Fehlertoleranzaspekte in Feldbussen

Sommer 1997

Alexander Bergolth

Hermann Himmelbauer

Albert Treytl

Inhaltsverzeichnis

Einleitung

Der Anwendungsbereich von Feldbussen erweitert sich stetig. Damit steigen auch die Anforderungen bezüglich Zuverlässigkeit, Verfügbarkeit und Sicherheit der Systeme. Selbst in Applikationen, die keine erhöhten Ansprüche an die Sicherheit des Systems stellen, führt die Anwendung von fehlertoleranten Prinzipien zu einer Steigerung der Qualität und der Verfügbarkeit – das Produktimage steigt.

Diese Arbeit soll einen Überblick über die zur Verfügung stehenden Instrumente und Methoden geben und im Anschluß daran die Implementierung von Fehlertoleranz in speziellen Feldbussen behandeln.

Im Speziellen soll erörtert werden, in wie weit die Funktionalität des Feldbusses bei Ausfall einzelner Busteilnehmer bzw. einzelner Buskomponenten (z.B. Leitungen) aufrecht erhalten werden kann.
Der Implementierungsteil soll insbesondere die Frage erörtern, ob die technisch realisierten Lösungen Fehlertoleranz im Gesamtsystem implementiert haben, oder ob es sich nur um spezielle Lösungen eines Teilaspektes handelt.

Abkürzungen

ARQ	Automatic Repeat Request
CAN	Controller Area Network
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidence
CRC	Cyclic Redundancy Check
DIS	Deutscher Industrie Standard
EMV	Elektromagnetische Verträglichkeit
FEC	Forward Error Correction
FED	Forward Error Detection
ISO	International Standardisation Organisation
NMR	N Times Modular Redundancy, Mehrheitsentscheidung aus N redundanten Systemen

Motivation

Dieses Theoriekapitel teilt die Realisierungsmöglichkeiten in zwei Grundkategorien: Hardware- und Softwarelösungen. Dies ist lediglich eine formale Aufteilung - Fehlertoleranz in einem (Bus)system darf nicht auf diesen beiden Ebenen getrennt betrachtet werden. Erst das planmäßige Zusammenspiel aller Aspekte ergibt ein fehlertolerantes System.
Fehlertoleranz muß Top-Down realisiert werden, sonst ist keinerlei Aussage über ihre Wirksamkeit möglich.

Ursachen für Störungen

Um Fehler[1] wirksam korrigieren zu können, ist es notwendig die Fehlerursachen zu kennen. Erst durch diese Kenntnis ist es möglich, das System gegen Fehler zu schützen, d.h. es fehlertolerant zu gestalten.
Im Zusammenhang mit Feldbussen sind insbesondere folgende Fehlerursachen von Bedeutung:

- **elektrische und elektromagnetische** Störungen - Das Einsatzgebiet der Feldbusse ist meist die Sensor- und Aktorebene in der Prozeßsteuerung. In dieser Ebene ist durch das Schalten von hohen Strömen in unmittelbarer Nähe der Kommunikationsleitungen eine große Belastung durch elektrische Störungen gegeben.
- **Überlastung des Kommunikationsmediums** - In der praktischen Anwendung steigt die Anzahl der Busteilnehmer stetig und damit wird auch das Datenaufkommen am Bus erhöht. Daher kann es zu Blockierungen und Überlasterscheinungen kommen - Diese Zustände stellen zwar keinen fehlerhaften Buszustand dar, können aber sehr wohl in der Anwendung zu Fehlern führen. Es ist nun die Kommunikation (Datenaustausch) so zu gestalten, daß erhöhte Kommunikation nicht den Betrieb stört. Je nach Anwendung muß diese Grenze definiert werden. Im Design wird dann meist ein gewisser Wahrscheinlichkeitswert für die Funktionsfähigkeit des Systems angegeben.
- **Ausfall von Komponenten** - Hier möchte ich nur die Busschnittstellen betrachtet wissen und nicht die dahinter befindliche "Peripherie". Unter diesem Gesichtspunkt ist meiner Meinung nach die mechanische Zerstörung[2] vernachlässigbar, da fast ausschließlich Halbleiterkomponenten verwendet werden. Es muß also der Ausfall von Halbleiterbaugruppen analysiert werden - In der Praxis kann mit einer Exponentialverteilung mit konstanter Ausfallsrate Lambda gerechnet werden.

Diese Zielsetzung der Fehlertoleranz kann nun unterschiedlich realisiert werden: Einerseits in Hardware oder andererseits in Software. Die Schlagworte Hardware und Software sind hier

[1] Im englischen differenziert sich Fehler in folgende drei Begriffe:

- Fault - Fehlerursache
- Error - Fehler(zustand)
- Failure - Ausfall.

Diese Differenzierung soll auch in dieser Arbeit beibehalten werden, um Zusammenhänge klarer darzustellen.
[2] siehe Abschnitt über Steckerproblematik

nicht so sehr als Implementierungsart gedacht sondern zur Unterscheidung zwischen Funktionalität (Software) und Gestaltung von Komponenten (Hardware).

Hardwarelösungen

Parallelredundanz

Ist Parallelisierung ein adäquates Mittel zur Erhöhung der Fehlertoleranz von Bussystemen? Diese Frage kann nicht uneingeschränkt mit Ja beantwortet werden, da in den meisten Bussystemen der Bus die vielen parallelen Kanäle auf ein singuläres Medium zwingt.

Wahrscheinlichkeitsrechnung

Hier nur einige grundlegende Regeln zur Berechnung der in den nächsten Abschnitten angeführten Ergebnisse.

- Die Intaktwahrscheinlichkeit eines Komponente A ist p(A)
- Die Wahrscheinlichkeit für eine Defekt ist q(A) = 1 - p(A).
- Die Intaktwahrscheinlichkeit der Parallelschaltung von n Komponenten berechnet sich zu: p(A1||A2||...||An) = 1- (q(A1)*...*q(An))
- Die Intaktwahrscheinlichkeit der Serienschaltung von n Komponenten berechnet sich zu: p(A1+A2+...+An) = p(A1)*...*p(An)

Modellierung

Um den Ausfall der Busschnittstelle, der sowohl aus dem Ausfall der Verbindung zur angeschlossenen Unit (Busteilnehmer) als auch der zum Bus resultieren kann, wird die Schnittstelle in zwei Komponenten zerlegt - der Unit-Schnittstelle US und der Busschnittstelle BS.
Abbildung 1 zeigt die einfache Busverbindung. Der Wert für die Zuverlässigkeit[3] Rges des Gesamtsystems berechnet sich zu: Rges = p(U) * p(US) * p(BS) * p(B)

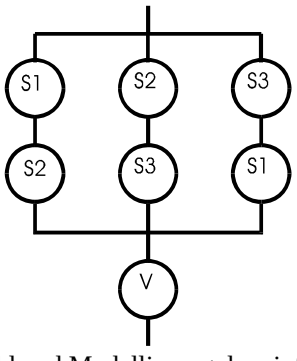

Abb. 1: Blockschaltbild und Modellierung der einfachen Busverbindung

[3]Zuverlässigkeit gibt eine Wahrscheinlichkeit an, daß die Funktionsfähigkeit bis zum Zeitpunkt t gegeben ist. In den Rechnungen wird sie durch die Intaktwahrscheinlichkeit ersetzt, die zeitunabhängig ist.

Um nun einem Fehler vorzubeugen kann man betroffene Komponenten mehrfach auslegen: Zum Beispiel kann die Busankopplung dreifach ausgelegt werden. (Abbildung 2a) Soll die Unit - Sie ist als IO-Baustein zur dahinterliegenden Informationsquelle gedacht. - vernachlässigt werden, kann man sie ausfallsfrei annehmen (p(U) = 1)

Trotz Parallelisierung sinkt die Zuverlässigkeit.(Rges = (1 - (1-p(U1)*p(US1))*(1-p(U2)*p(US2))*(1 - p(U3)*p(US3))* p(BS1) *p(BS2) *p(BS3) *p(B). Führt man auch den Bus parallelredundant aus kommt es zu einer Erhöhung der Zuverlässigkeit (Die Berechnung erfolgt anlog zu den vorherigen Auswertungen). Abbildung 2b zeigt das Modell für diesen Fall.

