

# Typ 6 SwiftNet

von Robert Crowder

## ***Eine kurze Einführung in SwiftNet***

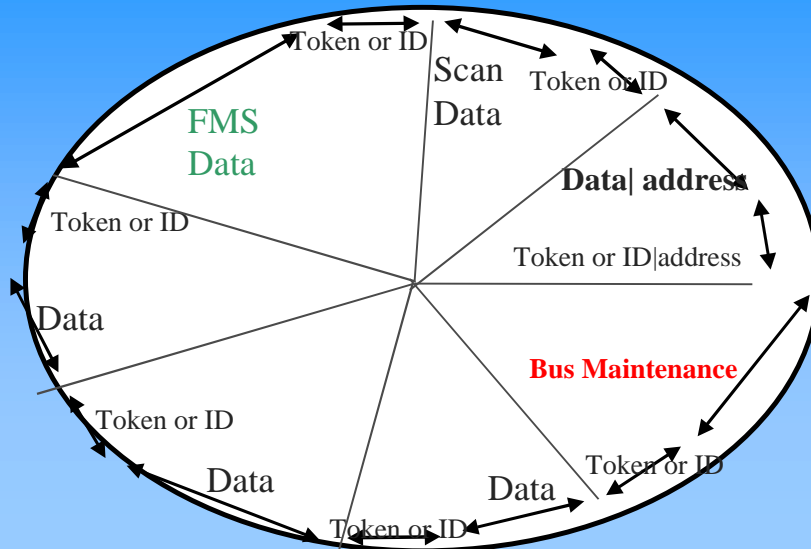
SwiftNet ist ein moderner, leistungsfähiger Feldbus. Er wurde entworfen, um die Anforderungen des Unternehmens Boeing für einen wirklich synchronen Bus mit hoher Geschwindigkeit für die Verwendung in kommerziellen Flugzeugen zu befriedigen (bis zu 120.000 Abtastungen/sek.). SwiftNet ist der Typ 6 der Internationalen Norm IEC 61158.

Indem SwiftNet die lokalen Zeitbasen aller Feldbusknoten synchronisiert und für Zeitstempelungen aller Alarme und Meldungen an ihrer Quelle sorgt, um jeglichen Zeitversatz auszuschließen, geht es deutlich über diese Aufgabenstellung hinaus. Dieses beseitigt wirksam Zeitversatz (Jitter) und Unregelmäßigkeiten, die in einigen anderen Feldbussen auftreten.

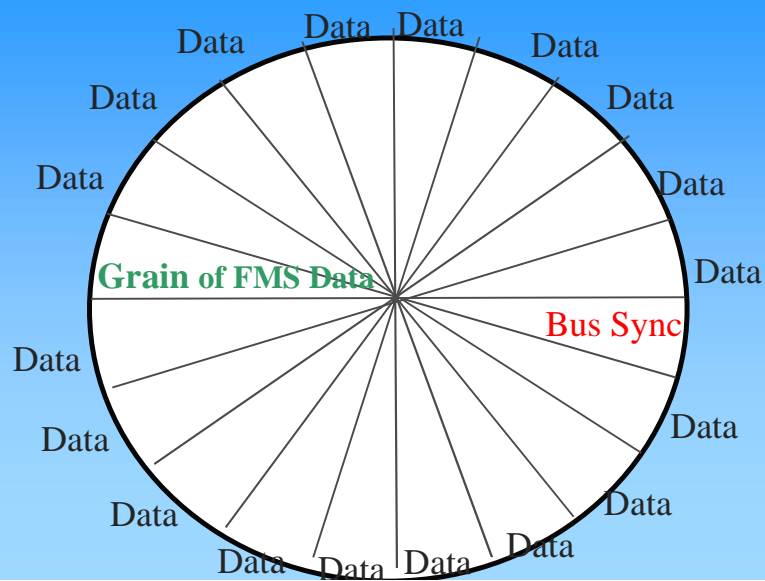
SwiftNet sorgt für synchronisierte globale oder einzelne Auslösungen der Aktion von Geräten. Weil diese Auslösungen unter Verwendung der lokalen Zeit eines Gerätes erfolgen, vermindern sie nicht die Datenübertragungsrate auf den Bus.

