DatenFunk

Technische Universität Clausthal
Institut für Informatik

Betreuer
Prof.Dr. Günter Kemnitz
Dipl.Ing. Hossam Addeen Ramadan

Studienarbeit von
Shi Wei und Xinli Yang

Clausthal-Zellerfeld, den 21. Januar 2010

Inhaltsverzeichnis

Abbildungsverzeichnis

Kapitel 1

Zielsetzung

Die Zielstellung der Studienarbeit ist, für die Kommunikation zwischen mehreren mobilen Robotern(Slave) und Computer(Master), ein echtzeitfähiges Datenfunksystem zu entwicken. Ein Master-Controller können theoretisch unbegrenzte Slave(Roboter) gleichzeitig drahtlos steuern. Die Studienarbeit sollte sich auf eine Versuch mit einem Master und zwei Slaves beschränken.

Beschreibung der Aufgabe:

- Die Informationen oder Daten, zwischen FPGA und PC über Schnittstelle RS232 senden oder empfangen, mit C-Sprache zu programmieren.
- Die Kommunikation zwischen FPGA und FPGA über Funkmodul, mit VHDL-Sprache zu programmieren.

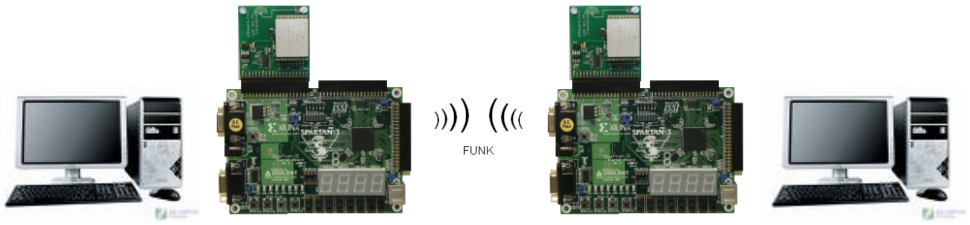

Abbildung 1.1: Prinzip der Verbindung

Kapitel 2

Hardware und Schnittstelle

In dieser Arbeit werden folgende Hardware im Anspruch genommen: die Xilinx Digilent Spartan-3 Starter Board(FPGA), Aurel Transceiver XTR-434, Funkadapter und Serielle Schnittstelle RS232 sind.
FPGA wird hauptsächlich als Antrieb für den XTR-434 Transceiver eingesetzt. Die Datenaustausch erfolgt über die serielle RS232-Schnittstelle. XTR-434-Transceiver-Modul, der sich auf einer separaten Platine befindet, ist für drahtlose Datenkommunikation zuständig.

2.1 Transceiver XTR-434

Miniature Daten Tranceivermodul(n -e) XTR-434 hat eine höchste Übertragungsgeschwindigkeit von 100Kbps und eine Betriebsfrequenz von 433,92MHz.

Abbildung 2.1: XTR-434

2.1.1 Pinbelegung und Blockschaltbild

Aurel XTR-434 Pinbelegung und Blockschaltbild

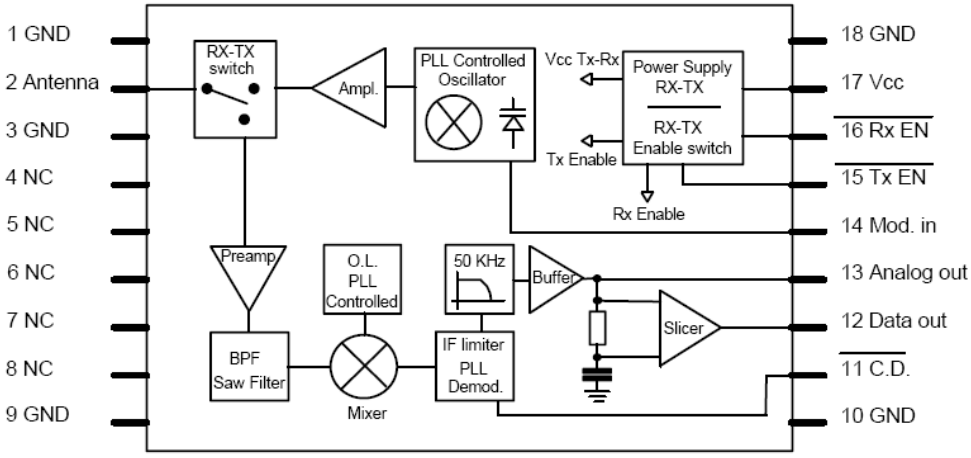

Abbildung 2.2: XTR-434 Block

2.1.2 XTR-434 Pin Verbindungen

Pin 1-3	RF GND	RF-Schaltung GND.
Pin 2	Antenna	50 Ω Impedanz Antennenanschluss.
Pin 9-10-18	GND	Verbindungen zu GND. Intern auch zur Abschirmung des Moduls.
Pin 11	CD	Carrier Detect. Bei einem aktiviertem Empfänger, ein Signal mit niedriger Spannung [Spannungspegel 0V] deutet einen HF-Träger hin. Die Zeile aktiviert mit einem HF-Signal von ca. -96 dBm bezogen auf 2-polig [Antenne]. Hoch Impedanz Ausgang nur für Lasten unter CMOS-Logik.
Pin 12	RXD	Empfänger-Daten ausgegeben. Lastimpedanz erlaubt: über 100KOmega. Kein Kapazität erlaubt.
Pin 13	AF	Gefiltert und gepufferte Ausgabe, die die analogen Ausgang des FM-Detektor ist. Lastimpedanz erlaubt: über 2KOmega und weniger als 100pF.
Pin 14	TXD	Eingang zum Sender; akzeptiert serielle Daten im TTL-Logik [0 5V] mit einer 10KOmega Lastimpedanz.
Pin 15	TX Enable	Aktiv, wenn gering [Spannungspegel 0V], ermöglicht es dem Sender Schaltung, benötigten Strom : 1mA.
Pin 16	RX Enable	Aktiv, wenn gering [Spannungspegel 0V], ermöglicht es dem Empfänger Schaltung, benötigten Strom : 1mA.
Pin 17	Vcc	Anschluss an den positiven Pol der Versorgungsspannung [5 V 10%].

2.1.3 Gerätenutzung

Um die Leistung des Gerätes auszunutzen und die Kondition, die durch die Zertifikate charakterisiert wird, zu erfüllen, muss der Sender wie Folgendes auf einer gedruckten Schaltung angepasst werden:

5V DC Versorgung :

- Der Transceiver muss an einer niedrigen Spannungsquelle angeschlossen. Zusätzlich muss er gegen Kurzschlüsse geschützt werden.

- Maximale Spannung Variation: +/- 0,5V.
- De-kupplung, neben dem Sender, durch ein Minimum 100.000pF Keramik-Kondensator.

Gerätenutzung in Netz-Systeme

Unter Bercksichtigung der minimalen Übertragungsgeschwindigkeit wurde das Gerät so entwickelt, dass es eine möglichst geringe Schaltzeit zwischen Sendung und Empfangen ermöglichen. Der XTR-434 braucht 2 ms nach Rx / Tx zu invertieren. Es wird empfehlen bei der Auswahl des Modells die minimale Inversionszeit zu berücksichtigen, da eine schnelle Inversion ein agierenderes Modulationsverfahren verlängt. Es ist in der Praxis sehr aufwendig zu implementieren.

