Sebastian Roll

Entwicklung eines Applikationskontrollers zur Untersuchung und Optimierung der Erfassungssicherheit eines RFID-Systems

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang Technische Informatik
am Fachbereich Elektrotechnik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. rer. nat. Reinhard Baran
Zweitgutachter: Prof. Dr. rer. nat. Gunter Klemke

Abgegeben am 18. März 2005

Sebastian Roll

Thema der Bachelorarbeit

Entwicklung eines Applikationskontrollers zur Untersuchung und Optimierung der Erfassungssicherheit eines RFID-Systems

Stichworte

Radio Frequenz Identifikation, Transponder ISO15693, Induktive Kopplung, Pulkerfassung, Mikrocontroller MSP430, RS-232, Lichtschranke, Auto-ID, Leserate

Kurzzusammenfassung

Diese Arbeit umfaßt die Entwicklung eines Gerätes, das der Optimierung der Erfassungssicherheit eines RFID-Systems dient (Radio Frequenz Identifikation). Das RFID-System ist eine Einheit zur automatischen Identifikation von Objekten in einem industriellen Prozeß (Fließband). Das System identifiziert Objekte, die mit einem Transponder versehen sind. Auf Anfrage sendet dieser Transponder seine ID mit Hilfe der RFID-Technik an ein Lesegerät. Der neu zu entwickelnde Applikationskontroller steuert den Ablauf der ID-Erfassung und berichtet das Ergebnis an eine höhere Instanz.

Sebastian Roll

Title of the paper

Development of an application controller for research and optimization the acquisition security of a RFID system

Keywords

Radio frequency identification, Transponder ISO15693, Inductive coupling, Anticollision sequence, Microcontroller MSP430, RS-232, Light barrier, Auto-ID, Reading rate

Abstract

This report describes the development of a technical equipment for optimizing the optimization the acquisition security of a RFID system (radio frequency identification). The RFID system is a unit for automatic identification of objects in an industrial process (assembly-line). The system identifies objects carrying a transponder. On request the transponder sends its ID with RFID technology to a reading device. The application controller to be developed should control the sequence of acquisition and report the result to a higher instance.

Inhaltsverzeichnis

1 Einleitung

In den letzten Jahren haben automatische Identifikationssysteme (Auto-ID) immer mehr an Bedeutung gewonnen. Aufgabe und Ziel der Auto-ID ist die Bereitstellung von Informationen zu Personen, Gütern und Waren. Zu den ersten Auto-ID zählen die Barcode-Papierstreifen. Sie sind recht billig, haben aber eine sehr geringe Speicherfähigkeit und können nicht umprogrammiert werden.

Eine technisch bessere Lösung ist die Speicherung der Daten auf einem Siliziumchip. Aus dem täglichen Leben ist hierzu die Chipkarte mit Kontaktfeld die bekannteste Bauform eines elektronischen Datenträgers, wie sie bei Telefon-, Versicherten- und Bankkarten eingesetzt wird. Die mechanische Kontaktierung wie bei der Chipkarte ist jedoch in vielen Fällen unzweckmäßig. Weitaus flexibler ist eine kontaktlose Übertragung der Daten zwischen dem Datenträger und einem zugehörigen Lesegerät. Idealerweise wird auch die zum Betrieb des elektronischen Datenträgers benötigte Energie durch das Lesegerät kontaktlos übertragen. Diese kontaktlose Auto-ID bezeichnet man als RFID-Systeme. Die technischen Verfahren hierzu wurden aus der Funk- und Radartechnik übernommen. Die Bezeichnung RFID steht deshalb für Radio Frequenz Identifikation, also Identifikation durch Radiowellen.

RFID hat gegenüber dem Barcode einige Vorteile, darunter fällt z.B. ein nicht mehr notwendiger Sichtkontakt zwischen Lesegerät und dem zu identifizierenden Objekt. In der Praxis erhöht sich somit die Erfassungsrate, weil Objekte schneller und effektiver erfaßt werden können. Zum anderen kann ein RFID-System bisher zentral gehaltene Daten in ein identifiziertes Objekt speichern. Dadurch wird die Datenhaltung dezentralisiert und gewissermaßen die Intelligenz in das Objekt hineinverlegt. Das heißt zum Beispiel, daß die Temperatur einer Tiefkühlpizza über den gesamten Weg von ihrer Herstellung bis zum Verkauf durch einen auf der Verpackung aufgedruckten Chip nachverfolgt werden kann.

Das Ziel dieser Arbeit ist es, ein Gerät zu entwickeln, das der Optimierung der Erfassungssicherheit eines RFID-Systems dient. Das RFID-System ist eine Einheit zur automatischen Identifikation von Objekten in einem industriellen Prozeß, z.B. ein Behälterumlaufsystem in einem Versandlager oder ein Fließband, auf dem Objekte verfahren. Das System identifiziert Objekte, die mit einem Transponder präpariert sind. Der Transponder ist zwingend erforderlich, da in ihm die Identifizierungsnummer (ID) gespeichert ist. Auf Anfrage sendet er seine ID mit Hilfe der RFID-Technik an ein Lesegerät.

Das zu entwickelnde Gerät steuert den Ablauf der Erfassung und berichtet das Ergebnis an eine höhere Instanz. Die Steuerung umfaßt einleitende Maßnahmen und eine immer wiederkehrende Abfolge des Lesens und Berichtens (Erfassung). Das Gerät soll mit Sensoren ausgestattet sein, so daß die Erfassung nur dann eingeleitet wird, wenn sich ein Objekt zum Erfassen in Reichweite befindet. Wesentlicher Bestandteil ist dabei die Aufbereitung der vom Lesegerät ausgelesenen Transponder-Daten, um sie der höheren Instanz zugänglich zu machen.

Das Zusammenspiel von Transponder, Lesegerät, Sensor und höherer Instanz ist das Verfahren (Applikation), das gesteuert werden soll. In dieser Arbeit wird das Gerät deshalb Applikationskontroller genannt. Er soll den Erfassungsvorgang automatisieren, dadurch kann zum ersten Mal eine große Anzahl an Erfassungen durchgeführt werden, die die Aussagekraft der Leserate eines RFID-Systems statistisch beweist. In der folgenden Arbeit werden zunächst die theoretischen Grundlagen dargestellt, die für die Entwicklung des Applikationskontrollers notwendig sind. Im Anschluß daran wird sein Aufbau beschrieben und mit eigenen Bildern, Diagrammen und Tabellen aus dem Projekt[1] dokumentiert.

1.1 Problemkreis

Bei der RFID- oder Transpondertechnologie handelt es sich um berührungslose Datenübertragung auf der physikalischen Basis elektromagnetischer Wechselfelder. Ein Chip und eine Antenne, die zum Datenaustausch, aber auch zur Übertragung der notwendigen Energie an den Chip dient, bilden den Transponder (auch Tag genannt), auf dem Informationen gespeichert werden können [Bild 1]. Diese Informationen können von einem Lesegerät, einem Reader, gelesen und ausgewertet werden. Ähnlich wie bei dem bereits etablierten Barcode werden also Informationen auf einem Datenträger an der Ware angebracht und hier elektromagnetisch durch einen Reader, dort optisch durch einen Scanner ausgelesen.

[1] Das Projekt wurde im Hause SICK Ibeo GmbH, Hamburg realisiert.

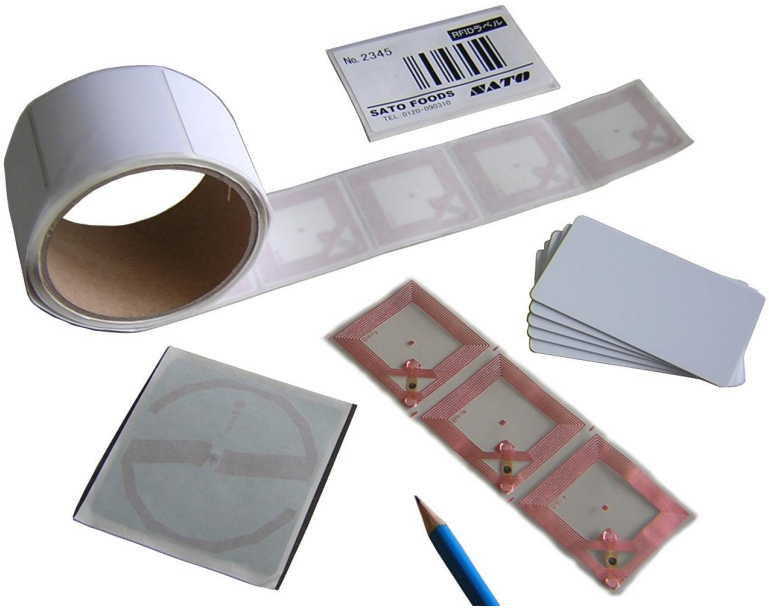

Bild 1 **Diverse Transponder: Als Aufkleber, in Chipkarte plastiniert, als Halbzeug und in Kombination mit aufgedrucktem Barcode. Abgebildet sind passive Transponder, die im Gegensatz zu aktiven Transpondern keine Batterie benötigen. Transponder, die flexibel (biegsam) sind, werden auch Smart-Labels genannt.**

In der Praxis werden Barcodes in vielen Bereichen eingesetzt, in denen auch mit Transpondern gearbeitet werden könnte. Eine Hemmschwelle für den Einsatz von Transpondern ist der Preis, der jedoch in den kommenden Jahren durch den Einsatz gedruckter Schaltungen drastisch sinken wird [Ident05]. Der Barcode hat sich in der Lager- und Distributionslogistik weitgehend durchgesetzt, nicht zuletzt aufgrund der weltweiten Standardisierung. Speziell der voll ASCII-fähige und selbstprüfende Code 128 findet Anwendung bei Förder- und Weiterreichsystemen für Paletten und Kartons, aber auch zur Identifikation von Lagerplätzen, Ladehilfsmitteln und Packstücken.

Der Transponder hat sich bisher nur im innerbetrieblichen Bereich und bei besonders hochwertigen Gütern durchsetzen können, da die Preise für den Masseneinsatz oder in Fällen, in denen der Transponder den Betrieb verläßt, noch unwirtschaftlich sind. Jedoch wird durch den Einsatz von Smart-Labels der derzeitige Preis von mindestens einem Euro für passive Transponder noch weiter sinken, und auch hier wird die RFID ihre Vorteile ausspielen können. Ob aber das RFID-System wirklich nur Vorteile gegenüber dem Barcode hat, wird sich noch im Laufe dieser und kommender Forschungsarbeiten zeigen müssen.

Hier vorab nur zwei der wichtigsten Argumente für das elektromagnetische Auslesen gegenüber dem Barcode: Der Reader braucht keinen Sichtkontakt zum Transponder. Das bedeutet, daß der Transponder auch innerhalb von Verpackungen oder Werkstücken untergebracht werden

kann und damit gegen äußere Einflüsse durch Feuchtigkeit, Temperatur, Schmutz oder Beschädigung weitgehend geschützt ist. Zum zweiten gibt es inzwischen eine funktionierende "Antikollisionstechnik“, die es ermöglicht, eine größere Anzahl von Transpondern gleichzeitig auszulesen. Der Reader schaltet dazu alle Transponder stumm, nachdem er sie identifiziert hat – jeder Transponder, der dem ISO-Standard 15693 entspricht, hat eine individuelle und weltweit einmalige Nummer – und kann diese in Bruchteilen von Sekunden nacheinander abfragen.

Es gibt zwei grundsätzlich verschiedene Typen von Transpondern: aktive und passive. Die aktiven Transponder verfügen über eine Batterie. Sie senden aktiv und ihre Lebensdauer beträgt – je nach verwendeter Batterie – bis zu fünf Jahre. Betrieben werden sie meist über Funkverfahren mit 868 MHz oder 2,45 GHz. Sie besitzen eine Reichweite von bis zu 100 m, sind jedoch sehr kostenintensiv und wegen der Batterie nur in einem eingeschränkten Temperaturbereich einsetzbar [Mieb03]. Passive Transponder gibt es für verschiedene Reichweiten: Long Range (Leseabstand bis zu 1,5 m) im Bereich 13,56 und 868 MHz oder 2,45 GHz bei ebenfalls vergleichsweise hohem Preis. Des weiteren werden Transponder im Kilohertz-Bereich (125 kHz), die einen Leseabstand bis zu 30 cm haben, angeboten. Allerdings handelt es sich hier um besonders robuste Industrie-Komponenten, die auch durch einen Wasserfilm hindurch lesbar sind.

Der Transponder mit der größten Zukunft ist der dem ISO-Standard 15693 entsprechende Typ im Megahertz-Bereich [Mieb03]. Die Betriebsfrequenz beträgt einheitlich 13,56 MHz, die Reichweite bis zu 1,5 m. Bei großen Stückzahlen erreichen die Smart-Labels einen akzeptablen Preis, und sie verfügen außerdem über die Antikollisionstechnik. Ihre Größe ist an die jeweilige Applikation anpaßbar, sie sind wiederbeschreibbar und wegen des Standards weltweit einsetzbar, daher erwartungsgemäß zukunftssicher. Diese Eigenschaften qualifizieren sie für den Einsatz in der Logistik vor allem in Branchen, in denen der Lebenslauf einzelner Komponenten oder Packstücke verfolgt und nachgewiesen werden muß, beispielsweise bei Lebensmitteln oder Postsendungen.

Die Kosten für Transponder betragen derzeit noch – abhängig vom Modell – zwischen 0,30 € und 3 € bei passiven und ca. 25 € bei aktiven Modellen für Spezialanwendungen. Smart-Labels werden in absehbarer Zeit im Bereich unter 0,20 € zu kaufen sein. [Mieb03]

1.2 Problemformen

Identifikationssysteme treten überall dort in Erscheinung, wo eine manipulationssichere und exakte Nachweisführung des Verlaufs und der Übergänge innerhalb eines logistischen Prozesses notwendig ist [Fhg01]. Was auch immer bei der Beförderung bewegt wird - Koffer, Paletten, Bücher oder Sendungen - es sind grundsätzlich Objekte, die für die effiziente Abwicklung von Logistikketten zu identifizieren sind. Das Spektrum der Einsatzgebiete für Identifikationssysteme in Produktion wie Logistik umfaßt daher u.a. Sendungsverfolgung, Lagerverwaltung und -steuerung, Fertigungsautomation. Die wichtigsten Arten von Speichermedien für Identifikationssysteme in der Fertigung sind Barcodes und Transponder:

Barcode-Systeme basieren auf der optischen Erkennung von Ziffern und Zeichen, die in Schwarz-Weiß-Strichkombinationen codiert sind. Damit stellen sie eine einfache und preiswerte Technologie dar, denn einerseits ist der technische Aufwand zum Lesen des Codes geringer als bei der Klarschrifterkennung (OCR - Optical Character Recognition) und andererseits kann die Erstellung der notwendigen Barcode-Etiketten kostengünstig mit handelsüblichen Druckern vorgenommen werden. Nachteilig ist beim Barcode allerdings die Empfindlichkeit des optischen Lesevorganges: Der Taststrahl des Laserscanners muß korrekt positioniert werden und das Barcode-Etikett darf nicht verschmutzt sein. Beim Barcode sind verschiedene Code-Systeme gebräuchlich, neben dem herkömmlichen eindimensionalen Strichcode gibt es weitere Codierungstechniken wie z.B. Stapel- oder Matrix-Codes. Letztere zeichnen sich durch eine höhere Datendichte und eine eingebaute Fehlerkorrektur aus. [Fhg01]

Bild 2 **Barcode** **Matrix-Code**

Transponder (Kunstwort aus Transmitter und Responder) sind eine Kombination aus elektronischem Chip und Antenne. Die im Chip gespeicherten Informationen werden über Radiowellen von einem externen Lesegerät ausgelesen. Daher bezeichnet man diese Technik auch als Radiofrequenz-Technik. Die passiven Modelle erhalten ihre Energie ausschließlich durch das vom Lesegerät aufgebaute Feld. Aktive besitzen eine eigene Energieversorgung zur Verstärkung des rückgesendeten Radiosignals. Außerdem wird bei beiden Modellen in read-only, d.h. nur auslesbare, und in read-write, d.h. lese- und schreibbare, Varianten differenziert.

Die genannten Systeme unterscheiden sich des weiteren durch die benutzte Frequenz, die Bauform (Folie, Röhrchen, Plastikkarte o.ä.), die Reichweite beim Schreib-/Lese-Vorgang und die Lesegeschwindigkeit. Die Grundinformation eines Transponders ist ein einprogrammierter, weltweit eindeutiger Nummerncode; je nach Ausführung speichern die Transponder weitere Informationen: z.B. Ort, Datum und Zeit der letzten Erfassung durch ein Lesegerät, Temperatur durch einen eingebauten Sensor oder an Konventionen des Barcodes (EAN 128, EAN 13) orientierte Informationen [Heis99].

Zusammengefaßt spielen drei Kriterien bei automatischen Identifikationssystemen (Auto-ID) eine wichtige Rolle: Zum einen die Erfassungssicherheit. Sie macht die Genauigkeit und die Sicherheit der Erfassung aus und ist notwendig, um fehlerhafte Eingaben bei der Datenverarbeitung zu vermeiden. Das zweite Kriterium ist die Geschwindigkeit. Sie ist nötig, um zeitlich die gewünschten Informationen rechtzeitig auszuwerten und auf dieser Basis die richtige Entscheidung treffen zu können. Der dritte Punkt ist die Eindeutigkeit der Codierung. Sie ist direkt verknüpft mit der entsprechend benötigten Geschwindigkeitsvorgabe und der Korrektheit der Lesung. Darüber hinaus trägt ein eindeutiger Code zur Kompatibilität zwischen verschiedenen Identifikationssystemen und der Außenwelt bei.

2 Projektspezifikation

Der schematische Aufbau der Applikation wird in [Bild 3] beispielhaft gezeigt und ist dem eines Behälterumlaufsystems in der Industrie nachempfunden. Ein Objekt mit Transponder wird auf einem Fließband an der Erfassungseinrichtung vorbeigeführt. Diese besteht aus einem RFID-Lesegerät mit Antenne, zwei Lichtschranken und dem Applikationskontroller. Lichtschranke 1 leitet den Erfassungsvorgang ein, sobald ihr Lichtkegel durchbrochen wird. Der Applikationskontroller liest durch das Lesegerät die ID des Transponders. Das Auslesen wird solange wiederholt, bis das Objekt, und damit der Transponder, die zweite Lichtschranke durchbrochen und wieder verlassen hat. Danach meldet der Applikationskontroller dem Host [siehe 2.1.1] die gelesene ID und wie oft er diese lesen konnte. Durch diesen Vorgang wird die Bestimmung der Leserate erstmalig automatisiert, was bisher nur per Hand möglich war. Das Ergebnis von, z.B. einstündigen Testläufen kann automatisch am Host visualisiert werden. Bei jedem Testlauf kann nun durch Optimierung von Parametern die Leserate, bezogen auf die jeweilige Applikation, erhöht werden. Diese Parameter [siehe 3.5, Übersicht Datenübertragung] werden allerdings nicht im zu entwickelnden Applikationskontroller, sondern im Lesegerät eingestellt.

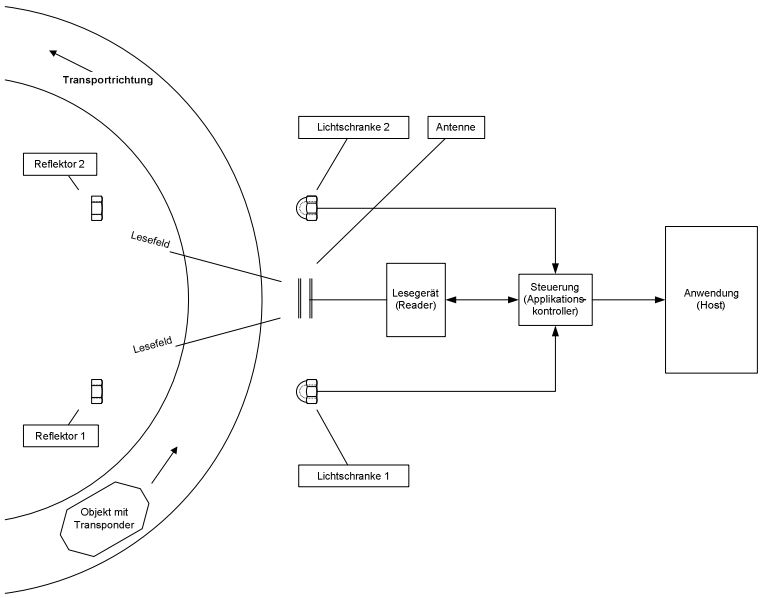

Bild 3 **Schematischer Aufbau der Applikation.**

2.1 Existierende Komponenten

Die Erfassungseinrichtung besteht aus mehreren Komponenten (siehe oben). In diesem Abschnitt werden die bereits vorhandenen und von dem in dieser Arbeit zu entwickelnden Applikationskontroller benutzten Geräte näher beschrieben. Der Applikationskontroller kann gem. [Bild 3] als Bindeglied zwischen Lesegerät (Reader) und Auswerteeinheit (Host) angesehen werden, wobei ihm die Steuerung des Readers und Koordination des Datentransfers zum Host obliegt. Ein solches Bindeglied existiert bisher nicht.

2.1.1 Host

Der Begriff "Host“ wird innerhalb dieser Arbeit synonym für das RDT400 benutzt. Das RDT400 (Remote Diagnostic Tool 400) wird bisher als Visualisierungs- und Diagnostik-Software für Barcode-Scanner-Systeme eingesetzt. Ein Scanner-System besteht aus mehreren einzelnen Barcode-Lesegeräten, die, als Tor aufgebaut, z.B. Gepäckstücke im Flughafen erfassen [Bild 4]. Als Tor deshalb, weil die Position des Barcodes (in Form einer Banderole um den Griff eines Koffers) in beliebiger Position am Scanner-System vorbeigeführt wird. Nun soll das RDT400 auch für die Visualisierung und Prüfung der Leistung für RFID-Systeme genutzt werden können.

Bild 4 **Gepäck wird mit Barcodescannern im Transfer- und Check-In-Bereich des Flughafens von Seoul identifiziert. Einzelne Scanner sind als Tor zusammengefaßt.**

Beschreibung des RDT400 im Überblick:

- Visualisierung und Monitoring von Scanner-Systemen, bis zur Ebene der individuellen Scanner
- Nutzung der existierenden Netzwerkstruktur (RS-232, Ethernet)
- Darstellung der aufbereiteten Daten auf jedem PC innerhalb des Intra-/Internet
- Abruf der aufbereiteten Daten über Standardtechnologien (Web Browser, TCP/IP)
- Entfernter Zugriff und Konfiguration möglich
- Kontinuierliches Monitoring der Scanner-System Leistung

Der RDT400 läuft als Web-Server auf einem Standard-PC (Win32). [Bild 5] zeigt die Benutzeranzeige (GUI) des RDT400. Alle Transponder-Identifikationsnummern (IDs), die der RDT400 vom Applikationskontroller entgegennimmt, werden in Logbuch-Dateien (Log) auf die Festplatte geschrieben. Für jeden Tag wird ein neuer Log erstellt. Greift ein Benutzer auf die entsprechende Webseite des RDT400 zu, erstellt dieser aus allen Logs entsprechende Tages-, Monats und Quartalsstatistiken über die Leistung der Erfassungseinrichtung.
Der Applikationskontroller sendet dem Host "Null-Daten", wenn ein Objekt beide Lichtschranken passiert hat, aber der Applikationskontroller keine ID aus dem Transponder auslesen konnte. Eine solche "Fehllesung" mindert also die Leserate der Erfassungseinheit und damit ihre Leistung. Die Leserate wird definiert als: Prozent der erfolgreichen Lesungen in bezug auf 100 Leseversuche eines bestimmten Datenträgers (hier: Transponder).

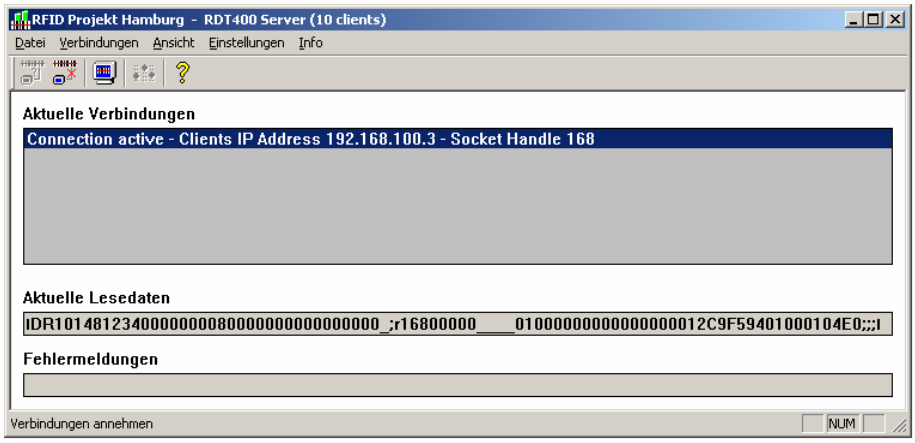

Bild 5 **Der Host zeigt die aktuellen Verbindungen zu Barcode- und RFID-Systemen an. Die Zeile "Aktuelle Lesedaten" zeigt den Unique Identifier des aktuell erkannten Objektes mit dessen Transponder (Hex C9 F5 94 01 00 01 04 E0).**

Der Host (RDT400) wird hauptsächlich zur Analyse und Optimierung automatischer Identifikationsportale in Transfergepäckanlagen mit Barcode-Scannern genutzt. Das vorherrschende

Problem besteht darin, daß diese Scanner-Technologie keine optimale Leserate aufweist, d.h. daß es noch zu viele Lesefehler gibt. Folglich kann das Gepäck oftmals nicht 100%ig korrekt identifiziert werden; es kommt zu Fehlleitungen. An Flughäfen ist zwar bekannt, ob ein Objekt identifiziert wurde oder nicht, ob der Lesefehler allerdings dem Barcode, dem Label oder dem Lesegerät zuzuschreiben war, ist nicht immer klar. Auftretende Fehler können i.d.R. in drei Gruppen unterteilt werden: die durch den Barcode verursachten (falscher Code), die durch das Label verursachten (verschmutzt, unvollständig) und die durch das Lesegerät verursachten Fehler. Die gleichen Fehlerursachen können auf die RFID Technologie übertragen werden. Ob denn nun ein Objekt richtig erkannt wurde oder ob nicht (und woran es lag) kann mit Hilfe des RDT400 besser bestimmt werden.

Inbetriebnahme und Voraussetzungen für den RDT400:
Die RDT400 Software ist ein 32 Bit Windows Programm der Firma SICK AG. Es läuft unter Windows NT4.0, 2000 oder XP. Das User-Interface läuft als Web-Application, basierend auf Microsoft's Internet Information Server (IIS). Deshalb wird das ISAPI (IIS Application Interface) für das Einlesen der täglichen Statistik-Log-Dateien benötigt, mit ihm werden die gewünschten Statistiken und Grafiken für den Benutzer (Web-Browser) dynamisch erzeugt. Der IIS wird mit der Server-Version von Microsoft NT4.0 ausgeliefert und gehört nicht zum Lieferumfang des RDT400. Alternativ dazu kann der Peer Web Service unter Microsoft NT Workstation genutzt werden, jedoch mit einer limitierten Anzahl an gleichzeitigen Verbindungen zu den Benutzern. Eine detaillierte Beschreibung über das Aufsetzen des IIS unter verschiedenen Betriebssystemen und die Installation des RDT400 steht in [Sick04].

2.1.2 Reader

Um dem Host mitzuteilen, ob ein Objekt identifiziert wurde, muß es zunächst ausgelesen werden, d.h. die ID des Transponders, der auf dem Objekt montiert ist, wird mit Hilfe eines Lesegeräts bestimmt. Der Begriff "Reader“ wird innerhalb dieser Arbeit synonym für ein solches Lesegerät verwendet.
Das RFID-Lesegerät beinhaltet ein Sende- und ein Empfangsmodul, eine Kontrolleinheit sowie ein Koppelelement (Antenne) zum Transponder. Viele Lesegeräte sind mit einer zusätzlichen Schnittstelle (RS-232, USB, etc.) ausgestattet, um die erhaltenen Daten an ein anderes System, z. B. eine Automatensteuerung, weiterzuleiten. Derzeit gibt es ca. 160 Anbieter solcher Lesegeräte [ABI05]. Ebenso hoch ist die Anzahl an Transponder-Herstellern und Systemintegratoren. Als Lesegerät für den hier zu entwickelnden Applikationskontroller sind die Reader der

Firma Scemtec ausgewählt worden. Neben den in [Kapitel 1.1] angesprochenen Transpondern, die der ISO 15693 Spezifikation entsprechen (siehe [Kapitel 3.6.1]), können von den Scemtec-Readern auch folgende Transponder ausgelesen werden:

- Philips I*Code, Hitag2
- Micron Hitag1
- Sokymat Unique, Titan, Nova
- Texas Instruments TagIt
- Temic e5530, e555x

Die Scemtec Baureihe SIR-2600 mit integrierter Antenne und SHL-2001 mit externer Antenne (Scemtec SAT-A6-LR-P 13,56 MHz) entsprechen den Anforderungen der in dieser Arbeit behandelten Applikation hinsichtlich Lesedistanz, Schnittstellen, Bauform und Technologie. In [Bild 6] sind beide Lesegeräte dargestellt. Der Reader wird beim Start der Applikation konfiguriert und danach auf Kommando angewiesen, die ID eines Transponders auszulesen und an den Applikationskontroller weiterzuleiten. Die Übermittlung soll über die gebräuchliche serielle Schnittstelle (RS-232, siehe [Kapitel 3.8]) erfolgen.

