

OSEK/VDX Network Management

Ausarbeitung zum Seminarvortrag
von Alexander Berger

Letzte Änderung: 31.10.2007

OSEK/VDX Network Management

Inhaltsverzeichnis

1 Motivation.....	5
2 Aufgaben des NM.....	5
3 Architektur.....	6
4 Konzept und Verhalten.....	7
4.1 Indirektes Netzwerk-Management.....	7
4.1.1 Knoten-Monitoring.....	7
4.2 Direktes Netzwerk-Management.....	8
4.2.1 Knoten-Monitoring.....	8
4.2.2 Logischer Ring.....	8
4.2.3 Verhalten des logischen Rings.....	9
4.2.4 Datenaustausch.....	11
4.2.5 Network Management Protocol Data Unit (NMPDU).....	11
4.2.6 Beispiel zum Wechsel in den globalen Ruhemodus.....	12
5 Skalierbarkeit.....	13
6 Ausblick.....	13
7 Erfahrungen in der Massenproduktion.....	14
7.1 Indirektes NM bei Renault.....	14
7.2 Direktes NM bei DaimlerChrysler.....	14
8 Quellen.....	15

1 Motivation

In den heutigen Fahrzeugen sind vielfältige Steuereinheiten / ECUs der verschiedensten Hersteller miteinander vernetzt. Um die Funktionsfähigkeit dieser Vernetzung garantieren zu können, benötigt man standardisierte Schnittstellen und Protokolle.

Eine Tatsache, die in einem komplexen Netzwerk zudem beachtet werden muss, ist die Beeinflussung des Verhaltens des gesamten Systems durch das Verhalten eines einzelnen Knotens dieses Netzwerks (und umgekehrt). Gerade in einem Fahrzeug, wo viele Informationen in kurzen Abständen bearbeitet und ausgewertet werden müssen, sind häufig mehrere ECUs bei der Erfüllung einer komplexeren Aufgabe beteiligt. Beispielsweise ist das Motormanagement eine Aufgabe, in der ständig neue Messwerte von Sensoren geliefert werden. Die erhaltenen Messwerte bilden dann bspw. die Grundlage zur Berechnung des benötigten Kraftstoff-Sauerstoff-Gemischs.

Fehlfunktionen, die auf gegenseitigen Beeinflussungen beruhen, müssen unbedingt vermieden werden.

Um dies zusichern zu können, muss der Status des Netzwerks überwacht werden. Zudem ist es notwendig, dass die einzelnen Knoten des Netzwerks miteinander in „Verhandlung“ stehen. So lässt sich bspw. verhindern, dass sich Knoten unkoordiniert vom Netzwerk abmelden, obwohl sie möglicherweise noch für die Bearbeitung einer Aufgabe benötigt werden.

Eine dedizierte Netzwerk-Management-Komponente soll diese Aufgaben erfüllen.

Ein weiteres Ziel des Netzwerk-Managements (im weiteren auch NM genannt), ist die Einsparung von Kosten und Entwicklungszeit, sowie die Portabilität durch den Einsatz vorgefertigter Module, die leicht an verschiedene Fahrzeugplattformen angepasst werden können.

2 Aufgaben des NM

Das NM soll den Anwendungsentwickler in erster Linie entlasten, indem der NM-Code vom eigentlichen Anwendungscode entkoppelt wird.

Die Standard-Management-Tasks, die das Netzwerk betreffen, übernimmt das NM.

Beispiele für solche Tasks, sind:

- Initialisierung der Kommunikationshardware
- die Möglichkeit verschiedene Busse an einem μ Controller anzuschließen
 - das NM bietet für jeden einzelnen Bus entsprechende Unterstützung / Überwachung an
- „start-up“ des Netzwerks & Steuerung der Betriebsmodi
 - problematisch kann hier der unkontrollierte Wechsel eines Knotens zwischen den Betriebsmodi sein
 - das NM koordiniert diese Betriebsmodi-Wechsel
- Ermittlung der aktuellen Netzwerkkonfiguration
 - das NM erkennt das Hinzufügen oder den Ausfall von Knoten im Netzwerk
- Unterstützung von Diagnose
 - Werkstätten können Fehlerquellen schneller finden und beheben
 - Fahrzeugführer haben idealerweise die Möglichkeit Fehlercodes auszulesen

3 Architektur

In Abbildung 1 erkennt man, wie sich das NM in den OSEK/VDX-Standard eingliedert und welche grundlegenden Dienste es anbietet.