Referenzkurven

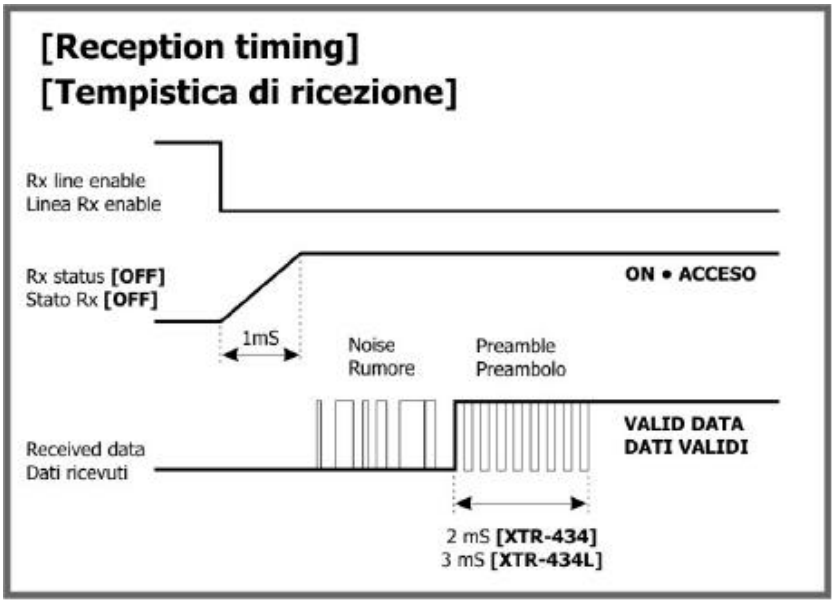

Abbildung 2.3: Referenzkurven

2.1.4 Antennenanschluss

Antennenanschluss:
1. Er kann als einen direkten Verbindungspunkt für die strahlende Peitschenantenne genutzt werden.
2. Er muss eine Verbindung der zentrale Leitung mit 50 Ω Koaxialkabel ertragen. Es ertragen kann die Verbindung der zentralen Draht einer 50 Omega Koaxialkabel. Es muss sicher gestellt werden, dass das Geflech an den Boden in einer nährer PositionS verschweißst wird.

Antenne:
Eine Kupfer Leitung Peitsche-Antenne, die eine Lange von 16,5 mm und einen Durchmesser von etwa 1 mm besitzt, muss an dem HF-Eingang des Transceivers angeschlossen werden.
Der Köerper der Antenne muss möglich gerade gestellt werden. Ein 5 cm minimaler Abstand von anderen Schaltung und Metallteile ist vorgeschrieben. Die Antenne kann sowohl vertikal als auch horizontal eingesetzt werden. Vorausgesetzt ist eine geeignete Grundplatte als Verbindungsstelle zwischen Antenne und Empfängerseingang eingesetzt wird.

Als Alternative zu der AM-Antenne ist es möglich, der Peitsche Modell von Aurel zu benutzen [siehe Daten Schutzdecken Application Notes]. Eine große Abweichung zur der Vorschrift beim Einbau des Peitschen Modells kann die CE-Zertifizierung nicht garantiert werden.

2.2 Datenfunkadapter

Anhand der Schaltung ist es ersichtlich, dass bei der Aktivierung des XTR-434s nur vier Ports benötigen werden. Nämlich Pin12 RXD, Pin14 TXD, Pin15 TX und Pin16 RX. In der folgenden Tabelle sind die Relationen zwischen den Ports des XTR-434 und Spartan-3 Starter Board aufgelistet.
Die Aktivierung der Pin9-10-18 GND und Pin17 Vcc kann durch eine direkt Verbindung mit S3Board A1 Stecker erfolgen.

XTR-434 Name	XTR-434 Pin	S3Board A1 Pin
RXD	12	4
TXD	14	22
TX Enable	15	20
RX Enable	16	21

Abbildung 2.4: Datenfunkadapter

2.3 FPGA Entwicklungsboard

Die Spartan-3 Starter Board bietet eine leistungsstarke Entwicklungsplattform für das Design mit neuen Spartan-3 FPGA von Xilinx. Es verfügt über eine 1000K Gattern Spartan-3, in board I/O Geräte und zwei große Speicherchips. Damit ist er ein perfektes Plattform zum Experimentieren mit allen neuen Design. Das Board enthält auch eine Plattform Flash JTAG-Programmable ROM, somit können Designs als Read-Only Daten in dem ROM gespeichert. Das Spartan-3 Starter Board ist voll kompatibel mit allen Versionen der Xilinx ISE-Tools, einschließlich der kostenlosen Webpack.

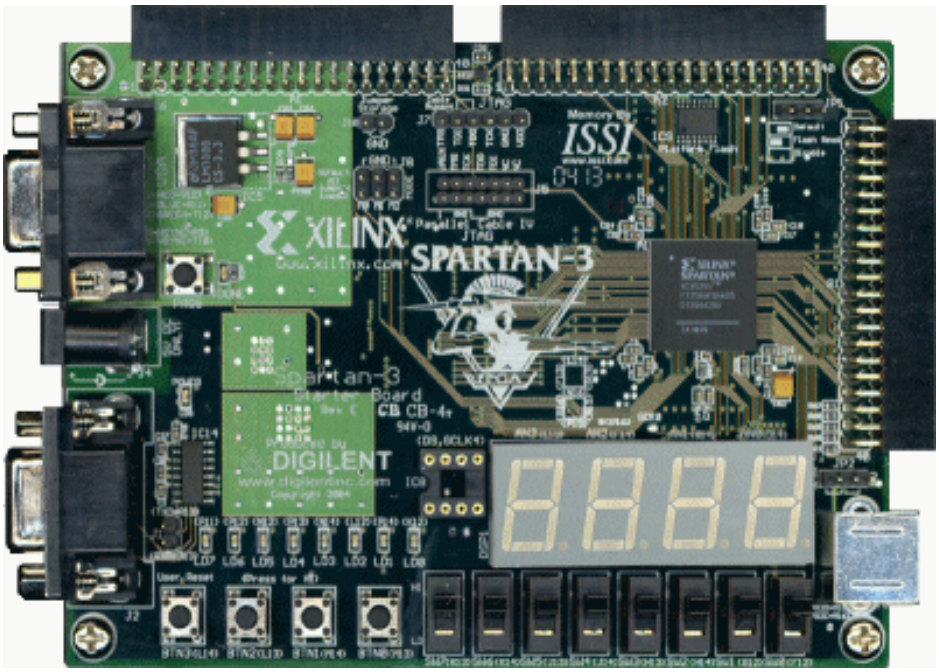

Abbildung 2.5: Spartan-3 Starter Kit Board

2.3.1 Merkmale

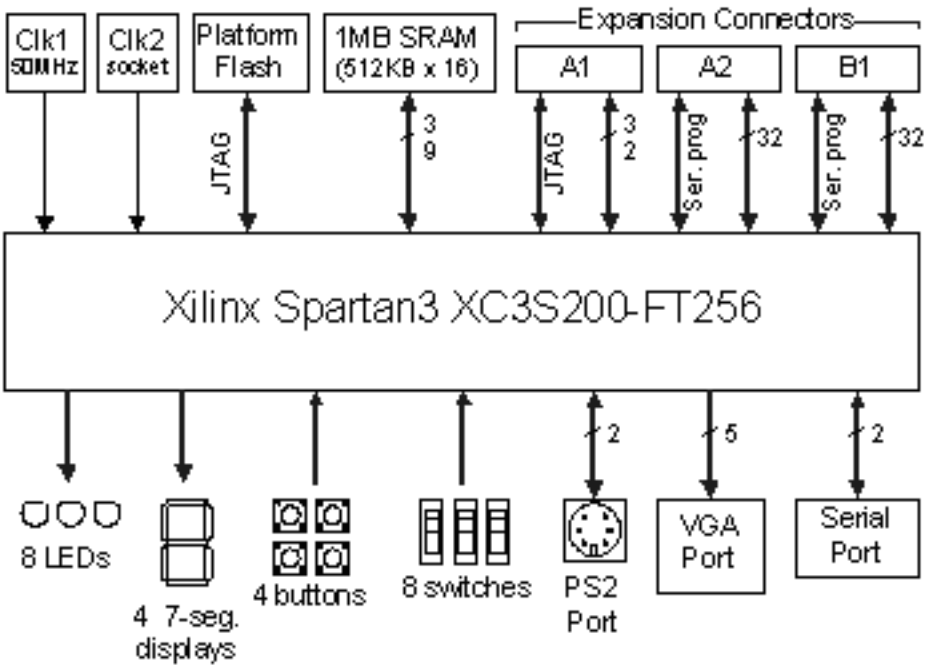

Abbildung 2.6: Spartan-3 Starter Kit Board Block

- Xilinx Spartan-3 FPGA w / zwölf 18-Bit-Multiplikatoren, 216Kbits Block-RAM und bis zu 500MHz internen Taktfrequenzen
- -200 Und -1000 Versionen verfügbar
- On-board 2Mbit Plattform Flash (XCF02S)
- 8 Schiebeschalter, 4 Tasten, 9 LEDs, und 4-stellige Sieben-Segment-Anzeige
- Serielle Schnittstelle, VGA-Anschluss und PS / 2 Maus / Tastatur-Anschluss
- Drei 40-Pin-Anschlüsse Expansion
- Drei Hochstrom-Spannungsregler (3,3 V, 2,5 V und 1,2 V)