Bild 6 **Links: Scemtec SHL-2001 Long Range Reader mit externer Antenne. Rechts: Scemtec SIR-2600 Mid Range Reader mit interner Antenne.**

Die Auswahl der Scemtec-Reader wurde anhand der Anforderungen an die Applikation getroffen. Hierzu zählen:

- 13,56 MHz RFID Leser für Long Range Applikationen
- Kompatibilität zum ISO/IEC 15693 Standard
- Lesedistanz bis zu 1,2 m mit Antenne und bis 1,6 m mit Antennen-Tor
- Antikollision: Mehrere Transponder im Antennenfeld können simultan gelesen werden
- RS-232 Schnittstelle
- Auswahl verschiedener Antennenformen zum Anschluß
- Erhöhter, industrietauglicher Arbeitsbereich bzgl. Temperatur
- CE Zulassung und EMV-Freigabe (Elektromagnetische Verträglichkeit)

Darüber hinaus erfüllen die Scemtec-Reader folgende Eigenschaften, die in bezug auf eine zukünftige Erweiterung der Applikation sinnvoll sind:

- Synchronisation, um mehrere Reader im selben Antennengebiet zu betreiben
- Multiplexing mehrerer angeschlossener Antennen
- RS-485 Schnittstelle (Master/Slave Betrieb)
- Schreibender Zugriff auf Transponder, d.h. IDs können nicht nur ausgelesen, sondern zusätzliche Daten können auf dem Transponder gespeichert werden.

2.1.3 Omni Tracking Controller (OTC)

Das omnidirektionale Lesesystem ist ein technisch optimiertes Komplettsystem zur Identifizierung von Barcodes auf Gütern und Waren. Die Optimierung beruht auf der zuvor angesprochenen Technik, mehrere Barcodescanner als Tor zusammenzufassen. Der OTC ist die Vermittlungsstelle zwischen den einzelnen Barcodescannern und dem Host (RDT400). Der OTC liest den Barcode über einen oder mehrere, der im Tor verbauten Barcodescanner ein. Die gelesenen Barcode-Daten werden vom OTC aufbereitet und an den Host gesandt. Der OTC fungiert auch als Einheit zur Konfiguration einzelner Barcodescanner und des gesamten Tores.

Bild 7 **Der OTC (Omni Tracking Controller) von der Firma SICK AG. Die gelesenen Barcode-Informationen werden durch ihn an einen Host übertragen.**

[Bild 7] zeigt den OTC im laufenden Betrieb im Barcode/RFID-Testcenter der Firma SICK AG in Reute (Freiburg). Das Lesesystem OTC auf einen Blick:

- Verarbeitung der Lesetor- und Weginformationen (Tracking)
- Fokus- und Lesesteuerung der Barcodescanner
- Bewertung und Filterung der Einzel-Leseergebnisse
- Zuordnung der Barcode-Informationen zu den Objekten
- Kommunikation mit dem Host
- Bereitstellung von Schaltausgängen
- Führung einer Statistik
- Systemüberwachung
- Anzeige des Systemstatus

Der OTC wird innerhalb dieser Arbeit nicht benutzt. Er hat jedoch in der Welt des Barcodes die Funktionalität, die dem zu entwickelnden Applikationskontroller in der RFID-Welt entspricht. D.h. ein Lesegerät steuern, über ein Lesegerät ein Objekt identifizieren und das Ergebnis an den Host senden. Der Applikationskontroller soll die Daten vom Lesegerät so aufbereiten, daß der Host sie versteht. Host und Lesegerät sind ohne den Applikationskontroller inkompatibel.

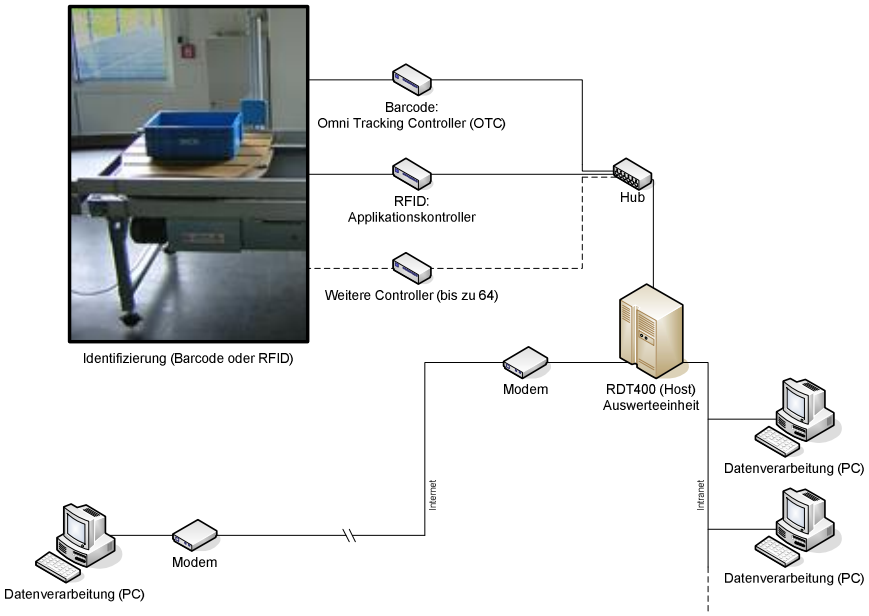

Bild 8 **Der Applikationskontroller fügt sich harmonisch in eine bestehende Topologie für automatische Objektidentifizierung (Auto-Ident) ein.**

Arbeitsstationen im Intranet eines Betriebes haben über die Auswerteeinheit RDT400 Zugriff auf Statistiken einzelner Identifizierungsstellen [Bild 8]. Auf den RDT400 hat man auch über das Internet Zugriff, da er den Ethernet Standard benutzt und als Webserver läuft. Eine Identifizierungsstelle kann mit Hilfe des zu entwickelnden Applikationskontrollers auch aus einem RFID-System bestehen. Insgesamt können am RDT400 bis zu 64 Barcode- oder RFID-Identifizierungsstellen angeschlossen werden.

2.2 Modulspezifikation

Die Anforderungen an den Applikationskontroller, der in dieser Arbeit zu entwickeln ist, sind:

- Parametrisierung Die Lichtschranken sollen einen Abstand zueinander besitzen, der größer als das zu erfassende Objekt ist. Die Entfernung zwischen Lichtschranke 1 und 2 soll einstellbar sein. Dadurch kann die Objektlänge und -geschwindigkeit ermittelt werden. Die Baudraten (Geschwindigkeit der Datenübermittlung) zur Kommunikation mit externen Geräten soll anpaßbar sein. Dem Applikationskontroller kann eine Gerätenummer (DeviceID) zugeordnet werden, sie wird zur Kommunikation zum Host verwendet.

- Technologie — Der Datenaustausch zwischen Transponder und Lesegerät soll mit der als "ISO 15693" bekannten Technologie [siehe Tabelle 1 auf Seite 30] realisiert werden.
- Verhalten — Das Verhalten gegenüber dem Host soll dem aus der Barcode-Welt bereits bekannten Omni Tracking Controller (OTC) [siehe 2.1.3] gleich sein. Die Kommunikation zum Host ist unidirektional. Die Kommunikation zum/vom Reader ist bidirektional. Bei der Pulkerfassung (mehrere Transponder im Lesefeld) sollen maximal 30 Transponder simultan verwaltet werden können.
- Protokolle — Zur Kommunikation zum RFID-Lesegerät soll das Scemtec STX/ETX Protokoll [Scem04] und zum Host der RDT400 Data String [siehe Anhang] verwendet werden. Die Konvertierung des STX/ETX Protokolls zum RDT400 Data String soll im Applikationskontroller stattfinden.
- Umfeld — Zwei Lichtschranken melden dem Applikationskontroller den Ein- und Austritt eines Objektes in/aus dem Lesefeld der Lesegerätantenne. Das Objekt fährt auf einem Fließband mit einer maximalen Geschwindigkeit von V_{MAX} = 5 m/s an der Lesegerätantenne vorbei.

2.3 Lösungsansatz

Im folgenden werden alle wichtigen Komponenten, die für die Durchführung des Projektes, bzw. den Bau eines Applikationskontroller benötigt werden, dargelegt:

Reader und Host sind zueinander inkompatibel, deshalb müssen die vom Lesegerät gelieferten IDs in das Protokoll des RDT400 konvertiert werden. Der Applikationskontroller übernimmt dies und sollte sich dafür zwischen den Reader und den Host schalten können. Der Reader und der Host haben eine RS-232 Schnittstelle, diese soll auch der Applikationskontroller zur Kommunikation mit beiden Geräten verwenden. Für die Konvertierung und Kommunikation braucht man einen Mikrocontroller, denn nur er ist in der Lage, die Anforderungen, bei gleichzeitig geringem Kostenaufwand, zu liefern. Darüber hinaus würde eine Mikrocontrollerlösung für geringen Platzverbrauch sorgen und er wäre zudem prädestiniert für den Anschluß von Sensoren. Diese Sensoren leiten den Erfassungsvorgang ein und sollten bewegte Objekte auf einem Fließband zuverlässig erkennen. Hierzu eignen sich Lichtschranken.

3 Grundlagen

3.1 Grundlegende Funktionsweise von RFID

In der Welt der RFID-Systeme gibt es zur Zeit viele unterschiedliche Techniken bezüglich Betriebsfrequenz, Reichweite, Energieversorgung, Bauform und Datenübertragung. Die Auswahl der geeigneten Technologie orientiert sich an den Anforderungen der gesamten Applikation. Sollen komplexe Datenstrukturen oder nur 1 Bit übertragen werden? Wie weit kann/soll der Transponder von der Antenne entfernt sein? Darf man im Umfeld der Anwendung mit Mikrowellen arbeiten? Für den Bereich der Automatisierungstechnik hat sich das 13,56 MHz Band als die geeignete Betriebsfrequenz erwiesen. Eine Zusammenfassung der bevorzugt eingesetzten Frequenzen hat das Fraunhofer Institut für Material und Logistik erstellt [Fhg04] und wird in [Bild 9] präsentiert. Die Norm 15693 [Wg899] der "International Organization for Standardization" (ISO) beschreibt die Energieversorgung, Datenübertragung und Abmessungen von passiven, kontaktlosen RFID Transpondern über lange Distanzen (Vicinity Chipkarten) im 13,56 MHz Band, wie sie in dieser Applikation verwendet werden.

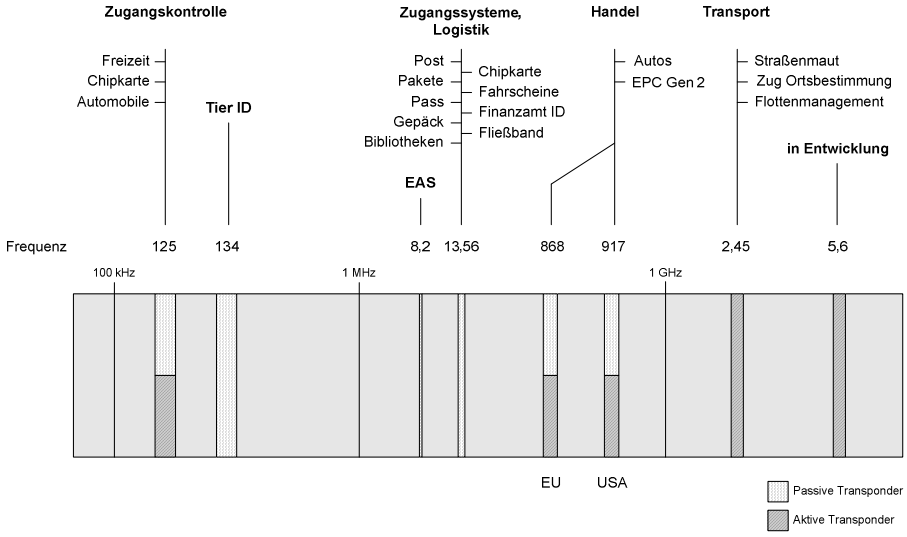

Bild 9 **RFID Systeme arbeiten in verschiedenen Frequenzbereichen (kHz, MHz, GHz). Die Arbeitsfrequenz ist das wichtigste Kriterium zur Bestimmung von Reichweite und Einsatzgebiet.**

In diesem Kapitel soll das grundsätzliche Zusammenwirken zwischen Transpondern und einem Lesegerät dargelegt werden, die gemäß der ISO 15693 Spezifikation arbeiten. Im ersten Abschnitt wird die Energieversorgung des Transponders besprochen, die auf dem physikalischen Gesetz der induktiven Kopplung beruht. Danach wird gezeigt, wie die Datenübertragung vom Lesegerät zum Transponder und in umgekehrter Richtung erfolgt. Dafür wird zuerst beschrieben, wie man Daten (Bits und Bytes) über die Luftschnittstelle auf das Trägerband (13,56 MHz) moduliert, das Transponder und Lesegerät "verbindet". Zum Schluß wird auf die Problematik der simultanen Pulkerfassung mehrerer Transponder näher eingegangen.

3.1.1 Induktive Kopplung

Induktiv gekoppelte Transponder werden fast ausschließlich passiv betrieben. Dies bedeutet, daß die gesamte zum Betrieb des Mikrochips notwendige Energie durch das Lesegerät zur Verfügung gestellt werden muß (bei aktiven Transpondern unterstützt eine Batterie den Betrieb). Da die Identifizierungsleistung jedoch nicht auf jene Objekte beschränkt sein darf, auf denen neben der Antenne und dem Mikrochip auch noch eine Batterie Platz und Gewicht benötigt, konzentrieren wir uns hier auf passive Transponder, auch wenn dadurch die Reichweite deutlich eingeschränkt ist (aktiv: 100 m, passiv: 1 m).

Von der Antennenspule des Lesegerätes wird für die induktive Kopplung zum Transponder ein hochfrequentes, elektromagnetisches Feld erzeugt, welches den Querschnitt der Spulenfläche und den Raum um die Spule durchdringt. Da die Wellenlänge λ des verwendeten Frequenzbereichs (13,56 MHz: 22,1 m) um ein Vielfaches größer ist als die Entfernung zwischen Leser-Antenne und Transponder, darf das elektromagnetische Feld im Abstand des Transponders zur Antenne mathematisch noch als einfaches magnetisches Wechselfeld H_L behandelt werden [Froh89]

Ein geringer Teil des ausgesendeten Feldes durchdringt die Antennenspule des Transponders, welcher sich in einiger Entfernung zur Spule des Lesegerätes befindet [Bild 10]. Durch Induktion wird dadurch an der Antennenspule des Transponders eine Spannung U_T erzeugt. Diese Spannung wird gleichgerichtet und dient der Energieversorgung des Datenträgers (Mikrochip). Der Antennenspule des Lesegerätes wird ein Kondensator C_L parallelgeschaltet; dessen Kapazität wird so gewählt, daß zusammen mit der Spuleninduktivität L_L der Antennenspule ein Parallelschwingkreis gebildet wird, dessen Resonanzfrequenz der Sendefrequenz des Lesegerätes entspricht. Durch Resonanzüberhöhung im Parallelschwingkreis werden in der Antennenspule

des Lesegerätes sehr hohe Ströme erreicht, womit die notwendigen Feldstärken auch zum Betrieb entfernter Transponder erzeugt werden können.

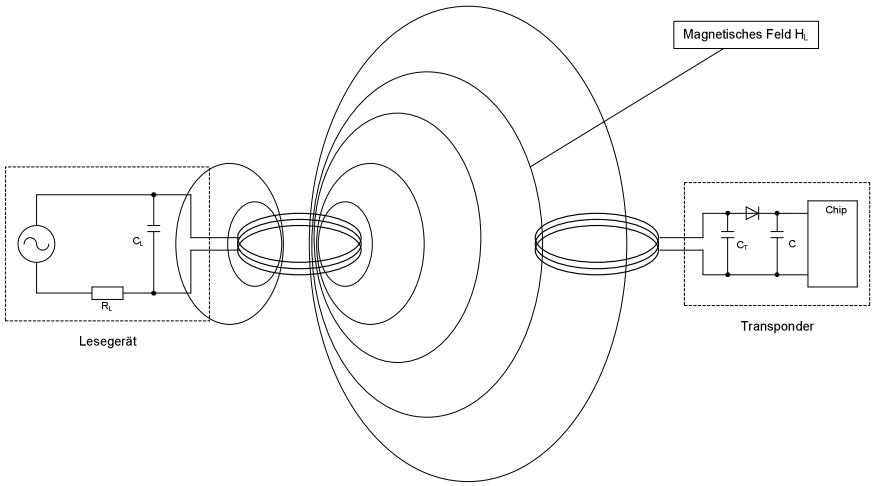

Bild 10 **Spannungsversorgung eines induktiv gekoppelten Transponders aus der Energie des magnetischen Wechselfeldes, das vom Lesegerät erzeugt wird.**

Die Antennenspule des Transponders bildet zusammen mit dem Kondensator C_T ebenfalls einen Schwingkreis, welcher auf die Sendefrequenz des Lesegerätes abgestimmt wird. Durch Resonanzüberhöhung im Parallelschwingkreis erreicht die Spannung U_T an der Transponderspule ein Maximum. Die zum Betrieb der Mikrochips minimal notwendige Versorgungsspannung liegt bei ca. 1,8 V, die Stromaufnahme bei ca. 1 mA (beide Angaben 13,56 MHz).

3.1.2 1 Bit / n Bit

Im Gegensatz zu 1 Bit Transpondern, welche meist durch die Anwendung einfacher physikalischer Effekte (LC-Schwingkreise, Flipflop) realisiert werden, verwenden die in der Objekt-Identifikation benutzten Transponder einen elektronischen Mikrochip als Datenträger. Auf diesem Datenträger können Datenmengen von mehreren kByte gespeichert werden. In dieser Arbeit werden nur (die ersten) 8 Byte benötigt, welche die eindeutige Zuordnung eines Transponders zu seinem Objekt garantieren [vgl. Kapitel 3.6.2]. 1 Bit Transponder werden seit den 60er Jahren beim Diebstahlschutz eingesetzt, z.B. auf CD-Hüllen im Supermarkt. Das Verfahren ist bekannt unter "EAS" - Electronic Article Surveillance (elektronische Artikelsicherung).

Als heute einfachste Variante verwendet man günstige Aufkleber (< 1 Cent) mit einem Schwingkreis, dessen Spule und Kondensator auf die jeweilige Resonanzfrequenz des Lesege-

räts abgeglichen sind, die meist bei 8,2 MHz liegt. Dabei fungiert die Spule als Antenne. Passiert man damit die Detektionsantennen am Ladenausgang, zieht der Schwingkreis durch Induktion Energie aus dem dazwischen bestehenden Wechselfeld ab. Diesen Energieschwund kann das mit den Detektionsantennen verbundene Lesegerät feststellen. Auf diese Weise gibt es die zwar unspezifische, aber völlig ausreichende Antwort „Hier ist ein nicht deaktiviertes Tag“. Bei gesetzestreuen Kunden hat die Kassiererin vorher den Transponder zerstört. Dazu wird durch ein starkes Magnetfeld eine so hohe Spannung induziert, daß der Folienkondensator durchschlägt. Damit ist der Schwingkreis irreversibel verstimmt, so daß der Transponder beim Durchqueren des Antennen-Tors schweigt.

Für Anwendungen der Objektidentifizierung sind 1 Bit Transponder nicht brauchbar, man benötigt mehr Informationen als 1 Bit, um mehrere Objekte voneinander zu unterscheiden. Es reicht also nicht aus, die Resonanzfrequenz durch Zerstörung des Kondensators auf dem Transponder zu verstimmen. Der Transponder sollte dem Lesegerät eine ganze Folge von Bits senden können, daher die Bezeichnung "n Bit". Das Lesegerät bestimmt dabei, wie und wann der Transponder senden soll. Ein Mikrochip nimmt auf dem Transponder Befehle vom Lesegerät entgegen und sendet gewünschte Daten zurück. Gleichzeitig wird der Mikrochip durch das magnetische Wechselfeld der Leserantenne auch mit Energie versogt. Der folgende Abschnitt beschreibt die dafür möglichen Verfahren.

3.1.3 Voll-, Halbduplex- und sequentielle Verfahren

Findet die Datenübertragung von Transponder in Richtung Lesegerät zeitversetzt mit der Datenübertragung vom Lesegerät zum Transponder statt, so bezeichnet man dies als Halbduplexverfahren (HDX). Erfolgt die Datenübertragung in beide Richtungen jedoch gleichzeitig, spricht man vom Vollduplexverfahren (FDX). Dabei kommen Verfahren zum Einsatz, bei denen die Daten des Transponders auf Teilfrequenzen des Lesegerätes, also einer subharmonischen, oder auf einer davon völlig unabhängigen Frequenz zum Lesegerät übertragen werden. Beiden Verfahren ist gemeinsam, daß die Energieübertragung vom Lesegerät zum Transponder kontinuierlich, also unabhängig von der Datenübertragungsrichtung stattfindet. Im Gegensatz dazu findet bei sequentiellen Systemen die Energieübertragung vom Transponder zum Lesegerät nur für eine begrenzte Zeitspanne statt (Pulsbetrieb). Die Datenübertragung vom Transponder zum Lesegerät wird in den Pausen zwischen der Energieversorgung des Transponders durchgeführt.

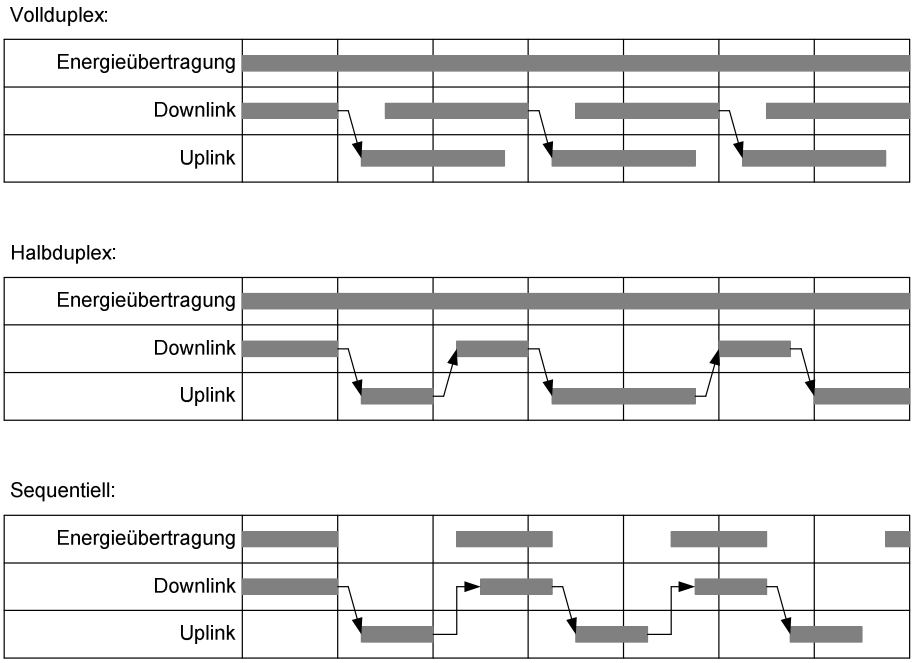

Bild 11 **Darstellung der zeitlichen Abläufe bei Voll-, Halbduplex- und sequentiellen Systemen. Die Datenübertragung vom Lesegerät zum Transponder wird in der Abbildung als downlink, die Datenübertragung vom Transponder zum Lesegerät als uplink bezeichnet.**

Oben stehendes Bild zeigt drei Verfahren, wie Daten zwischen Transponder und Lesegerät übertragen werden können. Das für diese Arbeit relevante Verfahren ist das Halbduplexverfahren und ergibt sich aus Teil 2 der ISO 15693 Norm. [Wg899]

3.2 Modulation

Wie in Kapitel [3.1.1, induktive Kopplung] gezeigt, wird durch elektromagnetische Wellen Energie von einer Antenne in den umgebenden Raum abgestrahlt. Durch gezielte Beeinflussung einer der drei Signalparameter - Leistung, Frequenz und Phasenlage - einer elektromagnetischen Welle können Nachrichten codiert und dadurch an jeden Punkt im Raum transportiert werden. Den Vorgang der Beeinflussung einer elektromagnetischen Welle durch Nachrichten (Daten) nennt man Modulation, eine unmodulierte elektromagnetische Welle wird als Träger (Carrier) bezeichnet. Untersucht man die Eigenschaften einer elektromagnetischen Welle an einem beliebigen Punkt im Raum, so kann man aus den Änderungen der empfangenen Leistung, Frequenz oder Phasenlage die der Welle aufgeprägte Nachricht rekonstruieren. Dieser Vorgang heißt Demodulation.

Im folgenden wird die Änderung der Amplitude einer Welle (Amplitude shift keying, ASK) genauer betrachtet, da sie bei 13,56 MHz Systemen vorzugsweise eingesetzt wird. Amplitude ist die physikalische Bezeichnung für die maximale Auslenkung einer Schwingung bzw. einer Welle aus der Mittellage heraus ("Höhe einer Welle"). Als Transponder werden "Vicinity-Chipkarten" [siehe Kapitel 3.6.1] benutzt. Neben Amplitudenmodulation kann auch Frequenzmodulation (FSK) oder Phasenmodulation zur Datenübertragung in RFID-Systemen zum Einsatz kommen. Eine nähere Beschreibung findet sich in [Fink02].

3.3 Datenübertragung Lesegerät → Transponder

Zur Datenübertragung von einem Lesegerät zu einer Vicinity-Chipkarte als Transponder kommt sowohl 10% ASK als auch 100% ASK-Modulation zum Einsatz. Die Prozentangabe definiert den Modulationsgrad (Verhältnis) der Auslenkung einer Amplitude zur Darstellung einer 0 (Null) gegenüber ihrer normalen Auslenkung zur Darstellung einer 1 (Eins), siehe [Bild 12]. Unabhängig vom gewählten Modulationsgrad kann zudem zwischen zwei verschiedenen Codierverfahren, einem "1 aus 265" sowie einem "1 aus 4"-Code ausgewählt werden. Eine Vicinity-Chipkarte muß dabei grundsätzlich beide Modulations- und Codierverfahren unterstützen. Die Auswahl der Verfahren zur Modulation und Codierung muß im Lesegerät eingestellt werden, nach dieser Einstellung richtet sich dann auch der Transponder.

3.3.1 Amplitudenmodulation

Das zu sendende binäre Codesignal besteht aus einer Folge von Nullen und Einsen (Bits). Bei der Modulation wird die Amplitude einer Trägerschwingung durch das Codesignal in zwei Zustände, z_{LOW} und z_{HIGH}, umgeschaltet. z_{LOW} kann dabei Werte zwischen z_{HIGH} und 0 annehmen. Das Verhältnis zwischen z_{LOW} und z_{HIGH} wird als *Tastgrad* bezeichnet, siehe [Bild 9]. Bei 100% ASK besitzt die Trägeramplitude für z_{LOW} den Wert 0.

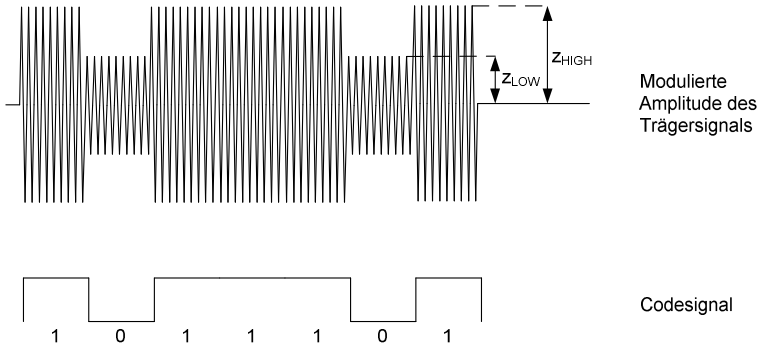

Bild 12 **Das Codesignal wird dem Träger aufmoduliert.**

Aus der empfangenen elektromagnetischen Welle, die aus Trägerschwingung und aufmoduliertem Codesignal besteht, wird das Codesignal mit Hilfe eines Bandpasses herausgefiltert und anschließend gleichgerichtet. Das resultierende Signal entspricht dem gesendeten Code.

3.3.2 Bitcodierung

"1 aus 256":
Bei diesem Codierverfahren handelt es sich um eine Puls Positions Modulation (PPM). Dies bedeutet, daß die Wertigkeit des zu übertragenden Zeichens im Wertebereich 0 bis 255 durch die zeitliche Lage eines Modulationspulses eindeutig definiert wird. Es können damit in einem Schritt 8 Bit (1 Byte) gleichzeitig übertragen werden. Die gesamte Übertragungsdauer eines Bytes beträgt 4,833 ms, daraus resultiert eine Datenrate von 1.650 Bit/s [Wg899]. Beginn und Ende einer Datenübertragung werden durch definierte Rahmensignale (Start of frame / End of frame) gekennzeichnet.

"1 aus 4":
Auch bei dieser Codierung bestimmt die zeitliche Lage eines Modulationspulses die Wertigkeit eines Zeichens. In einem Schritt werden dabei 2 Bit gleichzeitig übertragen, die Wertigkeit des zu übertragenen Zeichens liegt also im Wertebereich von 0 bis 3. Ein Byte wird in 75,52 µs übertragen, das entspricht einer Datenrate von 26.480 Bit/s (13,56 MHz / 512).

