

Aus dem  
Institut für Medizinische Physik  
Friedrich-Alexander-Universität Erlangen-Nürnberg  
Direktor: Prof. Dr. habil. Dr. med. h. c. mult. Willi A. Kalender, PhD

---

# **Steuerungskonzept für einen dedizierten Brust-CT-Scanner**

Inaugural-Dissertation zur  
Erlangung der Doktorwürde  
der Medizinischen Fakultät  
der Friedrich-Alexander-Universität Erlangen-Nürnberg  
(Dr. rer. biol. hum.)

vorgelegt von  
Alexander Peter Peetz  
aus Erlangen





Gedruckt mit Erlaubnis der Medizinischen Fakultät  
der Friedrich-Alexander-Universität Erlangen-Nürnberg

- Dekan:** Prof. Dr. med. Dr. h.c. Jürgen Schüttler  
Klinik für Anästhesiologie
- Referent:** Prof. Dr. Dr. med. h.c. mult. Willi A. Kalender  
Institut für Medizinische Physik
- 1. Korreferent:** Prof. Dr.-Ing. Jörg Franke  
Lehrstuhl für Fertigungsautomatisierung und  
Produktionssystematik
- 2. Korreferent:** PD Dr. med. Evelyn Wenkel  
Radiologisches Institut
- Tag der mündlichen Prüfung:** 17.07.2013



„Kenntnisse kann jedermann haben,  
aber die Kunst zu denken ist das seltenste Geschenk der Natur.“

*Friedrich II. (1712 - 1786)*



# Inhaltsverzeichnis

<b>Zusammenfassung</b> .....	<b>1</b>
<b>Summary</b> .....	<b>3</b>
<b>1 Einleitung</b> .....	<b>5</b>
1.1 Motivation.....	5
1.2 Stand der Technik Brust-CT.....	7
1.3 Zielsetzung.....	9
<b>2 Grundlagen und Techniken</b> .....	<b>11</b>
2.1 Technische Umsetzung eines klinisches CT-Scanners.....	11
2.2 Prozessautomatisierung.....	13
2.2.1 Technischer Prozess.....	13
2.2.2 Prozesssteuerung durch Steuerungseinheit und Kontrolleinheit.....	13
2.2.3 Ebenenmodell in der Prozessautomatisierung.....	14
2.3 Echtzeitsysteme.....	16
2.3.1 Echtzeitfähigkeit.....	16
2.3.2 Ausprägung von Echtzeitsystemen.....	18
2.3.3 Eingebettetes System.....	20
2.3.4 Echtzeitbetriebssysteme.....	21
2.4 Echtzeitkommunikation.....	23
2.4.1 ISO/OSI-Referenzmodell.....	23
2.4.2 Kommunikationsmodelle.....	24
2.4.3 Medienzugriffssteuerung.....	25
2.5 Kommunikationsstandards in der seriellen Datenübertragung.....	26
2.5.1 Ethernet nach IEEE 802.3.....	26
2.5.2 Controller Area Network.....	29
2.5.3 RS485.....	31
<b>3 Steuerungskonzept</b> .....	<b>33</b>
3.1 Zu steuernde Komponenten des Brust-CT-Scanners.....	33
3.1.1 Röntgenröhre.....	33
3.1.2 Röntgendetektor.....	35
3.1.3 Antriebe.....	37
3.2 Systemarchitektur.....	38
3.2.1 Stand der Technik.....	38

---

3.2.2	Probleme beim Stand der Technik.....	40
3.2.3	Ebenenmodell für den Brust-CT-Scanner .....	41
3.3	Steuerungsdaten-Kommunikationsmodell.....	42
3.3.1	Systemübersicht.....	42
3.3.2	CAN-Kommunikationsprotokoll „CT_Control_CAN“ .....	44
3.3.3	Ethernet-Kommunikationsprotokoll „CT_Control_Ethernet“ .....	47
3.4	Rohdatenübertragungsprotokoll „CT_Data_Ethernet“ .....	51
3.4.1	Stand der Technik .....	51
3.4.2	Probleme beim Stand der Technik.....	51
3.4.3	Sender- und Empfängerarchitektur für „CT_Data_Ethernet“ .....	52
3.4.4	Kommunikationsablauf „CT_Data_Ethernet“ .....	53
3.4.5	Paketaufbau „CT_Data_Ethernet“ .....	53
3.5	Synchronisation der Bildgebungs-komponenten .....	55
3.5.1	Stand der Technik .....	55
3.5.2	Probleme beim Stand der Technik.....	56
3.5.3	Zeitbasierter Projektionsbeginn mit Positionserfassung.....	56
3.5.4	Systemübersicht .....	57
3.5.5	Positionserfassung .....	58
3.5.6	Synchronisation der Röntgenröhre .....	60
3.5.7	Synchronisation der Detektormodule .....	60
3.5.8	Minimierung von Totzeiten .....	62
3.6	Sicherheitskonzept.....	64
3.6.1	CT-Scanner als sicherheitskritisches System .....	64
3.6.2	Maßnahmen zur Gefahrenvermeidung .....	65
3.6.3	Eskalationsstufen zur Fehlermeldung.....	66
<b>4</b>	<b>Realisierbarkeit des Steuerungskonzepts .....</b>	<b>69</b>
4.1	Hardwareebenen .....	69
4.1.1	Realisierung des Leitrechners.....	69
4.1.2	Realisierung der Steuerungseinheiten.....	71
4.1.3	Realisierung der Kontrolleinheiten.....	77
4.2	Sicherheitsfunktionen .....	77
4.2.1	Interlockkreis .....	77
4.2.2	Strahlungsfreigabe .....	78
4.3	Messung der Leistung ausgewählter Echtzeitbetriebssysteme .....	79
4.3.1	Einleitung.....	79
4.3.2	Material und Methoden.....	79

---

4.3.3	Ergebnisse .....	81
4.3.4	Diskussion.....	82
4.4	Versuche zum Datendurchsatz von 10-GBit/s-Ethernet.....	82
4.4.1	Einleitung.....	82
4.4.2	Material und Methoden.....	83
4.4.3	Ergebnisse .....	84
4.4.4	Diskussion.....	86
4.5	Messungen der Signallaufzeiten des Synchronisationssignals .....	87
4.5.1	Einleitung.....	87
4.5.2	Material und Methoden.....	87
4.5.3	Ergebnisse .....	88
4.5.4	Diskussion.....	89
<b>5</b>	<b>Zusammenfassung und Optimierungspotential .....</b>	<b>91</b>
5.1	Zusammenfassung der Ergebnisse.....	91
5.2	Optimierungspotential .....	93
	<b>Abkürzungsverzeichnis .....</b>	<b>95</b>
	<b>Literaturverzeichnis.....</b>	<b>99</b>
	<b>Abbildungsverzeichnis.....</b>	<b>117</b>
	<b>Tabellenverzeichnis.....</b>	<b>119</b>
	<b>Danksagung .....</b>	<b>121</b>



## Zusammenfassung

### Hintergrund und Ziele

Zur Brustkrebsfrüherkennung ist seit Jahren die Mammografie als Bildgebungsverfahren etabliert. Um die Spezifität und Sensitivität bei der Brustkrebsfrüherkennung zu steigern, entwickelt das Institut für medizinische Physik zusammen mit der Firma CT Imaging GmbH einen dedizierten Brust-CT-Scanner. Aufgrund der neuartigen verwendeten Komponenten können aus dem Stand der Technik bekannte Konzepte zur Steuerung von klinischen CT-Scannern nicht verwendet werden. Stattdessen wird ein neues Steuerungskonzept benötigt, das die speziellen Anforderungen der Brust-CT-Bildgebung erfüllt und sich vom Stand der Technik abhebt.

### Methoden

Zur Steuerung des Brust-CT-Scanners wird ein an die Prozessautomatisierung angelehntes, hierarchisches Ebenenmodell vorgestellt, das aus den drei Ebenen Leit-, Steuerungs- und Kontrollebene besteht. Da harte Echtzeitanforderungen für die Steuerungsebene bestehen, wurden als Steuerungseinheiten FPGA-basierte eingebettete Systeme (SOPCs) gewählt, auf denen ein NIOSII-Softcore-Prozessor instanziiert ist. Auf diesem Prozessortyp wurden verschiedene Echtzeitbetriebssysteme und deren Leistung anhand ihrer Reaktionszeiten getestet. Ferner erfolgte eine Bewertung mittels weiterer Betriebssystemeigenschaften, die für die Entwicklung des Brust-CT-Scanners relevant sind. Zur Kommunikation zwischen den drei Ebenen wurden eigene deterministische Protokolle „CT\_Control\_CAN“ und „CT\_Control\_Ethernet“ entwickelt, die auf der zweiten Schicht des OSI-Modells der Technologien CAN und Ethernet aufsetzen.

Für den dosiseffizienten Betrieb des Brust-CT-Scanners müssen Positionserfassung, Röntgenröhre und Detektor, sowie deren Totzeiten exakt miteinander synchronisiert sein. Anders als im Stand der Technik wird hierfür eine Synchronisation vorgestellt, die in die Steuerungseinheiten der Steuerungsebene integriert ist und den Schnittstellenstandard RS485 nutzt. Um die Genauigkeit der Synchronisation zu bewerten, wurden Messungen zur Signallaufzeit und -fluktuation (Jitter) an einem Synchronisationssignal der Steuerungseinheiten durchgeführt.

Aufgrund der hohen Auflösung und der hohen Bildrate des dedizierten Brust-CT-Scanners ergeben sich hohe Anforderungen an die Datenübertragung im System. Zur schnellen Übertragung der Bilddaten vom Detektor über einen Drehübertrager auf einen Rekonstruktionsserver wird das eigens hierfür entwickelte Kommunikationsprotokoll „CT\_Data\_Ethernet“ auf Basis der 10-GBit/s-Ethernet-Technologie vorgestellt, das mit Standard-Komponenten realisierbar ist und an das Datenformat der Projektionsdaten des Brust-CT-Scanners angepasst ist. 10-GBit/s-Ethernet-Komponenten weisen einen

theoretischen Datendurchsatz von 10 GBit/s auf, jedoch zeigen zahlreiche Veröffentlichungen geringere Datenraten.

Ebenfalls Betrachtung finden sicherheitskritische Aspekte. Um Verletzungen an Menschen sowie Schäden an Material ausschließen zu können, wird ein Sicherheitskonzept eingeführt, das über verschiedene Eskalationsstufen zur Meldung von Fehlern, einen Interlockkreis und ein sicheres Auslösen und Abschalten von Röntgenstrahlung verfügt.

## **Ergebnisse und Beobachtungen**

Die Leistungsmessungen an verschiedenen Echtzeitbetriebssystemen, die auf dem eingesetzten NIOSII-Prozessor instanziiert waren, zeigen zufriedenstellende Ergebnisse. Für den Brust-CT-Scanner wird nach Abwägung weiterer Kriterien, wie z.B. die Zertifizierung nach einschlägigen Normen, FreeRTOS bzw. SafeRTOS empfohlen.

Die Integration der Synchronisationsfunktion in die Steuerungseinheiten auf Basis eines FPGAs stellt eine Verbesserung zu den im Stand der Technik verwendeten einzelnen Synchronisationseinheiten dar. Die Ergebnisse der Messungen an einem Synchronisationssignal über RS485 zeigen eine Signallaufzeit von ca. 85 ns bei einem Jitter von 3 ns. Damit erfüllen die von den Steuerungseinheiten bereitgestellten Signale die Systemspezifikation eines Jitters kleiner als 1  $\mu$ s und einer Signallaufzeit von unter 10  $\mu$ s.

Zur Übertragung der Projektionsdaten wurde eine 10-GBit/s-Ethernet-Übertragung mit zugehörigem Protokoll entwickelt, die Standard-Ethernet-Komponenten verwendet. Eigene Messungen an einer solchen aus Standard-Komponenten aufgebauten 10-GBit/s Übertragungsstrecke ergaben Datenraten von bis zu 9,9 GBit/s und zeigten, dass die für den Brust-CT-Scanner geforderte Datenrate erreicht werden kann.

## **Praktische Schlussfolgerungen**

Es wurde ein neuartiges, hierarchisches Steuerungskonzept für den dedizierten Brust-CT-Scanner entwickelt. Das vorgestellte Steuerungskonzept erfüllt alle wesentlichen Anforderungen und zeichnet sich besonders durch seine hohe Flexibilität aus. Durch die Realisierung der Steuerungseinheiten auf Basis von FPGAs und bekannten, offenen Schnittstellentechnologien kann auf Änderungen von Anforderungen schnell reagiert werden, ohne das gesamte Steuerungskonzept ändern zu müssen.

Durch die in die Steuerungseinheiten integrierte Synchronisationsfunktion können hochgenaue Synchronisationssignale für die Bildgebungskomponenten bereitgestellt werden. Dadurch lassen sich technologisch bedingte Totzeiten minimieren und somit die Dosiseffizienz des gesamten Scanners steigern.

Anders als Steuerungskonzepte für CT-Scanner, die aus dem Stand der Technik bekannt sind, ist das vorgestellte Steuerungskonzept auf die Komponenten des Brust-CT-Scanners angepasst und optimiert. Es kann damit wesentlich zum Erfolg dieses Scanners in der Diagnose von Brustkrebs beitragen, ist aber durch seine hohe Flexibilität und Modularität auch auf andere Systeme der Medizintechnik übertragbar.

## Summary

### Background and aims

In the early diagnosis of breast cancer, mammography has been the established imaging technique for years. To improve both specificity and sensitivity of early diagnosis of breast cancer, the Institute of Medical Physics, in collaboration with the company CT Imaging GmbH, is developing a dedicated breast CT scanner. Due to the novel components being used for this approach, prior art concepts for controlling clinical CT scanners are not applicable. Instead a new controlling concept is needed, which fulfills the special requirements of breast CT imaging and distinguishes itself from state-of-the-art concepts.

### Methods

To control the breast CT scanner, a hierarchical three layered control architecture, based on the layers used in process automation, was developed. It consists of the layers coordination, control and command layer. Because of real-time requirements, the control units of the control layer were realized with FPGA-based embedded systems (SOPC) including an instantiated NIOSII softcore processor. The performance of different kinds of real-time operating systems was tested on this type of processor with respect to their reaction times. In addition an evaluation was performed using further operating system properties, which are relevant regarding to the development of the breast CT scanner. As means of communication between the three layers, deterministic protocols, “CT\_Control\_CAN“ and “CT\_Control\_Ethernet“, were developed by the author. They are based on the second level of the OSI model of CAN and Ethernet technology.

For dose-efficient operation of the breast CT scanner, the position tracking system, x-ray tube and detector, as well as their dead times, must be synchronized exactly. Therefore a synchronization is presented which, in contrast to the state-of-the art, is integrated in the control units of the control layer and uses the RS485 interface standard. To evaluate the precision of the synchronization, measurements on a synchronization signal of the control units were performed, measuring signal propagation time and its' Jitter.

The high resolution and high frame rate of the dedicated breast CT scanner lead to high requirements on the systems data transmission rate. In order to transmit the image data from the detector via slipping to a reconstruction server, a specially designed communication protocol “CT\_Data\_Ethernet“, based on 10-GBit/s-Ethernet technology, is introduced. It can be realized by using standard components and is adapted to the data format of the breast CT scanners projection data. 10 Gbit/s Ethernet components have a theoretical data throughput of 10 Gbit/s, however numerous papers report lower data transfer rates.

Safety-critical aspects are also considered. To avoid injury to humans as well as damage to the system, a safety concept is presented. It includes different levels of escalation for error messaging, an interlock circuit and a safe x-ray release and disabling mechanism.

## Results and Observations

The performance measurements carried out on various real-time operating systems instantiated on the used NIOSII processor all show satisfactory results. Taking this fact into account and balancing further criteria, e.g. certification according relevant standards, FreeRTOS is recommended for use in the breast CT scanner.

The integration of the synchronization function in the FPGA based control units is an improvement over the single synchronization units used in the state of the art. The results of the measurements on a synchronization signal sent over RS485 show a signal propagation time of about 85 ns and a jitter of 3 ns. The control unit's synchronization signals thereby fulfill the system requirement of a jitter of less than 1  $\mu$ s and a signal propagation time of less than 10  $\mu$ s.

For the transfer of the projection data a 10 GBit/s Ethernet transmission, based on standard components, was developed. Measurements performed by the author on such a 10 GBit/s Ethernet transmission, designed with standard components, showed data rates up to 9,9 GBit/s and that the required data transfer rate of the breast CT scanner can be achieved.

## Practical Conclusions

A novel, hierarchical control concept for the dedicated breast CT scanner was developed. The presented control concept fulfills all relevant requirements and is particularly characterized by its high flexibility. As a result of the realization of the control units based on FPGAs and well-known, open interface technology, it is possible to respond quickly to changes in requirements without modifying the whole control concept.

Due to the integration of the synchronization functionality in the control units, the imaging components can be provided with high-precision synchronization signals. Thus technologically caused dead times can be minimized and thereby the dose efficiency of the entire scanner increased.

Contrary to prior art control concepts, the presented control concept is well adapted and optimized for the components of the breast CT scanner. It can thus make an essential contribution to the success of the scanner in the diagnosis of breast cancer. Due to its high flexibility and modularity it is furthermore easily portable to other medical-technology systems.

# 1 Einleitung

## 1.1 Motivation

Jedes Jahr erkranken in Deutschland ca. 72.000 Frauen an Brustkrebs. Statistisch bedeutet dies, dass jede achte Frau in ihrem Leben an dieser Krebsart erkrankt. Mit einem prozentualen Anteil von 32% ist das Mammakarzinom somit die häufigste Krebsneuerkrankung unter Frauen. Der Anteil von Brustkrebs an allen krebsbedingten Sterbefällen bei Frauen beträgt 17%. Ein wichtiger Bestandteil einer erfolgreichen Therapie ist eine frühe Diagnose. Um diese zu ermöglichen, wurde in den Jahren 2005 bis 2008 in Deutschland ein staatliches Screening-Programm aufgebaut, welches für Frauen ab 30 Jahren einmal jährlich eine Tastuntersuchung vorsieht. Zusätzlich werden Frauen zwischen 50 und 69 alle zwei Jahre zu einer Untersuchung der Brust via Mammografie eingeladen. [193]

Bei der Mammografie handelt es sich um eine Röntgenuntersuchung bei der die Brust zwischen zwei Platten komprimiert fixiert und aus zwei Richtungen, meist kraniokaudal (kopf-fußwärts) und mediolateral-oblique (schräg-seitlich), durchstrahlt wird. Die sich daraus ergebenden Aufnahmen werden anschließend von Radiologen ausgewertet. [94] Wichtige mammografische Auffälligkeiten sind pathognomonische benigne Verkalkungen, Mikroverkalkungen, umschriebene Herde und sternförmige sowie diffuse Veränderungen. Besonderes Augenmerk gilt hier Mikrokalzifikationen, kleinsten Kalkablagerungen, welche bei 51% der auffälligen mammografischen Befunde vertreten sind. [99] Zur genaueren Abklärung können weitere Mammogramme wie z.B. Ergänzungsaufnahmen aus anderen Richtungen sowie Vergrößerungsaufnahmen erstellt werden. Ergänzende Möglichkeiten zur Abklärung sind die Sonografie (bei besonders dichten Brüsten) sowie die Magnetresonanztomografie nach Kontrastmittelgabe. Bei einer unklaren Befundung wird durch eine Biopsie an der entsprechenden Stelle Gewebe entnommen und anschließend histologisch untersucht, um eine eindeutige Aussage über die Dignität treffen zu können. [60; 99; 132]

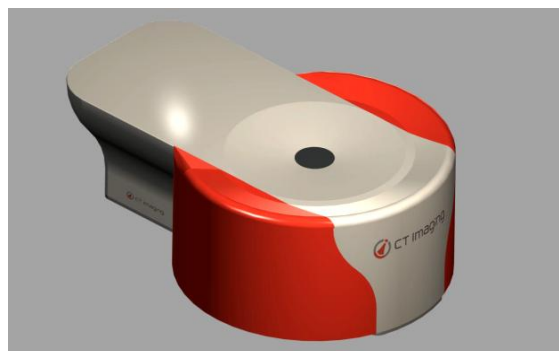
Mithilfe ausgereifter Röntgengeräte und speziell geschulter Ärzte, Fachkräfte und Histologen lässt sich mit dem Einsatz der Mammografie im Screening-Programm eine Sensitivität (Anteil von erkannten Karzinomen bei tatsächlich vorhandenen Karzinomen) von 62% bis 88% erreichen. [25; 129] Es stellt sich die Frage, warum diese nicht deutlich höher liegt. Ein Grund hierfür ist systembedingt und basiert auf der Tatsache, dass es sich bei der Mammografie um ein zweidimensionales Untersuchungsverfahren handelt. Beim Anfertigen eines Mamagramms kommt es zwangsläufig zu Überlagerung von Gewebeschichten und den darin enthaltenen Strukturen. Auf einem 2D-Röntgenbild

verdeckt Gewebe höherer Absorption, d.h. mit höherer Strahlendichte, weniger stark absorbierendes Gewebe. [127]

Obwohl durch Screening-Programme die Brustkrebsmortalität unter den Teilnehmerinnen um bis zu 30% gesenkt werden kann, nehmen nur etwa 53% der eingeladenen Frauen teil. [16] Eine mögliche Ursache für die geringe Akzeptanz unter Frauen ist die für die Mammografie notwendige Kompression der Brust. Diese Kompression wird von vielen Frauen als unangenehm und sogar schmerzhaft empfunden, weshalb potentielle Teilnehmerinnen schon vor dem ersten Screening verunsichert sind und nicht erscheinen oder einer weiteren Einladung nicht Folge leisten. [35]

Der Einsatz der Computertomografie (CT) anstelle der Mammografie könnte zu einer Steigerung der Sensitivität sowie zu höheren Teilnehmerquoten im Screening Programm führen. Diese Annahme ist darin begründet, da es sich bei der CT, anders als bei der Mammografie um ein dreidimensionales Bildgebungsverfahren handelt, das Schnittbilder von einem zu untersuchenden Objekt erstellt. Durch eine dreidimensionale Darstellung der Brust kommt es nicht zu Überlagerungseffekten, weshalb Strukturen besser aufgelöst erkennbar sind. Dies ist besonders bei der Untersuchung von Brüsten mit einer hohen Gewebedichte von Vorteil, da eine hohe Gewebedichte mit einer geringeren Sensitivität der Mammografie korreliert. [25] Außerdem bietet eine Untersuchung mit CT den großen Vorteil, dass die bei der Mammografie notwendige Kompression der weiblichen Brust nicht erfolgt. Dies könnte längerfristig zu einer besseren Akzeptanz unter den Frauen und somit einer höheren Teilnehmerquote führen. [127]

Aufgrund der sich durch die CT ergebenden Vorteile beschäftigen sich derzeit mehrere Forschungseinrichtungen, unter anderem auch das Institut für Medizinische Physik der Friedrich-Alexander-Universität Erlangen-Nürnberg (IMP) in Zusammenarbeit mit der Firma CT Imaging GmbH (CTI) mit der Konzeption eines Brust-CT-Scanners. Der von IMP und CTI geplante Brust-CT-Scanner, Abbildung 3, soll neue Maßstäbe setzen und der CT in der Brustbildgebung zum Durchbruch verhelfen. Ein solcher CT-Scanner stellt eine absolute technische Neuheit mit großem Potential dar und wird im Vergleich zur Mammografie über eine höhere Bildqualität, einen besseren Weichgewebekontrast und mehr Patientenkomfort verfügen, ohne jedoch mehr Dosis zu applizieren. [129]



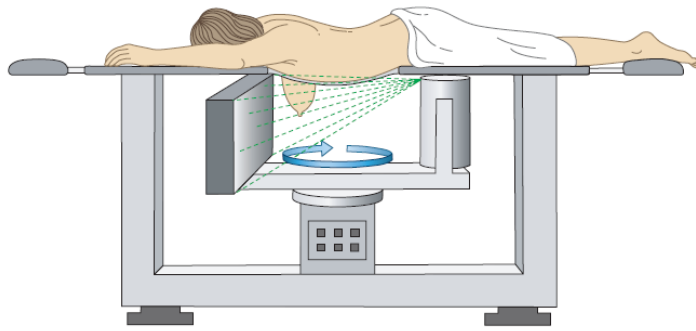
**Abbildung 1: Der dedizierte Brust-CT-Scanner von IMP und CTI.**

## 1.2 Stand der Technik Brust-CT

Schon kurz nach Erfindung der CT durch Godfrey Hounsfield im Jahre 1969 reifte in den 1970er Jahren die Idee, CT auch für die weibliche Brust zu nutzen. Dieser Gedanke wurde 1976 von General Electric umgesetzt und an zwei als CT/M bezeichneten CT-Scannern klinisch evaluiert. Die daraus entstandenen Studien von Chang et al. [27] und Gisvold et al. [86] kamen zu dem Ergebnis, dass die CT/M der Mammografie aufgrund der niedrigeren Ortsauflösung, der Notwendigkeit zum Einsatz von Kontrastmitteln und der höheren Dosisbelastung unterlegen sei. Daraufhin wurde von einer weiteren Vermarktung des CT/M-Konzepts abgesehen und die Weiterentwicklung der Prototypen eingestellt. [87]

In den folgenden Jahren entwickelte sich die CT technologisch schnell, wodurch die Ortsauflösung sowie das Kontrastverhältnis stetig zunahm. Gleichzeitig verkürzten sich die Scanzeiten, weshalb das Konzept CT zur Verwendung in der Brustbildgebung wieder attraktiver wurde. Etliche Studien zur Verwendung konventioneller Ganzkörper-CT-Scanner in der Brustbildgebung berichten (unter Einsatz von Kontrastmitteln) von hoher Sensitivität, aber geringer Spezifität. Es wurden also zu wenig gesunde Frauen auch als gesund erkannt. [115; 170; 195] Des Weiteren ist eine Untersuchung der Brust mit Ganzkörper-CT-Scannern mit einer relativ hohen Dosisbelastung verbunden. Der Grund hierfür ist, dass bei einer Untersuchung der Brust im konventionellen CT zwangsläufig der ganze Oberkörper durchstrahlt werden muss. Inoue et al. [115] schreibt von einer bis zu zehnmal höheren Dosis als bei einer Mammografie. Aufgrund der beiden genannten Nachteile, der niedrigen Spezifität und der hohen Dosisbelastung, eignet sich die konventionelle Ganzkörper-CT nicht für den Einsatz in Screening-Programmen zur Brustkrebs-Früherkennung.

Seit dem Aufkommen von digitalen Flachdetektoren in der Röntgendiagnostik findet diese Detektortechnologie in immer mehr Bereichen Anwendung. [214] Unter anderem werden Flachdetektoren in der Mammografie, Angiografie, digitalen Volumetomografie eingesetzt und es gibt Bestrebungen, diese auch für dedizierte Brust-CT-Scanner zu verwenden. So versuchen seit einigen Jahren verschiedene Forschergruppen Scanner auf Basis solcher Flachdetektoren zu entwickeln. Zu nennen sind hier unter anderem die Forschergruppe um John Boone an der University of California in Davis [18], Martin Tornai von der Duke University in Durham [228] sowie Ruola Ning von der University of Rochester [174]. Alle diese Ansätze basieren auf einem ähnlichen Prinzip, welches in Abbildung 2 dargestellt ist.

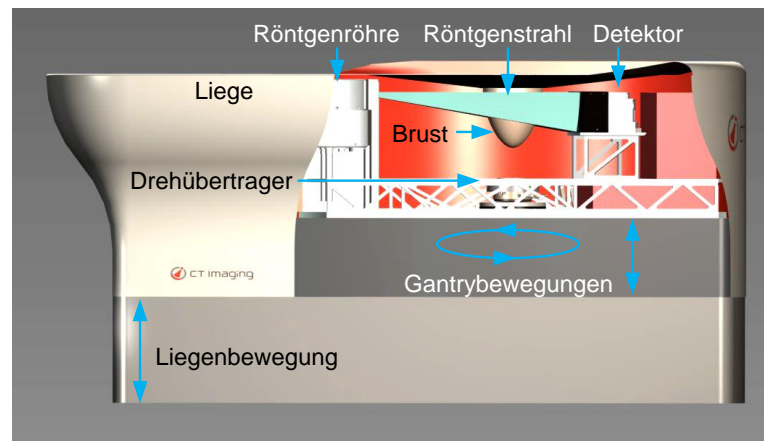


**Abbildung 2: Das Prinzip bisheriger Ansätze für Brust-CT-Scanner aus [87]:** Es basiert auf dem Einsatz eines Flachdetektors und einer Kegelstrahlröntgenröhre bei einer begrenzten Zahl von Gantryrotationen.

Die Patientin ruht auf einer Liege mit einer Öffnung, durch welche die zu untersuchende Brust frei hängen kann. Unter der Liege befindet sich ein drehbar gelagerter Aufbau, eine sogenannte Gantry, auf dem sich eine Kegelstrahl-Röntgenröhre sowie ein indirekt konvertierender, digitaler Flachdetektor befinden. Bei der Untersuchung rotiert die Gantry um die freihängende Brust und nimmt während einer 360°-Drehung die für eine Rekonstruktion benötigten Projektionen auf. Die zur Energieversorgung und Datenübertragung von Seiten der Gantry benötigten Leitungen sind hierbei an die begrenzte Drehbewegung der Gantry angepasst und als Schleppkabel ausgeführt. [18; 87; 174; 228]

Anders als die drei oben vorgestellten Forschergruppen verfolgen IMP und CTI einen anderen Ansatz. Ausgangspunkt sind die von Kalender et al. [129] definierten Anforderungen, welche ein dedizierter Brust-CT-Scanner erfüllen muss, um sich für den Einsatz in der Brustbildgebung auszuzeichnen. Der dedizierte Brust-CT-Scanner von IMP und CTI wird diese Anforderungen unter anderem dadurch erfüllen können, indem er spiralförmige Scans ausführt. Dies wird ermöglicht durch den Einsatz eines neuartigen Drehübertragers, über dessen Kontakte die Energieversorgung und Datenübertragung unterbrechungsfrei und unabhängig von der Drehbewegung der Gantry erfolgt. Um die Vorteile der Spiral-CT optimal zu nutzen, kommen eine speziell für den Brust-CT-Scanner entwickelte Röntgenröhre mit Fächerstrahl-Geometrie sowie ein neuartiger Detektor zum Einsatz. Hierbei handelt es sich um einen aus mehreren Modulen bestehenden, bogenförmigen Detektor, welcher über direkt konvertierendes Detektormaterial sowie eine neue Technologie zur Steigerung der Bildqualität verfügt. [127]

Abbildung 3 zeigt das Prinzip des dedizierten Brust-CT-Scanners von IMP und CTI. Die Patientin ruht auf einer höhenverstellbaren Liege, die zu Beginn der Untersuchung nach oben gefahren wird. Während der Untersuchung bewegt sich die Gantry, ausgehend von der Liegenunterseite spiralförmig um die Brust der Patientin nach unten. Die Anzahl der Rotationen ist wie beschrieben durch den Einsatz eines Drehübertragers unbegrenzt. In diesem Zusammenhang durchgeführte Simulationen zeigten, dass die genannten Anforderungen von Kalender et al. mit einem Spiral-CT-Scanner auf Basis der neuartigen Komponenten, in der vorgestellten Anordnung, erfüllbar sind. [129]



**Abbildung 3: Das Prinzip des dedizierten Brust-CT-Scanners von IMP und CTI:** Es handelt sich um einen Spiral-CT-Scanner mit einem Drehübertrager zur kontinuierlichen Gantryrotation sowie neuartigen Bildgebungskomponenten.

### 1.3 Zielsetzung

Bisherige Ansätze zur Entwicklung von Brust-CT-Scannern verwenden einen Flachdetektor in Kombination mit einer Mammografie-Röntgenröhre und entsprechen technologisch gesehen mehr einem Mammografie-Gerät als einem CT-Scanner. Der vom IMP geplante Scanner hingegen wird technologisch einem konventionellen Ganzkörper-CT-Scanner ähneln. Eine Anlehnung an bestehende Steuerungskonzepte von konventionellen CT-Scannern bekannter Hersteller ist jedoch aus verschiedenen Gründen (z.B. Neuartigkeit der Komponenten, Kosten, Geheimhaltung) nicht möglich. Hieraus ergibt sich die Notwendigkeit zur Entwicklung eines neuartigen Steuerungskonzepts für den dedizierten Brust-CT-Scanner. Ein Konzept, welches möglichst auf bekannten Technologien sowie Standardschnittstellen basiert, einfach zu erweitern und kostengünstig zu implementieren ist.

Hierfür ist zunächst eine Analyse der verwendeten Komponenten, besonders der neuartigen Bildgebungskomponenten Röntgenröhre und Detektor, notwendig. Darauf aufbauend muss ein Konzept zur Steuerung entwickelt werden, welches die spezifischen Eigenschaften dieser Komponenten berücksichtigt oder, wenn dies nicht möglich ist, Empfehlungen zu geben, wie diese Komponenten beschaffen sein müssen, um sich in ein mögliches Steuerungskonzept zu integrieren.

Zur Kommunikation mit den Komponenten des Scanners muss die Möglichkeit bestehen, diese über eine eventuell echtzeitfähige Kommunikationsverbindung anzusprechen, um z.B. Werte auszulesen, Parameter zu setzen oder Aktionen zu starten. Hierzu nötige Schnittstellen und Protokolle sind auszuwählen bzw. zu entwickeln. Eine wesentliche Randbedingung, die es zu berücksichtigen gilt, wird durch den verwendeten Drehübertrager vorgegeben. Dieser ermöglicht zwar eine kontinuierliche Drehung der Gantry, die Anzahl der enthaltenen Übertragungskanäle ist aufgrund der Baugröße aber limitiert.

Weitere Randbedingungen geben der neuartige Detektor sowie die Röntgenröhre vor. Um die erwartete Bildqualität erreichen zu können, müssen ihre Prozesse miteinander synchronisiert ablaufen. Dies muss das Steuerungskonzept berücksichtigen, da durch Ungenauigkeiten in der Synchronisation die verwendeten Technologien in der Röntgenröhre und dem Detektor nicht optimal nutzbar sind, was sich in einer Verschlechterung der Bildqualität äußert. Außerdem wäre die von der Röntgenröhre applizierte Dosis nicht effektiv nutzbar, wodurch den Patienten mehr Dosis appliziert würde, als medizinisch nötig wäre. Zur Synchronisation benötigen diese Komponenten ein Signal möglichst kurzer Laufzeit und hoher Genauigkeit, d.h. mit niedriger Fluktuation (Jitter). Die entsprechenden Synchronisationssignale sollen im Brust-CT-Scanner laut Spezifikation einen Jitter von kleiner als  $1 \mu\text{s}$  aufweisen.

Aufgrund der hohen Ortsauflösung des Brust-CT-Scanners ergibt sich ein hohes Datenvolumen auf Seiten des Detektors. Dieses Datenvolumen ist in möglichst kurzer Zeit über den Drehübertrager zu einem Rekonstruktionssystem zu übermitteln. Hierzu wird eine Schnittstellentechnologie zur Datenübertragung benötigt, welche möglichst wenig Protokolloverhead besitzt und hohe Datenraten zulässt. Für den Brust-CT-Scanner ist laut Spezifikation eine Datenübertragungsrate von mindestens 7 GBit/s gefordert.

Ein weiteres Ziel des Promotionsprojekts ist die Sicherstellung eines gefahrlosen Betriebs des CT-Scanners nach entsprechenden Normen für die Medizintechnik, wie z.B. DIN EN 60601-1 [46] und DIN EN 60601-2-44 [48]. Die Sicherstellung eines gefahrlosen Betriebs ist besonders für eine spätere Zertifizierung des Scanners für klinische Anwendungen wichtig, da eine solche Zertifizierung nur hierdurch möglich ist. Aufgrund der eingesetzten ionisierenden Strahlung sind besondere Sicherheitsstandards einzuhalten, um unnötige Dosisbelastung für die Patienten auszuschließen. So muss z.B. sichergestellt werden, dass die Röntgenröhre sofort abschaltet, wenn der Detektor aus unbestimmten Gründen keine Projektionen mehr aufnehmen kann. Des Weiteren muss sichergestellt werden, dass keine Gefahr durch drehbare Teile oder von der Hochspannung des Generators der Röntgenröhre ausgehen kann. Zusammengefasst ergeben sich für das Steuerungskonzept des Brust-CT-Scanners folgende, den Gesamtspezifikationen entnommene Anforderungen:

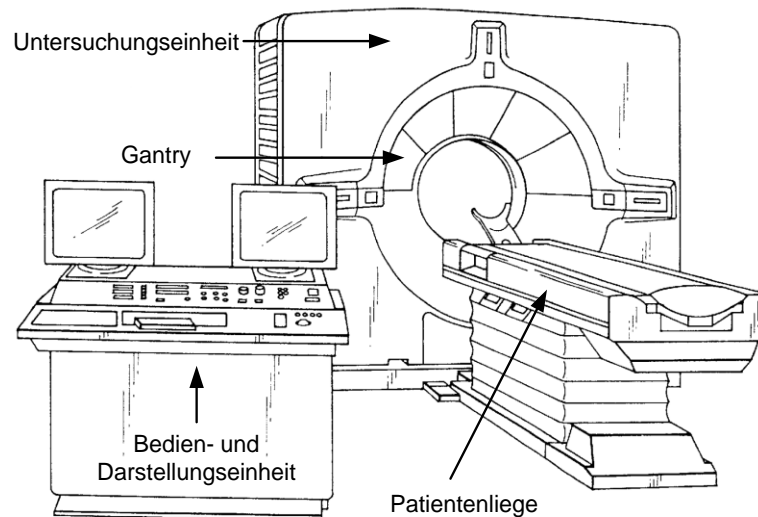
- Berücksichtigung neuartiger Bildgebungskomponenten durch die Steuerung
- Auslegung der Steuerung entsprechend gültiger Normen für die Medizintechnik
- Synchronisation von Positionserfassung, Röntgenröhre und Detektor
- Signallaufzeiten der Synchronisationssignale kleiner als  $10 \mu\text{s}$
- Jitter der Synchronisationssignale kleiner als  $1 \mu\text{s}$
- Übertragung der Projektionsdaten mit mindestens 7 GBit/s
- Einbindung eines adäquaten Sicherheitskonzepts

## 2 Grundlagen und Techniken

In den folgenden Abschnitten finden sich Grundlagen, die zur Lösung der Aufgabenstellung beigetragen haben, bzw. Techniken, die hierbei Anwendung fanden. Hierfür wird zunächst die technische Realisierung klinischer CT-Scanner erläutert. Daran anschließend werden Begriffe aus der Prozessautomatisierung eingeführt, da die Lösung der Aufgabenstellung hiervon stark beeinflusst ist. Da es sich bei einem CT-Scanner um ein sicherheitskritisches System handelt, welches unter anderem Echtzeitbedingungen erfüllen muss, wird der Begriff der Echtzeitsysteme definiert sowie deren Kommunikation untereinander beschrieben. Die letzten Abschnitte dieses Kapitels befassen sich mit den im weiteren Verlauf dieser Arbeit verwendeten Schnittstellen CAN, Ethernet und RS485, sowie deren Echtzeitverhalten.

### 2.1 Technische Umsetzung eines klinisches CT-Scanners

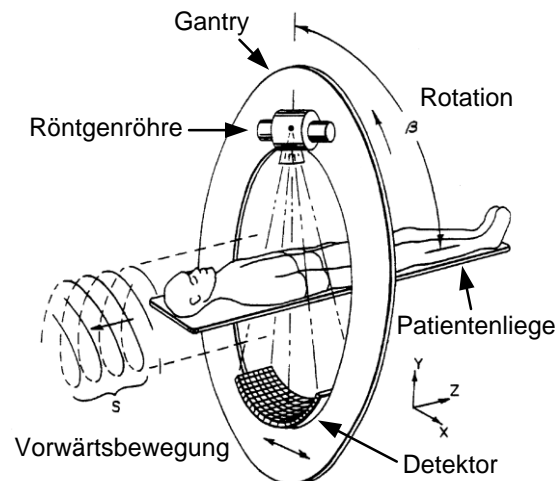
Ein konventioneller Ganzkörper-CT-Scanner, dargestellt in Abbildung 4, besteht aus Bedieneinheit mit Darstellungseinheit, Untersuchungseinheit und Patientenliege. Die Untersuchungseinheit beinhaltet unter anderem die Bildgebungskomponenten.



**Abbildung 4:** Ein konventionellen Ganzkörper-CT-Scanner aus [100]: Er besteht aus Bedieneinheit mit Darstellungseinheit, Untersuchungseinheit und Patientenliege.

Der zu untersuchende Patient ruht auf einer Patientenliege, welche sich vor- und zurückfahren lässt. Um den Patienten drehbar angeordnet befindet sich, entsprechend Abbildung 5, eine Einheit aus Röntgenröhre und Detektor, die als Gantry bezeichnet wird.

Vor Durchführung eines Scans wird dieser an der Bedieneinheit von einem Benutzer parametrisiert. Nach Einstellung der erforderlichen Parameter kann der Scan gestartet werden. Die Gantry, angetrieben durch einen Elektromotor, beginnt mit der Rotation um den Patienten. Nach Erreichen der Sollfrequenz sendet die Röntgenröhre fächerförmig Röntgenstrahlung aus, die nach Durchdringen des Patienten auf den auf der gegenüberliegenden Seite angebrachten Detektor trifft. Der Detektor besteht aus einem Sensorarray (unterteilt in Zeilen und Spalten) zur Messung der Röntgenintensität, sowie der dazu nötigen Ausleseelektronik. Durch die fächerförmige Röntgenstrahlung wird bei jedem Auslesevorgang des Detektors genau eine Projektion aufgenommen. Diese kann sich, abhängig von dem jeweiligen Detektor, über eine oder mehrere Schichten erstrecken. Während des Scans bewegt sich die Liege mit dem Patienten gemäß Kalender et al. [130], wie in Abbildung 5 gezeigt, kontinuierlich vorwärts (Spiral-CT), wodurch es zu einer unterbrechungsfreien Aufnahme von Projektionsdaten kommt. [88; 103; 106; 126; 128]



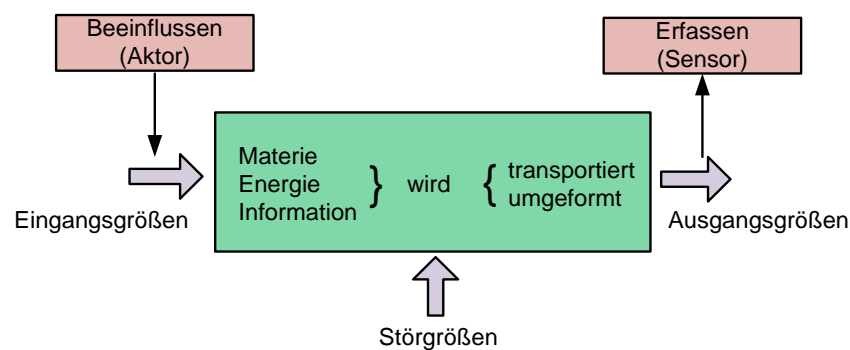
**Abbildung 5: Prinzipzeichnung einer Gantry mit Patientenliege aus [107] gemäß [130]:** Die Liege bewegt sich kontinuierlich vorwärts während sie von der Gantry umfahren wird – es entsteht eine spiralförmige Vorwärtsbewegung.

Die aufgenommenen Projektionsdaten werden während des Scans in Echtzeit oder gepuffert nach dem Scan an einen weiterverarbeitenden Computer gesendet. Dieser rekonstruiert durch Rückprojektion der Projektionsdaten die Schichtbilder und stellt sie dem Benutzer an einem Monitor zur Ansicht dar. Zur Energieversorgung sowie zum Austausch von Daten und Signalen mit Systemen auf der Gantry kommt ein Drehübertrager, z.B. in Form eines Schleifrings, zum Einsatz. Damit ist es der Gantry möglich, beliebig viele Umdrehungen auszuführen, ohne auf Schleppkabel Rücksicht nehmen zu müssen. [88; 103; 106; 126; 128]

## 2.2 Prozessautomatisierung

### 2.2.1 Technischer Prozess

Ein Prozess ist nach DIN 66201 [45] jeder Vorgang durch den eine Umformung, ein Transport oder ein Speichern von Materie, Energie oder Information durch ein technisches System erfolgt. Ein technischer Prozess verfügt zusätzlich noch über Eingangsgrößen, die mit Aktoren beeinflusst werden können, sowie über Ausgangsgrößen, die sich mit Sensoren erfassen lassen. Größen, welche den Prozess unerwünscht beeinflussen, werden als Störgrößen bezeichnet. [145; 189; 240]



**Abbildung 6:** Schematische Darstellung eines technischen (Teil)Prozesses angelehnt an [17]: Seine Ein- und Ausgangsgrößen sind durch Sensoren erfassbar und von Aktoren beeinflussbar.