- Arbeitet mit JTAG3 Digilent's, JTAG USB und JTAG USB Full Speed-Kabel, sowie P4 und MultiPro Kabel von Xilinx
- 1 MByte On-Board-10ns SRAM (256Kb x 32)

2.3.2 Expansion Connectors

Das Spartan-3 Starter Kit Bord hat drei 40-Pin-Expansion-Anschlüsse A1, A2 und B1. Die A1- und A2-Anschlüsse, befinden sich wie in Abbildung am oberen Rand des Boardes[1]. Connector A1 ist in der oberen linken und A2 ist in der oberen rechten Ecke. Die B1-Anschluss ist am rechten Rand des Boardes.

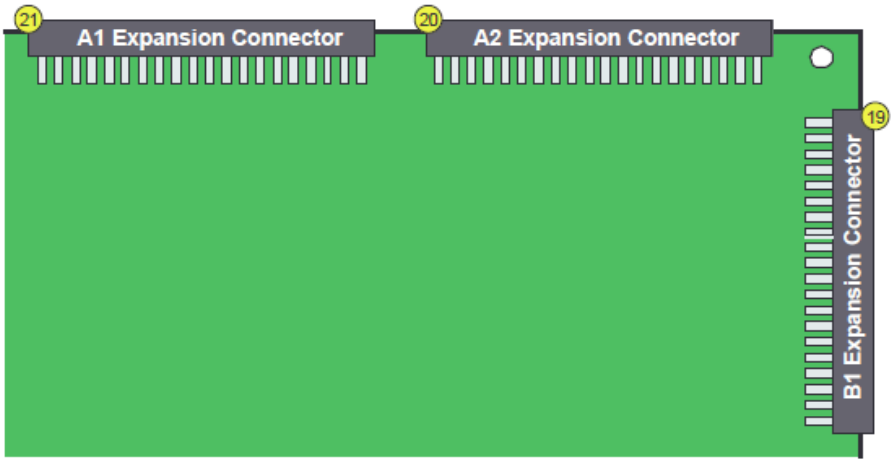

Abbildung 2.7: Spartan-3 Starter Kit Board Expansion Connectors

Es ist möglich durch jeden Port dem FPGA Spartan-3 Starter Kit Board zu programmieren. Zum Beispiel bietet Port A1 zusätzliche Logik um das FPGA und Plattform Flash JTAG-Chain zu steuern. Ebenso bieten Anschlüsse A2 und B1 Verbindungen für Master oder Slave Serial-Modus-Konfiguration. Schließlich bietet auch B1-Port Master oder Slave Parallel Konfigurationsmodus.

[1]siehe Abbildung 2.7

Jede 40-Pin-Header-Erweiterung, wie in Abbildung 2.8 gezeigt wird, verwendet einen 0,1-Zoll (100 mil) DIP Abstand. Pin 1 auf jedem Anschluss ist immer GND. Pin 2 auf allen Schlüsse ist als eien 5V DC-Ausgang festgelegt. Pin 3 dient als den Ausgang von 3,3 V DC Regulator.

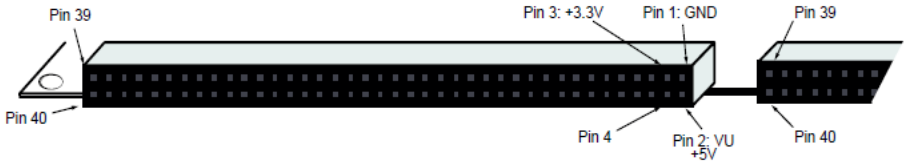

Abbildung 2.8: Spartan-3 Starter Kit Board 40-Pin Expansion Connector

A1 Steckers Pin

Namen	**FPGA Pin**	**Connector**		**FPGA Pin**	**Namen**
GND		1	2		VU(+5V)
Vcco(+3.3V)	Vcco	3	4	N8	ADR0
DB0	N7	5	6	L5	ARD1
DB1	T8	7	8	N3	ARD2
DB2	R6	9	10	M4	ARD3
DB3	T5	11	12	M3	ARD4
DB4	R5	13	14	L4	ARD5
DB5	C2	15	16	G3	WE
DB6	C1	17	18	K4	OE
DB7	B1	19	20	P9	CSA
LSBCLK	M7	21	22	M10	MA1-DB0
MA1-DB1	F3	23	24	G4	MA1-DB2
MA1-DB3	E3	25	26	F4	MA1-DB4
MA1-DB5	G5	27	28	E4	MA1-DB6
MA1-DB7	H4	29	30	H3	MA1-ASTB
MA1-DSTB	J3	31	32	J4	MA1-WRITE
MA1-WAIT	K5	33	34	K3	MA1-RESET
MA1-INT	L3	35	36	JTAG Isolation	JTAG Isolation
TMS	C13	37	38	C14	TCK
TDO-ROM	JTAG TDO	39	40	Header J7,pin3	TDO-A

2.3.3 ucf-Datei

In VHDL können nicht alle Entwurfsziele geeignet beschrieben werden. Es gibt deshalb zu jedem Projekt in unserem Entwurfssystem eine sog. ucf-Zusatzdatei, in der vor allem die Zuordnung der Schaltungsanschlüsse zu Schaltreisanschlüssen in der Form

NET 'Anschlussname' LOC = 'Pinnummer';

festgelegt wird.

```
NET "clk" LOC = "T9";

# --- RS 232 ---
NET "sio_RxD" LOC = "T13";
NET "sio_TXD" LOC = "R13";

# --- Pin 35 am B1 ---

NET "Funk_RxD" LOC = "N8";
NET "Funk_TxD" LOC = "M10";
NET "recvEn" LOC = "P9";
NET "sendEn" LOC = "M7";
```

Abbildung 2.9: Ein Beispiel fuer ucf-Datei

2.4 RS 232

Der Begriff EIA-232, ursprünglich RS-232, bezeichnet einen Standard für eine serielle Schnittstelle, die in dem frühen 1960 von einem US-amerikanischen Standardisierungskomitee (heute EIA C Electronic Industries Alliance) eingeführt werden.

EIA-232 definiert die Verbindung zwischen dem Terminal (data terminal equipment (DTE), Datenendeinrichtung) und dem Modem (Data Communication Equipment (DCE), Datenübertragungseinrichtung), was Timing, Spannungspegel, Protokoll und Stecker betrifft. Allgemein sind die Parameter unter Serielle Datenübertragung erläutert.
Mainframes und Text-Terminals sind unter Zuhilfenahme von Modems durch Punkt-zu-Punkt-Verbindungen über die Telefonleitung ähnlich wie Fernschreiber zusammengeschlossen worden. Die übertragung der Daten bei beiden Systemen erfolgte sequenziell. Durch den ursprünglichen Verwendungszweck bedingt weist die Schnittstelle einige Asymmetrien bei der Definition der Steuer-Leitungen auf, die bei den später üblich gewordenen Anwendungen in völlig anderen Bereichen zu Verschaltungsproblemen führen können.

