Zusammenfassung

Dezentrale Automation VO

Diese Zusammenfassung ist auf Basis [dieser](https://www.informatik-forum.at/forum/index.php?thread/101045-zusammenfassung/&postID=797686#post797686) Zusammenfassung aus dem Jahre 2013 von Bernhard B. entstanden. So wie das Original erhebt auch diese Lernhilfe keinen Anspruch auf Vollständigkeit oder Korrektheit. Es empfiehlt sich die Folien neben dieser Zusammenfassung trotzdem durchzugehen, da keine Bilder aus den Folien eingefügt sind und diese beim Verstehen doch helfen.

Solltest Du gerade dabei sein die nächste Version zu erstellen, ehre den Wunsch des ursprünglichen Urhebers und nenne die vorherigen Autoren, sodass ein Versionsverlauf entsteht.

Zum einfachen und schnellen Bearbeiten und aufgrund des fehlenden LaTex Sourcecodes, habe ich V2 in Word geschrieben und dabei das Roboto-Font-Set verwendet. Einige Kapitel und Absätze konnten aus dem Original 1:1 übernommen werden, andere Themengebiete sind dagegen neu hinzugekommen und wiederum andere weggefallen. Außerdem ist der Teil zu PROFIBus recht unübersichtlich – in der nächsten Überarbeitung sollte hier angesetzt werden.

Viel Erfolg beim Lernen und Glück bei der Prüfung!

Versionsverlauf:

15.03.2020 Version 2, Moritz C.

03.05.2013 Version 1, Bernhard B.

Foliensatz Grundlagen:

Prozessdefinition nach DIN 66201:

Ein technischer Prozess ist ein Prozess, dessen physikalische Größen mit technischen Mitteln erfasst und beeinflusst werden.

Was sind die 3 Aufgaben eines Automationssystems?

1. Messen, Steuern, Regeln
2. Überwachen (Anzeigen), Melden (Alarmieren)
3. Optimieren

Im Gegenzug dazu ist der Mensch für folgendes Zuständig:

1. Bedienen
2. Leiten
3. Führen

Automationsebenen & deren Geräte (& Programme):

Nach ISO 16484-2: Management, Automation, Field:

* Management und Leitebene:  
  Aufgaben: Aufgabenverwaltung, Logistik, Anlagenoptimierung, Datenarchivierung, Produktionsplanung  
  Geräte: Mainframes, Workstations & PCs  
  Programme: ERP (Enterprise Resource Planning, z.B. SAP), MES (Manufacturing Execution System, z.B. grafische Plantafel), SCADA (Supervisory Control and Data Acquisition, z.B. Siemens SIMATIC WinCC)  
  Anforderungen: hoher Datendurchsatz, Fremdsystemankopplung, Zeitverhalten in Minuten  
  Verbindungen: WAN, GAN, Internet
* Automationsebene:  
  Aufgaben: Überwachung, Steuerung, Regelung, Prozessdatenverarbeitung, Anlagenvisualisierung, Bedienung, Diagnose   
  Geräte: Industrietaugliche Prozessstationen (IPC/SPS, Anzeigetafeln, Bedienstationen, Scanner, BDE-Geräte  
  Anforderungen: Reaktionszeit < 100ms, mittlere Datensätze, fixe & ändernde Teilnehmer  
  Verbindungen: systemintegrierte Busse, Teilnehmer und Nachrichtenorientierte Feldbusse
* Feldebene:  
  Aufgaben: Erfassung von Signalen, Daten-Vorverarbeitung & -Verteilung, zunehmend auch Steuerung und Regelung von Teilprozessen  
  Geräte: Sensoren & Aktoren, Regler, Antriebstechnik & -steuerung, Analysegeräte  
  Verbindungen: Feldbussysteme

Interaktionsprinzipien:

* Client/Server (Unicast)
* Publish Subscriber(Multicast)
  + Pull-Modell
  + Push-Modell
* Producer/Consumer (Broadcast)

Fertigungsverfahren nach DIN 8580:

Umformen, Trennen, Fügen, Beschichten, Ändern von Stoffeigenschaften.

Arten der direkten Prozesskopplung:

1. Zustandserfassung (offene Kopplung): Rechner erzeugt Prozesszustandsabbild und Protokollierung der Prozessdaten, Mensch steuert mit diesen und dem Ziel den Prozess.
2. Prozessbeeinflussung (offene Kopplung): Vorstufe der vollständigen Automatisierung; Anlagenzustand und Ziel vom Menschen erfasst, er gibt über geeignete Schnittstelle Lenkzustand an Rechner vor. Dieser setzt die Stellsignale und protokolliert.
3. Closed Loop (geschlossene Kopplung): Mensch kennt Ziel und gibt Rechner über Terminal Anweisung. Rechner erfährt Messzustand, steuert den Prozess und erzeugt Protokoll. Mensch ist kein Bestandteil der Wirkungskette mehr.

Arten der indirekten Prozesskopplung:

1. Offline: Rechner nicht Teil der Wirkungskette; Mensch steuert Prozess auf Basis des Anlagenzustandes, des Ziels und des Betriebsprotokolls des Rechners und gibt dem Rechner Zustandsdaten weiter, welche dieser protokolliert.
2. Inline: Mensch erfasst Anlagenzustand und gibt sie an Rechner weiter, wartet während Rechner aus Ziel (Anweisung) Steuerdaten generiert und protokolliert und steuert auf Basis der Steuerdaten den Prozess.

Verfügbarkeit Formel und Einheit:

|  |  |
| --- | --- |
| V… Verfügbarkeit [Einheitenlos] MTBF… mean time between failure [Zeit] MTTR… mean time to repair [Zeit] |  |
|  |  |

Heiße Reserve:

“statische Redundanz“: es existiert (mindestens) ein statisch mitlaufender Prozessrechner, dem dieselben Messdaten zur Verfügung gestellt werden. Mittels eines m aus n Voters wird überprüft ob mindestens m aus n Prozessrechner das gleiche Ergebnis liefern. Erst dann wird eine Operation durchgeführt.

Kalte Reserve:

„dynamische Redundanz“: es gibt einen führenden Rechner und n mitwirkende Rechner im Stand-by-Betrieb; Falls ein Fehler/Ausfall durch z.B. online Prüfprogramme erkannt wurde wird auf einen der Rechner im Stand-by umgeschaltet.

4 Arten der Redundanz

1. Hardware Redundanz (kalte und heiße Reserve, siehe oben)
2. Software Redundanz: Betrieb mehrerer verschiedener Programme zur Realisierung einer Funktion
3. Messwert Redundanz: mehrfache Messung von Messgrößen
4. Zeitliche Redundanz: mehrfache Telegrammübertragung mit Vergleich und Quittungsverkehr

Was bedeutet "Industrie 4.0"?

Dies steht für die vierte Industrielle Revolution. Diese Zeichnet sich durch die Verwendung intelligenter und digital vernetzter Systeme aus. Ziel ist weitestgehend selbstorganisierte Produktion zu ermöglichen: Menschen, Maschinen, Anlagen, Logistik und Produkte kommunizieren und kooperieren direkt miteinander.

Wie kann man die 3 Phasen von technischen Systemen einteilen? Welche Kurve beschreibt dieses Verhalten?

Badewannenkurve

Welche Aufgabe hat der Network Layer (Schicht 3 im Schichtenmodell)? Warum wird die Schicht in manchen Automationsnetzen nicht implementiert?

Die Aufgaben der Vermittlungsschicht sind Routing, Flusskontrolle und verbindungsorientierte und verbindungslose Dienste. Weil in manchen Netzwerken Routing und Flusskontrolle nicht benötigt werden (?).

Foliensatz 2: Sensor/Aktor Ebene

Sensoren zur Entfernungsmessung:

1. Optische Sensoren mit Triangulationsprinzip: Ein Laserstrahl wird punktförmig auf das Messobjekt gerichtet und der Eintrittswinkel des zurückgestreuten Strahls mit einer Optik durch ein PSD (Position Sensitive Device) erfasst.
2. Sonarsensor: ein Schallgeber sendet kurz kegelförmige Schallwellen aus, welche von senkrecht zur Ausbreitungsfläche stehenden Gegenständen reflektiert werden. Durch Messung der Zeitdauer des Schalls vom Absenden bis zum Empfangen wird gemessen und ausgewertet.

Kapazitiver Näherungsschalter:

Vereinfacht bestehend aus zwei Komponenten: Einer Sensorelektrode, welche die Änderungen im elektrischen Feld vor seiner aktiven Zone erkennt, denn es wird die Kapazität zwischen der Elektrode und dem Erdpotential gemessen. Zusätzlich gibt es noch einen RC-Oszillator. Wird ein Gegenstand der Sensorelektrode angenähert ändert sich die Kapazität und beeinflusst damit die Schwingungsamplitude des Oszillators, woraufhin ein nachgeschalteter (Schmitt) Trigger schaltet. Dieser kann oft mit einem Potentiometer eingestellt werden.

Induktiver Näherungsschalter:

Meist sendet eine Spule als Bestandteil eines Schwingkreises mittels eines Ferritkerns ein elektromagnetisches Feld zur aktiven Fläche aus. Ein sich in der aktiven Zone befindliches Objekt entzieht dem Schwingkreis Energie, die Oszillatorspannung sinkt, ein Komparator detektiert dies und schaltet.

