

Systems Engineering & Consulting

für Embedded Applikationen

MIXED
MODE

FlexRay V1.0

Referent: Kurt Veit

FH-Schweinfurt 24.04.2007



- FlexRay Historie und Konsortium
- Vergleich LIN / CAN / FlexRay
- Anwendungsgebiete FlexRay
- FlexRay Protokoll
- FlexRay Controller / Transceiver
- FlexRay Design Tools
- FlexRay Analyzing Tools
- Ausblicke
- Links zu weiteren Informationen

- Diskussion

Das FlexRay Konsortium ist ein Zusammenschluss von Automobilherstellern, Halbleiter-Herstellern, Systemhäusern (HW, SW, Toolchains, Testtools)

Motivation:

- Steigende Anzahl der Steuergeräte (Powertrain, Safety, Chassis)
- Der Markt braucht einen skalierbaren Standard
- Höhere Datenraten werden erforderlich
- Es wird ein streng deterministisches und fehlertoleranter System Bus für Hochgeschwindigkeit und harte Echtzeit Anforderungen gefordert

Ziele:

- Entwicklung eines offenen Standards (Hohe Stückzahlen, Reuse, Second Source)
- Einführung der Technologie am Markt für Jedermann
- Entwicklung neuer Technologie auf zukunftsweisenden Applikationen und Domains (passive und aktive Sicherheitsanwendungen, neue Komfortfunktionen, ...)

- Gründung 1999 durch BMW und Daimler-Chrysler

Core Members:

- BMW, Bosch, Daimler-Chrysler, Freescale, GM, NXP, VW

Premium Associate Members (Auswahl):

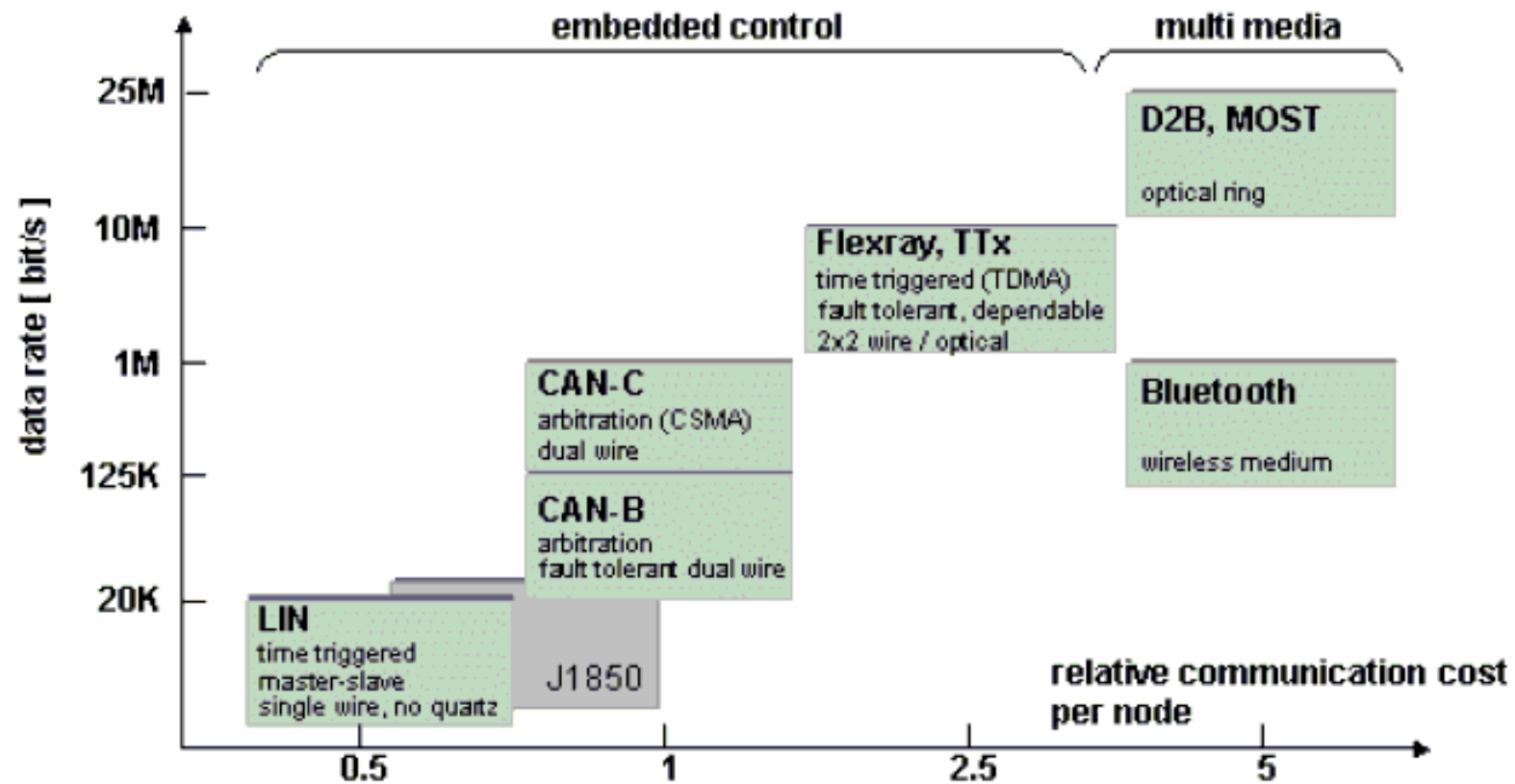
- Electrobit, AS, Conti Decomsys, Delphi, Elmos, Fiat, Ford, Honda, Nissan, Renault, TTAutomotive Software, Tyco, TZM, Vector-Informatik

Associate Members (Auswahl):

- Agilent, dSPACE, ETAS, Fujitsu, Göpel, Hella, IAV, K2L, Renesas, Softing, TDK, Toyota, TRW, Xilinx, Yokogawa

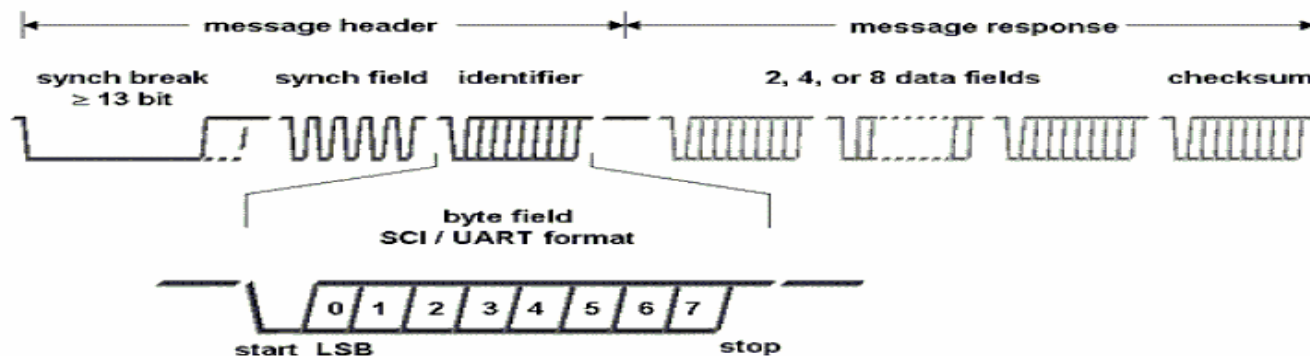
- FlexRay 2.1 Spezifikationen werden 2007 verabschiedet
- Die FlexRay 2.1 Konformitätstests werden sich 2007 / 2008 etablieren

Automotive Busse: Vergleich Performance - Kosten - Anwendungsbereiche



LIN

- Anwendungsbereich: z.B. Sitze, Schiebedach, Sensor Cluster
- Master / Slave (Nachrichtenpaar)
- Serielle Kommunikation über UART (Encoding NRZ 8N1)
- Kein Quarz / Eindraht / Geschwindigkeit ca. 20 KBit
- Fehlererkennung durch Parity Bit und Checksumme (Master / Slave)
- HW einfach und kostengünstig



Probleme LIN

- 13bit Sync macht Probleme bei Konfiguration „einfacher“ UART's
- Unterschiedlichste UART Implementierungen im Bordnetz (Bei Neuentwicklungen sollten nur μ Controller mit speziell entwickelten integrierten LIN Controller verwendet werden)
- LIN 2.1 nur mit komplexer Software realisierbar
- Hoher Integrationsaufwand bei mehreren LIN Knoten (Physikal Layer, Flankensteilheit, Messungen bei versch. Spannungen , Tests, ...)

CAN

- Anwendungsbereich: Body und Komfort Steuergeräte
- Multimaster / mehrere Empfänger
- Differentielles Signal (Encoding NRZ 5bit)
- Geschwindigkeit max. 1 MBit (500 KBit in der Praxis)
- Fehlererkennung über CRC, autonome Abschaltung fehlerhafter Knoten
- Garantierte Latenzzeit möglich
- Systemweite Datenkonsistenz
- Flashen der SW über CAN

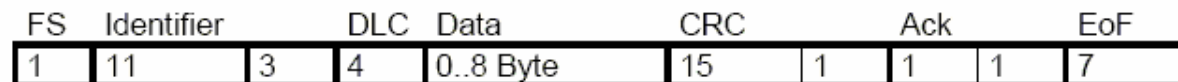


Abb. 6-11 : Erweitertes Datenformat (2.0B)

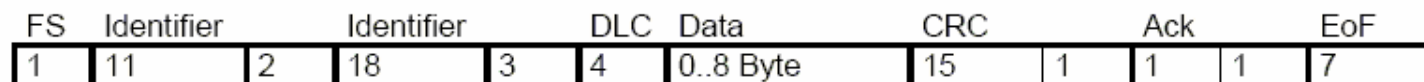


Abb. 6-12 : Aufbau der Telegramme bei CAN2.0A und CAN2.0B

Probleme CAN

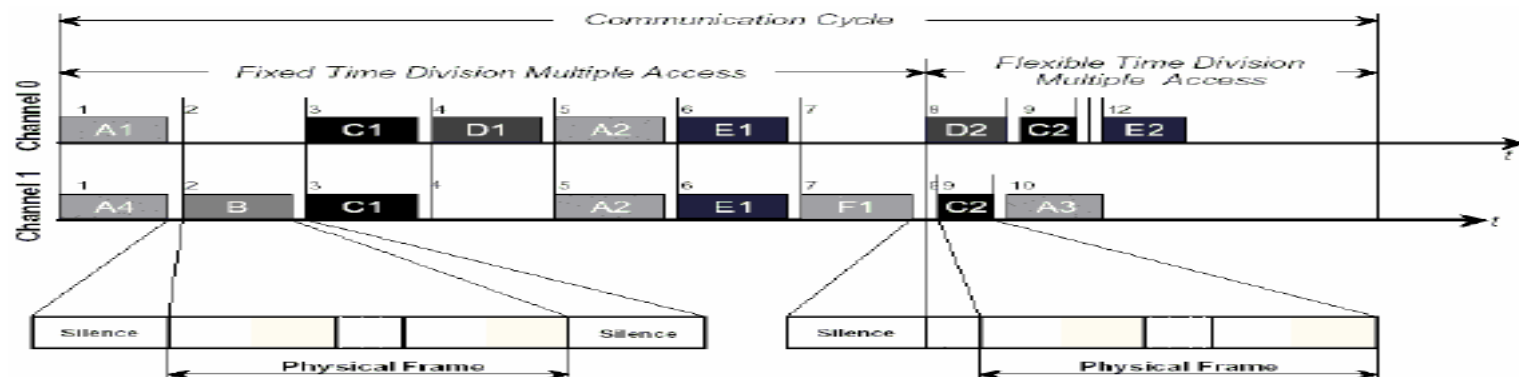
- Latenzzeit der Nachrichten u.a. abhängig von Anzahl der Nachrichten im System, ID und Buslast
- TTCAN Chips sind zu spät auf den Markt gekommen
- Protokoll-Overhead mehr als 50%
- Wakeup über CAN nicht bei highspeed CAN
- Nachträgliche Ergänzungen im Bordnetz sind knifflig (ID -> Latenzzeit der Nachrichten)
- Dimensionierung Abschlusswiderstand (Reuse der Steuergeräte in verschiedenen Fahrzeug Baureihen)
- Buspegel / Signalform bei hoher Anzahl von Steuergeräten (Sleep und Wakeup)

