

Modularisierung einer PROFIBUS Anschaltung für Embedded-Control

Prof. Max Felser

Berner Fachhochschule, Hochschule für Technik und Informatik, Kompetenzzentrum für Industrielle Kommunikation, Morgartenstrasse 2c, CH-3014 Bern, max.felser@hta-be.bfh.ch

Unterschiedliche Anwendungen haben unterschiedliche Anforderungen für die Vernetzung von Kleinststeuerungen für die Automatisierung. In diesem Beitrag werden die Anforderungen strukturiert und ein Beispiel einer modularen PROFIBUS-FDL Anschaltung für kostengünstige Steuerungen aufgezeigt.

1 AUSGANGSLAGE

1.1 Moderne Trends

Die Strukturen der Automatisierungstechnik verändern sich. Steuerungen werden immer kleiner, leistungsfähiger und kostengünstiger. Die Rechenleistung wird von zentralen Rechnern vermehrt auf dezentrale Rechner und intelligente Feldgeräte verteilt. Dies stellt neue Aufgaben an die Kommunikation, an die industriellen Netzwerke, die die einzelnen Geräte verbinden sollen. Als industrielle Netzwerke werden heute vor allem spezielle Feldbusse eingesetzt. Vermehrt sollen dabei auch Lösungen auf der Basis von Ethernet eingesetzt werden. Für einen Hersteller von kompakten und kostengünstigen Kleinststeuerungen stellt sich die Frage, welches industrielle Netzwerk, welchen Feldbus er in seinen embedded Steuerungen einsetzen soll. Zuerst wollen wir ein paar Grundlagen zum Aufbau einer solchen Anschaltung zusammenfassen.

1.2 Strukturen einer Anschaltung

Für die verschiedenen Kommunikationsanschlüssen kann die Hardware unterschiedlich strukturiert werden. In Anlehnung an Prof. Bender [1] teilen wir die verschiedenen Möglichkeiten in fünf Klassen ein (Abbildung 1).

In der Klasse 1 und 2 wird für die Kommunikation eine separate Baugruppe realisiert. Dies ist typischerweise bei leistungsfähigen Steuerungen wie

z.B. SPS oder PCs der Fall. Die Feldbusanschlusung ist ein vom Anwendungsprozessor vollständig getrennter Prozessor. Damit wird erreicht, dass die Kommunikation die Leistung der Steuerung nicht beeinträchtigt und stört. Diese Leistungssteigerung rechtfertigt in der Regel auch einen höheren Preis dieser Anschaltung. Die Prozessoren der Kommunikation und der Anwendung werden über eine geeignete Ankopplung zusammengeschaltet, die in der Regel auf einem gemeinsamen Speicher realisiert ist.

Die Klasse 1 unterscheidet sich von der Klasse 2 insofern, dass die eingestzten Prozessoren, die Kommunikation direkt eingebaut haben. Die Leistungsfähigkeit wird vom Umfang der Funktionen im intelligenten Netzwerkkontroller bestimmt. Beim PROFIBUS z.B. ist es mit einem einfachen eingebauten UART möglich Übertragungsgeschwindigkeiten bis zu 500 KBit/s zu erreichen. Für die Übertragungsgeschwindigkeiten 12 MBit/s ist ein intelligenter Netzwerkkontroller (z.B. ein ASIC) zwingend notwendig.

Bei der Klasse 3 und 4 wird ein Prozessor gleichzeitig für die Kommunikation und die Anwendung eingesetzt. Dies hat eine Reduktion der Leistung zur Folge, die einer Anwendung zur Verfügung steht, bringt aber Kosten und Platzvorteile die sich insbesondere bei kompakten Systemen positiv bemerkbar machen können. Klassische Vertreter für diesen An-