Drehwinkelgeber (inkl. Skizze):

Drehwinkelgeber werden zur Erfassung von Lageänderungen (zur Bestimmung der Wegstrecke und Wegrichtung) oder Winkeländerungen (zur Erfassung der Winkeländerung) eingesetzt. Im Gegensatz zu einem Absolutgeber ist mit einem Drehwinkelgeber die Bestimmung der absoluten Position nicht möglich! Das Funktionsprinzip eines Drehwinkelgebers ist sehr einfach erklärt: Eine Lichtquelle (zumeist eine Infrarot-Diode) sendet einen Lichtstrahl aus, der auf eine mit Schlitzen versehene Scheibe (Codescheibe) trifft. Hinter der Codescheibe sitzt zumeist eine Blende und eine Fotodiode, die zur Detektion des auftreffenden Lichtstrahls dient. Bewegt sich die Scheibe dann wird der Lichtstrahl zwischen Sender (Infrarot-Diode) und Empfänger (Fotodiode) periodisch moduliert. Dieses periodisch modulierte Sinussignal wird in einen Gray-Code umgesetzt und kann mit einer SPS ausgewertet werden.

Magnetischer Näherungsschalter (inkl. Skizze)?

Magnetische Näherungsschalter (auch Reed-Schalter genannt) sind Schalter, die durch ein Magnetfeld aktiviert werden. Das Funktionsprinzip eines solchen Schalters ist sehr schnell erklärt: In einer Glasampulle sind zwei ferromagnetische Schaltzungen eingeschmolzen, die sich nur wenige m überlappen. Im Ruhezustand berühren sich die zwei Schaltzungen nicht. Wird nun ein Magnetfeld an den Reed-Schalter angelegt, dann richten sich die ferromagnetischen Materialien in Richtung Nord - und Südpol aus. Das heißt: Die beiden Schaltzungen ziehen sich gegenseitig an, der Schalter wird geschlossen.

AS-Interface 2.0 Eckdaten <Daten für 3.0>:

ASi: Actuator Sensor Interface

* Beliebige Netzstruktur max. 100m dann Repeater
* Übertragungsmedium ist eine nicht geschirmte und nicht verdrillte Zweidrahtleitung (braun ASi pos., blau ASi neg) für Daten und Energie (schwarz bis 30V AC, rot bis 230V DC)
* Übertragungsprinzip: Single Master System mit zyklischem Polling, eine fixe Adresse pro Slave, max. 31 Slaves pro Strang und je max. 4 Sensoren/Aktuatoren pro Slave, max. 124 digitale Ein/Ausgänge <max. 62 Slaves, max. 248 digitale Eingänge, 186 Ausgänge>
* Zykluszeit passt sich an Anzahl der angeschlossenen Slaves an: 6 Slaves 1ms, 31 Slaves 5ms <62 Slaves: 10ms>
* Je 4 Bit Nutzdaten von Master zu Slave und umgekehrt

ASi-Protokoll:

Es handelt sich bei ASi um ein Master-Slave System. Das heißt, Master und Slaves arbeiten selbstständig miteinander zusammen, so dass sich das System an seinen Schnittstellen wie ein herkömmlicher Kabelbaum verhält. Es kommt in der untersten Steuerungsebene zum Einsatz und verbindet Sensoren/Aktoren mit der speicherprogrammierbaren Steuerungen. Ziel der ASi Entwicklung war eine Minimierung des Verdrahtungsaufwandes, bei gleichzeitiger Funktionserweiterung.  
Jeder Peripherie, die an den ASi-Bus angebunden werden soll, wird ein Slave-Chip vorgeschalten (wenn sie nicht bereits Asi kompatibel ist, also der Slave-Chip schon verbaut), welcher als Koppeleinheit zwischen Peripherie( Sensoren, Aktoren, etc.) und ASi-Leitung fungiert. Der Prozessrechnersteuerung (z.B. SPS) wird ein ASi-Master vorgeschalten, der den Datenverkehr auf der ASi-Leitung selbstständig organisiert und als Koppeleinheit zu übergeordneten Systemen fungiert.  
Der Master ruft alle Slaves zyklisch mit ihrer Adresse auf und tauscht mit ihnen standardisierte Nachrichten aus:  
Verlauf: Masteraufruf -> Masterpause -> Slaveantwort -> Slavepause; Dauer ca. 150 µs, dabei Masteraufruf (14 Bitzeiten): Startbit (0), SteuerBit, Adressbit 4, A3,A2,A1,A0,Information 3, I2, I1, I0, ParitätsBit, Endebit (1)  
Masterpause 3 bis 10 Bitzeiten lang  
Slaveantwort (7 Bitzeiten): ST (0), I3, I2, I1, I0, PB, EB(1)  
Slavepause (1-2 Bitzeiten)  
Dabei gibt es einmalige Übertragungen, wo bei Fehlerhaften oder ausbleibenden Antworten diese sofort gemeldet wird und keine Wiederholung stattfindet (z.B. Slave-Suche, oder wiederholbare Übertragunen)  
Je Strang können insgesamt 31 Slaves (5 Addressbits) angesprochen werden. Die Adresse 0 wurde dabei für die Sonderfunktion der automatischen Adressvergabe reserviert. An jeden dieser 31 Slaves können max. 4 Sensoren/Aktoren angeschlossen werden.   
Da die ASi-Leitung sowohl für die Übertragung der Daten als auch für die Versorgung der Komponenten zuständig ist, wird das Modulationsverfahren APM (alternierende Pulsmodulation) eingesetzt, welches gleichstromfreie Signale erzeugt   
Bei der alternierenden Pulsmodulation wird das zu übertragende Signal zuerst MAN Codiert (bei der Änderung des Grundsignals eine Phasenumtastung vorgenommen -> gleichstromfrei!). Dann werden die positiven und negativen Anteile des MAN Codierten Signals auf die zwei ASi Drahte aufgeteilt. Das Signal kann mithilfe zweier Komparatoren wiederhergestellt werden.  
Das ASi-Protokoll ist so störungssicher, dass bis zu drei Fehler pro Nachricht sicher erkannt werden können. Dabei wird das Signal beim jeweiligen Empfänger auf Störimpulse untersucht. ASi Leitungen benötigen keinen Leitungsabschlusswiderstände.

Phasen der ASi Master Ablaufkontrollebene:

Ablaufkontrollebene besteht aus: Nutzdaten (Eingangs-, Ausgangs-, Parameterdatenfeld), Konfigurationsdaten (-abbilder: E/A Konfig, ID.Codes; Liste erkannter Slaves LES, Liste aktivierter Slaves LAS), Projektierungsdaten, Flags über Masterzustand

* Initialisierung (Offline)
* Anlaufbetrieb
  + Erkennung:   
    Einlesen von EA- Konfiguration und ID-Codes; wenn oke, dann Slave in LES eintragen EA und ID speichern, das für alle Slaves
  + Aktivierung  
    der Datenaustausch wird mit allen Slaves getestet, ggf. werden Slaves aus LES gelöscht
* Zyklischer Normalbetrieb
  + Datenaustausch:  
    Slaves werden nacheinander mit Daten aufgerufen und eine Antowrt erwartet, nach dem dritten Fehler werden Slaves aus LAS und LES entfernt
  + Management:  
    hier werden azyklische Nachrichten verwendet; pro Zyklus kann nur ein Masteraufruf durchgeführt werden
  + Aufnahme:  
    am Ende jedes Zyklus wird nach neu hinzugekommenen Slaves gesucht, wenn ein neuer Slave antworte wird in nächster Aufnahmephase ID ermittelt und LES aktiviert (LAS adaptiert), dann das Eingangsbild aktualisiert

Erklären sie ASi Safety at Work:

Bei ASi safety at work werden in den ASi-Strang neben normalen Slaves, safety Slaves und ein ASi-Sicherheitsmonitor eingebunden. Jeder sicherheits Slave besitzt dabei eine eindeutige 8 x 4bit Codetabelle zur eindeutigen Identifizierung. Beim sogenannten „Teach-In“ liest der Sicherheitsmonitor die Codetabellen aller sicheren Slaves ein, um diese im Betrieb kontinuierlich überwachen zu können. Im Betrieb erfolgt ein permanenter Vergleich der erwarteten und der gesendeten Codetabellen. Wenn der Vergleich negativ ausfällt (die erwarteten Codetabellen stimmen nicht mit den gesendeten Codetabellen überein), dann erfolgt die Abschaltung durch den Sicherheitsmonitor. Response times: worst case 40ms, avg. 10ms

Was ist ein ASi-Koppelmodul?

In einem ASi-Koppelmodul befindet sich ein ASi-Slave, der es ermöglicht 4 Sensoren/Aktoren an den ASi-Bus anzubinden. Dadurch können nicht-ASi-kompatible Sensoren/Aktoren an den ASi-Bus angebunden werden.

Nennen Sie Punkte zur ASi-Datensicherung:

Es gibt ein paar einfach zu überprüfende Punkte, mit denen relativ rasch festgestellt werden kann, ob ein ASi-Telegramm gültig ist, oder nicht. Diese sind:

* Startbit/Endebit: der erste Impuls ist immer negativ, der letzte Impuls ist immer positiv
* Alternation: zwei aufeinanderfolgende Impulse haben immer unterschiedliche Polarität
* Impulspause: zwischen zwei Impulsen innerhalb eines Telegramms, maximal eine Impulslänge Pause
* Informationsgehalt: in der zweiten Bithälfte muss stets ein Impuls liegen
* Paritätsbit: gerade Paritätsüberprüfung
* Aufruflänge: nach dem Endebit des Masters kommt kein weiterer Impuls

Gerade Parität:

Das Paritätsbit wird so gesetzt, dass das Nachrichtenwort (inkl. Paritätsbit) auf eine gerade Anzahl von Einsen kommt. Zum Beispiel Nachrichtenwort aus 4 Datenbits + *1 Paritätsbit*: 0100*1*

Foliensatz 3: IEC 61131 & IEC 61499

Was ist eine SPS (nach IEC 61131-Programmable Controllers)?