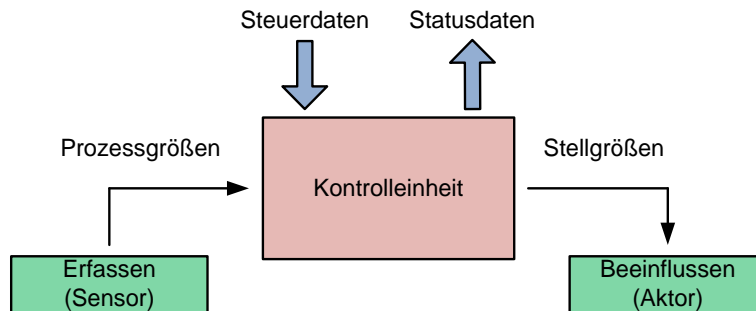
Im Falle des Brust-CT-Scanners sind die Prozesse primär durch die durch den Benutzer auslösbaren Aktionen wie CT-Bilddatensätze Aufnehmen, Übertragen und Rekonstruieren gekennzeichnet. Diese Prozesse lassen sich in verschiedene Teilprozesse unterteilen. Jeder einzelne dieser Teilprozesse ist durch das oben stehende Schema, Abbildung 6, darstellbar. Bei der Ausführung eines Scans wird beispielsweise durch Beschleunigen der Gantry elektrische Energie in Bewegungsenergie umgeformt und beim Auslesen der Bilddaten wird Information transportiert. [17]

### 2.2.2 Prozesssteuerung durch Steuerungseinheit und Kontrolleinheit

Quade [189] definiert eine Steuerung als die Summe der Rechenprozesse und deren Ablaufumgebung, bestehend aus Hardware und Software, zur Regelung, Führung und Optimierung eines Prozesses. Hierzu zählen:

- Erfassung der Zustandsgrößen des technischen Prozesses
- Beeinflussung des technischen Prozesses
- Koordination der Prozessabläufe
- Überwachung der Prozessabläufe

Die Erfassung von Zustandsgrößen eines technischen Prozesses erfolgt mittels Sensoren. Die erfassten Größen werden typischerweise von einer prozessnahen Kontrolleinheit zugeführt, die diese verarbeitet und den Prozess über Aktoren entsprechend beeinflusst, wie in Abbildung 7 gezeigt. Eine Kontrolleinheit kann, wenn der von ihr kontrollierte Prozess nur einen Teilprozess des Gesamtprozesses darstellt, einer übergeordneten Steuerungseinheit untergeordnet sein. Diese sendet Steuerdaten zur Kontrolleinheit, die der Koordination der Prozessabläufe dienen, und empfängt Statusdaten zur Überwachung dieser. [145; 189; 205; 240]

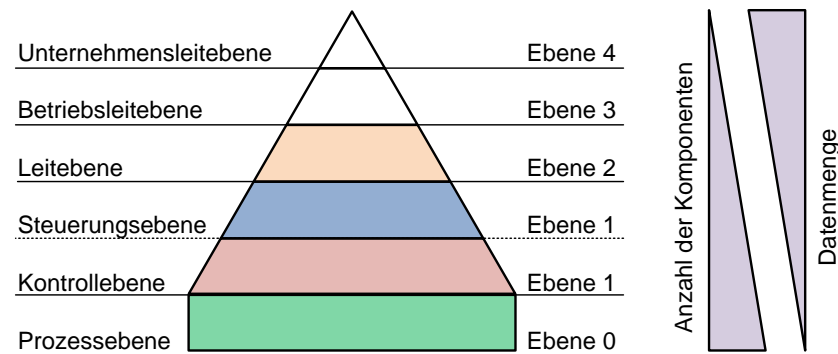


**Abbildung 7:** Schematische Darstellung einer Prozesssteuerung angelehnt an [205]: Sie ist als Kontrolleinheit ausgeführt und kommuniziert mit einer überlagerten Steuerungseinheit.

### 2.2.3 Ebenenmodell in der Prozessautomatisierung

Die Automatisierungspyramide ist ein mehrstufiges Ebenenmodell, das sich zur Beschreibung eines zu automatisierenden Gesamtprozesses durchgesetzt hat. Durch das Ebenenmodell wird der zu automatisierende Gesamtprozess hierarchisch in vier Ebenen unterteilt, welche in der Norm DIN EN 62264-1 [53] definiert sind. Diese Struktur dient dem leichteren Umgang mit komplexen Steuerungs- und Automatisierungslösungen und erhöht die Übersicht durch Schaffung einer Steuerungshierarchie. Zwar wachsen durch die hierarchische Struktur die Datenmengen, welche untere in höhere Ebenen kommunizieren, die Anzahl der benötigten Komponenten, sowie die Echtzeitanforderungen an diese, sinken jedoch. [77]

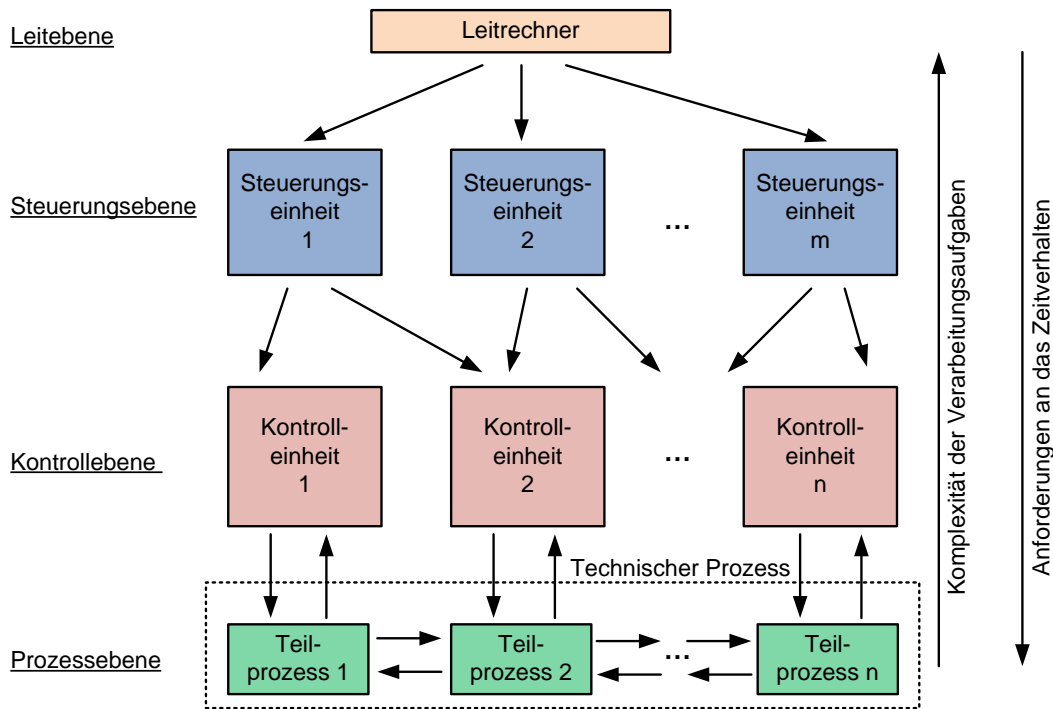
Die Automatisierung eines Prozesses ist nicht strikt an diese Ebenen gebunden und auch eine Implementierung aller Ebenen ist nicht zwingend erforderlich, sie sollen ausschließlich als Orientierung dienen. Die Namen der jeweiligen Ebenen sind in der Norm ebenfalls nicht vorgegeben. In der Norm sowie in der Literatur finden sich zahlreiche Beispiele für Automatisierungspyramiden, die sich in der Namensgebung der Ebenen unterscheiden und auf den jeweiligen Prozess bzw. Automatisierungsbereich angepasst sind. [166] Abgeleitet von Katzke [133] und der Norm DIN EN 62264-3 [54] sind die Ebenen in dieser Arbeit entsprechend Abbildung 8 benannt.



**Abbildung 8: Die Automatisierungspyramide angelehnt an die Norm DIN EN 62264-3 [54] und [133]:** Sie besteht aus vier abstrakten Ebenen zur Automatisierung und dem eigentlichen Prozess.

Die ersten beiden Ebenen, Unternehmens- und Betriebsleitebene, beschreiben strategische Funktionen zur Führung eines Unternehmens. Sie sind für die Steuerung kleinerer Anlagen oder Systeme nicht von Bedeutung und hier nur der Vollständigkeit halber aufgeführt. Die Leitebene hingegen überwacht und steuert übergeordnet den Gesamtprozess und interagiert mit dem Benutzer. Sie nimmt dessen Aufgaben an das Gesamtsystem entgegen und zeigt ihm System- und Statusmeldungen an. Ihr unterlagert ist die Steuerungsebene, die von der Leitebene ausgehende Teilaufgaben entgegennimmt und die dazu nötigen Aktionen auslöst. Hierzu kommuniziert sie mit den entsprechenden Kontrolleinheiten der Kontrollebene, die komplexe Steuer- und Regelaufgaben übernehmen und deren Status an die Steuerungsebene zurückmelden. Die Kontrollebene ist der Prozessebene direkt überlagert, weshalb ihre Kontrolleinheiten über Aktoren und Sensoren verfügen, um die von ihnen kontrollierten Prozesse direkt zu beeinflussen bzw. zu überwachen. Die Prozessebene bildet als unterste Ebene die Summe aus den jeweiligen Teilprozessen. [29; 62; 145; 185; 196–198]

Eine mögliche Umsetzung der abstrakt formulierten Automatisierungspyramide zeigt Abbildung 9. Zu erkennen ist ein dreistufiges Ebenenmodell, das nur die Ebenen 0–2 verwendet. Hierbei wird Ebene 2 von einem Leitreechner repräsentiert, der mit mehreren Steuerungseinheiten in Verbindung steht. Diese wiederum kommunizieren mit Kontrolleinheiten, die jeweils einen Teilprozess des Gesamtsystems kontrollieren. [240]

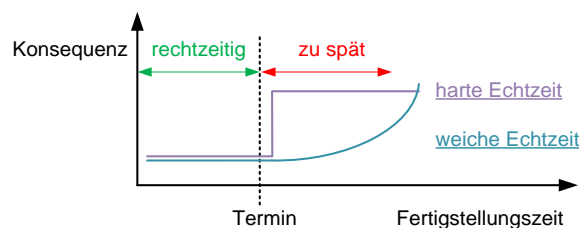


**Abbildung 9:** Eine mögliche Umsetzung der Automatisierungspyramide nach [240]: Das dreistufige Ebenenmodell besteht aus Leit-, Steuerungs- und Kontrollebene.

## 2.3 Echtzeitsysteme

### 2.3.1 Echtzeitfähigkeit

Echtzeit ist nach der Normenreihe DIN 44300 definiert als der Betrieb eines Rechensystems, bei dem Programme zur Verarbeitung anfallender Daten ständig betriebsbereit sind. Betriebsbereit in dem Sinne, dass die Verarbeitungsergebnisse innerhalb einer vorgegebenen Zeitspanne verfügbar sind. [44] Ein Echtzeitsystem ist somit ein System, das fähig ist, eine bestimmte Aufgabe in einem bestimmten Zeitfenster abzuarbeiten. Hierbei ist nicht entscheidend, eine Aufgabe möglichst schnell auszuführen, sondern eine festgelegte Zeitanforderung einzuhalten (Rechtzeitigkeit). Die Abarbeitung einer auszuführenden Aufgabe erfolgt rechtzeitig, wenn die entsprechende Zeitanforderung eingehalten wurde. Es wird zwischen harten und weichen Echtzeitanforderungen unterschieden, siehe hierzu Abbildung 10. [15; 140; 205]



**Abbildung 10:** Konsequenzen bei Nichterfüllen von Echtzeitschranken nach [15]: Die Konsequenzen bei Überschreiten der Fertigstellungszeit sind abhängig davon, ob es sich um harte oder weiche Echtzeit handelt, unterschiedlich schwer.

Wird bei einer harten Echtzeitbedingung eine Zeitschranke überschritten, so führt dies zu einem nicht tolerierbaren Fehler. Dabei kann unter anderem eine Gefahr für das System oder den Menschen ausgehen. In Bezug auf ein CT-System liegt eine harte Echtzeitbedingung vor, wenn ein Scan notfallbedingt abgebrochen werden muss. Hierzu schreibt die Norm DIN EN 60601-2-44 [48] vor, dass der Scanner innerhalb von 0,5 s vom Versorgungsnetz getrennt werden muss und Bilddaten nur noch bis zu einer zusätzlichen Gantryrotation aufgenommen werden dürfen.

Eine weiche Echtzeitbedingung hingegen lässt das Überschreiten einer Zeitschranke zu, das System toleriert dies und funktioniert weiterhin. Je länger die Zeitschranke jedoch überschritten wird, desto größer ist auch die negative Auswirkung auf das System. Bei einem CT-Scanner liegt eine weiche Echtzeitbedingung vor, wenn der Benutzer einen Scan startet und es zu einer Verzögerung zwischen Benutzerstart und Systemstart kommt. Diese Verzögerung wird, abhängig von ihrer Länge, von einem Benutzer als mehr oder weniger störend empfunden.

Neben der Rechtzeitigkeit muss ein echtzeitfähiges System auch die Eigenschaft der Vorhersehbarkeit und des Determinismus aufweisen. Das System darf sich nur in fest definierten Zuständen befinden, welche durch seine Ein- und Ausgaben definiert sind. Auf gleiche Eingaben müssen immer gleiche Ausgaben folgen. Dies ist insbesondere dann wichtig, wenn zwei Aufgaben gleichzeitig anfallen. Das Echtzeitsystem muss in der Lage sein, diese nach einem festen und reproduzierbaren Schema abzuarbeiten. [15; 140; 189; 205]

Besondere Bedeutung kommt dem Begriff der Echtzeit bei einer angestrebten Zertifizierung des Brust-CT-Scanners als Medizinprodukt zu. Hierbei muss Prüfern einer benannten Stelle für Medizinprodukte (z.B. TÜV) nachgewiesen werden, dass die Steuerung des CT-Scanners Aufgaben rechtzeitig, vorhersehbar und deterministisch ausführen kann und damit die Eigenschaft der Echtzeitfähigkeit aufweist. Nur wenn dies nachweisbar ist, wird eine Freigabe des Scanners gewährt. Als Beispiel für die Nichtgewährung einer Freigabe sei die Auslieferung von neuen ICE3 Zügen an die Deutsche Bahn genannt. Bei Aktivierung der Bremsen benötigte das Bremssignal des ICE3 eine Sekunde bis zur tatsächlichen Umsetzung des Kommandos. Dies bedeutete, dass sich der Bremsweg bei 250 km/h um 70 m verlängerte und stellte somit einen nicht tolerierbaren Sicherheitsmakel dar. [95]

### 2.3.2 Ausprägung von Echtzeitsystemen

Echtzeitsysteme lassen sich in die folgenden, an Quade [189] angelehnten Kategorien einteilen:

- Eingebettetes System
- Speicherprogrammierte Steuerung (SPS)
- Standard-PC mit echtzeitfähiger Software

Eingebettete Systeme stellen seit den 1960er Jahren den Standard zur Steuerung von industriellen Prozessen dar. Hierbei handelt es sich um Systeme, die für einen bestimmten Aufgabenbereich entwickelt werden und auf dessen Anforderungen spezifiziert sind. Ein eingebettetes System besteht aus Prozessor, Speicher sowie speziellen Schnittstellen für weitere Peripherie. Eingebettete Systeme zeichnen sich außerdem dadurch aus, dass sie in das zu steuernde Gerät bzw. die Anlage integriert sind. Dadurch erfüllen sie genau die Funktionen und besitzen nur die Schnittstellen, die tatsächlich benötigt werden. Dies macht das eingebettete System besonders robust und in großen Stückzahlen sehr günstig. [20; 91; 205; 233]

Als Weiterentwicklung von fest verdrahteten Ablaufsteuerungen mit Schützen und Relais entstanden in den 1970er Jahren speicherprogrammierbare Steuerungen (SPS), welche über einen modularen Aufbau, basierend auf verschiedenen Baugruppen, verfügen und an den jeweiligen Aufgabenbereich angepasst werden können. Je nach Anforderung wird eine standardisierte Hauptbaugruppe („Central Unit“) ausgewählt und mit weiteren standardisierten Peripheriebausteinen (dezentral oder zentral) bestückt. Über Schnittstellenbausteine können sie mit ihrer Umwelt kommunizieren. Die Programmabarbeitung erfolgt bei SPSen zyklisch oder ereignisgesteuert auf einer meist firmenspezifischen Ablaufumgebung (Firmware). SPS werden mit den in der Norm DIN EN 61131-3 standardisierten Programmiersprachen AWL, FUP, KOP, AS und ST programmiert. [150; 239; 240]

Seit den 1990er Jahren ist es möglich, Computersysteme auf Basis von Standard-PC-Komponenten (z.B. x86-Architekturen), welche nicht für echtzeitfähigen Betrieb ausgelegt sind, mit speziell angepassten Betriebssystemen (z.B. Real-Time Linux [153], QNX Neutrino [188]) bzw. spezieller Echtzeitsoftware (z.B. RTX Windows Realtime Extension [57], Kithara Real Time Suite [136]) Echtzeitfähigkeit zu verleihen, woraus ein echtzeitfähiger Standard-PC resultiert. Der sich hieraus ergebende Vorteil liegt in den kostengünstigen und gut verfügbaren Hardwarekomponenten, sowie der bekannten Technik. Nachteilig wirkt sich aber aus, dass Hard- und Software nicht für Automatisierungs- oder Steuerungsaufgaben ausgelegt sind und somit weniger robust und für geringere Lebensdauern konzipiert sind.

Abhilfe schaffen hier spezielle Industrie-PCs, die diese Nachteile teilweise ausgleichen können, sich architektonisch aber nicht von einem Standard-PC unterscheiden. [202]

Tabelle 1 vergleicht die oben genannten eingebetteten Systeme mit SPS und echtzeitfähigem Standard-PC anhand ihrer Flexibilität, ihres Aufwands, der Peripheriekosten und ihrer Leistung in Bezug auf eine zu lösende Steuerungsaufgabe.

**Tabelle 1: Vergleich verschiedener Typen von Echtzeitsystemen.**

Kriterium	eingebettetes System	SPS	echtzeitfähiger Standard-PC
SW-Flexibilität	●	○	●
HW-Flexibilität	●	●	○
Aufwand	●	○	●
Kosten der Peripherie	○	●	●
Leistung	●	●	○

Legende: ● = hoch, ● = mittel, ○ = niedrig

Wie Tabelle 1 zu entnehmen ist, weist ein eingebettetes System eine sehr hohe Flexibilität und Leistung auf. Allerdings ist hiermit auch der größte Aufwand verbunden. Echtzeitsysteme auf Basis eines Standard-PCs liegen leistungsmäßig unter einer SPS oder einem eingebetteten System, sind aber kostengünstiger und mit weniger Aufwand realisierbar als eingebettete Systeme, was auf ihren hohen Bekanntheits- bzw. Verbreitungsgrad zurückzuführen ist. Die SPS hingegen ist als fertige Lösung mit dem wenigsten Aufwand verbunden und stellt einen guten Kompromiss zwischen einem eingebetteten System und einem echtzeitfähigen Standard-PC dar.

Zwischen den oben genannten Systemen ist nicht immer eine klare Abgrenzung möglich. Der Übergang ist, aus technischer Sicht, fließend und eine Trennung erfolgt meist durch ihren Einsatzbereich. So enthält eine SPS, wie ein eingebettetes System einen Mikroprozessor und ähnliche Peripherie. Erst die unterschiedliche Programmierung, der modulare Aufbau, das typisch robuste Gehäuse und ihr Verwendungsort in der Automatisierung industrieller Anlagen macht dieses Echtzeitsystem zur SPS. Ein Standard-PC wiederum kann mit bestimmter Software eine SPS emulieren, es wird dann von einer „Soft-SPS“ gesprochen. Ähnlich verhält es sich mit einem sogenannten „Embedded-PC“, welcher oft als eingebettetes System beschrieben wird, wobei es sich aber um einen Standard-PC handelt, der sich durch besondere Robustheit und ein spezielles Gehäuse auszeichnet.

### 2.3.3 Eingebettetes System

Die im vorherigen Abschnitt vorgestellte Gruppe der eingebetteten Systeme, im englischen unter der Bezeichnung „Embedded Systems“ bekannt, umfasst eine große Bandbreite an unterschiedlichen Systemen. Sie lassen sich in die drei folgenden Untergruppen unterteilen: [92]

- „Fixed Processor“-System
- ASIC-basierendes “System on a Chip” (SOC)
- FPGA-basierendes “System on a programmable Chip” (SOPC)

Eingebettete Systeme bestehen meist aus fest („fixed“) verlöteten elektronischen Bauelementen sowie einem oder mehreren fest verbauten Mikroprozessoren bzw. -controllern. Je nach Anforderung werden Mikroprozessor, Speicher, Schnittstellen und weitere Bauelemente ausgewählt. Kriterien für die Auswahl von Komponenten können Leistungsfähigkeit, Architektur, Baugröße, Preis und Bauart sein. Durch die Verwendung von standardisierten elektronischen Bauelementen und deren großer Vielfalt sind „Fixed Processor“-Systeme relativ kostengünstig und sehr flexibel zu entwickeln und zu fertigen. [20; 74; 91; 92; 233]

Der Trend in der Entwicklung von eingebetteten Systemen geht immer mehr dazu über, Baugruppen auf einem einzigen Chip zu integrieren. Als Chip eines SOC kommt meist ein IC (Integrated Circuit) oder ASIC (Application Specific Integrated Circuit), zum Einsatz, um so die Kosten für Bauelemente und deren Montage zu senken. Eine große Rolle spielen hierbei wiederverwendbare Funktionsblöcke eines Chips, so genannte IP-Cores. Sie können entweder aus fertigen Codeblöcken in einer Hardwarebeschreibungssprache (Soft-IP-Cores) oder Layout-Blöcken (Hard IP-Core), welche in ein ASIC eingebunden werden, bestehen. [227] IP-Cores sind meist an die Funktion elektronischer Bauelemente angelehnt und bilden diese nach. Es existiert eine große Bandbreite an quelloffenen (Open Source) sowie kommerziellen IP-Cores. Eine besondere Form eines IP-Cores stellt der Prozessor eines eingebetteten Systems dar. Liegt dieser als Soft-IP-Core vor, handelt es sich um einen Softcore-Prozessor. Da es sich bei SOC um einen speziell angepassten Chip handelt, ist ein eingebettetes System auf Basis eines ASICs bzw. ICs eine sehr effektive und kompakte Lösung für eine Steuerungsaufgabe. Durch den hohen Aufwand bei der Fertigung sowie bei Änderungen eignet sich diese Lösung meist nur für sehr große Stückzahlen, das heißt, wenn die Kosteneinsparungen bei der Fertigung größer als die Entwicklungskosten sind. [20; 91; 119; 227]

Seit einigen Jahren werden „Field Programmable Arrays“ (FPGAs) in immer größeren Stückzahlen und zu immer günstigeren Preisen produziert. Dies führte zur Etablierung einer neuen Art der eingebetteten Systeme, dem SOPC. Hierbei wird der Grundgedanke der SOC's beibehalten, also die Integration möglichst vieler elektronischer Baugruppen sowie des Prozessors auf einem Chip. Diese Integration erfolgt beim SOPC nicht auf einem IC oder ASIC, welche nach der Herstellung nicht mehr veränderbar sind, sondern auf einem FPGA. FPGAs haben den Vorteil der leichten (Re-)Konfigurierbarkeit und (Wieder-)Programmierbarkeit. Hierzu finden vorwiegend die beiden Hardwarebeschreibungssprachen Verilog sowie VHDL Anwendung. Genau wie bei SOC's lassen sich auch bei SOPC's IP-Cores sowie benutzerspezifische Hardware verwenden. Hieraus ergibt sich ein sehr hohes Maß an Flexibilität, da das Chipdesign änderbar bleibt, ohne dass „echte“ Bauteile hinzugefügt oder spezielle ASIC's hergestellt werden müssen. Die Anzahl und Komplexität dieser IP-Cores ist nur begrenzt durch die Anzahl der verschaltbaren Logikblöcke des verwendeten FPGAs. [20; 21; 171; 216; 244]

Tabelle 2 vergleicht die oben beschriebenen Systeme anhand verschiedener Kriterien. „Fixed-Processor“ und ASIC-basierte Systeme unterscheiden sich besonders bei der Entwicklungszeit bzw. den Entwicklungskosten. SOPC's hingegen sind bei Energieeffizienz und Leistung den beiden erstgenannten Systemen unterlegen, weisen jedoch eine sehr hohe Flexibilität und Rekonfigurierbarkeit auf.

**Tabelle 2: Vergleich verschiedener Typen von eingebetteten Systemen nach [21; 92].**

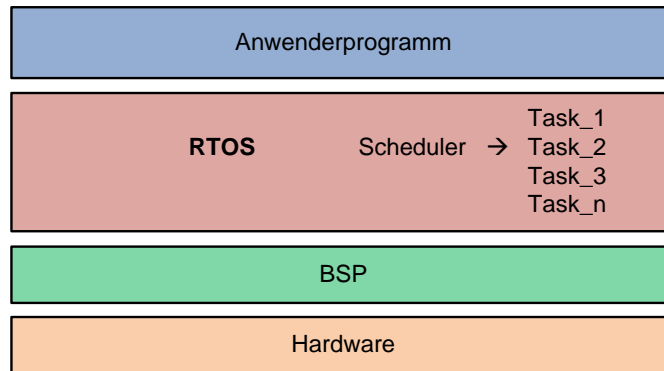
Kriterium	„Fixed-Processor	ASIC	SOPC
SW-Flexibilität	●	●	●
HW-Flexibilität	○	○	●
Rekonfigurierbarkeit	○	○	●
Entwicklungszeit/-kosten	●	○	●
Kosten der Peripherie	○	●	●
Leistung	●	●	○
Produktionskosten ( $n \gg 1$ )	●	●	○
Energieeffizienz	●	●	○

Legende: ● = hoch, ○ = mittel, ○ = niedrig

### 2.3.4 Echtzeitbetriebssysteme

Als Betriebssystem „Operating System“ (OS) werden nach DIN 44300 [44] diejenigen Programme eines digitalen Rechnersystems bezeichnet, die zusammen mit den Eigenschaften der Rechenanlage die Basis der möglichen Betriebsarten des digitalen Rechnersystems bilden und insbesondere die Abwicklung von Programmen steuern und überwachen. Hält ein Betriebssystem bei der Abarbeitung seiner Aufgaben harte Echtzeitbedingungen ein, handelt es sich um ein Echtzeitbetriebssystem bzw. ein „Real

Time Operating System“ (RTOS). Das (RT)OS befindet sich aus Systemsicht zwischen der Hardware, bestehend aus Prozessor und Peripherie, und dem Anwenderprogramm. Um ein RTOS an die entsprechende Hardware anzupassen, wird im Bereich der eingebetteten Systeme zusätzlich ein „Board Support Package“ (BSP) benötigt. Diese Softwarekomponente enthält die zum Betrieb der Hardware nötigen Schnittstellen und Treiber, wie Abbildung 11 verdeutlicht.



**Abbildung 11: Das RTOS zwischen Hardware und dem Anwenderprogramm nach [154].**

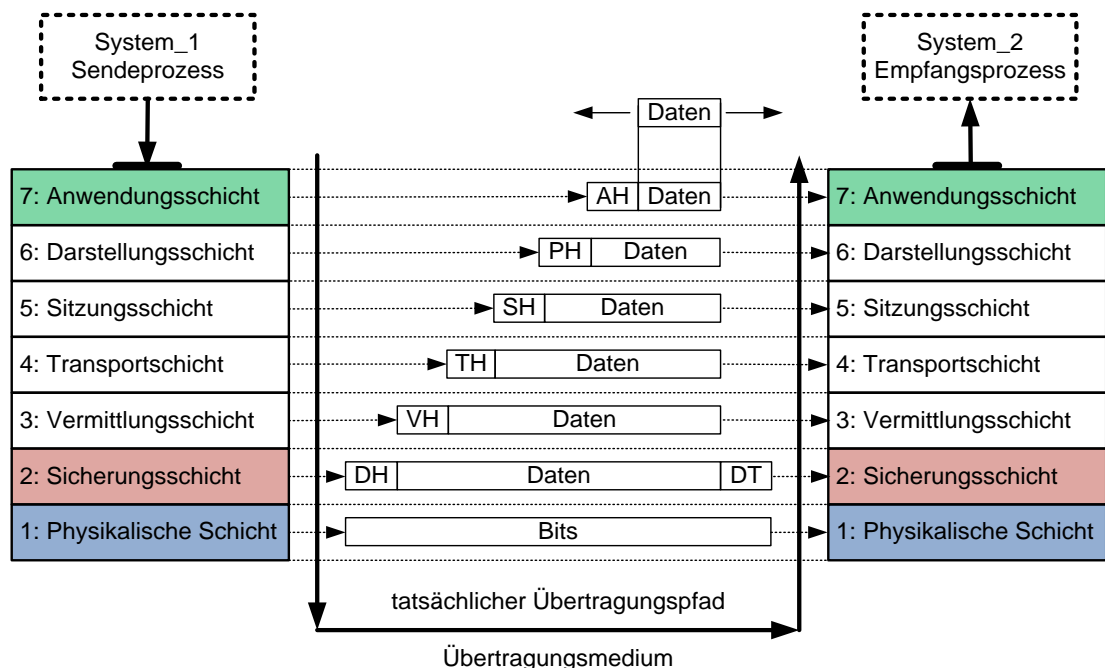
Wichtigster Bestandteil eines jeden RTOS ist der Scheduler, der die von der CPU zur Verfügung gestellte Rechenzeit auf die verschiedenen Bedürfnisse des Anwendungsprogramms verteilt. Dies geschieht durch Einteilung verschiedener Programmfunktionen des Anwenderprogramms in sogenannte Tasks oder Threads. Tasks können verschiedene Prioritäten zugewiesen werden. Anhand dieser Prioritäten legt der Scheduler fest, welcher Task wann und wie lange ausgeführt wird und verteilt so die Zugriffe auf die CPU. Kann ein Task in seinem zugewiesenen Zeitfenster seine Aufgabe nicht fertigstellen, wird seine Ausführung durch den Scheduler unterbrochen (Context Switch), die CPU einem anderen Task zugewiesen und der ursprüngliche Task erst zu einem späteren Zeitpunkt fortgesetzt. Dies geschieht nach speziellen Algorithmen, z.B. präemptiv oder Zeitscheiben-basierend. Somit können mehrere Tasks nahezu gleichzeitig ablaufen, auch wenn nur eine CPU vorhanden ist. Diese Eigenschaft eines OS wird auch als Nebenläufigkeit oder Multitasking bezeichnet.

Bei der Entwicklung einer Steuerung auf Basis eines eingebetteten Systems ist die Verwendung eines RTOS nicht unbedingt notwendig. Es erleichtert aber, ab einer gewissen Komplexität der Steuerungsaufgaben, den Entwicklungsprozess und hat Vorteile beim Nachweis der Systemsicherheit. Fällt die Entscheidung zugunsten der Verwendung eines RTOS, kann auf quelloffene, kommerzielle oder selbstentwickelte Produkte zurückgegriffen werden. Eine im Jahre 2005 durchgeführte Umfrage zur Verwendung von (RT)OS unter Systementwicklern [231] zeigte, dass eine klare Mehrheit von 44% ein kommerzielles RTOS verwendet, wobei die drei relevantesten Kriterien zur Auswahl eines solchen die Gesamtkosten, die Lizenzfreiheit sowie die Echtzeitperformance sind.

## 2.4 Echtzeitkommunikation

### 2.4.1 ISO/OSI-Referenzmodell

Das ISO/OSI-Referenzmodell (OSI-Modell) ist ein abstraktes Kommunikationsmodell, das 1977 entwickelt und 1983 von der internationalen Organisation für Normung (ISO) standardisiert wurde. Ziel war es, den Datenaustausch zwischen Computern zu vereinheitlichen, um herstellerunabhängig Protokolle zur Kommunikation entwickeln zu können. Hierzu wird die zur Kommunikation benötigte Funktionalität in eine Hierarchie aus verschiedenen Schichten aufgeteilt, wobei jede Schicht über ihr eigenes Protokoll verfügt. [36; 247]



**Abbildung 12: Das ISO/OSI-Referenzmodell zur Datenübertragung nach [240]:** Es besteht aus sieben Schichten, von der jede eine andere Aufgabe erfüllt.

Abbildung 12 zeigt das gängige Schichtenmodell in einer Beispieldatenübertragung von einem Sendeprozess in System\_1 zu einem Empfangsprozess in System\_2. Die zu sendenden Daten durchlaufen die Schichten sieben bis eins, von Anwendungsschicht bis physikalische Schicht. Sie werden anschließend Bit für Bit über das Übertragungsmedium gesendet und durchlaufen in System\_2 in umgekehrter Weise die physikalische Schicht bis zur Anwendungsschicht in Richtung Empfangsprozess. Jede Schicht hat hierbei eine bestimmte Aufgabe und überführt die von der vorherigen Schicht erhaltenen Daten in ihr eigenes Protokoll. Aufgrund der jeweils hinzugefügten Protokollinformationen, steigt die Datenmenge in Richtung der niedrigeren Schichten an. [36; 240; 247]

In der industriellen Echtzeitkommunikation zwischen Steuerungs- und Kontrollebene werden die Schichten drei bis sechs nur selten verwendet oder nur teilweise implementiert. [15; 183] Begründet wird dies durch die hohen Zeitanforderungen, sowie die ressourcenbeschränkten Endgeräte und den reduzierten Funktionsumfang. [15]

### 2.4.2 Kommunikationsmodelle

Das OSI-Referenzmodell zeigt, wie Daten zwischen zwei Systemen ausgetauscht, gesendet und empfangen werden. In welcher Beziehung diese Systeme zueinander stehen, zeigen die aus [15; 199; 223; 234; 250] bekannten Kommunikationsmodelle:

- Client/Server
- Producer/Consumer
- Publisher/Subscriber

Beim Client/Server-Modell steht der Client im Mittelpunkt, welcher Transaktionen auslöst und Informationen vom Server abrufen. Teilnehmer, die Informationen generieren, sind damit Server und antworten nur auf Anfragen eines Clients (Unicast oder Point-to-Point). Jeder Sender hat hierbei nur einen Empfänger. Bei Ausfall des Servers bleibt das System intakt, wenn die Clients auf andere Server gleicher Funktionalität zugreifen können. Beispiele für Bussysteme, die das Master/Slave-Modell verwenden sind Interbus, Profibus DP, P-Net und AS-Interface.

Beim Producer/Consumer-Modell steht die Information im Mittelpunkt. Es basiert darauf, dass ein Sender (Producer) eine Nachricht abschickt, die dann von einem oder mehreren interessierten Empfängern (Consumer) verarbeitet wird. Die Rollenverteilung von Producer und Consumer kann sich mit jeder Nachricht ändern. Durch das Senden einer Nachricht wird ein System für die Gültigkeit dieser Nachricht automatisch zum Producer. Hierbei ist nicht entscheidend, ob der Producer den oder die Consumer kennt oder umgekehrt. Die Nachricht wird von jedem Teilnehmer gelesen (Broadcast), der anhand des Inhalts der Nachricht über die Verarbeitung (Consumer) oder Nicht-Verarbeitung (kein Consumer) dieser entscheidet. Bei Ausfall eines Systems fällt abhängig von dessen versendeten Nachrichten nur ein Teil des Systems aus.

Eine erweiterte Version des Producer/Consumer-Modells ist das Publisher/Subscriber-Modell. Hierbei gehören Teilnehmer einer von zwei Gruppen an. Entweder der Gruppe der Publisher, welche Informationen senden, oder der Gruppe der Subscriber, welche Informationen empfangen. Wird eine Nachricht von einem Publisher versendet, wird sie ausschließlich von den Subscribern empfangen, die diese Nachricht verarbeiten (Multicast) können. Das Producer/Consumer- sowie das Publisher/Subscriber-Modell werden vorwiegend bei den Bussystemen CAN, CANOpen, DeviceNet, ControlNet, EIB und LON eingesetzt.

### 2.4.3 Medienzugriffssteuerung

Sind mehr als zwei Teilnehmer durch ein gemeinsames Übertragungsmedium verbunden, wird eine Koordination ihres Zugriffs benötigt. Hierfür stehen in der Echtzeitkommunikation verschiedene, aus [15; 189; 199; 212; 223] bekannte, Verfahren zu Verfügung, von denen die Wichtigsten kurz vorgestellt werden.

- Master/Slave
- Time Division Multiple Access (TDMA)
- Token-basiert
- Carrier Sense Multiple Access (CSMA)

Der Master/Slave-Medienzugriff stellt ein sehr einfaches und robustes Verfahren dar. Hierbei fragt ein Master-System mehrere Slave-Systeme in regelmäßigen Zyklen über ihren Zustand ab (Polling). Die Slaves dürfen nur auf Anfragen des Masters antworten, aber nicht selbstständig auf den Bus zugreifen. Sie verhalten sich passiv. Besonders Kommunikationstechnologien, die im Bereich der SPSen eingesetzt werden, wie z.B. Profibus DP, AS-Interface oder Interbus, verwenden diese Art der Zugriffssteuerung.

Beim zeitabschnittsbasierten Zugriff, auch als TDMA bezeichnet, erhält jeder Teilnehmer einen festen Zeitabschnitt, in dem er das Kommunikationsmedium nutzen darf. Hierzu wird entweder ein spezielles System zur Vergabe der jeweiligen Zeitfenster oder eine interne Uhr für jedes System, die mit allen anderen synchronisiert wird, benötigt. Zeitabschnittsbasierten Zugriff wird von TTP, SERCOS und ControlNet verwendet.

Token-basierte Verfahren steuern den Zugriff auf das Übertragungsmedium mithilfe eines eindeutigen Datums, dem Token, welches von einem System zum nächsten gesendet wird. Nur das System, welches das entsprechende Token besitzt, darf auf das Medium zugreifen. Möchte ein System eine Nachricht senden, muss dieses System warten bis es den Token (wieder) von einem anderen System übertragen bekommt. Token-basierte Zugriffssteuerungen finden sich z.B. in WorldFiP, Profibus FSM und P-Net. Sie werden häufig mit anderen Zugriffsverfahren, wie z.B. dem Master/Slave-Verfahren kombiniert, woraus sich ein Multi-Master-System ergibt.

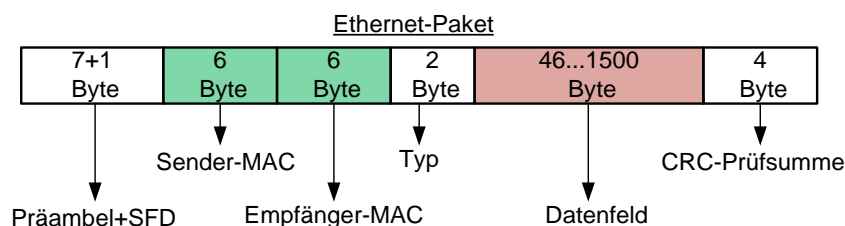
CSMA ist ein Konkurrenzverfahren, bei dem mehrere Stationen gleichzeitig auf das Medium zugreifen können. Die Stationen verfügen über Technologien, die es erlauben das Medium zu überwachen und nur zu senden, wenn kein Zugriff eines anderen Systems vorliegt. Senden zufällig zwei Systeme gleichzeitig, so bemerken die Stationen dies, und es setzt sich die höchstpriorie Station durch, CSMA/„Collision Avoidance“ (CSMA/CA), oder beide Stationen ziehen ihren Sendewunsch zurück CSMA/„Collision Detection“ (CSMA/CD). Bekannte Anwendungsbeispiele für CSMA/CA sind CAN, EIB sowie LON und für CSMA/CD insbesondere die Ethernet-Technologie.

## 2.5 Kommunikationsstandards in der seriellen Datenübertragung

### 2.5.1 Ethernet nach IEEE 802.3

#### 2.5.1.1 Übersicht

Ethernet ist eine Bustechnologie, welche in den 1970er Jahren entwickelt wurde und in den 1980er Jahren erstmals größere Verbreitung fand. Im Jahre 1985 entstand die Norm IEEE 802.3, welche sich vom ursprünglichen Ethernet in einigen Details unterscheidet, aber mit dem ursprünglichen Ethernet kompatibel ist. Ethernet hat sich im Sprachgebrauch als Synonym für den genormten Standard durchgesetzt und wird auch in dieser Arbeit entsprechend verwendet. [38] Ethernet nach IEEE 802.3 [111] definiert eine 10 MBit/s bis 100 GBit/s schnelle Datenübertragung auf den Schichten eins und zwei des OSI-Modells. Die physikalische Verbindung auf Schicht eins ist verantwortlich für die Übertragung der einzelnen Bits und wird mit einem der Übertragungsmedien Koaxialkabel (z.B. RG-8), „Twisted Pair“ oder Lichtwellenleiter realisiert. Die darauf aufbauende zweite Schicht fasst einzelne Bits der ersten Ebene zu Blöcken (Ethernet-Paketen) zusammen. [111]



**Abbildung 13: Aufbau eines Ethernet-Pakets entsprechend IEEE 802.3 [111].**

In Abbildung 13 ist ein Ethernet-Paket nach IEEE 802.3 dargestellt. Der Beginn eines Pakets wird durch die sieben Byte lange Präambel und dem ein Byte langen „Start Frame Delimiter“ (SFD) Feld signalisiert. Beide Felder dienen der Synchronisation von Sender und Empfänger. Daran anschließend befinden sich die „Media Access Control“-Adressen (MAC-Adressen) von Sender und Empfänger. Die MAC-Adressen sind eindeutig der jeweiligen Ethernet-Schnittstelle zugeordnet. Das zwei Byte lange Typ-Feld kennzeichnet die Art des Protokolls der nachfolgenden, nächsthöheren Schicht. In dem 46 bis 1500 Byte großen Datenfeld befinden sich die zu übertragenden Nutzdaten. Abgeschlossen wird ein Ethernet-Paket durch eine vier Byte lange „Cyclic Redundancy Check“-Prüfsumme (CRC-Prüfsumme) zur Erkennung von eventuellen Übertragungsfehlern. Die Prüfsumme umfasst die MAC-Adressen, das Typ-Feld und das Datenfeld des Pakets. [111]

Jede nächst höhere Schicht fügt einem Ethernet-Paket ihre eigenen Protokollinformationen an. Im Bereich der Internetkommunikation wäre das nächst höhere Protokoll auf Schicht drei das „Internet Protocol“ (IP). Daran anschließend folgt z.B. das „Transmission Control Protocol“ (TCP) oder „User Datagram Protocol“ (UDP) auf Schicht vier. Die Schichten fünf bis sieben werden repräsentiert durch z.B. das „Hypertext Transfer Protocol“ (HTTP) oder „File Transfer Protocol“ (FTP).

Aufgrund immer höherer Datenraten, speziell im Bereich von 10 bis 100 GBit/s, und immer zuverlässigerer Übertragung, ergab sich der Wunsch nach größeren Paketen, die somit auch über weniger Overhead verfügen. Diesem Wunsch entsprach die Einführung von sogenannten „Jumbo Frames“. Als „Jumbo Frames“ werden Ethernet-Pakete bezeichnet, deren Nutzdaten zwischen 1500 und 9000 Bytes liegen. Sie sind nicht Teil der Norm IEEE 802.3, da diese eine größere Nutzdatenmenge als 1500 Bytes nicht vorsieht. Vorteile beim Einsatz von „Jumbo Frames“ lassen sich deshalb nur erzielen, wenn die beteiligte Hard- und Software dafür ausgelegt ist. [28]

#### 2.5.1.2 Buszugriff und Echtzeitfähigkeit

Bei Ethernet wird über das CSMA/CD-Zugriffsverfahren auf das Übertragungsmedium, den Bus, zugegriffen. Das heißt, wenn zwei Stationen gleichzeitig ein Paket senden, kommt es zu einer Kollision. Beide Stationen erkennen diese, brechen die Übertragung ab und nehmen diese erst nach einer Wartezeit wieder auf. Diese Wartezeit ist zufallsverteilt, um eine erneute Kollision der beiden Stationen zu verhindern. Diese zufällige Verteilung macht Ethernet nicht deterministisch und damit auch nicht echtzeitfähig. [189] Aufgrund der weit verbreiteten Technologie und der kostengünstigen Komponenten gab es in den letzten Jahrzehnten immer wieder verschiedene Ansätze einer Realisierung von echtzeitfähigem Ethernet. Kern aller Ansätze ist die Vermeidung von Kollisionen auf dem Medium. Sind diese nicht zu verhindern, sollen die Konflikte minimiert oder eine deterministische Lösung für diese gefunden werden. Dies macht Ethernet zwar noch nicht echtzeitfähig, aber zumindest deterministisch, was als wichtige Voraussetzung für Echtzeitfähigkeit gilt. [38]

#### 2.5.1.3 Switched Ethernet

Eine Möglichkeit Kollisionen bei Ethernet vollständig zu vermeiden ist, Ethernet nur als Punkt-zu-Punkt-Verbindung zwischen zwei Systemen auszuführen. Aktuelle Ethernet-Transceiver besitzen einen Sende- und Empfangskanal, so dass jedes System Pakete gleichzeitig senden und empfangen kann. Eine Punkt-zu-Punkt-Verbindung zwischen mehreren Systemen lässt sich über Ethernet-Switches realisieren. Hierbei sind die Stationen sternförmig an ein Switch angeschlossen, welcher mithilfe einer Tabelle eine gegenseitige, vollduplexfähige Verbindung auf Schicht zwei des OSI-Modells zwischen genau einem Sender und einem Empfänger herstellt. Durch das sogenannte „Switched Ethernet“ können keine Kollisionen auf dem Medium auftreten, es kann allerdings

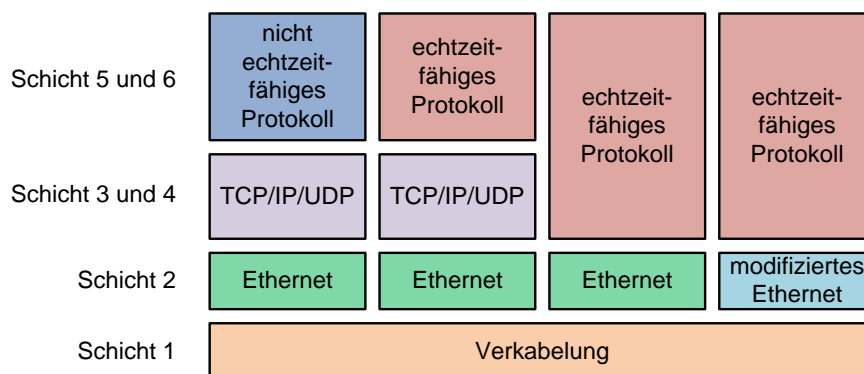
immer noch zu Paketverlusten und nichtdeterministischem Verhalten kommen, wenn sich mehrere Pakete unterschiedlicher Absender gleichzeitig an einen Empfänger richten. Der Switch besitzt zwar einen Puffer zur Zwischenspeicherung von Paketen, dieser ist aber begrenzt. Bei einem Pufferüberlauf werden überzählige Pakete durch den Switch verworfen. Um dieses Problem zu lösen erweitert IEEE 802.1Q [110] den Paketaufbau der IEEE 802.3 um vier Byte. Teil dieser Erweiterung ist unter anderem die Einführung eines drei Bit langen Prioritätsfelds „Priority Code Point“ (PCP). Dieses Feld dient der Priorisierung von Telegrammen und wird von einem entsprechenden Switch ausgewertet. Anhand der hier festgelegten Prioritäten delegiert ein Scheduler im Switch die Pakete an die Empfänger. Um einen dennoch möglichen Pufferüberlauf im Switch zu vermeiden, wird weiterhin eine Begrenzung der Datenmenge auf Seiten der Sender empfohlen. [66; 110; 120; 121; 147; 157; 160; 209; 245]

#### 2.5.1.4 Normierte Protokolle auf Basis von echtzeitfähigem Ethernet

Auf Basis des Ethernet-Standards entstand in den letzten Jahrzehnten eine Reihe von höheren Protokollen zur Umsetzungen von echtzeitfähigem Ethernet. Diese Ansätze sind in der Normenreihe DIN IEC 61158 beschrieben und lassen sich in drei Gruppen unterteilen: [70]

- Modifikation der zweiten Ethernet-Schicht
- echtzeitfähiges Protokoll aufgesetzt auf Schicht zwei
- echtzeitfähiges Protokoll aufgesetzt auf höherer Schicht (> Schicht zwei)

Ausgangspunkt ist Ethernet, wie es in IEEE 802.3 standardisiert ist. Im IT-Bereich finden meist nicht echtzeitfähige Protokolle wie HTTP oder FTP Verwendung, welche wiederum auf den Protokollen IP, TCP oder UDP aufsetzen, in Abbildung 14 ganz links dargestellt.