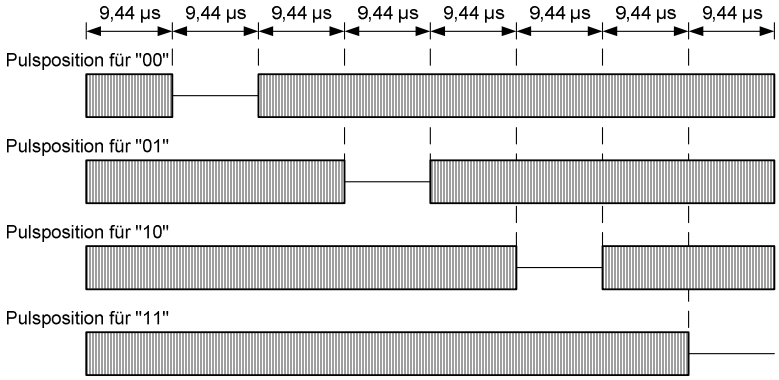

Bild 13 **Die "1 aus 4"-Codierung entsteht aus Aneinanderreihung von 8 Zeitabschnitten von je 9,44 µs Länge. Aus der zeitlichen Position eines Modulationspulses kann die Wertigkeit des zu übertragenen Zeichens ermittelt werden. Wie beim "1 aus 265"-Code kann dabei ein Modulationspuls nur zu einem ungeradzahligen Zeitabschnitt (1, 3, 5, …) auftreten. In den geradzahligen Zeitabschnitten werden Steuerzeichen übertragen (Start of frame / End of frame).**

Für eine Applikation, wie sie in dieser Arbeit beschrieben wird, sollte die Kombination 10% ASK mit "1 aus 4"-Code gewählt werden. Dadurch wird die Energieversorgung des Transponders auch bei großer Distanz zur Leserantenne gewährleistet und die ID kann trotzdem schnell ausgelesen werden. Die bei dieser Kombination, im Vergleich zur Feldstärke des 13,56 MHz Trägersignals niedrige Feldstärke der Modulationsseitenbänder, erlaubt die volle Ausnutzung der zulässigen magnetischen Feldstärke zur Energieversorgung des Transponders [Fink02].

3.4 Datenübertragung Transponder → Lesegerät

3.4.1 Lastmodulation

Bei Vicinity-Coupling-Systemen bedient man sich zur Datenübertragung vom Transponder zum Lesegerät der sogenannten Lastmodulation. Dies ist möglich, sofern sich der Transponder im Nahfeld (0,16 λ = 3,53 m) der Sendeantenne vom Lesegerät befindet. In diesem Falle spricht man von einer transformatorischen Kopplung. Die Wellenlänge λ ergibt sich bei 13,56 MHz Systemen aus [Gert95]:

Wellenlänge λ = Lichtgeschwindigkeit c / Frequenz f

λ = 299.792.458 m/s / 13.560.000 Hz

λ = 22,11 m

Bei einer solchen Kopplung wird dem magnetischen Wechselfeld der Antenne des Lesegerätes Energie durch den Transponder entzogen. Letzterer ist aus Sicht des Lesegerätes somit als "Last" anzusehen, welche als transformierte Impedanz Z_T bezeichnet wird. Eine Veränderung der Impedanz Z_T bewirkt eine Spannungsänderung ΔU_L an der Antenne des Lesegerätes. Durch Einfügen eines schaltbaren Lastwiderstandes an der Antenne des Transponders ist die Spannung am Lesegerät somit beeinflußbar und entspricht in ihrer Wirkung einer Amplitudenmodulation. Durch Ein- und Ausschalten des Lastwiderstandes mittels Daten können diese Daten vom Transponder zum Lesegerät übertragen werden.

Die Detektion eines Nutzsignals nach diesem Verfahren der Lastmodulation birgt jedoch Schwierigkeiten. Durch die geringe Kopplung bewegen sich die Spannungsänderungen am Lesegerät im Millivoltbereich, gleichzeitig können an der Antenne durch Resonanzüberhöhung Spannungen von ca. 100 V entstehen. Dies macht eine Detektion nur mit großem schaltungstechnischen Aufwand möglich. Einfacher ist es daher, sich die durch Amplitudenmodulation auftretenden Modulationsseitenbänder zunutze zu machen.

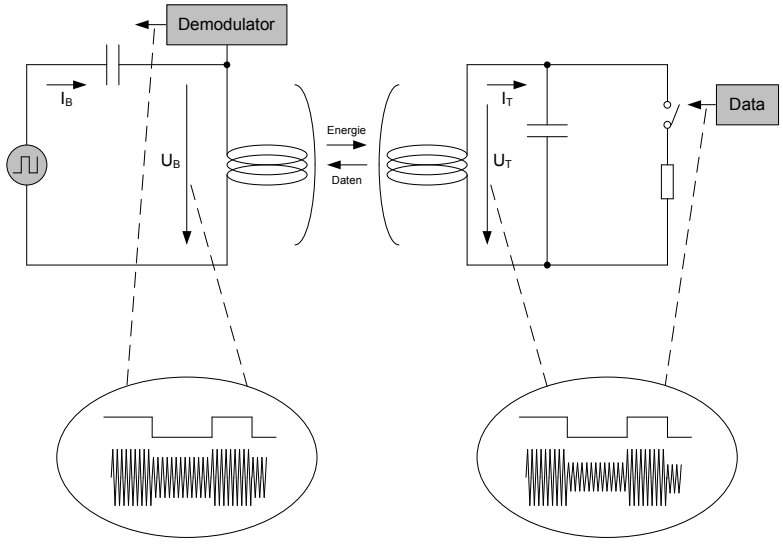

Bild 14 **Datenübertragung mit Hilfe des Prinzips des lose gekoppelten Transformators vom Transponder zum Lesegerät. In Abhängigkeit der Datenbits wird eine Last im Transponder gesteuert (Lastmodulation). Durch Änderung der Last auf dem Transponder erhöht sich auch der Strom im Lesegerät, so daß hier eine entsprechende Modulation erfaßt werden kann. Allerdings ist das Nutzsignal gegenüber dem Sendesignal sehr gering. Deshalb verwendet man einen Hilfsträger, der die Demodulation vereinfacht. [Elin04]**

3.4.2 Lastmodulation mit Hilfsträger

Aufgrund der geringen Kopplung zwischen Leseantenne und Transponderantenne sind die das Nutzsignal darstellenden Spannungsschwankungen an der Antenne des Lesegerätes um Größenordnungen kleiner als die Ausgangsspannung des Lesegerätes. Bei einem 13,56 MHz-System kann in der Praxis bei einer Antennenspannung von ca. 100 V (Spannungsüberhöhung durch Resonanz) mit einem Nutzsignal von etwa 10 mV gerechnet werden (= 80 dB Nutz/Störsignal-Verhältnis) [Fink02]. Da diese geringen Spannungsänderungen nur mit einem sehr großen schaltungstechnischen Aufwand zu detektieren sind, macht man sich die durch die Amplitudenmodulation der Antennenspannung entstehenden Modulationsseitenbänder zunutze: Wird der zusätzliche Lastwiderstand im Transponder mit sehr hoher Taktfrequenz f_H ein- und ausgeschaltet, so entstehen zwei Seitenbänder im Abstand $\pm f_H$ um die Sendefrequenz des Lesegerätes [Bild 15], die leicht detektiert werden können (es muß jedoch $f_H < f_{LESER}$ gelten). Im Sprachgebrauch der Funktechnik wird die zusätzlich eingeführte Taktfrequenz als Hilfsträger (Subcarrier) bezeichnet. Die Datenübertragung erfolgt durch Amplitudenmodulation des Hilfsträgers im Takt des Datenflusses.

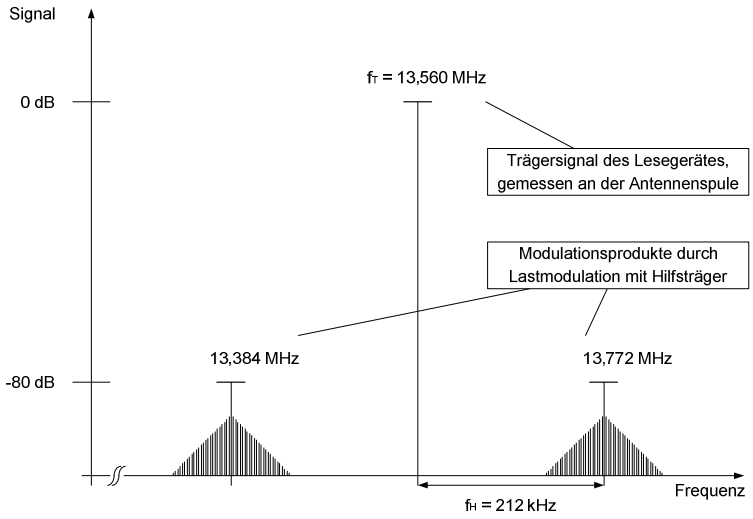

Bild 15 **Durch Lastmodulation mit Hilfsträger entstehen zwei Seitenbänder im Abstand der Hilfsträgerfrequenz f_H um die Sendefrequenz des Lesegerätes. Die eigentliche Information steckt in den Seitenbändern der beiden Hilfsträger-Seitenbänder, welche durch die Modulation selbst hervorgerufen wird.**

Lastmodulation mit Hilfsträger evoziert an der Antenne des Lesegerätes zwei Modulationsseitenbänder im Abstand der Hilfsträgerfrequenz um die Arbeitsfrequenz f_{LESER}. Diese Modulationsseitenbänder können durch eine Bandpaßfilterung (BP) auf einer der beiden Frequenzen

$f_{LESER} + f_H$ oder $f_{LESER} - f_H$, vom wesentlich stärkeren Signal des Lesegerätes getrennt werden. Nach anschließender Verstärkung ist das Hilfsträgersignal einfach zu demodulieren.

3.5 Übersicht Datenübertragung

Die folgende Übersicht faßt noch einmal die Möglichkeiten der Datenübertragung (Up- und Downlink) eines ISO 15693 RFID-Systems zusammen:

Parameter	Wert	Bemerkung
Energieversorgung	13,56 MHz ± 7 kHz	induktive Kopplung
Datenübertragung Leser → Karte		
Modulation	10% ASK, 100% ASK	Karte unterstützt beide
Bitcodierung	Long distance mode: "1 aus 256“ Fast mode: "1 aus 4“	Karte unterstützt beide
Baudrate	Long distance mode: 1,65 kBit/s Fast mode: 26,48 kBit/s	
Datenübertragung Karte → Leser		
Modulation	Lastmodulation mit Hilfsträger	
Bitcodierung	Manchester, Hilfsträger wird ASK (423 kHz) oder FSK (423/485 kHz) moduliert	
Baudrate	Long distance mode: 6,62 kBit/s Fast mode: 26,48 kBit/s	vom Lesegerät selektiert

Tabelle 1 **Modulations- und Codierverfahren bei ISO 15693. [Berg98]**

Neben der Norm für passive, kontaktlose RFID Transponder über lange Distanz (Vicinity Chipkarten, ISO 15693) gibt es auch Normen für kurze (Close-coupled Chipkarten, ISO 10536) und mittlere Reichweiten (Proximity Chipkarten, ISO 14443).

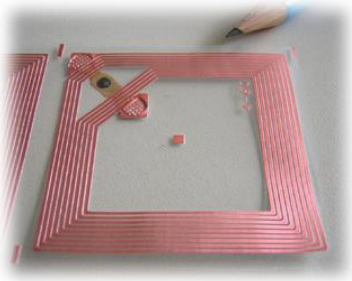

Bild 16 **ISO 15693 Transponder als Halbzeug auf Folie. Die Antenne hat 9 Windungen. Oben links sitzt der Mikrochip (schwarzer Fleck).**

3.6 Kollisionserkennung bei Pulkerfassung

Die technische Realisierung eines Vielfachzugriffs bei RFID-Systemen (mehrere Transponder versuchen "gleichzeitig" Daten an das Lesegerät zu senden) stellt einige Anforderungen an Transponder und Lesegerät. Es muß ohne spürbaren Zeitaufwand zuverlässig verhindert werden, daß die Daten der Transponder im Empfänger des Lesegerätes miteinander kollidieren und dadurch unlesbar werden. Ein technisches Verfahren (Protokoll), welches die störungsfreie Abwicklung eines Vielfachzugriffs ermöglicht, wird im Zusammenhang mit RFID als Antikollisionsverfahren (anticollision) bezeichnet.

3.6.1 ISO/IEC 15693 Anti-Kollisions- und Übertragungsprotokoll

Die "International Organization for Standardization" (ISO) und "International Electrotechnical Commission" (IEC) bilden die internationale Vereinigung der Standardisierungsgremien von 148 Ländern. Sie erarbeiten internationale Vorgaben in allen technischen Bereichen. Die Norm 15693 Teil 3 [Wg899] beschreibt das Protokoll der Anti-Kollision und Datenübertragung von kontaktlosen RFID Transpondern über lange Distanz (Vicinity Chipkarte), wie sie in dieser Applikation Verwendung findet.

3.6.2 Unique identifier (UID)

Transponder sind eindeutig identifizierbar durch einen 64 Bit "unique identifier" (UID). Mit ihm wird ein Transponder während der Anti-Kollisionsschleife und dem one-to-one Datenaustausch zwischen Transponder und Lesegerät einzeln und individuell adressiert. Die UID wird vom Transponderhersteller gemäß [Bild 17] permanent in den Mikrochip gesetzt.

MSB			LSB
64 57	56 49	48	1
Hex E0	Hersteller Code	Hersteller Serial Nr.	

Bild 17 **UID Format.**

- Die acht "Most significant bits" (MSB) enthalten Hex E0.
- Es folgt ein acht Bit "IC manufactor code", der durch ISO/IEC 7816-6/AM1 beschrieben ist. Er identifiziert den Transponderhersteller eindeutig.

- Die 48 "Least significant bits" (LSB) ergeben eine serielle Nummer und werden vom Transponderhersteller vergeben. 2^{48} Bits ergeben 281.474.976.710.656 Permutationen, die für eine weltweit eindeutige Identifizierung einer Vicinity-Chipkarte ausreichend sein dürfte.

Neben der UID enthält ein ISO 15693 Transponder:

- Der "Application family identifier" (AFI) repräsentiert den Typ der Anwendung, auf den ein RFID-System abzielt. Er wird zur Extraktion der im Lesefeld befindlichen Transponder genutzt. Der AFI ist ein Byte lang; es werden 16 Typen unterschieden (Transport, Medizin, Paketversand, Fluggepäck u.a.). Der AFI steht im Festspeicher des Transponders hinter der UID.
- Der "Data storage format identifier" (DSFID) beschreibt, wie Daten im Transponder strukturiert sind. Der ISO 15693 Standard unterstützt maximal 256 Seiten mit je 32 Byte, d.h. 8.192 ASCII-Zeichen. Der DSFID muß vom Lesegerät ausgewertet werden, um Daten neben dem UID und AFI lesen bzw. schreiben zu können.

3.6.3 Übertragungsprotokoll

Das Übertragungsprotokoll definiert den Mechanismus des bidirektionalen Austauschs von Instruktionen und Daten zwischen Reader und Transponder. Es basiert auf dem Konzept "Reader talks first", d.h. ein Transponder verhält sich still und sendet nichts, bis der Reader ihm die Aufforderung dazu erteilt. Request und Response sind jeweils gerahmt in "Start of frame" (SOF) und "End of frame" (EOF).

- Jeder Request enthält: Flags, Befehl, Befehls-Paramter, Daten, CRC
- Jeder Response enthält: Flags, optionale Parameter, Daten, CRC

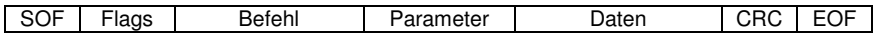

Bild 18 **Request Format.**

Das Feld "Flags" spezifiziert, wie der Transponder den Befehl ausführen soll und ob korrespondierende Parameter im Request vorhanden sind oder nicht.

Der "Cyclic redundancy check" (CRC) bezeichnet ein Prüfsummenverfahren, daß die Richtigkeit übertragener Daten sicherstellt. Er ist definiert in der Norm ISO/IEC 13239 und muß im Transponder sowie im Lesegerät implementiert sein.

Die Protokoll-Logik eines Transponders wird als Zustandsautomat implementiert. Die Zustände können sein:

1. Power-off state	Ein Transponder ist im Power-off Zustand, wenn er nicht vom Reader aktiviert werden kann.
2. Ready state	Wurde der Transponder vom Reader aktiviert, befindet er sich im Ready Zustand.
3. Quiet state	Requests werden nicht bearbeitet, bis der Transponder vom Reader adressiert wurde oder ein "Resert to ready" auftritt.
4. Selected State	Nur in diesem Zustand darf der Transponder auf Requests antworten.

Die Zustandsänderung des Automaten über Transitionen soll sicherstellen, daß sich jeweils nur ein Transponder zur Zeit im Zustand Selected befindet. Tritt ein Fehler auf (z.B. CRC-Fehler) verbleibt der Transponder in seinem momentanen Zustand.

3.6.4 Anti-Kollisionssequenz

Der Zweck der Anti-Kollisionssequenz besteht darin, ein Inventar aller im Lesefeld des Readers befindlichen Transponder anhand ihrer unique ID (UID) zu erstellen. Der Reader ist der Hauptkommunikator mit einem oder mehreren Transpondern. Er initiiert die Kommunikation durch Anforderung einer Inventur-Anfrage (Inventory request). Ein Transponder soll seinen Response in einem definierten Zeitfenster (Slot) senden oder nicht antworten, entsprechend dem unten beschriebenen Algorithmus. Wenn keine Transponder-Antwort erkannt wird, wechselt der Reader zum nächsten Slot durch Senden eines EOF. Wenn eine oder mehrere Transponder-Antworten erkannt werden, wartet der Reader bis alle Response-Rahmen komplett empfangen wurden, bevor er ein EOF sendet, um zum nächsten Slot zu wechseln. [Auto03]

```
// UID_Mask:         ID, mit der sich der Transponder selbst vergleichen soll.
// UID_Mask_Length:  Anzahl der Bits, für die UID_Mask gültig ist.
//                   Wenn 0: keine UID_Mask vorhanden.
// Slot_Frame:       Vom Lesegerät gesendetes, aktuelles Zeitfenster.
// LSB(wert, n):     Funktion liefert die n niederwertistgen Bits von wert

function Request_Process(UID_Mask, UID_Mask_Length, Slot_Frame)
   if (Slot_Frame == SOF) or (Slot_Frame == EOF)
      for (i=0; i<Total_Number_Of_Slots; i++)
         if LSB(myUID, UID_Mask_Length + i) == concat(i, LSB(UID_Mask, UID_Mask_Length))
         // transmit response to inventory request
         wait(Slot_Frame++)
         if (Slot_Frame == EOF)
            Return
```

Beispiel eines vom Transponder ausgeführten Inventory-Algorithmus. [Wg899]

3.6.5 Bitcodierung

Die in der ISO 15693 Norm definierte Anti-Kollisionssequenz wird im Reader als Binary-Search Algorithmus implementiert. Dies setzt die Notwendigkeit voraus, im Lesegerät die genaue Bitposition einer Datenkollision zu erkennen. Hierzu benötigt man eine geeignete Bitcodierung, weshalb hier zunächst das Kollisionsverhalten von NRZ- und Manchester-Codierung miteinander verglichen werden soll.

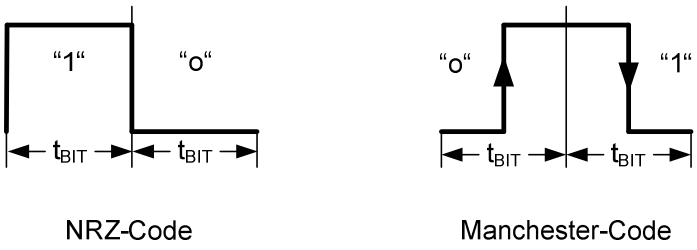

Bild 19 Bitcodierung im NRZ- und Manchester-Code.

NRZ-Code ("Non return to zero" Code): Die Wertigkeit eines Bits ist durch den statischen Pegel des Übertragungskanals innerhalb eines Bitfensters (t_{BIT}) definiert. In [Bild 19] wird eine logische 1 durch einen statischen high-Pegel, eine logische 0 durch einen statischen low-Pegel codiert. Sendet mindestens einer der Transponder eine 1, so wird dies vom Lesegerät als high-Pegel interpretiert und als logische 1 gewertet. Das Lesegerät kann nicht feststellen, ob die eingehende Bitfolge auf das Signal eines einzelnen Transponders oder auf die überlagerte Aussendung

mehrerer Transponder zurückzuführen ist. Die Verwendung einer Blockprüfsumme (Parity, CRC) ermöglicht lediglich die Feststellung eines Übertragungsfehlers "irgendwo" im Datenblock.

Manchester-Code: Die Wertigkeit eines Bits wird durch Pegelwechsel (positive/negative Flanke) innerhalb eines Bitfensters (t_{BIT}) definiert. Eine logische 0 ist durch eine positive Flanke, eine logische 1 durch eine negative Flanke codiert [Bild 19]. Der Zustand "keine Flanke" während der Datenübertragung ist nicht zulässig und wird als Fehler erkannt. Senden zwei (oder mehr) Transponder gleichzeitig Bits unterschiedlicher Wertigkeit, dann heben sich die positive und die negative Flanke der empfangenen Bits gegenseitig auf, so daß im Empfänger während einer ganzen Bitdauer nur das Hilfsträgersignal (423 kHz) empfangen wird. Dieser Zustand ist bei der Manchester-Codierung nicht definiert und führt zu einem Fehler. Auf diese Weise kann das Auftreten einer Kollision bitweise zurückverfolgt werden.

Durch Bitcodierung im Manchester-Code kann eine Kollision während der Datenübertragung von Transpondern zum Lesegerät bitgenau festgestellt werden [Bild 20]. Die Datenübertragung stellt eine sequentielle Folge einzelner Bits dar (Bitfolge). Die bis zur Kollision erfolgreich empfangene Bitfolge wird im Folgenden *UID_Mask* genannt (siehe auch "Inventory-Algorithmus", oben).

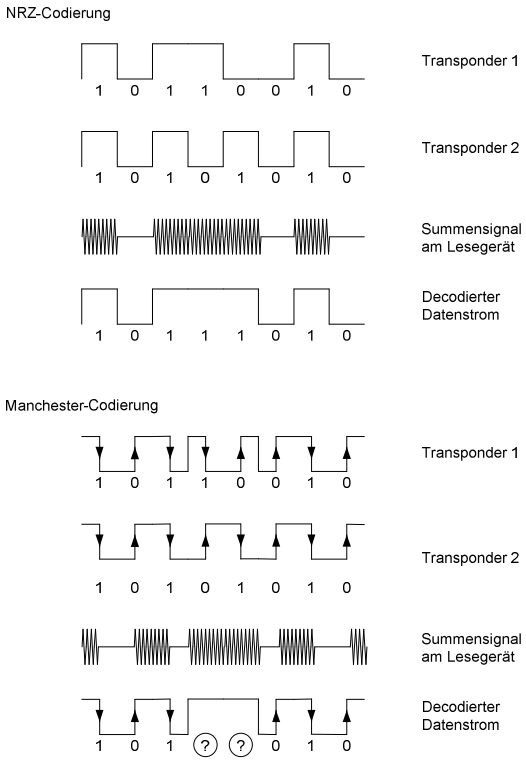

Bild 20 **Kollisionsverhalten mit NRZ- und Manchester-Code. Der Manchester-Code ermöglicht die bitweise Erkennung einer Kollision.**

3.6.6 Implementierung

Zur Implementierung des Binary-Search Algorithmus wird die hierzu geeignete Manchester-Codierung verwendet. Der Algorithmus besteht aus einer Abfolge von Request/Response Nachrichten zwischen einem Lesegerät und mehreren Transpondern im Lesefeld mit dem Ziel, einen beliebigen Transponder aus einer Gruppe auswählen zu können.

Binärer Suchbaum als Grundlage des Binary-Search Algorithmus:
Ein binärer Suchbaum ist ein Binärbaum (jeder Knoten besitzt höchstens zwei Kindknoten), der die Datenelemente in sortierter Reihenfolge enthält [Bild 21]. Die Sortierung ist dadurch definiert, daß ausgehend von einem beliebigen Baumknoten alle Datenschlüssel im linken Teilbaum kleiner als der Schlüssel des Knotens sind und alle Schlüssel des rechten Teilbaums größer sind [Schn04]. Ausgeglichene binäre Suchbäume (hier gegeben) ermöglichen eine garantierte effiziente Suche in *O*(log n). Der Baum kann so durchlaufen werden, daß die Datenelemente in sortierter Reihenfolge bearbeitet werden (Inorder-Travesierung).

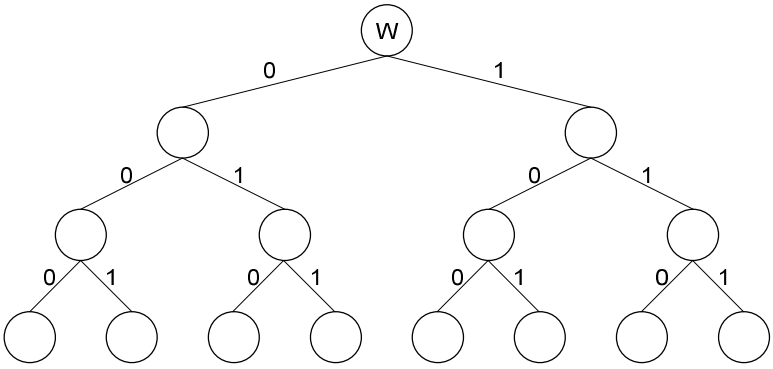

Bild 21 **Beispiel eines ausgeglichenen binären Suchbaumes mit der Höhe h = 3. Durch Traversierung links, rechts, rechts, beginnend an der Wuzel "W", erhält man die Bitfolge "011".**

Im folgenden wird der Ablauf des Binary-Search Algorithmus erklärt.
Das Lesegerät sendet im Request-Kommando nach einer festgestellten Kollision nur den bereits bekannten Teil (UID_Mask), der zu ermittelnden UID, als Suchkriterium. Alle Transponder, deren UID bis zum Bit UID_Mask_Length (auch im Request-Kommando enthalten) mit dem Suchkriterium übereinstimmen, antworten nun mit der Übertragung der restlichen Bits ihrer UID (vgl. Inventory-Algorithmus, oben). Das Zeitfenster (1 aus 16 Slot_Frame's), in dem der Transponder senden darf, ergibt sich aus den vier nachfolgenden Bits seiner UID ab UID_Mask_Length. Im besten Fall weisen alle im Pulk befindlichen Transponder unterschiedliche "nächste vier Bits" auf, ihre Sendezeitfenster (Slots) unterscheiden sich und es kommt zu keiner Kollision. Kollidieren die Antworten der Transponder dennoch, wird die UID_Mask vom

Reader um ein Bit verlängert und die Position sowie Wertigkeit des neuen Bits vermerkt, d.h. der entsprechende Knoten im binären Suchbaum markiert. Eine neue Runde beginnt, das Lesegerät sendet ein neues Request-Kommando mit der neuen UID_Mask und UID_Mask_Length. Nach erfolgreicher Selektion eines Transponders kann dieser, ungestört durch andere Transponder, vom Lesegerät ausgelesen oder beschrieben werden. Nach Abwicklung der Schreib-Leseoperation wird der selektierte Transponder in den Zustand 'Quiet state' gebracht (vgl. 4.5.3), wodurch er auf nachfolgende Request-Kommandos nicht mehr antwortet. Jetzt wird der konträre Ast im binären Suchbaum [Bild 21] ab der Stelle traversiert, an der die letzte Kollision auftrat.

Die durchschnittliche Anzahl an Iterationen L, die benötigt wird, um einen einzelnen Transponder aus einer größeren Menge zu ermitteln, hängt von der Gesamtzahl N der Transponder im Ansprechfeld des Lesegerätes ab und wird ermittelt durch:

$$L(N) = ld(N) + 1 = \frac{\log(N)}{\log(2)} + 1 \qquad \text{[Fink02]}$$

Die für die Pulkerfassung von Transpondern geforderten Merkmale der Kollisionserkennung und -auflösung wird bei ISO 15693 vollständig vom Lesegerät ausgeführt. Die Steuerungsform ist zentralisiert und in Abhängigkeit der Pulkgröße deterministisch. Die oben beschriebenen Methoden unterscheiden sich von den bekannten Medienzugriffsverfahren für Ethernet, CAN oder I^2C. Das Medium *Luftschnittstelle* kann nicht nach CSMA/CD zugeteilt werden, da die Transponder bzgl. der Schaltkomplexität stark begrenzt sind. Ausschlaggebend für die sichere Funktion des Binary-Tree-Suchalgorithmus ist die Synchronisation aller Transponder, so daß diese exakt zum gleichen Zeitpunkt mit der Übertragung ihrer Seriennummer beginnen (Lesegerät sendet ein SOF). Nur auf diese Weise ist die bitweise Bestimmung einer Kollision überhaupt möglich.

