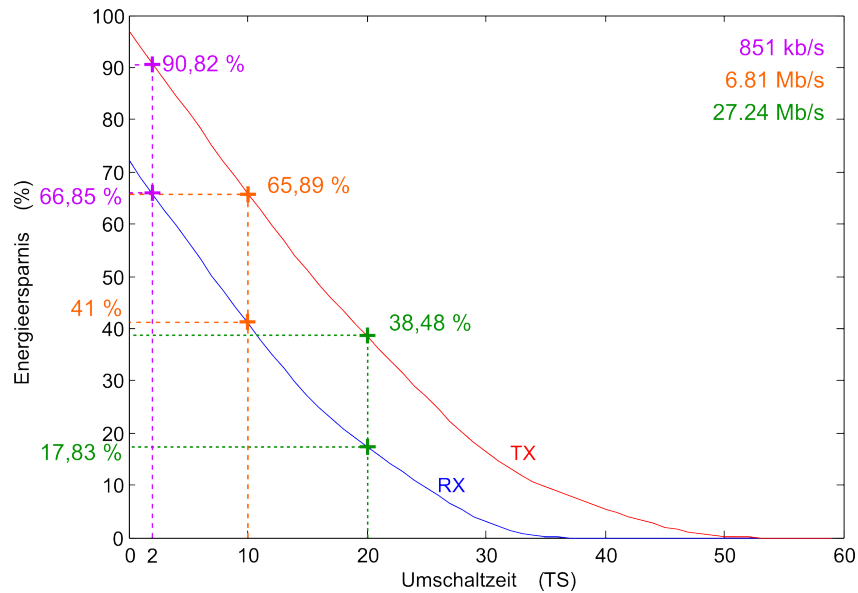


Abb. 4.17: Energieersparnis abhängig von der Umschaltzeit und der verwendeten Datenrate.

Zusammengefasst bedeutet das, dass für jedes übertragene Bit folgende durchschnittliche aktive Zeiten gelten:

- bei der Übertragungsrate von 851 kb/s :

$$\begin{aligned} \text{TX} &= 1024 \text{ ns} \cdot (1 - 90,82\%) = 94 \text{ ns} \\ \text{RX} &= 1024 \text{ ns} \cdot (1 - 66,85\%) = 339,5 \text{ ns} \\ \text{Durchschnitt} &= 216,75 \text{ ns} \end{aligned}$$

- bei der Übertragungsrate von 6,81 Mb/s :

$$\begin{aligned} \text{TX} &= 128 \text{ ns} \cdot (1 - 65,89\%) = 43,66 \text{ ns} \\ \text{RX} &= 128 \text{ ns} \cdot (1 - 41,00\%) = 75,52 \text{ ns} \\ \text{Durchschnitt} &= 59,59 \text{ ns} \end{aligned}$$

- bei der Übertragungsrate von 27,24 Mb/s :

$$\begin{aligned} \text{TX} &= 64 \text{ ns} \cdot (1 - 38,48\%) = 39,37 \text{ ns} \\ \text{RX} &= 64 \text{ ns} \cdot (1 - 17,83\%) = 52,59 \text{ ns} \\ \text{Durchschnitt} &= 45,98 \text{ ns} \end{aligned}$$

### Optimierte UWB-Datenübertragung

Im Gegensatz zur Standardübertragung, gibt es in der optimierten Übertragung aus

Kapitel 3 keine Bursts mehr. Allerdings sind die Positionen für die Impulse fest definiert und ermöglichen auch hier eine Optimierung.

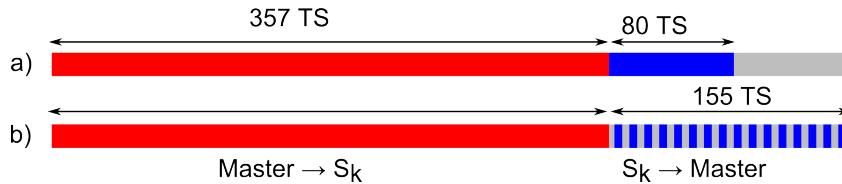
Der Master überträgt 16 Impulse innerhalb von  $357 \cdot TS \approx 714 \text{ ns}$ . Die Symboldauer liegt bei  $16 \cdot 32 \cdot TS \approx 1024 \text{ ns}$ . Die 16 Slaves übertragen ihre Daten jeweils mit 3 Impulsen innerhalb von  $5 \cdot TS$ . Demzufolge ergibt sich eine Gesamtdauer von  $16 \cdot 5 \cdot TS \approx 80 \text{ ns}$ .

Es müssen folgende zwei Fälle betrachtet werden:

1.  $TX_{S_k} \rightarrow RX_{Master}$ :

**Slave:** Der Abstand zwischen den gesendeten Impulsen ist zu gering, um den Sender während der Übertragung abschalten zu können. Daraus folgt eine aktive Zeit für  $TX_{S_k}$  von  $5 \cdot TS + Umschaltzeit = 25 \cdot TS \approx 50 ns$ . Das entspricht  $50/1024 = 4,88\%$  der Symboldauer.

**Master:** Der Master kann nur vorhersagen, wann welcher  $S_k$  aktiv ist und kann daher nur zwischen zwei aktiven Phasen von  $S_k$  und  $S_k + 1$  den Empfänger ausschalten. Dies hängt aber davon ab, wie die aktiven Phasen innerhalb des Symbolframes verteilt sind. In Abbildung 4.18 a) und b) sind zwei mögliche Fälle abgebildet. Der Fall b) beschreibt eine Verteilung mit möglichst großen Abstand zwischen den verschiedenen Slaves, was eine robustere Übertragung ermöglicht, da eventuell durch Reflexion verursachte Impulse vermieden werden. Allerdings sind die Abstände zwischen den aktiven Phasen nicht ausreichend groß, um den Empfänger des Masters abzuschalten. Der optimale Fall a) ist dann gegeben, wenn alle  $S_k$  nacheinander ohne Pause aktiv sind. Dann ergibt sich eine aktive Empfangsphase für den Master von  $16 \cdot 5 \cdot TS + Umschaltzeit = 90 \cdot TS \approx 180 ns$ . Das entspricht  $180/1024 = 17,58\%$  der Symboldauer.



**Abb. 4.18:** Optimale Verteilung der Sendephasen der Slaves.  
 a) gruppiert die Übertragungsintervalle aller Slaves innerhalb von 80 TS, um Energie zu sparen.  
 b) verteilt die Übertragungsintervalle aller Slaves auf 155 TS, um die Zuverlässigkeit zu erhöhen.

2.  $TX_{Master} \rightarrow RX_{S_k}$ :

**Slave:** Der einzelne Slave kann im Empfangsmodus nicht vorhersagen, wann welcher Impuls vom Master gesendet wird. Allerdings ist bekannt, dass der Master nur 3 Impulse aussendet. Somit kann die aktive Empfangsphase beendet werden, sobald 3 Impulse empfangen wurden. Um die Energieersparnis zu ermitteln, muss die durchschnittliche Zeit der Übertragung für 3 Impulse berechnet werden. In Abbildung 4.19 stellt die cyanfarbige Kurve die Häufigkeit dar, mit welcher der letzte der 14 Impulse bei jedem möglichen *time-slot* auftritt. In Abbildung 4.20 zeigt die cyanfarbige Kurve das durchschnittliche Auftreten des 14. Impulses je *time-slot*. Daraus folgt, dass der Slave durchschnittlich für  $339,6 \cdot TS + Umschaltzeit \approx 699,2 ns$  aktiv ist. Das entspricht

$699, 2/1024 = 68, 28\%$  der Symboldauer.

**Master:** Im Unterschied zu den Slaves, kann der Master vorhersagen wann er welchen Impuls aussendet. Demzufolge auch die Impulsabstände und kann, wenn diese ausreichend groß sind, den Sender zwischendurch abschalten. In Abbildung 4.19 zeigt die blaue Kurve, wie oft ein Impulsabstand von  $x \cdot TS$  auftritt und die grüne Kurve zeigt, wie oft der Sender des Masters für  $y \cdot TS$ , inklusive Umschaltzeit, aktiviert ist. In Abbildung 4.20 zeigt die blaue Kurve, wie wahrscheinlich es ist, dass der Impulsabstand größer als  $x \cdot TS$  ist. Die grüne Kurve zeigt, wie wahrscheinlich es ist, dass die aktive Sendezeit kleiner als  $y \cdot TS$  ist. Daraus folgt, dass der Master beim Senden durchschnittlich für  $201, 3 \cdot TS = 402, 6 ns$ , inklusive Umschaltzeit, aktiv ist. Das entspricht  $402, 6/1024 = 39, 32\%$  der Symboldauer.

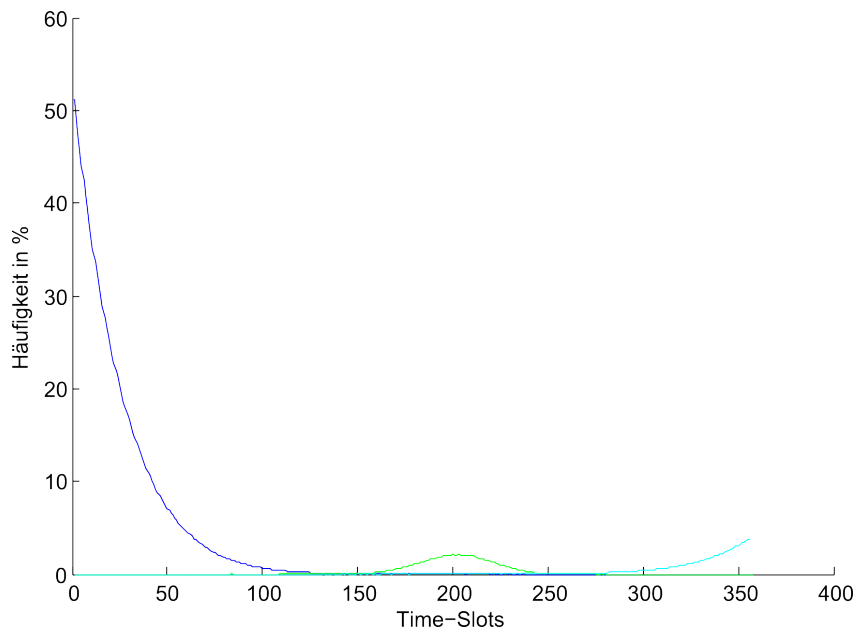


Abb. 4.19: Häufigkeiten verschiedener Parameter für den Fall  $TX_{Master} \rightarrow RX_{S_k}$

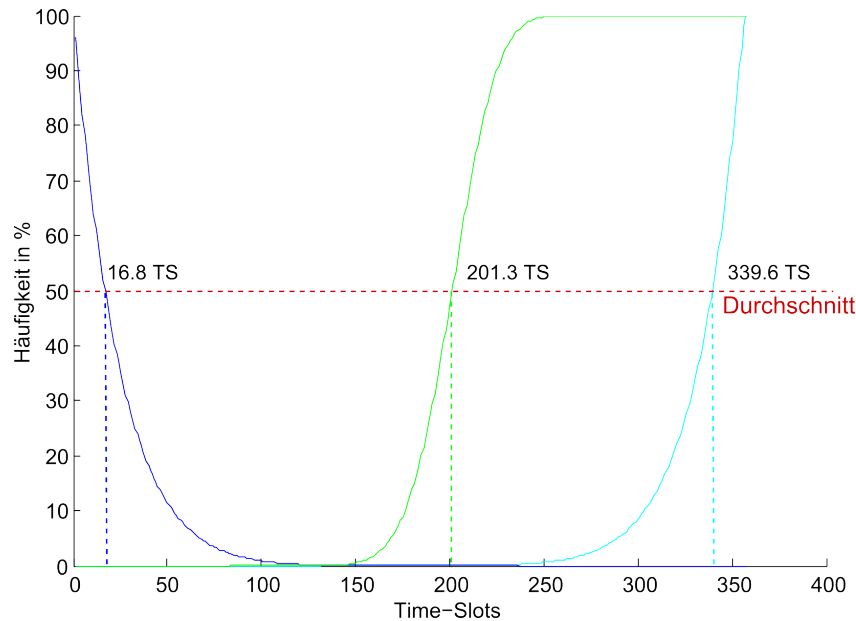


Abb. 4.20: Durchschnittliche Häufigkeiten verschiedener Parameter für den Fall  $TX_{Master} \rightarrow RX_{S_k}$

Zusammengefasst bedeutet das, dass für jedes übertragene Bit Folgende durchschnittliche aktive Zeiten gelten:

- Master:
  - TX=402,6 ns/96 = 4,19 ns
  - RX=180 ns/96 = 1,88 ns
  - Durchschnitt=3,04 ns
- Slave:
  - TX=50 ns/6 = 8,33 ns
  - RX=699,2 ns/(96/16) = 116,53 ns
  - Durchschnitt=62,43 ns

Im Vergleich zur standardkonformen Übertragung hat sich die aktive Zeit pro Bit stark verkürzt, außer beim Slave, wenn dieser Daten empfängt. In diesem Fall hat sich die aktive Zeit etwas mehr als verdoppelt.

Der Energieverbrauch ist minimal, wenn der Master mehr Daten von den Slaves empfängt als dieser Daten an die Slaves übermittelt. Dies ist der am häufigsten auftretende Fall bei Automatisierungssystemen. In solchen Systemen werden von einem Master überwiegend regelmäßig Statusinformationen von Sensoren abgefragt, was genau diesem Szenario entspricht.

### Auswirkung der Umschaltzeit auf die Synchronisation

Sowohl für die Standardübertragung als auch für die optimierte Übertragung bestehen die PREAMBLE-Symbole aus einzelnen vorhersagbaren Impulsen. Aus diesem Grund liegt hier die Umschaltzeit bei 2 ns und die aktive Zeit für TX pro

Impuls liegt bei 6 ns: 2 ns für das Einschalten, 2 ns für die Impulsübertragung und 2 ns für das Ausschalten. Daraus ergibt sich eine aktive Zeit pro PREAMBLE-Symbol von  $n \cdot 6$  ns, wobei  $n$  die Anzahl an Impulsen im PREAMBLE-Symbol darstellt. Tabelle 4.3 zeigt wie lang der Sender und der Empfänger, bei verschiedenen Übertragungsarten aktiv sind. Außerdem wird angezeigt um wieviel Prozent die Summe der aktiven Zeiten der Verschiedenen Verfahren, im Vergleich zum Standard-Verfahren, reduziert werden konnten.

**Tabelle 4.3:** Unterschiedliche aktive Zeiten des Senders und Empfängers, für eine komplette Übertragung, in Abhängigkeit der Übertragungsart

	Standard		Standard*		Optimiert		Optimiert*	
	TX	RX	TX	RX	TX	RX	TX	RX
$SYNC_M$ (ms)	3,02	3,02	3,02	3,02	0,34	0,34	0,34	0,34
$SYNC_S$ (ms)	3,02	3,02	3,02	3,02	0,17	5,43	0,17	5,43
$DATA_M$ (ms)	6,50	6,50	2,64	4,07	0,39	0,17	0,22	0,1
$DATA_S$ (ms)	6,50	6,50	2,64	4,07	0,43	6,16	0,43	6,16
$TOTAL_M$ (ms)	9,52	9,52	5,66	7,09	0,73	0,51	0,56	0,44
$TOTAL_S$ (ms)	9,52	9,52	5,66	7,09	0,6	11,63	0,6	11,63
<i>Summe</i> (ms)	38,08		25,5 (-33%)		13,47 (-64,6%)		13,23 (-65,2%)	

\* Aktive Zeit unter Berücksichtigung der Umschaltzeit

Dadurch ist die erste Frage, wie die Vorhersage der Burst zur Energieeinsparung eingesetzt werden kann, geklärt.

### Implementierung der Umschaltzeit

In diesem Abschnitt wird erklärt, wie das Basisband schnellstmöglich die Umschaltzeit umsetzen kann. Da das Basisband nur mit 31,2 MHz arbeitet, ist eine direkte Feinabstimmung der Ausschaltzeit bzw. Einschaltzeit nicht möglich. Allerdings könnte eine Auflösung von 2 ns erreicht werden, wenn wie bei den Daten ein Serialisierer zur Einstellung eingesetzt wird. Es wäre möglich, einen weiteren Serialisierer, wie in Abschnitt 4.2.3 beschrieben, einzusetzen, und auf diese Weise das Ziel zu erreichen. Dies würde aber einen weiteren kritischen Block hinzufügen, welcher auch wieder zusätzliche Fläche und Strom verbrauchen würde.

Aus diesem Grund wurde eine andere Möglichkeit gesucht, um dies mit der schon bestehenden Hardware realisieren zu können. Im Abschnitt 4.2.3 und insbesondere in Tabelle 4.1 wurde gezeigt, dass eine Kombination von *OOK* und *BPSK* nicht auftreten kann. Der Fall tritt auf, wenn  $OOK = 0$  und  $BPSK = 1$  sind. Diese Kombination hat keinerlei Auswirkung auf den Sender. Denn ein Impuls wird nur dann generiert wenn  $OOK = 1$  ist.

Die Idee ist nun, diese eine Kombination zu benutzen, um dem analogen Frontend mitzuteilen, dass der Sender bzw. Empfänger umgeschaltet werden soll. Da eine Umschaltung auch nur zwischen zwei Bursts erfolgen kann, wird es zu keiner Kollision zwischen Umschaltkommando und Burst-Sequenz kommen, siehe Abbildung 4.21.

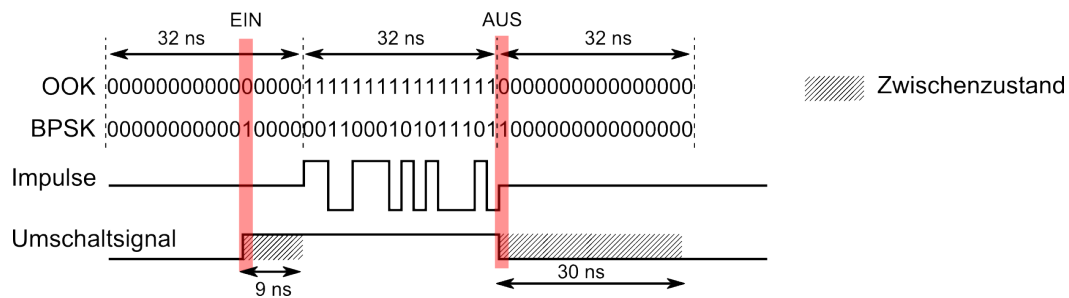


Abb. 4.21: Darstellung des Umschaltsignals, welches mittels dem Serialisierer generiert wird.

Es fehlt noch die Logik, die anhand von *OOK* und *BPSK* ein Umschaltsignal generiert. Dies kann mit nur 4 Gattern erreicht werden, siehe Abbildung 4.22. Diese Logik ist asynchron und kann wahlweise im Frontend bzw. im Basisband implementiert werden. Diese Logik ist nicht zeitkritisch.

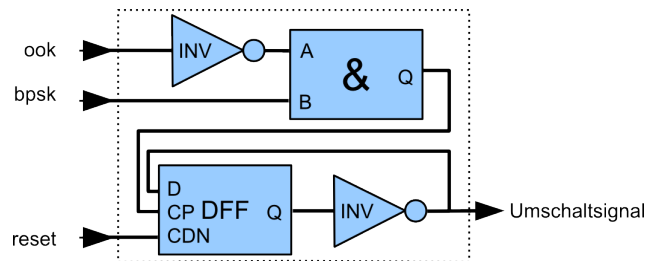


Abb. 4.22: Implementierung der Umschaltlogik

Die Energieersparnis galt bisher nur für das analoge Frontend, aber es wäre durchaus möglich, mittels *clock-gating* auch das Basisband zwischen zwei Bursts abzuschalten. In solch einem Fall würde sich der ausnutzbare Zeitabstand zwischen zwei Bursts um 64 ns verringern, da ein Taktzyklus 32 ns für das Einschalten des Frontends und ein Taktzyklus von 32 ns für das Ausschalten des Frontends benötigt wird.

## 4.4 ASIC Implementierung

Tabelle 4.4 listet die wichtigsten Eigenschaften der im IHP-gefertigten UWB-Chips auf. Ein Chipfoto ist in Abbildung 4.23 zu sehen. Abbildung 4.24 gibt einen Einblick in die Funktionsweise des analogen Frontends. Abschließend werden in Tabelle 4.5 einige andere aktuelle UWB-Entwicklungen mit der hier entwickelten Lösung verglichen. Bei der aktuellen Lösung wurden nicht alle aus Kapitel 3 und 4 vorgeschlagenen Optimierungen integriert. Daher gibt es auch noch Verbesserungspotential bzgl. des Leistungsverbrauchs. Die wesentlichen Features der letzten ASIC-Version vom 21.09.2014 sind:

- standardkonforme Übertragung für 851 kb/s, 6,81 Mb/s und 27,24 Mb/s
- variable PHR-Datenrate

- 3 Loop-Backs
- Broadcast-Nachrichten mit Nachrichtenlängen von bis zu  $2^{13} - 1 = 8191$  Byte
- getriggerte Übertragung
- Unterstützung für Streaming-Modus (Nachricht wird *on-the-fly* ohne Zwischenspeicherung kodiert und ausgesendet). Dieser Modus erfordert eine schnelle SPI-Verbindung.
- Built-in-self-tests
- integrierter CRC-Check
- variable Impulssequenz des PREAMBLE-Symbols
- variable Länge des PREAMBLE-Symbols
- einstellbare Anzahl an PREAMBLE-Symbolen im PREAMBLE-Teil von 0-4096
- integrierter digital steuerbarer Komparator im analogen Frontend
- Impulspolarität kann konstant gesetzt werden (dies kann helfen die Reichweite für nicht-kohärente Empfänger zu erhöhen)

**Tabelle 4.4:** Chip Parameter, entnommen aus [19].

Si technology	IHP SGB25V
Chip size	3.65 mm x 3.65 mm
Operation channel	5, 6, 8, 9 (mandatory)
Data rate	0.85, 6.81, 27.24 Mbps
Signal bandwidth	499.2 MHz
Reference clock	31.2 MHz
PLL settling time	< 10 $\mu$ s
Baseband processor	interface SPI
Baseband processor	clock 31.2 MHz
Transmitter RF output amplitude	250 mVpp
Transmitter modulation schemes impulse detection	BPM, BPSK Coherent
Receiver sensitivity	-65 dBm
Receiver maximum voltage gain	70 dB
Programmable gain setting	4 bit
Supply voltage	2.85 – 3.3 V
Max. power diss. – BB processor (including output CMOS pads)	60 mW, 112.2 mW
Max. power diss. – Frequency synthesizer	97.4 mW
Max. power diss. – Transmitter	104.3 mW
Max. power diss. – Receiver	121.6 mW



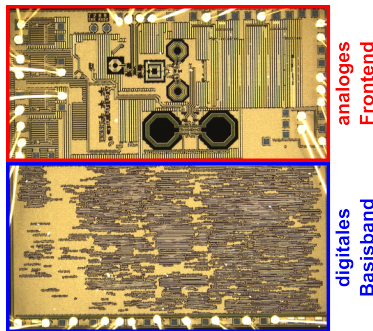


Abb. 4.23: Chipfoto, wobei die obere Hälfte das analoge Frontend und die untere Hälfte das digitale Basisband darstellt.

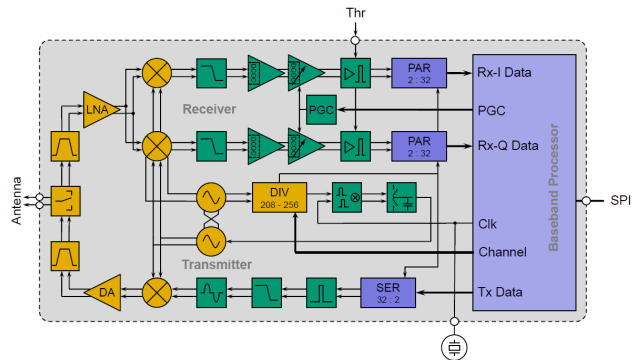


Abb. 4.24: Block-Diagramm des analogen Frontends

Tabelle 4.5: Vergleich verschiedener aktueller Transceiver, entnommen aus [19].

	[83]	[84]	[85]	TANDEM [17]	KUSZ [19]
Process	0.18 $\mu\text{m}$ CMOS	0.13 $\mu\text{m}$ CMOS	0.13 $\mu\text{m}$ CMOS	0.25 $\mu\text{m}$ BiCMOS	0.25 $\mu\text{m}$ BiCMOS
Supported channels	Ch1-Ch12	Ch3	Ch1-Ch3	Ch9	Ch5,6,8,9
Supported data rates [Mbps]	-	-	0.85	0.85	0.85, 6.81, 27.24
Supply	1.8 V	1.2 V	1.2 V	2.85 V	2.85 V
Power dissipation		Rx+Tx 34 mW			
Rx active	127 mW		36 mW	203 mW	219 mW
Tx active	86 mW		46 mW	134 mW	201 mW
Baseband		11 mW	245 mW	32 mW	112 mW
Receiver type	Coherent	Non-Coherent	Coherent	Non-Coherent	Coherent
Radio range	2.2 m	< 15 m	20 m	> 10 m	
Integration level	RF-frontend	Full Transceiver	Full Transceiver	Full Transceiver	Full Transceiver
Remarks		Mask violation	2 chips solution		

Im KUSZ-Projekt war es nicht vorgesehen, eine Vollduplex-Übertragung zu implementieren. Daher konnte dieses Übertragungsverfahren nicht praktisch getestet werden. Allerdings wurde bewiesen, dass alle Voraussetzungen für eine Vollduplex-Übertragung erfüllt sind. Es gelang Broadcastnachrichten zu übertragen und einzelne Impulse konnten ausgewertet werden. Außerdem konnte ein simultaner Betrieb von Sender und Empfänger mithilfe der internen Loop-Backs realisiert und getestet werden.

Aufgrund der implementierten variablen PREAMBLE-Symbolen, konnten verschiedene PREAMBLE-Symbole genutzt und deren Auswirkungen auf die Synchro-

nisation getestet werden.

#### 4.4.1 Ressourcenverbrauch

Die letzte Version des ASICS wurde am 21.08.2014 zur Fertigung eingereicht. In dieser Version wurde der Viterbi-Dekoder entfernt und die Synchronisation sowie die Datenerkennung wurden effizienter implementiert. Die verwendete Chipfläche konnte von  $4.8 \mu\text{m}^2$  auf  $3.13 \mu\text{m}^2$  gesenkt werden. Zusätzlich wurden Serialisierer und Parallelisierer integriert, um eine bessere Abgrenzung zwischen analoger und digitaler Domäne zu gewährleisten. Diese Abgrenzung soll helfen Störungen, in der analogen Domäne zu reduzieren.

Die wichtigsten Komponenten des Basisbands wurden bezüglich der Energieeffizienz mit der Software PrimeTime von Synopsys [86] untersucht. Es wurden keine *clock-gates* oder *power-gates* implementiert. Die Tabelle 4.6 gibt einen Überblick über den Leistungs- bzw. Flächenverbrauch des ganzen Systems und der wichtigsten Teilkomponenten.