Abbildung 2.10: RS 232

2.4.1 Verkabelung und Stecker

Um zwei Geräte über die serielle Schnittstelle zu verbinden, müssen die 'hörenden' mit den 'sprechenden' Leitungen verbunden werden. Bei Terminals bzw. Rechnern (DTE C data terminal equipment) sind 'sprechende' Leitungen TxD, RTS und DTR, 'hörende' Leitungen sind RxD, CTS, DSR, DCD und RI. Bei Modems (DCE C data circuit-terminating equipment) ist

es genau umgekehrt; es gibt die vom Terminal 'gesprochenen' Signale an die Gegenseite weiter und muss daher auf diese 'hören', andersherum werden die von der Gegenseite 'gehörten' Signale zum Terminal 'weitergesagt'.

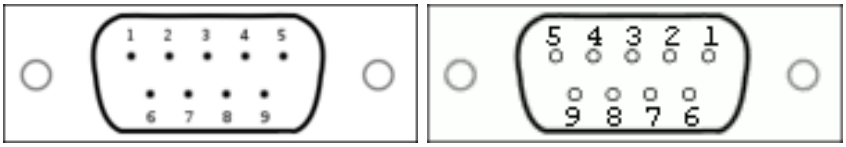

Abbildung 2.11: Zwei RS 232 Stecker

2.4.2 RS232 Pin Verbindungen

Name	**Pin Nr.**	**Beschreibung**	**In/Out**
TxD	3	Leitung für ausgehende (von DTE gesendete) Daten (negative Logik).	Out
RxD	2	Leitung für eingehende (von DTE zu empfangende) Daten (negative Logik).	In
RTS	7	'Sendeanforderung'; Ein High-Pegel an diesem Ausgang signalisiert, dass DTE Daten senden möchte.	Out
CTS	8	'Sendeerlaubnis'; Ein High-Pegel an diesem Eingang ist ein Signal der Gegenstelle, dass sie Daten von DTE entgegennehmen kann.	In
DSR	6	Ein High-Pegel an diesem Eingang ist ein Signal der Gegenstelle, dass sie im Prinzip einsatzbereit ist.	In
GND	5	Signalmasse. Die Signalspannungen werden gegen diese Leitung gemessen.	-
DCD	1	Mit einem High-Pegel an diesem Eingang signalisiert die Gegenstelle, dass sie einlaufende Daten auf der Leitung erkennt (dem Namen nach ist das die Modulationsträger-Erkennung) und an DTE weitergeben möchte.	In
DTR	4	Mit einem High-Pegel an diesem Ausgang signalisiert DTE seine Betriebsbereitschaft an die Gegenstelle. Damit kann die Gegenstelle, z. B. ein Modem, aktiviert oder auch zurückgesetzt werden. üblicherweise antwortet die Gegenstelle mit einem High-Pegel auf DSR.	Out
RI	9	Ein High-Pegel an diesem Eingang signalisiert dem DTE-Gerät, dass ein Anruf ankommt, d.h. dass jemand eine Datenverbindung aufzubauen wünscht, ('ring' ist engl. für 'klingeln'; besonders bei Modems).	In

Kapitel 3

Übertragungsprotokoll

TCP/IP-Referenzmodell

Zur Gliederung der Kommunikationsaufgaben werden in Netzwerken funktionale Ebenen, so genannte Schichten (layer), unterschieden. Für die Internetprotokollfamilie ist dabei das TCP/IP-Referenzmodell massgebend. Es beschreibt den Aufbau und das Zusammenwirken der Netzwerkprotokolle aus der Internet-Protokoll-Familie und gliedert sie in vier aufeinander aufbauende Schichten. TCP/IP steht für Transmission Control Protocol/Internet Protocol.
Das TCP/IP-Referenzmodell ist auf die Internet-Protokolle zugeschnitten, die den Datenaustausch über die Grenzen lokaler Netzwerke hinaus ermöglichen ('Internetworking'). Es wird weder der Zugriff auf ein übertragungsmedium noch die Daten übertragungstechnik definiert. Vielmehr sind die Internet-Protokolle dafür zuständig, Datenpakete über mehrere Punkt-zu-Punkt-Verbindungen (Hops) weiterzuvermitteln und auf dieser Basis Verbindungen zwischen Netzwerkteilnehmern über mehrere Hops herzustellen.
Um Probleme der Netzwerkkommunikation im Allgemeinen zu betrachten, greift man stattdessen auf das ISO/OSI-Referenzmodell zurück. Es ist jedoch zu beachten, dass sich die Benennung der einzelnen Schichten in den Modellen unterscheidet.

TCP/IP Referenzmodell Schicht:

1. Netzzugangsschicht
2. Internetschicht

3. Transportschicht

4. Anwendungsschicht

Die in der Arbeit entwickelte Übertragungsprotokoll basiert auf dem TCP/IP Referenzmodell. Abbildung 3.1 zeigt die Übertragungsprotokoll im Detail. Die Daten werden in der Anwendungsschicht mit dem Empfänger-IP-Header und die 'LEN'(die Länge der Daten) zusammengebaut. In dem Transportschicht wird das Datenpaket aus der Anwendungsschicht weiter Integriert. In der Internetschicht wird die Reihenfolge des Senders anhand seines Headers bestimmt. Die Netzzugangsschicht befindet sich auf der untersten Ebene. Dieses Schicht stellt mechanische, elektrische und weitere funktionale Hilfsmittel zur Verfügung, um physikalische Verbindungen zu ermöglichen.

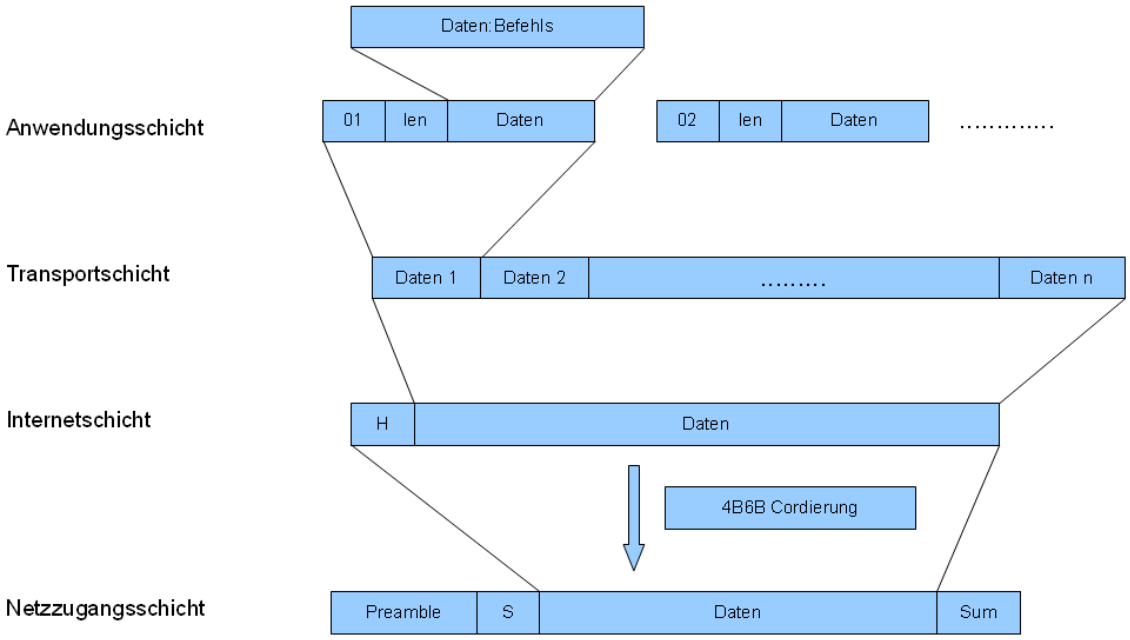

Abbildung 3.1: Datenfunk Protokoll

3.1 Netzzugangsschicht / Link Layer

Die Netzzugangsschicht (engl.: Link Layer) ist im TCP/IP-Referenzmodell spezifiziert, enthält jedoch keine Protokolle der TCP/IP-Familie. Sie ist vielmehr als Platzhalter für verschiedene Techniken zur Datenübertragung von Punkt zu Punkt zu verstehen.