FlexRay

- Anwendungsbereich: Motorsteuerung, Getriebesteuerung, Safety
- Deterministischer Bus mit statischem und dynamischem Segment für zeit- und eventgesteuerte Nachrichten (skalierbar)
- Differentielles Signal (Encoding MFM)
- Unterschiedlichste Topologien möglich (Passiver Bus, Stern)
- Geschwindigkeit bis 2 x 10 MBit (Redundanz <-> Performance)
- Fehlererkennung über CRC, Busuardian für statisches Segment
- Wecken über Bus
- Systemweite Datenkonsistenz
- Geringer Overhead 5 – 10 %
- Flashen der SW über FlexRay

FlexRay

- Im statischen Segment garantierte Latenzzeit und Jitter
- Globale Zeitbasis mit Synchronisierung
- Startup ist Bestandteil des Protokolls
- Aufgrund der garantierten Latenzzeit und Redundanz (2 Kanäle) für sicherheitsrelevante Systeme geeignet:
Abstandswarnsystem, Fahrbahnwechselassistent, EPS, Lenkwinkel, Aktives Fahrwerk, X-by-Wire,
- Wakeup, Startup und Sleep sind Bestandteil des Protokolls (definierte Zustandsübergänge)



Probleme FlexRay

- Neue Technologie / wenig qualifizierte HW verfügbar (Q1/2007)
- Anzahl der FlexRay Konfigurationsparameter ist sehr groß
- Nicht beliebig skalierbar
- Viele Toolchains mit unterschiedlichem Reifegrad (SW wird generiert)
- Standards wie Fibex oder Autosar sind keine Garantie für universelle Austauschbarkeit (Bordnetzbeschreibungsdateien, Toolchains, LLD Library, ...)

Höhere Performance und neue Funktionen erfordern neue Bussysteme

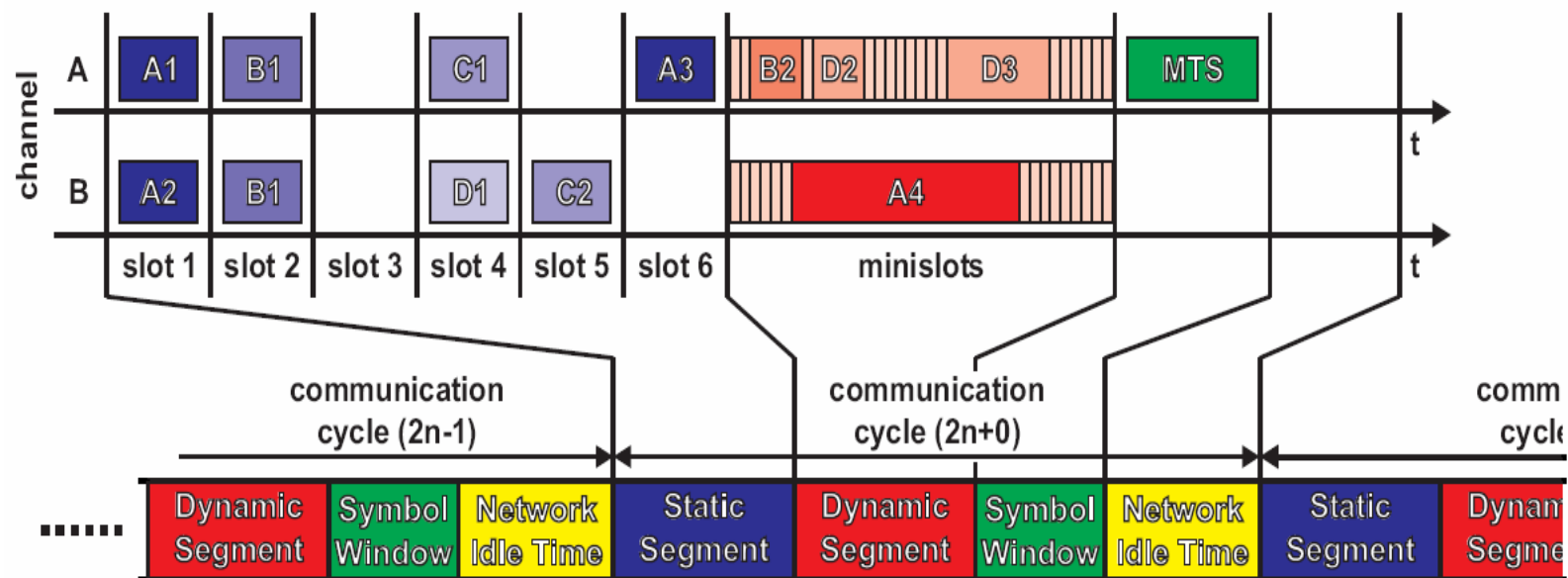
- Automotive (Normen, Group Standards)
- Anforderung Busse (Geschwindigkeit, harte Echtzeit, Sicherheit)
- Powertrain (Motorsteuerung, Getriebe, aktive Bremsen, aktives Fahrwerk, Steuerung der Allradkomponenten)
- Sicherheit (Crash Sensor, mehrstufiges Auslösen vieler Airbags)
- X-by-Wire Sicherheitssysteme (Zusammenspiel mehrerer Systeme -> Gyro Sensoren, Lenkwinkel, Radar für Abstand, Fahrbahnwechsel und Einparken)

-> Kommunikation zwischen den einzelnen Systemen im Fahrzeug nimmt zu

-> Konfigurierbarkeit und Anpassung jedes einzelnen Systems eines Fahrzeugs nimmt zu