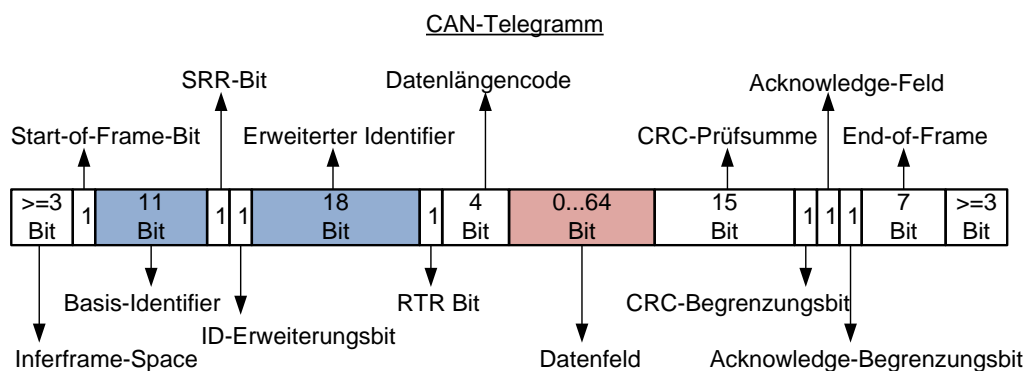
**Abbildung 14: Das Prinzip dreier Ansätze für echtzeitfähiges Ethernet nach [70]:** Die verschiedenen Ansätze setzen auf unterschiedlichen Ebenen des OSI-Modells auf. Zu Vergleichszwecken ist nichtechtzeitfähiges Ethernet ebenfalls angeführt.

Wird statt HTTP oder FTP ein echtzeitfähiges Protokoll benutzt, lässt sich weiterhin die darunterliegende Software sowie Hardware verwenden, allerdings können dann nur weiche Echtzeitbedingungen erfüllt werden, da darunterliegende Protokolle nicht für echtzeitfähigen Einsatz ausgelegt sind. Dies ist der Fall bei den Produkten MODBUS/TCP, Ethernet/IP, P-NET und Vnet/IP. Setzt ein echtzeitfähiges Protokoll direkt auf Schicht zwei auf, wie z.B. bei Ethernet Powerlink, TCnet, EPA und Profinet CBA, lässt sich auch weiterhin auf Standard-Hardware zurückgreifen. Modifikation der zweiten Schicht in Zusammenspiel mit einem echtzeitfähigen Protokoll stellt die effektivste Methode dar, Ethernet echtzeitfähig zu nutzen, da sich die zusätzliche Datenmenge durch den Wegfall weiterer Schichten verringert. Hierdurch lassen sich auch harte Echtzeitanforderungen erfüllen. Nach einer Modifikation auf Schicht zwei ist Ethernet aber nicht mehr mit dem Standard IEEE 802.3 kompatibel und es wird spezielle Hardware benötigt. Beispiele für diesen Ansatz sind SERCOS III, EtherCAT und PROFINET IO. [70]

## 2.5.2 Controller Area Network

### 2.5.2.1 Übersicht

„Controller Area Network“ (CAN) ist ein echtzeitfähiges, differentielles Bussystem, das 1983 von Bosch entwickelt wurde, um den Verdrahtungsaufwand in Kraftfahrzeugen zu reduzieren. In den 1990er Jahren, als CAN seine Zuverlässigkeit bereits durch jahrelangen Einsatz im Automobilbereich bewiesen hatte, wurde CAN auch in Industrie und Medizintechnik zum weit verbreiteten Standard echtzeitfähiger Kommunikation. CAN ist nach DIN EN 50325-1 spezifiziert und definiert wie Ethernet die Schichten eins und zwei des OSI-Modells. Mit CAN lassen sich Datenraten von 125 KBit/s bis 1 MBit/s, bei einer Nutzdatengröße von acht Byte pro Telegramm, realisieren. Als Übertragungsmedium auf Schicht eins ist „Twisted Pair“ spezifiziert. Die Bits der ersten Ebene sind in Ebene zwei zu CAN-Telegrammen zusammengefasst. [62; 68; 146; 192]



**Abbildung 15: Aufbau eines CAN-Telegramms entsprechend CAN 2.0B [192].**

Abbildung 15 veranschaulicht den Aufbau eines Datentelegramms nach CAN 2.0B. Ein Telegramm beginnt mit einem Start-of-Frame-Bit, welches der Synchronisation von Sender und Empfänger dient. Das Basis-Identifizier-Feld bildet zusammen mit dem erweiterten Identifizier-Feld den 29 Bit langen CAN-Identifizier (ID), der der eindeutigen Kennzeichnung des Telegramms dient. Absender und Empfänger sind hier nicht berücksichtigt, weshalb CAN in seiner ursprünglichen Form ein Producer/Consumer-Modell darstellt. Das Erweiterungsbit kennzeichnet die Verwendung der erweiterten ID nach CAN 2.0B. Das Datenfeld beinhaltet die bis zu acht Byte langen Nutzdaten. Die Länge der Nutzdaten darf nur ein Vielfaches eines Byte sein und wird durch den vier Bit Datenlängencode mit übertragen. Eine 15 Bit lange CRC-Prüfsumme, über die Felder von Start-of-Frame bis einschließlich des Datenfeldes, lässt die Empfänger eines Telegramms eventuelle Übertragungsfehler erkennen. Einen erkannten Fehler können Stationen mithilfe des, der CRC-Prüfsumme folgenden, Acknowledge-Feldes mitteilen. Hierzu legt eine Station, die ein Telegramm sendet, nach dem Senden der Prüfsumme ein rezessives Bit (logisch „1“) auf den Bus. Wird das Telegramm während des Sendens von allen Empfängern als fehlerfrei bestätigt, überschreiben diese das rezessive Bit des Senders mit einem dominanten Bit (logisch „0“). Empfängt eine Station dieses Telegramm als fehlerhaft, kann sie dies durch ein spezielles Fehlertelegramm mitteilen und den Sender zur Wiederholung des Telegramms auffordern. Abgeschlossen wird das Datentelegramm durch ein End-of-Frame-Bit. Neben dem gezeigten Datentelegramm existieren noch weitere Telegrammtypen mit deren Hilfe eine CAN-Station Fehler und Überlastungszustände allen anderen am Bus befindlichen CAN-Stationen automatisch mitteilen kann. [62; 68; 192]

#### 2.5.2.2 Buszugriffsverfahren und Echtzeitfähigkeit

Der Buszugriff erfolgt bei CAN durch das CSMA/CA-Verfahren. Möchte eine Station ein Telegramm über den Bus senden, wartet die entsprechende Station bis der Bus frei ist und beginnt mit der Übertragung. Die von der Station gesendeten Bits sind hierbei entweder dominant oder rezessiv. Beginnen zwei Stationen gleichzeitig mit dem Senden eines Telegramms, überschreiben dominante Bits des ID-Feldes der einen Station die rezessiven Bits des ID-Feldes der anderen Station. Ist dies der Fall, stoppt die unterlegene Station die Übertragung, ohne ein weiteres Bit zu senden. Das Telegramm mit dem niedrigsten Wert im ID-Feld, also der kleinsten ID, erhält somit die höchste Priorität. Diese resultierende Priorisierung von Telegrammen macht CAN deterministisch. Zusammen mit Fehler- und Überlastmechanismen ist der CAN-Bus deshalb für die Einhaltung harter Echtzeitbedingungen geeignet. [97; 189; 192; 224; 225]

### 2.5.2.3 CANopen

In den vergangenen Jahren entstanden eine Reihe von höheren Protokollen, die auf der CAN-Spezifikation von Bosch aufsetzen. Dazu zählen J1939, DeviceNet und CANopen. Besonders CANopen hat sich seit seiner Einführung im Jahre 1995 durch die Herstellervereinigung „CAN in Automation“ (CiA) in vielen Anwendungsgebieten, insbesondere in Europa, durchgesetzt und wurde in der Norm DIN EN 50325-4 standardisiert. [62; 183]

CANopen definiert auf Grundlage von CAN (spezifiziert in CiA201) ein objektorientiertes Protokoll (spezifiziert in CiA301), das Daten und Datentypen als Objekte behandelt. Jeder CANopen-Teilnehmer besitzt hierzu ein Objektverzeichnis (OD), in dem alle Objekte in einer Art indizierten Tabelle gespeichert sind. Andere CANopen Teilnehmer haben Zugriff auf das OD und wissen so, welche Objekte mit diesem Teilnehmer austauschbar sind. Die Objekte eines Teilnehmers sind abhängig von seiner Funktion im Prozess. CiA weist hierzu jedem Teilnehmer ein Geräte- oder Anwendungsprofil zu. Diese Profile sind in CiA401 bis CiA455 spezifiziert und unterliegen einem kontinuierlichen Entwicklungsprozess durch CiA. [22; 24; 183]

### 2.5.3 RS485

Anders als bei Ethernet und CAN, welche die Schichten eins und zwei des OSI-Modells definieren, handelt es sich bei RS485 um einen Schnittstellenstandard, der nur die erste Schicht des OSI-Modells, also die physikalische Verbindung, festlegt. Da sich mit RS485 bis zu 32 Stationen über ein Medium verbinden lassen, handelt es sich bei RS485, wie bei Ethernet und CAN, um ein Bussystem, jedoch ohne Protokoll zur Übertragung von Paketdaten. [150] Die serielle Übertragung von Daten erfolgt bei RS485 bitweise mit einer Datenrate von bis zu 10 MBit/s über ein differentielles Leitungspaar. Der Einsatz von differentieller Signalübertragung macht RS485 (genau wie CAN) besonders störsicher, da eingekoppelte Störungen durch Differenzbildung beider Leitungspotenziale nahezu aufgehoben werden können. [142] Realisiert wird eine RS485 Verbindung mit speziellen RS485-Transceivern, die über je einen Treiber- und einen Empfängerbaustein verfügen. Durch einen speziellen Eingang lässt sich ein Transceiver entweder in den Sende- oder Empfangsmodus schalten, wodurch mit RS485 nur wechselseitige, halbduplexfähige, Datenübertragungen möglich sind. Ein dem RS485 ähnlicher Standard ist RS422. Die technischen Unterschiede der beiden Schnittstellenstandards sind unwesentlich, jedoch ist RS422 nur teilweise mit RS485 kompatibel. Nicht zu verwechseln ist RS485 mit dem Schnittstellenstandard RS232, der nur für Punkt-zu-Punkt-Verbindungen über kurze Distanzen vorgesehen ist und für die serielle Kommunikation zwischen PCs und Modems eingesetzt wird. [74; 142; 150; 164; 210; 212]



### 3 Steuerungskonzept

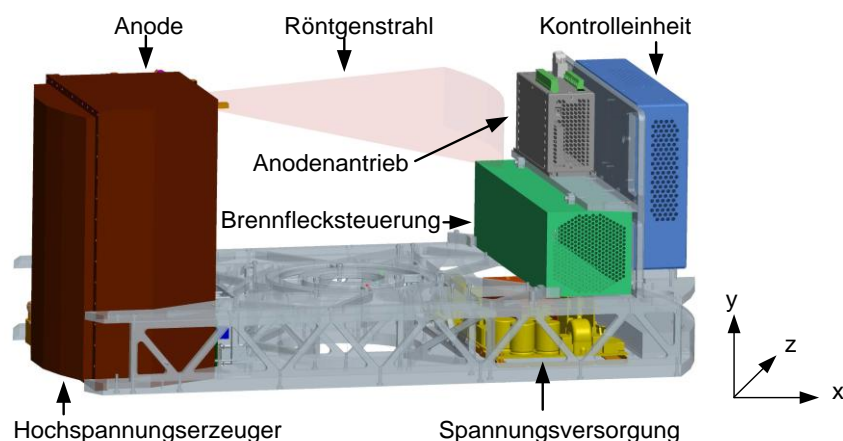
Das folgende Kapitel beschreibt ein neuartiges Konzept zur Steuerung eines Brust-CT-Scanners. Hierzu werden zuerst die für den Scanner entscheidenden Komponenten vorgestellt. Diese sind die auf der rotierenden Gantry befindliche Röntgenröhre und der Detektor, sowie auf der statischen Seite die Antriebe für Rotation, Translation und die Patientenliege. Ausgehend davon wird eine an das Ebenenmodell der Prozessautomatisierung der Industrie angelehnte, hierarchische Systemarchitektur eingeführt. Für die Kommunikation zwischen den jeweiligen Ebenen werden speziell entwickelte Protokolle auf Basis von Ethernet und CAN vorgestellt. Die Synchronisation der einzelnen Komponenten erfolgt mithilfe von RS485. Ebenfalls gezeigt wird ein Protokoll zur Übertragung von Projektionsdaten des Detektors zu einem Rekonstruktionsrechner.

#### 3.1 Zu steuernde Komponenten des Brust-CT-Scanners

##### 3.1.1 Röntgenröhre

###### 3.1.1.1 Übersicht

Bei der Röntgenröhre des Brust-CT-Scanners handelt es sich um eine eigens hierfür entwickelte Röhre mit Fächerstrahl-Geometrie und einem positionsgesteuerten Brennfleck. Zum Betrieb der Röntgenröhre benötigt diese einen für den Brust-CT-Scanner angepassten Hochspannungserzeuger, einen Anodenantrieb, sowie diverse Elektronik, wie in Abbildung 16 zu erkennen ist.



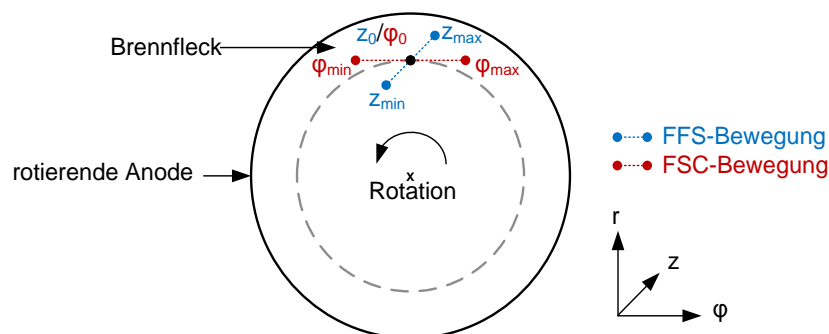
**Abbildung 16: Aufbau der Röntgenröhre aus mehreren Komponenten:** Diese sind der Anodenantrieb, der Hochspannungserzeuger, die Brennflecksteuerung, die Spannungsversorgung sowie die Kontrolleinheit.

Die durch die Röntgenröhre definierten Teilprozesse werden von der Kontrolleinheit der Röntgenröhre kontrolliert, die auch über eine Schnittstelle zur Steuerung des CT-Scanners verfügt. Da die Röntgenröhre ohne diese Teilkomponenten nicht funktionsfähig wäre, bezeichnet der Begriff Röntgenröhre im Weiteren die eigentliche Röhre inklusive ihrer Teilkomponenten.

### 3.1.1.2 Focal Spot Position Control (FSC)

Erzeugt wird die Röntgenstrahlung mithilfe eines thermisch geheizten Emitters, der Elektronen aussendet. Eine Hochspannung beschleunigt diese in Richtung einer Drehanode, an der durch Abbremsen der Elektronen Röntgenstrahlung entsteht. Die Entstehung der Röntgenstrahlung erfolgt am Brennfleck, dem Schnittpunkt zwischen Anodenteller und Elektronenstrahl. Zur Verbesserung der Bildqualität ist es üblich, den Elektronenstrahl durch ein elektromagnetisches Feld vertikal, in z-Richtung (siehe Koordinatensystem in Abbildung 17), abzulenken. Diese Ablenkung wird als „Flying Focal Spot“ (FFS) bezeichnet. Durch FFS wandert der Brennfleck auf der Anode und somit der Röntgenfokus im Drehzentrum um eine bestimmte Strecke, wie in Abbildung 17 blau gezeichnet. Werden während eines Scans, an der gleichen Winkelposition, zwei Projektionen kurz nacheinander aufgenommen, die Erste mit dem Brennfleck an der Ausgangsposition, sowie die Zweite mit dem Brennfleck an der Endposition, verdoppelt sich die Abtastrate in z-Richtung. [96; 123; 128; 201]

Bei der für den Brust-CT-Scanner entwickelten Röntgenröhre wird der Elektronenstrahl, anders als bei FFS, nicht in z-Richtung abgelenkt, sondern horizontal in  $\varphi$ -Richtung, wie in Abbildung 17 rot dargestellt. Diese Ablenkung des Fokus wird als „Focal Spot Position Control“ (FSC) bezeichnet. Die Bewegung des Brennflecks erfolgt kontinuierlich während einer Projektion von  $\varphi_{\min}$  nach  $\varphi_{\max}$  und entgegen der Rotationsrichtung. Die Verwendung eines solchen positionsgesteuerten Brennflecks ist dann sinnvoll, wenn der Detektor, wie im Brust-CT-Scanner der Fall, über einen speziellen Aufnahmemodus, einen „Time-Delayed Summation“-Modus, verfügt. Durch den Einsatz von FSC in Kombination mit TDS lässt sich eine erhebliche Verbesserung der Bildqualität, insbesondere in den peripheren Regionen eines Untersuchungsobjekts, erreichen. [176]

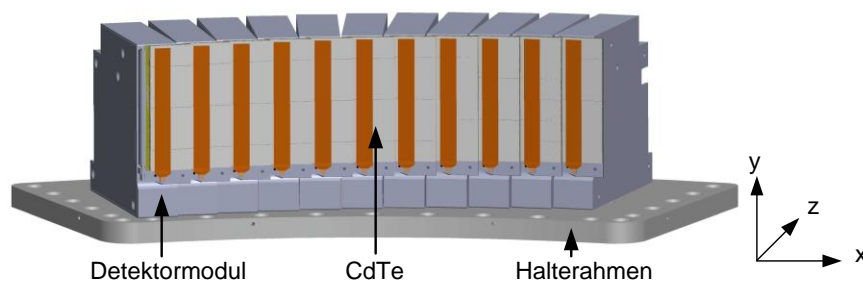


**Abbildung 17: Darstellung eines positionsgesteuerten Brennflecks:** Der Brennfleck ist bei FSC, anders als bei FFS in z-Richtung (blau), in  $\varphi$ -Richtung (rot) bewegbar.

### 3.1.2 Röntgendetektor

#### 3.1.2.1 Übersicht

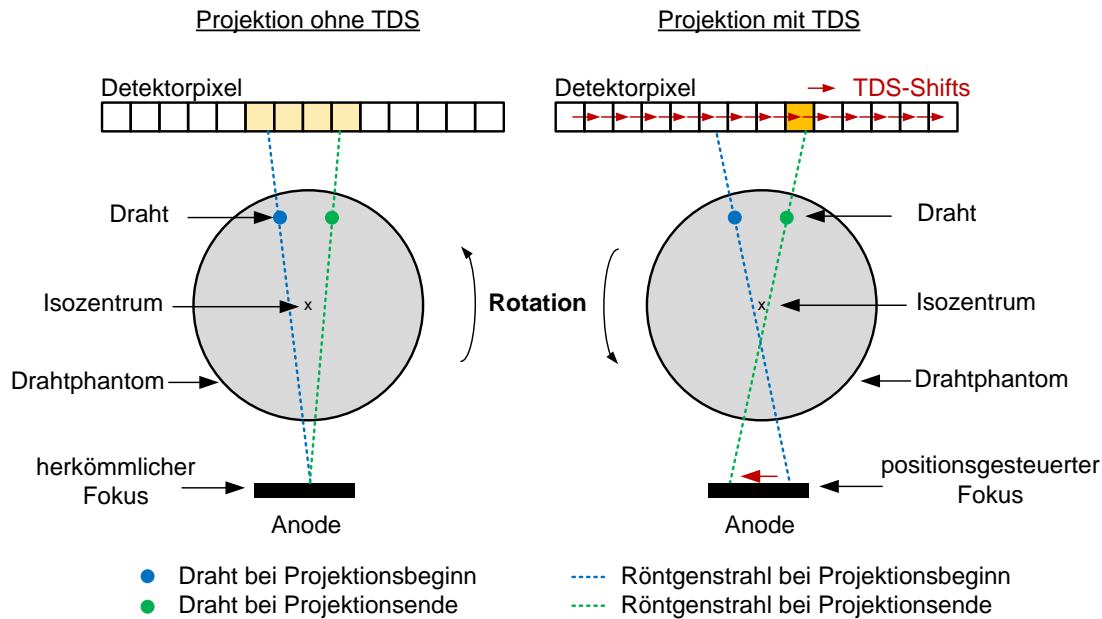
Anders als konventionelle Röntgendetektoren, welche die Röntgenintensität indirekt über den photoelektrischen Effekt mittels Fotodiode und Szintillationszähler (Szintillationsdetektor) messen, geschieht dies bei dem vorliegenden Detektor direkt über Einzelphotonenzählung durch Erzeugung von Elektronen-Loch-Paaren in der Sperrschicht eines Halbleiters (Halbleiterdetektor). Der speziell für den Brust-CT-Scanner entwickelte Detektor, Abbildung 18, verfügt hierzu über elf Einzelmodule. Jedes einzelne Detektormodul besitzt sechs Kacheln aus Cadmiumtellurid (CdTe) mit je  $256 \times 128$  Pixeln. Zum Auslesen der Photonenzählwerte befindet sich an jeder Kachel eine Ausleseelektronik auf Basis eines ASICs. Mit deren Hilfe lassen sich die registrierten Photonenzählwerte aller Kacheln eines Detektormoduls in ein als Kontrolleinheit dienendes FPGA übertragen. Jedes Detektormodul verfügt somit über ein FPGA als Kontrolleinheit und sechs ASICs mit je einer CdTe-Kachel. [85; 177; 219; 242]



**Abbildung 18: Der modulare Halbleiterdetektor des Brust-CT-Scanners.**

#### 3.1.2.2 Time-Delayed Summation

Charakteristisch für diesen Detektor ist außerdem der „Time-Delayed Summation“ (TDS)-Modus, mit dessen Hilfe sich Bewegungsunschärfe, die durch die Gantryrotation in den peripheren Bereichen eines Untersuchungsobjekts entsteht, reduzieren lässt. Um diese Unschärfe zu vermeiden, schiebt die Ausleseelektronik während einer Projektion von den Detektorpixeln registrierte Photonenzählwerte entgegen der Bewegungsrichtung des Detektors. Erfolgt die Verschiebung mit gleicher Geschwindigkeit wie die Gantryrotation, wird das Untersuchungsobjekt vor dem Detektor virtuell ortsfest gehalten und erscheint damit optisch scharf. Die Effektivität von TDS lässt sich noch steigern, wenn zusätzlich eine Röntgenröhre mit einem über FSC positionsgesteuerten Brennfleck Verwendung findet. [176; 177] Experimentelle „Charge-coupled Device“ (CCD)-Szintillationsdetektoren für die Mammografie sowie CMOS-Bildsensoren bei Luftbildkameras verfügen bereits über einen ähnlichen Modus zur Reduktion von Bewegungsunschärfe. Allerdings wird dieser aufgrund der andersartigen Detektortechnologie als „Time-Delayed Integration“ (TDI) bezeichnet. Abbildung 19 vergleicht die Aufnahme einer Projektion mit und ohne TDS und FSC. [149; 176; 177; 242; 243]

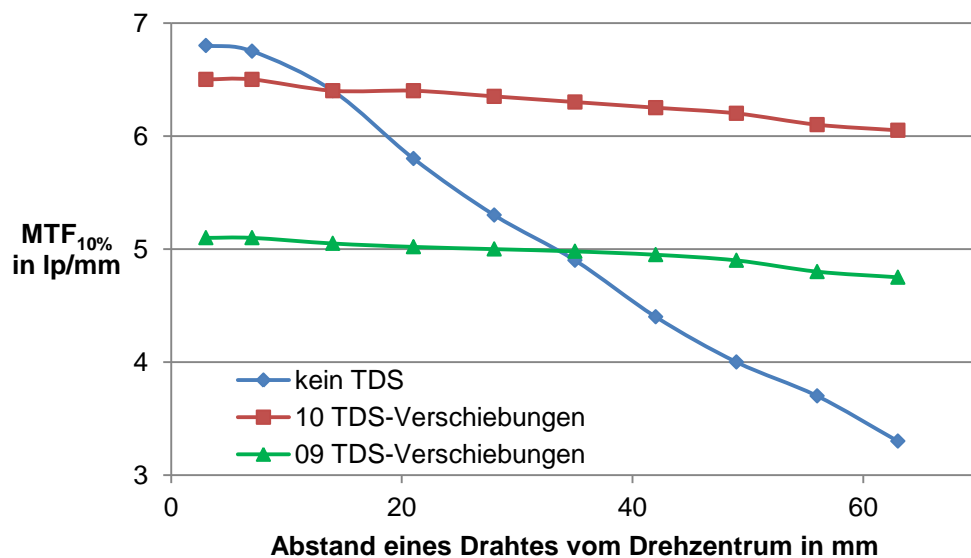


**Abbildung 19: Vergleich einer Projektionsaufnahme ohne und mit TDS und FSC nach [176]:** Ohne die Verwendung von TDS und FCS (Zeichnung links) wird der Draht eines Drahtphantoms auf mehrere Pixel abgebildet. Mit TDS und FCS (Zeichnung rechts) werden der Röhrenfokus und die Detektorpixel entgegen ihrer Bewegungsrichtung verschoben. Dadurch wird der Draht auf einem Pixel abgebildet und erscheint damit optisch scharf.

Die Zeichnung links zeigt die Aufnahme einer Projektion ohne die beiden Technologien. Hierbei wird bei Projektionsbeginn ein Draht von einem Röntgenstrahl (blau) durchstrahlt, der auf einen Detektorpixel (orange) trifft. Im Verlauf der Aufnahme bewegen sich Röntgenröhre und Detektor durch die Gantrybewegung um das Drahtphantom, wodurch der Röntgenstrahl weitere Pixel des Detektors belichtet. Bei Projektionsende hat der Röntgenstrahl den Draht (grün) auf vier Detektorpixel, und damit unscharf abgebildet. Bei Verwendung von FSC und TDS, Zeichnung rechts, werden der Brennfleck und damit der Fokus der Röntgenröhre während der Projektion entgegen der Bewegungsrichtung verschoben. Gleichzeitig werden die Pixelwerte aller Detektorpixel und damit auch der Pixelwert des von dem Röntgenstrahl (blau) bei Projektionsbeginn belichteten Pixels, ebenfalls entgegen der Bewegungsrichtung des Detektors vorschoben. Bei Projektionsende entspricht der Pixelwert des durch den Röntgenstrahl (grün) belichteten Pixels (orange) der Summe der vier belichteten Pixelwerte aus der Zeichnung links, ohne TDS und FSC. Der Draht erscheint dadurch, dass er nur auf einen Detektorpixel abgebildet wurde, deutlich schärfer.

Simulationen, die von Nowak et al. [176] zur Effektivität von TDS veröffentlicht wurden, zeigen, dass durch TDS die Ortsauflösung in den peripheren Regionen eines Untersuchungsobjekts deutlich erhöht werden kann. Abbildung 20 veranschaulicht hierzu ausgewählte Ergebnisse dieser Veröffentlichung bezogen auf die Geometrie des Brust-CT-Scanners. Simuliert wurden die 10%-Werte der Modulationsübertragungsfunktionen

( $MTF_{10\%}$ ) mit und ohne Verwendung von TDS und FSC in Abhängigkeit vom Abstand eines Drahtes vom Drehzentrum. MTF-Messungen sind in der CT zur Ermittlung der Ortsauflösung seit Jahren etabliert und nach ISO 15708 genormt. Die Ergebnisse der Simulationen zeigen ein kontinuierliches Sinken der  $MTF_{10\%}$  ohne die Verwendung von TDS (blau) mit steigendem Abstand vom Drehzentrum von ursprünglich 6,8 lp/mm bis auf 3,2 lp/mm. Unter Verwendung von FSC und, für diesen Simulationsaufbau idealen, zehn diskreten TDS-Verschiebungen pro Projektion (rot) ist die Auflösung nahe dem Drehzentrum minimal geringer als ohne TDS. Allerdings bleibt die  $MTF_{10\%}$  bei steigendem Abstand vom Drehzentrum fast konstant hoch ( $> 6$  lp/mm). In Abbildung 20 sind auch  $MTF_{10\%}$ -Werte einer nicht idealen Anzahl von neun TDS-Verschiebungen gezeigt. Ist die Zahl der TDS-Verschiebungen nicht ideal sind die  $MTF_{10\%}$ -Werte und somit auch die Ortsauflösung deutlich niedriger. [80; 122; 176; 211]



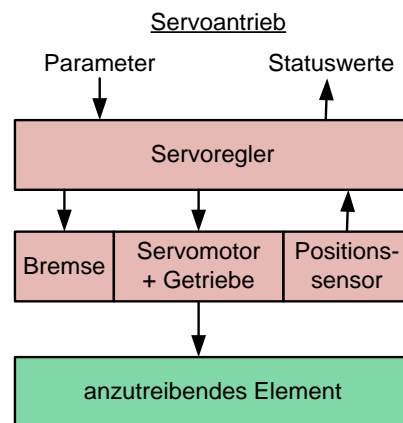
**Abbildung 20: Simulationsergebnisse zur Effektivität von TDS nach [176]:** Ohne TDS und FSC (blau) sinkt die Ortsauflösung mit zunehmendem Abstand vom Drehzentrum. Mit TDS-Verschiebungen (rot) bleibt die Ortsauflösung fast konstant.

### 3.1.3 Antriebe

Um die für einen Spiral-CT-Scanner typischen Scans ausführen zu können, verfügt der Brust-CT-Scanner über je einen Antrieb zur Rotations- und Translationsbewegung. Über einen dritten Antrieb lassen sich Patientenliege und Gantry gleichzeitig in der Höhe verstellen. Diese Antriebe sind als Servoantriebe ausgeführt, da ein Servoantrieb ein abgeschlossenes System bildet, welches vorgegebene Sollwerte durch selbständiges Regeln einhält. Ein solcher Antrieb, dargestellt in Abbildung 21, verfügt über einen Servoregler (Kontrolleinheit), der einen Servomotor antreibt.

Ein Servomotor bewegt in einem CT-Scanner selten direkt ein anzutreibendes Element, vielmehr wird sein Moment über Getriebe oder Riemen übertragen. Über eine integrierte Bremse lässt sich der Servomotor im Notfall sofort stoppen.

Zur Regelung des Motors verfügt dieser über einen Positionssensor, einen Winkellagegeber, der die aktuelle Position in Form eines digitalen Wertes erfasst und dem Servoregler zurückmeldet. Der Servoregler verfügt zur Parametrierung und Diagnose über eine Schnittstelle, von der aus ein übergeordnetes System (Steuerungseinheit) mit ihm kommunizieren kann. Über diese Schnittstelle lassen sich Statusdaten, wie z.B. Strom, Spannung sowie die aktuelle Position, auslesen und Steuerdaten setzen. [187; 238]

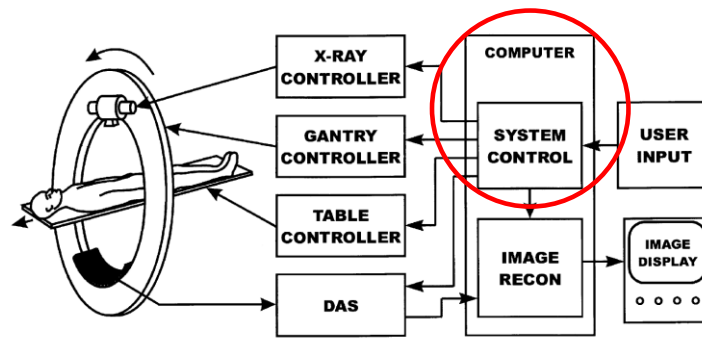


**Abbildung 21: Schematische Darstellung eines Servoantriebs:** Er besteht typischerweise aus einem Servoregler, einem Servomotor (mit Bremse und Getriebe) und einem Sensor zur Positionserfassung.

## 3.2 Systemarchitektur

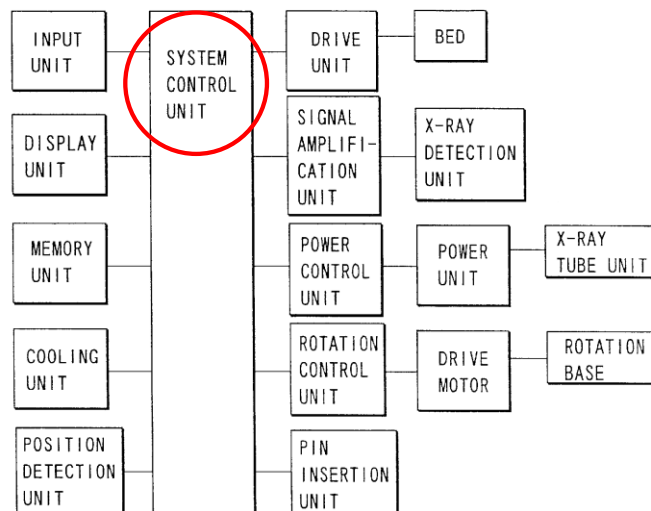
### 3.2.1 Stand der Technik

Eine vollständige Wiedergabe des Stands der Technik von Systemarchitekturen konventioneller CT-Scanner ist aufgrund der Verwendung von herstellerspezifischen Schnittstellen und der Geheimhaltung der jeweiligen Hersteller nicht in vollem Umfang möglich. Dennoch gibt eine umfangreiche Patentrecherche Einblicke in die Steuerungskonzepte bekannter Hersteller. Abbildung 22 und Abbildung 23 zeigen typische Architekturen, wie sie in Patentschriften zu finden sind. Gleiche oder ähnliche Architekturen sind auch in anderen Patentschriften sowie in Büchern als Stand der Technik zur Steuerung eines CT-Scanners aufgeführt, weshalb Abbildung 22 und Abbildung 23 hierfür als repräsentativ angesehen werden können. [10; 33; 37; 71; 79; 93; 100; 102; 107; 113; 135; 138; 158; 182; 229]



**Abbildung 22: Typisches Steuerungskonzept für einen CT-Scanner aus [135]:** Ein zentraler Computer steuert die Prozesse des CT-Scanners durch Interaktion mit den jeweiligen Komponenten.

In Abbildung 22 sind Patientenliege und Gantry, deren Bewegungen über jeweils einen Mikrocontroller „Table“ und „Gantry“ gesteuert werden, zu erkennen. Auf der Gantry befinden sich die Röntgenröhre und der Detektor. Die Röntgenröhre wird über eine in Abbildung 22 mit „X-Ray Controller“ bezeichnete Kontrolleinheit kontrolliert und die Projektionsdaten des Detektors über ein Datenerfassungssystem (DAS) ausgelesen. Ein zentrales Computersystem beinhaltet je eine Einheit zur Steuerung der Prozesse des Gesamtsystems „System Control“ sowie zur Rekonstruktion der Schichtbilder „Image Recon“. Benutzereingaben zur Parametrierung eines Scans sowie die Bilddarstellung der Schichtbilder erfolgt ebenfalls über dieses zentrale Computersystem. [135]



**Abbildung 23: Sternförmige Steuerungsarchitektur eines CT-Scanners aus [100]:** Die Komponenten des CT-Scanners sind sternförmig um eine zentrale Steuerungseinheit angeordnet.

Abbildung 23 zeigt ebenfalls ein Steuerungskonzept, bei dem alle Komponenten des CT-Scanners sternförmig um eine zentrale Steuerungseinheit angeordnet sind. Deutlich zu erkennen ist die große Anzahl an Schnittstellen, über die die Steuerungseinheit mit den verschiedensten Komponenten des CT-Scanners kommuniziert. [100]

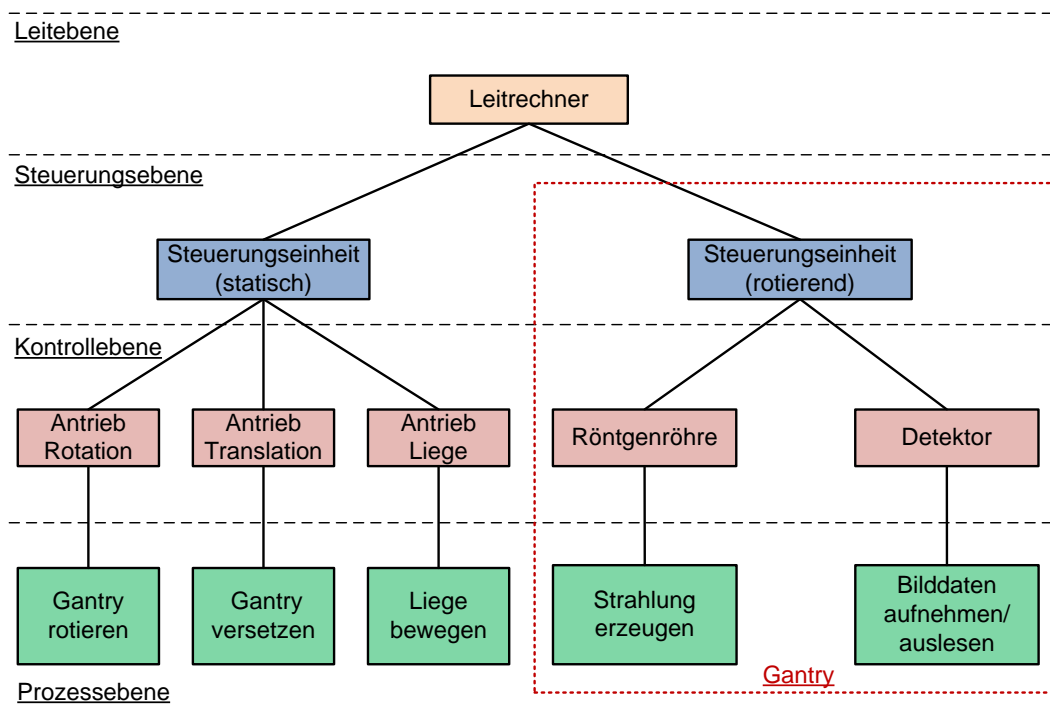
### 3.2.2 Probleme beim Stand der Technik

Aus dem Stand der Technik geht hervor, dass viele Steuerungskonzepte aus der grauen Literatur ein zentrales Computersystem zur Steuerung vorsehen. Bei diesen Computersystemen handelt es sich meist um für den jeweiligen Hersteller eines CT-Scanners angepasste, hochperformante Computersysteme auf Basis von Standard-PC-Komponenten oder speziellen eingebetteten Systemen, die Echtzeitanforderungen erfüllen können. Solche speziell entwickelte eingebettete Systeme eignen sich allerdings nicht für den Brust-CT-Scanner von IMP und CTI, da sie mit hohen Kosten verbunden sind und sich nur für Scanner eignen, deren Komponenten sowie Schnittstellen keinen Änderungsprozessen mehr unterworfen sind. Der Einsatz eines regulären PCs zur Systemsteuerung, der auf Standard-Komponenten basiert, im englischen auch als „Commercial Off-The-Shelf“-Komponenten (COTS) bezeichnet, ist ebenfalls nicht ideal. Ein Standard-PC verfügt im Allgemeinen nicht über ausreichend viele Schnittstellen und weist zu lange Reaktionszeiten für den Brust-CT-Scanner auf. Im Gegensatz zu einem kommerziellen CT-Scanner, bei dem große Hersteller Komponenten entweder selbst entwickeln oder an ihre bisherige Steuerungsarchitektur anpassen lassen, muss die Steuerungsarchitektur eines dedizierten Scanners an Komponenten von verschiedenen Herstellern anpassbar sein. Diese Forderung ist dadurch begründet, dass an einem neuartigen medizinischen Gerät, wie dem Brust-CT-Scanner, Bildgebungskomponenten verschiedener Hersteller zum Einsatz kommen bzw. getestet werden. Kann mit einer dieser Komponenten die Spezifikation nicht erreicht werden, oder gibt es inzwischen eine bessere Version dieser von z.B. einem anderen Hersteller, soll die entsprechende Komponente leicht zu ersetzen sein.

Dieses Problem wird durch ein hierarisches Steuerungskonzept gelöst, das sich an den hierarchischen Strukturen der Prozessautomatisierung orientiert. Hierzu wird die Steuerung des CT-Scanners in verschiedene Ebenen eingeteilt. Je ferner eine Ebene dem Prozess steht, desto geringer sind die (Echtzeit-)Anforderungen, die die in dieser Ebene enthaltenden Komponenten erfüllen müssen. Hierdurch muss nicht ein Computersystem alle Steuerungsaufgaben übernehmen, wie im Stand der Technik, sondern die Aufgaben verteilen sich auf verschiedene Systeme. Die einzelnen Komponenten müssen so nur nach den an sie gestellten Aufgaben und Anforderungen ausgewählt werden, wodurch sich eine erhebliche Kostenersparnis und eine erhöhte Flexibilität im Gegensatz zu einem zentralen Computersystem ergibt. Eine Änderung von z.B. einer Schnittstelle oder Komponente, zieht so nur noch eine Änderung einer Steuerungskomponente nach sich und nicht eine Änderung des gesamten Steuerungskonzepts bzw. -systems.

### 3.2.3 Ebenenmodell für den Brust-CT-Scanner

Auf Basis des in Kapitel 2.2 erörterten Ebenenmodells der Prozessautomatisierung zeigt Abbildung 24 eine dreistufige Steuerungshierarchie entsprechend der Norm DIN EN 62264 Anhang D [53] zur Steuerung der Prozesse des Brust-CT-Scanners. Jede Komponente des Scanners soll über eine eigene Kontrolleinheit verfügen, die die durch die Komponente definierten Prozesse kontrollier- und überwachbar macht. Die Komponenten bilden mit ihren jeweiligen Kontrolleinheiten ein abgeschlossenes System der Kontrollebene, das von einer übergeordneten Steuerungseinheit über eine definierte Schnittstelle und einen festen Befehlssatz angesprochen wird. Für den Brust-CT-Scanner bieten sich zwei Steuerungseinheiten an, die der Steuerungsebene zugehörig sind und räumlich zusammengehörige Komponenten zusammenfassen. Den beiden Steuerungseinheiten übergeordnet befindet sich auf der Leitebene ein Leitreechner, der mit dem Benutzer interagiert indem er Prozesse, wie z.B. das Auslösen eines Scans, initiiert und Statusmeldungen ausgibt.



**Abbildung 24: Das Ebenenmodell des Brust-CT-Scanners:** Es besteht aus einem Leitreechner, zwei Steuerungseinheiten und sechs Kontrolleinheiten.

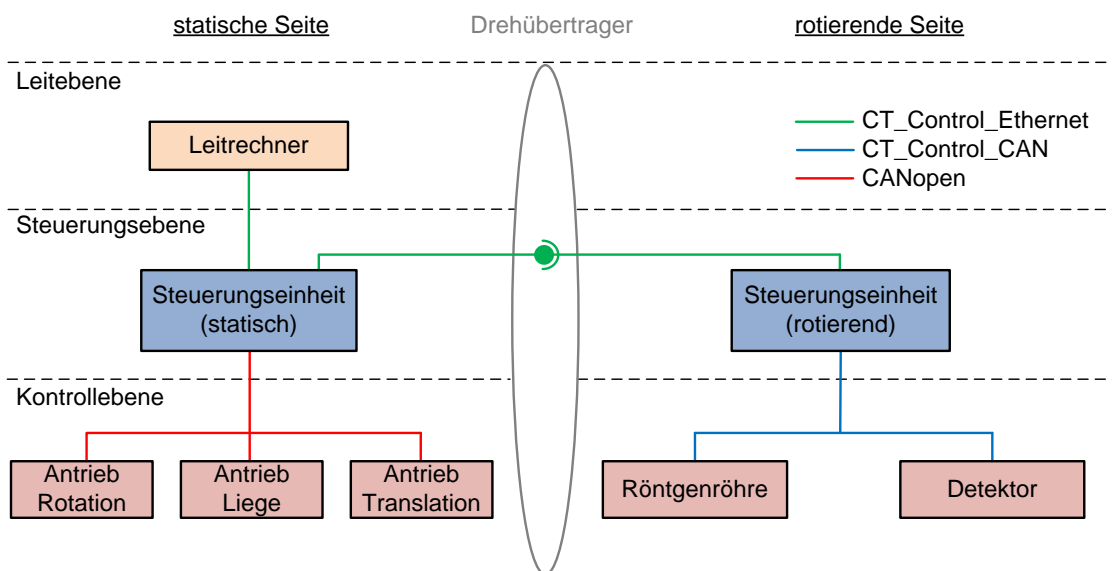
Die Zuordnung von Komponenten und Steuerungseinheiten erfolgt nach örtlichen Gesichtspunkten. Komponenten auf der statischen Seite sind einer statischen Steuerungseinheit, die der rotierenden Gantry sind einer rotierenden -einheit zugeordnet. Die jeweiligen Steuerungseinheiten steuern hierbei alle Teilprozesse der ihr unterlagerten Kontrolleinheiten, ohne dass diese mit anderen Komponenten kommunizieren müssen.