**Tabelle 4.6:** Flächen- und Leistungsverbrauch der wichtigsten Komponenten

	SWITCH (mW)	PEAK (mW)	INTERNAL (mW)	TOTAL (mW)	AREA ( $\mu\text{m}^2$ )	POWER- DENSITY ( $\text{mW}/\text{mm}^2$ )
<i>Gesamt</i>	5,71	219	16,9	22,70	3131975	7,25
<i>SER/PAR</i>	4,7	182	7,96	12,70	51903	244,69
<i>PAR/SER</i>	0,73	25,4	0,03	2,01	14803	135,78
<i>BASISBAND</i>	3,85	183	0,02	7,78	3065239 / 100%	2,54
<i>SYNC</i>	0	35,7	1,53	1,54	1221054 / 39,8%	1,26
<i>FIFO</i>	0,02	110	2,39	2,41	538118 / 17,6%	4,48
<i>TRANSMITTER</i>	0	19,6	0,49	0,49	211193 / 6,9%	2,32
<i>RS_DECODER</i>	0	31,7	1,37	1,38	713996 / 23,3%	1,93
<i>RS_ENCODER</i>	0	9,44	0,39	0,39	123183 / 4,0%	1,93
<i>INTERFACE</i>	0	38,5	0,78	0,79	86273 / 2,8%	9,16
<i>REST</i>	0,26	—	1,96	0,99	171452 / 5,6%	5,77

Bezüglich des Energieverbrauches gibt es 5 verschiedene Parameter:

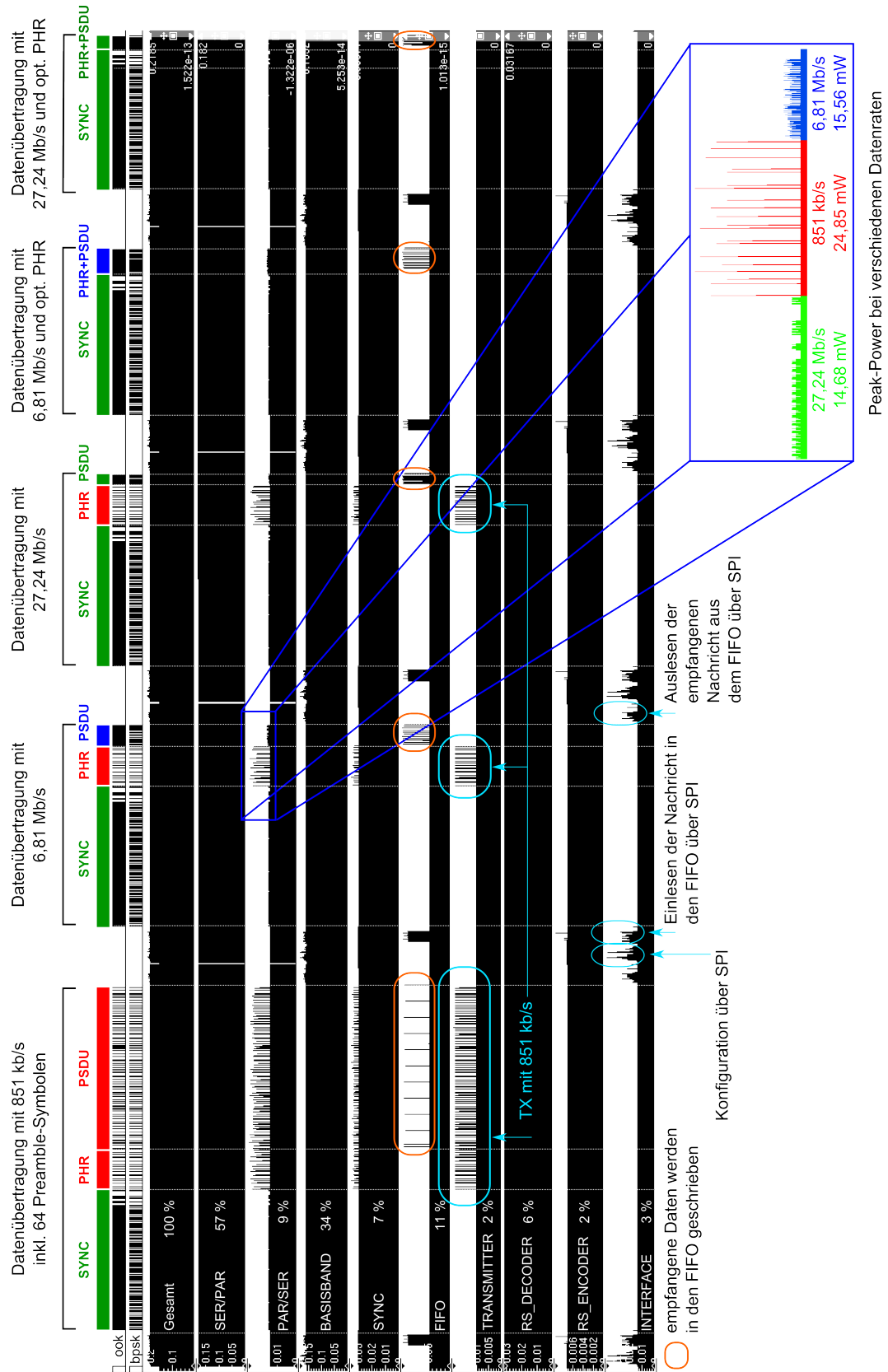
- *switching-power*: Das ist der durchschnittliche Leistungsverbrauch, welcher durch das Umschalten der verschiedenen Gatter verbraucht wird.
- *internal-power*: Dies ist der Leistungsverbrauch der auftritt, wenn sich die Eingangswerte der Gatter ändern, aber der Ausgang unverändert bleibt.
- *peak-power*: Dies ist der maximale Momentanleistungsverbrauch, welcher üblicherweise nur kurz auftritt.
- *total-power*: Dies ist der durchschnittliche Leistungsverbrauch.

- *leakage-power*: Dies ist die Leckleistung, welche durch Leckströme verursacht wird. Dieser Wert wird in dieser Arbeit nicht berücksichtigt da dieser gegenüber den anderen Werten vernachlässigbar ist. Die Leckleistung für das ganze Basisband liegt bei ungefähr  $20 \mu W$ .

Die Fläche der einzelnen Komponenten kann dazu verwendet werden, die Komplexität abzuschätzen.

Die *power-density* ist wichtig, um mögliche Hotspots zu erkennen. Hotspots sind Stellen, welche eine erhöhte Temperatur als der restliche Chip aufweisen. Größere Hotspots können dazu führen, dass der Chip nicht richtig funktioniert oder nur für eine begrenzte Zeit korrekt funktioniert.

Eine ausführlichere Analyse der einzelnen Komponenten ist mit Hilfe der Abbildung 4.25 möglich. In dieser Abbildung wurde der Leistungsverbrauch für 5 verschiedene Übertragungsarten aufgenommen. Die ersten drei Übertragungen sind standardkonform. Bei den beiden letzten Übertragungen werden sowohl PHR als auch PSDU mit derselben Datenrate übertragen. In allen 5 Fällen sind sowohl Sender als auch Empfänger gleichzeitig aktiviert. Damit kann abgeschätzt werden, welcher Leistungsverbrauch bei einem Vollduplex-Verfahren zu erwarten ist, da dort auch Sender und Empfänger gleichzeitig aktiviert sind.



**Abb. 4.25:** Momentanleistung bei verschiedenen Übertragungsraten. Die Momentanleistungen der wichtigsten Blöcke des UWB-Transceivers bei den verschiedenen Phasen eines UWB-Frames sind hier graphisch dargestellt.

Im Folgenden wird auf die einzelnen Komponenten eingegangen und deren Energieverbrauch genauer betrachtet. Das Gesamt-System besteht aus drei Hauptblöcken

"SER/PAR", "PAR/SER" und "BASISBAND".

**"SER/PAR"**: verbraucht am meisten Energie, da dieser Block mit 499,2 MHz getaktet ist und intern aus 4 Parallelisierer für  $I^+$ ,  $I^-$ ,  $Q^-$  und  $Q^+$  besteht. Der Energieverbrauch ist nicht von den Eingangsdaten abhängig. *Power-gating* bzw. *clock-gating* können hier erheblich Strom einsparen.

**"PAR/SER"** ist nur mit 9% am gesamten Energieverbrauch beteiligt, obwohl dieser Block auch mit 499,2 MHz getaktet ist. Im Unterschied zu "SER/PAR" ist dieser Block wesentlich kleiner und einfacher aufgebaut. Hier ist der Energieverbrauch, insbesondere die *peak-power*, von der gewählten Datenrate abhängig. Je höher die Anzahl an ausgesendeten Impulse ist, desto höher ist der Energieverbrauch. Der Energieverbrauch lässt sich reduzieren, indem eine höhere Datenrate gewählt wird. Dieses Verhalten ist für eine optimierte Datenübertragung, wie im Kapitel 3 beschrieben, vorteilhaft.

**"BASISBAND"**: ist nur mit 31,2 MHz getaktet, aber aufgrund der größeren Komplexität ist dieser Block mit 34% der zweitgrößte Energieverbraucher.

**"FIFO"**: ist ein relativ großer Block. Der FIFO verhält sich beim Schreiben von Daten wie ein Schieberegister. Das bedeutet, dass bei einem Schreibvorgang alle Register den Wert des vorherigen Registers übernehmen. Dies ist der Grund, warum die meiste Energie beim Schreiben verbraucht wird. Schreibvorgänge finden im Sendemodus nur am Anfang statt, um die Nachricht in den FIFO zu schreiben. Im Empfangsmodus werden die empfangenen Daten nach und nach in den FIFO geschrieben.

Beim Lesen von Daten wird nur ein Adresszeiger verändert, aber die Inhalte der Register verändern sich nicht. Mögliche Maßnahmen zur Reduktion des Energieverbrauchs sind:

- FIFO durch einen Speicher (RAM) ersetzen.
- Die FIFO- oder Speicher-Größe auf 48 Byte begrenzen, was immer noch für den angestrebten Anwendungsfall ausreichend wäre, um eine komplette Nachricht zwischenspeichern.
- FIFO oder RAM verwerfen und nur einen *on-the-fly* Betrieb unterstützen. In diesem Fall ist ein schnelles Interface erforderlich.

**"SYNC"**: Hier wird der Energieverbrauch von den 16 parallel arbeitenden Korrelatoren verursacht. Auch hier könnte *power-gating* viel Energie einsparen, da dieser

Block nur für die Synchronisationsphase relevant ist.

**"TRANSMITTER"**: ist im Vergleich zu den anderen Blöcken relativ einfach aufgebaut, da hier nur die Modulation angewendet wird. Allerdings ist auch hier ein erhöhter Energieverbrauch bei 851 kb/s zu sehen. Auch hier ist eine schnellere Datenrate von Vorteil.

**"RS\_ENCODER" & "RS\_DECODER"** : Beide Blöcke sind relativ komplex, besonders der RS\_DECODER. Außerdem müssen die Daten aufwendig vorverarbeitet werden, damit diese von den Blöcken prozessiert werden können. Der Energieverbrauch ist nicht von der Datenrate abhängig. Diese Blöcke würden bei der optimierten Datenübertragung wegfallen.

**"INTERFACE"**: Es handelt sich hierbei um ein SPI-Interface, welches mit bis zu 33 MHz betrieben werden kann. Der Energieverbrauch hängt von der Nachrichtenlänge ab. *Power-gating* könnte auch hier eine erhebliche Energieersparnis bringen, da das Interface nur am Anfang und am Ende einer Datenübertragung benötigt wird.

#### 4.4.2 Messergebnisse

In diesem Abschnitt werden die implementierten Features mittels Messungen nachgewiesen. Der Messaufbau besteht aus einem Chip, welcher in einer Fassung platziert wurde, um diesen direkt mit einer Spannungsversorgung, einem Taktgenerator (31,2 MHz) und mit dem Computer zu verbinden. Die zu messenden Signale werden an einem Oszilloskop dargestellt und aufgenommen. Der Aufbau ist in Abbildung 4.26 dargestellt.

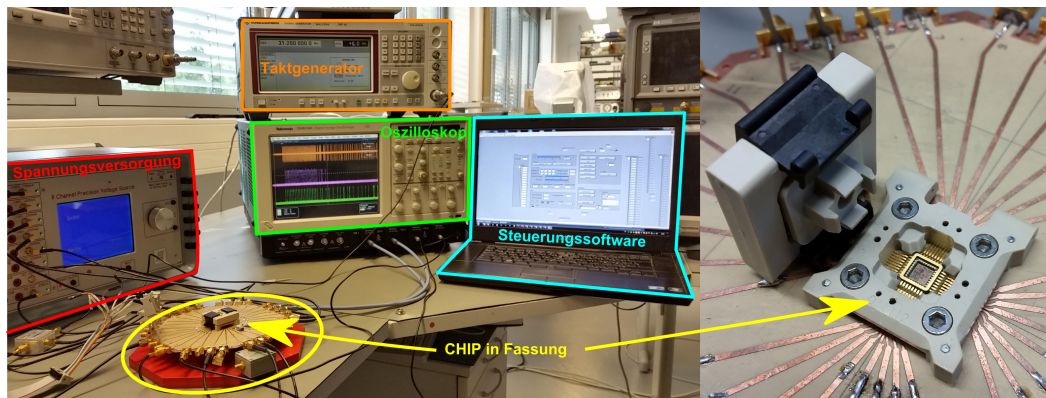
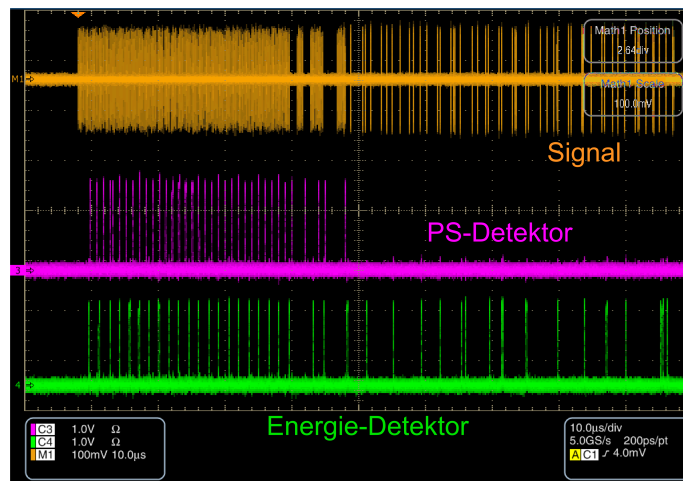


Abb. 4.26: Messaufbau für die Chip-Messung

Alle folgenden Abbildungen stellen die vom Oszilloskop aufgenommenen Signale dar. Die Signale wurden lediglich teilweise eingefärbt oder anders angeordnet. Dies ermöglicht es, die Signale besser zu vergleichen und die implementierten Features (siehe Seite 137) darzustellen.

#### 4.4.2.1 CCA Clear Channel Assessment

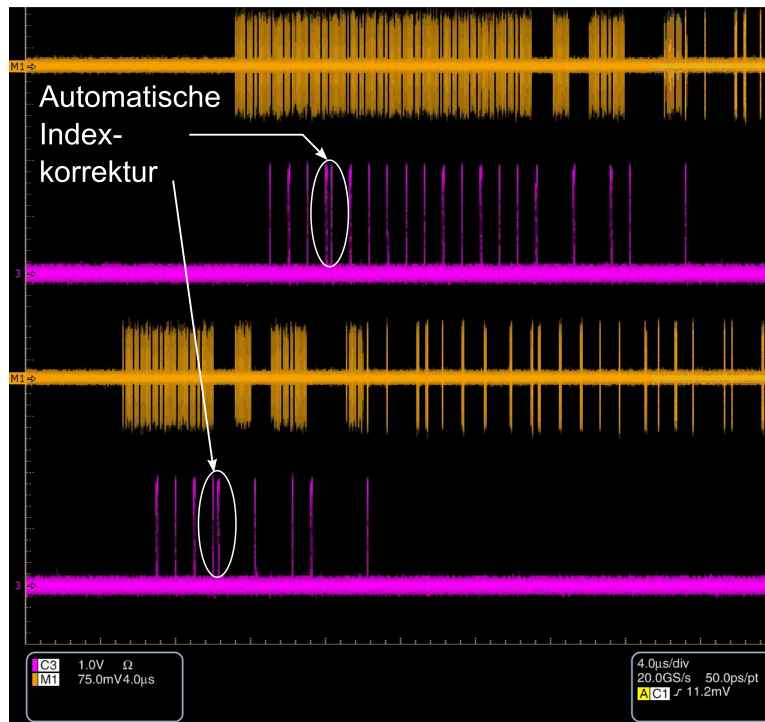
Dieses Feature wurde in Kapitel 4.3.3 beschrieben. In Abbildung 4.27 sind sowohl das Interrupt-Signal des PREAMBLE-Symbol-Detektors (lila) als auch das Interrupt-Signal des Energie-Detektors (grün) dargestellt. Das analysierte UWB-Signal (orange) besteht aus 32 PREAMBLE-Symbolen und nutzt eine Datenrate von 851 kb/s. Da in diesem Fall der Sender und Empfänger gleichzeitig aktiviert sind, beweist dies, dass beide parallel betrieben werden können.



**Abb. 4.27:** CCA: Das analysierte UWB-Signal ist orange markiert, das Interrupt-Signal des PS-Detektors ist lila markiert und das Interrupt-Signal des Energie-Detektors ist grün markiert.

#### 4.4.2.2 Synchronisation

Abbildung 4.28 zeigt einerseits, dass auch wenige PREAMBLE-Symbole ausreichen, um eine erfolgreiche Synchronisation zu erreichen. Andererseits ist hier die Indexkorrektur des Analyseintervalls gut zu erkennen. In beiden Fällen wird das letzte PREAMBLE-Symbol zum gleichen Zeitpunkt erfolgreich detektiert.



**Abb. 4.28:** Indexkorrektur während der Synchronisation

Im oberen orangen Signal besteht der PREAMBLE-Teil aus 16 PREAMBLE-Symbolen.

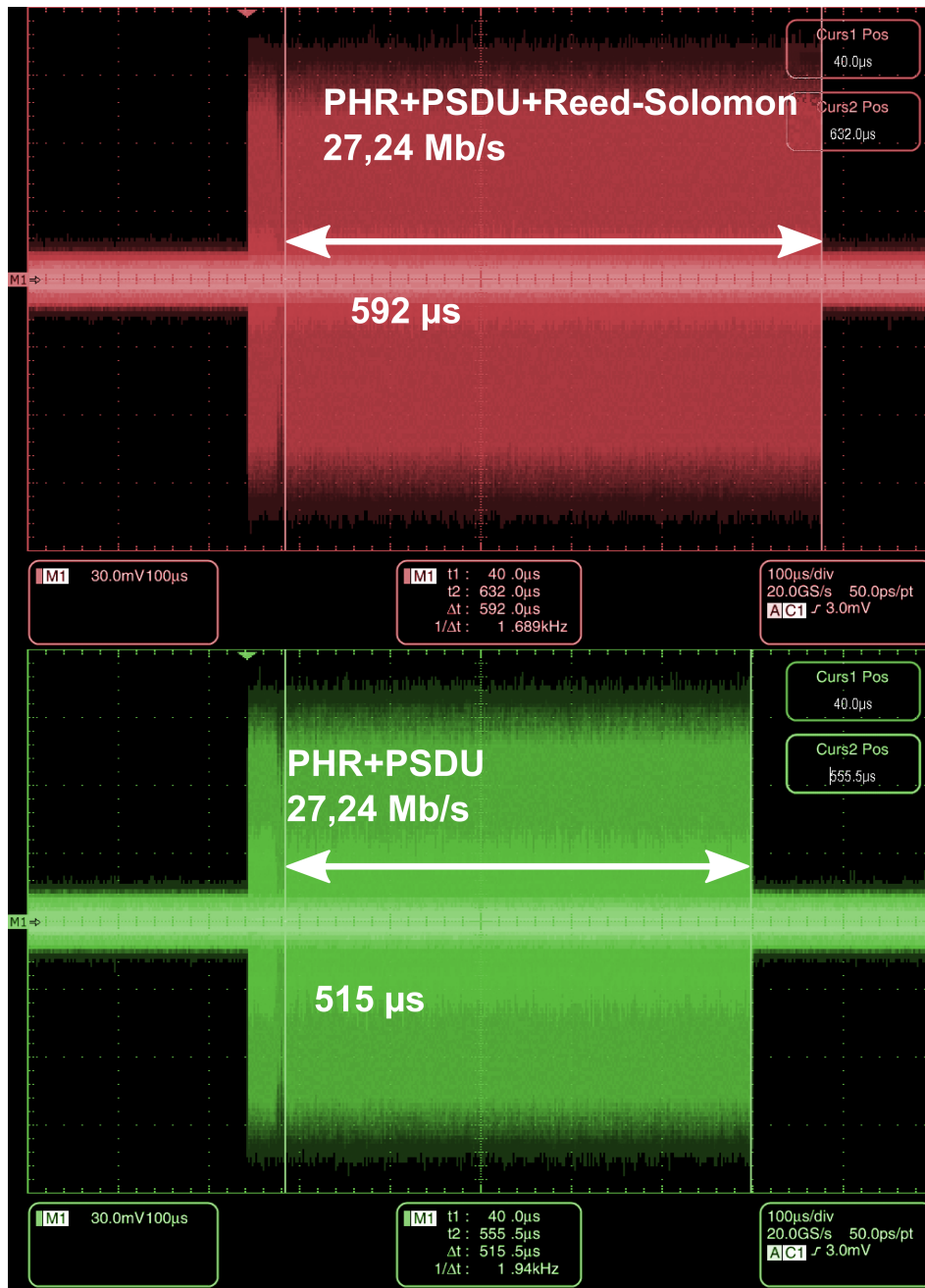
Im unteren orangen Signal besteht der PREAMBLE-Teil aus 4 PREAMBLE-Symbolen.

Für jedes UWB-Signal gibt es ein dazugehöriges lila markiertes Interrupt-Signal des PS-Detektors.



#### 4.4.2.3 Broadcast-Nachrichten

Abbildung 4.29 zeigt zwei UWB-Signale für eine Nachrichtenübertragung mit 1000 Byte Nutzdaten. Beide Signale weisen eine nicht standardkonforme kontinuierliche Datenrate von 27,24 Mb/s auf. Der Unterschied zwischen dem roten und dem grünen UWB-Signal ist der Reed-Solomon-Overhead. Der Reed-Solomon-Overhead macht einen Unterschied von  $77 \mu\text{s}$  aus.



**Abb. 4.29:** Broadcast-Nachricht mit einer Länge von 1000 Byte Nutzdaten mit Overhead des Reed-Solomon-Codes (rot markiert) und ohne Fehlerkorrektur (grün markiert).

Anhand dieser gemessenen Signale, lässt sich das Timing der Impulsgenerierung überprüfen. Ein zeitlicher Fehler wäre bei dieser sehr langen Nachricht gut zu erkennen, da sich ein Fehler aufsummieren würde. Für eine Übertragung von 1000 Byte Nutzdaten, berechnet sich die Datenübertragungszeit  $T_{DATA\ ohne\ RS}$  wie folgt:

$$\begin{aligned} T_{DATA\ ohne\ RS} &= T_{PHR} + T_{PSDU} \\ &= 39 \cdot \frac{32}{499,2MHz} + 1000 \cdot 8 \cdot \frac{32}{499,2MHz} \\ &\approx 2,5\ \mu s + 512,82\ \mu s \\ &\approx 515,32\ \mu s \end{aligned}$$

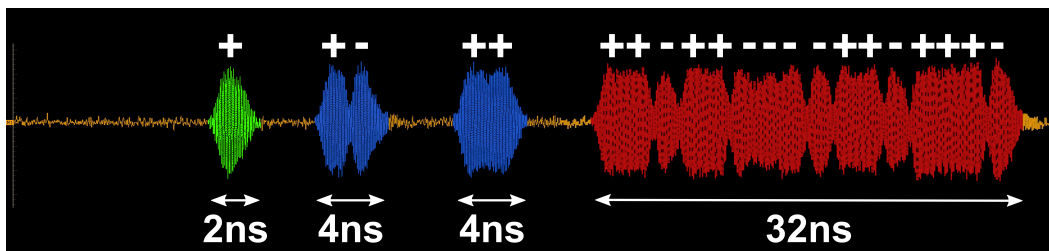
Wird zusätzlich der Reed-Solomon-Overhead berücksichtigt, so erhält man:

$$\begin{aligned} T_{DATA\ mit\ RS} &= T_{DATA\ ohne\ RS} + T_{RS-Overhead} = 515,32\ \mu s + \left\lceil \frac{1000}{41} \right\rceil \cdot 48 \cdot \frac{32}{499,2MHz} \\ &= 515,32\ \mu s + 25 \cdot 48 \cdot \frac{32}{499,2MHz} \\ &\approx 515,32\ \mu s + 76,92\ \mu s \\ &\approx 592,24\ \mu s \end{aligned}$$

Da diese berechneten Werte von  $515,32\ \mu s$  und  $592,24\ \mu s$  mit den gemessenen Werten aus Abbildung 4.29 Überstimmen, bestätigt dies die korrekte Funktionalität.

#### 4.4.2.4 Impulsgenerierung

Abbildung 4.30 zeigt anhand einiger Beispiele wie die generierten Impulse bzw. Bursts für jede unterstützte Datenrate aussehen. Die Impulse wurden je nach verwendeter Datenrate eingefärbt.



**Abb. 4.30:** Gemessene Bursts für alle Datenraten.  
 rot markiert: für 851 kb/s bzw. 16 Impulse je Burst  
 blau markiert: für 6,81 Mb/s bzw. 2 Impulse je Burst  
 grün markiert: für 27,24 MB/s bzw. 1 Impuls je Burst

#### 4.4.2.5 Unterstützte Datenraten

Es wurden insgesamt 5 verschiedene Übertragungsmodi implementiert. Abbildung 4.31 stellt alle 5 Modi dar. Die Signal-Abschnitte wurden je nach verwendeter Datenrate entsprechend eingefärbt. Die verschiedenen Signale wurden vertikal angeordnet,

um diese besser vergleichen zu können. Es handelt sich bei jeder Übertragung um die gleiche Nachricht <sup>7</sup> mit der Länge von 17 Byte. Es ist erkennbar, wie sich die zeitlichen Verhältnisse zwischen der Synchronisation und der Datenübertragung umkehren. Im obersten Signal ist die Datenübertragung maßgeblich verantwortlich für die Übertragungszeit. Im untersten Signal ist die Synchronisation maßgeblich verantwortlich für die Übertragungszeit.