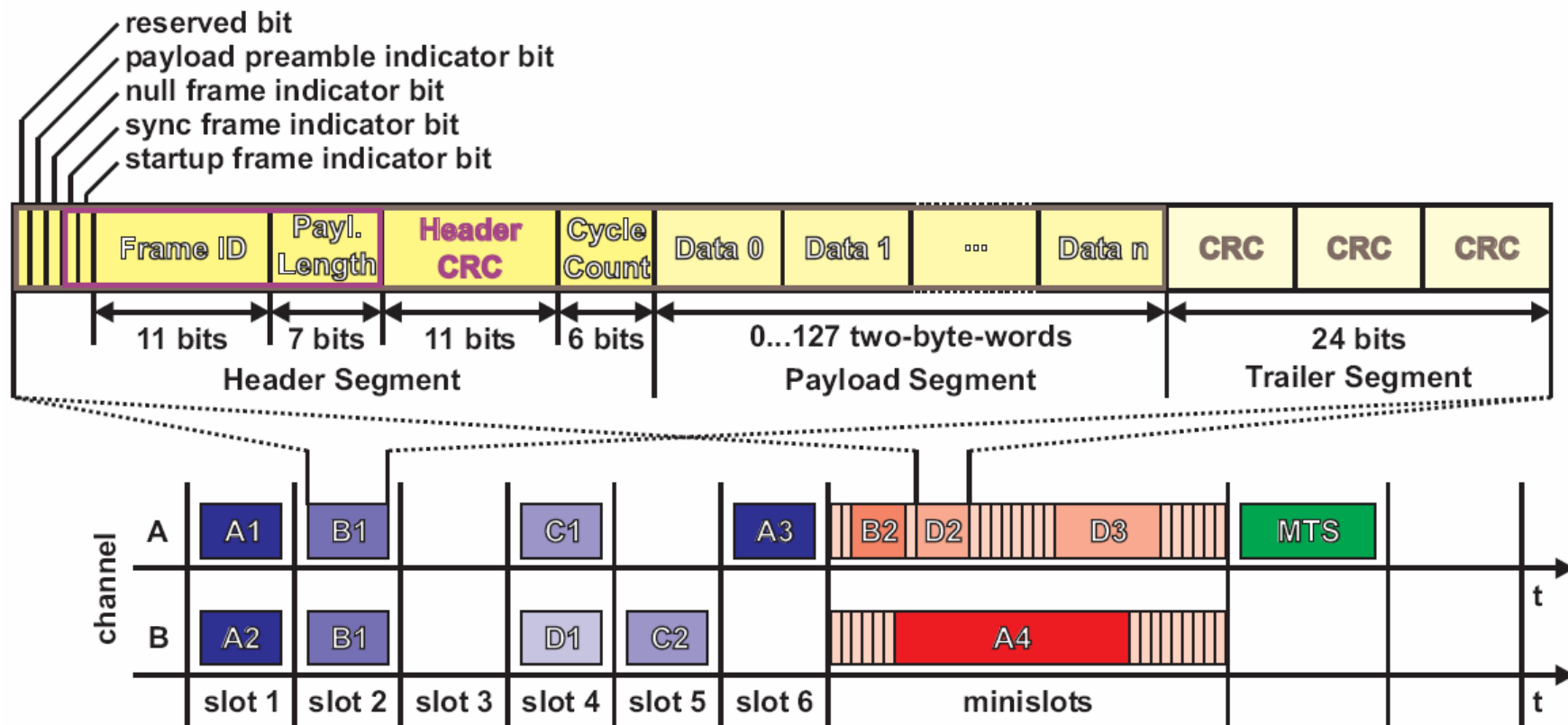
FlexRay Protokoll Communication Cycle

Statisches Segment / Dynamisches Segment / Protokoll Steuerung / Idle

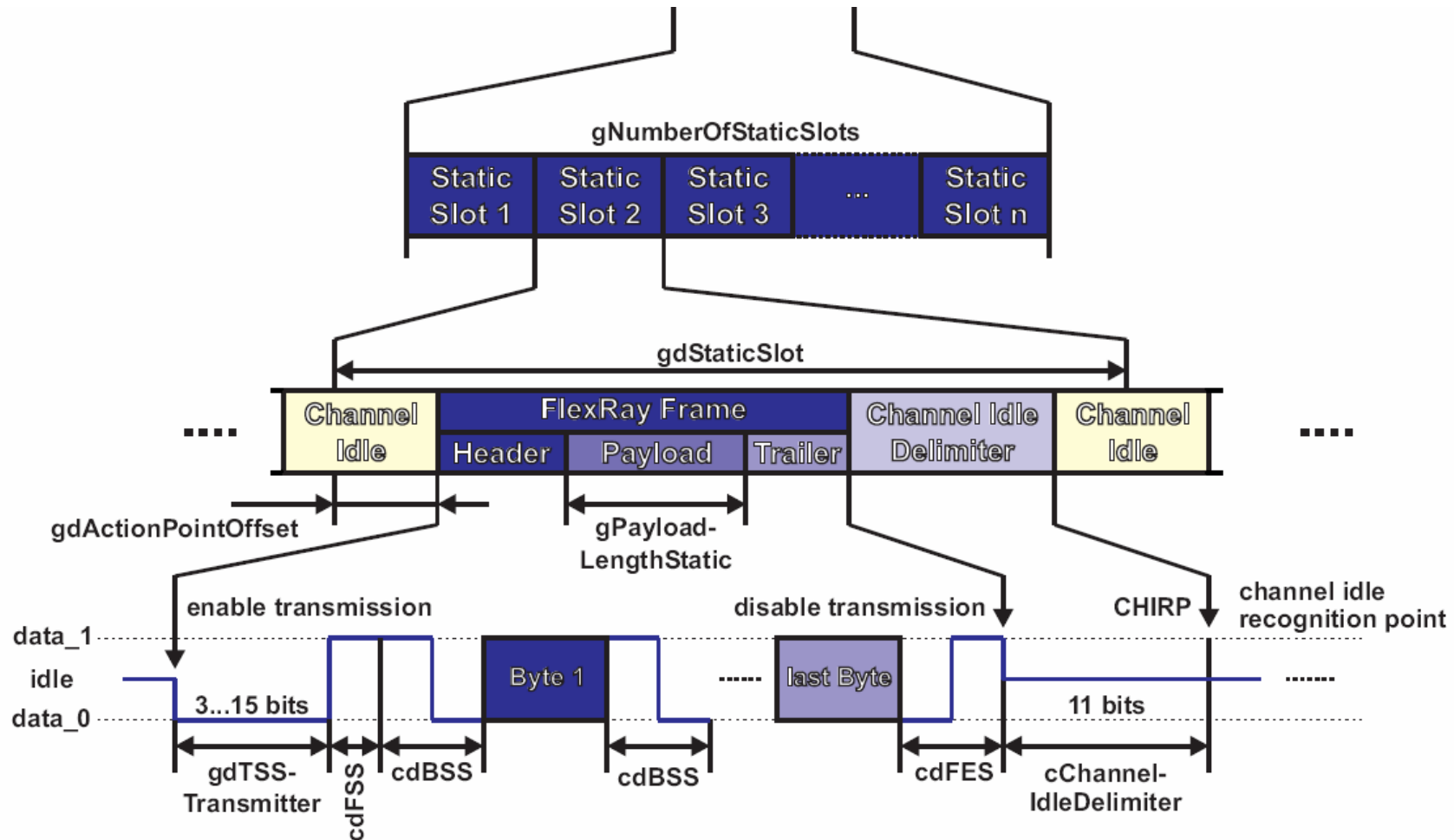


FlexRay Protokoll Frame

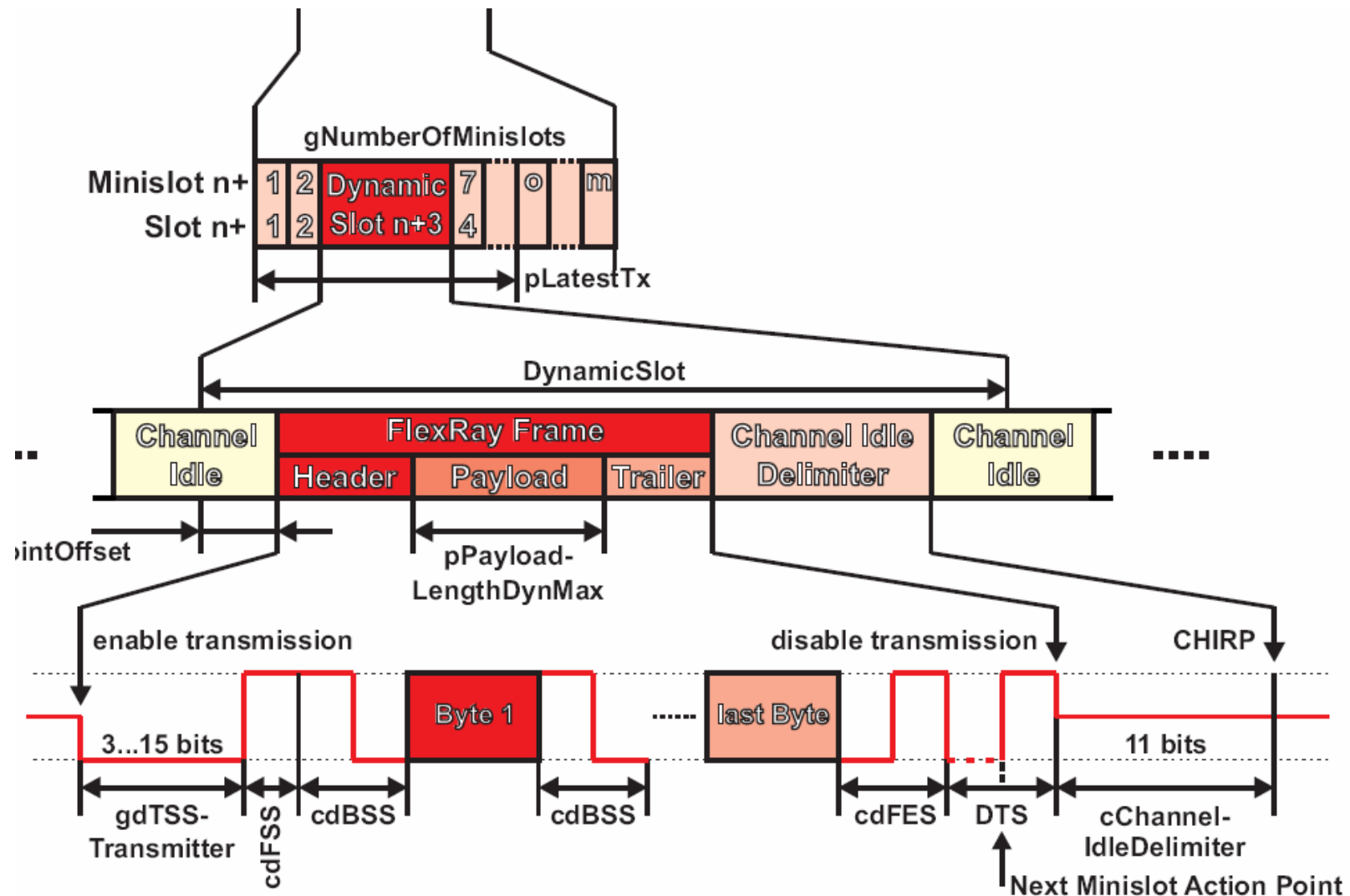
FlexRay Frame im statischen und dynamischen Segment



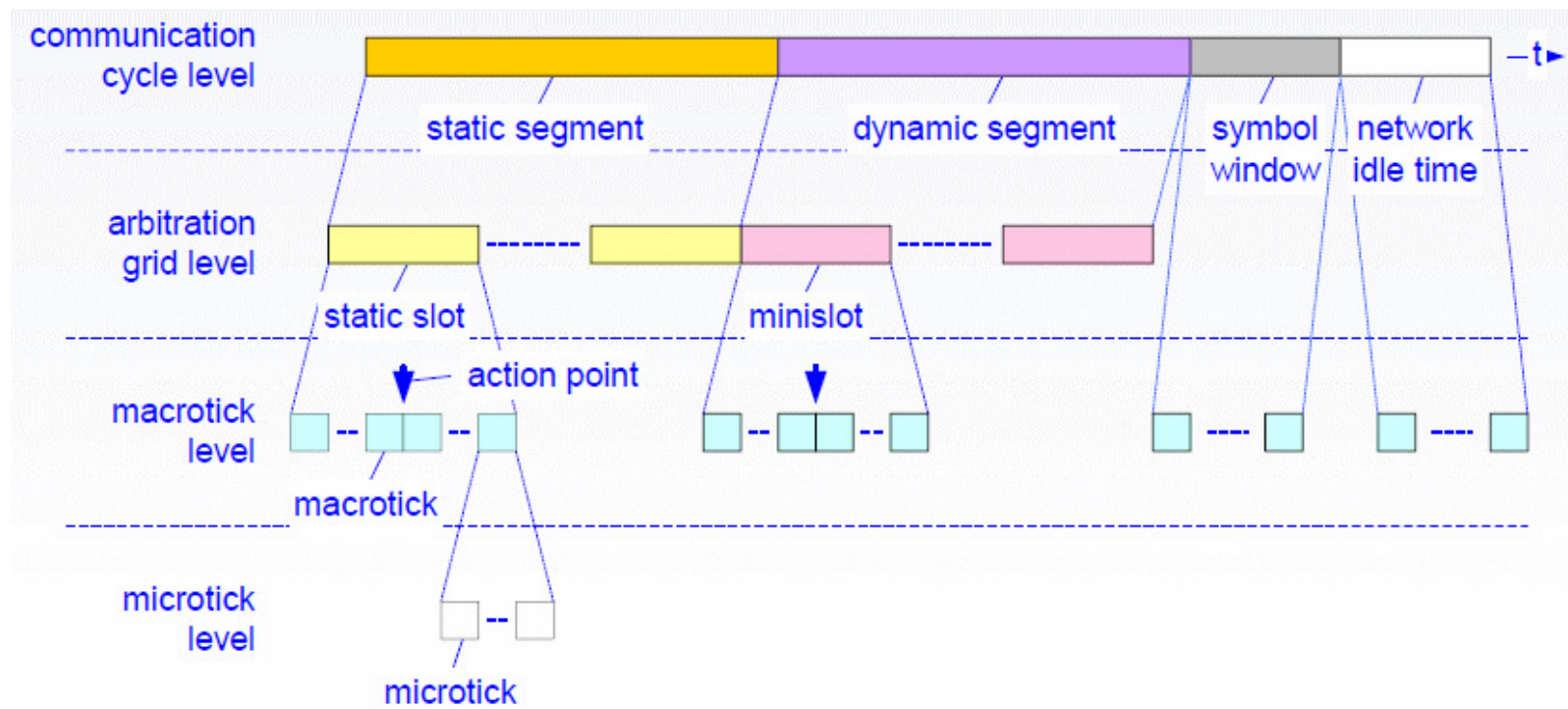
Statisches Segment und Konfigurationsparameter