Der Datenaustausch von Steuerungsdaten zwischen der statischen und der rotierenden Seite erfolgt kontaktbehaftet über einen Drehübertrager. Die Übertragungsqualität eines Drehübertragers ist niedriger als bei einer direkten Verdrahtung und wird durch das Bitfehlerverhältnis (auch als Bitfehlerrate (BER) bezeichnet) ausgedrückt. Es beträgt bei kontaktbehafteter Übertragung ca.  $1 \times 10^{-9}$  [172] bzw. bei kontaktloser Übertragung ca.  $1 \times 10^{-12}$  [204]. Aufgrund der Übertragungsqualitätsminderung durch den Drehübertrager ist es von Vorteil nur Daten möglichst hoher Ebenen über den Drehübertrager auszutauschen, da die Datenmenge in Richtung höherer Ebenen sinkt und sich eine niedrigere Übertragungsqualität somit weniger stark auswirkt. Dies findet bei der Architektur des Brust-CT-Scanners Berücksichtigung, indem nur Daten der Steuerungsebene über den Drehübertrager ausgetauscht werden.

### 3.3 Steuerungsdaten-Kommunikationsmodell

#### 3.3.1 Systemübersicht

Das in Kapitel 3.2.3 eingeführte Ebenenmodell zeigt die hierarchische Aufteilung der Steuerung in mehrere Ebenen. Zur Kommunikation zwischen den Ebenen werden entsprechende Kommunikationstechnologien und -protokolle benötigt. Das für das Steuerungskonzept des Brust-CT-Scanners entwickelte Kommunikationsmodell ist in Abbildung 25 dargestellt.



**Abbildung 25: Übersicht über die Steuerungskommunikation des Brust-CT-Scanners:** Die Leit- und Steuerungsebene kommunizieren über ein Ethernet-Protokoll, die Steuerungs- und Kontrollebene über zwei verschiedene CAN-Protokolle.

Auf Kontrollebene sind die durch die Komponenten bestimmten Teilprozesse über ihre Kontrolleinheiten kontrollier- und überwachbar. Als Schnittstelle zwischen Komponenten der Kontrollebene und den Steuerungseinheiten der Steuerungsebene eignet sich

besonders der CAN-Bus. Er verfügt über hohe Flexibilität sowie hohe Übertragungssicherheit und dient als Grundlage für die ganzheitliche Kommunikation zwischen Kontrollebene und Steuerungsebene.

Für Komponenten der statischen Seite, im Falle des Brust-CT-Scanners sind dies die zugekauften Antriebskomponenten, ist die Auswahl an Protokollen und Schnittstellen begrenzt. Es empfiehlt sich die Verwendung eines offenen Standards, den idealerweise viele Hersteller verwenden. Im Bereich der Antriebskomponenten, wie sie für die Bewegungsabläufe des Brust-CT-Systems nötig sind, ist dies der CANopen-Standard. CANopen ist in Europa der mit am weitesten verbreitete, offene Schnittstellenstandard für Komponenten der Prozessautomatisierung. [62; 183] In Abbildung 25 sind Kommunikationsverbindungen mit CANopen rot eingezeichnet. Hierbei besteht auf Seiten der statischen Steuerungseinheit nicht das Bedürfnis nach der ganzen Funktionalität eines kompletten CANopen-Protokolls, die in Form eines sogenannten CANopen-Stacks kostenpflichtig beziehbar ist. Stattdessen ist es ausreichend nur einzelne Funktionen zu implementieren, die zum Betrieb der Antriebe nötig sind und aus deren Datenblättern hervorgehen.

Bei den Komponenten auf Seiten der rotierenden Gantry handelt es sich um die Röntgenröhre und den Detektor. Die Kontrolleinheiten dieser Bildgebungskomponenten sind via CAN von einer übergeordneten Steuerungseinheit ansprechbar. Anders als die Kontrolleinheiten der Antriebe sind sie aber nicht mit einem CANopen-Protokoll-Stack versehen, da dies ihre Entwicklungskosten deutlich erhöht hätte und sich CANopen aufgrund der Neuartigkeit der Komponenten nicht eignet. Stattdessen kommt ein speziell für den Brust-CT-Scanner entwickeltes Protokoll auf Basis von CAN, in Abbildung 25 blau dargestellt, mit dem Namen „CT\_Control\_CAN“ zum Einsatz. Eine genaue Beschreibung dieses Protokolls folgt in Kapitel 3.3.2.

Die Steuerungseinheiten der Steuerungsebene stehen mit dem Leitrechner der Leitebene über Ethernet in Verbindung. Protokolle auf Basis von Ethernet bieten sich für die Kommunikation zwischen Steuerungsebene und Leitebene besonders an, da Zeitanforderungen in Richtung höherer Ebenen abnehmen und Datenmengen ansteigen. Ethernet besitzt hierfür im Gegensatz zu CAN größere Paketdatenfelder und eine höhere Datenübertragungsrate. Außerdem handelt es sich bei einem Leitrechner typischerweise um einen Standard-PC, der bereits über eine Ethernet-Schnittstelle verfügt.

Zur deterministischen/echtzeitfähigen Kommunikation zwischen Leitrechner und mehreren Steuerungseinheiten bietet sich außerdem der Einsatz eines Switches an, welcher diese Systeme kollisionsfrei miteinander verbindet. In der Steuerungsebene des Brust-CT-Scanners finden nur zwei Steuerungseinheiten Verwendung, weshalb die Funktion des Switches in eines der beiden Steuerungseinheiten integriert wird.

Idealerweise ist das eingesetzte, auf Ethernet basierende Protokoll echtzeitfähig. Dies ist durch den Einsatz kommerzieller, echtzeitfähiger Protokollstacks und der dazu gehörigen Hardware erreichbar. Sie eignen sich für den Einsatz im Brust-CT-Scanner nur bedingt, da sie für industrielle Automation erdacht wurden und mit hohen Kosten für Lizenzen und Hardware verbunden sind. Um dennoch Echtzeitfähigkeit oder zumindest Determinismus erreichen zu können, kommt ein eigenes Protokoll mit dem Namen „CT\_Control\_Ethernet“ zum Einsatz, in Abbildung 25 grün gezeichnet, das in Kapitel 3.3.3 explizit beschrieben wird.

### **3.3.2 CAN-Kommunikationsprotokoll „CT\_Control\_CAN“**

#### **3.3.2.1 Stand der Technik**

Zur Kommunikation sind in der Industrie unter anderem CAN-basierende Bussysteme etabliert, die in der Norm DIN EN 61784-1 [51] beschrieben sind. In der Medizintechnik ist CAN ebenfalls weit verbreitet, wobei Hersteller von CT-Scannern ihre eigenen CAN-basierenden Protokolle bevorzugen oder auf Komponenten von Fremdherstellern zurückgreifen, die einem Anwendungs- oder Geräteprofil nach CANopen entsprechen. CANopen unterstützt durch entsprechende Profile unter anderem bereits die Ansteuerung von Patientenliegen, Röntgengeneratoren und -kollimatoren. [23] Neben CAN finden auch noch andere Schnittstellen, wie z.B. RS232 oder RS485, Verwendung, deren Kommunikationsprotokolle aber keiner allgemeinen Norm entsprechen. Welche Schnittstelle bzw. welches Protokoll sich für eine Komponente eignet, ist abhängig von der Anzahl und Größe der mit ihr auszutauschenden Daten sowie der Häufigkeit mit der Daten ausgetauscht werden sollen. [40; 90; 179; 246]

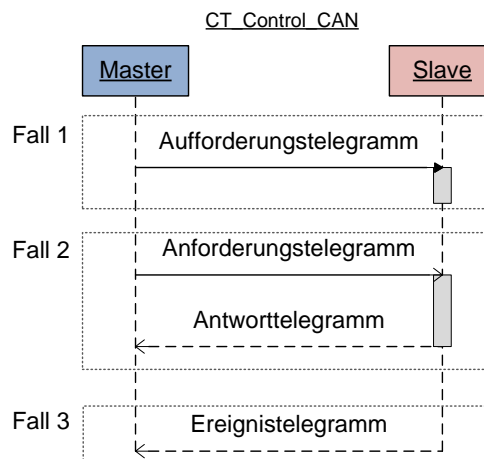
#### **3.3.2.2 Probleme beim Stand der Technik**

Die Verwendung von CAN zur Kommunikation in der Medizintechnik erfolgt, wie im Stand der Technik dargelegt, unter anderem mithilfe von CAN-Protokollen. Herstellerspezifische CAN-Protokolle implementieren hierbei meist ein Producer/Consumer-Modell, welches sich für eine hierarchische Strukturierung der Kommunikation nicht eignet. Eine Anpassung der Kommunikation von Röntgenröhre und Detektor an die bekannten CANopen-Profile für die Medizintechnik ist keine praktikable Lösung, da die genannten Profile nicht über die nötigen Funktionen zur Ansteuerung der Bildgebungskomponenten des Brust-CT-Scanners verfügen. Um CANopen für diese Komponenten nutzen zu können, müssten CANopen-Profile entsprechend erweitert oder neu geschaffen werden. Dies ist mit hohem Entwicklungsaufwand und -kosten verbunden. Desweiteren sind standardisierte Protokolle wie CANopen dafür geschaffen, möglichst viele Anwendungsbereiche und Hersteller zu erreichen, was zu einem hohen Protokoll-overhead führt. Das Datenvolumen steigt und die Leistungsfähigkeit des Protokolls sinkt, verglichen mit einem speziellen Protokoll, welches nur für die geforderten Aufgaben konzipiert wurde.

Diese Problematik wird durch ein Protokoll gelöst, das den Funktionsumfang der genannten neuartigen CT-Komponenten abdeckt, echtzeitfähig ist und minimalen Protokolloverhead aufweist.

### 3.3.2.3 Kommunikationsablauf von „CT\_Control\_CAN“

Der Kommunikationsablauf des „CT\_Control\_CAN“-Protokolls ist ereignisgesteuert und basiert auf dem Master/Slave-Buszugriffsverfahren und einem darauf aufsetzenden Client/Server-Konzept. Der Ablauf lässt sich in die in Abbildung 26 gezeigten drei Fälle einteilen. Zum Setzen von Werten oder zum Auslösen von Aktionen sendet der Master, z.B. eine Steuerungseinheit, ein Aufforderungstelegramm und erwartet keine Antwort von dem entsprechenden Slave, z.B. einer Kontrolleinheit (Fall 1). Möchte der Master hingegen Daten vom Slave abfragen, z.B. das Auslesen von Werten, sendet er ein Anforderungstelegramm und wartet auf die Antwort des Slave (Fall 2). Der Slave verarbeitet die Anforderung des Masters und sendet ein Antwortpaket mit den entsprechenden Daten. Das Antwortpaket muss innerhalb einer festgelegten Zeit erfolgen, da ansonsten ein Timeout auf Seiten des Masters ausgelöst wird. Ein Master sendet während dieser Zeit keine weiteren Telegramme, was einer synchronen Kommunikation entspricht. Für asynchrone oder unerwartete Ereignisse, z.B. Fertigstellung eines Prozesses oder Fehlers, kann ein Slave jederzeit ein Ereignistelegramm senden, um dieses Ereignis dem Master mitzuteilen (Fall 3). [1; 139; 200]

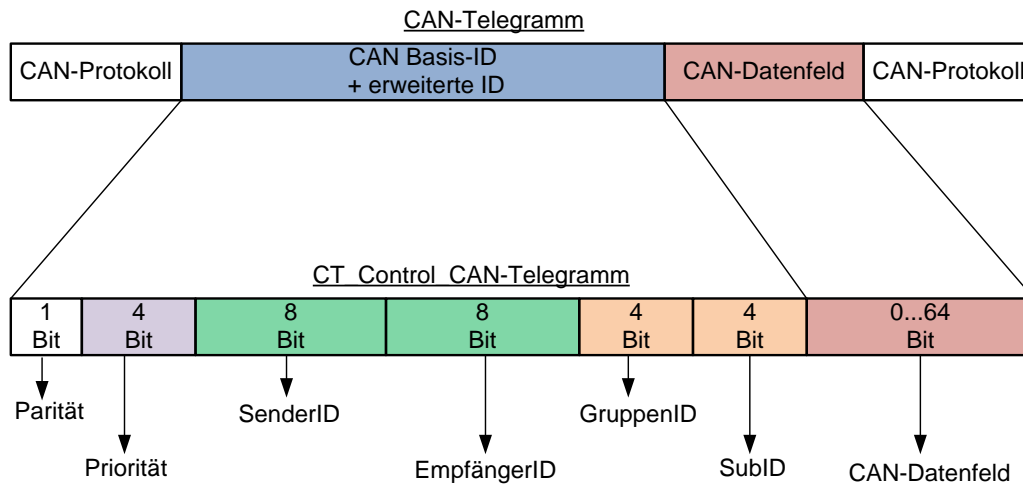


**Abbildung 26: Kommunikationsablauf des „CT\_Control\_CAN“-Protokolls:** Es existieren drei mögliche Fälle der Kommunikation, die mit verschiedenartigen CAN-Telegrammen abgewickelt werden.

### 3.3.2.4 Telegrammaufbau von „CT\_Control\_CAN“

Die zweite Schicht von CAN definiert, wie in Kapitel 2.5.2 beschrieben, kein Protokoll zur Datenübertragung, sondern ein ID-Feld mit dem ein Producer/Consumer-Modell realisiert werden kann. Zur Realisierung eines Master/Slave-basierten Client/Server-Konzepts, teilt „CT\_Control\_CAN“ das ID-Feld entsprechend Abbildung 27 in weitere Felder ein. Aus mehreren Veröffentlichungen ist bekannt, dass sich das 29 Bit große

ID-Feld aus dem CAN-Standard 2.0B für weitere Protokollfunktionen nutzen lässt. Dies hat den Vorteil, dass das bei CAN ohnehin nur 64 Bit große Datenfeld weiterhin in voller Größe für Daten zu Verfügung steht. [125; 155; 191; 232; 249].



**Abbildung 27: Aufteilung eines CAN-Telegramms durch „CT\_Control\_CAN“.**

Ein Telegramm von „CT\_Control\_CAN“ beginnt mit dem Paritätsbit, welches dazu dient, das 29 Bit lange ID-Feld von CAN in weitere Felder aufzuteilen, die jeweils eine Länge von vier bzw. acht Bit aufweisen. Daran anschließend folgt ein vier Bit langes Prioritätsfeld, das den Empfängern zur Priorisierung eines Telegramms dient, falls mehrere Teilnehmer gleichzeitig über den Bus senden. Es lassen sich damit 16 verschiedene Prioritäten festlegen. Da sich das Prioritätsfeld am Anfang des ID-Feldes befindet, ist sichergestellt, dass Telegramme mit höheren Prioritäten die Arbitrierung des Busses gewinnen. Die Felder „SenderID“ und „EmpfängerID“ sind für eine Adressierung jedes Telegramms zuständig. Hierdurch ist festgelegt, welcher Busteilnehmer ein Telegramm gesendet hat und für wen dieses bestimmt ist.

Um verschiedene Telegramme logisch zusammenzufassen und ihnen eine Priorität geben zu können, existiert die „GruppenID“. Sie repräsentiert eine Gruppe von Telegrammen, die ähnliche Funktionen auf der jeweiligen Komponente auslösen. Tabelle 3 zeigt die im Brust-CT-Scanner verwendeten Telegrammtypen. Diese sind bei Bedarf noch auf bis zu 16 Typen erweiterbar. Jeder „GruppenID“ ist entsprechend ihrer Relevanz einer Priorität von 1 bis 16 zugewiesen. Je wichtiger eine Telegrammgruppe ist, desto höher ist ihre Priorität und desto niedriger ihre Prioritätsnummer.

**Tabelle 3: Die verschiedenen Telegrammtypen von „CT\_Control\_CAN“.**

Typ	Priorität	Beschreibung
Kommando	0x9	Starten und Stoppen von Aktionen
Parameter Schreiben	0x8	Setzen von Parametern und Werten
Parameter Lesen	0x7	Auslesen von Parametern und Werten
Speicherzugriff	0x6	Übertragung großer Datenmengen für Updates etc.
Antwort	0x5	Antwort auf ein Telegramm, das Rückgabe erwartet
Nachricht	0x4	Senden einer informativen Nachricht
Warnung	0x3	Senden einer Warnung
Fehler	0x2	Senden eines Fehlers
Fataler Fehler	0x1	Senden eines fatalen Fehlers

Die „SubID“ kennzeichnet genau ein Telegramm der „GruppenID“. Dies entspricht einer ausführbaren Funktion auf Seiten einer bestimmten, durch die „EmpfängerID“ festgelegten Komponente. Ein Telegramm zum Setzen eines Parameters auf Seiten einer Kontrolleinheit entspricht beispielsweise folgender Funktion:

```
writeParameter(RegisterOffset: uint_16, Value: uint_32)
```

Die für diese Funktion benötigte Datenmenge von 48 Bit ist kleiner als die für CAN maximal zulässige Datenmenge von 64 Bit und lässt sich damit in ein Telegramm der Gruppe „Parameter Schreiben“ mit z.B. der „GruppenID“ 0x4 und der „SubID“ 0x01 einordnen. Die ausführbaren Funktionen auf den Kontrolleinheiten der Komponenten sind so gewählt, dass sich jede Funktion einer Kontrolleinheit mit einem einzigen CAN-Telegramm abbilden lässt. Zur Übertragung größerer Datensätze, wie sie z.B. bei einem Firmware-Update vorkommen, existiert eine spezielle Telegrammgruppe, die die hierfür nötigen Telegramme zu Verfügung stellt.

### 3.3.3 Ethernet-Kommunikationsprotokoll „CT\_Control\_Ethernet“

#### 3.3.3.1 Stand der Technik

Zur Kommunikation zwischen Leit- und Steuerungsebene in der Industrie sowie in Bereichen der Medizintechnik finden je nach Anwendungsbereich Protokolle auf Basis von echtzeitfähigem oder nichtechtzeitfähigem Ethernet Verwendung. Da Letzteres nicht garantieren kann, Pakete in einer definierten Zeit zuzustellen, ist der Einsatz von nichtechtzeitfähigem Ethernet in einem Medizinprodukt mit entsprechenden Schwierigkeiten verbunden bzw. nur möglich, wenn nachgewiesen werden kann, dass es durch die Nichterfüllung von Zeitanforderungen unter keinen Umständen zu einer Gefährdung von Menschen kommen kann. In einem CT-Scanner ist dieser Nachweis nur möglich, indem vollständig darauf verzichtet wird, sicherheitsrelevante Daten über nichtechtzeitfähiges Ethernet auszutauschen. Der Einsatz von echtzeitfähigem bzw. deterministischem Ethernet hingegen macht diesen Nachweis überflüssig. Hierzu existieren mehrere kommerzielle, in der DIN EN 61784-2 [52] festgehaltene, echtzeitfähige Ethernet-Technologien, wie die in Abschnitt 2.5.1.4 benannten. [38; 46; 70; 182; 199; 206; 223]

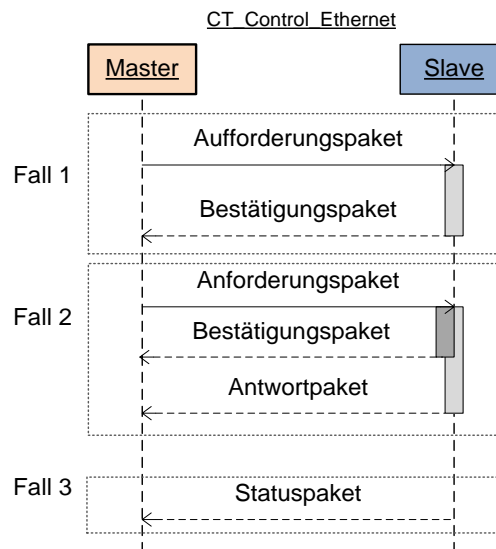
### 3.3.3.2 Probleme beim Stand der Technik

Durch das nichtdeterministische Buszugriffsverfahren, und der daraus resultierenden Absenz von Echtzeitverhalten, eignen sich nichtechtzeitfähige Protokolle wie z.B. TCP/IP/UDP nur mit sehr starken Einschränkungen für den Einsatz in Echtzeitsystemen. [69] Der Nachweis, dass hierdurch kein Gefährdungspotential für Menschen vorliegt, ist nicht ohne Weiteres zu erbringen, da ein CT-Scanner durch elektrischen Strom, Bewegung und Röntgenstrahlung generell eine Gefahr für Menschen darstellt. Außerdem beinhaltet ein nicht zertifizierter Protokollstack, z.B. für TCP, fremden Programmcode, dessen Funktion nicht bis ins letzte Detail bekannt ist, wodurch zusätzliche Unsicherheitsfaktoren auftreten. Andere auf Ethernet basierende Technologien, die den Anspruch der Echtzeitfähigkeit erheben, sind mit sehr hohen Kosten verbunden, benötigen teilweise spezielle Hardware und besitzen, ebenso wie bekannte CAN-Protokollstandards, Overhead, der die Leistungsfähigkeit mindern kann. Des Weiteren befinden sich IP-Cores für echtzeitfähiges Ethernet für die im weiteren Verlauf dieser Arbeit verwendeten SOPCs noch in der Entwicklungsphase und werden erst in der zweiten Jahreshälfte 2013 verfügbar sein. [3; 8; 203] Hieraus ergibt sich der Wunsch, für den Brust-CT-Scanner ein performantes Ethernet-Protokoll zu entwickeln, welches Standardhardware verwendet, deterministisch ist und ebenfalls minimalen Protokoll-overhead aufweist.

### 3.3.3.3 Kommunikationsablauf von „CT\_Control\_Ethernet“

„CT\_Control\_Ethernet“ dient der Kommunikation zwischen einem Master, z.B. dem Leitrechner, und mehreren Slaves, z.B. den Steuerungseinheiten. Die Architektur dieses Protokolls sieht eine Verbindung von einem Master mit jeweils einem Slave, entweder direkt oder über einen voll duplexfähigen Switch nach IEEE 802.1Q [110], vor. Hierdurch ist sichergestellt, dass beide Punkt-zu-Punkt verbunden sind und bereits eine Priorisierung über das in IEEE 802.1Q definierte Prioritätsfeld „Priority Code Point“ möglich ist. [67; 181; 245]

Ein Master, z.B. der Leitrechner, kommuniziert mit einem Slave, z.B. einer Steuerungseinheit, nach dem in Abbildung 28 gezeigten Ablauf. Das Senden von Paketen wird in den ersten beiden Fällen durch den Master initiiert. Möchte der Master einem Slave Daten senden, und erwartet keine Daten zurück, überträgt er ein entsprechendes Anforderungspaket, welches vom Slave mit einem Bestätigungspaket quittiert wird (Fall 1). Der Slave beginnt erst nach dem Senden des Bestätigungspakets mit der Verarbeitung der Daten. Möchte der Master hingegen Daten von einem Slave abfragen, überträgt er diesem ein Anforderungspaket, dessen Empfang vom Slave zeitnah mit einem Bestätigungspaket quittiert wird (Fall 2). Danach beginnt der Slave mit der Zusammenstellung der gewünschten Daten, welche es dem Master nach einer gewissen Verarbeitungszeit mit einem Antwortpaket überträgt. Hierbei ist durch den Slave eine Zeitbedingung einzuhalten, da ansonsten ein Timeout ausgelöst wird.

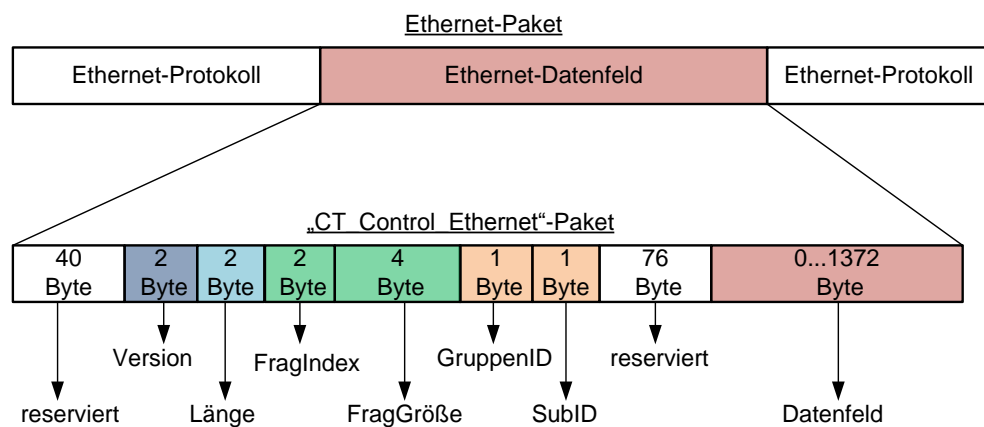


**Abbildung 28: Kommunikationsablauf des „CT\_Control\_Ethernet“-Protokolls:** Es existieren drei mögliche Fälle der Kommunikation, die mit verschiedenartigen Ethernet-Paketen abgewickelt werden.

Alle weiteren Fälle der Datenübertragung sind durch ein Statuspaket abgedeckt, welches jeder Slave in festen Zeitabständen sendet (Fall 3). Das Nichteintreffen eines solchen Pakets löst ebenfalls ein Timeout aus. Somit ist sichergestellt, dass unvorhergesehene Ereignisse, insbesondere Fehler, nach einer festen Zeit dem Master gemeldet werden. [1; 139; 200]

#### 3.3.3.4 Paketaufbau von „CT\_Control\_Ethernet“

Das „CT\_Control\_Ethernet“-Protokoll ist entsprechend Abbildung 29 in ein Datenpaket eines Standard-Ethernet-Pakets integriert. Weniger komplexe Aufteilungen finden sich in den wissenschaftlichen Veröffentlichungen von Martínez et al. [163] und Pedreiras et al. [180], welche jeweils ein echtzeitfähiges Protokoll für Ethernet vorstellen.



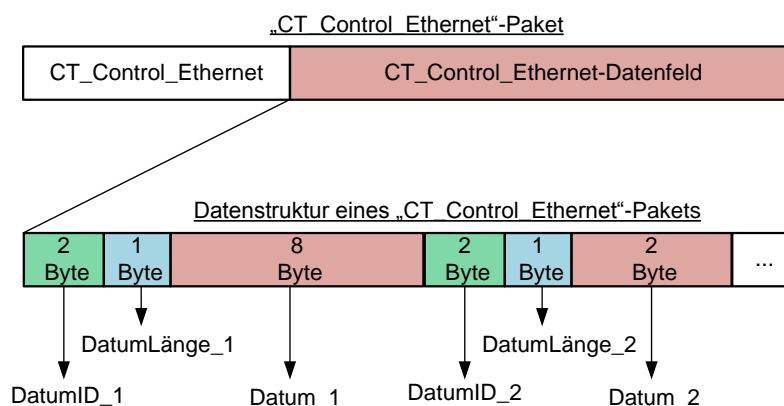
**Abbildung 29: Aufteilung eines Ethernet-Pakets durch „CT\_Control\_Ethernet“.**

Die ersten 40 Byte eines Pakets von „CT\_Control\_Ethernet“ bleiben aus Gründen der Kompatibilität für einen eventuellen, späteren Einsatz eines kommerziellen Protokolls reserviert. Eine daran anschließende zwei Byte große Versionsnummer lässt auf die Version des Protokolls schließen. Ein ebenfalls zwei Byte großes Längensfeld gibt die Länge der im Datenfeld transportierten Daten für das Paket an. Das Datenfeld kann eine Größe von bis zu 1372 Byte erreichen.

Um Daten fragmentiert übertragen zu können, deren Größe das Datenfeld überschreitet, existiert ein „FragIndex“. Der Wert dieses Index entspricht der Anzahl der noch zu sendenden Pakete und wird mit jedem erfolgreich empfangenen Paket dekrementiert. Die dem „FragIndex“ folgende „FragGröße“ gibt die Gesamtgröße der zu sendenden fragmentierten Daten in Byte an.

Die drei Byte große „GruppenID“ kennzeichnet, um welchen Typ von Paket es sich handelt. Jede „GruppenID“ fasst eine bestimmte Anzahl von logisch zusammengehörigen Paketen zusammen. Jedes Paket einer Gruppe verfügt zudem über eine in der „GruppenID“ eindeutigen „SubID“. Die durch jeweils eine „GruppenID“ und „SubID“ eindeutig gekennzeichneten Pakete lösen fest vorgegebene Funktionen auf Seiten des Empfängers aus, wie z.B. das Setzen von Parametern oder das Starten einer Aktion. Die letzten 76 Byte sind für zukünftige Erweiterungen reserviert und stellen sicher, dass die Größe des Paketkopfes immer 128 Byte beträgt.

Aufgrund der Größe des Datenfeldes von bis zu 1372 Byte lassen sich im Gegensatz zu CAN, welches nur über ein 8 Byte großes Datenfeld verfügt, sehr viele Daten mit nur einem Paket übertragen. Zur besseren Unterscheidbarkeit der Daten im Datenfeld sind diese entsprechend Abbildung 30 sortiert. Sie zeigt beispielhaft den Anfang eines Paketdatenfelds mit jeweils einem zwei und einem acht Byte großen Datum. Jedem Datum ist eine ID zugewiesen, die „DatumID“. Zusätzlich wird seine Größe, „DatumLänge“, übertragen. Damit lassen sich Daten beim Empfänger leichter aus einem Paket auslesen, z.B. mithilfe eines Parsers, einem speziellen Programm zur Verarbeitung von Datenströmen, da die Größe jedes Datums jeweils vor dessen Beginn angegeben ist.



**Abbildung 30: Struktur des Datenfeldes von „CT\_Control\_Ethernet“.**

## **3.4 Rohdatenübertragungsprotokoll „CT\_Data\_Ethernet“**

### **3.4.1 Stand der Technik**

In der CT fallen aufgrund ständig steigender Ortsauflösung durch bessere Detektoren bei der Aufnahme von Projektionsdaten sehr große Datenmengen an, die es zu übertragen gilt. Dies trifft besonders auf den dedizierten Brust-CT-Scanner zu, der pro Scan ca. 100 GByte an Projektionsdaten erzeugt. Aus dem Stand der Technik geht hervor, dass die Übertragung von Projektionsdaten in klinischen sowie präklinischen CTs vorwiegend kontaktlos über Drehübertrager erfolgt, wobei kapazitive, hochfrequenztechnische oder optische Verfahren zum Einsatz kommen. Die Projektionsdaten liegen hierbei in Segmente unterteilt vor und werden von speziellen Sende- und Empfangsmodulen sequentiell als Bitstrom auf der physikalischen Ebene des OSI-Modells übertragen. [12; 30; 103; 128]

Zur Übertragung wird der Datenstrom auf Seiten des Sendemoduls serialisiert und auf Seiten des Empfängermoduls wieder deserialisiert, wobei meist eine Einheit aus Serializer/Deserializer (SerDes) und eine 8Bit/10Bit-Kodierung bei Übertragungsraten im Bereich mehrerer GBit/s Verwendung findet. Zur weiteren Verarbeitung kommen Protokolle aus dem Bereich der Hochgeschwindigkeitsdatenübertragung, wie z.B. „Fibre Channel“, „SATA“, „SAS“ und „Infiniband“ zum Einsatz. Durch Verwendung mehrerer paralleler Kanäle können so hohe Übertragungsraten erreicht werden. [12; 30; 34; 90; 103; 108; 124; 194; 235]

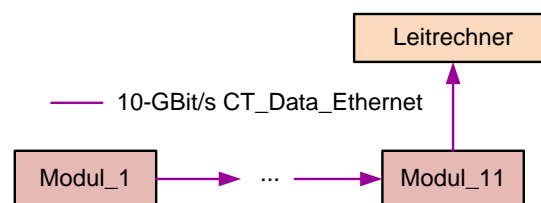
### **3.4.2 Probleme beim Stand der Technik**

Wie bei klinischen CT-Scannern üblich, soll auch für den dedizierten Brust-CT-Scanner die Übertragung der Projektionsdaten kontaktlos, mittels Lichtwellenleiter, erfolgen. Aufgrund der geringeren Gantryabmessungen im Vergleich zu einem Ganzkörper-CT-Scanner verfügt der Drehübertrager des Brust-CT-Scanners jedoch nicht über mehrere parallele Kanäle, sondern nur über einen Kanal zur Projektionsdatenübertragung. Die Entwicklung von Standard-SerDes-Komponenten für die Übertragung über diesen Kanal erfordert spezielle Hardware auf Seiten des Empfängercomputers und ist mit einem hohen Aufwand verbunden. [124] Außerdem wird eine zusätzliche Funktionalität zum Erkennen von Übertragungsfehlern benötigt, weshalb hiervon aus Kostengründen abgesehen und stattdessen der Einsatz von Standard-Komponenten bzw. COTS-Komponenten bevorzugt wird. Hierzu bieten sich 10-GBit/s-Ethernet-Komponenten an, deren Hersteller Datentransferraten von bis zu 10 GBit/s angeben und deren Kosten in den vergangenen Jahren stark gesunken sind. Die Verwendung von Standard-Ethernet-Komponenten wurde bereits in [61; 108; 114] gezeigt, allerdings mit einer Übertragungsrate von maximal 1 GBit/s sowie in Zusammenhang mit einem TCP/UDP Protokoll und den sich daraus ergebenden Nachteilen des großen Overheads der höheren

Schichten. Für den Brust-CT-Scanner wird zur Datenübertragung von Projektionsdaten eine 10-GBit/s-Ethernet Verbindung verwendet, die auf Standard-Komponenten basiert. Als Übertragungsprotokoll kommt das im Rahmen dieser Arbeit hierfür entwickelte „CT\_Data\_Ethernet“-Protokoll zum Einsatz. [34; 215]

### 3.4.3 Sender- und Empfängerarchitektur für „CT\_Data\_Ethernet“

Auf Basis der 10-GBit/s-Ethernet-Technologie wurde ein Datenübertragungsprotokoll entworfen, welches die Übertragung der Projektionsdaten des Brust-CT-Scanners vom Detektor zu einem Standard-PC mit Rekonstruktionssoftware (Rekonstruktionsrechner) sicherstellt. Die hierzu nötige Architektur wird im Folgenden beschrieben. Als Rekonstruktionsrechner eignet sich für den Brust-CT-Scanner besonders der bereits im Rahmen des Steuerungskonzepts vorhandene Leitrechner. Dieser muss entsprechend leistungsfähig sein und zusätzlich über eine 10-GBit/s-Ethernet-Netzwerkkarte verfügen. Aufgrund des nur einkanaligen, aber duplexfähigen Lichtwellenleiters im Drehübertrager eignet sich zur Übertragung der Projektionsdaten eine Reihenschaltung aller Detektormodule via Ethernet. Auf Seiten des Detektors benötigt jedes Detektormodul hierfür zwei physikalische Ethernet-Schnittstellen, jeweils eine um Projektionsdaten von einem Nachbarn zu empfangen und eine um Projektionsdaten einem anderen Nachbarn zu senden. Ein Detektor verfügt somit entsprechend Abbildung 31 über ein erstes Detektormodul mit nur einem Nachbarn und ein abschließendes Detektormodul, das mit dem Rekonstruktionsrechner verbunden ist.



**Abbildung 31: Schematische Darstellung der Projektionsdatenübertragung:** Alle Detektormodule sind in Reihe geschaltet und senden ihre Projektionsdaten an ihren jeweiligen Nachbarn, das letzte Modul sendet sie dem Rekonstruktionsrechner.

Der Auslesevorgang wird durch den Rekonstruktionsrechner (Leitrechner) initiiert und von der rotierenden Steuerungseinheit gesteuert. Jedes Detektormodul sendet bei einem Datentransfer seine gepufferten Projektionsdaten einem benachbarten Detektormodul, das abschließende Detektormodul entsprechend zum Leitrechner. Durch diese Reihenschaltung lassen sich mit nur einem Lichtwellenleiter im Drehübertrager und nur einer 10-GBit/s-Ethernet-Netzwerkkarte auf Seiten des Rekonstruktionsrechners die Projektionsdaten schnell und effizient übertragen.

### 3.4.4 Kommunikationsablauf „CT\_Data\_Ethernet“

Für einen erfolgreichen Kommunikationsablauf, siehe hierzu Abbildung 32, müssen Sender und Empfänger Punkt-zu-Punkt verbunden sein. Der Sender beginnt mit der Übertragung von Datenpaketen, welche die Projektionsdaten enthalten. Der Empfänger bestätigt den korrekten Erhalt von  $n$  Datenpaketen mit einem Bestätigungspaket. Nach erfolgreicher Bestätigung von  $n$  Datenpaketen beginnt der Sender wieder mit der Übertragung von  $n$  Datenpaketen. Dieser Vorgang wiederholt sich  $m$ -mal, bis alle Projektionsdaten übertragen wurden. Treten bei der Übertragung fehlerhafte Datenpakete auf oder kommt es zu einem Paketverlust, kann der Empfänger dies über ein Fehlerpaket mitteilen, worauf der Sender die Übertragung ab dem fehlerhaften oder verlorenen Paket wiederholt. Zusätzlich verfügen beide Teilnehmer über ein Timeout, der auslöst, falls die Verbindung komplett unterbrochen wird. [144]

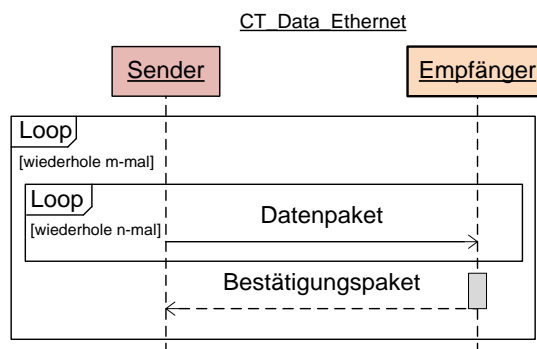


Abbildung 32: Kommunikationsablauf des „CT\_Data\_Ethernet“-Protokolls.

### 3.4.5 Paketaufbau „CT\_Data\_Ethernet“

„CT\_Data\_Ethernet“ basiert auf der zweiten Ebene des Ethernet-Standards und verwendet unter anderem Pakete im „Jumbo Frame“-Format, um einen möglichst hohen Datendurchsatz gewährleisten zu können. Datenpakete von „CT\_Data\_Ethernet“ sind wie in Abbildung 33 gezeigt, in das Datenfeld eines Ethernet-Paketes im „Jumbo Frame“-Format integriert. Ein Datenpaket enthält einen zwei Byte großen Index, „PaketIndex“, der mit jedem Datenpaket erhöht wird und der dem Empfänger zur Bestätigung erfolgreich empfangener Datenpakete dient. Gefolgt wird dieser Index von einem 18 Byte großen Feld „ProjektionsInfo“, welches projektionsspezifische Daten enthält. Abgeschlossen wird das Datenpaket durch ein bis zu 8144 Byte großes Datenfeld, das für die eigentlichen Projektionsdaten vorgesehen ist. Die Informationen im „ProjektionsInfo“-Feld sowie die Projektionsdaten sind in einer Struktur angeordnet, die nach dem Abspeichern in den Hauptspeicher auf Seiten des Empfängers in einem für die Rekonstruktionssoftware besonders günstigem Datenformat vorliegen und sofort verarbeitet werden können.

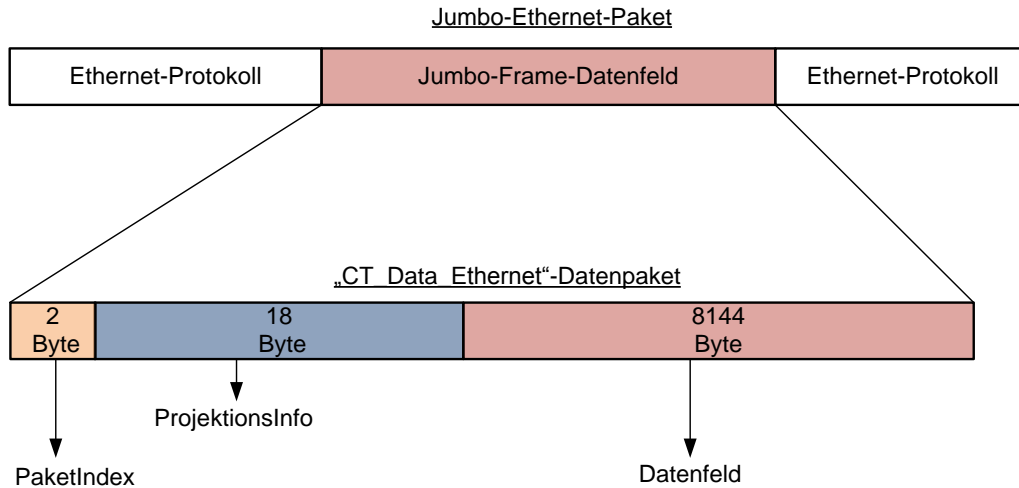


Abbildung 33: Paketaufbau eines „CT\_Data\_Ethernet“-Datenpakets.

Der Empfänger quittiert den Erhalt einer bestimmten Anzahl von  $n$  Datenpaketen mit einem Bestätigungspaket, gezeigt in Abbildung 34. Es enthält ein Feld „LetzterPaket-Index“, das den Index des letzten erfolgreich empfangenen Datenpakets sowie ein Bestätigungsfeld „Ack“, zur Bestätigung der  $n$  Datenpakete, enthält. Ist das Bestätigungsfeld nicht gesetzt, muss der Sender die Übertragung ab dem letzten erfolgreich empfangenen Datenpaket, gekennzeichnet durch das Feld „LetzterPaketIndex“, wiederholen. Ansonsten fährt er mit der Übertragung fort.

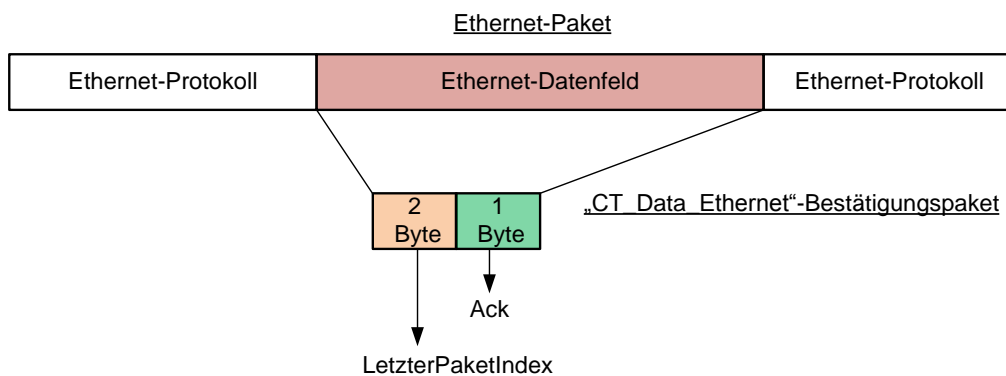


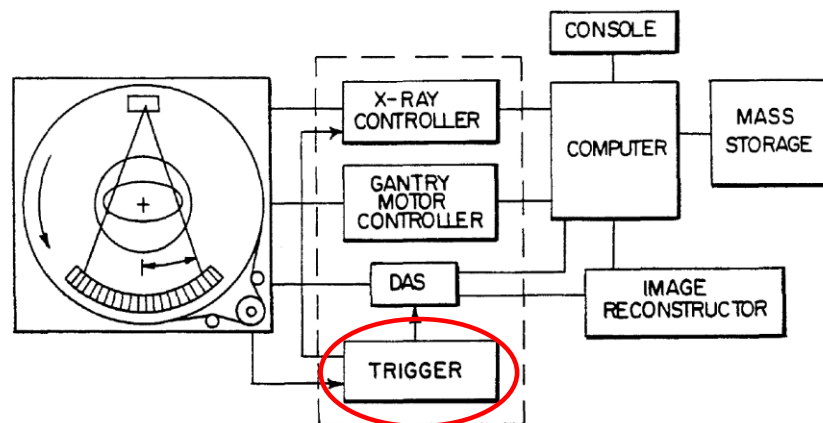
Abbildung 34: Paketaufbau eines „CT\_Data\_Ethernet“-Bestätigungspaketes.

## 3.5 Synchronisation der Bildgebungskomponenten

### 3.5.1 Stand der Technik

Die Strahlungsauslösung der Röntgenröhre, das Auslesen des Detektors und das Erfassen der Positionen zu jeder Projektion müssen durch eine entsprechende Synchronisationseinrichtung aufeinander abgestimmt werden. Dies ist notwendig, da zur Rekonstruktion der aufgenommenen Schichtbilder in der CT zu jeder Projektion die entsprechenden Positionswerte vorliegen müssen. Diese sind jeweils die Winkelposition der Gantry ( $\alpha$ -Wert) und ihre Translationsposition ( $z$ -Wert). Die Güte dieser Abstimmung hat somit große Bedeutung, da sie einen wesentlichen Einfluss auf die Bildqualität hat.

In klinischen CT-Scannern sowie Detektorsystemen aus dem Forschungsbereich wird hierzu zusätzlich zum zentralen Steuerungscomputer eine Synchronisationseinrichtung in Form eines „Triggermoduls“ verwendet, welches nur die Aufgabe der Erzeugung von Synchronisationssignalen übernimmt. Hierbei kann es sich je nach Anforderung um ein eingebettetes System oder eine Schaltung aus Logikblöcken auf Basis eines FPGAs oder ASICs handeln. [39; 41; 43; 78; 89; 161; 215; 217; 218; 221; 230; 241]



**Abbildung 35: Synchronisation eines CT-Scanners mit einem „Triggermodul“** aus [26]: Ein solches Modul zur Synchronisation der Komponenten findet häufig Verwendung in CT-Scannern.

