

eBUS Spezifikation

Netzwerkmanagement



V 1.0.2

03/2007

Inhalt

1	ÄNDERUNGSLISTE	3
2	REFERENZDOKUMENTE	3
3	ÜBER DIESES DOKUMENT	4
4	EINLEITUNG	4
5	EINORDNUNG DES NETZWERKMANAGEMENTS IN DEN EBUS KNOTEN	5
6	ZUSTÄNDE DES NETZWERKMANAGEMENTS	6
7	SCHNITTSTELLENSPEZIFIKATION: DATENINHALTE	7
7.1	ZYKLUSZEITEN	7
7.2	ZUSTANDSTABELLE	7
7.3	NETZSTATUS	8
7.4	STARTFLAG	8
7.5	SOLLKONFIGURATION	8
7.5.1	<i>STATISCHE SOLLKONFIGURATION</i>	9
7.5.2	<i>DYNAMISCHE ERZEUGUNG DER SOLLKONFIGURATION</i>	9
7.5.3	<i>KOMBINATION STATISCHE SOLLKONFIGURATION UND DYNAMISCHE ERZEUGUNG DER SOLLKONFIGURATION</i>	9
8	SCHNITTSTELLENSPEZIFIKATION: DIENSTE	10
8.1	SCHNITTSTELLENDIENSTE ZWISCHEN NETZWERKMANAGEMENT UND APPLIKATION	10
8.2	SCHNITTSTELLENDIENSTE ZWISCHEN NETZWERKMANAGEMENT UND EBUS TREIBER	10
8.3	NETZWERKMANAGEMENT BOTSCHAFTEN	11
9	ABLAUFDIAGRAMME DES NETZWERKMANAGEMENTS	12
9.1	ABLAUFDIAGRAMME NMINIT, NMRESET UND NMNORMAL	12
9.2	VORSCHLAG FÜR EINE NETZWERKMANAGEMENT IMPLEMENTIERUNG MIT APPLIKATIONSSPEZIFISCHEN REGELN ZUR DYNAMISCHEN ERZEUGUNG DER SOLLKONFIGURATION	14
10	IMPLEMENTIERUNGSVORSCHLÄGE	19
10.1	VORSCHLAG EINER IMPLEMENTIERUNG MIT STATISCHER SOLLKONFIGURATION UND EINER DEFAULT ZYKLUSZEIT	19
10.2	VORSCHLAG EINER IMPLEMENTIERUNG MIT STATISCHER SOLLKONFIGURATION UND SPEZIFISCHEN ZYKLUSZEITEN	20
10.3	VORSCHLAG EINER IMPLEMENTIERUNG MIT DYNAMISCHER ERZEUGUNG DER SOLLKONFIGURATION UND EINER DEFAULT ZYKLUSZEIT	22
10.4	VORSCHLAG EINER IMPLEMENTIERUNG MIT DYNAMISCHER ERZEUGUNG DER SOLLKONFIGURATION UND SPEZIFISCHEN ZYKLUSZEITEN	25
11	ABBILDUNGSVERZEICHNIS	27

1 Änderungsliste

Version	Datum	Bemerkungen
1.0	Mai 2000	Autoren: <i>M. Scheurer c&s group</i> <i>F. Fischer, c&s group</i> <i>R. Sanders, Karl Dungs GmbH & Co</i> <i>M. Bünnemeyer, Kromschröder</i> <i>T. Maier, Lamberti</i> <i>H. Falk, TEM</i> <i>B. Scheffold, Max Weishaupt GmbH</i> <i>A. Konzelmann, Wikon</i>
1.0.1	Februar 2001	Korrektur der Bilder 6 und 8
1.0.2	März 2007	Änderung Fußnoten Text: © User Club eBUS e.V.  www.eBUS.de nach © eBUS Interest Group  www.eBUS.de

2 Referenzdokumente

Referenz für die Entwicklung des Netzwerkmanagements für den eBUS ist das Dokument 'OSEK/VDX Networkmanagement, Concept and Application Programming Interface'.

© eBUS Interest Group  www.eBUS.de	eBUS Spezifikation Netzwerkmanagement	Version 1.0.2	Ausgabe 03 / 07	Seite 3
--	--	---------------	--------------------	------------

3 Über dieses Dokument

Die Spezifikation Netzwerkmanagement für den eBUS umfaßt folgende Inhalte:

- Beschreibung der Funktionalität der Dienste zwischen Netzwerkmanagement und Applikation
- Beschreibung der Funktionalität der Dienste zwischen Netzwerkmanagement und eBUS Treiber
- Beschreibung der stationsinternen und stationsübergreifenden Algorithmen zur Gewährleistung der Funktionalität der beschriebenen Dienste des Netzwerkmanagements
- Randbedingungen und Auslegung des Netzwerkmanagements

Die Art der Implementierung wird weitgehend offen gelassen, so daß applikationsspezifische Optimierung möglich ist.

4 Einleitung

Die Aufgabe des Netzwerkmanagements ist die Gewährleistung des sicheren Betriebs und Zusammenspiels aller Knoten am Netz. Diese Aufgabe erfüllt das Netzwerkmanagement durch


- Bestimmen der Konfiguration derjenigen Teilnehmer am Bus, die er für seine Funktionalität benötigt
- Überwachen der Konfiguration derjenigen Teilnehmer am Bus, die er für seine Funktionalität benötigt
- Überwachen des eigenen Knotens
- Bereitstellung von Statusinformationen für die Applikation, so daß anwendungsspezifisch reagiert werden kann

Das Netzwerkmanagement für den eBUS folgt dem Konzept des Indirekten Netzwerkmanagements¹. Das Indirekte Netzwerkmanagement bestimmt und überwacht die Konfiguration anhand zyklischer Anwendungsbotschaften. Zyklische Botschaften der einzelnen Knoten sind daher für die Funktionalität des Netzwerkmanagements zwingend erforderlich.

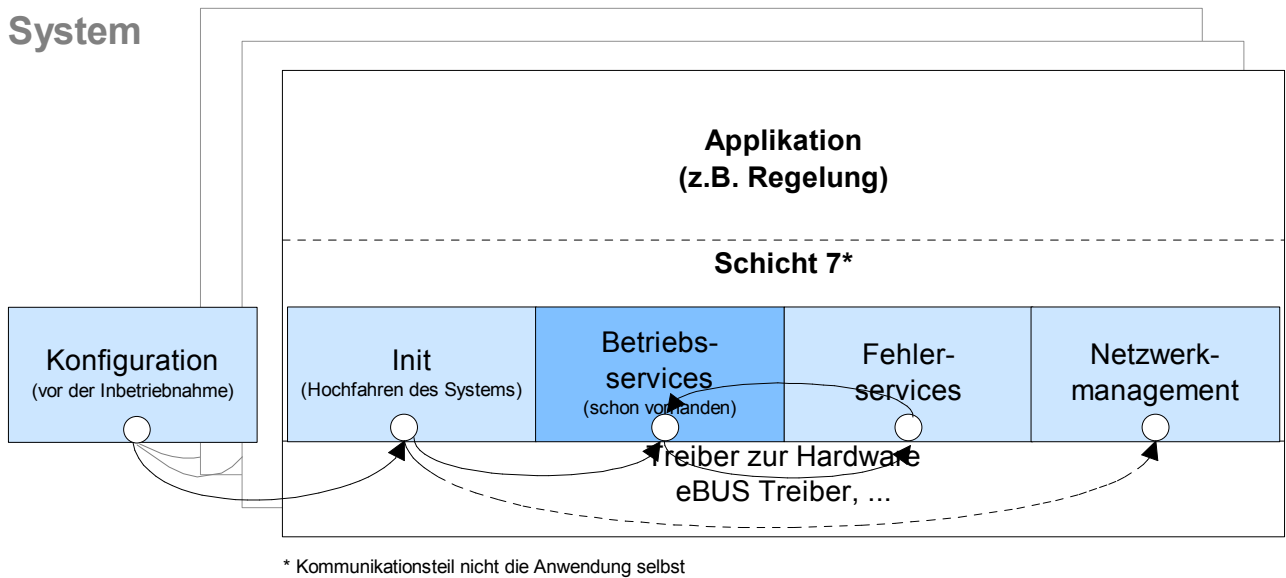
Die vorliegende Spezifikation des Netzwerkmanagements berücksichtigt die folgenden geforderten Randbedingungen:

- minimaler Ressourcen Bedarf
- geringer Implementierungsaufwand
- keine zusätzliche Buslast
- Definition der Funktionalität der Schnittstellen zur Applikation und zum eBUS Treiber
- Datenformate der Schnittstellen nicht vorgeschrieben
- standardisierte Fehlermeldungen
- Knoten am Bus, die keine Netzwerkmanagement Implementierung haben, beeinflussen die Funktionalität des Netzwerkmanagements in den anderen Knoten im Netz nicht
⇒ die Implementierung des Netzwerkmanagements ist optional
- Slaves haben kein Netzwerkmanagement, jeder Slave wird von all den Mastern überwacht, die ihn für ihre Funktionalität benötigen
- Anwendbarkeit in großen und in kleinen Systemen
- gute Testbarkeit

¹ OSEK/VDX Networkmanagement, Concept and Application Programming Interface, Version 2.5

 <p>© eBUS Interest Group www.eBUS.de</p>	<p>eBUS Spezifikation</p> <p>Netzwerkmanagement</p>	<p>Version 1.0.2</p>	<p>Ausgabe 03 / 07</p>	<p>Seite 4</p>
--	---	----------------------	----------------------------	--------------------

5 Einordnung des Netzwerkmanagements in den eBUS Knoten



- vorhanden in den Services der Schicht 7 Spezifikation
- nur teilweise definiert

Abbildung 1: Einordnung des Netzwerkmanagements in den eBUS Knoten

6 Zustände des Netzwerkmanagements

Das Netzwerkmanagement verfügt über drei Zustände:

- Initialisierungsphase (NMInit)
- Resetphase (NMReset)
- Überwachungsphase (NMNormal)

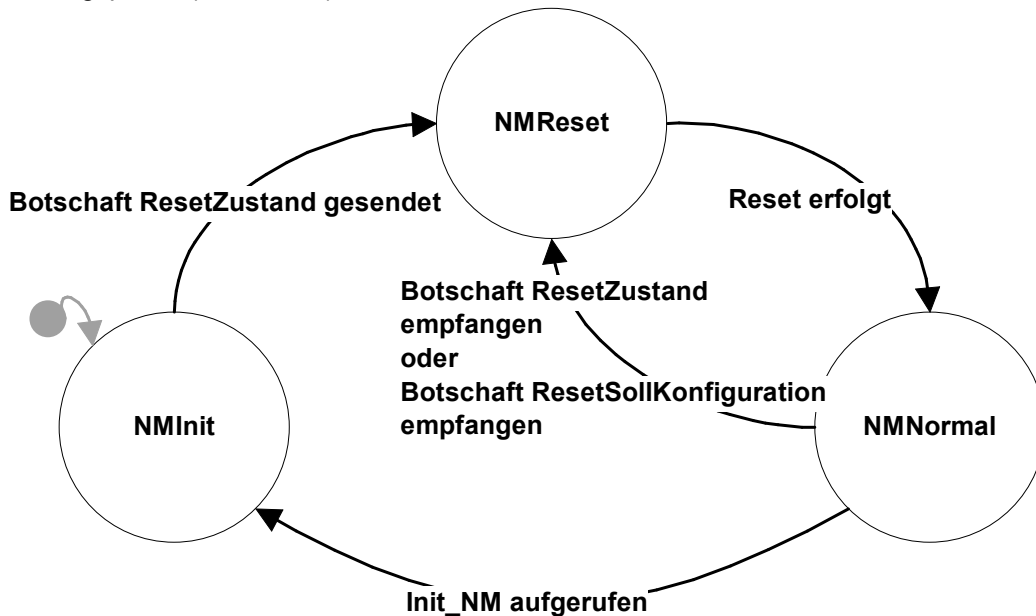


Abbildung 2: Zustandsdiagramm des Netzwerkmanagements

Nachdem ein Teilnehmer an den eBUS angeschlossen wurde, befindet sich das Netzwerkmanagement des neu hinzugekommenen Knotens im Zustand NMInit. In diesem Zustand wird eine Botschaft ResetZustand abgesetzt, die bei den bereits vorhandenen Teilnehmern am Bus einen Übergang in den Zustand NMReset veranlasst. Der neu hinzugekommene Teilnehmer geht ebenfalls in den Zustand NMReset über. In diesem Zustand werden Zustandstabelle, Netzstatus und interne Variable der Implementation auf ihre Default Werte gesetzt (Reset). Das Startflag wird gesetzt.

Wenn eine dynamische Sollkonfiguration² eingesetzt wird und diese noch nicht erzeugt wurde, wird die Sollkonfiguration initialisiert.

Hat ein Teilnehmer die Botschaft ResetSollKonfiguration erhalten, so wird eine bereits erzeugte dynamische Sollkonfiguration gelöscht und die Sollkonfiguration neu initialisiert.

Anschließend erfolgt ein sofortiger Übergang in den Zustand NMNormal. In diesem Zustand wird gegebenenfalls die Sollkonfiguration dynamisch erzeugt, die dynamisch erzeugte oder statische Sollkonfiguration wird überwacht.


Alle zu überwachenden Knoten werden zunächst als 'in Ordnung' (OK) betrachtet, der Netzstatus ist daher OK. Mit dem im Zustand NMReset gesetzten Startflag bietet das Netzwerkmanagement der Applikation die Möglichkeit zu unterscheiden, ob ein Knoten in der Zustandstabelle als OK eingetragen ist, weil ein Reset durchgeführt wurde (Zykluszeit noch nicht abgelaufen, also noch nicht überprüft) oder ob er im laufenden Überwachungsbetrieb als OK gilt.

Das Startflag wird zurückgesetzt, wenn die maximale (längste) Zykluszeit mit der Botschaften erwartet werden, abgelaufen ist. Alle zu überwachenden Knoten, die nach Zurücksetzen des Startflags als OK eingetragen sind, gelten als vorhanden.

Bei neu erkannten Fehlern (eigener Knoten nicht sendefähig oder ein zu überwachender Knoten ausgefallen) wird einmalig eine Fehlermeldung³ (Ausfall_Botschaft) abgesetzt. Diese Ausfallbotschaft kann optional mit Mindestabständen von 15 Minuten wiederholt werden, solange mindestens ein Knoten als ausgefallen gilt. Wird im Überwachungsbetrieb der Dienst Init_NM aufgerufen, wechselt das Netzwerkmanagement des Knotens in den Zustand NMInit.

² dynamische Sollkonfiguration s. Kapitel 'Sollkonfiguration'

³ Fehlermeldungen standardisiert in eBUS Spezifikation Anwendungsschicht - OSI 7 V 1.4

© eBUS Interest Group  www.eBUS.de	eBUS Spezifikation Netzwerkmanagement	Version 1.0.2	Ausgabe 03 / 07	Seite 6
---	--	---------------	--------------------	------------

7 Schnittstellenspezifikation: Dateninhalte

Die Schnittstellenspezifikation der Dateninhalte des Netzwerkmanagements legt einerseits die Daten fest, die dem Netzwerkmanagement zur Verfügung gestellt werden müssen, andererseits aber auch die Daten, die vom Netzwerkmanagement zur Verfügung gestellt werden.

Die entsprechenden Dateninhalte werden im folgenden aufgeführt. Die Datentypen werden nicht festgelegt, die Art der Implementierung bleibt weitestgehend Entwickler überlassen.

Sollkonfiguration	Die Sollkonfiguration ist die Auflistung der zu überwachenden Knoten und implizit des eigenen Knotens. Als Identifikation dienen die eBUS Adressen der jeweiligen Knoten
Zustandstabelle	Die Zustandstabelle enthält für jeden der zu überwachenden Knoten (einschließlich des eigenen Knotens) einen Eintrag, der den Zustand des jeweiligen Knotens widerspiegelt (Knoten vorhanden / Knoten nicht vorhanden)
Zykluszeiten	Die Zeiten innerhalb derer von den zu überwachenden Knoten eine Botschaft erwartet wird bzw. innerhalb der vom eigenen Knoten eine Nachricht abgesetzt werden muß. Es gibt zwei Varianten: <ul style="list-style-type: none"> - die Default Zykluszeit gilt einheitlich für alle zu überwachenden Knoten - spezifische Zykluszeit für jeden einzelnen Knoten
Startflag	Das Startflag wird nach einem Reset gesetzt und wird zurückgesetzt, wenn die längste zu überwachende Zykluszeit einmal abgelaufen ist.
optional: Netzstatus	Der Netzstatus stellt eine Summeninformation dar. Netzstatus = OK wenn alle zu überwachenden Knoten vorhanden sind und der eigene Knoten in Ordnung ist, andernfalls gilt Netzstatus = NOK

7.1 Zykluszeiten

Ein Netzwerkmanagement Knoten muß für jeden der von ihm zu überwachenden Knoten eine Zykluszeit kennen, innerhalb der eine Botschaft von dem jeweiligen Knoten erwartet wird.

Das Netzwerkmanagement muß außerdem die Zeit kennen, innerhalb der er eine Nachricht senden muß.

Die Zykluszeit ist applikationsspezifisch und muß daher an die Anwendung angepaßt werden. Zwei Varianten der Festlegung der Zykluszeiten sind möglich:

- Nutzung einer Default Zykluszeit einheitlich für alle zu überwachenden Knoten
- Nutzung von spezifischen Zykluszeiten für jeden einzelnen Knoten

Spezifische Zykluszeiten müssen den zu überwachenden Knoten und dem eigenen Knoten fest zugeordnet werden und dem Netzwerkmanagement zur Verfügung gestellt werden.

7.2 Zustandstabelle

In der Zustandstabelle muß für jeden der zu überwachenden Knoten und den eigenen Knoten ein Eintrag vorhanden sein. Die Einträge müssen den jeweiligen Knoten fest zugeordnet sein.

Hat ein Knoten sich innerhalb der ihm zugeordneten Zykluszeit nicht gemeldet oder konnte der eigene Knoten innerhalb der ihm zugeordneten Zykluszeit keine Botschaft absetzen, so wird der zugehörige Eintrag in der Zustandstabelle auf NOK gesetzt. Sobald das Netzwerkmanagement eine Nachricht von einem zu überwachenden Knoten empfängt oder eine Nachricht vom eigenen Knoten abgesetzt wurde, wird der zugehörige Eintrag in der Zustandstabelle auf OK gesetzt.

© eBUS Interest Group  www.eBUS.de	eBUS Spezifikation Netzwerkmanagement	Version 1.0.2	Ausgabe 03 / 07	Seite 7
---	--	---------------	--------------------	------------

Die Zustandstabelle wird vom Netzwerkmanagement beim Senden und Empfangen einer Botschaft oder auf Anforderung (Dienst GetStatus) aktualisiert und der Applikation zur Verfügung gestellt.

Nach einem Reset sind die Werte der Zustandstabelle mit dem Default Wert OK belegt. Ein OK in der Zustandstabelle trifft keine Aussage über den wirklichen Zustand des zu überwachenden Knotens solange das Startflag gesetzt ist. Das Startflag wird zurückgesetzt, wenn die längste der zu überwachenden Zykluszeiten einmal abgelaufen ist. Erst dann wurden sicher alle zu überwachenden Knoten auf das Senden einer Botschaft innerhalb der ihnen jeweils zugeordneten Zykluszeiten überprüft.