Dynamisches Segment und Konfigurationsparameter



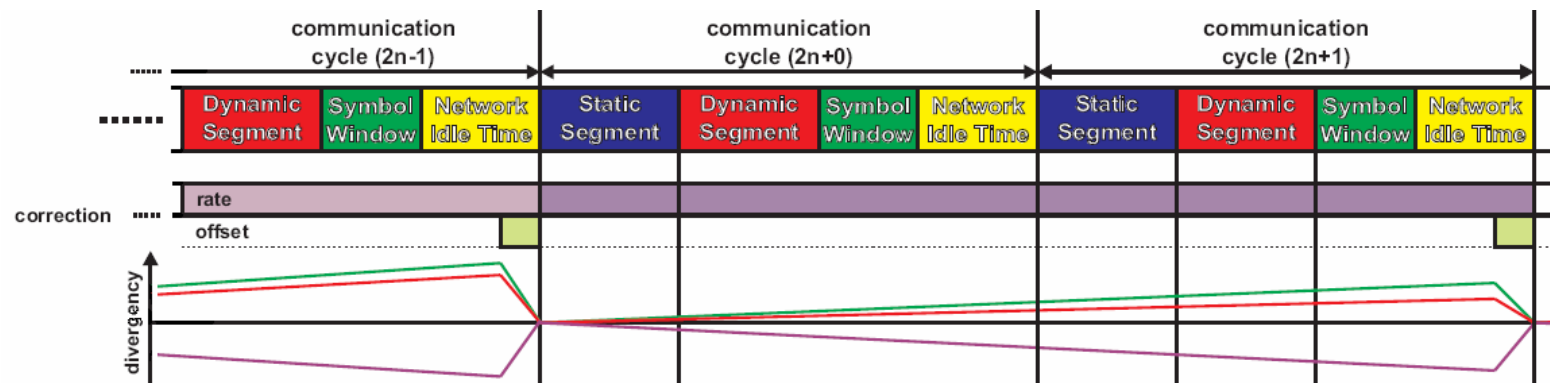
Macroticks und Microticks



Uhrensynchronisation (global Time) mit Sync Frames im statischen Segment

Synchronisation Physical Layer:

- Die Synchronisation erfolgt durch Sync Frames am Ende eines Kommunikationszyklus (NIT)
- Korrigiert werden Clock Rate und Offset



Synchronisation Physical Layer

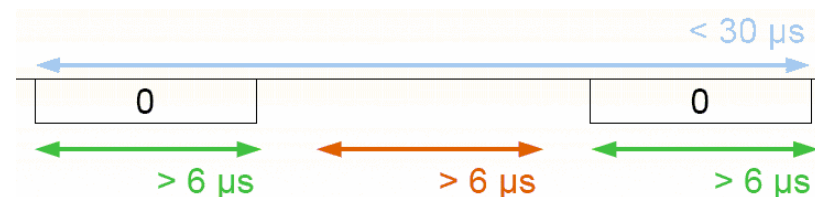
Wakeup wird durch Coldstart Nodes gesteuert und ist Bestandteil des FlexRay Protokolls

Local Wakeup

- Wakeup Pin (FR-Controller oder Transceiver)

Wakeup via FlexRay Bus

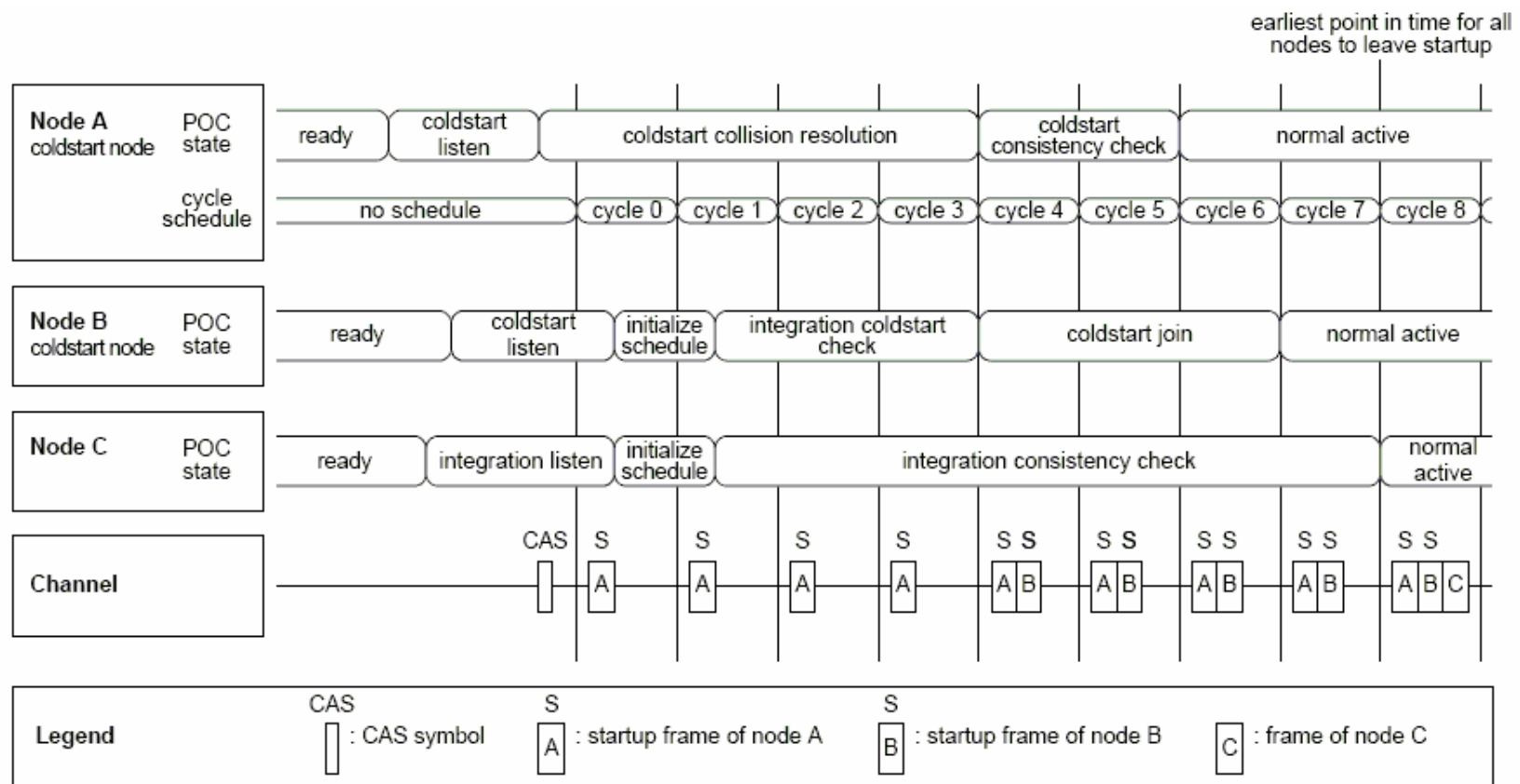
- FlexRay-Frame mit gültigen Daten
- Wakeup Pattern im Symbol Window



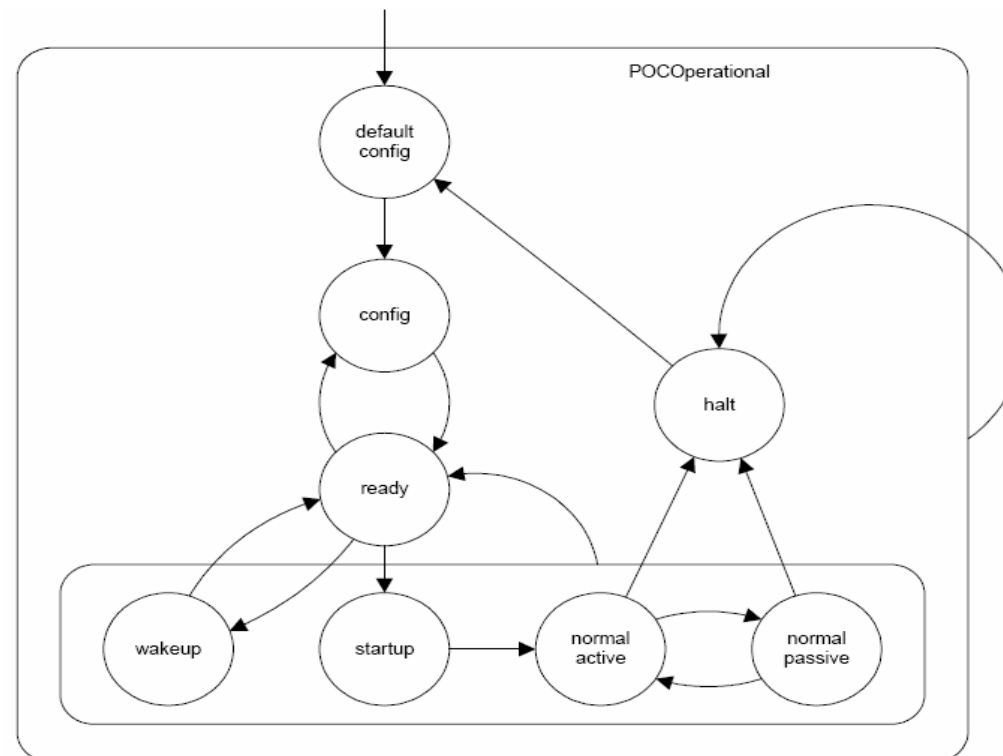
Wakeup Pattern

FlexRay Protokoll Startup

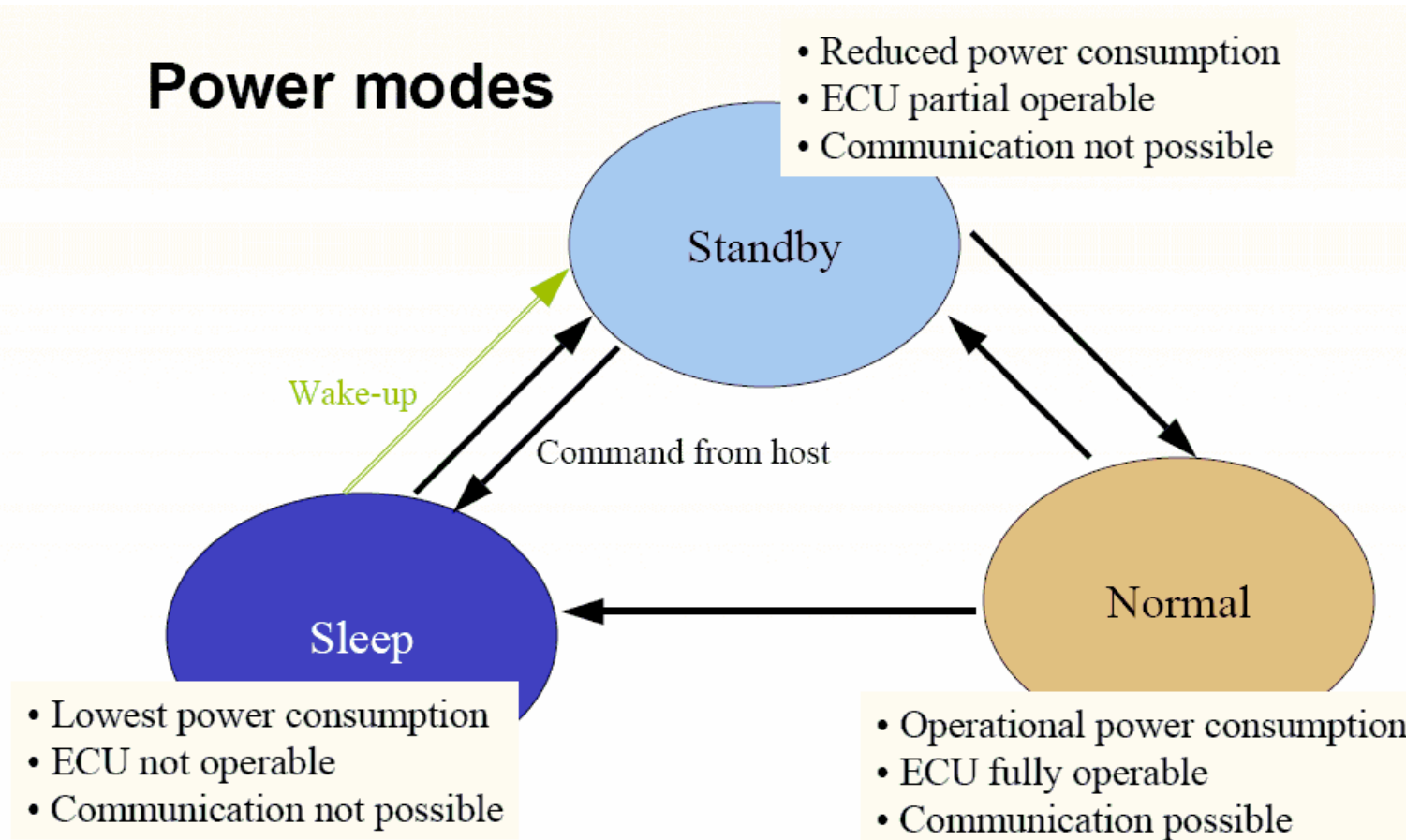
- Mindestens 2 dedizierte Knoten (Leading Coldstart Node und Following Coldstart Node) steuern den Startup
- Nur die Coldstart Nodes können einen Cluster Startup initiieren