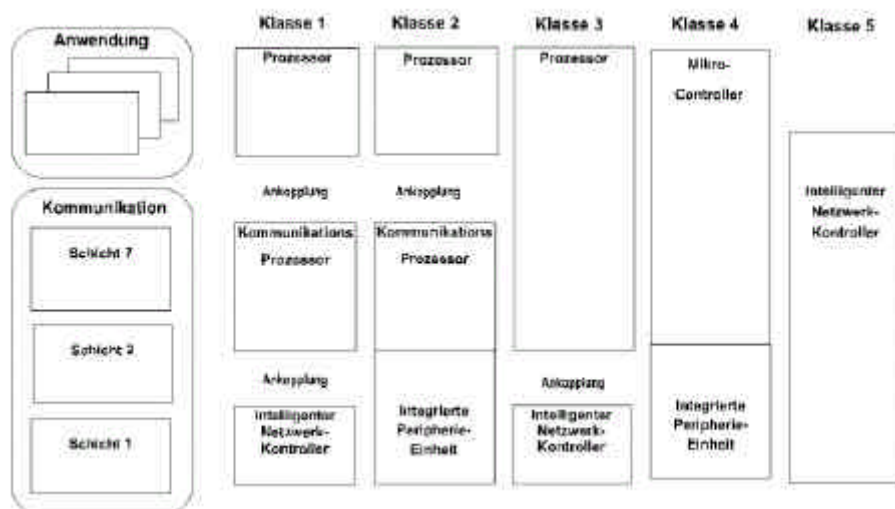


Abbildung 1 Klassierung der Anschaltungen

wendungsbereich sind u.A. die CAN Anschaltungen oder der DPC31 für den PROFIBUS, wo Teile der Netzwerkanschaltung direkt in den Mikrokontroller als intelligente Netzwerk-Controller integriert worden sind. Die Klasse 4 ist vor allem dort interessant, wo Standardmikrokontroller, die heute für ein paar Franken zu haben sind, neben der einfachen Anwendung auch gleich die Kommunikation übernehmen können. Dies ist zum Beispiel bei einfachen PROFIBUS Anschaltungen der Fall, die keine speziellen Hardwareanschaltungen benötigen.

Die Klasse 5 ist der Feldbusanschluss für den Sensor-Aktor Bereich. Es wird keine lokale Vorverarbeitung vorgenommen und der Netzwerk-Controller übernimmt die Abarbeitung des gesamten Protokolls. Dies ist mit vertretbarem Aufwand nur für einfache Schicht 7 Protokolle möglich, wie Sie im Bereich der Dezentralen Peripherie ohne lokale Vorverarbeitung

Die gleichen Überlegungen müssen auch über die vorgesehenen Stückzahlen angestellt werden. Sollen nur einzelne Anschlüsse für spezielle Messgeräte oder Sensoren erstellt werden, so können kaum eigene Entwicklungsleistungen amortisiert werden. Soll aber ein Produkt entwickelt werden, das nachher in grossen Stückzahlen auf den Markt kommt, so kann mit eigener Entwicklungsleistung unter Umständen der Herstellungspreis der Kommunikationsanschaltung reduziert werden. Für alle industriellen Kommunikationssysteme ist es heute möglich, von einfachen Chips mit Schemas und Aufbauanleitungen bis zur kompletten Anschaltbaugruppe auf einer Platine alle Arten von Anschaltungen von entsprechenden Technologielieferanten zu erhalten. Nur in seltenen Fällen von grossen Massenherstellern lohnt es sich, die Technologie der industriellen Kommunikation selber zu erarbeiten.

Bezeichnung	Anwendung	Forderungen	Bitrate
PA/GA	Prozessautomation Gebäudeautomation	Grosse Distanzen Langsame Prozesse	93.75 KBit/s
MPI	Programmierung	Kompatibel mit SIMATIC	187.5 KBit/s
FMS	Master-Master Kommunikation	(nur noch wenig eingesetzt)	500 KBit/s
DP	Dezentrale Peripherie	Effiziente Protokolle	1.5 MBit/s
MC	Motion Control	Kurze Zykluszeiten	12 Mbit/s

Tabelle 1 Anforderungen unterschiedlicher Anwendungsgebiete

und Sensor-Aktor-Busse eingesetzt werden. Diese Anschaltungen haben sehr oft die Funktion von einem Multiplexer.

2 ANFORDERUNGEN

Die Anforderungen an eine Feldbusanschaltung können nun nach weiteren Kriterien zusammengefasst werden [2,3]: Kostengünstig, leistungsfähig, marktgängig und Offenheit.

2.1 Kostengünstig

Die verschiedenen Kommunikationsanschaltungen haben nicht nur unterschiedliche Leistungsmerkmale, sondern unterscheiden sich auch betreffend dem Preis, den eine solche Anschaltung kosten darf. Wenn wir als Faustformel davon ausgehen, dass die Netzwerkanschaltung eines Systemelementes nicht mehr als 30% der Elementkosten ausmachen soll, dann haben wir die möglichen Preisdimensionen aufgeführt: Ein Anschluss an ein Industriesteuerungssystem von 10'000 Fr. darf ohne weiteres in der Grössenordnung von 3'000 Fr. liegen. Wenn aber ein einzelner Sensor von weniger als 100 Fr. angeschlossen wird, muss der Feldbusanschluss für weniger als 30 Fr. möglich sein. Ja, wenn ein einzelner Kontaktschalter von unter einem Franken angeschlossen werden soll, muss der Anschluss für einen Sensor-Aktor-Bus auch unter einem Franken möglich sein. Insbesondere der letzte hier geschilderte Bereich wird heute in der Praxis noch nicht erreicht. Bei den heutigen Sensor-Aktor-Busse können die Anschlusskosten (je nach Technologie und Bussystem) zwischen 10 bis 40 Franken pro Anschlusspunkt betragen.