Diese positiven Verhaltenseigenschaften von SwiftNet sind in dem strukturiertem, synchronen Busprotokoll (TDMA) begründet, das alle Adress- und Steuerfelder von den Datentelegrammen entfernt und synchronisierte Tätigkeiten aller Knoten sicherstellt. Mehr als 99 % der Bandbreite werden bei SwiftNet für die Datenübertragung verwendet. Ein Vergleich der TDMA-Wirkungsweise mit herkömmlichen Feldbussen wird durch die beiden untenstehenden Bilder vereinfacht dargestellt, wobei die Frequenz der Synchronisierungstelegrammrahmen auf den SwiftNet-Bus weniger als 1/256 sind, auch wenn Quarzkristalle geringerer Genauigkeit verwendet werden.

## Traditional Message Based Bus



## Structured SwiftNet Bus



Das Producer-Consumer-Prinzip, bei dem mehrere Knoten gleichzeitig die gleichen Daten abhören können, wird zur Verbesserung sowohl der Leistung als auch der vielfältigen Anwendbarkeit verwendet. SwiftNet unterstützt viele Anwendungsprotokolle einschließlich FMS. Ein sehr fein aufteilbares Protokoll unterstützt Anwendungsdaten in Pakete, die bis zu 1792 Bytes lang sein können. Die Anwendungsdatenpakete stören nicht die Zeitsteuerung der Datenübertragung.

Die Datenübertragungsraten für das Physical Layer nach EIA 485 betragen 5 Mbit/sek. bis 78 125 kbits/sek., wie in der u. s. Tabelle dargestellt. Sie stellt ebenfalls die größte Segmentlänge und die größte Länge eines Busstranges ohne Verwendung von Bridges als Funktion der Datenrate:

<b>Datenrate</b>	<b>Höchste Segmentlänge (m)</b>	<b>Höchste Buslänge ohne Verwendung von Bridges (m)</b>
5 Mbit/s	100	400
2.5 Mbit/s	100	400
1.25 Mbit/s	100	400
625 Kbit/s	225	900
312.5 Kbit/s	350	1400
156.25 Kbit/s	600	2400
78.125 Kbit/s	1200	4800

Faseroptischer Medien können ebenfalls eingesetzt werden. Die Buskonfiguration wird vollständig durch den Benutzer festgelegt. Sie wird bei jedem Einschalten des Busses automatisch abgeprüft. Dies erlaubt eine leichte Anpassung der Buskonfiguration bei der Umstellung auf eine andere Aufgabe, wie es in Laboratorien oder Prüffeldern des öfteren vorkommt.

SwiftNet kann ebenfalls Netzwerke mit einer hohen Anzahl von Knoten und Variablen, die durch Brücken untereinander verbunden sind, synchronisieren. Diese Synchronisation zwischen Knoten in Netzwerken, die untereinander durch Bridges verbunden sind, wird auf 50 µsek gehalten.

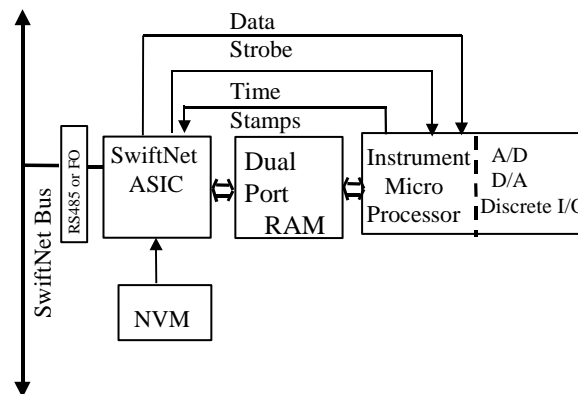
### **SwiftNet und unternehmensweite Netzwerke**

SwiftNet ist eine ideale Schnittstelle zwischen Geräten und einem unternehmensweiten Netzwerk, das das Internetprotokoll oder andere Netzwerkprotokolle benutzt, da es die Steuerung des Feldbusses und seiner Synchronisation ohne Einwirkung von Software aus den oberen Schichten aufrecht erhält. Dieses beseitigt wirksam den Einfluss von Übertragungstotzeiten auf dem Netzwerk durch eingeschränkte Leistungsfähigkeit von Geräten oder Berichterstattung und hilft, dem Einfluss von Nichtrealzeit-Betriebssystem wie Windows NT entgegenzuwirken.