Abbildung 35 zeigt eine typische Systemarchitektur eines CT-Scanners mit „Triggermodul“ aus einer Patentschrift von Casey et al. [26]. In der Patentliteratur finden sich ebenfalls Ausführungen in denen der Detektor ein Synchronisationssignal bereitstellt, das „Triggermodul“ ist somit im Detektor integriert. [131; 173] Als Technologie zur Signalübertragung der Synchronisationssignale wird unter anderem der Übertragungsstandard RS485 verwendet. [40; 215] Durch seinen einfachen Aufbau, der störungssicheren differentiellen Übertragung und den hohen Übertragungsgeschwindigkeiten ist er hierfür besonders gut geeignet.

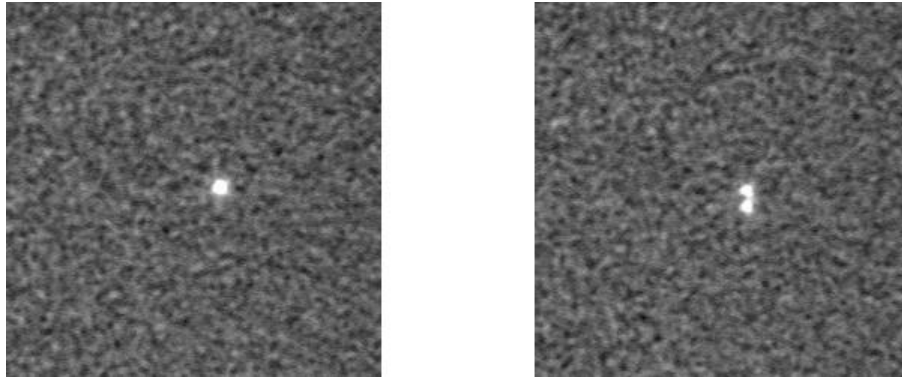
### 3.5.2 Probleme beim Stand der Technik

Die Integration eines „Triggermoduls“ in den Detektor oder einer anderen Komponente auf der Kontrollebene des Brust-CT-Scanners stellt keine praktikable Lösung dar, da der hier verwendete Detektor aus mehreren exakt gleichen Modulen besteht und somit jedes dieser kompakten Detektormodule damit bestückt sein müsste, was deren Kosten und Größe erheblich erhöhen würde. Die Verwendung eines einzelnen „Triggermoduls“ hingegen erhöht zum einen die Anzahl der Gesamtkomponenten, zum anderen wird eine Schnittstelle zur Kommunikation mit einer Steuerungseinheit benötigt, um beispielsweise deren Zeitverhalten festzulegen. Außerdem verfügen die Bildgebungskomponenten des Brust-CT-Scanners über die Technologien TDS und FSC. Eine Synchronisation von Bildgebungskomponenten, die diese Technologien enthalten, ist im Stand der Technik bisher nicht berücksichtigt. Aufgrund der Verwendung dieser Technologien benötigen die Bildgebungskomponenten eine besonders genaue Synchronisation, deren Güte und Funktionalität die bekannten Lösungen aus dem Stand der Technik übertrifft.

Diese Problematik wird durch Integration der Synchronisationsfunktion in die Steuerungsebene – und somit in die Steuerungseinheiten – gelöst. Um Signallaufzeiten auszugleichen, die durch unterschiedlich schnelle Signalerfassung bzw. -verarbeitung in den Komponenten der Steuerebene entstehen, wird für jede Komponente ein individuell konfigurierbares Synchronisationssignal bereitgestellt. Dieses dient der exakten Synchronisation der Komponenten und stellt das Erreichen höchster Bildqualität sicher.

### 3.5.3 Zeitbasierter Projektionsbeginn mit Positionserfassung

Eine Synchronisationseinheit für die Synchronisation der Bildgebungskomponenten legt den Beginn einer Projektion fest. Diese Festlegung eines Projektionsbeginns kann zeit- oder auch positionsbasiert erfolgen. Bei der zeitbasierten Methode wird der Projektionsbeginn nach festen Zeitabständen ausgelöst. Da die Zeitabstände konstant sind, gilt dies auch für die von der Röntgenröhre ausgesandte Röntgenintensität  $I_0$ . Aufgrund von mechanischen Toleranzen ist die Winkelgeschwindigkeit der Gantry allerdings nicht konstant. [41; 131] Wird bei der zeitbasierten Methode nicht gleichzeitig auch die Position erfasst, sondern nur anhand der Rotationsgeschwindigkeit geschätzt, macht sich dies in der Bildqualität negativ bemerkbar. Abbildung 36 zeigt hierzu die Rekonstruktion eines Drahtphantoms, das mit einem  $\mu$ CT-Scanner für Kleintierbildgebung aufgenommen wurde. Auf dem Bild links erfolgte die Rekonstruktion mithilfe der gemessenen Winkelpositionen, auf dem Bild rechts mithilfe einer Winkelliste, die sich aus gleichen Winkelinkrementen, berechnet aus der Scandauer sowie Anfangs- und Endposition, zusammensetzt. Die  $MTF_{10\%}$  auf dem Bild links beträgt 6,3 lp/mm, die auf dem Bild rechts lediglich 3,5 lp/mm. Auf dem Bild rechts sind deutliche Defizite in der Bildqualität erkennbar.

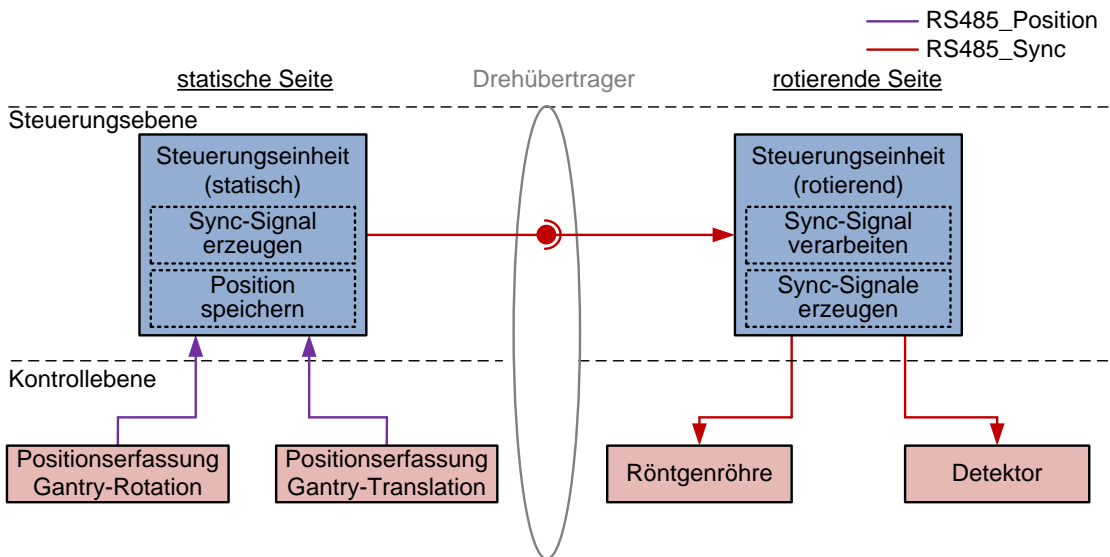


**Abbildung 36: Vergleich der Bildqualität bei zeitbasierter Synchronisation:** Das Bild links ( $MTF_{10\%} = 6,3 \text{ lp/mm}$ ) zeigt eine Rekonstruktion eines Drahtphantoms anhand einer gemessenen Winkelliste, das Bild rechts ( $MTF_{10\%} = 3,5 \text{ lp/mm}$ ) anhand einer berechneten Winkelliste. Das Bild rechts gibt den Draht verfälscht und unscharf wieder.

Ebenso lässt sich eine positionsbasierte Methode zur Synchronisation verwenden, bei der eine Projektion nach einem bestimmten Winkelinkrement der Gantry ausgelöst wird. [26] Hierbei kommt es durch die unterschiedlich hohe Winkelgeschwindigkeit der Gantry zu unterschiedlich hohen Röntgenintensitäten  $I_0$  bei den Projektionen. Durch technische Mittel lässt sich die Röntgenintensität messen und in der Rekonstruktion berücksichtigen. Geschieht dies nicht, hat das ebenfalls negative Auswirkungen auf die Bildqualität. Im Rahmen dieser Arbeit wird eine zeitbasierte Synchronisation mit hochgenauer Positionserfassung favorisiert, da diese kostengünstiger und mit weniger Aufwand realisierbar ist als eine exakte Bestimmung der Röntgenintensität mit einem zusätzlichen Intensitätsmesser im Detektor bzw. der Röntgenröhre.

### 3.5.4 Systemübersicht

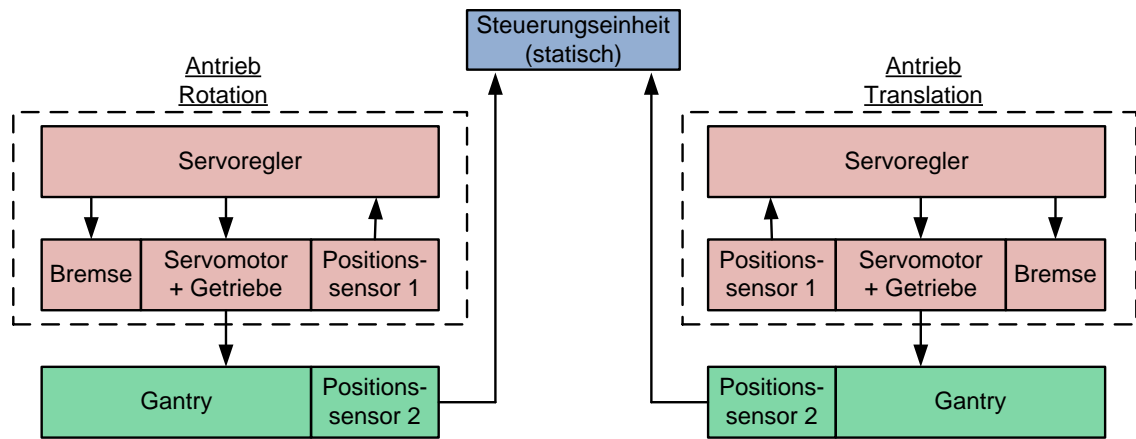
Die Synchronisation ist entsprechend Abbildung 37 aufgebaut. Die Position der Gantry (Rotation und Translation) wird von hochgenauen Sensoren erfasst, die ihre Positionssignale, „RS485\_Position“ (lila), kontinuierlich über RS485 an die statische Steuerungseinheit senden. Diese enthält einen hochgenauen Taktgenerator, der zu bestimmten Zeitpunkten den Beginn einer Projektion initiiert. Bei Projektionsbeginn legt die statische Steuerungseinheit die aktuelle Position der Gantry in einem Speicher ab, erzeugt ein Synchronisationssignal, „RS485\_Sync“ (rot), und sendet es über den Drehübertrager zur rotierenden Steuerungseinheit. Die rotierende Steuerungseinheit verarbeitet dieses Synchronisationssignal und erzeugt individuelle Synchronisationssignale für die Bildgebungskomponenten. Diese Synchronisationssignale werden jeweils, um interne Signallaufzeiten der Komponenten auszugleichen, mit einer individuellen Verzögerungszeit beaufschlagt und den entsprechenden Kontrolleinheiten von Röntgenröhre und Detektormodulen gesendet. Das initiale Synchronisationssignal der statischen Steuerungseinheit erfolgt entsprechend um die längste Verzögerungszeit versetzt, so dass dieses vor Erreichen des für die Projektion prädestinierten Zeitpunktes erfolgt.



**Abbildung 37: Übersicht der Synchronisation des Brust-CT-Scanners über RS485:** Die statische Steuerungseinheit erfasst die Position der beweglichen Gantry und initiiert zeitbasiert den Beginn einer Projektion durch das Senden eines Synchronisationssignals für die statische Steuerungseinheit.

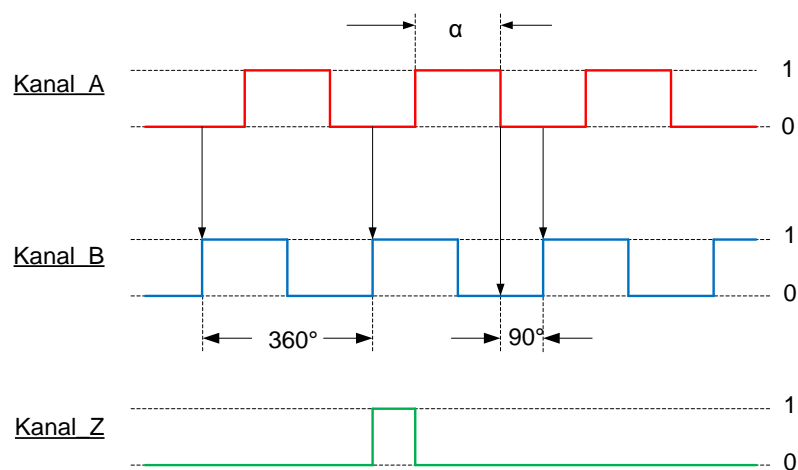
### 3.5.5 Positionserfassung

Die für den Brust-CT-Scanner verwendeten Servoantriebe zur Rotation und Translation der Gantry verfügen systembedingt über Sensoren zur Messung der aktuellen Position, vergleiche Abbildung 21 aus Kapitel 3.1.3. Die Werte dieser Sensoren lassen sich von den Kontrolleinheiten der Antriebe, also den Servoreglern, über CANOpen von den übergeordneten Steuerungseinheiten abfragen. Für eine Rekonstruktion werden zu jeder Projektion die entsprechenden Positionswerte der Gantryrotation und der Gantrytranslation benötigt. Bei einer Projektionsrate des Scanners von bis zu 1000 Projektionen pro Sekunde ergeben sich somit 2000 Positionswerte pro Sekunde, die es auszulesen gilt. Die beim Auslesen dieser Werte resultierende Last für den CAN-Bus wäre erheblich und überträfe seine Leistungsfähigkeit bei Weitem. Außerdem messen motorinterne Positionssensoren die Position des Motors, nicht aber die Position des damit angetriebenen Elements. Alle zur Kraftübertragung verwendeten Maschinenelemente wie, z.B. Getriebe und Riemen, weisen mechanische Abweichungen (Spiel) auf, wodurch sich eine Diskrepanz zwischen errechneter Position aus der Winkelposition der Motorwelle und tatsächlicher Position der angetriebenen Komponente ergibt. Aus diesem Grund verfügen alle von einem Servomotor angetriebenen Elemente, wie Gantry und Patientenliege, über einen zusätzlichen, hochgenauen Positionssensor, wie in Abbildung 38 zu sehen.



**Abbildung 38: Erfassung der Gantryposition:** Durch zusätzliche Positionssensoren können Translations- und Rotationsbewegung der Gantry exakt bestimmt werden.

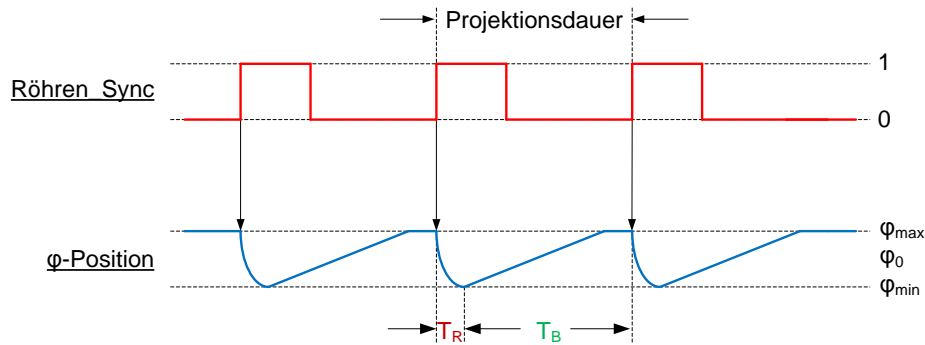
Als Schnittstellentechnologie hierfür bietet sich RS485 an, da diese Schnittstelle herstellerunabhängig ist, kein Protokoll benötigt und eine hohe Störsicherheit aufweist. Die Sensorelektronik kommuniziert den Positionswert mithilfe dreier RS485-Kanäle an die statische Steuerungseinheit. Auf jedem Kanal wird ein Rechtecksignal, entsprechend Abbildung 39, gesendet, das durch Interpolation von Abtastwerten eines Strichgitters bzw. einer Abtastplatte entsteht. Die Rechtecksignale der beiden Kanäle „Kanal\_A“ und „Kanal\_B“ sind zueinander um  $90^\circ$  phasenverschoben. Durch Zählen der Rechteckimpulse und Berücksichtigung ihrer Phasenlage wird durch die statische Steuerungseinheit die relative Position des angetriebenen Elements errechnet. Über den dritten Kanal, „Kanal\_Z“, wird an bestimmten Positionen, z.B. dem Nulldurchgang, ein Referenzimpuls gesendet. Mithilfe dieses Referenzimpulses und der relativen Position, die sich aus den anderen beiden Kanälen ergibt, lässt sich die absolute Position bestimmen. [58; 59; 112]



**Abbildung 39: Ausgangssignale eines Positionssensors:** Über drei RS485-Signale kann die Position eines angetriebenen Elements exakt bestimmt werden.

### 3.5.6 Synchronisation der Röntgenröhre

Die Röntgenröhre verfügt, wie bereits beschrieben, über einen positionsgesteuerten Brennfleck. Um die Position des Brennflecks von einer Steuerungseinheit vorzugeben, verfügt die Röntgenröhre über eine RS485-Schnittstelle. Die Bewegung wird gemäß dem Zeitverlaufdiagramm in Abbildung 40 über ein spezielles Synchronisationssignal „Röhren\_Sync“ vorgegeben.



**Abbildung 40: Zeitverlaufdiagramm der Röntgenröhrensynchronisation:** Die Position des Brennflecks wird durch ein Synchronisationssignal gesteuert.

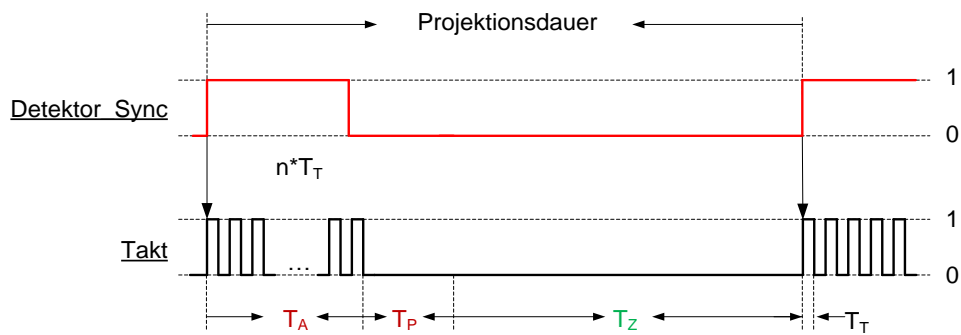
Der Ausgangspunkt des Brennflecks ist zu Beginn eines Scans, nach Aktivierung der Röntgenstrahlung, in der Position maximal positiver Auslenkung  $\varphi_{\max}$ . Wird von der rotierenden Steuerungseinheit ein „Röhren\_Sync“-Signal gesendet, lenkt die Kontrolleinheit der Röntgenröhre bei Erkennen der Flanke den Brennfleck auf die Position maximal negativer Auslenkung  $\varphi_{\min}$ . Die hierfür benötigte Zeit wird als Rücksprungzeit  $T_R$  bezeichnet und ist unabhängig von der Projektionsrate. Nach Erreichen von  $\varphi_{\min}$  bewegt sich der Brennfleck mit konstanter Geschwindigkeit nach  $\varphi_{\max}$ . Die hierfür benötigte Zeit, als Bewegungszeit  $T_B$  bezeichnet, ist durch die rotierende Steuerungseinheit variabel einstellbar und abhängig von der Projektionsrate des aktuellen Scans. Die effektive Aufnahme einer Projektion erfolgt nur während der Zeit  $T_B$ , also der Bewegung des Brennflecks. Während der Zeit  $T_R$  wird weiterhin Strahlung erzeugt und damit Dosis appliziert, diese ist aber medizinisch nicht nutzbar.  $T_R$  bezeichnet deshalb die Totzeit der Röntgenröhre während der Projektionsdauer.

### 3.5.7 Synchronisation der Detektormodule

Die verschiedenen Module des Detektors sollen über ein einziges zentrales Signal miteinander und mit der Röntgenröhre synchronisiert werden können. Hierzu erfolgt eine Unterteilung einer Projektion aus Sicht eines Detektormoduls in drei Zeitabschnitte.

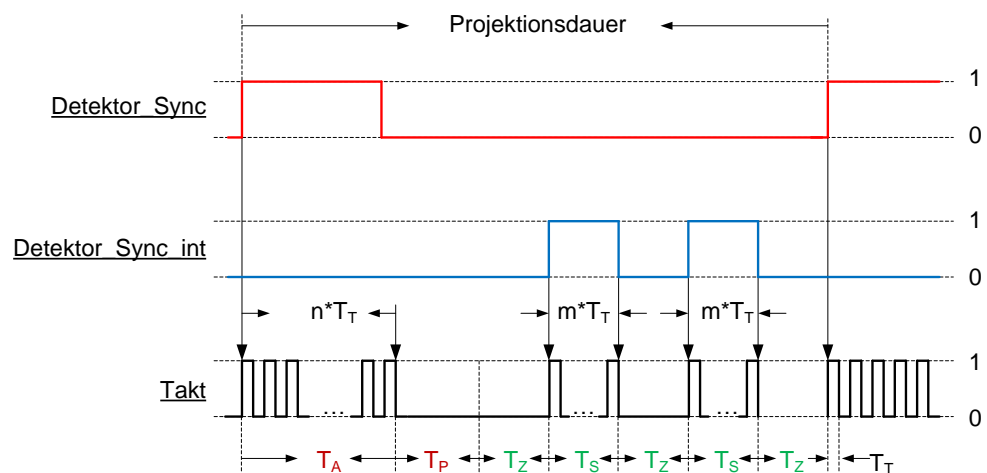
- Photonenwerte auslesen ( $T_A$ )
- Photonen zählen ( $T_Z$ )
- Photonenwerte schieben ( $T_S$ )

Im Rahmen dieser Arbeit wurde der Ablauf, entsprechend Abbildung 41, gewählt.



**Abbildung 41: Zeitverlaufsdiagramm der Detektorsynchronisation ohne TDS:** Das Auslesen von Detektorpixeln wird durch ein Synchronisationssignal gestartet.

Am Anfang eines regulären Scans, ohne Einsatz von TDS, beginnt die Kontrolleinheit eines Detektormoduls die Zählwerte der vorherigen Projektion auszulesen. Dieses Auslesen wird initiiert durch ein von einer Steuerungseinheit gesendetes Synchronisationssignal, hier mit „Detektor\_Sync“ bezeichnet. Die Kontrolleinheit eines Detektormoduls benötigt für diesen Auslesevorgang für jede auszulesende Pixelzeile ein internes Takt-signal der Dauer  $T_T$ , wie bei Detektoren dieser Bauart üblich. Die hierfür benötigte Zeit  $T_A$  ist abhängig von der Anzahl der Pixelspalten  $n$  und der Takt-dauer  $T_T$  und ist für jeden Scan konstant. Nach einer variablen Pausenzeit  $T_P$ , die dazu dient eventuelle Verzögerungen beim Auslesevorgang auszugleichen, beginnt das Detektormodul damit Photonen zu zählen, deren Anzahl pro Zeiteinheit die Höhe der Röntgenintensität repräsentiert. Dieser Zählvorgang dauert bis zum nächsten Eintreffen des Synchronisations-signals und wird mit  $T_Z$  bezeichnet. Der Ablauf wiederholt sich für jede Projektion bis zum Ende des Scans. Die Verwendung von TDS und FSC hingegen beinhaltet das Schieben von Detektorpixelwerten während der Aufnahme, allerdings nicht kontinuierlich, sondern diskret, wie Abbildung 42 verdeutlicht.

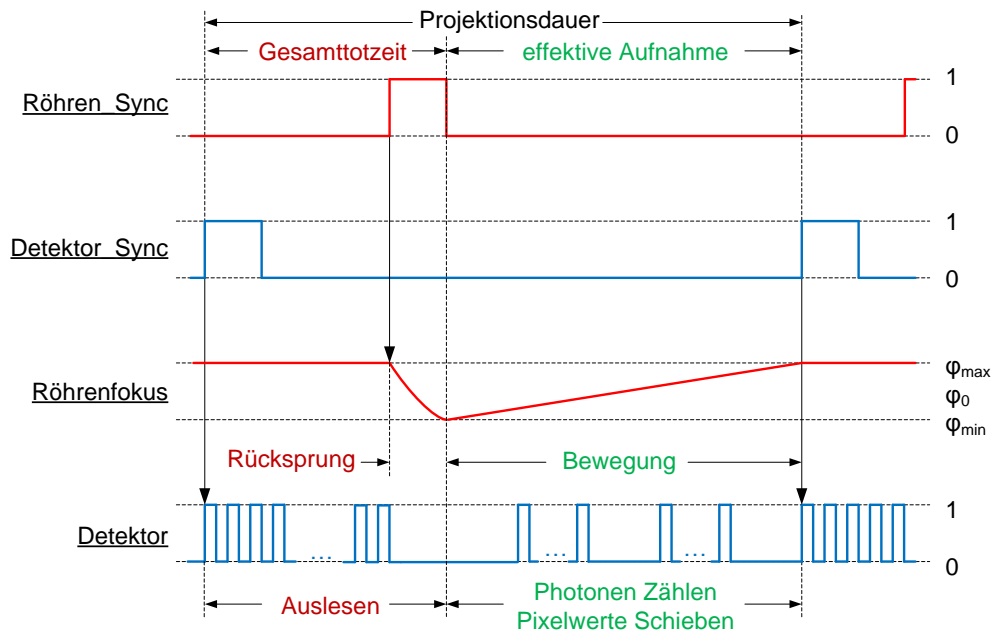


**Abbildung 42: Zeitverlaufsdiagramm der Detektorsynchronisation mit TDS:** Das Zählen von Photonen wird zum Verschieben von Pixelwerten unterbrochen.

Über die rotierende Steuerungseinheit lässt sich eine bestimmte Anzahl von TDS-Verschiebungen pro Projektion wählen. Ähnlich Abbildung 41 liest ein Detektormodul bei Beginn einer Projektion die Zählwerte der letzten Projektion für eine Zeit  $T_A$  aus und pausiert eine Zeit  $T_P$ . Der Zählvorgang beginnt ebenfalls wie in Abbildung 41 nach  $T_P$ , allerdings unterbricht die Kontrolleinheit des Detektormoduls das Zählen von Photonen und fügt einen Verschiebevorgang ein. Hierzu generiert die Kontrolleinheit aus dem Signal „Detektor\_Sync“ nach einer konstanten Zeit  $T_A + T_P + T_Z$  ein internes Signal, „Detektor\_Sync\_int“, das den Zählvorgang nach  $T_Z$  unterbricht. Während einer Unterbrechung durch „Detektor\_Sync\_int“ schiebt das Detektormodul für eine bestimmte Zeit  $T_S$  Detektorpixelwerte entgegen der Bewegungsrichtung der Gantry. Für jede Pixelzeile benötigt ein Detektormodul eine bestimmte Anzahl von  $m$  Taktsignalen. Anschließend zählt das Detektormodul wieder für eine gewisse Zeit  $T_Z$  Photonen, bis dieser Zählvorgang wieder unterbrochen wird. Abbildung 42 zeigt hierzu exemplarisch die Aufnahme einer Projektion mit zwei TDS-Verschiebungen. Da während  $T_A$  und  $T_P$  die von der Röntgenröhre applizierte Strahlung nicht genutzt werden, bilden diese Zeiten die Totzeiten des Detektors.

### 3.5.8 Minimierung von Totzeiten

Die während eines Scans von der Röntgenröhre kontinuierlich erzeugte Strahlung kann bei Verwendung von TDS und FSC nur medizinisch genutzt werden, wenn sich der Röntgenröhrenfokus kontinuierlich bewegt und die Detektormodule Photonen zählen bzw. schieben (effektive Aufnahme). Bei Rücksprung des Fokus und beim Auslesen der Zählwerte durch den Detektor ist die Röntgenstrahlung medizinisch nicht nutzbar, weshalb diese Zeiten die Totzeiten der Komponenten darstellen. Durch die Synchronisation des Brust-CT-Steuerungskonzepts werden die Totzeiten von Röntgenröhre und Detektor durch Überlagerung auf ein Minimum reduziert, sodass sich die Projektion aus einer minimalen Gesamttotzeit und der effektiven Aufnahme zusammensetzt. Abbildung 43 verdeutlicht die Überlagerung durch entsprechende Synchronisation.



**Abbildung 43: Zeitverlaufsdiagramm Bildkomponentensynchronisation:** Durch Berücksichtigung und Überlagerung von Totzeiten der Bildgebungscomponenten durch die Synchronisation überlagern sich diese, und es kommt zu einer Erhöhung der Dosiseffizienz des Brust-CT-Scanners.

Die Gesamtzeit ist dominiert durch die Totzeit der Detektormodule, die je nach Aufnahmemodi unterschiedlich lang ausfällt, aber immer länger als die Rücksprungzeit des Röntgenröhrenfokus ist. Das Auslesen der Detektormodule und damit der Beginn der Projektion werden initiiert durch das Synchronisationssignal „Detektor\_Sync“. Durch Kopplung des Synchronisationssignals „Röhren\_Sync“, zum Start des Rücksprungs des Röntgenröhrenfokus, an das Synchronisationssignal „Detektor\_Sync“, erfolgt der Rücksprung noch während des Auslesevorgangs der Detektormodule. Im Anschluss beginnt die kontinuierliche Bewegung des Röhrenfokus gleichzeitig mit dem Zählen von Photonen durch die Detektormodule. Um diese beiden Signale entsprechend Abbildung 43 nacheinander auszulösen, müssen die Steuerungseinheiten in der Lage sein, solche Signale mit kurzen Signallaufzeiten und minimaler Fluktuation (Jitter) zu erzeugen, da sonst die Bildqualität, wie in Abschnitt 3.1.2.2 beschrieben, rapide absinkt. Für den Brust-CT-Scanner wurde deshalb spezifiziert, dass die Signallaufzeit der Synchronisationssignale nur maximal ein Hundertstel der Projektionszeit betragen darf. Der hierbei auftretende Jitter darf nur maximal einem Tausendstel der Projektionszeit bzw. 10% der Rücksprungzeit entsprechen. Dies bedeutet für eine Projektionszeit des Scanners von 1 ms einen maximalen Jitter von 1  $\mu$ s bei einer maximalen Signallaufzeit von 10  $\mu$ s. Die durch die Kopplung der beiden hochgenauen Synchronisationssignale resultierende Überlagerung der Totzeiten erhöht die effektive Aufnahmedauer in Bezug zur Projektionsdauer und steigert so die Dosiseffizienz des gesamten Brust-CT-Scanners.

## 3.6 Sicherheitskonzept

### 3.6.1 CT-Scanner als sicherheitskritisches System

In technischen Systemen, wie im Falle eines Brust-CT-Scanners, können Fehler durch Menschen oder technische Komponenten verursacht werden. Wenn durch das Auftreten eines solchen Fehlers eine Gefährdung für Menschen entstehen kann, wie bei einem CT-Scanner, handelt es sich um ein sicherheitskritisches System. Die möglichen für das Steuerungskonzept relevanten Gefährdungen, welche von dem sicherheitskritischen System Brust-CT-Scanner für den Menschen ausgehen können, lassen sich in vier Teilbereiche gliedern.

- Gefährdung durch mechanische Bewegungen
- Gefährdung durch elektrischen Strom
- Gefährdung durch Röntgenstrahlung
- Gefährdung durch falsche bzw. zu hohe Parameter

Solche Gefährdungen auszuschließen oder zumindest kalkulierbar zu machen, ist die Aufgabe eines adäquaten Sicherheitskonzepts. Als eines der bekanntesten Beispiele für ein nicht ausreichendes oder falsches Sicherheitskonzept bei einem medizinischen Gerät sei der Linearbeschleuniger Therac-25 genannt. Durch einen schweren Softwarefehler und falsches Reagieren darauf, kamen in den Jahren 1985 bis 1987 mehrere Menschen ums Leben bzw. wurden schwer verletzt. [151; 152] Grundlage für ein richtiges Sicherheitskonzept ist eine ausführliche und vollständige Gefahrenanalyse entsprechend der Norm DIN EN ISO 14971 [56], wie sie von jedem Hersteller eines medizinisches Gerätes durchgeführt werden muss. Auf sie wird im Rahmen dieser Arbeit nicht weiter eingegangen. Stattdessen werden die für das Steuerungskonzept entscheidenden Maßnahmen zur Gefahrenvermeidung beschrieben. Weitere Normen, die beim Entwurf eines adäquaten Sicherheitskonzepts eines CT-Scanners berücksichtigt werden müssen, ist die Norm für die Herstellung medizinisch elektrischer Geräte DIN EN 60601-1 [46], die Norm für Medizingeräte-Software DIN EN 62304 [55] sowie die Grundnorm für funktionale Sicherheit DIN EN 61508-1 [50]. [13; 32; 76; 81–83; 98; 101; 137]

### 3.6.2 Maßnahmen zur Gefahrenvermeidung

Großes Gefährdungspotential durch den Brust-CT-Scanner für Menschen geht, neben der Röntgenstrahlung, von elektrischem Strom und von mechanischen Bewegungen aus. Um diese Gefährdungspotentiale zu minimieren, sind elektrische und mechanische Systeme nach entsprechenden gültigen Normen für Medizingeräte DIN EN 60601-1 [46] und funktionaler Sicherheit DIN EN 61508-1 [50] aufzubauen. Zusätzlich überwacht im Brust-CT-Scanner ein Stromkreis, hier als Interlock bezeichnet, sicherheitskritische Komponenten, um im Falle eines Fehlers, Gefährdungen jederzeit zu unterbinden und einen sicheren Zustand herzustellen. Jede Komponente, die einen Fehler erkennen kann, verfügt über einen Öffner, um damit den Interlock öffnen zu können. Abhängig von der Art des geöffneten Interlocks wird im Anschluss entweder mechanische oder elektrische Sicherheit, z.B. durch Abschalten von elektrischen Spannungen oder Stoppen von Antrieben, hergestellt.

Ebenfalls gefährlich für den Menschen, insbesondere den Patienten, ist die Applikation von zu viel oder unnötiger Strahlendosis. Der Brust-CT-Scanner ist dafür optimiert, höchste Bildqualität zu ermöglichen, dabei aber nur ein Mindestmaß an Dosis zu applizieren. Im Fehlerfall sind aber dennoch Szenarien denkbar, in denen zu viel Dosis appliziert werden könnte, z.B. bei falscher Parameterübernahme auf Seiten der Kontrolleinheit der Röntgenröhre. Um dies auszuschließen, müssen die Steuerungseinheiten die einstellbaren Parameter aller Komponenten des Brust-CT-Scanners vor Beginn eines Scans auslesen und mit den vorgegebenen Werten des Leitrechners vergleichen. Zur Applikation von unnötiger Dosis hingegen kann es kommen, wenn während eines Scans der Detektor ausfällt. In diesem Szenario wäre die applizierte Dosis unnötig, da ohne Projektionsdaten kein diagnostischer Wert vorliegt. Aus diesem Grund ist jede Komponente so ausgelegt, dass bei Erkennen eines Fehlers die Kontrolleinheit oder die übergeordnete Steuerungseinheit echtzeitfähig ist und mit entsprechenden Maßnahmen darauf reagieren kann.

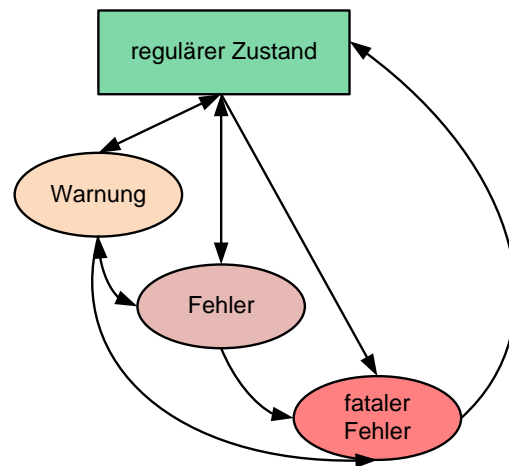
Komponenten des Brust-CT-Scanners sind insbesondere durch falsche Parametrierung gefährdet. Erhitzt sich, z.B. durch einen hohen Röntgenröhrenstrom während eines unverhältnismäßig lang andauernden Scans, die Röntgenröhrenelektronik zu stark, kann diese Schaden nehmen. Um die Übergabe falscher Parameter auszuschließen, verfügen Komponenten auf der Kontrollebene über eine interne Begrenzung ihres Wertebereichs und über die Möglichkeit einer Plausibilitätsprüfung ihrer einstellbaren Parameter. Wenn versucht wird, ungültige Parameter zu setzen, reagiert die entsprechende Steuerungseinheit mit einer Fehlermeldung. Zusätzlich sind kritische Werte einer Komponente jederzeit durch die übergeordnete Steuerungseinheit abrufbar und werden von der entsprechenden Kontrolleinheit der Komponente selbst überwacht. Bei Überschreiten fest definierter Grenzwerte erfolgt eine entsprechende Meldung an die Steuerungseinheit.

Die von der Steuerung auf den verschiedenen Ebenen getroffenen Maßnahmen zur Gefährdungsvermeidung lassen sich folgendermaßen zusammenfassen:

- Echtzeitfähige Fehlermeldungen durch Interlockkreis oder Bussystem
- Parameterabfrage und Überwachung durch die Steuerungseinheiten
- Interne Wertebegrenzung und Plausibilitätsprüfung durch die Kontrolleinheiten

### 3.6.3 Eskalationsstufen zur Fehlermeldung

Um einen Fehler zu vermeiden oder auf sein Auftreten entsprechend zu reagieren, müssen Komponenten, die eine Abweichung von einer Anforderung feststellen, diese einer übergeordneten Einheit mitteilen. Zur Mitteilung einer Abweichung wird für den Brust-CT-Scanner folgende Einteilung von Meldungen in unterschiedlichen Eskalationsstufen vorgesehen:



**Abbildung 44: Die Eskalationsstufen des Brust-CT-Scanners:** Das Auftreten eines schwerwiegenden Fehlers kann die höchste Eskalationsstufe „fataler Fehler“ auslösen.

„Warnungen“ sind Abweichungen von einem regulären Zustand, die über einem bestimmten Grenzwert liegen. Dies kann z.B. der Messwert eines Temperatursensors sein, der sich über eine vorher festgelegte Temperaturschwelle erhöht hat. Erhält eine Steuerungseinheit von einer untergebenen Kontrolleinheit eine Warnung, kann sie Gegenmaßnahmen ergreifen, damit sich der Zustand wieder normalisiert. „Warnungen“ haben keinen Einfluss auf die Qualität des Scans.

Wird auf eine „Warnung“ nicht entsprechend reagiert und es wird ein weiterer Grenzwert überschritten kann es zu einem Fehler kommen. Hierzu kommt es auch wenn eine unwesentliche Komponente des Brust-CT-Scanners beschädigt wird oder ausfällt. Kommt es zu einem „Fehler“ wird die Qualität eines Scans unter Umständen beeinträchtigt, der Scan wird aber nicht abgebrochen.

Bei einer Häufung von Fehlern oder beim Erkennen eines besonders schweren Fehlers, wie z.B. der Ausfall einer wesentlichen Komponente des Brust-CT-Scanners, wird ein „fataler Fehler“ erzeugt. Dieser tritt auf, wenn akute Gefahr für Menschen oder Komponenten des CT-Scanners besteht, sowie, wenn die Qualität eines Scans durch eine Häufung von Fehlern zu stark beeinträchtigt wurde. Bei Auftreten eines „fatalen Fehlers“ wird der Scan sofort abgebrochen, die Komponente, welche den Fehler erkannt hat, geht in einen speziellen Fehlerzustand und der Brust-CT-Scanner stellt einen sicheren Gesamtzustand her. Die Meldung eines „fatalen Fehlers“ erfolgt hierbei in Echtzeit über den Interlockkreis mithilfe fehlersicherer Komponenten. Um diesen Fehlerzustand wieder verlassen zu können, muss die Fehlerursache behoben und die Behebung explizit durch den Benutzer bestätigt werden.



## **4 Realisierbarkeit des Steuerungskonzepts**

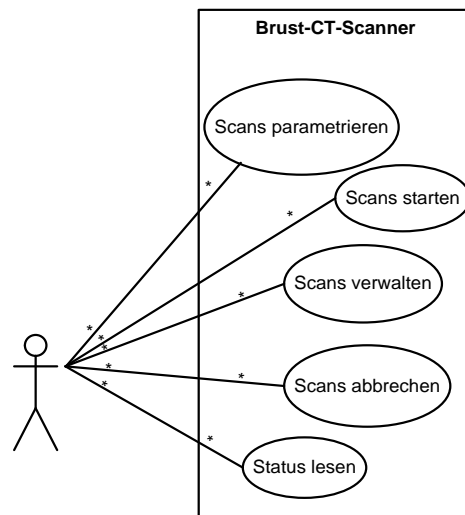
Kapitel 3 beschreibt ein mögliches Konzept zur Steuerung des Brust-CT-Scanners. Das folgende Kapitel zeigt die Umsetzung dieses Konzepts und wie damit die Anforderungen des Brust-CT-Scanners erfüllt werden. Hierzu wird auf die Realisierung der einzelnen Schichten des Ebenenmodells des Brust-CT-Scanners eingegangen und besonders die Realisierung der Steuerungseinheiten detailliert betrachtet, da diese den Kern des Steuerungskonzepts bilden. Die Auswahl eines RTOS für die Steuerungseinheiten ist auf Grund verschiedener Faktoren nicht trivial, weshalb Messungen an vier verschiedenen RTOS auf den Steuerungseinheiten durchgeführt wurden. Auf Basis dieser Messungen und den Angaben der Hersteller wird ein RTOS für die Verwendung im Brust-CT-Scanner empfohlen. Eine weitere Herausforderung bei der Realisierung des Konzepts ist die für die Übertragung der Projektionsdaten notwendige Übertragungsrate von mindestens 7 GBit/s. Diese ist für 10-GBit/s-Ethernet-Netzwerkkarten in verschiedenen Veröffentlichungen unterschiedlich hoch angegeben. [105; 109; 186] Messungen hierzu sollen zeigen, ob die für den Brust-CT-Scanner geforderte Übertragungsrate von 7 GBit/s mit aktueller Hardware erreichbar ist. Neben der Übertragungsrate ist auch die Fluktuation der Synchronisationssignale, ausgedrückt als Jitter, die die Synchronisation des Brust-CT-Scanners mindestens erreichen muss, festgelegt. Dieser Jitter muss weniger als 1  $\mu$ s betragen um die Bildqualität nicht negativ zu beeinflussen. Um zu überprüfen ob die Synchronisation der Steuerungseinheiten diese Forderung erfüllen kann, wurden Messungen hierzu durchgeführt, welche dieses Kapitel abschließen.

### **4.1 Hardwareebenen**

#### **4.1.1 Realisierung des Leitrechners**

##### **4.1.1.1 Software**

Der Leitrechner repräsentiert im Brust-CT-Scanner die Leitebene (mit Benutzerschnittstelle) und das Rekonstruktionssystem. Als Benutzerschnittstelle muss er die in Abbildung 45 gezeigten möglichen Anwendungsfälle für den Benutzer bereitstellen können. Dies geschieht mit der Software „BreastCtHMI“, die speziell für die Steuerung des dedizierten Brust-CT-Scanners entwickelt wurde.