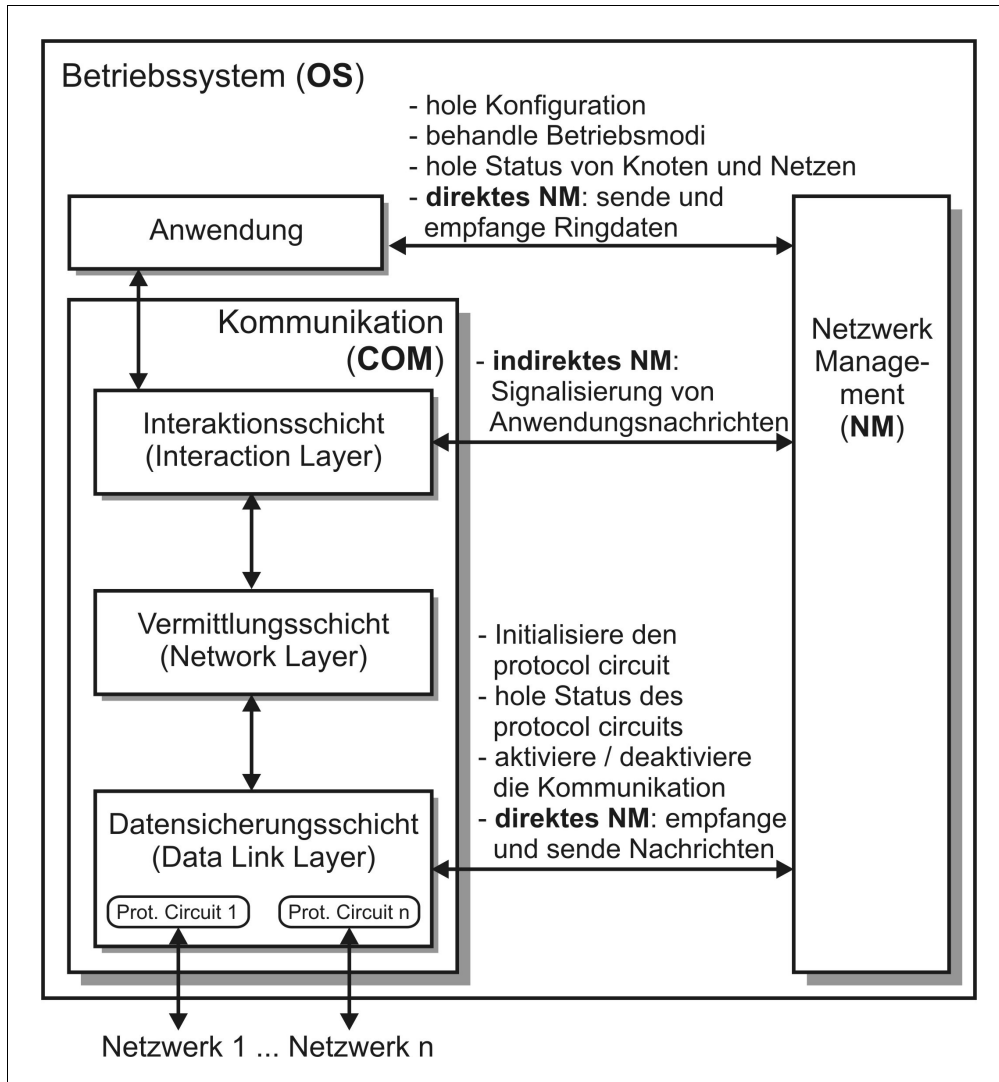


Abbildung 1: Model der OSEK/VDX Architektur

Das NM hat Schnittstellen zur Anwendung, zur Interaktionsschicht und zur Datensicherungsschicht.

- Die Schnittstelle zur Anwendung (API) bietet dem Anwendungsentwickler den Zugriff auf die Funktionen des NM. Die Anwendung kann somit bspw. die aktuelle Konfiguration des Netzwerks in Erfahrung bringen, ohne das Netzwerk selbst überwachen zu müssen.
- Die Schnittstelle zur Interaktionsschicht bietet dem NM die Möglichkeit, über eingehende (und ausgehende) Anwendungsnachrichten informiert zu werden.
- Die Schnittstelle zur Datensicherungsschicht bietet dem NM Zugang zur Kommunikationshardware und ermöglicht es dem NM eigene Nachrichten zu senden und zu empfangen.

