

Norm	RCN-213 DCC-Protokoll Betriebsbefehle für Zubehördecoder	RailCommunity
Ausgabe 08.12.2014		RailCommunity – Verband der Hersteller Digitaler Modellbahnprodukte e.V.

Inhalt

1 Allgemeines	1
1.1 Zweck der Norm	1
1.2 Anforderungen	2
1.3 Formatdefinitionen.....	2
2 Zubehördecoder Paketformat	3
2.1 Einfache Zubehördecoder Paketformat.....	3
2.2 Notaus-Befehl für Einfache Zubehördecoder	5
2.3 Erweitertes Zubehördecodersteuerung Packet Format	6
2.4 Notaus-Befehl für Erweiterte Zubehördecoder.....	6
2.5 NOP Befehl für einfache und erweiterte Zubehördecoder.....	6
Anhang A: Verweise auf andere Normen.....	7
A.1 Normative Verweise	7
A.2 Informative Verweise.....	7
Anhang B: Historie.....	7

1 Allgemeines

1.1 Zweck der Norm

Diese Norm beschreibt die Pakete des DCC-Protokolls für Zubehördecoder im normalen Anlagenbetrieb. Die Befehle zur Konfiguration von Decodern sind in [RCN214] festgelegt und werden hier nur in sofern erwähnt, wie sie sich in die Befehlsstruktur einfügen. Alle Pakete folgen der in [RCN211] beschriebenen Paketstruktur.

Diese Norm entspricht im Wesentlichen Abschnitt D von [S921], wobei aber die Befehle zur Konfiguration von Decodern nach [RCN214] ausgelagert wurden.

1.2 Anforderungen

Um diese Norm zu erfüllen ist es nicht erforderlich alle definierten Pakete zu unterstützen. Folgende Befehle müssen aber von Zentralen für Zubehörsteuerung und allen Zubehördecodern mindestens unterstützt werden:

- Alle Decoder und Zentralen:
 - Rücksetzbefehl (aus [RCN211])
- Alle Zentralen für Zubehörsteuerung
 - Zubehördecoderbefehle im zwei Byte Format entsprechend 2.1
- Einfache Zubehördecoder:
 - Zubehördecoderbefehle im zwei Byte Format entsprechend 2.1
- Erweiterte Zubehördecoder:
 - Zubehördecoderbefehle im drei Byte Format entsprechend 2.3

Zentralen, die mehr als das Minimum unterstützen, dürfen als DCC-Protokoll für den Betrieb von Zubehör nur die hier definierten Befehle senden. Ausnahmen sind von der RailCommunity freigegebene Pakete zur Unterstützung von weiteren Entwicklungen.

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 möglichen Befehle, die sie **nicht** unterstützen, ignorieren und diese Befehle dürfen nicht zu einer Fehlfunktion des Decoders führen. Das gilt auch für alle bisher nicht definierten Befehlscodes.

Die Normen [RCN210] und [RCN211] 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

D Datenbit

R Richtungsbit

x Bits, die an der entsprechenden Stelle nicht betrachtet werden

2 Zubehördecoder Paketformat

Digitale Zubehördecoder sind zur Steuerung einer Zahl von einfachen Funktionen bestimmt, wie ein Weichenantrieb oder Lichter ein- und auszuschalten. Es ist zulässig Decoder zu entwickeln, die auf mehrere Adressen reagieren, um mehr Geräte von einem einzelnen Decoder steuern zu können.

Bei den Befehlen für Zubehördecoder gibt es keine klare Trennung zwischen den Bytes für die Adresse und den Befehlen. Daher wird hier immer das gesamte Paket, allerdings ohne Synchronbits, Startbit, Null-Bits zwischen den Bytes, Prüfbyte und Paketendebyte dargestellt. Also nur das, was bei den Paketen für Fahrzeugdecoder die Adress- und Befehlsbytes sind.

Es kann immer nur ein Befehl in einem Paket für Zubehördecoder sein.

Wenn zukünftig eine Betriebsmodusquittung definiert ist und diese in dem Decoder freigegeben wurde, muss der Empfang jedes dieser Befehle mit einer Betriebsmodusquittung beantwortet werden.

2.1 Einfache Zubehördecoder Paketformat

Das Format für einfache Zubehördecoder ist: **10AA-AAAA 1AAA-DAAR** (zwei Byte Format)

Digitale Zubehördecoder können entworfen werden, um kurzzeitig oder permanent eingeschaltete Geräte (z.B. Magnetartikel, Lichter, Lichtsignale, Motoren, Servos) zu steuern.

