Norm	RCN-214 **DCC-Protokoll** **Konfigurationsbefehle**	RailCommunity
Ausgabe 02.12.2018		RailCommunity – Verband der Hersteller Digitaler Modellbahnprodukte e.V.

Inhalt

1 Allgemeines

1.1 Zweck der Norm

Diese Norm beschreibt die Pakete des DCC-Protokolls zur Konfiguration eines Decoders. Alle Pakete folgen der in [RCN211] beschriebenen Paketstruktur.

Die Konfiguration von Decoder erfolgt über das setzen von Variablen, die jeweils ein Byte umfassen. Es gibt aber auch Befehle, die nur ein Bit eines Bytes betreffen. Der Adressbereich für die Konfigurationsvariablen umfasst 1024 Byte. Er kann durch indirekte Zugriffe vergrößert werden. Die Konfigurationsvariablen sind in [RCN225] definiert.

Befehle zur Konfiguration können sowohl im normalen Betrieb als auch im Programmiermodus ausgeführt werden. Der Wechsel zwischen diesen Betriebsarten ist in [RCN216] beschrieben.

Diese Norm entspricht den in [S921] definierten Befehlen zur Konfiguration von Decodern und den Abschnitten D bis F von [S923].

1.2 Anforderungen

Diese Norm betrifft nur Geräte, die Konfigurationen über DCC-Befehle unterstützen, sei es durch senden der entsprechenden Pakete (Zentralen) oder empfangen und verarbeiten (Decoder).

Eine Zentrale, die die Konfiguration von Decodern über DCC-Befehle unterstützt, muss mindestens die Lange Form des Konfigurationsvariablen Zugriffsbefehls entweder im Betriebsmodus mit allen Adressierungen, die für Betriebsbefehle unterstützt werden, oder im Programmiermodus unterstützen. Es müssen alle 1024 CVs erreichbar sein.

Zentralen, die den Zugriffsbefehl im Registermodus unterstützen, müssen alle acht Register adressieren können und bei jedem Zugriff das Register 6 setzen. Die Sonderform "Paged Mode" muss nicht unterstützt werden.

Zentralen, die mehr als das Minimum unterstützen, dürfen als DCC Protokoll zur Konfiguration eines Decoders nur die hier definierten Befehle senden. Ausnahmen sind von der RailCommunity freigegebene Pakete zur Unterstützung von weiteren Entwicklungen.

Ein Fahrzeugdecoder, der die Konfigurationen über DCC-Befehle unterstützt, muss mindestens die lange Form des Konfigurationsvariablen Zugriffsbefehls entweder im Betriebsmodus sowohl mit der 7 Bit als auch mit der 14 Bit Adressierung oder im Programmiermodus unterstützen.

Ein Zubehördecoder, der die Konfigurationen über DCC-Befehle unterstützt, muss mindestens die lange Form des Konfigurationsvariablen Zugriffsbefehls entweder im Betriebsmodus mit seiner einfachen oder erweiterten 11-Bit-Adressierung oder im Programmiermodus unterstützen.

Decoder, die den Zugriffsbefehl im Registermodus unterstützen, müssen alle acht Register adressieren können. Wird in einen Betriebsmodus geschaltet, muss intern das Register 6 auf den Wert 1 gesetzt werden. Die Sonderformen "Paged Mode" und "Decoder Factory Reset" müssen aber nicht unterstützt werden.

Decoder, die mehr als das Minimum unterstützen, müssen alle Befehle, die sie unterstützen und die hier beschrieben sind, genau so interpretieren, wie hier festgelegt. Decoder müssen alle Befehle, die sie **nicht** unterstützen, ignorieren und diese Befehle dürfen nicht zu einer Fehlfunktion des Decoders führen.

Es kann immer nur ein Befehl in einem Paket zur Konfiguration sein.

Die Forderung der NMRA, beim Schreiben von CV #1 im Decoder CV #29 Bit 5 (Verwendung der 4-stellige Adresse) und CV #19 (Mehrfachtraktionsadresse) zu löschen, also auf `0` zu setzen, wird nicht aufrechterhalten. Ein derartiges Verhalten widerspricht aber nicht dieser Norm.

Die Normen [RCN210] und [RCN211] sowie bei Unterstützung des Programmiermodus [RCN216] sind einzuhalten.

