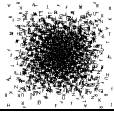


Projekt

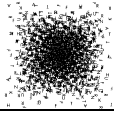
Profibusmonitor

Vorgelegt von: Martin Harndt
Betreuer: Prof. Dr. Volker Pfeiffer
Frankfurt, den 22.08.2013



Inhalt

1	Einleitung	1-1
2	Aufgabenstellung	2-1
3	Profibus	3-1
3.1	Normen	3-1
3.2	Varianten	3-2
3.3	Einordnung in das ISO/OSI-Referenzmodell	3-3
3.4	Topologie	3-3
3.5	Datenaustausch durch Telegramme	3-4
3.5.1	SD1: Telegramm ohne Daten	3-5
3.5.2	SD2: Telegramm mit Daten variabler Länge	3-5
3.5.3	SD3: Telegramm mit Daten fester Länge	3-5
3.5.4	SD4: Token-Telegramm	3-6
3.5.5	SC: Kurzquittung	3-6
3.6	Dienste	3-6
3.7	Adressierung	3-7
3.8	Zeichenkodierung	3-7
3.9	Signalübertragung mittels RS-485	3-7
3.10	Kabel und Stecker	3-9
4	Profibusmonitor	4-1
4.1	Stand der Technik	4-1
4.2	Warum Entwicklung eines eigenen Profibusmonitors?	4-2
4.3	Konzeption eines eigenen Profibusmonitors	4-2
4.4	Vorgegebene Hardware (Spartan 3 Starter Kit)	4-3
5	Realisierte Lösung	5-1
5.1	Anzeige	5-1
5.2	Telegrammaufzeichnung	5-2
5.3	Schnittstelle Profibus	5-3
5.4	Verarbeitungseinheit / Ablaufsteuerung	5-4
5.4.1	Modul InAB_INPUT	5-6
5.4.2	Modul BIT_REGISTER	5-13
5.4.3	Modul TELEGRAM_CHECK	5-15
5.5	Schnittstelle Anzeige	5-23
5.5.1	Modul RS232_TX	5-24
6	Testaufbau	6-1
6.1	Schnittstelle Profibus Aufbau	6-3
6.2	Stromversorgung	6-6
7	Zusammenfassung und Ausblick	7-1
8	Anhang	8-2
8.1	Variablendefinition Steuersignale Steuerung	8-2
8.2	Variablendefinition Steuerzeichen Verarbeitungseinheit	8-2
8.3	Variablendefinition der Ausgänge Modul InAB_INPUT	8-2
8.4	Variablendefinition Ausgänge Modul BIT_REGISTER	8-2
8.5	Variablendefinition Ausgänge Modul RS232_TX	8-3
8.6	Umsetzung der Module in VHDL	8-3
8.7	Verwendetes Automatenmodell	8-3
8.8	Belegung der Ausgangsvariablen	8-4
8.9	Literatur / Web-Seiten	8-4



8.10	Verzeichnis Bilder.....	8-6
8.11	Verzeichnis Tabellen.....	8-7
8.12	Quelle Bilder	8-7
8.13	Verzeichnis Abkürzungen.....	8-7



1 Einleitung

Diese Dokumentation enthält das Konzept und die Realisierung eines Profibusmonitors.

Ein Profibusmonitor kann man sich so ähnlich vorstellen wie die Software /WIRESHARK/ nur in einer Hardwareausführung. Die Daten und Signale des Profibus sollen mitgehört und analysiert werden und anschließend auf einer Anzeige zur Verfügung stehen.

Mit den Daten und Signalen des Profibus kann eine Signalanalyse, Telegrammanalyse, ein Netzwerkmanagement und eine Überwachung des Profibus durchgeführt werden. Als Anzeige für die Daten und Signale könnte vieles dienen, von einer 7-Segment Anzeige bis hin zu einem Rechner oder ein Smartphone, welches die Darstellung der Daten via App übernimmt.

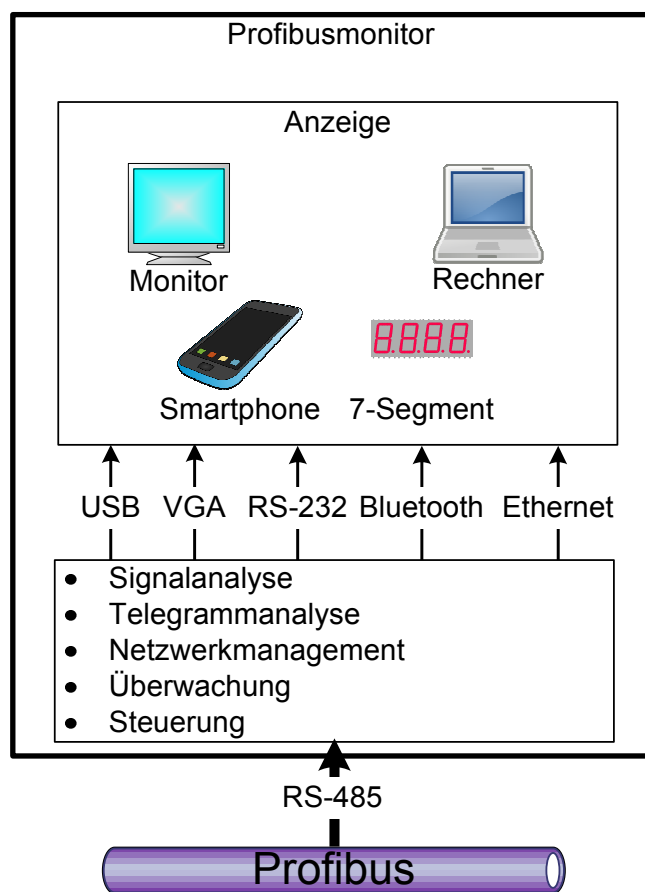
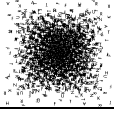


Abbildung 1-1: Konzept Profibusmonitor
/LAPTOP/ /PHONE/ /7SEG/



2 Aufgabenstellung

Ziel dieses Projektes ist die Realisierung eines Profibusmonitors welches den auf einem Profibusnetzwerk ausgetauschten Datenverkehr sichtbar machen kann.

Der Profibusmonitor soll folgende Eigenschaften aufweisen:

- Anschluss an ein Profibusnetzwerk
- Mithören der Signale des Profibusnetzwerks
- Erkennen von Telegramm
- Prüfung auf fehlerhafte Telegramme
- Anzeige der Telegramme
- Steuerung des Ablaufs

Der Profibusmonitor soll mit Hilfe des Spartan 3 Starter Kit und dessen Programmierbaren Chip (FPGA) in VHDL realisiert werden.

Alle im Labor vorhanden Materialien und Geräte können zur Realisierung des Profibusmonitors benutzt werden.



3 Profibus

PROFIBUS (**Process Field Bus**) ist ein Standard für die Feldbus-Kommunikation in der Automatisierungstechnik. /WIKIP_ROFI/

3.1 Normen

Feldbusse, wie der Profibus, werden in der Norm IEC 61158 standardisiert. In der Norm IEC 61784 werden die Communication Profile Families (CPF) der Feldbusse definiert.

Die Norm IEC 61158 unterteilt sich in mehrere Dokumente. Jeder Felsbustyp hat einen eigenen Teil der Norm. In der Norm wird der Profibus als „Type 3“ bezeichnet. Für den Profibus relevant sind daher nur die Teile mit der „Type 3“-Bezeichnung.

Die Norm gliedert sich grob nach den Schichten des OSI-Modells. Wobei hier eine spezielle Variante, das 3-Schichten Feldbusmodell, zum Einsatz kommt.

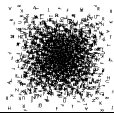
Bezeichnung	Inhalt	Erforderlicher Teil für Profibus
IEC 61158-1	Aufbau und Inhalt der Norm IEC 61158	/IEC 61158-1/
IEC 61158-2	Spezifikation der Bitübertragungsschicht (Physical Layer)	/IEC 61158-2/
IEC 61158-3	Dienstfestlegungen der Sicherungsschicht (Data Link Layer)	/IEC 61158-3-3/
IEC 61158-4	Protokoll der Sicherungsschicht (Data Link Layer)	/IEC 61158-4-3/
IEC 61158-5	Dienstfestlegungen der Anwendungsschicht (Application Layer)	/IEC 61158-5-3/
IEC 61158-6	Protokoll der Anwendungsschicht (Application Layer)	/IEC 61158-6-3/

Tabelle 3-1: Aufbau und Inhalt der Norm IEC 61158

Die Norm IEC 61784 unterteilt sich ebenfalls in mehrer Dokumente. Sie definiert für jeden Feldbus die verschiedenen Kommunikationsprofile. Für den Profibus gilt die CPF3 welche sich wiederum in drei Gruppen unterteilt:

- CPF3/1 für PROFIBUS DP
- CPF3/2 für PROFIBUS PA
- CPF3/3 für PROFINET CBA

Die Norm /IEC 61784-1/verweist auf die Norm IEC 61158 und weist den Kapiteln von IEC 61158 der jeweiligen Profibusvariante DP oder PA zu. /IEC 61784-1/ ist für eine Implementierung daher unbedingt erforderlich.



Bezeichnung	Inhalt	Erforderlicher Teil für Profibus
IEC 61784-1	Zuordnung der CPF zu Norm IEC 61158	/IEC 61784-1/
IEC 61784-2	Kommunikationsprofile für Echtzeit-Ethernet Nur für PROFINET (CPF3/3)	-
IEC 61784-3	Funktionssicherheit von Feldbussen Hier definiert als PROFISAFE (FSCP 3/1)	/IEC 61784-3-3/
IEC 61784-4	Profile für sichere Kommunikation Ausgelagert zur Norm IEC 62443	-
IEC 61784-5	Installationsprofile für Feldbusse	/IEC 61784-5-3/

Tabelle 3-2: Aufbau und Inhalt der Norm IEC 61784

3.2 Varianten

Es gibt drei Varianten des Profibus:

- Profibus DP
- Profibus PA
- Profibus FMS

Profibus FMS wurde durch Profibus DP abgelöst und ist im Profibusstandard nicht mehr enthalten. /WIKIP_ROFI/

Profibus DP, Kommunikationsprofil CPF3/1 siehe /IEC 61784-1/ Kapitel 7.2.

Meistgenutzte Variante des Profibus.

Zur Ansteuerung von Sensoren und Aktoren durch eine zentrale Steuerung und die Vernetzung von mehreren Steuerungen untereinander. Wird meist in der Fertigungstechnik eingesetzt. /WIKIP_ROFI/

Profibus DP verwendet violette Kabel, siehe /IEC 61158-2/ Kapitel 22.1.2.2.

Definition der Bitübertragungsschicht, siehe /IEC 61158-2/ Kapitel 22.1.

➔ In der vorliegenden Arbeit wird Profibus DP verwendet.

Profibus PA, Kommunikationsprofil CPF3/2 siehe /IEC 61784-1/ Kapitel 7.3.

Zur Kommunikation zwischen Mess- und Prozessgeräten, Aktoren und Prozessleitsystem.

Wird meist in der Prozess- und Verfahrenstechnik eingesetzt. /WIKIP_ROFI/

Profibus PA verwendet hellblaue Kabel, siehe /IEC 61158-2/ Kapitel 22.2.2.2.

PA ist die Eigensichere Variante des Profibus mit einer für explosionsgefährdete Bereiche definierten Bitübertragungsschicht (Physical Layer), siehe /IEC 61158-2/ Kapitel 22.2.

➔ In der vorliegenden Arbeit wird nicht auf den Profibus PA eingegangen.

Gemeinsamkeiten:

Der Profibus kann die Energieversorgung der angeschlossenen Geräte bereitstellen, siehe /IEC 61158-2/ Kapitel 21.12.

Profibus DP und Profibus PA Segmente können über einen Repeater verbunden werden.

Siehe /IEC 61158-2/ Abbildung 103 und 104.m

Die Sicherungsschichtdienste (Data Link Layer Services) sind bei beiden Varianten gleich, siehe /IEC 61784-1/ Kapitel 7.3.2.1.

Die Anwendungsschichtdienste (Application Layer Services) sind bei beiden Varianten gleich, siehe /IEC 61784-1/ Kapitel 7.3.3.

Unterschiede:

Die Bitübertragungsschicht (Physical Layer), Bitübertragungsschichtdienste (Physical Layer Services), Sicherungsschicht (Data Link Layer) und Anwendungsschicht (Application Layer) bei DP und PA weisen Gemeinsamkeiten auf unterscheiden sich aber voneinander in mehreren Punkten, siehe /IEC 61784-1/ Kapitel 7.2 für Profibus DP und /IEC 61784-1/ Kapitel 7.3.

3.3 Einordnung in das ISO/OSI-Referenzmodell

Feldbusse wie der Profibus benutzen das Feldbusreferenzmodell, siehe /IEC 61158-1/ Kapitel 6. Es ist ein auf drei Schichten vereinfachtes ISO/OSI-Referenzmodell. Die Funktionen der OSI-Schichten 3 und 4 werden dabei von der Schicht 2 (Sicherungsschicht) übernommen und die der Schichten 5 und 6 von der Schicht 7 (Anwendungsschicht). Das Feldbusreferenzmodell besteht daher nur aus der Bitübertragungsschicht (Schicht 1), der Sicherungsschicht und der Anwendungsschicht. Jede Schicht verfügt über eine eigene Systemverwaltung für das Management der von der jeweiligen Schicht benutzten Protokolle. Zwischen der Schichten regeln die Dienste den Datenaustausch mit der oberen Schicht. Die Anwendungsschichtdienste regeln den Datenaustausch mit Programmen oder Anwendungen der Feldbusse und gewähren so Zugriff auf die darunter liegenden Schichten.

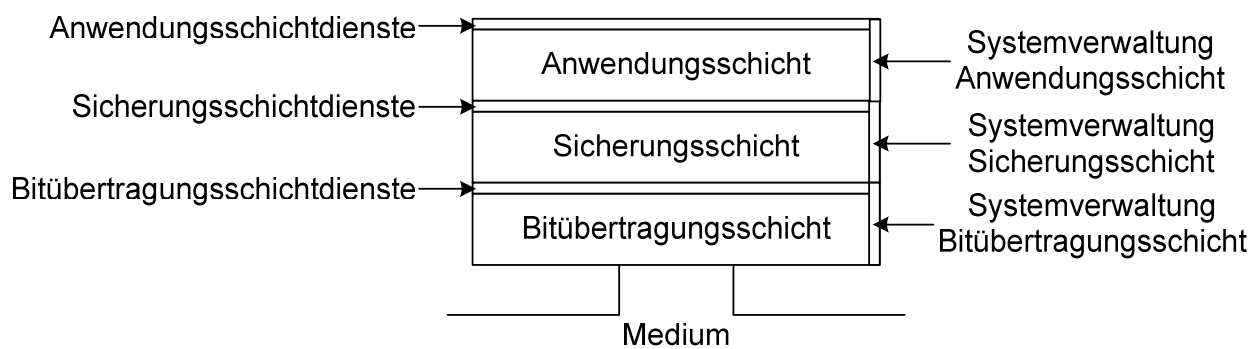


Abbildung 3-1: 3-Schichten Feldbusreferenzmodell

3.4 Topologie

Beim Profibus sind die Geräte an Abzweigungen entlang eines Hauptkabels angebracht, siehe **Abbildung 3-2** Seite 3-4. Diese Abzweigungen können in den Geräten selbst untergebracht sein, siehe /IEC 61158-2/ Kapitel 12.3.2.

Der Profibus kennt zwei Gerätekategorien, siehe /IEC 61158-2/, Kapitel 3.7.

- **Master:** kontrolliert mit dem Token-Passing den Medienzugriff der Master im Netzwerk und den Datenaustausch der Slaves zum Master. Er ist die Zentrale eines Profibusnetzwerks, steuert und koordiniert die Arbeit der Slaves.
- **Slave:** Erhält vom Master seine Aufgaben und führt dessen Befehle aus. Der Datenaustausch vom Slave zum Master erfolgt nur auf Anweisung vom Master.

Das Token-Passing zwischen den Mastern und der Datenaustausch zwischen Master und Slave erfolgt durch Telegramme, siehe 3.5 Datenaustausch durch Telegramme Seite 3-4.

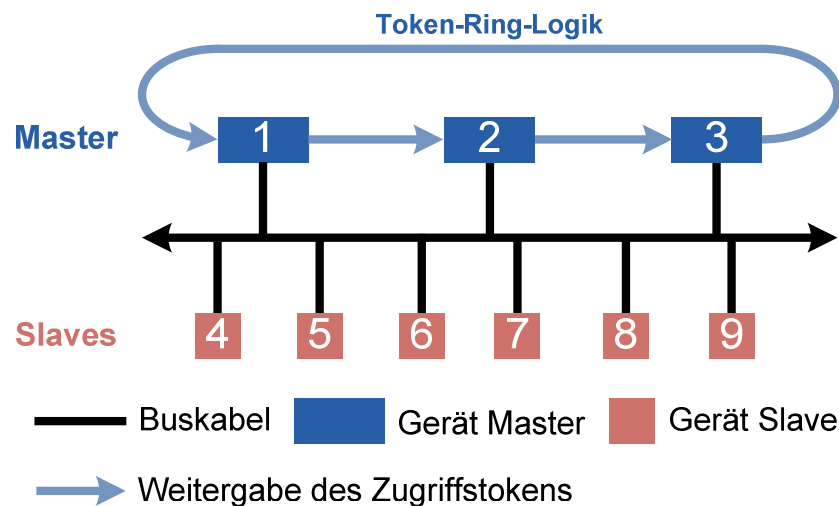


Abbildung 3-2: Master/Slave in der Bustopologie
/WIKIP_ROFI/

Der Profibus arbeitet mit einem hybriden Zugriffsverfahren, siehe Norm /IEC 61158-4-3/, Kapitel 5.1.

- Dezentral: Token-Passing (Token-Ring-Logik) zwischen den Masterstationen, ein Zugriffstoken wird dabei von Master zu Master gegeben. Das erfolgt nur bei mehreren Masters im Profibusnetzwerk
- Zentral: Master-Slave-Verfahren zwischen Master und Slave

Das Token-Passing erlaubt einen fairen Zugriff auf das Medium bei mehr Master am Profibus, Multi-Master-Mode. Nur der Master, welcher aktuell den Zugriffstoken hält, kann Daten über das Busnetzwerk senden.

3.5 Datenaustausch durch Telegramme

Der Datenaustausch, der Master/Slave-Zugriff und die Tokenweitergabe zwischen den Profibusgeräten erfolgt mittels Telegrammen, siehe /IEC 61158-4-3/ Kapitel 7.1.1.

Es gibt 5 Telegrammtypen die anhand ihrer Startsequenz (SD: Start Delimiter) identifiziert werden können, siehe /FELSER_WEB/.

- SD1: Telegramm ohne Daten
- SD2: Telegramm mit Daten variabler Länge
- SD3: Telegramm mit Daten fester Länge
- SD4: Token-Telegramm (Länge: 3 Byte)
- SC: Kurzquittung (Länge: 1 Byte)



Bedeutung der Abkürzungen:

SD*: Start Delimiter 1-4

DA: Destination Address

SA: Source Address

FC: Function Code

FCS: Frame Check Sequence

ED: End Delimiter

LE: Length

LEr: Length repeated

PDU: Protocol Data Unit

SC: Short Confirmation

Die Start Delimiter (SD1 – SD4), Short Confirmation (SC) und das End Delimiter (SD) haben feste Werte, dargestellt als hexadezimaler Wert. Variable Werte sind mit „var.“ bezeichnet.

Im Destination Address (DA) Feld können dezimale Werte von 0 bis 127 vorkommen, siehe 3.7 Adressierung Seite 3-7.

Im Source Address (SA) Feld können dezimale Werte von 0 bis 126 vorkommen, siehe 3.7 Adressierung Seite 3-7.

Die Felder Length (LE) und Length repeated (LEr) enthalten jeweils die Anzahl der Bits der Felder DA, SA, FC und PDU. Die Felder LE und LEr können so den Wert von 4 bis 249 enthalten.

3.5.1 SD1: Telegramm ohne Daten

Sendet der Master aus um neue Stationen im Profibusnetzwerk zu finden.

Gesamtlänge: 6 Byte

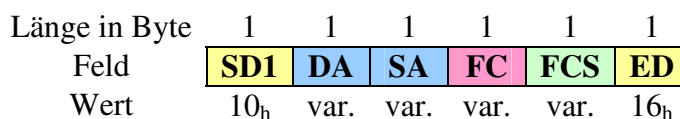


Abbildung 3-3: Aufbau Telegram ohne Daten

3.5.2 SD2: Telegramm mit Daten variabler Länge

Wird benutzt um Daten variabler Länge zu transportieren. Benutzt im Dienst SRD, siehe 3.6 Dienste Seite 3-6.

Gesamtlänge: 10 bis 255 Byte

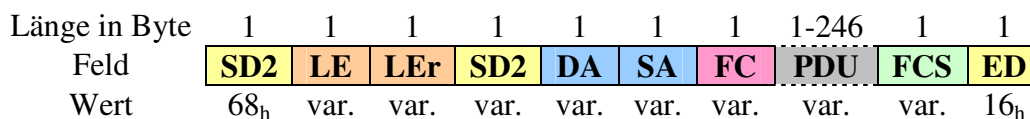


Abbildung 3-4: Aufbau Telegramm mit Daten variabler Länge

3.5.3 SD3: Telegramm mit Daten fester Länge

Wird benutzt als Antworttelegramm um Daten mit der festen Länge von 8 Bit zu transportieren.