3.7 Mikrocontroller

Der zu entwickelnde Applikationskontroller soll Objekte identifizieren, die an der Antenne eines RFID-Lesegeräts vorbeifahren. Dazu sind auf den Objekten Transponder montiert. Der Applikationskontroller steuert das Lesegerät und sendet nach der Identifizierung entsprechende Daten an den RDT400. Für die Realisierung der Aufgaben "Steuerung" und "Kommunikation" eignet

sich der Einsatz eines Mikrocontrollers (vgl. Kapitel 2.3, Lösungsansatz). Dieser Abschnitt beschreibt die Grundlagen einer solchen Einheit.

Der Mikrocontroller beinhaltet auf einem Chip einen kompletten Kleinst-Computer. Auf dem Chip befinden sich die CPU, ein ROM-Speicher für das Programm, ein RAM-Speicher für variable Daten sowie parallele und serielle Ein- und Ausgabeports. Die CPU ist über ein internes Bussystem mit dem Speicher und den Schnittstellen-Baugruppen verbunden. Der Mikrocontroller wird hauptsächlich im Bereich der Automatisierungs- und Steuerungstechnik eingesetzt, wie in dieser Applikation. Speziell für derartige Anwendungsgebiete sind außer den aufgeführten Standard-Baugruppen noch eine Reihe zusätzlicher Funktionseinheiten in den Mikrocontroller integriert. Solche Funktionseinheiten sind z.B. schnelle Zähler (Timer), A/D-Wandler (Analog zu Digital Wandler) oder Interrupt-Controller. [Bild 22] zeigt den in dieser Arbeit verwendeten Mikrocontroller (SMD Bauweise).

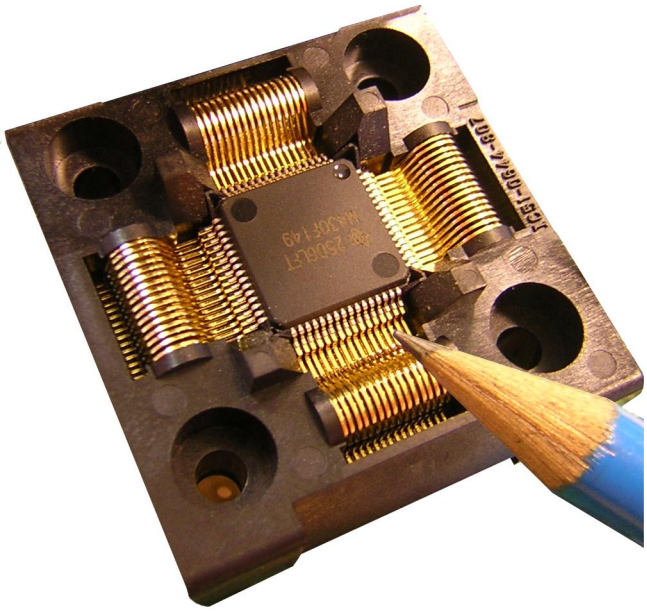

Bild 22 **Der Mikrocontroller MSP430F149 von Texas Instruments im Sockel. Durch den Sockel wird die Kontaktierung seiner 64 Pins auf eine Platine erleichtert.**

Die oben erwähnte Aufgabe "Steuerung" beinhaltet auch die Auswertung der Signale, die von den Sensoren geliefert werden. Dies soll so schnell wie möglich erfolgen, deshalb wird die Funktionalität des im Mikrocontroller eingebauten Interrupt-Controllers genutzt:

Die Funktion des Mikrocontrollers wird durch das Programm, welches er bearbeitet, vorgegeben. Bei Steuerungen zur Automatisierung von Maschinen oder Prozessen erfolgt die Programmbearbeitung zyklisch. Die Reaktionszeit auf Eingangssignale beträgt also maximal

grammbearbeitung zyklisch. Die Reaktionszeit auf Eingangssignale beträgt also maximal eine Zykluszeit und liegt damit im Millisekundenbereich. Bei dieser Art der Programmierung werden Eingangssignale oder interne Zustände erst registriert, wenn sie durch das Programm abgefragt werden. Diese Methode wird Polling genannt. Soll die Reaktionszeit auf ein Eingangssignal verringert werden, muß die Möglichkeit bestehen, den Mikrocontroller während seines laufenden Programms zu unterbrechen, um auf das Ereignis (hier: Sensor-Signale) sofort zu reagieren. Solch eine Unterbrechungsanforderung nennt man Interrupt. In dieser Applikation werden die Lichtschranken solche Interrupts auslösen, um sofort mit der Erfassung der Transponder-IDs zu starten. Auf den Interrupt kann dann der Mikrocontroller mit der Interrupt-Service-Routine reagieren. Der Interrupt wird bei externen Ereignissen über Porteingänge ausgelöst und setzt dabei ein bestimmtes Bit (Flag) im Spezial-Funktions-Register. Die interne Interrupt-Steuerung (Interrupt-Controller) fragt jedes Flag ab, ob es gesetzt ist. Zu jedem Flag gehört noch ein Freigabe-Bit (Interrupt Enable).

3.8 Serielle Schnittstelle (RS-232)

Die serielle Schnittstelle dient dem zu entwickelnden Applikationskontroller zur Kommunikation mit dem RFID-Lesegerät und dem RDT400. Sie ist in den oben beschriebenen Mikrocontroller als Peripherie-Modul integriert.

RS-232 (1969 von der EIA als Recommended Standard 232 eingeführt) ist eine Spannungsschnittstelle, d.h. verschiedene Spannungspegel stellen die Information dar. Sie entspricht einer V.24 ISO-Schnittstelle hinsichtlich Signalsemantik, Elektrik und Steckerbelegung. Eine RS-232 Verbindung stellt eine serielle Datenübertragung dar. Die Bits werden hintereinander auf einer Leitung übertragen - im Gegensatz zur parallelen Datenübertragung, bei der mehrere Bits gleichzeitig auf mehreren verschiedenen Leitungen übermittelt werden. Der Spannungsbereich für die logische Eins reicht von -3 bis -15 Volt und die logische Null wird durch Spannungen zwischen +3 und +15 Volt abgebildet. Die Abbildung der logischen Eins als negative Spannung und der logischen Null als positive Spannung nennt man negative Logik. Da die Spannung mit der Länge einer Leitung (wegen des größer werdenden elektrischen Widerstandes) immer kleiner wird, ist die Leitungslänge begrenzt (auf ca. 25 bis 100 m, je nach Kabel- und Steckverbindungsqualität). Als Steckverbindung wurden ursprünglich 25-polige Stecker und Buchsen (Sub-D) benutzt. Da die Leitungen i.d.R. reine Drucker- bzw. Terminal-Steuerleitungen sind, die für die meisten Verbindungen nicht benötigt werden, haben sich 9-polige Sub-D-Stecker und Buchsen etabliert.

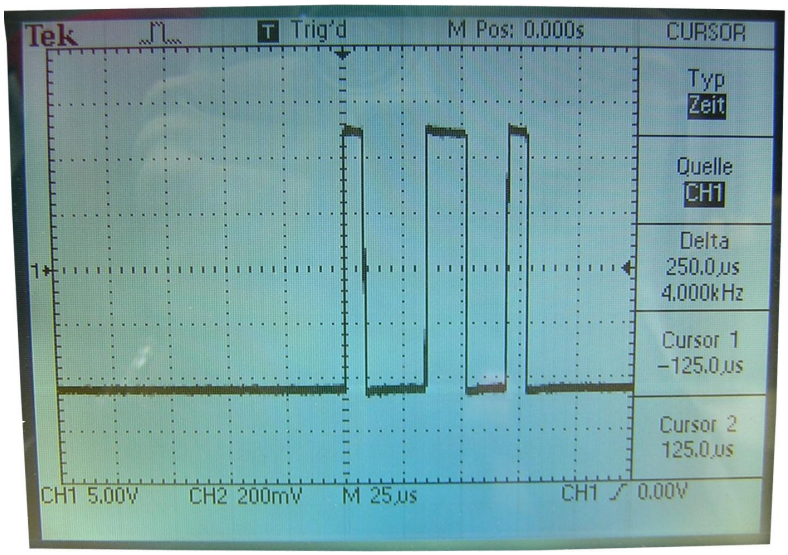

Bild 23 **Schnappschuß des RS-232 Signals für das ASCII-Zeichens 'g'. Die Y-Achse zeigt die Spannung des Signals in Volt. Ein Kästchen entspricht 5 V. Die X-Achse zeigt den zeitlichen Verlauf des Signals. Ein Kästchen entspricht 25 µs. Die Baudrate beträgt 115.200 Bit/s.**

Der "American Standard Code for Information Interchange" (ASCII) definiert die ersten 128 Zeichen eines 8 Bit Zeichensatzes. [Bild 23] zeigt die Übertragung des Zeichens 'g' (kleines g) vom Applikationskontroller zum Lesegerät über die serielle Schnittstelle (RS-232). 'g' besitzt den ASCII Wert Hex 67, binär entspricht dies: 01100111.

Der Ruhezustand der Übertragungsleitung entspricht dem Pegel einer logischen Eins (-15 V). Die Übertragung eines Zeichens (ein Byte) beginnt mit einem vorangestellten Startbit, das als logische Null gesendet wird. Anschließend werden nacheinander 8 Datenbits, beginnend mit dem niederwertigsten Bit (LSB) ausgegeben. Bei den Datenleitungen (Sendeleitung, Empfangsleitung) ist die Spannungslage invertiert; sie liegt im Ruhezustand auf negativem Spannungspegel. Das Ende eines zu übertragenden, seriellen Zeichens wird durch ein Stopbit gebildet, das der vorangegangenen Information angehängt wird. Damit erkennt der Empfänger, daß die Übertragung des Zeichens beendet ist. Die Bitcodierung erfolgt bei RS-232 im NRZ-Code [siehe Kapitel 3.6.5].

4 Systemimplementierung

Wie Kapitel [2.3, Lösungsansatz] zeigt, soll die Zentrale zur Steuerung und Lenkung des zu entwickelnden Applikationskontrollers durch einen Mikrocontroller realisiert werden, wie er in [Kapitel 3.7] beschrieben ist. Schon in der frühen Phase des Projektes muß ein geeigneter Mikrocontroller ausgewählt werden. Das Projekt zerfällt nach dieser Entscheidung in zwei Teilprojekte, wobei das eine die Hardware, das andere die Software entwickelt. Eine nachträgliche Korrektur der Auswahl ist in der Regel nicht möglich, da die Software auf spezielle, herstellerabhängige Features des Mikrocontrollers zugeschnitten wird. Für diese Entscheidung gilt es deshalb sorgfältig zu überprüfen, ob die sehr verschiedenartigen Anforderungen einer Applikation erfüllt werden.

4.1 Selektion der Hardwarebauteile

Für die Selektion der Bauteile wird die Hardware in vier Teile zerlegt:

- CPU. Die zentrale Verarbeitungseinheit, kurz als CPU (Central Processing Unit) oder Prozessor bezeichnet, übernimmt sowohl die eigentliche Datenverarbeitung als auch die Koordination aller recheninternen Aktivitäten [Beier02]. Der Kandidat MSP430 wird bei einer Betriebsfrequenz von 8 MHz mit einer Spannung von V_{CC} = 3,3 V betrieben und verbraucht maximal 560 µA. Nach außen führen 64 Pins, die mit Pinsockelleisten auf die Platine kontaktiert werden.

- Schnittstelle. Über sie kommuniziert die CPU mit der Außenwelt. Der gewählte RS-232 Standard schreibt Signalpegel von ± 15 V vor. Der MSP430 treibt eine logische Eins auf jedem Pin mit maximal V_{CC} – 0,3 V = 3,0 V. Das Sendesignal (TX) muß daher auf RS-232 Pegel abgebildet werden. Das Empfangssignal (RX) muß auf die maximal von der CPU erlaubte Eingangsspannung von GND bis 0,8 * V_{CC} = 0 bis 2,7 V transferiert werden. [Ti04]

- Sensoren. Sie nehmen Informationen über den Zustand eines technischen Prozesses durch Messung einer physikalischen Größe auf und leiten diese an die CPU. Die Zustandswechsel für den Objektein- bzw. -austritt werden durch zwei optische Sensoren

(Reflexions-Lichtschranken) herbeigeführt. Die Versorgungsspannung liegt zwischen 10 und 30 Volt.

- Spannungsversorgung. Sie stellt die Anforderungen an die benötigte elektrische Leistung obiger Bauteile zur Verfügung.

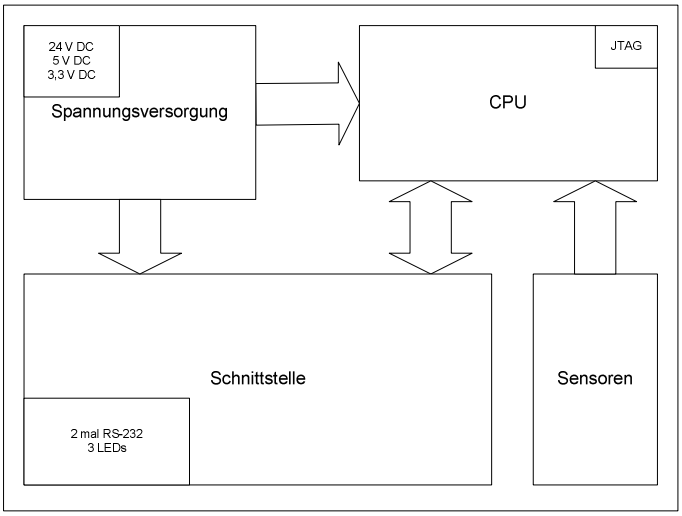

Bild 24 **Hardwarekomponenten. Die Applikation verlangt Strom und Kommunikation. Die Schnittstelle ist für Signalverstärkung, Pegelumsetzung und galvanische Trennung verantwortlich.**

Im nachfolgenden werden die vier Teile der Hardware genauer beschrieben, um die geeigneten Bauteile zur Erstellung des Applikationskontrollers zu selektieren.

4.1.1 CPU

Nun gilt es anhand verschiedener Kriterien die Applikation zu klassifizieren, um den geeigneten Mikrocontroller zu finden.

Echtzeit-Anforderung:

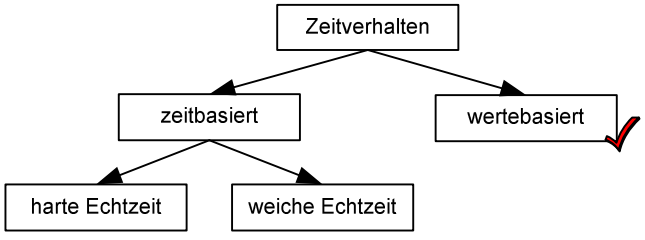

Bild 25 **Verschiedene zeitliche Anforderungen von Anwendungen.**

Bezüglich des Limits der Ausführungszeit ist die Applikation wertebasiert, da die Antwort auf Eingangsdaten nicht innerhalb einer eindeutig begrenzten Zeitspanne generiert werden muß. Die Richtigkeit des Wertes ist jedoch für das Funktionieren der Applikation von hoher Bedeutung. Benötigt werden gute mittlere Performance, tiefe Pipeline und Speicher zum Halten der Werte.

Ablaufsteuerung:

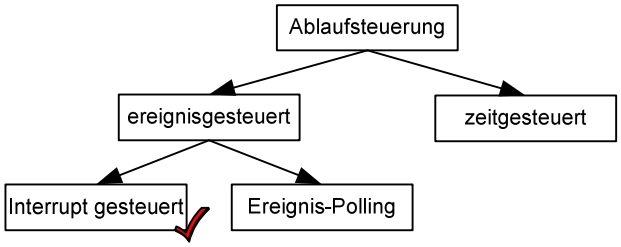

Bild 26 **Verschiedene Arten der Ablaufsteuerung von Anwendungen.**

Jede Aktivität wird zeitunabhängig (asynchron) durch äußere Ereignisse initiiert. Die Ereignisse können durch spezielle Hardware direkt den Programmfluß tangieren. Die Verarbeitung von Interrupts sollte keinen großen Overhead aufweisen und geringe IRQ-Antwortzeiten haben.

Partitionierung:

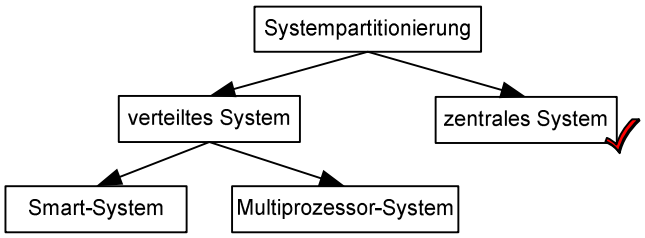

Bild 27 **Arten der Partitionierung von Systemen.**

Es bedarf nur einer Einheit, mit der alle Ein- und Ausgangssignale verarbeitet werden. Alle Sensoren und Aktoren sind mit diskreten Leitungen mit der zentralen Einheit verbunden. Es wird eine hohe Anzahl an Ports benötigt, die IO sollte gut adressierbar sein (Memory-Mapped-IO).

Fehlerverhalten:

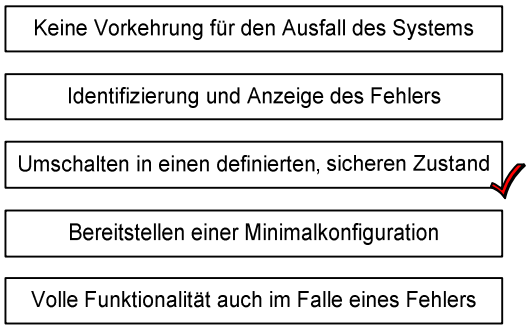

Bild 28 **Unterschiedliches Verhalten im Falle eines Fehlers.**

Falls ein Fehler im System auftritt, soll in einen sicheren Zustand umgeschaltet werden. Die Erkennung soll durch Hardware-Redundanz unterstützt werden. Eine einfache Form stellt der Watchdog-Timer dar. Er muß periodisch innerhalb einer bestimmten Zeitspanne ein Signal vom Mikroprozessor erhalten. Wenn dieses Signal durch einen Soft- oder Hardware-Fehler nicht zeitlich korrekt generiert wird, wird das System in einen sicheren Zustand [siehe Kapitel 6.2.6, Fehlerbehandlung] gebracht, eine Fehlermeldung ausgegeben und der Mikroprozessor neu gestartet.

Die Klassifizierung des geeigneten Mikrocontrollers wird im folgenden mit einer Übersicht der benötigten Fähigkeiten [Tabelle 2] sublimiert:

Peripherie	Gefordert	Bemerkung
ROM als Flash	Ja	Mehrfach beschreibbar ("Coding on Chip“)
ADC	Nein	Keine Verarbeitung analoger Signale
Low Power	Ja	Warum Energie verschwenden?
Display LCD	Nein	Interaktion mit Anwender nur selten nötig
CAN / IIC / Ethernet	Nein	Externe Bridge benutzen
UART	Ja	2 mal RS-232 mit Interrupt-Unterstützung
Watchdog	Ja	Für definiertes Fehlerverhalten
Extended RAM	Ja	Möglichkeit zur erweiterten Datenaufbereitung
PWM	Nein	Keine Verarbeitung von PWM-Signalen
Comparator	Nein	Keine Verarbeitung analoger Signale
Brown Out Detection	Nein	Spannungsversorgung ist stabillisiert
I/O Ports	Ja	Anschluß Lichtschranken, LEDs
Code Security Fuse	Nein	Anwendung nicht außer Haus
Elektromagnetische Veträglichkeit	Ja	Steuerung einer RFID-Applikation
Timer	Ja	Messung Objektlänge und -geschwindigkeit

Tabelle 2 **Anforderungen an den Mikrocontroller bezüglich seiner Peripherie.**

Aufgrund dieser genannten Anforderungen fällt die Wahl des richtigen Mikrocontrollers leicht. Es wird sich hierbei um einen der typischen 8 oder 16 Bit Mikrocontroller handeln, wie er von vielen Herstellern angeboten wird. Ein Merkmal sticht jedoch heraus: zwei UARTs.

Die “Universal Asynchronous Receiver and Transmitter"-Einheit (UART) stellt die physikalische Schnittstelle zur Kommunikation dar. Sie führt die Signalgenerierung aus, die die darüberliegende RS-232 Schnittstelle [siehe 3.8] braucht. Die erste UART-Einheit wird dem Reader, die zweite dem Host zugeordnet. Zum Senden eines ASCII-Zeichens über den UART wird das Zeichen in ein spezielles 8 Bit Schieberegister kopiert, das der Mikrocontroller zur Verfügung stellt. Die UART-Einheit schiebt dann Bit für Bit auf die Sendeleitung der RS-232 Schnittstelle in einer zuvor eingestellten Baudrate und fügt evtl. Start-/Stop/Parity-Bits dazu. Entsprechend wird ein empfangenes Zeichen behandelt, die UART-Einheit signalisiert das Eintreffen eines ASCII-Zeichens per Interrupt, so daß der Anwender in seiner selbst geschriebenen Interrupt-Service-Routine (ISR) darauf reagieren kann. Durch "Memory-Mapped IO“ resultiert die Aktion "ein ASCII-Zeichen senden" in genau einem Assembler-Befehl (move txByte, $0123). Da die Transponder schnellstmöglich ausgelesen werden sollen, um an den Host geschickt zu werden, wurde das Kriterium UART besonders beachtet.

In der Tat bieten derzeit nur wenige Hersteller Mikrocontroller mit 2 UARTs an, z.B.:

- Texas Instruments (Typ MSP430)
- Atmel (Typ Atmega64 und Atmega128)
- NEC (Typ K-line)

Dabei sollten auch Entwicklungstools und Support beachtet werden. Die Wahl fiel auf den MSP430, da die Entwicklungsumgebung (IDE) für diesen Mikrocontroller im Hause SICK präsent war.

MSP430 Architektur:
Der Mikrocontroller MSP430 von Texas Instruments hat eine 16 Bit RISC (Reduced Instruction Set Computer) CPU. Die CPU ist mit der Peripherie durch einen getrennten Adreß- und Daten-Bus verbunden ("von Neumann"-Architektur). Ein flexibles Clock-System generiert für die Peripherie unterschiedliche Takte. Alle Module der Peripherie sind Memory-Mapped, was den Zugriff erleichtert. Das ROM (Flash) hat 60 kByte, das RAM umfaßt 2 kByte. Von den 16 CPU Registern (jedes 16 Bit breit) sind die ersten vier reserviert für: Programm Counter (PC), Stack Pointer (SP), Status-Flags und Konstanten-Generator. Die Arithmetic Logic Unit (ALU) kann 27

verschiede Befehle mit sieben unterschiedlichen Adreßmodi verarbeiten. Jeder Befehl wird in einem Zyklus verarbeitet. [Ti04]

Clock-System des MSP430:
Dieser Abschnitt beschreibt, wie das Clock-System im Mikrocontroller MSP430 arbeitet (Taktgenerierung). Die Taktgenerierung ist im MSP430 so entworfen, daß sie flexibel und energiearm arbeitet. Die Implementierung dieser Ziele basiert zum Großteil auf der Fähigkeit, unterschiedliche Taktgeber für unterschiedliche Teile des Chips auszuwählen. Durch Wahl der minimalen Taktgeschwindigkeit, die für ein gegebenes Modul notwendig ist, wird die Leistungsaufnahme verringert, wobei die Bedürfnisse des Moduls bezüglich seiner Synchronisation noch erfüllt werden. Der MSP430 besitzt drei Taktquellen für das Clock-System und drei Taktschienen, aus denen die Module wählen können. Die Taktquellen werden als Basis für die Taktschienen verwendet. Dies erlaubt eine Mischung aus nieder- und hochfrequenten Takten, die im ganzen System genutzt werden können. Die drei Taktquellen sind:

Low Frequency Crystal Clock (LFXTCLK)
Dieses Signal ist gedacht für einen extern angeschlossenen Uhrenquarz. Der Quarz wird an die Pins XIN und XOUT kontaktiert und sollte mit 32 kHz oszillieren. Diese Quelle kann über das OSCOFF Flag im Statusregister (SR) an- und abgeschaltet werden.

Crystal 2 Clock (XT2CLK)
Dieses Signal ist die zweite externe Taktquelle und wird über Pin XT2IN, XT2OUT angeschlossen. Üblicherweise wird das Signal als hochfrequente Quelle genutzt. In dieser Arbeit ist ein 8 MHz Quarz angeschlossen, was die maximal nutzbare Frequenz darstellt.

Digitally Controlled Oscillator Clock (DCOCLK)
Dies ist die einzige intern generierte Taktquelle und stellt die Werkseinstellung nach einem Reset dar. Üblicherweise läuft dieses Signal mit ungefähr 900 kHz, kann aber durch die Flags RSELx, MODx und DCOx nach unten manipuliert werden. Genaue Angaben macht Texas Instruments bei der Auslieferung des Mikrocontrollers über die Frequenz dieser Quelle nicht. Präzise Messungen (Zeit) sind mit dieser Quelle nicht möglich.

Die drei Taktschienen sind:

Master Clock (MCLK)
Diese Taktschiene ist die Quelle für den MSP430 CPU-Kern; er muß anliegen, damit der Prozessor Instruktionen abarbeiten kann. Die MCLK verfügt über die meisten Optionen, eine Quelle

zu selektieren. Die Quelle wird über die SELMx Flags des Basic Clock System Control Register 2 (BCSCTL2) eingestellt. Der Taktteiler wird über DIVMx im BCSCTL2 selektiert. Die CPU kann auch ganz abgeschaltet werden, indem das CPUOFF Flag im SR gesetzt wird. Nur ein Interrupt kann aus diesem Zustand wieder herausführen.

Submaster Clock (SMCLK)
Diese Taktschiene wird für den Großteil der Peripherie verwendet und als Quelle dient entweder DCOCLK oder XT2CLK. Einstellungen für die Quelle werden über SELS und SCG Flags im BCSCTL2 und SR vorgenommen. Der Teiler wird über DIVSx Bits im BCSCTL2 bestimmt.

Auxiliary Clock (ACLK)
Die Quelle dieser Taktschiene ist immer LFXTCLK. Sie wird als Option angeboten, um langsame Subsysteme mit noch weniger Energie zu versorgen. Dieser Takt kann über DIVAx im Basic Clock System Control Register 1 (BCSCTL1) geteilt werden.

Das Taktsystem im MSP430 hat einen Vorteiler am Eingang jeder Taktschiene und an den meisten Peripherie-Eingängen. Dies erlaubt für jedes Peripherie-Modul ein individuelles Timing. Für Entwicklungszwecke sind die schnellsten Taktgeber normalerweise die nützlichsten, ihr Nachteil ist jedoch eine erhöhte Leistungsaufnahme.

Taktteiler:
Während der gesamten Takteinrichtung am MSP430 werden Taktteiler benutzt. Ein Taktteiler verringert die Frequenz eines eingehenden Taktes und gibt die geteilte Frequenz an seinem Ausgang aus. Die einfachsten Teiler arbeiten auf der Basis des Vielfachen von zwei, so daß das Signal am Ausgang eines Taktteilers die Hälfte, ein Viertel oder ein Achtel der Eingangsfrequenz entspricht.

4.1.2 Sensorik

Lichttaster und Lichtschranken erfassen Objekte berührungslos über größere Entfernungen. Sie sind unverzichtbare Bestandteile in vielen automatisierten Prozessen. Die Erfassung von Objekten soll in dieser Applikation schnell und zuverlässig erfolgen. Die robuste und wartungsarme Technik der Reflexions-Lichtschranken kommt daher hier zum Einsatz.

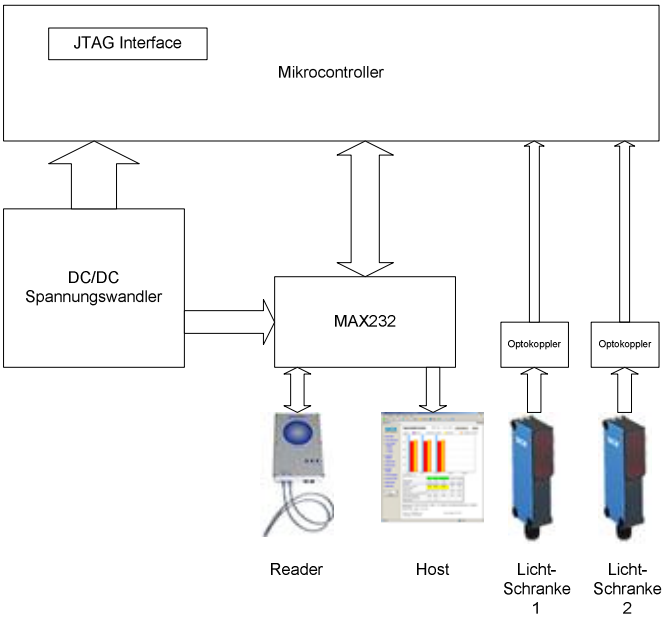

Bild 29 **Blockschaltbild der Systemimplementierung. Zwei Lichtschranken dienen dem Mikrocontroller als Sensoren. Durch die Lichtschranken wird der Ein- und Austritt von Objekten erkannt.**

Die Reflexions-Lichtschranke WL18 von der Firma SICK AG ist ein opto-elektronischer Sensor und wird zum optischen, berührungslosen Erfassen von Objekten eingesetzt. Zum Betrieb ist ein Reflektor erforderlich. Die Lichtschranke verfügt über antivalente Schaltausgänge Q und nicht-Q. Damit kann der Pegel des Ausgangssignals (High oder Low) für Lichtunterbrechung und -empfang gewählt werden. In dieser Arbeit wurde der Lichtunterbrechung ein High-Signal zugeordnet, da dieser Zustand nur für kurze Zeit eintritt (Objektein/-austritt) und dadurch die meiste Zeit der energiearme Low-Pegel (0 V) anliegt. Ein geeigneter Reflektor muß gegenüber der Lichtschranke montiert und ausgerichtet werden. Über einen Drehknopf wird die Lichtschranke justiert, bei optimalem Lichtempfang leuchtet eine Empfangsanzeige (LED) permanent. Die Reichweite beträgt 0 bis 7 m bei einem Lichtdurchmesser von 0,04 bis 2 m. Die Versorgungsspannung kann von 10 bis 30 V DC gewählt werden, wobei der Pegel eines High-Ausgangssignals der Versorgungsspannung gleich ist.