1.3 Formatdefinitionen

Innerhalb dieser Norm gelten folgende Festlegungen:

- Die Bits eines Bytes werden von 0 bis 7 gezählt. Bit 0 ist das niederwertigste Bit und steht ganz rechts und Bit 7 ist das höchstwertigste und steht ganz links.
- Zwischen den Bits 4 und 3 wird für eine bessere Lesbarkeit ein Strich eingefügt: `7654-3210`.
- Abgesehen von der Beschreibung des Aufbaus eines DCC-Paketes werden die Nullen zwischen den Bytes eines DCC-Paketes nicht dargestellt.
- Es wird häufig das ganze Byte dargestellt; Bits ohne Bedeutung im aktuellen Kontext werden mit '**`x`**' gekennzeichnet.
- Folgende Zeichen werden zur Kennzeichnung der Bedeutung eines Bits verwendet:

  `0` Bitwert 0
  `1` Bitwert 1
  `A` Adressbit
  `B` Bitadressbit beim Zugriff auf Konfigurationsvariable
  `D` Datenbit
  `K` Unterbefehlsbit
  `V` Konfigurations**v**ariablen- und Registeradressbits
  `x` Platzhalter für ein Bit, dessen Wert von der Art des Pakets und des Befehls abhängt und an der Stelle nicht näher betrachtet wird.

2 Konfigurationsvariablen Zugriffsbefehl - Lange Form

Die lange Form erlaubt die direkte Manipulation aller CVs. Dieser Befehl ist drei Byte lang und hat die Formate:

1110-KKVV VVVV-VVVV DDDD-DDDD	Im Betriebsmodus für Zugriffe auf Bytes
1110-10VV VVVV-VVVV 111K-DBBB	Im Betriebsmodus für Zugriffe auf einzelne Bits
0111-KKVV VVVV-VVVV DDDD-DDDD	Im Programmiermodus für Zugriffe auf Bytes
0111-10VV VVVV-VVVV 111K-DBBB	Im Programmiermodus für Zugriffe auf einzelne Bits

2.1 Adressierung

Zur Ausführung dieses Befehls kann der Decoder folgender Maßen adressiert werden:

<<keine Adresse>>	Ohne Adresse im Programmiermodus.
0AAA-AAAA	Fahrzeugdecoder mit 7 Bit Adressen
11AA-AAAA AAAA-AAAA	Fahrzeugdecoder mit 14 Bit Adressen
10AA-AAAA 1AAA-1AA0	Einfache Zubehördecoder mit 11 Bit Adressen und
10AA-AAAA 0AAA-0AA1	Erweiterte Zubehördecoder mit 11-Bit Adressen

Anmerkungen:

- Die 7 Bit Adresse eines Fahrzeugdecoders muss eine Basisadresse sein, auf Konfigurationsbefehle an die Mehrfachtraktionsadresse darf nicht reagiert werden.
- Zubehördecoder können zur Konfiguration über die und nur die Adressen angesprochen werden, auf denen sie auf Schaltbefehle reagieren. Dabei ist es zulässig alle Ausgänge über eine Adresse zu konfigurieren.
- Das von der NMRA bei den einfachen Zubehördecodern definierte Flagbit **C** in Bit 3 des zweiten Adressbytes wird nicht übernommen:
  - Programmiergeräte müssen Bit 3 des zweiten Adressbytes immer auf 1 setzen.
  - Decoder, die vier aufeinander folgende Adressen unterstützen, dürfen gemäß den Vorgaben der NMRA auf eine 0 im Bit 3 des zweiten Adressbytes reagieren. Dieses dient aber ausschließlich der Rückwärtskompatibilität.
- Bei den Erweiterte Zubehördecodern stellt Verwendung einer **0** in Bit 3 des zweiten Adressbytes sicher, dass dieses Paket nicht mit veralteten Zubehördecoderprogrammier-paketen verwechselt werden kann (siehe Anhang B).
- Bei den Befehlen für Zubehördecoder kann eine Unterscheidung zwischen Betriebsbe-fehlen und Konfigurationsvariablen Zugriffsbefehlen nur über die Länge des Paketes erfolgen.