2.2 Leistungsfähig

Die Anforderungen an eine Feldbusanschaltung sind auch unterschiedlich entsprechend dem Anwendungsgebiet. Je nach Einsatz verändern sich die Anforderungen an die Reaktionszeiten und Datenmengen die übertragen werden müssen. Aufgrund dieser Anforderungen können z.B. für den PROFIBUS unterschiedliche Leistungsmerkmale zusammengestellt werden (siehe Tabelle 1).

Die Preisstruktur ist somit auch für jedes Profil wieder unterschiedlich. Es ist ganz klar, dass je höher die Anforderung, desto höher die Kosten für eine Anschaltung. Somit muss es auch möglich sein, durch Reduktion der Anforderungen die Kosten für eine Anschaltung zu senken.

Beim PROFIBUS haben wir noch ein anderes Unterscheidungsmerkmal: Die einzelnen Geräte können entweder aktive Master oder passive Slave sein. Der Aufwand für die Realisierung eines aktiven Masters ist um einen Faktor 10 grösser als für einen passiven Slave.

Die zu übertragende Datenmenge ist ein weiteres Kriterium. Die Prozessdaten sind in der Regel nicht besonders umfangreich. Es setzt sich heute durch, für einen Prozesspunkt einen Float-Wert von 4 Byte gefolgt von einem Statusbyte einzusetzen. Dies bedeutet, dass pro Wert 5 Byte übertragen werden müssen. Beim PROFIBUS können so mehr als 40 Prozesspunkte in einem Gerät auf einmal übertragen werden während beim CAN für jeden Prozesspunkt ein eigenes Telegramm eingesetzt werden muss. Für die Parametrierung und das Engineering werden grössere Datenmengen übertragen. Bei einem Te-

temperatursensor nach dem Profil PROFIBUS-PA sind dies z.B. 172 verschiedene Parameter mit unterschiedlichem Umfang.

2.3 Marktgängig

Warum spielt die Marktgängigkeit eine wichtige Rolle? Ein einzelner Hersteller kann nicht mehr alle Produkte die es zur Integration eines komplexen Systems braucht selber entwickeln und herstellen. Darum greift man zu offenen Schnittstellen, die mit marktgängigen Produkten abgedeckt sind. Der Hersteller beschränkt sich auf die Entwicklung und Produktion desjenigen Produktes, wo seine Kernkompetenzen sind und ergänzt seine Produkte für den Systemaufbau mit denjenigen von anderen Herstellern. Ein Feldbus ist somit umso geeigneter, desto mehr Hersteller und Produkte es von diesem System gibt.

Ebenso wichtig ist die Zukunftssicherheit. Wenn in eine neue Technologie investiert wird, will man sicher sein, dass die Kosten der Entwicklung amortisiert werden können. In der Automatisierungstechnik ist der Produktlebenszyklus wesentlich grösser und länger als in der Informationstechnik üblich. Bei der Informationstechnik veraltet die Hardware nach einem Jahr, die Betriebssysteme werden alle drei Jahre ausgewechselt und die Anwendungsprogramme spätestens nach zehn Jahren. In der Automatisierungstechnik will man ein Gebäude 30 Jahre mit derselben Infrastruktur nutzen, in der Chemischen Industrie haben wir noch heute 20 jährige Informatik im Produktionseinsatz und eine Fertigungsstrasse bleibt mindestens 10 Jahre in Betrieb.

Mehr als 50% aller Feldbus-Systeme in der Automatisierungstechnik werden heute mit PROFIBUS nach IEC 61158 realisiert. Die Prognosen deuten darauf hin, dass dies auch in der Zukunft so bleiben wird. Eine Alternative stellt im Moment nur Ethernet basierte Technologie dar, die aber noch in mehrere Varianten aufgeteilt ist. Mit einem einheitlichen Massenmarkt ist erst in der nächsten Gerätegeneration zu rechnen.

2.4 Offenheit

Feldbusse sind auf bestimmte Anwendungen zugeschnitten. Der PROFIBUS-DP ist optimiert zur Anschaltung von Dezentraler Peripherie (DP) an eine Steuerung, der PROFIBUS-PA wurde speziell für die Prozess-Automation (PA) entwickelt. Wenn jetzt aber Steuerungen auf einen solchen Feldbus geschaltet werden, haben Sie zusätzliche Anforderungen: Die Kommunikation soll auch zwischen den Steuerungen möglich sein. Ebenso müssen Programmierer und Engineeringwerkzeuge über denselben Bus auch auf die Steuerungen zugreifen können. Die Steuerung soll über den Bus in Betrieb genommen werden können.