Bleibt eine Botschaft eines Knotens mit einer kürzeren Zykluszeit innerhalb dieser Zeit aus, wird der Zustand dieses Knotens auf NOK gesetzt, ein NOK stellt eine reelle Aussage über den Zustand eines Knotens auch bei gesetztem Startflag dar.

7.3 Netzstatus

Der Netzstatus stellt eine Summeninformation über die Zustände aller zu überwachenden Knoten und des eigenen Knotens dar.

Der Netzstatus gilt als 'in Ordnung' (OK) wenn der Zustand aller zu überwachenden Knoten in der Zustandstabelle als OK gekennzeichnet ist und der überwachende Knoten sendefähig ist. Er gilt als 'nicht in Ordnung' wenn mindestens ein Knoten in der Zustandstabelle als NOK gekennzeichnet ist oder der überwachende Knoten nicht sendefähig ist.

Nach einem Reset ist der Netzstatus mit seinem Default Wert OK belegt. Ein OK des Netzstatus trifft keine wirkliche Aussage über den tatsächlichen Zustand aller Teilnehmer am Bus solange das Startflag gesetzt ist. Erst wenn das Startflag zurückgesetzt ist, wurden sicher alle zu überwachenden Knoten auf das Senden einer Botschaft innerhalb der ihnen jeweils zugeordneten Zykluszeiten überprüft. Ein NOK des Netzstatus ist dagegen auch bei gesetztem Startflag ein Indikator dafür, daß mindestens ein Knoten als fehlerhaft erkannt wurde.

Die Bereitstellung des Netzstatus durch das Netzwerkmanagement ist optional, wird aber empfohlen. Wird auf den Netzstatus als Summeninformation verzichtet, muß die Applikation jeweils die gesamte Zustandstabelle durchlaufen, um eine Aussage über den Zustand der zu überwachenden Sollkonfiguration zu erhalten. Bei Bereitstellung einer Summeninformation ist das Durchlaufen der Zustandstabelle nur dann notwendig, wenn ein NOK angezeigt wird und die fehlerhaften Knoten bestimmt werden sollen.

7.4 Startflag

Alle zu überwachenden Knoten werden in der Zustandstabelle zunächst als 'in Ordnung' (OK) betrachtet, der Netzstatus ist daher OK. Mit dem im Zustand NMRreset gesetztem Startflag bietet das Netzwerkmanagement der Applikation die Möglichkeit zu unterscheiden, ob ein Knoten in der Zustandstabelle als OK eingetragen ist, weil ein Reset durchgeführt wurde (Zykluszeit für den Knoten unter Umständen noch nicht abgelaufen und Knoten daher noch nicht geprüft) oder ob er im laufenden Überwachungsbetrieb als OK gilt.

Das Startflag wird zurückgesetzt, wenn die maximale (längste) Zykluszeit mit der Botschaften erwartet werden, abgelaufen ist. Alle zu überwachenden Knoten, die nach Zurücksetzen des Startflags als OK eingetragen sind, gelten definitiv als vorhanden.

7.5 Sollkonfiguration

Das Netzwerkmanagement eines Knotens überwacht diejenigen Teilnehmer, die der Knoten für seine Funktionalität benötigt.

Die Adressen dieser Teilnehmer müssen ihm als Sollkonfiguration bekannt gemacht werden. Hierfür gibt es zwei Möglichkeiten:

- die Sollkonfiguration kann statisch vorgegeben werden, soweit diese bei Inbetriebnahme bekannt ist und sich während des Betriebs nicht ändert
- die Sollkonfiguration kann dynamisch während der Inbetriebnahme erzeugt werden.

In der Sollkonfiguration dienen die eBUS Adressen als Identifikation der einzelnen zu überwachenden Knoten. In die Sollkonfiguration werden sowohl Slaves als auch Master aufgenommen. In die Sollkonfiguration muß die eBUS Adresse des eigenen Knotens nicht explizit aufgenommen werden, da diese Adresse bekannt ist.

© eBUS Interest Group  www.eBUS.de	eBUS Spezifikation Netzwerkmanagement	Version 1.0.2	Ausgabe 03 / 07	Seite 8
---	--	---------------	--------------------	------------

Beispiel:

Knoten 1: benötigt die Außentemperatur, die Innentemperatur und die Uhrzeit.

Knoten 2(Slave): kann auf Abfrage die Außentemperatur liefern.

Knoten 3(Master): liefert zyklisch ein Telegramm, in der Innentemperatur enthalten ist

Knoten 5(Master): liefert zyklisch ein Telegramm, in der die Uhrzeit enthalten ist

⇒ In der Sollkonfiguration von Knoten 1 sind explizit die eBUS Adressen der Knoten 2, 3 und 5 enthalten und implizit die eBUS Adresse des eigenen Knotens.

7.5.1 Statische Sollkonfiguration

Die Sollkonfiguration kann dem Netzwerkmanagement statisch vorgegeben werden, soweit die zu überwachenden Adressen bei Inbetriebnahme bekannt sind und sich diese während des Betriebs nicht mehr ändern.

Die statische Sollkonfiguration muß dem Netzwerkmanagement während des gesamten Betriebs zur Verfügung stehen.

Sollen auf die Überwachung der Knoten von der Default Zykluszeit abweichende Zykluszeiten angewandt werden, so muß jedem Knoten in der Sollkonfiguration die entsprechende Zykluszeit fest zugeordnet werden.

7.5.2 Dynamische Erzeugung der Sollkonfiguration

Wenn die zu überwachenden Adressen bei Inbetriebnahme nicht bekannt sind oder sich die zu überwachenden Adressen während des Betriebs ändern können, muß die Sollkonfiguration während des Betriebs dynamisch erzeugt werden.

In diesem Fall wird die Sollkonfiguration im Zustand NMReset zusätzlich zur eBUS Adresse des eigenen Knotens mit je einem Wert 0xAA für z. B. jeden geforderten Service gefüllt.

Für jeden -optionalen- Service, den ein Knoten nicht zwingend für seine Funktionalität benötigt, den er aber nutzt, sofern ihm dieser Service geliefert wird, wird eine 'Adresse' 0xA9 in die Sollkonfiguration geschrieben. Diesen 'Adressen' 0xAA und 0xA9 werden die zu den Services gehörenden Zykluszeiten zugeordnet.

Im Zustand NMNormal wird die Sollkonfiguration in Abhängigkeit von applikationsspezifischen Regeln gefüllt. Diese Initialisierung der Sollkonfiguration wird nur dann vorgenommen, wenn die Sollkonfiguration leer ist oder wenn die Botschaft ResetSollkonfiguration empfangen wurde.

Beim Empfang eines geforderten Services wird der zugehörige Eintrag 0xAA durch die eBUS Adresse des Senders überschrieben bzw. gelöscht wenn der Sender bereits in der Sollkonfiguration vorhanden ist.

Das Vorhandensein von 0xAA in der Sollkonfiguration führt zum Absetzen der Ausfall_Botschaft wenn die zugeordnete zu überwachende Zykluszeit abgelaufen ist. Diese Botschaft kennzeichnet im Datenteil die Adresse 0xAA als 'ausgefallen'.

Beim Empfang eines optionalen Services wird der zugehörige Eintrag 0xA9 durch die eBUS Adresse des Senders überschrieben bzw. gelöscht wenn der Sender bereits in der Sollkonfiguration vorhanden ist.


Das Vorhandensein von 0xA9 in der Sollkonfiguration führt nicht zum Absetzen der Ausfall_Botschaft wenn die zugeordnete zu überwachende Zykluszeit abgelaufen ist.

Ein Vorschlag für eine Netzwerkmanagement Implementierung mit applikationsspezifischen Regeln zur dynamischen Erzeugung der Sollkonfiguration findet sich im Kapitel 'Ablaufdiagramme des Netzwerkmanagements'.

7.5.3 Kombination statische Sollkonfiguration und dynamische Erzeugung der Sollkonfiguration

Falls nur ein Teil der zu überwachenden Adressen bei Inbetriebnahme nicht bekannt ist, ist es möglich eine Kombination aus statischer Sollkonfiguration und dynamischer Erzeugung der Sollkonfiguration zu nutzen.

In diesem Fall wird die Sollkonfiguration sowohl mit den bekannten eBUS Adressen als auch mit den 'Adressen' 0xAA für z.B. die geforderten Services und den 'Adressen' 0xA9 für optionale Services initialisiert. Die Zuordnung der Zykluszeiten erfolgt wie bereits beschrieben.

© eBUS Interest Group  www.eBUS.de	eBUS Spezifikation Netzwerkmanagement	Version 1.0.2	Ausgabe 03 / 07	Seite 9
---	--	---------------	--------------------	------------

8 Schnittstellenspezifikation: Dienste

Die Schnittstelle zwischen Netzwerkmanagement und Applikation bzw. Netzwerkmanagement und eBUS Treiber wird im folgenden durch Dienstaufrufe beschrieben⁴.

Die Daten (s. auch Kapitel 'Schnittstellenspezifikation: Dateninhalte') können durch direkte Referenzierung bei einem Dienstaufruf oder aber auch über einen gemeinsamen Zugriff auf globale Variable ausgetauscht werden. Die Bearbeitung der spezifizierten Aufgaben der Dienste obliegt dem Implementierer.