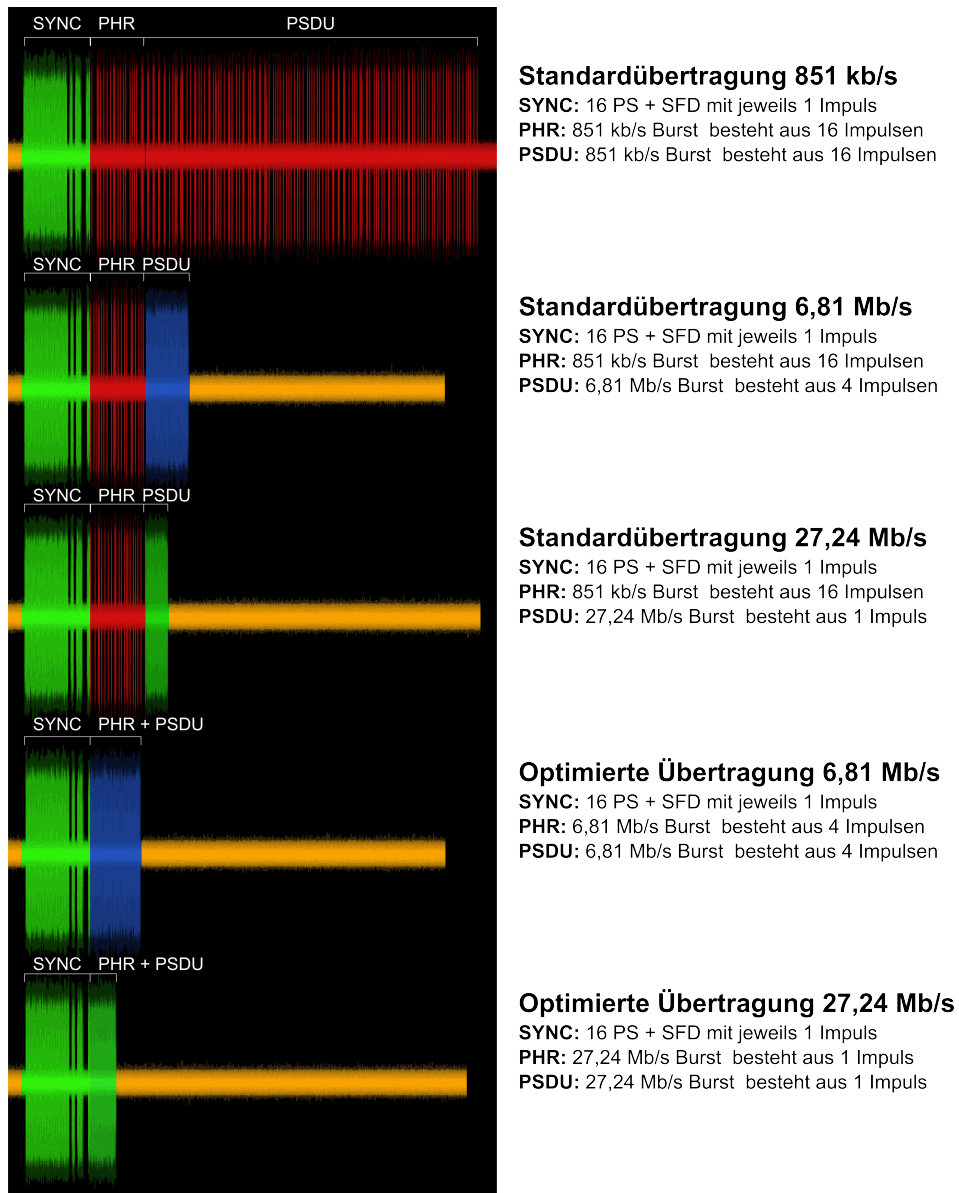


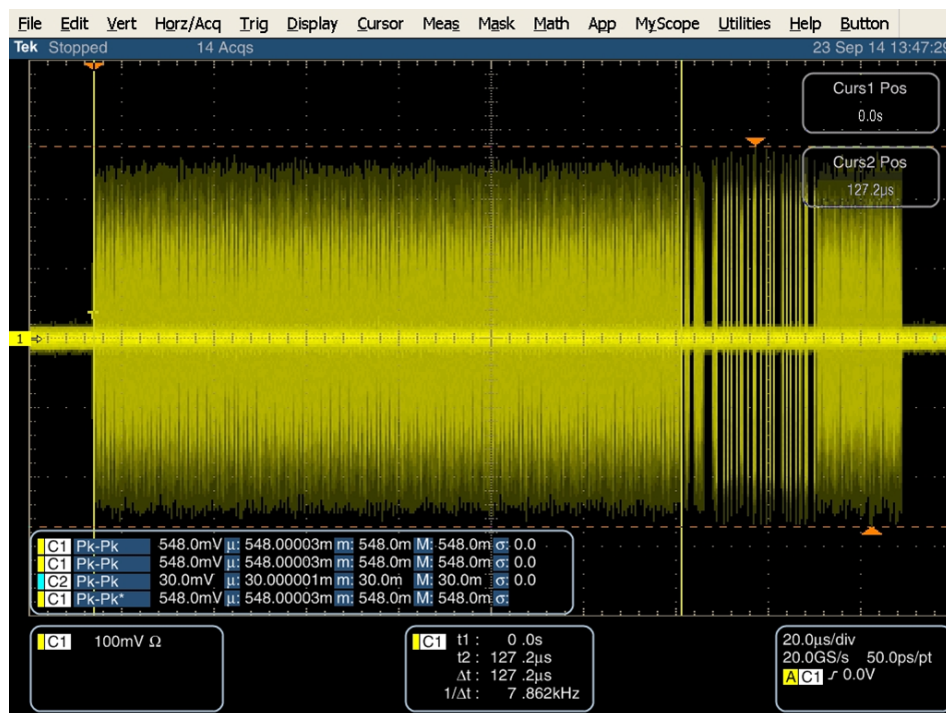
Abb. 4.31: Alle unterstützten Übertragungsmodi des gefertigten ASICs.

<sup>7</sup>Es handelt sich hierbei um die Beispielnachricht aus dem Standard IEEE 802.15.4a angegeben ist.

### 4.4.3 Vergleich mit einem anderen UWB-Anbieter

Um die Qualität des vom ASIC generierten IR-UWB-Signals besser zu beurteilen, wurden die gesendeten Signale mit dem eines gekauften IR-UWB-Entwicklungsmoduls (ScenSor EVK1000 Evaluation Kit [87]) von der Firma DecaWave [71] verglichen. Dieses Modul wurde 2013 veröffentlicht und ist in der Lage, mittels two-way-ranging eine Abstandsmessung mit einer Genauigkeit von  $\pm 10$  cm durchzuführen. Um *two-way-ranging*<sup>8</sup> zu betreiben, müssen die IR-UWB-Module untereinander Daten austauschen. Dies wird mittels standardkonformer Datenframes realisiert. Diese Frames wurden mit einem Oszilloskop aufgenommen und die Ergebnisse sind in den folgenden Abbildungen dargestellt. Das IR-UWB-Modul von DecaWave unterstützt die Datenraten 6,81 Mb/s und 110 kb/s. Da letztere nicht in dieser Arbeit berücksichtigt wurde, wird nur die Datenübertragung mit 6,81 Mb/s betrachtet. Außerdem wurden einige nicht standardkonforme Features hinzugefügt. Es werden zusätzliche Anzahlen an PREAMBLE-Symbolen im PREAMBLE-Teil unterstützt (128, 512 und 1536), sowie ein modifizierter SFD-Teil. Beide Features dienen dazu, die Zuverlässigkeit der Synchronisationserkennung zu verbessern.

Abbildungen 4.32 bis 4.37, stellen die erzeugten Signale der DecaWave-Module dar.



**Abb. 4.32:** DecaWave-Modul: IR-UWB Frame mit 6,81 Mb/s:  
Alle 4 Teile des UWB-Frames PREAMBLE, SFD, PHR und PSDU sind gut zu erkennen.

<sup>8</sup>Dies ist eine Methode, bei der zwei Geräte Zeitinformationen austauschen, mit welcher die Laufzeit (engl. *Time-Of-Flight*, TOF) des Signals und somit auch der Abstand zwischen den beiden Geräten bestimmt werden kann.

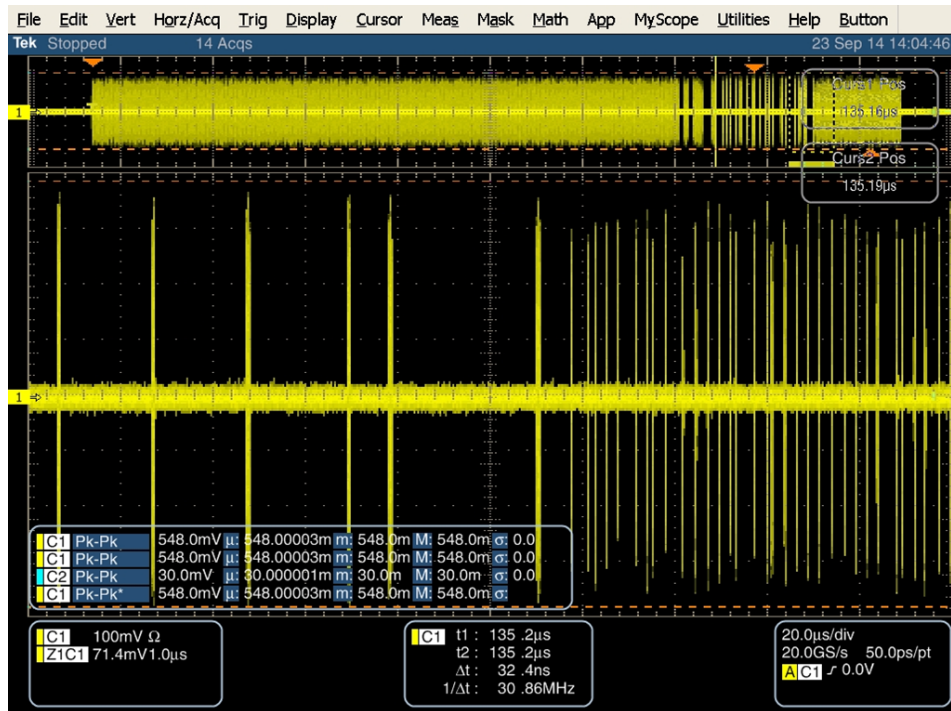


Abb. 4.33: DecaWave-Modul: Ausschnitt aus einem IR-UWB Frame mit 6,81 Mb/s: Übergang vom PHR-Teil (851 kb/s) zum PSDU-Teil (6,81 Mb/s)

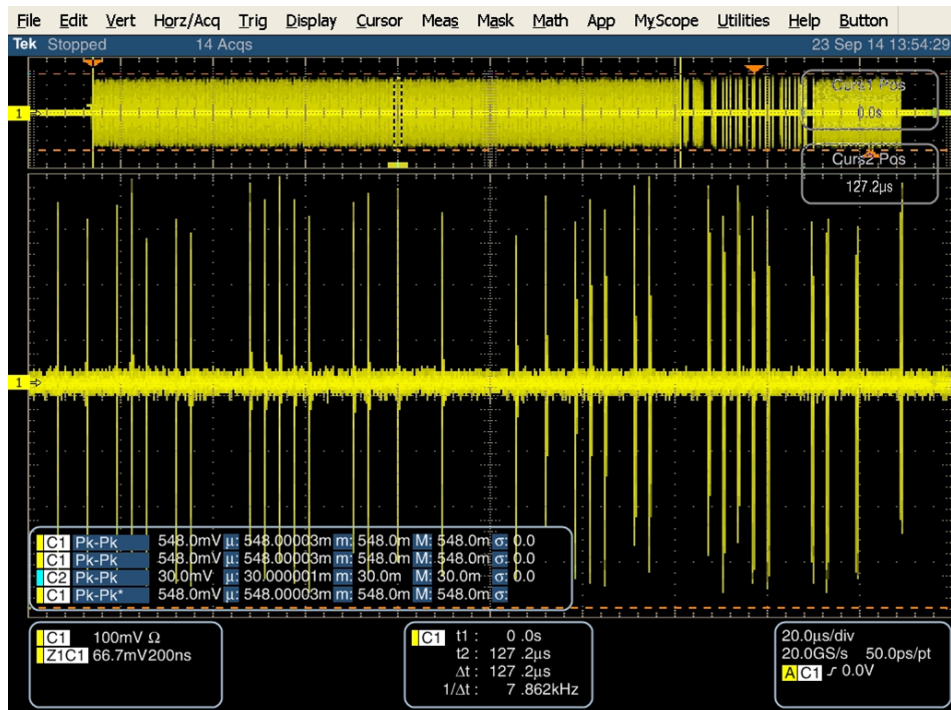


Abb. 4.34: DecaWave-Modul: Ausschnitt aus einem IR-UWB Frame: Vergrößerte Darstellung eines PREAMBLE-Symbols

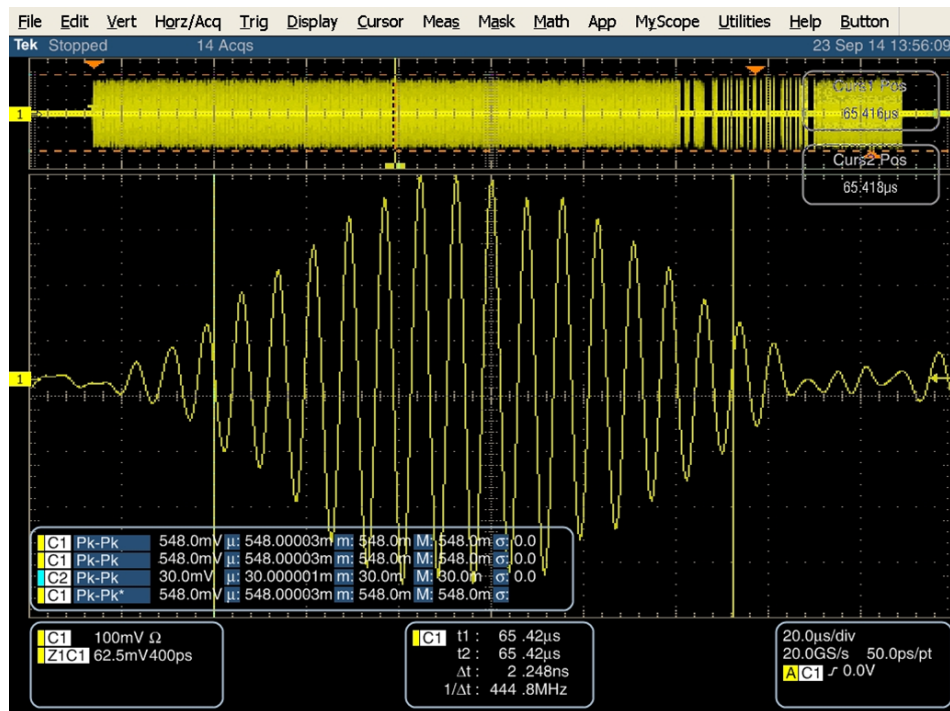


Abb. 4.35: DecaWave-Modul: Ausschnitt aus einem IR-UWB Frame: Ein einzelner Impuls, wie er bei der Synchronisation oder bei einer Datenrate von 27,24 Mb/s vorkommt.

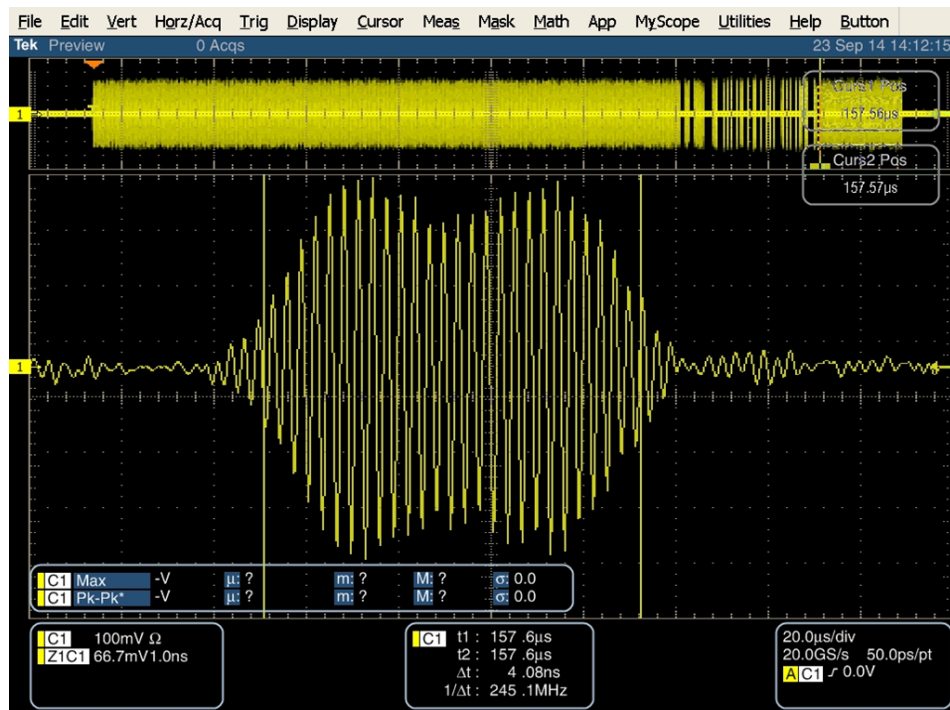
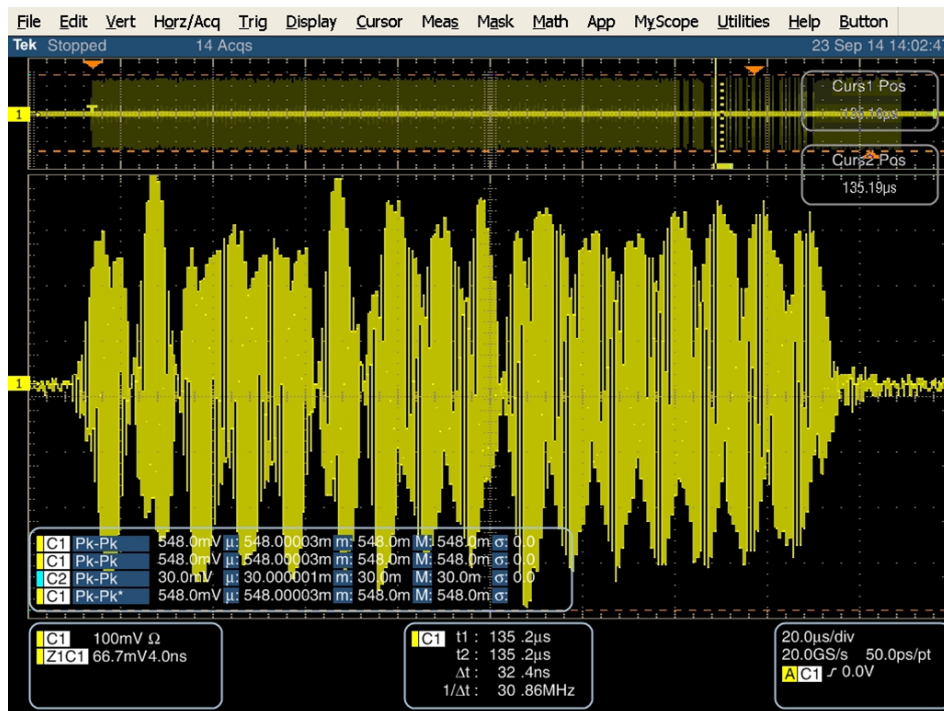


Abb. 4.36: DecaWave-Modul: Ausschnitt aus einem IR-UWB Frame: Ein Burst, bestehend aus zwei Impulsen, wie er bei einer Datenrate von 6,81 Mb/s vorkommt.



**Abb. 4.37:** DecaWave-Modul: Ausschnitt aus einem IR-UWB Frame:  
Ein Burst, bestehend aus 16 Impulsen, wie er bei einer Datenrate von 851 kb/s vorkommt.

Die Signale von den DecaWave-Modulen sind ähnlich den Signalen des in dieser Arbeit genutzten Frontends. Der einzige signifikante Unterschied ist die Ausgangsleistung. Bei DecaWave erreicht das Ausgangssignal eine Amplitude von 548 mV Peak-to-Peak und bei dem hier vorgestellten Chip sind es 200 mV Peak-to-Peak. Die erreichbare Reichweite liegt bei DecaWave bei rund 10 m bei 6,81 Mb/s und einer Frequenz von 6,489 GHz. Da nicht genau bekannt ist, wie der Empfänger implementiert wurde, ist ein tiefergehender Vergleich nicht möglich. Allerdings konnte bei dem Chip aus dem TANDEM Projekt, welcher einen ähnlichen Sender enthält, eine Reichweite von bis zu 15 m bei einer Ausgangsleistung von 260 mV erreicht werden.

## 4.5 FPGA Implementierung

Der zuvor in Kapitel 4.4 beschriebene ASIC wurde auch auf einem FPGA Virtex5 von AVNET [75] implementiert, siehe Abbildung 4.38. Das Basisband wurde generisch in VHDL implementiert, so dass keine Veränderungen notwendig waren, um es auf dem FPGA zu testen. Allerdings wurden die Blöcke "SER/PAR" und "PAR/SER" speziell für die IHP-Technologie entwickelt und optimiert und können somit nicht für den FPGA wiederverwendet werden. Diese Blöcke sind aber für die hier beschriebenen Zuverlässigkeitstests nicht relevant. Da der VHDL-Code des ASIC voll kompatibel zum FPGA ist, kann der FPGA taktgenau das Verhalten des ASICs reproduzieren.



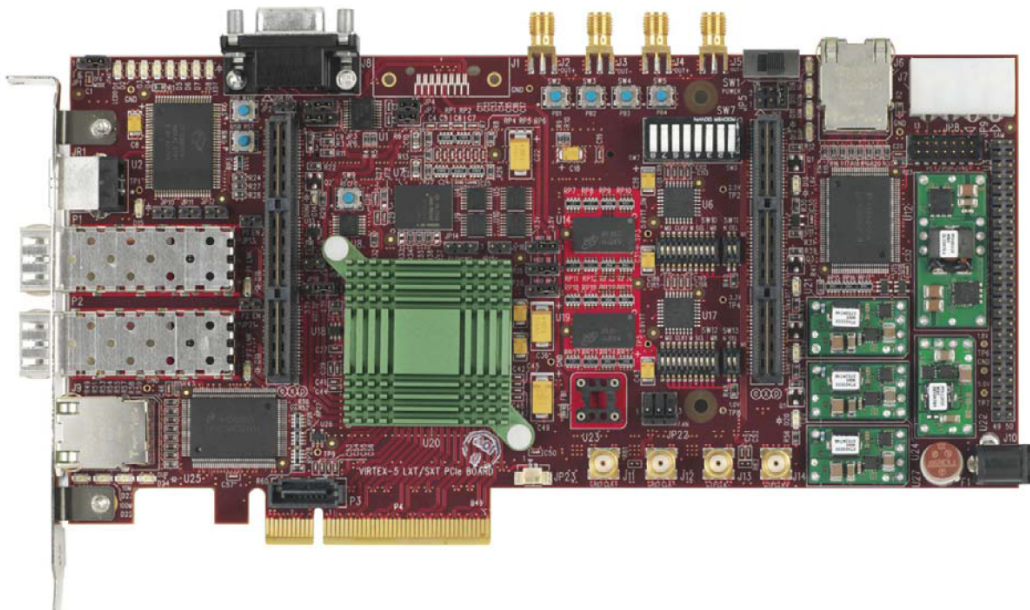


Abb. 4.38: Xilinx Virtex-5 FXT PCIe Development Kit

### 4.5.1 Zuverlässigkeitstests

Es wurden zwei identische Basisbänder in den FPGA integriert. Ein Basisband generiert und empfängt das Originalsignal und das andere Basisband empfängt ein gestörtes Signal. Die Störung erfolgt mittels einer Komponente, genannt Interferer, siehe Abbildung 4.39. Um die Testparameter verändern zu können, wurde eine GUI in Labview programmiert. Mit dieser GUI können alle möglichen Konfigurationen eingestellt werden. Beispielsweise kann die Datenrate verändert werden, der Viterbi-Dekoder und/oder der Reed-Solomon-Dekoder können unabhängig voneinander aktiviert/deaktiviert werden. Die Fehlerrate kann verändert werden. Es können zufällige Fehler, Burst-Fehler, ein schwaches Signal oder ein stark verrauschtes Signal generiert werden. Die empfangenen Daten beider Basisbänder werden verglichen und können somit genau analysiert werden.

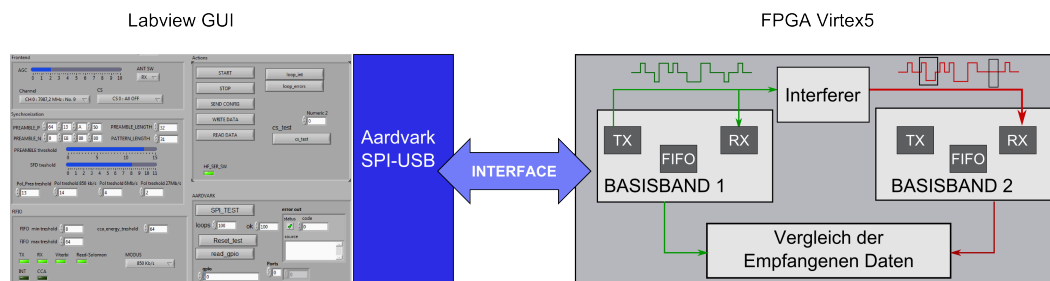
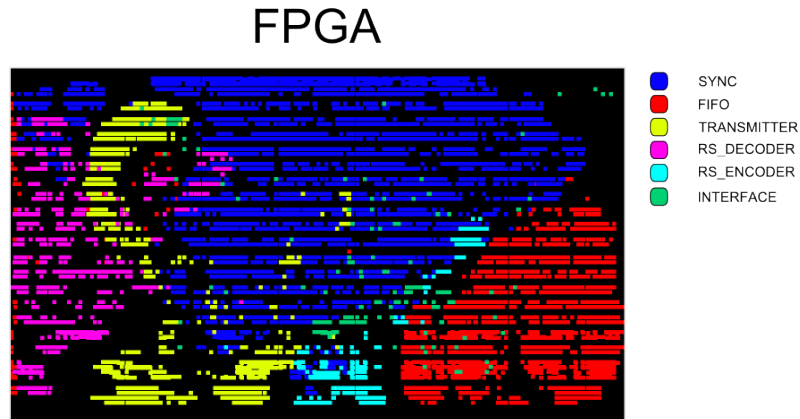


Abb. 4.39: FPGA Test

### 4.5.2 Ressourcenverbrauch

Der gesamte Energieverbrauch des Designs für ein Basisband beläuft sich auf 1.565 W, darauf entfallen 86 mW auf die *switching-power* und 1.479 W auf die *leakage-power*.

Die FPGA-Auslastung liegt bei 35,5 %. Abbildung 4.40 stellt den Ressourcenverbrauch der einzelnen Komponenten grafisch dar.



**Abb. 4.40:** FPGA Ressourcenverbrauch

In Tabelle 4.7 ist die Ausnutzung der verschiedenen FPGA-Komponenten genauer angegeben. Die Bedeutung der angegebenen Parameter ist wie folgt:

#FF: Anzahl an *Flip-Flops*

#LUT: Anzahl an *Look-up Tables*

#SRL: Anzahl an *Shift-Register LUT*

#DSP: Anzahl an *Digital Signal Processor*

#CARRY4: Anzahl an *Fast Carry Logic mit Look Ahead*

**Tabelle 4.7:** FPGA Ressourcenverbrauch

	#FFs	#LUTs	#SRLs	#DSPs	#CARRY4s	%
<i>FPGA</i> verfügbar	58880	58880	24320	640	—	—
<i>BASISBAND</i>	6786	22564	58	8	662	100 %
<i>SYNC</i>	3015	10151	0	0	3	43,78 %
<i>FIFO</i>	1518	4670	0	0	22	20,65 %
<i>TRANSMITTER</i>	288	3611	0	0	286	13,91 %
<i>RS_DECODER</i>	861	2062	58	6	145	10,41 %
<i>RS_ENCODER</i>	338	700	0	2	76	3,71 %
<i>INTERFACE</i>	326	268	0	0	20	2,04 %
<i>REST</i>	440	1102	0	0	110	5,5 %

Die Aufteilung der einzelnen Komponenten entspricht ungefähr der Aufteilung im ASIC. Nur der "TRANSMITTER" und der "RS\_DECODER" weichen stark ab.

Im ASIC liegt der Anteil des "TRANSMITTER" bei 6,89% und der "RS\_DECODER" liegt bei 23,29 %. Die Ursache liegt darin, dass der FPGA manche VHDL-Strukturen besser optimieren kann als der ASIC und umgekehrt.