3.1.1 Physical

Der Aufbau der physikalischen Schicht ist seitens des verwendeten Funkmoduls XTR-434 fest vorgegeben.

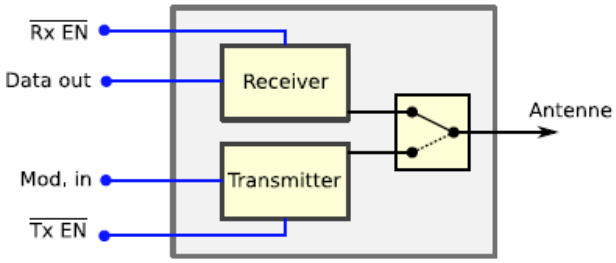

Abbildung 3.2: Prinzipieller Aufbau des Funkmoduls

- Siehe Abbildung, das Transceiver Modul wird über 4 Pin gesteuert:

  - 'Rx EN' : Receive Enable
  - 'Tx EN' : Transmit Enable
  - 'Mod.in' : Sender-Daten
  - 'Data out' : Empfänger-Daten

- Sender und Empfänger des Transceiver Modul können nur wechselweise aktiviert werden.

- Puls-Amplituden-Zeit : den Eigenschaften der Schaltungen bestimmen die Länge der Zeit zwischen je zwei aufeinander folgenden übergängen auf der Ebene der seriellen Signal. Für den ordnungsgemäßen Betrieb

des Transceiver Modul, Suchzeit muss zwischen 10 μS und 200 μS enthalten sein.

- Einschwingszeit von der Data Slicer erfordert, dass für 1 mS bevor die Daten, eine Preamble, die von einem quadratischen Welle zusammensetzt, übertragen, um zu prüfen, zuverlässige Daten, die sich aus der RXD-Leitung.

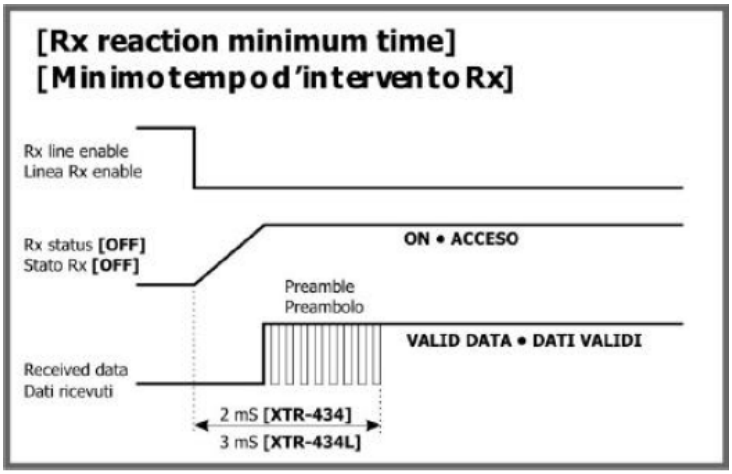

Abbildung 3.3: Rx Reaktion minimum Time

- Das Sendesignal sollte idealerweise ein Tastverhältnis von 50:50 aufweisen.

3.1.2 Data Link

Die Datenübertragung auf der Sicherungsschicht erfolgt nach dem NRZ-Prinzip ('non-return-to-zero'), d.h. der Bitwert wird durch den elektrischen Pegel kodiert. Während der Symbol-/Bitzeit ('1 Takt') ist der Pegel dabei konstant. Ein Pegelwechsel ist immer nach Ende eines Taktes möglich. Die Länge des Taktes beträgt 13 us, sodass sich eine Brutto-Übertragungsrate (ohne Berücksichtigung des Leitungscodes) von 76,9 kBit ergibt.

Abbildung 3.4 zeigt den Aufbau des Rahmens. Er besteht aus 4 Teilen:

- 'Preamble': Rechtecksignal mit einer Dauer von ca. 2ms. Nach jedem Takt erfolgt ein Pegelwechsel. Die Preambel ist erforderlich, damit der

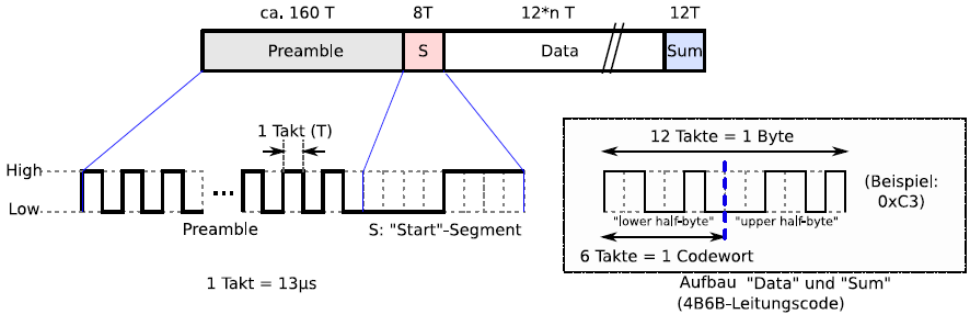

Abbildung 3.4: Rahmenformat

Empfänger sich auf das Signal einregeln kann (Verstärkungregelung, Entscheiderschwelle für Datenausgang, etc.).

- 'Start (S)': Sequenz bestehend aus vier Takten Low-Pegel und darauf folgend vier Takten High-Pegel. Sie signalisiert dem Empfänger, dass nun die eigentlichen Daten gesendet werden. Da die Länge der Preambel auf Empfängerseite schwanken kann (die Zeit des Übergangs vom Rauschen (zufällige Pegelwechsel im 'Data out'-Signal) bis zur korrekten Wiedergabe des Rechtecksignals hängt von verschiedenen Parametern ab wie Entfernung des Senders, Fertigungstoleranzen des Moduls, Störungen, etc.), ist die Start-Sequenz zur Synchronisation des Empfängers zwingend erforderlich. Dazu kann der Empfänger z.B. auf die Folge von vier Low-Pegeln warten und dann mit der steigenden Flanke die Abtastung beginnen (wobei die ersten vier Werte verworfen werden).

Die folgenden 3 Teile des Rahmens enthalten Daten (also eine Menge von Bytes) und werden mit einem '4B6B' Leitungscode übertragen, d.h. eine Bitsequenz bestehend aus 4 Bits wird eindeutig auf ein 6 Bit Codewort abgebildet, wobei jedes dieser Codeswörter genau drei Nullen und drei Einsen aufweist (Tastverhältnis 50:50). Die Abbildung ist dargestellt, wobei die vorne stehende Ziffer zeitlich immer zuerst übertragen wird. Nebenbei sei vermerkt, dass durch die dargestellte Auswahl der Codeworte aus der Menge aller möglichen Codeworte garantiert ist, dass maximal 4 Nullen bzw. 4 Einsen aufeinander folgen (die Codeworte '111000' und '000111' sind ungültig). Der Empfänger muss also (theoretisch) lediglich über 4 Takte tolerant ge-

genüber dem maximal möglichen Geschwindigkeitsunterschied von Sender- und Empf'angeruhr sein (in dieser Zeit ist wegen NRZ keine Taktrückgewinnung möglich).