Digital arbeitendes, elektronisches System für industrielle Umgebung; verwendet programmierbaren Speicher zur internen Speicherung, um besondere Funktionen auszuführen und steuert durch digitale oder analoge Ein und Ausgänge Maschinen und Prozesse.  
Besteht aus CPU (+ Watchdog), Ein/Ausgabeverwaltung, RAM & EEPROM (beide mit eigenem parity controller) an einem Bussystem.

Zyklus einer SPS & charakteristische Zeiten:

Eingänge Lesen -> Bearbeiten des Programms -> Bearbeiten der Kommunikationsanforderungen -> Selbstdiagnose der CPU -> Schreiben der Ausgänge

Reaktionszeit ist im Worstcase: tR = 2 \* (tZ, tE, tA)  
tE… Eingangsverzögerung  
tA… Ausgangsverzögerung  
tZ… Zykluszeit = Summe aller internen Bearbeitungszeiten von Anweisungen: ∑tI

Was definiert IEC 61131-3

* Struktur des Steuerungsprojekts (Konfiguration, Ressource, Programme, Tasks)
* Programmstruktur (Organisationsbausteine, Funktionen, Funktionsbausteine)
* Standardfunktionen und -bausteine
* Datentypen und-formate (-> direkte [bit, byte, word, doubleword] Addressierung)
* Programmiersprachen

Programmiersprachen für SPS nach IEC 61131-3:

* Kontaktplan (KOP): graphische Programmiersprache, ähnelt elektrischen Schaltplänen
* Funktionsplan (FUP): simuliert Datenfluss
* Anweisungsliste (AWL): stackorientierte Abarbeitung, Assambler-ähnlich
* Strukturierter Text (ST): Struktur, Syntax PASCAL ähnlich
* Ablaufssprache (AS): Schritte, Transitionen

IEC 61499:

Definiert verteilte Architekturen/Applikationen, Functionblocks, Ereignis und Dateninterfaces, ist komponentenorientiert und offen: portable, interoperable, configureable

Funktionsblocks nach IEC 61499:

In IEC 61499 sind Funktionsbausteine für industrielle Leitsysteme spezifiziert. Dabei gibt es zwei Verschiedene Arten:

* Simple function block: besteht aus internen variablen, Algorithmus (auch mehrere), und Execution Control Charts (ECC); die internen Daten sind von außen nicht zugänglich, Algorithmen in beliebiger Sprache  
  ECCs sind state-machines; wenn ein Event auftritt kann die ECC einen Algorithmus starten: anhand Eingangsdaten werden dann neue Ausgangsdaten erzeugt, welche durch ein Output-Event signalisiert werden und so anderen Funktionsblöcken als Input dienen können
* Complex function block: „Netzwerk“ aus verbundenen (simple oder complex) Funktionsblöcken; Regeln: kein „splitten“, „mergen“ von Events, kein „mergen“ von Daten

Aufgaben von ECCs nach IEC 61499:

ECCs sind State Machines welche Events bearbeiten und so festlegen wie sich ein Functionblock bei bestimmten Events verhält indem sie (unterschiedliche) Algorithmen starten.

Phasen eines Funktionsblocks:

Setzen der Eingangswerte -> Eintreffen der Ereignisse -> Aktivieren der Ausführungskontrolle -> Aktivieren des Algorithmus -> Erzeugung von Ausgangsdaten -> Erkennen des Endes des Algorithmus durch Ausführungskontrolle -> Weitergeben der Daten an Ausgänge -> Erzeugen des Ausgangsevent

Service Primitives bei IEC 61499 (Folien S. 46):

Application initiated transactions: Anwendung schreibt zuerst Ergebnisse zur Ressource, dann Einlesen der Inputs von Ressource  
Resource initiated transactions: Ressource schreibt Inputs zur Anwendung, dann liest sie Ergebnisse von Anwendung

Foliensatz 4: PROFIBus, PROFINet

Welche drei Profibus-Varianten gibt es und wo werden sie eingesetzt?

* Profibus FSM (Fieldbus Message Specification): wurde früher im Bereich der allgemeinen Automatisierung eingesetzt, hat aber mittlerweile an Stellenwert verloren und wird kaum mehr verwendet.
* Profibus DP (Dezentrale Peripherie): wird zur Ansteuerung von Sensoren/Aktoren in der Fertigungsautomatisierung eingesetzt.
* Profibus PA (Prozess-Automation): wird vorallem in der Prozessautomatisierung eingesetzt. Das Besondere an dieser Profibus Variante ist, dass in den Busleitungen nur begrenzt viel Strom fließt. Dies hat zur Folge, dass auch im Störfall keine explosionsfähigen Funken entstehen können, und das Bussystem somit in explosionsgefährdeten Bereichen eingesetzt werden kann.

Profibus Topologie:

Linienförmig mit Stichleitungen (< 0.3m), dabei ist die Leitungslänge von der Übertragungsrate abhängig. Pro Linie sind 32 Teilnehmer möglich und zwischen zwei Teilnehmern max. 3 repeater. Maximale Leitungslänge 4800m, maximale Teilnehmeranzahl: 122

Was ist das FDL Protokoll bei Profibus?

Das FDL (Fieldbus Data Link) Protokoll ist das Basisprotokoll auf dem Profibus aufbaut. Das FDL Protokoll belegt im OSI-7-Schichtenmodell Schicht 2 (Data Link Layer). Aufgrund der Hardwarenähe des Protokolls ist ein hoher Telegrammdurchsatz möglich (~ hohe Geschwindigkeit). Um eine hohe Übertragungssicherheit zu gewährleisten, weisen alle Telegramme einen Hamming-Abstand von 4 (HD = 4) auf. Wenn bei der Übertragung eines Telegramms ein Fehler auftritt, dann wird dieses automatisch wiederholt gesendet. Folgende Übertragungsdienste werden unterstützt:

* SDA: Send Data with Acknowledge
* SDN: Send Data with no Acknowledge
* SRD: Send and Request Data with Acknowledge
* CSRD: Cyclic Send and Request Data

Das FDL beinhaltet das Token Passing Verfahren (Master – Master) und Master – Slave Kommunikationsverfahren. Es ermöglicht neben Punkt zu Punkt Datenübertragung auch Broadcasts und Multicasts.

Beschreiben Sie das Profibus-Protokoll (= FDL):

Profibus wurde so spezifiziert, dass es möglich ist, Mono oder Multi-Master-Systeme aufzubauen. Um sicherzustellen, dass jeder Master für eine bestimmte Zeit mit seinen Slaves kommunizieren kann, erfolgt die Auswahl des Masters mit Hilfe des Token-Passing Verfahrens. Dabei werden die Master-Geräte in einem logischen Ring angeordnet und der Token (die Buszugriffsberechtigung) wird innerhalb dieses Rings nach einer bestimmten Zeit an den nächsten Nachbarn weitergegeben. Wichtig dabei ist, dass im laufenden Betrieb defekte oder inaktive Master aus dem Ring entfernt und neu hinzugekommene aktive Teilnehmer in den Ring aufgenommen werden. Jener Master, der im Besitz des Tokens ist, kann nun in einer Master-Slave Kommunikation mit seinen Slaves kommunizieren.

Nennen Sie die drei Profibus-DP Geräte

* DP-Master Klasse 1 (DPM1): in einem festgelegten Nachrichtenzyklus werden zyklisch Daten mit den DP-Slaves ausgetauscht. Typische Geräte sind SPS oder PC.
* DP-Master Klasse 2 (DPM2): bei diesen Mastern handelt es sich Projektierungs-, Überwachungs- oder Engineering-Geräte, die azyklisch auf den Bus zugreifen und dadurch die Konfiguration und Parametrisierung intelligenter Feldgeräte ermöglichen.
* DP-Slaves: ist ein Peripheriegerät (E/A, Antriebe, Ventile, etc.), das Eingangsinformationen einliest und Ausgangsinformationen an die Peripherie ausgibt.

Erklären Sie den RS-485 Schnittstellen-Standard:

Die RS-485 Schnittstelle wurde entwickelt, um Daten über große Entfernungen zuverlässig und schnell übertragen zu können. Bei RS-485 handelt es sich um eine sogenannte symmetrische Schnittstelle. Das heißt, es werden immer zwei Leitungen benötigt, um ein Signal zu übertragen. Auf der einen Leitung wird der invertierte und auf der anderen Leitung der nichtinvertierte Pegel übertragen. Beim Empfänger wird aus der Differenz der beiden Spannungspegel das ursprüngliche Datensignal rekonstruiert. Die Zeichen werden über das Vorzeichen der Spannungsdifferenz identifiziert (z.B.: logisch 1 (+200mV); logisch 0 (-200mV)). Da beide Leitungen für die Datenübertragung benötigt werden, ist die RS-485 Schnittstelle im Gegensatz zur RS-232 Schnittstelle nur half-duplex (es kann also immer nur gesendet oder empfangen werden) fähig. Daten werden üblicherweise nach dem UART-Protokoll (1 Startbit; 8 Datenbits; 1 Paritätsbit; 1 Stoppbit) übertragen. RS-485 wird heutzutage vor allem im Automatisierungssektor (z.B.: Profibus-FMS, Profibus-DP) sehr gerne eingesetzt.

Wie funktioniert der Buszugriff bei Profibus? Welche Formel gilt für den Tokenumlauf?