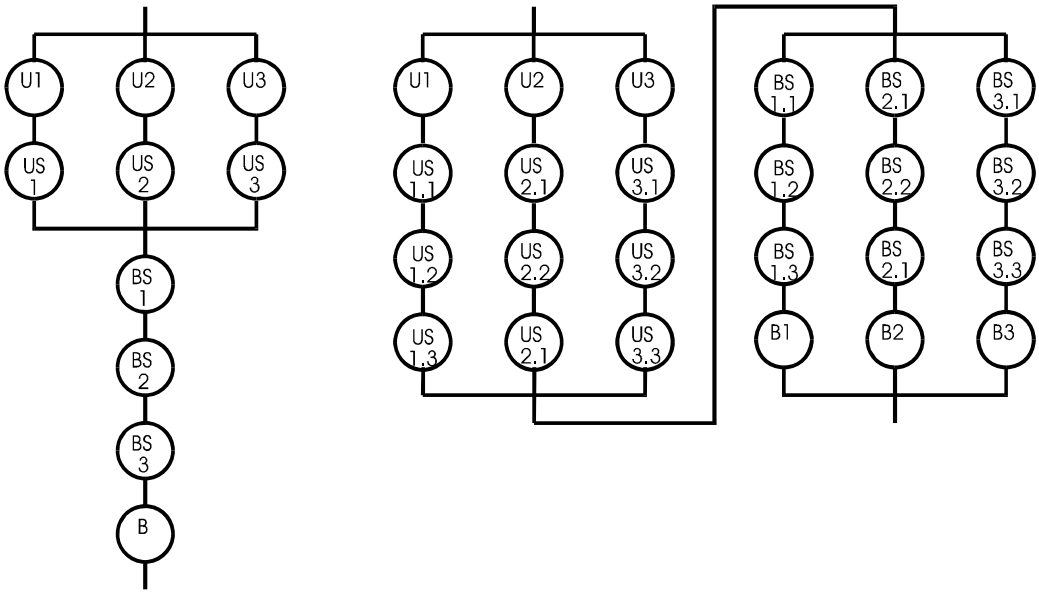

Abb. 2: (a)Modell für eine dreifach ausgelegte Busankopplung (b) Modell für totale dreifach Parallelität des gesamten Bussystems

Grenzen in Theorie und Anwendung

Nach numerischer Auswertung erhält man ein faszinierendes Ergebnis: Die Zuverlässigkeit kann durch Parallelisierung nur auf ein bestimmtes Maximum gehoben werden. Fügt man weitere Parallelsysteme hinzu senkt man die Zuverlässigkeit wieder (Serienpfade im Modell). In [LIT 2] finden sich neben der exakten Zuverlässigkeitsberechnung auch Tabellen, die diesen Sachverhalt erklären und veranschaulichen.

Praktisch wird die Parallelisierung primär durch den Kostenfaktor und eventuell durch die Systemkomplexität begrenzt.

NMR-Systeme

Diese Systeme sind die „klassischen“ Parallelsysteme. Mehrere parallel ausgeführte Komponenten bilden über einen Voter das Ausgangssignal - mindestens m aus n Systemen müssen den gleichen Ausgangswert haben, damit der Ausgang diesen Wert annimmt. Die Zuverlässigkeit ist innerhalb einer Missionszeit sehr hoch, nimmt dann jedoch rapide ab.

Schwachstelle dieser Konstruktion ist der Voter, da er nur singulär ausgeführt ist. Die Zuverlässigkeit des Systems hängt daher extrem von der Ausführung und dem Entscheidungskonstrukt des Voters ab.

Totalvermaschung - demokratische Entscheidung

In diesen Systemen sind alle Komponenten parallel ausgeführt und miteinander verbunden (Totalvermaschung). Die Entscheidung wird letztendlich demokratisch durchgeführt - dies erfordert eine spezielle Voterkonstruktion - z.B. in Form eines mechanischen Teils an dem alle Aktoren angreifen. Fehlentscheidungen fließen im Gegensatz zu NMR-Systemen in das Endergebnis zwar ein, werden aber durch die Mehrheit der „richtigen“ Ergebnisse abgeschwächt. Damit stellt das Hinzufügen von Komponenten einen Gewinn an Zuverlässigkeit dar.

Sicherheit vs. Zuverlässigkeit

Ein kurzer Einwurf zum Widerstreit zwischen Sicherheit und Zuverlässigkeit.
Sicherheit ist ein meist subjektives Maß, das den Schutz von Mensch, Tier und Umwelt beschreibt.
Zuverlässigkeit hingegen ist die Wahrscheinlichkeit, daß ein System vom Zeitpunkt t0 bis t1 im Rahmen seiner Spezifikation einwandfrei arbeitet
Die Einführung von parallelen Komponenten führt immer zu einem Sicherheitsgewinn jedoch nicht notwendigerweise zu einer Erhöhung der Zuverlässigkeit.

Selbsttest und Abkopplung von Busteilnehmern:

Ist ein Busteilnehmer defekt und stört somit die Übertragung auf dem Gesamtsystem ergibt sich die einfache Lösung, ihn von Bus abzutrennen. Dies ist aus der Sicht des Busbetriebs die richtige Lösung - das Gesamtsystem kann andere Forderungen stellen. Sind die Bremsen eines Autos via CAN angesteuert, dürfen diese nicht einfach durch Abschaltung aus dem Prozeßgeschehen (Fahren) entfernt werden. Für die Abschaltung und Kontrolle - es muß festgestellt werden, ob der Busteilnehmer defekt ist - bieten sich prinzipiell zwei Möglichkeiten:

1. Kommandierte Abschaltung:
2. Eigenverantwortliche Abschaltung:

Kommandierte Abschaltung

Testeinheiten sollen die Funktionstüchtigkeit einer Baugruppe feststellen und, falls ein Fehler gefunden wird, entsprechende Aktionen einleiten. Die Möglichkeiten für solche Aktionen reichen von einem Neustart der Einheit bis zur Abschaltung der fehlerhaften Komponente. Der Nachteil der befohlenen Abkopplung bzw Abschaltung liegt in der Anfälligkeit für Verklemmungssituationen (Deadlock). Dazu ein Beispiel: Zwei Computer haben eine bestimmte Aufgabe zu erfüllen und sich aus Sicherheitsgründen zu kontrollieren. Erhalten die beiden Computer unterschiedliche Ergebnisse, veranlaßt jeder die Abschaltung des anderen - Wer hat das größere Gewicht in der Entscheidungsfindung ? - Ein Deadlock ist bei Gleichberechtigung unausweichlich.

Es mag jetzt der Einwand folgen, daß dieses Beispiel unrealistisch ist und NMR-System zu bevorzugen sind. Dem möchte ich entgegen halten, daß auf einem Bus die Abschaltung eines Teilnehmers von Nöten ist, wenn dieser den Bus durch ständiges Senden blockiert und auf Busarbitrationssignale nicht reagiert.
Bezüglich NMR-Systemen sei auf den Abschnitt über Parallelisierung verwiesen.

Eigenverantwortliche Abschaltung

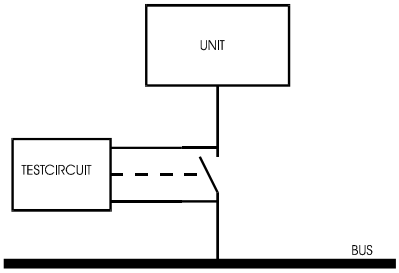

Abbildung 3: Busteilnehmer mit Selbsttesteinheit

Abbildung 3 soll dieses System veranschaulichen:
Die Selbsttesteinheit (Testcircuit) ist in der Lage den Busteilnehmer vom Bus zu trennen. Dies geschieht zu zwei Zeitpunkten:

1. in der Testphase - Durch die Auftrennung ist es möglich die Fehlerursache zu lokalisieren, und diese dem Bus oder dem Busteilnehmer (Unit) zuzuordnen. Entsprechend dieser Entscheidung können dann weitere Maßnahmen eingeleitet werden, wie zum Beispiel eine Abschaltung der Unit, Aufrechterhaltung der Busabkopplung, umschalten auf andere Units (Standby-Systeme[4]) usw.
2. im Fehlerfall

Die Testeinheit muß selbst in hohem Maß fehlertolerant sein, da sie über die Abkoppelung oder Abschaltung der Unit entscheidet und damit einen kritischen Faktor darstellt.
Der CAN-Bus[5]verwirklicht diese Philosophie der Eigenanalyse und -verantwortung, die Blockierungen weitgehend ausschließt. Ebenso gibt es in Flugzeugen (MIL-STD-1553B) vorgeschriebene Testzyklen, um die Systemfunktionen zu überprüfen.

Übertragungstechnik

Die Übertragungsart trägt in hohem Maße dazu bei, die Fehlerquellen des verwendeten Kanals zu minimieren. Ich möchte hiermit drei Übertragungsverfahren vorstellen, die jeweils auf spezielle Störgrößen zugeschnitten sind.

Differentielle Übertragung

[4]Cold-Standby: es gibt ein Ersatzgerät;
Hot-Standby: ein anderes Gerät stellt seine eigentliche Arbeit ein, um die Aufgabe des ausgefallenen Geräts zu übernehmen. Es kommt zu einer Systemdegration (Fail-Soft Prinzip).
[5]siehe folgende Kapitel

Das Prinzip der differentiellen Übertragung besteht darin, daß gleichzeitig das Signal und sein inverses Signal auf einer anderen Leitung übertragen wird[6]. Der Vorteil dieser Übertragungstechnik besteht darin, daß erstens Gleichtaktstörungen durch die Abtastung automatisch ausgefiltert werden und daß zweitens Störungen den Datentransport weniger behindern.