Diese Funktionen sind nicht weiter normiert, da die Erfahrung gezeigt hat, dass hier sich die einzelnen Steuerungsbauer sehr stark unterscheiden. Die Programmierschnittstelle ist und bleibt herstellerspezifisch. Ein Feldbus für Embedded Control muss somit mit firmenspezifischen Erweiterungen ergänzt

werden können, ohne die Kompatibilität mit der Normierung zu verlieren.

Offenheit hat verschiedene Aspekte: technische und juristische Offenheit. Die technische Offenheit bedeutet, dass wir auf die technische Dokumentation zugreifen können und eine genaue Beschreibung zugänglich ist, die für eine Implementierung verwendet werden kann. Oftmals wird die technische Offenheit heute mit internationalen Normen dokumentiert. Als Beispiel kann hier IEC 61158 dienen, wo 10 verschiedene Feldbusse für die allgemeine Automatisierung normiert sind.

Die juristische Offenheit ist viel komplexer. Wenn ein Feldbus eine internationale Norm ist, kann er trotzdem einen Patentrechtlichen Schutz umfassen, d.h. wir brauchen eine Lizenz um diese Norm in einem Produkt implementieren zu dürfen. Die Normierungsgremien verlangen nur, dass diese Einschränkungen bei der Normierung deklariert werden, und alle Hersteller zu denselben Bedingungen eine Lizenz bekommen. Oftmals besteht eine generelle Freigabe, die sich auf die normierten Teile beschränkt, d.h. man darf den normierten Teil implementieren, aber keine herstellerspezifischen Ergänzungen vornehmen. Diese Einschränkungen gelten z.B. bei der DeviceNet und Foundation Fieldbus Technologie nach IEC 61158. Für den PROFIBUS gibt es zwar Patente von Siemens, aber keine Einschränkungen betreffend den Nutzungsrechten und Lizenzen: Es dürfen auch nur Teile der Norm implementiert werden und beliebige Ergänzungen vorgenommen werden.

3 STRUKTUR

Damit die unterschiedlichen Implementierungsmöglichkeiten besser aufgezeigt und ausgenutzt werden können, wird ein Teil des PROFIBUS, der PROFIBUS-FDL (FDL = Fieldbus Data Link) hier in einzelne Module zerlegt.

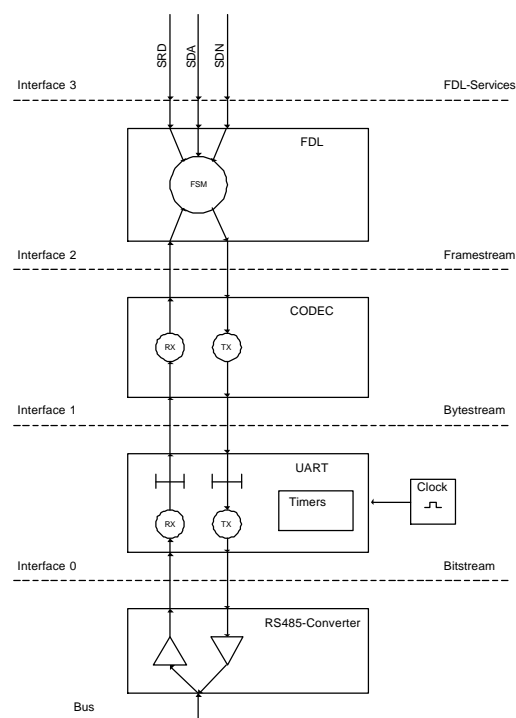


Abbildung 2 Systemstruktur

3.1 Systemstruktur

Die Struktur des Systems wird in der Abbildung 2 dargestellt.

3.2 Modul FDL

Im Modul FDL (FDL = Fieldbus Data Link) werden die Dienstleistungen der Kommunikation bereitgestellt. In der Norm sind diese Zustandsmaschinen (FSM = Final State Maschine) abschliessend, und in der neuen IEC 61158 auch formal vollständig definiert [4,5].