Die Auswahl des Masters erfolgt durch Token-Passing. Beim Token-Passing werden alle Master Geräte in einem logischen Ring angeordnet und das Token (die Buszugriffsberechtigung) wird innerhalb dieses Rings nach aufsteigender Adresse weitergereicht. Dabei muss sichergestellt werden, dass jeder Master ausreichend Zeit zur Erfüllung seiner Kommunikationsaufgaben erhält. Um dies zu gewährleisten, wird die Tokenumlaufzeit vom Anwender projektiert. Weiters muss sichergestellt werden, dass defekte oder abgeschaltete Master erkannt und aus dem Ring ausgetragen werden. Wenn ein Master im Besitz des Tokens ist, dann können die entsprechenden Slaves via Polling abgefragt werden. Die Tokenhaltezeit (also jene Zeit, die dem Master für das Erledigen der Kommunikationsaufgaben zur Verfügung

steht) berechnet sich zu: TTH = TTR - TRR

TTH...Token Hold Time (Tokenhaltezeit)

TTR...Target Token Rotation Time (Ziel-Tokenumlaufzeit)

TRR...Real Token Rotation Time (tatsächliche Tokenumlaufzeit)

Erklären Sie den Begriff FREEZE

Wenn der Befehl FREEZE vom Master an einen Slave gesendet wird, dann werden die Zustände der Eingänge auf den momentanen Wert eingefroren. Die Eingänge werden erst dann wieder aktualisiert, wenn der Master einen erneuten FREEZE Befehl sendet. Zur Synchronisation von Eingängen

Erklären Sie den Begriff SYNC

Wenn der Master den Befehl SYNC an einen Slave sendet, dann werden die Ausgänge des adressierten Slaves eingefroren. Die Ausgangsdaten werden erst dann wieder an die physikalischen Ausgänge durchgeschalten, wenn ein erneuter SYNC Befehl gesendet wird. Zur Synchronisation von Ausgängen

Erklären Sie den Begriff CLEAR

Teil der global controls von Mastern. Erlaubt die Rücksetzung der Ausgänge in einen sicheren Zustand

Welche Aufgaben führt ein DP-Master im Zustand CLEAR durch?

Im Zustand CLEAR wird die Konfigurierung und Parametrierung der Slaves durchgeführt und das erste Ausgabe-Telegramm gesendet.

Welche Phasen durchläuft ein DP-Slave?

Parametrisierung-, Konfigurations-, Nutzdatenübertragungsphase

Welche Zustände kann ein Profibus-DP Master einnehmen?

* OFFLINE: Startzustand (kein Datenaustausch zwischen Master und Slave)
* STOP: DP Master Klasse 2 kann Diagnose-Informationen auslesen
* CLEAR: Konfigurierung und Parametrierung der Slaves; senden des ersten Ausgabe-Telegramms
* OPERATE: zyklischer Datenaustausch mit den Slaves

Service Access Points bei PROFIBus:

Werden für Interaktion zwischen Application Layser und Data Link Layer verwendet.  
Source Service Access Points (SSAP) sind Dienstzugangspunkte der anfragenden Station  
Destination Service Access Points (DSAP) sind Dienstzugangspunkte der antwortenden Station  
SAPs haben bei DP die verschiedene Funktionen: Default Nutzdatentausch aber auch Master – Master Funktionen, Set / Change Slave Address, Eingänge/Ausgänge lesen…

Wie lauten die zwei Aufgaben des DTM und von wem werden sie zur Verfügung gestellt?

Ein Device Type Manager (DTM) erlaubt die Parametrisierung von Feldgeräten verschiedener Hersteller mit nur einem Programm. Ein DTM ist nichts anderes als ein Treiber, der von seinem spezifischen Gerät Daten lesen und senden kann (über Feldgerätetreiber FDT). Der Device Type Manager wird vom jeweiligen Hersteller zur Verfügung gestellt, enthält Dokumentation in Form von Hilfedateien, unterstützt gerätespezifische Diagnosefunktionen und kennt alle gerätespezifischen Daten und Regeln für die Kommunikation mit dem Gerät.

Ein Feldgerätetreiber (FDT) beschreibt Schnittstellen und Verhaltensweisen zwischen Rahmenapplikation und DTM.

Profinet Protokollstufen:

* UDP/IP: Geräteparametrierung und Konfiguration  
  Lesen von Diagnosedaten  
  Aushandeln des Nutzdatenkanals
* RT (Realtime) mit VLAN-TAG: zyklische Nutzdatenübertragung  
  Ereignisgesteuerte Meldungen/Alarme
* IRT (Isochronous realtime): taktsynchrone Übertragung, Jitter < µs  
  zuerst synchronisierung, dann taktsynchrone, deterministische Kommunikation (IRT), dann Standard (UDP, TCP, RT) Kommunikation

Profinet Geräterollen:

IO-Controller: Kontrolle über den Prozess (auch wenn auf mehrere Feldgeräte verteilt), Anwenderprogramm läuft hier (SPS oder PC)

IO-Supervisor: Engineering-Station, welche temporären Zugriff (z.B. Inbetriebnahme) auf die Feldgeräte besitzen kann.

IO-Device: dezentral angeschlossenes, prozessnahes Feldgerät, erwartet Konfiguration durch Controller oder Supervisor und überträgt seine Prozessdaten zyklisch an den Controller. Kann mehrere Verbindungen zu diversen Supervisors und Controllers gleichzeitig aufrechterhalten.

Foliensatz 5: CAN, LIN, FLEXRAY

Anforderungen und Herausforderungen bei Netzwerken in Autos:

* Comfort and basic service systems:   
  Applikationen werden zumeist als State- Machines, welche durch Events getriggert werden, modelliert. Das Zeitintervall (von einem Aufruf bis zum nächsten Aufruf einer Komfort-Funktion) erstreckt sich von Bruchteilen von Sekunden bis hin zu einigen Tagen. In diesem Bereich ist CAN und auch LIN weit verbreitet.   
  Das Power Management und die Komplexität der State-Machine stellen eine große Herausforderung im Komfort-Bereich dar.
* Drive train and chassis control:  
  Applikationen werden zumeist als Event getriggerte Echtzeitanwendungen modelliert. Vorallem im Bereich der Elektronik des Motors, sind die Perioden nicht immer gleich groß, sondern hängen zum Beispiel von der Umdrehung der Kurbelwelle ab. In diesem Bereich ist CAN sehr weit verbreitet und wird sogar in günstigeren Autos eingesetzt.  
  Eine besondere Herausforderung stellt die Bandbreite, Fehlertoleranz und Zuverlässigkeit dar.
* Safety critical systems:  
  Sicherheitfeatures dominieren diese Anwendungen. Aufgrund der höher benötigten Verlässlichkeit, werden in diesem Bereich zeitgesteuerte (time-triggered) Protokolle verwendet.  
  Eine besondere Herausforderung stellt die Verlässlichkeit, Zuverlässigkeit und Fehlertoleranz dar.
* Multimedia:  
  in diesem Bereich werden einfache Infotainment Systeme modelliert. Da die Übertragung von Video und Audio sehr viel Bandbreite verschlingt, muss dafür ausreichend vorhanden sein. Auf die Fehlertoleranz und Zuverlässigkeit kaum Wert gelegt. CAN, der MOST-Bus und diverse Wireless- Lösungen sind in diesem Sektor weit verbreitet.

CAN-Übersicht:

Can ist eine Multi-Master Architektur, jeder der bis zu 30 Busteilnehmer hat Zugang zum Übertragungsmedium, bei gleichzeitigem Zugriff erfolgt die Auflösung selbstständig. Die Übertragungsrate ist variabel bis 1 Mbit/s, bei reduzierter Übertragungsrate sind bis zu 1000m langen Leitungen möglich. CAN verwendet kurze Nachrichten (0-8 Byte), dies hält die Übertragungszeit kurz und garantiert die Latenzzeit. Außerdem ist CAN sehr störsicher.

CAN-Genauer:

CAN überträgt ein Signal über zwei Leitungen (-> Sicherheit). Der Zustand mit zwei (groß) unterschiedlichen Pegeln auf den beiden Leitungen wird als dominanter Zustand (logische 0) und der Zustand mit zwei gleichen (wenig unterschiedlichen) Pegeln auf beiden Leitungen wird als rezessiver Zustand (logische 1) bezeichnet. Bei der Datenübertragung in einem CAN-Bus werden keine Knoten adressiert, sondern der Inhalt einer Nachricht (z.B: Temperatur, Luftfeuchte) wird durch einen eindeutigen 11bit langen (bei CAN 2.0A) bzw. 29bit langen (bei CAN 2.0B) Identifier festgelegt.   
Ein wichtiger Unterschied im Gegensatz zu Profibus ist, dass Teilnehmer ohne vorher vom Master angesprochen worden zu sein, Daten senden können. Dadurch kann es jedoch dazu kommen, dass mehrere Teilnehmer gleichzeitig Daten auf den Datenbus legen - es kommt zu einem Buszugriffskonflikt. Dieser Konflikt wird durch ein „Wired-And-Busarbitrierungsschema“ aufgelöst. Das heißt, es setzt sich jene Nachricht mit der höchsten Priorität (entspricht dem niedrigsten Identifier) ohne Zeit- bzw. Bitverlust durch. Eine Kollision wird dadurch erkannt, dass ein Sender den gesendeten Identifier selbst zurückliest und vergleicht. Da ein dominates Bit ein rezessives Bit überschreibt, sind alle Knoten, die ein rezessives Bit anlegen, aber ein dominantes Bit empfangen ab sofort nur noch Empfänger und senden erst dann wieder, wenn der Bus wieder frei ist.