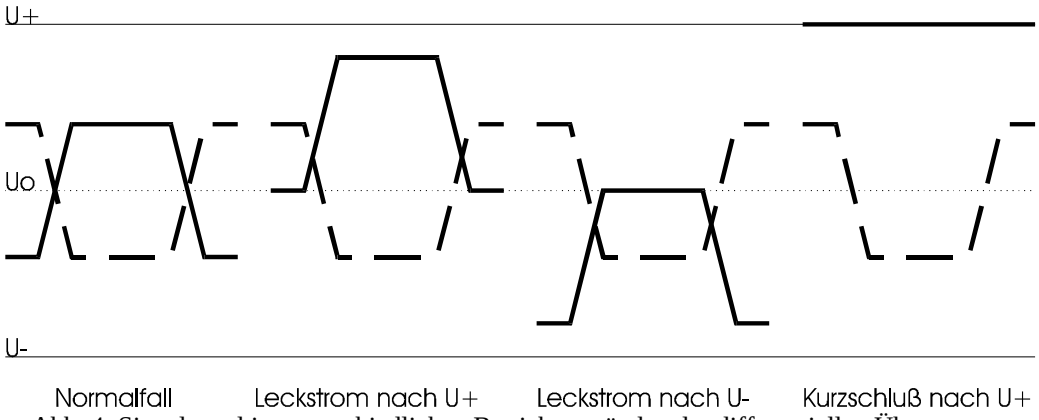

Abb. 4: Signalpegel in unterschiedlichen Betriebszuständen der differentiellen Übertragung

Abbildung 4 zeigt unterschiedliche Fehlerzustände. Da immer die Differenzspannung[7] den entsprechenden Datenwert (qualitatives Merkmal positiv/negativ) kodiert, kann mittels einer zusätzlichen quantitativen Messung auch der Fehlerzustand bestimmt werden. Man sieht sofort, daß nicht alle Fehlerursachen zu einer Störung der Übertragung führen. Ein einfacher Leckstrom zum Beispiel stört die Übertragung nicht. Ein weiterer Pluspunkt dieser Übertragungstechnik ist, daß mit entsprechenden intelligenten Schaltungen Fehlerzustände automatisch einer Reparaturstelle mitgeteilt werden können bevor das System vollständig außer Funktion ist.

Multiple Abtastung

Das Datenbit wird nicht nur einmal abgetastet, sondern mehrmals. Die so erhaltenen Abtastwerte werden dann einem Datenwert zugeordnet. Je nach Übertragungsart und -strecke können bestimmte Verfahren zur Auswertung herangezogen werden. So macht es fast immer Sinn, den ersten und letzten Abtastwert zu verwerfen, um Einschwingvorgänge bei Übergängen von zwischen zwei unterschiedlichen Signalzuständen vernachlässigen zu können. In diesem Zusammenhang sei auch auf die Problematik der Synchronisation hingewiesen. Alle hier getätigten Aussagen beziehen sich nämlich auf ideale Synchronisation!

Spread-Spektrum-Signale

Diese Signale lassen sich dadurch charakterisieren, daß die Übertragungsbandbreite wesentlich höher als die übertragene Informationsbandbreite ist. Diese Übertragungsart hat jedoch entscheidende Vorteile in stark gestörten Übertragungsmedien und für den Fall, daß ein Kanal von mehreren Teilnehmern gleichzeitig benutzt werden soll.

[6]das Referenzpotential muß nicht mehr übertragen werden, da die Abtastung ebenfalls differentiell erfolgt; Bei den Abtastschaltungen ist auf Gleichtaktfehler zu achten!
[7]Signalspannung zwischen durchgezogener und unterbrochener Linie gemessen.

Vereinfacht dargestellt werden die Informationsbits durch eine pseudo-zufällige Sequenz (z.B.: M-Folgen, Gold-Folgen, Kasami-Folgen) von 0 und 1 kodiert. Der Empfang erfolgt nur mehr korrelativ.
Durch diesen Vergleich während des Empfangs werden Störsignale[8] größtenteils automatisch eliminiert. Es soll hier aber nicht verleugnet werden, daß dieser Korrelationsempfang der kritische Punkt der Spread-Spektrum-Technologie ist. Im Vergleich mit direkt abtastenden System benötigt der korrelative Empfang aufwendigere schaltungstechnische Maßnahmen und erhöht die Synchronisationszeiträume drastisch.
Im Zusammenhang mit Mehrbenutzersystemen soll noch die Orthogonalität der Codesequenzen erwähnt werden. Unter Orthogonalität versteht man in diesem Zusammenhang die Unterscheidbarkeit zweier Folgen, d.h. wie gut die Informationssequenzen eines anderen Teilnehmers als Störsignale unterdrückt werden.

Hauptanwendungsgebiet der Spread-Sprectrum-Signale ist die drahtlose Kommunikation und die Kommunikation über Powerlines, die insbesondere in der Gebäudevernetzung von Interesse sind.

Steckerproblematik

Die Reduktion von Steckverbindungen ist ein entscheidender Schritt zur Erhöhung der Zuverlässigkeit von Bussystemen. Die heutige Halbleitertechnologie ermöglicht qualitativ so hochwertige Bauelemente, daß deren Ausfallswahrscheinlichkeit kleiner oder gleich der von Lötstellen und mechanischen Verbindungen ist. Der Stecker, der noch dazu nicht in einem schützenden Chip eingegossen ist, wird damit der immer fehleranfälligere Teil des Systems.

Software- und Protokollösungen

Kodierungsverfahren

Die Kryptographie ist ein probates Mittel, um Fehlererkennung, Fehlerkorrektur und Rückfrageprotokolle (ARQ) zu realisieren. Es soll hier nicht auf die unterschiedlichen Kodierungsverfahren, wie zum Beispiel Hamming-Codes, BCH-Codes, Fire-Codes, usw. eingegangen werden, die jeweils spezielle Eigenschaften hinsichtlich der Fehlerbehandlung haben (Einzelfehler, Büschelfehler,...), sondern die prinzipielle Anwendung in einem Bussystem und ihrer Auswirkungen auf die Fehlertoleranzaspekte beschrieben werden.

Fehlererkennung (FED)

Sie ist der Grundstein für Fehlertoleranz. Erst durch erfolgreiche Erkennung ist es möglich einen Fehler zu bekämpfen. Durch die Einbringung zusätzlicher Korrekturstellen in den Datenstrom ist es möglich, diesen auf seine Korrektheit hin zu überprüfen. Es sei hier auf

[8]Das Nutzsignal kann im Störsignal vollständig untergehen - dies ist jedoch meist nur in militärischen Anwendungen von Interesse.

weiterführende Literatur [LIT 16] und [LIT 15] verwiesen. Die Praxisbeispiele in den folgenden Kapiteln werden die bei den unterschiedlichen Bussystemen verwendeten Codes und ihren Eigenschaften und Auswirkungen auf die Gesamtfehlertoleranz der Busse aufzeigen.

Fehlerkorrektur (FEC)

Die Mächtigkeit der verwendeten Fehlerkorrektur ist stark vom verwendeten Kodierungsverfahren abhängig. Nichtsdestoweniger induziert Fehlerkorrektur zusätzliche Fehlerquellen, da die Möglichkeit besteht auch "falsch" zu korrigieren. So können aus fehlerhaften Werten richtige Datenworte geformt werden - Es findet eine nicht detektierbare Störung der Übertragung statt. [LIT 15]

Automatische Rückfrage (ARQ)

Ist ein Fehler aufgetreten, kommt es zur automatischen Aufforderung an den Sender, die gesendeten Daten zu wiederholen, solange bis eine fehlerfreie Übertragung zustande gekommen ist bzw. ein Abbruchkriterium [9]erreicht wurde.
Je nach Applikation ergeben Mischformen dieser 3 Basiselemente die für den Anwender die besten Resultate.[LIT 16]

Netzprotokolle

Ein Bussystem ist die Kommunikationsschiene zwischen unterschiedlichen Teilnehmern, die ich nun Knoten nennen möchte. Werden nun Nachrichten ausgetauscht, kommt es zu Problemen bei der Busarbitrierung.- Diese Probleme werden im Rahmen des Bussystems behandelt.
Es soll nun dargestellt werden, inwiefern sich die Eigenheiten der einzelnen Verfahren auf die Fehlertoleranz eines Gesamtsystems im Falle wechselnder Belastung auswirken[10].

Zentrale Busarbitrierung

Das Datenaufkommen ist mit dieser Methode am einfachsten zu handhaben. Die zentrale Arbitrationseinheit stellt aber einen kritischen Schwachpunkt für das System dar und muß daher besonders gegen Ausfälle gesichert werden.
Im Rahmen von verteilten (Feld)bussystemen ist noch die Anzahl der zusätzlichen Steuerleitungen nicht zu vernachlässigen!