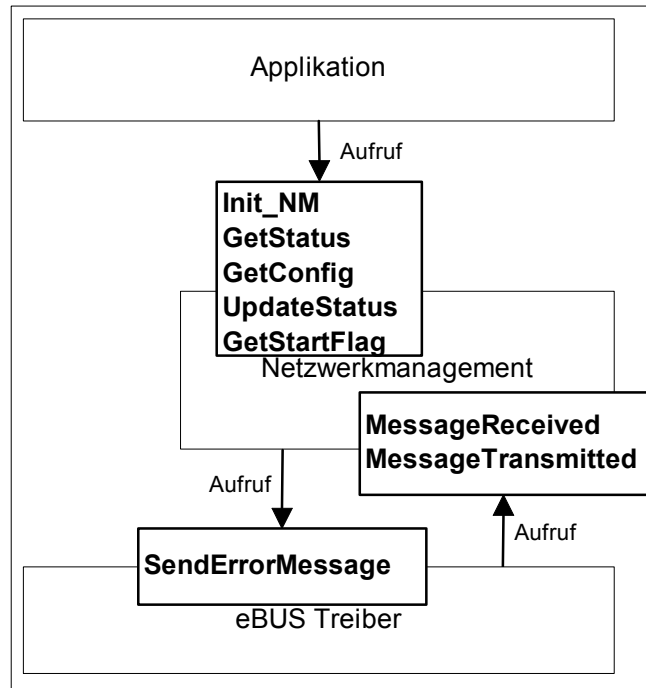


Abbildung 3: Architektur Modell

8.1 Schnittstellendienste zwischen Netzwerkmanagement und Applikation

Init_NM	Startet das Netzwerkmanagement.
UpdateStatus	Aktualisiert Zustandstabelle und gegebenenfalls den Netzstatus.
GetConfig	Liefert die Zustandstabelle mit den Zuständen der zu überwachenden Knoten.
GetStartFlag	Liefert das Startflag.
optional: GetStatus	Liefert eine Summeninformation über den Status der zu überwachenden Konfiguration und des eigenen Knotens (Netzstatus)

8.2 Schnittstellendienste zwischen Netzwerkmanagement und eBUS Treiber

MessageReceived	Veranlaßt Aktualisierung von Zustandstabelle und gegebenenfalls dem Netzstatus (UpdateStatus).
MessageTransmitted	Veranlaßt Aktualisierung von Zustandstabelle und gegebenenfalls dem Netzstatus (UpdateStatus).
SendErrorMessage	Sendet ein Fehlertelegramm auf den eBUS aus,

⁴ eine Implementation als Dienst ist nicht zwingend erforderlich, Beschreibung dient zum weiteren Verständnis der funktionalen Definition des Netzwerkmanagements

sobald ein überwachter Knoten ausgefallen ist.

8.3 Netzwerkmanagement Botschaften

In der *eBUS Spezifikation Anwendungsschicht – OSI 7* sind entsprechende Services definiert. Bei der Implementierung eines Netzwerkmanagements ist mindestens die Unterstützung der folgenden Services erforderlich:

Service FFh 00h	Reset Zustand NM
Service FFh 02h	Ausfallbotschaft

Die Unterstützung der weiteren Services zum Netzwerkmanagement ist optional.

9 Ablaufdiagramme des Netzwerkmanagements

9.1 Ablaufdiagramme NMInit, NMReset und NMNormal

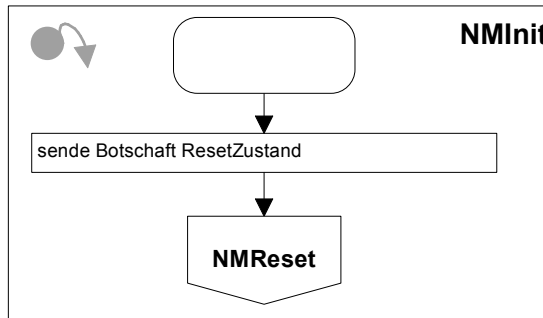


Abbildung 4: Ablaufdiagramm NMInit

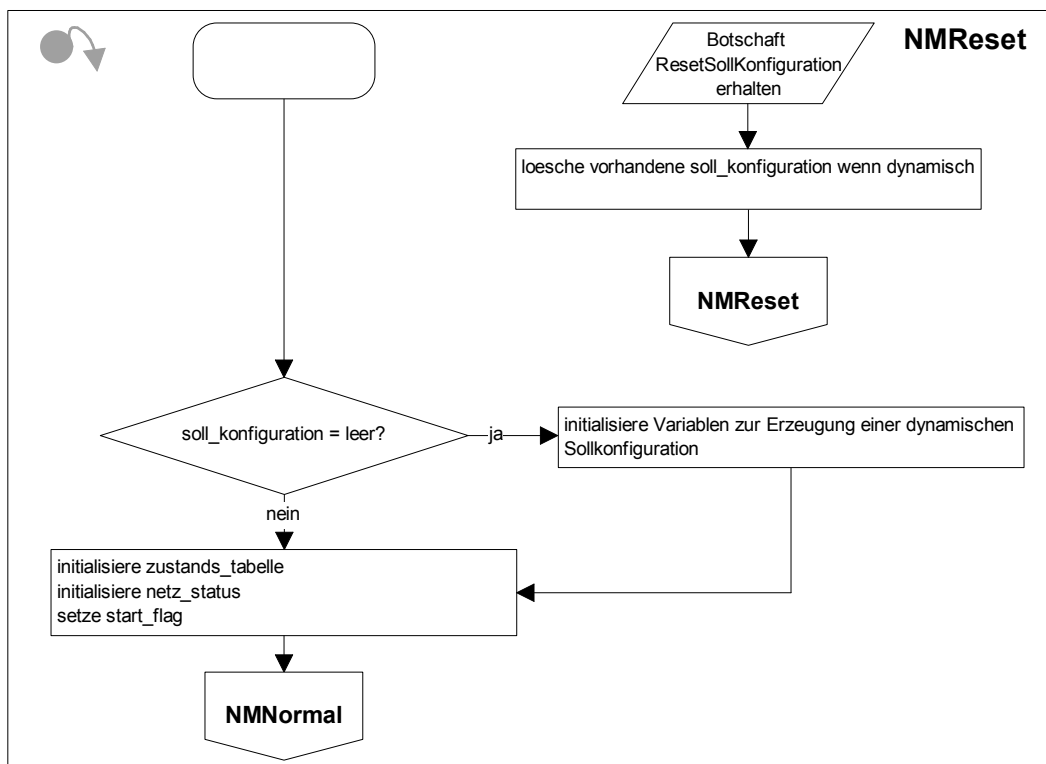


Abbildung 5: Ablaufdiagramm NMReset

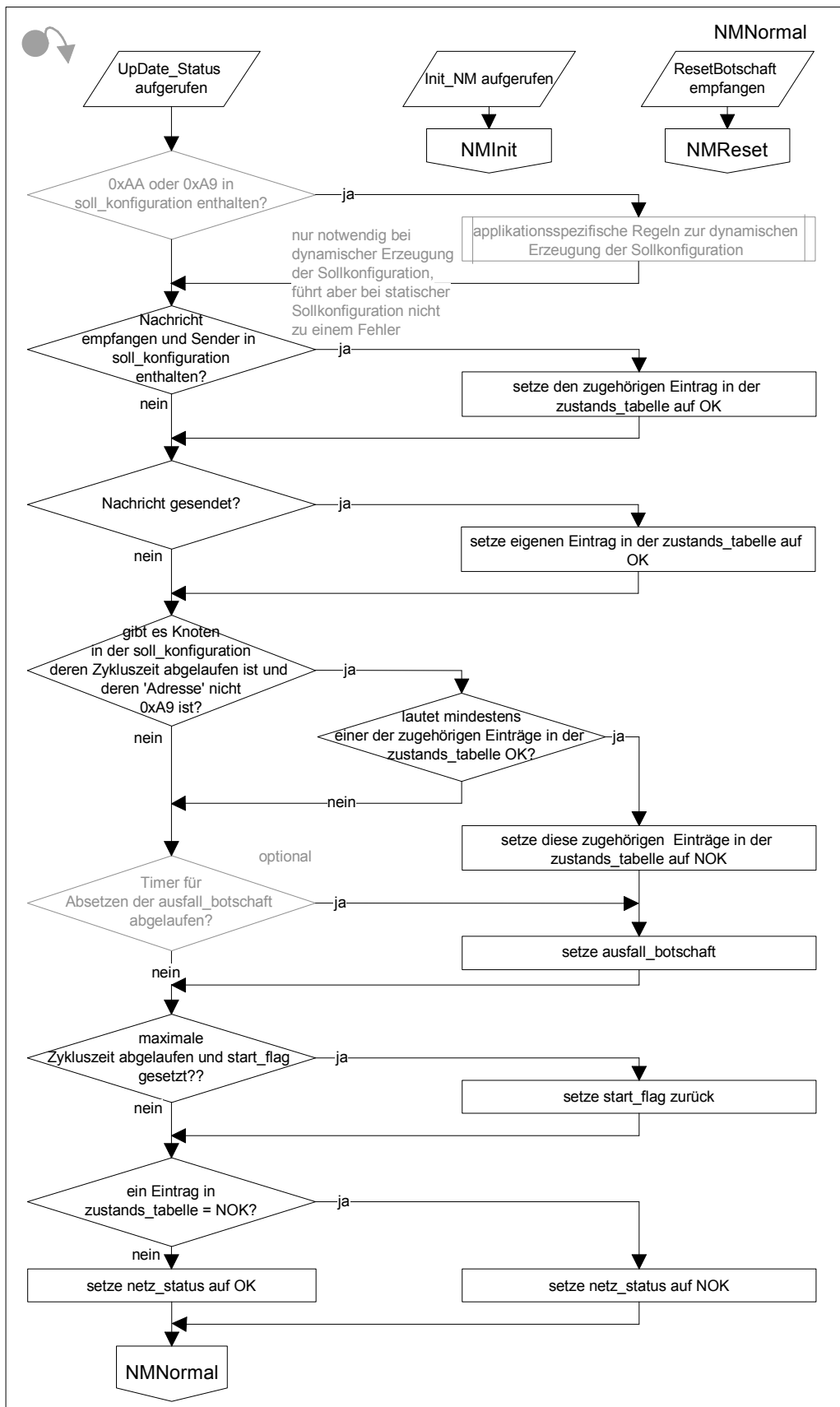


Abbildung 6: Ablaufdiagramm NMNormal

9.2 Vorschlag für eine Netzwerkmanagement Implementierung mit applikationsspezifischen Regeln zur dynamischen Erzeugung der Sollkonfiguration

Ein Knoten benötigt bestimmte Daten, die ihm von einem anderen Teilnehmer mit dem Service PB SB zur Verfügung gestellt werden. Die Adresse dieses Teilnehmers ist bei der Inbetriebnahme nicht bekannt. Die Sollkonfiguration muß daher während des Betriebs erzeugt werden.