Bitsequenz	**Codewort**	**Bitsequenz**	**Codewort**
0000(0x0)	110100	1000(0x8)	110001
0001(0x1)	101100	1001(0x9)	101001
0010(0x2)	011100	1010(0xA)	011001
0011(0x3)	110010	1011(0xB)	100101
0100(0x4)	011010	1100(0xC)	001101
0101(0x5)	100110	1101(0xD)	100011
0110(0x6)	010110	1110(0xE)	010011
0111(0x7)	001110	1111(0xF)	001011

Tabelle 3.1: Codetabelle (vorderstes Bit wird zuerst gesendet)

Demzufolge wird jedes Byte durch zwei Codewörter mit insgesamt 12 Bits (also 12 Takte) dargestellt. Bei der Übertragung wird das niederwertige Halbbyte zuerst gesendet.

- 'Data' (n Bytes, 12*n Takte): Hier werden die Nutzdaten übertragen.
- 'Sum' (1 Byte, 12 Takte): Prüfsumme über das 'Data'-Segment. Sie wird durch Modulo-Addition(XOR) aller Datenbytes berechnet.
  z.B. :
  Sum = DataPaket[0] XOR DataPaket[1],
  Sum = Sum XOR DataPaket[2],
  .....
  Sum = Sum XOR DataPaket[7].

3.2 Internetschicht / Internet Layer

Die Internetschicht (engl.: Internet Layer) ist für die Weiterleitung von Paketen und die Wegewahl (Routing) zuständig. Auf diesem Schicht und den darunterliegenden Schichten werden direkte Verbindungen betrachtet. Die Aufgabe dieses Schichtens ist es, zu einem empfangenen Paket das nächste Zwischenziel zu ermitteln und das Paket dorthin weiterzuleiten.

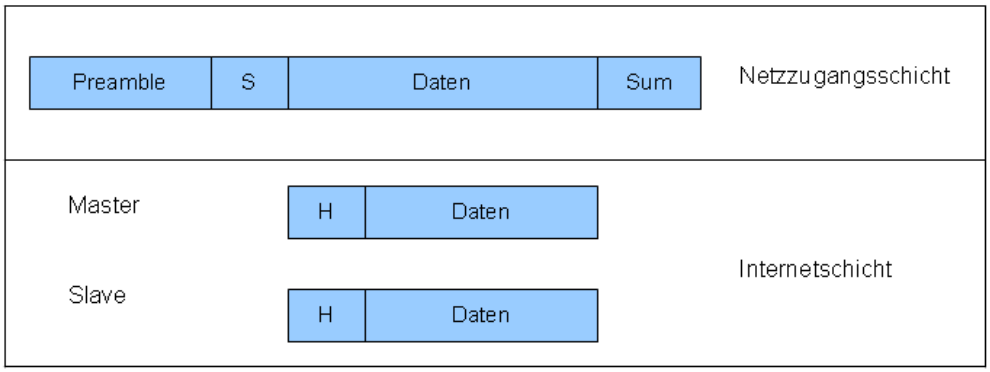

Abbildung 3.5: Internetschicht

In der vorliegenden Aufgabe, wird die Datenübertragung zwischen einem Master und mehreren Slaves simuliert.
An dieser Stelle wird der 'Broadcast'-Modus für die Propagation des Paketes eingesetzt. 'Broadcast'-Modus ist in der Lage, Daten gleichzeitig an mehreren Empfänger zu senden. In dem Broadcast-Prozess darf nur ein Sender gleichzeitig existieren. [1]

Auf diesem Schicht wird zuerst ein ID in der Form von 'Header-Data' vor das Datenpaket hinzugeügt. Danach wird das Datenpaket mit Header an allen Empfänger geschickt. Derjeniger Empfänger, der das gleiche ID besitzt, wird nach dem Empfang des Paketes als neuen Sender initialisiert. Der originäre Sender wird zu einem Empfänger umgewandelt.

[1] Siehe Abbildung 3.6

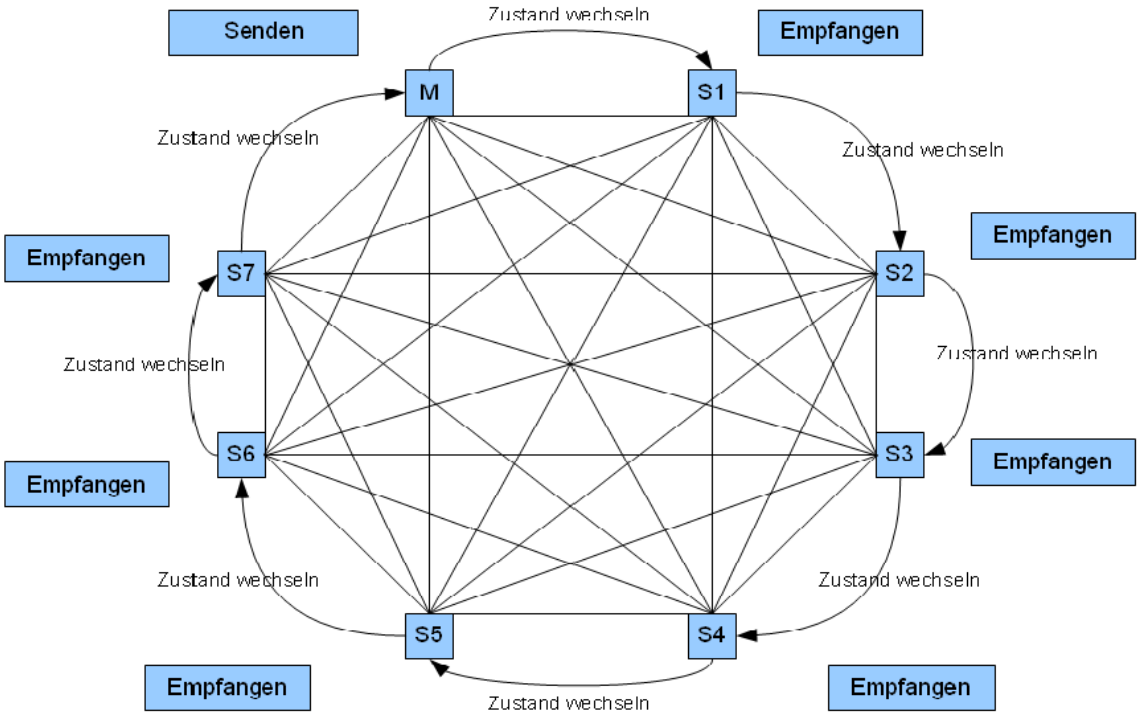

Abbildung 3.6: Master-Slave-Topology

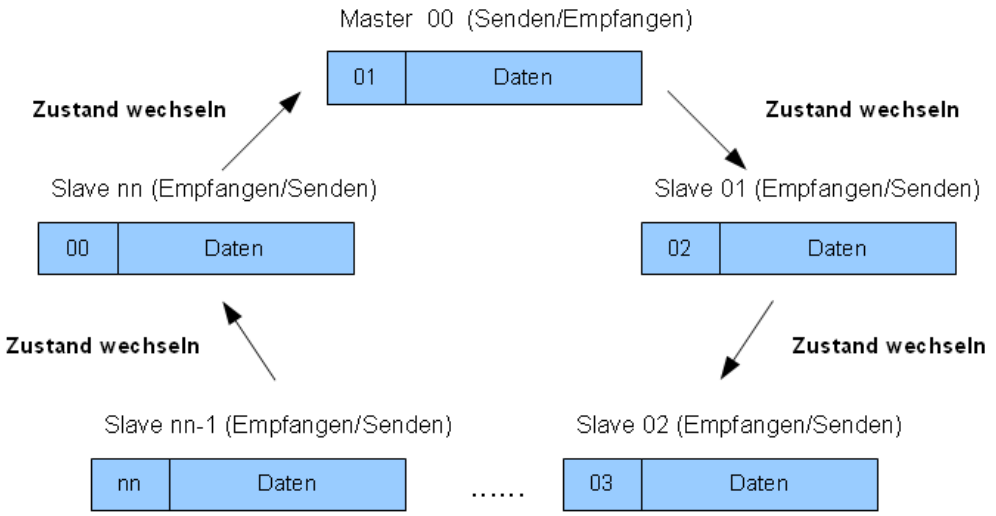

Abbildung 3.7: Ein Beispiel des Protokolls

Das Prinzip wird mit folgenden Beispiel[2] erläutert: Das Datenpaket mit 'Header' 01 wird an allen Empf"angern geschickt. Nur der Empf"anger, der ID 01 hat, wird nach dem Empfang des Paketes als einen Sender initialisiert. Nach der Sendung des Datenpaketes wird der Sender zu einem Empfänger umgewandelt. Der neue Sender mit ID 01 packt ein neues ID vor das Paket und schickt das Datenpaket wiederum an allen Empfänger. Auf dieser Weise wird das Paket durch das ganze Netz propagiert bis es dem gewünschten Empfänger erreicht.