Kollisionsverfahren

Aufgrund des Kollisionsdesigns verschlechtert sich das Systemverhalten mit zusätzliche Belastung. Je nach Auslastung kann ein gewisser Wahrscheinlichkeitswert für eine

[9]Die Effizienzansprüche, geforderte Restfehlerwahrscheinlichkeit und die Kanaleigenschaften bestimmen die Wahl des Abbruchkriteriums.
[10]Fehlerkorrektureigenschaften wurden im Abschnitt Kodierung behandelt.

erfolgreiche bzw. fehlerhafte Übertragung angegeben werden. Der Entwickler hat nun über die Verwendbarkeit zu entscheiden.

Timeslotverfahren

Sie stellen in bezug auf die Belastung die sichersten Systeme dar, da jeder Knoten nur eine bestimmte Zeitscheibe zur Verfügung hat, um Daten zu übermitteln. Nachteilig wirkt sich dieser starre Rahmen nur insofern aus, als daß spezielle Synchronisationsmechanismen vorhanden sein müssen, die wiederum vor Fehlerursachen geschützt werden müssen. Als Vorteil wäre noch anzuführen, daß zusätzliche Testzyklen einfach zu integrieren sind.

Tokenverfahren

Bei diesem System benötigt jeder Knoten eine „Staffel“ (Token), die ihm signalisiert, daß er die Sendeberechtigung besitzt. Nachteilig an diesem System ist die komplexe Fehlerbehandlung falls der Token verloren geht oder verstümmelt wird. Der Profibus verwendet solch ein System.

Prioritätsverfahren

Sind in einem System unterschiedliche Nachrichten nicht von gleicher Bedeutung, so kann durch Prioritätscodes der Überlastfall gegliedert und akzeptabel abgearbeitet werden. Je nach Applikation ist darauf Rücksicht zu nehmen, daß auch niederpriore Nachrichten nicht beliebig lange zurückgehalten werden dürfen.
Die Prioritätssteuerung kann z.B. über spezielle Buspegel, zusätzliche Leitungen (Da in Feldbussystemen die geringe Leitungsanzahl eine große Rolle spielt, ist diese Art der Steuerung eher selten) oder innerhalb bestimmter Zeitschlitze erfolgen. Eine weitere Möglichkeit der Priorisierung ist zum Beispiel der Aufbau eine Daisy-Chain.[LIT 4]

Fehlersuchstrategien:

Einige gängige und wichtige Schlagworte sind
- reservierte Testzyklen in Timeslotverfahren
- Builtin Self Test Units
- Wrap-Around Systeme
- Boundary-Scan
- Watchdog-Schaltungen

Diese Aufzählung ist sicher unvollständig. Zur Vertiefung sei auf weitere Literatur [LIT 1] und die folgenden Kapitel, in denen ausführlicher die jeweils verwendeten Verfahren beschrieben werden, verwiesen.

Fehlertoleranzaskpekte eines Bussystems anhand eines praktischen Beispiels: Der XR-III Bus

Anforderungen

Der XR-III Bus ist als Beispiel für einen fehlertoleranten Bus sehr gut geeignet, da die Anforderungen bezüglich der Zuverlässigkeit in einem elektronikfeindlichen Einsatzgebiet sehr hoch sind. Der XR-III Bus wurde als Bussystem für Datenerfassungs- und Prozeßsteuerungsaufgaben konzipiert. Einige der Anforderungen waren:

- Vernetzung von Sensoren und Aktoren
- Lichtwellenleitertauglichkeit
- Hohe Datensicherheit
- Große Datenrate
- Mehrere getrennt verlegte parallele Übertragungswege

Der letzte Anforderungspunkt resultierte aus der Überlegung, daß im industriellen Bereich Leitungsstörungen einesteils nur schwer lokalisierbar sind und andererseits eine Reparatur oder ein Austausch schwierig bzw. nicht möglich sind. Ich möchte an dieser Stelle einige Überlegungen über die Vor- und Nachteile eines Bussystems bezogen auf dessen Fehlertoleranz verschiedener Netztopologien einbringen.

Netztopologien

Ringvernetzung

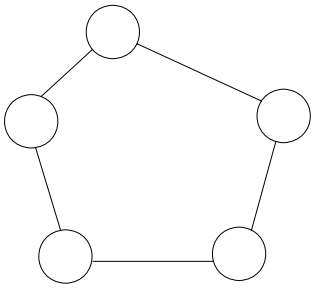

Abb. 5: Ringstruktur

Merkmale:

- Einzelne Knoten
- Point to Point Verbindungen (2 Ports pro Knoten)
- Einfaches Protokoll
- Hohe Ausfallswahrscheinlichkeit – Ein Ausfall eines Knotens führt zu einem Systemausfall

- Relativ einfache Implementierung von Redundanzen
- Unterscheidung zwischen einem aktiven und passiven Empfangsmodus
- Möglichkeit eines Master-Slave Systems mit einer Tokenstruktur

Störungen wirken bei Ringsystemen primär nur auf das jeweilige Segment; Der Fehler bleibt lokal und kann im nächsten Knoten gegebenenfalls korrigiert werden. Es ist nicht möglich, daß räumlich weit entfernt auftretende Störungen einander überlagern. Ein Ausfall einer aktiven Station führt allerdings zu einem Komplettausfall des Systems. Dieselbe Wirkung haben Leitungsunterbrechungen und Kurzschlüsse. Allerdings ist es vom Standpunkt der Zuverlässigkeit oft besser wenn das Gesamtsystem einen Ausfall hat, als wenn Teilsysteme unkontrollierbare Zustände annehmen, die eventuell zu Fehlern führen, die die Sicherheit beeinträchtigen können. Die Lokalisierung eines Leitungsausfalles ist einfacher als bei einer Busverbindung, da das Segment einfach durch geeignete Mittel gefunden werden kann.

Vernetzung mittels einer Sternstruktur

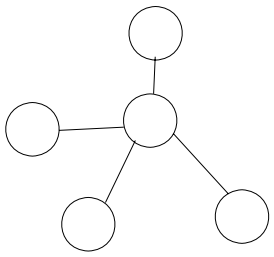

Abb. 6: Sternstruktur

Merkmale:

- Ein zentraler Knoten
- Point to Point Verbindungen
- Kommunikationsintelligenz im zentralen Knoten
- Ausfall des Systems bei Ausfall des zentralen Knotens
- Ausfall einer peripheren Station hat keine Auswirkungen
- Einfache Anbindung neuer Teilnehmer (In den Grenzen der Spezifikationen)

Ein Merkmal der Sternstruktur ist, daß der zentrale Knoten (Master) als Vermittlungsstelle einen hohen Implementierungsaufwand besitzt. Die Ausbaufähigkeit des Netzes hängt von den Spezifikationen des Masters ab. Weiters hängt auch die Datenrate zwischen zwei Slaves von der Kapazität des Masters ab. Ein weiteres Problem dieser Struktur ist der beträchtliche Verkabelungsaufwand.

Vernetzung über einen Bus (Linie)

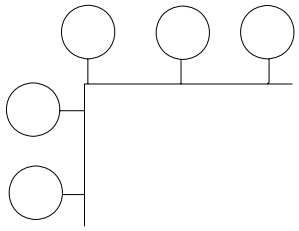

Abb. 7: Busstruktur (Linie)

Merkmale:

- Einzelne Knoten sind an ein Übertragungsmedium angeschlossen
- Notwendigkeit der Busarbitration
- Tokenprotokolle, Random Access Methoden, Timeslot Verfahren
- Singlemaster/Multimaster Systeme
- Durch die Leistung limitierte Leitungslänge
- Ein Teilnehmer kann den Bus blockieren
- Störungen machen sich im gesamten Netz bemerkbar

Bei Bustopologien kann ein Teilnehmer passiv auskoppeln, was keine Beeinträchtigung der Kommunikation der anderen Teilnehmer bedeutet. In Multimastersystemen führt somit auch der passive Ausfall eines Masters zu keinem Fehlerfall. Falls jedoch ein Teilnehmer den Bus auf einen dominanten Pegel festlegt, bricht die gesamte Kommunikation zusammen. Kurzschlüsse, Leitungsbrüche und andere Störungen machen sich im ganzen Netz bemerkbar. Eine Leitungsunterbrechung segmentiert das Netz, wobei die Teile für sich noch funktionstüchtig sein können. Ein wesentliches Merkmal ist die relativ komplexe Arbitration des Bussystems. Hierfür existieren verschiedene Strategien, die aber in jedem Fall einen relativ hohen Implementierungsaufwand bedeuten.

Baumstruktur

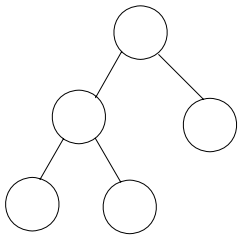

Abb. 8: Baumstruktur

Merkmale:

- Untergliederung von Kopfstationen und Substationen, die ihrerseits wieder Kopfstationen bilden können.
- Point to Point Verbindungen zwischen den Stationen
- Der Ausfall einer Kopfstation führt zum Ausfall der Kommunikation der Substationen

Die Baumstruktur hat starke Ähnlichkeiten mit der Sterntopologie, womit viele Eigenschaften vom Stern übernommen werden können.