**Abbildung 45: Die verschiedenen Anwendungsfälle des Brust-CT-Scanners.**

„BreastCtHMI“ ist ebenfalls zuständig für die leittechnischen Aufgaben, die sich auf das Auslösen entsprechender Aktionen auf der Seite der Steuerungseinheiten beschränken, sowie auf das Senden von Parametern und Empfangen von Statusmeldungen. Da BreastCtHMI auf dem Leitrechner somit einen Teil der Gesamtsteuerung des Brust-CT-Scanners darstellt, wurde diese Softwarekomponente entsprechend der Norm für programmierbare elektrische medizinische Systeme, DIN EN 60601-1-4 [49], und der Norm für Software-Lebenszyklus-Prozesse in Medizingeräten, DIN EN 62304 [55], implementiert. [82; 83]

Das Empfangen, Rekonstruieren und Darstellen von Projektionsdaten wird von dem Softwarepaket „ImpactToolchain“ übernommen. Dieses Paket ist bereits in anderen Scannern im Einsatz, weshalb auf ihre Anwendungsfälle und Funktionen nicht näher eingegangen wird. Die „ImpactToolchain“ ist nicht Teil der Steuerung des Brust-CT-Scanners, weshalb sie nicht der Norm DIN EN 60601-1-4 unterworfen ist, sondern nur der Norm DIN EN 62304 [55]. [82; 83]

#### 4.1.1.2 Hardware

Der Leitrechner befindet sich auf oberster Ebene des Ebenenmodells für die Steuerung des Brust-CT-Scanners, weshalb die an ihn gestellten zeitlichen Anforderungen geringer sind als in unteren Ebenen. Des Weiteren wird er räumlich, vom Brust-CT-Scanner getrennt, in einem Bedienraum aufgestellt, wodurch er nicht den härteren Bedingungen des Untersuchungsraumes ausgesetzt ist. Aus diesen beiden Gründen ist es zweckmäßig, den Leitrechner auf Basis eines Standard-PCs aufzubauen. Neben den allgemeinen Anforderungen, die an einen PC für medizinisches Umfeld zu stellen sind, muss der Leitrechner über entsprechend leistungsfähige Komponenten verfügen, die den Anforderungen der Rekonstruktionssoftware gerecht werden. Außerdem benötigt er eine 10-GBit/s-Ethernet-Netzwerkkarte zum Empfangen der Projektionsdaten. [47; 84] Die Anforderungen ergeben sich damit zu:

- Intel Core i7 CPU mit mindestens sechs Prozessorkernen
- Grafikkarte der neuesten Generation von NVIDIA (Geforce GTX Titan)
- 512 GByte SSD-Festplatte für die Rekonstruktion der Projektionsdaten
- vier 2 TByte HDD-Festplatten für die Archivierung der Projektionsdaten
- 128 GByte DDR3-SDRAM Hauptspeicher als Puffer für die Projektionsdaten
- 10-GBit/s-Ethernet-Netzwerkkarte für den Empfang der Projektionsdaten
- elektromagnetisch verträglich nach DIN EN 60601-1-2

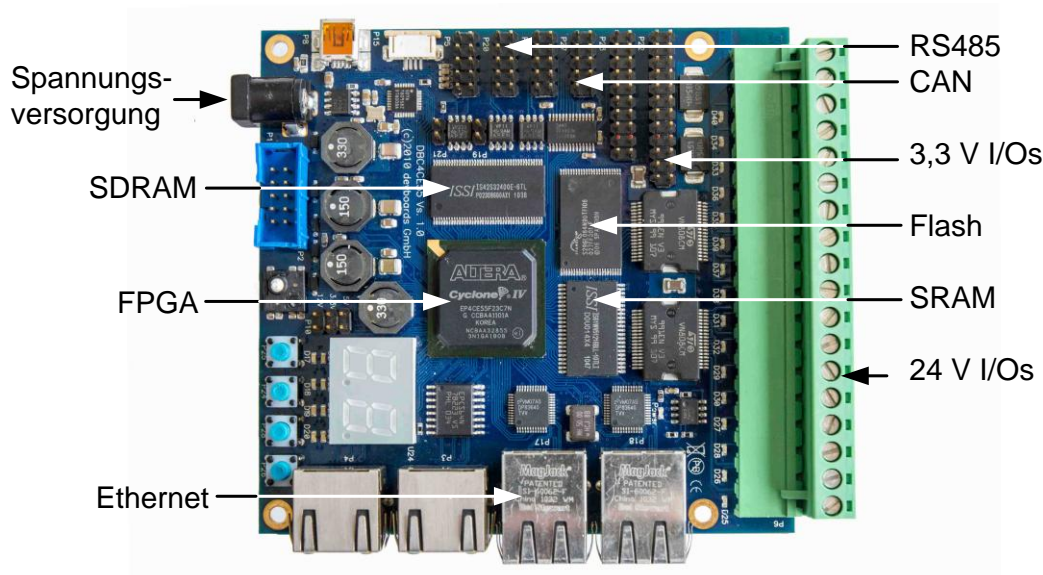
#### 4.1.2 Realisierung der Steuerungseinheiten

##### 4.1.2.1 Grundplatine DBC4CE55

Als Bindeglied zwischen Leitreechner und Kontrolleinheiten kommt den Steuerungseinheiten auf rotierender und statischer Seite besondere Bedeutung zu. Aufgrund ihres ähnlichen Aufgabenbereichs und um Kosten für eine Entwicklungsumgebung möglichst niedrig zu halten, ist es vorteilhaft, den gleichen Systemtyp auf beiden Seiten zu verwenden. Hier eignet sich besonders ein eingebettetes System auf Basis eines SOPC, da diese Systeme ein Höchstmaß an Flexibilität aufweisen und sich mit ihnen schnell auf Änderung der Spezifikationen bei Erstellung eines neuartigen Systems reagieren lässt, wie in Kapitel 2.3.3 beschrieben. Zahlreiche Veröffentlichungen zeigten bereits die Vorteile dieser Technologie bei der Erfüllung anspruchsvoller Steuerungsaufgaben. [9; 14; 19; 143; 159; 244] Weitere Anforderungen an die Steuerungseinheiten ergeben sich aus der in Kapitel 3 vorgestellten Systemarchitektur, dessen Kommunikationsmodell und der Synchronisierung der Bildgebungskomponenten. Die Anforderungen lassen sich zu folgender Liste zusammenfassen:

- Mikroprozessor und FPGA auf einem Chip (SOPC)
- zwei Ethernet-Schnittstellen
- zwei CAN-Schnittstellen
- sieben RS485-Kanäle
- kompakte Bauform
- zertifizierbares/zertifiziertes RTOS

Für die Steuerungseinheiten des Brust-CT-Scanners wurde je ein SOPC-Development-Board mit der Bezeichnung DBC4CE55 gewählt, siehe Abbildung 46, da es zum Zeitpunkt dieser Arbeit das einzige kommerzielle SOPC-Development-Board war, das über eine entsprechende Anzahl an CAN und Ethernet Schnittstellen, sowie einen kompakten Aufbau, verfügte. Außerdem bietet der Hersteller dieses FPGAs ein Zertifizierungspaket für das auf dem Board enthaltende FPGA an, mit dem sich eine Zertifizierung entsprechend DIN EN 60601-1 [46] und DIN EN 61508-1 [50] durchführen lässt. [4; 7]



**Abbildung 46: Das SOPC-Development-Board im Detail:** Es besitzt bei kompakter Bauweise eine Vielzahl von Schnittstellen sowie ein leistungsfähiges FPGA.

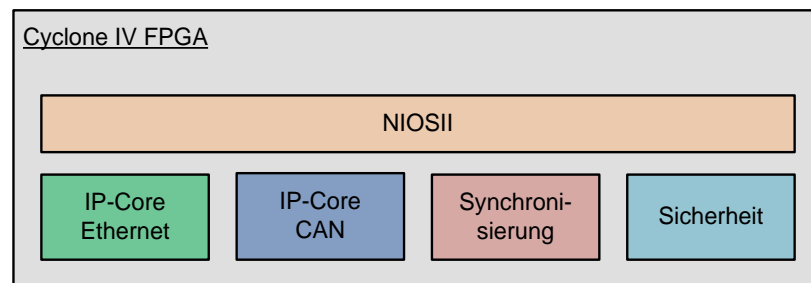
Das SOPC-Development-Board besitzt folgende Merkmale und Schnittstellen [42]:

- Altera Cyclone IV FPGA
- SDRAM (16 MByte)
- SRAM (1 MByte)
- Flash-Speicher (8 MByte)
- zwei Ethernet-Schnittstellen (zwei Ethernet-Transceiver)
- zwei CAN-Schnittstellen (zwei CAN-Transceiver)
- vier RS485-Kanäle (vier RS485-Transceiver)
- 24 V Ein-/Ausgabe (16 Bit I/Os)
- 3,3 V Ein-/Ausgabe (32 Bit I/Os)

Auf dem FPGA kommt ein in der Hardwarebeschreibungssprache VHDL vorliegender Softcore-Prozessor des Typs NIOSII zum Einsatz. FPGA und NIOSII-Prozessor wurden an die Anforderungen des Brust-CT-Scanners angepasst (Peripherie, Schnittstellen sowie Ein-/Ausgänge). Der verwendete NIOSII-Prozessor verfügt über eine Taktrate von 75 MHz, eine „General Purpose Processor“ (GPP)-Architektur sowie einen 32 Bit „Reduced Instruction Set Computer“- (RISC) Befehlssatz. NIOSII-Prozessoren sind in drei verschiedenen leistungsfähigen Versionen auf einem FPGA instanzierbar. Je leistungsfähiger der Prozessor gewählt ist, desto größer ist sein Platzbedarf auf dem FPGA. [5; 227; 244]

- NIOSII/E (Economy): Kleinste, aber langsamste Version des NIOSII.
- NIOSII/S (Standard): Standard-Version des NIOSII.
- NIOSII/F (Fast): Schnellste, aber größte Version des NIOSII.

Die Transceiver für CAN und Ethernet realisieren nur die erste Schicht des OSI-Modells. Die zum Betrieb benötigten zweiten Schichten können als IP-Core für den NIOSII-Prozessor erworben und entsprechend den jeweiligen Lizenzen verwendet werden. Abbildung 47 gibt eine Übersicht über die auf dem FPGA verwendeten VHDL-Codeblöcke. Neben dem NIOSII Softcore-Prozessor und den IP-Cores für CAN und Ethernet befinden sich Verknüpfungsschaltungen für Synchronisations- sowie Sicherheitsfunktionen auf dem FPGA. Da es sich bei VHDL um eine Hardwarebeschreibungssprache handelt, deren Anweisungen auf dem FPGA in Logikgatter realisiert werden, liegt die damit umgesetzte Funktionalität nicht in Software, sondern in Hardware vor. Im Gegensatz zu Software, welche generell als fehlerbehaftet angesehen werden kann, ist Hardware eindeutig test- und prüfbar, weshalb eine Umsetzung von Funktionalität durch Logikgatter – und damit in Hardware – die Fehlersicherheit eines Systems erhöht und deren Nachweis erleichtert. [55; 76; 190]



**Abbildung 47: Die VHDL-Funktionsblöcke der Steuerungseinheiten:** Die FPGAs der Steuerungseinheiten enthalten einem NIOSII Softcore-Prozessor, zwei kommerzielle IP-Cores für CAN und Ethernet sowie je einen Codeblock für Synchronisations- und Sicherheitsfunktionen.

#### 4.1.2.2 Echtzeitbetriebssysteme für den NIOSII Softcore-Prozessor

Beide im vorherigen Kapitel beschriebenen Steuerungseinheiten sind ohne Betriebssystem lauffähig. Mit der Programmiersprache C sowie mit einer vom Hersteller mitgelieferten Entwicklungsumgebung kann ein Anwenderprogramm erstellt werden. Aufgrund der Komplexität der umzusetzenden Koordinierungsaufgaben und aus Gesichtspunkten der funktionalen Sicherheit ist der Einsatz eines RTOS empfehlenswert. Für SOPCs existiert eine Vielzahl von RTOSs, die allerdings nur bestimmte Prozessortypen unterstützen. Der verwendete NIOSII-Prozessor wird von folgenden RTOS unterstützt: [2; 6]

- eCos
- embOS
- Erika Enterprise
- Euros RTOS
- FreeRTOS/SafeRTOS
- Linux
- $\mu$ C/OSII

- osCAN
- ThreadX
- $\mu$ CLinux
- VxWorks

Die genannten RTOS unterscheiden sich anhand verschiedener Eigenschaften, die den Herstellerangaben entnehmbar sind. Tabelle 4 vergleicht fünf der oben aufgelisteten RTOS (von diesen waren Demoversionen zum Download verfügbar) anhand ihrer Kosten, technischer Daten sowie eventuell vorhandener Zertifikate bzw. Zertifizierungspakete entsprechend gültiger Normen. Verfügt ein RTOS laut Tabelle 4 über ein entsprechendes Zertifikat, so bedeutet dies, dass dieses Zertifikat zusätzlich zu den Lizenzen vom Hersteller des RTOS erworben werden muss. Hierzu wird dem RTOS-Hersteller der CPU-Typ sowie der Compiler mitgeteilt, auf dem das RTOS kompiliert werden soll. Der RTOS-Hersteller lässt das RTOS daraufhin für diese CPU-Compiler-Kombination bei der benannten Stelle nach der gewünschten gültigen Norm zertifizieren. Dieses Vorgehen basiert darauf, dass der jeweilige RTOS-Hersteller bereits eine „Grundversion“ seines RTOS auf einer Referenz-CPU zertifizieren hat lassen. Anhand dieses Exempels kann er weitere CPU-Typen zertifizieren und die Kosten hierfür abschätzen. [6; 11; 63–65; 167–169; 207; 236; 237]

**Tabelle 4: Vergleich von fünf RTOS für den NIOSII-Prozessor.**

Eigenschaft	embOS	FreeRTOS	SafeRTOS	$\mu$ C/OSII	ThreadX
Gesamtkosten <sup>1</sup>	niedrig	keine	hoch	hoch	hoch
unterstützte Prozessoren	~ 50	~ 30		~ 60	~ 40
Kernelgröße (ROM)	3 KByte	4 – 9 KByte	6 – 15 KByte	6 – 26 KByte	~ 5 KByte
Prioritäten	255	$\infty$		64	1024
Tasks	$\infty$	$\infty$		254	$\infty$
FDA510(k)	nein	nein	ja	ja	ja
IEC-62304	nein	nein	ja	ja	ja
IEC-60601	nein	nein	nein	ja	ja
ISO-14971	nein	nein	nein	ja	ja
IEC-61508	nein	nein	ja	ja	ja

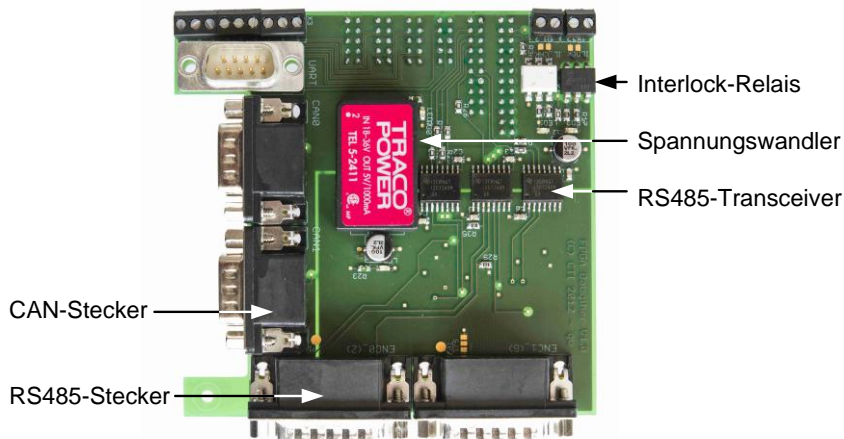
<sup>1</sup>Die Gesamtkosten für ein RTOS beziehen sich auf eine Lizenz für den Einsatz mehrerer dieser RTOS in jedem Brust-CT-Scanner („Product Line Licence“) und, wenn möglich, ein Zertifikat entsprechend IEC-61508. Trotz unterschiedlich hoher Lizenz- und Zertifizierungskosten der jeweiligen RTOS-Hersteller, ergibt sich in Summe für alle zertifizierbaren RTOS ein ähnlich hoher Betrag, der hier aus Datenschutzgründen nicht genannt wird.

Aus Tabelle 4 geht hervor, dass alle fünf RTOS über eine kleine Kernelgröße, mindestens 64 verschiedene Prioritäten und eine große Anzahl an Tasks verfügen. Eine große Anzahl von Prioritäten und Tasks erhöht die Flexibilität bei der Softwareentwicklung. Eine kleine Kernelgröße ist von Vorteil, da hierdurch mehr Speicher für die eigene Anwendung zu Verfügung steht. Falls in zukünftigen Weiterentwicklungen des Brust-CT-Scanners ein anderer Prozessor statt des NIOSII eingesetzt wird, ist wichtig, dass ein RTOS auch eine Vielzahl von häufig anzutreffenden Prozessortypen unterstützt, was ebenfalls alle fünf RTOS sicherstellen.

Für eine Zertifizierung des Brust-CT-Scanners sind die einem RTOS beigefügten Zertifikate von besonderer Bedeutung. Das RTOS embOS ist entsprechend der Herstellerangaben nicht nach einer Norm für sicherheitskritische Systeme zertifizierbar. ThreadX und  $\mu$ C/OSII lassen sich nach allen für die funktionale Sicherheit im Allgemeinen (IEC-61508) und im Bereich der Medizintechnik (FDA510(k), IEC-62304, IEC-60601, ISO-14971) gültigen Normen zertifizieren. FreeRTOS stellt einen speziellen Fall dar, da dieses ohne Zertifikate und kostenfrei beziehbar ist. Bei Bedarf ist es durch das kostenpflichtige SafeRTOS ersetzbar, das über die gleiche interne Softwarestruktur verfügt und nach FDA510(k), IEC-62304 und IEC-61508 zertifizierbar ist. In den Produktdatenblättern von SafeRTOS ist eine Möglichkeit zur Zertifizierung nach IEC-60601 und ISO-14971 nicht zu finden. Nachfragen beim Hersteller ergaben aber, dass dies prinzipiell möglich sein müsste.

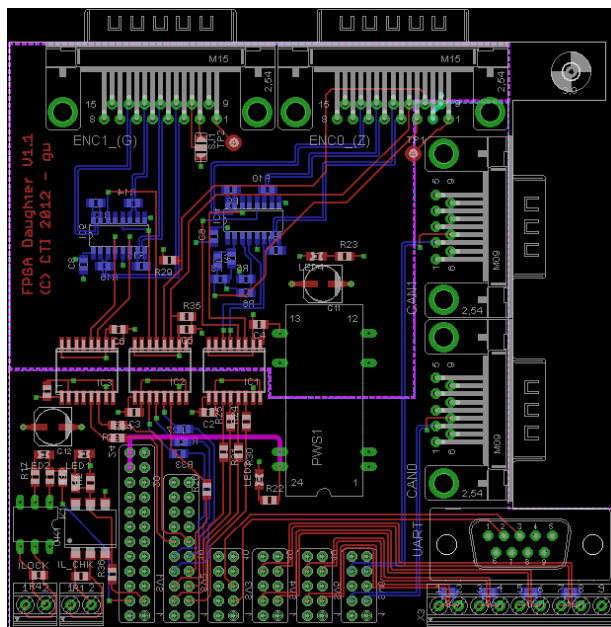
#### 4.1.2.3 Erweiterungsplatine für das DBC4CE55 Development-Board

Die Steuerungseinheiten übernehmen im Brust-CT-Scanner auch die Aufgabe der Synchronisation. Hierzu muss die statische Steuerungseinheit die für die Rekonstruktion nötigen Positionen der Gantryrotation und –translation erfassen. Da jeder Positionssensor seine Werte über drei RS485-Kanäle ausgibt, siehe Kapitel 3.5.5, benötigt die statische Steuerungseinheit zur Positionserfassung sechs RS485-Kanäle. Über einen weiteren Kanal wird ein Synchronisationssignal auf die rotierende Seite übertragen, was die Anzahl der auf statischer Seite nötigen RS485-Kanäle auf sieben erhöht. Auf rotierender Seite wird das Signal entsprechend empfangen und an Röntgenröhre und Detektor verteilt, wofür auf rotierender Seite insgesamt drei RS485-Kanäle erforderlich sind. Für eventuelle Erweiterungen sind auf rotierender Seite zwei weitere RS485-Kanäle vorzusehen. Da die Anzahl der für die Steuerungseinheiten benötigten RS485-Kanäle die Anzahl der auf dem DBC4CE55-Board vorhandenen übersteigt, wurde im Rahmen dieser Arbeit mit der Firma CTI eine Erweiterungsplatine, siehe Abbildung 48, entwickelt.



**Abbildung 48: Erweiterungsplatine für das DBC4CE55 FPGA-Board:** Sie ergänzt das Board um weitere RS485-Transceiver sowie einen Öffner für den Interlockkreis.

Um auf das DBC4CE55-Board aufgesteckt werden zu können, ist es an dessen Form angepasst und verfügt über ein kompaktes, zweilagiges Layout, wie in Abbildung 49 zu erkennen, mit den entsprechenden Schnittstellen. Es erweitert die Anzahl der auf dem Board vorhandenen RS485-Kanäle um weitere sechs, auf insgesamt zehn Kanäle. Außerdem verfügt die Erweiterungsplatine über einen Öffner, um im Falle eines „fatalen Fehlers“ den Interlockkreis öffnen zu können. Hierauf wird in Kapitel 4.2.1 näher eingegangen.



**Abbildung 49: Zweilagiges Layout der Erweiterungsplatine für das FPGA-Board.**

Über sechs RS485-Kanäle erfasst die statische Kontrollleinheit mit einem Codeblock in VHDL kontinuierlich die Position der Gantry. Nach fest kalkulierbaren Zeiten wird die aktuelle Position der Gantry in ein Register des FPGAs geschrieben. Das Abspeichern der Positionen erfolgt im FPGA in wenigen Takten und damit ohne großen Zeitversatz.

Zeitgleich mit dem Abspeichern wird vom FPGA ein Synchronisationssignal zum Starten der Projektion erzeugt und an die rotierende Steuerungseinheit gesendet. Diese empfängt das Signal und generiert, ebenfalls mit einem VHDL-Codeblock, die Synchronisationssignale für die Röntgenröhre und die Detektormodule. Eventuell auftretende Signallaufzeiten und Latenzen in den Komponenten lassen sich bis in den Bereich eines FPGA-Takts (20 ns) ausgleichen. Im Anschluss an den Beginn einer Projektion liest das auf dem NIOSII-Prozessor der statischen Steuerungseinheit ablaufende Anwenderprogramm die im FPGA gepuffert vorliegende Gantryposition zeitunkritisch aus. [141; 248]

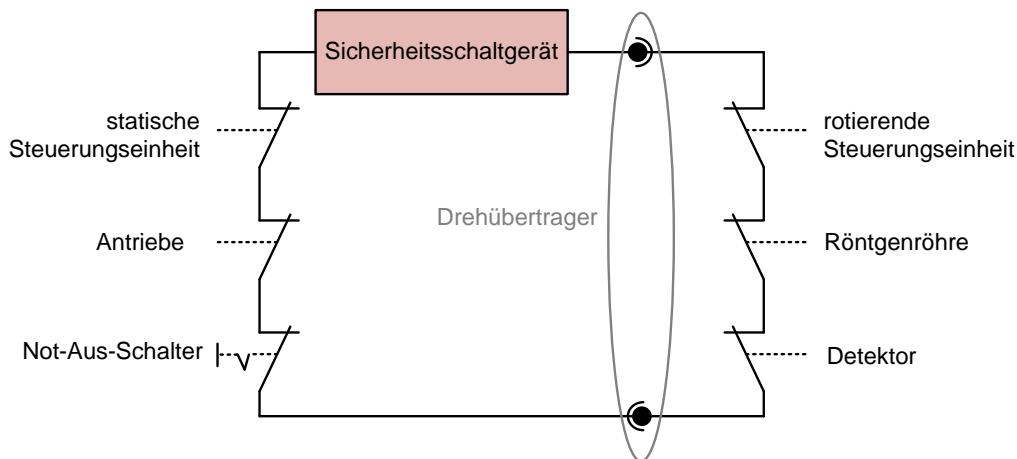
### 4.1.3 Realisierung der Kontrolleinheiten

Da es sich bei den Komponenten des Brust-CT-Scanners entweder um Standard-Komponenten oder um speziell entwickelte Einzelkomponenten handelt, die von ihren jeweiligen Herstellern gefertigt werden, ist das Hard- und Softwaredesign der Kontrolleinheiten nicht direkt beeinflussbar, aber auch nicht relevant, sofern die Komponenten die an sie gestellten Anforderungen erfüllen. Die Anforderungen sind im Rahmen einer Spezifikation festgelegt, die für jede Komponente eine Kommunikationsschnittstelle und einen definierten Befehlssatz sowie auslesbare Stadien, die von den Kontrolleinheiten der Komponenten im Rahmen eines endlichen Automatenmodells abgearbeitet werden, definiert. Hierfür muss der Hersteller garantieren.

## 4.2 Sicherheitsfunktionen

### 4.2.1 Interlockkreis

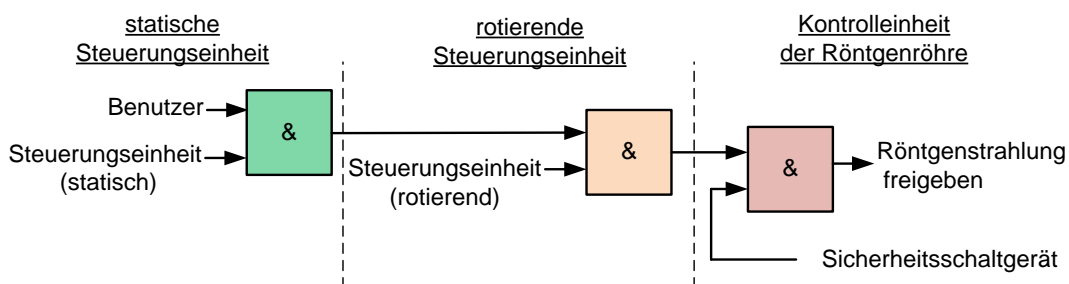
Im Falle des Auftretens eines „fatalen Fehlers“ muss die Reaktion auf diesen, also das Erreichen eines sicheren Zustandes durch Stoppen der Strahlung und aller mechanischen Bewegungen, entsprechend der Norm 60601-2-44 [48] erfolgen. Hierfür verfügt jede Komponente der Kontrollebene und der Steuerungsebene des Brust-CT-Scanners über ein spezielles Relais zum Öffnen eines Interlockkreises. Der Interlockkreis wird von einem Sicherheitsschaltgerät vom Typ Pilz PNOZ\_S3 gespeist und überwacht. Die sicherheitstechnischen Kenndaten dieses Sicherheitsschaltgeräts entsprechen unter anderem der Norm DIN EN 61511, die als Umsetzung der Norm DIN EN 61508 für die Prozessindustrie gilt, und ist damit als funktional sicher anzusehen. Es stellt bei Unterbrechung des Interlockkreises, innerhalb einer fest definierten Zeit, mechanische und elektrische Sicherheit her, indem es alle Antriebe und Spannungen abschaltet, die gefährlich für Menschen sein können. Abbildung 50 zeigt das Sicherheitsschaltgerät und den Interlockkreis mit den entsprechenden Öffner-Kontakten der Komponenten, die einen schweren Fehler erkennen können, sowie einen Not-Aus-Schalter. [184]



**Abbildung 50: Interlockkreis des Brust-CT-Scanners mit Sicherheitsschaltgerät:** Jede Kontroll- oder Steuerungseinheit kann den Interlockkreis öffnen.

## 4.2.2 Strahlungsfreigabe

Um auszuschließen, dass aufgrund eines Fehlers Strahlung erzeugt wird, muss das Auslösen von Strahlung laut DIN EN 60601-2-44 [48] besonders abgesichert sein. Das Auslösen von Strahlung im Brust-CT-Scanner setzt daher voraus, dass der Interlockkreis geschlossen ist und kein Fehler in einer der Komponenten vorliegt. Zusätzlich müssen die Steuerungseinheiten durch ein Signal ihrerseits das Auslösen von Strahlung explizit bestätigen, wie in Abbildung 51 zu erkennen. Ist dies der Fall kann der Benutzer eine Taste drücken, deren Signal mit dem der Steuerungseinheiten und des Sicherheitsschaltgerätes über mehrere UND-Gatter verknüpft wird. Erst wenn alle vier Signale aktiv sind, und es auch während des ganzen Scans durchgängig bleiben, erzeugt die Röntgenröhre Strahlung. Eine kurzzeitige Unterbrechung des Tastendrucks führt sofort zu einem Abbruch der Bildaufnahme (Totmannschalter). Die zur Realisierung verwendeten UND-Gatter befinden sich jeweils in den FPGAs der beiden Steuerungseinheiten sowie der Kontrolleinheit der Röntgenröhre. Sie liegen somit in Logik vor, was für eine Sicherheitsfunktion dieser Art bei einer Zertifizierung von großem Vorteil ist, wie bereits in Abschnitt 4.1.2.1 beschrieben. [31; 75; 98]



**Abbildung 51: Freigabe der Röntgenstrahlung durch Logikgatter:** Sie befinden sich auf den FPGAs der Steuerungseinheiten sowie der Röntgenröhrenkontrolleinheit.

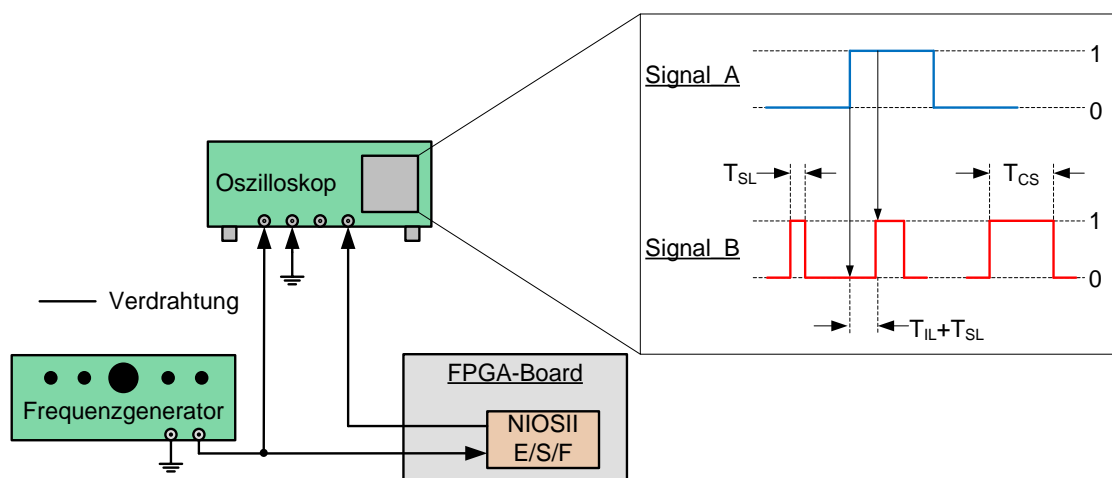
## 4.3 Messung der Leistung ausgewählter Echtzeitbetriebssysteme

### 4.3.1 Einleitung

Die in Abschnitt 4.1.2.2 vorgestellten RTOS, welche sich für den NIOSII-Prozessor eignen, sind anhand ihrer Eigenschaften laut Datenblatt, nicht jedoch anhand ihrer tatsächlichen Leistungsfähigkeit auf dem entsprechenden NIOSII-Zielprozessor vergleichbar, da entsprechende Veröffentlichungen zum Zeitpunkt dieser Arbeit nicht vorhanden bzw. bekannt waren. Es existieren zwar Veröffentlichungen, die Ergebnisse von Leistungsmessungen dieser RTOS beschreiben, allerdings auf anderen Zielprozessoren. [162; 175; 220] Ebenso sind Veröffentlichungen zu finden, die die Leistung von anderen RTOS als die hier aufgeführten auf dem NIOSII-Prozessor vergleichen. [165] Aus diesem Grund wurde in den folgenden Messungen die Interrupt-Latenz, „Interrupt-Latency“  $T_{IL}$ , sowie die „Context-Switching-Time“  $T_{CS}$  von verschiedenen, für den NIOSII-Prozessor verfügbaren RTOS bestimmt, um Rückschlüsse auf ihre Leistung ziehen zu können.

### 4.3.2 Material und Methoden

Als RTOS wurden embOS, FreeRTOS,  $\mu\text{C}/\text{OSII}$  und ThreadX ausgewählt, da für diese RTOS für den NIOSII-Prozessor angepasste Demoverionen mit entsprechenden BSPs verfügbar waren. Die Eigenschaften dieser RTOS sind bereits in Kapitel 4.1.2.2 hinreichend beschrieben. Sie wurden einzeln auf ein DBC4CE55 FPGA-Board (Steuerungseinheit) installiert, auf dem ein NIOSII-Prozessor für jede Messung jeweils in der Leistungsklasse „Economy“, „Standard“ und „Fast“, instanziiert war. Der Versuchsaufbau ist in Abbildung 52 gezeigt.



**Abbildung 52: Versuchsaufbau zur Messung der Leistungsfähigkeit eines RTOS:** Gemessen wurde die „Interrupt Latency“ und die „Context Switching Time“.

Die RTOS wurden entsprechend ihrer Datenblätter auf ihre, wenn vorhanden, jeweils höchste Leistungsstufe parametrisiert und anschließend jeweils die Zeit  $T_{IL}$  und  $T_{CS}$  durch Messung bestimmt. Beide Zeiten sind wesentliche Merkmale, um die Leistung eines RTOS zu bestimmen. [134; 213] Sie wurden bereits in zahlreichen Veröffentlichungen als solche herangezogen. [162; 175; 178; 220] Für die Messungen wurden entsprechende Testprogramme geschrieben und mit Optimierungsstufe O2 auf der GNU Compiler Kollektion (GCC) kompiliert.

Die „Interrupt-Latency“  $T_{IL}$  ist die Zeit, die ein RTOS benötigt, um auf ein plötzlich auftretendes externes Ereignis (Interrupt) zu reagieren.  $T_{IL}$  wird messbar durch Anlegen eines Taktsignals, „Signal\_A“, an einen als Interrupt definierten Eingang des Boards. Beim Erkennen des Signalwechsels schaltet das RTOS via „Interrupt Service Routine“ einen entsprechenden Ausgang, „Signal\_B“, auf logisch „1“. Beide Signale wurden wie in Abbildung 52 gezeigt mit einem Oszilloskop 1000-mal aufgezeichnet und der Mittelwert ihrer Differenz gemessen, während auf dem RTOS zwei Tasks abliefen, die durch Berechnungsroutinen Prozessorlast simulierten. Die Zeit, die zwischen dem Auftreten der Flanke von „Signal\_A“ und „Signal\_B“ vergeht, setzt sich aus  $T_{IL}$  und einer systembedingten Signallaufzeit  $T_{SL}$  zusammen. Zur Bestimmung von  $T_{SL}$  wurde vor jeder Messung der entsprechende Ausgang des Boards einmal ein- und sofort wieder ausgeschaltet. Um Unterbrechungen durch Interrupts auszuschließen wurden diese während der Messung von  $T_{SL}$  deaktiviert. Die Zeit  $T_{SL}$  wurde durch mathematische Operation direkt am Oszilloskop von der gemessenen Zeit  $T_{SL} + T_{IL}$  abgezogen, woraus sich  $T_{IL}$  ergab. [175; 213]

Als „Context-Switching-Time“  $T_{CS}$  hingegen wird die Zeit verstanden, die ein RTOS benötigt, um einen Task zu unterbrechen und einen anderen Task fortzuführen. Um  $T_{CS}$  zu messen, wurde für jedes RTOS ein Testprogramm geschrieben, welches über die Funktionen des Betriebssystems zwei Tasks erzeugt. Ein Task, „Task\_1“, setzt hierbei einen Ausgang des Boards auf logisch „1“ und gibt im Anschluss den Prozessor an einen zweiten Task, „Task\_2“, ab. „Task\_2“ schaltet den von „Task\_1“ gesetzten Ausgang wieder auf logisch „0“ und gibt den Prozessor wieder frei, siehe hierzu auch Abbildung 52. Um die Signallaufzeit zu messen, schaltet bei jeder Messung ein Task den Ausgang ein und sofort wieder aus. Das Ausgangssignal wurde ebenfalls mit einem Oszilloskop 1000-mal aufgezeichnet und der Mittelwert aus den gemessenen Zeiten gebildet, woraus sich die Zeit  $T_{CS}$  ergab. [175; 213]

### 4.3.3 Ergebnisse

Die Ergebnisse der Messung von  $T_{IL}$  sind in Abbildung 53 veranschaulicht. Als Referenz gilt die Messung von  $T_{IL}$  ohne RTOS. Der Einsatz von embOS zeigte keine Verlängerung von  $T_{IL}$  im Vergleich zum Verzicht auf ein RTOS, was als sehr gut zu bewerten ist und auch durch den Hersteller so angegeben wird. [208] Die Zeiten  $T_{IL}$  von FreeRTOS und  $\mu C/OSII$  waren nur wenige  $\mu s$  länger als die von embOS. ThreadX hingegen benötigte zur Verarbeitung eines Interrupts mehr als doppelt so lange wie embOS.

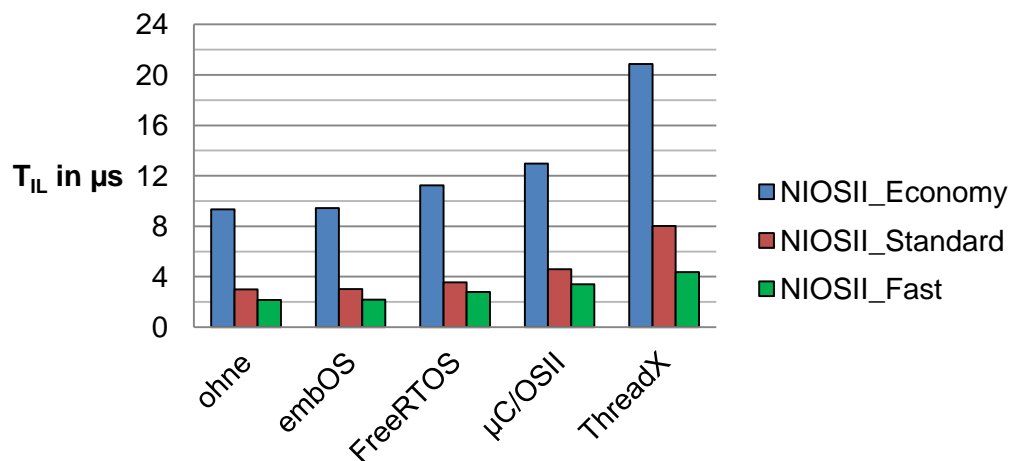


Abbildung 53: Messung der „Interrupt-Latency“ mit Codeoptimierungsstufe O2.

Bei der Messung der „Context-Switching-Time“  $T_{CS}$ , Abbildung 54, zeichnete sich ThreadX als schnellstes RTOS aus, gefolgt von embOS, FreeRTOS und  $\mu C/OSII$ . FreeRTOS benötigte für einen „Context-Switch“ ungefähr die vierfache Zeit der restlichen RTOS.

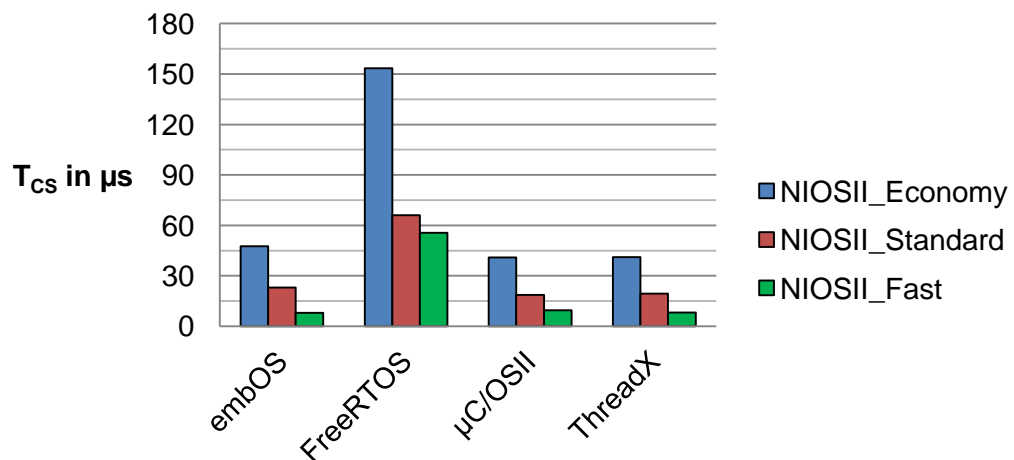


Abbildung 54: Messung der „Context-Switching-Time“ mit Codeoptimierungsstufe O2.

#### 4.3.4 Diskussion

Mithilfe der Messungen aus 4.3.3 wurden vier RTOS für den NIOSII-Prozessor anhand der Zeiten  $T_{IL}$  und  $T_{CS}$  verglichen. embOS zeigte in beiden Messungen die höchste Leistung, verfügt aber laut Tabelle 4 aus Kapitel 3.4.1 nicht über die nötigen Zertifikate für einen Einsatz in sicherheitskritischen Systemen oder Medizingeräten.  $\mu C/OSII$  und ThreadX besitzen entsprechende Zertifikate, ihre Verwendung ist allerdings mit hohen Kosten verbunden. FreeRTOS benötigte relativ lange Zeiten  $T_{CS}$  für einen „Context Switch“. Diese sind aber, bezogen auf eine Verwendung von FreeRTOS im Brust-CT-Scanner, noch akzeptabel. Allerdings verfügt FreeRTOS, wie embOS, nicht über entsprechende Zertifikate für den Einsatz in Medizingeräten. Der große Vorteil dieses RTOS besteht aber darin, dass es sich um ein vollwertiges und zugleich kostenfreies RTOS handelt, das bei Bedarf, wenn eine Zertifizierung des Brust-CT-Scanners als Medizinprodukt durchgeführt werden soll, durch SafeRTOS austauschbar ist. SafeRTOS verfügt über alle hierfür nötigen Zertifikate und ist mit vergleichbaren Kosten verbunden, wie andere zertifizierte RTOS. Diese Kosten müssten bei FreeRTOS allerdings nicht zu Beginn der Entwicklung aufgebracht werden, sondern erst wenn der Brust-CT-Scanner realisiert ist und er seine Zuverlässigkeit, mit dem entsprechenden RTOS, in ausführlichen Tests bewiesen hat. Aus diesen Gründen wird die Verwendung eines vollwertigen und kostenlosen RTOS, wie FreeRTOS, für den Brust-CT-Scanner empfohlen, wenn sich dieses leicht durch ein kommerzielles, zertifiziertes RTOS, wie SafeRTOS, austauschen lässt.

### 4.4 Versuche zum Datendurchsatz von 10-GBit/s-Ethernet

#### 4.4.1 Einleitung

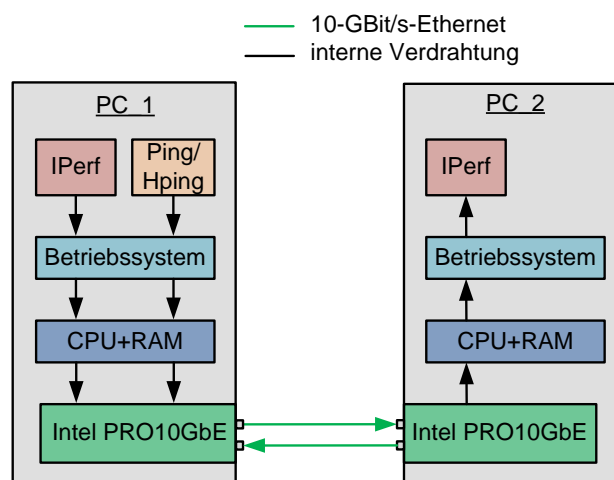
Für die Übertragung der Projektionsdaten des Brust-CT-Scanners sind Datentransferraten von mindestens 7 GBit/s gefordert. Durch den Einsatz von 10 GBit/s schnellem Ethernet soll dieser Forderung nachgekommen werden. Laut den Herstellerangaben von 10-GBit/s-Ethernet-Komponenten sind Datenraten von bis zu 10 GBit/s zwar möglich, allerdings existieren zahlreiche Veröffentlichungen, die den Datendurchsatz solcher Komponenten wesentlich niedriger einstufen. In diesen Veröffentlichungen wurden Versuche mit 10-GBit/s-Ethernet-Übertragungstrecken unter Einsatz von Standard-Komponenten durchgeführt und der angegebene Datendurchsatz von 10 GBit/s nur teilweise erreicht, wobei sich die Ergebnisse stark unterscheiden. So erreichten Pope et al. maximal 6,8 GBit/s [186], Hughes-Jones et al. 5,7 GBit/s [105] und Hurwitz et al. 4 GBit/s [109]. Die folgenden Messungen sollen zeigen, ähnlich wie die Versuche aus früheren Veröffentlichungen, welcher Datendurchsatz mit 10-GBit/s-Standard-Ethernet-Komponenten tatsächlich erreichbar ist und ob damit die für den Brust-CT-Scanner geforderte Datenrate von 7 GBit/s erreicht werden kann. Neben dem Datendurchsatz

wurde auch die Paketumlaufzeit gemessen. Sie bezeichnet die Zeit, die ein Ethernet-Paket benötigt, wenn es von einer Station zu einer anderen gesendet wird und von dieser wieder zurück zur Ausgangsstation. Sie ist ebenfalls ein Indiz für die Güte einer Ethernet-Verbindung. [72; 73; 234]

#### 4.4.2 Material und Methoden

Als Messaufbau, siehe hierzu Abbildung 55, dienten zwei Standard-PCs mit einem Intel DP67BG Mainboard, Intel i7-2600 Prozessor und 8 GByte DDR3-SDRAM Hauptspeicher. Ferner waren beide PCs mit je einer 10-GBit/s-Ethernet-Netzwerkkarte des Typs „Intel PRO10GbE LR“ [118] bestückt und über einen Lichtwellenleiter miteinander verbunden. Als Betriebssystem kam Windows 7 Pro, sowie die Linux-Distribution Knoppix 6.7 zum Einsatz. Betriebssysteme und Netzwerkkarten verfügen über eine Reihe von einstellbaren Parametern, die Datendurchsatz und Paketumlaufzeit beeinflussen können. Sie wurden entweder den Herstellerempfehlungen entsprechend gesetzt oder auf einem Standardwert belassen, da sich in vorausgehenden Testmessungen kein Einfluss dieser Parameter auf die Ergebnisse zeigte.

Dies trifft allerdings nicht auf die Verwendung von sogenannten „TCP/IP Offload Engines“ (TOEs) zu. Hierbei übernimmt die Netzwerkkarte beim Entpacken eines TCP-Pakets einen Teil der ansonsten vom Prozessor des PCs ausgeführten Aufgaben – respektive das Berechnen der CRC – und entlastet diesen so. Die Messungen von Datendurchsatz und Paketumlaufzeit wurden deshalb nicht nur für jedes Betriebssystem, sondern jeweils auch mit ein- und ausgeschalteter TOE durchgeführt. Messungen mit und ohne Verwendung von TOE sind deshalb von Interesse, da das entwickelte Protokoll zur Datenübertragung der Projektionsdaten „CT\_Data\_Ethernet“ nicht auf dem TCP-Protokoll basiert und deshalb nicht abgeschätzt werden konnte, ob eine TOE zur Steigerung der Leistungsfähigkeit beitragen kann. [28; 72; 116; 117; 148]



**Abbildung 55: Versuchsaufbau zu Messungen einer 10-GBit/s-Ethernet-Strecke:** Er besteht aus zwei PCs, von denen jeder mit einer 10-GBit/s-Ethernet-Netzwerkkarte und der benötigten Testsoftware bestückt ist.

Gemessen wurde zum einen der Datendurchsatz in Abhängigkeit von der Paketgröße, der Puffergröße und der TCP-Fenstergröße. Mit TCP-Fenstergröße ist die Datenmenge gemeint, nach welcher ein Empfänger den Erhalt der Daten mit einem Paket explizit bestätigen muss. Als Testprogramm kam die Testsoftware IPerf zum Einsatz. Sie misst den Datendurchsatz von Netzwerkverbindungen, indem sie Datenströme auf der Seite des Senders generiert, über das TCP-Protokoll an einen Empfänger (ebenfalls mit einer IPerf-Version) sendet und dort auswertet. Als Datenmenge wurden jeweils 10 GByte übertragen, da vorausgehende Testmessungen keine Änderung des Datendurchsatzes bei größeren Datenmengen zeigten. [104; 226]

Zum anderen wurde die Paketumlaufzeit, mit den Tools HrPing unter Windows sowie HPing unter Knoppix, gemessen. Beide Programme senden ein Paket auf Basis des zu TCP/IP gehörigen „Internet Control Message Protocol“ (ICMP) und warten auf die Antwort der Gegenstelle. Die hierfür benötigte Zeit ist die Paketumlaufzeit. Um repräsentative Ergebnisse erzielen zu können, wurde für jede Messung die Paketumlaufzeit 1000-mal gemessen und der Mittelwert der Ergebnisse gebildet.