Gesamtlänge: 14 Byte

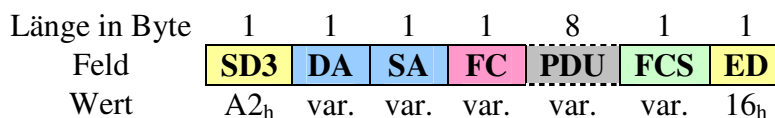
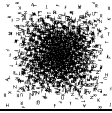


Abbildung 3-5: Aufbau Telegramm mit Daten fester Länge



3.5.4 SD4: Token-Telegramm

Wird als Token genutzt um den Mastern nach dem den Zugriff auf den Bus zur gewähren, siehe 3.4 Topologie Seite 3-3.

Gesamtlänge: 3 Byte

Länge in Byte	1	1	1
Feld	SD4	DA	SA
Wert	DC _h	var.	var.

Abbildung 3-6: Aufbau Token-Telegramm

3.5.5 SC: Kurzquittung

Wird als kurze positive Bestätigung genutzt.

Gesamtlänge: 1 Byte

Länge in Byte	1
Feld	SC
Wert	E5 _h

Abbildung 3-7: Aufbau Kurzquittung

➔ Für den Profibusmonitor wurden nur die Felder SD1 bis SD4, SC, ED, LE und LEr ausgewertet. Das ermöglicht eine exakte Bestimmung des Telegrammtyps, der Telegrammlänge und dem Ende des Telegramms.

3.6 Dienste

Die auf der Sicherungsschicht und Anwendungsschicht bereitgestellten Dienste werden nur anhand eines Beispiels erklärt.

Dienst der Datenübertragungsschicht:

- SDR: Send and Request Data

Wird benutzt um Daten an einen Slave zu versenden und gleichzeitig Daten vom Slave anzufordern. Die Antwort des Slaves enthält gleichzeitig auch die angeforderten Daten. Siehe dazu auch /FELSER_WEB/.

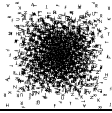
Dienst der Anwendungsschicht:

- Get_Configuration

Wird benutzt damit der Master die aktuelle gültige Konfiguration aus dem Slave auslesen kann. Siehe dazu auch /FELSER_WEB/.

Eine weiterführende Beschreibung findet sich in den Normen:

- Dienste der Sicherungsschicht, siehe Norm /IEC 61158-3-3/
- Dienste der Anwendungsschicht, siehe Norm /IEC 61158-5-3/



3.7 Adressierung

Jede Master- und Slave-Station in einem Profibusnetzwerk hat eine Adresse im Bereich von 0 bis 127. Wobei 127 die globale Adresse für Broadcast- und Multicast-Nachrichten ist, siehe /IEC 61158-4-3/, Kapitel 5.1.

Laut /FELSER_WEB/ hat sich folgende Konvention bewährt:

- 0: Reserviert für Diagnosewerkzeuge
- 1 bis n: Masteradressen, möglichst niedrig
- n bis 125: Slaveadressen, mit einem Master bleiben 124 Adressen für Slaves
- 126: Reserviert für Stationen im Auslieferungszustand
- 127: Reserviert als Broadcast- und Multicastadresse

3.8 Zeichenkodierung

Die einzelnen Zeichen werden dabei als in der UART-Zeichencodierung übertragen, siehe /IEC 61158-4-3/ Kapitel 6.1.1 und /ISO/IEC 1177/.

Die Codierung erfolgt byteweise und hat folgenden Aufbau, siehe /FELSER_WEB/.

Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Parität	Stop
-------	-------	-------	-------	-------	-------	-------	-------	-------	---------	------

Start = 0 ; Parität = gerade ; Stop = 1

Abbildung 3-8: Bits der UART-Codierung

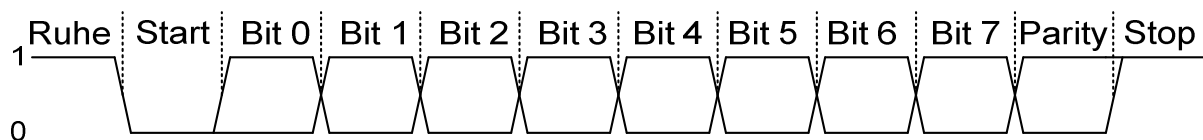


Abbildung 3-9: Impulsdiagramm UART-Codierung

Übertragungsregeln: /FELSER_WEB/

- Ruhezustand ist logisch 1
- Die Übertragung beginnt mit einem SYN (logisch 1) von mindestens 33 Bit Dauer
- Zwischen einzelnen Zeichen eines Telegramms sind keine Ruhezeiten erlaubt.
- Kontrolle des Empfängers pro Zeichen: Startbit, Stoppbit und Paritätsbit

3.9 Signalübertragung mittels RS-485

Die physikalische Signalübertragung in der Bitübertragungsschicht erfolgt mittels RS-485, siehe /IEC 61158-2/ Kapitel 22.1.1 und /TIA-485-A/.

RS-485 verwendet zwei Leiter zur Signalübertragung, siehe **Abbildung 3-12** Seite 3-8.

Das Signal des Leiters A ist invertiert gegenüber dem des Leiters B, siehe **Abbildung 3-10** Seite 3-8.

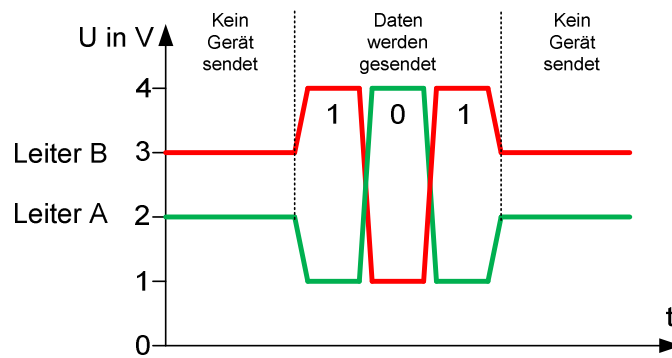
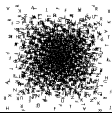


Abbildung 3-10: Impulsdiagramm der Spannungspegel Leiter B und Leiter A

	Leiter A	Leiter B
kein Gerät sendet	2V	3V
0 wird gesendet	4V	1V
1 wird gesendet	1V	4V

Tabelle 3-3. Werte der Spannungspegel RS-485

Die Spannungsdifference zwischen High- und Low-Pegel, gemessen zwischen B und A ergibt das übertragene das Bit 1 oder 0, siehe /FELSER_WEB/.

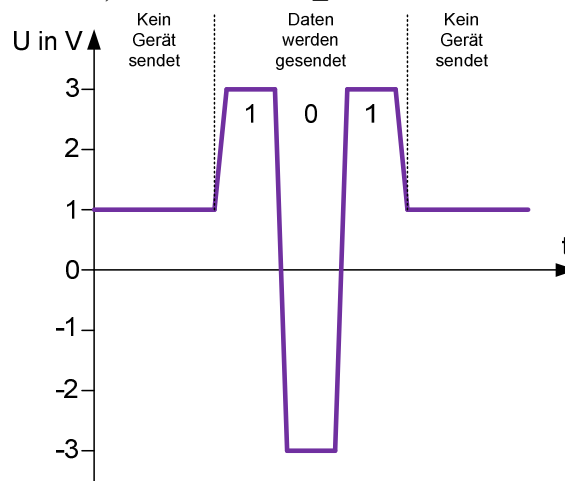


Abbildung 3-11: Impulsdiagramm der Spannungsdifference zwischen Leiter B und Leiter A

Die Bitübertragung erfolgt mit der Leitungscodierung „Non-Return-to-Zero“ (NRZ). Um die verwendeten Signalpegel aus der 5V Speisespannung (VP) zu erzeugen müssen die Enden des Profibuskabels mit einem Busabschluss terminiert sein, siehe /IEC 61158-2/ Kapitel 12.8.5.

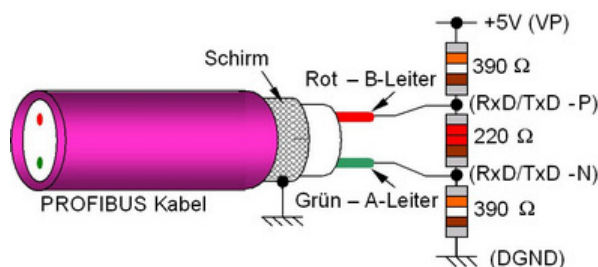


Abbildung 3-12: Busabschluss am Profibus Kabeltyp A /FELSER_WEB/



Der Busabschluss wird häufig im D-Sub 9 Stecker (Profibusstecker) des Profibuskabels verbaut und kann bei Lage am Busende eingeschaltet werden, siehe /FELSER_WEB/.

3.10 Kabel und Stecker

Für Kabelspezifikationen und Steckerbelegung siehe /IEC 61158-2/ Kapitel 22.1.2 und /FELSER_WEB/.

Als Medium für die Datenübertragung kommt ein geschirmtes, twisted-pair Kabel mit zwei Adern zum Einsatz, siehe /IEC 61158-2/ Kapitel 22.1.2.2 und **Abbildung 3-12** Seite 3-8.

Es gibt zwei Typen von Kabeln:

- Type A: aktuelle Kabelversion
- Type B: ältere Kabelversion, nicht geeignet für Neuinstallationen

Laut /FELSER_WEB/ wird die Verwendung des Type B Kabels nicht mehr empfohlen und bei einer Neuinstallation sollte nur noch der Kabeltyp Type A verwendet werden.

➔ Diese Arbeit bezieht sich daher auf den Kabeltyp Type A.

Beschreibung	Einheit	Werte								
Datenrate	kbit/s	9,6	19,2	93,75	187,5	500	1500	3000	6000	12000
Kabellänge Type A	m	1200	1200	1200	1000	400	200	100	100	100

Tabelle 3-4: Datenrate und Kabellänge Kabeltyp Type A /IEC 61158-2/

Es werden 9-polige D-SUB Profibus Stecker am Profibuskabel verwendet, welche einen zuschaltbaren Busabschluss enthalten. Die Buchsen dazu befinden sich am Profibusgerät, siehe /IEC 61158-2/ Anhang I.2

Pin Nr.	Signal	Funktion	Hinweis
1	Schirm	Schutzerde	nicht empfohlen
2	M24	Masse für 24V Spannung	nicht empfohlen
3	RxD/TxD-P	Datenleitung Plus (B-Leiter)	Pflicht
4	CNTR-P	Repeater Richtungskontrolle	Optional
5	DGND	Daten Masse	Pflicht
6	VP	+5V Speisung für Busabschluss	Pflicht
7	P24	+24V Speisung	nicht empfohlen
8	RxD/TxD-N	Datenleitung Minus (A-Leiter)	Pflicht
9	CNTR-N	Repeater Richtungskontrolle	Optional

Tabelle 3-5: Pinbelegung D-SUB Stecker für Profibus /FELSER_WEB/

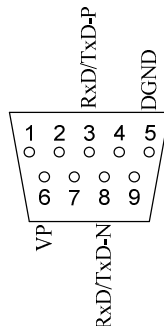


Abbildung 3-13: Pinlayout mit Pflichtbelegung; Frontalansicht Stecker; Rückansicht Buchse

4 Profibusmonitor

4.1 Stand der Technik

Der heutigen verfügbaren Profibusmonitore bieten laut /FELSER_WEB/ folgende Funktionen an:

- Signalanalyse: Messung der elektrischen Signale und Darstellung als zeitlicher Verlauf
- Telegrammanalyse: Aufzeichnung der Telegramme und Analyse und Auswertung der Inhalte
- Netzwerkmanagement: Ermittlung und Abbildung der Netzwerkstruktur sowie Zuordnung der Messungen zu den einzelnen Stationen
- Überwachung: Einsetzbar als permanentes Überwachungsgerät des Profibusnetzwerks mit Anbindung an Netzwerkmanagementdienste (SNMP) und konfigurierbaren Mailversand
- Steuerung: Steuerung der Signalanalyse, der Telegrammanalyse, des Netzwerkmanagement und Überwachungsfunktionen sowie deren Ablauf.
- Anzeige: Als Anzeige für die Profibusmonitore können unterschiedliche Geräte verwendet werden. Häufig handelt es sich dabei um einen via USB oder Ethernet angeschlossenen Rechner.

Eine Zusatzfunktion ist die Integration eines eigenen Webservers zu Bereitstellung der Daten über Ethernet z.B. zum Anzeigen der Messdaten auf mobilen Geräten wie Smartphone oder Tablet.

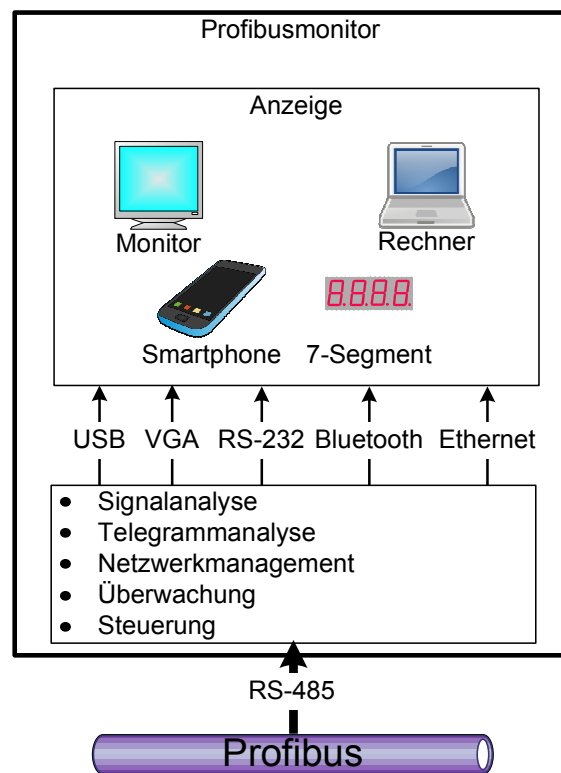


Abbildung 4-1: Aufbau Profibusmonitor

4.2 Warum Entwicklung eines eigenen Profibusmonitors?

Wenn es auf dem Markt schon fertige Profibusmonitore gibt, warum sollte man einen eigenen Profibusmonitor entwickeln?

Es gibt mehrere Gründe für die Entwicklung eines eigene Profibusmonitors:

- Ausbildung: Die FH Frankfurt kann jedes Teil des Profibusmonitors für Ausbildungszwecke verwenden, Hardware genauso wie die Software.
- Einblick in den Profibus: Der Profibusmonitor ermöglicht Einblick in die Funktionsweise des Profibusnetzwerks
- Realisierung mit FPGA: Die Realisierung erfolgt mit einem FPGA was deren Leitungsfähigkeit und Einsatzmöglichkeit demonstriert.
- Parallele Realisierung der Algorithmen: Die Algorithmen im eigenen Profibusmonitor liegen offen und sind ohne Copyright eines Herstellers
- Nur Telegrammanzeige: Der erste Entwicklungsschritt sieht einen auf die Anzeige der Telegramme reduzierten Funktionsumfang vor.
- Modulare Erweiterung: Schrittweise modulare Erweiterungen um weitere Funktionen sind möglich.

4.3 Konzeption eines eigenen Profibusmonitors

Folgende Funktionen aus 4.1 Stand der Technik Seite 4-1 werden in dieser Arbeit umgesetzt:

- a) Telegrammaufzeichnung aus der Telegrammanalyse: Erfassung, Verarbeitung und Aufzeichnung der Telegramme vom Profibus
- b) Anzeige von Telegrammen: Darstellung der Telegramme auf einer Anzeige.
- c) Ablaufsteuerung der Telegrammaufzeichnung: Steuerung des Ablaufs der Telegrammaufzeichnung, wird in die Telegrammaufzeichnung integriert

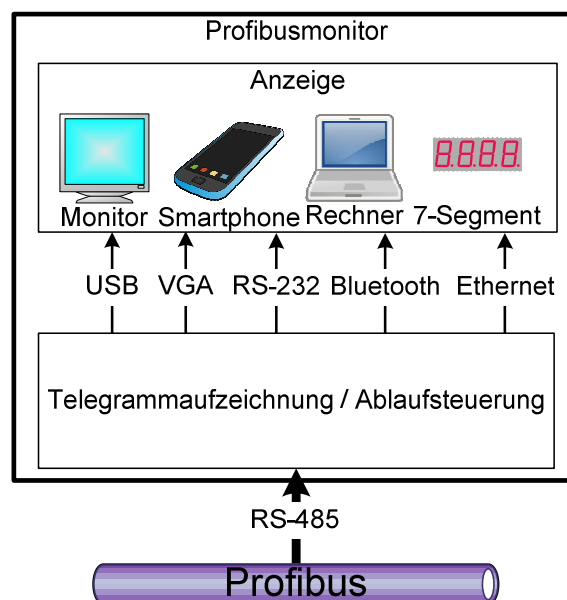


Abbildung 4-2: Konzept eigener Profibusmonitor

4.4 Vorgegebene Hardware (Spartan 3 Starter Kit)

Als Plattform für die Realisierung des eigenen Profibusmonitors soll das Spartan 3 Starter Kit, **Abbildung 4-3**, verwendet werden, siehe 2 Aufgabenstellung Seite 2-1.

Das Spartan 3 Starter Kit, siehe /GUIDE/ ist ein Entwicklungsplatine mit folgenden Elementen:

- Programmierbaren Chip (FPGA)
- Anzeigemöglichkeit aus vier 7-Segmentanzeigen und 8 LEDs
- sechs Schnittstellen (RS-232, VGA, PS/2 und drei 40-poligen Erweiterungsschnittstellen)
- Steuerungsmöglichkeit über 4 Taster und 8 Schalter

Das Spartan 3 Starter Kit wurde vorgegeben, da an den Arbeitsplätzen eine komplette Entwicklungsumgebung vorhanden ist und schon Erfahrungen in der Entwicklung damit vorhanden ist.

für die Umsetzung eines eigenen Profibusmonitors könnten z.B. auch folgende Plattformen genutzt werden:

- Rechner
- Mikrocontroller

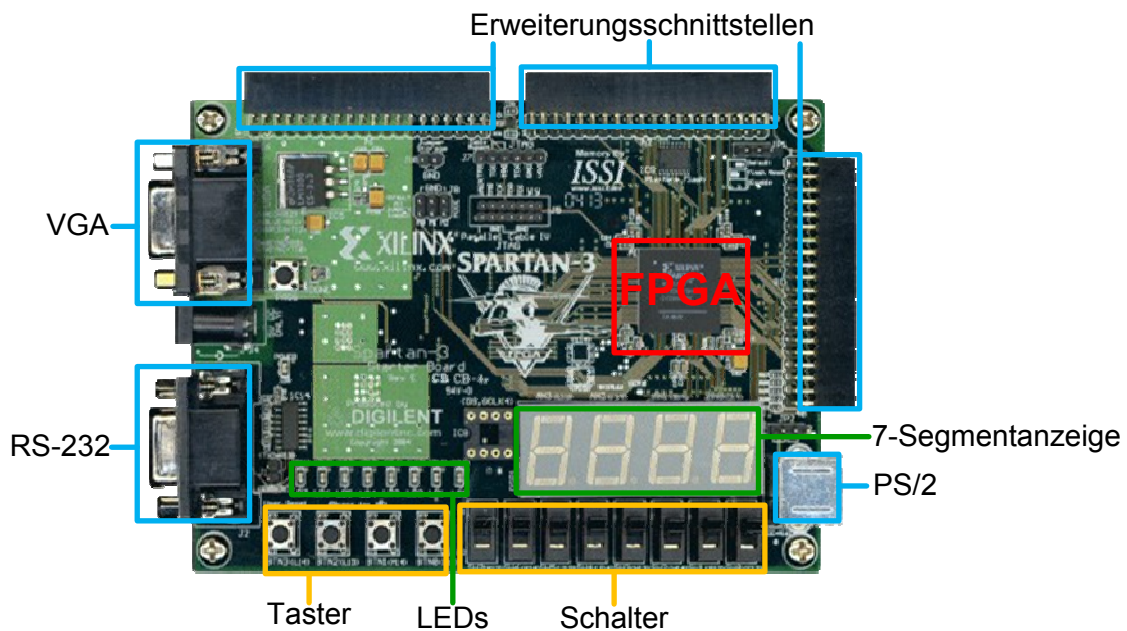


Abbildung 4-3: Spartan 3 Starter Kit mit einzelnen Elementen
/SPARTANBOARD/

5 Realisierte Lösung

5.1 Anzeige

Als Anzeige kommen viele Geräte in betracht wie z.B. ein Monitor, Smartphone, Rechner oder eine 7-Segmentanzeige. Diese Anzeigegeräte können auf verschiedene Art mit der Telegrammaufzeichnung verbunden sein wie z.B. über USB, VGA, RS-232, Bluetooth oder Ethernet.

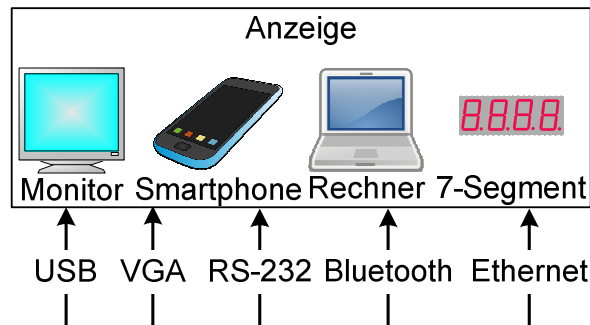


Abbildung 5-1: Konzept Anzeige

Das Spartan 3 Starter Kit, siehe **Abbildung 4-3** Seite 4-3, bietet nur VGA und RS-232 an. Eine Lösung über RS-232 mit einem angeschlossenen Rechner auf dem die Telegramme in einem Terminalprogramm dargestellt werden lässt sich mit wenig Aufwand realisieren. Siehe dazu auch /MILLER/ und Kapitel 7 im /GUIDE/. Die Lösung über VGA einen Monitor als Anzeigegerät zu verwenden ist ebenfalls möglich, erfordert aber einen weit höheren Realisierungsaufwand, siehe /GUIDE/ Kapitel 5.