4 Konzept und Verhalten

Das Netzwerk-Management basiert auf dem Überwachen der Knoten eines Netzwerks. Dabei kann man zwischen zwei Methoden der Knotenüberwachung unterscheiden:

- *indirektes Knoten-Monitoring* durch Überwachen von Anwendungsnachrichten
- *direktes Knoten-Monitoring* durch Austausch spezieller Nachrichten

Unabhängig von der gewählten Methode, ist die Knotenüberwachung Grundlage für das Konfigurationsmanagement des NM, um die aktuelle Netzwerkkonfiguration ermitteln zu können. Es kann dabei zwischen verschiedenen Konfigurationen unterschieden werden:

- Die *tatsächliche Konfiguration* ist die Menge der Knoten, auf die zugegriffen werden kann
- Die *Ziel-Konfiguration* wird von der Anwendung vorgegeben und mit der *tatsächlichen Konfiguration* verglichen (bspw. zur Ermittlung von aufzuweckenden Knoten)
- Die „*limp-home*“-Konfiguration ist die Menge der Knoten, die aufgrund von Fehlern im „*limp-home*“-Zustand, d.h. ausgefallen, sind

Die „*limp-home*“-Konfiguration kann allerdings nur vom direkten NM ermittelt werden.

4.1 Indirektes Netzwerk-Management

4.1.1 Knoten-Monitoring

Die Anwendungen, die auf den einzelnen Knoten ausgeführt werden, tauschen typischerweise in bestimmten Zeitabständen Nachrichten aus. Diese Nachrichten können z.B. Messwerte, Steuernachrichten usw. sein.

Das indirekte NM basiert auf der Überwachung dieses normalen Nachrichtenaustausches. Dabei wird der Empfang und das Senden von Nachrichten als Lebenssignal des entsprechenden Knotens interpretiert. Das Ausbleiben einer Nachricht über einen bestimmten Zeitraum führt hingegen dazu, dass eine Fehlfunktion des entsprechenden Knotens angenommen wird.

Wie man sieht, ist das indirekte Knoten-Monitoring äußerst simpel und bietet sich bei sehr einfachen oder zeitkritischen Anwendungen an, bei denen die Rechenleistung des Knotens nicht für „stärkere“ Management-Tasks ausreicht.

Der größte Nachteil des indirekten Knoten-Monitorings ist, dass aufgrund des passiven Verhaltens nicht alle im Abschnitt „Motivation“ angesprochenen Aufgaben erfüllt werden können. Da das indirekte NM keine eigenen Nachrichten sendet, bedeutet dies z.B., dass die aktive Steuerung anderer Knoten im Netzwerk (bspw. den Wechsel zwischen Betriebsmodi auslösen) nicht möglich ist.

Die Algorithmen des indirekten NM zeichnen sich durch die folgenden Eigenschaften aus:

- *passiv*
Indirektes NM kann in jedem Knoten optional angewendet werden. Es hat keinen Einfluss auf das Netzwerk und benötigt keine Buskapazität.
- *statisch*
Es handelt sich um einen simplen Algorithmus, der sich zur Entwicklungszeit an die Anwendung anpassen lässt.
- *Geschwindigkeit*
Die Überwachungsperiode kann an die Häufigkeit des Auftretens der Anwendungsnachrichten angepasst werden.

4.2 Direktes Netzwerk-Management

4.2.1 Knoten-Monitoring

Beim direkten NM überwachen sich die Knoten des Netzwerks aktiv gegenseitig, indem sie spezielle NM-Nachrichten senden und empfangen.

Jeder Knoten sendet sein eigenes „Lebenssignal“ und empfängt die Lebenssignale aller anderen Knoten.

Um die aktuelle Konfiguration des Netzwerks zu bestimmen, werden die empfangenen Lebenssignale „aufsummiert“, d.h. die Menge der empfangenen Lebenssignale repräsentiert die Menge der aktiven Knoten auf die zugegriffen werden kann.

Zu Austausch dieser speziellen NM-Nachrichten wird ein logischer Ring benutzt. Dies ermöglicht es die Last auf dem Bus, die durch den Austausch der NM-Nachrichten entsteht, konstant zu halten.

4.2.2 Logischer Ring

Jeder Knoten bekommt eine eindeutige ID zugewiesen. Diese IDs folgen einer logischen Anordnung (z.B. 1, 2, 3, ...), so dass für jeden Knoten ein eindeutiger logischer Nachfolger