#### 4.4.3 Ergebnisse

Erwartungsgemäß korreliert der Datendurchsatz mit der Paketgröße, der Puffergröße und der TCP-Fenstergröße. Besonders deutlich ist die Abhängigkeit von der Paketgröße in Abbildung 56 zu erkennen. Mit zunehmender Paketgröße steigt der Datendurchsatz schnell an. Bei Paketgrößen, die über 1500 Byte liegen, speziell durch den Einsatz von „Jumbo Frames“, lassen sich Datenraten von fast 10 GBit/s (9,9 GBit/s) erreichen. Durch den Einsatz von TOE erhöht sich die Datenrate nur in geringem Maß. Ein gravierender Unterschied zwischen den Betriebssystemen wurde nicht festgestellt.

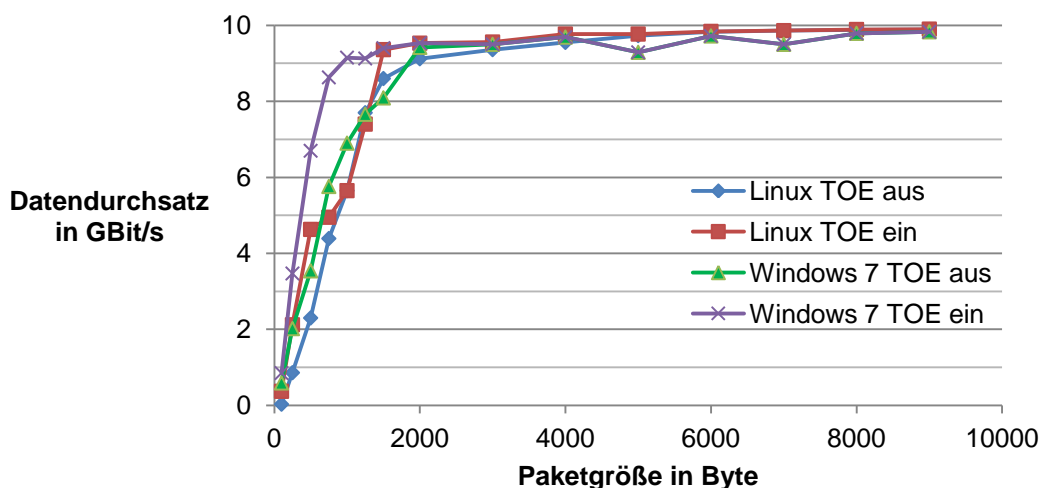


Abbildung 56: Messung des Datendurchsatzes in Abhängigkeit der Paketgröße.

Außerdem gemessen wurde der Datendurchsatz in Abhängigkeit der Puffergröße, siehe hierzu Abbildung 57. Die Puffergröße bezeichnet die Größe, welche das Betriebssystem für eingehende Ethernet-Pakete vorhält. Ab einer Puffergröße von ca. 300 KByte zeigt sich entsprechend Abbildung 57 kein Einfluss mehr auf den Datendurchsatz. Die Puffergröße ist unter Knoppix aufgrund von nicht ausreichender Treiberfunktionalität nicht veränderbar und wurde deshalb nicht gemessen. [116]

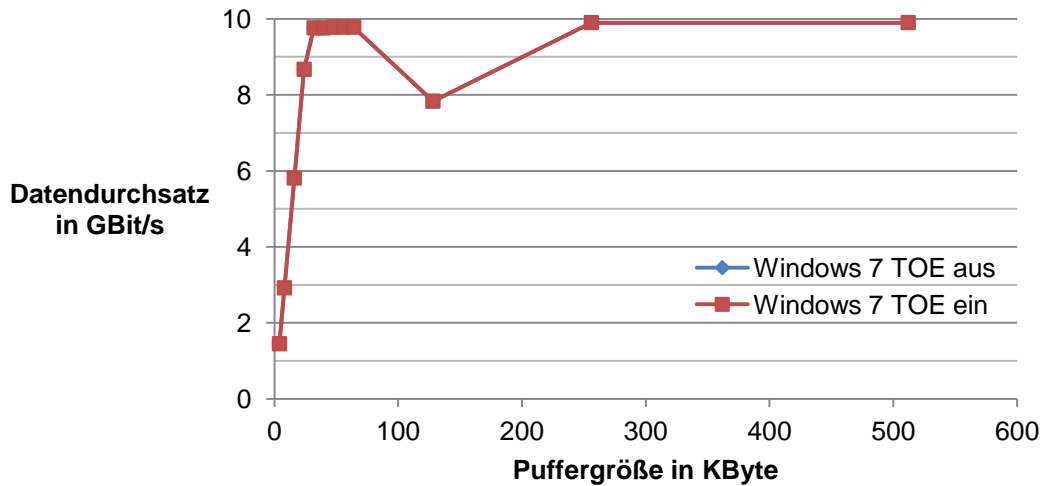


Abbildung 57: Messung des Datendurchsatz in Abhängigkeit der Puffergröße.

Ähnlich verhält es sich bei der Messung des Datendurchsatzes in Abhängigkeit von der TCP-Fenstergröße. Sie wurde ebenfalls nur für Windows 7 gemessen, da die vorliegenden Treiber der Linux Distribution ein Verändern der TCP-Fenstergröße nicht zuließen. Bei einer Puffergröße von 500 KByte blieb der Datendurchsatz von ca. 10 GBit/s konstant, wie Abbildung 58 zeigt. [116]

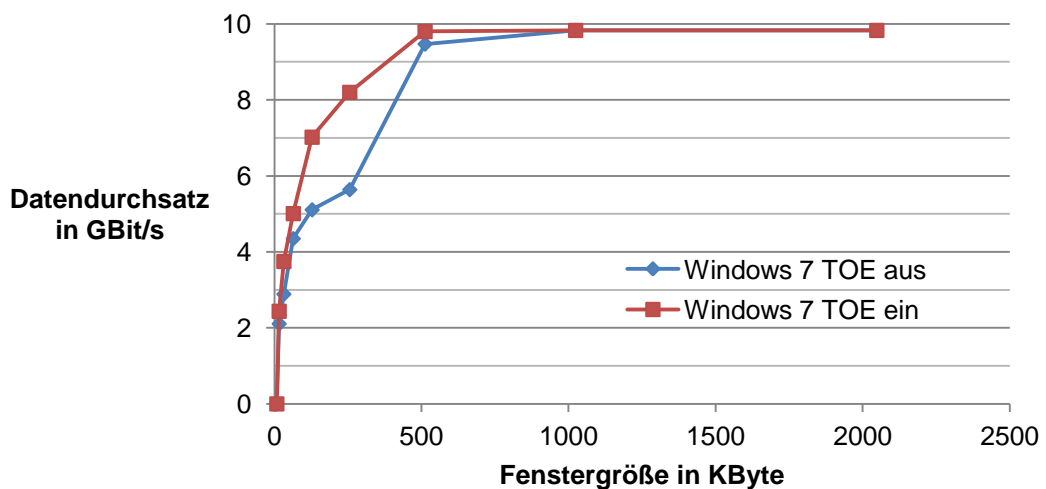


Abbildung 58: Messung des Datendurchsatz in Abhängigkeit der Fenstergröße.

Bei den Messungen der Paketumlaufzeit, siehe Abbildung 59, zeigen sich bessere Ergebnisse bei Einsatz der Linux Distribution, die eine deutlich geringere Paketumlaufzeit aufweist. Durch die höhere Paketgröße ergeben sich bei Verwendung von 9000 Byte großen „Jumbo Frames“ entsprechend längere Paketumlaufzeiten. Die Ergebnisse aus Abbildung 59 entsprechen weitestgehend den Ergebnissen von Pope et al. [186]. Durch die ähnlich hohen Paketumlaufzeiten wie in [186], kann auf eine ähnlich hohe Güte der 10-GBit/s-Ethernet-Übertragungsstrecken rückgeschlossen werden. Die aufgebauten Übertragungsstrecken sind somit von vergleichbarer Qualität.

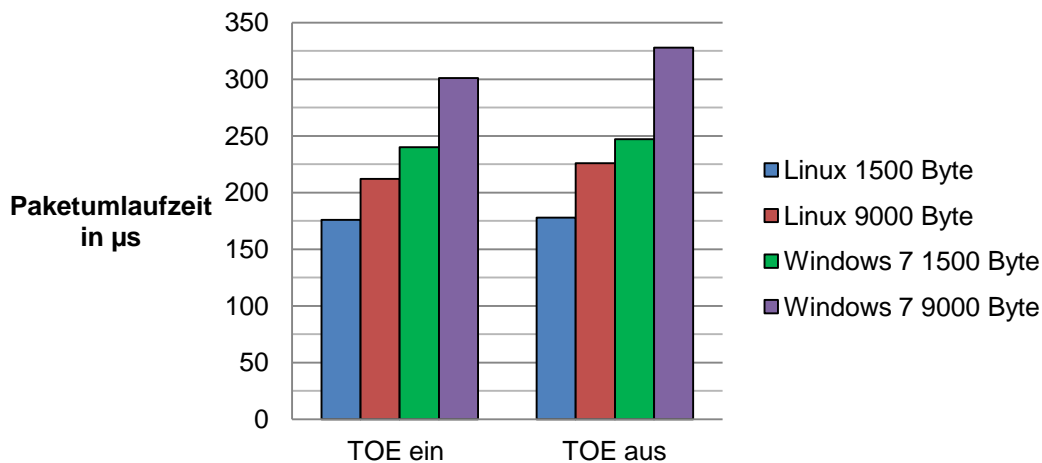


Abbildung 59: Messung der Paketumlaufzeit in Abhängigkeit der Paketgröße.

#### 4.4.4 Diskussion

Die Ergebnisse der Messungen lassen erkennen, dass sich eine 10-GBit/s-Ethernet-Datenübertragungsstrecke mithilfe von Standard-Komponenten aufbauen lässt, die durch den Einsatz von „Jumbo Frames“, fähig ist, Daten mit bis zu 9,9 GBit/s zu übertragen. Hierbei zeigten sich zwischen Linux und Windows 7 keine großen Unterschiede. Allerdings waren bestimmte Parameter nicht in jedem Betriebssystem veränderbar, bzw. eine Veränderung zeigte aufgrund unzureichender Treiberfunktion keine Auswirkung.

Die Ergebnisse stehen in Diskrepanz zu den Veröffentlichungen von Pope et al. [186], Hughes-Jones et al. [105] und Hurwitz et al. [109], welche ähnliche Netzwerkkarten verwendeten, aber von einem geringeren Datendurchsatz berichteten. Die niedrigeren Datendurchsatzraten aus früheren Veröffentlichungen sind dadurch erklärbar, dass sich die Technologie bezüglich der für den Datendurchsatz entscheidenden Komponenten stark weiterentwickelt hat. Dies trifft besonders auf den Datendurchsatz von Prozessoren und Hauptspeichern zu, deren Taktraten nun deutlich höher liegen als noch vor einigen Jahren. Auf die Paketumlaufzeit wirkt sich die Leistung dieser Komponenten weniger bis gar nicht aus, da die Verarbeitung einzelner Pakete keine Belastung für den Prozessor bzw. Hauptspeicher darstellt. [156]

## 4.5 Messungen der Signallaufzeiten des Synchronisationssignals

### 4.5.1 Einleitung

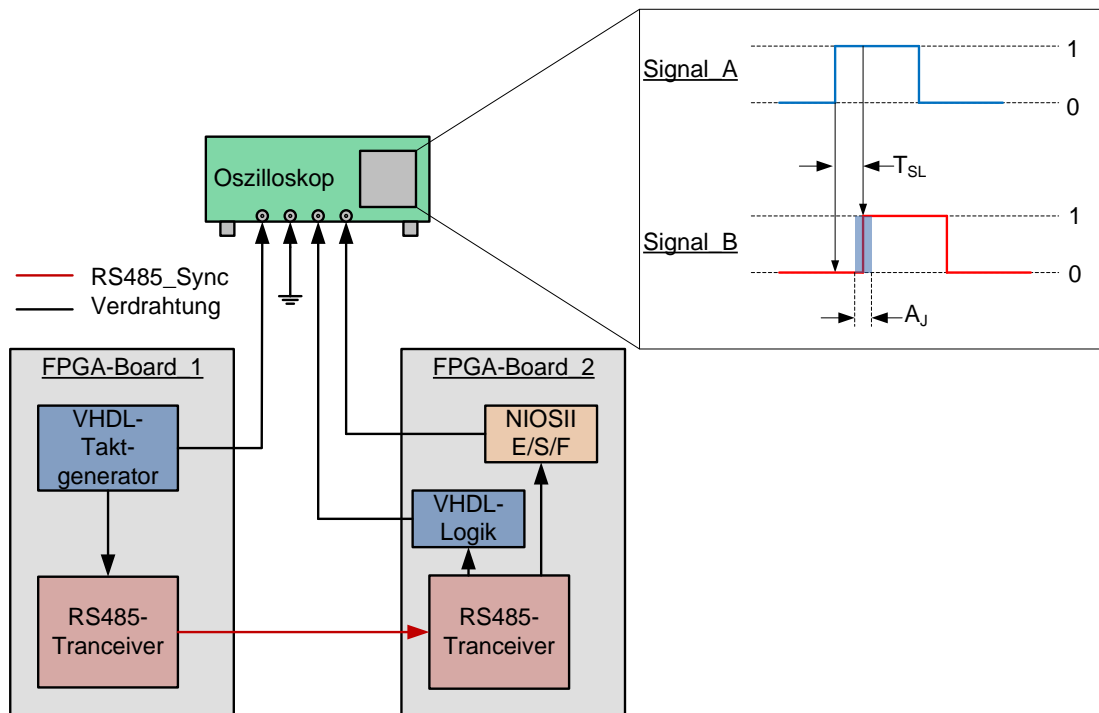
Die statische Steuerungseinheit generiert ein initiales Synchronisationssignal und sendet es der rotierenden Steuerungseinheit, die daraus weitere Synchronisationssignale erzeugt. Hierbei auftretende Signallaufzeiten können ausgeglichen werden, indem die statische Steuerungseinheit das Synchronisationssignal um den Betrag der Signallaufzeit früher versendet. Laut Spezifikation dürfen diese Signallaufzeiten allerdings nicht mehr als 10  $\mu\text{s}$  betragen, vergleiche hierzu Abschnitt 3.5.8. Um die Bildqualität nicht negativ zu beeinflussen, ist für den Brust-CT-Scanner außerdem ein Jitter der Synchronisationssignale von maximal 1  $\mu\text{s}$  spezifiziert, da dieser nicht ohne Weiteres ausgeglichen werden kann. Die folgende Messung bestimmt die Signallaufzeit sowie deren Jitter, die beim Senden und Empfangen von RS485-Signalen zwischen den verwendeten Steuerungseinheiten auftreten. Da diese Synchronisationssignale mit oder ohne NIOSII-Prozessor von den Steuerungseinheiten verarbeitet werden können, wurden beide Möglichkeiten berücksichtigt. Bei Verarbeitung mittels eines NIOSII-Prozessors erfolgten die Messungen unter Verwendung aller drei Versionen (Economy/Standard/Fast) des NIOSII-Prozessors.

### 4.5.2 Material und Methoden

Als Sende- und Empfangssystem dient jeweils ein FPGA-Development-Board (DBC4CE55), das auch im Brust-CT-Scanner für die Realisierung der Steuerungseinheiten eingesetzt wird. Beide Systeme verfügen somit über den gleichen RS485-Transceiver, einen Texas Instruments SN75HVD11D mit einer Übertragungsrate von 10 MBit/s. [42; 222]

Auf dem Sendesystem wurde über einen Taktgenerator in VHDL ein Synchronisationssignal fester Frequenz erzeugt, hier mir „Signal\_A“ bezeichnet. „Signal\_A“ wurde über den RS485-Transceiver des Sendesystems an den RS485-Transceiver des Empfangssystems übertragen, siehe hierzu Abbildung 60. In einer Messung leitete das Empfangssystem das empfangene Signal durch ein Logikgatter (Inverter) in VHDL direkt an einen Ausgangspin des FPGAs weiter, wodurch sich ein Ausgangssignal „Signal\_B“ ergab. In weiteren Messungen war ein NIOSII Softcore-Prozessor auf dem FPGA instanziiert, auf dem durch das empfangene Signal ein Interrupt erzeugt wurde. Dieser Interrupt schaltete daraufhin einen Ausgangspin, woraus das Signal „Signal\_B“ entstand. Auf dem Prozessor lief während der Messung ein Testprogramm ab, um Prozessorlast zu simulieren. Beide Synchronisationssignale wurden 10.000-mal mit einem Oszilloskop aufgezeichnet und die sich daraus ergebende Signallaufzeit  $T_{SL}$  bestimmt. Die Summe aus maximal negativer und maximal positiver Abweichung vom Mittelwert von  $T_{SL}$  ergibt die

Amplitude des Jitters  $A_J$ . Die Messungen wurden für alle drei Leistungsstufen des NIOSII-Prozessors durchgeführt, wobei die eingestellte Codeoptimierungsstufe O2, der FPGA-Takt 50 MHz und der Prozessortakt 75 MHz betragen. Zur Kompilierung wurde GCC verwendet. Um Signallaufzeiten auf den Leitungen zwischen den FPGA-Boards vernachlässigbar gering zu halten, betrug die Leitungslänge lediglich 0,2 m.



**Abbildung 60: Aufbau zur Bestimmung von Signallaufzeit und Jitter:** Die Messung erfolgte mithilfe eines Oszilloskops an einem Synchronisationssignal, das zwischen zwei FPGA-Boards (Steuerungseinheiten) über RS485 ausgetauscht wurde.

### 4.5.3 Ergebnisse

**Tabelle 5: Signallaufzeiten des Synchronisationssignals mit und ohne NIOSII-Prozessor.**

Empfänger	Mittelwert von $T_{SL}$	Jitter-Amplitude $A_J$	Jitter in % von $T_{SL}$
NIOSII/E	9810 ns	1577 ns	16,1 %
NIOSII/S	3155 ns	320 ns	10,1 %
NIOSII/F	2350 ns	534 ns	22,7 %
VHDL-Logik	85 ns	3 ns	3,5 %

Die Ergebnisse der Messung sind in Tabelle 5 aufgeführt. Deutlich erkennbar sind die hohen Signallaufzeiten bei Verwendung eines NIOSII-Prozessors. Je leistungsfähiger der Prozessor gewählt war, desto kürzere Laufzeiten wurden erreicht. Die Signallaufzeiten  $T_{SL}$  lagen zwischen 9810 ns bei einem NIOSII „Economy“ und 2350 ns bei einem NIOSII-Prozessor der Version „Fast“. Der gemessene Jitter betrug minimal 10,1% bei der Version „Standard“ und 22,6% bei der Version „Fast“.

Ohne Verwendung eines NIOSII-Prozessors, bei direkter Verarbeitung im FPGA, ergibt sich eine Signallaufzeit von 85 ns. Dies ist eine Reduktion der Laufzeit um einen Faktor von 115 im Vergleich zu einer Laufzeit unter Verwendung eines NIOSII-Prozessors der Version „Economy“. Der hierbei zu messende Jitter von nur 3 ns ist mit 3,5% bis um einen Faktor von sechs, bezogen auf die jeweiligen Signallaufzeiten, kleiner als der Jitter bei Verwendung eines NIOSII-Prozessors der Version „Fast“.

#### 4.5.4 Diskussion

Die relativ hohen Signallaufzeiten bei Verwendung eines NIOSII-Prozessors sind durch die Latenz erklärbar, die der Prozessor benötigt, um einen Interrupt zu erkennen und abzuarbeiten (im englischen als „Interrupt-Latency“ bezeichnet). Diese Zeit ist abhängig von verschiedenen Faktoren, wie z.B. der Taktrate des Prozessors sowie der Zeit, die dieser benötigt um die aktuelle Aufgabe zu unterbrechen und abzuspeichern (im englischen als „Context-Save-Time“ bezeichnet). [5] Für die Signallaufzeit ohne Prozessor ergaben sich Signallaufzeiten, die mit 85 ns deutlich unter dem Erwartungswert von 110 ns lagen. Der Erwartungswert ergab sich aus dem Datenblatt der verwendeten RS485-Transceiver, deren typische Durchlaufzeit mit je 55 ns (30 bis 70 ns) angegeben war. Zusätzlich zu den Laufzeiten in den Transceivern kommen noch Laufzeiten auf den Leitungen und in den FPGAs hinzu, die mit insgesamt 20 ns angenommen wurden. Besonders positiv zu bewerten ist weiterhin der geringe Jitter von nur 3 ns bzw. 3,51%. [222]

Auf den Brust-CT-Scanner bezogen bedeuten die Ergebnisse, dass ein NIOSII-Prozessor für die Signalverarbeitung des Synchronisationssignals nicht verwendet werden sollte, da die hierbei auftretenden Signallaufzeiten zu nahe an dem maximal spezifizierten Grenzwert von 10  $\mu$ s liegen. Gleiches gilt für den auftretenden Jitter, der, unter Verwendung eines NIOSII/E-Prozessors zur Signalverarbeitung, den maximal spezifizierten Grenzwert von 1  $\mu$ s sogar überschritten hat. Hierdurch wäre die Bildqualität des Brust-CT-Scanners beeinträchtigt, besonders bei Verwendung von TDS und FSC.

Die Ergebnisse zeigen aber auch, dass eine Steuerungseinheit auf Basis eines SOPCs (mit FPGA) Synchronisationssignale zur Verfügung stellen kann, deren Signallaufzeiten mit 85 ns und Jitter mit 3 ns deutlich unter den geforderten Grenzwerten von 10  $\mu$ s bzw. 1  $\mu$ s liegen. Aufgrund dieser positiven Ergebnisse erfolgt die Positionserfassung sowie die Bereitstellung und Verarbeitung von Synchronisationssignalen im Brust-CT-Scanner in VHDL auf FPGA-Basis. Diese FPGAs sind Kern der als SOPCs ausgeführten Steuerungseinheiten.



## 5 Zusammenfassung und Optimierungspotential

### 5.1 Zusammenfassung der Ergebnisse

Anders als bisherige Ansätze zur Entwicklung eines Brust-CT-Scanners basiert der von IMP und CTI geplante dedizierte Brust-CT-Scanner auf dem Prinzip der Spiral-CT und verwendet neuartige Bildgebungskomponenten. Dem Stand der Technik entsprechende Steuerungskonzepte sind unter anderem aus diesen und anderen Gründen (Verwendung eines Drehübertrags, Geheimhaltung) nicht zur Steuerung dieses Scanners geeignet, weshalb ein neuartiges Steuerungskonzept benötigt wurde. Die Erstellung eines solchen Konzepts stellte das Ziel dieser Arbeit dar. Hiervon ausgehend wurde ein neuartiges, hierarchisches Steuerungskonzept für den dedizierten Brust-CT-Scanner entwickelt, das aus vier Ebenen besteht und über entsprechende Funktionalität zur Synchronisation der Bildgebungskomponenten verfügt.

Durch die hierarchische Aufteilung ergibt sich für die Kommunikation eine Reduzierung der Datenmenge, sowie der (Echt-)Zeitanforderung in Richtung höherer Ebenen. Dies ist notwendig, da für die Datenübertragung zwischen statischer und rotierender Seite des Scanners ein Drehübertrager verwendet wird, der eine begrenzte Anzahl von Datenübertragungskanälen sowie eine höhere Bitfehlerrate, als direkte Verbindungen, besitzt. Zur Kommunikation zwischen den Ebenen dienen die eigens hierfür entwickelten deterministischen Kommunikationsprotokolle „CT\_Control\_CAN“ auf Basis von CAN sowie „CT\_Control\_Ethernet“ auf Basis von Ethernet. Durch die Verwendung von bekannten und offenen Schnittstellen, lizenzfreier Technologien sowie der hierarchischen Architektur, ist das Konzept nicht nur für den Brust-CT-Scanner, sondern auch für andere Geräte aus der Medizintechnik einsetzbar.

Einen weiteren wichtigen Teil des Steuerungskonzepts stellt die Synchronisation der Bildgebungskomponenten dar. Um höchste Bildqualität zu gewährleisten, muss die Erfassung der Gantrypositionswerte synchron mit der Projektionsaufnahme durch die Röntgenröhre und den Detektor erfolgen. Dies trifft besonders bei Verwendung der Technologien TDS und FSC zu. Bei Verwendung dieser Technologien müssen den Bildgebungskomponenten hochgenaue Synchronisationssignale mit Signallaufzeiten kleiner als 10  $\mu\text{s}$  und einem Jitter kleiner als 1  $\mu\text{s}$  zugeführt werden.

Die hierzu nötige Synchronisationsfunktionalität wurde den Steuerungseinheiten der Steuerungsebene des hierarchischen Steuerungskonzepts hinzugefügt. Hierzu verfügt die statische Steuerungseinheit über eine spezielle Positionserfassung und startet zeitbasiert eine Projektion, indem sie ein Synchronisationssignal für die rotierende Steuerungseinheit erzeugt.

Die rotierende Steuerungseinheit generiert daraus die spezifischen Synchronisationssignale für die Bildgebungskomponenten, die sie in einer bestimmten Abfolge versendet. Die Abfolge ist so gewählt, dass sich in den Bildgebungskomponenten vorhandene, technologisch bedingte Totzeiten überlagern und hierdurch die Dosiseffizienz des Brust-CT-Scanners gesteigert wird. Die von den Steuerungseinheiten erzeugten Synchronisationssignale weisen eine Signallaufzeit von 85 ns bei einem Jitter von 3 ns auf, und liegen damit deutlich unter den spezifizierten Grenzwerten von 10  $\mu$ s bzw. 1  $\mu$ s.

Bei der Realisierung des Steuerungskonzepts kommt besonders den Steuerungseinheiten der Steuerungsebene zentrale Bedeutung zu. Sie sind das Bindeglied zwischen zugelieferten Komponenten der Kontrollebene und des Leitrechners auf der Leitebene. Außerdem übernehmen sie die Aufgabe der Synchronisation der Bildgebungskomponenten und die der Positionserfassung, was eine zentrale Auswirkung auf die Bildqualität und die Dosiseffizienz des gesamten Brust-CT-Scanners hat. Um diese Aufgaben erfüllen zu können, wurden für die Realisierung der Steuerungseinheiten FPGA-Boards auf Basis von SOPCs gewählt, da diese Systeme ein Höchstmaß an Flexibilität aufweisen und sich mit ihnen schnell auf Änderungen, wie sie bei Erstellung eines neuartigen medizinischen Geräts üblich sind, reagieren lässt. Auf den FPGAs der Systeme kommen Softcore-Prozessoren der Architektur NIOSII zum Einsatz, von denen drei verschiedenen leistungsfähige Versionen entsprechend den Anforderungen wählbar sind. Auf der NIOSII-Prozessorarchitektur können eine Reihe verschiedener RTOS betrieben werden. Da die Leistung dieser Betriebssysteme abhängig vom eingesetzten Prozessor ist und nicht immer den Herstellerangaben entnommen werden kann, wurden hierzu Messungen durchgeführt. Basierend auf den Ergebnissen dieser Messungen und aus den Eigenschaften laut Datenblatt, wurde für den Brust-CT-Scanner das hierfür am besten geeignete RTOS, FreeRTOS für die Entwicklung bzw. SafeRTOS für den zu zertifizierenden Scanner, ausgewählt.

Um die beim Brust-CT-Scanner auf der rotierenden Seite anfallenden Projektionsdaten auf die statische Seite übertragen zu können, bestand außerdem der Bedarf einer hochperformanten Datenübertragung. Diese sollte die Projektionsdaten mit mindestens 7 GBit/s über den eingesetzten Drehübertrager übertragen. Um diese Forderung zu erfüllen, wurde eine 10-GBit/s-Ethernet-Datenstrecke gewählt, die mit Standard-Komponenten realisierbar ist und mit der in Versuchen Datenübertragungsraten von bis zu 9,9 GBit/s nachgewiesen werden konnten. Als höhere Schicht, welche auf Ethernet aufsetzt, kommt ein speziell für die Übertragung der Projektionsdaten des Brust-CT-Scanners entwickeltes Protokoll „CT\_Data\_Ethernet“ zum Einsatz. Es weist durch die Verwendung von „Jumbo Frames“ sowie aufgrund des speziellen Aufbaus wenig Protokoll-overhead und eine hohe Leistungsfähigkeit auf.

Ein weiteres Ziel des Promotionsprojekts war die Sicherstellung eines gefahrlosen Betriebs des CT-Scanners. Um diesen zu ermöglichen verfügt das Steuerungskonzept des Brust-CT-Scanners über ein Sicherheitskonzept, welches drei verschiedene Eskalationsstufen zur Meldung irregulärer Zustände kennt. Dies sind die Stufen „Warnung“, „Fehler“ und „fatale Fehler“. Das Auftreten eines besonders kritischen irregulären Zustands des Scanners löst hierbei die Meldung eines „fatalen Fehlers“ und Unterbrechung eines Interlockkreises aus. Hierauf wird mit der Herstellung eines sicheren Zustands reagiert. Eine zusätzliche Absicherung in Logik schützt vor einem versehentlichen oder fehlerhaften Auslösen der Röntgenstrahlung und schaltet diese auch sicher wieder ab.

## 5.2 Optimierungspotential

Ein mögliches Optimierungspotential des Steuerungskonzepts betrifft die Realisierung der Steuerungseinheiten in der Steuerungsebene. Die vorgestellten FPGA-Development-Boards für die SOC's der Steuerungseinheiten stellen einen nicht unerheblichen Kostenfaktor dar. Durch Einsatz einer Prozessorarchitektur, die nicht als Soft-core-, sondern als „fixed Processor“-System vorliegt, wie z.B. einem ARM-Prozessor, könnten die Kosten erheblich gesenkt werden. Die zeit- und sicherheitskritischen VHDL-Codeblöcke könnten dann, anstatt auf einem FPGA, auf einem CPLD („Complex Programmable Logic Device“) ausgeführt werden, das mit dem „fixed Processor“ über eine definierte Schnittstelle kommuniziert. Bei einem CPLD handelt es sich um eine günstige Alternative zum FPGA. Es ist wie ein FPGA mit VHDL wiederbeschreibbar, allerdings weist es einen einfacheren Aufbau sowie eine deutlich geringere Anzahl von Logikblöcken auf. Die Verwendung eines „fixed Processor“-Systems in Kombination mit einem CPLD würde allerdings eine Einschränkung der Flexibilität bedeuten und ist nur sinnvoll, wenn die Funktionen und Schnittstellen des damit zu entwickelnden Scanners genau bekannt sind. Dies ist nur möglich, wenn das Konzept des Brust-CT-Scanners erfolgreich anhand eines Prototyps evaluiert und getestet wurde. Denn nur, wenn sich die Anforderungen an die Steuerung nicht mehr ändern, ist es sinnvoll, sich auf ein konkretes System zur Realisierung der Steuerungseinheiten festzulegen, das nur die benötigte Funktionalität enthält und kostenoptimiert wurde.



## Abkürzungsverzeichnis

μCT	Mikro-Computertomografie
ARM	Advanced RISC Machines
AS	Ablaufsprache
ASIC	Application-specific Integrated Circuit
AS-Interface	Actuator-Sensor-Interface
AWL	Anweisungsliste
BER	Bitfehlerrate
BSP	Board Support Package
CA	Collision Avoidance
CAN	Controller Area Network
CBA	Component Based Automation
CCD	Charge-coupled Device
CD	Collision Detection
CdTe	Cadmiumtellurid
CiA	CAN in Automation
CMOS	Complementary Metal Oxide Semiconductor
COTS	Commercial Off-The-Shelf
CPLD	Complex Programmable Logic Device
CRC	Cyclic Redundancy Check
CSMA	Carrier Sense Multiple Access
CT	Computertomografie
CTI	CT Imaging GmbH
CUP	Central Processing Unit
DAS	Data Acquisition System
DDR	Double Data Rate
DIN	Deutsches Institut für Normung
DP	Decentralized Peripherals

---

EIB	Europäischer Installationsbus
EN	Europäische Norm
EPA	Ethernet for Plant Automation
FDA	Food and Drug Administration
FFS	Flying Focal Spot
FIP	Flux Information Processus
FPGA	Field Programmable Gate Array
FSC	Flying Focal Spot Control
FSM	Fieldbus-Message-Specification
FTP	File Transfer Protocol
FUP	Funktionsplan
GCC	GNU Compiler Collection
GNU	GNU's Not Unix
GPP	General Purpose Processor
HDD	Hard Disk Drive
HTTP	Hypertext Transfer Protocol
IC	Integrated Circuit
ICE	Intercity-Express
ICMP	Internet Control Message Protocol
ID	Identifier
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IMP	Institut für Medizinische Physik
IP	Internet Protocol
IP-Core	Intellectual Property Core
ISO	International Organization for Standardization
KOP	Kontaktplan
LON	Local Operating Network
LP	Line Pairs
MAC	Media Access Control

---

MTF	Modulationsübertragungsfunktionen
OD	Object Dictionary
OS	Operating System
OSI	Open Systems Interconnection
OSI-Modell	Open Systems Interconnection 7-Schichtenmodell
PC	Personal Computer
PCP	Priority Code Point
RAM	Random Access Memory
RISC	Reduced Instruction Set Computer
RTOS	Real-time Operating System
SAS	Serial Attached Small Computer System Interface
SATA	Serial Advanced Technology Attachment
SDRAM	Synchronous Dynamic Random Access Memory
SERCOS	Serial Realtime COmmunication System
SerDes	Serializer/Deserializer
SFD	Start Frame Delimiter
SOPC	System on a Programmable Chip
SPS	Speicherprogrammierbare Steuerung
SRAM	Static random-access memory
SSD	Solid-State-Drive
ST	Strukturierter Text
TCP	Transmission Control Protocol
TDI	Time-delayed Integration
TDMA	Time Division Multiple Access
TDS	Time-delayed Summation
TOE	TCP/IP Offload Engine
TTP	Time-Triggered Protocol
TÜV	Technischer Überwachungsverein
UDP	User Datagram Protocol
VHDL	Very High Speed Integrated Circuit Hardware Description Language



## Literaturverzeichnis

1. Albert, A.: *Comparison of Event-Triggered and Time-Triggered Concepts with Regard to Distributed Control Systems*. In: *Proceedings of the Embedded World*, 2004. S. 235–252.
2. Altera Corporation (Hg.): *Altera Embedded Alliance: Embedded Operating Systems*. Online verfügbar unter <http://www.altera.com/devices/processor/embedded-alliance/proc-embed-partners.html#os>, zuletzt geprüft am 18.04.2013.
3. Altera Corporation (Hg.): *Design for Multiple Industrial Ethernet Protocols: Designing Industrial Ethernet with Softing and Altera*. Online verfügbar unter <http://www.altera.com/end-markets/industrial/automation/ethernet/protocols/ind-multiple-protocols.html>, zuletzt geprüft am 11.02.2013.
4. Altera Corporation (Hg.): *Developing Functional Safety Systems with TÜV-Qualified FPGAs*, 2010.
5. Altera Corporation (Hg.): *Nios II Processor Reference Handbook*. Version 11.0.0, 2011.
6. Altera Corporation (Hg.): *Nios II Software Developer Handbook*. Version 11.0, 2011.
7. Altera Corporation (Hg.): *Reducing Steps to Achieve Safety Certification*, 2011.
8. Altera Corporation (Hg.): *Support All Industrial Ethernet Standards: WP-01186*, November/2012.
9. Amin, A. F.; Aris, I.; Halin, I. B.; Abdullah, R. S.; Sahbudin, R. K.; Abdullah, M. K.: *Development of vehicle communication system using FPGA*. In: *Australian Journal of Basic and Applied Sciences*, Nr. 6, 2011. S. 1026–1038.
10. Atsushi, H.; Takeshi, M.; Michito, N.: *X-ray CT apparatus and control method for X-ray CT apparatus*. Veröffentlichungsnr: US020110222649, 2011.
11. Barry, R.: *FreeRTOS Features*. Hg. v. Real Time Engineers Ltd. Online verfügbar unter [http://www.freertos.org/FreeRTOS\\_Features.html](http://www.freertos.org/FreeRTOS_Features.html), zuletzt geprüft am 17.04.2013.
12. Bauer, M.; Edler, W.; Neubauer, G.; Popescu, S.: *Method and apparatus for error-tolerant data transfer for a CT system*. Veröffentlichungsnr: US7519208, 2009.
13. Bell, R.: *Introduction to IEC 61508*. In: *Proceedings of the 10th Australian workshop on Safety critical systems and software*, 2006. S. 3–12.
14. Ben Salem, A. K.; Ben Othman, S.; Ben Saoud, S.: *Hard and soft-core implementation of embedded control application using RTOS*. In: *IEEE International Symposium on Industrial Electronics*, 2008. S. 1896–1901.

15. Benra, Juliane T. (Hg.): *Software-Entwicklung für Echtzeitsysteme*. Springer. Berlin, Heidelberg, 2009.
16. Biesheuvel, C.; Weigel, S.; Heindel, W.: *Mammography Screening: Evidence, History and Current Practice in Germany and Other European Countries*. In: *Breast care*, Nr. 2, 2011. S. 104–109.
17. Bolch, G.: *Prozessautomatisierung: Material zur Vorlesung Informatik 4 der Universität Erlangen-Nürnberg*. Erlangen, 2002.
18. Boone, J. M.; Kwan, A. L.; Yang, K.; Burkett, G. W.; Lindfors, K. K.; Nelson, T. R.: *Computed tomography for imaging the breast*. In: *Journal of mammary gland biology and neoplasia*, Nr. 2, 2006. S. 103–111.
19. Börcsök, J.; Hayek, A.; Machmur, B.; Umar, M.: *Design and Implementation of an Embedded System on a FPGA with uClinux*. Technical Report, Institute of Computer Architecture and System Programming, University of Kassel, Germany. Kassel, 2009.
20. Brinkschulte, U.; Ungerer, T.: *Mikrocontroller und Mikroprozessoren*. 2. Aufl., Springer. Berlin/Heidelberg, 2007.
21. Calderón, H.; Elena, C.; Vassiliadis, S.: *Soft Core Processors and Embedded Processing: A survey and analysis*. In: *Conference Proceedings of ProRISC*, 2005. S. 483–488.
22. CAN in Automation (Hg.): *CiA201: CAN Application layer for industrial applications*, 1997.
23. CAN in Automation (Hg.): *CiA412: Profiles for medical devices*, 2005.
24. CAN in Automation (Hg.): *CiA301: Application layer and communication profile*, 2011.
25. Carney, P. A.; Miglioretti, D. L.; Yankaskas, B. C.; Kerlikowske, K.; Rosenberg, R.; Rutter, C. M. et al.: *Individual and combined effects of age, breast density, and hormone replacement therapy use on the accuracy of screening mammography*. In: *Annals of Internal Medicine*, Nr. 3, 2003. S. 168–175.
26. Casey, C. J.; Clifford, J. A.; Dalman, H.; Hum, R. W.; Pearson, P. E.: *Gantry position reference for tomographic scanners*. Veröffentlichungsnr: US5175754, 1992.
27. Chang, C. H.; Sibala, J. L.; Fritz, S. L.; Gallagher, J. H.; Dwyer, S. J.; Templeton, A. W.: *Computed tomographic evaluation of the breast*. In: *American Journal of Roentgenology*, Nr. 3, 1978. S. 459–464.
28. Chase, J.; Gallatin, A.; Yocum, K.: *End system optimizations for high-speed TCP*. In: *IEEE Communications Magazine*, Nr. 4, 2001. S. 68–74.
29. Chemane, L. A.; Nunes, A. F., JR.; Hancke, G. P.: *Industrial information infrastructure and intelligent instrumentation-the options*. In: *Proceedings of the IEEE International Symposium on Industrial Electronics*, Bd. 1, 1997. S. 44–48.

30. Ching-Ming, L.: *Method of and apparatus for collecting more views of data in a CT scanner with limited data transfer rate*. Veröffentlichungsnr: US6259761, 2001.
31. Conmy, P. M.; Pygott, C.; Bate, I.: *VHDL guidance for safe and certifiable FPGA design*. In: *5th IET International Conference on System Safety*, 2010. S. 1–6.
32. Cousins, T.: *Safety and Reliability Engineering in Medical Electronic Products - A Case Study*. In: *Electricity+Control*, Nr. 3, 2007.
33. Crawford, C. R.; King, K. F.; Toth, T. L.: *Helical scanning computed tomography apparatus with constrained tracking of the x-ray source*. Veröffentlichungsnr: US5224136, 1993.
34. Crawford, C. R.; Ling, Y.; Wegener, A. W.: *Kompression und Speicherung von Projektionsdaten in einem Computertomographiesystem*. Veröffentlichungsnr: DE112009003525, 2013.
35. Davey, B.: *Pain during mammography: Possible risk factors and ways to alleviate pain*. In: *Radiography*, Nr. 3, 2007. S. 229–234.
36. Day, J.; Zimmermann, H.: *The OSI reference model*. In: *Proceedings of the IEEE*, Bd. 71, 1983. S. 1334–1340.
37. De Man, B. K.; Muller, S. L.: *System and method for breast imaging using x-ray computed tomography*. Veröffentlichungsnr: US20120128120, 2012.
38. Decotignie, J.-D.: *Ethernet-Based Real-Time and Industrial Communications*. In: *Proceedings of the IEEE*, Bd. 93, 2005. S. 1102–1117.
39. Dedic, J.; Hasanovic, A.; Golob, D.; Plesko, M.: *Extremely low-jitter FPGA based synchronization timing system*. In: *IEEE Particle Accelerator Conference*, 2007. S. 296–298.
40. Delgado, E. C.; Ren, Z.: *Stationary computed tomography system and method*. Veröffentlichungsnr: US20070003004, 2007.
41. Demharter, N.: *Vorrichtung zur Erzeugung positionsbasierter Triggersignale für Datenaufnehmer in CT-Gantries*. Veröffentlichungsnr: DE10140741, 2003.
42. devboards GmbH (Hg.): *Datasheet DBC4CE55 Cyclone IV Development Board*. Version 1.01, 2011.
43. Di Odoardo, A.; Geraci, A.; Ripamonti, G.; Seminari, M.: *A fully digital architecture for trigger circuits in programmable logic*. In: *IEEE Transactions on Nuclear Science*, Nr. 6, 2002. S. 3314–3321.
44. DIN 44300-2:1988 (1988-11): *Informationsverarbeitung - Begriffe - Informationsdarstellung*.
45. DIN 66201:1981 (1981-05): *Prozeßrechensysteme*.

46. DIN EN 60601-1:2006 (2007-12): *Medizinische elektrische Geräte - Teil 1: Allgemeine Festlegungen für die Sicherheit einschließlich der wesentlichen Leistungsmerkmale.*
47. DIN EN 60601-1-2:2007 (2007-12): *Medizinische elektrische Geräte - Teil 1-2: Allgemeine Festlegungen für die Sicherheit einschließlich der wesentlichen Leistungsmerkmale - Ergänzungsnorm: Elektromagnetische Verträglichkeit - Anforderungen und Prüfungen.*
48. DIN EN 60601-2-44:2009 (2010-02): *Medizinische elektrische Geräte - Teil 2-44: Besondere Festlegungen für die Sicherheit einschließlich der wesentlichen Leistungsmerkmale von Röntgeneinrichtungen für die Computertomographie (IEC 60601-2-44:2009).*
49. DIN EN 60601-4:1999 (2001-04): *Medizinische elektrische Geräte - Teil 1-4: Allgemeine Festlegungen für die Sicherheit; Ergänzungsnorm: Programmierbare elektrische medizinische Systeme.*
50. DIN EN 61508-1:2010 (2011-02): *Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme - Teil 1: Allgemeine Anforderungen.*
51. DIN EN 61784-1:2010 (2011-04): *Industrielle Kommunikationsnetze - Profile - Teil 1: Felddbusprofile (IEC 61784-1:2010).*
52. DIN EN 61784-2:2010 (2011-04): *Industrielle Kommunikationsnetze - Profile - Teil 2: Zusätzliche Felddbusprofile für Echtzeitnetzwerke basierend auf ISO/IEC 8802-3 (IEC 61784-2:2010).*
53. DIN EN 62264-1:2008 (2008-06): *Integration von Unternehmensführungs- und Leitsystemen - Teil 1: Modelle und Terminologie (IEC 62264-1:2003).*
54. DIN EN 62264-3:2007 (2008-01): *Integration von Unternehmens-EDV und Leitsystemen - Teil 3: Aktivitätsmodelle für das operative Produktionsmanagement (IEC 62264-3:2007).*
55. DIN EN 62304:2006 (2007-03): *Medizingeräte-Software - Software-Lebenszyklus-Prozesse (IEC 62304:2006).*
56. DIN EN ISO 14971:2012 (2013-04): *Medizinprodukte - Anwendung des Risikomanagements auf Medizinprodukte.*
57. Direct Insight Ltd (Hg.): *RTX / RTX SMP Real-Time Extension for Windows.* Online verfügbar unter <http://www.directinsight.co.uk/products/venturcom/rtx.html>, zuletzt geprüft am 18.04.2013.
58. Dr. Johannes Heidenhain GmbH (Hg.): *Interface-Elektroniken: Produktkatalog, 2007.*
59. Dr. Johannes Heidenhain GmbH (Hg.): *Winkelmessgeräte: Produktkatalog, 2007.*
60. Duda, V.; Schulz-Wendtland, R.: *Mammadiagnostik: Komplementärer Einsatz aller Verfahren.* 1. Aufl., Springer. Berlin, 2004.