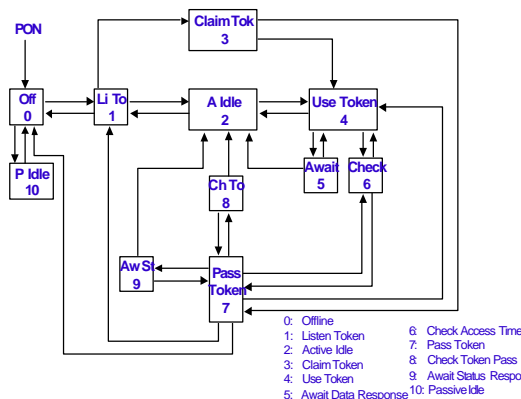


Abbildung 3 FDL Zustandsmaschine

Für einen Master (mit Initiator und/oder Responder Funktion) muss diese Zustandsmaschine vollständig implementiert werden. Für einen Slave (nur Responder Funktion) müssen nur die Zustände 0 (Off) und 10 (P Idle) implementiert werden, d.h. der Funktionsblock ist praktisch leer.

Die Zustandsmaschine ist zeitabhängig, d.h. einzelne Zustandsübergänge werden durch die Zeit bestimmt. Diese Zeit kann aber in einem grossen Rahmen durch Parameter verstellt werden, so dass auch langsame Implementierungen noch normgerecht möglich sind. Die Grenzen werden aber hier durch die praktische Einsetzbarkeit gegeben.

3.3 Modul CODEC

Im Modul CODEC werden aus den Nutzdaten die Telegramme zum Senden gebildet und die ankommenden Telegramme decodiert und auf Übertragungsfehler kontrolliert. Fehlerhafte Telegramme werden weggeworfen. Ebenso finden wir in diesem Block den Adressfilter, d.h. nur Telegramme mit der physikalischen Adresse der Station werden weiter ausgewertet. Die Telegramme werden aufgrund des Funktionscodes dem richtigen SAP übergeben.

Diese Rahmenbildung (Framing) ist zeitkritisch, d.h. es müssen bestimmte Bedingungen zwingend eingehalten werden:

- Synchronisation und schlupfflose Übertragung
- Antwortverzögerung (Station Delay of Responder)

Synchronisation

Damit der Beginn eines Telegramms genau definiert und festgestellt werden kann, wird eine schlupffreie Übertragung gefordert d.h. die einzelnen UART Zeichen eines Telegramms müssen ohne Pause übertragen werden. Nach dem Stop-Bit folgt direkt

das Start-Bit des nächsten Zeichens. Der Beginn eines Telegramms wird mit einer Pause grösser 33 Bit, d.h. mit mindestens drei Zeichen Abstand signalisiert.

Die ersten Implementierungen des PROFIBUS waren hier noch sehr tolerant, d.h. auch innerhalb eines Telegramms haben sie Pausen akzeptiert, wenn diese Pausen die 33 Bit nicht überschreiten. Neuere ASICs sind hier sehr strickt und verwerfen Telegramme, die einen Schlupf aufweisen.

Somit muss bei der Implementierung das Zusammenwirken des Framing und des Rx/Tx Teiles sehr eng sein und die Zeitverhältnisse müssen genau kontrolliert werden.

Antwortverzögerung

Mit der TSDR (SDR = Station Delay of Responder) wird vorgeschrieben, in welchen Zeiten der Responder einem Initiator frühestens (minTSDR) und spätestens (maxTSDR) eine Antwort zustellen muss.

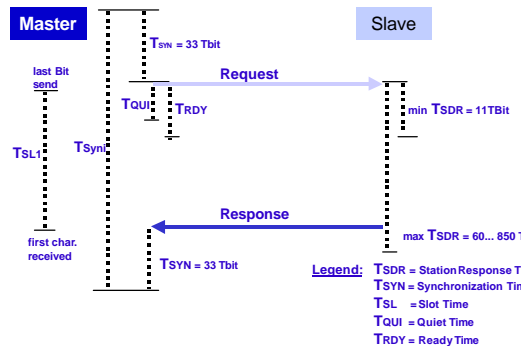


Abbildung 4 Zeitverhältnisse

Die maxTSDR kann durch den Anwender fast beliebig gross gemacht werden. Somit können auch langsame Softwarelösungen realisiert werden. In der Praxis wird die Systemreaktionszeit (z.B. Slot Time) durch den langsamsten DP-Slave bestimmt, darum werden in der Norm enge Grenzen für max TSDR empfohlen.

Der Master (oder Initiator) muss innerhalb der TRDY (Ready Time) bereit sein, die Antwort des Responder zu empfangen. Bei den oben aufgeführten Werten muss er somit eine TRDY von weniger als 11 Tbit aufweisen. Die erhöhten Werte des min TSDR werden im Mischbetrieb nötig, damit auch mit SW realisierten "alten" FMS-Knoten den neueren mit ASIC realisierten DP-Slave folgen können und die Stationen korrekt erkannt werden. Praktische Messungen haben gezeigt, dass dies z.B. bei der Life-List zu Problemen führt, d.h. der Master kann trotzdem nicht alle Slave erkennen da nicht alle ASICs die minTSDR für alle Dienste korrekt anpassen.