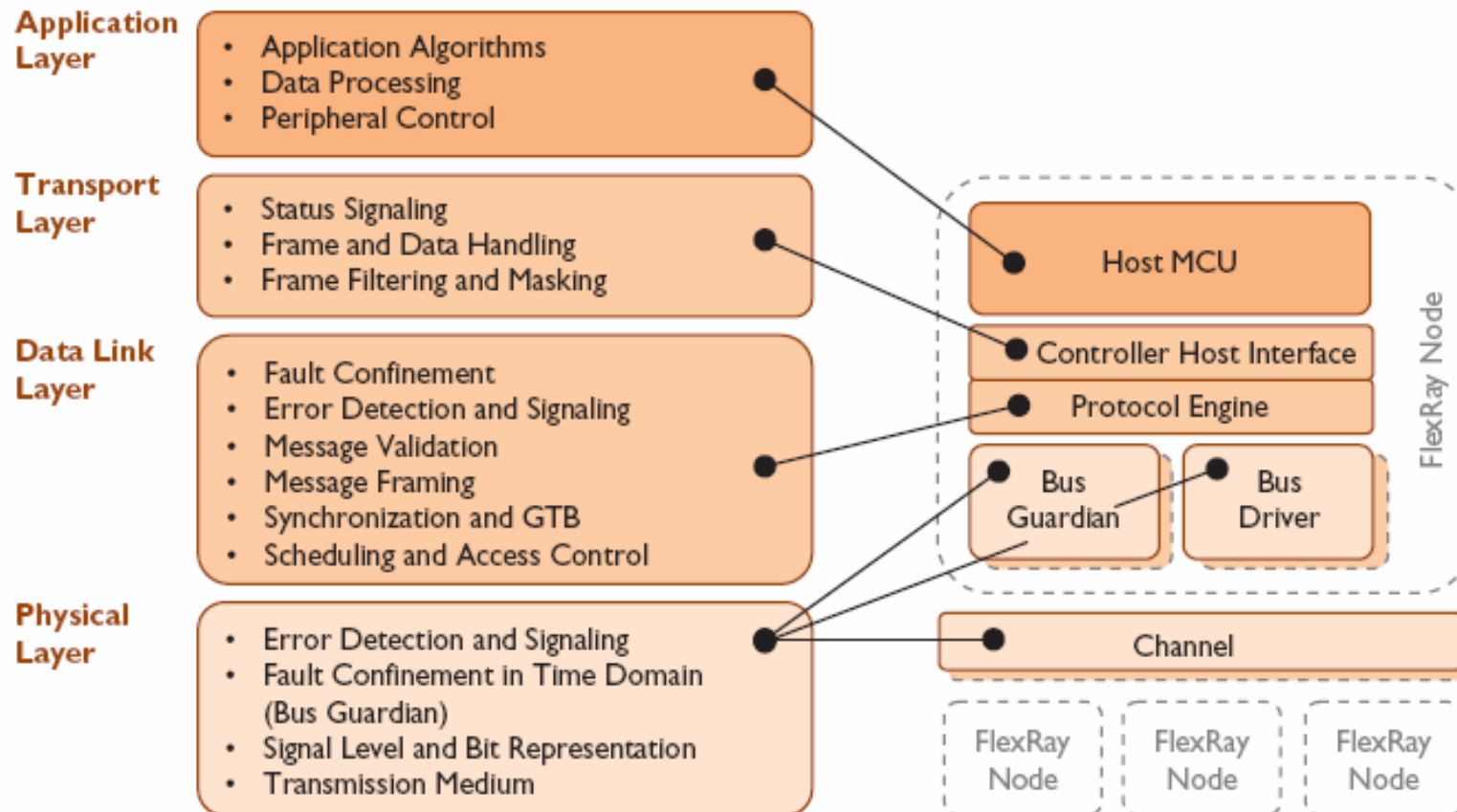


- Fehlererkennung durch Protokoll (CRC, Clock Rate, Offset, ...) und Bus-Guardian
- Degradation Model:
Applikation steuert die Zustandsübergänge, die im Protokoll möglich sind („normal aktiv“ -> „normal passiv“ -> „halt“)

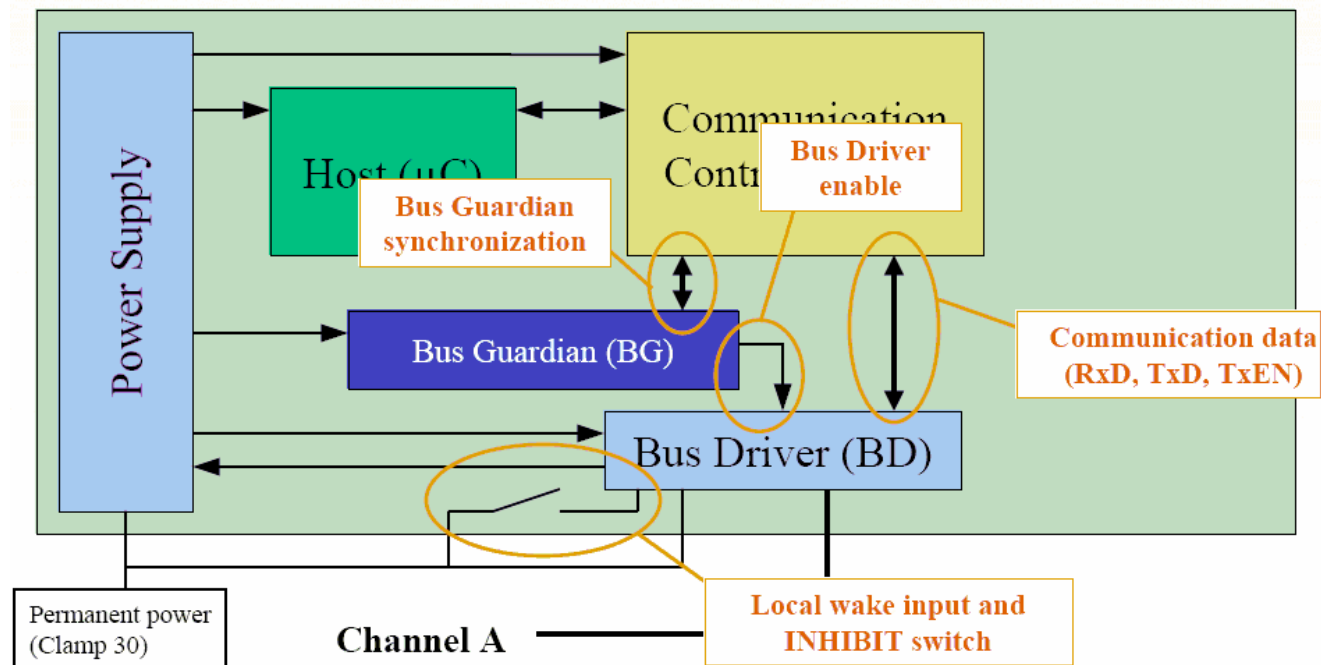


Power modes

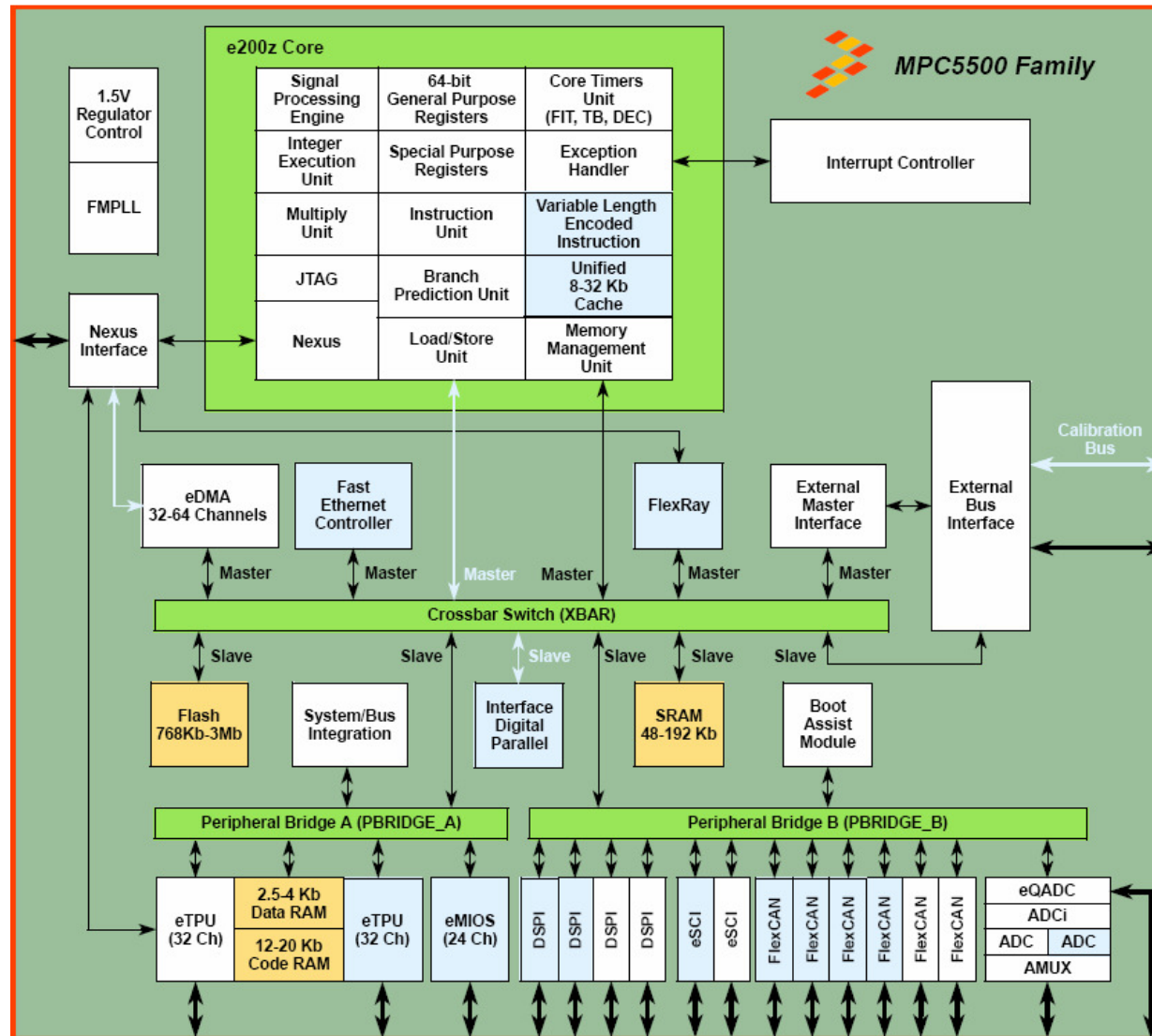




- μ Controller mit integriertem FlexRay
- FlexRay Communication Controller
- FlexRay Transceiver