Funktionsweise:
In der Lichtschranke dient eine lichtemittierende Infrarot-Leuchtdiode als Fotoaktor. Sie ist zusammen mit einer lichtdetektierenden Fotodiode in einem Gehäuse verbaut. Die Leuchtdiode wird mit 1 kHz gepulst und sendet Licht durch eine Linse auf einen entfernt montierten Reflektor. Dieser reflektiert das Licht auf die Fotodiode. Dort wird der *innere lichtelektrische Effekt* ausgenutzt. Der gepulste Betrieb erlaubt es, unerwünschtes Fremdlicht zu filtern.

Der innere lichtelektrische Effekt:
Durch die Bestrahlung von halbleitendem Material mit elektromagnetischen Wellen (Licht) erhöht sich dessen Leitfähigkeit. Die Lichtquanten bewirken, daß einzelne oder mehrere Elektronen in das Leitungsband gehoben werden, wodurch sich Ladungsträgerpaare (Elektronen- und Defektelektronen) bilden [Bild 30]. Den Strom, der durch diese Bestrahlung im Material hervorgerufen wird, bezeichnet man als Fotostrom. [Zenk05]

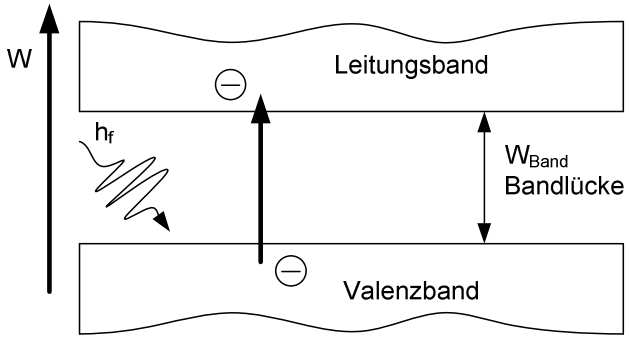

Bild 30 **Der innere lichtelektrische Effekt: Photonen brechen Kristallverbindungen auf und heben Elektronen in das Leitungsband. Wenn die Energie der absorbierten Photonen $W = h_f$ größer als die der Bandlücke W_{Band} ist, werden Elektronen vom Valenzband in das Leitungsband gehoben. Der Halbleiter leitet dann Strom.**

Tritt ein Objekt zwischen Lichtschranke und Reflektor, wird die Reflexion unterbrochen und der Fotostrom bricht ein. Dies wird schaltungstechnisch erkannt und von der Lichtschranke als Ausgangssignal aufbereitet.

4.1.3 Schnittstelle

Die in diesem Kapitel als Schnittstelle bezeichnete Einheit übernimmt die Aufgabe der Pegelwandlung von Lichtschranken- und RS-232 -Signalen. Die Lösung der Aufgabe wird mit Hilfe von Optokopplern (ein Koppler pro Lichtschranke) und dem Baustein MAX232 erreicht.

Optokoppler CNY17-2:
Beide Lichtschranken werden mit einer Spannung der Stärke V_{LS} = 24 V DC versorgt und liefern als Signal (während der Unterbrechung durch ein Objekt) ebenso 24 V zurück. Zur Weiterverarbeitung im Mikrocontroller muß das Signal auf TTL-Pegel (Transistor-Transistor-Logik) gebracht werden. Dafür wird ein elektronisches Koppelelement benutzt (Optokoppler).
Optokoppler sind Signalübertragungsglieder, die aus einem Fotoaktor (LED) und einem Fotosensor (Fotodiode, Fototransinstor oder Fotohyristor) bestehen. Lichtsender und Lichtempfän-

ger sind optisch gekoppelt, aber galvanisch getrennt in einem Gehäuse untergebracht. Das Verhältnis des Ausgangsstroms zum Eingangsstrom heißt Koppelfaktor. Er beträgt beim CNY17-2 63 bis 125%. [Fair04]

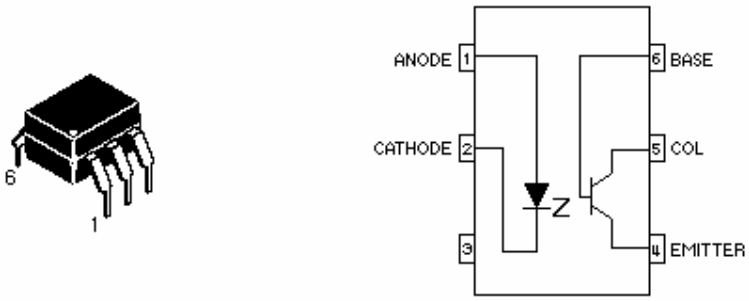

Bild 31 **Optokoppler CNY17-2 der Firma Fairchild Semiconductor. Er dient zur galvanischen Trennung des Sensorsignals einer Lichtschranke zum Applikationskontroller. (Bildquelle: Fair04)**

Galvanische Trennung:

Zwei Stromkreise sind galvanisch voneinander getrennt, wenn es keinen Weg gibt, über den Strom aus einem Stromkreis in den anderen fließen kann, also keinerlei direkte Verbindung über ein leitendes Material wie Eisen oder Kupfer besteht. Problematisch ist bei galvanisch getrennten Stromkreisen, wenn trotzdem ein Signal von einem Stromkreis zum anderen übertragen werden soll. Es wird dann zwischen den beiden Stromkreisen eine Art von Kopplung hergestellt, die Strom blockiert, ein Nutzsignal jedoch durchläßt.

MAX232:

Der Baustein MAX232 der Firma Maxim ist ein weitverbreiteter Pegelkonverter für die serielle RS-232 Schnittstelle. Vom Mikrocontroller aus stehen nur TTL Pegel (0 V für Low, 2,7 V für High) zur Verfügung, trotzdem soll mit dem Reader und dem Host nach dem RS-232 Standard kommuniziert werden. Es stehen also nicht die geforderten Spannungen von +15 V und -15 V zur Verfügung. Eine eigene Spannungsversorgung nur für die RS-232 Schnittstelle wäre ein zu hoher Aufwand. Abhilfe schafft hier der MAX232. Diesem genügt eine 5 V Versorgungsspannung, um eine Wandlung von TTL auf RS-232 Pegel zu realisieren. Die benötigten +15 V und -15 V werden von dem Baustein selbst erzeugt. Als externe Beschaltung genügen 5 Kondensatoren mit 100 nF Kapazität, sie dienen als Ladungspumpen. Daneben wird der Applikationskontroller durch das Vorschalten des MAX232 gegen elektrostatische Beschädigung über die serielle Schnittstelle geschützt. [Max04]

Bild 32 **Baustein MAX232 mit 5 Kondensatoren, die ihm als Ladungspumpen dienen.**

Zur Funktion:
Der MAX232 verfügt über 2 integrierte DC/DC-Wandler: einen zur Spannungserhöhung (+15 V) und einen als Spannungsinverter (-15 V). Für diese Wandler werden 5 Kondensatoren benötigt [Bild 32]. Um eine korrekte Wandlung eines TTL Signals auf ein RS-232 Signal zu ermöglichen muß eine Pegelumsetzung auf die höheren Pegel erfolgen, und zusätzlich muß das Signal invertiert werden, da eine logische Eins bei TTL +2,7 V und bei RS-232 -15 V entspricht. Die Wandlung von RS-232 auf TTL erfolgt analog zu oben, ein Pegel von -15 V wird auf +2,7 V umgesetzt. Der MAX232 verfügt über zwei TTL zu RS-232 und RS-232 zu TTL Stufen. Eine Stufe dient dem Senden/Empfangen zum bzw. vom Reader. Über die zweite Stufe wird mit dem Host kommuniziert.

4.1.4 Spannungsversorgung

Die Betriebsspannungen für die Bauteile "on Board" weichen voneinander ab. Der Mikrocontroller läuft mit 3,3 V. Der Baustein MAX232 benötigt eine Eingangsspannung von 5 V um die RS-232 Signalpegel von ±15 V zu treiben. Ein DC/DC-Wandler transformiert die Eingangsspannung des Applikationskontrollers von V_{in} = 24 V DC auf 5 V DC. Damit wird der MAX232 versorgt. Über einen Festspannungsregler erhält der Mikrocontroller weiter transformierte 3,3 V.

Auswahl des DC/DC Wandlers:
Die extern zugeführte Spannungsversorgung von 24 V DC entspricht dem Industriestandard. Der Applikationskontroller läßt sich dadurch an den gängigen Einsatzorten betreiben. Die Spannung muß zum Betrieb der TTL-Bausteine im Applikationskontroller auf 5 V DC runtertransformiert werden. Vorteil dabei ist, daß extreme Spannungsschwankungen unter schwierigen Netzbedingungen kompensiert und die eng toleranten Versorgungsspannungen der TTL-Bauteile dadurch gewährleistet werden. [Tabelle 3] zeigt die Dimensionierung des DC/DC Wandlers nach angeschlossenen Verbrauchern.

Verbraucher	Wert	Einheit	Anmerkung
MSP430	4,5	mA	8 MHz Betriebstakt bei V_{cc} = 3,3 V
3 LED's	60	mA	3 mal 20 mA
UART	24	mA	4 mal 6 mA (TX0/RX0, TX1/RX1)
MAX232	190	mA	Datenblatt [Max04]
Spannungsregler LM1086	120	mA	Baureihe TI3.3 [Nat01]
Gesamt	398,5	mA	

Tabelle 3 **Berechnung der Stromaufnahme On Board.**

Die Stromaufnahme erhöht sich weiter durch diverse Klein-Bauteile, die in der Stückliste [siehe Anhang] aufgeführt sind. Um für eventuelle, zukünftige Erweiterungen des Applikationskontrollers gewappnet zu sein wurde ein DC/DC Wandler gewählt, der ausreichend Leistungsreserven bietet. Der "TEN10-2411" der Firma Traco Electronic AG, Schweiz liefert bei einer Eingangsspannung von 18 bis 36 V DC eine Ausgangspannung von 5 V DC mit maximal 2000 mA Ausgangsstrom [Trac04]. Dieses Bauelement wurde als geeigneter DC/DC Wandler für den Applikationskontroller ausgewählt.

4.1.5 Externe Komponenten

Bridge Lantronix UDS-10 (RS-232 → TCP/IP):
Die Bridge bietet einen schnellen, einfachen und kostengünstigen Weg, den Datenzugriff und die Konfiguration des Applikationskontrollers in ein bestehendes Netz einzubinden. Sie erlaubt Geräten mit serieller Schnittstelle (RS-232) die Verbindung und Kommunikation mit einem Ethernet Netzwerk (TCP/IP). Daneben werden auch RS-422, RS-485 und UDP Datagramme unterstützt. [Lan04]

Dazu muß die Bridge für das Netwerk konfiguriert werden (IP Adresse, Subnetzmaske, Gateway). Ein eingebauter Webserver erleichtert diesen Vorgang (auch über Telnet-Console oder Setup-Assistent möglich).

Bild 33 **Die Bridge "Lantronix UDS-10" konvertiert RS-232 auf Ethernet und umgekehrt.**

Funktionsweise:
Genutzt wird die Methode, der "Seriellen Tunnelung". Serielle Daten, die der Applikationskontroller über die RS-232 Schnittstelle sendet und empfängt, werden von der Bridge in TCP/IP Pakete eingekapselt und über das Ethernet transportiert. Mit zwei Bridges, verbunden durch ein Netzwerk, können virtuelle serielle Verbindungen über das Firmengelände oder um die ganze Welt hergestellt werden. Dabei nutzt die Bridge das Internet Protokoll (IP) für die Netzwerk-Kommunikation und das Transmission Control Protokoll (TCP) um sicherzustellen, daß Daten weder verlorengehen noch dupliziert werden (gesicherte Punkt zu Punkt Verbindung).

4.2 Schaltungsentwurf und Layout

[Bild 34] zeigt das (selbsterstellte) Layout aller zuvor beschriebenen Bauteile. Auf Grundlage dieser wurde der Applikationskontroller gebaut. Eine Eurorasterplatine mit einem Lochrastermaß von 2,54 mm nimmt die Bauteile auf. Die Bestückungsliste kann dem Anhang dieser Arbeit entnommen werden.

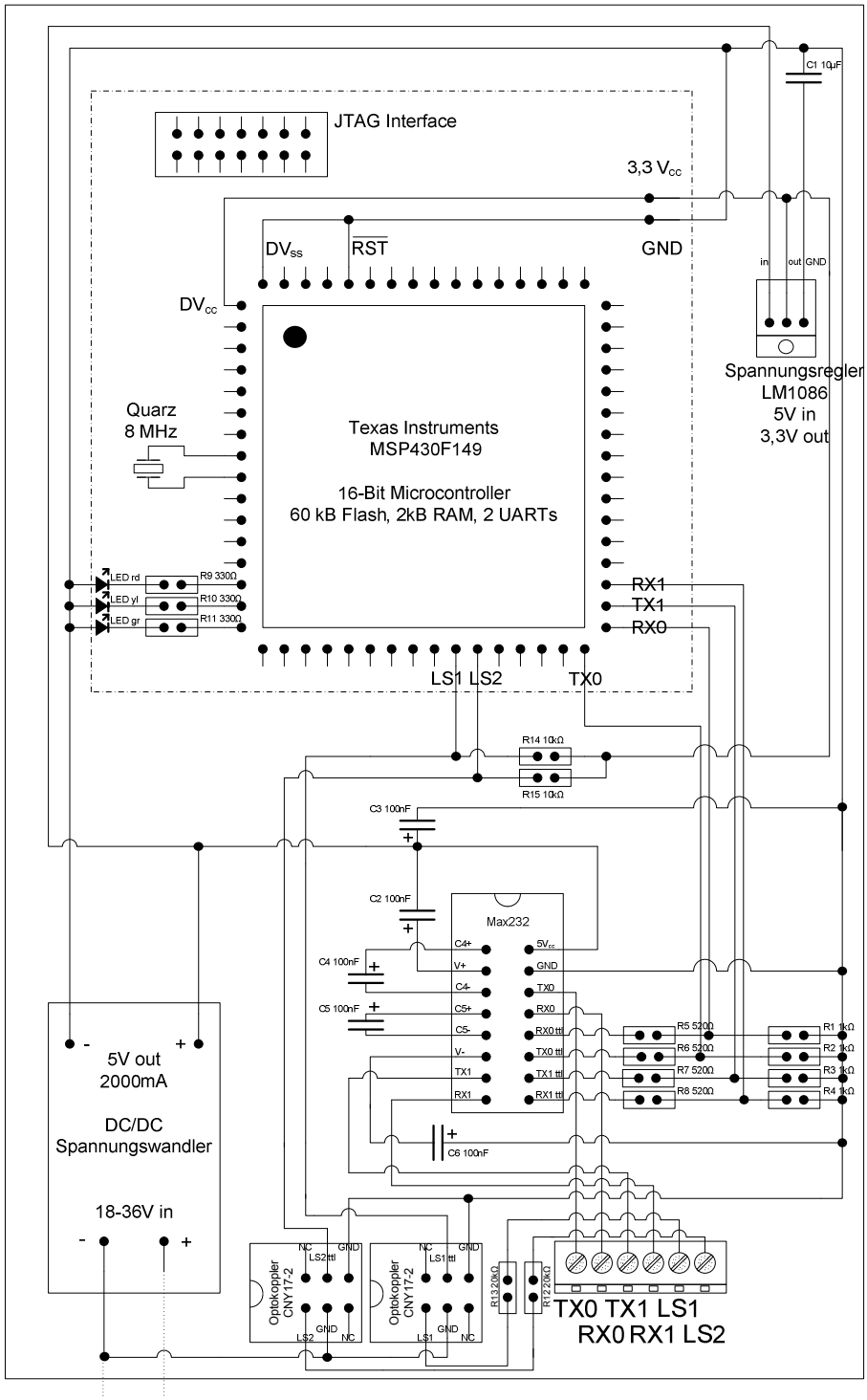

Bild 34 **Platinen-Layout des Applikationskontrollers. An die sechs Schraubklemmen (unten rechts im Bild) werden Sende - und Empfangsleitungen der seriellen Kommunikation zum Reader (TX0/RX0) und zum Host (TX1/RX1) sowie Signalleitungen beider Lichtschranken (LS1/LS2) angeschlossen.**

Der Applikationskontroller ist in einem Gehäuse der Firma Bopla GmbH, Bünde verbaut. In den Gehäuseschalen befinden sich Nocken für selbstformende Schrauben. Die Nockenmaße sind speziell auf Europakartenformat abgestimmt und nehmen damit die elektronischen Bauteile sicher auf. Die Verschraubung am Einsatzort kann am Ober- oder Unterteil des Gehäuses erfolgen.

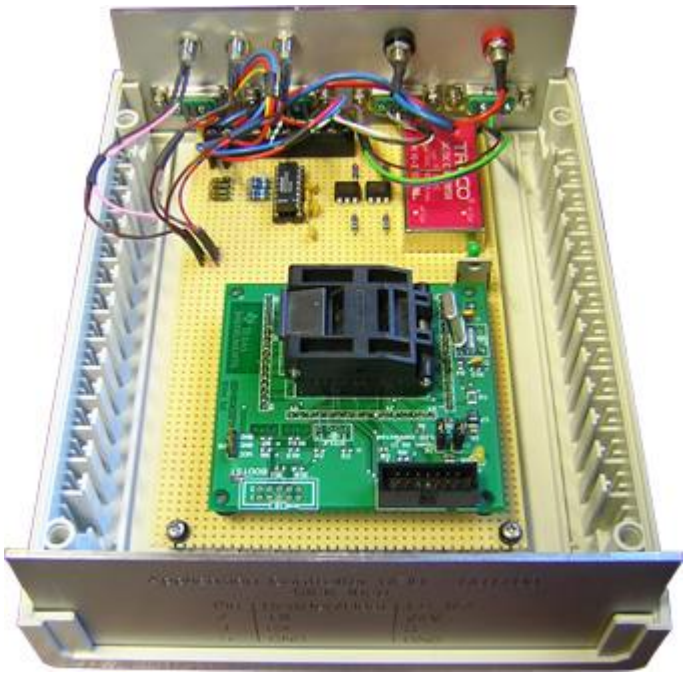

Bild 35 **Der Applikationskontroller mit geöffnetem Gehäuse.**

Die beiden Frontplatten für Vorder- und Rückwand - Maßanfertigungen der Firma Schaeffer Apparatebau KG, Berlin - sind in das Gehäuse eingeschoben [Bild 35]. Ihre Fräsung, Bohrung und Gravierung wurden mit dem Tool "Frontplatten Designer V3.2" erstellt und bei der Firma Schaeffer in Auftrag gegeben. Lüftungsschlitze an Ober- und Unterteil des Gehäuses gewährleisten den Wärmeabtransport durch Luftströmung. Die Strömung entsteht dabei schon durch den Dichteunterschied zwischen der im Gerät erwärmten Luft und der kühleren Umgebung. [Elek05]

5 Entwicklungssystem

Ein Entwicklungssystem dient der Entwicklung von Programmen, z.B. zur Automatisierung eines Gerätes. Diese Programme werden nach der Funktionsprüfung in den ROM-Speicher (Flash) des Mikrocontrollers gebrannt. Die Programme sind in Assembler oder einer Hochsprache (hier: C) geschrieben und werden mit Hilfe des Entwicklungssystems in Maschinensprache übersetzt und in den Mikrocontroller gespeichert. Bevor die Programme in das Flash gespeichert werden, sind sie auf richtige Funktion zu prüfen, um in der Serienherstellung des Applikationskontrollers keinen Ausschuß zu produzieren. Diese Funktionsprüfung findet im Entwicklungssystem statt, das in seinem Aufbau dem späteren Zielsystem entsprechen muß.

5.1 Entwicklungsumgebung (IDE)

Eine integrierte Entwicklungsumgebung stellt ein Anwendungsprogramm zur Entwicklung von Software dar. Meistens wird synonym der englische Begriff Integrated Development Environment (IDE) verwendet. Das Programm, das den Applikationskontroller und damit auch den Reader, die Lichtschranken und die Datenaufbereitung für den Host steuert wurde mit Hilfe einer IDE erstellt. IDEs verfügen in der Regel über folgende Komponenten:

- Texteditor — Für die Eingabe des Sourcecodes, evtl. mit Syntax-Highlighting, Formatierung und Programmierhilfen.
- Compiler — Zur Übersetzung des Sourcecodes in Maschinensprache (Programmmodul).
- Linker — Stellt einzelne Programmodule zu einem ausführbaren Programm zusammen.
- Debugger — Für das Auffinden und Diagnostizieren von Fehlern im Programm.

Umfangreichere IDEs beinhalten oft weitere hilfreiche Komponenten wie Versionsverwaltung, Projektmanagement, UML-Modellierung oder die Möglichkeit der einfachen Erstellung von GUI's (Graphical User Interfaces). IDEs sind hilfreiche Werkzeuge, die dem Software-Entwickler häufig wiederkehrende Aufgaben abnehmen und einen schnellen Zugriff auf wichtige Funktionen bieten. Der Entwickler kann sich dadurch ganz auf seine eigentliche Aufgabe, die Programmierung, konzentrieren.

Die Software für den RFID-Applikationskontroller wurde mit der "Embedded Workbench" der Firma IAR GmbH, Parsdorf programmiert. Hierbei handelt es sich um eine IDE, die speziell auf die Bedürfnisse der Mikrocontrollerprogrammierung zugeschnitten ist. Die Entwicklungsumgebung unterstützt zur Zeit 38 verschiedene Typen von Mikroprozessoren, auch den für dieses Projekt gewählte MSP430 von Texas Instruments. Bei der gewünschten Programmiersprache kann zwischen C, C++ und EC++ gewählt werden. Übersichtlich gestaltete Header-Files werden von IAR für jeden Prozessortyp mitgeliefert. Sie erleichtern den Zugriff auf prozessorspezifische Besonderheiten, z.B. den Adressen der Memory-Mapped-IO oder Interrupt-Vektoren. Alle Einstellungen für Compiler, Linker und Debugger werden über komfortable Fensterdialoge vorgenommen. Der Texteditor bleibt beim Debuggen erhalten und dient als Monitor des Singlesteppings. Während des Debuggings können zusätzliche Informationen angezeigt werden, die bei der Fehlersuche helfen: Sourcecode, Disassembly, Memory, Register, Watch und Locals, Call Stack und Terminal I/O. Über das Terminal I/O Fenster können während der Programmausführung Ausgaben mit "prinft()" getätigt werden. Dies erleichtert die Programmierung von Mikrocontrollern, die in der Regel über keinen angeschlossenen Bildschirm verfügen. Die "Embedded Workbench"[2] konnte mit dem in [Kapitel 5.2] angesprochenen FET über die JTAG-Schnittstelle kommunizieren. Ein integrierter "Loader" brennt vor jeder Debug-Sizung das Programm in das Flash-ROM des MSP430.

Als Alternativen für die Entwicklung von Software auf dem MSP430 gibt es u.a.:

CrossWorks for MSP430:
Kmpl. IDE von Rowley Associates Ltd, England. 30 Tage Eval-Version für Win32 und Linux unter http://www.crossstudio.co.uk/crossworks/Evaluating.htm. Die Vollversion kostet ca. 700 Euro.

ICCIDE:
C Compiler, Loader und Texteditor von ImageCraft Creations Inc., USA. 30 Tage Demo-Version für Win32 unter http://www.imagecraft.com/software/demos.html. Die Vollversion kostet ca. 350 Euro. Dazu wird der NoICE430 Debugger für ca. 100 Euro empfohlen.

Code Composer Essentials:

[2] Vier Handbücher mit insgesamt über 600 Seiten Inhalt beschreiben detailliert den Umgang mit der IDE, Debugger, Compiler und Linker. Auch die Funktionen der mitgelieferten Standard C Bibliotheken werden darin gut erklärt. Die angesprochen Features haben allerdings auch ihren Preis, IAR verlangt ca. 2.800,- Euro für die "Embedded Workbench".

Kmpl. Opensource IDE basierend auf der Eclipse-Plattform von Texas Instruments, USA. Eine auf 8 kB Codegröße begrenzte Beta-Version für Win32 gibt es unter http://www.go-dsp.com/beta/code_composer_essentials/index.htm. Die Vollversion wird kostenlos sein.

MSPGCC:
Toolchain der erfolgreichen UNIX Opensource GNU Compiler Collection. Unter http://mspgcc.sourceforge.net kann der für den MSP430 portierte C Compiler und dazugehörige Binutils für Win32, BSD und Linux kostenlos bezogen werden. GCC ist ein Kommandozeilen-Programm, es gibt also kein graphisches Benutzerinterface, keinen Texteditor mit Syntax-Highlighting oder Wizards.

AQ430:
Kpml. IDE von Quadravox Inc., USA. 30 Tage Demo Version für Win32 unter http://www.quadravox.com/AQ430.htm. Die Vollversion kostet ca. 400 Euro.

5.2 Evaluation Board

Für den MSP430 bietet die Firma Texas Instruments das Flash Emulation Tool (FET) an [Bild 36]. Es unterstützt die komplette Entwicklung "In-System" durch Programmierung (ISP), Debugging auf Assembler- oder C-Quellcode-Ebene, Single-Stepping und Möglichkeit zum Setzen mehrerer Hardware-Breakpoints. Die Performance des MSP430 bleibt beim Debuggen weitgehend erhalten, da der Kern des Mikroprozessors den Industriestandard JTAG unterstützt. Anders als bei klassischen Hintergrund-Debuggern ist kein Time-Sharing der seriellen Schnittstelle nötig - durch die eingebettete Debug-Unterstützung im MSP430 selbst. Texas Instruments liefert das FET mit einer Kickstart-Version der IAR Entwicklungsumgebung aus, die auf eine ladbare Programmgröße von maximal 2 kByte begrenzt ist. Ähnliche Versionen des FET werden auch von Drittanbietern vertrieben.

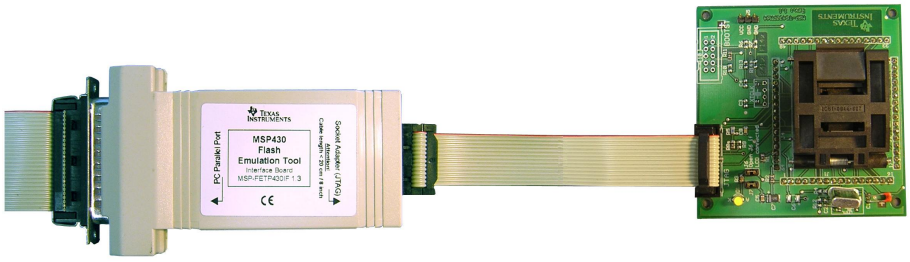

Bild 36 **Das Flasch Emulation Tool (FET, links) wird zwischen Paralleler- (PC) und JTAG- (Mikrocontroller) Schnittstelle gehängt. Es erlaubt das "In system programming" (ISP). Der Mikrocontroller sitzt im Sockel (geschlossen, rechts).**

Das JTAG Interface dient als Programmierschnittstelle und versorgt während der Implementierungs- und Debugphase den MSP430 mit 3,3 V. Der Programmspeicher (Flash-ROM) wird beim Beschreiben und Löschen mit erhöhten 3,6 V versorgt.

5.3 Softwaretools

Die Software des Applikationskontrollers verwendet für die Kommunikation zum Reader und zum Host die RS-232 Schnittstelle. Über diese werden Befehle und Daten als eine Folge einzelner ASCII-Zeichen dargestellt und in reader- bzw. hostspezifische Protokolle verpackt (Pakete). Das Einpacken sollte nach Möglichkeit automatisch, also generisch erfolgen. Die eigens geschriebenen Software-Module "reader.c" und "host.c" übernehmen u.a. diese Aufgabe [siehe 6.2.4, Strukturierung der Software]. Für den Test der Kommunikationsfähigkeit des Applikationskontrollers wurde im Rahmen dieses Projektes ein spezielles Programm mit dem Namen "ListenPort" geschrieben.

Hintergrund: Das RDT400 Serverprogramm, das auf dem Host läuft, kann bis zu 64 getrennte Verbindungen zu Barcode- und/oder RFID-Applikationskontrollern halten. Dazu ist die RS-232 Schnittstelle ungeeignet, da sie nicht für 1-zu-n Verbindungen ausgelegt ist. Besser geeignet und als Standard bereits in vielen Unternehmen eingeführt, ist hierfür das Ethernet. Die Barcode-/RFID-Applikationskontroller sollten also für die Kommunikation zum Host das Ethernet unterstützen. Der in dieser Arbeit entwickelte Applikationskontroller bedient sich zum Einbinden

dieser Funktionalität einer Bridge [siehe 4.1.5, Externe Komponenten] als zusätzlicher Hardwareeinheit.