3.4 UART

In diesem Block UART (= Universal Asynchronous Receiver Transmitter) werden die einzelnen Zeichen gesendet und empfangen. Sie werden dazu mit einem Start-Bit, einer Parität und einem Stop-Bit versehen.

Damit der Empfänger die einzelnen Bits richtig erkennen kann, braucht er einen Clock, der

mindestens 4x Grösser als die Bitrate ist. Nur mit einem solchen Clock kann der korrekte Pegel in der Mitte der Bitdauer mit einer minimalen Toleranz festgestellt werden.

Für Slaveanschlüssen ist es üblich, hier noch eine Bitratenerkennung anzufügen. Eine Bitraten-Erkennung funktioniert nach dem folgenden Prinzip:

Ein Timer überwacht die Kommunikation. Wenn ein korrektes Zeichen (Parität und Stop-Bit) empfangen wird, wird der Timer zurückgesetzt. Wenn der Timer abläuft wird eine neue Bitrate eingestellt. Immer wenn zwei fehlerhafte Zeichen eingetroffen sind, wird die Bitrate um eine Stufe verstellt bis richtige Zeichen erkannt werden können.

3.5 Zeitanforderungen

Die Anforderungen an die Ausführungszeit sind für einzelne Teile sehr unterschiedlich und sollen hier etwas zusammengestellt werden. Es sind dazu die folgenden Berechnungen und Abschätzungen gemacht worden:

UART: Das NRZ-Datensignal muss mit einem mindestens 4 mal grösserem Clock als die Bitrate abgetastet werden (Kommerzielle UART verwendet in der Regel einen 16x grösseren Clock als die Bitrate).

CODEC: Alle 11 Bit erreicht diesen Block ein neues Zeichen.

FDL-Responder: Die Antwort sollte spätestens nach maxTSDR bereitstehen.

FDL-Initiator: Der Initiator muss nach maximal minTSDR, d.h. nach TDRY bereit sein, die Antwort des Responders zu empfangen.

Für einzelne Bitraten gelten somit die Zeiten gemäss Tabelle 2.

Praxis wird heute auch bei zeitkritischen Anwendungen 90 bis 95% des Codes in C und nur ein kleiner Rest in Assembler geschrieben.

Für die Realisierung von PROFIBUS-FDL hat Software den Vorteil, dass schnelle Entwicklungszyklen realisiert werden können und flexibel auf neue Anforderungen reagiert werden kann. Dies ist insbesondere in einem frühen Projektstand von Vorteil.

Für erste Versuche sind die beschriebenen Module in C auf der seriellen Schnittstelle unter Windows NT realisiert worden. Als Einschränkung wurde das Modul FDL nur als Single-Master, d.h. ohne Token-Passing implementiert. Die praktischen Tests haben gezeigt, dass diese Schnittstelle bis zu 19200 Bit/s eingesetzt werden kann [6,7], was vor allem auf Einschränkungen des Standardtreibers unter Windows-NT zurückzuführen ist.

Für industrielle Steuerungen sind diese Protokolle inklusive Token-Passing in einem Mikrocontroller realisiert worden. Messungen haben gezeigt, dass Bitraten bis zu 187,5 KBit/s mit dem im Mikrocontroller eingebauten UART ohne weiteres erreicht werden können. Für eine weitere Beschleunigung bis 1,5Mbit/s muss ein externer UART eingesetzt werden.

4.2 Hardware

Für den PROFIBUS stehen heute verschiedene ASICs mit fest eingebauten Funktionen zur Verfügung. Eine gewisse Anzahl dieser ASICs verfügt auch über eine PROFIBUS-FDL Schnittstelle (z.B. auch der SPC4 von Siemens). Für eine kostengünstige Alternative ist aber die Frage, wie weit sich die einzelnen Module auch als Hardware nachbilden

Bitrate (Tbit)	9.6 kBit/s (104µs)	187,5 kBit/s (5,3 µs)	1.5 Mbit/s (0,66 µs)	12 Mbit/s (0,083µs)
Clock im UART	> 38,4 kHz (<26 µs)	>0,75 MHz (<1,3 µs)	> 6 MHz (<0,166µs)	> 48 MHz (<0,02µs)
Clock im CODEC	>872 Hz (<110 µs)	>17 kHz (<58 µs)	> (7,3 µs)	> 4 MHz (<0,9 µs)
FDL-Responder	< 30 Tbit (< 3 ms)	< 60 Tbit (< 320 µs)	< 150 Tbit (< 100µs)	< 800 Tbit (< 66 µs)
FDL-Initiator	11 Tbit (110 µs)	11 Tbit (58 µs)	11 Tbit (7,3 µs)	11 Tbit (0,9 µs)

Tabelle 2 Unterschiedliche Zeitanforderungen

Wenn wir davon ausgehen, dass 10 bis 20 Instruktionen eines Mikrocontrollers mindestens 50 µs benötigen erkennen wir die möglichen Grenzen einer Realisierung in Software. Wenn für die Bearbeitung einer Aufgabe weniger als 50 µs zur Verfügung stehen, ist eine Realisierung in Hardware anzustreben.