Vernetzung mittels einer Totalvermaschung

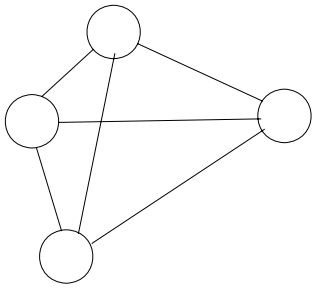

Abb. 9: Totalvermaschung

Merkmale:
- Jeder Knoten ist mit jedem verbunden
- Point to Point Verbindungen
- Hohes Kommunikationsaufkommen
- Geringe Ausfallswahrscheinlichkeit
- Hohe Fehlertoleranzen möglich
- Hohes Kommunikationsaufkommen
- Hohe Kosten

Die Vernetzung mittels Totalvermaschung umgeht jegliche Probleme wie Buszugriffsrechte etc. Eine defekte Verbindung zwischen zwei Teilnehmern kann durch eine Verbindung über einen dritten Teilnehmer ersetzt werden, was eine sehr hohe Fehlertoleranz ermöglicht. Dieser hohe (finanzielle) Aufwand wird vor allem für Systeme mit sehr hohen Sicherheitsansprüchen, wie sie z.B. in der Flug- und Raumfahrttechnik gefordert werden, in Kauf genommen.

Die Topologie des XR-III Bus

Für den XR-III Bus wurde eine Ringstruktur als geeignet erachtet. Dadurch bleiben Fehler lokal. Als Zugriffsverfahren wurde das Master-Slave Prinzip gewählt, da Sensoren kaum mit Masterfunktionen ausgestattet sind und sich damit auch das Protokoll vereinfacht. Durch den fehlenden Overhead des Protokolls wird zusätzlich eine effizientere Datenübertragung erreicht.

Um die Forderung nach der hohen Verfügbarkeit zu erreichen, werden drei parallele Leitungen verwendet (zweifache Redundanz). Ein wichtiger Punkt ist die Verlegung der Leitungen auf verschiedenen Wegen von einem Knoten zum nächsten. Somit können zwei Leitungen ausfallen, ohne daß die Gesamtfunktion beeinträchtigt wird. Außerdem können die Knoten als Repeater eingesetzt werden, die neben der Signalverstärkung auch für die Fehlerüberprüfung und Fehlerbehandlung eingesetzt werden können, wodurch die Fehlertoleranz beträchtlich erhöht werden kann.

Übertragungsmedium und -protokoll

Da sich der XR-III Bus in seiner Spezifikation auf kein Übertragungsmedium festlegt, wurde ein Manchester-Code als Übertragungsprotokoll gewählt. Hierbei wird ein Nullbit des NRZ- (no return to zero-) Codes durch eine Null-Eins-Kombination auf der Leitung repräsentiert, ein Einsbit in den Daten durch eine Eins-Null Kombination (siehe Abbildung 10).

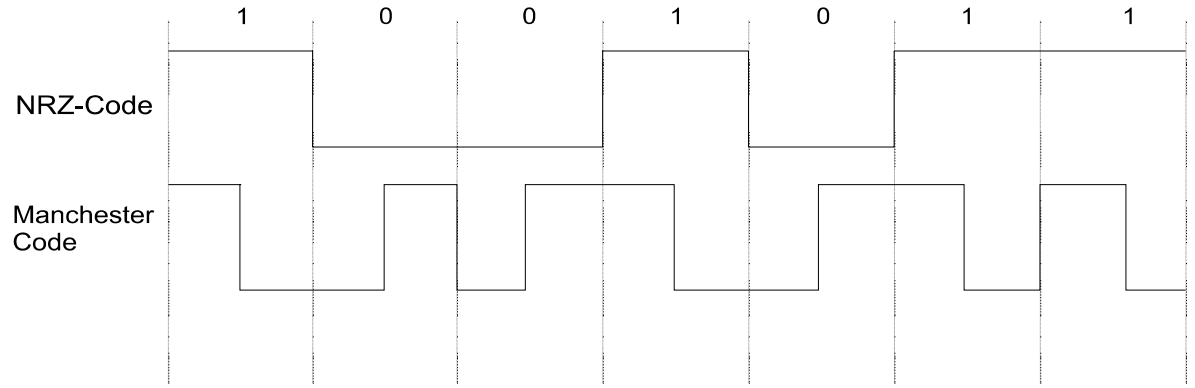

Abb 10: Pegelverläufe bei NRZ- und Manchesterkodierung

Der Vorteil des Manchester-Codes liegt in der Nichtbelastung der Leitung durch einen Gleichanteil, was besonders in der optischen Signalübertragung Vorteile bringt. Allerdings bringt der Code den Nachteil mit sich, daß bei einer unvorhergesehenen Phasenverschiebung um 180 Grad Daten für den Empfänger als korrekt scheinen, obwohl sie in invertierter Form vorliegen. Codes, die diese Möglichkeit ausschließen, sind allerdings vom Implementierungsaufwand deutlich aufwendiger, als der relativ einfache Manchester-Code. Die in jedem Bit vorhandenen Signalübergänge machen eine einfache Synchronisation möglich. Da nach zwei halben Takten auf jeden Fall ein Wechsel des Signalzustandes erfolgen muß, wird eine weitere Erkennung von Übertragungsfehlern und damit eine Verbesserung der Fehlertoleranz ermöglicht. Außerdem ergibt sich durch die Gleichanteilfreiheit des Codes die Möglichkeit der galvanischen Trennung, die besonders in industriellen Einsatzgebieten Vorteile bringt.

Folgende Übertragungsmedien können eingesetzt werden:

- Verdrillte Zweidraht-Leitungen
- Koaxialkabel
- Lichtwellenleiter

Die aus der Sicht der Fehlertoleranz beste Lösung ist sicherlich der Lichtwellenleiter, da dieser nicht auf elektromagnetische Störungen anspricht und eine hohe Bandbreite und damit eine hohe Datenrate erlaubt. Außerdem treten beim optischen Übertragungsweg vor allem stochastische Fehler und seltener die schwerer zu korrigierenden Büschelfehler auf.

Fehlerbehandlungsstrategien

Für die Fehlerbehandlung existieren vier Vorgangsweisen:

- ARQ (Automatic Repeat Request) – Beim Empfang eines fehlerhaften Datenpakets bittet der Empfänger um eine wiederholte Übertragung.
- FEC (Forward Error Correction) – In das Protokoll sind Coderedundanzen eingebaut, die eine Fehlerkorrektur ermöglichen.
- Hybridform – Eine Mischform aus ARQ und FEC.
- Fehlererkennung – Der Empfänger verwirft als fehlerhaft erkannte Datenpakete.

Eine ARQ-Strategie ist in der Implementierung zwar einfacher, allerdings ist ein ARQ-Protokoll nicht echtzeitfähig und für die Anwendung im XR-III Bus nicht geeignet. Da der Verlust neuer Datenpakete während einer Wiederholsequenz eines fehlerhaften Datenpakets kritischer als die Verwerfung ist, muß ein FEC-Protokoll verwendet werden.

Fehlersicherungscode

Die Fehlersicherung soll gegen folgende für einen Bus typische Einflüsse optimiert werden:

- Rauschen
- Intersymbol-Interferenz
- Elektromagnetische Störungen

Diese Einflüsse äußern sich auch in Büschelfehlern, die im Vergleich zu stochastischen Fehlern nur schwer zu korrigieren sind. Gegen Fehler dieser Art könnten gespreizte oder zusammengesetzte Codes verwendet werden, auf die nicht näher eingegangen werden soll.

Da die Fehlertoleranz des XR-III Busses sehr hoch sein soll, muß ein Code eingesetzt werden, der in der Lage ist, Zweifachfehler zu korrigieren. Für diese Aufgabe eignet sich nach [LIT 12] ein BCH(31,21) Code (21 Datenbits und 10 Kontrollbits). Dieser Code hat den Hammingabstand 5, womit zwei Fehler korrigiert und 3 Fehler erkannt werden können. Büschelfehler ab der Länge 4 bleiben unerkannt.

Leitungsredundanz

Leitungsredundanz kann auf folgende zwei Arten implementiert werden:

- Passiv: Es wird immer nur eine Leitung zur Übertragung verwendet; Die redundanten Leitungen bleiben unbenutzt. Im Falle eines Leitungsbruchs oder im Fall einer starken Leitungsstörung wird auf die nächste Leitung umgeschaltet.
- Aktiv: Alle redundanten Leitungen sind ständig in Betrieb. Die Knoten hören alle Leitungen ab, gleichen Laufzeitunterschiede aus und nutzen alle Daten für die Dekodierung.