2.2 Befehlsaufbau

Die gewünschte Konfigurationsvariable wird mit der 10 Bit Adresse (**xxxx-xxVV VVVV-VVVV**) ausgewählt, wobei die beiden Adressbits **VV** im ersten Befehlsbyte die

höchstwertigsten Bits der Konfigurationsvariablen-Adresse sind. Die adressierte Konfigurationsvariable ist die gegebene 10 Bit Adresse plus 1. Z.B. um CV #1 zu adressieren ist die 10 Bit Adresse **VV VVVV-VVVV** = **00 0000-0000** auszuwählen. Die programmierenden Geräte müssen den gesamten möglichen Adressbereich unterstützen.

Die für den Befehlstyp (**xxxx-KKxx**) im ersten Befehlsbyte festgelegten Werte sind:

KK = **00** – reserviert
KK = **01** – Byte Überprüfen
KK = **11** – Byte Schreiben
KK = **10** – Bit Manipulation

Wenn einer der drei Befehlstypen Byte Überprüfen, Byte Schreiben oder Bit Manipulation unterstützt wird, müssen all drei Befehlstypen zur Verfügung gestellt werden.

KK = 01 Byte Überprüfen
Der Inhalt der durch die 10 Bit Adresse gekennzeichneten Konfigurationsvariable wird mit dem Datenbyte **DDDD-DDDD** verglichen.

KK = 11 Byte Schreiben
Der Inhalt der durch die 10 Bit Adresse gekennzeichneten Konfigurationsvariable wird durch das Datenbyte **DDDD-DDDD** ersetzt.

KK = 10 Bit Manipulation
Die Bit-Manipulationsbefehle verwenden ein spezielles Format für das dritte Befehlsbyte **111K-DBBB**, wobei **BBB** die Bitposition innerhalb der CV darstellt, **D** enthält den Wert des Bits, das geprüft oder geschrieben werden soll, und **K** bestimmt, ob die Aktion ein Bit Prüfen oder Bit Schreiben Zugriff ist.

K = **1** – Bit Schreiben
K = **0** – Bit Überprüfen

Die **Bit Überprüfen** und **Bit Schreiben** Befehle arbeiten in ähnlicher Art wie die **Byte Überprüfen** und **Byte Schreiben** Befehle (aber bearbeiten ein einzelnes Bit).

2.3 Befehlssequenz im Programmiermodus

Die Befehlssequenz zum Programmieren im Programmiermodus sind in [RCN216] im Abschnitt "Befehlsfolgen" festgelegt.

2.4 Befehlssequenz im Betriebsmodus

Beim Schreiben einer Konfigurationsvariablen sind zwei identische Pakete erforderlich, bevor der Decoder eine Konfigurationsvariable ersetzen darf. Diese beiden Pakete müssen auf dem Gleis nicht direkt aufeinander folgen. Allerdings wird jedes andere Paket an denselben Decoder die Schreiboperation ungültig machen. Dieses schließt Broadcast Pakete ein. Bestätigt wird nur der erfolgreiche Empfang des zweiten Paketes.

3 Konfigurationsvariablen Zugriffsbefehl - Kurze Form

Die kurze Form erlaubt die direkte Manipulation weniger ausgewählter CVs ausschließlich im Betriebsmodus. Dieser Befehl ist zwei oder drei Byte lang und hat das Format **1111-KKKK DDDD-DDDD** bzw. **1111-KKKK DDDD-DDDD DDDD-DDDD**.

3.1 Adressierung

Zur Ausführung dieses Befehls kann der Decoder folgender Maßen adressiert werden:

0AAA-AAAA	Fahrzeugdecoder mit 7 Bit Adressen
11AA-AAAA AAAA-AAAA	Fahrzeugdecoder mit 14 Bit Adressen

Anmerkungen:

- Die 7 Bit Adresse eines Fahrzeugdecoders muss eine Basisadresse sein, auf Konfigurationsbefehle an die Mehrfachtraktionsadresse darf nicht reagiert werden.
- Diesen Befehl gibt es nur im Betriebsmodus.