Der Knoten führt daher eine `erzeuger_liste`, diese enthält:

- die Services, die der Knoten für seine Funktionalität benötigt und deren Sender er überwachen soll
- zu jedem Service ein Flag `received/not received`
- zu jedem Service die zu überwachende Zykluszeit, sofern für die Services eine spezifische Zykluszeit gelten soll

Aus der `erzeuger_liste` wird während des Betriebs die Sollkonfiguration erzeugt. Im folgenden ist ein möglicher Ablauf der Sollkonfigurationsfindung dargestellt (Verhalten in `NMInit` wie bei statischer Sollkonfiguration).

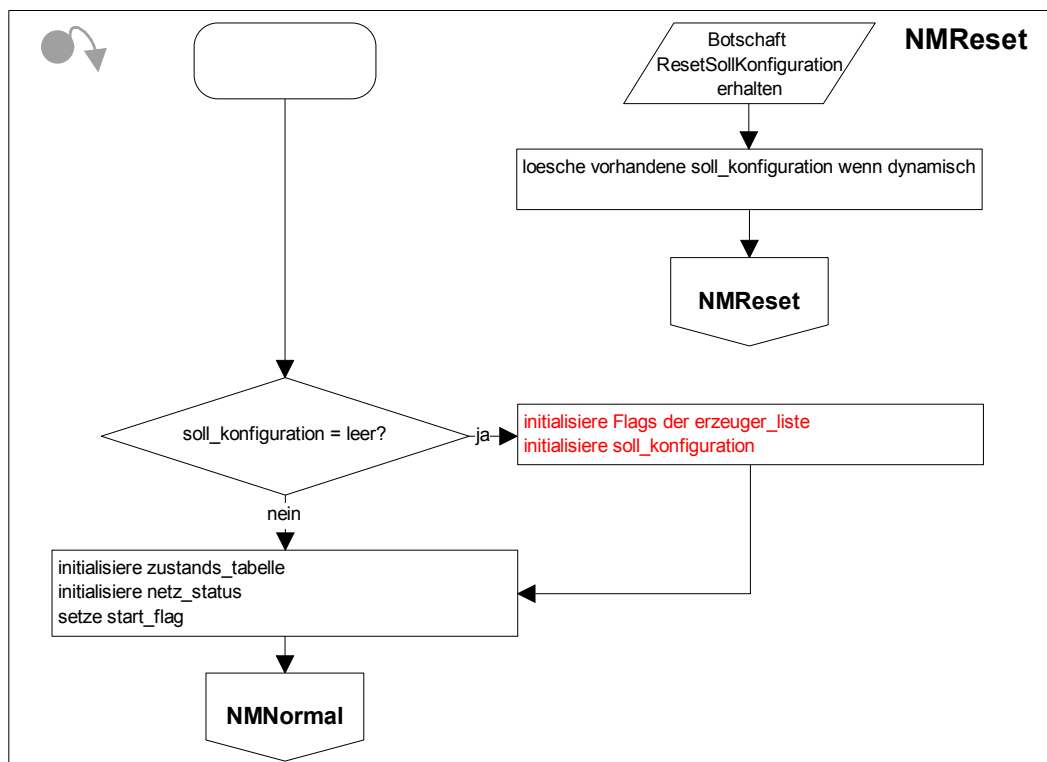


Abbildung 7: Vorschlag zur dynamischen Erzeugung der Sollkonfiguration - Ablaufdiagramm für den Zustand `NMReset`

Beim Übergang in den Zustand `NMNormal` sind `erzeuger_liste`, `soll_konfiguration`, `zustands_tabelle`, `start_flag` und `netz_status` wie folgt initialisiert:

PB 1	SB 1	Zeit 1	not received	...	PB n	SB n	Zeit n	not received	erzeuger_liste
0xAA	...	0xAA	eigene Adresse						soll_konfiguration
Zeit 1	...	Zeit n	eigene Zeit						Zykluszeiten
OK	...	OK	OK						zustands_tabelle
OK									netz_status
gesetzt									start_flag

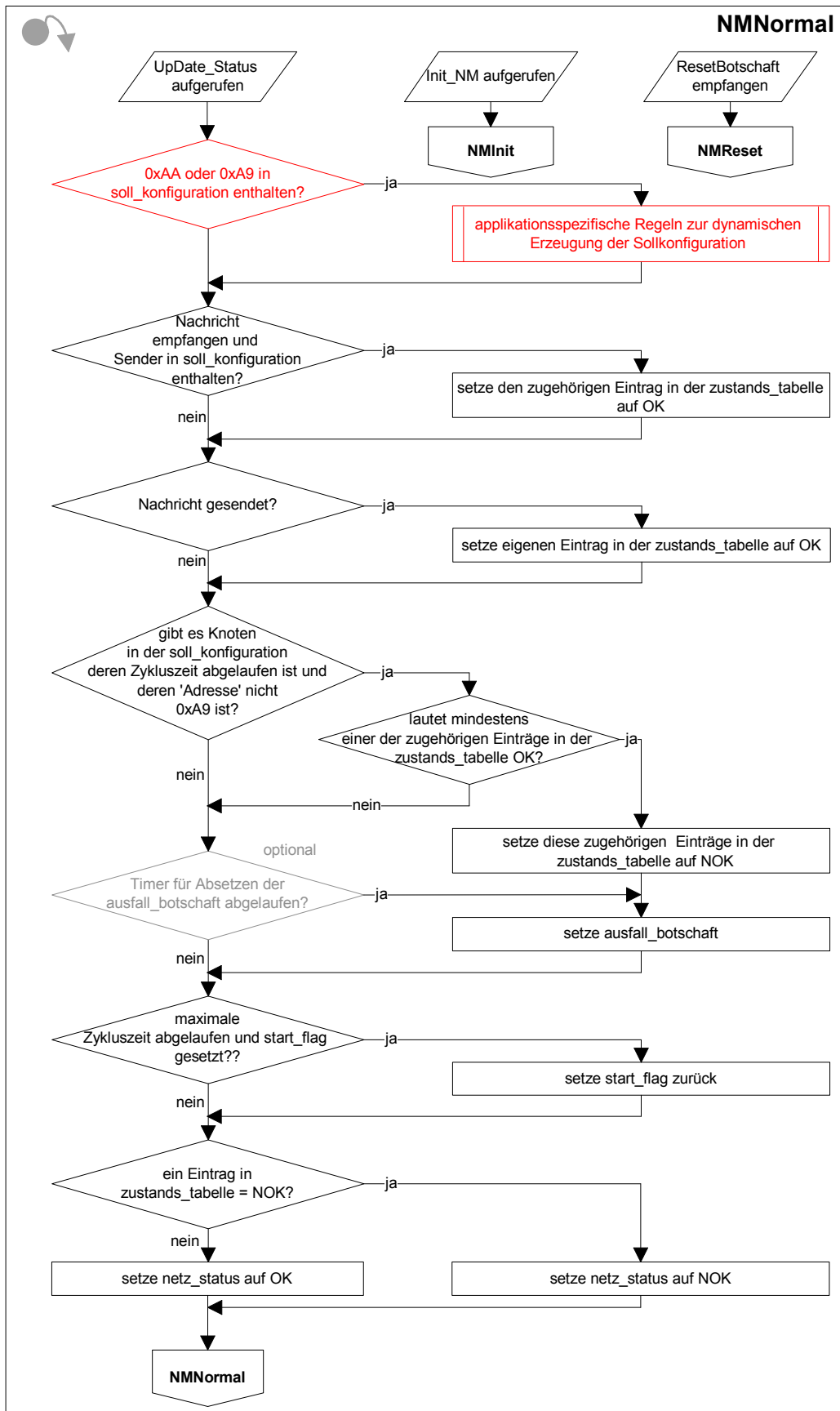


Abbildung 8: Vorschlag zur dynamischen Erzeugung der Sollkonfiguration - Ablaufdiagramm für den Zustand NMNormal

