

## 1 Einführung in das Protokoll

INTERBUS ist ein digitales serielles Kommunikationssystem zur Anbindung von Sensoren und Aktoren an Automatisierungssysteme (z. B. speicherprogrammierbare Steuerungen). Typische Geräte sind einfache digitale Ein- und Ausgänge, Sensoren, Meßaufnehmer aber auch komplexe Geräte wie Regler, Antriebs- und Schweißsteuerungen. INTERBUS ist besonders geeignet für die Anwendung in der Fabrikautomation, jedoch nicht auf diese beschränkt.

Das Protokoll ist entsprechend dem ISO/OSI-Modell in drei Schichten aufgebaut:

- Physical Layer (Schicht 1)  
Hier werden sowohl die Zeitbedingungen, wie Baudrate, zulässiger Jitter usw., als auch die Formate für die Leitungskodierung festgelegt.
- Data Link Layer (Schicht 2)  
Hier wird die Integrität der Daten sichergestellt.
- Application Layer (Schicht 7)  
Definiert die Schnittstelle zur Anwendung.

Die Haupteigenschaften sind:

- Master/Slave Konzept
- Ringstruktur mit Summenrahmenprotokoll
- Baumstruktur für die Geräteanordnung
- geringer Protokollaufwand
- Unterscheidung zwischen Ein-/Ausgangsdaten (Prozeßdaten) und Nachrichten (Parameter)
- Prozeßdaten und Parameter werden im Summenrahmen gleichzeitig übertragen
- Parameterdatenübertragung beeinflusst die Prozeßdatenübertragung nur mit einer geringen, vordefinierten Zeit
- Gestaffelte Fehlererkennungsmechanismen, CRC, Stopbit, Time-Out und Loopback-Word (überprüft den Summenrahmen)

Die Anwendervorteile sind:

- feste, berechenbare Zykluszeit für eine konkrete Konfiguration
- zeitlich konstante und konsistente Übertragung der Prozeßdaten
- einfache Installation und Wartung, da keine Geräteadressen benötigt werden
- gute Diagnosemöglichkeit, da Fehler direkt dem betroffenen Gerät / Bussegment zugeordnet werden können
- einfache und komplexe Geräte können beliebig in einer Konfiguration gemischt werden

## 2 Physical Layer (PhL - Schicht 1)

Der Type 8 PhL ist entsprechend dem Modell der IEC 61158-2 in drei Ebenen modelliert:

- medium-independent sublayer (MIS)
- medium-dependent sublayer (MDS)
- medium attachment unit (MAU).

Vom MIS werden die PhL Dienste unabhängig von der darunter liegenden physikalischen Schnittstelle bereit gestellt. Der MIS bildet die Schnittstelle zwischen DLL und PhL.

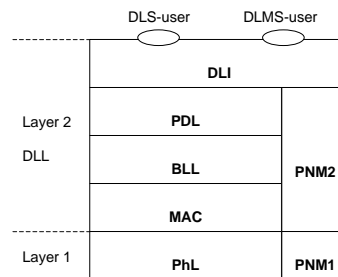
Der Type 8 PhL spezifiziert die MAU-Typen:

- drahtgebunden (ANSI TIA/EIA-422-B)
- Lichtwellenleiter (Polymerfaser und HCS-Faser)

Für beide Medien sind full-duplex Übertragung mit Non Return to Zero (NRZ) Kodierung sowie verschiedene Baudraten zwischen 500 kbit/s bis zu 16 Mbit/s definiert.

### 3 Data Link Layer (DLL - Schicht 2).

Der Type 8 DLL ist in vier Ebenen modelliert (siehe nachfolgendes Bild). Das spezifizierte Protokoll ist voll kompatibel zum INTERBUS Protokoll.



Das Data Link Layer Interface (DLI) stellt die Dienstprimitiven für den DLL Anwender, d. h. den Application Layer, und für den DL Management Benutzer, d. h. System Management, bereit.

Die folgenden Dienste sind definiert:

1. Data Link Service Anwender
  - Put Buffer Schreiben der zyklischen Prozeßdaten
  - Get Buffer Lesen der zyklischen Prozeßdaten
  - Buffer received signalisiert den einwandfreien Empfang eines Summenrahmens
  - Normal Data TransfERNachrichtenorientierte Datenübertragung für Parameterdaten
2. Data Link Management Service user
  - Reset Rücksetzen der DL Protokollschicht
  - Set value Schreiben von DL Konfigurationsparametern
  - Read value Lesen von DL Konfigurationsparametern
  - Event signalisiert DL Ereignisse an den DL Management Anwender
  - Get current configuration Lesen der augenblicklichen (physikalischen) DL Konfiguration (master only)
  - Get active configuration Lesen der aktiven DL Konfiguration (master only)
  - Set active configuration Schreiben der aktiven DL Konfiguration (master only)

Der Peripherals Data Link (PDL) wandelt die über den DLI eingehenden Dienstanforderungen in die zyklischen Dienste des BLL. Hier werden die Service Data Units (SDU) von mehreren Dienstanforderungen zu einer gemeinsamen für den BLL zusammengestellt und umgekehrt die SDU vom BLL aufgeteilt. Der PDL ist zuständig für die Verwaltung des Empfangs- (receive update memory) und des Sende-Buffers (transmit update memory), in denen die Daten eines Summenrahmens gehalten werden.