[2]Siehe Abbildung 3.7

3.3 Transportschicht / Transport Layer

Die Transportschicht (engl.: Transport Layer) stellt eine Ende-zu-Ende-Verbindung her. Diese Schicht und alle darüber liegenden werden in C Programm implementiert. Die Schnittstelle besteht dabei aus einem Serielle Schnittstelle für die empfangenen und zu sendenden Daten.

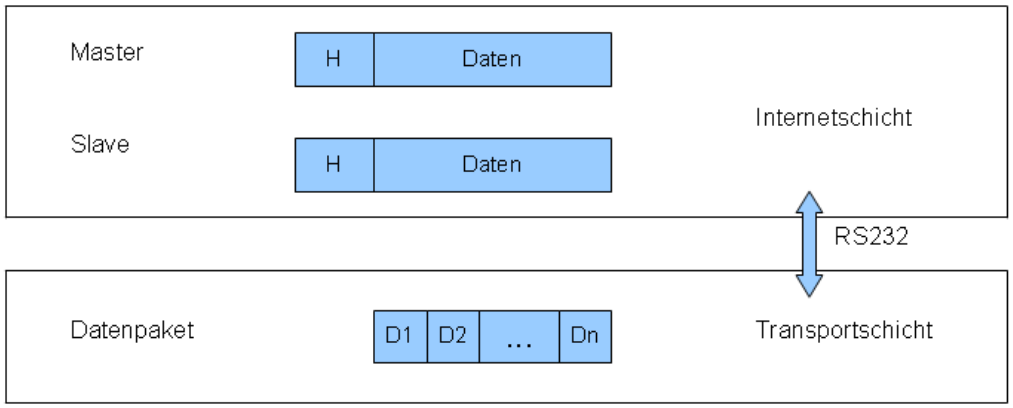

Abbildung 3.8: Transportschicht

Spartan 3 Starter Kit Board hat einen Anschluss für eine solche serielle Schnittstelle, über die die Baugruppe mit einem PC kommunizieren kann. Von dem 9-poligen SUBD-Stecker einer seriellen Schnittstelle werden nur 2 Leitungen genutzt:

- Steckerkontakt 2 : RxD, Eingang Empfänger
- Steckerkontakt 3 : TxD, Ausgang Sender

Eine Null wird als negative und eine Eins als positive Spannung zwischen Signalleitung übertragen. Die Umwandlung der üblichen Logikpegal (große Spannung, kleine Spannung) in (positive Spannung, negative Spannung) erfolgt in einem speziellen Schaltkreis, in unserem Versuchsaufbau. In der ucf-Datei heißt der Empfängereingang sio_RxD und der Senderausgang sio_TxD.

Die Übertragung über RS232 erfolgt byteweise. Während einer Übertragungspause ist das Sendesignal TxD='1'. Jedes Datenpaket beginnt mit

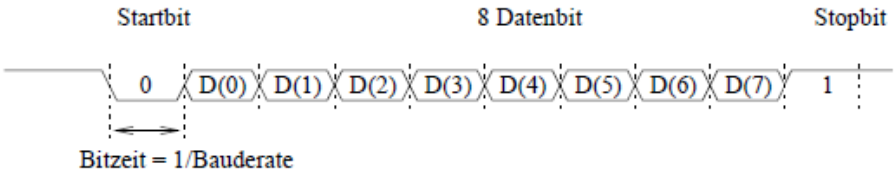

Abbildung 3.9: RS232 Datenpaket

einem Startbit=’0’ gefolgt von den 8 Datenbits, und einem Stopbit=’1’. Danach kann sich eine Pause oder das Startbit der nächsten Übertragung anschließen. Wenn mehre Bytes Daten über RS232 senden möchte, Nach einem Startbit=’0’, gefolgt von den 16, 24, oder 32, also n*8 bits Daten, dann einem Stopbit=’1’.

Das in dem Array gespeicherte ASCII Zeichen wurde vor der Sendung ins 8 bit Binärdaten konvertiert. Anschließende wird die Binärdaten über RS232 Anschluss auf das Board übertragen.[3]

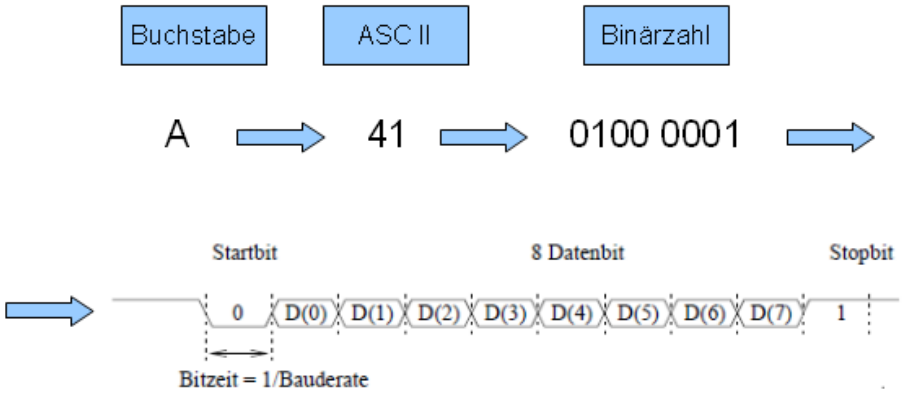

Abbildung 3.10: Konvertierung von Buchstaben zur Bits

[3]Siehe Abbildung 3.10

3.4 Anwendungsschicht / Application Layer

Die Anwendungsschicht (engl.: Application Layer) umfasst alle Protokolle, die mit Anwendungsprogrammen zusammenarbeiten und die Infrastruktur der Netzwerk für den Austausch anwendungsspezifischer Daten nutzen. Diese Schicht und alle darüber liegenden werden in C Programm implementiert.

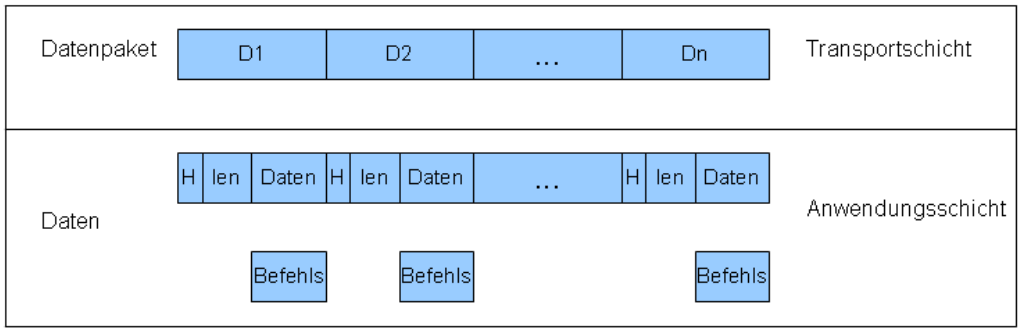

Abbildung 3.11: Anwendungsschicht

- Der Befehls werden aus Textdatei gelesen.
- Sie werden in Array von String gespeichert.
- Header und Len werden vor der Daten hinzufügt.
- Alle String kombiniert in einem Datenpaket.