4 REALISIERUNGEN

Um den PROFIBUS-FDL zu realisieren können verschiedene Technologien eingesetzt werden.

4.1 Software

Bei der Realisierung in Software steht die Programmierung mit C (möglichst unabhängig von einem Prozessor) im Vordergrund. Das beschränkende Element ist hier die mögliche Verarbeitungsgeschwindigkeit.

Beim Einsatz von modernen Compilern ist nicht damit zu rechnen, dass eine wesentliche Beschleunigung der Abarbeitungszeiten durch eine Programmierung in Assembler erreicht werden kann. In der

lassen.

Die Realisierung einer Hardware erfolgt heute am zukunftssichersten mit einer Beschreibung mit VHDL (Very high speed Integrated circuit Hardware Description Language). Die Beschreibungen einer Hardware in VHDL können anschliessend wahlweise in einen FPGA oder bei genügenden Stückzahlen in einem ASIC realisiert werden.

VHDL erlaubt auch eine prozedurale Beschreibung, die einer Realisierung in C sehr nahe sein kann. Mit dem Einsatz eines FPGA können auch kurze Entwicklungszyklen realisiert werden. Die grösste Anforderung an die Entwicklung in VHDL ist aber ein systematischer Test der definierten Funktionen.

Für die Überprüfung der Machbarkeit ist das Modul UART für eine Bitrate von 12 MBit/s in ein CPLD mit VHDL realisiert worden [8]. In einfachen Funktionstests ist die korrekte Funktion überprüft worden. Somit ist es grundsätzlich möglich auch eine reine Softwarestruktur mit diesem UART bis 12

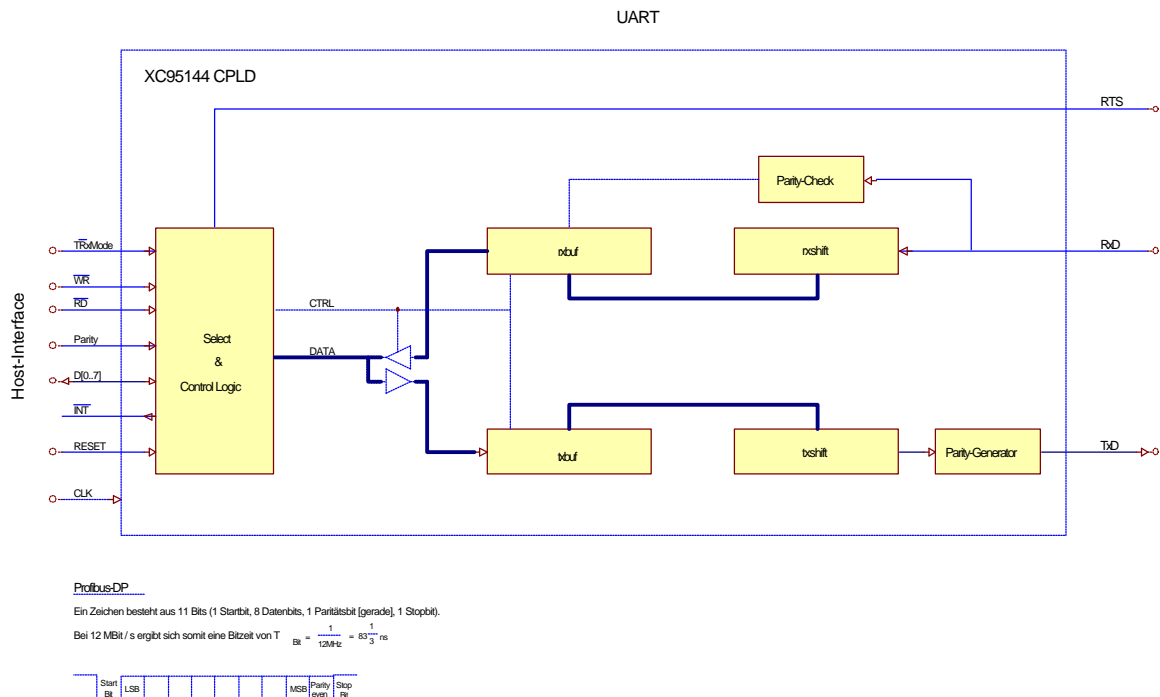


Abbildung 5 Struktur des 12 MBit/s UART im CPLD

MBit/s zu realisieren.