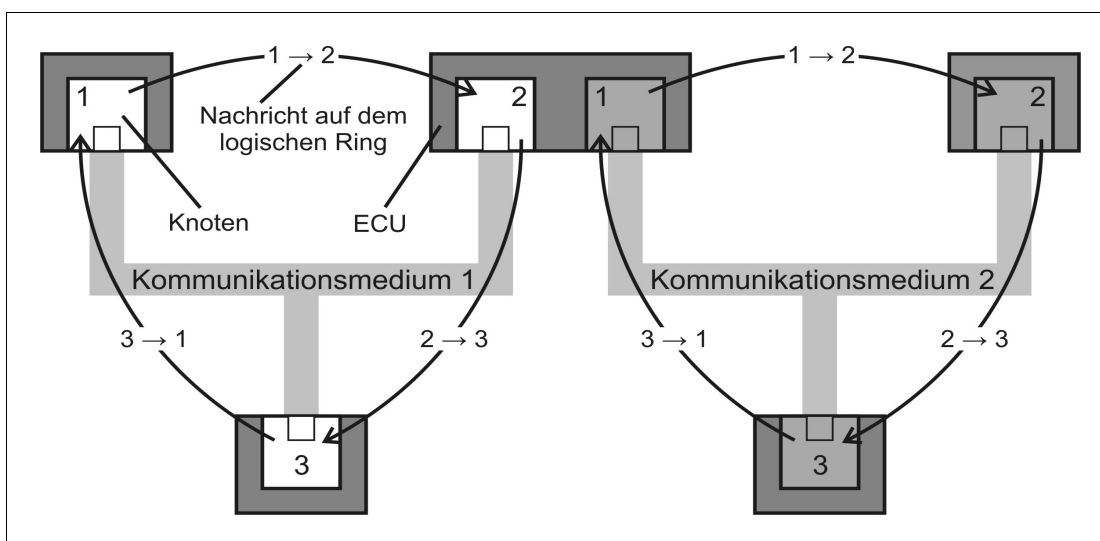


Abbildung 2: Schema des logischen Rings

bestimmt werden kann (z.B. $1 \rightarrow 2$).

Die NM-Nachrichten werden dann über das physikalische Kommunikationsmedium verteilt, aber an den logischen Nachfolger des Absenders adressiert. Das NM jedes Knotens kann alle NM-Nachrichten auf dem Medium empfangen, aber nur der adressierte Knoten darf als nächstes sein „Lebenssignal“ senden.

Die Algorithmen zur Realisierung des logischen Rings zeichnen sich durch folgende Eigenschaften aus:

- *dezentral*
Alle Knoten enthalten identische NM-Software, abgesehen von speziellen Anpassungen für den jeweiligen Knoten, d.h. alle NM-Dienste und Betriebsmodi sind in jedem Knoten verfügbar.
- *dynamisch*
Der logische Ring kann zwischen zwei beliebigen Knoten eingerichtet werden. Das NM kann sich schnell an veränderte Netzwerkzustände anpassen.
- *keine History*
Das NM beschränkt sich auf die schnelle Anpassung an jeden beliebigen Netzwerkstatus. Die Anwendungen benötigen allerdings für bestimmte Aufgaben ein Wissen der „Vergangenheit“, z.B. um den Ausfall eines Knotens entdecken zu können.

Das NM unterstützt die Anwendungen dabei durch spezielle API-Dienste, um die aktuelle Netzwerkkonfiguration auslesen zu können.

4.2.3 Verhalten des logischen Rings

Das OSEK/VDX NM legt dezentrale und dynamische Algorithmen für das direkte NM fest, die einen logischen Ring zur Kommunikation nutzen.

Beim normalen und störungsfreien Ablauf gibt es nur zwei Aktivitäten:

- Integration eines Knoten in den logischen Ring
- Erkennen eines fehlerhaften Knotens und Neukonfiguration des logischen Rings

Die NM-Nachrichten treten dabei periodisch auf dem Bus auf.

Nach dem Start eines Knotens (siehe Abbildung 3) oder der Erkennung eines Knotenfehlers (siehe Abbildung 4), meldet der (neu hinzugekommene oder einen Fehler erkennende) Knoten seine Anwesenheit im Netzwerk durch das einmalige Senden seines „Lebenssignals“. Anschließend nimmt der Knoten am Austausch der Ringnachrichten teil. Die Ringnachrichten werden dabei an den logischen Nachfolger adressiert, soweit der Knoten Kenntnis von diesem hat. Stellt ein Knoten fest, dass er bei den Ringnachrichten übersprungen wurde, sendet auch dieser Knoten einmalig sein „Lebenssignal“ um seine Anwesenheit im Ring bekannt zu machen.

Alle Nachrichten im logischen Ring unterliegen fest vorgegebenen Zeitintervallen. Um Überschneidungen von Nachrichten zu verhindern, werden die „Lebenssignale“ asynchron zu den normalen Ringnachrichten gesendet.