Für den XR-III Bus wurde aus folgenden Gründen die aktive Variante gewählt:

- Wegfall von Umschaltkriterien – Algorithmen für die Umschaltung von einer defekten auf eine funktionierende Leitung fallen bei der aktiven Variante weg.
- Kein Synchronisationsverlust bei Fehlerfällen – Bei der passiven Variante muß bei einem Leitungsausfall neu synchronisiert werden.
- Einfache Leitungsüberwachung – Durch die ständige Verwendung aller Leitungen werden diese auch ständig überprüft. Um bei der passiven Methode dem Fall vorzubeugen, daß unbenutzte Leitungen aufgrund äußerer Einflüsse unbrauchbar werden ohne Kenntnis davon zu haben, muß eine Leitungsüberwachung implementiert werden.
- Zusätzliche Fehlersicherheit – Durch die Möglichkeit des Vergleichs aller Datenpakete der verschiedenen Leitungen wird die Fehlertoleranz gesteigert.
- Einfache Implementierung – Durch den Verzicht auf Hardware für Umschaltvorgänge kann der Implementierungsaufwand relativ gering gehalten werden.

Coderedundanz vs. Leitungsredundanz

Um Leitungsredundanz einzusparen und trotzdem gleiche Datensicherheit zu gewährleisten, müßten sehr leistungsfähige Codes eingesetzt werden. Diese verlangsamen einerseits das System und bedürfen außerdem kompliziertere Algorithmen zur Dekodierung. Dazu kommt die Tatsache, daß das System bei einem Leitungsausfall lahmgelegt wird, was ja besonders durch eine Verlegung redundanter Leitungen in verschiedenen Wegen präventiert wird.

Schlußfolgerungen

Der Aufbau und die Spezifikationen des XR-III Busses zeigen deutlich, daß die Anforderung „Fehlertoleranz“ an sich zu wenig aussagekräftig ist. Das Design eines Bussystems bezogen auf die Fehlertoleranz muß sich zumindest aus den folgenden Eingangsparametern ableiten:

- Datenrate
- Restfehlerwahrscheinlichkeit, die für die Applikation genügt
- Störungseinflüsse (elektromagnetisch, mechanisch etc.)
- Echtzeitfähigkeit des Systems
- Zuverlässigkeit
- Verfügbarkeit
- Sicherheit
- Kostenfaktoren

Aus diesen Punkten leitet sich dann nach eingehender Berechnung die Beschaffenheit der Fehlertoleranz und des gesamten Designs ab. Einige gefundene Ausgangsparameter bezogen auf die Fehlertoleranz wären:

- Coderedundanz (BCH, Hamming, etc.)
- Leitungscodierung (NRZ, Manchester etc.)
- Übertragungsmedien (Koaxialkabel, Zweidrahtleitung, Glasfaser)
- Leitungsredundanz
- Protokoll (ARQ/FEC, Hybridform, Fehlererkennung)
- Topologie (Ring, Bus etc.)

Wie in jedem Design müssen also die Anforderungen an das System geklärt werden. Es ist sinnlos, für eine Anwendung, die eine Restfehlerwahrscheinlichkeit von z.B. 10^{-9} benötigt, eine Restfehlerwahrscheinlichkeit von 10^{-13} zu implementieren oder ein teures optisches Übertragungsmedium auszuwählen, obwohl das System keinen elektromagnetischen Störungen ausgesetzt ist und die Leitungslänge sehr kurz ist. Fehlertoleranz ist also in dem Ausmaß einzusetzen, wie sie benötigt wird. Dies läßt überdies den Schluß zu, daß ein Top-Down Design für ein fehlertolerantes Bussystem unerläßlich ist, da Bottom-Up Designs keine eindeutigen Schlüsse über Fehlertoleranzaspekte geben können.

Der CAN-Bus

Einleitung

Ursprünglich wurde der CAN-Bus von der Firma BOSCH in Zusammenarbeit mit INTEL für den Einsatz im Automobil entwickelt. Das „Controller Area Network“ ist international durch ISO/DIS 11898 und ISO/DIS 11519-1 genormt, worin sowohl das eigentliche CAN-Protokoll, als auch Aussagen zum physikalischen Übertragungsmedium enthalten sind. Aufrund der großen Störsicherheit und der hohen Datenübertragungsgeschwindingkeit für geringe Entfernungen (bis zu 1 Mbit/s für eine Ausdehnung bis zu 40 Meter) hat sich der CAN-Bus mittlerweile auch in der Automatisierungstechnik einen Platz erworben.

Anforderungen

Der Einsatzbereich Automobil schreibt bereits ein Höchstmaß an Sicherheit vor. Die serielle Kommunikation muß auch in einer stark gestörten Umgebung fehlerlos möglich sein. Problembereiche finden sich im Kfz-Bereich zum Beispiel im Bereich EMV, Fehler können durch Störeinstrahlung (Autoradio, Airbag) oder durch Störabstrahlung (Zündung) entstehen. Aber auch die hohen Temperaturschwankungen, sowie der Wunsch nach billigen, ungeschirmten Kabeln, und natürlich die Tatsache, daß sicherheitskritische Daten übertragen werden müssen, fordern ein sehr sicheres Bussystem.

Die geforderten Antwortzeiten schwanken gerade im Kfz-Bereich sehr stark. Man unterscheidet zwischen der Karosserieelektronik, die zum Beispiel Beleuchtungssteuerung oder Komfortelektronik beinhaltet und wo Latenzzeiten von 1/10 Sekunde durchaus noch akzeptabel sind und dem Motormanagement, wo es einige wenige Steuergeräte gibt, die auf Zykluszeiten im Bereich einiger Millisekunden angewiesen sind.

Grundkonzepte

Der CAN-Bus ist ein multimasterfähiges System, das heißt mehrere CAN-Knoten können gleichzeitig, ohne einen Buscontroller zu benötigen, den Bus anfordern.

Die Übertragung ist nachrichten- und nicht adreßorientiert, während bei den meisten anderen Netzen die Informationen an bestimmte Knotenadressen geschickt werden, setzen die Entwickler des CAN-Busses auf Identifizierung durch Objektnummern. Jede Nachricht erreicht daher alle Knoten, die dann aufgrund der Objektnummer des Datenpaketes entscheiden müssen, ob sie die Nachricht verwerten, oder nicht.

Fehlertoleranz durch Hardwareredundanz

Grundsätzlich ist CAN für die serielle Datenübertragung in einer Bus-Topologie ausgelegt. Als physikalisches Medium ist in ISO/DIS 11898 die Redundanz einer Zweidrahtleitung (RS 485) vorgesehen. Das Signal wird über eine positive und eine negative Leitung differentiell übertragen. Auf diese Weise werden Störungen, die aus zu großem Massepotentialversatz oder von elektromagnetischen Strahlungen herrühren vermieden, oder zumindest verringert.

Ebenso ist in der Norm vorgesehen, daß die Stationen im Falle der Unterbrechung einer Leitung, aus unvorhersehbaren Gründen, die Kommunikation auf Eindrahtbetrieb umstellen können. In diesem Notbetrieb können aber natürlich die Vorteile der differentiellen Übertragung nicht mehr genutzt werden.

Viele CAN-Transceiver bieten außerdem einstellbare Flankensteilheit, wodurch sich, auf Kosten der Übertragungsgeschwindigkeit, die beim CAN-Bus frei wählbar ist, die elektromagnetischen Abstrahlungen reduzieren lassen.

Kein Routing, einfache Diagnose

Aufgrund des Fehlens von Knotenadressen entfällt beim CAN das Problem des Routings, der Sender muß nicht wissen, wer die Information erhalten soll, jede Nachricht erreicht ja alle Stationen, was die Fehleranfälligkeit natürlich ebenfalls stark reduziert. Das erleichtert auch den Einsatz von Diagnosegeräten, das Zuschalten eines weiteren Busteilnehmers bleibt ja für den Rest des Systems unbemerkt. Der Diagnosechip muß also keine Nachrichten umleiten, er empfängt einfach den gesamten Verkehr. Außerdem bedeutet das, daß zum Beispiel der Ausfall eines Displays für die Funktion des Gesamtsystems keine Rolle spielt, die restlichen Stationen, die auf die selben Informationen angewiesen sind, können weiterhin fehlerfrei arbeiten.

Kollisionsvermeidende Busarbitrierung:

Als Zugriffsverfahren verwendet der CAN-Bus CSMA/CA (Carrier Sense Multiple Access with Collision Avoidence). Dazu ist es wichtig, zu wissen, daß der Buspegel zu jedem Zeitpunkt definiert ist. Man unterscheidet zwischen dominanten (Bit = 0) und rezessiven (Bit = 1) Spannungspegeln. Jeder sendende Knoten zieht bei einer „0“ die gesamte Busleitung auf diesen Zustand. (Die Busleitung funktioniert wie ein „Wired-And“).