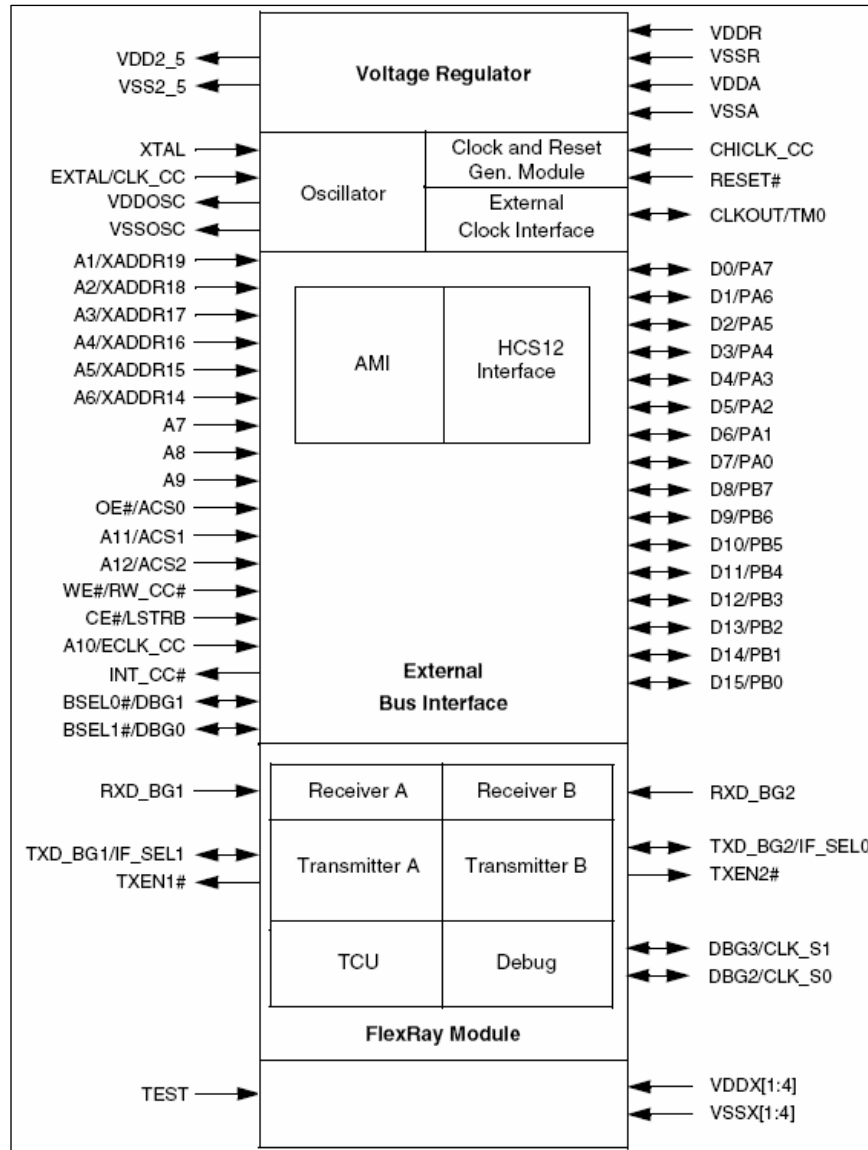
FlexRay Interface μ Controller mit int. FlexRay



Freescale MPC5567

- PBGA416
- -40 bis +125 °C
- 132 MHz
- 2 MB Flash
- 64 KB SRAM
- 5x CAN
- 2 CH FlexRay
- 1x Fast Ethernet

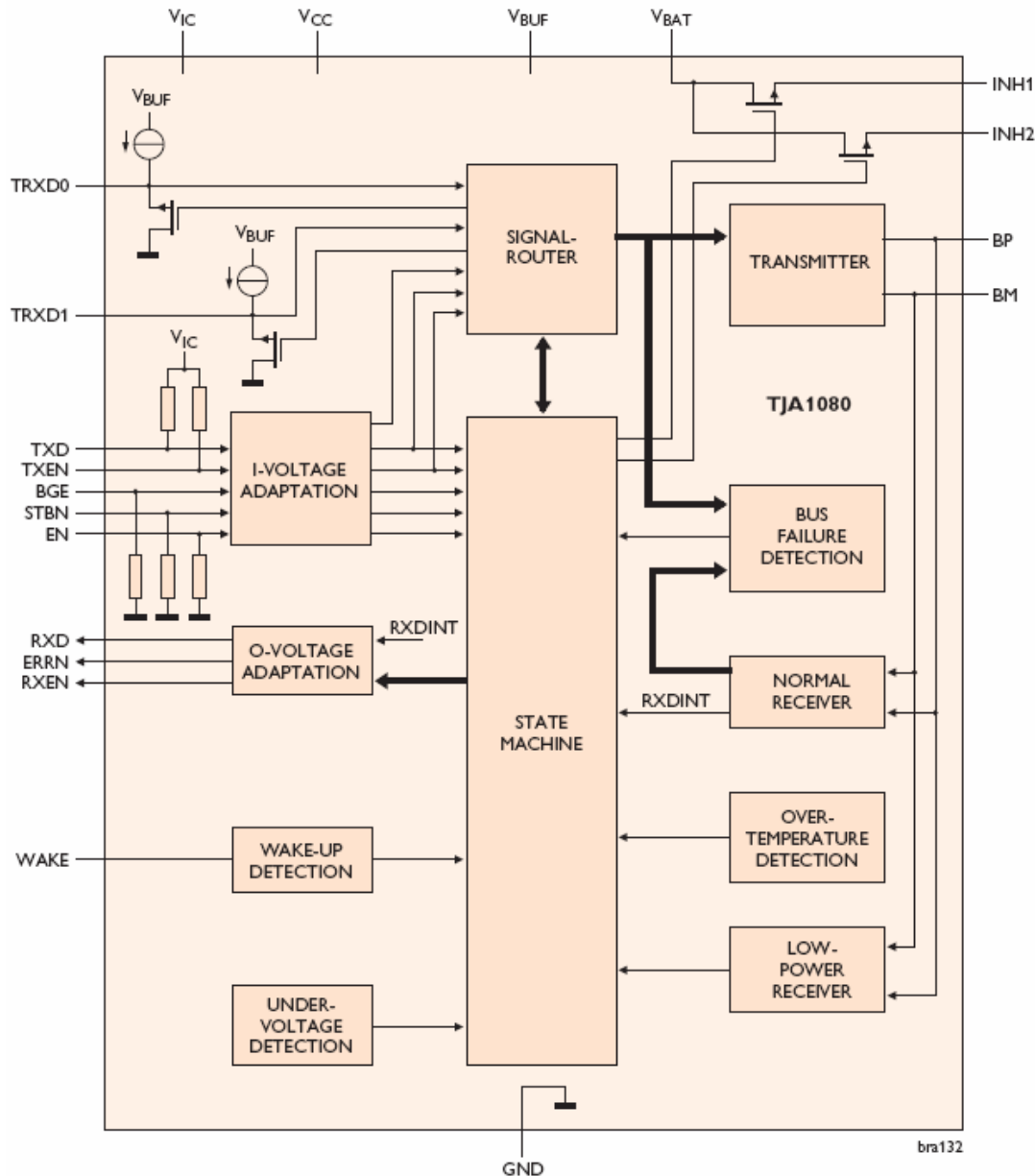
FlexRay Interface Communication Controller