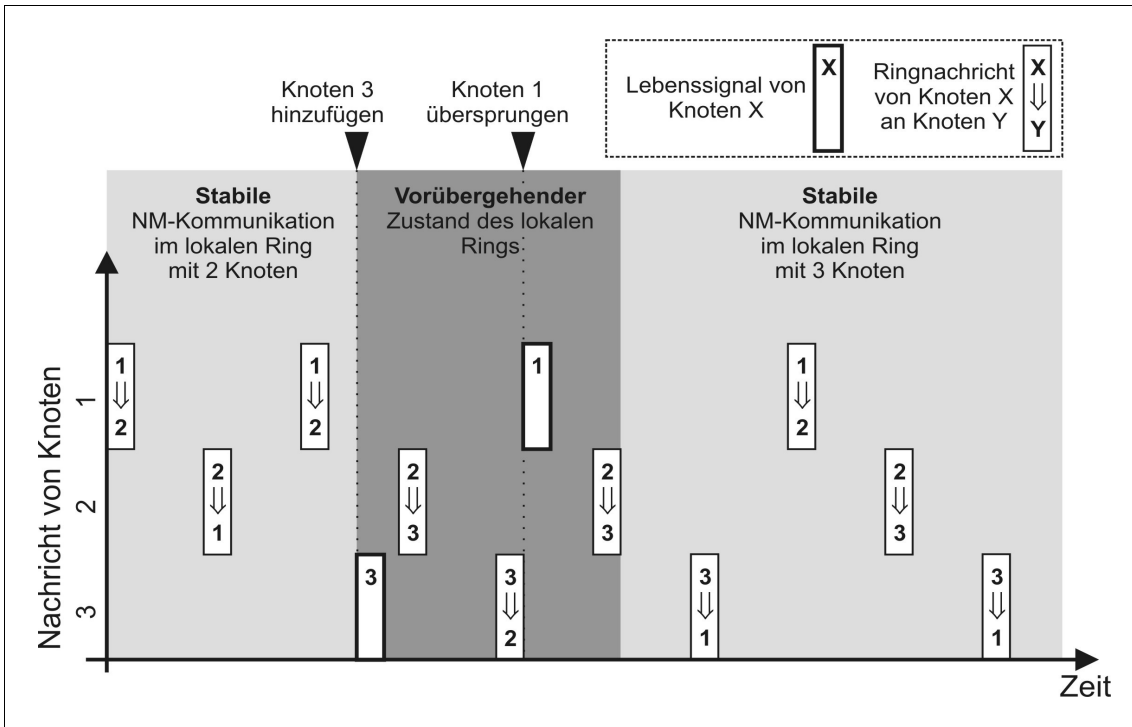


Abbildung 3: Integration eines Knotens in den logischen Ring

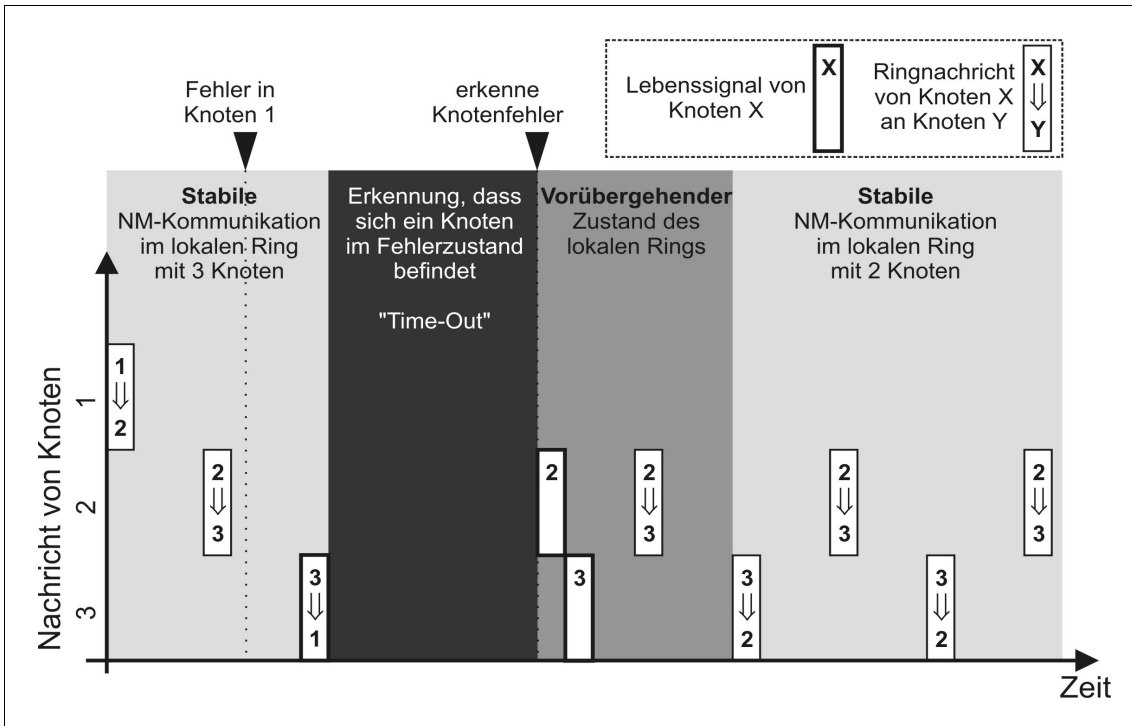


Abbildung 4: Erkennung eines Knotenfehlers und Neukonfiguration des logischen Rings

4.2.4 Datenaustausch

Das NM unterstützt den Austausch von Anwendungsdaten über den logischen Ring. Dies ist möglich, da zwischen dem Empfang und dem Senden der Ringnachrichten eine Verzögerung eingehalten werden muss. Während dieser Verzögerung darf die Anwendung die Daten (eines speziellen Datenfeldes, siehe NMPDU) verändern.

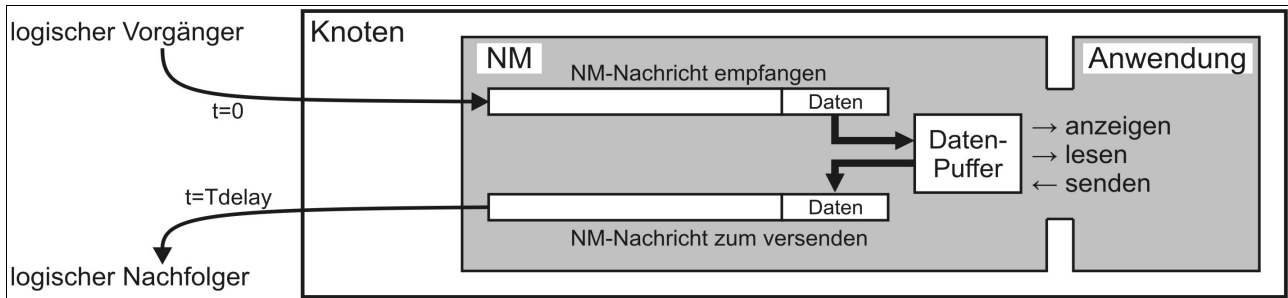


Abbildung 5: Austausch von Anwendungsdaten über den logischen Ring

Da dies allerdings nicht den Standardalgorithmen des OSEK/VDX-Standards entspricht, sind entsprechende Anpassungen der Algorithmen notwendig.

4.2.5 Network Management Protocol Data Unit (NMPDU)

Das OSEK/VDX NM unterstützt die Interoperabilität von Knoten, die von verschiedenen Zulieferern entwickelt wurden. Daher musste für das NM-Protokoll und die Repräsentation von NM-Daten ein einheitliches Format festgelegt werden, woraus die NMPDU hervorgegangen ist.

Die NMPDU unterteilt sich in ein Adressfeld, ein Steuerfeld und ein optionales Datenfeld.