61. Edler, W.; Grottel, J.; Karl, H.; Mehldau, H.: *Computertomograph*. Veröffentlichungsnr: DE102012201850, 2012.
62. Etschberger, K.: *Controller-Area-Network: Grundlagen, Protokolle, Bausteine, Anwendungen*. 3. Aufl., Hanser. München, 2002.
63. Express Logic, I.: *Preisanfrage ThreadX*. Email, April 2013.
64. Express Logic, Inc (Hg.): *ThreadX Certification*. Online verfügbar unter [http://rtos.com/products/threadx/certification\\_pack](http://rtos.com/products/threadx/certification_pack), zuletzt geprüft am 17.05.2012.
65. Express Logic, Inc (Hg.): *ThreadX Features*. Online verfügbar unter <http://rtos.com/products/threadx/>, zuletzt geprüft am 17.04.2013.
66. Fan, X.; Jonsson, M.: *Guaranteed real-time services over standard switched Ethernet*. In: *IEEE Conference on Local Computer Networks*, 2005. S. 483–492.
67. Fan, X.; Wang, Z.; Sun, Y.: *How to guarantee factory communication with switched Ethernet: survey of its emerging technology*. In: *28th Annual Conference of the IEEE Industrial Electronics Society, Bd. 3*, 2002. S. 2525–2530.
68. Farsi, M.; Ratcliff, K.; Barbosa, M.: *An overview of controller area network*. In: *Computing Control Engineering Journal*, Nr. 3, 1999. S. 113–120.
69. Felser, M.: *Ethernet TCP/IP in automation: a short introduction to real-time requirements*. In: *IEEE International Conference on Emerging Technologies and Factory Automation, Bd. 2*, 2001. S. 501–504.
70. Felser, M.: *Real-Time Ethernet - Industry Prospective*. In: *Proceedings of the IEEE*, 2005. S. 1118–1129.
71. Feng, R.; Changyun, M.; Chenxi, S.: *The System of Signal Acquisition and Processing of Industrial CT*. In: *International Conference on Control, Automation and Systems Engineering*, 2011. S. 1–4.
72. Feng, W.; Balaji, P.; Baron, C.; Bhuyan, L. N.; Panda, D. K.: *Performance characterization of a 10-Gigabit Ethernet TOE*. In: *13th Symposium on High Performance Interconnects*, 2005. S. 58–63.
73. Feng, W.; Hurwitz, J.; Newman, H.; Ravot, S.; Les Cottrell, R.; Martin, O. et al.: *Optimizing 10-Gigabit Ethernet for Networks of Workstations, Clusters, and Grids: A Case Study*. In: *ACM/IEEE Supercomputing Conference*, 2003. S. 15–21.
74. Flik, T.: *Mikroprozessortechnik und Rechnerstrukturen*. 7. Aufl., Springer. Berlin/Heidelberg, 2005.
75. Förster, F.: *Mit FPGAs auf der sicheren Seite*. In: *Elektronik Informationen*, Nr. 6, 2010. S. 40–42.
76. Fowler, K.: *Mission-critical and safety-critical systems handbook: Design and development for embedded applications*. 1. Aufl., Elsevier/Newnes. Amsterdam, 2010.

77. Freytag, M.: *Datenkommunikation in der Leittechnik: Status und Trend*. In: *Automatisierungstechnische Praxis*, Nr. 5, 1986. S. 215–222.
78. Fröhlich, I.; Kajetanowicz, M.; Korcyl, K.; Krzemien, W.; Palka, M.; Salabura, P. et al.: *A General Purpose Trigger and Readout Board for HADES and FAIR-Experiments*. In: *IEEE Transactions on Nuclear Science*, Nr. 1, 2008. S. 59–66.
79. Fujimoto, H.; Manabe, Y.: *Positional adjustment of resolution in radiation CT scanner*. Veröffentlichungsnr: US5386446, 1995.
80. Fujita, H.; Tsai, D.-Y.; Itoh, T.; Doi, K.; Morishita, J.; Ueda, K.; Ohtsuka, A.: *A simple method for determining the modulation transfer function in digital radiography*. In: *IEEE Transactions on Medical Imaging*, Nr. 1, 1992. S. 34–39.
81. Gärtner, A.: *Elektrische Sicherheit in der Medizintechnik*. 1. Aufl., TÜV Media. Köln, 2008.
82. Gärtner, A.: *Medizinproduktesicherheit*. 1. Aufl., TÜV Media. Köln, 2008.
83. Gärtner, A.: *Normen in der Medizintechnik*. 1. Aufl., TÜV Media. Köln, 2009.
84. Gärtner, A.: *Der PC – (K)ein Medizinprodukt: Formalrechtliche und sicherheitstechnische Kriterien der Kombination PC und Medizinprodukt nach dem MPG*. Hg. v. Baaske Medical GmbH & Co.KG, 2012.
85. Gebert, U.: *Untersuchung von Eigenschaften photonenzählender pixelierter Halbleiterdetektoren der Medipix-Familie*. Diplomarbeit, Physikalisches Institut, Friedrich-Alexander-Universität Erlangen-Nürnberg. Erlangen, 2007.
86. Gisvold, J. J.; Reese, D. F.; Karsell, P. R.: *Computed tomographic mammography (CTM)*. In: *American Journal of Roentgenology*, Nr. 6, 1979. S. 1143–1149.
87. Glick, S. J.: *Breast CT*. In: *Annual review of biomedical engineering*, 2007. S. 501–526.
88. Goldman, L. W.: *Principles of CT and CT Technology*. In: *Journal of Nuclear Medicine Technology*, Nr. 3, 2007. S. 115–128.
89. Gong, G.; Xue, T.: *Embedded Controller for Industrial CT Trigger Module*. In: *Proceedings of the 8th International Workshop on Personal Computers and Particle Accelerators (PCaPAC)*, 2010. S. 201–203.
90. Grottel, J.; Gruber, K.; Popescu, S.: *Verfahren zur Daten- und Signalübertragung zwischen unterschiedlichen Teileinheiten einer medizintechnischen Anlage*. Veröffentlichungsnr: DE10354494, 2005.
91. Häberlein, T.: *Technische Informatik: Ein Tutorium der Maschinenprogrammierung und Rechnertechnik*. 1. Aufl., Vieweg + Teubner. Wiesbaden, 2011.
92. Hamblen, J. O.; Hall, T. S.; Furman, M. D.: *Rapid prototyping of digital systems: SOPC Edition*. 1. Aufl., Springer. New York, 2008.

93. Hangartner, T. N.: *Microprocessor control of a computed tomography scanner*. In: *Proceedings of the Annual International Conference of the IEEE*, 1988. S. 386–387.
94. Harish, M. G.; Konda, S. D.; MacMahon, H.; Newstead, G. M.: *Breast lesions incidentally detected with CT: What the general radiologist needs to know*. In: *Radiographics*, Suppl 1, 2007. S. 37–51.
95. Hebel, C., 2012: *Eine Sekunde bis zum Stopp*. Hg. v. Spiegel Online. Online verfügbar unter <http://www.spiegel.de/wirtschaft/unternehmen/spiegel-neue-ice-modelle-haben-probleme-mit-steuerungssoftware-a-869137.html>, zuletzt geprüft am 30.01.2013.
96. Hell, E.; Mattern, D.; Schardt, P.: *X-ray tube with flying focus*. Veröffentlichungsnr: US6292538, 2001.
97. Herpel, T.; Hielscher, K.-S.; Klehmet, U.; German, R.: *Stochastic and deterministic performance evaluation of automotive CAN communication*. In: *Computer Networks*, Nr. 8, 2009. S. 1171–1185.
98. Hilton, A.; Hall, J.: *On Applying Software Development Best Practice to FPGAs in Safety-Critical Systems*. In: Reiner Hartenstein und Herbert Grünbacher (Hg.): *Field-Programmable Logic and Applications*, Bd. 1896, Springer. Berlin, 2000. S. 793–796.
99. Hilz, D.: *Ergebnisse der Abklärung klinisch nicht palpabler mammographischer Veränderungen der Mamma*. Dissertation, Medizinische Fakultät, Ludwig-Maximilian-Universität zu München. München, 2002.
100. Hisashi, T.: *Arrangements for mounting units in a computed tomography system*. Veröffentlichungsnr: US6314157, 2001.
101. Hölzer-Klüpfel, M.: *Software-Entwicklung für Medizinprodukte: Herausforderungen in einem regulierten Umfeld*. In: *Elektronik Informationen*, Nr. 12, 2008. S. 68–73.
102. Horinouchi, M.; Iso, M.; Nakanishi, S.; Nakashima, S.; Okumura, M.; Shinno, T. et al.: *X-ray computed tomography apparatus*. Veröffentlichungsnr: US7085343, 2006.
103. Hsieh, J.: *Computed tomography: Principles, design, artifacts, and recent advances*. 2. Aufl., Wiley Interscience. Hoboken, N.J., 2009.
104. Huang, R.; Chien, A.: *Benchmarking High Bandwidth Delay Product Networks*. Technical Report, Department of Computer Science, University of California, San Diego. San Diego, 2004.
105. Hughes-Jones, R.; Clarke, P.; Dallison, S.: *Performance of 1 and 10 Gigabit Ethernet cards with server quality motherboards*. In: *Future Generation Computer Systems*, Nr. 4, 2005. S. 469–488.
106. Hu, H.: *Multi-slice helical CT: scan and reconstruction*. In: *Medical Physics*, Nr. 1, 1999. S. 5–18.

107. Hu, H.; Pelc, N.; Pfoh, A.: *Reconstruction method for helical scanning computed tomography apparatus with multi-row detector array employing overlapping beams*. Veröffentlichungsnr: US5430783, 1995.
108. Humbert, B.; Bard, P.; Fuch, C.; Brasse, D.: *Gigabit High Speed TCP Data Transmission for Positron Emission Tomography*. In: *IEEE Nuclear Science Symposium Conference Record*, 2009. S. 597–601.
109. Hurwitz, J.; Wu-chun Feng: *Initial end-to-end performance evaluation of 10-Gigabit Ethernet*. In: *11th Symposium on High Performance Interconnects*, 2003. S. 116–121.
110. IEEE 802.1Q (2011): *Media Access Control (MAC) Bridges and Virtual Bridge Local Area Networks*.
111. IEEE 802.3 (2005): *Carrier sense multiple access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications*.
112. IFM Electronic GmbH (Hg.): *Schulungsunterlagen für Drehgeber/Winkelcodierer*, Jan/2004.
113. Imai, Y.: *X-ray CT apparatus*. Veröffentlichungsnr: US7706498, 2010.
114. Imrek, J.; Novak, D.; Hegyesi, G.; Kalinka, G.; Molnar, J.; Vegh, J. et al.: *Development of an FPGA-Based Data Acquisition Module for Small Animal PET*. In: *IEEE Transactions on Nuclear Science*, Nr. 5, 2006. S. 2698–2703.
115. Inoue, M.; Sano, T.; Watai, R.; Ashikaga, R.; Ueda, K.; Watatani, M.; Nishimura, Y.: *Dynamic multidetector CT of breast tumors: Diagnostic features and comparison with conventional techniques*. In: *American Journal of Roentgenology*, Nr. 3, 2003. S. 679–686.
116. Intel Corporation (Hg.): *Intel Server Adapters: Advanced settings for Intel PRO/10GbE Server Adapters*. Online verfügbar unter <http://www.intel.com/support/network/adapter/pro100/sb/cs-010619.htm>, zuletzt geprüft am 12.03.2012.
117. Intel Corporation (Hg.): *Network Connectivity: Tuning Intel Ethernet Adapter throughput performance*. Online verfügbar unter <http://www.intel.com/support/network/sb/CS-025829.htm>, zuletzt geprüft am 12.03.2012.
118. Intel Corporation (Hg.): *Intel PRO/10GbE LR Server Adapter: Industry-leading 10 Gigabit Single-mode Fiber Server Connection*, 2004.
119. Jacomet, M.; Goette, J.; Breitenstein, J.; Hager, M.: *On a development environment for real-time information processing in system-on-chip solutions*. In: *Proceedings of the 14th symposium on integrated circuits and systems design*, 2001. S. 28–31.
120. Jasperneite, J.; Neumann, P.: *Switched Ethernet for factory communication*. In: *8th IEEE International Conference on Emerging Technologies and Factory Automation, Bd. 1*, 2001. S. 205–212.

121. Jasperneite, J.; Neumann, P.; Theis, M.; Watson, K.: *Deterministic real-time communication with switched Ethernet*. In: *4th IEEE International Workshop on Factory Communication Systems*, 2002. S. 11–18.
122. Judy, P. F.: *The line spread function and modulation transfer function of a computed tomographic scanner*. In: *Medical Physics*, Nr. 3, 1976. S. 233–236.
123. Kachelriess, M.; Knaup, M.; Penssel, C.; Kalender, W.: *Flying focal spot (FFS) in cone-beam CT*. In: *IEEE Nuclear Science Symposium Conference Record*, Bd. 6, 2004. S. 3759–3763.
124. Kaempfer, G.; Tomer, M.; Yariv, M.; Rapoport, A.: *Method and System for Error Resilient Compression and Decompression of Computed Tomography Data*. Veröffentlichungsnr: US201213424578, 2012.
125. Kaiser, J.; Mock, M.: *Implementing the real-time publisher/subscriber model on the controller area network (CAN)*. In: *2nd IEEE International Symposium on Object-Oriented Real-Time Distributed Computing*, 1999. S. 172–181.
126. Kalender, W. A.: *X-ray computed tomography*. In: *Physics in Medicine and Biology*, Nr. 13, 2006. S. 29–43.
127. Kalender, W. A.: *Concepts for High-Resolution CT of the Breast*. In: Joan Martí, Arnau Oliver, Jordi Freixenet und Robert Martí (Hg.): *Digital Mammography*, Bd. 6136, Springer. Berlin/Heidelberg, 2010. S. 421–427.
128. Kalender, W. A.: *Computed tomography: Fundamentals, system technology, image quality, applications*. 3. Aufl., Publicis. Erlangen, 2011.
129. Kalender, W. A.; Beister, M.; Boone, J. M.; Kolditz, D.; Vollmar, S. V.; Weigel, M. C.: *High-resolution spiral CT of the breast at very low dose concept and feasibility considerations*. In: *European Radiology*, Nr. 1, 2011. S. 1–8.
130. Kalender, W. A.; Seissler, W.; Klotz, E.; Vock, P.: *Spiral volumetric CT with single-breath-hold technique, continuous transport, and continuous scanner rotation*. In: *Radiology*, Nr. 1, 1990. S. 181–183.
131. Kämmerer, P.; Reichel, T.: *Jitterfreie Ortstriggerung eines Detektors eines CT-Systems*. Veröffentlichungsnr: DE102009019841, 2010.
132. Kampmann-Küster, I.: *Zuverlässigkeit bildgebender Verfahren in der Früherkennung des Mammakarzinoms*. Dissertation, Medizinische Fakultät, Georg-August-Universität zu Göttingen. Göttingen, 2012.
133. Katzke, U.: *Spezifikation und Anwendung einer Modellierungssprache für die Automatisierungstechnik auf Basis der Unified Modeling Language (UML)*. Dissertation, Fachbereich Elektrotechnik/Informatik, Universität Kassel. Kassel, 2009.
134. Khan, Z.; Hussain, K.; Mir, S. A.: *Identification and analysis of performance metrics for real time operating system*. In: *IEEE International Conference on Emerging Technologies*, 2009. S. 183–187.

135. Kinsinger, R. E.: *Helical computed tomography with feedback scan control*. Veröffentlichungsnr: US6185271, 2001.
136. Kithara Software GmbH (Hg.): *Kithara »RealTime Suite«*. Online verfügbar unter <http://www.kithara.de/de/produkte/realtime-suite>, zuletzt geprüft am 18.04.2013.
137. Knight, J.: *Safety critical systems: challenges and directions*. In: *Proceedings of the 24rd International Conference on Software Engineering*, 2002. S. 547–550.
138. Kobayashi, M.: *X-ray CT scanner and control program thereof*. Veröffentlichungsnr: US20110228897, 2011.
139. Kopetz, H.: *Event-triggered versus time-triggered real-time systems*. In: Arthur Karshmer und Jürgen Nehmer (Hg.): *Operating Systems of the 90s and Beyond*, Bd. 563, Springer. Berlin/Heidelberg, 1991. S. 86–101.
140. Kopetz, H.: *Real-time systems: Design principles for distributed embedded applications*. 2. Aufl., Springer. New York, 2011.
141. Krah, J. O.; Klarenbach, C.: *FPGA based Field Oriented Current Controller for High Performance Servo Drives*. Technical Report, Institut für Automatisierungstechnik, Fachhochschule Köln. Köln, 2008.
142. Kugelstadt, T.: *The RS-485 Design Guide: Application Report*. Hg. v. Texas Instruments Incorporated, 2008.
143. Kung, Y.-S.; Shu, G.-S.: *Design and implementation of a control IC for vertical articulated robot arm using SOPC technology*. In: *IEEE International Conference on Mechatronics*, 2005. S. 532–536.
144. La Porta, T.; Schwartz, M.: *Architectures, features, and implementation of high-speed transport protocols*. In: *IEEE Network*, Nr. 3, 1991. S. 14–22.
145. Lauber, R.: *Prozessautomatisierung*. 3. Aufl., Springer. Berlin/Heidelberg, 1999.
146. Lawrenz, W.: *CAN Controller Area Network: Grundlagen und Praxis*. 5. Aufl., Hüthig. Heidelberg, 2009.
147. Lee, K. C.; Lee, S.: *Performance evaluation of switched Ethernet for real-time industrial communications*. In: *Computer Standards & Interfaces*, Nr. 5, 2002. S. 411–423.
148. Leitao, B. H.: *Tuning 10Gb network cards on Linux: A basic introduction to concepts used to tune fast network cards*. In: *Proceedings of the Linux Symposium*, 2009. S. 169–188.
149. Lepage, G.; Bogaerts, J.; Meynants, G.: *Time-Delay-Integration Architectures in CMOS Image Sensors*. In: *IEEE Transactions on Electron Devices*, Nr. 11, 2009. S. 2524–2533.
150. Lerch, R.: *Elektrische Messtechnik: Analoge, digitale und computergestützte Verfahren*. 6. Aufl., Springer Vieweg. Berlin, 2012.

151. Leveson, N. G.: *Safeware: System safety and computers: A guide to preventing accidents and losses caused by technology*. 1. Aufl., Addison-Wesley. Reading, 1995.
152. Leveson, N.; Turner, C.: *An investigation of the Therac-25 accidents*. In: *Computer*, Nr. 7, 1993. S. 18–41.
153. Linux Kernel Organization Inc. (Hg.): *Real-Time Linux Wiki*. Online verfügbar unter [https://rt.wiki.kernel.org/index.php/Main\\_Page](https://rt.wiki.kernel.org/index.php/Main_Page), zuletzt geprüft am 18.04.2013.
154. Li, Q.; Yao, C.: *Real-time concepts for embedded systems*. 1. Aufl., CMP Books. San Francisco, CA, 2003.
155. Livani, M. A.; Kaiser, J.; Jia, W.: *Scheduling Hard and Soft Real-Time Communication in a Controller Area Network*. In: *Control Engineering Practice*, Nr. 12, 1999. S. 1515–1523.
156. Loeb, M.; Rindos, A.; Holland, W.; Woolet, S.: *Gigabit Ethernet PCI adapter performance*. In: *IEEE Network*, Nr. 2, 2001. S. 42–47.
157. Loeser, J.; Haertig, H.: *Low-latency hard real-time communication over switched Ethernet*. In: *16th Euromicro Conference on Real-Time Systems*, 2004. S. 13–22.
158. Lutz, A.: *Method and apparatus for radiological examination of cardiac phases of a patient*. Veröffentlichungsnr: US5832051, 1998.
159. Lu, Z.; Zhang, X.; Sun, C.: *An Embedded System with uClinux Based on FPGA*. In: *IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, 2008. S. 691–694.
160. Marau, R.; Almeida, L.; Pedreiras, P.: *Enhancing real-time communication over cots ethernet switches*. In: *IEEE International Workshop on Factory Communication Systems*, 2006. S. 295–302.
161. Marciniowski, P.; Fransson, K.; Gustafsson, L.; Calen, H.: *A fast programmable trigger system for the WASA experiment*. In: *IEEE Transactions on Nuclear Science*, Nr. 1, 2002. S. 312–317.
162. Marieska, M. D.; Hariyanto, P. G.; Fauzan, M. F.; Kistijantoro, A. I.; Manaf, A.: *On performance of kernel based and embedded Real-Time Operating System: Benchmarking and analysis*. In: *International Conference on Advanced Computer Science and Information System*, 2011. S. 401–406.
163. Martínez, J. M.; Harbour, M. G.: *RT-EP: A Fixed-Priority Real Time Communication Protocol over Standard Ethernet*. In: *10th International Conference on Reliable Software Technologies*, 2005. S. 180–195.
164. Maxim Integrated Products (Hg.): *Fundamentals of RS-232 Serial Communications: Application Note 83*, 2005.
165. McLean, P.; Ayoub, H.: *uCOS II vs uClinux*. Technical Report, Computer Architecture Research Group, University of Ottawa. Ottawa, 2009.

166. Metzger, M.; Polakow, G.: *A Survey on Applications of Agent Technology in Industrial Process Control*. In: *IEEE Transactions on Industrial Informatics*, Nr. 4, 2011. S. 570–581.
167. Micrium Inc. (Hg.): *μC/OS-II Features*. Online verfügbar unter <http://micrium.com/rtos/ucosii/features/>, zuletzt geprüft am 17.04.2013.
168. Micrium Inc. (Hg.): *The Micrium/Validated Software Advantage*. Online verfügbar unter [http://micrium.com/rtos/micrium\\_validated/medical\\_industries](http://micrium.com/rtos/micrium_validated/medical_industries), zuletzt geprüft am 17.05.2012.
169. Micrium Inc.: *Preis Anfrage μC/OS-II*. Email, Mai 2012.
170. Miyake, K.; Hayakawa, K.; Nishino, M.; Nakamura, Y.; Morimoto, T.; Urata, Y. et al.: *Benign or malignant?: Differentiating breast lesions with computed tomography attenuation values on dynamic computed tomography mammography*. In: *Journal of computer assisted tomography*, Nr. 6, 2005. S. 772–779.
171. Monmasson, E.; Cirstea, M. N.: *FPGA Design Methodology for Industrial Control Systems—A Review*. In: *IEEE Transactions on Industrial Electronics*, Nr. 4, 2007. S. 1824–1842.
172. Moog Components Group (Hg.): *Motion Technology Catalog: Produktkatalog*, 2011.
173. Nanni, E.; Pasini, A.; Righini, D.: *Method for synchronising an emitter and a detector in a computed tomography scanner*. Veröffentlichungsnr: US7751526, 2010.
174. Ning, R.; Conover, D.; Yu, Y.; Zhang, Y.; Cai, W.; Betancourt-Benitez, R.; Lu, X.: *A novel cone beam breast CT scanner: system evaluation*. In: *Proceedings of SPIE 6510*, 2007. S. 651030-651030-9.
175. Nisar, S. K.; Ahmed, M.; Ayub, H.; Baig, I.: *Operating System Performance Analyzer for Low-End Embedded Systems*. In: *International Journal of Computer Science Issues (IJCSI)*, Vol. 8, Issue 6, No 3, 2011. S. 341–348.
176. Nowak, T.; Hupfer, M.; Althoff, F.; Brauweiler, R.; Eisa, F.; Steiding, C.; Kalender, W. A.: *Time-delayed summation as a means of improving resolution on fast rotating computed tomography systems*. In: *Medical Physics*, Nr. 4, 2012. S. 2249–2260.
177. Nowak, T.; Schilling, H.; Kalender, W. Prof. Dr.: *Vorrichtung und Verfahren zur zeitverzögerten Integration auf einem aus mehreren Detektormodulen zusammengesetzten Röntgendetektoren*. Veröffentlichungsnr: DE102009045092, 2010.
178. Oh, J.; Kim, W.; Lee, W.; Chung, Y.: *Scalable RTOS for SoC Platform Environments*. In: *IEEE World Congress on Computer Science and Information Engineering*, 2009. S. 443–447.
179. Pastors, F.: *CANopen in der Medizintechnik: Benefits und aktuelle Implementierungsstrategien*. In: *MEDIZIN+elektronik*, Nr. 4, 2012.

180. Pedreiras, P.; Gai, P.; Almeida, L.; Buttazzo, G.: *FTT-Ethernet: A flexible real-time communication protocol that supports dynamic QoS management on Ethernet-based systems*. In: *IEEE Transactions on Industrial Informatics*, Nr. 3, 2005. S. 162–172.
181. Pedreiras, P.; Leite, R.; Almeida, L.: *Characterizing the Real-Time Behavior of Prioritized Switched-Ethernet*. In: *2nd International Workshop on Real-Time LANS in the Internet Age*, 2003.
182. Peter, F.: *CT-Gerät*. Veröffentlichungsnr: DE19837442, 2000.
183. Pfeiffer, O.; Ayre, A.; Keydel, C.: *Embedded Networking with CAN and CANopen*. 1. Aufl., Copperhill Technologies Corp. Greenfield, MA, 2008.
184. Pilz GmbH & Co. KG (Hg.): *Pilz NOT-AUS-Schaltgerät PNOZ\_s3: Datenblatt 1001969*, 2011.
185. Polke, M.; Epple, U.: *Prozessleittechnik*. 2. Aufl., Oldenbourg. München, 1994.
186. Pope, S.; Riddoch, D.: *10Gb/s Ethernet performance and retrospective*. In: *SIGCOMM Computer Communication Review*, Nr. 2, 2007. S. 89–92.
187. Probst, U.: *Servoantriebe in der Automatisierungstechnik: Komponenten, Aufbau und Regelverfahren*. 1. Aufl., Vieweg + Teubner. 2011.
188. QNX Software Systems Ltd (Hg.): *QNX Neutrino RTOS*. Online verfügbar unter <http://www.qnx.com/products/neutrino-rtos/neutrino-rtos.html>, zuletzt geprüft am 18.04.2013.
189. Quade, J.: *Harte und weiche Echtzeitsysteme: Material zur Vorlesung Echtzeitsysteme I+II der Hochschule Niederrhein*. Niederrhein, 2004.
190. Rechenberg, P.: *Informatik-Handbuch*. 4. Aufl., Hanser. München/Wien, 2006.
191. Richardson, P.; Haniak, P.; Sieh, L.; U. S. Army; Vetronics Tank-automotive: *A Real-Time Control Network Protocol for Embedded Systems Using Controller Area Network*. In: *IEEE Electronic and Information Technology Conference*, 2001. S. 1–10.
192. Robert Bosch GmbH (Hg.): *CAN Specification Version 2.0*, 1991.
193. Robert Koch-Institut; Gesellschaft der epidemiologischen Krebsregister in Deutschland e. V. (Hg.): *Krebs in Deutschland 2007/2008*, 2012.
194. Roosevelt, G.; Bueno, D.; Haque, J.; Roper, W.; Romanko, T.: *Rad-Hard high speed serial communication using Honeywell SerDes macros*. In: *IEEE Aerospace conference*, 2009. S. 1–10.
195. Sardanelli, F.; Calabrese, M.; Zandrino, F.; Melani, E.; Parodi, R.; Imperiale, A. et al.: *Dynamic helical CT of breast tumors*. In: *Journal of computer assisted tomography*, Nr. 3, 1998. S. 398–407.

196. Sauter, T.: *Integration aspects in automation - a technology survey*. In: *10th IEEE Conference on Emerging Technologies and Factory Automation, Bd. 2*, 2005. S. 255–263.
197. Sauter, T.: *The continuing evolution of integration in manufacturing automation*. In: *IEEE Industrial Electronics Magazine*, Nr. 1, 2007. S. 10–19.
198. Sauter, T.; Soucek, S.; Kastner, W.; Dietrich, D.: *The Evolution of Factory and Building Automation*. In: *IEEE Industrial Electronics Magazine*, Nr. 3, 2011. S. 35–48.
199. Sauter, T.; Wollschlaeger, M.: *Feldbussysteme - Historie, Eigenschaften und Entwicklungstrends*. In: *it - Information Technology*, Nr. 4, 2000. S. 7–17.
200. Scarlett, J.; Brennan, R.: *Re-evaluating Event-Triggered and Time-Triggered Systems*. In: *IEEE Conference on Emerging Technologies and Factory Automation*, 2006. S. 655–661.
201. Schardt, P.; Deuringer, J.; Freudenberger, J.; Hell, E.; Knüpfer, W.; Mattern, D.; Schild, M.: *New x-ray tube performance in computed tomography by introducing the rotating envelope tube technology*. In: *Medical Physics*, Nr. 9, 2004. S. 2699–2706.
202. Schiffmann, W.; Bähring, H.; Hönig, U.: *Grundlagen der PC-Technologie*. 1. Aufl., Springer. Berlin/Heidelberg, 2011.
203. Schlegel, C.: *Implementierung von Industrial Real-Time Ethernet Schnittstellen: Universelle Lösung auf Basis der FPGA-Technologie*. In: *A&D*, Nr. 4, 2009. S. 55–58.
204. Schleifring und Apparatebau GmbH (Hg.): *Slip Ring Solutions - Industrial Applications: Produktkatalog*, 2012.
205. Scholz, P.: *Softwareentwicklung eingebetteter Systeme: Grundlagen, Modellierung, Qualitätssicherung*. 1. Aufl., Springer. Berlin/Heidelberg, 2005.
206. Schweizer, H.: *Medizinisches Gerät mit zentraler Schnittstelle für Interaktionen zwischen Benutzer und dem Gerät am 04.01.2011*. Veröffentlichungsnr: DE102011002426, 2011.
207. SEGGER Microcontroller GmbH & Co. KG (Hg.): *embOS Features*. Online verfügbar unter <http://www.segger.com/embos.html>, zuletzt geprüft am 17.04.2013.
208. SEGGER Microcontroller GmbH & Co. KG (Hg.): *embOS Interrupt Latency*. Online verfügbar unter <http://www.segger.com/interrupt-latency.html>, zuletzt geprüft am 17.04.2013.
209. Skeie, T.; Johannessen, S.; Holmeide, O.: *The road to an end-to-end deterministic Ethernet*. In: *4th IEEE International Workshop on Factory Communication Systems*, 2002. S. 3–9.

210. Soltero, M.; Zhang, J.; Cockril, C.: *RS-422 and RS-485 Standards Overview and System Configurations: Application Report*. Hg. v. Texas Instruments Incorporated, 2010.
211. Staude, A.; Goebbels, J.: *Ortsauflösung in der Computertomographie: Vergleich von MTF und Linienpaarstrukturen*. In: *DGZfP-Jahrestagung*, 2011.
212. Steinke, P.: *Bussysteme in der Automatisierungs- und Prozesstechnik: Grundlagen, Systeme*. 8. Aufl., Vieweg+Teubner. Wiesbaden, 2012.
213. Stewart, D. B.: *Measuring Execution Time and Real-Time Performance*. In: *Proceedings of the Embedded Systems Conference*, 2002. S. 1–15.
214. Strotzer, M.: *Flachdetektoren in der digitalen Radiographie: Sofortbilder hoher diagnostischer Qualität bei reduzierter Strahlenbelastung*. In: *Dtsch Arztebl International*, Nr. 38, 2002. S. 2484–2488.
215. Sukhanov, A.; Kulinich, P.; Sarin, P.: *A gigabit/s data acquisition system*. In: *14th IEEE-NPSS Real Time Conference*, 2005. S. 524–526.
216. Sun, F.; Li, X.; Wang, Q.; Tang, C.: *FPGA-based embedded system design*. In: *IEEE Asia Pacific Conference on : Circuits and Systems*, 2008. S. 733–736.
217. Szadkowski, Z.: *Prototype trigger and readout-electronics built on unified Altera platform for underground muon counters triggered by surface detectors*. In: *IEEE Real Time Conference*, 2012. S. 1–8.
218. Szadkowski, Z.; Becker, K.-H.; Kampert, K.-H.: *Development of a new first level trigger for the surface array in the Pierre Auger Observatory based on the CycloneTM Altera FPGA*. In: *Nuclear Instruments and Methods in Physics Research Section A*, Nr. 3, 2005. S. 793–802.
219. Szeles, C.: *CdZnTe and CdTe materials for X-ray and gamma ray radiation detector applications*. In: *Physica Status Solidi B*, Nr. 3, 2004. S. 783–790.
220. Tan, S.-L.; Anh, T. N.: *Real-time operating system (RTOS) for small (16-bit) microcontroller*. In: *IEEE 13th International Symposium on Consumer Electronics*, 2009. S. 1007–1011.
221. Tao, X.; Gong, G.-H.: *Trigger system for industrial CT*. In: *16th IEEE NPSS Real Time Conference*, 2009. S. 490–491.
222. Texas Instruments Incorporated (Hg.): *RS485 Transceiver TI SN65HVD10*, 2003.
223. Thomesse, J.-P.: *Fieldbus Technology in Industrial Automation*. In: *Proceedings of the IEEE*, Nr. 6, 2005. S. 1073–1101.
224. Tindell, K.; Burns, A.: *Guaranteed Message Latencies for Distributed Safety-Critical Hard Real-Time Control Networks*. Technical Report, Department of Computer Science, University of York. York, 1994.
225. Tindell, K.; Burns, A.; Wellings, A.: *Calculating Controller Area Network (CAN) Message Response Times*. In: *Control Engineering Practice*, 1995. S. 1163–1169.

226. Tirumala, A.; Cottrell, L.; Dunigan, T.: *Measuring end-to-end bandwidth with Iperf using Web100*. In: *Proceedings of the Fifth Passive and Active Measurement Workshop*, 2003. S. 2003.
227. Tong, J.; Anderson, I.; Khalid, M.: *Soft-Core Processors for Embedded Systems*. In: *International Conference on Microelectronics*, 2006. S. 170–173.
228. Tornai, M. P.; McKinley, R. L.; Bryzmialkiewicz, C. N.; Madhav, P.; Cutler, S. J.; Crotty, D. J. et al.: *Design and development of a fully 3D dedicated x-ray computed mammatomography system*. In: *Proceedings of SPIE 5745*, 2005. S. 189–197.
229. Toth, T. L.: *Variable dose application by modulation of x-ray tube current during CT scanning*. Veröffentlichungsnr: US5379333, 1995.
230. Traxler, M.; Becker, R.; Frohlich, I.; Kuhn, W.; Lehnert, J.; Lichtblau, C. et al.: *The multi-level trigger system of the HADES detector*. In: *11th IEEE Real Time Conference*, 1999. S. 529–532.
231. Turley, J., 2005: *Embedded systems survey: Operating systems up for grabs*. Hg. v. UBM LLC. Online verfügbar unter <http://www.eetimes.com/General/PrintView/4025539>, zuletzt geprüft am 07.02.2013.
232. Upender, B. P.; Brajczewsk, D. C.; Dean, A. G.; Hatzirallis, S. A.: *Message routing in control area network (CAN) protocol*. Veröffentlichungsnr: US5854454, 1998.
233. Urbanek, P.: *Embedded Systems: Ein umfassendes Grundlagenwerk mit vielen Anwendungsbeispielen*. 1. Aufl., HSU. Nürnberg, 2007.
234. Veríssimo, P.; Rodrigues, L.: *Distributed systems for system architects*. 1. Aufl., Kluwer Academic. Boston, 2001.
235. Voruganti, K.; Sarkar, P.: *An analysis of three gigabit networking protocols for storage area networks*. In: *IEEE International Conference on Performance, Computing, and Communications*, 2001. S. 259–265.
236. WA&S Ltd. (Hg.): *SafeRTOS Features*. Online verfügbar unter <http://www.highintegritysystems.com/safertos/features/>, zuletzt geprüft am 17.04.2013.
237. WA&S Ltd.: *Preis Anfrage SafeRTOS*. Email, Mai 2012.
238. Weidauer, J.: *Elektrische Antriebstechnik: Grundlagen, Auslegung, Anwendungen, Lösungen*. 2. Aufl., Publicis. Erlangen, 2011.
239. Wellenreuther, G.; Zastrow, D.: *Automatisieren mit SPS: Theorie und Praxis*. 5. Aufl., Vieweg + Teubner. Wiesbaden, 2011.
240. Wörn, H.; Brinkschulte, U.: *Echtzeitsysteme: Grundlagen, Funktionsweisen, Anwendungen*. 1. Aufl., Springer. Berlin/Heidelberg, 2005.

241. Xiao, Y.; Chen, Z.; Zhang, L.: *Design of Synchronized Data Acquisition System based on Switched Integrator Amplifier for the Industrial CT*. In: *9th International Conference on Electronic Measurement & Instruments (ICEMI)*, 2009. S. 928–931.
242. Yaffe, M.: *Detectors for Digital Mammography*. In: U. Bick und F. Diekmann (Hg.): *Digital Mammography*, Springer. Berlin/Heidelberg, 2010. S. 13–31.
243. Yaffe, M. J.; Rowlands, J. A.: *X-ray detectors for digital radiography*. In: *Physics in Medicine and Biology*, Nr. 1, 1997. S. 13–31.
244. Yiannacouras, P.; Steffan, J. G.; Rose, J.: *Exploration and Customization of FPGA-Based Soft Processors*. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Nr. 2, 2007. S. 266–277.
245. Yiming, A.; Eisaka, T.: *A switched Ethernet protocol for hard real-time embedded system applications*. In: *19th International Conference on Advanced Information Networking and Applications*, Bd. 2, 2005. S. 41–44.
246. Zeltwanger, H.: *CAN in Medizingeräten*. In: *MED engineering*, Nr. 5, 2011. S. 56–59.
247. Zimmermann, H.: *OSI Reference Model-The ISO Model of Architecture for Open Systems Interconnection*. In: *IEEE Transactions on Communications*, Nr. 4, 1980. S. 425–432.
248. Zimmermann, W.; Kampe, G.: *Digitales Winkel und Lagemessverfahren: Systementwurf und Realisierung mit VHDL und Schaltungssynthese*. Technical Report, Fachbereich Informationstechnik, Hochschule für Technik Esslingen. Esslingen, 1999.
249. Zuberi, K. M.; Shin, K. G.: *Scheduling messages on controller area network for real-time CIM applications*. In: *IEEE Transactions on Robotics and Automation*, Nr. 2, 1997. S. 310–316.
250. Zurawski, R.: *The Industrial Communication Technology Handbook*. 1. Aufl., Taylor & Francis. 2005.



## Abbildungsverzeichnis

Abbildung 1:	Der dedizierte Brust-CT-Scanner von IMP und CTI.....	6
Abbildung 2:	Das Prinzip bisheriger Ansätze für Brust-CT-Scanner.....	8
Abbildung 3:	Das Prinzip des dedizierten Brust-CT-Scanners von IMP und CTI.....	9
Abbildung 4:	Ein konventionellen Ganzkörper-CT-Scanner .....	11
Abbildung 5:	Prinzipzeichnung einer Gantry mit Patientenliege .....	12
Abbildung 6:	Schematische Darstellung eines technischen (Teil)Prozesses .....	13
Abbildung 7:	Schematische Darstellung einer Prozesssteuerung .....	14
Abbildung 8:	Die Automatisierungspyramide .....	15
Abbildung 9:	Eine mögliche Umsetzung der Automatisierungspyramide .....	16
Abbildung 10:	Konsequenzen bei Nichterfüllen von Echtzeitschranken .....	16
Abbildung 11:	Das RTOS zwischen Hardware und dem Anwenderprogramm .....	22
Abbildung 12:	Das ISO/OSI-Referenzmodell zur Datenübertragung .....	23
Abbildung 13:	Aufbau eines Ethernet-Pakets entsprechend IEEE 802.3 .....	26
Abbildung 14:	Das Prinzip dreier Ansätze für echtzeitfähiges Ethernet .....	28
Abbildung 15:	Aufbau eines CAN-Telegramms entsprechend CAN 2.0B .....	29
Abbildung 16:	Aufbau der Röntgenröhre aus mehreren Komponenten .....	33
Abbildung 17:	Darstellung eines positionsgesteuerten Brennflecks .....	34
Abbildung 18:	Der modulare Halbleiterdetektor des Brust-CT-Scanners .....	35
Abbildung 19:	Vergleich einer Projektionsaufnahme ohne und mit TDS und FSC... ..	36
Abbildung 20:	Simulationsergebnisse zur Effektivität von TDS .....	37
Abbildung 21:	Schematische Darstellung eines Servoantriebs.....	38
Abbildung 22:	Typisches Steuerungskonzept für einen CT-Scanner .....	39
Abbildung 23:	Sternförmige Steuerungsarchitektur eines CT-Scanners .....	39
Abbildung 24:	Das Ebenenmodell des Brust-CT-Scanners.....	41
Abbildung 25:	Übersicht über die Steuerungskommunikation.....	42
Abbildung 26:	Kommunikationsablauf des „CT_Control_CAN“-Protokolls .....	45
Abbildung 27:	Aufteilung eines CAN-Telegramms durch „CT_Control_CAN“ .....	46
Abbildung 28:	Kommunikationsablauf des „CT_Control_Ethernet“-Protokolls .....	49
Abbildung 29:	Aufteilung eines Ethernet-Pakets durch „CT_Control_Ethernet“ .....	49
Abbildung 30:	Struktur des Datenfeldes von „CT_Control_Ethernet“ .....	50
Abbildung 31:	Schematische Darstellung der Projektionsdatenübertragung .....	52
Abbildung 32:	Kommunikationsablauf des „CT_Data_Ethernet“-Protokolls.....	53
Abbildung 33:	Paketaufbau eines „CT_Data_Ethernet“-Datenpakets.....	54

Abbildung 34:	Paketaufbau eines „CT_Data_Ethernet“-Bestätigungspakets .....	54
Abbildung 35:	Synchronisation eines CT-Scanners mit einem „Triggermodul“ .....	55
Abbildung 36:	Vergleich der Bildqualität bei zeitbasierter Synchronisation .....	57
Abbildung 37:	Übersicht der Synchronisation des Brust-CT-Scanners .....	58
Abbildung 38:	Erfassung der Gantryposition .....	59
Abbildung 39:	Ausgangssignale eines Positionssensors .....	59
Abbildung 40:	Zeitverlaufdiagramm der Röntgenröhrensynchronisation .....	60
Abbildung 41:	Zeitverlaufdiagramm der Detektorsynchronisation ohne TDS .....	61
Abbildung 42:	Zeitverlaufdiagramm der Detektorsynchronisation mit TDS .....	61
Abbildung 43:	Zeitverlaufdiagramm Bildkomponentensynchronisation.....	63
Abbildung 44:	Die Eskalationsstufen des Brust-CT-Scanners .....	66
Abbildung 45:	Die verschiedenen Anwendungsfälle des Brust-CT-Scanners .....	70
Abbildung 46:	Das SOPC-Development-Board im Detail .....	72
Abbildung 47:	Die VHDL-Funktionsblöcke der Steuerungseinheiten.....	73
Abbildung 48:	Erweiterungsplatine für das DBC4CE55 FPGA-Board .....	76
Abbildung 49:	Zweilagiges Layout der Erweiterungsplatine .....	76
Abbildung 50:	Interlockkreis des Brust-CT-Scanners mit Sicherheitsschaltgerät .....	78
Abbildung 51:	Freigabe der Röntgenstrahlung durch Logikgatter .....	78
Abbildung 52:	Versuchsaufbau zur Messung der Leistungsfähigkeit eines RTOS....	79
Abbildung 53:	Messung der „Interrupt-Latency“ .....	81
Abbildung 54:	Messung der „Context-Switching-Time“ .....	81
Abbildung 55:	Versuchsaufbau zu Messungen einer 10-GBit/s-Ethernet-Strecke.....	83
Abbildung 56:	Messung des Datendurchsatzes in Abhängigkeit der Paketgröße .....	84
Abbildung 57:	Messung des Datendurchsatzes in Abhängigkeit der Puffergröße.....	85
Abbildung 58:	Messung des Datendurchsatzes in Abhängigkeit der Fenstergröße.....	85
Abbildung 59:	Messung der Paketumlaufzeit in Abhängigkeit der Paketgröße.....	86
Abbildung 60:	Aufbau zur Bestimmung von Signallaufzeit und Jitter.....	88

## Tabellenverzeichnis

Tabelle 1:	Vergleich verschiedener Typen von Echtzeitsystemen .....	19
Tabelle 2:	Vergleich verschiedener Typen von eingebetteten Systemen .....	21
Tabelle 3:	Die verschiedenen Telegrammtypen von „CT_Control_CAN“ .....	47
Tabelle 4:	Vergleich von fünf RTOS für den NIOSII-Prozessor .....	74
Tabelle 5:	Signallaufzeiten des Synchronisationssignals .....	88



## Danksagung

Als erstes möchte ich natürlich meinem Doktorvater, Prof. Dr. Willi A. Kalender, dafür danken, mir die Chance gegeben zu haben an seinem Institut eine Promotion aufnehmen zu dürfen und mich immer wieder dabei unterstützt hat diese auch erfolgreich abschließen zu können.

Besonderer Dank gebührt auch Herrn Professor Dr. Bruno Lurz, der mir an meiner Diplomfeier im Jahre 2008 das IMP als herausragendes Institut und Herrn Prof. Dr. Willi A. Kalender als herausragenden Doktorvater aufs Beste empfahl. Ohne seine Empfehlungen wäre ich wohl nun nicht in der glücklichen Lage, diese Zeilen schreiben zu dürfen.

Weiterer Dank gebührt meinen Eltern, Großeltern sowie meinem Bruder, die mich während meiner Zeit am IMP unterstützt haben und immer für mich da waren, als ich sie brauchte.

Zu guter Letzt bedanke ich mich bei meinen Kollegen vom IMP sowie der CTI. Sie waren jederzeit für mich da und unterstützten das Entstehen dieser Arbeit durch vielerlei Anregungen und Hinweise. Namentlich seien hier genannt von Seite des IMP Dr. Felix Althoff, Dr. Martin Hupfer, Dr. Daniel Kolditz, Dr. Robert Brauweiler, Dr. Fabian Eisa, Dr. Michael Meyer, Ferdinand Lück, Christian Steiding und Ronny Hendrych sowie von Seite der CTI Harry Schilling, Juliane Krämer, Dr. Marcel Beister, Günther Weninger und Franz Hirschbeck.