Jede Nachricht (Frame) besteht aus einem Frame-Identifier, der die Art der zu übertragenden Information angibt. Zum Beispiel könnten die aktuelle Kühlwassertemperatur, oder der Drehzahlstand durch solche Identifier definiert sein. Es können 2048 verschiedene Identifier vergeben werden. Anhand dieser Identifikation erkennen die Stationen, ob die aktuelle Nachricht für sie relevant ist, oder nicht. Der Frame-Identifier legt aber auch die Priorität des Datenpaketes fest. (Siehe weiter unten.)

Die eigentlichen Nutzdaten werden im sogenannten Data-Frame gesendet, ein Frame kann eine ganzzahlige Anzahl an Bytes (8 Bit) enthalten, wobei maximal 8 Bytes erlaubt sind. Jeder Frame beinhaltet eine 15-bit CRC-Prüfsumme.

Das Senden am Bus darf nur erfolgen, wenn kein anderer Teilnehmen gerade aktiv auf das Netz zugreift. Die Übertragung wird durch das Senden eines dominanten Bits eingeleitet. Danach folgt der Frame Identifier und schließlich die Nutzdaten.

Bitcheck

Während des Sendevorganges überprüft der Transmitter ständig, ob das, was er aussendet auch am Bus anliegt. Da die „0“ gegenüber der „1“ dominant ist könnte nämlich ein anderer Busteilnehmer, bei gleichzeitigem Beginn der Übertragung, den Kampf um den Bus gewinnen, wenn ein rezessives Bit von einem, von der der anderen Station gesendeten, dominanten Bit am Bus „überschrieben“ wird. Sobald die am Bus anliegenden Daten nicht mehr mit den gesendeten übereinstimmen, muß sich der betreffende Transmitter sofort vom Bus zurückziehen. Da der Frame-Identifier am Anfang der Nachricht steht, gewinnt bei einem Zugriffskonflikt der Teilnehmer, der den niedrigsten Frame-Identifier aussendet, da dieser mehr (dominante) Nullen enthält.

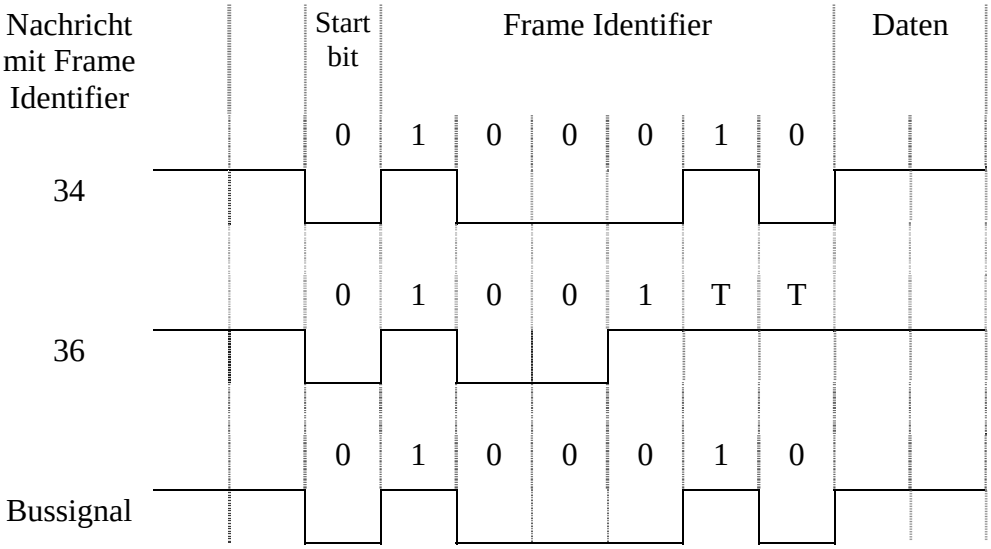

Abb. 11: Dominanz der Signalpegel

In diesem Beispiel (Abbildung 11) dominiert die Nachricht mit dem Identifier 34. Wie bereits erwähnt, prüft jeder Transmitter, ob seine Daten auch am Bus anliegen. Der Transmitter, der versucht, den Frame „36“ über den Bus zu bringen, erkennt nach dem Senden des 5. Bits, daß seine „1“ von der dominanten „0“ der konkurrierenden Einheit überschrieben wurde und überläßt dieser den Bus.

Das bedeutet, daß in jedem Fall eine Nachricht übertragen wird, es kann nie passieren, daß beide zum Abbruch gezwungen werden, der Transmitter, der den niedrigeren Frame-Identifier sendet, erhält den Bus. Diese Art der Busarbitrierung wird daher als kollisionsvermeidend bezeichnet. Durch geeignete Vergabe der niedrigsten Frame-Identifier an die Meldungen, die niedrige Latenzzeiten erfordern, kann man garantierte Zugriffszeiten auf den Bus für hochpriore Nachrichten erreichen. (Die wichtigste Nachricht wartet maximal bis die aktuelle Übertragung zu Ende ist.) Das bedeutet natürlich auch, daß bei starkem Busverkehr Daten mit hohen Identifiern unter Umständen lange auf die Übertragung warten müssen, daher spielt die Vergabe dieser Nummern eine entscheidende Rolle bei der Planung des CAN-Busses.

Auf jeden Fall muß aber vermieden werden, daß zwei Stationen zur gleichen Zeit versuchen, Daten mit derselben Identifikation zu versenden, da das zu einem unlösbaren Busarbitrierungskonflikt führen würde. Diese Situation würde zu Fehlern führen und darf daher nicht auftreten. Auch daher ist die Planung ein wichtiger Schritt zu einem fehlertoleranten CAN-Bus.

Fehlererkennung

CRC-Check

Beim CAN-Bus enthält jedes Datenpaket eine 15Bit Prüfsumme, die vom Transmitter berechnet, mitgeschickt und schließlich von jeder empfangenden Station mit dem selbst berechneten Ergebnis verglichen wird.

Form-Check

Die Daten in einem Frame sind durch ein genau definiertes Format festgelegt. Zum Beispiel legt ein Bit-Stuffing Gesetz fest, daß maximal 5 Bits mit gleicher Polarität auftreten dürfen. Außerdem werden die Datenfelder auf ihre Länge und das korrekte Ende überprüft. Jede empfangende Einheit achtet auf eine Verletzung dieses Formates und meldet diese.

Acknowledgement Check

Der Transmitter erkennt, wenn sein Datenpaket von keiner Station korrekt empfangen wurde.

Fehlerbehandlung

Error aktiv – Error passiv

Jede Station kann sowohl „error-aktiv“ als auch „error-passiv“ sein. Eine „error-aktive“ Station nimmt an, selbst nicht die Ursache für den Fehler zu sein, sondern den Fehler durch einen Leitungsdefekt oder von einer anderen fehlerhaften Station erhalten zu haben. Wenn eine „error-aktive“ Station einen Fehler entdeckt, sendet sie einen „active-error frame“, was bewirkt, daß sämtliche Empfänger dieser Informationen einen Form-Error entdecken und das Datenpaket verwerfen.

Eine „error-passive“ Station sieht sich selbst als Grund des Fehlers und kann daher, falls sie einen Fehler entdeckt, nur einen „passive-error frame“ versenden, der jedoch von einer „error-aktiven“ Station überschrieben werden kann, falls diese keinen Fehler bemerkt. Wenn das der Fall ist, wird das betreffende Datenpaket nicht verworfen. Außerdem muß eine „error-passive“ Station nach der Übermittlung eines Paketes sieben rezessive Bits aussenden und warten, ob eine andere Station ein Datenpaket senden will. Wenn das der Fall ist, empfängt die „error-aktive“ Station dieses Paket, anstelle das eigene (möglicherweise falsche) an den Bus zu legen.

Jeder Busteilnehmer verwaltet eine Fehlerstatistik in der die Anzahl und Art der Fehler festgehalten werden und abhängig von dieser kann die Station den Zustand von „aktiv“ nach „passiv“ wechseln.

Bus-Off

Werden von einer Station zu viele Fehler entdeckt, kann diese sich auch völlig vom Bus abkoppeln, was bedeutet, daß sie keinerlei Daten mehr senden oder empfangen kann. Eine Station die nur „error-passiv“ ist, kann jedoch noch empfangen, sowie (eingeschränkt) senden. Ein defekter Teilnehmer zieht sich also schrittweise vom Bus zurück.

Im Normalfall haben in einem CAN-Bus sämtliche Busteilnehmer den Status „error-aktiv“, was sicherstellt, daß ein Datenpaket entweder von allen, oder von keiner Station empfangen wird.

Fehlerkorrektur

Die Fehlerkorrektur funktioniert beim CAN-Bus nach einem einfachen Prinzip: Automatische Wiederholung. Das fehlerhafte Paket wird noch einmal übertragen.