## SwiftNet Architektur

SwiftNet ermöglicht durch die Verwendung eines SwiftNet-ASIC mit einer DPRAM-Schnittstelle für jeden Mikroprozessor in einem intelligenten Feldgerät eine Netzwerkarchitektur zu niedrigen Kosten (siehe Bild unten).

## SwiftNet Device as Conventional Smart Instrument



Der SwiftNet-ASIC führt alle Realzeitfunktionen aus, um die Interoperabilität zwischen Geräten verschiedener Hersteller sicherzustellen. Dieser Ansatz vermindert die Kostengröße und Leistungsaufnahme der Geräte. Übergeordnete Geräte verwenden den SwiftNet-ASIC zusammen mit einer Speichereinheit für jedes Feldgerät. Bei steigender Geräteanzahl kann ein übergeordnetes Gerät erhebliche Ansprüche stellen.

## SwiftNet Anwendungen

SwiftNet ist für Anwendungen, die eine oder mehrere Eigenschaften aufweisen, vorgesehen.

- hohe Geschwindigkeit
- synchronisierte Datenerfassung und -steuerung
- viele Busteilnehmer
- große räumliche Ausdehnung
- Synchronisierung der Schnittstellen mit einem unternehmensweiten Netzwerk

Einige Anwendungen, bei denen SwiftNet deutliche Vorteile hat:

- Prüfanordnung im fliegenden Flugzeug
- Flugsimulation
- drahtgebundene Fernsteuerung von Flugzeugen
- numerisch gesteuerte Werkzeugmaschinen
- betriebkomplexer Maschinen und Systeme
- Transportsysteme

### ***Konformität mit SwiftNet***

Die Konformität wird in verschiedenen Ebenen, die unten angegeben sind, festgelegt. Dies ermöglicht verschiedene Konformitätsklassen, die auf die jeweiligen Aufgaben zugeschnitten sind. Damit werden die Kosten und die Komplexität der SwiftNet-Knoten verringert.

- a. Eigenschaften, die in jedem SwiftNet-Knoten ausgeprägt sein müssen
- b. zusätzliche Eigenschaften, die in jedem SwiftNet-Knoten, der eine höhere Rolle in einem SwiftNet-Netzwerk spielt, ausgeprägt sein müssen
- c. zusätzliche Eigenschaften, die in jedem SwiftNet-Knoten, der eine bestimmte Kanalart oder die Auslösefunktion unterstützt, ausgeprägt sein müssen
- d. zusätzliche Eigenschaften, die für die Konformitätsklasse, zu der dieser Knoten gehört, ausgeprägt sein müssen

### ***SwiftNet-Eigenschaften***

SwiftNet ist ein sehr hochleistungsfähiger Feldbus mit hoher, synchronisierter Abtastrate. SwiftNet erlaubt die Hochgeschwindigkeitsdatenübertragung von analogen Ein- und Ausgängen (bis zu 120 016 bit Abtastungen pro sek bei einer Datenrate von 5 Mbs), die auf einem gemeinsamen Bus mit einer großen Anzahl diskreter Ein-/Ausgabe-Punkte und FMS-Übertragung betrieben werden.

SwiftNet verringert und begrenzt die zeitlichen Schwankungen (Jitter) sowie die ungünstigen Auswirkungen von Protokoll Overhead auf synchronisierte Abtastung. Die ungünstigste Abweichung von der synchronisierten Operation beträgt 25  $\mu$ sek.

Realzeitfähiges SwiftNet kann in einfachen Geräten mit dem gewöhnlichen SwiftNet-ASIC (keine weitere Software benötigt) vollständig verwirklicht werden, bei einfacher Implementierung tatsächlicher Interoperabilität und geringen Gerätekosten.