Bit 3 des zweiten Bytes "**D**" wird benutzt, um das adressierte Gerät zu aktivieren (**D = 1**) oder zu deaktivieren (**D = 0**). Über diesen Weg ergibt sich, wenn der optionale Befehl zum Deaktivieren verwendet wird, die maximale Schaltzeit für den adressierten Ausgang. Wird während der Schaltzeit der Aktivierungsbefehl (**D = 1**) regelmäßig wiederholt, bestimmt sich dadurch die Länge der Schaltzeit auch ohne Deaktivierungsbefehl (**D = 0**). Wird das adressierte Gerät (z.B. Lichtsignal, Motor oder Servo) vom Decoder zeitlich individuell angesteuert, ist die von der Zentrale vorgegebene Schaltzeit für dieses Gerät nicht relevant.

Handelt es sich um reine Schaltausgänge, sollte zum Schutz der Antriebe der Ausgang eine gewisse Zeit nach einem Aktivierungsbefehl (**D = 1**) automatisch wieder abgeschaltet werden. Diese Zeit muss konfigurierbar sein, sofern der Decoder nicht für eine spezifische Anwendung fest vorkonfiguriert ist. In [RCN225] werden die dafür vorgesehenen CVs festgelegt. Soll ein Schaltausgang länger als die vom Decoder vorgegebene Zeit aktiv bleiben, muss die Zentrale wiederholt Aktivierungsbefehle (**D = 1**) senden.

Die Ausgänge sind paarweise adressiert. Bit 0 des zweiten Bytes **R** wird benutzt um den Ausgang innerhalb des Paares auszuwählen. Bei Standardanwendungen bedeutet **R = 0** Weiche auf Abzweig oder Fahrtrichtung links bzw. Signal auf Halt und **R = 1** Weiche gerade oder Fahrtrichtung rechts bzw. Signal auf Fahrt. Anmerkung: Bei Großbahnen ist die Definition der Weichenlage über rechts und links üblich, bei den kleineren Bahnen über Abzweig und Gerade. Bei Decodern, die fest mit einer Weiche verbunden sind, ist eine Konfigurationsmöglichkeit vorzusehen, um die Weichenlage anpassen zu können. Die Ausgänge eines Paares können bei entsprechender Konfiguration auch getrennt geschaltet werden.

Die höchstwertigsten Bits der 11-Bit-Adresse sind die Bits 4 bis 6 des zweiten Bytes **xAAA-xxxx**. Diese Bits werden im Einerkomplement übertragen. Danach kommen die Bits 0 bis 5 des ersten Bytes **xxAA-AAAA**. Die niederwertigsten Bits der 11-Bit-Adresse sind die Bits 1 bis 2 des zweiten Bytes **xxxx-xAAx**.

Daraus ergeben sich folgende Wertigkeiten, wobei ein '/' ein invertiertes Adressbit kennzeichnet:

Paket: **1 0 A A - A A A A | 1 A A A - D A A R**

Adresse: – – A7 A6 - A5 A4 A3 A2 – /A10 /A9 /A8 – – A1 A0 –

Aus Gründen der Kompatibilität zu existierenden Zentralen ist die erste angesprochene Adresse 4: **1000-0001 1111-D00R**. Diese Adresse wird in Anwenderdialogen als Adresse 1 dargestellt.

Bei einigen existierenden Zentralen folgt auf die Adresse 255 nicht die Adresse 256, sondern es werden die Adressen 0 bis 3 eingefügt. Danach geht es bei 260 weiter. Entsprechend bei ab Adresse 511, 767, 1023 usw.:

255 **1011-1111 1111-D11R**

0 **1000-0000 1111-D00R**

1 **1000-0000 1111-D01R**

2 **1000-0000 1111-D10R**

3 **1000-0000 1111-D11R**

260 **1000-0001 1110-D00R**

261 **1000-0001 1110-D01R**

...

511 **1011-1111 1110-D00R**

256 **1000-0000 1110-D00R**

257 **1000-0000 1110-D01R**

258 **1000-0000 1110-D10R**

259 **1000-0000 1110-D11R**

516 **1000-0001 1101-D00R**

517 **1000-0001 1101-D01R**

...

Es ist jedoch eine lineare Adressfolge anzustreben:

```

255 1011-1111 1111-D11R
256 1000-0000 1110-D00R
257 1000-0000 1110-D01R
258 1000-0000 1110-D10R
259 1000-0000 1110-D11R
260 1000-0001 1110-D00R
261 1000-0001 1110-D00R
...
511 1011-1111 1111-D00R
512 1000-0000 1110-D00R
513 1000-0000 1110-D01R
514 1000-0000 1110-D10R
515 1000-0000 1110-D11R
516 1000-0001 1101-D00R
517 1000-0001 1101-D01R
...