Bluetooth, USB und Ethernet benötigen spezielle Hardware und einen sehr viel höheren Realisierungsaufwand.

Die Benutzung der im Spartan 3 Starter Kit integrierten 7-Segmentanzeige bietet aufgrund der nur vier darstellbaren Zeichen zu wenig Platz zur Anzeige der Telegramme.

Die Entscheidung fiel daher für eine Lösung mit RS 232 und Rechner, siehe **Abbildung 5-2**.

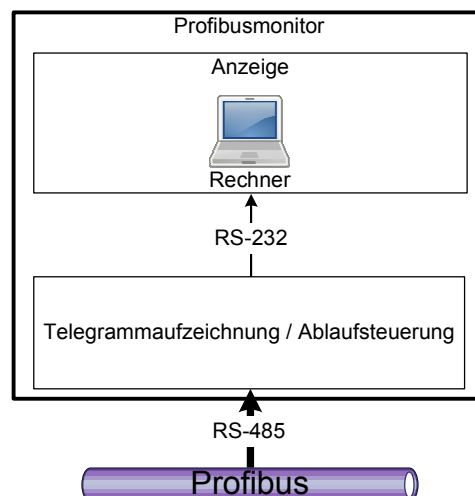


Abbildung 5-2: Konzept eigener Profibusmonitor mit Anzeige via RS-232 und Rechner

5.2 Telegrammaufzeichnung

In der Telegrammaufzeichnung wird das RS-485 Signal erfasst und die darin enthaltenen Telegramme aufgezeichnet. Der **Abbildung 4-2** Seite 4-2 kann entnommen werden, dass die Telegrammaufzeichnung eine Schnittstelle zum Profibus und eine Schnittstelle zur Anzeige benötigt. Zwischen den beiden Schnittstellen sitzt eine Verarbeitungseinheit die für eine Umwandlung der RS-485 Signale zu Telegrammen sorgt. In der Verarbeitungseinheit integriert wird die Ablaufsteuerung. Durch eine Steuerung kann der Ablauf der Verarbeitung direkt gesteuert werden.

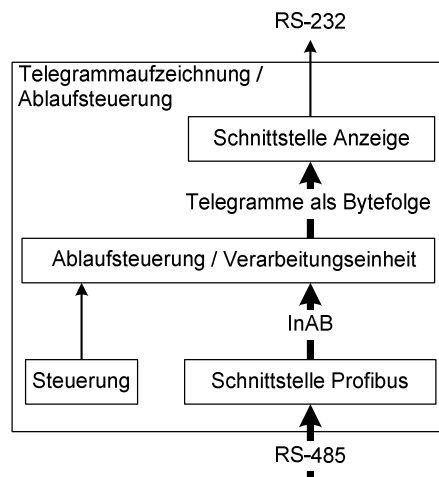


Abbildung 5-3: Aufbau Telegrammaufzeichnung

Die Schnittstelle Anzeige welche das RS-232 Signal für die Anzeige bereitstellt ist die RS-232 Schnittstelle des Spartan 3 Kit, siehe **Abbildung 4-3** Seite 4-3. Diese benötigt die Telegramme als Bytefolge um sie via RS-232 Signal an die Anzeige senden zu können.

Die Verarbeitungseinheit und die Ablaufsteuerung befindet sich im FPGA, dem programmierbaren Chip des Spartan 3 Starter Kit, siehe **Abbildung 4-3** Seite 4-3. Die Telegramme werden als Bytefolge von der Verarbeitungseinheit ausgegeben.

Die Schnittstelle Profibus kann nicht auf dem Spartan 3 Kit realisiert werden, da dort keine RS-485 Schnittstelle zur Verfügung steht. Sie muss daher extern mit eigenständiger Hardware realisiert werden, siehe Kapitel 5.3 Schnittstelle Profibus Seite 5-3. Dort werden die RS-485 Signalpegel in das Signal InAB umgewandelt.

Um den Ablauf der Telegrammaufzeichnung beeinflussen zu können, wird die Ablaufsteuerung ebenfalls in den FPGA integriert. Die Ablaufsteuerung wird bedient durch eine Steuerung welche mittels Schalter und Taster des Spartan 3 Starter Kit, siehe **Abbildung 4-3** Seite 4-3, erfolgt.

Die **Abbildung 5-4** Seite 5-3 zeigt diese Zuordnung der Bestandteile der Telegrammaufzeichnung zum Spartan 3 Starter Kit.

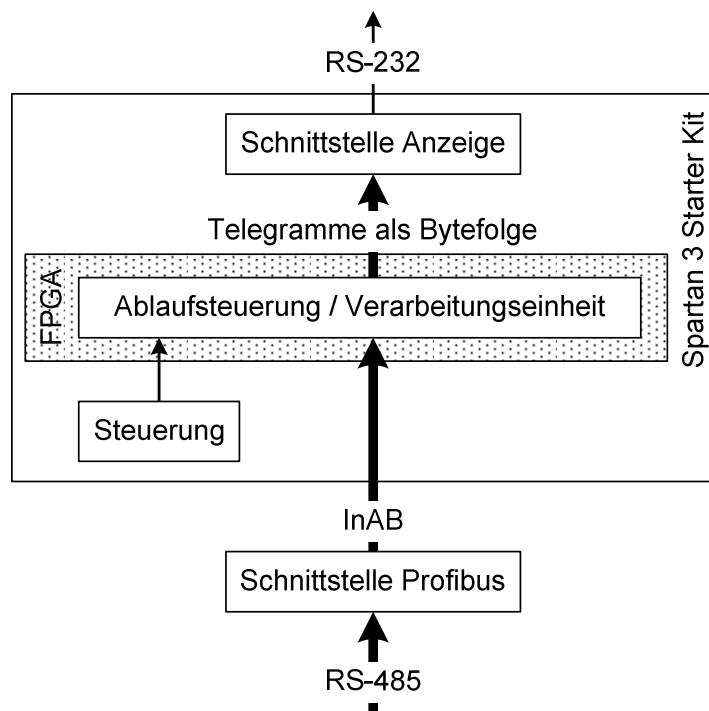


Abbildung 5-4: Zuordnung der Bestandteile der Telegrammaufzeichnung

5.3 Schnittstelle Profibus

Die Schnittstelle Profibus hat zwei Aufgaben:

- Umsetzung der RS-485 Signalpegel vom Profibus zum Signal InAB
- Reduzierung möglicher Störungen

Als Grundlage dient eine Schaltung aus den Büchern von Manfred Popp, siehe /POPP/ Kapitel 5.4 Abbildung 5.7 Seite 22. Diese Schaltung kombiniert die RS-485 Signalpegel der Leiter A und Leiter B zu dem Signal InAB.

Die Reduzierung möglicher Störungen wird durch eine elektrische Abtrennung des Profibusnetzwerks durch einen Optokoppler vorgenommen. Die Hardware der Schnittstelle Profibus besteht aus zwei getrennten Stromkreisen. Die RS-485 Signalpegel der Leiter A und B gelangen zu einem BUS Transceiver. Der BUS Transceiver wandelt die Signalpegel in das Signal InAB um, welches dann in den Optokoppler geleitet wird. Im Optokoppler erfolgt dann eine potentialfreie Weiterleitung des Signals InAB. Das so potentialfreie Signal InAB gelangt dann zum Spartan 3 Starter Kit.

Ein DC/DC-Wandler übernimmt die Stromversorgung der Stromkreise und sorgt gleichzeitig für eine Potentialtrennung, siehe **Abbildung 5-5** Seite 5-4.

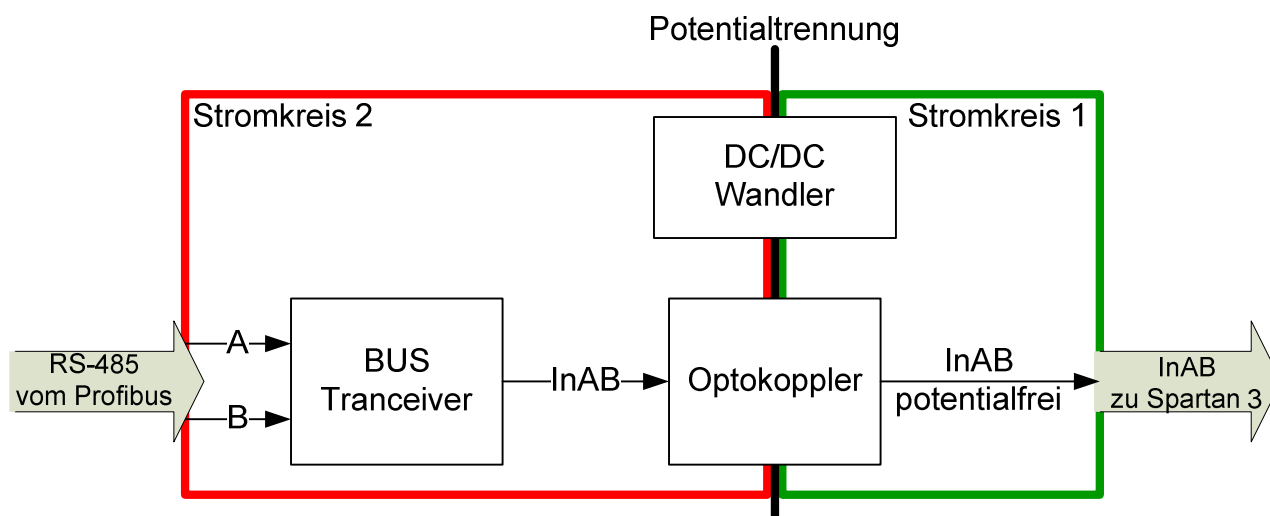


Abbildung 5-5: Grundlegender Aufbau Schnittstelle Profibus

Die Schnittstelle Profibus sorgt für eine Umwandlung der RS-485 Pegel in das Signal InAB mit folgenden Pegeln:

	InAB
kein Gerät sendet	5V
0 wird gesendet	0V
1 wird gesendet	5V

Tabelle 5-1: Signalpegel Bitfolge InAB

Die Zeichencodierung des Profibus, siehe 3.8 Zeichencodierung Seite 3-7, ändert sich dabei nicht.

5.4 Verarbeitungseinheit / Ablaufsteuerung

Um vom Signal InAB aus der Schnittstelle Profibus zu den Telegrammen als Bytefolge für die Schnittstelle Anzeige zu kommen ist eine Zerlegung der Ablaufsteuerung / Verarbeitungseinheit notwendig, siehe **Abbildung 5-4** Seite 5-3. Diese noch kleineren Einheiten werden Module genannt da sie innerhalb des Programmierbaren Chips (FPGA) des Spartan 3 Starter Kit, siehe **Abbildung 4-3** Seite 4-3, realisiert werden.

Die Umwandlung des Signals InAB in EN_BIT_i und BIT_VALUE erfolgt im Modul InAB_INPUT. EN_BIT_i und BIT_VALUE werden dann im Modul BIT_REGISTER zu der Bytefolge BYTE_OUT umgewandelt. Diese Bytefolge BYTE_OUT wird nun einer Analyse unterzogen um die Telegramme darin zu erkennen. Die Analyse der Bytefolge BYTE_OUT erfolgt im Modul TELEGRAM_CHECK. Dieses Modul gibt Steuerzeichen aus, die es ermöglichen die Telegramme in der Bytefolge BYTE_OUT zu kennzeichnen, siehe **Abbildung 5-6**.

Die Steuerzeichen in Verbindung mit der Bytefolge BYTE_OUT ergeben dann die Telegramme als Bytefolge welche an die Schnittstelle Anzeige weitergeleitet werden.

Die Ablaufsteuerung gliedert sich in zwei Teile auf:

- Ablaufsteuerung 1 im Modul InAB_INPUT
- Ablaufsteuerung 2 im Modul TELEGRAM_CHECK

Beide können mit der Steuerung, siehe **Abbildung 5-3** Seite 5-2, und deren Funktionen ERROR_CTRL, TELEGRAM_STOP, TELEGRAM_RUN und CHOSE-VALUE direkt in das Verhalten der Module TELEGRAM_CHECK und InAB_INPUT eingreifen, siehe **Abbildung 5-6**.

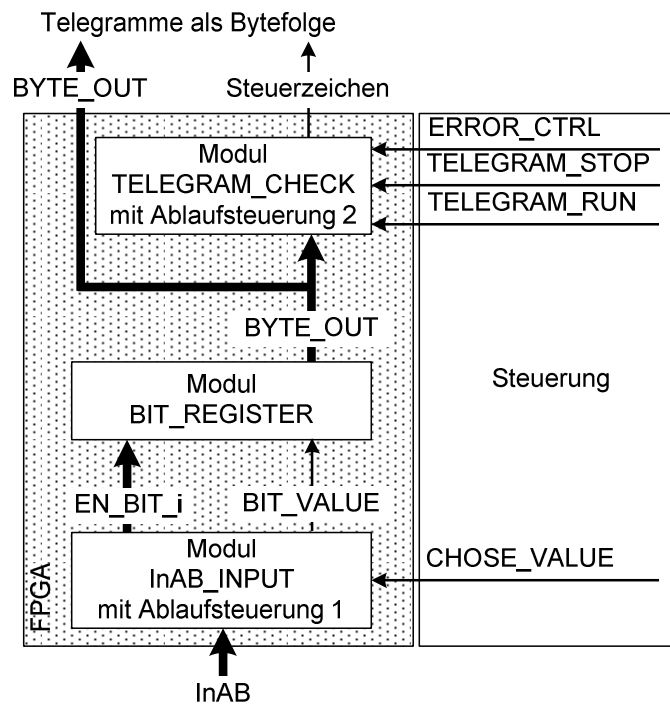


Abbildung 5-6: Aufbau Verarbeitungseinheit / Ablaufsteuerung

Der sich aus **Abbildung 5-6** ergebende Wirkungsplan der Verarbeitungseinheit / Ablaufsteuerung ist in der **Abbildung 5-7** Seite 5-6 dargestellt.

Das Signal InAB kommt über den Eingang InAB in die Verarbeitungseinheit / Ablaufsteuerung.

Zu den Steuersignalen der Steuerung, siehe **Abbildung 5-3** Seite 5-2, für die Ablaufsteuerung 1 und 2, siehe **Abbildung 5-6** Seite 5-5, gehören folgende Eingänge:

- CHOSE_VALUE: Umschalten der Zähler zwischen Normal- und Schrittbetrieb
- TELEGRAM_RUN: Start der Analyse der Bytefolge nach Telegrammen
- TELEGRAM_STOP: Stopp der Analyse der Bytefolge nach einem Telegramm
- ERROR_CTRL: Bestätigung und Löschen von Fehlern bei der Telegrammerkennung

Variablendefinition der Steuersignale siehe 8.1 Variablendefinition Steuersignale Steuerung Seite 8-2.

Die Bytefolge, welche die Telegramme enthält wird am Ausgang BYTE_OUT ausgegeben.

Zu den Steuerzeichen, siehe **Abbildung 5-6** Seite 5-5, gehören folgende Ausgänge:

- SEND_OUT: Ein Byte eines Telegramms wurde korrekt erkannt.
- T_LENGTH: Ausgabe der Länge des aktuellen Telegramms, siehe Kapitel 3.5.
- T_TYPE: Ausgabe des aktuellen Telegrammtyps, siehe Kapitel 3.5.

- T_END: Ende des aktuellen Telegramms erreicht.
- PARITY_FAIL: Fehler in der Parität des aktuellen Bytes
- NO_ED: Fehler, da kein End Delimiter erkannt wurde, siehe Kapitel 3.5.
- WORKING: Anzeige, das die Analyse der Bytefolge aktiv ist.
- KNOWN_T: Anzeige, das ein Telegramm erkannt wurde.
- UNKNOWN_BYTE: Fehler, kein Telegramm in der aktuellen Bytefolge erkannt.

Variablendefinition der Steuerzeichen siehe 8.2

Variablendefinition Steuerzeichen Verarbeitungseinheit Seite 8-2.

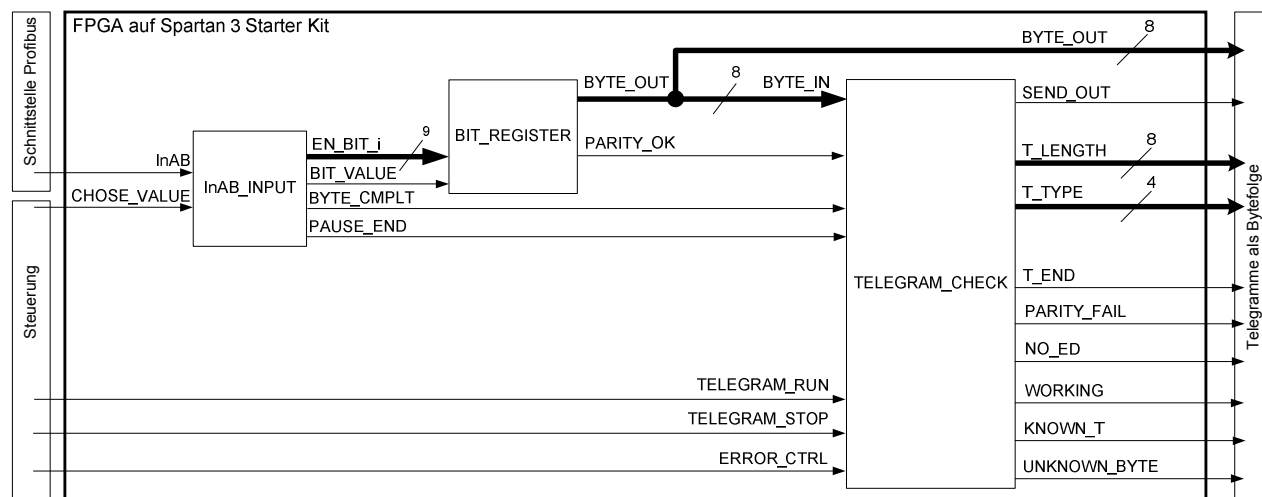


Abbildung 5-7: Wirkungsplan Verarbeitungseinheit / Ablaufsteuerung

5.4.1 Modul InAB_INPUT

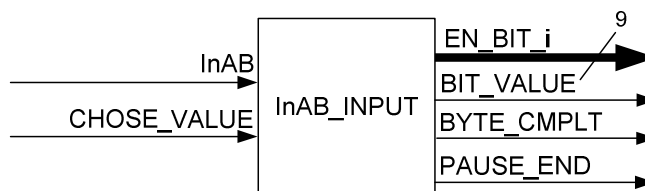


Abbildung 5-8: Wirkungsplan Modul InAB_INPUT

Das Modul InAB_INPUT erfasst die Bits aus dem Signal InAB am Eingang InAB und gibt diese weiter an das Modul BIT_REGISTER, siehe 5.4.2 Modul BIT_REGISTER Seite 5-13, über die Ausgänge EN_BIT_i und BIT_VALUE.

Der Eingang CHOSE_VALUE dient zum Umschalten zwischen Normalbetrieb, logisch 0 am Eingang oder Schrittbetrieb, logisch 1 am Eingang.

Der Ausgang BYTE_CMPLT gibt eine logische 1 aus, wenn ein komplettes Byte aus dem Signal InAB vom Modul InAB_INPUT erfasst wurde.

Der Ausgang PAUSE_END gibt eine logische 1 aus wenn das Ende eines SYN erkannt wurde.

Variablendefinition der Ausgänge, siehe 8.3

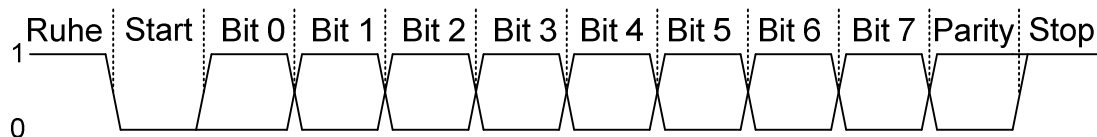
Variablendefinition der Ausgänge Modul InAB_INPUT Seite 8-2



Das Signals InAB enthält eine Bitfolge, welche hat die selbe Zeichencodierung wie der Profibus hat, siehe 5.3 Schnittstelle Profibus Seite 5-3 und 3.8 Zeichenkodierung Seite 3-7.

	InAB
kein Gerät sendet	5V
0 wird gesendet	0V
1 wird gesendet	5V

Siehe Tabelle 5-1: Signalpegel Bitfolge InAB Seite 5-4



Siehe Abbildung 3-9: Impulsdigramm UART-Codierung Seite 3-7

Das Modul InAB_INPUT muss daher die Pegel des Signals InAB erfassen und als Null oder Eins interpretieren. Mit Hilfe des Start- und Stoppbits kann jeweils der Anfang und das Ende der aktuellen Bitfolge erkannt werden. Dabei sind die Übertragungsregeln des Profibus zu beachten, siehe 3.8 Zeichenkodierung Seite 3-7.

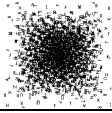
Übertragungsregeln: /FELSER_WEB/

- Ruhezustand ist logisch 1
- Die Übertragung beginnt mit einem SYN (logisch 1) von mindestens 33 Bit Dauer
- Zwischen einzelnen Zeichen eines Telegramms sind keine Ruhezeiten erlaubt.
- Kontrolle des Empfängers pro Zeichen: Startbit, Stoppbit und Paritätsbit

Die **Abbildung 5-9** Seite 5-10 zeigt beispielhaft eine Bitfolge des Signals InAB. Mit Hilfe von Zählern wird bestimmt, wann welches Teilstück der Bitfolge des Signals InAB gerade als Spannungspegel am Modul InAB_INPUT ankommt und erfasst werden soll.