3.2 Befehlsaufbau

Die 8 Bit Daten im zweiten und ggf. dritten Befehlsbyte **DDDD-DDDD** werden in der bzw. den Konfigurationsvariablen abgelegt, die durch die Bits 0-3 im ersten Befehlsbyte **KKKK** entsprechend der folgenden Tabelle gekennzeichnet ist bzw. sind. Bei zwei Konfigurationsvariablen befinden sich die Daten für die CV mit der kleineren Nummer im zweiten Befehlsbyte, die Daten für die CV mit der größeren Nummer im dritten Befehlsbyte.

KKKK = **0000** – Darf nicht verwendet werden

KKKK = **0010** – Beschleunigungswert in einer Mehrfachtraktion (CV #23)
Nur ein einzelnes Paket ist erforderlich um diese Konfigurationsvariable zu ändern.

KKKK = **0011** – Verzögerungswert in einer Mehrfachtraktion (CV #24)
Nur ein einzelnes Paket ist erforderlich um diese Konfigurationsvariable zu ändern.

KKKK = **0100** – Schreiben von CV #17 und CV #18 gleichzeitig sowie Setzen des Bit 5 in CV #29 (Umschaltung auf lange Adressen)
Wie bei der langen Form sind zwei identische Pakete erforderlich, bevor der Decoder die Konfigurationsvariablen ersetzen darf.

KKKK = **0101** – Schreiben von CV #31 und CV #32 gleichzeitig (Index CVs)
Wie bei der langen Form sind zwei identische Pakete erforderlich, bevor der Decoder die Konfigurationsvariablen ersetzen darf.

KKKK = **1001** – Reserviert, siehe Anhang C

Die restlichen Werte von **KKKK** sind reserviert und werden von der RailCommunity bestimmt, wenn eine Notwendigkeit festgestellt wird.

Zum Schreiben der CV 17 und 18 ist bevorzugt die kurze Befehlsform zu nutzen, bei der es nicht zu einer Änderung nur einer CV kommen kann.

4 Konfigurationsvariablen Zugriffsbefehl XPOM

Dieser Befehl dient dazu im Betriebsmodus den Zugriff auf Konfigurationsvariablen zu beschleunigen und den ansonsten nur über die CVs #31 und #32 erreichbaren erweiterten Bereich linear zu adressieren. Zudem können bis zu 4 Bytes mit einem Befehl geschrieben werden. Wenn der Decoder RailCom unterstützt und diese in dem Decoder freigegeben wurde, ist auch das gleichzeitige Lesen von 4 CVs möglich.

Dieser Befehl ist vier bis acht Byte lang und hat das allgemeine Format:

1110-KKSS VVVV-VVVV VVVV-VVVV VVVV-VVVV {DDDD-DDDD {DDDD-DDDD {DDDD-DDDD {DDDD-DDDD}}}}

Die Adressierung erfolgt wie beim Befehl „Zugriffsbefehl - Lange Form“ im Betriebsmodus. Der XPOM Befehl kann von diesem nur durch seine Länge unterschieden werden. Dabei definiert KK wie beim „Zugriffsbefehl - Lange Form“ die Zugriffsart:

- **KK** = **00** – reserviert
- **KK** = **01** – Bytes Lesen
- **KK** = **11** – Byte(s) Schreiben
- **KK** = **10** – Bit schreiben.

SS ist eine Sequenznummer, die für die Rückmeldung über RailCom beim Blockweisen Schreiben und Lesen vieler CVs erforderlich ist. Ohne RailCom sind dies Bits ohne Bedeutung. Daher wird die Verwendung der Sequenznummer in der [RCN217] definiert. Bei Zentralen ohne RailCom sollten diese Bits auf **SS** = **00** gesetzt werden.

VVVV-VVVV VVVV-VVVV VVVV-VVVV ist die 24 Bit lange Adresse der ersten zu schreibenden oder zu lesenden CV. Damit können bis zu 2^{24} = 16 777 216 CVs direkt adressiert werden. Es werden die höherwertigen Adressbits im ersten Byte übertragen.
Im Vergleich zum Zugriff über die CVs #31 und #32 ergibt sich damit, dass das erste **VVVV-VVVV** Byte dem Inhalt der CV #31 entspricht, das zweite **VVVV-VVVV** Byte dem Inhalt von CV #32 und das dritte **VVVV-VVVV** Byte dem zweiten Byte des „Zugriffsbefehl - Lange Form“ entspricht.