# Zusammenfassung und Ausblick

---

## Inhaltsverzeichnis

5.1	Zusammenfassung . . . . .	159
5.2	Ausblick . . . . .	161

---

In diesem Kapitel werden die erzielten Ergebnisse dieser Arbeit zusammengefasst und weiterführende Forschungsrichtungen erörtert.

### 5.1 Zusammenfassung

In dieser Arbeit wurden zuerst die besonderen Anforderungen und Probleme von drahtlosen Sensornetzwerken im industriellen Umfeld und insbesondere für die Automatisierungsindustrie ermittelt und dargestellt. Die wichtigste Anforderung war eine Zykluszeit unter 1 ms für den Worst Case, d.h ein Netzwerk mit einem Master und 127 Knoten mit einer bidirektionalen Kommunikation und einer maximalen Nutzdatenmenge von bis zu 48 Byte je Knoten.

Des Weiteren gab es strenge Anforderungen an die Zuverlässigkeit. Es muss gewährleistet werden, dass alle Nachrichten korrekt übermittelt werden. Die Wahrscheinlichkeit, dass eine Übertragung 3 mal in Folge fehlschlägt, darf nicht höher als  $10^{-9}$  sein.

Bei all diesen Optimierungen wurden die Beschränkungen seitens der UWB-Regulierungen insbesondere die Impulswiederholfrequenz (PRF) berücksichtigt und eingehalten. Dies war eine der größten Herausforderung in dieser Doktorarbeit. Eine weitere Herausforderung war es, Synchronisations- und Datenübertragungsverfahren zu entwickeln, die eine variable Anzahl an Sensorknoten unterstützten.

Es wurde gezeigt, dass es theoretisch und technisch möglich ist die Anforderungen zu erfüllen. Allerdings ist es mit dem aktuellen UWB-Standard IEEE.802.15.4a und den damit einhergehenden Restriktionen nicht möglich die hohen Anforderungen zu erfüllen.

Im Rahmen dieser Arbeit wurde eine schrittweise Optimierung durchgeführt, um möglichst geringe Abweichungen zum UWB-Standard gewährleisten zu können.

Als Ausgangspunkt für die Optimierungen wurde untersucht, welche kürzesten Übertragungszeiten mit dem vorhandenen UWB-Standard erreicht werden können. Eine Punkt-zu-Punkt Kommunikation benötigt im Worst case  $80,4 \mu s$ . Daraus folgt wiederum eine Zykluszeit von 20,43 ms. Im Vergleich zu der geforderten Zykluszeit von 1 ms, ist diese 20 mal höher.

Im ersten Optimierungsschritt wurden, alle Teile eines UWB-Frames einzeln optimiert. Diese Optimierung konnte die Anforderungen für eine Punkt-zu-Punkt Kommunikation von unter  $50 \mu s$  erfüllen, aber war nicht ausreichend, um die Anforderung für eine Punkt-zu-Multipunkt Kommunikation zu erfüllen. Denn diese konnte in diesem Schritt nur auf  $8,82 \text{ ms}$  gesenkt werden.

Für die Punkt-zu-Multipunkt Kommunikation wurde die Möglichkeit der Nutzung von Broadcastnachrichten untersucht und implementiert. Damit konnte die reine Übertragungszeit auf  $6,21 \text{ ms}$  reduziert werden.

Daraufhin wurde eine Vollduplex-Kommunikation zwischen zwei Knoten ausgearbeitet. Hierfür war es nötig, ein neues Synchronisationsverfahren zu entwickeln und optimierte PREAMBLE-Symbole zu verwenden. Damit wurde eine Zykluszeit von  $4,82 \text{ ms}$  (mit *hopping-positions*) und von  $1,71 \text{ ms}$  (ohne *hopping-positions*) erreicht. Um die Übertragungszeit weiter reduzieren zu können, wurden basierend auf diesen Ergebnissen sowohl die Datenübertragung als auch die Synchronisierung für eine variable Anzahl an gleichzeitig sendenden Knoten erweitert. Bei der parallelen Synchronisation mit mehreren Knoten gleichzeitig wurde darauf geachtet, dass mit höherer Anzahl an genutzter Knoten die gesamte Synchronisationszeit im Vergleich zur gesamten sequentiellen Synchronisationszeit abnimmt.

Mit diesen Optimierungen konnten die zeitlichen Anforderungen der Automatisierungsindustrie von unter  $1 \text{ ms}$  erfüllt werden. Je nach Anzahl verwendeter PREAMBLE-Symbole und deren Länge, konnten Zykluszeiten von  $0.6 \text{ ms}$  erreicht werden. Allerdings wurde hierbei die Zuverlässigkeit noch nicht berücksichtigt.

Die Zuverlässigkeitsanforderungen konnten mittels eines speziell für ternäre Impulse angepassten Fehlerkorrekturverfahrens und Mehrfachübertragungen erfüllt werden. Der hierfür entwickelte Umwandlungsalgorithmus, welcher eine Impulssequenz mit einer konstanten Anzahl an Impulsen in eine Bitsequenz überführt und umgekehrt, ist eine der herausragenden Innovationen dieser Arbeit. Durch Erhöhung der Zuverlässigkeit, erhöhte sich auch die zuvor optimierte Zykluszeit auf über  $0,7 \text{ ms}$ . Es wurde gezeigt, dass die hohen Anforderungen der Automatisierungsindustrie bzgl. Zuverlässigkeit und Zykluszeit trotzdem erreicht werden können, wenn bestimmte Frame-Error-Rates nicht überschritten werden. Die Zykluszeit wurde im Vergleich zum UWB-Standard um mehr als  $95\%$  reduziert.

Es wurden weitere Optimierungen der Hardware-Implementierung bzgl. Energieverbrauch, Chip-Fläche und Komplexität untersucht. Einige dieser Optimierungen wie Serialisierer, Parallelisierer und optimierte Synchronisation wurden in dieser Arbeit implementiert. Es hat sich gezeigt, dass ein modularer Aufbau und schnelle Umschaltzeiten den Energieverbrauch stark senken können. Addiert man die aktiven Sende- und Empfangs-Zeiten während eines kompletten Zykluses für den Master und alle  $127$  Slaves, so beträgt dieser Wert bei einer Übertragung entsprechend dem UWB-Standard  $38,08 \text{ ms}$  und bei Verwendung des in dieser Arbeit entwickelten optimierten Übertragungsverfahrens sind es  $13,23 \text{ ms}$ . Dies entspricht einem um  $65\%$  reduzierten Energieverbrauch.

Der erste Chip-Entwurf, der als Ausgangspunkt für diese Arbeit diente, unterstützte nur wenige Features. Es wurde nur eine Datenrate von 851 kb/s unterstützt und ein kohärenter Empfang war nicht möglich. Das Basisband benötigte  $5,52 \text{ mm}^2$  Fläche.

Der in dieser Arbeit entwickelte Chip bietet deutlich mehr Funktionalitäten, die wichtigsten sind:

- Senden und Empfangen von Broadcast-Nachrichten
- Kohärenter Empfang mit einer Genauigkeit von 2 ns
- Unterstützung von mehreren Datenraten sowohl für den PSDU-Teil als auch für den PHR-Teil
- Variierbare PREAMBLE-Symbole
- Variierbare Anzahl an PREAMBLE-Symbolen

Obwohl dieser Chip viel mehr Features aufweist, konnte die Fläche aufgrund der in dieser Arbeit beschriebenen Optimierungen auf  $3,13 \text{ mm}^2$  reduziert werden. Dies entspricht einer Reduzierung im Vergleich zum ersten Chip um 43%.

Bei dem hier verwendeten optimierten Synchronisationsverfahren werden auch gleichzeitig Abstandsinformationen ausgetauscht, welche beispielsweise zur Verbesserung der Datensicherheit genutzt werden können.

In dieser Doktorarbeit wurde gezeigt, dass es möglich ist mit vielen „kleinen“ Optimierungen und unter Berücksichtigung der UWB-Regulierungen die Anforderungen der Automatisierungsindustrie zu erfüllen.

## 5.2 Ausblick

Aufbauend auf den Ergebnissen dieser Arbeit können weitere Aspekte untersucht und implementiert werden.

Die in dieser Arbeit beschriebenen Übertragungsverfahren zur Unterstützung paralleler Datenübertragung mehrerer Sensorknoten gleichzeitig, können im Rahmen eines weiteren Projektes implementiert und weiterentwickelt werden. Die Datenrate lässt sich steigern, indem noch größere Impulsverteilungsintervalle zugelassen werden. Bei größeren Impulsverteilungsintervallen kann, aufgrund von größeren Lücken während der Übertragungsphase, mehr Energie eingespart werden. Der in dieser Arbeit entwickelte Chip bietet eine gute Grundlage dafür, da dieser schon einige Voraussetzungen für solch eine Implementierung bereitstellt, wie z.B. gleichzeitiger Betrieb von Sender und Empfänger.

Es wurde gezeigt, dass schnelle Umschaltzeiten zwischen Sender und Empfänger im analogen Frontend sehr viel zur Energieeinsparung beitragen können. Es wäre daher sinnvoll, in dieser Richtung zu forschen. In diesem Fall müssen aber Änderungen sowohl im Basisband als auch im analogen Frontend durchgeführt werden. Der Schwerpunkt liegt hierbei im analogen Frontend, da viele analoge Komponenten eine gewisse Anlaufzeit brauchen, um funktionsbereit zu sein. Es muss auch darauf geachtet werden, dass die Umschaltzeiten durch die Geschwindigkeit des digitalen Basisbands begrenzt werden. Für den in dieser Arbeit entwickelten Chip liegt die Begrenzung der Umschaltgeschwindigkeit bei 6 ns.

Eine weitere Forschungsrichtung sind Fehlerkorrekturverfahren, die die Fehler auf Impulsebene korrigieren können. Auch die Kombination von mehreren Fehlerkorrekturverfahren kann Vorteile bringen. In Kapitel 3 wurde z.B. gezeigt, dass eine intelligente Kombination von Hamming-Code und Reed-Solomon-Code sowohl den Overhead als auch die Verarbeitungszeit reduzieren kann. Eventuell können auch weitere FEC-Verfahren wie BCH-Codes oder Faltungscodes oder auch andere Hamming-Codes mit ARQ-Verfahren kombiniert werden, um die Zuverlässigkeit weiter zu erhöhen und um die Anzahl an Mehrfachübertragungen zu senken.

In dieser Arbeit waren die Beschränkungen bzgl. der Impulswiederholrate von maximal 15,6 MHz sehr schwer zu erfüllen. Aus dieser Arbeit geht hervor, dass eine auch nur leicht höhere Impulswiederholrate (z.B. 16,1 MHz) eine wesentlich höhere Datenrate und Zuverlässigkeit ermöglichen würde. Es müsste erforscht werden, wie sich höhere Impulswiederholungsraten auf die Koexistenz mit anderen Funksystemen auswirken und ob die UWB-Regulierungen verletzt werden.

Ein weiteres wichtiges Anwendungsfeld von UWB ist Ranging. Aufgrund der Möglichkeit, während der Synchronisation auch Abstandsinformation auszutauschen, wäre eine Verbesserung oder eine Erweiterung dieser Fähigkeit sehr nützlich. Ein System, welches mittels UWB hohe Datenraten und präzise Abstandsmessungen ermöglicht, gibt es noch nicht. Wenn die Positionen der Netzwerkteilnehmer bekannt sind, kann mittels Abstandsinformationen die Sicherheit erhöht werden. Wenn zusätzlich gerichtete Antennen verwendet werden, ist auch eine Lokalisierung möglich und es kann ausgeschlossen werden, dass ein Knoten gleichzeitig mehrere Identitäten annimmt.



# Anhang A

---

## Inhaltsverzeichnis

---

A.1	Eingesetzte Funkstandards in der Automatisierungsindustrie . . . . .	<b>163</b>
A.2	UWB Forschungsprojekte im detail . . . . .	<b>166</b>
A.3	Ermittlung der maximalen Datenrate . . . . .	<b>177</b>
A.4	Berechnung optimaler Impulssequenzen für PREAMBLE-Symbole .	<b>178</b>
A.5	Optimale Preamble-Sequenzen . . . . .	<b>181</b>
A.6	Optimale Bitraten . . . . .	<b>191</b>

---

## A.1 Eingesetzte Funkstandards in der Automatisierungsindustrie

In diesem Abschnitt werden die wichtigsten Funktechnologien für die Automatisierung kurz dargestellt. Bisher wurde kein einziger Standard explizit für die Automatisierungsindustrie entwickelt. Alle derzeit angewandte Standards bzw. Protokolle basieren oder sind eine Erweiterung von existierenden Standards, welche nicht vorrangig für den Einsatz in der Automatisierung konzipiert wurden.

### A.1.0.1 IEEE 802.15.4

Wesentliche Entwicklungsziele dieses Funkstandards IEEE 802.15.4 [88] waren eine geringe Leistungsaufnahme für einen langen Betrieb über Batterieversorgung, kostengünstige Hardware, sichere Übertragung, Nutzung der lizenzfreien ISM-Bänder. Außerdem wurde darauf geachtet andere Protokolle wie WLAN und BLUETOOTH, welche im selben Frequenzbereich operieren, nicht zu beeinträchtigen. Aufgrund dieser Eigenschaften eignet sich der Standard IEEE 802.15.4 vor allem für drahtlose Sensornetze und für direkt am Körper getragene Sensoren und Aktoren WBAN (engl. Wireless Body Area Network). Mehrere Protokolle basieren auf diesem Standard [89] [90]:

- **ZigBee** [91] ist ein Industriestandard für günstige und sparsame drahtlose Netzwerke. ZigBee wird hauptsächlich für Energieverwaltung, Gesundheit, Haus und Heimautomatisierung verwendet. Das ZigBee Protokoll operiert im

2,4 GHz Frequenzband mit einer Datenrate von 250kb/s. ZigBee-Module können eine 128 bit AES (engl. Advanced Encryption Standard) Verschlüsselung nutzen.

- **6LoWPAN** (engl. IPv6 over Low power Wireless Personal Area Networks) [92] ist ein Protokoll mit ähnlichen Anwendungsgebieten und Eigenschaften wie ZigBee. Im Unterschied zu ZigBee kann dieses Protokoll auf IP-Ebene mit WLAN und Ethernet interagieren. Außerdem verwendet es Header-Kompressionsverfahren um die Effizienz zu steigern.
- **Wireless HART** [93] ist ein industrieller Standard für Prozessüberwachung und Prozesssteuerung und basiert auf IEEE 802.15.4-2006. Er nutzt das freie 2,4 GHz Frequenzband. In der Industrie wird dieser Standard für WSN (engl. Wireless Sensor Actuator Network) in der Prozessautomatisierung (engl. Process Automation, PA) genutzt. Daher werden diese Netzwerke auch oft als WSN-PA [94] bezeichnet.
- **ISA100** [95] ist ein Protokoll, welches wie Wireless Hart auf IEEE 802.15.4-2006 basiert und das 2,4 GHz Frequenzband nutzt. Im Gegensatz zu WirelessHART bietet ISA 100 mehr Flexibilität indem es mit mehreren existierenden Standards im Automatisierungsumfeld (HART [96], Profibus [97], Foundation Fieldbus [98] und Device Net [99]) kompatibel ist.
- **WiMi Mesh** [100] und **WiMi P2P** [101] basieren auf IEEE 802.15.4-2003 und nutzen beide das Frequenzband von 2,45 GHz. Es sind proprietäre Protokolle von der Firma Microchip [102]. Diese Protokolle laufen nur auf den von der Firma Microchip hergestellten PIC-Mikroprozessoren. Die Protokolle sind wesentlich einfacher als ZigBee und benötigen 3-17 kB. Im Gegensatz dazu benötigt ZigBee 40-100 kB.
- **SimpliTI** [103] ist ein proprietäres Protokoll von Texas Instruments und operiert bei 433 MHz, 868 MHz und 915 MHz mit einer maximalen Datenrate von 300 kb/s und einer maximalen Reichweite von 100 Metern.

#### A.1.0.2 KNX-RF

KNX-RF (engl. Konnex Radio Frequency) [104] ist ein Kommunikationssystem für Heim und Gebäudeautomatisierung welches auf dem internationalen Standard KNX (ISO/IEC14543-3) [105] basiert. Die genutzte Frequenz ist 868,3 MHz. Die maximale Datenrate von 16.384 kb/s wird mittels einer Frequenzumtastung (engl. Frequency Shift Keying, FSK) erreicht. Die Übertragungsleistung liegt im Bereich von 1 bis 25 mW. KNX-Knoten können mit zusätzlichen Steckern ausgestattet werden, um die Kommunikation mit anderen Knoten außer Reichweite zu ermöglichen [106]. Der KNX Standard ist nicht lizenzfrei.

### A.1.0.3 IEEE 802.15.1 BLUETOOTH

Bluetooth ist ein durch die Bluetooth SIG (engl. Special Interest Group) entwickelter Industriestandard gemäß des Standards IEEE 802.15.1. BLUETOOTH wurde vorwiegend für Datenübertragungen zwischen Geräten über kurze Distanzen wie WPAN (engl. Wide Personal Area Network) entwickelt. Es werden Übertragungen von Punkt-zu-Punkt und Ad-hoc-<sup>1</sup> oder Piconetze <sup>2</sup> unterstützt.

Eine zuverlässige und störungsresistente Funkverbindung wird durch ein adaptives Frequenzsprungverfahren (engl. Adaptive Frequency-Hopping, AFH) gewährleistet. Dieser Vorteil von Bluetooth gegenüber anderen Funktechnologien wurde frühzeitig von verschiedenen Herstellern wie Phoenix Contact [107], WAGO Kontakttechnik GmbH & Co. KG [39] und Schildknecht AG [108] für Automatisierungsprodukte erkannt. Daraufhin wurden Bluetooth basierende Industrieprodukte realisiert, die in verschiedensten Bereichen der Industrie eingesetzt werden, um kabellos zwischen verschiedenen Komponenten in Maschinen zu kommunizieren zu. In der Industrie wird dieser Standard insbesondere für WSA in der Fabrikautomatisierung (engl. Fabric Automation, FA) genutzt. Daher werden diese Netzwerke auch oft als WSA-FA [94] bezeichnet.

Mittlerweile hat die PROFIBUS Nutzerorganisation e. V. (PNO) [109] Bluetooth neben WLAN als Trägertechnologie für kabellose Übertragung von PROFIBUS [110] Datenpaketen definiert. Auch die Vereinigung CAN <sup>3</sup> in Automation (CiA) [111] plant die Spezifikation eines auf Bluetooth basierenden Systems für die kabellose Übertragung von CAN-Telegrammen.

### A.1.0.4 WISA

WISA (engl. Wireless Interface for Sensors and Actors) [40] von der Firma ABB [112] ist ein Funkübertragungssystem für die Steuerung von Automatisierungsanlagen. Das Datenübertragungsprotokoll entspricht dem in Bluetooth verwendeten Standard IEEE 802.15.1 [25]. Im Gegensatz zu BLUETOOTH wird hier das Frequenzsprungverfahren FHSS (engl. Frequency Hopping Spread Spectrum) mit 79 Sprungfrequenzen genutzt. Das Frequenzsprungverfahren kombiniert mit einem Zeitschlitzverfahren TDMA (Time Division Multiple Access) ermöglicht den Einsatz von WISA für Echtzeitanwendungen. WISA zeichnet sich wie ZigBee durch einen äußerst geringen Energieverbrauch aus. Aufgrund der niedrigen Sendeleistung von 1 mW, bestehen die zellularen Netze aus vielen Picozellen, um größere Bereiche wie Fabrikhallen abzudecken.

---

<sup>1</sup>Ein Ad-hoc-Netz bezeichnet in der Informationstechnik eine drahtlose Netzwerktopologie zwischen zwei oder mehr Endgeräten, die ohne feste Infrastruktur auskommt.

<sup>2</sup>Piconetze sind Kleinstfunkzellen. Sie bestehen mindestens aus einem Master und bis zu 7 Slaves.

<sup>3</sup>engl. Controller Area Network

### A.1.0.5 Industrial WLAN

WLAN 802.11 steht für eine standardisierte Funktechnologie, wie es auch im Heimbereich für die PC Funkvernetzung verwendet wird. WLAN ist auf hohe Datenübertragungsraten optimiert. Dieser Standard wird permanent erweitert und ergänzt, daher gibt es derzeit eine Vielzahl an WLAN Standards. Tabelle A.1 stellt die Frequenzbänder und Übertragungsraten [113] der verschiedenen WLAN-Standards dar.

**Tabelle A.1:** Unterschied zwischen diversen Wlan-Standards

Standard	802.11	802.11b	802.11 a/h/j	802.11 g	802.11 n	802.11 ac
Frequenzbänder (GHz)	2,4	2,4	5	2,4	2,4 und 5	5
Brutto (Mb/s)	2	11	54	54	150 - 600	1.300
Netto (Mb/s)	0,5-1	5	bis 25	25	bis 200	bis 650
Reichweite Indoor (m)	20	20	38	38	20	20
Reichweite Outdoor (m)	100	120	bis 2 km	100	100	50

IWLAN (engl. Industrial WLAN)[114] ist eine Weiterentwicklung des bekannten WLAN für den industriellen Einsatz durch die Firma Siemens AG [115].

Der wesentliche Unterschied im Vergleich mit dem im Privatbereich stark verbreiteten WLAN liegt in der zeitgenauen Übertragung der Steuer- und Datensätze, was für den Einsatz in der Industrie zwingend erforderlich ist, um Maschinen sicher steuern zu können. Außerdem sind solche Geräte für einen größeren Temperaturbereich von  $-40^{\circ}\text{C}$  bis  $+70^{\circ}\text{C}$  ausgelegt. Das Netz arbeitet mit einem speziellen Verschlüsselungsverfahren, um Manipulationen zu unterbinden. Aktuelle IWLAN-Produkte von Siemens basieren auf dem Standard IEEE 802.11n.

## A.2 UWB Forschungsprojekte im detail

In diesem Abschnitt werden die vorgestellten UWB Forschungsprojekte aus Kapitel 2 detaillierter vorgestellt. Abbildung A.1 ist eine übersichtliche Darstellung aller Projekte.

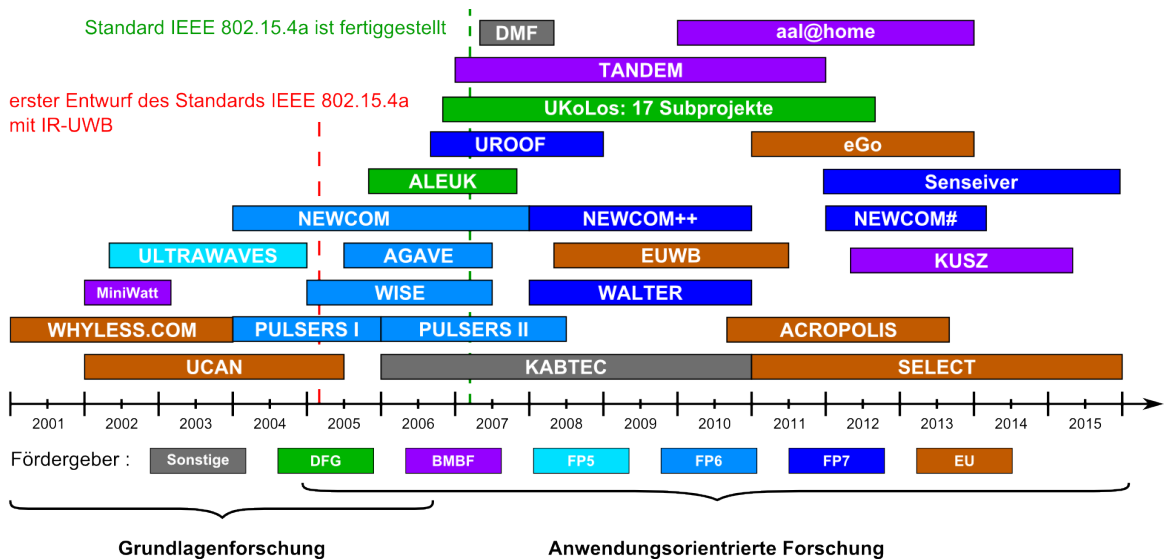


Abb. A.1: Forschungsprojekte im Bereich UWB von 2001 bis 2015

**WHYLESS.COM****Laufzeit:** 01/2001 - 12/2003**Förderung durch:** EU IST-Rahmenprogramm

**Partner:** MST GmbH (Germany); University of Rome "La Sapienza" (Italy); Dresden University of Technology (Germany); University of Perugia (Italy); Noventum Consulting (Germany); Dresden University of Technology (Germany); University of California, Los Angeles (USA); Exodus (Greece)

**Kurzbeschreibung:** Ziel des EU-Vorhabens ist die Entwicklung von Business-Szenarien auf Basis der Highspeed-Funktechnologie UWB (Ultra-Wideband). Unter der Leitung der IMST GmbH, Kamp-Lintfort, erarbeitet die noventum consulting GmbH dazu gemeinsam mit fünf anderen Partnern Netzwerkarchitekturen, bei denen nicht der Anbieter, sondern die Belange der Nutzer im Vordergrund stehen.

**MiniWatt - Alternative Funksysteme mit minimaler Strahlungsleistungsdichte im digitalen Rundfunk, Mobilfunk, drahtlosen LANs****Laufzeit:** 01/2002 - 03/2003**Förderung durch:** BMBF

**Partner:** acht Universitätsinstitute, ein Fraunhofer-Institut sowie mehrere Industrieunternehmen und mittelständische Betriebe

**Kurzbeschreibung:** In dem breit angelegten Forschungsvorhaben, wurden verschiedene technologische Ansätze verfolgt, drahtlose Kommunikation und Information mit möglichst geringen Leistungsflussdichten zu realisieren. Das Forschungsvorhaben berücksichtigt unter anderem auch verschiedene WPAN-Anwendungen, u.a. mit UWB und anderen alternativen Übertragungsverfahren.

**UCAN - UWB Concepts for Ad-hoc Networks****Laufzeit:** 01/2002 - 06/2005

**Förderung durch:** EU

**Partner:** Gesellschaft für Wissens- und Technologietransfer Dresden mbH (Deutschland); Advanced Communication Research and Development S.A. (Spanien); University of Cantabria (Spanien); Centre de Recherche Motorola, Motorola S.A. (Frankreich); STMicroelectronics NV (Schweiz); THALES Communications (Frankreich); University of Rome "La Sapienza" (Italien); Digital DNA Laboratory Frankfurt, Motorola GmbH (Deutschland); CEA-LETI (Frankreich)

**Kurzbeschreibung:** Das Ziel von U.C.A.N. ist sowohl die theoretische als auch experimentelle Untersuchung von nachrichtentechnischen Systemen auf Basis der Ultra-Wideband (UWB) Übertragungstechnik. Ein wesentlicher Aspekt der durchzuführenden Arbeiten ist die Analyse einer speziellen physikalischen Übertragungsschicht. Das dabei angewendete sogenannte Impulse Radio (IR) Übertragungsverfahren ist von besonderem Interesse, da es auf die Verwendung von Trägersignalen verzichten kann. In Verbindung mit IR werden zur Modulation die Verfahren Puls Position Modulation (PPM), Puls Amplituden Modulation (PAM), sowie Mischformen in verschiedenen Varianten betrachtet.