Die **Abbildung 5-10** Seite 5-12 zeigt die Verhaltensbeschreibung des Moduls InAB_INPUT als Programmablaufgraph (PAG) für:

- Start einer Übertragung (SYN von 33 Bit) erkennen
- Startbit erkennen
- Erkennen von Bit1 bis Bit9
- Erkennung des Stoppbits
- Endeerkennung



Start einer Übertragung (SYN von 33 Bit) erkennen:

Der Ruhezustand ist logisch 1 und eine Übertragung beginnt mit einem SYN (logisch 1) von mindestens 33 Bit Dauer, siehe **Abbildung 5-9** Seite 5-10 und **Abbildung 5-10** Seite 5-12.

- Sobald eine logische 1 an Eingang InAB erkannt wurde startet der Zähler 1($t_{\text{SYN}0}$).
- Zähler 1 zählt solange eine logische 1 übertragen wird bis zum Zeitpunkt ($t_{\text{SYN}30}$) an dem 30 von 33 Bit des SYN übertragen wurden und das SYN damit als erkannt gilt.
- Anschließend wird auf den Anfang der Bitfolge gewartet, welche mit dem Startbit (logisch 0) beginnt
- Ausgang PAUSE_END und BYTE_CMPLT sind während der Erkennung des Starts der Übertragung logisch 0.

Startbit erkennen:

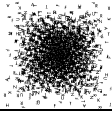
Nachdem das SYN erkannt wurde, folgt das Startbit welches logisch 0 ist, , siehe **Abbildung 5-9** Seite 5-10 und **Abbildung 5-10** Seite 5-12.

- Zähler 2 startet sobald eine logische 0 erkannt wurde (t_0).
- Ausgang PAUSE_END wird logisch 1 da SYN beendet ist.
- Erreicht Zähler 2 die zeitliche Mitte (t_1) des Startbits (Bit0) wird der aktuelle Spannungspegel erfasst und als logische 0 oder 1 interpretiert.
- Bei einer logischen 0 gilt das Startbit als erkannt, es folgt das Erkennen von Bit1
- Bei einer logischen 1 wurde kein Startbit erkannt und es wird auf den nächsten Start einer Übertragung gewartet.
- Ausgang PAUSE_END wird wieder logisch 0.
- Ausgang BYTE_CMPLT ist während der Erkennung des Startbits logisch 0.

Erkennen von Bit1 bis Bit9:

Nachdem erkennen des Startbit (Bit0) ist der Zähler 2 weitergelaufen, siehe **Abbildung 5-9** Seite 5-10 und **Abbildung 5-10** Seite 5-12.

- Erreicht Zähler 2 die zeitliche Mitte (t_2) des Bit1 wird der aktuelle Spannungspegel erfasst und als logische 0 oder 1 interpretiert.
- Der Zähler 2 läuft weiter bis zur Mitte (t_3 bis t_{10}) des nächsten Bit (Bit2 – Bit9).
- Der aktuelle Spannungspegel wird erfasst und als logische 0 oder 1 für das Bit (Bit2 – Bit9) interpretiert.



Erkennung des Stoppbits:

Nach dem Paritätsbit (Bit9) folgt die Erkennung des Stoppbits (Bit10) welches logisch 1 ist, siehe **Abbildung 5-9** Seite 5-10 und **Abbildung 5-10** Seite 5-12.

- Von der Mitte (t_{10}) des Bit9 läuft der Zähler 2 weiter bis zur Mitte (t_{11}) des Stoppbits (Bit10).
- Dort wird der aktuelle Spannungspegel erfasst und als logische 0 oder 1 interpretiert.
- Bei einer logischen 1 gilt das Stoppbit als erkannt.
- Ausgang BYTE_CMPLT wird logisch 1 da das Stoppbit erkannt wurde, es folgt die Endeerkennung.
- Bei einer logischen 0 für das Stoppbit wurde kein Stoppbit erkannt und es wird auf den nächsten Start einer Übertragung gewartet.
- Ausgang BYTE_CMPLT bleibt logisch 0.
- Ausgang PAUSE_END ist während der Erkennung des Stoppbits logisch 0.

Endeerkennung:

Nachdem das Stoppbit (Bit10) erkannt wurde, muss das Ende der Bitfolge erkannt und auf den Beginn einer neuen Bitfolge (neues Startbit) oder den Start einer neuen Übertragung (SYN) gewartet werden, siehe **Abbildung 5-9** Seite 5-10 und **Abbildung 5-10** Seite 5-12.

- Von der Mitte (t_{10}) des Stoppbit (Bit10) läuft der Zähler 2 weiter bis 70% des Stoppbits übertragen wurden (t_{12}).
- Dort wird der aktuelle Spannungspegel erfasst und als logische 0 oder 1 interpretiert bis der Zähler 30% (t_{13}) des folgenden Bits (Startbit oder SYN).
- Bei einer logischen 1 wurde kein neues Startbit erkannt und es wird auf den nächsten Start einer Übertragung gewartet
- Bei einer logischen 0 wurde ein Startbit erkannt und es folgt das Erkennen von Bit1 bis Bit9.

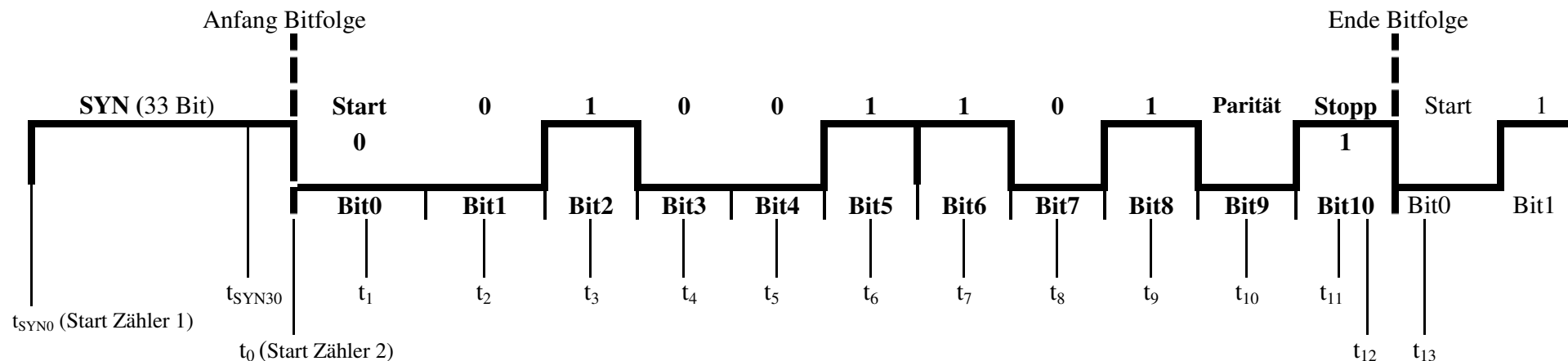


Abbildung 5-9: Beispiel Impulsdiagramm der Bitfolge InAB mit Zählerzeitpunkten

Der Zähler 1 läuft vom Beginn des SYN t_{SYN0} (Start Zähler 1) bis zum Zählerwert t_{SYN30}. Der Zähler 2 läuft von t₀ (Start Zähler 2) bis zum Zählerwert t₁₃. Zählerwerte t_{SYN0}, t_{SYN30} und t₀ bis t₁₃ siehe **Tabelle 5-2** Seite 5-11.

Berechnung der Zählerwerte der Zählerzeitpunkte:

Der Systemtakt des Spartan 3 Starter Kit ist 50.000.000 Hz (50 MHz), siehe /GUIDE/ Kapitel 8. Die Baudrate beträgt 9600 Bit/s, fest eingestellt am Profibus Master.

- Berechnung der Übertragungsdauer von n Bits (t_{BIT}):
 - $t_{\text{BIT}} = n \text{ Bit} / \text{Baudrate}$
- Berechnung der Systemtakte (k) für die Übertragung von n Bits (t_{BIT}):
 - $t_{\text{BIT}} = k / \text{Systemtakt}$
 - $k = t_{\text{BIT}} * \text{Systemtakt}$
 - $k = (n \text{ Bit} / \text{Baudrate}) * \text{Systemtakt}$



Zählerzeitpunkte	Berechnung (n Bit / Baudrate) * Systemtakt	Zählerwert lang, dezimal	Zählerwert lang, hexadezimal	Zählerwert kurz, dezimal	Zählerwert kurz, hexadezimal	Variablenname
t _{SYN0}	÷	0	0	0	0	÷
t _{SYN30}	(30 / 9600) * 50000000	156250	2625A	10	A	CNTS30
t ₀	÷	0	0	0	0	÷
t ₁	(0,5 / 9600) * 50000000	2604	0A2C	3	3	CNTT01
t ₂	(1,5 / 9600) * 50000000	7812	1E84	6	6	CNTT02
t ₃	(2,5 / 9600) * 50000000	13020	32DC	9	9	CNTT03
t ₄	(3,5 / 9600) * 50000000	18229	4735	12	C	CNTT04
t ₅	(4,5 / 9600) * 50000000	23435	5B8B	15	F	CNTT05
t ₆	(5,5 / 9600) * 50000000	28644	6FE4	18	12	CNTT06
t ₇	(6,5 / 9600) * 50000000	33854	8441	21	15	CNTT07
t ₈	(7,5 / 9600) * 50000000	39062	9872	24	18	CNTT08
t ₉	(8,5 / 9600) * 50000000	44270	ACEE	27	1B	CNTT09
t ₁₀	(9,5 / 9600) * 50000000	49479	C147	30	1E	CNTT10
t ₁₁	(10,5 / 9600) * 50000000	54687	D59F	33	21	CNTT11
t ₁₂	(10,7 / 9600) * 50000000	55729	D9B1	36	24	CNTT12
t ₁₃	(11,3 / 9600) * 50000000	58854	E5E6	42	2A	CNTT13

Tabelle 5-2: Zählerwerte Modul InAB_INPUT

Die Ergebnisse der Berechnung der „Zählerwerte lang, dezimal“ wurden abgerundet.

Der „Zählerwert lang, dezimal“ und „Zählerwert lang, hexadezimal“ sind die Zählerwerte im Normalbetrieb.

Der „Zählerwert kurz, dezimal“ und „Zählerwert kurz, hexadezimal“ sind die Werte für den Schrittbetrieb.

Der Variablenname wird in der Verhaltensbeschreibung als Programmablaufgraph (PAG) des Moduls InAB_INPUT genutzt, siehe **Abbildung 5-10** Seite 5-12.

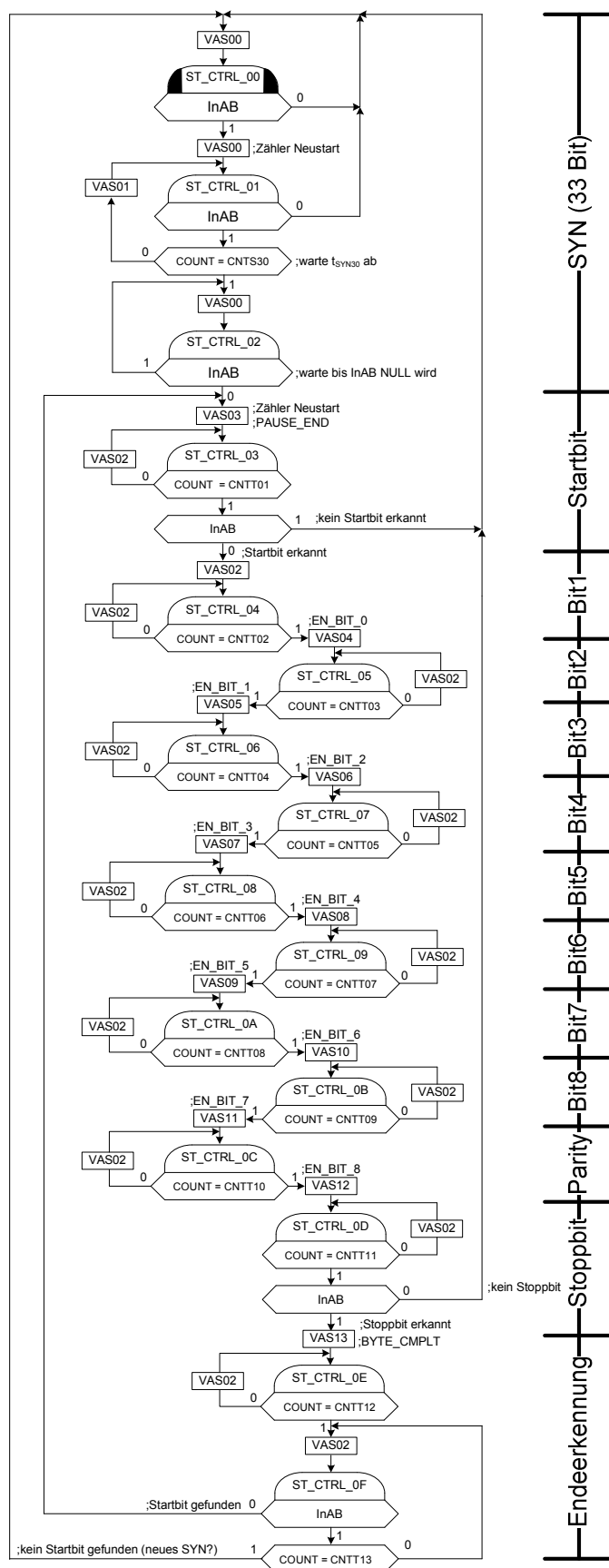


Abbildung 5-10: Verhaltensbeschreibung als PAG Modul InAB_INPUT

5.4.2 Modul BIT_REGISTER

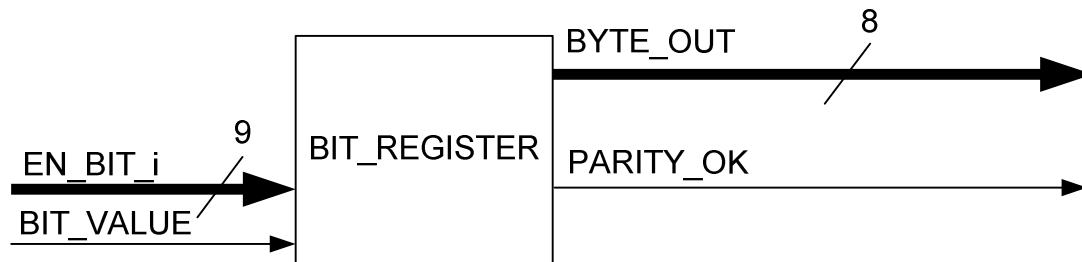


Abbildung 5-11: Wirkungsplan Modul BIT_REGISTER

Das Modul BIT_REGISTER speichert die Ausgabe EN_BIT_i und BIT_VALUE vom Modul InAB_INPUT, siehe 5.4.1 Modul InAB_INPUT Seite 5-6, zwischen und wandelt sie in eine Bytefolge (BYTE_OUT) um. Zusätzlich erfolgt im Modul die Berechnung der Parität des der Bitfolge InAB, siehe **Abbildung 5-9** Seite 5-10.

Es gibt neun Register (BIT_REGISTER_EN_BIT_0 bis BIT_REGISTER_EN_BIT_8) im Modul. Mit dem Ansprechen von EN_BIT_0 bis EN_BIT_8 am Eingang EN_BIT_i wird das dazugehörige Register im Modul freigegeben auf dem der logische Wert, welcher zeitgleich am Eingang BIT_VALUE anliegt, gespeichert wird, siehe **Abbildung 5-12**. Die Register werden nacheinander vom Modul InAB_INPUT angesprochen, wenn dort die Erkennung der Bits1 bis Bit9, siehe Seite 5-8 und **Abbildung 5-10** Seite 5-12, erfolgt.

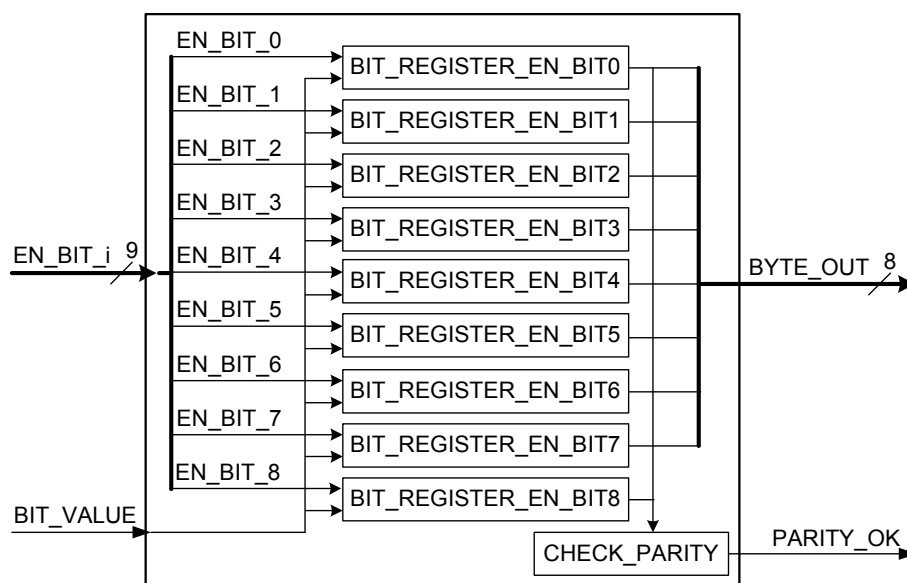


Abbildung 5-12: Register im Modul BIT_REGISTER

Am Ausgang BYTE_OUT liegen ständig die in den Registern BIT_REGISTER_EN_BIT0 bis BIT_REGISTER_EN_BIT7 gespeicherten Werte an, siehe 8.4 Variablendefinition Ausgänge Modul BIT_REGISTER Seite 8-2.

Die Register arbeiten nachdem in **Abbildung 5-13** dargestellten Programmablaufgraphen (PAG).

Im Zustand ST_BR_00 ist der Ausgang BYTE_OUT(i) (i steht für die Zahlen 1 bis 8) logisch 0. Wenn der Eingang EN_BIT_i mit einer logischen 1 angesprochen wird und der logische Wert des Eingangs BIT_VALUE ebenfalls 1 ist, wird in den Zustand ST_BR_01 gewechselt. Im Zustand ST_BR_01 ist der Ausgang BYTE_OUT(i) logisch 1. In den Zustand ST_BR_00 kann nur zurück gewechselt werden wenn EN_BIT_i mit einer logischen 1 angesprochen wird und der Wert des Eingangs BIT_VALUE logisch 0 ist

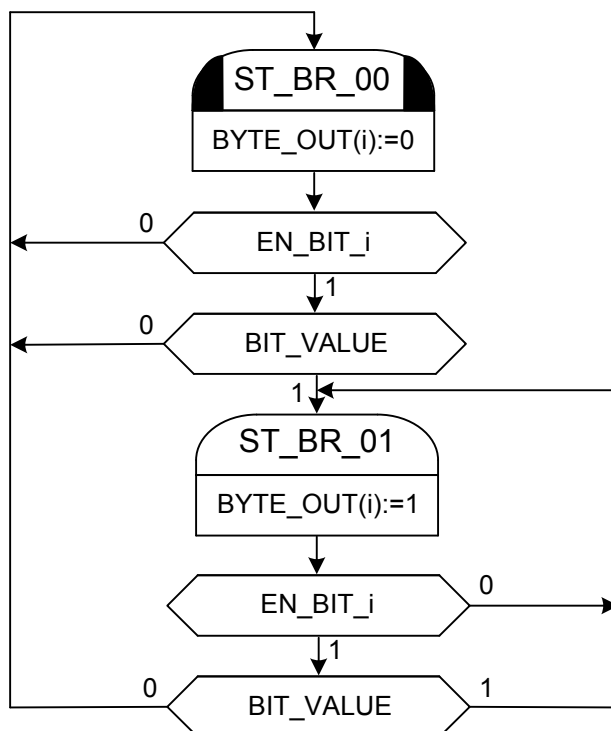


Abbildung 5-13: Verhaltensbeschreibung als PAG Modul BIT_REGISTER

In der Untereinheit CHECK_PARITY, siehe **Abbildung 5-12** Seite 5-13, des Moduls BIT_REGISTER erfolgt eine laufende gerade Paritätsprüfung der Werte in den Registern BIT_REGISTER_EN_BIT0 bis BIT_REGISTER_EN_BIT7 welche mit dem aus dem Signal InAB, siehe **Abbildung 5-7** Seite 5-6, erkannten Wert des Paritätsbit im Register BIT_REGISTER_EN_BIT8 verglichen wird.

Bei der geraden Paritätsprüfung wird am Bit9 eine logische 1 ergänzt wenn die Summe der logischen 1 in den Bits1 bis 8 ungerade ist. Es wird eine logische 0 ergänzt wenn die Summe ungerade ist.

Die Paritätsprüfung erfolgt mittels XOR-Verknüpfung der einzelnen Bit:

$$\text{Parität} = ((\text{Bit1 XOR Bit2}) \text{ XOR } (\text{Bit3 XOR Bit4})) \text{ XOR } ((\text{Bit5 XOR Bit6}) \text{ XOR } (\text{Bit7 XOR Bit8}))$$

Diese über die Bit1 bis Bit 8 errechnete Parität wird dann mit dem Paritätsbit (Bit9), siehe **Abbildung 3-8** Seite 3-7 und **Abbildung 5-9** Seite 5-10, verglichen. Stimmen beide überein ist der Ausgang PARITY_OK logisch 1 (Parität ok) ansonsten logisch 0 (Parität fehlerhaft), siehe 8.4 Variablendefinition Ausgänge Modul BIT_REGISTER Seite 8-2.