### ***SwiftNet-Eigenschaften***

SwiftNet hat die Eigenschaften, die in modernen Feldbussen benötigt werden, so wie es nachstehend dargestellt ist.

- das benutzte Medium bestimmt die höchste synchronisierte Abtastungsrate
- Auslöseimpuls, um die Geräte untereinander und mit der Abtastung zu synchronisieren
- kleinste zeitliche Schwankungen
- ein Producer von abgetasteten Daten mit N-Konsumer sind möglich
- konfigurierbare Länge der abgetasteten Daten

- Zeitstempelung von Alarmen und Fehlermeldungen in den Geräten
- gemeinsame Zeitbasis ist in allen Geräte verfügbar
- unterstützt übliche Anwendungsprotokolle, FMS und andere bis zu 1792 bytes von Anwendungsdaten
- kann vollständig durch den Endbenutzer konfiguriert werden
- es wird keine SwiftNet-Software benötigt, um abgetastete Daten zu übertragen oder zu konfigurieren oder die Synchronität des Busses und der Auslöseimpulse sicher zustellen
- unterstützt große synchronisierte Netzwerke, die mit Bridges untereinander verbunden sind
- unterstützt netzwerkweite Stempelung mit einer gemeinsamen Zeit

## ***Verhältnis zu anderen Spezifikationen***

SwiftNet ist Typ 6 der IEC 61158. Es ist nicht in EN 50170 und EN 50254 enthalten.

Der Physical Layer von SwiftNet war ein Anhang zur 2. Ausgabe der IEC 61158-4. Es wird jetzt Bestandteil der 3. Ausgabe der IEC 61158-2.

## ***Technische Übersicht über SwiftNet***

### *Übersicht über das Physical Layer*

SwiftNet benutzt ein Physical Layer nach EIA 485, bei dem jedes Bussegment an beiden Enden mit seinem Wellenwiderstand abgeschlossen wird.

### **Datenübertragungsraten und Topologie**

Der Physical Layer nach EIA 485 ermöglicht Linear-, Baum- und Stern und Verbindungen von diesen.

Die höchste Anzahl von Knoten pro Physical Layer-Segment ist 32. Die höchste Anzahl von Knoten in einem lokalen Link unter Verwendung von Repeatern ist 122 für eine lineare Bustopologie und 922 für Sterntopologie.

Ein Zweidrahtbus ohne galvanische Trennung darf verwendet werden.

D-Subminiatursteckverbinder mit neun Stiften werden in der Baumkonfiguration verwendet, die den Verlust der Kommunikation verhindert, wenn ein Knoten entfernt wird.

### **Kodierung und Datenintegrität**

Zur Verringerung von Fehlern und zur Gewährleistung von speziellen Endzeichen für den Datenrahmen zur Synchronisation des Busses wird die Manchester-Kodierung verwendet. Zwei Rahmenbegrenzungszeichen werden jeweils nur hierfür verwendet. Das eine bedeutet das Ende eines Buszyklus, während das andere die Zeitabstimmung ermöglicht, so dass auch Quarzkristalle niedriger Genauigkeit in den SwiftNet-Knoten verwendet werden können.

Ein 16 bit FCS-Polynom gewährleistet im Data Link Layer eine Hamming-Distanz von 4 für alle Telegrammrahmen und eine Hamming-Distanz von 5 für kurze Telegrammrahmen.

### *Übersicht über das Datalinklayer*

Das SwiftNet-Datalinklayer benutzt eng synchronisierte TDMA-Prozeduren für den Medienzugriff und weiterhin stille Rahmenadressen und Steuerfelder, um die Abtastrate für ihre Realzeitdaten zu erhöhen. Diese Struktur ermöglicht bei einer gegebenen Datenübertragungsrate eine wesentliche höhere Rahmenübertragungsrate als ein nachrichtenbasierendes System.