**ULTRAWAVES**

**Laufzeit:** 04/2002 - 12/2004

**Förderung durch:** EU FP5

**Partner:** Philips (Netherlands and UK); Wisair Ltd (Israel); ENSTA, ARMINES (France); Radiolabs (Italy); University of Oulu (Finland); Chalmers University of Technology (Sweden)

**Kurzbeschreibung:** ULTRAWAVES objective is to provide a high performance and low cost wireless home connectivity solution, supporting applications requiring home multi streaming of high quality video and broadband multimedia as well as the IPv6 emerging protocol.

**PULSERS I - Pervasive Ultra-wideband Low Spectral Energy Radio Systems**

**Laufzeit:** 01.2004 - 12.2005

**Förderung durch:** EU, IST FP6

**Partner:** 23 Partner aus 8 EU Länder; 3 Schweiz; 1 Israel; 1 Russland; 2 Singapur

**Kurzbeschreibung:** Das Ziel von Pulsers ist es, einen Beitrag zur Realisierung der Vision "optimal verbunden an jedem Ort zu jeder Zeit" zu leisten. Dies soll durch wissenschaftliche Untersuchungen und durch innovative Entwicklung von drahtlosen Systemen und Architekturen für kurze Distanzen auf Basis der Ultra-Wideband (UWB) Übertragungstechnik erreicht werden. Zum einen sollen dafür neue Konzepte für die physikalische Übertragungsschicht (PHY) und die Media Access Control (MAC) Struktur erarbeitet werden und zum anderen Hard- und Softwareplattformen bereitgestellt werden, welche einen großen Bereich an Einsatzgebieten abdecken.

**PULSERS II - Pervasive Ultra-wideband Low Spectral Energy Radio Systems**

**Laufzeit:** 01/2006 - 06/2008

**Förderung durch:** EU, IST FP6

**Partner:** 39 Partner

**Kurzbeschreibung:** In Phase II des Projektes PULSERS erfolgt eine Fokussierung auf konkrete Anwendungsfälle der Ultra-Wideband-Technik (UWB), wobei grundsätzlich zwischen Very High Data Rate Systems (VHDR) und Low Data Rate Systems (LDR) unterschieden wird. Für Zieldatenraten bis 1,6 GBit/s zur drahtlosen Übertragung von hochwertigem Videomaterial wird ein Multiband-OFDM-System realisiert. Daneben wird ein Multiband-Impulsradio-Ansatz verfolgt, für den ebenfalls ein Demonstratorsystem entwickelt werden soll. Im Bereich niedriger Datenraten werden drei unterschiedliche Konzepte bearbeitet. Zum einen wird ein System entwickelt, das neben der Kommunikation auch die vielversprechenden Möglichkeiten der Lokalisierung, die ein Impulsradiosystem bietet, demonstrieren soll (Low Data Rate-Location and Tracking, LDR-LT). Zum zweiten werden Sensornetzwerke untersucht und zum dritten geht es um Body-Area-Networks (BAN), die z.B. im medizinischen Bereich Anwendung finden werden. Neben den technischen Entwicklungszielen besteht eine weitere wichtige Aufgabe des Projektes in der Unterstützung der Aktivitäten zur europäischen Regulierung der UWB-Technik.

#### **NEWCOM - Network of Excellence**

**Laufzeit:** 2004 - 2007

**Förderung durch:** EU FP6

**Partner:** CNIT(Modena, Milano, Firenze); Supélec; Eure com; Thalès; TUI; TUA; AAU; ISMB; Chalmers

**Kurzbeschreibung:** NEWCOM aims at creating a European network that links in a cooperative way a large number of leading research groups addressing the Strategic Objective "Mobile and wireless systems beyond 3G", a frontier research area of the Priority Thematic Area of IST. The main dimensions of the NEWCOM vision are: - Strengthening, development and integration of research in the field - Empowerment of groups and individuals via dissemination activities - Effective use of produced knowledge via exploitation-commercialisation-standardization strategies. To achieve these dimensions, NEWCOM will implement an elaborate plan of initiatives which revolve around the key notion and strategic choice of a Virtual Knowledge Centre: NEWCOM will effectively act as a distributed (decentralised) university, organised in a matrix fashion. The columns will represent the seven NEWCOM (Disciplinary) Departments, characterised by basic research on well-established topics and grouping leading European researchers active in those topics. The rows will represent NEWCOM Projects, identified by important, "hot" problems whose solution requires multidisciplinary skills drawn from NEWCOM Departments and aggregated in a meaningful way to promote the problem solution. The Joint Programme of Activities involves researcher exchanges, organisation of workshops and conferences, the preparation of graduate courses coordinated with the PhD programs of the academic partners to be diffused via NEWCOM high-speed network, the broad dissemination of scientific results, the promotion of entrepreneurship among its researchers, by

setting up a policy of IPR encouragement and their exploitation through the creation of start-ups inside its distributed campus. The "glue" connecting everything together is a set of tools for Integration, the unifying thread making all objectives and goals a feasible vision, and for Management, which maintains a clear separation between "administrative" and "scientific" tasks.

**NEWCOM++ - Network of Excellence**

**Laufzeit: 2008 - 2010**

**Förderung durch: EU FP7**

**Partner: 17 Partner aus 13 Länder**

**Kurzbeschreibung:** The objective of the NEWCOM++ project is creating a trans-European virtual research centre linking a proper number of leading European research groups in a highly integrated, carefully harmonized, cooperative fashion. This collection of groups follows a carefully tailored joint program of research activities heterogeneous Radio Access Techniques and Radio Access Networks characterized by very different bit rates, frequency bands and bandwidths, medium access strategies, etc.

**NEWCOM# - Network of Excellence**

**Laufzeit: 2012 - 2014**

**Förderung durch: EU FP7**

**Partner:** CNIT (IT); Aalborg University (DK); Bilkent University (TK), CNRS (FR); CTTC (ES); IASA (GR); INOV (P); Poznan University of Technology (PL); Technion (IL); Technische Universitaet Dresden (D); University of Cambridge (UK); Université de Louvain (BE); Oulun Yliopisto (FIN); Technische Universitaet Wien (A)

**Kurzbeschreibung:** NEWCOM# pursues long-term, interdisciplinary research on the most advanced aspects of wireless communications like Finding the Ultimate Limits of Communication Networks, Opportunistic and Cooperative Communications, or Energy- and Bandwidth-Efficient Communications and Networking.

**WISE - Integrated Wireless SEnsing Laufzeit: 01/2005 - 06/2007**

**Förderung durch: EU FP6**

**Partner:** Brno University of Technology; Advanced Communications Research and Development S.A.; TES Electronic Solutions; EADS Deutschland GmbH; Eurocopter; Saarland University; Messier-Bugatti; CSEM - Recherche et Dveloppement - SA'; Warsaw University of Technology; Teletel; DASSAULT Aviation

**Kurzbeschreibung:** WISE war ein Projekt welches als Ziel hatte verschiedene Technologien zu untersuchen und zu entwickeln welche in der Luftfahrt eingesetzt werden können; unter anderem wurde auch UWB untersucht und es entstand ein Demonstrator, basierend auf FPGAs, welcher den Standard IEEE 802.15.4a unterstützt.

**AGAVE - AGv Navigation system Based on Flexible and Innovative UWB Positioning**



**Laufzeit:** 07/2005 - 07/2007

**Förderung durch:** FP6

**Partner:** LABOR S.R.L (IT); NUOVAFIMA S.P.A (IT); PRO-CERAM GMBH (DE); DIRK SCHUMANN GMBH (DE); COSMOTEK S.R.L (IT); MESUREX S.L. (ES); UNIVERSITÀ DEGLI STUDI DI UDINE - DIPARTIMENTO DI INGEGNERIA ELETTRICA GESTIONALE E MECCANICA (IT); ISTITUTO AGILUS DE INOVAÇÃO EM TECNOLOGIAS; DE INFORMAÇÃO LDA (PT); UNIVERSIDAD DE MÁLAGA (ES)

**Kurzbeschreibung:** The AGAVE project overall objective is the development of an advanced guidance system for Automatic Guided Vehicles, focusing on a next generation positioning system development, allowing indoor and outdoor goods transportation within the warehouse through a flexible material handling system. The AGAVE system is a free-range system allowing a high level of optimisation in the routes to be executed, being its performances superior to those currently available on the market. It will be an integrated system easily adaptable to production environment changes and able to improve the overall quality, reducing the total lead-time. The following outcomes are expected: 1) A stand-alone plug-in positioning system, based on UltraWideBand technology, exploiting computational hardware commercialised by the company TimeDomain, with a time to market of about 15 months after project completion. 2) A navigation system, installed onboard, based on the improvement of the existing systems. 3) An integrated software for fleet management, route optimisation and integration with operating machines, installed on a fixed host. The AGAVE system main technical advantages are: - operate in a very wide warehouse (average indoor route: 150-200 m) - realise a very flexible system, easily adaptable to outdoor applications (expected average route: 4-5 Km) - be compatible with all AGV configurations, even platform carriers, which is currently impossible with laser based positioning systems - be compatible with all warehouse layouts, with narrow corridors or the occupation of the overhead volume now kept free to ensure line of sight conditions for laser systems - be easily manageable and configurable through an advanced user interface - well automate goods transportation logistic phase, ensuring a smooth functioning of the pick and drop phases, in order to reduce the total lead-time of the entire industrial process.

#### **ALEUK Analyse von alternativen Empfängerkonzepten für die Ultra-Wideband Kommunikation**

**Laufzeit:** 09/2005 - 08/2007

**Förderung durch:** DFG

**Partner:** Technische Universität Dresden (TUT)

**Kurzbeschreibung:** Seit der erfolgreichen Einführung der drahtlosen Kommunikation mittels Mobiltelefonen ist ein stetig wachsendes Interesse an vergleichbaren Techniken für verschiedene Einsatzgebiete zu verzeichnen. Derzeit sind Begriffe wie Wireless Local Area Network (WLAN) oder Bluetooth (BT) in aller Munde. Eine Folge dieser Entwicklung ist die immer knapper werdende Ressource Frequenz, wie beispielsweise die Versteigerung der UMTS-Lizenzen im Jahr 2000 zeigte. Somit müssen neue Wege beschritten werden, um diesen Engpass zu regulieren. Bei

ALEUK wird eine technisch mögliche Lösung für dieses Problem im Mittelpunkt stehen. Sie beruht auf der Anwendung extrem breitbandiger Übertragungstechniken wie Ultra-Wideband (UWB). Aufgrund der ihr zu Grunde liegenden Funktionsweise ist der gleichzeitige Betrieb mehrerer UWB-Geräte im gleichen Frequenzbereich möglich. Eine Anwendung dieser Technik besteht im Einsatz sehr kurzzeitiger Impulse.

**KABTEC (SecureAirCom)****Laufzeit: 2006 - 2010****Förderung durch:** DLR Bonn**Partner:** Airbus; AIDA Development GmbH; Apparatebau Gauting GmbH; Diehl Aerospace GmbH; DLR e.V.; EADS Deutschland GmbH; IMST GmbH; Monogram Systems GmbH; Rheinmetall Defence Electronics GmbH; Sell GmbH; WITTENSTEIN A& S GmbH; Comtronic GmbH; Hochschule Pforzheim (IAF); Opsira GmbH**Kurzbeschreibung:** Development of an ultra-wideband location and communication system in airplane cabins. The main project, KABTEC, seeks to improve total cabin architecture, including seats, using innovative technology. The project is based on a modular approach to the use of components, offering greater comfort, efficiency, and utility even in economy class. The sub-project, SecureAirCom, focuses on the development of an ultra-wideband system. This location and communication system will be used by the crew to improve crisis management during an emergency. A special problem is the specific propagation channel conditions in the cabin; these have to be taken into account to optimize the configuration of the RF base stations integrated in the cabin ceiling and the mobile stations, which are tags worn by the crew.**UROOF - Photonic components for Ultra-Wideband radio over optical fibre Proposal****Laufzeit: 07/2006 - 12/2008****Förderung durch:** EU FP7**Partner:** HOLON INSTITUTE OF TECHNOLOGY (ISRAEL); TES ELECTRONIC SOLUTIONS GmbH (GERMANY); WISAIR LTD. (ISRAEL); THE CENTRE FOR INTEGRATED PHOTONICS LIMITED (UNITED KINGDOM); INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE (FRANCE); INESC PORTO - INSTITUTO DE ENGENHARIA DE SISTEMAS E COMPUTADORES DO PORTO PORTUGAL); UNIVERSITY OF ESSEX (UNITED KINGDOM); THALES COMMUNICATIONS SA (FRANCE); UNIVERSIDAD POLITECNICA DE VALENCIA (SPAIN)**Kurzbeschreibung:** The key goal for UROOF is to investigate building blocks for enabling the delivery of Ultra-wideband (UWB) radio signal over low-cost optical fibre. The aims are therefore to study, develop, test and implement very low cost conversion solutions for direct optical-to-UWB (O/UWB) and UWB-to-optical (UWB/O) based on innovative microwave photonic concepts. Unlike state of the art radio-over-fiber (RoF) technologies that are used in the backbone of the wireless access systems, UROOF will address the challenges of the low-cost wireless personal

area networks (WPAN).

### **TANDEM**

**Laufzeit:** 2007 - 2011

**Förderung durch:** BMBF

**Partner:** IHP GmbH - Innovations for High Performance Microelectronics; Green-Way Systeme GmbH; lesswire AG; Sensys Sensorik & Systemtechnologie GmbH; neun weitere Firmen des Unternehmensnetzwerks Präzisionssensorik (PräSens)

**Kurzbeschreibung:** Das Vorhaben beinhaltet die Untersuchung und Entwicklung einer Sensorkommunikationstechnologie, welche die Basis eines vielseitig anwendbaren, intelligenten, energieeffizienten und funkbasierten Sensornetzes werden soll. Dieses Sensornetz soll vorrangig für die Anwendungsbereiche Verkehrstechnik, Logistik, Sicherheitstechnik und Haus- automatisierung ausgelegt werden, aber auch in anderen Bereichen wie Medizin, Aktuator- und Industrietechnik oder zur Kennzeichnung benutzt werden können. Mit einem Sensor- und Aktuatorkommunikationssystem können vorhandene Produkte verbessert werden. Außerdem eröffnet es den Firmen neue Produktvisionen.

**Walter - Wireless ALliance for Testing Experiment and Research**

**Laufzeit:** 2008 - 2010

**Förderung durch:** EU FP7

**Partner:** inno AG, ETSI, TMC, JRC, CTL, AT4 wireless, Wisair and CSR

**Kurzbeschreibung:** The project WALTER addresses the need for broadband testbeds by overcoming the technical issues of measuring low level, high datarate radio signals. The resulting worldwide interconnected testbeds will address the short term needs of industry and regulators but also the longer-term needs of research communities. Further information on the WALTER project is available at [www.walter-uwband.eu](http://www.walter-uwband.eu)

### **EUWB - European Ultra-Wideband**

**Laufzeit:** 04/2008 - 07/2011

**Förderung durch:** EU

**Partner:** TES Electronic Solutions (DE); Philips (NL); Bosch (DE); CEA-Léti (FR); Leibniz Universität Hannover (DE); CREATE-NET (IT); VTT Electronics (FI); EADS (DE); University of Oulu (FI); Telefónica (ES); THALES Communications (FR); Wisair (IL); TU Ilmenau (DE); Universidad de Zaragoza (ES); ACORDE TECHNOLOGIES (ES); TES Electronic Solutions (UK), HTW Dresden (DE); Università degli Studi di Bologna (IT); FBConsulting (LU); Universität Duisburg-Essen (DE); Veebeam (UK); University of Bucharest (RO); Bitgear (RS); EIT+ Wrocław (PL); Czech Technical University in Prague

**Kurzbeschreibung:** Its key objective is to explore the enormous economic potential of the innovative and disruptive Ultra-Wideband (UWB) radio technology for key industrial sectors in Europe. The project aims to consolidate technical advances in scientific areas related to UWB, such as cognitive signalling, intelligent multiple antenna and multiband/ multimode concepts. These advances are envis-

aged to lead to the definition of system concepts for four key application areas in major European industry sectors. Based on a strong demand from these industry sectors the four prominent examples in which applications will be implemented are the Automotive environment (driven by Bosch), the Public transport environment (driven by EADS), the Intelligent home environment (driven by Philips) and the next generation (NG) of heterogeneous public access network environment (driven by Telefónica). Results will be provided by four integrated application areas delivered on the EUWB open UWB technology platforms.

**aal@home****Laufzeit: 2010 - 2013****Förderung durch: BMBF**

**Partner:** Hausarztpraxis Adendorf; Kieback& Peter GmbH Co. KG, Hamburg/Berlin; Klinikum Lüneburg GmbH, Lehrkrankenhaus der Uni Göttingen; Leuphana Universität Lüneburg - Institut für verteilte autonome Systeme und Technologien; Panasonic Industrial Devices Europe GmbH; Paritätischer Wohlfahrtsverband Niedersachsen e.V., Kreisverband Lüneburg; Pflegedienst Lilienthal GmbH; telemed Online Service für Heilberufe GmbH, Koblenz; Technische Universität Ilmenau, FB Elektronische Messtechnik; Universität Kassel, FB Maschinenbau, Arbeits- und Organisationspsychologie

**Kurzbeschreibung:** Im Projekt aal@home entsteht ein Assistenzsystem, das die Sicherheit und damit die Unabhängigkeit älterer, allein lebender Menschen gewährleisten soll. Für die Senioren wird ein engmaschiges Versorgungsnetz aufgebaut, in das Angehörige, Pflegedienst, Hausarzt und Klinik eingebunden werden können. Alle Beteiligten haben ständig einen Einblick, wie es den alten Menschen gerade geht und ob Hilfe erforderlich ist. Möglich wird dies über hoch innovative UWB-Sensoren (UWB: Ultrabreitband /Ultra- Wideband), die kontinuierlich den Gesundheitszustand der Senioren überprüfen können. Neben den Vitaldaten kann auch die aktuelle Position in der Wohnung gemessen werden. Dafür werden die Messgeräte in den Wänden installiert. Bis zu einer Entfernung von acht Metern können die Daten erfasst werden. Damit die Sensoren jeden Bewohner klar identifizieren können, trägt jeder eine individuelle Sensorkarte bei sich. Auch aus der Haustechnik lassen sich Informationen über die Aktivitäten der Senioren ableiten, beispielsweise über die Nutzung der Lichtschalter. Die verschiedenen Daten ergeben ein aussagefähiges Abbild der Lebenssituation, das von einem verteilten, lernfähigen Assistenzsystem zusammengefasst und interpretiert wird. Über ein Medizindatenkommunikationssystem wird der aktuelle Lagebericht an ein Notruf- und Sicherheitssystem weitergeleitet, das Angehörige, Pflegedienste, Hausärzte und Kliniken gleichermaßen einbezieht. Das Projekt schließt damit die Lücke zwischen dem bestehenden Hausnotrufprinzip und telemedizinischen Systemen. Zusätzlich bekommen die Senioren ein einfaches mobiles Endgerät an die Hand, mit dem sie ihre Haustechnik steuern, aber auch mit entfernten Internetdiensten und Personen Kontakt aufnehmen können.

**ACROPOLIS - Advanced Coexistence Technologies for Radio Optimisation in Licensed and Unlicensed Spectrum****Laufzeit:** 10/2010 - 09/2013**Förderung durch:**EU**Partner:** RWTH Aachen (DE);University of Rome "La Sapienza" (IT); Institute of Accelerating Systems and Applications (GR); University of Piraeus Research Center (GR); Wrocław Research Center EIT+ (PL); King's College London (UK); KTH Royal Institute of Technology (SE); SS. Cyril and Methodius University Skopje (MK); Centre Tecnologic de Telecomunicacions de Catalunya (ES); Joint Research Centre European Commission (BE); Poznan University of Technology (PL); University of Surrey (UK); EURECOM (FR); University of Leeds (UK); EDAS Innovation Works (DE)**Kurzbeschreibung:** The vision of the ACROPOLIS consortium is to integrate European research efforts in the domains of cooperative and cognitive communications such that coexistence and capabilities of future networks operating both in the licensed and open spectrum bands can be significantly increased. The key issue here is the pooling of our knowledge on flexible radios, coexistence methods, cognitive radio technologies, and also regulatory and economical issues, in order to facilitate harmonised and interdisciplinary joint research and roadmapping.**eGo Project Laufzeit: 01/2011-12/2013****Förderung durch:** EU EUREKA (CATRENE)**Partner:** CIT; Continental Automotive; Decawave; Gemalto; IDEX ASA; Inria; Lincor Solutions Ltd; Precise Biometrics; STMicroelectronics; Tyndall; Worldline**Kurzbeschreibung:** The eGo consortium, funded by EU initiative Catrene, proposes a new and innovative way to establish secure, bidirectional wireless channels between objects or individuals in the future internet of things, based on a bootstrapping scheme using the electrical conductivity of human skin. This will open the path to new and intuitive ways of interaction for electronic transactions. DecaWave is providing the Ultra Wideband communications component for the eGo project.**KUSZ - Kognitives IR-UWB-System Laufzeit: 04/2012 - 03/2015****Förderung durch:** BMBF**Partner:** IHP GmbH - Innovations for High Performance Microelectronics; EADS Innovation Works Germany; Festo AG & Co. KG; Fraunhofer Institut für Telekommunikation; TES Electronic Solutions; TU Dortmund; WAGO Kontakttechnik GmbH & Co. KG**Kurzbeschreibung:** Im Projekt wird ein kognitives UWB-Funksystem für industrielle Anwendungen entwickelt, bei denen hohe Zuverlässigkeit, kurze Latenzzeit und flexible Anpassung an die Umgebung gefordert wird. Koexistenzprobleme werden systembedingt vermieden.**SELECT Laufzeit: 2011-2015****Förderung durch:** EU

**Partner: Kurzbeschreibung:** The objective of the SELECT project is the design of a cheap smart wireless network, composed of several wireless cooperating microsystems where the detection, identification, and tracking of objects are integrated, going beyond the limitation of the existing techniques for automatic identification (AutoID) and Real Time Location Systems (RTLS). Limitations are for instance, the unavailability of location systems for semi-passive tags, inflexible, cabled network structures of RTLS, the missing compatibility to prior systems and the missing unification of the above functionalities. Envisioned technologies to overcome these limitations include radio frequency identification (RFID), ultra-wideband (UWB) radio and radar as well as up to date data fusion.

**Senseiver - Low-cost and energy-efficient LTCC sensor/IR-UWB transceiver solutions for sustainable healthy environment**

**Laufzeit:** 12/2011-11/2015

**Förderung durch:** EU FP7

**Partner:** University of Novi Sad (Serbien); Vienna University of Technology (Österreich); INESC - Porto (Portugal); Institute of Electron Technology; Technical University of Iasi; TES Electronic Solutions GMBH; North Point Ltd. (Großbritannien); Ministry of Science and Technological Development (Serbien); Foundation for promoting advanced research "New Wave Science Mates" (Rumänien); Municipal Waterworks and Sewer Enterprise in Kraków Joint Stock Company (Polen)

**Kurzbeschreibung:** The main goal of the network is to improve career perspectives of early-stage and experienced researchers from involved organizations, to structure initial research training at the EU level and to spread knowledge and skills in the field of new sensors, materials, transceivers and data acquisition systems, for application in environmental parameters monitoring.

**DMF - Deutsches Mobilfunk Forschungsprogramm: Bestimmung der Exposition durch Ultra-Wideband Technologie Laufzeit: 04/2007 - 04/2008**

**Förderung durch:** —

**Partner:** Ziel dieses Forschungsvorhabens war es, möglichst frühzeitig eine belastbare Datenbasis für die von verschiedenen Ultra-Wideband Funkanwendungen verursachten maximal möglichen und durchschnittlich zu erwartenden Expositionen zu schaffen. Hierzu sollten bestehende Mess- und Berechnungsverfahren auf ihre Eignung geprüft und ggf. angepasst werden.

**UKoLoS - Ultra-Wideband Radio Technologies for Communications, Localization and Sensor Applications**

**Laufzeit:** 2006 - 2012

**Partner: DFG Kurzbeschreibung:** UKoLoS steht für "Ultra-Wideband Radio Technologies for Communications, Localization and Sensor Applications" und ist ein DFG-Schwerpunktprogramm. Dieses Schwerpunktprogramm soll mit interdisziplinären Verbundprojekten eine deutschlandweite Vernetzung von Forschungsaktivitäten im UWB-Bereich anregen und zur Einbindung in internationale Forschungsnet-

zwerke beitragen. Außerdem sollen die Grundlagen der UWB-Technik für besonders innovative Anwendungsfelder erarbeitet werden.

**Subprojekte:**

- MIRA
- UltraEfficiency
- Multi-User UWB communication
- PAOLA
- Low complexity UWB communication
- Chip2Chip
- HaLOS
- RoboLoc
- ultraMEDIS
- UWB in Medicine
- CoLOR
- ESPURS
- 60 GHz Short Range Communication
- CoMoDe IR-UWB
- ISOPerm
- UltraMEDIS-II
- UWB4WSN

### A.3 Ermittlung der maximalen Datenrate

Die Datenrate ist die Anzahl an Bits, welche pro Sekunde übertragen werden können. Die Anzahl an Bits wird anhand der Anzahl an verschiedenen Zuständen ermittelt.

Sei  $n = 512$  die Anzahl an *time-slots*, sei  $\#I$  die maximale die Anzahl an Impulsen und sei  $\#P_{(n,\#I)}$  die Anzahl an möglichen Zustände für exakt  $\#I$  Impulse. Jeder Impuls kann negativ oder positiv sein.  $\#P_{(n,\#I)}$  kann wie folgt aus  $\#P_{(n,k)}$  berechnet werden.

$$\#P_{(n,\#I)} = \sum_{k=0}^{\#I} \#P_{(n,k)}$$

Zur Ermittlung von  $\#P_{(n,k)}$  wird zuerst der Fall von  $n$  *time-slots* und exakt  $k$  Impulsen ohne Berücksichtigung der Polarität der Impulse betrachtet.

Die Impulse werden nach und nach platziert. Beim ersten Impuls gibt es  $n$ , beim zweiten Impuls  $(n-1)$  und beim  $k$ -ten Impuls  $(n-k)$  mögliche Zustände. Dies entspricht  $n \cdot (n-1) \cdot \dots \cdot (n-(k-1)) = \frac{n!}{(n-k)!}$  mögliche Zuständen.

Da die Impulse aber nicht unterscheidbar sind, müssen noch alle Permutationen der Impulse von den möglichen Zuständen entfernt werden. Die Anzahl an Permutationen für  $k$  Objekte ist  $k!$ . Demzufolge gibt es  $\frac{n!}{k! \cdot (n-k)!}$  Mögliche Zustände.

Es bleibt noch die Polarität der verschiedenen Impulse zu berücksichtigen. Es gibt für jeden Zustand  $2^k$  Varianten für die Polarität. Daraus ergibt sich für  $\#P_{(n,k)}$ :

$$\#P_{(n,k)} = \frac{n!}{k! \cdot (n-k)!} \cdot 2^k = \binom{n}{k} \cdot 2^k$$

Demzufolge ist die Anzahl an verschiedenen Zuständen für  $\#I \leq 16$  gleich:

$$\#P_{(n,\#I)} = \sum_{k=0}^{16} \#P_{(n,k)} = \sum_{k=0}^{16} \frac{n!}{k! \cdot (n-k)!} \cdot 2^k = \sum_{k=0}^{16} \binom{n}{k} \cdot 2^k$$

Die maximal mögliche Datenrate ist dann:

$$\text{Datenrate} = \frac{\lfloor \log_2 (\#P_{(n,\#I)}) \rfloor}{n \cdot 2ns}$$

## A.4 Berechnung optimaler Impulssequenzen für PREAMBLE-Symbole

### A.4.1 Vorüberlegungen

Wie zuvor in Kapitel 3.2 erwähnt, wurden optimale Impulssequenzen für PREAMBLE-Symbole mittels FPGA berechnet. Hier wird nun erklärt wie dabei vorgegangen wurde.