Fazit

Der CAN-Bus ist ein Feldbus der für eine Umgebung entwickelt wurde, in der starke Störeinflüsse vorhanden sind und hohe Übertragungsraten über kurze Strecken gefragt sind. Durch die implementierten Fehlerbehandlungsstrategien erreicht man eine Hamming-Distanz von 6, daß heißt, daß bis zu 5 Bitfehler innerhalb einer Nachricht erkannt werden.

$$\text{Hamming-Distanz} - 1 = \text{Anzahl der erkennbaren Bitfehler}$$

Die Fehlerwahrscheinlichkeit wird beim CAN-Bus mit 10^{-13} angegeben.

Feldbus nach MIL-STD-1553B

Einleitung

MIL-STD-1553 ist ein Standard des Department of Defense (DoD), der die elektrischen, mechanischen und Timing-Spezifikationen eines Kommunikationsnetzwerkes spezifiziert. Der MIL-STD-1553 Bus ist ein sehr zuverlässiger Bus, mit einer Fehlerrate von unter 10^{-7}. Haupteinsatzgebiet ist der Flugzeugbereich (sowohl im zivilen als auch im militärischen Bereich) außerdem wird er in der Raumfahrt, aber auch in Unterseebooten oder Satelliten eingesetzt. Der MIL-STD-1553 ist ein Hochgeschwindigkeitsbus mit einer Übertragungsrate von 1 Mbit/s. Die Ausdehnung des Busses ist auf 100 Meter beschränkt, die Verkabelung sollte mit geschirmten und verdrillten Zweidrahtkabeln erfolgen.

Parallelredundanz

Ein System nach MIL-STD-1553 verfügt mindestens über 2 voneinander unabhängige Busse. Im Regelfall werden sämtliche Informationen über den Primärbus übertragen. Nur wenn dieser ausfällt tritt der Sekundärbus in Kraft, abhängig von der eingesetzten Strategie kann ein fehlgeschlagener Transfer über diesen zweiten Bus wiederholt werden.

Hierarchische Struktur

Der Bus ist hierarchisch aufgebaut, ein Buscontroller (BC) steuert bis zu 31 Remote-Terminals (RT's). Jedes Remote-Terminal verfügt über eine eindeutige Adresse. Ein Remote-Terminal kann 30 verschiedene Nachrichten senden und wiederum 30 verschiedene empfangen. Die Nummer der Nachricht wird durch die Subadresse angegeben. Der Buscontroller überwacht das Senden und Empfangen von Informationen über den Bus, nur er kann eine Übertragung einleiten. Die Rolle des Buscontrollers übernimmt normalerweise eine der mächtigeren Einheiten am Bus, zum Beispiel eine Steuerungseinheit.

In einem typischen MIL-STD-1553-Bus hat der Buscontroller eine vordefinierte Sequenz von Befehlen, die er zyklisch wiederholt. Diese Befehlsfolge wird als „Buslist" bezeichnet. Die Anordnung und Häufigkeit der Befehle muß mit Rücksicht auf die zu übertragenden Informationen gewählt werden, zum Beispiel werden Informationen über Tragflächenvibrationen wesentlich häufiger gesendet werden, als der aktuelle Treibstoffstand.

Der Buscontroller kann vier verschiedene Arten des Nachrichtenaustausches einleiten:
- Die Kommunikation zwischen zwei Remote-Terminals
- Eine Nachricht eines Remote-Terminals an den Buscontroller
- Eine Nachricht des Buscontrollers an ein Remote-Terminal
- Statusmeldungen und Kommandos, die der Verwaltung des Busses dienen.

In jedem Fall muß der Buscontroller die zu steuernden Einheiten anweisen, die Kommunikation vorzubereiten, bei der Übertragung zwischen zwei Remote-Terminals muß nur der Controller den Sender und den Empfänger kennen.

Die Art der zu übertragenden Information wird durch ein Sync-Muster bestimmt, das den Informationen vorangestellt wird. Man unterscheidet zwischen:
- Datenwörtern (sie enthalten die Nutzdaten)
- Befehlswörtern (sie enthalten Anweisungen des Buscontrollers)
- und Statuswörtern.

Ein Wort kann im MIL-STD-1553B immer 16 Bits eine Nachricht kann 32 Wörter enthalten.

Fehlererkennung

Beim MIL-STD-1553B-Bus unterscheidet man zwischen zwei Gruppen von Fehlern:
- Fehler, die von einem Remote-Terminal gemeldet werden, und
- Fehler, die der Buscontroller erkennt

Fehlererkennung durch zyklische Selbsttests

Ein Remote-Terminal kann einen Fehler durch ein spezielles Fehlerbit im Statuswort senden, daß es an den Controller sendet. Der Buscontroller kann ein Remote-Terminal durch ein spezielles Kommando anweisen, einen Selbsttest durchzuführen, was in einer typischen Anwendung in regelmäßigen Abständen durchgeführt wird.

Ein Buscontroller muß Bitlevel Fehler und Wortfehler erkennen können. Erstere entstehen zum Beispiel durch ein unzulässiges Sync-Muster oder durch eine Parity Fehler. Außerdem können Wörter zu viele oder zu wenige Bits enthalten. Wortfehler entstehen durch unzulässige Pausen zwischen zwei Wörtern oder, wenn eine Wortanzahl über den Bus gesendet wird, die nicht mit der angekündigten übereinstimmt.

Der Buscontroller verwaltet eine Fehlerstatistik, er kann zum Beispiel ein Timeout bei der Kommunikation mit einem Remote-Terminal erhalten, oder aber bemerken, daß ein bestimmtes Terminal regelmäßig Fehler am Bus verursacht.

Fehlerbehandlung

Da es sich beim MIL-STD-1553B-um einen Bus mit mindestens Parallelredundanz handelt, gibt es mehrere Möglichkeiten auf einen aufgetretenen Fehler zu reagieren: Je nach Wichtigkeit der Nachricht und Art des Fehlers kann der Buscontroller mehrere Strategien einsetzten. Er kann

- versuchen, die Nachricht am selben Bus noch einmal zu senden
- versuchen auf den anderen Bus auszuweichen

- den Fehler ignorieren und mit der nächstfälligen Übertragung fortfahren. (z.B. bei Nachrichten mit niedriger Priorität)

Im Normalfall wird eine fehlgeschlagene Übertragung nicht beliebig oft wiederholt werden. Der Buscontroller muß die Anzahl der Wiederholungen, mit Rücksicht auf die Busbelastung und auf die Antwortzeiten der anderen Nachrichten beschränken.

Bus-Off

Die Kommandos des Buscontrollers im Bereich Fehlerbehandlung beschränken sich aber nicht nur auf die Anweisung zum Selbsttest oder die Wiederholung von Übertragungen. Er kann, je nach Häufigkeit und Gewichtigkeit der Fehler auch das Abschalten von Remote-Terminals befehlen. Im Extremfall kann der Buscontroller sogar einem Remote-Terminal (das die entsprechenden Fähigkeiten besitzt) auch anweisen, seine Rolle zu übernehmen!

Literaturverzeichnis

LIT 1 Prof. Dietmar Dietrich:
Fehlertolerante Prozeßrechensysteme
Institut für Computertechnik der TU-Wien, 1996

LIT 2 Prof. Dietmar Dietrich:
Bussysteme und Rechnerkommunikation in der Prozeßautomatisation
Institut für Computertechnik der TU-Wien, 1996

LIT 3 Alois Goiser:
VO: Digitale Spread-Spectrum Systeme
Institut für Allgemeine Elektrotechnik und Elektronik der TU-Wien, 1996

LIT 4 Prof. Richard Eier:
Mikrocomputer
Institut für Computertechnik der TU-Wien, 1994

LIT 5 MIL-STD-1553B:
Interface Standard for Digital Time Division Command/response
Multiplex Data-Bus Department of Defense, 1978

LIT 6 Georg Färber:
Bussysteme
Oldenbourg, München-Wien,1984

LIT 7 iNet'92 Tagungsband
Sindelfingen bei Stuttgart, 1992

LIT 8 iNet'93 Tagungsband
Karlsruhe, 1993

LIT 9 Peter Kohlendorfer:
Diplomarbeit: Feldbusknoten für paket- und
verbindungsorientierte Datenübertragung
Wien, 1995

LIT 10 Elektronik Report Ausgabe 9A,
Bussysteme: Dezentral mit CAN
Wien-1995

LIT 11 Prof. Dr. Ing. Georg Färber:
Feldbussysteme – Oberseminar Prozeßrechentechnik WS 94/95
Lehrstuhl für Prozeßrechner TU-München, 1994

LIT 12 Thilo Sauter:
Diplomarbeit: XRIII-BUS und XR-III Buscontroller
Wien 1992

LIT 14 Alfredo Baginski, Martin Müller:
InterBus-S Grundlagen und Praxis, Hüthing Verlag
Heidelberg, 1994

LIT15 Dr. techn. Joachim Swoboda:
Codierung zur Fehlerkorrektur und Fehlererkennung
Oldenbourg Verlag Wien, 1973

LIT16 Prof. Richard Eier:
Datensicherung
Institut für Computertechnik der TU-Wien, 1994