Kapitel 4

Programm Flussdiagramm

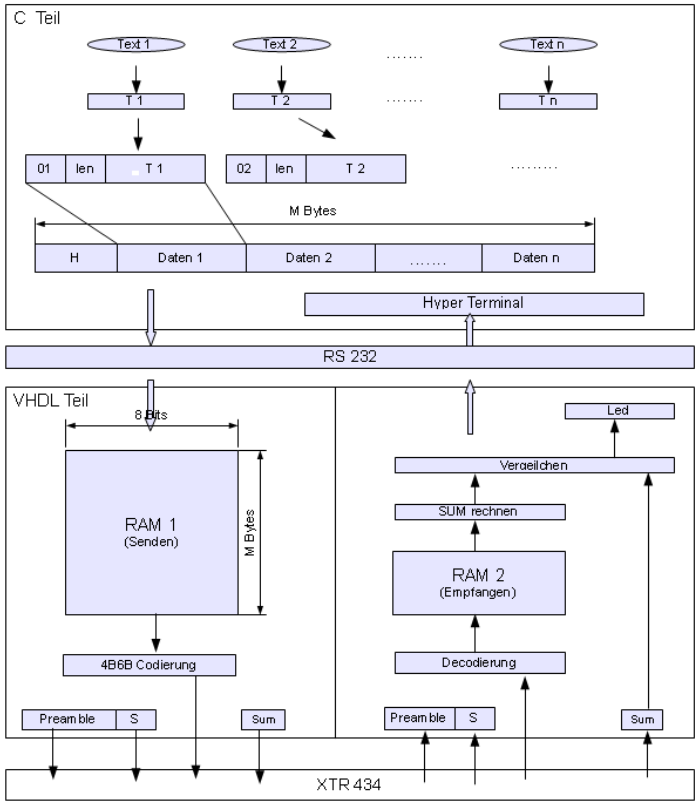

Abbildung 4.1: Diagramm

4.1 VHDL Programm Flussdiagramm

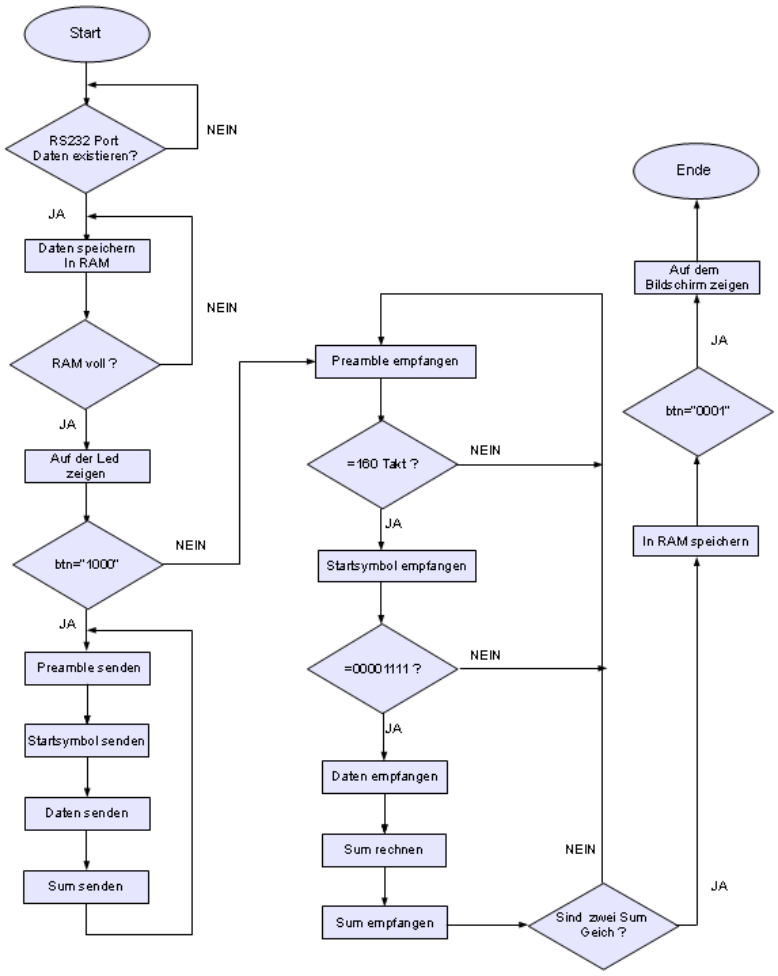

Abbildung 4.2: VHDL Flussdiagramm

4.2 C Programm Flussdiagramm

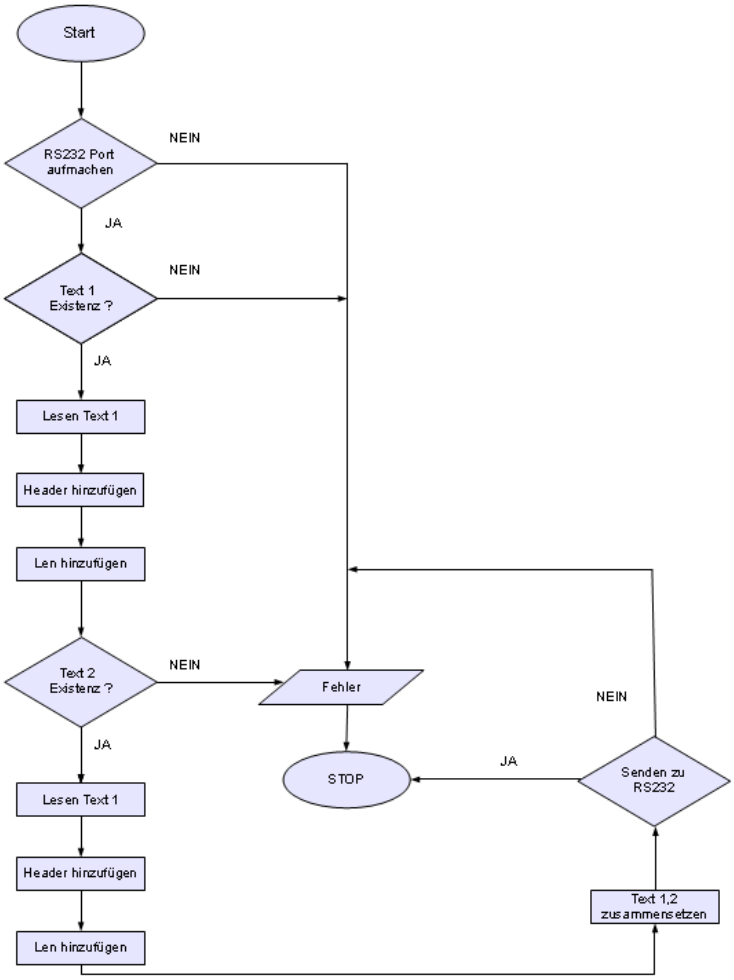

Abbildung 4.3: C Flussdiagramm

Literatur

- Projekt: DatenFunk Dokumentationen, Funkprotokoll, http://techwww.in.tu-clausthal.de/site/Projekte/Datenfunk/.
- Script zur Vorlesung Rechnertechnologie, G.Kemnitz, 3.Februar 2005.
- VHDL Praktikum Serielle Schnittstelle (RS232), G.Kemnitz, 19.Juli.2007, http://techwww.in.tu-clausthal.de/site/Lehre/Praktikum_VHDL/Aufgaben/VHDL-RS232.pdf.
- Spartan-3 Starter Kit Board User Guide, UG130(v1.0), April 26, 2004, XLINX. http://www.xilinx.com.
- XTR-434xxx Instruction Manual, http://www.aurel.it.
- Data Transceiving by XTR-434 transceiver, http://www.aurel.it.
- EIA-232, RS232, http://de.wikipedia.org/wiki/EIA-232.
- American Standard Code for Information Interchange(ASCII), http://de.wikipedia.org/wiki/ASCII.