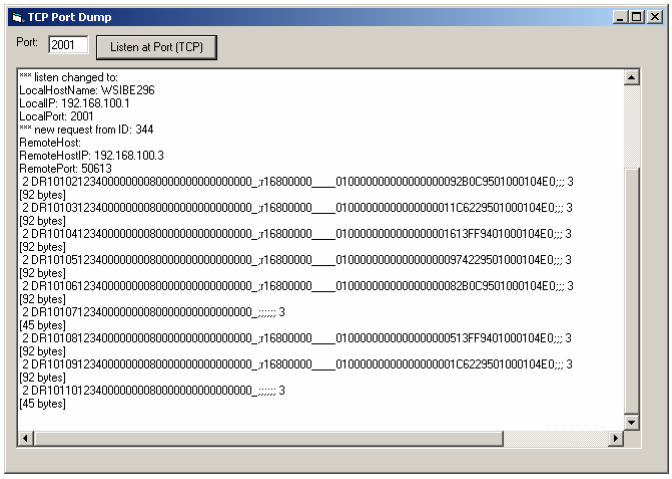

Bild 37 **"TCP Port Dump" ist ein eigens geschriebenes Programm zur Kontrolle der vom Applikationskontroller gesendeten "RDT400 Data Strings".**

Zum Test dieser Konfiguration - also Erstellen der Host-Pakete, Versand über RS-232, nochmaliges Verpacken in TCP/IP (Ethernet) durch ein externes Gerät - dient das oben erwähnte Programm. Es wurde mit Visual Basic 6.0 geschrieben und läuft an Stelle des RDT400-Servers auf dem Host. Seine Aufgabe ist das Horchen auf eingehende Verbindungsanfragen vom Applikationskontroller und die Ausgabe der Rohdaten (dump) auf dem Bildschirm. Die Rohdaten enthalten ASCII-Steuerzeichen, die von einem gewöhnlichen Tool ignoriert werden würden. Diese Steuerzeichen bestehen aus den ersten 32 Zeichen des ASCII-Zeichensatzes [siehe 3.8], sie werden von "ListenPort" in Hexadezimalwerte umgerechnet und zusammen mit echten ASCII-Zeichen auf einem Textfeld übersichtlich ausgegeben [siehe Bild 37]. Für den Betrieb des Softwaretools ist die Installation der "Visual Basic Runtime Environment" auf dem Host-Rechner nötig ("msvbvm60.dll" von Microsoft). Die Kommunikation konnte mit Hilfe des Softwaretools erfolgreich getestet werden.

6 Softwareentwicklung

In der Projektspezifikation [Kapitel 2] wurde durch Analyse der Anforderungen festgelegt, was der Applikationskontroller leisten soll, jedoch noch nicht wie diese Leistungsmerkmale erreicht werden. Diese Lücke soll im vorliegenden Kapitel geschlossen werden, d.h. es wird der Bauplan der Software, die Software-Architektur, entwickelt. Dazu wird das System in Teilsysteme/Module zerlegt sowie deren Schnittstellen und Algorithmen ausgearbeitet. Die Software soll mit der Bereitstellung und systematischen Verwendung von Methoden, Techniken und Werkzeugen der Softwaretechnik hergestellt werden.

6.1 Allgemeine Konventionen

In der Softwaretechnik sind Prinzipien (Abstraktionsprinzip, Prinzip der Strukturierung) durch Erfahrung hergeleitet und bestätigt worden, die bei der Problemlösung zugrunde gelegt werden sollten. Das *Abstraktionsprinzip* beinhaltet die Verringerung der Komplexität durch Vernachlässigung von Nebenaspekten und Details. Mit dem Blick auf die wesentlichen Aspekte ist es möglich, Modelle zu entwickeln. Ein Modell ist in diesem Zusammenhang eine Abstraktion der Realität aus einem spezifischen Blickwinkel [Schn04]. Das *Prinzip der Strukturierung* greift auf der Ebene von Programmkomponenten durch die ausschließliche Verwendung von Grundstrukturen (z.B. Folge, Auswahl, Wiederholung). Auf der Ebene von Softwaresystemen erfolgt die Strukturierung durch Hierarchisierung und Modularisierung. In diesem Kapitel soll gezeigt werden, wie mit Hilfe der Softwaretechnik die genannten Prinzipien eingehalten werden können. Dafür werden Anwendungsfälle und deren Ablauf ausgearbeitet. Ziel ist die Implementierung eines Automatenmodells, das den Applikationskontroller steuert.

6.2 Spezifikation der Software

Die Frage nach der gewünschten Leistung des geplanten Systems steht am Beginn jeder Systementwicklung. Eine fundierte Beantwortung bewahrt davor, im Detail zu versinken, bevor man weiß, was vom System überhaupt erwartet wird. Ein Use-Case Diagramm (Anwendungsfall-Diagramm) zeigt das externe Verhalten eines Systems aus der Sicht der Nutzer, indem es

die Nutzer (in UML "Akteure" genannt), die Use-Cases und deren Beziehungen zueinander darstellt. Ein Nutzer kann eine Person, aber auch ein Nachbarsystem sein. Use-Cases bilden dabei die Reaktion des Systems auf Ereignisse seiner Umwelt ab und fassen dabei Teile der Systemdienstleistung zusammen [Doug98].

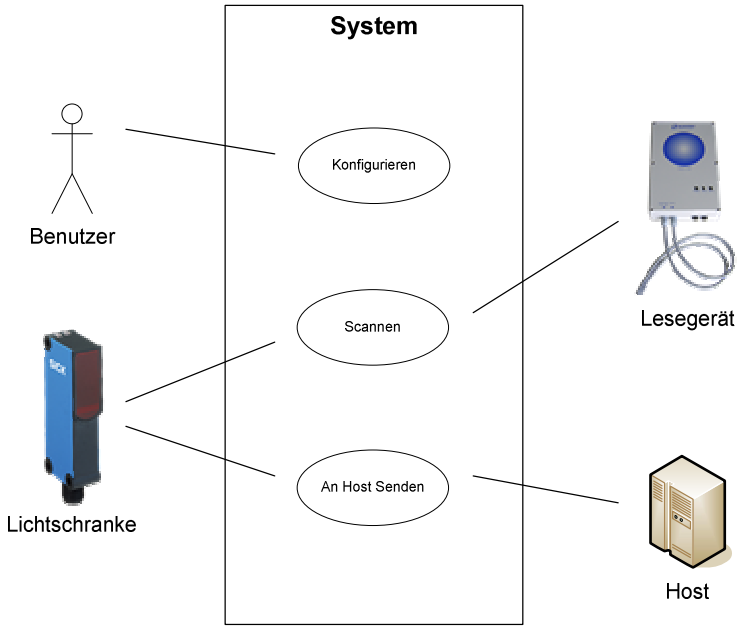

Bild 38 **Detailierung der Applikation durch ein Use-Case Diagramm.**

Das System ist diejenige Einheit, die das Verhalten, welches durch die Use-Cases beschrieben wird, realisiert und anbietet, wobei die Kanten des Systems die Systemgrenzen markieren. Der Aktor "Benutzer" interagiert mit dem Use-Case "Konfigurieren". Er soll dabei vom System geführt werden. Das Konfigurieren wird durch ein Login in das System vom Benutzer eingeleitet. Parameter, die dem System bekannt sein müssen, sind:

- Abstand von Lichtschranke 1 zu Lichtschranke 2
- Controller-Kennung

Mit Hilfe eines vom System selbst generierten Datums (Zeit) kann die Objektlänge und -geschwindigkeit ermittelt werden. Anhand der Controller-Kennung vermag der Host (RDT400) mehrere angeschlossene Applikationskontroller zu unterscheiden und zu verwalten.

Die Akteure "Lichtschranke" stoßen die Anwendungsfälle "Scannen" und "An Host Senden" an. Dabei sind sie hart an das System gekoppelt und liefern das Signal zum Anstoß per Interrupt. Das "Lesegerät" übernimmt übernimmt als Aktor die Rolle des ID-Lieferanten. Der Datenaus-

tausch zwischen System und Lesegerät erfolgt bidirektional und in bezug auf die Ermittlung der Leserate zeitkritisch, da er nicht verzögernd wirken soll. Dem "Host" kommt die Rolle des ID-Emfängers zu.

6.2.1 Programmablauf

Die Assoziation zwischen "Lichtschranke", "Scannen" und "An Host Senden" soll durch geeignete Technik weiter verfeinert werden. Der zeitliche Ablauf einer Objekterfassung läßt sich gut durch ein Sequenzdiagramm [Bild 39] darstellen. Da der Mikrocontroller nicht in einer objektorientierten Sprache programmiert ist, wurden für die im Sequenzdiagramm üblicherweise verwendeten Klassennamen die Modulnamen verwendet.

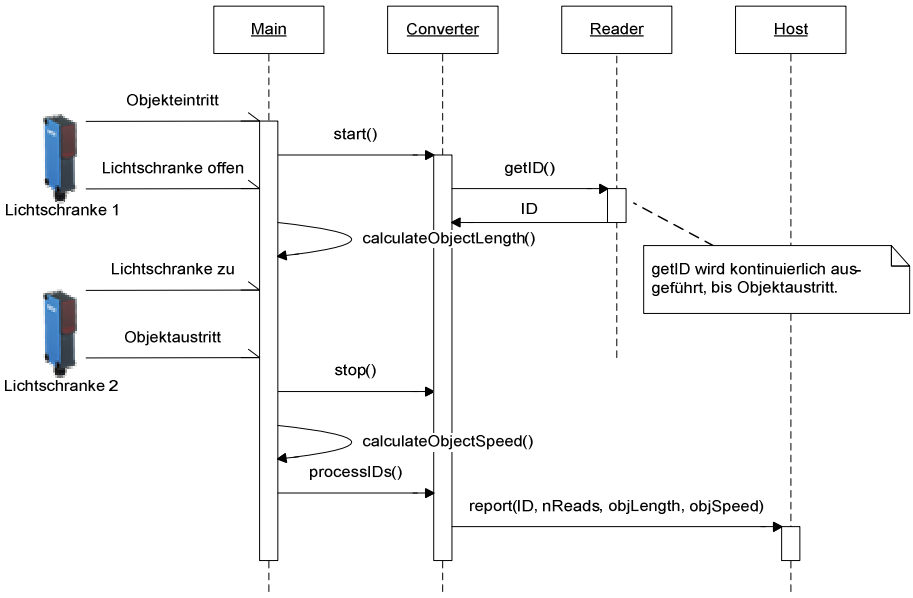

Bild 39 **Programmablauf bei Eintritt eines Objektes als Sequenzdiagramm. Horizontal erkennt man die beteiligten Module. Vertikal läuft die Zeit.**

Im Sequenzdiagramm werden die Botschaften, die zwischen den Modulen ausgetauscht werden, durch Pfeile dargestellt. Eine Botschaft ist in diesem Sinne ein Aufruf einer Operation eines (Empfangs-) Moduls durch ein (Sende-) Modul. Bei diesen Botschaftsaufrufen können auch Parameter übergeben werden. Asynchrone Botschaften werden mit einer halben Pfeilspitze dargestellt. In [Bild 39] triggert Lichtschranke 1 bei Objekteintritt das Modul "Main" mittels Interrupt an. "Main" startet den Erfassungsvorgang, indem es die Funktion "start" im Modul "Converter" aufruft. Danach ruft "Converter" die Funktion "getID" des Moduls "Reader" auf und erhält

daraufhin die IDs aller im Lesefeld der Antenne befindlichen Transponder. Dies geschieht so lange, bis Lichtschranke 2 den Objektaustritt an Modul "Main" meldet und dieses den "Converter" stoppt. Das Modul "Main" ruft die moduleigenen Funktionen zur Bestimmung der Objektlänge ("calculateObjectLength") und Objektgeschwindigkeit ("calculateObjectSpeed") mit den fallenden Flanken von Lichtschranke 1 und 2 auf. Im Modul "Host" wird daraufhin die Funktion "report" aufgerufen, die für jede ermittelte ID einen RDT400 Data String an den Host sendet. Damit ist die Erfassung eines Objektes abgeschlossen. Der Applikationskontroller wartet nun auf die erneute Triggerung durch Lichtschranke 1, um die beschriebene Erfassungssequenz erneut zu starten.

6.2.2 Automatenmodell

Das Automatenmodell dient dazu, die Ablaufsteuerung des Applikationskontrollers möglichst fehlerfrei zu programmieren. Dafür wird der zuvor beschriebene Programmablauf als Automat modelliert. Die Reaktion auf ein Ereignis dokumentiert sich in Ausgaben des Systems und/oder in dessen internen Veränderungen. Die Ereignisse, die der Applikationskontroller empfangen kann, sind:

Ereignis	Bemerkung
Lichtschranke 1 ↑	Lichtunterbrechung
Lichtschranke 1 ↓	Lichtempfang
Lichtschranke 2 ↑	Lichtunterbrechung
Lichtschranke 2 ↓	Lichtempfang
Login	Konfiguration durch Benutzer

Tabelle 4 **Externe Eingangsereignisse. Auf die Ereignisse reagiert der Automat mit Aktionen/Aktivitäten und/oder Zustandswechsel.**

Zur Beschreibung des Verhaltens solcher Systeme sind Automatenmodelle das adäquate Stilmittel. Ein Automatenmodell definiert die Reaktion eines Systems auf ein Ereignis unter Berücksichtigung des aktuellen Systemzustandes. Es beinhaltet eine Menge von Zuständen und definierten Übergängen zwischen ihnen. Für jeden Übergang in einen anderen Zustand ist eine Eingabe angegeben, deren Eintreffen diesen Übergang veranlaßt. Die Ausgaben des Automaten sind entweder an die Übergänge (Mealy-Automaten) oder an die Zustände (Moore-Automaten) gebunden. [Fohl04]

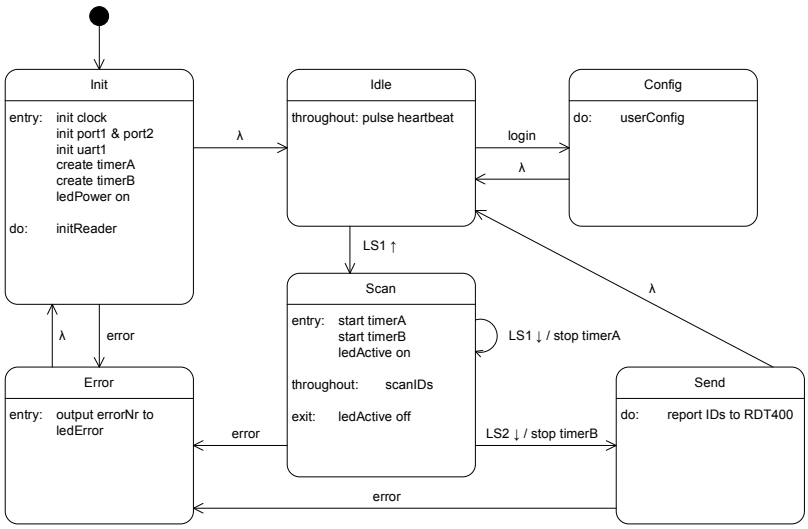

Bild 40 **Für das Design der Software wurde die robuste Technik des Harel-Automaten-Modells genutzt.**

Noch flexibler ist der *Harel-Automat*, der Mealy- und Moore-Automaten hybrid verbindet und zudem bedingte Zustandsübergänge, hierarchische Zustandsautomaten, Zustände mit Gedächtnis und nebenläufige Zustände einführt. Diese Erweiterungen dienen im wesentlichen der besseren Lesbarkeit und höheren Abstraktion. Die Eingabe in einen endlichen Automaten entspricht den Ereignissen in einem Harel-Automaten. Die Ausgabe in einem endlichen Automaten verhält sich äquivalent zu einer Aktion oder einer Aktivität im entsprechenden Harel-Automaten. [Sam02]. Das für den Applikationskontroller erstellte Automatenmodell nach Harel ist in [Bild 40] dargestellt.

6.2.3 Verwendete Protokolle

Art und Umfang der Protokolle, welche Reader und Host verwenden, sind verschieden. Das RDT400 Data String Protokoll, welches vom Host erwartet wird, besitzt eine größere Anzahl an Feldern, als durch die Daten des Readers gefüllt werden können. Daher sollten diese Felder durch sinnvolle Default-Werte belegt werden.

STX/ETX-Protokoll der Reader SHL-2001 und SIR-2600:
Das Protokoll der hier verwendeten RFID-Reader beinhaltet Funktionen zur deren Konfiguration sowie zum Lesen und Schreiben von verschiedenen Transpondern. Hierbei handelt es sich um ein Protokoll für die RS232-Schnittstelle. Die Datentelegramme setzen sich aus Start- und Stopzeichen, Prüfsumme, einer Funktionsnummer und gelesenen bzw. zu schreibenden Daten

zusammen. Die Daten (IDs) mehrerer bzw. aller im Feld der Antenne vorhandenen Transponder können in einem Telegramm übermittelt werden.

RDT400 Data String Protokoll:
Der Host erwartet ein Datentelegramm, in welchem neben den eigentlichen Daten (IDs) Zusatzinformationen enthalten sind. Diese Informationen sind vom RFID-System selbst nicht lieferbar. Im Protokoll werden sie gebraucht und genutzt, um Daten über Qualität und Fehlersicherheit der Barcode-Scannerergebnisse zu liefern (der RDT400 wurde ursprünglich als Host für Barcode-Scanner entwickelt). Zusätzlich werden Informationen über die Lage des gelesenen Barcodes, Objektgröße und Objektabstand zum Scanner übermittelt. Für den Betrieb eines RFID-Systems mit dem RDT400 können diese Felder mit sinnvollen Werten gefüllt werden. Um die Auswertung möglich zu machen, sind Form und Struktur des "emulierten" Protokolls genau einzuhalten. Im Anhang dieser Arbeit befindet sich eine Tabelle mit allen Feldern des Protokolltelegramms samt Füllwerten.

6.2.4 Strukturierung der Software

Die Software ist nach Aufgaben modularisiert. Physikalisch vorhandene Geräte und Einheiten sind in der Codestruktur wiederzuerkennen. Die in einem Modul realisierten Aufgaben/Abläufe besitzen einen engen inhaltlichen Zusammenhang (Kopplung). Zudem sollten die Module untereinander eine möglichst geringe Kopplung aufweisen, was die Schnittstellen zwischen den einzelnen Modulen minimiert. So bleibt der Rest des Softwaresystems vor Änderungen unberührt, falls ein Modul ausgetauscht wird. Durch Nachbildung der realen Welt in Software erhält man wiederverwendbare, für den Benutzer logisch gegliederte Software [Doug98]. Die Software umfaßt die nachstehenden sieben Module (Reader, Host, Converter, UART, Main, Timer und Tools), die im folgenden beschrieben werden.

Modul Reader:
Abstrahiert das RFID-Lesegerät vom Anwender. Es muß an verschiedene Lesegeräte angepaßt werden, falls sie nicht das Scemtec STX/ETX Protokoll unterstützen. Es stellt Funktionen zum Initialisieren des Readers, Testen der Verbindung zum Reader, Finden von IDs und Optimierung der Leserate bereit. Dabei werden dem Lesegerät Befehle im STX/ETX Protokoll verpackt und im ASCII-Format über RS-232 gesendet.

02	6C21	03	xor
STX	Command	ETX	<Checksum>

(Hex)

Bild 41 **Befehl (Request) zum Erfassen aller IDs im Lesefeld.**

Das Lesegerät führt den Befehl aus und sendet an das Modul Reader alle IDs, die sich im Lesefeld befinden.

06	02	6C21	02	D829950100E0…	03	xor
ACK	STX	Command	ID-Count	IDs	ETX	<CS>

(Hex)

Bild 42 **Antwort (Response) vom Lesegerät, es wurden zwei Transponder gefunden.**

Das Lesegerät sendet nur auf Befehl eine Antwort, so daß das Modul bis zu einem definiertem Timeout solange auf Antworten wartet, bis es einen Kommunikationsfehler erkennt. Es handelt sich also um synchrone Kommunikation, was das Zusammenspiel erleichtert und im embedded Bereich üblich ist. Dem Nutzer des Moduls (Aufrufer) werden im Fehlerfall aussagekräftige Fehlercodes zurückgegeben. Jede Antwort vom Lesegerät wird von dem Modul auf Formatierung, Zusammengehörigkeit, Framing und Checksumme geprüft.

Modul Host:
Es stellt das Interface zum Versenden von Daten an den Host zur Verfügung. Die Verbindung zum Host ist unidirektional und läßt auch einen Heartbeat zu, mit dem man signalisieren kann, das die Verbindung noch aktiv ist. Daten werden hier, wie vom/zum Reader, in das STX/ETX Protokoll verpackt. Der RDT400 verzichtet jedoch auf eine Prüfsumme, was ein Sicherheitsrisiko darstellt. Bei hoher Baudrate, langem Kabel oder elektromagnetischen Störquellen kann ein Bit auf der seriellen Verbindung schnell kippen (Transition 0→1 oder 1→0).

Modul Converter:
Es enthält den zentralen Algorithmus [Bild 43] zur Ermittlung der Erfassungssicherheit. Daten (IDs) vom Reader werden hier gesammelt und, nachdem das/die Objekte das Lesefeld verlassen haben, an den Host gesendet. Das Modul hält die Datenstrukturen "Array von ISO 15693-Transpondern" und "RDT400 Data String". Das Modul übernimmt die Steuerung von Begin des Objekteintrittes bis zum Objektaustritt und bedient sich dabei an den Funktionen aller anderen Module. Der Converter kann nicht erkennen, welcher Typ von Reader ihm die IDs zuspielt. Er kann also mit jedem Typ zusammenarbeiten, solange eine Funktion getIDs() bereitsteht. Hier tritt der Vorteil von Modularisierung besonders deutlich zu Tage.

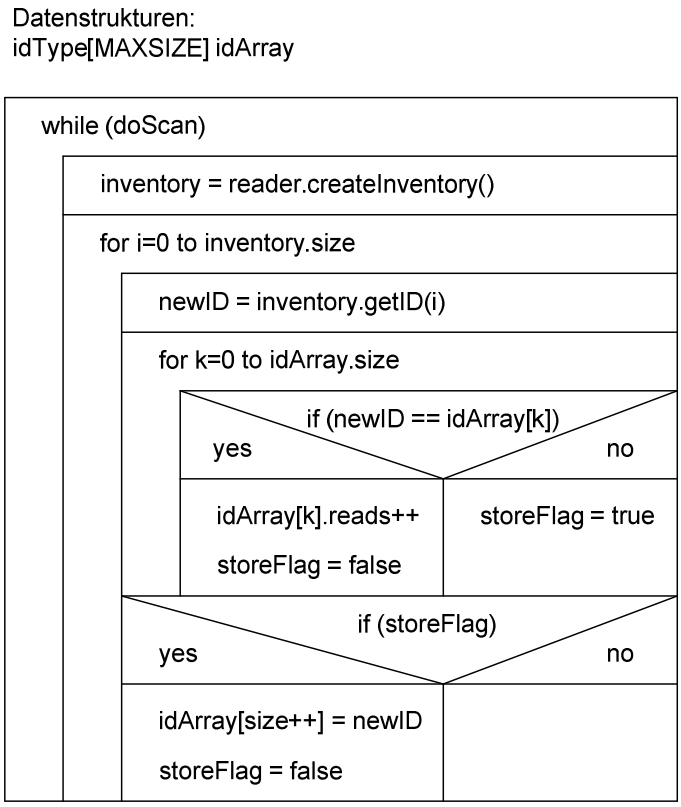

Bild 43 **Algorithmus zur Bestimmung der Leserate. Der Zähler einer bereits erkannten ID wird inkrementiert, neue IDs werden gespeichert.**

Mit fallender Flanke von Lichtschranke 2 (Interrupt) bricht der Algorithmus ab. Für jede erkannte ID wird ein RDT400 Data String erzeugt; die ID, Anzahl der Lesungen, Objektlänge/geschwindigkeit eingetragen und an den Host per STX/ETX Protokoll gesandt.

Modul UART:
Die Module UART0 und UART1 bieten Zugang zu beiden seriellen Schnittstellen. Sie müssen bei der Initialisierung konfiguriert werden, da es beim RS-232 Protokoll verschiedene Varianten gibt [Camp88]. Hier die gängigsten Parameter, die das Modul UART anbietet:

- Datenbits (7 oder 8)
- Parität (keine, gerade, ungerade)
- Stopbits (1 oder 2)
- Baud (9600 bis 115200 Bit/s)

Die Parameter zur Kommunikation mit Reader/Host sind in dieser Applikation wie folgt eingestellt: 8 Datenbits, keine Parität, 1 Stopbit ("8N1") bei 115200 Bit/s. Danach stehen zur Kommunikation mit den Geräten benutzerfreundliche Funktionen zur Verfügung:

```
sendByte(unsigned char b)
unsigned char b = receiveByte()
unsigned char buf[] = getBuffer(unsigned short* BufLen)
```

Modul Main:

Im Mikrocontroller wird nach Abarbeitung des Startup-Codes automatisch die Hauptroutine main() aufgerufen. In main() werden zunächst folgende Peripherien initialisiert: Watchdogtimer, Ports (LEDs, Interrupts für Lichtschranken), Clocksystem und Timer. Diese einleitenden Maßnahmen werden nach jedem Einschalt- oder Resetvorgang ausgeführt. Der Applikationskontroller arbeitet als Zustandsautomat [siehe 6.2.2], die beschriebene Initialisierung ist Teil des Zustands "Init". Das Modul Main enthält auch Routinen für die Konfiguration des Systems (Abstand zwischen Lichtschranke 1 und 2 in cm, Baudrate, DeviceID) von außen. Hierfür kann das windowseigene Hyperterminal genutzt werden. Empfängt der Controller drei 'x' während er im Zustand Idle ist, werden Parameter über die Konsole abgefragt und gespeichert.

Modul Timer:

Zeitgeber gehören zu den essentiellen Bestandteilen eines jedes Betriebssystems. Da es hier kein Betriebssystem unterhalb der Anwendung gibt, wurde die Zeiterfassung (wie alle anderen Module) selbstentwickelt. Normalerweise wird hierfür ein spezieller Uhrenquarz mit einer Frequenz von 32768 Hz eingesetzt, was 2 hoch 15 Hz entspricht. Durch 15-faches Teilen durch 2 erhält man somit einen Sekundentakt. Die Zeitmessung soll jedoch auf mindestens 1 ms genau erfolgen, da vorbeifahrende Objekte per Spezifikation mit bis zu 5 m/s schnell sein können. Bei einer Objektgröße von z.B. 0,2 m wird eine Lichtschranke für t_O = 0,2 m / 5 m/s = 40 ms durchbrochen.

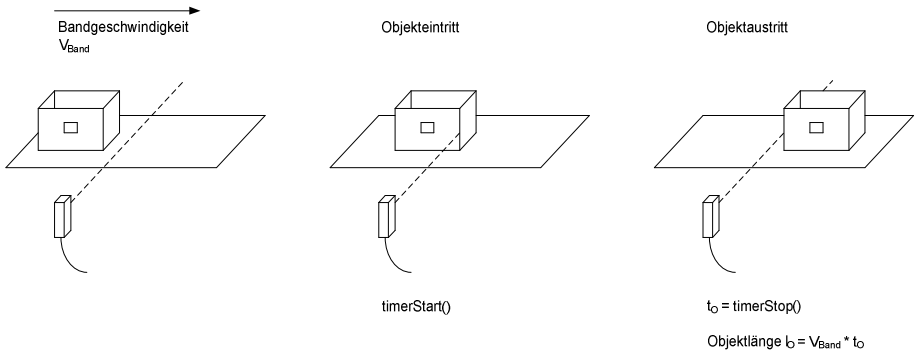

Bild 44 **Die Objektlänge wird ermittelt und an den Host gesendet.**

Hierfür wird der Systemtakt von 8 MHz zur Zeitmessung gebraucht. Der Mikrocontroller MSP430 hat zwei 16 Bit Timer mit insgesamt zehn Compare-Registern. Ein Timer wird mit jedem Takt (8 Millionen mal pro Sekunde) um eins inkrementiert und mit einem statisch geladenen Wert W = 8000 verglichen. Stimmt der Vergleich überein, dann wird vom Timer ein Interrupt ausgelöst, und für den Zeitmesser ist eine Millisekunde vergangen. Der Wert von 8000 ergibt sich aus W = 8 MHz / 1000 Hz und wurde mit einem Oszilloskop unter Betriebsbedingungen verifiziert. Die Objektgröße kann mit einer Granularität von G = 1 ms * 5 m/s = 5 mm bestimmt werden. Für den Anwender arbeitet das Modul wie eine Stopuhr, wobei mehrere Stopuhren parallel betrieben werden können. Mit timerInit() erhält man eine Referenz auf eine Uhr und kann beliebig oft start/stop aufrufen. Die Funktion stop() gibt die Zeit in ms seit dem Aufruf von start() zurück. Ein Überlauf einer Stopuhr tritt nach 49 Tagen (Hex FFFFFFFF * 1 ms = 1.193 Stunden) auf und wird mit einem Fehlercode quittiert.

Modul Tools:
Es enthält nützliche Routinen für den Umgang mit dem MSP430 (clock), der Konsole (printf) und sonstige Funktionen (Konvertierung ASCII zu long, xor Checksummen, Speicherblöcke kopieren u.a.). Die Routinen sind der Std-C Lib nachempfunden, ohne jedoch den Overhead dieser Bibliotheken nach sich zu ziehen.