5.4.3 Modul TELEGRAM_CHECK

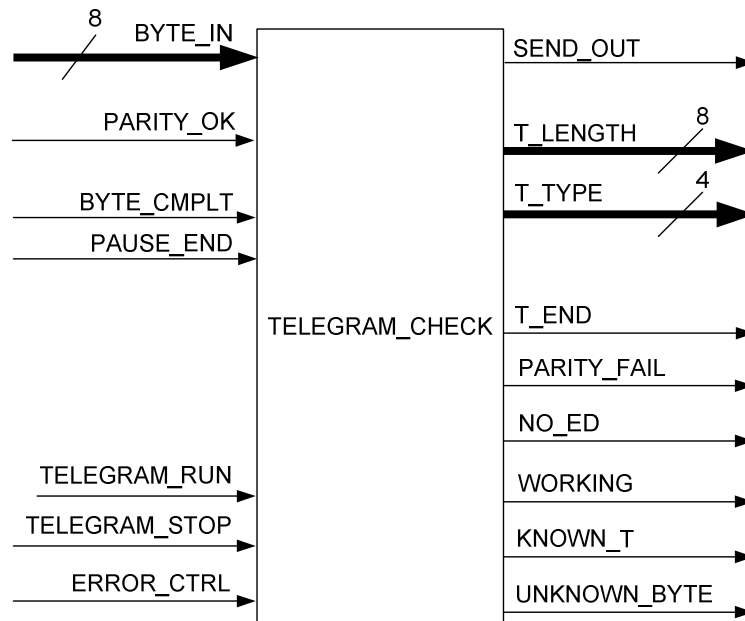


Abbildung 5-14: Wirkungsplan Modul TELEGRAM_CHECK

Das Modul TELEGRAM_CHECK ist für die Erkennung der Telegramme in der Bytefolge BYTE_OUT, siehe **Abbildung 5-6** Seite 5-5, in **Abbildung 5-14** dargestellt als BYTE_IN, verantwortlich.

Es benutzt dazu die eingehenden Signale der Ausgänge PARITY_OK des Moduls BIT_REGISTER, siehe 5.4.2 Modul BIT_REGISTER Seite 5-13, und BYTE_CMPLT sowie PAUSE_END des Moduls InAB_INPUT, siehe 5.4.1 Modul InAB_INPUT Seite 5-6.

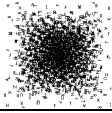
Eine Übersicht ist im Wirkungsplan Verarbeitungseinheit / Ablaufsteuerung der **Abbildung 5-7** Seite 5-6 gegeben.

Die im Modul TELEGRAM_CHECK enthaltene Ablaufsteuerung 2, siehe 5.4 Verarbeitungseinheit / Ablaufsteuerung Seite 5-4, wird durch die Steuersignale TELEGRAM_RUN, TELEGRAM_STOP und ERROR_CTRL der Steuerung, siehe 5.2 Telegrammaufzeichnung Seite 5-2, gesteuert.

Die Ausgänge des Moduls TELEGRAM_CHECK geben die Steuerzeichen, 5.4 Verarbeitungseinheit / Ablaufsteuerung Seite 5-4ff, für die Bytefolge BYTE_OUT, siehe 5.4.2 Modul BIT_REGISTER Seite 5-13, aus.

Für die Variablendefinition der Ausgänge siehe 8.2

Variablendefinition Steuerzeichen Verarbeitungseinheit Seite 8-2.



Die **Abbildung 5-15** Seite 5-22 zeigt die Verhaltensbeschreibung des Moduls TELEGRAM_CHEK als Programmablaufgraph (PAG) für:

- Start der Telegrammerkennung
- Erkennung Telegramm mit SD1
- Erkennung Telegramm mit SD2
- Erkennung Telegramm mit SD3
- Erkennung Telegramm mit SD4
- Erkennung Telegramm mit SC
- Behandlung TELEGRAM_STOP 1
- Behandlung TELEGRAM_STOP 2
- Fehlerbehandlung Parität falsch
- Fehlerbehandlung kein ED

Start der Telegrammerkennung:

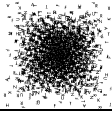
Der Start der Telegrammerkennung beginnt im Initialzustand ST_TC_00.

- Setzen aller Ausgänge auf logisch 0 und des Zählers COUNT auf null.
- Abfrage des Steuersignals TELEGRAM_RUN auf logisch 0 oder 1
- Bei einer logischen 0 wird der Start der Telegrammerkennung wiederholt.
- Bei einer logischen 1 wird das Signal PAUSE_END vom Modul InAB_INPUT auf logisch 0 oder 1 abgefragt.
- Bei einer logischen 0 wird der Start der Telegrammerkennung wiederholt.
- Bei einer logischen 1 wird am Ausgang WORKING bis zum Ende von Erkennung Telegramm mit SC eine logische 1 ausgegeben um die laufende Telegrammerkennung anzuzeigen.
- Wechsel in den Zustand ST_TC_01
- Abfrage des Signal BYTE_CMPLT vom Modul InAB_INPUT auf logisch 0 oder 1.
- Bei einer logischen 0 wird im Zustand ST_TC_01 verharrt und das Signal BYTE_CMPLT erneut abgefragt.
- Bei einer logischen 1 wird das Signal PARITY_OK vom Modul BIT_REGISTER auf logisch 0 oder 1 abgefragt.
- Bei einer logischen 0 die ein Fehler in der Parität erkannt worden was zur Fehlerbehandlung Parität falsch führt.
- Bei einer logischen 1 ist die Parität korrekt und es folgt die Erkennung Telegramm mit SD1.

Erkennung Telegramm mit SD1:

Die Parität des ersten Bytes ist korrekt wurde in Start der Telegrammerkennung festgestellt.

- Abfrage des Eingangs BYTE_IN (Ausgang BYTE_OUT vom Modul BIT_REGISTER) auf Übereinstimmung mit dem den hexadezimalen Wert 10_h. (Start Delimiter des Telegramms SD1, siehe 3.5.1 SD1: Telegramm ohne Daten Seite 3-5.)
- Wenn keine Übereinstimmung mit 10_h folgt die Erkennung Telegramm mit SD2.
- Wenn Übereinstimmung mit 10_h Zähler COUNT (aktueller Wert 0) um 1 erhöhen.
- Ausgang SEND_OUT wird logisch 1 (vorher 0)
- Ausgang T_LENGTH gibt den aktuellen Wert des Zählers COUNT aus.
- Ausgang T_TYPE gibt den binären Wert 0001_b für das Telegramm mit SD1 aus bis zum Ende Erkennung Telegramm mit SD1.

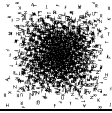


- Wechsel in den Zustand ST_TC_02.
- Abfrage des Signals BYTE_CMPLT vom Modul InAB_INPUT auf logisch 0 oder 1.
- Bei einer logischen 0 wird im Zustand ST_TC_02 verharrt und das Signal BYTE_CMPLT erneut abgefragt, SEND_OUT logisch 0.
- Bei einer logischen 1 wird das Signal PARITY_OK vom Modul BIT_REGISTER auf logisch 0 oder 1 abgefragt.
- Bei einer logischen 0 die ein Fehler in der Parität erkannt worden was zur Fehlerbehandlung Parität falsch führt.
- Bei einer logischen 1 ist die Parität korrekt und es wird der aktuelle Wert des Zählers COUNT abgefragt.
- Wenn der aktuelle Wert des Zählers COUNT ungleich 5 (Maximale Gesamtlänge des Telegramm SD1, siehe 3.5.1 SD1: Telegramm ohne Daten Seite 3-5.) ist wird im Zustand ST_TC_02 verharrt bis der aktuelle Wert des Zählers COUNT gleich 5 ist.
- Wenn der aktuelle Wert des Zählers COUNT gleich 5 ist wird der Eingang BYTE_IN auf Übereinstimmung mit dem den hexadezimalen Wert 16_h geprüft. (End Delimiter (ED) des Telegramms SD1, siehe 3.5.1 SD1: Telegramm ohne Daten Seite 3-5.)
- Wenn keine Übereinstimmung mit 16_h des aktuellen Wert des Zählers COUNT folgt die Fehlerbehandlung kein ED.
- Wenn Übereinstimmung mit 16_h des aktuellen Wert des Zählers COUNT wurde ein End Delimiter (ED) erkannt und der Ausgang SEND_OUT ist weiterhin logisch 1.
- Ausgang T_LENGTH gibt den aktuellen Wert des Zählers COUNT (aktueller Wert 5) aus.
- Ausgang T_END wird logisch 1 um das Ende des Telegramms anzuzeigen und folgt Behandlung TELEGRAM_STOP 2.

Erkennung Telegramm mit SD2:

Die Parität des ersten Bytes ist korrekt wurde in Start der Telegrammerkennung festgestellt.

- Abfrage des Eingangs BYTE_IN (Ausgang BYTE_OUT vom Modul BIT_REGISTER) auf Übereinstimmung mit dem den hexadezimalen Wert 68_h (Start Delimiter des Telegramms SD2, siehe 3.5.2 SD2: Telegramm mit Daten variabler Länge Seite 3-5.)
- Wenn keine Übereinstimmung mit 68_h folgt die Erkennung Telegramm mit SD3.
- Wenn Übereinstimmung mit 68_h Zähler COUNT (aktueller Wert 0) um 1 erhöhen.
- Ausgang SEND_OUT wird logisch 1 (vorher 0)
- Ausgang T_LENGTH gibt den aktuellen Wert des Zählers COUNT aus.
- Ausgang T_TYPE gibt den binären Wert 0010_b für das Telegramm mit SD2 aus bis zum Ende Erkennung Telegramm mit SD2.
- Wechsel in den Zustand ST_TC_03.
- Abfrage des Signals BYTE_CMPLT vom Modul InAB_INPUT auf logisch 0 oder 1.
- Bei einer logischen 0 wird im Zustand ST_TC_03 verharrt und das Signal BYTE_CMPLT erneut abgefragt, SEND_OUT logisch 0.
- Bei einer logischen 1 wird das Signal PARITY_OK vom Modul BIT_REGISTER auf logisch 0 oder 1 abgefragt.
- Bei einer logischen 0 die ein Fehler in der Parität erkannt worden was zur Fehlerbehandlung Parität falsch führt.

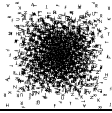


- Bei einer logischen 1 ist die Parität korrekt und es wird der Eingang BYTE_IN auf Übereinstimmung mit dem den hexadezimalen Wert 16_h geprüft. (End Delimiter des Telegramms SD1, siehe 3.5.2 SD2: Telegramm mit Daten variabler Länge Seite 3-5.)
- Wenn keine Übereinstimmung mit 16_h wird der aktuelle Wert des Zählers COUNT abgefragt.
- Wenn der aktuelle Wert des Zählers COUNT ungleich 254 (Maximale Gesamtlänge des Telegramm SD2, siehe 3.5.2 SD2: Telegramm mit Daten variabler Länge Seite 3-5.) ist wird im Zustand ST_TC_03 verharrt bis der aktuelle Wert des Zählers COUNT gleich 254 ist oder der aktuelle Wert des Zählers COUNT mit 16_h übereinstimmt.
- Wenn der aktuelle Wert des Zählers COUNT gleich 254 wurde kein End Delimiter (ED) erkannt und es folgt die Fehlerbehandlung kein ED.
- Wenn Übereinstimmung mit 16_h des aktuellen Wert des Zählers COUNT wurde ein End Delimiter erkannt und der Ausgang SEND_OUT ist weiterhin logisch 1.
- Ausgang T_LENGTH gibt den aktuellen Wert des Zählers COUNT (Aktueller Wert abhängig von den Daten im Telegramm, siehe 3.5.2 SD2: Telegramm mit Daten variabler Länge Seite 3-5.) aus.
- Ausgang T_END wird logisch 1 um das Ende des Telegramms anzuzeigen und folgt Behandlung TELEGRAM_STOP 2.

Erkennung Telegramm mit SD3:

Die Parität des ersten Bytes ist korrekt wurde in Start der Telegrammerkennung festgestellt.

- Abfrage des Eingangs BYTE_IN (Ausgang BYTE_OUT vom Modul BIT_REGISTER) auf Übereinstimmung mit dem den hexadezimalen Wert A2_h. (Start Delimiter des Telegramms SD3, siehe 3.5.3 SD3: Telegramm mit Daten fester Länge Seite 3-5.)
- Wenn keine Übereinstimmung mit A2_h folgt die Erkennung Telegramm mit SD4.
- Wenn Übereinstimmung mit A2_h Zähler COUNT (aktueller Wert 0) um 1 erhöhen.
- Ausgang SEND_OUT wird logisch 1 (vorher 0)
- Ausgang T_LENGTH gibt den aktuellen Wert des Zählers COUNT aus.
- Ausgang T_TYPE gibt den binären Wert 0011_b für das Telegramm mit SD3 aus bis zum Ende Erkennung Telegramm mit SD3.
- Wechsel in den Zustand ST_TC_04.
- Abfrage des Signals BYTE_CMPLT vom Modul InAB_INPUT auf logisch 0 oder 1.
- Bei einer logischen 0 wird im Zustand ST_TC_04 verharrt und das Signal BYTE_CMPLT erneut abgefragt, SEND_OUT logisch 0.
- Bei einer logischen 1 wird das Signal PARITY_OK vom Modul BIT_REGISTER auf logisch 0 oder 1 abgefragt.
- Bei einer logischen 0 die ein Fehler in der Parität erkannt worden was zur Fehlerbehandlung Parität falsch führt.
- Bei einer logischen 1 ist die Parität korrekt und es wird der aktuelle Wert des Zählers COUNT abgefragt.
- Wenn der aktuelle Wert des Zählers COUNT ungleich 13 (Maximale Gesamtlänge des Telegramm SD3, siehe 3.5.3 SD3: Telegramm mit Daten fester Länge Seite 3-5.) ist wird im Zustand ST_TC_04 verharrt bis der aktuelle Wert des Zählers COUNT gleich 13 ist.
- Wenn der aktuelle Wert des Zählers COUNT gleich 13 ist wird der Eingang BYTE_IN auf Übereinstimmung mit dem den hexadezimalen Wert 16_h geprüft. (End



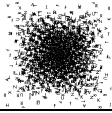
Delimiter (ED) des Telegramms SD3, siehe 3.5.3 SD3: Telegramm mit Daten fester Länge Seite 3-5.)

- Wenn keine Übereinstimmung mit 16_h des aktuellen Wert des Zählers COUNT folgt die Fehlerbehandlung kein ED.
- Wenn Übereinstimmung mit 16_h des aktuellen Wert des Zählers COUNT wurde ein End Delimiter erkannt und der Ausgang SEND_OUT ist weiterhin logisch 1.
- Ausgang T_LENGTH gibt den aktuellen Wert des Zählers COUNT (aktueller Wert 13) aus.
- Ausgang T_END wird logisch 1 um das Ende des Telegramms anzuzeigen und folgt Behandlung TELEGRAM_STOP 2.

Erkennung Telegramm mit SD4:

Die Parität des ersten Bytes ist korrekt wurde in Start der Telegrammerkennung festgestellt.

- Abfrage des Eingangs BYTE_IN (Ausgang BYTE_OUT vom Modul BIT_REGISTER) auf Übereinstimmung mit dem den hexadezimalen Wert DC_h . (Start Delimiter des Telegramms SD4, siehe 3.5.4 SD4: Token-Telegramm Seite 3-6.)
- Wenn keine Übereinstimmung mit DC_h folgt die Erkennung Telegramm mit SC.
- Wenn Übereinstimmung mit DC_h Zähler COUNT (aktueller Wert 0) um 1 erhöhen.
- Ausgang SEND_OUT wird logisch 1 (vorher 0)
- Ausgang T_LENGTH gibt den aktuellen Wert des Zählers COUNT aus.
- Ausgang T_TYPE gibt den binären Wert 0100_b für das Telegramm mit SD4 aus bis zum Ende Erkennung Telegramm mit SD4.
- Wechsel in den Zustand ST_TC_05.
- Abfrage des Signals BYTE_CMPLT vom Modul InAB_INPUT auf logisch 0 oder 1.
- Bei einer logischen 0 wird im Zustand ST_TC_05 verharrt und das Signal BYTE_CMPLT erneut abgefragt, SEND_OUT logisch 0.
- Bei einer logischen 1 wird das Signal PARITY_OK vom Modul BIT_REGISTER auf logisch 0 oder 1 abgefragt.
- Bei einer logischen 0 die ein Fehler in der Parität erkannt worden was zur Fehlerbehandlung Parität falsch führt.
- Bei einer logischen 1 ist die Parität korrekt und es wird der aktuelle Wert des Zählers COUNT abgefragt.
- Wenn der aktuelle Wert des Zählers COUNT ungleich 2 (Maximale Gesamtlänge des Telegramm SD4, siehe 3.5.4 SD4: Token-Telegramm Seite 3-6.) ist wird im Zustand ST_TC_05 verharrt bis der aktuelle Wert des Zählers COUNT gleich 2 ist.
- Wenn der aktuelle Wert des Zählers COUNT gleich 2 ist bleibt der Ausgang SEND_OUT ist weiterhin logisch 1 (Kein End Delimiter (ED) im Telegramm mit SD4, siehe 3.5.4 SD4: Token-Telegramm Seite 3-6.)
- Ausgang T_LENGTH gibt den aktuellen Wert des Zählers COUNT (aktueller Wert 13) aus.
- Ausgang T_END wird logisch 1 um das Ende des Telegramms anzuzeigen und folgt Behandlung TELEGRAM_STOP 2.



Erkennung Telegramm mit SC:

Die Parität des ersten Bytes ist korrekt wurde in Start der Telegrammerkennung festgestellt.

- Abfrage des Eingangs BYTE_IN (Ausgang BYTE_OUT vom Modul BIT_REGISTER) auf Übereinstimmung mit dem den hexadezimalen Wert E5_h. (Start Delimiter des Telegramms SD4, siehe 3.5.4 SD4: Token-Telegramm Seite 3-6.)
- Wenn keine Übereinstimmung mit E5_h folgt die Behandlung TELEGRAM_STOP 1.
- Wenn Übereinstimmung mit E5_h Zähler COUNT (aktueller Wert 0) um 1 erhöhen.
- Ausgang SEND_OUT wird logisch 1 (vorher 0)
- Ausgang T_LENGTH gibt den aktuellen Wert des Zählers COUNT aus.
- Ausgang T_TYPE gibt den binären Wert 1000_b für das Telegramm mit SC aus bis zum Ende Erkennung Telegramm mit SC.
- Ausgang T_END wird logisch 1 um das Ende des Telegramms anzuzeigen und folgt Behandlung TELEGRAM_STOP 2.

Behandlung TELEGRAM_STOP 1:

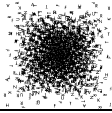
Die Parität des ersten Bytes ist korrekt wurde in Start der Telegrammerkennung festgestellt aber es wurde kein Start Delimiter (SD) eines bekannten Telegramms gefunden.

- Das Signal der Steuerung TELEGRAM_STOP wird auf logisch 0 oder 1 abgefragt.
- Bei einer logischen 0 wird folgt Start der Telegrammerkennung.
- Bei einer logischen 1 wird Ausgang UNKNOWN_BYTE auf logisch 1 gesetzt und alle Ausgänge auf logisch 0.
- Wechsel in den Zustand ST_TC_06.
- Abfrage des Steuersignals TELEGRAM_STOPT auf logisch 0 oder 1.
- Bei einer logischen 0 wird folgt Start der Telegrammerkennung.
- Bei einer logischen 1 wird im Zustand ST_TC_06 verharrt und das Steuersignal TELEGRAM_STOP erneut abgefragt, UNKNOWN_BYTE weiterhin auf logisch 1.

Behandlung TELEGRAM_STOP 2:

Es wurde ein bekanntes Telegramm, mit der richtigen Länge und Ende erkannt.

- Wechsel in den Zustand ST_TC_07.
- Das Signal der Steuerung TELEGRAM_STOP wird auf logisch 0 oder 1 abgefragt.
- Bei einer logischen 0 wird folgt Start der Telegrammerkennung.
- Bei einer logischen 1 wird im Zustand ST_TC_07 verharrt und das Steuersignal TELEGRAM_STOP erneut abgefragt, der Ausgang KNOWN_T wird auf logisch 1 gesetzt.



Fehlerbehandlung Parität falsch:

Die Prüfung der Parität eines Bytes hat einen Fehler erkannt.

- Der Ausgang PARITY_FAIL wird auf logisch 1 gesetzt und alle Ausgänge auf logisch 0.
- Wechsel in den Zustand ST_TC_08.
- Das Signal der Steuerung ERROR_CTRL wird auf logisch 0 oder 1 abgefragt.
- Bei einer logischen 0 wird im Zustand ST_TC_08 verharret, der Ausgang PARITY_FAIL bleibt auf logisch 1 gesetzt und alle Ausgänge auf logisch 0.
- Bei einer logischen 1 wird im Zustand ST_TC_09 gewechselt und der Fehler gilt als bestätigt.
- Das Signal der Steuerung ERROR_CTRL wird auf logisch 0 oder 1 abgefragt.
- Bei einer logischen 0 wird im Zustand ST_TC_09 verharret, der Ausgang PARITY_FAIL bleibt auf logisch 1 gesetzt und alle Ausgänge auf logisch 0.
- Bei einer logischen 1 gilt der Fehler als behoben und es folgt Start der Telegrammerkennung.

Fehlerbehandlung kein ED:

Die Prüfung des End Delimiter (ED) eines Bytes hat kein ED im aktuellen Telegramm festgestellt.