Freescale MFR4310

- LQFP64
- -40 bis +125 °C
- 40 MHz

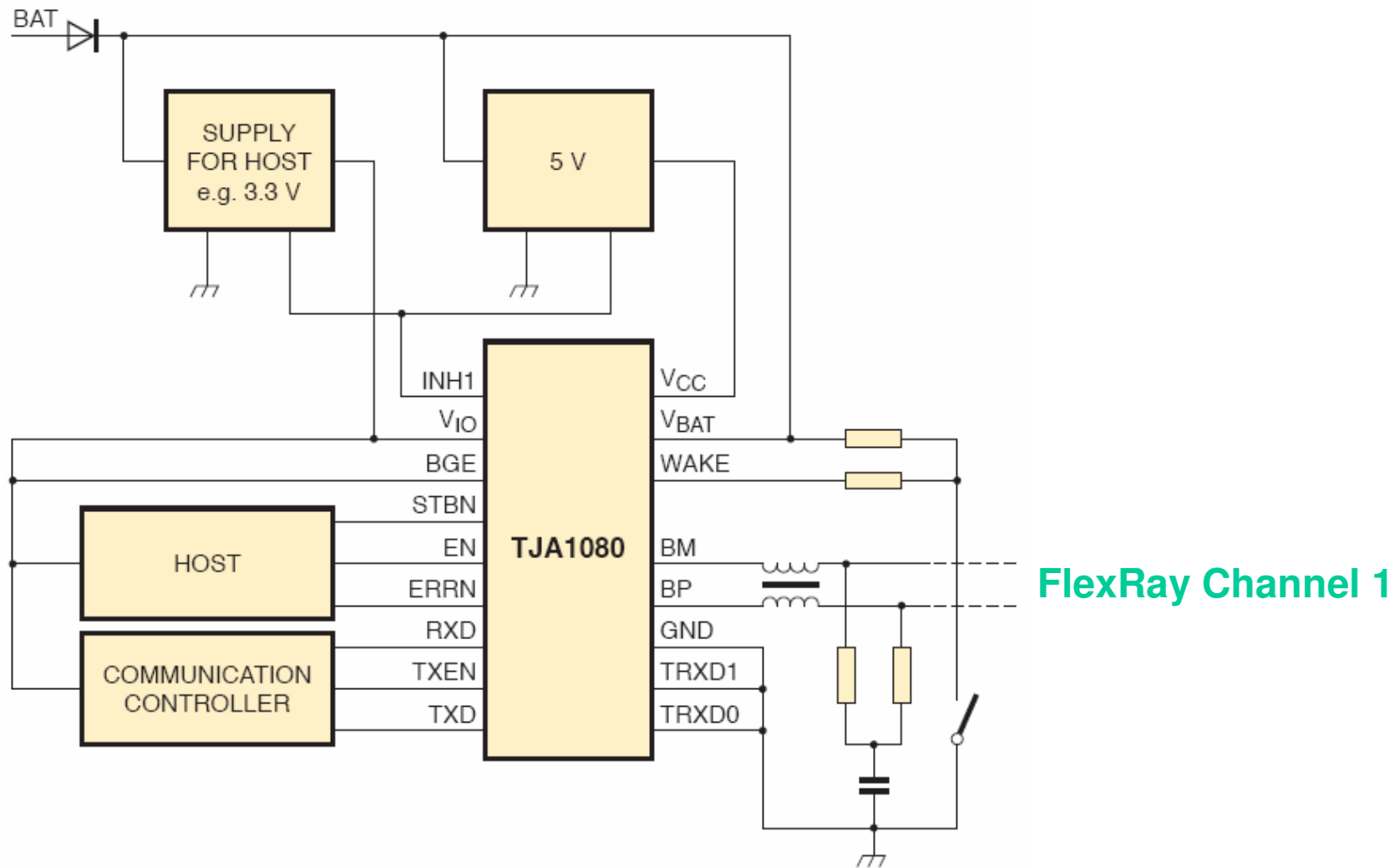
FlexRay Interface Bus Transceiver



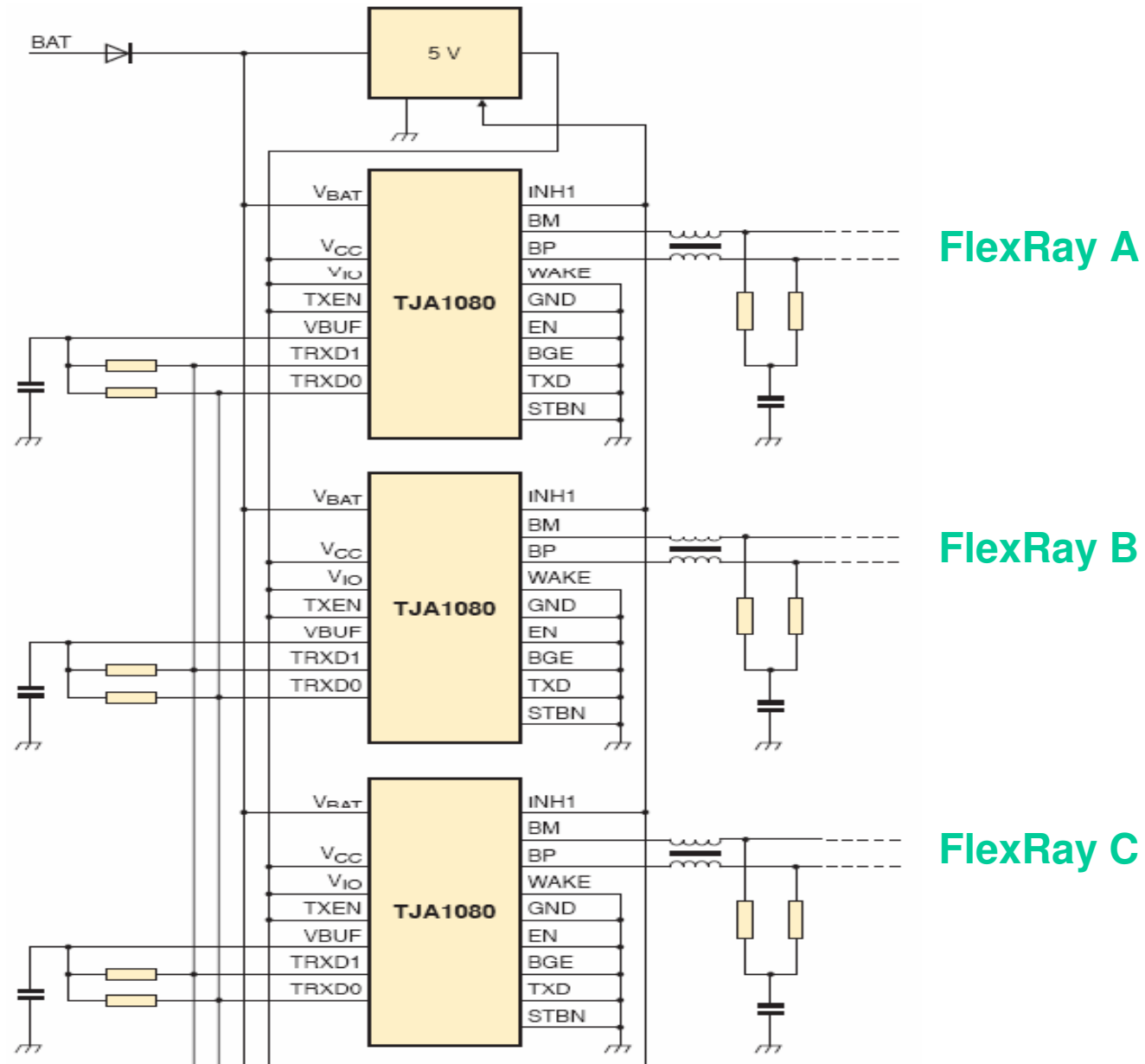
NXP TJA1080

- SSOP20
- -40 bis +125 °C
- 50 µA Ruhestrom
- +/- 8 KV ESD

FlexRay Interface Bus Transceiver (Knoten)



FlexRay Interface Bus Transceiver (Active Star)

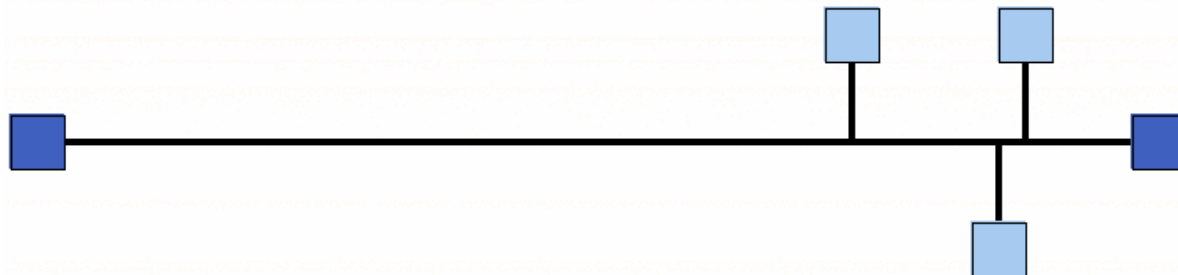


Passive Bus

- Max 8 Stubs 0,2m 10Mbit
- Max. 12m Kabellänge

Nachteil:

- Akzeptables EMV Verhalten
- Praktikabel, aber keine Fehlererkennung (z.B. Kurzschluss)



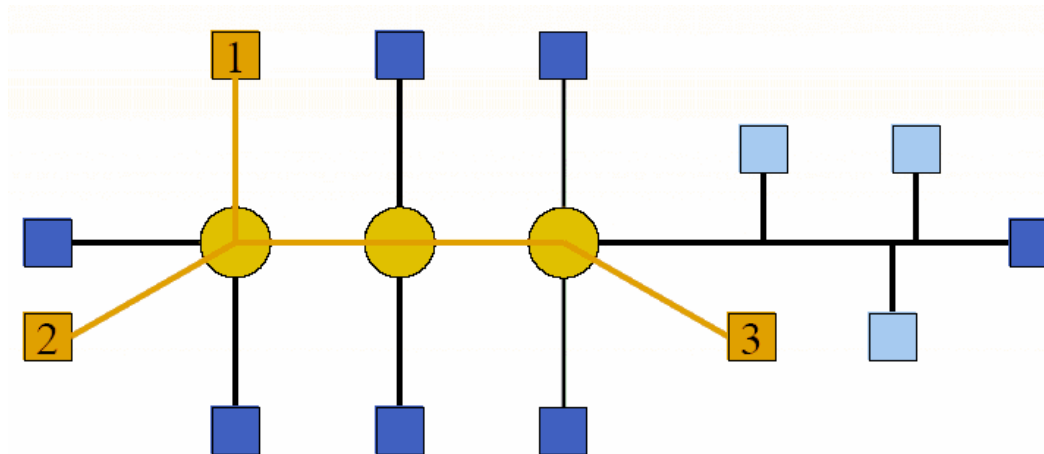
Aktive Star

- 16 Verzweigungen (max. 12m Länge oder 24m Länge ohne Stichleitung)
- Verbessertes EMV Verhalten
- Fehler Management on Branches
- Autonomes Nachrichten Routing
- Autonomes Powermanagement (Wakeup, Sleep, ...)



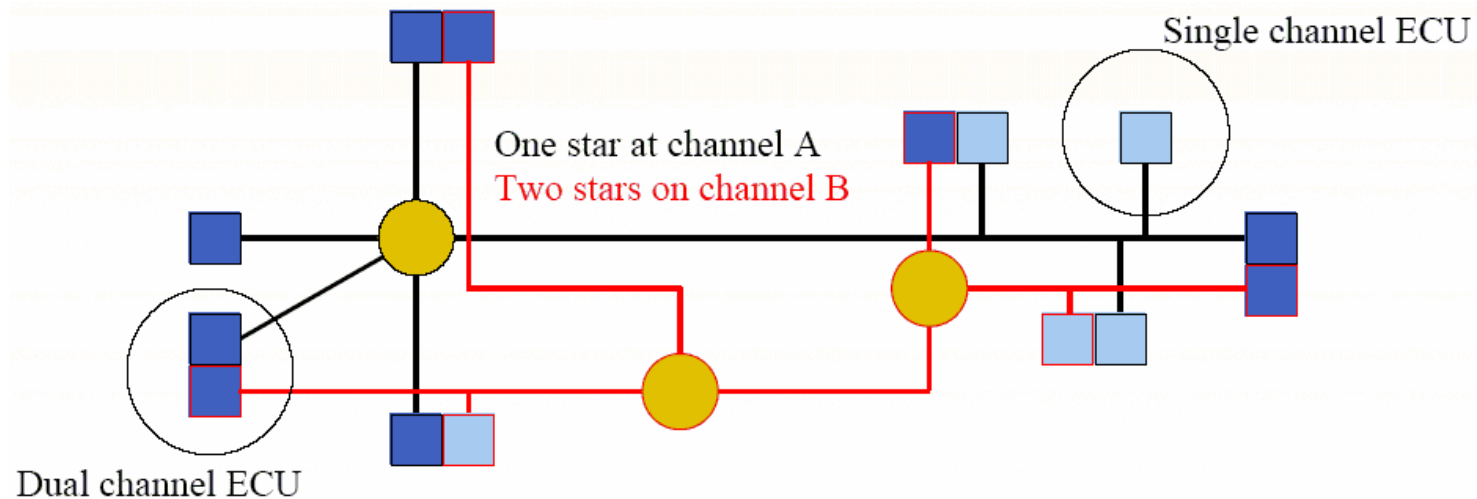
Kaskadierte Aktive Stars

- max. 3 aktive Stars zwischen zwei Knoten
- Uhren Synchronisierungs Algorithmus korrigiert unterschiedliche Propagation Delays