Zuerst wurde das Format für die PREAMBLE-Sequenzen wie folgt festgelegt. Eine Impulssequenz für PREAMBLE-Symbole besteht aus 2 Vektoren *data* und *polarity* der Länge 31.

```

1 data      : std_logic_vector(30 downto 0);
2 polarity  : std_logic_vector(30 downto 0);

```

Wenn  $data(i) = '0'$  dann ist dies eine 0 bzw. ein leerer Impuls.

Wenn  $data(i) = '1'$  dann ist dies ein Impuls und die Polarität ergibt sich durch *polarity(i)* (1 ist ein negativer Impuls und 0 ein positiver Impuls).



Das folgende Beispiel illustriert dies:

```

Sequenz  : 0+00-0-0++0000--+00-+0+-++0+00
data     : 0100101011000011100110111110100
polarity : 0000101000000011000100001000000

```

Angenommen, der FPGA arbeitet mit einer Frequenz von 100 MHz und bei jedem Taktzyklus kann  $A_p$  für eine Kombination aus *data* und *polarity* berechnet werden, dann dauert die Berechnung:

$$\begin{aligned}
TIME &= \frac{10 \text{ ns} \cdot (2^{31} - 1) \cdot (2^{31} - 1)}{10^{-9}} = 4,6117 \cdot 10^{10} \text{ Sekunden} \\
TIME &= 1,281 \cdot 10^7 \text{ Stunden} \\
TIME &= 5,3376 \cdot 10^5 \text{ Tage} \\
TIME &= 1462 \text{ JAHRE}
\end{aligned}$$

Dies ist offensichtlich zu lang, daher muss die auszuprobierenden Menge reduziert werden.

Der FPGA benötigt 21,4 Sekunden, um alle Möglichkeiten  $2^{31} - 1$  für *data* zu durchlaufen. Dies kann auf  $2^{28} - 1$  reduziert werden, wenn angenommen wird, dass jede Sequenz, welche alle Anforderungen erfüllt, mindestens 3 aufeinanderfolgende Nullen beinhaltet. Weiterhin müssen nur Sequenzen berechnet werden, wo die Quersumme von *data* = 15 ist. Dies bedeutet, dass *data* genau 15 Impulse aufweisen muss. Dies würde aber nur eine Einsparung von wenigen Sekunden bringen, weshalb es sich nicht lohnt diese Sequenzen herauszufiltern. Viel wichtiger ist es, die Anzahl an Möglichkeiten für *polarity* zu reduzieren. Da die Polarität nur für genau 15 Impulse in *data* Anwendung findet, kann die Anzahl auf  $2^{15} - 1$  reduziert werden. Daraus ergibt sich folgende Dauer.

$$\begin{aligned}
TIME &= \frac{10 \text{ ns} \cdot (2^{28} - 1) \cdot (2^{15} - 1)}{10^{-9}} = 8,79 \cdot 10^4 \text{ Sekunden} \\
TIME &= 24,43 \text{ Stunden}
\end{aligned}$$

Um die Werte auszulesen, wurde ein SPI-Interface implementiert und mit einer Labview-GUI können die Werte gespeichert oder auch diverse Einstellungen vorgenommen werden.

Einstellbar sind die Startwerte, Sequenzlänge und Treshold. Zusätzlich dazu ist es möglich die aktuellen Werte abzufragen.

Mit Hilfe dieser Einstellungen ist es möglich, die Berechnung an einem bestimmten Punkt fortzusetzen, falls die Berechnung aus irgendeinem Grund abgebrochen wurde.

### A.4.2 Resultate

Für die Verarbeitung und Berechnungen der einzelnen Sequenzen, werden folgende Laufzeiten benötigt:

- Laufzeit für *data* Berechnung= 1 Taktzyklus
- Laufzeit für *polarity* Berechnung= 1 Taktzyklus
- Laufzeit für  $A_p$  Berechnung = 5 Taktzyklen (Pipelining)
- Laufzeit für die Ausgabe über SPI  $\approx 10$  ms (Rechnerabhängig)  
Diese Laufzeit kann vernachlässigt werden, da nur ein neuer Wert ausgegeben wird, wenn dieser eine bessere  $A_p$  als der letzte Wert hat. Somit gibt es nur maximal 60 Ausgaben.

Dieses VHDL-Programm wurde für jede Sequenzlänge zwischen 1 und 31 erstellt und ausgeführt. In Kapitel 3 Tabelle 3.5 sind die Resultate zusammengefasst abgebildet. Im Anhang A.5 sind detailliertere Informationen und Beispielsequenzen angegeben.

Da ternäre Sequenzen häufig in der Nachrichtentechnik verwendet werden, sind diese Resultate auch für andere Anwendungen wichtig. Außerdem konnten mit demselben Programm auch alle optimalen binären Sequenzen ermittelt werden, siehe dazu A.5.

## A.5 Optimale Preamble-Sequenzen

In diesem Abschnitt sind die Ergebnisse aus der Brute-force Berechnung der Korrelationssequenzen mit verschiedenen Längen  $n$  von 2 bis 31 angegeben. Zuerst werden die optimalen Werte für binäre Sequenzen in Tabelle A.2 angegeben. Für  $n = 2, 3, 4, 5, 7, 11, 13$  entsprechen die Sequenzen den Barker-Codes<sup>4</sup>. Daraufgehend werden dann alle Ergebnisse in den Tabellen A.2-A.7 für ternäre Sequenzen, sortiert nach  $n$  und  $k$ , angegeben. Für jede Kombination von  $n$  und  $k$  wird eine Beispielsequenz, welche eine maximale  $A_p$  aufweist, angegeben.

**Tabelle A.2:** Autokorrelationsgüte für optimale binäre Sequenzen

n	Code sequence	$A_p$	n	Code sequence	$A_p$
2	01	-	17	0 0000 0100 1101 0111	8
3	001	-	18	00 0000 0011 0010 1011	8
4	0001	2	19	000 0101 0111 1001 0011	10
5	0 0001	2	20	0000 0010 1011 0110 0111	10
6	00 0001	2	21	0 0000 0010 1001 1001 0111	10
7	000 1011	4	22	00 0000 0010 0110 1010 0111	10
8	0000 1011	4	23	000 0010 1001 1001 1010 1111	12
9	0 0000 1011	4	24	0000 0001 0101 1001 1010 0111	12
10	00 0000 1011	4	25	0 0000 0001 1010 0110 1010 0111	12
11	000 1001 0111	6	26	00 0000 0001 0101 1001 1010 0111	12
12	0000 0110 1011	6	27	000 0000 0001 0101 1001 1010 0111	12
13	0 0000 0101 0011	6	28	0000 0001 1001 0110 1010 1100 1111	14
14	00 0000 0101 0011	6	29	0 0000 0001 0101 1000 1101 1001 0111	14
15	000 0101 0011 0111	8	30	00 0000 0001 0011 0100 0111 0010 1011	14
16	0000 0100 1110 1011	8	31	000 0010 0011 1010 1001 0111 1001 1011	16

<sup>4</sup>Der Barker-Code ist ein in der Nachrichtentechnik verwendeter Binär-Code mit minimaler Autokorrelation, welcher im Bereich der Synchronisation und Radartechnik Verwendung findet

Tabelle A.3: Autokorrelationsgüte für optimale ternäre Sequenzen der Länge 2-10

$n = 2$		
k	Code sequence	$A_p$
1	0+	2
2	+ -	4
$n = 3$		
k	Code sequence	$A_p$
1	00+	2
2	0+-	4
3	++-	6
$n = 4$		
k	Code sequence	$A_p$
1	000+	2
2	00+-	4
3	0++-	4
4	+++ -	4
$n = 5$		
k	Code sequence	$A_p$
1	0 000+	2
2	0 00+-	4
3	0 0++-	4
4	0 +- -+	6
5	+ +++-	4
$n = 6$		
k	Code sequence	$A_p$
1	00 000+	2
2	00 00+-	4
3	00 0++-	4
4	0+ 0+--	6
5	0+ +-+-	6
6	++ +++-	4
$n = 7$		
k	Code sequence	$A_p$
1	000 000+	2
2	000 00+-	4
3	000 0++-	4
4	000 ++-+	6
5	00+ ++-+	6
6	0++ -+--	8
7	+++ -+--	8

$n = 8$		
k	Code sequence	$A_p$
1	0000 000+	2
2	0000 00+-	4
3	0000 0++-	4
4	0000 ++-+	6
5	000+ ++-+	6
6	00++ -+--	8
7	0+++ -+--	8
8	++++ -+--	8
$n = 9$		
k	Code sequence	$A_p$
1	0 0000 000+	2
2	0 0000 00+-	4
3	0 0000 0++-	4
4	0 0000 ++-+	6
5	0 00+0 +- -+	8
6	0 00++ -+--	8
7	0 0+++ -+--	8
8	0 ++++ -+--	8
9	+ ++++ -+--	8
$n = 10$		
k	Code sequence	$A_p$
1	00 0000 000+	2
2	00 0000 00+-	4
3	00 0000 0++-	4
4	00 0000 ++-+	6
5	00 00+0 -+--	8
6	00 00++ -+--	8
7	00 0++- -+--	10
8	00 +- -+ -+--	10
9	0+ +- -+ -+--	10
10	++ ++++ -+--	8

Tabelle A.4: Autokorrelationsgüte für optimale ternäre Sequenzen der Länge 11-15

$n = 11$		
k	Code sequence	$A_p$
1	000 0000 000+	2
2	000 0000 00+-	4
3	000 0000 0+-+	4
4	000 0000 +-+-	6
5	000 00+0 +--+	8
6	000 00++ -+--	8
7	000 0+-+ -+-+	10
8	000 +++- +--+	10
9	00+ +++- -+-+	10
10	0+- +++- -+--	12
11	+++ -+-+ +---	12
$n = 12$		
k	Code sequence	$A_p$
1	0000 0000 000+	2
2	0000 0000 00+-	4
3	0000 0000 0+-+	4
4	0000 0000 +-+-	6
5	0000 00+0 +--+	8
6	0000 00++ -+--	8
7	0000 0+-+ -+-+	10
8	0000 +++- +--+	10
9	00+0 -+++ -+--	12
10	00++ -+++ -+--	12
11	0+++ -+-+ +---	12
12	++++ +-+ +---	12
$n = 13$		
k	Code sequence	$A_p$
1	0 0000 0000 000+	2
2	0 0000 0000 00+-	4
3	0 0000 0000 0+-+	4
4	0 0000 0000 +-+-	6
5	0 0000 00+0 +--+	8
6	0 0000 +0-- 0+-+	10
7	0 0000 0+-+ -+-+	10
8	0 0000 +-+- +--+	12
9	0 000+ +-+- +--+	12
10	0 00++ -+++ -+--	12
11	0 0+++ +-+- -+--	12
12	0 +-++ -+++ +--+	14
13	+ +++++ +-+ +---	12

$n = 14$		
k	Code sequence	$A_p$
1	00 0000 0000 000+	2
2	00 0000 0000 00+-	4
3	00 0000 0000 0+-+	4
4	00 0000 0000 +-+-	6
5	00 0000 00+0 +--+	8
6	00 0000 00++ -+--	8
7	00 0000 0+-+ -+-+	10
8	00 000+ 0-- +--+	12
9	00 000+ +-+- +--+	12
10	00 00++ +-+- +--+	12
11	00 0+-+ +--- +--+	14
12	00 +++- +-+ -+--	14
13	0+ +-+ -+-- +--+	14
14	++ +++++ +-+ +---	12
$n = 15$		
k	Code sequence	$A_p$
1	000 0000 0000 000+	2
2	000 0000 0000 00+-	4
3	000 0000 0000 0+-+	4
4	000 0000 0000 +-+-	6
5	000 0000 00+0 +--+	8
6	000 0000 +0-- 0+-+	10
7	000 0000 0+-+ -+-+	10
8	000 000+ 0-- +--+	12
9	000 000+ +-+- +--+	12
10	000 +0+- 0-- -+--	14
11	000 0+-+ +-+- -+-+	14
12	000 +++++ -+- +--+	14
13	00+ +++++ -+-- -+-+	14
14	0++ +-+ +--- +--+	14
15	+++ +-+ +--+ +---	16

Tabelle A.5: Autokorrelationsgüte für optimale ternäre Sequenzen der Länge 16-19

$n = 16$		
k	Code sequence	$A_p$
1	0000 0000 0000 000+	2
2	0000 0000 0000 00+-	4
3	0000 0000 0000 0++-	4
4	0000 0000 0000 ++-+	6
5	0000 0000 00+0 +--+	8
6	0000 0000 +0-- 0+-+	10
7	0000 0000 0+-+ -+-+	10
8	0000 000+ 0--+ +-+-	12
9	0000 000+ +-+- ++--	12
10	0000 0+0- +--- +---	14
11	0000 0+++ --++ -+-+	14
12	00+- 0+-+ +0+- ----	16
13	00+0 +-+- -+-+ ----	16
14	00++ +-+- -+-+ ++--	16
15	0+++ +-++ ---- -+-+	16
16	++++ +-++ ---- -+-+	16
$n=17$		
k	Code sequence	$A_p$
1	0 0000 0000 0000 000+	2
2	0 0000 0000 0000 00+-	4
3	0 0000 0000 0000 0+-+	4
4	0 0000 0000 0000 ++-+	6
5	0 0000 0000 00+0 +--+	8
6	0 0000 0000 +0-- 0+-+	10
7	0 0000 0000 0+-+ -+-+	10
8	0 0000 000+ 0--+ +-+-	12
9	0 0000 0+0- 00++ --+-	12
10	0 0000 0+0- -+-+ --++	14
11	0 0000 0+++ -+-+ +---	14
12	0 000+ 0--++ -+-+ ++--	16
13	0 000+ ++-- +-+- -+-+	16
14	0 00++ +-+- -+-+ -+-+	16
15	0 0+++ +-+- -+-+ ++--	16
16	0 +-+- ---- +--- +-++	18
17	+ +++++ +-++ --+- +---	16

$n = 18$		
k	Code sequence	$A_p$
1	00 0000 0000 0000 000+	2
2	00 0000 0000 0000 00+-	4
3	00 0000 0000 0000 0+-+	4
4	00 0000 0000 0000 ++-+	6
5	00 0000 0000 00+0 +--+	8
6	00 0000 0000 +0-- 0+-+	10
7	00 0000 0000 0+-+ -+-+	10
8	00 0000 000+ 0--+ +-+-	12
9	00 0000 00+- +++- +---	14
10	00 0000 0+0- +--- +---	14
11	00 0000 0+++ -+-+ +---	14
12	00 0000 ++-- +-+- ++--	16
13	00 000+ ++-- +-+- -+-+	16
14	00 00++ +-+- -+-+ -+-+	16
15	00 0+-+ -+-+ ---- +-+-	18
16	00 +++- --+- --+- +-+-	18
17	0+ +++- --++ --+- -+-+	18
18	++ +++++ +-++ +-+- -+-+	16
$n = 19$		
k	Code sequence	$A_p$
1	000 0000 0000 0000 000+	2
2	000 0000 0000 0000 00+-	4
3	000 0000 0000 0000 0+-+	4
4	000 0000 0000 0000 ++-+	6
5	000 0000 0000 00+0 +--+	8
6	000 0000 0000 +0-- 0+-+	10
7	000 0000 0+0- -+0+ 00+-	10
8	000 0000 000+ 0--+ +-+-	12
9	000 0000 0+0- 00++ --+-	14
10	000 0000 0+0- +--- +---	14
11	000 0000 +-+- 00++ --+-	14
12	000 0000 ++-- +-+- ++--	16
13	000 +0-+ 0--+ -0-+ +++-	16
14	000 00++ -+-- -+-+ -+-+	16
15	000 0+++ -+-- ++-- +-+-	18
16	000 +++++ -+-+ +--- +-+-	18
17	00+ +++++ -+-- ++-- +-+-	18
18	0+- -+++ +-+- +--- -+-+	16
19	+++ +-+- +--- -+-+ ++--	16

Tabelle A.6: Autokorrelationsgüte für optimale ternäre Sequenzen der Länge 20-22

$n = 20$		
k	Code sequence	$A_p$
1	0000 0000 0000 0000 000+	2
2	0000 0000 0000 0000 00+-	4
3	0000 0000 0000 0000 0++-	4
4	0000 0000 0000 0000 ++-+	6
5	0000 0000 0000 00+0 +- -+	8
6	0000 0000 0000 +0-- 0+-+	10
7	0000 0000 0+0- -+0+ 00+-	10
8	0000 0000 000+ 0--+ +-+-	12
9	0000 0000 0+-0 -0++ --+-	14
10	0000 0000 0+0- +-++ +- -+	14
11	0000 000+ 0-+- --+0 +- -+	14
12	0000 0000 +-+- +-+- +- -+	16
13	0000 00++ --0- +-+- +- -+	16
14	0000 0+0+ -+++ --+- +- -+	16
15	0000 0+++ +- -+ +-+- +- -+	18
16	0000 ++++ +-+- +- -+ +- -+	18
17	00+0 -+++ +-+- +- -+ +- -+	18
18	00++ -+++ +-+- +- -+ +- -+	16
19	0+++ +-+- +- -+ +- -+ +- -+	16
20	++++ +-+- +- -+ +- -+ +- -+	16
$n = 21$		
k	Code sequence	$A_p$
1	0 0000 0000 0000 0000 000+	2
2	0 0000 0000 0000 0000 00+-	4
3	0 0000 0000 0000 0000 0++-	4
4	0 0000 0000 0000 0000 ++-+	6
5	0 0000 0000 0000 00+0 +- -+	8
6	0 0000 0000 0000 +0-- 0+-+	10
7	0 0000 0000 0+0- -+0+ 00+-	12
8	0 0000 0000 000+ 0--+ +-+-	12
9	0 0000 0000 0+-0 -0++ --+-	14
10	0 0000 000+ +-0- -0++ 0-+-	16
11	0 0000 0000 +-+- -0++ --+-	16
12	0 0000 0000 +-+- +-+- +- -+	16
13	0 0000 00+- +-0- +-+- --++	18
14	0 0000 00++ +- -+ +- -+ +- -+	18
15	0 000+ 0-+- 0-+- -+++ +- -+	20
16	0 0000 +-+- +++- --++ +- -+	20
17	0 000+ +++- +- -+ +- -+ +- -+	20
18	0 00++ +-+- +- -+ +- -+ +- -+	20
19	0 0+++ +++- +- -+ +- -+ +- -+	20
20	0 +++++ +-+- +- -+ +- -+ +- -+	20
21	+ +++++ +-+- +- -+ +- -+ +- -+	20

$n = 22$		
k	Code sequence	$A_p$
1	00 0000 0000 0000 0000 000+	2
2	00 0000 0000 0000 0000 00+-	4
3	00 0000 0000 0000 0000 0++-	4
4	00 0000 0000 0000 0000 ++-+	6
5	00 0000 0000 0000 00+0 +- -+	8
6	00 0000 0000 0000 +0-- 0+-+	10
7	00 0000 0000 0+0- -+0+ 00+-	12
8	00 0000 0000 000+ 0--+ +-+-	12
9	00 0000 0000 0+-0 -0++ --+-	14
10	00 0000 000+ +-0- -0++ 0-+-	16
11	00 0000 0000 +-+- -0++ --+-	16
12	00 0000 0+0- -0+0 +-+- +- -+	18
13	00 0000 00+- 00-- +-+- +- -+	18
14	00 00+0 +0-0 +- -+ +- -+ +- -+	20
15	00 0000 +-0- +- -+ +- -+ +- -+	20
16	00 0000 +- -+ +- -+ +- -+ +- -+	20
17	00 000+ +++- +- -+ +- -+ +- -+	20
18	00 +0+0 -+++ - -+ - -+ - -+ - -+	22
19	00 0+++ +- -+ +- -+ +- -+ +- -+	22
20	00 +++- +- -+ +- -+ +- -+ +- -+	22
21	0+ +++- +- -+ +- -+ +- -+ +- -+	22
22	++ +++++ +- -+ +- -+ +- -+ +- -+	20

Tabelle A.7: Autokorrelationsgüte für optimale ternäre Sequenzen der Länge 23-24

$n = 23$		
k	Code sequence	$A_p$
1	000 0000 0000 0000 0000 000+	2
2	000 0000 0000 0000 0000 00+-	4
3	000 0000 0000 0000 0000 0++-	4
4	000 0000 0000 0000 0000 ++-+	6
5	000 0000 0000 0000 00+0 +- -+	8
6	000 0000 0000 0000 +0-- 0+-+	10
7	000 0000 0000 0+0- -+0+ 00+-	12
8	000 0000 0000 000+ 0-- +--+	12
9	000 0000 0000 0+0- -0++ --+-	14
10	000 0000 000+ -+0- -0++ 0--+	16
11	000 0000 0000 +-+- -0++ --+-	16
12	000 0000 0+0- -0+0 +-+- -++-	18
13	000 0000 00+- 00-- +++- +- -+	18
14	000 00+0 +0-0 +- -+ +- -0 --+-	20
15	000 0000 +-0- +- -+ +- -+ --+-	20
16	000 0000 +- -+ +- -+ +- -+ --+-	20
17	000 000+ +++- +- -+ +- -+ +- -+ --+-	20
18	000 +0+0 -+++ - - -+ - - -+ - - -+ --	22
19	000 0+++ -+-+ - - -+ - - -+ - - -+ --	22
20	000 +++- +++- +- -+ +- -+ +- -+ --	22
21	00+ +++- -+-+ +- -+ +++- +- -+ --	22
22	0++ +++++ +- -+ +- -+ +- -+ +- -+ --	20
23	0++ +++++ +- -+ +- -+ +- -+ +- -+ --	20

$n = 24$		
k	Code sequence	$A_p$
1	0000 0000 0000 0000 0000 000+	2
2	0000 0000 0000 0000 0000 00+-	4
3	0000 0000 0000 0000 0000 0++-	4
4	0000 0000 0000 0000 0000 ++-+	6
5	0000 0000 0000 0000 00+0 +- -+	8
6	0000 0000 0000 0000 +0-- 0+-+	10
7	0000 0000 0000 0+0- -+0+ 00+-	12
8	0000 0000 0000 000+ 0-- +--+	12
9	0000 0000 0000 0+0- -0++ --+-	14
10	0000 0000 000+ -+0- -0++ 0--+	16
11	0000 0000 0000 +-+- -0++ --+-	16
12	0000 0000 0+0- -0+0 +-+- -++-	18
13	0000 0000 00+- 00-- +++- +- -+	18
14	0000 00+0 +0-0 +- -+ +- -0 --+-	20
15	0000 0000 +-0- +- -+ +- -+ --+-	20
16	0000 0000 +- -+ +- -+ +- -+ --+-	20
17	0000 000+ +++- +- -+ +- -+ +- -+ --+-	20
18	0000 +0+0 -+++ - - -+ - - -+ - - -+ --	22
19	0000 0+++ -+-+ - - -+ - - -+ - - -+ --	22
20	0000 +++- +++- +- -+ +- -+ +- -+ --	22
21	000+ +++- -+-+ +- -+ +++- +- -+ --	22
22	00++ +++++ +- -+ +- -+ +- -+ +- -+ --	20
23	00++ +++++ +- -+ +- -+ +- -+ +- -+ --	20
24	00++ +++++ +- -+ +- -+ +- -+ +- -+ --	20



Tabelle A.8: Autokorrelationsgüte für optimale ternäre Sequenzen der Länge 25-26

$n = 25$		
k	Code sequence	$A_p$
1	0 0000 0000 0000 0000 0000 000+	2
2	0 0000 0000 0000 0000 0000 00+-	4
3	0 0000 0000 0000 0000 0000 0+-+	4
4	0 0000 0000 0000 0000 0000 +-+-	6
5	0 0000 0000 0000 0000 00+0 +--+	8
6	0 0000 0000 0000 0000 +0-- 0+-+	10
7	0 0000 0000 0000 0+0- -+0+ 00+-	12
8	0 0000 0000 0000 000+ 0--+	12
9	0 0000 0000 0000 0+-0 -0++ --+-	14
10	0 0000 0000 000+ --0+ 00+- -+-+	16
11	0 0000 0000 0000 +-+- -0++ --+-	16
12	0 0000 0000 00+- -++0 -0-+ -+-+	18
13	0 0000 000+ -+0- 0+-+ -00- +-++	20
14	0 0000 0000 +-+0 -+-+ +0+- -+-+	20
15	0 0000 0000 +-0- +-++ -+-+ --++	20
16	0 0000 00+- 0-+- 0+++ --+- +-+-	22
17	0 0000 00+0 --+- ++++ -+-+ --++	22
18	0 00+0 -+++ -00- -+-0 --+- ++++	24
19	0 0000 +0+- --+- ++++ ---+ +-+-	24
20	0 0000 +++- -+-+ -+-+ -+-+ -+-+	24
21	0 000+ +++- +++- +-+- +-+- +-+-	24
22	0 00++ +++- +++- +-+- +-+- +-+-	24
23	0 0+++ +++- +-+- -+-+ -+-+ -+-+	24
24	0 ++++ +++- -+-+ -+-+ -+-+ -+-+	24
25	+ ++++ +++- -+-+ -+-+ -+-+ -+-+	24

$n = 26$		
k	Code sequence	$A_p$
1	00 0000 0000 0000 0000 0000 000+	2
2	00 0000 0000 0000 0000 0000 00+-	4
3	00 0000 0000 0000 0000 0000 0+-+	4
4	00 0000 0000 0000 0000 0000 +-+-	6
5	00 0000 0000 0000 0000 00+0 +--+	8
6	00 0000 0000 0000 0000 +0-- 0+-+	10
7	00 0000 0000 0000 0+0- -+0+ 00+-	12
8	00 0000 000+ 0000 +-+0 -00+ 00+-	14
9	00 0000 0000 0000 0+-0 -0++ --+-	14
10	00 0000 0000 000+ --0+ 00+- -+-+	16
11	00 0000 000+ 0-0+ 0-+- 00-0 +++-	18
12	00 0000 0000 00+- -++0 -0-+ -+-+	18
13	00 0000 000+ -0-+ 0+-+ 0-0+ --++	20
14	00 0000 0000 +-+0 -+-+ 0-+- -+-+	20
15	00 0000 0+-0 ++0- +-+0 -+++ -0-+	22
16	00 0000 00+0 -+-+ --0+ -+-+ --++	22
17	00 0000 00+0 -+-+ ---+ +-+- +-+-	22
18	00 0000 +0-- +-++ --0- +-++ +-+-	24
19	00 0000 0+-+ -+-+ +--- +-+- +-+-	24
20	00 0000 +++- +-+- -+-+ -+-+ -+-+	24
21	00 000+ +++- +-+- -+++ --++ +-+-	24
22	00 0+-+ +-++ +0-- -+-+ +-+- +-+-	26
23	00 +0++ -+-+ +--- -+-+ +++- +-+-	26
24	00 ++++ -+-+ -+-+ +--- +++- -+-+	26
25	0+ ++++ -+-+ -+-+ -+-+ +--- -+-+	26
26	++ ++++ +++- +-+- -+-+ -+-+ +---	24