KK = **01 Bytes Lesen**
Es werden keine Datenbytes **DDDD-DDDD** übertragen. Der Inhalt der ausgewählten CV und der drei darauf folgenden CVs wird per RailCom übertragen.

KK = **11 Byte(s) Schreiben**
Die bis zu vier **DDDD-DDDD** Bytes enthalten die Daten, die in die ausgewählte CV und ggf. den bis zu drei darauf folgenden CVs geschrieben werden. Es werden immer die Werte von vier CVs per RailCom zurückgemeldet.

KK = **10 Bit Schreiben**
Das fünfte Befehlsbyte hat wie beim „Zugriffsbefehl - Lange Form“ das Format **1111-DBBB**, wobei **BBB** die Bitposition innerhalb der CV darstellt, **D** enthält den Wert des zu schreiben Bits. Es werden immer die Werte von vier CVs per RailCom zurückgemeldet.

5 Zugriffsbefehl im Registermodus

Der Registermodus ist der älteste Mechanismus zur Manipulation von acht Konfigurationsvariablen, die als Register bezeichnet werden. Dieser Befehlstyp ist als veraltet zu betrachten und sollte nicht verwendet werden. Er wird hier nur aus Gründen der Rückwärtskompatibilität dokumentiert.

Dieser Befehl ist zwei Byte lang und hat das Format **`0111-KVVV DDDD-DDDD`**. Er ist nur im Programmiermodus ohne Adresse gültig, woraus sich ein drei Byte Paket ergibt. Dieser Befehl kann vom Decoder nur aufgrund seiner Länge vom Konfigurationsvariablen Zugriffsbefehl - Lange Form im Programmiermodus unterschieden werden.

5.1 Befehlsaufbau

Das gewünschte Register wird mit der 3 Bit Adresse **`VVV`** ausgewählt. Dabei entspricht der Wert **`000`** der Adresse von Register 1 und der Wert **`111`** der Adresse für Register 8.

Die Register 1 bis 8 entsprechen aber nicht (immer) den CVs 1 bis 8. Register 5 entspricht CV #29 und Register 6 ist für den als "Paged Mode" bezeichneten indizierten Zugriff reserviert. Die Register 1 bis 4 entsprechen nur dann den entsprechenden CVs, wenn das Register 6 den Wert 1 hat. Die Register 7 und 8 entsprechen immer den CVs 7 und 8.

Die für den Befehlstyp **`K`** im ersten Befehlsbyte festgelegten Werte sind:

- **`K`** = **`0`** – Byte Überprüfen
- **`K`** = **`1`** – Byte Schreiben

`K` = **`0`** **Byte Überprüfen**
Der Inhalt des durch die 3 Bit Adresse gekennzeichneten Registers wird mit dem Datenbyte **`DDDD-DDDD`** im zweiten Befehlsbyte verglichen.

`K` = **`1`** **Byte Schreiben**
Der Inhalt des durch die 3 Bit Adresse gekennzeichneten Registers wird durch das Datenbyte **`DDDD-DDDD`** im zweiten Befehlsbyte ersetzt.

5.2 Befehlssequenz im Registermodus

Die Befehlssequenz zum Programmieren im Registermodus sind in [RCN216] im Abschnitt "Registermodus" festgelegt.

5.3 Sonderformen des Registermodus

5.3.1 Address Only Mode

In der NMRA Norm S-9.2.3 wird der Zugriff auf CV #1 über Register 1 als "Address Only Mode" bezeichnet. Dieses dient nur dazu, diesen Zugriff als eigenständige Konfigurations-methode in die Liste der Minimalanforderungen aufzunehmen.

5.3.2 Decoder Factory Reset

In der NMRA Norm S-9.2.3 wird das Schreiben des Wertes 8 in Register 8 über Register 8 als "Decoder Factory Reset" bezeichnet. Mit diesem Schreibbefehl auf die nicht änderbare Konfigurationsvariable wird ein Überschreiben **aller** CVs mit dem jeweiligen vom Hersteller vorgegebenen Standardwert initialisiert.

Da die Befehlssequenz nicht ausreichend Zeit bietet, um alle CVs zu überschreiben, werden die internen Schreibbefehle bei den folgenden Spannungszyklen durchgeführt. Solange diese Operation nicht abgeschlossen ist wird CV 8 temporär auf 255 gesetzt.