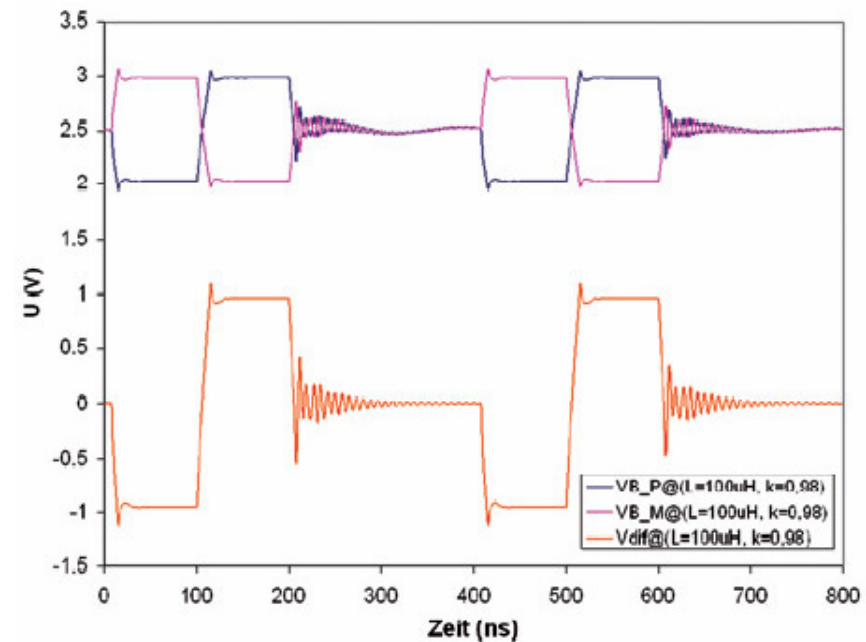
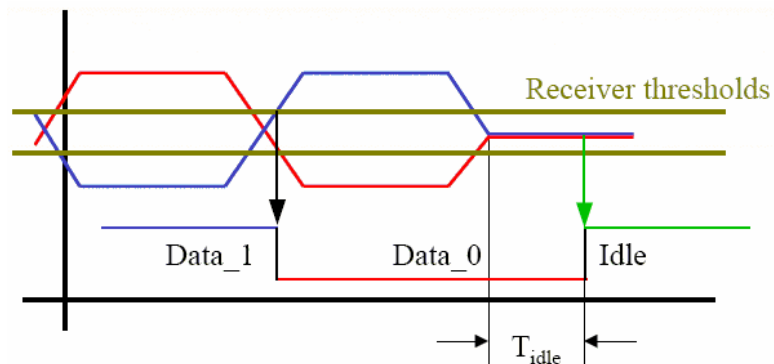
Features eines Zwei-Kanal Netzwerks:

- Unterschiedliche Channel Layouts
- Mischung von Single und Dual Channel Knoten
- Redundanz \leftrightarrow Bandbreite



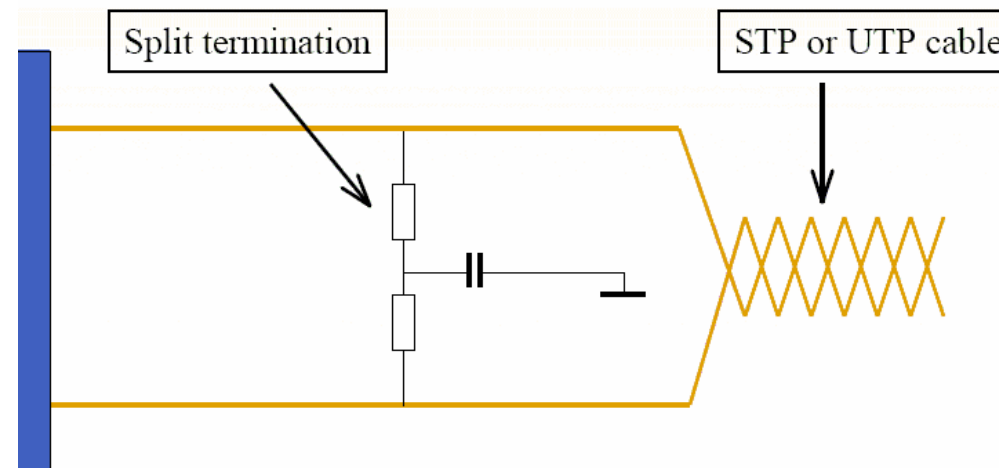
Bus Pegel

- Beide Signal Zustände sind dominant
- Nur der Idle Zustand ist recessive
- Weite Hysterese
- Threshold symmetrisch zum Idle Pegel



Terminierung

- Nötig für genau definierte Bus Pegel
- Verhindert Reflexionen
- Nötig bei End Knoten und Stars
- Nicht notwendig bei Stichleitungen



Ziele:

- Automatisierung (Konfiguration, Code Generierung, Integration in AUTOSAR SW, Test, ...)
- Kurze Entwicklungszeit
- Steigerung von Reuse
- Steigerung der Qualität
- Integration in bestehende Prozesse

Automatisierte Tests des Flexray Knotens gegen die Protokoll Spezifikation

- Integration der Host Applikation, Stimuli Vektoren, Antwort Vektoren
- Automatisierter Testflow
- Automatisierte Auswertung der Ergebnisse

Test Grundlage

- Physical Layer Conformance Test Specification V1.0
- Data Link Layer Conformance Test Specification V2.1

Test Toolchains z.B. erhältlich von Decomsys, IXXAT und SMART

Physical Layer Conformance Test

- ESD
- Termination / Undervoltage
- Short Circuit / Communication Failure
- Signal Shape / Signal Offset / Threshold / Eye Diagrams
- Stress Test
- Babbling Idiot

Data Link Layer Conformance Test

- Startup / Wakeup / Sleep
- Indicators / ID / PayloadLength / CRC
- Konfigurationsdaten / Synchronisation / Timer
- Transmit Buffer Assignment

Tracer / Analyzer / Multibus Analyzer

- PC Tools mit FlexCard / Analyzer HW
- Autarker Einsatz im Fahrzeug mit internem Speicher
- Offline Analyse Tools

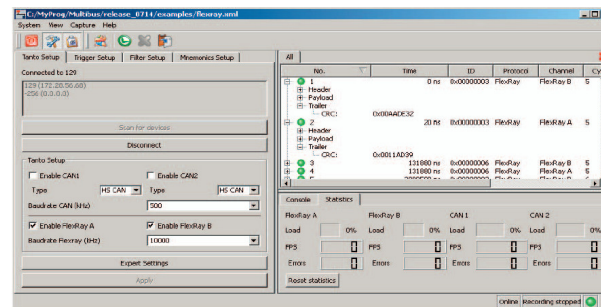
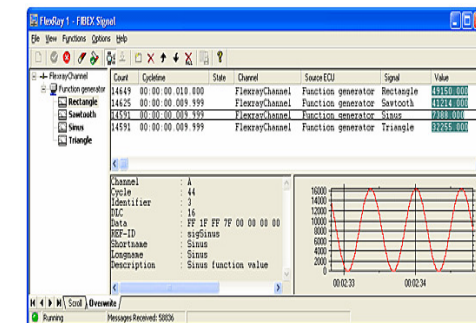
Mixed-Mode / Hitex Tanto2 multiBus Analyzer

Vector-Informatik Canoe

IXXAT Multibus Analyzer

Decomsys Busdoctor

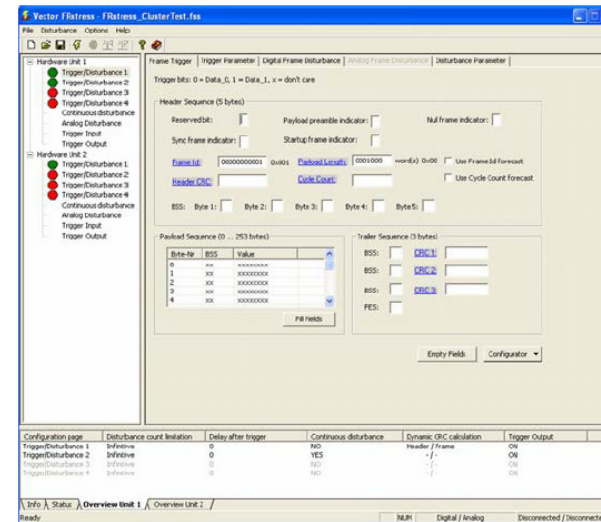
Yokogawa Signal Analyzer



Test Tools

- Restbussimulation
- Stresstest
- Fehlerinjektion

Vector-Informatik
IXXAT
Smart



FlexRay Konfiguratoren / Generatoren

- Konfiguratoren für Bordnetz / FlexRay / Fibex
- Code Generatoren für FlexRay Controller
- Code Generatoren für Autosar SW
- Konfigurator für OSEKtime Betriebssystem

SW-Systemhäuser

- Electrobit (Autosar Basis SW)
- Vektor-Informatik (Autosar Basis SW)
- Mentor Graphics (SW-HW-Integration)
- ETAS (SW-HW-Integration)
- CRST (SW-HW-Integration)
- HL-Hersteller (Autosar LLD)

Automotive Technologien im Fahrzeug

- FlexRay „Light“
- Fast Ethernet (Flashen von SW gleichzeitig in versch. Bussegmente)
- Carring II (TU Clausthal):
 - Gigabit Ring, max. 255 Ringe, beliebige Topologien, beliebig skalierbar, große Redundanz

Unterhaltungstechnik im Fahrzeug

- USB (HDD Navigation, mobile DVD, mobile MP3, iPod)
- Flash Speicherkarten (Multimedia, Navigation, Ersatz von DVD / CD-Wechsler)

Spezifikationen

- www.FlexRay.com FlexRay Spezifikation, Information
- www.can-cia.org CAN Spezifikation, Information
- www.lin-subbus.org LIN Spezifikation, Information

Tools

- www.mixed-mode.de Testtools, Dienstleistungen
- www.ixxat.de Testtools, Dienstleistungen
- www.tzm.de Testtools, Dienstleistungen
- www.decomsys.de Testtools, Dienstleistungen
- www.vector-informatik.de Testtools, Dienstleistungen
- www.smart-gmbh.de Testtools, Dienstleistungen

Flexray μ Controller, Flexray Communication Controller / Transceiver

- www.amis.com FlexRay Transceiver
- www.freescale.com FlexRay μ Controller, Transceiver
- www.fujitsu.de FlexRay μ Controller
- www.infineon.com FlexRay μ Controller, Transceiver
- www.nec.de FlexRay μ Controller
- www.nxp.com FlexRay Transceiver



Danke für ihre Aufmerksamkeit!

Haben sie noch Fragen?

Kurt Veit

kurt.veit@mixed-mode.de



FlexRay

Diskussion



- Statisches und dynamisches Segment