Tabelle A.9: Autokorrelationsgüte für optimale ternäre Sequenzen der Länge 27-28

$n = 27$		
k	Code sequence	$A_p$
1	000 0000 0000 0000 0000 0000 000+	2
2	000 0000 0000 0000 0000 0000 00+-	4
3	000 0000 0000 0000 0000 0000 0++-	4
4	000 0000 0000 0000 0000 0000 ++-+	6
5	000 0000 0000 0000 0000 00+0 +--+	8
6	000 0000 0000 0000 0000 +0-- 0+-+	10
7	000 0000 0000 0000 0+0- -+0+ 00+-	12
8	000 0000 0000 0+0- -0+0 0+00 0+-+	14
9	000 0000 0000 0000 0+0- -0++ --+-	14
10	000 0000 0000 000+ --0+ 00-+ -++-	16
11	000 0000 0000 +-0+ --00 ++00 -+-+	18
12	000 0000 0000 00+- -++0 -0-+ -++-	18
13	000 0000 000+ -0+- -00+ +-0- +---	20
14	000 0000 0000 +-0- +-+0 -+-+ +---	20
15	000 0000 00+- +0-- +-+0 0+-- -++-	22
16	000 0000 00+0 -+-+ --0+ -+++ -+++	22
17	000 0000 +-+- 0-++ -+++ 0+0- -+++	24
18	000 0000 0+-- +-+- +0-- -+++ +-+-	24
19	000 0000 0+-- +-+- +++- -+++ -+++	24
20	000 0+00 -+++ +0+- -+++ +-+- -+++	26
21	000 00++ 0--+- +-+- +-+- -+++ -+++	26
22	000 00++ +++- +-+- +-+- -+++ -+++	26
23	000 0+++ +-+- +++- +-+- +-+- -+++	26
24	000 +++++ +-+- -+++ -+-+ -+++ +-+-	26
25	00+ +++++ +-+- -+++ +++- -+++ +-+-	26
26	00+ +++++ +-+- -+++ +++- -+++ +-+-	26
27	00+ +++++ +-+- -+++ +++- -+++ +-+-	24

$n = 28$		
k	Code sequence	$A_p$
1	0000 0000 0000 0000 0000 0000 000+	2
2	0000 0000 0000 0000 0000 0000 00+-	4
3	0000 0000 0000 0000 0000 0000 0++-	4
4	0000 0000 0000 0000 0000 0000 ++-+	6
5	0000 0000 0000 0000 0000 00+0 +--+	8
6	0000 0000 0000 0000 0000 +0-- 0+-+	10
7	0000 0000 0000 0000 0+0- -+0+ 00+-	12
8	0000 0000 0000 0+0- 0-+0 +000 +0--	14
9	0000 0000 0000 0000 0+0- -0++ --+-	14
10	0000 0000 0000 000+ --0+ 00-+ -++-	16
11	0000 0000 0000 ++-0 -0+0 +0-+ 0--+	18
12	0000 0000 0000 00+- -++0 -0-+ -++-	18
13	0000 0000 000+ 0-+- 0-++ 00-- +---	20
14	0000 0000 0000 +-0- +-+0 -+-+ +---	20
15	0000 0000 00+- -0++ 0+-- 0+-- -+++	22
16	0000 0000 00+0 -+-+ --0+ -+++ -+++	22
17	0000 0000 +0-+ -+-+ +-0+ --0+ +---	24
18	0000 0000 0+-- +-0+ +-+- -+++ +---	24
19	0000 0+00 -+++ +0++ +-+- 0-+- 0--+	26
20	0000 000+ -+++ 0-+- ++++ -+++ +---	26
21	0000 000+ +++- +-+- +-+- -+++ +---	26
22	0000 00++ +++- +-+- +-+- -+++ +---	26
23	0000 0+++ +++- +-+- +-+- -+++ +---	26
24	000+ +-0- +-+- +-+- -+++ -+++ +---	28
25	00+0 -+-+ +++- +-+- +-+- -+++ +---	28
26	00++ +-+- +++- +-+- +-+- -+++ +---	28
27	0+++ +-+- +-+- +-+- +-+- -+++ +---	28
28	++++ +++- +-+- +-+- +-+- -+++ +---	28

Tabelle A.10: Autokorrelationsgüte für optimale ternäre Sequenzen der Länge 29-30

$n = 29$			$n = 30$		
k	Code sequence	$A_p$	k	Code sequence	$A_p$
1	0 0000 0000 0000 0000 0000 0000 000+	2	1	00 0000 0000 0000 0000 0000 0000 000+	2
2	0 0000 0000 0000 0000 0000 0000 00+-	4	2	00 0000 0000 0000 0000 0000 0000 00+-	4
3	0 0000 0000 0000 0000 0000 0000 0++-	4	3	00 0000 0000 0000 0000 0000 0000 0++-	4
4	0 0000 0000 0000 0000 0000 0000 ++-+	6	4	00 0000 0000 0000 0000 0000 0000 ++-+	6
5	0 0000 0000 0000 0000 0000 00+0 +--+	8	5	00 0000 0000 0000 0000 0000 00+0 +--+	8
6	0 0000 0000 0000 0000 0000 +0-- 0+-+	10	6	00 0000 0000 0000 0000 0000 +0-- 0+-+	10
7	0 0000 0000 0000 0000 0+0- -+0+ 00+-	12	7	00 0000 0000 0000 0000 0+0- -+0+ 00+-	12
8	0 0000 0000 0000 0+0- -0+0 0+00 0+-+	14	8	00 0000 0000 0000 0+0- -0+0 0+00 0+-+	14
9	0 0000 0000 0000 0000 0+-0 -0++ --+-	14	9	00 0000 0000 0000 0000 0+-0 -0++ --+-	14
10	0 0000 0000 0000 000+ --0+ 00-+ -++-	16	10	00 0000 0000 0000 000+ --0+ 00-+ -++-	16
11	0 0000 0000 0000 +-0+ --00 ++00 -+-+	18	11	00 0000 0000 0000 +-0+ --00 ++00 -+-+	18
12	0 0000 0000 0000 00+- -++0 -0-+ -++-	18	12	00 0000 0000 0000 00+- -++0 -0-+ -++-	18
13	0 0000 0000 000+ +0-0 -+-+ 00+- -++-	20	13	00 0000 0000 000+ 0-+- 0-++ 00-- -+++	20
14	0 0000 0000 +0-0 ++-- 0-+0 +00- -++-	22	14	00 0000 0000 +0-0 -+-+ ++00 0--+ 0+-+	22
15	0 0000 0000 00+- -0++ 0+-+ 0+-+ -++-	22	15	00 0000 0000 00+- -0++ 0+-+ 0+-+ -++-	22
16	0 0000 00+0 --+0 00+- -0-+ 0-+- +++-	24	16	00 0000 000+ -0-- ++0+ -0-0 -+-+ 0++-	24
17	0 0000 0000 +0-+ -+-+ +-0+ --0+ +++-	24	17	00 0000 0000 +0-- -+++ -0+- -+0- +-+-	24
18	0 0000 0000 0+-+ +-0+ +-+- +--+ +++-	24	18	00 0000 0000 0+-+ +-0+ +-+- +--+ +++-	24
19	0 0000 000+ -++ 0-+- +--0 +++- +--+	26	19	00 0000 000+ -+- 0+-+ +-0- +--+ +++-	26
20	0 0000 000+ +--- 0++- +-+- -+-+ +--+	26	20	00 0000 000+ -+0- --++ +-+- --++ -+-+	26
21	0 0000 000+ +-+- +-+- -+-+ -+-+ +--+	26	21	00 0000 +-+- 0+-+ +-+- +-0+ ++0- -+-+	28
22	0 000+ 0+0- -++ -+0- +-+- -+-+ -++-	28	22	00 0000 0+-+ -+0- +-+- +-+- -+-+ -++-	28
23	0 0000 +0-- ++++ -+-+ +-+- +--- +-+-	28	23	00 0000 0++- -+-+ +-+- +-+- -+-+ +-+-	28
24	0 0000 +++- -++ -+-+ -+-+ +-+- -++-	28	24	00 0000 ++++ -+-+ +-+- -+-+ +--- +-+-	28
25	0 000+ ++++ -+-+ +-+- -++ -+-+ +-+-	28	25	00 +0-- -+0- ++++ -+-+ -+0- -+-+ +---	30
26	0 00++ ++++ +-+- -+-+ -+-+ -+-+ +---	28	26	00 0+-+ +--- --+0 --++ -+-+ --++ +-+-	30
27	0 0+++ ++++ -++ -+-+ +-+- +--+ +---	28	27	00 +0-+ -+-+ ++++ +-+- +--- -+++ -+-+	30
28	0 +-+- ++++ +--- +--+ -+-+ -+-+ -+++ -+-+	30	28	00 +++- -+-+ -+-+ -+-+ -+++ +--- +-+-	30
29	+ ++++ ++++ +-+- -+++ -+-+ -+-+ +---	28	29	0+ +++- +-+- -++ -+-+ +--- -++ -+-+	30
			30	++ ++++ ++++ +-+- +-+- +--- +-+- -+-+	28

Tabelle A.11: Autokorrelationsgüte für optimale ternäre Sequenzen der Länge 31

$n = 30$		
k	Code sequence	$A_p$
1	000 0000 0000 0000 0000 0000 0000 000+	2
2	000 0000 0000 0000 0000 0000 0000 00+-	4
3	000 0000 0000 0000 0000 0000 0000 0++-	4
4	000 0000 0000 0000 0000 0000 0000 ++-+	6
5	000 0000 0000 0000 0000 0000 00+0 +--+	8
6	000 0000 0000 0000 0000 0000 +0-- 0+--	10
7	000 0000 0000 0000 0000 0+0- -+0+ 00+-	12
8	000 0000 0000 0000 0+0- -0+0 0+00 0+--	14
9	000 0000 0000 0000 0000 0+-0 -0++ --+-	14
10	000 0000 0000 0000 000+ --0+ 00-- -++-	16
11	000 0000 0000 0000 +-0+ --00 ++00 -++-	18
12	000 0000 0000 0+-0 -+0- 00++ 00-- 0+--	20
13	000 0000 0000 000+ 0+-+ 0-++ 00-- ++-+	20
14	000 0000 0000 +0-- 0+0+ --+0 0-0+ --++	22
15	000 0000 0000 00+- -0++ 0+-- 0+-- -++-	22
16	000 0000 000+ 0+-+ 0-0+ 0-++ -0-- +++-	24
17	000 0000 0000 +0+- -+0- -+++ --0+ -++-	24
18	000 0000 00+- +0-+ 0+-0 -+++ 0--+ -+++	26
19	000 0000 000+ -+-- 0+-+ +-0- +--+ ++--	26
20	000 000+ -0-0 0+-+ -+++ +0+- 0+-- -++-	28
21	000 0000 +0+- -++0 0--+ -+-+ +++- +--+	28
22	000 0000 0+-- 0++- +-+- +++- -+++ -++-	28
23	000 0000 0++- +++- +-+- +--- ++++ -+--	28
24	000 00+0 ---+ ---+ 0+-+ -+++ ---+ +++-	30
25	000 000+ +--+ ---+ -+-+ ---+ +--- +++-	30
26	000 00++ ++++ +-+- -+-+ ---+ -+++ ---+	30
27	000 0+++ ++++ -+-- +++- -+-- ++++ ---+	30
28	000 ++++ +++- +-+- ---+ ++++ ---+ ++++	30
29	00+ ++++ +-+- +++- ---+ +++- +++- +-+-	30
30	0++ -+++ ++++ ---+ -+-+ +++- ---+ -+--	32
31	+++ +-+- +++- -+-+ -++- +--- -++- -+--	32

## A.6 Optimale Bitraten

Für die Berechnung der Übertragungszeit wurde eine Nachrichtenlänge von 408 Bits und für die Synchronisation wurde  $n=15$  angenommen.

**Tabelle A.12:** Fehlerrate und Übertragungszeit Master→ Slave

1 Impuls												
$S_k$	$M \rightarrow S_k$			$M \rightarrow S_k$			GB	TRANS	IER			GT
	$n'_M$	$k_M$	BR	$n'_s$	$k'_S$	BR			%	M	S	
1	16	1	5	16	1	5	0.3125	5248	96.2	132	23	1142856
2	32	1	6	4	1	3	0.3	17408	86.1	148	7	1643096
3	32	1	6	2	1	2	0.3158	26112	82.4	155	3	1668575
2 Impulse												
1	33	2	11	33	2	11	0.3333	5016 ns	93	136	26	1145936
2	46	2	12	7	2	6	0.3750	8704 ns	85.8	149	6	1005750
3	46	2	12	4	2	4	0.3750	13056 ns	82	155	4	994015
4	46	2	12	4	1	3	0.3750	17408 ns	81.1	157	7	980425
5	24	2	10	2	1	2	0.3750	26112 ns	85.3	149	3	1048960
6	46	2	12	2	1	2	0.3750	26112 ns	76.2	167	3	1014859
3 Impulse												
1	48	3	17	48	3	17	0.3542	4608 ns	93.3	137		1098466
2	60	3	18	17	2	9	0.3750	8832 ns	86.6	147	11	1001658
3	60	3	18	7	2	6	0.3750	13056 ns	84.7	150	6	961950
4	94	3	20	4	3	5	0.3636	18040 ns	75.2	169	1	1082065
5	94	3	20	4	2	4	0.3509	23256 ns	70	182	4	1172662
6	60	3	18	4	1	3	0.3750	32096 ns	75.1	170	7	1202636
4 Impulse												
1	62	4	23	62	4	23	0.3594	4608 ns	92.9	137	16	1098466
2	102	4	26	13	4	13	0.4063	8192 ns	86.3	148	4	961112
3	121	4	27	7	4	9	0.3803	13064 ns	77.3	165	2	1058585
4	73	4	24	7	2	6	0.3636	17408 ns	81.5	156	6	974181
5	86	4	25	5	2	5	0.3509	20992 ns	78	159	4	952473
6	73	4	24	8	1	4	0.3750	26112 ns	77.4	165	14	1002705
5 Impulse												
1	68	5	28	68	5	28	0.3488	4800 ns	94	136	17	1116560
2	113	5	32	20	4	16	0.4000	8320 ns	84.3	150	8	983700
3	129	5	33	10	4	11	0.4125	12160 ns	80.8	158	5	966064
4	113	5	32	7	3	8	0.4000	16320 ns	77.5	164	3	979531
5	86	5	30	7	2	6	0.3750	21760 ns	80.2	159	6	976896
6	86	5	30	5	2	5	0.3750	26240 ns	77.9	164	4	1000127
6 Impulse												
1	88	6	23	88	6	23	0.3646	6912 ns	92.6	138	19	1424436
2	155	6	26	18	6	13	0.4167	12288 ns	84.5	150	2	1281300
3	138	6	27	13	4	9	0.4063	17664 ns	78.9	161	4	1279789
4	155	6	24	8	4	6	0.4160	26112 ns	75.1	170	5	1431528
5	155	6	25	7	3	5	0.4167	31488 ns	72.2	176	3	1423770
6	99	6	24	7	2	4	0.3750	39168 ns	77.2	165	6	1361745



# Literaturverzeichnis

- [1] P. D. P. Langendörfer, *TANDEM*, 2007. <http://www.tandem-projekt.de/>.
- [2] D. S. Olonbayar, *KUSZ*, 2012. <http://www.kusz-projekt.de/>.
- [3] D. Kreiser and S. Olonbayar, "Efficient synchronization method for ir-uwband 802.15. 4a non-coherent energy detection receiver," in *Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*, pp. 521–526, IEEE Computer Society, 2010.
- [4] D. Kreiser and S. Olonbayar, "Design and asic implementation of an ir-uwband baseband transceiver for ieee 802.15. 4a," in *Wireless Communications and Signal Processing (WCSP), 2011 International Conference on*, pp. 1–6, IEEE, 2011.
- [5] D. Kreiser and S. Olonbayar, "Improvements of ieee 802.15. 4a for non-coherent energy detection receiver," in *Signals, Systems, and Electronics (ISSSE), 2012 International Symposium on*, pp. 1–5, IEEE, 2012.
- [6] D. Kreiser, M. Brzozowski, G. Panic, S. Olonbayar, T. Basmer, and P. Langendörfer, "The all ihp sensor node: highly integrated sensor nodes using ihp components," in *Proceedings of the 10th ACM international symposium on Mobility management and wireless access*, pp. 147–148, ACM, 2012.
- [7] D. Kreiser and S. Olonbayar, "Improvements of ieee 802.15.4a for usage in automation systems," in *Energy Aware Computing Systems and Applications (ICEAC), 2013 4th Annual International Conference on*, pp. 131–134, IEEE, 2013.
- [8] D. Kreiser and S. Olonbayar, "Multiple sensor node synchronization for parallel uwband communication and ranging," in *The 2014 IEEE International Conference on Ultra-Wideband (ICUWB)*, IEEE, 2014.
- [9] D. Kreiser and S. Olonbayar, "Collision-free full-duplex uwband communication based on the standard ieee.802.15.4a," in *2015 IEEE Wireless Communications and Networking Conference (WCNC)*, IEEE, 2015.
- [10] D. Kreiser, D. Martynenko, O. Klymenko, and G. Fischer, "Simple and efficient localization method for ir-uwband systems based on two-way ranging," in *2015 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM), 2015*, IEEE, 2015.
- [11] D. Kreiser, "Simple and efficient two-way ranging method for ieee802.15.4a," in *1st URSI Atlantic Radio Science Conference (URSI AT-RASC), 2015*, URSI, 2015.

- [12] S. Olonbayar, D. Kreiser, and R. Kraemer, "Performance and design of ir-uwband transceiver baseband for wireless sensors," in *Ultra-Wideband, 2009. ICUWB 2009. IEEE International Conference on*, pp. 809–813, IEEE, 2009.
- [13] S. Olonbayar, G. Fischer, D. Kreiser, and R. Kraemer, "Automatic gain controlling in ir-uwband communications designed for wireless sensors," in *Ultra-Wideband (ICUWB), 2010 IEEE International Conference on*, vol. 1, pp. 1–4, IEEE, 2010.
- [14] S. Olonbayar, D. Kreiser, and R. Kraemer, "Ieee 802.15. 4a ir-uwband noncoherent receiver design and performance," IEEE, 2011.
- [15] S. Olonbayar, D. Kreiser, D. Martynenko, G. Fischer, O. Klymenko, and R. Kraemer, "Board implementation and its performance for ir-uwband ieee. 802.15. 4a from multiple asic chips," in *European Wireless, 2012. EW. 18th European Wireless Conference*, pp. 1–8, VDE, 2012.
- [16] D. Martynenko, G. Fischer, O. Klymenko, D. Kreiser, and S. Olonbayar, "High-band ultra-wideband transmitter for ieee 802.15. 4a standard," in *Ultra-Wideband (ICUWB), 2013 IEEE International Conference on*, pp. 165–169, IEEE, 2013.
- [17] G. Fischer, D. Martynenko, O. Klymenko, S. Olonbayar, D. Kreiser, J. Diggel, M. Masini, M. Grozing, and R. Kraemer, "Ir-uwband single-chip transceiver for high-band operation compliant to ieee 802.15. 4a," in *Ultra-Wideband (ICUWB), 2013 IEEE International Conference on*, pp. 270–277, IEEE, 2013.
- [18] S. Olonbayar, D. Kreiser, and R. Kraemer, "Performance and implementation of a multi-rate ir-uwband baseband transceiver for ieee802. 15.4 a," in *Ultra-Wideband (ICUWB), 2013 IEEE International Conference on*, pp. 231–237, IEEE, 2013.
- [19] G. Fischer, D. Martynenko, O. Klymenko, S. Olonbayar, and D. Kreiser, "Multi-channel and multi-data rate ir-uwband single-chip transceiver compliant to ieee 802.15.4a," in *The 2014 IEEE International Conference on Ultra-Wideband (ICUWB)*, IEEE, 2014.
- [20] S. Olonbayar, D. Kreiser, and R. Kraemer, "Fpga and asic implementation and testing of ir-uwband baseband transceiver for ieee802.15.4a," in *The 2014 IEEE International Conference on Ultra-Wideband (ICUWB)*, IEEE, 2014.
- [21] D. Martynenko, D. Kreiser, O. Klymenko, and G. Fischer, "Evaluation platform for verification of an ir-uwband ranging system," in *12th Workshop in Positioning, Navigation and Communication (WPNC'15)*, IEEE, 2015.
- [22] S. Olonbayar, D. Kreiser, R. Kraemer, G. Fischer, D. Martynenko, and O. Klymenko, "Polarity reception for ir-uwband in wireless fading channel," in *European Wireless EW2015, 2015*, IEEE, 2015.



- [23] IEEE, *IEEE 802.11<sup>TM</sup>: Wireless LANs*, August 2007. <http://standards.ieee.org/about/get/802/802.11.html>.
- [24] IEEE, *IEEE 802.15.1 - 2005*. [http://standards.ieee.org/getieee802/download/802.15.1-2005\\_part1.pdf](http://standards.ieee.org/getieee802/download/802.15.1-2005_part1.pdf).
- [25] IEEE, *IEEE 802.15.1 - 2002 specification*. <http://standards.ieee.org/findstds/standard/802.15.1-2002.html>.
- [26] UMTS. <http://www.etsi.org/technologies-clusters/technologies/mobile/umts>.
- [27] LTE. <http://www.3gpp.org/technologies/keywords-acronyms/98-lte>.
- [28] Fraunhofer, "Pressemitteilung November 2013," [http://www.ipa.fraunhofer.de/Innovationspreis\\_der\\_Automatisierungsindustrie\\_2013.2634.0.html](http://www.ipa.fraunhofer.de/Innovationspreis_der_Automatisierungsindustrie_2013.2634.0.html).
- [29] "Automatisierungsindustrie überwindet Krise im Rekordtempo," <http://www.produktion.de/unternehmen-maerkte/automatisierungsindustrie-ueberwindet-krise-im-rekordtempo/>.
- [30] "Produktion - Automatisierungsindustrie legt bei Umsatz und Mitarbeitern zu," <http://www.ingenieur.de/Themen/Produktion/Automatisierungsindustrie-legt-Umsatz-Mitarbeitern-zu>.
- [31] Frost and Sullivan, "Industrial Automation and Process Control," <http://www.frost.com/prod/servlet/press-release.pag?docid=268331746>.
- [32] D. F. für Künstliche Intelligenz (DFKI) GmbH. <http://www.dfki.de/>.
- [33] V. D. Ingenieure. <http://www.vdi.de/>.
- [34] V. GMA. [http://www.vde.com/de/fg/GMA/Seiten/default\\_0.aspx](http://www.vde.com/de/fg/GMA/Seiten/default_0.aspx).
- [35] V. G. für Mess-und Automatisierungstechnik, "Automation 2020," Juni 2009. [https://www.vdi.de/fileadmin/vdi\\_de/redakteur\\_dateien/gma\\_dateien/AT\\_2020\\_INTERNET.pdf](https://www.vdi.de/fileadmin/vdi_de/redakteur_dateien/gma_dateien/AT_2020_INTERNET.pdf).
- [36] M. Masini, *Störungsunempfindliche Informationsübertragung in dynamischer Produktionsumgebung* -. Norderstedt: BoD – Books on Demand, 2015.
- [37] IEEE, *IEEE 802.15.4a specification*, August 2007.
- [38] FESTO. <http://www.festo.de/>.
- [39] WAGO. <http://www.wago.de/>.
- [40] J. E. A. C. R. Richard Steigmann ABB STOTZ-KONTAKT GmbH, "Introduction to wisa wisa - wireless interface for sensors and actuators," 2006.

- [41] F. Nekoogar, *Ultra-Wideband Communications - Fundamentals and Applications*. New Jersey: Prentice Hall PTR, 2011.
- [42] G. S. M. C. Safety, *Exposition durch Ultra-Wideband Technologien*, 2008. [http://www.emf-forschungsprogramm.de/home/forschung/dosimetrie/dosimetrie\\_abges/dosi\\_092\\_Praesentationen.pdf](http://www.emf-forschungsprogramm.de/home/forschung/dosimetrie/dosimetrie_abges/dosi_092_Praesentationen.pdf).
- [43] H.-U. Dehner, *Interferenzuntersuchungen für inkohärente Multiband-Ultra-Breitband(UWB)-Übertragung* -. Inst. für Nachrichtentechnik, 2012.
- [44] G. Adamiuk, *Methoden zur Realisierung von dual-orthogonal, linear polarisierten Antennen für die UWB-Technik* -. Karlsruhe: KIT Scientific Publishing, 2010.
- [45] H. Nikookar and R. Prasad, *Introduction to Ultra Wideband for Wireless Communications* -. Berlin, Heidelberg: Springer, 2008.
- [46] F. C. Commission, *New public safety applications and broadband internet access among uses envisioned by fcc authorisation of ultrawideband technology*. FCC, 2002.
- [47] M.-G. D. Benedetto and G. Giancola, *Understanding Ultra Wide Band Radio Fundamentals* -. Amsterdam: Pearson Education, 2004.
- [48] I. P. W. G. for Wireless Personal Area Networks (WPANs) [http://grouper.ieee.org/groups/802/15/pub/2002/Jul02/02255r1P802-15\\_SG3a-July02-Minutes.doc](http://grouper.ieee.org/groups/802/15/pub/2002/Jul02/02255r1P802-15_SG3a-July02-Minutes.doc).
- [49] *The 2014 IEEE International Conference on Ultra-Wideband (ICUWB 2014)*. <http://www.icuwb2014.org/>.
- [50] V. Niemela, M. Hamalainen, and J. Iinatti, "Improved usage of time slots of the ieee 802.15. 4a uwb system model," in *Wireless Personal Multimedia Communications (WPMC), 2011 14th International Symposium on*, pp. 1–5, IEEE, 2011.
- [51] I. Dotlic and R. Miura, "Ieee 802.15. 4a-based impulse radio ultra-wideband system with joint channel equalization and decoding," in *Ultra-WideBand (ICUWB), 2014 IEEE International Conference on*, pp. 234–239, IEEE, 2014.
- [52] C. Jing, I. Jeon, and Y. Kim, "Enhanced phy symbol structure based on the ieee 802.15. 4a standardization and orthogonal pulses for 3d positioning," *3D Research*, vol. 2, no. 4, pp. 1–6, 2011.
- [53] J. Hund, C. Schwingenschlogl, and R. Kraemer, "Extension of ieee 802.15. 4a to improve resilience for wireless automation using soft-bit combination," in *Ultra-Wideband (ICUWB), 2011 IEEE International Conference on*, pp. 605–609, IEEE, 2011.