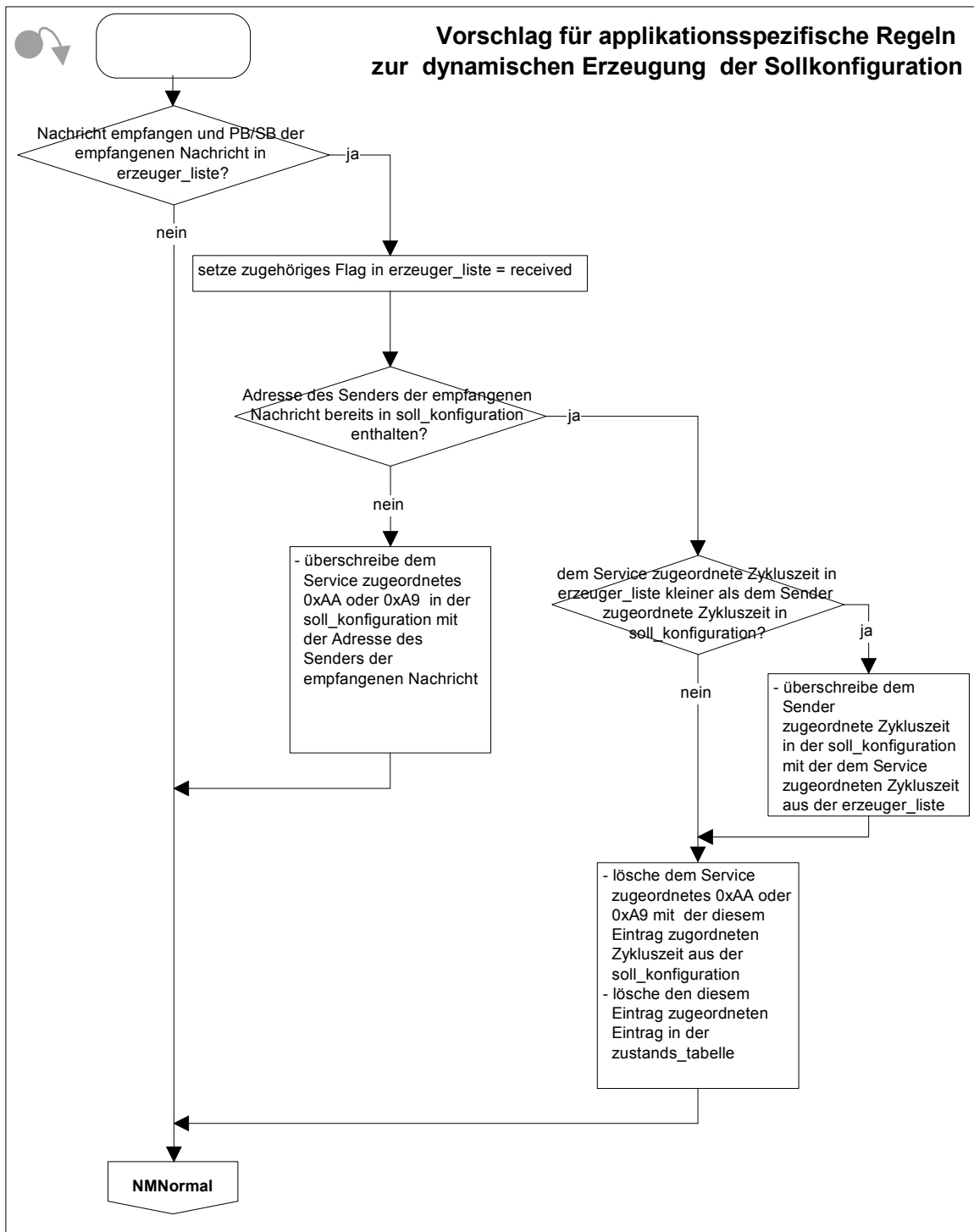


Abbildung 9: Vorschlag zur dynamischen Erzeugung der Sollkonfiguration - Ablaufdiagramm für applikationsspezifische Regeln

Nach dem Senden einer eigenen Botschaft und dem Empfang einer Nachricht mit PB = PB1 und SB = SB1 Adresse 0xXX ergeben sich erzeuger_liste, soll_konfiguration, zustands_tabelle und netz_status wie folgt (Änderungen grau hinterlegt):

PB 1	SB 1	Zeit 1	received	...	PB n	SB n	Zeit n	not received	erzeuger_liste
0xXX	...	0xAA	eigene Adresse						soll_konfiguration
Zeit 1	...	Zeit n	eigene Zeit						Zykluszeiten
OK	...	OK	OK						zustands_tabelle
OK									netz_status
gesetzt									start_flag

Nach Empfang weiterer Nachrichten die den geforderten Services entsprechen, ergeben sich erzeuger_liste, soll_konfiguration, zustands_tabelle und netz_status wie folgt, wenn der letzte benötigte Service empfangen wurde (Änderungen nach Empfang der letzten Botschaft dunkelgrau hinterlegt):

PB 1	SB 1	Zeit 1	received	...	PB n	SB n	Zeit n	received	erzeuger_liste
0xXX	...	0xZZ	eigene Adresse						soll_konfiguration
Zeit 1	...	Zeit n	eigene Zeit						Zykluszeiten
OK	...	OK	OK						zustands_tabelle
OK									netz_status
nicht gesetzt									start_flag

Das Netzwerkmanagement kennt jetzt die von ihm zu überwachende Sollkonfiguration. Eine Änderung der Netzkonfiguration (z.B. Austausch des Knotens, der einen geforderten Service unterstützt, gegen einen Knoten mit einer anderen Adresse) führt über NMRreset zu einer Neuerzeugung der Sollkonfiguration.

Der hier vorgestellte Lösungsansatz zur dynamischen Erzeugung einer Sollkonfiguration ist so aufgebaut, daß eine entsprechende Implementierung ohne weiteres auch für einen Netzwerkmanagement Knoten mit statischer Sollkonfiguration eingesetzt werden kann.

10 Implementierungsvorschläge

Die nachfolgenden Implementierungsbeispiele zeigen jeweils den Speicherbedarf pro zu überwachenden Knoten sowie den Speicherbedarf, der grundsätzlich vorhanden sein muß.

Jeder Implementierungsvorschlag wird anhand einer Abbildung veranschaulicht. Diese Abbildung bezieht sich auf ein Netz mit drei Knoten in dem ein Knoten 0x0F die Knoten 0x03 und 0x07 und sich selbst überwacht.

Hinweis:

Für die Darstellung der Zykluszeiten wurden im folgenden jeweils 8 Bit zugrunde gelegt. Bei Wahl einer geeigneten Auflösung kann die Zykluszeit jedoch unter Umständen mit einer geringeren Anzahl an Bits dargestellt werden. Der Ressourcenbedarf würde damit insbesondere bei Anwendung spezifischer Zykluszeiten drastisch reduziert!

10.1 Vorschlag einer Implementierung mit statischer Sollkonfiguration und einer Default Zykluszeit

Definition der Daten der Schnittstellen:

Sollkonfiguration	pro zu überwachenden Knoten wird ein Byte für die jeweilige eBUS Adresse benötigt
Zustandstabelle	für die eBUS Adresse des eigenen Knotens wird kein Byte benötigt, da die eigene Adresse bekannt ist pro zu überwachenden Knoten ein Bit: der Zustand des jeweiligen Knotens wird durch das Bit repräsentiert (1 = OK / 0 = NOK)
Zykluszeit	für den Zustand des eigenen Knotens wird ein Bit benötigt
Startflag	ein Byte
Netzstatus	ein Bit (1 = ON / 0 = OFF) ein Bit (1 = OK / 0 = NOK)

Definition der internen Variablen der Netzwerkmanagement Implementierung

aktuelle Zeitwerte (zur Überwachung der Zykluszeiten)	pro zu überwachenden Knoten ein Byte (wie Zykluszeit)
	für den eigenen Knoten ein Byte

Damit werden für jeden zu überwachenden Knoten 17 Bit benötigt:

Sollkonfiguration	8 Bit
Zustandstabelle	1 Bit
aktuelle Zeitwerte	8 Bit
	Σ17 Bit

Zur Überwachung des eigenen Knotens und für Startflag und Netzstatus werden zusätzlich 11 Bit benötigt:

Zustandstabelle	1 Bit
aktuelle Zeitwerte	8 Bit
Startflag	1 Bit
Netzstatus	1 Bit
	Σ11 Bit

Die Default Zykluszeit wird durch 8 Bit dargestellt.

Summe: 19 Bit + (Anzahl zu überwachender Knoten · 17 Bit)

Die nachfolgende Abbildung zeigt eine mögliche Realisierung des Implementierungsvorschlags für einen Master 0x0F.

Master 0x0F überwacht Master 0x03 und Master 0x07 und sich selbst.

Die Default Zykluszeit ist 10min.

Ressourcenbedarf ohne Quellcode: 53 Bit

Hinweis: bei Darstellung der Zykluszeit durch 4 Bit liegt der Ressourcenbedarf für das Beispiel bei 37 Bit

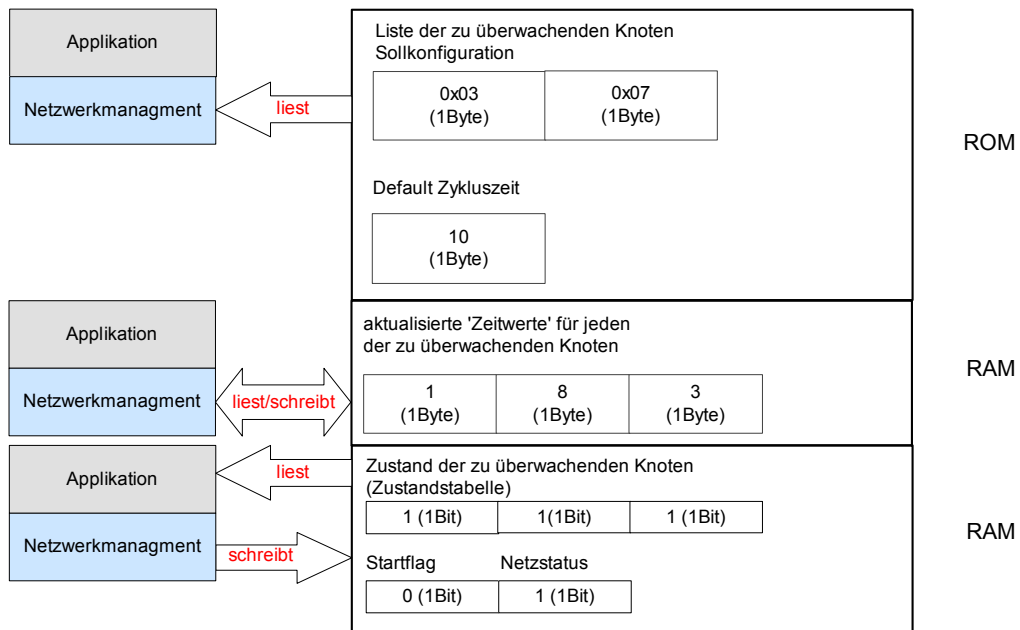


Abbildung 10: statische Sollkonfiguration mit Default Zykluszeit - Realisierung eines Implementierungsvorschlags