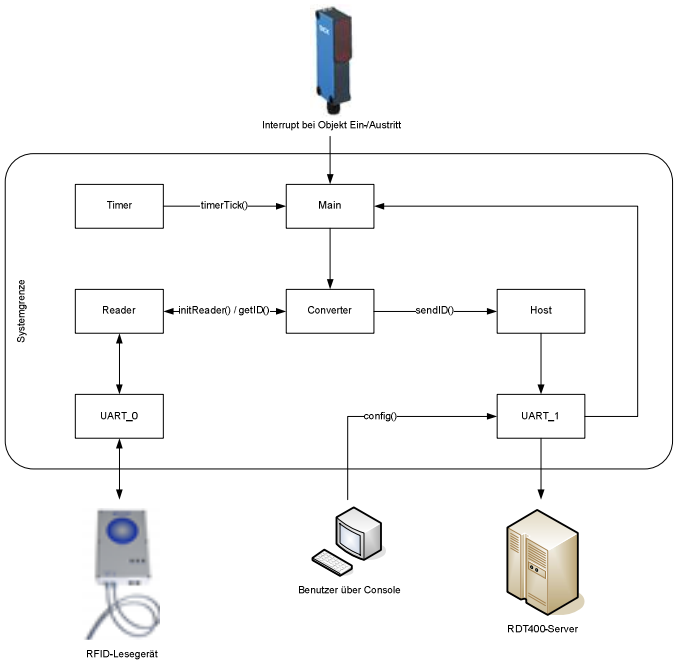

Bild 45 **Zusammenspiel der Software mit der Außenwelt.**

[Bild 45] zeigt die oben genannten sieben Module und die Richtung ihrer gesendeten und empfangenen Nachrichten zusammen mit externen Geräten.

6.2.5 Spezielle Methoden

Baudratengenerierung:
Durch den "Universal Synchronous/Asynchronous Receive/Transmit Communication Interface" (USART) erhält der Mikrocontroller MSP430 die Möglichkeit der seriellen Kommunikation über RS-232. Der USART kann im synchronen oder asynchronen Modus betrieben werden. Der asynchrone Modus (UART) ist der am weitesten verbreitete Modus, die Synchronisation von Sender und Empfänger erfolgt hier über Start- und Stopbits vor und nach einem Zeichenrahmen, was einen gewissen Overhead produziert.

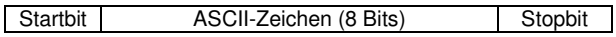

Bild 46 **Bei der asynchronen Kommunikation (UART) wird ein Zeichenrahmen durch ein Start- und ein Stopbit begrenzt.**

Der Vorteil asynchroner Kommunikation besteht im geringen gerätetechnischen Aufwand. Des weiteren kann der USART im MSP430 mit oder ohne Interrupt-Unterstützung betrieben werden. Für diese Arbeit wurde der asynchrone Modus mit Interrupts gewählt, da die Kommunikationspartner (Reader, Host, Console) den gleichen Modus benutzen und der Overhead vertretbar ist. Besonderes Augenmerk verlangt der USART bei der Baudratengenerierung und der Fehlererkennung, hervorgerufen durch das hochflexible Clock-System des MSP430 (vgl. 4.1.1).

Der USART wird durch sieben Kontroll-Register und einem read-only-Register gesteuert (jedes 8 Bit breit) [Ti99]. Die gewünschte Baudrate wird aus der in diesem Projekt relativ hohen Systemfrequenz von 8 MHz erzeugt. Im USART wird dafür ein Taktteiler benutzt (vgl. 4.1.1), durch den jedoch zwangsläufig ein Baudratenfehler entsteht. Der durch die Division resultierende Baudratenfehler ist relativ klein, wegen des großen ganzzahligen Anteils des Quotienten im Vergleich zu seinem Nachkomma-Anteil. Bsp.: 8.000.000 Hz / 9.600 Hz = 833,34 (833 ist 2.450-fach größer als 0,34). Der Baudratenfehler sollte dennoch weiter vermindert werden, um die Synchronisation der letzten Bits eines Nachrichtenrahmens zu gewährleisten. Aus diesem Grund bietet der MSP430 ein Modulationsregister (UMCTL). Es enthält acht Bits als Information, um die Baudrate des Sende- und Empfangssignals zu korrigieren. Die acht Bits beschreiben für den USART, wie der Taktteiler (Register UBR0 und UBR1) für die einzelnen Bits eines Zeichenrahmens benutzt werden soll.

Einstellung der Register UBR0, UBR1 und UMCTL für die gewünschte Baudrate:
Die Systemfrequenz wird durch die gewünschte Baudrate geteilt. Der ganzzahlige Anteil des Quotienten wird in die Hexadezimaldarstellung umgerechnet und auf die beiden Register UBR0 und UBR1 verteilt. Somit können 16 Bit breite Werte auf zwei 8 Bit breite Register abgebildet werden. Im Register UMCTL wird der verbleibende Nachkomma-Anteil moduliert. Der Wert für UMCTL ergibt sich aus der Anzahl der Bits, die geteilt durch die Breite von Register UMCTL dem Nachkomma-Anteil am nächsten kommen. Dabei sollte die Position der Bits im Register UMCTL gleichmäßig verteilt sein. Ein Implementierungsbeispiel verdeutlicht die Vorgehensweise:

```
Function UART_Set_Baud_Rate(rate)
   switch (rate)
      case 9600:  UBR0  = 0x41   // 8 MHz / 9600 = 0x0341 (Der ganzzahlige Teil)
                  UBR1  = 0x03   //
                  UMCTL = 0x89   // Modulation: 0.334 (Nachkomma-Anteil von 8MHz/9600)
                                 // 3/8 = 0.375 (das kommt 0.34 sehr nahe, also müssen 3
                                 // Bits in UMCTL gesetzt werden, z.B. 0x89 = %10001001)
      case 19200: UBR0  = 0xA0   // 8 MHz / 19200 = 0x010A - (Der ganzzahlige Teil)
                  UBR1  = 0x01   //
                  UMCTL = 0xD5   // Modulation: 0.667 (Nachkomma-Anteil von 8MHz/19200)
                                 // 5/8 = 0.625 (das kommt 0.67 sehr nahe, also müssen 5
                                 // Bits in UMCTL gesetzt werden. Z.B. 0xD5 = %11010101)
```

Unter [GCC05] steht ein Berechnungsprogramm für die oben beschriebene Vorgehensweise zur Verfügung. Eine Tabelle mit den häufigsten Baudraten findet man in [Ti03, Kapitel 14-17].

Bit-Timing Fehler:
Das Timing (Zeiteinteilung) für jedes übertragene Zeichen ist die Summe der individuellen Bit-Timings. Durch Modulation eines jeden Bits mit Hilfe von UMCTL wird der kumulative Fehler verringert. Der individuelle Bitfehler wird nach [Ti03] berechnet durch:

$$Error[\%] = \left\{ \frac{Baudrate}{Takt} \times \left[(j+1) \times UBRx + \sum_{i=0}^{n-1} m_i \right] - (j+1) \right\} \times 100\%$$

Mit:

	Baudrate:	die gewünschte Baudrate
	Takt:	dem USART zugeführte Taktschiene (MCLK, SMCLK, ACLK)
	j:	Bitposition (0 für Startbit, 1 für ASCII-Bit[0], 2 für ASCII-Bit[1] usw.)
	UBRx:	Inhalt der Register UBR0 und UBR1 (der ganzzahlige Teil der Division)
	m_i:	Inhalt des korrespondierenden Modulations-Bits aus UMCTL

Für die Übertragungsregeln dieser Arbeit gilt:

Baudrate: 115.200 Bits/s

Takt: 8.000.000 Hz

UBRx: Hex 45 (entspricht Dez 69)

UMCTL: Bin 10101010 (Vier gesetzte Bits)

Daraus ergeben sich trotz Modulation durch UMCTL folgende Bitfehler [Tabelle 5]:

Bit	Soll [µs]	Ist [µs]	Fehler [µs]	Fehler [%]
Startbit	8,68	8,62	+0,0555	+0,64
ASCII-Bit[0]	17,36	17,38	-0,0138	-0,16
ASCII-Bit[1]	26,04	26,00	+0,0416	+0,48
ASCII-Bit[2]	34,72	34,75	-0,0277	-0,32
ASCII-Bit[3]	43,30	43,38	+0,0277	+0,32
ASCII-Bit[4]	52,08	52,12	-0,0416	-0,48
ASCII-Bit[5]	60,76	60,75	+0,0138	+0,16
ASCII-Bit[6]	69,44	69,50	-0,0555	-0,64
ASCII-Bit[7]	78,12	78,12	+0,0000	+0,00
Stopbit	86,81	86,87	-0,0694	-0,80

Tabelle 5 **Reduzierte Bitfehler durch Modulation**

Das Ergebnis zeigt einen Maximalen Bit-Timing Fehler von 0,80% einer ganzen Bit-Periode. Statt mit der gewünschten Baudrate von 115.200 Bit/s wird der USART mit 115.108 Bit/s senden.

6.2.6 Fehlerbehandlung

Der Fehlererkennung und -behandlung sollte in einem embedded System, wie es hier vorliegt, besondere Beachtung geschenkt werden. Der Benutzer sollte den Applikationskontroller anschließen können ohne sich weiter darum kümmern zu müssen. Tritt während des Betriebs dennoch ein Fehler auf, so muß in geeigneter Weise auf ihn reagiert werden. Für die Reaktion wurde eine Fehlerbehandlungsroutine implementiert, die dem Benutzer das Auftreten und die Herkunft des Fehlers anzeigt. Wegen der kompakten Bauart des Applikationskontrollers stehen dafür nur begrenzte Möglichkeiten zur Verfügung. Auf den Einbau eines LCD-Displays wurde daher verzichtet. Statt dessen gibt der Applikationskontroller den Fehler optisch über eine Leuchtdiode aus und versucht danach die erneute Initialisierung aller Komponenten. Schlägt dies fehl, z.B. weil kein Reader oder Lichtschranken angeschlossen sind, wiederholt sich die Fehlerbehandlungsroutine, bis Reader und Lichtschranken angeschlossen sind.

Bei einem Programmabsturz kommt die für Mikrocontroller bekannte Technik des Watchdogs zum Einsatz. Ein Watchdog ist eine Schaltung (im Mikrocontroller integriert), die in einem solchen Fall einen Reset auslöst, damit der Mikrocontroller seine Aufgabe wieder aufnehmen kann. Technisch realisiert wird dies über einen Zeitgeber (Timer), der per Software regelmäßig zurückgesetzt werden muß. Ist das ausführende Programm im Mikrocontroller abgestürzt, kann es diesen Timer auch nicht mehr zurücksetzen und der Watchdog löst einen Reset des Systems aus.

Anzeige der Fehler durch Leuchtdiode (LED):
In der Frontplatte des Applikationskontrollers sind drei LEDs angeordnet, die über den Betriebszustand des Gerätes informieren. Die LED "Power" leuchtet, wenn das Gerät mit Spannung versorgt wird. Die LED "Active" leuchtet, wenn das Gerät sich in einem Lesevorgang befindet oder die Initialisierung des Readers (Reset) vornimmt. LED "Error" blinkt im Falle eines Fehlers; die Anzahl der Blinkpulse leitet sich aus dem Fehlercode ab, der in den Fehlerzustand geführt hat. Die Fehlercodes sind im Anhang aufgeführt. Dabei steht die Nummer hinter jedem Fehlercode für die Anzahl der Blinkpulse der "Error"-LED. Dadurch kann der Benutzer die Fehlerursache feststellen.

7 Systemverifizierung

Was der Applikationskontroller leisten soll, wurde in den Anforderungen [Kapitel 2, Projektspezifikation] definiert. Die Grundlagen, die für das Verständnis der Applikation und deren Arbeitsweise nötig sind, wurden in [Kapitel 3, Grundlagen] dargestellt. Die Auswahl und Erarbeitung der Hardware sowie Entwicklung und Programmierung der Software sind in [Kapitel 4, Systemimplementierung] und [Kapitel 6, Softwareentwicklung] ausgeführt worden. Nun soll die Frage beantwortet werden, ob die einzelnen Schritte der Entwicklung zum gewünschten Ziel führten.

Bild 47 **Der Applikationskontroller besitzt zwei serielle Schnittstellen (RS-232) und zwei Eingänge für Lichtschranke 1 und 2. Drei Leuchtdioden signalisieren den momentanen Zustand.**

Die sieben Module der Software wurden zunächst einzeln getestet. Testtreiber stellten sicher, daß die Kommunikation zu den externen Partnern (Reader und Host) funktionierte. Alle für den Betrieb erforderlichen Befehle an den Reader (z.B. setParameter(), getID()) sind erfolgreich ausgeführt worden. Die Berichterstattung ausgelesener Transponder-IDs an den Host wurde sichergestellt. Die Module wurden zu einem Programm zusammengebunden und das Zusammenspiel aller Module wurde verifiziert. Das in [Kapitel 6.2.2] entworfene Automatenmodell wurde implementiert und lief stabil. Das Programm wurde in die zeitlich parallel aufgebaute Hardware des Applikationskontrollers eingespielt. Das Ergebnis zeigt [Bild 47].

Im laufenden Betrieb zeigte der Applikationskontroller sein erwartetes Verhalten. Nach dem Anschluß von Lichtschranken, Reader und Host wurden die Transponder-IDs korrekt an den RDT400 gesendet [Bild 48]. Dazu wurden sie per Hand an Lichtschranke 1 und 2 vorbeigeführt,

so daß sie das Feld der Lesegerätantenne durchdrangen. Die Anzahl der Lesungen hing wesentlich von der Zeit ab, die sie im Lesefeld verbrachten. Die Leserate kann somit bestimmt werden und unter verschiedenen Bedingungen weiter getestet werden.

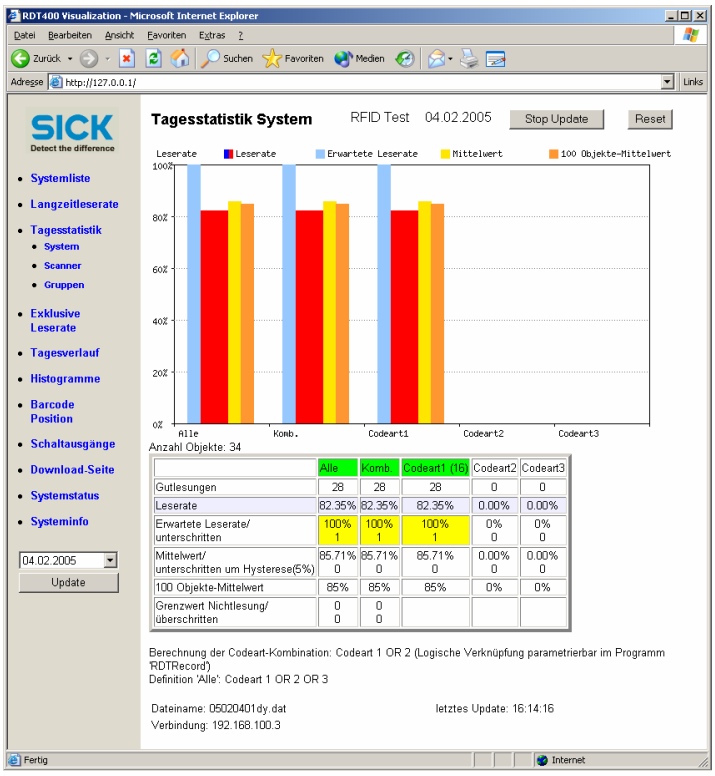

Bild 48 **Über einen Webbrowser kann auf den RDT400-Server zugegriffen werden. Er liefert Statistiken über die Erfassungssicherheit des RFID-Systems.**

8 Überlegungen zur Erfassungssicherheit

Im folgenden Kapitel werden einige Überlegungen bezüglich der Erfassungssicherheit dargelegt, die mit Hilfe des entwickelten Applikationskontrollers noch zu evaluieren wären:

Die RFID-Technik nach ISO 15693 läßt eine große Bandbreite möglicher Einsatzzwecke zu. Daher muß das RFID-System auf die jeweilige Applikation konfiguriert werden. Dabei kommen wichtige Fragen auf, die vor dem Einsatz geklärt werden müssen. Wie z.B. die Frage nach der Lesedistanz. Das heißt, auf welche Entfernung soll ein Objekt erfaßt werden. Dann muß beantwortet werden, was überhaupt erfaßt werden soll: Mensch, Tier, statische/bewegte Objekte, Material/Anzahl der Objekte? Wichtig ist auch die Frage nach der Umgebung des Systems: Das heißt, wird die Erfassung unter erschwerten Bedingungen wie z.B. Absorption, Reflexion, elektromagnetische Störquellen etc. betrieben. Einige dieser Störfaktoren werden im kommenden Abschnitt näher durchleuchtet.

Bei der Absorption geht Wellenenergie in andere Energieformen (Wärme) über, daß heißt Wellenenergie wird verzehrt. Im Gegensatz zur Streuung, bei der die Strahlung nur ihre Ausbreitungsrichtung verändert, aber trotzdem Strahlung bleibt [Vogel95]. Der Einfluß von Metall (Reflexion) und Flüssigkeiten (Absorption) variiert je nach genutzter Sendefrequenz. Generell kann gesagt werden, daß die Störanfälligkeit bei Flüssigkeiten mit höheren Frequenzen zunimmt. Wasser nimmt die Strahlungsenergie auf (vgl. Mikrowellenherd). Bei den hier betrachteten 13,56 MHz kann ein Transponder jedoch ohne weiteres auf einen Kanister Wasser montiert werden, ohne daß Störungen auftreten. Bei Metall im Lesefeld verhält es sich genau umgekehrt. Die Störwirkung von Metall nimmt von nieder- nach hochfrequenten RFID-Systemen ab. Bei 13,56 MHz hat Metall einen negativen Einfluß, dabei hindern nicht die magnetischen Eigenschaften, sondern die elektrische Leitfähigkeit der Metalle. Daher können auch Aluminiumkonstruktionen im Feld das Auslesen durch Wirbelströme beeinträchtigen. Es werden aber mittlerweile RFID-Transponder angeboten, die den Einsatz auf metallischen Objekten erlauben (RFID On Metal Label von Schreiner-Logidata, München).

Einen wichtigen Einfluß auf die Erfassungssicherheit hat zudem der Leseabstand zwischen Transponder und Leseantenne. Im 13,56 MHz Band gilt bei 4 Watt Sendeleistung ein typischer Leseabstand von 1 bis max. 1,6 Meter. [Bild 49] zeigt die Leserate des in dieser Arbeit benutzten RFID-Systems in bezug auf die Distanz zwischen Transponder und Leseantenne. Die Daten wurden im Testcenter Reute (Freiburg) der Firma SICK AG für die vorliegende Arbeit erhoben.

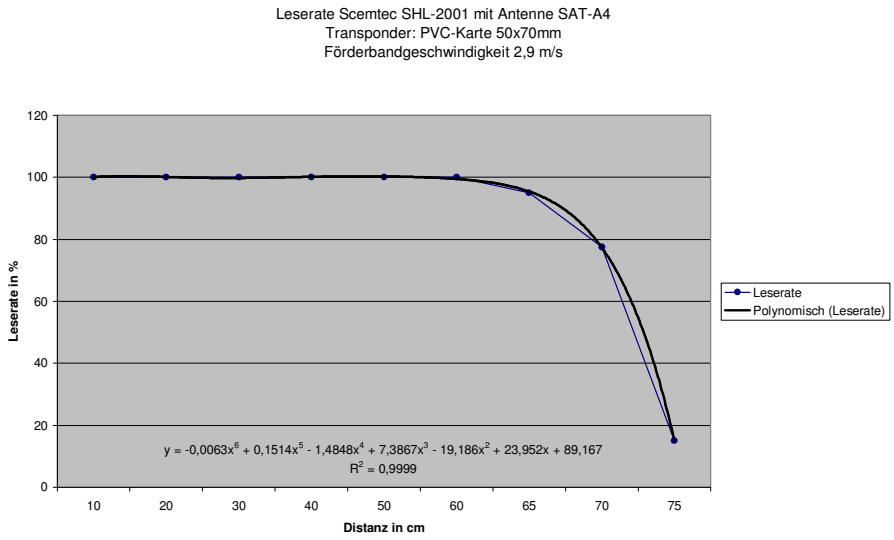

Bild 49 **Leserate in Abhängigkeit der Distanz zwischen Transponder und Leseantenne,**

Ein deutlicher Abfall der Leserate ist ab ca. 60 cm festzustellen, was die Bedeutung der Lesedistanz für die jeweilige Applikation verdeutlicht.

Daneben gibt es noch weitere Faktoren, die die Erfassungssicherheit eines RFID-Systems mindern. Dazu zählen z.B. elektromagnetische "Störsender": Bei großer Distanz zwischen RFID-System und Störsender können jedoch nur Störungen auftreten, wenn die Störsender stark genug sind. Die Zerstörung durch Feldeinwirkung ist bei den herkömmlichen Transpondern zur Elektronischen Artikelsicherung [vgl. Kapitel 3.1.2, 1 Bit / n Bit] standardmäßig vorgesehen (Deaktivierung an der Kasse). Alle Transponder können durch ein ausreichend starkes elektromagnetisches Feld zerstört werden. Generell ist dies wegen der hohen erforderlichen Feldstärke nur aus unmittelbarer Nähe möglich. Es gibt Hinweise darauf, daß Funkeninduktoren oder in unmittelbarer Nähe stattfindende Hochspannungs-Schaltvorgänge ausreichende Spannungsspitzen in Transpondern induzieren, um die Chips zu beschädigen.

Eine Störung im Nahbereich kann aber ebenso entstehen, wenn zwei Leserantennen parallel betrieben werden. Dabei müßte der Wellenberg der einen Antenne in das Wellental der zweiten Antenne ausgerichtet sein. Die Wechselfelder (gleicher Frequenz) löschen sich dann aus, weil die Summe beider Wellen im Punkt des Transponders Null ist. Dieser Effekt kann sich aber auch zu Nutze gemacht werden, indem man mehrere Antennen phasenrichtig ausrichtet. Wenn ein Wellenberg auf einen weiteren Wellenberg trifft ergibt sich theoretisch eine Verdopplung der Feldstärke.

Desweiteren können Transponder von der Erfassung abgeschirmt werden, indem man sie in metallische Folie einwickelt (Alufolie) oder in eine mit Metallstreifen ausgestattete Reisetasche legt. Zwar arbeitet das RFID-System ohne Sichtkontakt zwischen Sender und Leser, für die Anwendung muß jedoch sichergestellt werden, daß der Transponder nicht abgeschirmt wird.

9 Ausblick

Neben den vorab aufgezählten Störfaktoren, die bedacht werden müssen, sollen in diesem Ausblick eine Reihe an Verbesserungen der Performance bei Bestimmung der Leserate vorgestellt werden.

Bei einem RFID-Erfassungssystem, wie es in dieser Arbeit eingesetzt wird, könnte eine Verbesserung bereits durch den Einsatz von Background-Funktionen (BF) im Reader erreicht werden. Die BF wird bei Initialisierung des Readers eingeschaltet und läuft danach selbständig. Bei erfolgreichem Lesevorgang sendet der Reader automatisch und asynchron die gelesene ID an den Applikationskontroller. Dieser speichert die ID und vergleicht sie mit bereits empfangenen IDs, um die Leserate zu bestimmen. Die asynchrone Kommunikation zwischen Reader und Applikationskontroller kann dabei durch den eingehenden Interrupt auf der RX-Leitung der RS-232 Schnittstelle behandelt werden. Das Polling auf Erzeugen eines neuen "Inventorys" (Bestandsliste aller Transponder im Lesefeld) und damit erhöhter Last auf der RS-232 Schnittstelle könnte damit vermieden werden. Auf Anfrage bei der Firma Scemtec GmbH werden zur Zeit keine Background-Funktionen im Betriebsmodus "ISO 15693" unterstützt und befinden sich auch nicht in Planung (für I*Code und TagIt Transponder werden BF angeboten).

Die serielle Schnittstelle fungiert als das Nadelöhr, durch das die Befehle und Daten zwischen Reader und Applikationskontroller ausgetauscht werden. [Bild 50] zeigt die Anzahl der Lesezyklen in Abhängigkeit der Baudrate unter Laborbedingungen. Mit steigender Baudrate erhöht sich die Anzahl der ausgelesenen Transponder pro Zeiteinheit (hier: 60 Minuten). Dabei wird die besondere Anforderung der Pulkerfassung gegenüber der Einzelerfassung miteinbezogen, da sich insgesamt vier Transponder im Lesefeld der Antenne aufhielten.

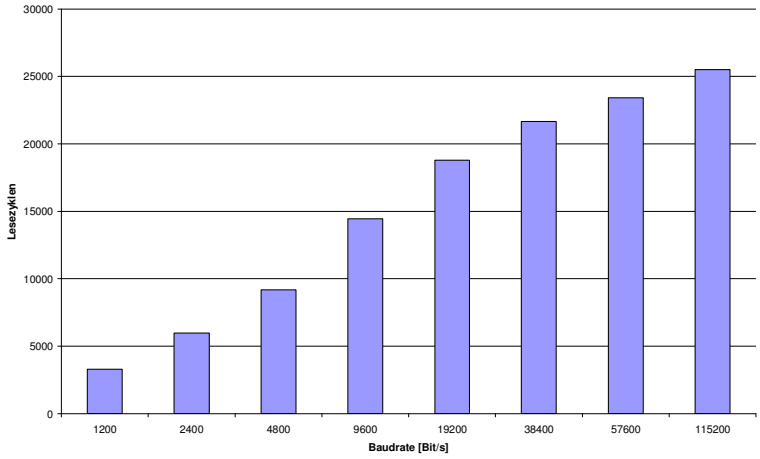

Bild 50 **Lesezyklen in bezug auf die verwendetet Baudrate. Im Lesefeld der Antenne befanden sich vier Transponder und wurden 60 Minuten lang ausgelesen.**

Wenn davon ausgegangen wird, daß sich bei der Objekterfassung am Fließband nur ein einzelner Transponder im Lesefeld der Antenne befände, könnte eine weitere Verbesserung der Performance durch Verwendung des Befehls "Looped Address Scan" erreicht werden. Dieser Befehl wird vom STX/ETX Protokoll der Firma Scemtec unterstützt. Der Reader wiederholt dabei kontinuierlich einen Single-Timeslot-Inventory Versuch, bis ein Transponder entdeckt wurde. Da dem Transponder nur ein Zeitfenster (Timeslot) zum Antworten (Senden seiner ID) gegeben wird, handelt es sich implizit um eine Funktion, die das Merkmal "Anti-Kollision" nicht unterstützt. Durch Umgehen des Anti-Kollisions-Algorithmus wird sich der Zeitintervall zwischen den Lesezyklen verringern. Tests zur Implementierung und Messung der resultierenden Leserate könnten Bestandteil weiterer Forschungsarbeit sein.

Die größte Verbesserung der Performance dürfte sich durch Zusammenlegen der beiden Funktionalitäten "ID lesen" und "ID sammeln" ergeben. Das Nadelöhr der seriellen Datenübertragung über RS-232 würde wegfallen, wenn der in [Bild 43] aufgezeigte Algorithmus zur Bestimmung der Leserate im IC des Lesegeräts zur Verfügung stünde. Das ein- und ausleitende Triggern über Lichtschranken sollte weiter bestehen bleiben, um das Magnetfeld der Leseantenne nur bei Bedarf aufzubauen und damit Energie zu sparen. Der nachfolgende Versand der Leseraten-Information (ID und Anzahl der Lesungen) an den Host und damit auch die Implementierung des RDT400 Protokolls müßten ebenso im IC des Readers vorgenommen werden.

Um die Erfassungssicherheit weiter zu erhöhen, könnte man mehrere Antennen gleichzeitig an das Lesegerät anschließen (bei Scemtec SHL-2001: bis zu vier). Diese Antennen können parallel mit halbierter Leistung (Splitbetrieb) oder zeitlich abwechselnd (Multiplexbetrieb) angesteuert werden. Dadurch könnte sich die Ausleuchtung des Lesefelds verbessern und ungünstig posi-

tionierte Transponder sicherer ausgelesen werden. Vor allem, wenn man die Antennen versetzt (Ausrichtung oder Winkel) anordnet. Der Splitbetrieb wäre einmalig bei Initialisierung des Readers vorzunehmen, der Multiplexbetrieb müßte dynamisch zwischen dem Senden von Lesebefehlen erfolgen.

Der, im Rahmen dieser Arbeit erstellte, Applikationskontroller kommuniziert zum Host über eine RS-232 Schnittstelle. Der Host stellt auch eine Ethernet-Schnittstelle zur Verfügung, die den Vorteil multipler Verbindungen zu mehreren Applikationskontrollern innerhalb eines Firmennetzwerkes bietet. Zum jetzigen Zeitpunkt übernimmt ein dem Applikationskontroller nachgeschaltetes, externes Gerät (Bridge) die Konvertierung von RS-232 auf Ethernet. Bestandteil weiterer Arbeiten am Applikationskontroller sollte daher die Implementierung eines TCP/IP-Stacks für die Kommunikation zum Host sein. Als Grundlage der Implementierung könnten hierfür der "uIP Stack" von Adam Dunkel (Opensource) oder der "EasyWeb2 TCP/IP Stack" von Texas Instruments dienen. Für beide Varianten könnte der "Crystal LAN Controller" der Firma Cirrus Logic, USA die Aufgaben der physikalischen Schicht (PHY) übernehmen.