4.3 Hybride Lösungen

In der Praxis werden hybride Lösungen am meisten eingesetzt. Die schnellen Blöcke werden in Hardware realisiert und die komplexen „langsamen“ Funktionen in Software. Die Kunst ist es, für jedes Anforderungsprofil die richtige Schnittstelle und Aufteilung zu finden.

Eine modulare Struktur und definierte Schnittstellen zwischen den verschiedenen Blöcken erlauben den Aufbau von Systemen mit unterschiedlichen Eigenschaften. Eine mögliche Struktur ist in der Tabelle 3 zusammengestellt.

Bitrate	≤ 187.5 KBit/s	≤ 1.5 MBit/s	≤ 12 MBit/s
UART Block	HW	HW	HW
CODEC Block	SW	HW	HW
FDL Block	SW	SW	HW

Tabelle 3 Unterschiedliche Realisierungen

Die erste Lösung läuft schon auf einer Standard-HW (mit UART). Somit ist es möglich eine Realisierung zu finden, die ohne zusätzliche Hardware auskommt. Es ist grundsätzlich möglich, jede serielle Schnittstelle auf einer Steuerung mit diesem Protokoll auszurüsten, ohne zusätzliche Hardwarekosten. Der einschränkende Punkt ist hier die zur Verfügung stehenden Softwareressourcen punkto Prozessorleistung und Speicher.

Andere Baudraten können mit einem „normalen“ UART nur eingestellt werden, wenn spezielle Quarze eingesetzt werden. Dies wird in der praktischen Realisierung die einschränkende Grösse sein. Es ist im Detail zu prüfen, wieweit die Bitraten 45,45 und 93,75 KBit/s bei der vorhandenen Hardware einge-

stellt werden können und welches die Einschränkungen bei der Prozessorlast sein werden.

Die zweite Lösung kann mit einem FPGA oder CPLD realisiert werden. Es ist dabei anzustreben, dass eine Lösung gefunden wird, die möglichst in allen Steuerungen in der gleichen Konfiguration eingesetzt werden kann. Damit können grosse Stückzahlen erreicht werden und somit die Chipherstellkosten reduziert werden.

Bei der dritten Lösung ist nicht sicher, dass die geforderte Komplexität der Hardwarebausteine die Auswahl von kostengünstigeren Bausteinen erlaubt, als heute fix-fertige ASICs auf dem Markt verfügbar sind. Die geforderten Funktionen sind so umfang-

reich und komplex, dass FPGA in der Preislage von klar über 30 CHF eingesetzt werden müssen.

5 SCHLUSSFOLGERUNGEN

Wer einen Feldbus einsetzen will, verwendet grundsätzlich vorhandene Bausteine und baut so seine Feldbusanschaltung auf. Dies ist der übliche Weg. Wir haben hier gezeigt, dass es spezielle Anforderungen gibt, die es wünschenswert erscheinen lassen, dass der Aufbau der Anschaltung vollständig kontrolliert und selber aufgebaut wird. Unter diesen Voraussetzungen wird es möglich, Feldbusanschaltungen, insbesondere für spezielle Anwendungsfelder, kostengünstig zu realisieren.

REFERENZEN

- [1] Bender, K.: *Entwicklungstrends und Perspektiven*, TR-Fachkongress, Bern 14. bis 16. Oktober 1992, Seiten 49 bis 57
- [2] Felser, M.: *Feldbusse - die "unbekannten Wesen"*; 7-teilige Artikelserie in Precision Automation Ausgaben 3/95 bis 9/95; Verlag Vogt-Schild AG Solothurn
- [3] Felser, M.: *Technologie für Industrielle Kommunikation, Wie komme ich zu einer Feldbus-Anschaltung?* MegaLink 7/96, Seiten 14 bis 16
- [4] IEC 61158, *Digital data communication for measurement and control – Fieldbus for use in industrial control systems*, CDV January 2002
- [5] IEC 61784, *Profile sets for continuous and discrete manufacturing relative to fieldbus use in industrial control systems*, CDV January 2002
- [6] Harder, T.: *PROFIBUS-DP Terminal*, Diplomarbeit an der Softwareschule Schweiz T34.1, 1996
- [7] Lüscher, M., Hilfiker T.: *RS232-Library für den PROFIBUS*, Diplomarbeit an der Softwareschule Schweiz, T41.2, 2000
- [8] Aebersold M.: *12Mbit UART in einem CPLD*, technischer Report der HTA-BE, Juli 2001