Für Werte, die mit dem Konfigurationsvariablen Zugriffsbefehl - Lange Form in CV #8 geschrieben werden um den Decoder zurückzusetzen, wird auf [RCN225] verwiesen.

5.3.3 Paged Mode

Wie im Abschnitt zum Befehlsaufbau angesprochen, kann das Register 6 als Indexregister verwendet werden. Andere Nutzungen dieses Registers sind nicht zulässig.

Der Wert im Register 6 bildet eine 4 Byte große Seite von CVs auf die Register 1 bis 4 ab. Über die 256 möglichen Werte in Register 6 lassen sich alle 1024 CVs erreichen. Die CV-Nummer, auf die über Register 1 zugegriffen wird ergibt sich nach der Formel:

<Inhalt Register 6> * 4 - 3

Über die Register 2 bis 4 wird auf die drei folgenden CVs zugegriffen.

Umgekehrt kann man den Zugriff auf ein bestimmtes CV folgender Maßen ermitteln:

Register 6 := 1 + ganzzahliger Teil von (<CV-Nummer> - 1) / 4
Register-Nummer für Zugriff = 1 + Rest aus der Division.

Der indizierte Zugriff auf die CVs erfolgt mit dem im Abschnitt Befehlssequenz im Registermode beschriebenen Ablauf mit zwei Änderungen:

- Anstatt den festen Wert 1 in Register 6 zu schreiben wird der für den indizierten Zugriff erforderliche Wert in Register 6 geschrieben.
- Das optionale Abschalten der Betriebsspannung nach dem Schreiben von Register 6 ist nicht zulässig.

Auf die Zuordnung der Register 5, 7 und 8 zu den CVs 29, 7 und 8 hat der Wert in Register 6 keinen Einfluss.

6 Bestätigung

6.1 Bestätigung im Programmiermodus

Im Programmiermodus muss jede erfolgreiche Schreiboperation und ein Prüfbefehl, bei dem der angefragte Wert mit dem Wert im Decoder übereinstimmt, mit einer Bestätigung beantwortet werden. Diese ist in [RCN216] im Abschnitt "Bestätigung" festgelegt.

Beim Lesen einer nicht implementierten CV, d.h. einer CV, der keine physikalische Speicherstelle entspricht, wird folgendes Verhalten empfohlen:

- Bei einem „Bit Überprüfen“ wird immer geantwortet, egal ob auf Bit = 0 oder Bit = 1 abgefragt wird.
- Bei einem „Byte Überprüfen“ wird nie geantwortet.

Damit muss bei der standardmäßigen Abfrage nicht 9-mal auf das Ende des Zeitfensters gewartet werden und es kann eine nicht implementierte CV sowohl von einer CV mit beliebigem Wert als auch von einem aus anderem Grund nicht antwortenden Decoder unterschieden werden.

6.2 Bestätigung im Betriebsmodus

Wenn der Decoder RailCom unterstützt und diese in dem Decoder freigegeben wurde, muss auf die Konfigurationsvariablen Zugriffsbefehle mit dem Inhalt der adressierten Konfigurationsvariablen geantwortet werden. Dabei ist bei Schreiboperationen in der langen Form auf den zweiten Befehl und bei Prüfoperationen und Schreiboperationen in der kurzen Form auf jeden Befehl mit einer Bestätigung zu antworten.

Anhang A: Verweise auf andere Normen

A.1 Normative Verweise

Die hier aufgeführten Normen sind ganz oder in dem beim Zitat angegebenen Rahmen einzuhalten, um diese Norm zu erfüllen.

[RCN210] RCN-210 DCC Bit-Übertragung

[RCN211] RCN-211 DCC Paketstruktur

[RCN215] RCN-215 DCC Übergänge der Betriebsarten[1]

[RCN216] RCN-216 DCC Programmierumgebung

A.2 Informative Verweise

Die hier aufgeführten Normen und Dokumente haben rein informativen Charakter und sind nicht Bestandteil dieser Norm.