Jeder Buszyklus wird in Zeitscheiben gleicher Länge unterteilt. Typischerweise werden  $2^{16}$  Scheiben pro Buszyklus vorgesehen. Diese Scheiben werden bei Intervallen von  $2^N$  durch die Konfiguration in den Kanälen zugewiesen. Jeder Kanal ist wiederum ist einer Dateneinheit zugewiesen. Diese Dateneinheit kann eine

abgetastete Variable oder ein Bruchteil einer längeren Variable oder FMS-Nachricht sein. Wenn N= 1 ist, kann genau die Hälfte der Bandbreite für den synchronisierten Zugang zu einer Variablen verwendet werden. Der Betrieb eines wesentlich verkürzten Busses ist im untenstehenden Bild dargestellt, wobei der Kanal 1 bei der höchsten Abtastrate abgetastet wird. Kanäle B und C mit 25 % und D, E und F mit 12,5 %.

Start abbreviated Bus Cycle 1 shown for an artificial Bus-Cycle with 16 Slots and 3 Scan Classes the typical Bus-Cycle has 64K (65536) Slots and 15 Scan Classes															End	Start of Bus Cycle 2		
Slot 0	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7	Slot 8	Slot 9	Slot 10	Slot 11	Slot 12	Slot 13	Slot 14	Slot 15	Slot 0	End-user Selected Scan Rates	
Ch A		Ch A		Ch A		Ch A		Ch A		Ch A		Ch A		Ch A	Bus Sync period	Ch A		Scan-Class 1, Channel A  @ max. Scan Rate
	Ch B				Ch C				Ch B				Ch C		Bus Sync period			Scan-Class 2, Channels B, C  @ 25% of max. Scan Rate
			Ch D				Ch E Syn Ch				Ch F				Bus Sync period			Scan-Class 3, Channels D ,E, F  @ 12.5% of max Scan Rate

Ein bestimmter ausgewählter Knoten auf jedem Link, der Anführer, synchronisiert alle Knoten mit Hilfe von Bussynchronisationsrahmen, die durch eine bestimmte Menge von Physical Layer-Rahmenbegrenzern dargestellt werden.

Redundante übergeordnete Geräte, von denen jedes der Anführer werden kann, werden ebenfalls unterstützt.

Der Anführer des Top-Links eines Netzwerkes, das mit Hilfe von Bridges mit anderen verbunden ist, synchronisiert das gesamte Netzwerk mit Hilfe der Physical Layer-Synchronisationssignale zwischen den Bridges.

### Leistungsfähigkeit

SwiftNet kann bis zu 120 016 bit Abtastungen/sek. über den 5 Mbits/sek. EIA 485 Physical Layer übertragen mit einem Jitter weniger als 25 µsek. Die Uhr eines einzelnen Knotens ist bis auf 12,5 µsek bei 5 Mbits/sek. genau.

### Dienste des Data Link Layer

Das Data Link Layer stellt eine Übertragung mit hoher Geschwindigkeit und Synchronisation (auf einen einzelnen Link) von Daten sowohl in der Prozess- als auch in der Fertigungsindustrie zur Verfügung, zusammen mit einer Unterstützung für allgemeine Anwendungsprotokolle, wie FMS, IEEE 1451-1 und IEEE 1451-2.

- Realzeitübertragungen benutzen das Producer/Consumer-Modell, so dass verschiedene Knoten gleichzeitig die gleichen Dateneinheiten empfangen können
- das Data Link Layer zerteilt und überträgt lange Anwendungsnachrichten innerhalb der Byte, die für die Übertragung kurzer Abtastdaten vorgesehen ist, so dass diese Nachrichten nicht den Realzeitverkehr stören
- wiederholte Übertragsversuche von fehlerhaften Nachrichten werden sowohl auf den GPA- und GPC-Kanälen unterstützt
- das Aushandeln der Qualität des Übertragungsdienstes und Anforderungs-Antwort-Operation werden ebenfalls durch die GPC-Kanäle unterstützt

Das Data Link Layer benutzt Verbindungsorientiertheit für alle Typen von Betriebsdatenübertragung.