10.2 Vorschlag einer Implementierung mit statischer Sollkonfiguration und spezifischen Zykluszeiten

Definition der Daten der Schnittstellen:

Sollkonfiguration	pro zu überwachenden Knoten wird ein Byte für die jeweilige eBUS Adresse benötigt
Zustandstabelle	für die eBUS Adresse des eigenen Knotens wird kein Byte benötigt, da die eigene Adresse bekannt ist pro zu überwachenden Knoten ein Bit: der Zustand des jeweiligen Knotens wird durch das Bit repräsentiert (1 = OK / 0 = NOK)
Zykluszeit	für den Zustand des eigenen Knotens wird ein Bit benötigt pro zu überwachenden Knoten ein Byte
Startflag	für den eigenen Knoten ein Byte
Netzstatus	ein Bit (1 = ON / 0 = OFF) ein Bit (1 = OK / 0 = NOK)

Definition der internen Variablen der Netzwerkmanagement Implementierung

aktuelle Zeitwerte (zur Überwachung der Zykluszeiten)	pro zu überwachenden Knoten ein Byte (wie Zykluszeit)
	für den eigenen Knoten ein Byte

Damit werden für jeden zu überwachenden Knoten 25 Bit benötigt:

Sollkonfiguration	8 Bit
Zustandstabelle	1 Bit
Zykluszeit	8 Bit
aktuelle Zeitwerte	8 Bit
Σ 25 Bit	

Zur Überwachung des eigenen Knotens und für Startflag und Netzstatus werden zusätzlich 11 Bit benötigt:

Zustandstabelle	1 Bit
Zykluszeit	8 Bit
aktuelle Zeitwerte	8 Bit
Startflag	1 Bit
Netzstatus	1 Bit
Σ19 Bit	

Summe: 19 Bit + (Anzahl zu überwachender Knoten · 25 Bit)

Die nachfolgende Abbildung zeigt die mögliche Realisierung des Implementierungsvorschlags für einen Master 0x0F.

Master 0x0F überwacht Master 0x03 und Master 0x07 und sich selbst.

Von Master 0x03 wird spätestens nach 1s eine Botschaft erwartet, von Master 0x07 wird spätestens nach 10min eine Botschaft erwartet. Master 0x0F soll spätestens nach 5min eine Botschaft absetzen.

Ressourcenbedarf ohne Quellcode: 69 Bit

Hinweis: bei Darstellung der Zykluszeit durch 4 Bit liegt der Ressourcenbedarf für das Beispiel bei 45 Bit

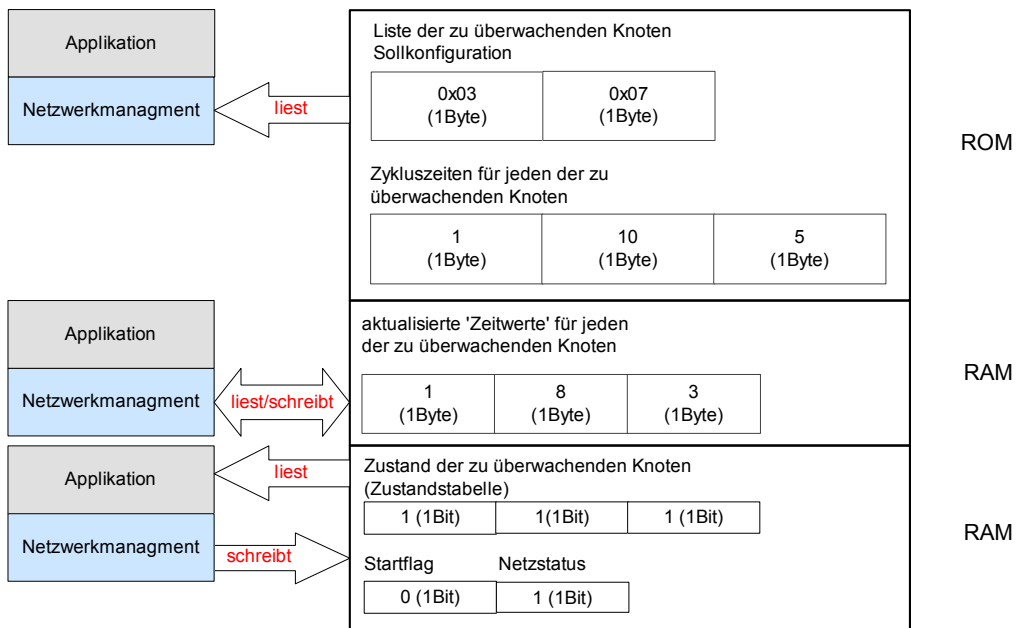


Abbildung 11: statische Sollkonfiguration mit spezifischen Zykluszeiten - Realisierung eines Implementierungsvorschlags

10.3 Vorschlag einer Implementierung mit dynamischer Erzeugung der Sollkonfiguration und einer Default Zykluszeit

Definition der Daten der Schnittstellen:

Erzeuger_Liste	pro benötigtem Service werden zwei Byte für Primär- und Sekundärbefehl und ein Bit für das Flag 'Received/not received' benötigt
Sollkonfiguration	pro zu überwachenden Knoten wird ein Byte für die jeweilige eBUS Adresse benötigt
Zustandstabelle	für die eBUS Adresse des eigenen Knotens wird kein Byte benötigt, da die eigene Adresse bekannt ist pro zu überwachenden Knoten ein Bit: der Zustand des jeweiligen Knotens wird durch das Bit repräsentiert (1 = OK / 0 = NOK)
Zykluszeit	für den Zustand des eigenen Knotens wird ein Bit benötigt
Startflag	ein Byte
Netzstatus	ein Bit (1 = ON / 0 = OFF)
	ein Bit (1 = OK / 0 = NOK)

Definition der internen Variablen der Netzwerkmanagement Implementierung

aktuelle Zeitwerte (zur Überwachung der Zykluszeiten)	pro zu überwachenden Knoten ein Byte (wie Zykluszeit)
	für den eigenen Knoten ein Byte

Damit werden für jeden zu überwachenden Knoten 17 Bit benötigt:

Sollkonfiguration	8 Bit
Zustandstabelle	1 Bit
aktuelle Zeitwerte	8 Bit
	Σ17 Bit

Für jeden geforderten Service werden in der Erzeugerliste 17 Bit benötigt:

Primärbefehl	8 Bit
Sekundärbefehl	8 Bit
Flag	1 Bit
	Σ 17 Bit


Zur Überwachung des eigenen Knotens und für Startflag und Netzstatus werden zusätzlich 11 Bit benötigt:

Zustandstabelle	1 Bit
aktuelle Zeitwerte	8 Bit
Startflag	1 Bit
Netzstatus	1 Bit
	Σ11 Bit

Die Default Zykluszeit wird durch 8 Bit dargestellt.

Summe: 19 Bit + (Anzahl zu überwachender Knoten · 17 Bit) + (Anzahl geforderter Services · 17 Bit)⁵

⁵ 'Worst Case' Betrachtung: jeder der geforderten Services wird von einem anderen Knoten geliefert ⇒ Anzahl geforderter Services = Anzahl zu überwachender Knoten (Maximaler Bedarf an Speicherplatz lässt sich berechnen)

<p>© eBUS Interest Group</p>  <p>www.eBUS.de</p>	<p>eBUS Spezifikation</p> <p>Netzwerkmanagement</p>	<p>Version 1.0.2</p>	<p>Ausgabe</p> <p>03 / 07</p>	<p>Seite</p> <p>23</p>
--	---	----------------------	-------------------------------	------------------------

Die nachfolgende Abbildung zeigt wiederum die mögliche Realisierung des Implementierungsvorschlags für einen Master 0x0F.
 Master 0x0F benötigt Service 1 und Service 2 für seine Funktionalität. Während des Betriebs erhält er die Daten des Service 1 von Master 0x03 und die Daten des Service 2 von Master 0x07.
 Damit überwacht Master 0x0F den Master 0x03, den Master 0x07 und sich selbst.
 Die Default Zykluszeit ist 10min.

Ressourcenbedarf ohne Quellcode: 87 Bit

Hinweis: bei Darstellung der Zykluszeit durch 4 Bit liegt der Ressourcenbedarf für das Beispiel bei Bit

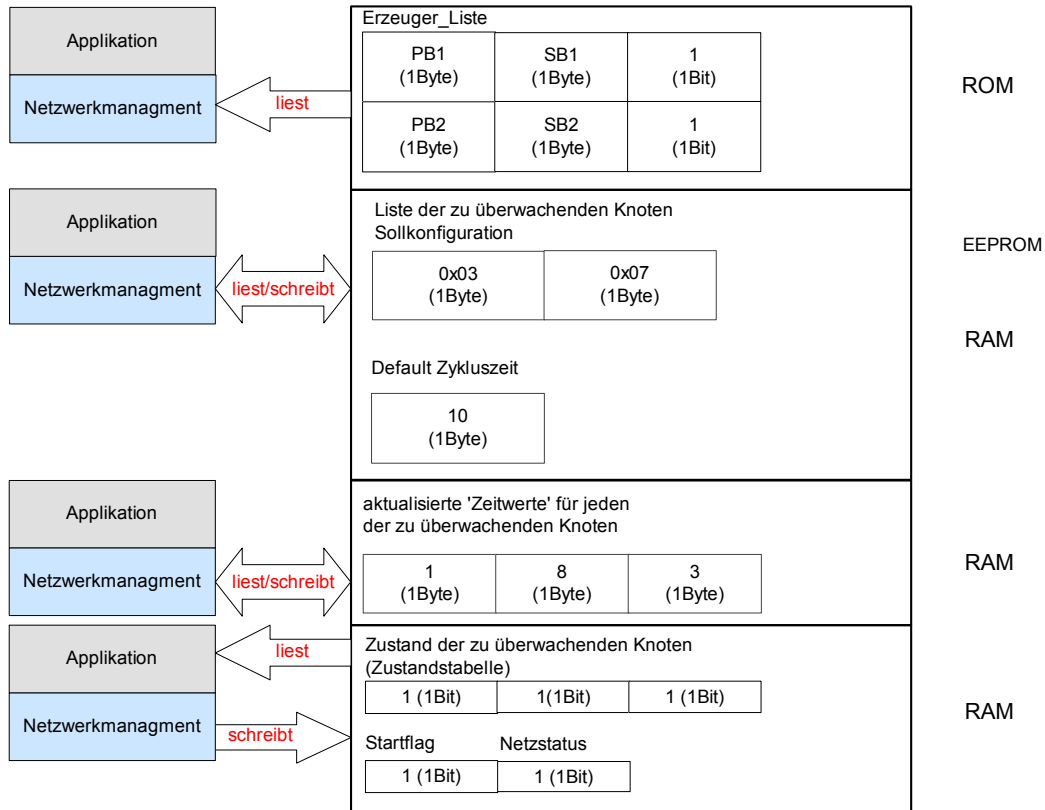


Abbildung 12: dynamische Erzeugung der Sollkonfiguration mit Default Zykluszeit - Realisierung eines Implementierungsvorschlags