[RCN217] RCN-217 DCC-Rückmeldeprotokoll – RailCom

[RCN225] RCN-225 DCC Konfigurationsvariablen

[S921] NMRA: S-9.2.1 DCC Extended Packet Formats

[S923] NMRA: S-9.2.3 DCC Service Mode

[1] Die RCN-215 war zum Zeitpunkt der Erstellung dieser Norm noch in Vorbereitung.

Anhang B: Historie

Datum	Kapitel	Änderungen gegenüber der jeweils vorhergehenden Version
02.12.2018	4	Neu: XPOM Befehl eingefügt.
22.07.2018	1.2 3.2 5.1	Nur ein Befehl in einem Paket zur Konfiguration Bevorzugte Verwendung der Kurzen Befehlsform für die CV 17 und 18 Verhalten bei nicht implementierter CV
03.08.2014	Alle	Erste Version

Anhang C: Veralteter Befehl

Dieser Anhang enthält ein veraltetes und daher gesperrtes Befehlsformat. Decoder dürfen dieses Paketformat nicht unterstützen, weil es mit dem NOP-Befehl für Zubehördecoder kollidiert. Zentralen dürfen dieses Paketformat nicht verwenden.

C.1 Zubehördecoder Konfigurationsvariablen Zugriffsbefehl

Das Paketformat für den veralteten Zubehördecoder Konfigurationsvariablen Zugriffsbefehl ist: **`10AA-AAAA 0AAA-11VV VVVV-VVVV DDDD-DDDD`**

Die Bitmuster beschrieben durch **`VV VVVV-VVVV`** in dem zweiten und dritten Byte und **`DDDD-DDDD`** in dem vierten Byte sind ebenfalls identisch zu den entsprechenden Bits in dem Konfigurationsvariablen Zugriffsbefehl - Lange Form. Dieser Befehl hatte die gleiche Aufgabe und darf nicht mehr verwendet werden.

Anhang D: F9-Befehl

Dieser Anhang enthält zwei veraltete und daher gesperrte Befehlsformate, die beide den gleichen Befehlscode verwenden. Decoder dürfen dieses Paketformat unterstützen, Zentralen dürfen dieses Paketformat aber nicht senden.

D.1 Adresssuchbefehl

Dieses ist ein alter Befehl, um die Adresse eines Decoders zu ermitteln. Dazu werden der Reihe nach die möglichen Werte für CV #1 abgefragt. Ein Lesen des CV #1 ist zu bevorzugen!

Das Befehlsformat lautet:

`{20 Synchronbits} 0 0AAA-AAAA 0 1111-1001 0 PPPP-PPPP 1`

Falls die Adresse im Befehl mit der Adresse des Decoders in CV #1 übereinstimmt, sendet der Decoder eine Bestätigung mittels Strompuls wie im Abschnitt Bestätigung der [RCN215] beschrieben.

Dieser Befehl ist nur für den Adressbereich 1 bis 111 gültig!

D.2 "Service Mode Decoder Lock Instruction"

Dieser Befehl wird in der Technical Note TN-1-05 der NMRA näher beschrieben.

Die "Service Mode Decoder Lock Instruction" (SMDLI) verhindert die Programmierung einiger Decoder, während andere auf dem gleichen Gleis programmiert werden können. Es werden dazu Befehle für den Programmiermodus auf dem normalen Fahrgleis verwendet, womit man nicht den Vorteil der strombegrenzten Testumgebung hat und Decoder, die diesen Befehl nicht unterstützen, ungewollt umprogrammiert werden können.

Das Befehlsformat lautet:

{20 Synchronbits} 0 0000-0000 0 1111-1001 0 0AAA-AAAA 0 PPPP-PPPP 1

Dabei steht das **AAA-AAAA** für die kurze Adresse des Decoders, der weiterhin Befehle des Programmiermodus ausführen wird.

Ein Decoder, der diesen Befehl unterstützt, vergleicht seine Adresse in CV #1 gegen die Adresse in dem Befehl. Stimmen die Adressen nicht überein, geht er in einen gesperrten Zustand über, in dem er auch nach Aus- und wieder Einschalten der Spannung verbleibt. In diesem Zustand ignoriert er alle Befehle des Programmiermodus.

Um diesen gesperrten Zustand zu verlassen muss er entweder einen Befehle des Betriebsmodus oder eine SMDLI mit übereinstimmender Adresse empfangen. Danach kann der Decoder wieder programmiert werden.