- Der Ausgang NO_ED wird auf logisch 1 gesetzt und alle Ausgänge auf logisch 0.
- Wechsel in den Zustand ST_TC_10.
- Das Signal der Steuerung ERROR_CTRL wird auf logisch 0 oder 1 abgefragt.
- Bei einer logischen 0 wird im Zustand ST_TC_10 verharret, der Ausgang NO_ED bleibt auf logisch 1 gesetzt und alle Ausgänge auf logisch 0.
- Bei einer logischen 1 wird im Zustand ST_TC_11 gewechselt und der Fehler gilt als bestätigt.
- Das Signal der Steuerung ERROR_CTRL wird auf logisch 0 oder 1 abgefragt.
- Bei einer logischen 0 wird im Zustand ST_TC_11 verharret, der Ausgang NO_ED bleibt auf logisch 1 gesetzt und alle Ausgänge auf logisch 0.
- Bei einer logischen 1 gilt der Fehler als behoben und es folgt Start der Telegrammerkennung

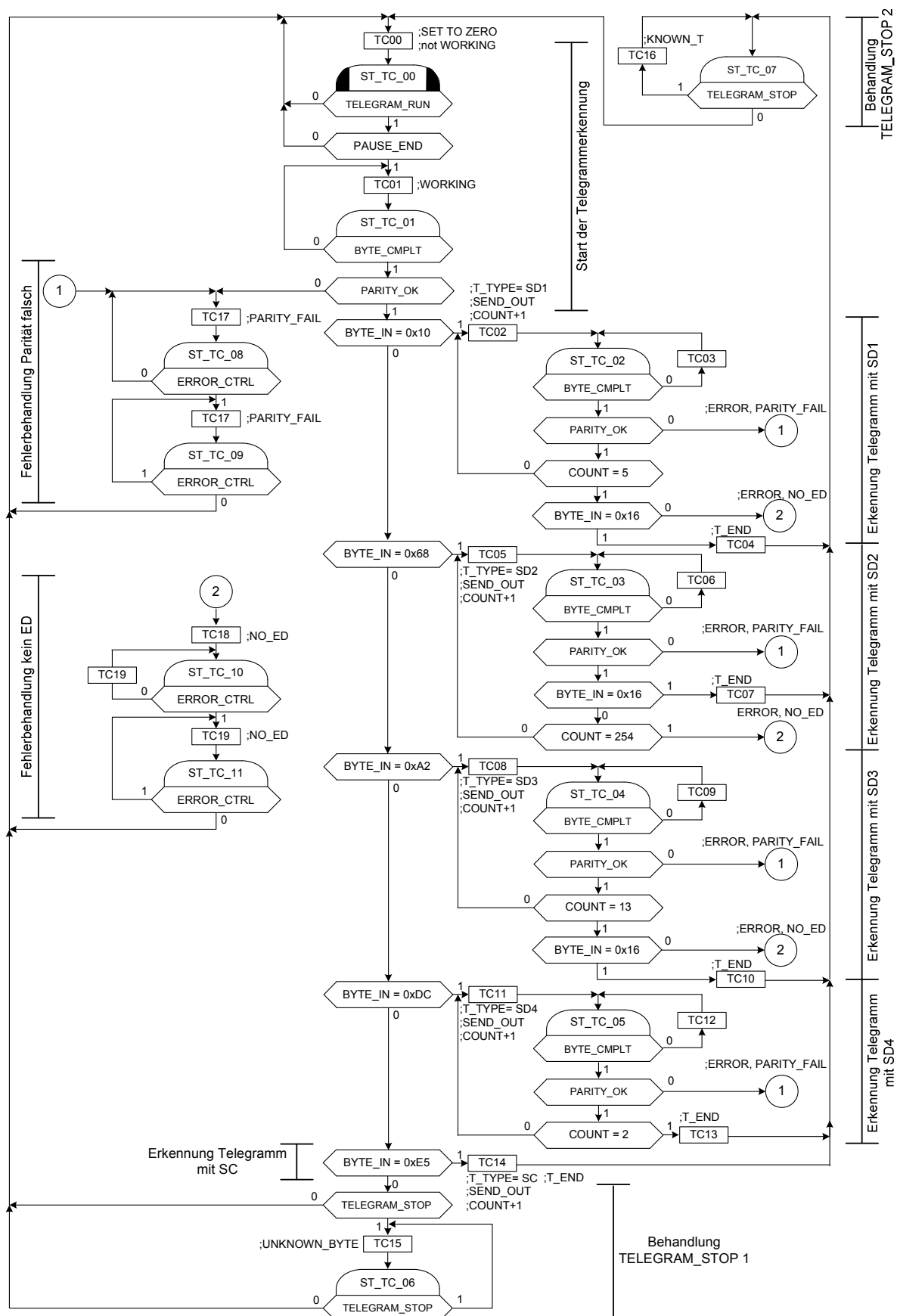
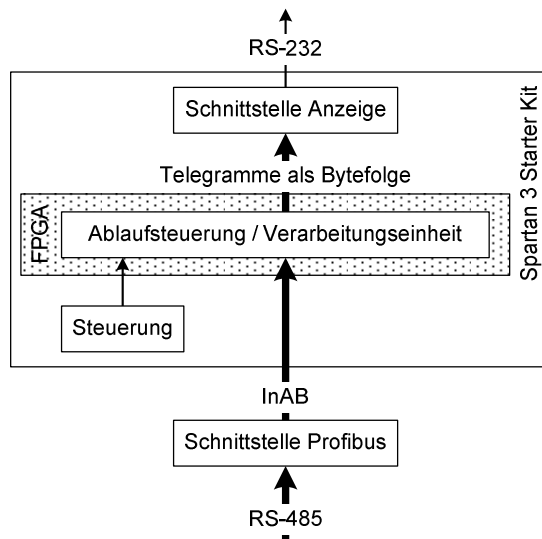


Abbildung 5-15: Verhaltensbeschreibung als PAG Modul TELEGRAM_CHECK

5.5 Schnittstelle Anzeige

Um von der Telegramme als Bytefolge, siehe 5.4 Verarbeitungseinheit / Ablaufsteuerung Seite 5-4, zur Anzeige senden zu können wird die Schnittstelle Anzeige benötigt, siehe 5.2 Telegrammaufzeichnung Seite 5-2.



Siehe Abbildung 5-4: Zuordnung der Bestandteile der Telegrammaufzeichnung Seite 5-3

Die Anzeige, die die Telegramme via RS-232 empfängt, siehe **Abbildung 5-2** Seite 5-1. Sie wird mit der RS-232 Schnittstelle des Spartan 3 Starter Kit, siehe **Abbildung 4-3** Seite 4-3, verbunden. Die Schnittstelle Anzeige muss also die Telegramme als Bytefolge an die RS-232 Schnittstelle übergeben.

Die Erzeugung der RS-232 Signalpegel erledigt die RS-232 Schnittstelle des Spartan 3 Starter Kit, siehe /GUIDE/ Kapitel 7.

RS-232 verwendet ebenfalls die UART-Zeichencodierung, siehe 3.8 Zeichenkodierung Seite 3-7, allerdings ohne Paritätsbit.

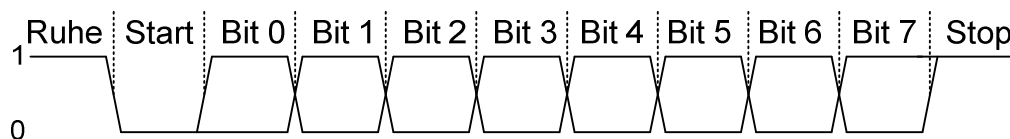


Abbildung 5-16: Impulsdigramm UART-Codierung ohne Paritätsbit

Um die Telegramme als Bytefolge in die UART-Zeichencodierung umzuwandeln wird das Modul RS232_TX verwendet.

5.5.1 Modul RS232_TX

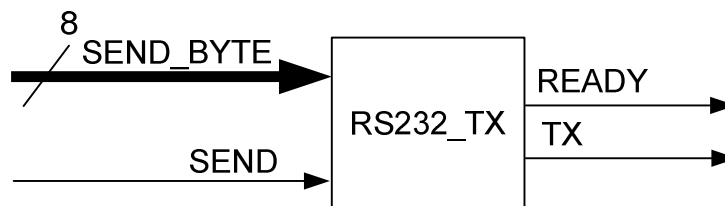


Abbildung 5-17: Wirkungsplan Modul RS232_TX

Das Modul RS232_TX erhält die Telegramme als Bytefolge des Signal BYTE_OUT vom Modul BIT_REGISTER, siehe 5.4.2 Modul BIT_REGISTER Seite 5-13, in **Abbildung 5-17** als SEND_BYTE dargestellt.

Das Signal SEND_OUT vom Modul TELEGRAM_CHECK, siehe 5.4.3 Modul TELEGRAM_CHECK Seite 5-15, in **Abbildung 5-17** als SEND dargestellt, wird genutzt um zu signalisieren das ein Byte geprüft (Byte vollständig und Parität korrekt) und einem Telegramm zugeordnet wurde. Es ist somit bereit zum senden.

Der Ausgang READY des Modul RS232_TX dient zur Anzeige wenn nichts via RS-232 gesendet (logisch 1) wird und das Modul RS232_TX somit bereit zum senden ist.

Der Ausgang TX des Modul RS232_TX übergibt die Telegramme als Bytefolge an die RS-232 Schnittstelle des Spartan 3 Starter Kit, siehe **Abbildung 4-3** Seite 4-3.

Für die Variablendefinition der Ausgänge siehe 8.5

Variablendefinition Ausgänge Modul RS232_TX Siehe 8-3.

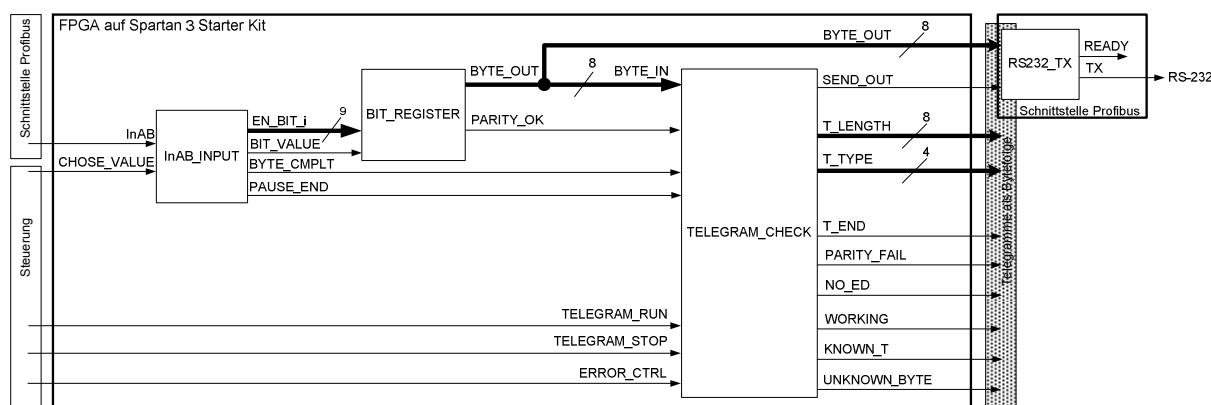
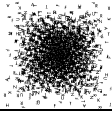


Abbildung 5-18: Einordnung Modul RS232_TX mit Wirkungsplan Verarbeitungseinheit / Ablaufsteuerung

Die **Abbildung 5-19** Seite 5-27 zeigt Beispielhaft eine Bitfolge des Ausgangs TX des Moduls RS232_TX. Mit Hilfe eines Zählers wird bestimmt, wann welches Teilstück der Bitfolge über den Ausgang TX an die RS-232 Schnittstelle des Spartan 3 Starter Kit gesendet wird.



Die **Abbildung 5-20** Seite 5-1 zeigt die Verhaltensbeschreibung des Moduls RS232_TX als Programmablaufgraph (PAG) für:

- Idle 1
- Startbit
- Bit0 bis Bit 7
- Stoppbit
- Idle 2

Idle 1:

Der Ruhezustand ist logisch 1, siehe **Abbildung 5-16** Seite 5-23.

- Ausgang TX des Modul RS232_TX wird logisch 1 um den Ruhezustand zu signalisieren.
- Ausgang READY des Modul RS232_TX ist logisch 1 um Bereit zum Senden zu signalisieren
- Wechsel in den Initialzustand ST_TX_00.
- Abfrage des Eingangs SEND (Ausgang SEND_OUT vom Modul TELEGRAM_CHECK) auf logisch 0 oder 1.
- Bei einer logischen 0 wird im Initialzustand ST_TX_00 verharrt und der Eingang SEND erneut abgefragt, TX und READY logisch 1.
- Bei einer logischen 1 folgt Startbit.

Startbit:

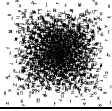
Am Eingang SEND wurde logisch 1 empfangen. Die Übertragung beginnt mit dem Startbit.

- Ausgang TX des Modul RS232_TX wird logisch 0, das Startbit wird gesendet.
- Ausgang READY des Modul RS232_TX wird logisch 0 bis zum Ende des Stoppbits, Modul RS232_TX sendet.
- Der Zähler COUNT wird auf null gesetzt (t_0).
- Wechsel in den Zustand ST_TX_01.
- Abfrage des Zähler COUNT ob das das Ende des Startbits (t_1) erreicht wurde (COUNT = CNT01).
- Wurde das Ende des Startbits (t_1) nicht erreicht wird im Zustand ST_TX_01 verharrt, Zähler COUNT wird um 1 erhöht, Ausgang TX weiterhin logisch 0.
- Wurde das Ende des Startbits (t_1) erreicht folgt Bit 0 bis Bit 7.

Bit0 bis Bit7:

Das Ende des Startbits (t_1) wurde erreicht und ist der Beginn der Übertragung von Bit0 bis Bit 7.

- Ausgang TX des Modul RS232_TX sendet nun den logischen Wert des Eingangs SEND_BYTE, Ausgang READY weiterhin logisch 0.
- Wechsel zum jeweiligen nächsten Zustand (ST_TX_02 bis ST_TX_09).
- Abfrage des Zähler COUNT ob das das Ende des jeweiligen Bit (t_2 bis t_9) erreicht wurde (COUNT = CNT02 bis COUNT = CNT09).
- Wurde das Ende des jeweiligen Bit (t_2 bis t_9) nicht erreicht wird im Zustand (ST_TX_02 bis ST_TX_09) verharrt, Zähler COUNT wird um 1 erhöht, Ausgang TX sendet weiterhin den logischen Wert des Eingangs SEND_BYTE.
- Wurde das Ende des jeweiligen Bit (t_2 bis t_9) erreicht folgt das Stoppbit.



Stoppbit:

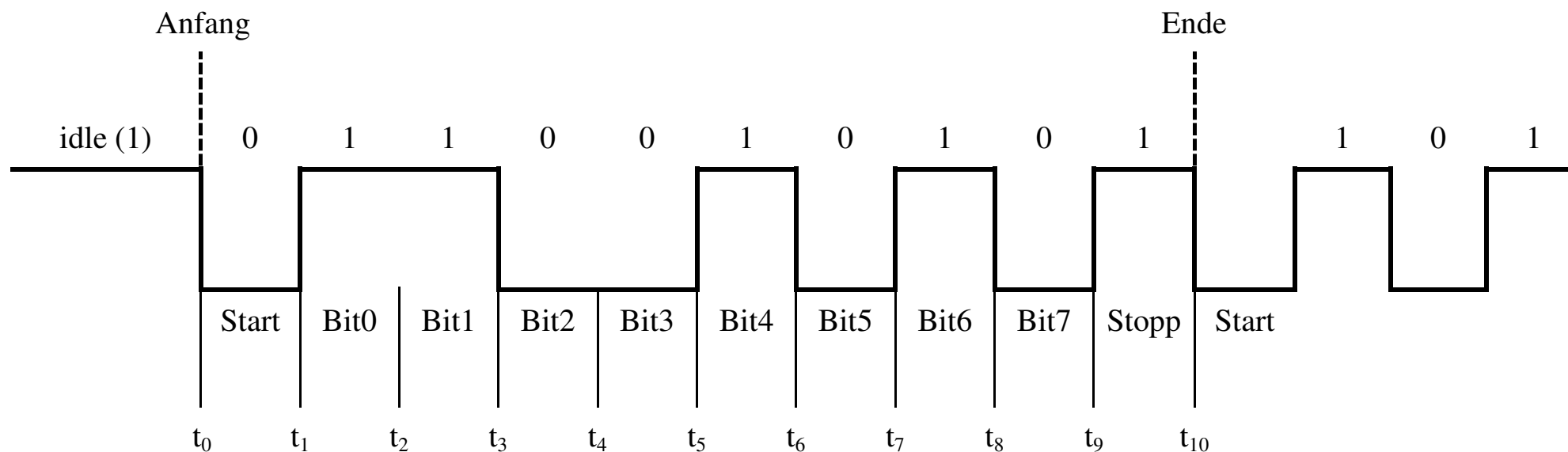
Das Ende des Bit7 (t_9) wurde erreicht und ist der Beginn der Übertragung des Stoppbits.

- Ausgang TX des Modul RS232_TX wird logisch 1, das Stoppbit wird gesendet.
- Ausgang READY des Modul RS232_TX weiterhin logisch 0.
- Wechsel zum Zustand ST_TX_10.
- Abfrage des Zähler COUNT ob das das Ende des Stoppbits (t_{10}) erreicht wurde (COUNT = CNT10).
- Wurde das Ende des Stoppbits (t_{10}) nicht erreicht wird im Zustand ST_TX_10 verharret, Zähler COUNT wird um 1 erhöht, Ausgang TX weiterhin logisch 1.
- Wurde das Ende des Stoppbits (t_{10}) erreicht folgt Idle 2.

Idle 2:

Das Ende des Stoppbits (t_{10}) wurde erreicht, der Ruhezustand ist logisch 1, siehe **Abbildung 5-16** Seite 5-23.

- Ausgang TX des Modul RS232_TX ist weiterhin logisch 1 um den Ruhezustand zu signalisieren.
- Ausgang READY des Modul RS232_TX wird logisch 1 um Bereit zum Senden zu signalisieren
- Wechsel in den Zustand ST_TX_11.
- Abfrage des Eingangs SEND (Ausgang SEND_OUT vom Modul TELEGRAM_CHECK) auf logisch 0 oder 1.
- Bei einer logischen 0 folgt Idle 1, weitere Bits können gesendet werden.
- Bei einer logischen 1 wird im Zustand ST_TX_11 verharret und der Eingang SEND erneut abgefragt.



Der Zähler COUNT läuft von t_0 bis t_{10} . Zählerwerte t_0 bis t_{10} , siehe **Tabelle 5-3** Seite 5-28.

Berechnung der Zählerwerte der Zählerzeitpunkte:

Der Systemtakt des Spartan 3 Starter Kit ist 50.000.000 Hz (50 MHz), siehe /GUIDE/ Kapitel 8. Die Baudrate wurde festgelegt auf 19200 Bit/s.

- Berechnung der Übertragungsdauer von n Bits (t_{BIT}):
 - $t_{\text{BIT}} = n \text{ Bit} / \text{Baudrate}$
- Berechnung der Systemtakte (k) für die Übertragung von n Bits (t_{BIT}):
 - $t_{\text{BIT}} = k / \text{Systemtakt}$
 - $k = t_{\text{BIT}} * \text{Systemtakt}$
 - $k = (n \text{ Bit} / \text{Baudrate}) * \text{Systemtakt}$



Zählerzeitpunkt	Berechnung (n Bit / Baudrate) * Systemtakt	Zählerwert dezimal	Zählerwert hexadezimal	Variablenname
t ₀		0	0	÷
t ₁	(1 / 19200) * 50.000.000	2604	0A2C	CNT01
t ₂	(2 / 19200) * 50.000.000	5208	1458	CNT02
t ₃	(3 / 19200) * 50.000.000	7812	1E84	CNT03
t ₄	(4 / 19200) * 50.000.000	10416	28B0	CNT04
t ₅	(5 / 19200) * 50.000.000	13020	32DC	CNT05
t ₆	(6 / 19200) * 50.000.000	15625	3D09	CNT06
t ₇	(7 / 19200) * 50.000.000	18229	4735	CNT07
t ₈	(8 / 19200) * 50.000.000	20833	5161	CNT08
t ₉	(9 / 19200) * 50.000.000	23437	5B8D	CNT09
t ₁₀	(10 / 19200) * 50.000.000	26041	65B9	CNT10

Tabelle 5-3: Zählerwerte Modul RS232_TX

Die Ergebnisse der Berechnung der „Zählerwerte dezimal“ wurden abgerundet.

Der Variablenname wird in der Verhaltensbeschreibung als Programmablaufgraph (PAG) des Moduls RS232_TX genutzt, siehe **Abbildung 5-20** Seite 5-1.