SwiftNet sieht 4 Typen von Data Link Layer-Verbindungen vor:

<b>Connection Type</b>	<b>End Point Interaction</b>	<b>Range of DLPDU sizes - octets</b>	<b>Typical Usage</b>
SCAN	Producer Consumer	0, 2 to 33	typical scanned variable
Extended SCAN (EXSCAN)	Producer Consumer	0 to 1792	long scanned variable
General Purpose Confirmed (GPC)	Peer to peer	0 to 1792	request / response, confirmed request, or event notification (with time stamp)
General Purpose Acknowledged (GPA)	Peer to peer	0 to 1792	event notification (with time stamp) - not used with FMS

Verbindungen werden implizit durch die Zeitscheibenummer der ersten Scheibe, die sie auf jedem Link besetzen, identifiziert. Dieses vereinfacht die Konfiguration und verringert den Kommunikationsoverhead.

SwiftNet sieht zusätzliche Realzeitdienste vor einschließlich:

- synchronisieren der lokalen Zeiten von allen Geräten auf einem einzigen Bus
- Zeiten werden ebenfalls zwischen Netzwerken, die durch Bridges verbunden sind, synchronisiert
- synchronisierte Träger für lokale und globale Aktionen
- Zeitstempelung von Ereignisnachrichten mit Hilfe der Buszeit an dem Punkt, wo sich das Ereignis tatsächlich ereignete.

### **Data Link Layer-Konfiguration**

Alle Datenübertragungen, Auslöseimpulse, Zeitstempelungen und Übertragungspfade durch Bridges in einem SwiftNet-Netzwerk (Netzwerk mit Bridges oder ein einzelnes Link) werden beim Aufstarten des Busses durch den Anführer mit Hilfe einer Knotenliste konfiguriert, die durch den Endnutzer vorbereitet wurden.

Die Anwesenheit und ordnungsgemäß Konfiguration jedes Gerätes der Knoten-Liste und die Abwesenheit von nicht aufgeführten Geräte wird während der Aufstartprozedur des Busses bestätigt.

SwiftNet benutzt verbindungslose Dienste für die Buskonfiguration. Verbindungslose Linkadressen können eins bis drei Okette lang sein, so dass eine große Anzahl von Links und Knoten, die in einem Netzwerk durch Bridges untereinander verbunden sind, unterstützt wird.

### **Übersicht über den Application Layer**

Das SwiftNet Data Link Layer kann mit jedem Anwendungsprotokoll verwendet werden, das von seinem Data Link Diensten Gebrauch macht.

SwiftNet sieht auch eine große Breite von Diensten des Application Layer vor, so wie sie in Typ 9 FMS festgelegt sind, mit Ausnahme von zwei nichtbenötigten Diensten. Diese Dienste werden in Anwendungsdienstelemente (Application Service Elements= ASEs) wie folgt unterteilt:

- a. Virtual Field Device-ASE einschl. Status nicht angeforderter Status und Identifikation der VFDs

- b. Objektverzeichnis-ASE einschl. Herholen und Hinterlegen von Objektbeschreibung
- c. Umgebungsmanagement-ASE einschl. des Herbeiführens und des Verwerfens und des Zurückweisens von Diensten
- d. Anwendungsbeziehungs-ASE einschl. des unbestätigten Sendens, bestätigten Sendens, Verwerfens, Aufbauens, Erreichen von gepufferten Nachrichten, Status und XON-OFF-Dienste
- e. Zugang zu Variablen-ASE einschl. Lesen, Schreiben und Berichten von einfachen Array und Berichtsvariablen und Variablenlisten
- f. Ereignis-ASE einschl. Benachrichtigung, Bestätigung, Änderung der Anzeigen von Bedingungen und Zeitstempelung von Ereignissen
- g. Lade-Bereich-ASE einschl. Heraufladen und Herunterladen von Bereichen
- h. Funktionsaufruf-ASE einschl. Starten, Stoppen, Beenden, Rücksetzen, Beseitigen, Schaffen und Löschen von Funktionsaufrufen

Informationsberichte und Ereignisbenachrichtigungen erhalten die Realzeiteigenschaften des SwiftNet Data Link Layer.