Der Basic Link Layer (BLL) ist in den Geräten für die Steuerung des Buszugriffes zuständig. Der BLL stellt, im Master die Dienste für die Datenzyklen, mit Daten, zur Verfügung. Im Slave stellt der BLL sicher, daß die richtig empfangenen Daten zum PDL gelangen und das Daten vom PDL gesendet werden.

Der Medium Access (MAC) überwacht den Zugriff auf die Übertragungsstrecke und sichert die empfangenen und gesendeten Daten mit einem 16-bit CRC Polynom.

INTERBUS verwendet ein Summenrahmenverfahren, hierbei wird jedem Teilnehmer ein, seiner Datenbreite angepasster, Zeitschlitz zugeordnet. Die Reihenfolge der Datenpakete im Summenrahmen entspricht der physikalischen Lage des Gerätes im Ring. Die zyklischen Prozeßdaten und auch die nicht zyklischen Parameterdaten, sofern solche zu übertragen sind, werden gleichzeitig übertragen. Geräte mit Parameterdaten belegen dafür eine feste Länge von 2, 4 oder 8 Octets (Bytes) im Summenrahmen, längere Nachrichten werden im DLL segmentiert.

Die Datenübertragung erfolgt Vollduplex. Ein Übertragungszyklus beginnt mit einer Datensequenz. Diese Datensequenz enthält sendeseitig das Loopback-Word, gefolgt von den Ausgangsdaten. Während diese Datenausgabe vom Master zu den Slaves durchgeführt wird, erfolgt gleichzeitig die Übertragung der Eingabedaten von den Slaves an den Master. Nachdem so der gesamte Summenrahmen ausgegeben und gleichzeitig wieder eingelesen wurde, sind alle Ausgangsdaten in den einzelnen Teilnehmer richtig positioniert.

Die sich anschließende 32-Bit lange Frame-Check-Sequence (FCS) dient der Überprüfung der in der Datensequenz übertragenen Nutzdaten. Diese Datensicherung erfolgt durch ein Prüfsummenverfahren mit einem 16-bit CRC-Polynom. Bedingt durch die physikalische Punkt-zu-Punkt Struktur findet der Datensicherungsmechanismus immer zwischen zwei benachbarten Teilnehmern statt. Der Austausch und Vergleich der CRC Polynom-Reste erfolgt, gesteuert durch die FCS, gleichzeitig zwischen allen Geräten, so daß nur der Platz für ein 16-bit CRC Wort für den gesamten Summenrahmen benötigt wird. In den restlichen 16-Bit der FCS wird der Prüfsummenstatus übertragen. Sofern der Prüfsummenstatus keine Fehlererkennung enthält wird die „Buffer Received Indication“ erzeugt.

Jeder Busteilnehmer hat einen Identifikationscode, der Auskunft über die Teilnehmerart und dessen Datenlänge im Summenrahmen gibt. Zusätzlich zu dem oben beschriebenen Datenzyklus ist im DLL noch ein Identifikationszyklus für das Busmanagement, definiert, mit welchem der Identifikationscode von angeschlossenen Slaves gelesen werden kann.

#### **4 Application Layer (Schicht 7)**

Der Type 8 Application Layer (AL) ist auf Basis des gemeinsamen Application Layer Modells (siehe IEC 61158-5 und IEC 61158-6) spezifiziert. Sowohl das Objekt-Modell als auch die Struktur der Protokollmaschinen werden angewandt. Type 8 verwendet eigene optimierte Encoding-Regeln (PER – Peripherals Encoding Rules), damit wird sichergestellt, daß die PDUs kompatibel zur bestehenden INTERBUS Spezifikation sind.

Type 8 spezifiziert zwei Übertragungsarten:

- Zyklische Datenübertragung (für Prozeßdaten) in einer hocheffizienten Weise auf Basis des push publisher/subscriber Modells  
Verwendete ARPM: Buffered Network-Scheduled Uni-directional (BNU)
- Übertragung von nicht zyklischen Nachrichten (für Parameterdaten) auf Basis eines Client/Server Modells, dabei können sowohl Master als auch Slave als Client oder Server arbeiten.  
Verwendete ARPM: Queued User-triggered Bi-directional with Flow Control (QUB-FC)

ARPM: Application Relationship Protocol Machine

Ein Type 8 Application Layer Anwender (Applikationsprozeß) erreicht die AL Dienste über die folgenden Application Service Elements (ASE):

1. Object Management ASE - Dienst:
  - Get Attributes
  
2. Application Process ASE - Dienste:
  - Identify
  - Get Status
  - Initiate
  - Terminate
  - Reject

3. Application Relationship ASE - Dienste:
  - AR-Unconfirmed Send
  - AR-Establish
  - AR-Abort
  
4. Variable ASE - Dienste für einfache und Array Variablen:
  - Read
  - Write
  - Information Report

Verwendet werden die definierten Datentypen:

- Boolean, BitString
- Integer8, Integer16, Integer32, Unsigned8, Unsigned16, Unsigned32
- Float32
- BinaryDate, BinaryDate2000
- TimeOfDay, TimeDifference
- VisibleString, OctetString

5. Function Invocation ASE Dienste:
  - Start
  - Stop
  - Resume
  - Reset

Die folgende Abbildung zeigt das Zusammenspiel zwischen den Protokollmaschinen und Diensten