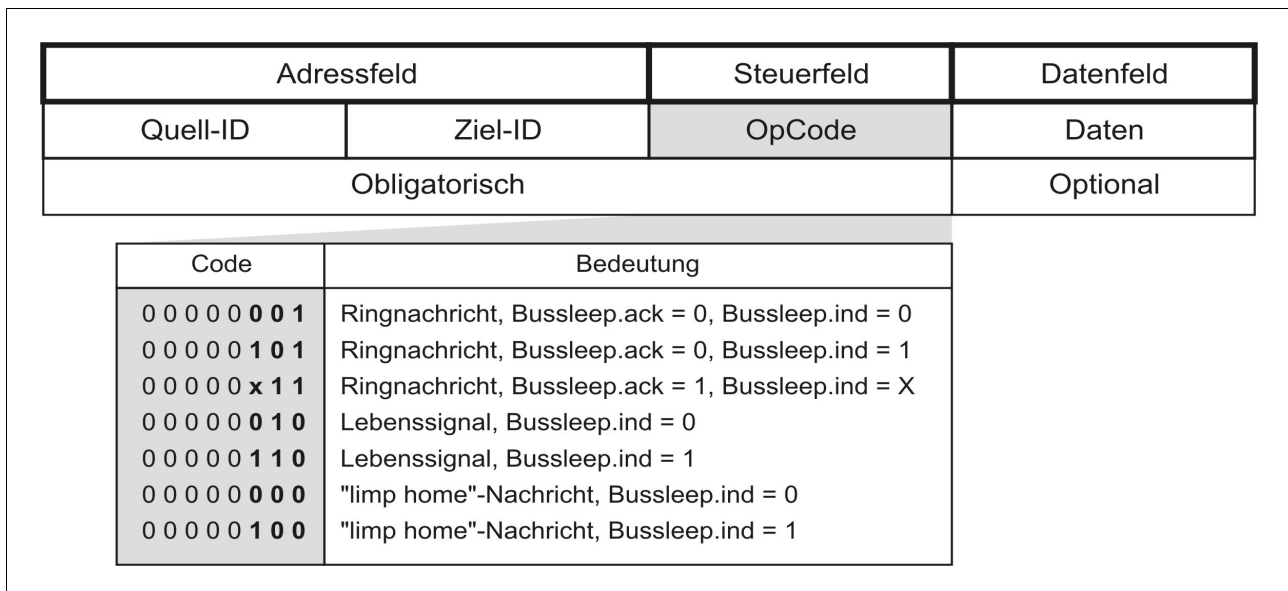


Abbildung 6: NMPDU

Das Adressfeld der NMPDU enthält:

- die ID des Absenders
- die ID des Empfängers

Das Steuerfeld der NMPDU enthält:

- einen vordefinierten OpCode zur Codierung der NM-Nachrichtentypen
- einen reservierten OpCode zur Codierung von Erweiterungen zukünftiger Standardisierungen oder zur Codierung von systemspezifischen Parametern

Das Datenfeld der NMPDU enthält:

- Daten der Anwendung
- bei Lebenssignalen z.B. die Weckursache des Knotens
- bei Ringnachrichten z.B. Fehlermeldungen

4.2.6 Beispiel zum Wechsel in den globalen Ruhemodus

Der Ruhezustand versetzt die ECU entweder in einen Low-Power-Modus oder schaltet die ECU komplett ab. Der Low-Power-Modus ist notwendig, falls die ECU bspw. noch Überwachungsaufgaben im Hintergrund ausführen soll.

Insgesamt soll die Stromaufnahme minimiert werden, so dass auch bei längerer Standzeit des Fahrzeuges die Batterie(en) nicht vollständig entladen werden.

Jeder Knoten kann versuchen das Netz in den globalen Ruhemodus zu versetzen. Im folgenden werden die einzelnen Schritte dazu erläutert:

- Möchte eine Anwendung den Ruhemodus einleiten, so setzt sie in der Ringnachricht das *Bussleep.ind*-Bit auf 1.
- Alle anderen Knoten im Netz können auf zwei Arten reagieren:
 - Ein Knoten, der bereit ist in den Ruhemodus zu wechseln, reicht das *Bussleep.ind*-Bit unverändert weiter.
 - Ein Knoten, der nicht bereit ist in den Ruhemodus zu wechseln (weil er z.B. noch Funktionen ausführen muss), setzt das *Bussleep.ind*-Bit auf 0 zurück.
- Nach einem Durchlauf im logischen Ring wird wieder eine Ringnachricht beim Initiator eintreffen.
 - Wurde das *Bussleep.ind*-Bit zurückgesetzt (auf 0), so bleibt das Netz aktiv.
 - Kommt das *Bussleep.ind*-Bit unverändert (auf 1 gesetzt) an, so sind alle Knoten für den Ruhemodus bereit.
- Der Knoten sendet nun eine Ringnachricht, in der er das *Bussleep.ack*-Bit auf 1 gesetzt hat. Diese Nachricht wird von allen Knoten gleichzeitig empfangen.
- Alle Knoten gehen nun, nach einer bestimmten Wartezeit, in den Ruhemodus über.
- Sollte während dieser Wartezeit eine weitere Ringnachricht empfangen werden, so wird das Netz wieder aufgebaut.

Ein Knoten im Ruhezustand kann aufgeweckt werden durch:

- einschalten durch den Benutzer
- Strom an Klemme 15
- die Anwendung selbst
- Signale auf dem Bus

5 Skalierbarkeit

Die grundlegenden Netzwerk-Management-Dienste sollten auf praktisch jeder Art von Knoten vorhanden sein. Bei der Vielzahl an verfügbaren Controllern würde dies aber bedeuten, dass eine low-cost Ein-Chip-Lösung mit minimaler Rechenleistung und minimaler Speicherausstattung die gleichen Dienste anzubieten hat wie eine high-tech-Lösung mit gewaltiger Rechenleistung und riesiger Speicherausstattung. Dies ist natürlich nicht machbar und auch nicht notwendig.