- [54] T. Paso, V. Niemela, J. Haapola, and J. Iinatti, "Novel modulation adaptation techniques for ieee 802.15. 4a uwb system," in *Medical Information and Communication Technology (ISMICT), 2012 6th International Symposium on*, pp. 1–4, IEEE, 2012.
- [55] T. N. Duong, M. T. Hoang, M. H. Tran, and Q. H. Dang, "A practical synchronization algorithm for ieee 802.15. 4a uwb receivers," in *Advanced Technologies for Communications (ATC), 2013 International Conference on*, pp. 170–175, IEEE, 2013.
- [56] H. Karvonen, C. Pomalaza-Ráez, and M. Hämäläinen, "A cross-layer optimization approach for lower layers of the protocol stack in sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 11, no. 1, p. 16, 2014.
- [57] P. Losco, S. Bourdel, J. Gaubert, N. Dehaese, S. Meillere, O. Ramos, R. Vauche, and H. Barthelemy, "Analysis of the ieee 802.15. 4a uwb phy layer for optimizing the power consumption of the transmitter," in *Ultra-Wideband (ICUWB), 2013 IEEE International Conference on*, pp. 189–194, IEEE, 2013.
- [58] M. Flury, R. Merz, and J.-Y. Le Boudec, "Synchronization for impulse-radio uwb with energy-detection and multi-user interference: Algorithms and application to ieee 802.15. 4a," *Signal Processing, IEEE Transactions on*, vol. 59, no. 11, pp. 5458–5472, 2011.
- [59] J. Hund, M. Bahr, C. Schwingenschlögl, R. Kraemer, and S. Olonbayar, "Exploiting short mac superframe cycles for fast bit synchronization in ieee 802.15. 4 uwb-ir," in *Multiple Access Communications*, pp. 158–169, Springer, 2012.
- [60] J.-Y. Le Boudec, R. Merz, B. Radunovic, and J. Widmer, "Dcc-mac: a decentralized mac protocol for 802.15. 4a-like uwb mobile ad-hoc networks based on dynamic channel coding," in *Broadband Networks, 2004. BroadNets 2004. Proceedings. First International Conference on*, pp. 396–405, IEEE, 2004.
- [61] G. Kolumbán, T. Krébesz, C. K. Tse, and F. C.-M. Lau, "Improving the coverage of ultra wideband impulse radio by pulse compression," in *Circuits and Systems (ISCAS), 2012 IEEE International Symposium on*, pp. 1939–1942, IEEE, 2012.
- [62] S. Cui and F. Xiong, "Adaptive threshold to mitigate the effect of cross-modulation interference and synchronisation errors in energy detection pulse position modulation ultra-wideband systems," *Communications, IET*, vol. 6, no. 9, pp. 1032–1037, 2012.
- [63] A. Dezfooliyan and A. M. Weiner, "Phase compensation communication technique against time reversal for ultra-wideband channels," *Communications, IET*, vol. 7, no. 12, pp. 1287–1295, 2013.

- [64] S. Kim, Y. Kim, X. Li, and J. Kang, "Orthogonal pulse design in consideration of fcc and ieee 802.15. 4a constraints," *Communications Letters, IEEE*, vol. 17, no. 5, pp. 896–899, 2013.
- [65] I. Barraç, H. Trabelsi, and M. Masmoudi, "A new uwb pulse shaping for ieee 802.15. 4a," in *Systems, Signals & Devices (SSD), 2013 10th International Multi-Conference on*, pp. 1–5, IEEE, 2013.
- [66] S. Morosi, T. Bianchi, and F. Gei, "Frequency domain multi-user receivers for ieee 802.15. 4a short-range communication network," *Transactions on Emerging Telecommunications Technologies*, vol. 24, no. 6, pp. 545–551, 2013.
- [67] T. A. Holdings <http://www.timedomain.com/>.
- [68] N. T. GmbH <http://www.nanotron.com/>.
- [69] A. L. Dr. Andreas Schuerzinger, "Smartfactory – UWB-Funk-Ortungssysteme in der Automobilproduktion im Einsatz," 2010. <http://www.all-electronics.de/media/file/2706>.
- [70] <http://www2.imec.be>, "State-of-the-art integrated IR-UWB enabling global high-quality low-power mobile applications," [http://www2.imec.be/be\\_en/press/imec-news/uwb2011.html](http://www2.imec.be/be_en/press/imec-news/uwb2011.html).
- [71] decaWave, "DecaWave's DW1000 is the world's first single chip UWB transceiver," <http://www.decawave.com/>.
- [72] B. SAS <http://www.bespoon.com/>.
- [73] X. S. Technology <https://www.xethru.com/en/>.
- [74]
- [75] AVNET, "Xilinx Virtex-5 FXT PCIe Development Kit," <http://www.em.avnet.com/en-us/design/drc/Pages/Xilinx-Virtex-5-FXT-PCIe-Development-Kit.aspx>.
- [76] T. Volkhausen, D. H. Woldegebreal, and H. Karl, "Improving network coded cooperation by soft information," in *Sensor, Mesh and Ad Hoc Communications and Networks Workshops, 2009. SECON Workshops' 09. 6th Annual IEEE Communications Society Conference on*, pp. 1–6, IEEE, 2009.
- [77] D. H. Woldegebreal, *Network-Coded Cooperation in Wireless Networks: Theoretical Analysis and Performance Evaluation*. dissertation, Universität Paderborn, 2010.
- [78] R. Morelos-Zaragoza, "UWB channel model," <http://www.mathworks.com/matlabcentral/fileexchange/6697-uwb-channel-model>.

- [79] N. N. Instruments, "Systemdesignsoftware NI LabVIEW," <http://www.ni.com/labview/d/>.
- [80] Totalphase, "Aardvark I2C/SPI Host Adapter," <http://www.totalphase.com/products/aardvark-i2cspi/>.
- [81] H. Homayoun, S. Golshan, E. Bozorgzadeh, A. Veidenbaum, and F. J. Kurdahi, "On leakage power optimization in clock tree networks for ASICs and general-purpose processors," *Surface Science*, vol. 1, pp. 75–87, 2011.
- [82] Fujitsu, "Design of low power consumption LSIs," <http://www.fujitsu.com/cn/fsp/asic/lowpower.html>.
- [83] Y. Zheng, M. A. Arasu, K.-W. Wong, Y. J. The, A. Suan, D. D. Tran, W. G. Yeoh, and D.-L. Kwong, "A 0.18  $\mu\text{m}$  cmos 802.15.4a uwb transceiver for communication and localization," *IEEE International Solid-State Circuits Conference - Digest of Technical Papers, ISSCC 2008*, 2008.
- [84] D. Lachartre, B. Denis, D. Morche, L. Ouvry, M. Pezzin, B. Piaget, J. Prouvee, , and P. Vincent, "A 1.1nj/b 802.15.4a-compliant fully integrated uwb transceiver in 0.13  $\mu\text{m}$  cmos," *IEEE International Solid-State Circuits Conference - Digest of Technical Papers, ISSCC 2009*, 2009.
- [85] S. Joo, W. Chen, T. Choi, M. Oh, J. Park, J. Kim, and B. Jung, "A fully integrated 802.15. 4a ir-uwb transceiver in 0.13  $\mu\text{m}$  cmos with digital rrc synthesis," *IEEE International Solid-State Circuits Conference - Digest of Technical Papers, ISSCC 2010*, 2010.
- [86] Synopsys, "The Synopsys PrimeTime suite," <http://www.synopsys.com/Tools/Implementation/SignOff/PrimeTime/Pages/default.aspx>.
- [87] DecaWave, "ScenSor EVK1000 Evaluation Kit," <http://www.decawave.com/products/evk1000-evaluation-kit>.
- [88] *IEEE 802.15.4*. <http://standards.ieee.org/getieee802/download/802.15.4-2011.pdf>.
- [89] C. G. Klaus Gravogl, Jan Haase, *Choosing the best wireless protocol for typical applications*. Institute of Computer Technology, 2011.
- [90] N. F. T. Waqas Ikram, "Wireless communication in process automation: A survey of opportunities, requirements, concerns and challenges," in *Ultra-Wideband (ICUWB), 2013 IEEE International Conference on*, 2010.
- [91] *ZigBee IEEE 802.15.4*. <http://zigbee.org/Standards/Overview.aspx>.
- [92] *IPv6 over Low power WPAN*. <https://tools.ietf.org/wg/6lowpan/>.
- [93] *802.15.4-2006*. <http://standards.ieee.org/findstds/standard/802.15.4-2006.html>.

- [94] *WSAN-PA / WSAN-FA.* <http://www.profibus.com/nc/pi-organization/regional-pi-associations/homepage-italia/events/storico-eventi/damdownload/05-wireless-sensor-and-actuator-networks-automazione-di-fabbrica-wireless-emiliano-sisinni/download/>.
- [95] *ISA100.11a Technology Standard.* <http://www.nivis.com/technology/ISA100.11a.php>.
- [96] *HART Communication Foundation.* <http://de.hartcomm.org/>.
- [97] *Profibus - Profinet.* <http://www.profibus.com>.
- [98] *Fieldbus Foundation.* <http://www.fieldbus.org/index.html>.
- [99] *ODVA - DeviceNet.* <http://www.odva.org/Home/ODVATECHNOLOGIES/DeviceNet.aspx>.
- [100] *Microchip MiWi PRO Wireless Networking Protocol.* <http://www.microchip.com/wwwAppNotes/AppNotes.aspx?appnote=en554286>.
- [101] *Microchip MiWi P2P Wireless Protocol.* <http://www.microchip.com/wwwAppNotes/AppNotes.aspx?appnote=en536181>.
- [102] *MICROCHIP.* <http://www.microchip.com/>.
- [103] *Texas Instruments - SimpliCI™ - RF Made Easy.* [http://www.ti.com/corp/docs/landing/simpliCI/index.htm?DCMP=hpa\\_rf\\_general&HQS=NotApplicable+OT+simpliCI](http://www.ti.com/corp/docs/landing/simpliCI/index.htm?DCMP=hpa_rf_general&HQS=NotApplicable+OT+simpliCI).
- [104] *KNX Journal, 2006.* [http://www.knx.org/fileadmin/downloads/07%20-%20News%20%26%20Press/01%20-%20KNX%20Journal/2006/2006%20I%20KNX%20Journal\\_PDF%20and%20Text/2006\\_I\\_KNX%20Journal.pdf](http://www.knx.org/fileadmin/downloads/07%20-%20News%20%26%20Press/01%20-%20KNX%20Journal/2006/2006%20I%20KNX%20Journal_PDF%20and%20Text/2006_I_KNX%20Journal.pdf).
- [105] *KNX ist jetzt der internationale Standard ISO/IEC 14543-3.* <http://www.knx.org/fileadmin/news/120524713218185PR20061201-DE.pdf>.
- [106] K. J. 2007 [http://www.weinzierl.de/download/references/KnxJournal\\_2007-2\\_Seite\\_13-14.pdf](http://www.weinzierl.de/download/references/KnxJournal_2007-2_Seite_13-14.pdf).
- [107] *PHOENIX CONTACT Deutschland.* <https://www.phoenixcontact.com>.
- [108] *Schildknecht - Smart Data Communication.* <http://www.schildknecht.ag/>.
- [109] *Schildknecht - Smart Data Communication.* <https://www.xing.com/companies/profibusnutzerorganisatione.v>.
- [110] *PROFINET - Systembeschreibung.* <http://www.profibus.com/nc/download/technical-descriptions-books/downloads/profinet-technology-and-application-system-description/download/11865/>.

- 
- [111] *CiA - CAN in Automation*. <http://www.can-cia.org/>.
- [112] *ABB - Power and productivity for a better world*. <http://new.abb.com/de>.
- [113] *Übertragungsgeschwindigkeit bei WLAN-Verbindungen*. [http://avm.de/nc/service/fritzbox/fritzbox-7490/wissensdatenbank/publication/show/514\\_Geringe-Uebertragungsgeschwindigkeit-bei-WLAN-Verbindungen/](http://avm.de/nc/service/fritzbox/fritzbox-7490/wissensdatenbank/publication/show/514_Geringe-Uebertragungsgeschwindigkeit-bei-WLAN-Verbindungen/).
- [114] *SIEMENS - IWLAN*. <http://w3.siemens.com/mcms/automation/de/industrielle-kommunikation/iwlan-industrial-wireless-communication/seiten/default.aspx>.
- [115] *SIEMENS AG*. <http://www.siemens.com/>.





# Abbildungsverzeichnis

1.1	Industrielle Revolutionen bis heute . . . . .	2
1.2	Ergebnis einer Mitgliederumfrage, der Gesellschaft für Mess- und Automatisierungstechnik 2012 . . . . .	3
1.3	Zuverlässigkeit und Wirtschaftlichkeit sind die Hauptanforderungen der Automatisierungsindustrie. . . . .	3
1.4	Überschneidung der Anforderungen aus der Automatisierungsindustrie mit kabelgebundenen und drahtlosen Sensornetzwerke und Funktechnologien. . . . .	4
1.5	Bereits im Einsatz befindliche Funktechnologien in der Automatisierungsindustrie. . . . .	5
1.6	Anforderungen der Automatisierungsindustrie. . . . .	6
1.7	Sterntopologie und bidirektionale Datenübertragung. . . . .	8
1.8	Übersicht drahtloser Energieversorgungen für WSN. . . . .	11
2.1	Wechselwirkungen zwischen verschiedenen Eigenschaften von WSN. . . . .	18
2.2	Elektrische Schaltung eines Knallfunksender . . . . .	20
2.3	Europäische und amerikanische Regulierungsmaske der Leistungsdichte für Indoor UWB-Geräte . . . . .	22
2.4	Europäische Regulierungsmaske mit und ohne LDC. . . . .	22
2.5	Globale UWB-Frequenzuteilung . . . . .	23
2.6	Übersicht von IR-UWB-Standards . . . . .	23
2.7	Übersicht der Forschungsprojekte im Bereich UWB von 2000 bis 2015. . . . .	28
2.8	Für diese Arbeit relevante Veröffentlichungen und deren Optimierungsschwerpunkte . . . . .	29
2.9	Übersicht der Markteinführung kommerzieller Produkte im Bereich IR-UWB. . . . .	34
3.1	Struktur eines UWB-Frames. . . . .	38
3.2	Ternäre Darstellung der PREAMBLE-Symbole. . . . .	39
3.3	Darstellung eines Bit-Signals für die UWB-Übertragung. . . . .	41
3.4	Vergleich der bestmöglichen Übertragungszeiten für 3 und 48 Byte Nutzdaten . . . . .	43
3.5	Grafische Darstellung der Autokorrelationsgüte für zwei ternäre Sequenzen $p_1$ und $p_2$ . . . . .	52
3.6	PHR nach IEEE 802.15.4a . . . . .	57

3.7	Overhead in Bits in Abhängigkeit der verwendeten Fehlerkorrektur-Verfahren . . . . .	59
3.8	Zeitlicher Overhead in ns inkl. Verarbeitungszeit für verschiedene Fehlerkorrektur-Verfahren . . . . .	60
3.9	Vergleich zwischen einer standardkonformen und einer optimierten Punkt-zu-Punkt Übertragung . . . . .	63
3.10	Bidirektionale Kommunikation zwischen einem Master und $n$ Slaves .	64
3.11	Erweiterte PHR für Broadcast-Nachrichten . . . . .	66
3.12	Vergleich zwischen Halbduplex und Vollduplex. . . . .	68
3.13	Relative Verzögerungen zwischen $A$ und $B$ bei einer Datenrate von 27,24 Mb/ . . . . .	69
3.14	Verschiedene Typen von Kollisionen . . . . .	70
3.15	Aufbau und Funktionsweise des LFSR entsprechend dem UWB-Standard	70
3.16	Durchschnittliche Kollisionsverteilung . . . . .	71
3.17	Vergrößerung der Kollisionsverteilung aus Abbildung 3.16. . . . .	71
3.18	Auswirkungen der verschiedenen Verzögerungen von $B$ relativ zu $A$ .	72
3.19	Auswirkungen der verschiedenen Verzögerungen von $A$ relativ zu $B$ .	73
3.20	Kollisionsunterdrückung relativ zu $B$ . . . . .	73
3.21	Alle mögliche Hopping-Positionen für $j = 0$ . . . . .	75
3.22	Alle mögliche time-slot-Kombinationen für $j \in \{-8, \dots, 8\}$ und $A_{h^{(k+i)}} = 6$ . . . . .	76
3.23	Darstellung eines Synchronisationsverfahren zwischen $A$ und $B$ , welches eine Vollduplex-Kommunikation ermöglicht. . . . .	77
3.24	Unterschiedliche Darstellungen eines PREAMBLE-Symbols aus IEEE 802.15.4a . . . . .	79
3.25	Vergleich verschiedener Impulsverteilungen für $A$ und $B$ . . . . .	80
3.26	Globale Übertragungszeit $T_{TRANS}$ für verschiedene PS-Sequenzlängen	83
3.27	Verschiedene Bitraten mit 4 Impulsen innerhalb von 128 TS . . . . .	85
3.28	Verschiedene Bitraten für verschiedene Intervalllängen für zwei Sensorknoten $A$ und $B$ . . . . .	86
3.29	Verschiedene Bitraten für verschiedene Intervalllängen für drei Sensorknoten $A$ , $B$ und $C$ . . . . .	86
3.30	Anzahl an Synchronisationen abhängig von der Anzahl an genutzten Sensorknoten . . . . .	89
3.31	$T_{TRANS}$ für verschiedene Sensorknoten . . . . .	90
3.32	3 verschiedene Synchronisationsverfahren zwischen einem Master und mehreren Sensorknoten . . . . .	91
3.33	Globale Übertragungsdauer $T_{TRANS}$ bei verschiedenen Längen der Impulssequenz. . . . .	95
3.34	Optimierte PHR inkl. Nachrichtenlänge, Prüfsumme und ID . . . . .	98
3.35	Fehlerkorrektur durch Mehrfachübertragung. Die Fehlerhaften Stellen in der Übertragung sind rot markiert. Die Korrigierten Stellen in der Übertragung sind grün markiert. . . . .	104

3.36	Symbolfehlerrate für verschiedene Impulsfehlerraten und Anzahl an Impulsen . . . . .	107
4.1	Ablaufplan der Implementierung Positive Ergebnisse der Funktionstests sind grün markiert. Negative Ergebnisse der Funktionstests sind rot markiert. . . . .	114
4.2	UWB-Transceiver Blockschaltbild . . . . .	115
4.3	Blockschaltbild des UWB-Basisbands . . . . .	116
4.4	Kernkomponenten des Serialisierers . . . . .	119
4.5	Aufbau des Serialisierers . . . . .	119
4.6	Kernkomponenten des Parallelisierers . . . . .	120
4.7	Aufbau des Parallelisierers . . . . .	120
4.8	Korrektur des Analyse-Intervalls für eine verzögerungsfreie PS-Detektion.	122
4.9	CCA Energie- und PREAMBLE-Symbol-Detektor . . . . .	123
4.10	Alle implementierten Loop-Backs . . . . .	125
4.11	Darstellung der entwickelten Labview GUI mit vielen Einstellungsmöglichkeiten. 127	
4.12	Beispiel eines <i>clock-trees</i> . . . . .	127
4.13	Power-Gating . . . . .	128
4.14	Burst-Abstand . . . . .	129
4.15	Burst-Position-Vorhersage . . . . .	130
4.16	Durchschnittlicher Burstabstand abhängig von der Umschaltzeit und der verwendeten Datenrate. . . . .	131
4.17	Energieersparnis abhängig von der Umschaltzeit und der verwendeten Datenrate. . . . .	131
4.18	Optimale Verteilung der Sendephasen der Slaves . . . . .	133
4.19	Häufigkeiten verschiedener Parameter für den Fall $TX_{Master} \rightarrow RX_{S_k}$	134
4.20	Durchschnittliche Häufigkeiten verschiedener Parameter für den Fall $TX_{Master} \rightarrow RX_{S_k}$ . . . . .	135
4.21	Darstellung des Umschaltsignals, welches mittels dem Serialisierer generiert wird. . . . .	137
4.22	Implementierung der Umschaltlogik . . . . .	137
4.23	Chipfoto, wobei die obere Hälfte das analoge Frontend und die untere Hälfte das digitale Basisband darstellt. 139	
4.24	Block-Diagramm des analogen Frontends . . . . .	139
4.25	Momentanleistung bei verschiedenen Übertragungsraten . . . . .	142
4.26	Messaufbau für die Chip-Messung . . . . .	144
4.27	CCA: Das analysierte UWB-Signal ist orange markiert, das Interrupt-Signal des PS-Detektors ist lila markiert und das Interrupt-Signal des Energie-Detektors ist grün markiert. . . . .	145

4.28	Indexkorrektur während der Synchronisation	
	Im oberen orangen Signal besteht der PREAMBLE-Teil aus 16 PREAMBLE-Symbolen.	
	Im unteren orangen Signal besteht der PREAMBLE-Teil aus 4 PREAMBLE-Symbolen.	
	Für jedes UWB-Signal gibt es ein dazugehöriges lila markiertes Interrupt-Signal des PS-Detektors. . . . .	146
4.29	Broadcast-Nachricht mit einer Länge von 1000 Byte Nutzdaten mit Overhead des Reed-Solomon-Codes (rot markiert) und ohne Fehlerkorrektur (grün markiert). . . . .	148
4.30	Gemessene Bursts für alle Datenraten.	
	rot markiert: für 851 kb/s bzw. 16 Impulse je Burst	
	blau markiert: für 6,81 Mb/s bzw. 2 Impulse je Burst	
	grün markiert: für 27,24 MB/s bzw. 1 Impuls je Burst . . . . .	149
4.31	Alle unterstützten Übertragungsmodi des gefertigten ASICs. . . . .	150
4.32	DecaWave-Modul: IR-UWB Frame mit 6,81 Mb/s:	
	Alle 4 Teile des UWB-Frames PREAMBLE, SFD, PHR und PSDU sind gut zu erkennen. . . . .	151
4.33	DecaWave-Modul: Ausschnitt aus einem IR-UWB Frame mit 6,81 Mb/s:	
	Übergang vom PHR-Teil (851 kb/s) zum PSDU-Teil (6,81 Mb/s) . . .	152
4.34	DecaWave-Modul: Ausschnitt aus einem IR-UWB Frame:	
	Vergrößerte Darstellung eines PREAMBLE-Symbols . . . . .	152
4.35	DecaWave-Modul: Ausschnitt aus einem IR-UWB Frame:	
	Ein einzelner Impuls, wie er bei der Synchronisation oder bei einer Datenrate von 27,24 Mb/s vorkommt. . . . .	153
4.36	DecaWave-Modul: Ausschnitt aus einem IR-UWB Frame:	
	Ein Burst, bestehend aus zwei Impulsen, wie er bei einer Datenrate von 6,81 Mb/s vorkommt. . . . .	153
4.37	DecaWave-Modul: Ausschnitt aus einem IR-UWB Frame:	
	Ein Burst, bestehend aus 16 Impulsen, wie er bei einer Datenrate von 851 kb/s vorkommt. . . . .	154
4.38	Xilinx Virtex-5 FXT PCIe Development Kit . . . . .	155
4.39	FPGA Test . . . . .	155
4.40	FPGA Ressourcenverbrauch . . . . .	156
A.1	Forschungsprojekte im Bereich UWB von 2001 bis 2015 . . . . .	167

# Tabellenverzeichnis

2.1	EIRP-Grenzwerte der FCC für Indoor- und Handheld UWB-Geräte . . . . .	21
2.2	Übersicht und Vergleich der IR-UWB Standards . . . . .	25
3.1	Parameter eines Bits im PSDU-Teil in Abhängigkeit von der Datenrate. . . . .	42
3.2	Kürzeste Dauer des UWB-Frames für 3 Byte Nutzdaten (minimale angeforderte Datenmenge) . . . . .	43
3.3	Theoretisch erreichbare Bitraten unter der Annahme, dass die Impulse des <i>Bursts</i> innerhalb des ganzen Bit-Signals verteilt sind. . . . .	45
3.4	Alle Zustände für $p_i^j$ und $b_i$ und Gewichtungskoeffizienten . . . . .	51
3.5	Optimale Autokorrelationsgütern ternärer Sequenzen der Länge $n = 31$ mit $k$ nicht leeren Stellen. Fälle mit $2 \cdot k > n$ sind auf weißem und Fälle mit $2 \cdot k \leq n$ , welche das Kriterium (3.6) der technischen Beschränkungen erfüllen, auf grauem Hintergrund dargestellt. . . . .	53
3.6	Autokorrelationsgüte für Impulssequenzen der PREAMBLE-Symbole, entnommen aus dem UWB-Standard [37]. . . . .	54
3.7	Maximal mögliche Anzahl an Impulsen $\#IPS$ je SFD-Sequenzlänge $y$ und Impulssequenzlänge $n$ der PS für $\#PS = 16$ im PREAMBLE-Teil. Die in rot markierten Einträge stellen die jeweils optimale Anzahl an Impulsen $\#IPS$ für eine bestimmte PREAMBLE-Symbol-Länge dar. Der in grün markierte Eintrag ist die kürzeste Dauer der optimierten Synchronisationsphase $T_{SYNC_{opt}}$ , welche alle Anforderungen erfüllt. . . . .	56
3.8	Maximal mögliche Anzahl an Impulsen $\#IPS$ je SFD-Sequenzlänge $y$ , PREAMBLE-Symbol-Länge $n$ und Anzahl an PREAMBLE-Symbolen $\#PS$ im PREAMBLE-Teil. . . . .	56
3.9	Anzahl an Impulsen $\#I$ in einem PREAMBLE-Symbol in Abhängigkeit von $n$ und $X$ . . . . .	82
3.10	maximale $A_p$ der PREAMBLE-Symbole in Abhängigkeit von $n$ und $X$ . . . . .	83
3.11	Vergleich des Impulsverhältnisses während der Synchronisation für verschiedene Sequenzlängen der PREAMBLE-Symbole . . . . .	89
3.12	Vergleich des Impulsverhältnisses während der Synchronisation für verschiedene Sequenzlängen der PREAMBLE . . . . .	96
3.13	Unterschiedliche Übertragungszeiten für $BM_v$ und $BM_c$ . . . . .	100
3.14	$S_{m,l}$ für Impulskodierung . . . . .	103

3.15	Impulsfehlerkorrektur . . . . .	104
3.16	Impulsfehlerkorrektur mittels Prüfsumme . . . . .	105
3.17	Fehlerrate Master→ Slave abhängig von der Impulsfehlerrate für einzelne Datensymbole . . . . .	108
3.18	Fehlerrate Master→ Slave abhängig von der Impulsfehlerrate für einzelne Datensymbole . . . . .	109
3.19	Vergleich des Impulsverhältnisses während der Synchronisation für verschiedene Sequenzlängen der PREAMBLE . . . . .	109
3.20	Zusammenfassung der erzielten Ergebnisse . . . . .	111
4.1	Wahrheitstabelle für OOK und BPSK . . . . .	118
4.2	On-Chip Testmodes . . . . .	124
4.3	Unterschiedliche aktive Zeiten des Senders und Empfängers, für eine komplette Übertragung, in Abhängigkeit der Übertragungsart . . . . .	136
4.4	Chip Parameter, entnommen aus [19]. . . . .	138
4.5	Vergleich verschiedener aktueller Transceiver, entnommen aus [19]. . . . .	139
4.6	Flächen- und Leistungsverbrauch der wichtigsten Komponenten . . . . .	140
4.7	FPGA Ressourcenverbrauch . . . . .	156
A.1	Unterschied zwischen diversen Wlan-Standards . . . . .	166
A.2	Autokorrelationsgüte für optimale binäre Sequenzen . . . . .	181
A.3	Autokorrelationsgüte für optimale ternäre Sequenzen der Länge 2-10 . . . . .	182
A.4	Autokorrelationsgüte für optimale ternäre Sequenzen der Länge 11-15 . . . . .	183
A.5	Autokorrelationsgüte für optimale ternäre Sequenzen der Länge 16-19 . . . . .	184
A.6	Autokorrelationsgüte für optimale ternäre Sequenzen der Länge 20-22 . . . . .	185
A.7	Autokorrelationsgüte für optimale ternäre Sequenzen der Länge 23-24 . . . . .	186
A.8	Autokorrelationsgüte für optimale ternäre Sequenzen der Länge 25-26 . . . . .	187
A.9	Autokorrelationsgüte für optimale ternäre Sequenzen der Länge 27-28 . . . . .	188
A.10	Autokorrelationsgüte für optimale ternäre Sequenzen der Länge 29-30 . . . . .	189
A.11	Autokorrelationsgüte für optimale ternäre Sequenzen der Länge 31 . . . . .	190
A.12	Fehlerrate und Übertragungszeit Master→ Slave . . . . .	191