10.4 Vorschlag einer Implementierung mit dynamischer Erzeugung der Sollkonfiguration und spezifischen Zykluszeiten

Definition der Daten der Schnittstellen:

Erzeuger_Liste	pro benötigtem Service werden zwei Byte für Primär- und Sekundärbefehl, ein Bit für das Flag 'Received/not received' und ein Byte für die spezifische Zykluszeit benötigt
Sollkonfiguration	pro zu überwachenden Knoten wird ein Byte für die jeweilige eBUS Adresse benötigt
Zustandstabelle	für die eBUS Adresse des eigenen Knotens wird kein Byte benötigt, da die eigene Adresse bekannt ist pro zu überwachenden Knoten ein Bit: der Zustand des jeweiligen Knotens wird durch das Bit repräsentiert (1 = OK / 0 = NOK)
Zykluszeit	für den Zustand des eigenen Knotens wird ein Bit benötigt pro zu überwachenden Knoten ein Byte
Startflag	für den eigenen Knoten ein Byte
Netzstatus	ein Bit (1 = ON / 0 = OFF) ein Bit (1 = OK / 0 = NOK)

Definition der internen Variablen der Netzwerkmanagement Implementierung

aktuelle Zeitwerte (zur Überwachung der Zykluszeiten)	pro zu überwachenden Knoten ein Byte (wie Zykluszeit)
	für den eigenen Knoten ein Byte

Damit werden für jeden zu überwachenden Knoten 25 Bit benötigt:

Sollkonfiguration	8 Bit
Zustandstabelle	1 Bit
Zykluszeit	8 Bit
aktuelle Zeitwerte	8 Bit
Σ	25 Bit

Für jeden geforderten Service werden in der Erzeugerliste 25 Bit benötigt:

Primärbefehl	8 Bit
Sekundärbefehl	8 Bit
Zykluszeit	8 Bit
Flag	1 Bit
Σ	25 Bit

Zur Überwachung des eigenen Knotens und für Startflag und Netzstatus werden zusätzlich 19 Bit benötigt:

Zustandstabelle	1 Bit
Zykluszeit	8 Bit
aktuelle Zeitwerte	8 Bit
Startflag	1 Bit
Netzstatus	1 Bit
Σ	19 Bit

Summe: 19 Bit + (Anzahl zu überwachender Knoten · 25 Bit) + (Anzahl geforderter Services · 25 Bit)⁶

Die nachfolgende Abbildung zeigt die mögliche Realisierung des Implementierungsvorschlags für einen Master 0x0F.

Master 0x0F benötigt Service 1 nach spätestens 1min und Service 2 nach spätestens 10min für seine Funktionalität. Während des Betriebs erhält er die Daten des Service 1 von Master 0x03 und die Daten des Service 2 von Master 0x07.

Damit überwacht Master 0x0F den Master 0x03, den Master 0x07 und sich selbst.

Ressourcenbedarf ohne Quellcode: 119 Bit

Hinweis: bei Darstellung der Zykluszeit durch 4 Bit liegt der Ressourcenbedarf für das Beispiel bei 87 Bit

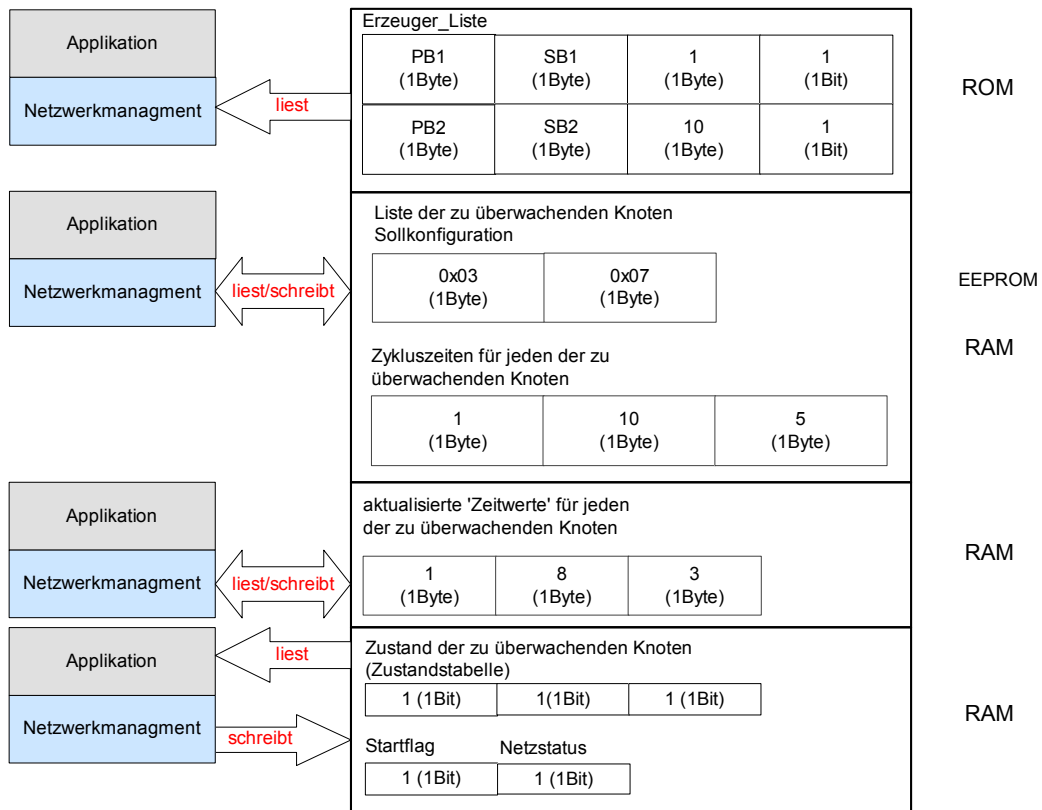


Abbildung 13: dynamische Erzeugung der Sollkonfiguration mit spezifischen Zykluszeiten - Realisierung eines Implementierungsvorschlags

⁶ 'Worst Case' Betrachtung: jeder der geforderten Services wird von einem anderen Knoten geliefert ⇒ Anzahl geforderter Services = Anzahl zu überwachender Knoten (Maximaler Bedarf an Speicherplatz lässt sich berechnen)

11 Abbildungsverzeichnis

ABBILDUNG 1: EINORDNUNG DES NETZWERKMANAGEMENTS IN DEN eBUS KNOTEN	5
ABBILDUNG 2: ZUSTANDSDIAGRAMM DES NETZWERKMANAGEMENTS	6
ABBILDUNG 3: ARCHITEKTUR MODELL	10
ABBILDUNG 4: ABLAUFDIAGRAMM NMINIT	12
ABBILDUNG 5: ABLAUFDIAGRAMM NMRESET	12
ABBILDUNG 6: ABLAUFDIAGRAMM NMNORMAL	13
ABBILDUNG 7: VORSCHLAG ZUR DYNAMISCHEN ERZEUGUNG DER SOLLKONFIGURATION - ABLAUFDIAGRAMM FÜR DEN ZUSTAND NMRESET	14
ABBILDUNG 8: VORSCHLAG ZUR DYNAMISCHEN ERZEUGUNG DER SOLLKONFIGURATION - ABLAUFDIAGRAMM FÜR DEN ZUSTAND NMNORMAL	16
ABBILDUNG 9: VORSCHLAG ZUR DYNAMISCHEN ERZEUGUNG DER SOLLKONFIGURATION - ABLAUFDIAGRAMM FÜR APPLIKATIONSSPEZIFISCHE REGELN	17
ABBILDUNG 10: STATISCHE SOLLKONFIGURATION MIT DEFAULT ZYKLUSZEIT - REALISIERUNG EINES IMPLEMENTIERUNGSVORSCHLAGS	20
ABBILDUNG 11: STATISCHE SOLLKONFIGURATION MIT SPEZIFISCHEN ZYKLUSZEITEN - REALISIERUNG EINES IMPLEMENTIERUNGSVORSCHLAGS	21
ABBILDUNG 12: DYNAMISCHE ERZEUGUNG DER SOLLKONFIGURATION MIT DEFAULT ZYKLUSZEIT - REALISIERUNG EINES IMPLEMENTIERUNGSVORSCHLAGS	24
ABBILDUNG 13: DYNAMISCHE ERZEUGUNG DER SOLLKONFIGURATION MIT SPEZIFISCHEN ZYKLUSZEITEN - REALISIERUNG EINES IMPLEMENTIERUNGSVORSCHLAGS	26