Aus diesem Grund wurde die Spezifikation in elementare Kern-Dienste und optionale Dienste unterteilt.

Zu den Kern-Diensten des indirekten NM zählen dabei nur die Initialisierung der Hardware, der Neustart nach einem Fehler und das Abschalten des Busses.

Zu den Kern-Diensten des direkten NM zählen das Ermitteln des logischen Nachfolgers, die Übertragung von NM-Nachrichten über den logischen Ring und das „Neu-Einloggen“ im Ring, falls der Knoten übersprungen wurde.

Optionale Dienste sind bspw. das Übertragen von Anwendungsdaten im Ring, die Erkennung von fehlerhaften Knoten per „Time-Out“, Wechseln in den Ruhezustand, usw.

6 Ausblick

Wie in Abschnitt 2.4.5 (NMPDU) bereits beschrieben, werden NM-Nachrichten mit einem bestimmten Datenformat im logischen Ring ausgetauscht. Da diese Nachrichten jedoch über ein physikalisches Kommunikationsmedium übertragen werden, ist es notwendig, dass eine NMPDU auf eine Nachricht des zugrunde liegenden Busprotokolls abgebildet wird. Die Kfz-Hersteller sind derzeit dabei diese Abbildung für CAN-Nachrichten festzulegen.

Zukünftige Anforderungen an das Netzwerk-Management sind bereits erkennbar. Dazu gehören:

- *Subnetz-Betrieb*
gezieltes aufwecken / schlafen-schicken von Knoten über die Busleitungen
 - Eine Untermenge von Knoten ist aktiv
 - Eine Untermenge von Knoten ist im low-power-Modus
- *Gateway-Unterstützung*
 - Überwachung von Knoten über verschiedene Netze hinweg
 - Verhandlung über die Betriebsmodi über verschiedene Netze hinweg

7 Erfahrungen in der Massenproduktion

Der OSEK/VDX Standard wird bereits seit einigen Jahren in Fahrzeugen eingesetzt. Je nach Hersteller wird dabei das direkte oder das indirekte Netzwerk-Management genutzt.

7.1 Indirektes NM bei Renault

In jedem neuen Fahrzeug von Renault wird das indirekte NM auf den ECUs des Antriebsstrangs eingesetzt (CAN-Bus).

Renault hat ein SDL-Modell des in seinen Fahrzeugen genutzten NM erstellt. ECU-Zulieferern wird dieses SDL-Modell zur Verfügung gestellt und dient als Basis für ihre Entwicklungen.

Den Zulieferern werden zudem Testprogramme zur Verfügung gestellt, um ihre Entwicklungen überprüfen zu können. Ein letzter Test wird allerdings dennoch direkt von Renault durchgeführt, die die einzelnen ECUs zu einem Gesamtsystem zusammenfügen.

Die notwendigen Ressourcen für das indirekte NM sind:

- 1 – 2 kByte ROM, inklusive Fehlerspeicher
- 0% Buslast (es gibt keine NM-spezifischen Nachrichten)

7.2 Direktes NM bei DaimlerChrysler

Mercedes-Benz setzt in seinen Fahrzeugen das direkte NM ein.

DaimlerChrysler hat eine Implementierung der NM-Algorithmen auf C-Code-Ebene entwickelt, die in jedem ihrer Fahrzeuge Anwendung findet. Dieser Code kann von den Zulieferern als Grundlage für eigene Entwicklungen genutzt werden.

Die notwendigen Ressourcen für direktes NM sind:

- 1,5 – 1,7 kByte ROM
- 4 – 8 kByte RAM
- 1% Buslast

8 Quellen

- [1] OSEK/VDX Portal. URL: <http://www.osek-vdx.org/> [Stand: 15.09.2007].
- [2] OSEK/VDX NM Spezifikation Version 2.5.3.
URL: <http://portal.osek-vdx.org/files/pdf/specs/nm253.pdf>
[Stand: 15.09.2007].
- [3] VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik (Hg.):
OSEK-VDX. open systems in automotive networks. Düsseldorf: VDI-Verlag, 2000.
- [4] Prof. Dr.-Ing. Thomas Form (2007): Folien zur Vorlesung „Datenbussysteme in
Straßenfahrzeugen“ an der TU Braunschweig.
URL: http://www.ifr.ing.tu-bs.de/lehre/downloads/skripte/skript_dbf_38.pdf
[Stand: 15.10.2007]