Nennen Sie die vier verschiedenen CAN-Telegrammformate

* Datentelegramm: Datenübertragung von einem Sender zu einem oder mehreren Empfängern auf Initiative des Senders
* Datenanforderungstelegramm: Ein Teilnehmer (Empfänger) fordert das Senden eines Datentelegramms von einer Quelle (Sender) an
* Fehlertelegramm: Kennzeichnet einen Fehler an alle Teilnehmer
* Überlasttelegramm: Signal von CAN-Teilnehmern, dass sie überlastet sind. Realisiert Verzögerung zwischen Datenanforderungstelegrammen

Wodurch unterscheidet sich ein CAN-Fehlertelegramm von einer Datennachricht?

Ein Error-Frame unterscheidet sich deutlich von einem herkömmlichen Data-Frame: Es besteht aus 6-12 dominanten Bits (Error-Flag) gefolgt von 8 rezessiven Bits (Error-Delimiter). Das Error-Flag überschreibt ggf. auf dem Bus gesendete Bits und führt andererseits eine Codeverletzung herbei (Bitstuffing-Regel wurde verletzt). Falls mehrere Knoten gleichzeitig einen Fehler melden, dann können sich die Error-Flag Felder überlagern, wodurch maximal 12 dominante Bits hintereinander gesendet werden.  
Da durch das Bitstuffing maximal 5 Bits gleicher Polarität aufeinander folgen dürfen, bemerken alle teilnehmenden Netzknoten diese Codeverletzung und verwerfen den Datenrahmen.

Erklären Sie Bitstuffing:

Um die Teilnehmer synchron zu halten werden lange monotone Sequenzen am Bus verhindert. Deshalb werden „Bits gestopft“: nach fünf gleichen Bits wird ein inverses Bit eingefügt. Dieses wird vom Empfänger als gestopftes Bit erkannt und wieder aus der Bitfolge entfernt. Ist das sechste Bit dagegen gleich wie die fünf vorherigen dann erkennt der Empfänger einen Übertragungsfehler.

Fehlereingrenzungsverfahren bei CAN:

Um zu vermeiden, dass einzelne Teilnehmer durch irrtümlich erkannte Fehlerbedingungen dauerhaft den Nachrichtentransport blockieren, enthält jeder Teilnehmer Fehlerzähler für Sende- und Empfangsfehler. Diese Zähler erlauben nach den Regeln der Spezifikation, einen fehlerhaft arbeitenden Teilnehmer in zwei Stufen des Betriebszustands vom Bus zu trennen, wenn er wiederholt Fehler erkennt, die andere Teilnehmer nicht erkennen, oder wiederholt fehlerhafte Frames versendet. Die Zustände nennen sich fehleraktiv (normal), fehlerpassiv (Teilnehmer darf nur noch passive – das heißt rezessive – Error-Frames senden) und vom Bus ausgeschaltet (Teilnehmer darf nicht mehr senden). Verschiedene Fehlerfälle führen zu einer unterschiedlich großen Erhöhung der Fehlerzähler.

LIN-Übersicht:

Local Interconnect Network ist ein serielles (und sehr einfach aufgebautes) Bussystem, welches für die kostengünstige Kommunikation von Sensoren und Aktoren in Kraftfahrzeugen entwickelt wurde. Die Buszugriffssteuerung bei LIN beruht auf dem Master-Slave-Prinzip. Die Kommunikation wird dabei immer vom Master initiiert. Der Master kann dabei Daten von einem Slave anfordern (Slave -> Master), Daten an den Slave schicken (Master -> Slave) oder eine Kommunikation zwischen zwei Slaves initiieren (Slave #1 -> Slave #2). Die Kommunikation erfolgt dabei zeitgesteuert (schedule). Ein LIN-Header ist folgendermaßen aufgebaut:

* Synchronisationsbreak: um den Beginn eines neuen Datenrahmens anzuzeigen, wird ein 13bit langer 0-Pegel gesendet.
* Sync-Feld: es folgt ein Synchronisationsbyte (01010101, 0x55), zur Taktsynchronisation
* Identifier: definiert Richtung, Bedeutung und Länge des zu übertragenden Datenfelds.
* 1-8 Datenbytes (Antwort)
* Checksumme (Checksumme)

Da die Kommunikation immer vom Master ausgeht, wird sichergestellt, dass zu jedem Zeitpunkt immer nur eine LIN-Botschaft übertragen wird. Kollision sind somit nicht möglich. Abschließend noch ein paar wichtige Informationen zu LIN:

* Datenrate: 19,2kBit/s
* max. Leitungslänge: 20m
* max. 16 LIN-Knoten auf einem Bus
* halbduplex Verfahren (gleichzeitiges lesen und schreiben ist nicht möglich, da nur eine Datenleitung existiert!)

Welche verschiedenen Typen von Datenpaketen gibt es bei LIN?

* Unconditional Frame (Standardframe): dient zur Signalübertragung zyklischer Daten (z.B. Sensordaten)
* Event-triggered Frame: damit werden mehrere Standardframes verschiedener Slaves zusammengefasst. Da nun nicht mehr jeder Slave einzeln angesprochen wird, kann dadurch wertvolle Übertragungskapazität eingespart werden. Wenn mehrere Slaves gleichzeitig antworten, kann es jedoch zu Kollisionen kommen. Diese werden dann mit Hilfe einer Collision Resolving Schedule Table aufgelöst.
* Sporadic Frame: ein Sporadic Frames dienen dem Masterknoten quasi als Platzhalter für beliebige Datenpakete. Der Master kann so in einem Slot unterschiedliche Datenpakete verschicken (z.B einmal Daten des Temperatursensors und ein anderes Mal Daten des Feuchtesensors) oder den Slot einfach unbenutzt lassen.
* Diagnostic Frame

LIN Workflow:

NCF (Node Capability File) enthält leistungsumfang eines Knotens  
LDF (LIN Description File) enthält alle Eigenschaften des LIN-Clusters, schedule, …  
Aus LDF werden mit einem LIN Cluster Generator die Software Komponenten der Knoten erstellt

FlexRay

Flexray ist ein serielles, deterministisches, fehlertolerantes Bussystem, welches spezifiziert wurde, um die erhöhten Anforderungen (höhere Datenübertragungsrate, Echtzeit-Fähigkeit, Ausfallsicherheit) zukünftiger Vernetzung in Fahrzeugen erfüllen soll.  
Nachrichten werden in festen Zeitfenstern auf zwei Kanälen (um eine fehlertolerante und redundante Übertragung zu gewährleisten) übertragen. In diesen festen Zeitfenstern hat der jeweilige Knoten exklusiven Buszugriff. Diese Zeitfenster wiederholen sich in einem gewissen Zyklus. Es kann dadurch vorausgesagt werden, zu welchem Zeitpunkt sich welche Nachricht auf dem Bus befindet. Ein Nachteil dieser festen Zuordnung ist, dass die Bandbreite nicht voll ausgeschöpft werden kann. Aus diesem Grund wird der Zyklus in statische und dynamische Abschnitte unterteilt.   
Am Anfang eines Kommunikationszykluses befindet sich der statische Abschnitt, in dem jeder Knoten ein fixes Zeitfenster zugeteilt bekommt, in dem er senden muss. Wenn keine Daten zum Senden vorliegen, dann wird ein Nullframe gesendet.  
Im darauffolgenden dynamischen Abschnitt, werden die Zeitfenster dynamisch zugeteilt. Das dynamische Segment wird von Knoten benutzt, die längere (die Breite des statischen Slots reicht hierfür nicht aus) oder zusätzliche Nachrichten senden möchten. Jedem Knoten wird dazu ein Minislot zugewiesen. Wenn nun ein Knoten eine Nachricht absetzen möchte, dann wird der Minislot für die Dauer der Übertragung verlängert. Wenn es hingegen nichts zu übertragen gibt, dann läuft der Minislot einfach ab.   
Neben einem statischen und dynamischen Abschnitt befindet sich in einem Zyklus auch noch Abschnitte für das Symbolfenster (symbol window) und die NIT (Network Idle Time). Das Symbolfenster war für den Test des Buszugriffs vorgesehen und wird mittlerweile nicht mehr verwendet. Die NIT, in der keine Kommunikation zwischen den Knoten stattfindet, soll dazu dienen, dass sich die Knoten wieder exakt mit dem Bus synchronisieren können (Uhrensynchronisation).

Bus Guardian: Was macht er? Wo wird er eingesetzt?

Der Bus-Guardian wird u.a bei Flexray eingesetzt und stellt sicher, dass Netzwerkteilnehmer nur in ihrem jeweiligen Zeitslot senden. Babbling-Idiots (Knoten, die Nachrichten unsynchronisiert versenden wollen) werden somit erkannt und vom Bus getrennt.

Nennen Sie die vier aufeinanderfolgenden Segmente eines Kommunikationszyklus bei FlexRay

* statisches Segment
* dynamisches Segment
* Symbolfenster
* Network Idle Time

Wie ist ein Flexray-Knoten aufgebaut (inkl. Skizze und Erklärung)?

Ein Flexray-Knoten besteht aus einem Host-Prozessor, einem Communication-Controller (CC), dem Bus Guardian und einem Bustreiber (je Kanal). Daten vom Host-Prozessor werden in bestimmten Zeitabständen (Zeitslots) vom Communication-Controller versendet. Die vom Communication-Controller versendeten Nachrichten werden von den Bustreibern in physikalische Signale umgewandelt und auf die entsprechenden Leitungen geschrieben. Der Communication- Controller hat zudem die Aufgabe den Knoten mit dem Restnetzwerk zu synchronisieren. Der Bus-Guardian überwacht den Buszugang und aktiviert nur dann die Bustreiber, wenn der jeweilige Zeitslot erreicht ist.