```

Die vier ersten Adressen können am Ende des Adressbereichs angesprochen werden, wobei die Adresse 2047 als Adresse für den Notaus-Befehl übersprungen wird:

```

2045 1011-1111 1000-D01R
2046 1011-1111 1000-D10R
0    1000-0000 1111-D00R
1    1000-0000 1111-D01R
2    1000-0000 1111-D10R
3    1000-0000 1111-D11R

```

Bei Neuentwicklungen muss die lineare Fortschaltung unterstützt werden. Um die Kompatibilität zu vorhandenen Anlagen zu erhalten, kann eine Einstellbarkeit zwischen den eingefügten zurückliegenden Vierblöcken und der linearen Adressfolge vorgesehen werden.

2.2 Notaus-Befehl für Einfache Zubehördecoder

Das Format für den Notaus-Befehl ist: **1011-1111 1000-0110**. Es entspricht einem Ausschaltbefehl (**D = 0**) des ersten Ausgangs (**R = 0**) auf Adresse 2047. Dabei sind alle Ausgänge zu deaktivieren.

Dieses Paket muss von allen einfachen Zubehördecodern ausgeführt werden.

2.3 Erweitertes Zubehördecodersteuerung Packet Format

Das Format für erweiterte Zubehördecoder ist: **10AA-AAAA 0AAA-0AA1 DDDD-DDDD** (drei Byte Format).

Das erweiterte Zubehördecodersteuerungspaket wurde eingeführt, um bis zu 256 Zustände an komplexere Zubehördecoder zu übertragen. Jeder Adresse ist immer nur ein Gerät zugeordnet.

Aus der Kompatibilität zum Basispaket ergeben sich bei folgende Wertigkeiten, wobei ein '/' ein invertiertes Adressbit kennzeichnet:

Paket: **1 0 A A - A A A A | 0 A A A - 0 A A 1**

Adresse: – – A7 A6 - A5 A4 A3 A2 – /A10 /A9 /A8 – – A1 A0 –

Über die Bits 0 bis 7 im dritten Byte **DDDD-DDDD** werden die 256 möglichen Zustände übertragen. Handelt es sich bei dem angeschlossenen Gerät um ein Lichtsignal, können bis zu 256 Signalbegriffe angezeigt werden. Ein Wert von **DDDD-DDDD = 0000-0000** bedeutet den absoluten Haltbegriff. Alle anderen durch **DDDD-DDDD** vertretenen Signalbilder sind von dem verwendete Signalsystem bzw. dem nachgebildete Vorbild abhängig und werden daher hier nicht festgelegt. In der [NEM694] gibt es Vorgaben für die Codierung der Signalbegriffe.

2.4 Notaus-Befehl für Erweiterte Zubehördecoder

Das Format für den Notaus-Befehl ist: **1011-1111 0000-0111 0000-0000**

Alle Erweiterten Zubehördecoder müssen dieses Paket ausführen. Es entspricht dem Wert 0 (= Halt bei Lichtsignalen) für die Adresse 2047. Da in dem Abschnitt für das Erweiterte Zubehördecoder Paketformat nur für **DDDD-DDDD = 0000-0000** eine einheitliche Festlegung gibt, ist nur dieser Wert im Notaus-Befehl erlaubt.

2.5 NOP Befehl für einfache und erweiterte Zubehördecoder

Das Format für den NOP Befehl ist: **10AA-AAAA 0AAA-1AAT**

Der NOP-Befehl ermöglicht RailCom-fähigen Zubehör-Decodern einen Service-Request (SRQ) an die Zentrale zu senden, ohne dass der aktuelle Zustand verändert wird. Dieser Befehl wird von allen nicht RailCom-fähigen einfachen und erweiterten Zubehör-Decodern als ungültig erkannt und damit ignoriert.

T = 0: Einfacher Zubehördecoder

T = 1: Erweiterter Zubehördecoder

Nähere Informationen in der RailCom-Spezifikation.

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

A.2 Informative Verweise

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

[RCN214] [RCN-214](#) DCC Konfigurationsbefehle

[RCN225] RCN-225 DCC Konfigurationsvariablen¹

[NEM694] [NEM 694](#) Bus-Protokolle für Steuermodule

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

Anhang B: Historie

Datum	Änderungen seit der vorhergehenden Version
08.12.2014	Erste Version

Copyright 2014 RailCommunity – Verband der Hersteller Digitaler Modellbahnprodukte e.V.

¹ Die RCN-225 war zum Zeitpunkt der Erstellung dieser Norm noch in Vorbereitung. Sie wird auf der Basis der NMRA S-9.2.2 erstellt.