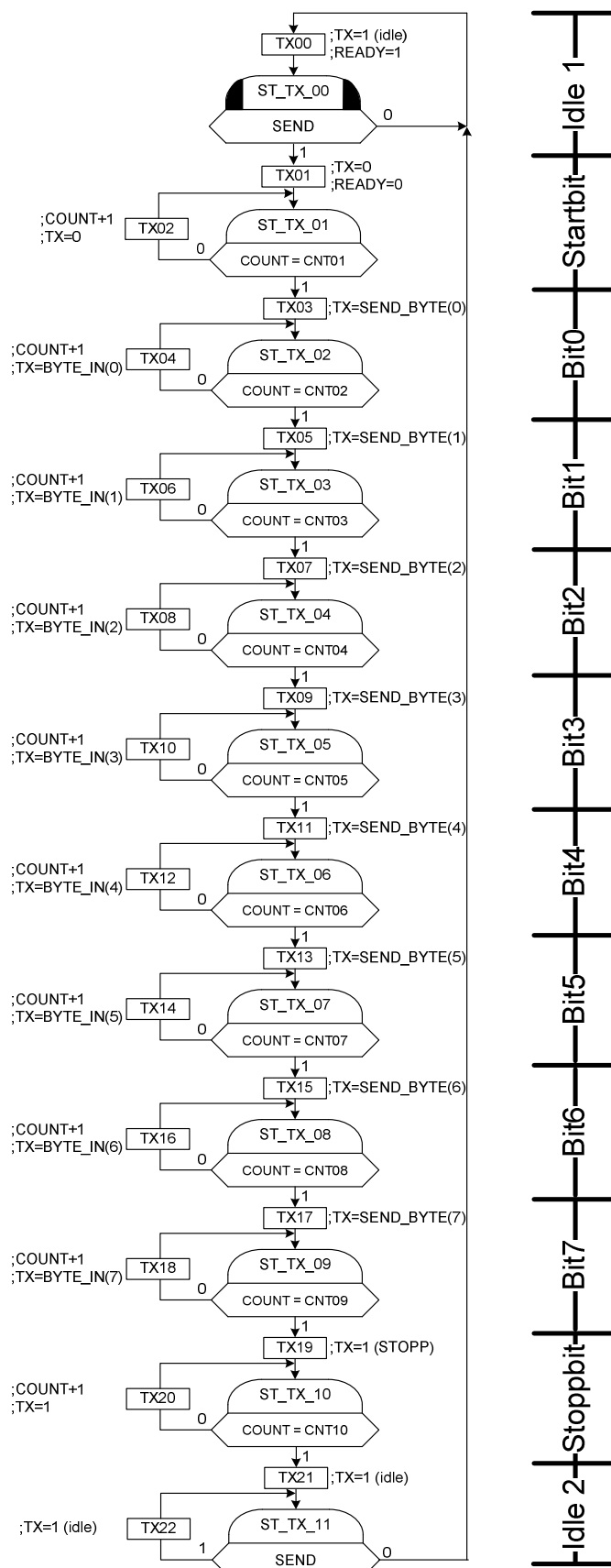
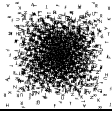


Abbildung 5-20: Verhaltensbeschreibung als PAG Modul RS232_TX



6 Testaufbau

Der Testaufbau ist einfach gehalten. Verwendet wird ein:

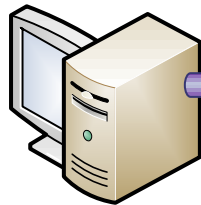
- Profibusmaster
- Profibuskabel
- Profibusslave

Dabei entsteht ein einfaches aber voll funktionsfähiges Profibusnetzwerk. Welche Geräte genau als Master und Slave verwendet werden ist nicht relevant.

Für den Testaufbau wurde ein Rechner mit der Profibusmasterkarte „FC3101“ von Beckhoff ausgestattet, siehe /PCMASTER/, und als Profibus Master konfiguriert.

Als Profibusslave dient ein Schneider Electric „Modicon TSX 170 BDM 344 01“, siehe /SLAVE/.

Profibus Master



Profibuskabel

Profibus Slave

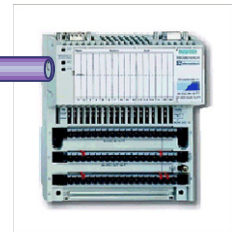


Abbildung 6-1: Aufbau einfaches Profibusnetzwerk
/PROFI_SLAVE/

Der Profibusmonitor wird nun an das Profibusnetzwerk angeschlossen.

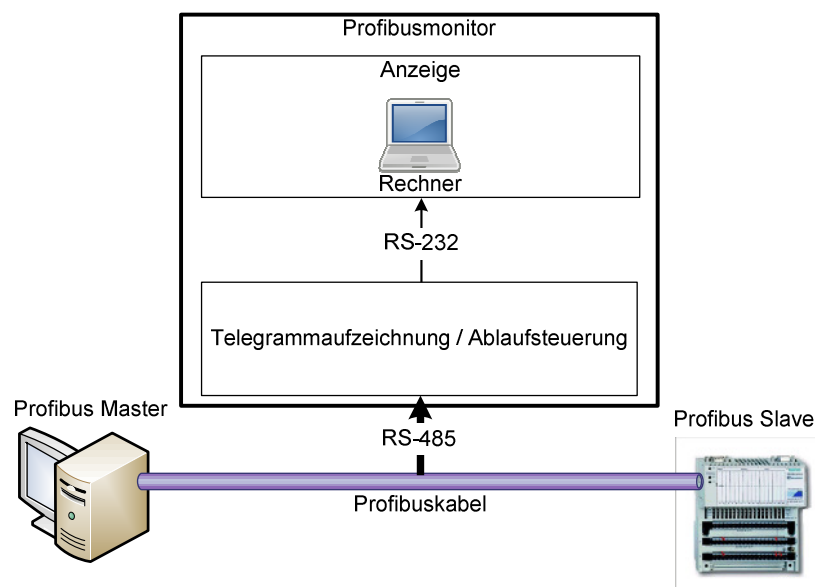


Abbildung 6-2: Testaufbau Profibusmonitor
/PROFI_SLAVE/



Verbindung Profibus mit Schnittstelle Profibus:

Die Verbindung des Profibus mit Schnittstelle Profibus, siehe 5.3 Schnittstelle Profibus Seite 5-3, erfolgt über das Profibuskabel mit einem D-SUB DE9 Stecker, siehe *Abbildung 3-13* Seite 3-9. Die Schnittstelle Profibus verfügt dazu über eine D-SUB DE9 Buchse, siehe 3.10 Kabel und Stecker Seite 3-9.

Verbindung Schnittstelle Profibus mit Spartan 3 Starter Kit:

Die Verbindung der Schnittstelle Profibus, siehe 5.3 Schnittstelle Profibus Seite 5-3, mit dem Spartan 3 Starter Kit, siehe *Abbildung 4-3* Seite 4-3, erfolgt über ein VGA-Kabel mit einem D-SUB DE15 Stecker, siehe /WIKI_DSUB15/. Das Spartan 3 Starter Kit und die Schnittstelle Profibus verfügen dazu über eine D-SUB DE15 Buchse, siehe /WIKI_DSUB15/.

Verbindung Spartan 3 Starter Kit mit Anzeige (Rechner):

Die Verbindung des Spartan 3 Starter Kit, siehe *Abbildung 4-3* Seite 4-3, mit dem Rechner der Anzeige, siehe *Abbildung 5-2* Seite 5-1, erfolgt über ein RS-232 Kabel mit D-SUB DE9 Stecker für die Seite des Spartan 3 Starter Bit und D-SUB DE9 Buchse für den Rechner.

Konfiguration Profibus Master:

Auf dem Rechner des Profibus Master muss eine Steuerungssoftware laufen. Es wurde TwinCAT 2 von Beckhoff Automation GmbH genutzt, siehe /TWINCAT/.

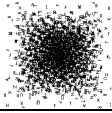
Der Profibusmaster wird so konfiguriert das er die Adresse 1 hat und die Geschwindigkeit des Profibusnetzwerks auf 9600Bit/s regelt.

Konfiguration Profibus Slave:

Der Profibus Slave wurde auf die Adresse 8 eingestellt.

Terminal Programm zum Anzeigen der Telegramme auf dem Rechner:

Als Terminalprogramm wurde HTERM, siehe /HTERM/, verwendet. Die Einstellung um Daten via RS-232 zu empfangen sind 19200 Baud, 8 Datenbits, keine Parität, 1 Stoppbit.



6.1 Schnittstelle Profibus Aufbau

Stromversorgung: Siehe *Abbildung 6-3* Seite 6-5.

- Rechte Seite, oben, Bezeichnung „Buchse 4,5 -9V“, Hohlsteckerbuchse zum Anschluss der Versorgungsspannung des DC/DC-Wandlers (IN) und damit die Speisung des Stromkreises 2, hier mit 2P5 (5V) und 2M (Masse) bezeichnet. Die Kondensatoren C8 (0,1 μ F) und C9 (100 μ F) sind als Abblockkondensatoren für die Stabilisierung der Spannung und Reduzierung daraus entstehender Störungen notwendig.
- Der DC/DC-Wandler, Bezeichnung „TEL 3-0511“, welche die Speisung des Stromkreises 2 übernimmt hat am Ausgang (OUT) ebenfalls Abblockkondensatoren erhalten. Die Eingangsspannung des TEL 3-0511 von Traco Power, siehe /TRACO/, kann zwischen 4,5V und 9V sein. Die Ausgangsspannung liegt dann bei 5V. Da der TEL 3-0511 bis zu 600mA Ausgangsstrom liefert, aber nur einen Wirkungsgrad von 70% besitzt sollte die Stromversorgung mindestens 857mA konstant liefern können.
- Rechte Seite, mittig, Bezeichnung „USB“, USB-Buchse Typ B zur Speisung des Stromkreises 1 mit 5V, hier mit der Bezeichnung P5 (5V) und M (Masse). Auch hier werden die Kondensatoren C1 (0,1 μ F) und C2 (100 μ F) als Abblockkondensatoren eingesetzt. LED2 dient als Anzeige für das korrekte Funktionieren der Stromversorgung.

Anschlüsse: Siehe *Abbildung 6-3* Seite 6-5.

- Linke Seite, Bezeichnung „Profibus“, D-Sub9 Buchse zum Anschluss eines Profibussteckers. Auf Pin 3, Bezeichnung „X1-3“ liegt die Leitung B des Profibusteckers. Die Leitung A ist am Pin 8 „X1-8“ angeschlossen. Beide Leitungen führen zum BUS Transceiver, Bezeichnung „75ALS176P“. Die Bezeichnung der Eingänge des „75ALS176P“ mit A und B hat nichts mit den Bezeichnungen der Leitungen A und B des Profibus zu tun, siehe *Abbildung 3-12* Seite 3-8. Pin 5 „X1-5“ und 6 „X1-6“ sind für die Speisung der Abschlusswiderstände des Profibussteckers zuständig und an den Stromkreis 2 angeschlossen.
- Rechte Seite, unten, Bezeichnung „VGA (Spartan 3 Kit)“ ist ein D-Sub15 VGA Anschluss für das VGA-Kabel zum Spartan 3 Starter Kit. Es werden nur die Pins 2 „X2-1“ und 11 „X2-11“ verwendet. Pin 1 leitet das verstärkte Profibuseingangssignal vom Optokoppler, Bezeichnung „HCPL7721“, an das Spartan 3 Starter Kit weiter. Pin 11 verbindet die Masse des Stromkreises 1 mit GND des Spartan 3 Starter Kit, siehe /GUIDE/, Seite 21.

BUS Transceiver 75ALS176P: Siehe *Abbildung 6-3* Seite 6-5.

Der 75ALS176P, siehe /TRANSCIVER/ wandelt die eingehenden Profibussignale der Leitungen A und B in ein binäres Signal um. Die Leitung B ist dazu am Pin 6 und die Leitung A am Pin 7 angeschlossen. Das binäre Signal wird am Pin 1 ausgegeben. Pin 2 muss für die korrekte Funktion des Pin 1 auf Masse gesetzt sein, andernfalls ist Pin 1 hochohmig geschaltet. Pin 3 und 4 sind ebenfalls an Masse gelegt. Auf dem Schaltplan ist nicht dargestellt ist die Stromversorgung über den Stromkreis 2, Pin 8 kommt an 2P5 und Pin 5 an 2M. Dazwischen wieder ein Abblockkondensator mit 0,1 μ F.



Optokoppler HCPL7721: Siehe *Abbildung 6-3* Seite 6-5.

Der Optokoppler spielt bei der Potentialtrennung eine wichtige Rolle da hier das Signal via LED per Licht und somit potentialfrei übertragen wird. Vom Pin 1 des BUS Transceivers empfängt der HCPL7721 das Profibuseingangssignal am Pin 2. Diese Seite des Optokopplers wird vom Stromkreis 2, Pin 1 an 2P5 und Pin 4 an 2M, versorgt. Pin 3 ist die Anode der internen LED und darf nicht angeschlossen werden, siehe /OPTOKOPPLER/. Der Kondensator C6 (0,1µF) zwischen Pin 1 und Pin 4 muss verwendet werden, genauso wie C7 (0,1µF) zwischen Pin 8 und Pin 5, die am Stromkreis 1 angeschlossen sind. Pin 8 geht auf P5, Pin 5 an M des Stromkreises 1. Vom Pin 6 geht dann das Profibuseingangssignal in die Verstärkerschaltung. Der Pin 7 des Optokopplers ist intern im Optokoppler nicht verbunden.

Verstärkerschaltung 74HC125N: Siehe *Abbildung 6-3* Seite 6-5.

Die Verstärkerschaltung dient zur Verstärkung des Profibuseingangssignals, welches vom Pin 6 des Optokopplers kommt. Im 74HC125N gibt es insgesamt vier Verstärkerschaltungen, von denen nur die ersten beiden genutzt und im Schaltplan getrennt dargestellt sind, Bezeichnung „IC2A“ und „IC2B“. „IC2B“ übernimmt die Verstärkung des am Eingang, Pin 5, angelegten Profibuseingangssignal. Dieses Signal kommt verstärkt am Pin 6 an und ist mit dem Pin 1 „X2-1“ des VGA-Anschlusses verbunden. Das Profibuseingangssignal geht aber auch in die Verstärkerschaltung „IC2A“, Pin 2, deren Ausgang, Pin 3, wiederum an eine LED angeschlossen ist. Dadurch kann optisch kontrolliert werden ob das Signal vom Profibus korrekt ankommt. Bei beiden genutzten Verstärkerschaltungen müssen die Pins 1 und 4 auf Masse liegen, da die Pins 3 und 6 sonst hochohmig geschaltet werden.

Schirmung: Siehe *Abbildung 6-3* Seite 6-5.

Um etwaige Störungen von Außen abzuhalten sollte das Hardwareinterface mit einem Schirm versehen werden. Der Schirm ist über ein RC-Glied C5 (4,7nF) und R1 (1MΩ) mit der Masse M verbunden. Die Schirmung ist dabei zusätzlich mit der des VGA-Buchse und bei Anschluss eines VGA-Kabels mit dessen Masse verbunden. Die Masse der D-Sub9 Buchse für den Anschluss des Profibussteckers ist davon ausgenommen, da sich Störungen auf der Masse sonst über das Hardwareinterface hinweg ausbreiten würden. Als Schirm sollten zwei metallbeschichtete Leiterplatten, auf und unter dem Hardwareinterface angebracht werden.

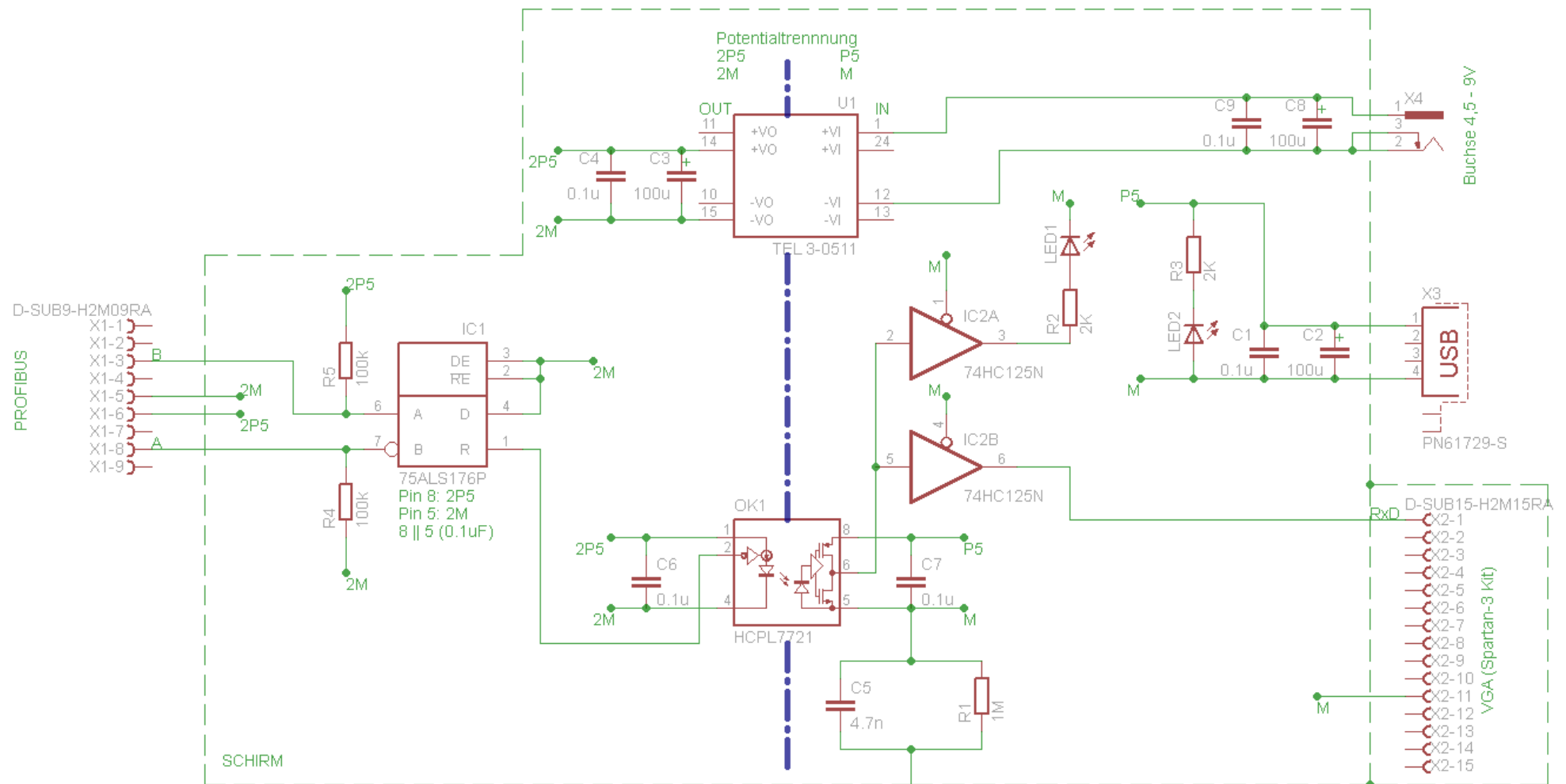
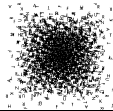


Abbildung 6-3: Schaltplan Profibus Interface

6.2 Stromversorgung

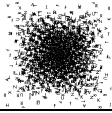
Die Stromversorgung des Spartan 3 Starter Kits und der Schnittstelle Profibus erfolgt über eine vom Netz getrennte Stromversorgung mittels sechs Batterien. Es hat sich während der Entwicklung des Profibusmonitors und des Hardwareinterfaces gezeigt, dass die zur Verfügung stehenden Netzteile Störungen verursachen. Deren Ursache konnte nicht geklärt werden. Es wurde daher entschieden, eine eigene Stromversorgung zu entwickeln.

Sechs Akkus à 1,2V liefern eine Gesamtspannung von 7,2V. Diese Spannung wird direkt über einen Hohlstecker an die Hochsteckerbuchse der Schnittstelle Profibus weitergeleitet und dort für den DC/DC-Wandler genutzt. Auf der Stromversorgung selbst wird damit ebenfalls ein DC/DC-Wandler vom gleichen Typ wie auf der Schnittstelle Profibus, TEL 3-0511, versorgt. Dieser speist zwei USB-Buchsen Typ A. An diesen kann die Schnittstelle Profibus mit einem USB-Kabel angeschlossen werden. USB-Buchsen liefern eine Ausgangsspannung von 5V bei einem Strom von 600mA.

→ Der Aufbau der Stromversorgung ist nicht Bestandteil dieser Arbeit. Sie wurde von einem Mitstudenten an der FH Frankfurt Herr Phillip Brocar realisiert. Vielen Dank.



Abbildung 6-4: Stromversorgung



7 Zusammenfassung und Ausblick

Der Profibusmonitor kann bisher nur die vom Profibus empfangene Bytes via RS-232 an einen Rechner zur Ausgabe weiterleiten.

Eine Erweiterung könnte daher ein Speicher zur Speicherung der empfangen Telegramme auf dem Spartan 3 Starter Kit sein. Diese Telegramme könnte man wiederum von Rechner über ein Programm via RS-232 auslesen und anzeigen. Die Telegramme könnten in dem Programm in einer sehr viel übersichtlichere und besser verständlichen Form dargestellt werden.

Auch eine Anzeige der Telegramme im Speicher direkt vom Spartan 3 Starter Kit auf einer 7-Segmentanzeige oder einem Monitor könnte realisiert werden.

Die Steuerung dafür könnte dabei wahlweise mit den Schaltern und Tastern des Spartan 3 Starter Kit erfolgen, externen Tastern und Schaltern oder über eine angeschlossene Tastatur oder Maus.

Durch eine Erweiterung des Spartan 3 Starter Kit mit zusätzlichen Schnittstellen (WLAN, Bluetooth, USB, Ethernet) wären weitere Arten der Anzeige, Steuerung und Weitergabe der Telegramme möglich.

Funktionell könnte der Profibusmonitor um weitere Funktionen wie Signalanalyse, Telegrammanalyse, Netzwerkmanagement und Überwachung erweitert werden.