Foliensatz 6: InterBus

Wie funktioniert Interbus?

Interbus ist ein Feldbussystem, welches sowohl in der Sensor/Aktor Ebene, als auch der Prozessautomatisierung - und Überwachung Verwendung findet. Topologisch betrachtet, ist Interbus ein Ringsystem. Das heißt alle Teilnehmer sind in einer Ringstruktur angeordnet und leiten das ankommende Signal (nach einer Signalaufbereitung) an den nächsten Teilnehmer weiter. Im Gegensatz zu anderen Ringsystemen wird bei Interbus sowohl die Hin - als auch die Rückleitung innerhalb eines Kabels paarweise verdrillt und an sämtliche Teilnehmer geführt. Dadurch ergibt sich physisch eine linienartige (auch baumartige) Verkabelungsstruktur.  
Die Kommunikation zwischen den Teilnehmern wird durch ein Master/Slave Zugriffsverfahren geregelt, wobei der Master zusätzlich für die Kopplung an das übergeordnete Bussystem zuständig ist. In das Interbus-System können bis zu 512 Slaves integriert werden.  
Durch die integrierte Verstärkerfunktion in jedem Fernbusteilnehmer lässt sich ein Interbus-System mit einer Länge von bis zu 13km realisieren. Eine weitere Besonderheit von Interbus ist die Möglichkeit Subringsysteme zu bilden. Die Ankopplung solcher Subringsysteme erfolgt durch Buskoppler (Busklemmen). Als Beispiel soll hier der Aufbau eines Interbus Systems mit einem Hauptring und zwei Subringsystemen erläutert werden.  
Von der Anschaltbaugruppe (Master des Interbus) ausgehend wird der Fernbus installiert. An den Fernbus werden Fernbusgeräte und Buskoppler angeschlossen. Jeder Buskoppler verbindet den Fernbus mit einem Subringsystem. Es gibt dabei drei verschiedene Ausprägungen von Subringsystemen: Lokalbus, Fernbusstich und Interbus-Loop.

* Lokalbus: entspricht Peripheriebus
  + kann nicht weiter verzweigt werden
  + Versorgung erfolgt direkt aus integriertem Netzteil der Buskopplung
  + max. Anzahl der Teilnehmer: 8
  + kann einzeln vom Fernbus getrennt werden
  + für geringe Distanzen (1,5m) zwischen Modulen ausgelegt
* Fernbusstich: verbindet Fernbusgeräte miteinander
  + max. Abstand zwischen zwei Fernbusgeräten: 400m
  + mit Fernbusstichen können komplexe Netztopologien realisiert werden
  + Verdrahtung erfolgt üblicherweise mit paarweise verdrillten, einfach geschirmten
  + Signalleitungen (alternativ: Glasfaserkabel)
* Interbus-Loop:
  + echte Ringstruktur
  + max. Anzahl der Teilnehmer: 63
  + max. Abstand zwischen zwei Teilnehmern: 20m
  + Versorgung der Teilnehmer erfolgt über den Bus

Der Busmaster steuert alle Abläufe auf dem Ring und ist als Anschaltbaugruppe für verschiedene Hostsysteme (SPS, Industrie-PC) verfügbar. Er steuert das zyklische InterBus Protokoll, ist für die Kommunikation mit intelligenten Teilnehmern, das Busmanagement, der Fehlerdiagnose und -ortung und den Transfer der E/A Daten zu den Modulen und zum überlagerten Steuersystem zuständig.  
Der Buskoppler verbindet (unterschiedliche Segmente). Er ist für die Versorgung der Busgeräte zuständig und dient als Signalumsetzer (innerhalb eines Interbus-Systems können somit unterschiedliche Übertragungsmedien eingesetzt werden). Weiters zeigt er Fehler über einen potentialfreien Alarm-Ausgang und Diagnose Daten via LEDs an.  
E/A Module sind Kommunikationsslaves im Ringbus, die mittels einer Hin - und Rückleitung ins Interbus-System eingebunden werden. Zusätzlich besitzen sie eine entsprechende Schnittstelle (binär, digital, analog) um eine Verbindung zu den Prozesssignalen herzustellen. Datensicherheit wird durch CRC, Timeout-Kontrolle und Stopbit-Check erreicht.  
Allgemeine Infos:

* schnelle und einfache Inbetriebnahme
* gute Diagnoswerkzeuge integriert
* niedrige Kosten
* Übertragungsrate: 500kbit/s
* Gesamtlänge eines Interbus-Netzes: max. 13km
* Protokolleffizienz konstant 60%

Wie funktioniert die Datenübertragung bei Interbus?

Das Interbus-Protokoll arbeitet nach dem Summenrahmenverfahren, welches eine hohe Effizienz bei der Datenübertragung und das gleichzeitige Senden und Empfangen von Daten (Vollduplexbetrieb) bei deterministischer Laufzeit ermöglicht.  
Der Summenrahmen besteht aus dem Loopback, den Steuerdaten, einem Prüfsummenfeld und den Daten für alle(!) Teilnehmer. Dieser Summenrahmen wird wie bei einem Schieberegister beginnend mit Loopback durch den Ring geschoben. Wenn der Loopback im Empfangsregister ansteht, dann wurde der Summenrahmen durch alle Teilnehmer geschoben und alle Teilnehmerdaten liegen beim richtigen Teilnehmer an. Die Teilnehmer lesen die Eingangsdaten von ihrem Slot und schreiben die Ausgangsdaten in den Slot zurück.  
Die einzelnen Busteilnehmer werden bei Interbus also nicht über eine Adresse adressiert, sondern über die Position im Ring. Die Länge des Datenrahmens ergibt sich aus der Anzahl der Teilnehmer und der Nutzdaten, die pro Teilnehmer übertragen werden. Da alle Teilnehmer ihre Daten zur selben Zeit lesen, kommt es zu keinen Dateninkonsistenzen.  
Beim Starten der Anlage wird vom Busmaster als erstes ein Identifikationszyklus (ID-Zyklus) initiiert. Dabei schalten alle Busteilnehmer die ID-Register in den Interbus-Datenring, damit der Busmaster diese identifizieren kann. Erst danach folgen die Datenzyklen zur Nutzdatenübertragung.   
Jeder Zyklus besteht aus 2 Sequenzen (Daten- und Checksequenz). Master wählt mittels Select Signal ID- oder Datenzyklus, dann sendet er Daten beginnend mit Loopback, empfängt # der Daten gesendet und Loopback. Er leitet CRC Sequenz ein: jeder Teilnehmer sendet seine Prüfsumme an den nächsten Eingang.   
Das 16bit breite ID-Register (welches jeder Slave besitzt) ist folgendermaßen aufgebaut:

* Bits 0-7: Gerätegruppe (digital, analog, FB, IB, LB-Teilnehmer)
* Bits 8-12: Datenbreite des Registers
* Bits 13-15: Management Bits (CRC-Fehler, Modulfehler, Re-Konfigurtionsanforderung)

Folgende Signale warden zur Datenübertragung von InterBus benötigt:  
Bustakt, Datensignal, Select-Signal, Control-Signal, Reset-Signal

IEC 61508 – Safety

Funktionale Sicherheit elektrischer/elektronischer/ programmierbar elektronischer Systeme. InterBus besitzt das Safety Integrity Level 3 nach IEC 61508. Dazu wird das Black-Channel-Prinzip verwendet: ein „black-channel“ ist ein Kommunikationskanal, der den Sicherheitsanforderungen nicht entspricht. Um diesen Kanal nun verwenden zu können wird zwischen Anwendung und Kanal ein Safety-Layer eingefügt, welcher den unsicheren Kanal überwacht.

Foliensatz 7: Common Industrial Protocol

CIP

CIP (Common Industrial Protocol) ist ein Anwendungsprotokoll, welches den Übergang von Feldbus-Systemen in industrielles Ethernet (EtherNet/IP, DeviceNet...) und IP-Netze ermöglicht. Das CIP Protokoll liegt oberhalb der Transportschicht und erweitert die reinen Transportdienste um Kommunikationsdienste für die Automatisierungstechnik.

Netzwerktechnologien bei CIP

?

DeviceNet

DeviceNet ist eine Implementierung von CIP auf Basis von CAN. Es besteht aus vier Leitungen (CAN-H, CAN-L, Vcc, Ground). Von einer Stammleitung (Trunk) gehen Stichleitungen (Drops) fort, maximal können 64 Teilnehmer an einer Leitung hängen, die Übertragungsraten sind 125, 250 und 500 kbps. DeviceNet ist ein objektorientiertes Bussystem, welches nach dem Producer-Consumer-Prinzip arbeitet. DeviceNet Geräte können Client, Server oder beides sein. Clients und Server wiederrum können Consumer, Producer oder beides sein. Die Kommunikationsbeziehungen zwischen Master und Slave können folgendermaßen kategorisiert werden:

* Polled-I/O: Daten des Slaves werden zyklisch durch Master abgefragt
* Explicit: azyklische Kommunikation
* Bit-Strobed: Nachricht von Master an alle Slaves, welche daraufhin Eingangsinformationen senden
* Change-of-State: Slave sendet bei Eingangsänderung automatisch Daten an Master
* Cyclic: Slave sendet seine Daten nach Ablauf einer Zykluszeit selbstständig

ControlNet

ControlNet ermöglicht es, die zyklische E/A-Datenübertragung und die azyklische Datenkommunikation für Konfiguration und Programm-Up-/Download gleichzeitig auszuführen. Dabei ist sichergestellt, dass weder Durchsatz noch Determinismus oder zyklische Wiederholbarkeit beeinflusst werden: Zeitkritische Daten haben garantierte Übertragungszeitpunkte (während des „scheduled Service“), der restliche Datenverkehr lässt sich hinsichtlich seiner maximalen Übertragungszeiten eindeutig vorherbestimmen (zeitunkritisch: „unscheduled service“, nach round robin verfahren slot vergeben), jeweils gibt es einen Guardband Slot zur Synchronisation.  
Die max. Datenrate liegt bei 5 Mbit/s, es sind Trunklines und Droplines, aber auch Sterne und Ringe möglich, maximal 99 Teilnehmer.