10 Schlußwort

Ziel dieser Arbeit bestand darin, einen Applikationskontroller zu erstellen, der zur Optimierung der Erfassungssicherheit des beschriebenen RFID-Systems dient. Der selbstentwickelte Applikationskontroller kann zwar nicht von sich aus die Erfassungssicherheit optimieren, aber das Novum an ihm ist, daß er den Erfassungsvorgang automatisiert. Dadurch kann zum ersten Mal eine große Anzahl an Erfassungen durchgeführt werden, die die Aussagekraft der Leserate eines RFID-Systems statistisch beweist. Solch ein Applikationskontroller liegt nun am Ende dieser Arbeit vor. Hiermit ist die Grundlage für weitergehende Entwicklungen bezüglich der Erfassungssicherheit gegeben.

Für die Forschung ist der Applikationskontroller von enormer Wichtigkeit, nicht nur, weil RFID so viele Vorteile gegenüber dem Barcode aufweist, sondern auch, weil der Wirtschaftsmarkt und große Konzerne weltweit eine Auseinandersetzung mit diesem Thema erfordern. In den USA z.B. verlangen Konzerne wie Wal-Mart schon heute von ihren Lieferanten, daß sie bis zum Jahr 2006 mit RFID anstelle des Barcodes arbeiten [Ident05].

Doch sollte man nicht außer acht lassen, daß das RFID-System noch nicht so weit erforscht ist, daß man 100%ig davon ausgehen kann, daß es wirklich ein adäquater Ersatz für den Barcode ist. Man bedenke z.B. die hohe Umweltbelastung, wenn von nun an jedes zu verkaufende Produkt statt einfacher Striche Silizium und Kupfer in Form eines Transponders enthält. Auch preislich liegt der Transponder noch auf lange Sicht im Nachteil gegenüber dem Barcode.

Nichtsdestoweniger lohnt es sich, die Forschungsarbeit in diese Richtung weiterzubetreiben, denn wie in dieser Arbeit deutlich wird, könnte sich ein RFID-System als sehr nutzbringend und effektiv für die Wirtschaft erweisen.

11 Glossar

Aktiver Transponder Ein RFID-Transponder, der einen eigenen Sender zur Rückgabe der Information besitzt, statt das Signal des Readers nur moduliert zu reflektieren wie die passiven Transponder. Die meisten aktiven Transponder besitzen auch eine eigene Energieversorgung in Form einer Batterie. Aktive Transponder können auf eine Entfernung bis zu 100 m ausgelesen werden, sind aber teuer (> 20 €). Sie werden vorwiegend benutzt, um wertvolle Gegenstände über eine weite Entfernung zu verfolgen. Aktive Transponder können über größere Distanz zur Leseantenne angesprochen werden als passive Transponder.

Amplitude Physikalische Bezeichnung für die maximale Auslenkung einer Schwingung bzw. einer Welle aus der Mittellage.

Antenne Wandelt hochfrequente Wechselströme in elektromagnetische Wellen (Sendeantenne) um und umgekehrt (Empfangsantenne). In Form einer Spirale erhöht sich der Wirkungsgrad des verwendeten Stroms in Bezug auf die Feldstärke.

Applikation Anwendung, Verwendung, Nutzung oder Gebrauch.

Automatische Identifikation (Auto-ID) Ein Begriff, der sämtliche Methoden beschreibt, Daten zu sammeln und diese direkt (ohne menschliches Eingreifen) in ein Computersystem einzugeben. Technologien, die zur Auto-ID gezählt werden, sind z.B. Barcode, Biometrie, RFID und Spracherkennung.

Barcode Der Strichcode (bar = Strich) ist eine maschinenlesbare Schrift, die aus verschieden breiten Strichen und Lücken besteht. Sie kann über optische Abtaster, sogenannte Strichcodelesegeräte (auch Scanner) maschinell gelesen und in der EDV weiterverarbeitet werden. Erstmals am 26. Juni 1974 wurde ein Strichcode von einem Scanner in einem Supermarkt in Ohio, USA eingelesen: auf einer Packung Kaugummi der Marke Wrigleys.

Binär	Das binäre Zahlensystem benutzt im Gegensatz zum bekannten Dezimalsystem (Zahlen von 0 bis 9) nur die Wert 1 und 0 (bzw. Zustände "Ja" und "Nein", "ON" und "OFF"). Es ist Grundlage digitaler Datenübermittlung und wird auch "duales System" genannt.
Bit	Ein Bit ist die kleinste digitale Einheit, mit der ein Computer "rechnen" kann. Die Einheit entspricht entweder logisch 0 (Null) oder 1 (Eins). Die Namensgebung setzt sich aus den Begriffen "binary" (binär) und "digit" (Ziffer) zusammen.
CSMA/CD	"Carrier Sense Multiple Access with Collision Detection" Stochastische Zugriffsmethode auf ein Kommunikationsmedium mit Signalabtastung und Kollisionsfeststellung. Sendebereite Stationen hören das physikalische Kommunikationsmedium ab und belegen es, falls keine Nachrichtensignale anderer Stationen festgestellt werden. Während des Sendens wird der Übertragungskanal weiter abgehört. Eine Sendung wird abgebrochen, falls sich die eigene Nachricht mit der Nachricht einer anderen Station irreversibel überlagert, d.h. wenn eine Kollision auftritt. Danach wartet die sendebereite Station eine zufällige Zeit bis zum erneuten Senden. [Schn04]
Erfassungssicherheit	Genauigkeit und Kontrolle der Erfassung. Genauigkeit wird erhöht durch Validitäts- und Plausibilitätsprüfungen, durch maschinell lesbare Darstellung der Daten, durch Ausschaltung manueller Eingriffe, durch definierte Umgebungsvariablen (Temperatur, Feuchtigkeit, EMV), durch Unterdrückung von äußeren Einflüssen. Fehleingaben werden verringert durch Plausibilitätsabgleich.
Frequenz	Die Anzahl der Wiederholungen einer kompletten Welle in einer Sekunde, ausgedrückt in Hertz (Hz). 1 Hz entspricht dabei einer vollständigen Welle in einer Sekunde. 1 kHz entspricht 1.000 Wellen in einer Sekunde. 1 MHz entspricht 1.000.000 Wellen in einer Sekunde. RFID-Transponder benutzen Low, High, Ultra-High und Mikrowellen-Frequenzen. Jede Frequenz besitzt spezifische Vor- und Nachteile, so daß die optimale Frequenz für eine Anwendung ermittelt werden muß.

Host Der Host nimmt in dieser Arbeit Transponder-IDs vom Applikationskontroller entgegen (incl. Anzahl der Lesungen). Er visualisiert und speichert alle erfaßten IDs. Siehe RDT400.

ID Auch UID (Unique identifier) genannt. Weltweit einzigartige Kennung, mit der ein Transponder (und damit ein Objekt) identifiziert werden kann. 2^{64} Kombinationsmöglichkeiten ergeben 18.446.744.073.709.551.616 Kennungen. Zentrale Vergabe durch ISO/IEC 7816-6/AM1 (ähnlich ISBN-Nummer für Buchtitel).

JTAG Verfahren zum Debugging elektrischer Hardware und steht für "Joint Test Action Group". Es entstand durch einen Zusammenschluß von Halbleiterherstellern im Jahre 1985/86. Es wurde ein Standard erarbeitet, welcher in der Norm IEEE 1149.1-1990 festgehalten ist. Das Verfahren ist auch als Boundary Scan Test bekannt. Zweck ist, integrierte Schaltkreise, welche sich bereits in einer Arbeitsumgebung befinden, beispielsweise verlötet auf einer Platine, auf Funktion zu testen. Dazu werden integrierte Schaltkreise in ihrem inneren Aufbau durch JTAG-Komponenten ergänzt.

Kollision Irreversible Überlagerung zweier Sendesignale.

Lesefeld Räumlicher Bereich um die Antenne eines RFID-Lesegeräts. Meist in Keulenform. Transponder außerhalb des Lesefeldes werden vom magnetischen Wechselfeld der Antenne nicht durchdrungen und können deshalb nicht ausgelesen werden. Je höher die Arbeitsfrequenz eines RFID-System ist (ca. 100 kHz bis 20 GHz), desto größer ist i.d.R. die Reichweite des Lesefelds.

Leserate Prozent der erfolgreichen Lesungen in Bezug auf 100 Leseversuche eines bestimmten Datenträgers (Transponder).

MAX232 Elektronischer Baustein der Firma Maxim. Pegelwandler, um Ausgangssignale eines Mikrocontrollers von 0 bis 3 V auf -15 bis +15 V abzubilden. Industriestandard bei Kommunikation über RS-232.

Mikrocontroller Winziger PC (ein Chip) ohne Betriebssystem. Neben CPU, ROM und

RAM enthält ein Mikrocontroller zudem Peripherie: Timer, Interrupttunterstützung, Analog Digital Wandler und führt Pins nach außen zur Kommunikation (UART, Ethernet, CAN, I^2C). Wird meist in den Sprachen Assembler oder C programmiert.

Modulation Ein Verfahren, durch das Verändern von Funkwellen, die zwischen Reader und Transponder wandern, Informationen zu übertragen. Die Wellen, auf verschiedene Art geändert, können vom Reader aufgenommen und in binäre Einsen und Nullen umgesetzt werden. So können die Amplitude der Wellen erhöht oder verringert (Amplitudenmodulation) bzw. nach vorne verlagert (Phasenmodulation) werden. Man kann die Frequenz variieren (Frequenzmodulation) oder die Daten können über die Variation der Dauer des Impulses übertragen werden (Impulsdauermodulation).

Passiver Transponder Eine RFID Einheit, die seine Energie aus dem Magnetfeld der Sende-/Empfangseinheit bezieht. Vergleich "Aktiver Transponder".

Reader RFID-Lesegerät. Liest auf Anfrage Transponder aus, die sich im Lesefeld seiner Antenne befinden. Er kann auch Daten auf einen Transponder schreiben. Siehe Scemtec SHL-2001.

RFID Radio Frequenz Identifikation. Sie wird in der Automatisierungstechnik zur drahtlosen, automatischen Erfassung von Objekten benutzt. Siehe Barcode. RFID ist keine neue Technologie. Beim US Militär werden RFID oder deren Vorgängertechnologien bereits seit ca. 1940 genutzt, um den Verbleib von Nachschub wie Treibstoff oder Munition zu verfolgen oder die Freund-Feind-Erkennung alliierter Flugzeuge zu ermöglichen. Seit 1977 sind RFID-Systeme für zivile Anwendungen freigegeben. Zu den ersten Anwendungen zählten Ende der 80er Jahre Transponder für die Tieridentifikation. [Krem04]

RS-232 Name eines seriellen asynchronen Übertragungsstandards, genormt durch die amerikanische Electronic Industries Alliance (EIA). Arbeitet mit Signalpegeln im Bereich von +3 bis +15 V zur Darstellung einer logischen 0 und -3 bis -15 V zur Darstellung einer logischen 1. Datenraten bis 115200 Bits/s möglich. Verbindung via SUB-D Stecker

(9-polig) bis zu 20 m.

SHL-2001 Gerät zum Auslesen und Beschreiben verschiedener Typen von Transpondern (Philips I*Code, TagIt, Hitag, ISO 15693) der Firma Scemtec GmbH. Klasse 13,56 MHz.

Transponder Drahtloses Kommunikationsgerät, das eingehende Signale aufnimmt und automatisch antwortet. Der Begriff Transponder ist zusammengesetzt aus den Begriffen Transmitter (Sender) und Responder (Empfänger). Transponder können passiv oder aktiv sein.

UART Universal Asynchronous Receiver Transmitter. Gängige serielle Schnittstelle an PCs und Mikrocontrollern. Siehe RS-232.

Verstimmen HF- und UHF-Antennen sind darauf abgestimmt, RFID-Wellen einer bestimmten Länge von einem Reader zu empfangen. Stehen HF- und UHF-Antennen nahe an Metall oder metallischen Materialien, so können die Antennen verstimmt werden. Das Ergebnis ist eine schlechte Leseperformance.

12 Geräteverzeichnis

TI's MSP430F149 wurde programmiert unter MS Windows 2000 mit Intel Pentium Prozessor.

Dazu wurde folgende Hardware genutzt:

- Evaluationboard TS430PM64 Rev.1.1 von Texas Instruments
- Flash Emulation Tool MSP-FET P430IF V1.3 von Texas Instruments

Entwicklungsumgebung:

- IDE von Firma IAR Systems AG, Parsdorf V.2.32
- Kompiliert mit IAR MSP430 C Compiler V2.21B/W32
- Gelinkt mit IAR Universal Linker V4.56E/386
- Eingestellte Optionen für Konfiguration Release (abweichend von Einstellung "default"):
    - Device = MSP430F149
    - Hardware multiplier = On
    - Double floating point size = 32 Bit
    - Output file = Executable
    - Library = IAR CLIB (C Library), Compact math libraries
    - Formatted write = Medium
    - Formatted read = Medium

Externe Komponenten:

- Scemtec SIR-2600 RFID Mid Range Reader, Gerätenummer 0013
- Scemtec SHL-2001 RFID Long Range Reader, Gerätenummer 0009
- Scemtec Loopantenne
- Scemtec Antenne SAT-A4
- Scemtec Vicinity-Chipkarten Transponder ISO 15693
- SICK Reflexionslichtschranke WL18 2P430
- SICK Remote Diagnostic Tool 400 (RDT400) V2.0
- Tektronix TDS220 Oszilloskop, Gerätenummer 102005

Der Applikationskontroller unterstützt:

- Scemtec STX/ETX Protokoll in Version 4.13
- SICK RDT400 Data String Format in Version 2.0

13 Literaturverzeichnis

[ABI05] "RFID Vendor Assessment - Identifying RFID Player, Product & Service Leadership", http://www.abiresearch.com/reports/RVA.html, Aufruf vom 29.01.2005

[Auto03] "Technical Report: 13.56 MHz ISM Band Class 1 Radio Frequency Identification Tag Interface Specification: Recommended Standards, Version 1.0.0", AUTO-ID CENTER (2003), http://www.epcglobalinc.org/standards_technology/Secure/v1.0/HF-Class1.pdf, Abruf vom 13.01.2005

[Beier02] "Taschenbuch Mikroprozessortechnik", 2. Auflage, T. Beierlein / O. Hagenbruch, Fachbuchverlag Leibzig, 2001

[Berg98] "Contactless smart card standards and test methods“ von D. Berger, IEEE Workshop Tagungsband, Berlin, 1998

[BSI05] "Abhörmöglichkeiten der Kommunikation zwischen Lesegerät und Transponder am Beispiel eines ISO 14443 Systems" von T. Finke, H. Kelter, BSI, http://www.bsi.de/fachthem/rfid/Abh_RFID.pdf, Abruf vom 13.01.2005

[Camp88] "V24/RS-232 Kommunikation", Joe Campbell, 6. Auflage, Sybex-Verlag Düsseldorf, 1988

[Chris00] "Induktiv gekoppelte passive Transpondersysteme – Analyse, Modellierung, Simulation, Verifikation, Optimierung" von N. Christoffers, Diplomarbeit der Gerhard-Mercator-Universität Duisburg, 2000

[Dank97] "Praxis der C Programmierung" von Jürgen Dankert, B.G. Teubner Stuttgart, 1997

[Doug98] “Real-Time UML, Developing Efficient Objects For Embedded Systems" von Bruce Powel Douglass, Addison Wesley Longman Inc., 1998

[Elek05] "Grundlagen der Elektrokühlung" von E. Reiff, Elektronik Ausgabe 05/2005, Weka Verlag Poing, 2005

[Elin04] "Elektronik Industrie" von D. Morgenroth, Ausgabe 12/2004, Hüthig Verlag, 2004

[Erle02] "C Programmieren von Anfang an" von Helmut Erlenkötter, 5. Auflage, Rowohlt Hamburg, 2002

[Fair04] "CNY17-2 Phototransistor Optocouplers Datasheet" von Fairchild Semiconductor Corporation, 2004, http://www.fairchildsemi.com/ds/CN/CNY17-2.pdf, Abruf vom 07.02.2004

[Fhg01] "Praxishandbuch Informations- und Kommunikationstechnologie" von Fraunhofer Institut für Materialfluß und Logistik, 2001, http://www01.iml.fhg.de/~praxishb/phb_offiziell.pdf, Abruf vom 14.02.1005

[Fhg04] "Das Internet der Dinge" von T. Hompel, Fraunhofer Institut für Materialfluß und Logistik (IML), 2004, http://www.vdeb.de/index.htm?/studien.rfid.htm, Abruf vom 24.02.2005

[Fink02] "RFID-Handbuch" von Klaus Finkenzeller, 3. Auflage, Carl Hanser Verlag München, 2002

[Fohl04] "Verhaltensbeschreibung reaktiver Systeme" von W. Fohl, Vorlesungsskript Prozeßlenkung, 2004, http://users.etech.haw-hamburg.de/users/fohl/pl/wse6harel.PDF, Abruf vom 21.02.2005

[Froh89] "Elektrische und magnetische Felder, Einführung in die Elektrotechnik" von H. Frohne, E. Ueckert, Teubner-Studienskripte, 1989

[GCC05] "GCC toolchain for MSP430" von Chris Liechti, Sourceforge Project, 2005, http://mspgcc.sourceforge.net/baudrate.html, Abruf vom 19.02.2005

[Gert95] "Gerthsen Physik" von Helmut Vogel, 18. Auflage, Springer-Verlag Berlin-Heidelberg, 1995

[Heis99] "Warum immer Barcode?" von K. Heise, Logistik Heute Ausgabe 32/1999, Huss Verlag München, 1999

[Ident05] "Schulterschluß im Handel" von Maria Meriemque-Aha, Ident-Magazin für Automatische Datenerfassung & Identifikation, 02/2005, http://ident.de/index.php?id=526, Abruf vom 17.02.2005

[Koln01] "Drahtlose Signal- und Energieübertragung mit Hilfe von Hochfrequenztechnik in CMOS-Sensorsystemen" von Stephan Kolnsberg, Dissertation der Gerhard-Mercator-Universität Duisburg 2001

[Krem04] "Das Internet der Dinge“ von S. Krempl, Computerworld 5/2004, http://viadrina.euv-frankfurt-o.de/~sk/Pub/rfid-cw04.html, Abruf vom 11.01.2005

[Lan04] "Lantronix UDS-10/100 User Guide" von Lantronix, CA, USA, 2005, http://www.lantronix.com/pdf/UDS10-UDS100_UG.pdf, Abruf vom 10.01.2005

[Max04] "MAX232 Datasheet (Revision 14)“ von Maxim Integrated Products, 2004, http://pdfserv.maxim-ic.com/en/ds/MAX220-MAX249.pdf, Abruf vom 8.12.2004

[Mieb03] " Wann löst das Smart-Label den Barcode ab?" von S. Eikel, Miebach Logistik GmbH, 2003, http://www.miebach.com/deutsch/public_interface_d/fachartikel_d/transponder-eikel_d.html, Abruf vom 22.01.2005

[Nat01] "Lowdrop Festspannungsregler LM1086“ von National Semiconductor Corporation, 2001, http://www.national.com/pf/LM/LM1086.html, Abruf vom 6.1.2005

[Sam02] "Practical Statecharts in C/C++" von Miro Samek, CMP Books Kansas USA, 2002

[Scem04] "Scemtec STX/ETX Protocol Specification V4.11" von Scemtec Transponder Technology GmbH Reichshof, 2004

[Sick04] "RDT400 Remote Diagnostic Tool V2.0 Operating Instructions" von SICK AG Division Auto Ident, 2004

[Stark96] "Grundlagen der Funk- und Kommunikationstechnik" von L. Starke, Dr. Alfred Hüthig Verlag Heidelberg, 1996

[Ti03] "MSP430x4xx Family User's Guide (Revision D)" von Texas Instruments 2003, http://www.gaw.ru/pdf/TI/micros/msp430/slau056d.pdf, Abruf vom 10.11.04

[Ti04] "MSP430F149 Datasheet (Revision F)" von Texas Instruments, 2004, http://focus.ti.com/lit/ds/symlink/msp430f149.pdf, Abruf vom 13.11.2004

[Ti99] "MSP430 Universal Synchronous/Asynchronous Receive/Transmit Communication Interface (Revision A)" von Texas Instruments, 1999, www.gaw.ru/pdf/TI/app/msp430/slaa049.pdf, Abruf vom 10.11.04

[Trac04] "Traco Power DC/DC-Konverter Datasheet TEN 10 Serie" von Traco Electronic AG, Zürich, Schweiz, 2004 http://www.tracopower.com/datasheet_g/ten10-d.pdf, Abruf vom 20.01.2005

[Wg899] "ISO/IEC JTC1/SC17/WG8 Contactless Integrated Circuit(s) Cards", Working Group 8 within the subcommittee ISO/IEC JTC1/SC17, 1999, http://gippacb.dd6338.kasserver.com/_wg8De/index.html, Abruf vom 22.12.2004

[Zenk05] "Der Lichtelektrische Effekt" von Michael Zenk, Treasure Island Museum, http://www.zes-elektrotechnik.de/ts/tube/beschreibung/lichtelektrischer_effekt.htm, Abruf vom 14.02.2005

14 Anhang

Protokoll zum Versuch, die Leserate durch Ändern verschiedener Parameter bei Initialisierung des Lesegeräts zu verbessern.

Reading 4 Transponder-IDs (ISO 15963 - each 64bit) simultaneously - looped 60 minutes.
Software is irrTool v.0.93 from scemTec
Device is SIR-2600 from scemTec (No. 0013)

Baudrate RS-232 [bit/s]	Tag Request Mode [slow / fast]	Tag Response Mode [slow / fast]	Tag Request Modulation [20% / 100%]	Tag Response Techinque [ASK / FSK]	Delay [ms]	Antenna Power [Watt]	**Cycles [n Reads]**	Acquisition Security [%]	Comment
1200	fast	fast	100%	FSK	-0140h	4	**3290**	100	
2400	fast	fast	100%	FSK	-0140h	4	**5973**	100	
4800	fast	fast	100%	FSK	-0140h	4	**9175**	100	
9600	fast	fast	100%	FSK	-0140h	4	**14454**	100	
19200	fast	fast	100%	FSK	-0140h	4	**18803**	100	
38400	fast	fast	100%	FSK	-0140h	4	**21669**	100	
57600	fast	fast	100%	FSK	-0140h	4	**23415**	100	
115200	fast	fast	100%	FSK	-0140h	4	**25503**	100	
115200	slow	fast	100%	FSK	-0140h	4	**8357**	100	
115200	fast	slow	100%	FSK	-0140h	4	**16398**	100	
115200	fast	fast	100%	FSK	-017Eh	4	**25446**	100	(delay: auto-calibrate)
115200	fast	fast	100%	FSK	-0140h	1	**25479**	100	
115200	fast	fast	20%	FSK	-0140h	4	**0**	0	(0 out of 4 Tags read per cycle)
115200	fast	fast	100%	ASK	-0140h	4	**7319**	38	(1,53 out of 4 Tags read per cy
115200	slow	slow	100%	ASK	-0140h	4	**7159**	100	

Stückliste des Applikationskontrollers:

Pos	Menge	Bezeichnung	Task	Lieferant
1	1	MSP430 EvalBoard	ok	TI
2	4	StiftSockelLeiste für MSP430 (16 Pin)	ok	SICK
3	1	Max232 (DIL-16 Gehäuse)	ok	SICK
4	5	C max232 (100 nF)	ok	SICK
5	1	Sockel für max232 (16 Pin)	ok	SICK
6	4	R max232 (1 kOhm)	ok	SICK
7	4	R max232 (0,5 kOhm)	ok	SICK
8	1	Quarz 8 MHz (Bauart HC18)	ok	Philips
9	1	DC/DC Wandler (24V->5V, 3W)	ok	Traco
10	1	Schraubklemmleiste (10er)	ok	SICK
11	1	Festspannungsregler LM1086 it3,3 (5V->3,3V)	ok	Reichelt
12	1	C festSpRegler (10 uF)	ok	SICK
13	2	Optokoppler CNY 17/ii (24V->3,3V, Lichtschranken)	ok	Reichelt
14	2	R opto (20 kOhm)	ok	SICK
15	2	R opto (10 kOhm)	ok	SICK
16	1	Euro Lochrasterplatine (2,54 mm Raster)	ok	Bopla
17	1	Gehäuse	ok	Reichelt
18	1	DB9 Buchse männlich	ok	SICK
19	3	DB9 Buchse weiblich	ok	SICK
20	1	Bananenstecker Buchse rot	ok	SICK
21	1	Bananenstecker Buchse schwarz	ok	SICK
22	3	LED mit Fassung rot, gelb, grün (2V, 30mA)	ok	Reichelt
23	3	R led (330 Ohm)	ok	SICK

Bopla: BOPLA Gehäuse Systeme GmbH, Borsigstr. 17-25, 32257 Bünde

Philips: Philips Semiconductors GmbH, Stresemannallee 101, 22529 Hamburg

Reichelt: Reichelt Elektronik e. Kfr., Elektronikring 1, 26452 Sande

SICK: SICK IBEO GmbH, Fahrenkrön 125, 22179 Hamburg

TI: Texas Instruments Inc., 13532 N. Central Expressway, M/S 3807 Dallas, Texas, USA

Traco: TRACO Electronic AG, P.O. Box, Jenatschstrasse 1, 8002 Zurich, Schweiz

Der Applikationskontroller zeigt im Fehlerfall folgende Blinkintervalle über die Error-LED an:

Fehler	Blinkintervall
ERROR_NOERR	0
ERROR_UNKNOWN	1
ERROR_TIMEOUT	2
ERROR_BUFFER_TOO_SMALL	3
ERROR_NULL_POINTER	4
ERROR_BUFFER_EMPTY	5
ERROR_BUFFER_OVERFLOW	6
ERROR_UART_FRAMING	7
ERROR_UART_CRC	8
ERROR_READER_NO_CONNECTION	9

Die zu dieser Arbeit beigefügte CD enthält folgende Dokumente:

- Quellcode des Applikationskontrollers
- Datenblätter der Bauteile
- ISO 15693 Spezifikation
- Scemtec STX/ETX Protokoll V4.11 Spezifikation
- Scemtec RFID Lesegerät SHL-2001 Handbuch
- SICK Remote Diagnostic Tool 400 V2.0 Handbuch

Die aufgeführten Dokumente können eingesehen werden bei:

- Prof. Dr. rer. nat. Reinhard Baran, HAW-Hamburg, Berliner Tor 3, 20099 Hamburg
- Prof. Dr. rer. nat. Gunter Klemke, HAW-Hamburg, Berliner Tor 3, 20099 Hamburg

RDT400 Data String mit den vom Applikationskontroller verwendeten Füllwerten:

Feld	**Größe (Byte)**	**Standard-Inhalt (Beispiel)**	**RFID-Inhalt (Beispiel)**	**Bemerkung**
Device ID	3	DC4	DCX	
OTC Nummer	2	01	01	Sollte auf 01 bleiben, da z.Z. nur ein Reader über RS-232 angeschlossen wird
Lfd. Nummer	2	03	03	Lfd. Nr. der Telegramme
Index	4	9999	0001	Lfd. Nr. der erkannten Behälter
Objektlänge	3	015	000	Wird über Lichtschrankensystem ermittelt
Objektabstand	3	406	000	-„-
Objektgeschwindigkeit	2	10	00	-„-
Device Liste	6	680000	800000	Liste der am OTC angeschlossenen Scanner, bei RFID nur einer
Device Fehler Liste	6	000000	000000	Liste, welche Scanner Fehler melden
Outputs	2	E0	80	Status der Ausgänge
Fehlernummer	3	000	000	Fehlernummer
Reserviert	1	_	_	
Separator	1	;	;	
Code-ID	1	e	e	ID für RFID, eventuell noch festzulegen
Codelänge	2	10	16	Abh. von gelesenen Daten und Systemeinstellung, z.B. Transponder-ID oder Transponderdaten
Gelesen-Liste	6	480000	800000	Liste, welche Scanner den Code gelesen haben, bei RFID nur einer
Reserviert	4	____	____	vier Underscore
Scanner, am besten gelesen	2	02	01	Bei RFID der erste und einzige Reader
X-Position	3	000	000	Kann vom RFID-System selbst nicht bestimmt werden
Y-Position	3	-08	000	Kann vom RFID-System selbst nicht bestimmt werden
Z-Position	3	-10	000	Kann vom RFID-System selbst nicht bestimmt werden
Max. X-Position	3	000	000	Kann vom RFID-System selbst nicht bestimmt werden
Min. X-Position	3	000	000	Kann vom RFID-System selbst nicht bestimmt werden
Fehlersicherheit	3	108	010	Bei Barcode die Anzahl der erfolgreichen Scans pro Code, bei RFID Anzahl der erfolgreichen Leseversuche zwischen den 2 Lichtschranken
Gelesener Code	<Codelänge>	1234560001	ABCDEF01 23456789	
<Eventuell weiterer gelesener Code vom Typ 1>	18 + <Codelänge>			
Separator	1	;	;	
<Gelesene Codes vom Typ 2>	<abh. von Codetyp und Länge>			
Separator	1	;	;	
<Gelesene Codes vom Typ 3>	<abh. von Codetyp und Länge>			

Versicherung über die Selbständigkeit

Hiermit versichere ich, daß ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach § 22 (4) ohne fremde Hilfe selbständig verfaßt und nur die angegebenen Hilfsmittel benutzt habe.

(Ort, Datum) (Unterschrift)