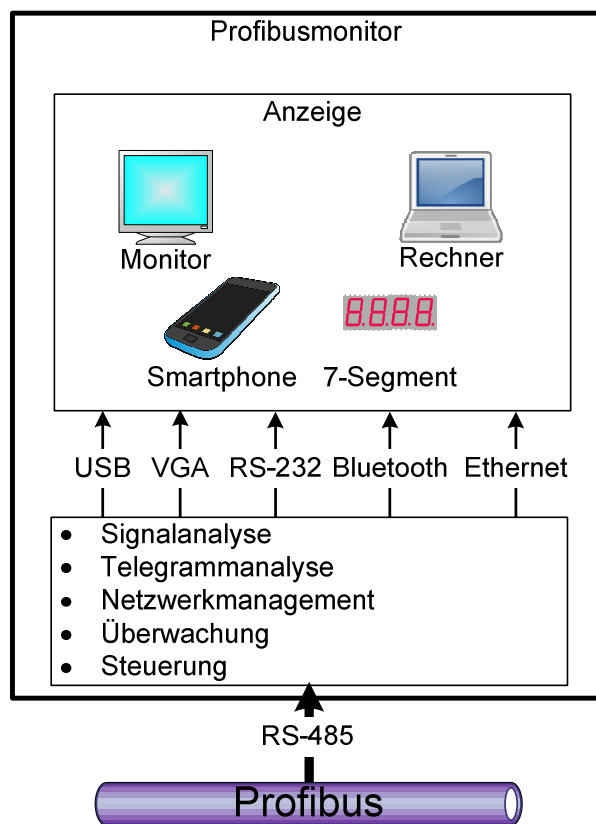


Abbildung 7-1: Konzept Profibusmonitor mit zusätzlichen Erweiterungen
/7SEG/ /LAPTOP/



8 Anhang

8.1 Variablendefinition Steuersignale Steuerung

Variablenname	Datentyp	Variablentyp	Informationen / Anweisungen
CHOSE_VALUE	BOOL	Eingang	1: Zählerwerte, Schrittbetrieb; 0: normale Zählerwerte
TELEGRAM_RUN	BOOL	Eingang	1: nächstes Telegramm
TELEGRAM_STOP	BOOL	Eingang	1: Stopp nach einem Telegramm
ERROR_CTRL	BOOL	Eingang	1: nach Fehler fortfahren

8.2 Variablendefinition Steuerzeichen Verarbeitungseinheit

Variablenname	Datentyp	Variablentyp	Informationen / Anweisungen
SEND_OUT	BOOL	Ausgang	1: aktuelles Byte senden
T_LENGTH	VECTOR, 8 Bit	Ausgang	Telegrammlänge, binär
T_TYPE	VECTOR, 4 Bit	Ausgang	Telegrammtyp (0000: kein Telegramm erkannt, 0001: Telegrammtyp SD1 0010: Telegrammtyp SD2 0011: Telegrammtyp SD3 0100: Telegrammtyp SD4 1000: Telegrammtyp SC)
T_END	BOOL	Ausgang	1: Ende des aktuellen Telegramms
PARITY_FAIL	BOOL	Ausgang	1: Paritätsfehler
NO_ED	BOOL	Ausgang	1: Kein Enddelimitier festgestellt
WORKING	BOOL	Ausgang	1: Modul TELEGRAM_CHECK arbeitet
KNOWN_T	BOOL	Ausgang	1: Telegramm erkannt
UNKNOWN_BYTE	BOOL	Ausgang	1: Byte nicht erkannt

8.3 Variablendefinition der Ausgänge Modul InAB_INPUT

Variablenname	Datentyp	Variablentyp	Informationen / Anweisungen
EN_BIT_i	VECTOR, 9 Bit	Ausgang	Einschalten Bit 0-8
BIT_VALUE	BOOL	Ausgang	1: Bitwert ist 1
BYTE_CMPLT	BOOL	Ausgang	1: Byte komplett empfangen
PAUSE_END	BOOL	Ausgang	1: SYN von 33 Bit beendet

8.4 Variablendefinition Ausgänge Modul BIT_REGISTER

Variablenname	Datentyp	Variablentyp	Informationen / Anweisungen
BYTE_OUT	VECTOR, 8 Bit	Ausgang	Ausgabe Datenbyte



Variablenname	Datentyp	Variablentyp	Informationen / Anweisungen
PARITY_OK	BOOL	Ausgang	1: Parität ist korrekt

8.5 Variablendefinition Ausgänge Modul RS232_TX

Variablenname	Datentyp	Variablentyp	Informationen / Anweisungen
TX	BOOL	Ausgang	1: Bit ist 1, 0: Bit ist 0
READY	BOOL	Ausgang	1: bereit zum Senden

8.6 Umsetzung der Module in VHDL

Die Umsetzung in VHDL der Module:

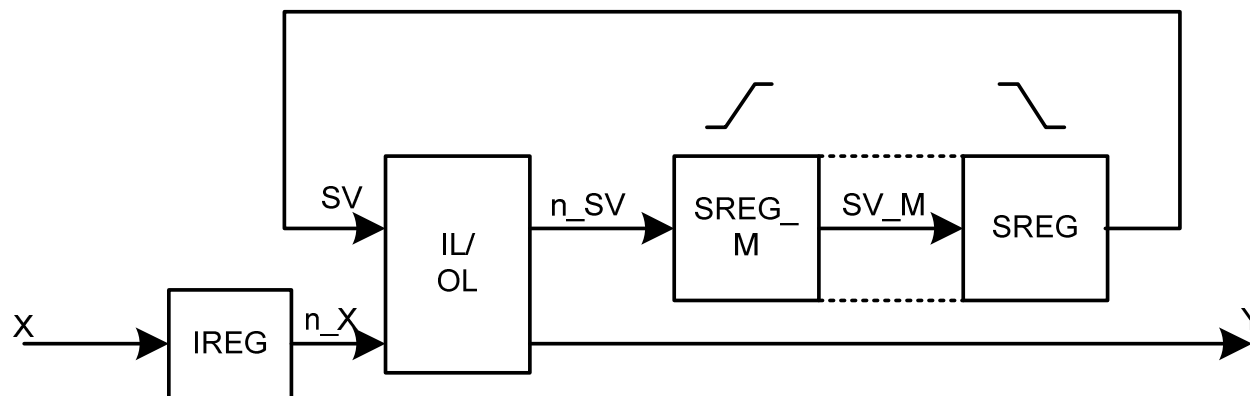
- InAB_INPUT, siehe 5.4.1 Modul InAB_INPUT Seite 5-6
- BIT_REGISTER, siehe 5.4.2 Modul BIT_REGISTER Seite 5-13
- TELEGRAM_CHECK, siehe 5.4.3 Modul TELEGRAM_CHECK Seite 5-15
- RS232_TX, siehe 5.5.1 Modul RS232_TX Seite 5-24

befindet sich im Ordner „VHDL_Bausteine“.

Es wurde der Huffmanautomat verwendet, siehe 8.7 Verwendetes Automatenmodell Seite 8-3.

8.7 Verwendetes Automatenmodell

/PFEIFFER/



IL / OL Input / Output Logic
Eingangs- / Ausgangslogik
I REG Input Register
Eingangsregister
S REG_M State Register Master
Zustandsregister Master
S REG State Register
Zustandsregister (Slave)



Informationsverarbeitung erfolgt mit der 0/1-Flanke
der Taktvariablen



Informationsverarbeitung erfolgt mit der 1/0-Flanke
der Taktvariablen

X Eingangsvariablen
n_X Eingangsvariablen, zwischengespeichert
SV State Variables
Zustandsvariablen
n_SV State Variables
Zustandsvariablen
neuer Wert
SV_M State Variables
Zustandsvariablen
Master
Y Ausgangsvariablen

Abbildung 8-1: Huffmanautomat, Master-/ Slavesteuerung, ohne Ausgangsregister



8.8 Belegung der Ausgangsvariablen

Die Belegung der Ausgangsvariablen vom PAG der Module:

- InAB_INPUT, siehe 5.4.1 Modul InAB_INPUT Seite 5-6
- TELEGRAM_CHECK, siehe 5.4.3 Modul TELEGRAM_CHECK Seite 5-15
- RS232_TX, siehe 5.5.1 Modul RS232_TX Seite 5-24

Befindet sich im Ordner „Belegung_der_Ausgangsvariablen“.

8.9 Literatur / Web-Seiten

/74HC125/	SN54HC125, SN74HC125 Quadruple Bus Buffer Gates With 3-State Outputs Texas Instruments Google.de Suchbegriff: 74HC125 http://www.ti.com/product/sn74hc125 (abgerufen am 16.05.2013)
/FELSER_BUCH/	Prof. Max Felser PROFIBUS Handbuch – Eine Sammlung von Erläuterungen zu PROFIBUS Netzwerken Version vom: 17.08.2009 Berner Fachhochschule (BFH) http://www.see-solutions.de/sonstiges/PROFIBUS%20Handbuch%208_2009.pdf (abgerufen am 14.05.2013)
/FELSER_WEB/	Prof. Max Felser PROFIBUS Handbuch – Eine Sammlung von Erläuterungen zu PROFIBUS Netzwerken Ausgabe 1.1.5 vom Thursday, December 29, 2011 Berner Fachhochschule (BFH) Onlineversion http://www.profibus.felser.ch/ (abgerufen am 17.04.2013)
/GUIDE/	Spartan 3 Starter Kit User Guide http://www.digilentinc.com Suchbegriff Spartan 3 http://www.digilentinc.com/Data/Products/S3BOARD/S3BOARD_RM.pdf (abgerufen am 13.05.2013)
/HTERM/	Webseite des Terminalprogramms Hterm http://www.der-hammer.info/terminal/ (abgerufen am 14.05.2013)
/IEC 61158-1/	International Electrotechnical Commission IEC/TR 61158-1 TECHNICAL REPORT Industrial communication networks – Fieldbus specifications – Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series Edition 3.0 2010-08 ISBN 978-2-88912-137-3 http://www.iec.ch (abgerufen am 31.07.2013)
/IEC 61158-2/	International Electrotechnical Commission IEC 61158-2 INTERNATIONAL STANDARD Industrial communication networks – Fieldbus specifications – Part 2: Physical layer specification and service definition Edition 5.0 2010-07 ISBN 978-2-88912-805-1 http://www.iec.ch (abgerufen am 31.07.2013)
/IEC 61158-3-3/	International Electrotechnical Commission IEC 61158-3-3 INTERNATIONAL STANDARD Industrial communication networks – Fieldbus specifications – Part 3-3: Data-link layer service definition – Type 3 elements Edition 1.0 2007-12 ISBN 2-8318-9412-3 http://www.iec.ch (abgerufen am 31.07.2013)
/IEC 61158-4-3/	International Electrotechnical Commission IEC 61158-4-3 INTERNATIONAL STANDARD Industrial communication networks – Fieldbus specifications – Part 4-3: Data-link layer protocol specification – Type 3 elements Edition 2.0 2010-08 ISBN 978-2-83220-127-5 http://www.iec.ch (abgerufen am 31.07.2013)



/IEC 61158-5-3/	International Electrotechnical Commission IEC 61158-5-3 INTERNATIONAL STANDARD Industrial communication networks – Fieldbus specifications – Part 5-3: Application layer service definition – Type 3 elements Edition 2.0 2010-08 ISBN 978-2-88912-106-9 http://www.iec.ch (abgerufen am 31.07.2013)
/IEC 61158-6-3/	International Electrotechnical Commission IEC 61158-6-3 INTERNATIONAL STANDARD Industrial communication networks – Fieldbus specifications – Part 6-3: Application layer protocol specification – Type 3 elements Edition 2.0 2010-08 ISBN 978-2-83220-128-2 http://www.iec.ch (abgerufen am 31.07.2013)
/IEC 61784-1/	International Electrotechnical Commission IEC 61784-1 INTERNATIONAL STANDARD Industrial communication networks – Profiles – Part 1: Fieldbus profiles Edition 3.0 2010-07 ISBN 978-2-88912-807-5 http://www.iec.ch (abgerufen am 31.07.2013)
/IEC 61784-3-3/	International Electrotechnical Commission IEC 61784-3-3 INTERNATIONAL STANDARD Industrial communication networks – Profiles – Part 3-3: Functional safety fieldbuses – Additional specifications for CPF 3 Edition 2.0 2010-06 ISBN 978-2-88910-978-4 http://www.iec.ch (abgerufen am 31.07.2013)
/IEC 61784-5-3/	International Electrotechnical Commission IEC 61784-5-3 INTERNATIONAL STANDARD Industrial communication networks – Profiles – Part 5-3: Installation of fieldbuses – Installation profiles for CPF 3 Edition 2.0 2010-07 ISBN 978-2-88912-056-7 http://www.iec.ch (abgerufen am 31.07.2013)
/ISO/IEC 1177/	International Organization for Standardization International Electrotechnical Commission ISO/IEC 1177 auch bezeichnet als DIN ISO 1177 oder ISO 1177 INTERNATIONAL STANDARD Data communication; representation of characters for serial data transmission Edition 1.0 1989-11 http://www.beuth.de Suchbegriff: ISO 1177 (abgerufen am 05.08.2013)
/MILLER/	Implementierung einer RS232 Schnittstelle in VHDL Lothar Miller, Am Bläsistock 8, 88483 Rot Lothar@Lothar-Miller.de http://www.lothar-miller.de/s9y/categories/42-RS232 (abgerufen am 13.05.2013)
/OPTOKOPPLER/ /	Optokoppler HCPL-7721 http://www.avagotech.com Suchbegriff HCPL-7721 http://www.avagotech.com/pages/en/optocouplers_plastic/plastic_digital_optocoupler/high_speed_cmos_logic_gate/hcpl-7721/ (abgerufen am 22.08.2012)
/PCMASTER/	Webseite Beckhoff Suchbegriff Profibusmasterkarte FC3101 http://www.beckhoff.de/default.asp?pc_cards_switches/fc3101_fc3102.htm (abgerufen am 15.05.2013)
/PFEIFFER/	Prof. Dr. Volker Pfeiffer Fachhochschule Frankfurt am Main Vorlesung Modellbasiertes Entwerfen und Projektieren 8. Zeitliches Verhalten Oktober 2010
/POPP/	Manfred Popp Profibus-DP : Grundlagen, Tipps und Tricks für Anwender Heidelberg : Hüthig, 1998



	ISBN: 3-7785-2676-6
/SLAVE/	PDF-Dokument Modicon TSX 170 BDM 344 01 Google.de Suchbegriff Modicon TSX 170 BDM 344 01 http://www.global-download.schneider-electric.com/85257849002EB8CB/all/4AFFD128C03A97108525787000816F3B/\$File/33000241_k01_000_01.pdf (abgerufen am 15.05.2013)
/SPARTAN3/	Webseite mit Informationen über das Spartan 3 Starter Kit http://www.digilentinc.com/Products/Detail.cfm?NavPath=2,400,799&Prod=S3BOARD (abgerufen am 13.05.2013)
/TIA-485-A/	Telecommunications Industry Association TIA-485-A ELECTRICAL CHARACTERISTICS OF GENERATORS AND RECEIVERS FOR USE IN BALANCED DIGITAL MULTIPOINT SYSTEMS Revision / Edition: A 12.07.2012 http://www.tiaonline.org/ (abgerufen am 31.07.2013)
/TRACO/	Datenblatt des Traco Power TEL 3-0511 http://www.tracopower.com/datasheet_g/tel3-d.pdf (abgerufen am 16.05.2013)
/TRANSCEIVER/	SN65ALS176, SN75ALS176, SN75ALS176A, SN75ALS176B DIFFERENTIAL BUS TRANSCEIVERS Texas Instruments Google.de Suchbegriff: SN65ALS176 http://www.alldatasheet.com/view.jsp?Searchword=75ALS176P (abgerufen am 16.05.2013)
/TWINCAT/	Webseite der Firma Beckhoff Automation GmbH Suchbegriff TwinCAT TwinCAT 2 Software http://www.beckhoff.de/default.asp?twincat/default.htm (abgerufen am 14.05.2013)
/WIKI_DSUB15/	de.Wikipedia.org Suchbegriff: D-Sub 15 http://de.wikipedia.org/wiki/D-SUB_15 (abgerufen am 22.08.2013)
/WIKIP_ROFI/	de.Wikipedia.org Suchbegriff: Profibus http://de.wikipedia.org/wiki/Profibus (abgerufen am 14.05.2013)
/WIRESHARK/	de.Wikipedia.org Suchbegriff Wireshark http://de.wikipedia.org/wiki/Wireshark (abgerufen am 10.05.2013)

8.10 Verzeichnis Bilder

Abbildung 1-1: Konzept Profibusmonitor	1-1
Abbildung 3-1: 3-Schichten Feldbusreferenzmodell	3-3
Abbildung 3-2: Master/Slave in der Bustopologie	3-4
Abbildung 3-3: Aufbau Telegramm ohne Daten	3-5
Abbildung 3-4: Aufbau Telegramm mit Daten variabler Länge	3-5
Abbildung 3-5: Aufbau Telegramm mit Daten fester Länge	3-5
Abbildung 3-6: Aufbau Token-Telegramm	3-6
Abbildung 3-7: Aufbau Kurzquittung	3-6
Abbildung 3-8: Bits der UART-Codierung	3-7
Abbildung 3-9: Impulsdiagramm UART-Codierung	3-7
Abbildung 3-10: Impulsdiagramm der Spannungspegel Leiter B und Leiter A	3-8
Abbildung 3-11: Impulsdiagramm der Spannungsdifferenz zwischen Leiter B und Leiter A	3-8
Abbildung 3-12: Busabschluss am Profibus Kabeltyp A /FELSER_WEB/	3-8
Abbildung 3-13: Pinlayout mit Pflichtbelegung; Frontalansicht Stecker; Rückansicht Buchse	3-9
Abbildung 4-1: Aufbau Profibusmonitor	4-1
Abbildung 4-2: Konzept eigener Profibusmonitor	4-2
Abbildung 4-3: Spartan 3 Starter Kit mit einzelnen Elementen	4-3
Abbildung 5-1: Konzept Anzeige	5-1
Abbildung 5-2: Konzept eigener Profibusmonitor mit Anzeige via RS-232 und Rechner	5-1
Abbildung 5-3: Aufbau Telegrammaufzeichnung	5-2
Abbildung 5-4: Zuordnung der Bestandteile der Telegrammaufzeichnung	5-3
Abbildung 5-5: Grundlegender Aufbau Schnittstelle Profibus	5-4
Abbildung 5-6: Aufbau Verarbeitungseinheit / Ablaufsteuerung	5-5
Abbildung 5-7: Wirkungsplan Verarbeitungseinheit / Ablaufsteuerung	5-6
Abbildung 5-8: Wirkungsplan Modul InAB_INPUT	5-6



Abbildung 5-9: Beispiel Impulsdigramm der Bitfolge InAB mit Zählerzeitpunkten	5-10
Abbildung 5-10: Verhaltensbeschreibung als PAG Modul InAB_INPUT	5-12
Abbildung 5-11: Wirkungsplan Modul BIT_REGISTER	5-13
Abbildung 5-12: Register im Modul BIT_REGISTER	5-13
Abbildung 5-13: Verhaltensbeschreibung als PAG Modul BIT_REGISTER	5-14
Abbildung 5-14: Wirkungsplan Modul TELEGRAM_CHECK	5-15
Abbildung 5-15: Verhaltensbeschreibung als PAG Modul TELEGRAM_CHECK	5-22
Abbildung 5-16: Impulsdigramm UART-Codierung ohne Paritätsbit	5-23
Abbildung 5-17: Wirkungsplan Modul RS232_TX	5-24
Abbildung 5-18: Einordnung Modul RS232_TX mit Wirkungsplan Verarbeitungseinheit / Ablaufsteuerung ...	5-24
Abbildung 5-19: Beispiel Impulsdigramm Modul RS232_TX	5-27
Abbildung 5-20: Verhaltensbeschreibung als PAG Modul RS232_TX	5-1
Abbildung 6-1: Aufbau einfaches Profibusnetzwerk	6-1
Abbildung 6-2: Testaufbau Profibusmonitor	6-1
Abbildung 6-3: Schaltplan Profibus Interface	6-5
Abbildung 6-4: Stromversorgung	6-6
Abbildung 7-1: Konzept Profibusmonitor mit zusätzlichen Erweiterungen	7-1
Abbildung 8-1: Huffmanautomat, Master-/ Slavesteuerung, ohne Ausgangsregister	8-3

8.11 Verzeichnis Tabellen

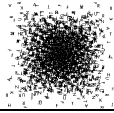
Tabelle 3-1: Aufbau und Inhalt der Norm IEC 61158	3-1
Tabelle 3-2: Aufbau und Inhalt der Norm IEC 61784	3-2
Tabelle 3-3: Werte der Spannungspegel RS-485	3-8
Tabelle 3-4: Datenrate und Kabellänge Kabeltyp Type A /IEC 61158-2/	3-9
Tabelle 3-5: Pinbelegung D-SUB Stecker für Profibus /FELSER_WEB/	3-9
Tabelle 5-1: Signalpegel Bitfolge InAB	5-4
Tabelle 5-2: Zählerwerte Modul InAB_INPUT	5-11
Tabelle 5-3: Zählerwerte Modul RS232_TX	5-28

8.12 Quelle Bilder

/7SEG/	http://www.nemsim.com/ece395blimp/fritzing/parts/svg/core/breadboard/7-segment%20display.svg (abgerufen am 17.04.2013)
/LAPTOP/	http://web.fh-ludwigshafen.de/rz/home.nsf/Files/745C9DE7E8FC59C6C12576D3002ECB19/\$File/200px-Gnome-laptop.svg.png (abgerufen am 17.04.2013)
/PROFI_SLAVE/	http://www.kintercontrol.com/images/product/Modicon%20Momentum.jpg (abgerufen am 17.04.2013)
/SPARTANBOARD/	http://www.digilentinc.com/Data/Products/S3BOARD/S3BOARD-top-400.gif (abgerufen am 17.04.2013)
/PHONE/	http://www.clker.com/cliparts/P/g/N/M/O/w/smartphone-hi.png (abgerufen am 10.05.2013)

8.13 Verzeichnis Abkürzungen

BUS	Binary Unit System
CBA	Component Based Automation
CMPLT	Complete
COUNT	Counter
CPF	Communication Profile Families
CTRL	Control
DA	Destination Address
DC	Direct Current
DP	Dezentrale Peripherie
ED	End Delimiter
FC	Function Code
FCS	Frame Check Sequence



FMS	Fieldbus Message Specification
FPGA	Field Programmable Gate Array
IEC	International Electrotechnical Commission
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
LE	Length
Ler	Length repeated
NRZ	No-Return-to-Zero
OSI	Open Systems Interconnection
PA	Process Automation
PC	Personal Computer
PDU	Protocol Data Unit
Profibus	Process Field Bus
PS/2	Personal System /2
SA	Source Address
SC	Short Confirmation
SD	Start Delimiter
SDR	Send and Request Data
SPS	Speicher Programmierbare Steuerung
SYN	synchronize
TX	Transmit
UART	Universal Asynchronous Receiver
USB	Universal Serial Bus
VGA	Video Graphics Array
VHDL	Very High Speed Integrated Circuit Hardware Description Language