Ethernet/IP

EtherNet/IP ist ein industrielles Kommunikationsprotokoll auf Basis von Ethernet, welches auf den beiden weit verbreiteten Transportprotokollen TCP/IP und UDP/IP aufbaut. Dazu wurde Ethernet um das Common Industrial Protocol (CIP) erweitert. Zur Datenübertragung werden üblicherweise normale Twisted-Pair Kabel oder Lichtwellenleiter verwendet. Da bei Ethernet (mit seinem CSMA/CD Datalink- Layer) bei höherer Belastung überproportional viele Kollisionen auftreten, hat man sich in der Automatisierungstechnik lange Zeit dagegen gesträubt Ethernet einzusetzen. Durch höhere Übertragungsgeschwindigkeiten (100 Mbps und höher) und den Einsatz von Switched - statt Shared-Hubs reduziert sich die Anzahl der Kollisionen jedoch drastisch. Man sollte jedoch im Hinterkopf behalten, dass sich Ethernet/IP für „harte“ Realtime-Anwendungen (< 1ms) nach wie vor nicht eignet. Wie bereits oben erwähnt, setzt man bei Ethernet auf die beiden Transportprotokolle TCP/IP und UDP/IP. TCP/IP wird immer dann eingesetzt, wenn man Daten fehlerfrei übertragen möchte (z.B. Programm-Download). Kommt es bei der Übertragung eines Datenpakets zu einem Fehler, dann sorgt TCP/IP dafür, dass das Paket erneut gesendet wird. Für die Übertragung von Prozessdaten (z.B Daten eines Sensors) eignet sich das Protokoll hingegen weniger. Hier wäre es besser, immer den aktuellsten Wert zu übertragen, anstatt einen alten Daten zu wiederholen, bei dem ein Fehler aufgetreten ist. In diesem Fall verwendet man besser UDP/IP.

Zugriffverfahren bei ControlNet

?

Foliensatz 8: Wireless

Wireless Communication Vorteile, Herausforderungen, Probleme, Lösungen

Vorteile:

* Weniger Installationskosten
* Knotenpunkte können auch dort platziert werden, wo man mit Kabeln nicht hinkommt
* Knoten sind mobil: können flexibel verbunden werden, bewegt werden, erlauben User ad-hoc Interaktion

Herausforderungen:

* Kaum Ressourcen: Energie, Speicher und Rechenleistung
* Knoten Beweglichkeit
* Heterogene Gerätearchitekturen: Manche „full blown“ (kaum Einschränkungen), andere mit nur minimalen Ressourcen/Fähigkeiten
* Interaktion mit „transmit-only-devices”
* Interferenz: unintentional & intentional (physical access no longer needed for security attacks)

Probleme:

* Offenes Medium: keine Direktionsmöglichkeit der Informationen, jeder kann hören, jeder kann angreifen
* Unverlässlich: im Vergleich zu kabelgebundenen Netzwerken, Verlässlichkeit kostet viele Ressourcen (Bandbreite, Strom, Zeit,…)
* Hidden Node Problem: wenn A Hub sieht, B Hub sieht, aber A B nicht sieht entsteht eine Kollision wenn beide gleichzeitig senden.
* Layer 2 (MAC) interference
* „exposed Node Problem”: Broadcast Range von Knoten überlappt sich und Knoten kann auch bei unterschiedlichen Empfängern nicht senden

Lösungen:

* Low power Designs
* Energie ergattern: Photovoltaik, Temperatur, Druck,…
* Gegen intentional interference: Kryptographie
* Topologien:
  + Stern: P2P Verbindungen mit Gateway, unidirektionelle Kommunikation möglich
  + Mesh: redundante Kommunikationspfade, Weiterleiten von Daten, selbst Heilend, große Reichweite, Robust gegen Ausfall von Knoten und Interferenzen
* Wireless Sensor Network (WSN): Für verteilte Umwelt Überwachung, geringe Kosten, Leistung, Größe,…; ein einziger starker Knoten als Gateway zu User, Plug and Play, Selbstorganisierend, Anwendungen: Logistik, Militärwesen, Umweltmonitoring

Wired VS Wireless Communication

|  |  |
| --- | --- |
| Wired | Wireless |
| Hohe Bandbreite | Niedrig bis mittlere Bandbreite |
| High performance | High latency |
| Robust | Interferenzen |
| Reliable | Unreliable by nature |
| Teuer | Billig |
| Viele Ressourcen | Sehr begrenzte Ressourcen |
| Statisches Netzwerk | Mobiles Netz |
| Security nicht so wichtig da phys. Zugriff notwendig | Security wichtig da kein phys. Zugriff notwendig |

Wireless Basics

Elektromagnetische Energie wird vom Sender zum Empfänger übertragen, ohne Medium übertragen. Dieses Signal breitet sich im Idealfall Kreisförmig aus und wird mit Wellenlänge(λ) und Frequenz charakterisiert.  
Wichtige Frequenzbänder sind:  
VHF/UHF Bänder für mobiles Radio (λ ~ dm)  
SHF Bänder für Statelliten (λ ~cm)  
UHF/SHF Band für Wireless LANs, Mobiltelefone  
ISM Bänder  
Die Ausbreitung erfolgt im wesentlichen direkt (wie Licht), hängt jedoch von Frequenz und Antenne ab. Die maximale Weite hängt ebenfalls von Distanz, Antenne, Frequenz, Leistung, Umgebung, Wetter,… ab.  
Multiplexen von Übertragungen um gemeinsam das Medium verwenden zu können (Access Möglichkeiten):

* Space (SDMA): Kanäle im Raum verteilt
* Time (TDMA): der Kanal belegt gesamte Frequenz für definierten Zeitslot; hoher throughput, aber genaue Synchronisation benötigt
* Frequenz (FDMA): Die gesamte Bandbreite wird in Frequenzsegmente aufgeteilt, jeder Kanal bekommt ein Segment. Benötigt keine Koordination, gut für Analoge Signale aber Unflexibel und möglicherweise Verschwendung von Bandbreite
* Code (CDMA): jeder Sender hat einen einzigartigen orthogonalen Code, alle Sender senden gleichzeitig. Keine Koordination notwendig, Bandbreite effizient genutzt, Daten gegen Interferenzen geschützt, jedoch Datenrate limitiert und komplexe Signalerzeugung
* Time and frequency: Mischung von TDMA und FDMA (z.B. GSM), sehr sicher, hohe Datenraten, aber sehr genaue Koordination erforderlich

IEEE 802.15.4

Ist ein offenen spezifizierter wireless-communication Standard für Sensor/Aktuator Netzwerken. Gedacht für kurze Strecken, wenig Energie, geringe Kosten. Er ist flexibel und mächtig und unterstützt security Geräte.   
Es gibt kein Frequenz-hopping, Stern oder Peer to Peer Topologien, der Datentransfer erfolgt mit Bestätigungen und CSMA/CA oder Guaranteed Time Slots. Es wird zwischen Full Function Devices (FFDs) und Reduced Function Devices (RFDs) unterschieden.   
FFDs implementieren den kompletten Kommunikationsstack, sind extern Stromversorgt. können mit FFDs und RFDs kommunizieren, sind daher PAN Koordinatoren oder Endgeräte.  
RFDs sind dagegen meist batteriebetriebene Endgeräte und können nur mit FFDs kommunizieren.  
Jeder IEEE 802.15.4 Knoten hat eine einzigartige 64 bit Adresse (wie z.B. MAC Adresse) und zusätzlich eine 16 Bit Kurzadresse, welche einzigartig im Personal Area Network (PAN) sein muss. Durch PAN lassen sich mehrere Stern-Netzwerke zu einem Netzwerk zusammenfassen.  
Es gibt zwei Möglichkeiten für die Superframe-Struktur:   
non-beacon enabled: keine Synchronisation, nur CSMA/CA  
Beacon enabled: Superframe wird von Koordinator vorgegeben, Beacons Synchronisieren das Netzwerk  
Der Datentransfer erfolgt direkt oder indirekt:  
direkt: Geräte schicken Daten(anfrage) direkt an anderes Gerät  
indirekt: Geräte kommunizieren mit Koordinator (nur angefragte Daten werden gesendet)

Foliensatz 9: Gebäudeautomation

Wie ist der Begriff der Gebäudeautomation in der ISO 16484 definiert?

„System aus Produkten und Dienstleistungen für automatische Steuerung und Regelung, Überwachung, Optimierung, Betrieb sowie für manuelle Eingriffe und Management zum energieeffizienten, wirtschaftlichen und sicheren Gebäudebetrieb.“

Der Prozess Gebäude:

Beinhaltet Gebäudeinneres und Umfeld, hat meist unkritisches Zeitverhalten, jedoch mittlere Anforderungen für Verfügbarkeit und Zuverlässigkeit (sehr hoch bei Security/Safety)

typische Anwendungen für die Gebäudeautomation:

* Heizung, Klima, Lüftung (Zonenregelung, Einzelraumregelung, Start-/Stoppautomatik...)
* Beleuchtung (Bewegungsmelder, Sonnenschutz, transparent schaltbare Verglasungen...)
* Sicherheitstechnik (Zutrittskontrollen, Gefahrenmeldungen, Videoüberwachung, Notbeleuchtung...)

Herausforderungen bei der Gebäudeautomation:

* Stochastische Störgrößen: Wechselnde Witterungsverhältnisse, Personenanzahl, Verhalten
* kostensensitiv
* Lange Lebenszyklen: Technisch konservativ (aber zukunftssicher); Ausschreibungen verlangen oft Normenkonformität
* Potentiell außerordentlich große Ausdehnung: Lokalität der Daten hoch -> dezentraler Ansatz

Offene Systeme

Bei offenen Systemen ist die Spezifikation öffentlich zugänglich (nicht notwendigerweise kostenlos), sodass Reparatur, Änderungen, Erweiterungen nicht nur ursprünglicher Hersteller vornehmen kann. Dadurch steigt der Engineering Aufwand meist, da die Schnittstellen nicht als „Plug&Play“ gedacht sind. Offene Systeme sind nicht mit Open-Source Systemen vergleichbar. BACnet, LonWorks und KNX sind offene Systeme.

Anforderungen an Netzwerke bei Gebäudeautomation:

Das Control Network (Feldebene) hat hohe Knotenzahlen, niedrige Datenraten, muss Kosteneffizient, robust, einfach zu installieren sein und akzeptable Reaktionszeiten liefern. Daher greift man hier auf Feldbussysteme zurück.  
Der Backbone dagegen benötigt eine hohe Netzwerkbandbreite um mit tausenden Datenpunkten umgehen zu können, dafür gibt es nur moderate Anforderungen an Robustheit und Rechtzeitigkeit. Hier werden Office (IP) Technologien eingesetzt.

KNX (IEC/ISO 14543)

KNX wird auf der Feld und Automatisierungsebene eingesetzt. Es verwendet eine Drei-Ebenen-Struktur bestehend aus (Haupt-, Sub-) Linien mit IP Tunneling/Routing. Geräte haben eine individuelle Adresse und Gruppenadresse. Der Kontrolldatenaustausch erfolgt auf dem Publish/Subscribe Modell, der Entwicklungsdatenaustausch auf Basis des Client/Server Modells.   
Für den Netzaufbau werden vier Arten von Geräten benötigt:   
Bus Coupling Units (BCUs) mit KNX/EIB Stack und TPUARTs mit KNX Layer 1&2, diese sind generic devices.   
Applikationsmodule (z.B. Lichtschalter) und Hostcontroller sind für den Task specific devices.

LonWorks

LonWorks wird auf der Feld und Automatisierungsebene eingesetzt. Es ist ein allgemein verwendbares, ereignisgesteuertes Embedded-Netzwerk für Steuerungszwecke. Es besteht aus drei Komponenten: dem LonTalks Protokoll, „Neuron“ Chip (umfasst Schicht 1-2 MAC, Netzwerkstack 3-6 und Schicht 7 Benutzerapplikation) und ein Netzwerk Management Tool (LNS).  
Datenübertragung kann Unacknowledged (repeated), acknowledged, oder mittels request/response erfolgen.

BACNet

BacNet wird auf der Management und Automatisierungsebene eingesetzt. Das Building Automation and Control Network ist in ISO 16484-5 definiert und kann lizenzfrei implementiert werden. Es ist Netzwerkneutral und ausschließlich ein Protokoll (läuft zum Beispiel auf LonTalk, Ethernet oder ZigBee Unterbau). Es definiert Objekte und Dienste.

Foliensatz 10: Industrial Ether-, Internet

Industrial Ethernet

Zielt auf Ethernet bis zu den Endknoten (also ohne Feldbusse) ab. Dadurch soll Anschaltung kostengünstig werden, da man z.B. bestehende Verkabelungen verwenden könnte und es eine einheitliche Lösung sein soll. Außerdem sind keine Zwischenknoten notwendig.  
Bisher hat es Ethernet in der Industrie aufgrund von fehlender Echtzeitfähigkeit durch den Shared-Medium-Ansatz nicht gegeben.  
Auch mit full-duplex-Kommunikation ist Ethernet noch immer nicht voll echtzeitfähig, Protokollstacks und Netzwerk-Interfaces erzeugen unvorhersebare Verzögerungen. Außerdem stellen Switches Flaschenhälse dar: Puffer und Queues sind beschränkt, Uplink hat begrenzte Kapazität,… Dies kann jedoch mit Flusssteuerung gelöst werden. Weiters sind Anwendungen ebenfalls Flaschenhälse.  
Die Effizienz in Ethernet beträgt nur 11%.   
Bis jetzt ist Industrial Ethernet keine standardisierte Lösung, es gibt viele verschiedene Ansätze:  
TCP/IP Pakete über INTERBUS zu schicken  
Ethernet/IP (Industrial Protocol) zu verwenden, da dies schon bekannt ist, Kompatibilität zu bereits verwendeten Anwendungen (DeviceNet) hat und die Buslast durch Publish/subscribe Modell gering gehalten wird. Jedoch ist dies nur eine Tunnellösung mit allen Problemen von TCP/UDP/IP und für Echtzeit werden starke Rechner benötigt.  
Neue Echtzeitprotokolle auf Ethernet sind spezielle Protokolle ohne Rückwärtskompatibilität.  
Generell gibt es die Problematiken der Kombination von (Nicht-)Echtzeitdaten, der Integration von IP Daten und der sinnvollen Aufteilung der Bandbreite.  
Ethernet Powerlink ist ein neues Echtzeitprotokoll und ersetzte TCP/IP und MAC durch spezielle Funktionen. Dadurch werden kurze Zykluszeiten, deterministisches Verhalten, kleiner Jitter, hoher Datendurchsatz (Ausnutzung > 90%) möglich. Jedoch ist es nicht konform mit Standard Ethernet, der Bus Master ist eine kritische Einheit, es braucht spezielle Router und Gateways zur Anbindung an Office-Netzwerke und es ist nicht erweiterbar.  
  
Industrial Ethernet ersetzt keine Feldbusse im untersten Bereich, es ist nicht so kostengünstig wie behauptet und bietet noch keine Eigensicherheit.

Bisherige Integrationsarten von industrial Ethernet:

* Feldbus über Internet: Tunneln des Feldbus-Protokolls, daher Feldbus-Know-How bei Client notwendig -> nicht benutzerfreundlich
* Internet über Feldbus: Tunneln des Internet-Protokolls, daher spezielle Software im Feldgerät notwendig, dies braucht viele Ressourcen.
* Gateway zwischen Feldbus und Internet: Standardsoftware beim Client; ist Feldbus-unabhängig, Benutzerfreundlich, aber mit sehr hohem Aufwand (viele Protokollkombinationen) verbunden.

Service orientierte Architekturen

Service Orientierung ist ein Designansatz bei dem große komplexe Systeme durch individuelle kleine Services aufgebaut werden. Dabei ist ein Service eine Implementation einer Betriebsaktivität mit einem spezifizierten Ergebnis. Die Funktionalität von Services muss, well-defined, reusable, self-contained sein.  
Um dies zu ermöglichen muss folgende Infrastruktur gegeben sein:   
Services müssen beschreibbar sein, Provider müssen ihre Services veröffentlichen und User müssen passende Services finden und verwenden.  
Das Web Service Framework und Representional State Transfer (REST) bieten Frameworks für solche verteilten Anwendungen.

In keinem Foliensatz

Wie kann das Zeitmultiplexverfahren unterteilt werden?

Beim Zeitmultiplexverfahren wird zwischen synchronem und asynchronem Verfahren unterschieden. Weiters kann das asynchrone Verfahren noch in Verfahren mit kontrolliertem Buszugriff (Zentrale Busteilung, Dezentrale Busteilung) sowie Verfahren mit zufälligem Buszugriff (Carrier Abtastung) unterteilt werden.

* synchrones Verfahren: Beim synchronen Verfahren wird jedem Sender ein fester Zeitabschnitt zur Übertragung zugewiesen. Das heißt, jeder Teilnehmer kann seine Daten mit derselben Datenübertragungsrate übertragen. Wenn ein Teilnehmer jedoch keine Daten zu übertragen hat, dann bleibt der Zeitslot ungenutzt.
* asynchrones Verfahren: Beim asynchronen Verfahren wird der Nachteil des synchronen Verfahrens (ungenutzte Zeitslots) vermieden. Dies geschieht dadurch, indem nur jene Teilnehmer auf den Übertragungskanal zugreifen dürfen, die auch tatsächlich etwas zu übertragen haben. Um die Datenpakete am Ende des Übertragungskanals auch wieder den entsprechenden Sendern zuweisen zu können, wird ein Channel Identifier zu jedem Datenpaket hinzugefügt.
* CSMA (Carrier Sense Multiple Access): Bei diesem Verfahren beobachten alle Teilnehmer den Status des Übertragungskanals und übertragen nur dann Daten, wenn gerade kein anderer Teilnehmer sendet, der Kanal also frei ist. Weiters existieren für CSMA verschiedene Ansätze, die Kollisionen erkennen und behandeln (Collision Detection, Collision Resolution) bzw. vermeiden (Collision Avoidance).
* CSMA/CD (Carrier Sense Multiple Access/Collision Detection): erkennt Kollisionen und versucht die Konkurrenzsituationen durch Abbruch der aktuellen Sendung und anschließende unterschiedliche (zufällige) Sendeverzögerung zu vermeiden.