

Normvorschlag	RCN-217 RailCom DCC-Rückmeldeprotokoll	RailCommunity
Ausgabe 18.12.2016		RailCommunity – Verband der Hersteller Digitaler Modellbahnprodukte e.V.

Inhalt

1 Allgemeines.....	3
2 Physical Layer	3
2.1 Allgemeines	3
2.2 RailCom – Sender im Decoder	4
2.3 Der RailCom Detector	5
2.4 Timing.....	5
2.5 Datensicherung	7
3 Packet Layer.....	8
3.1 RailCom Befehlsart MOB.....	9
3.2 RailCom Befehlsart STAT.....	10
4 RailCom CVs und DCC-Befehle	12
4.1 Systemanforderungen	12
4.2 CVs	12
4.2.1 CV28 RailCom Konfiguration	12
4.2.2 CV29	12
4.2.3 CV31, CV32.....	12
4.2.4 RailCom Page.....	13
4.3 DCC-Befehle.....	14
4.3.1 Systemkommandos.....	14
4.3.2 Zusätzliche Funktionskommandos	14
4.3.3 Erweiterte Programmierbefehle	14
4.3.4 NOP für Zubehördecoder	15
5 Anwendungen (Application/APP) Layer für Fahrzeugdecoder	17
5.1 POM.....	17
5.1.1 Byte Lesen.....	18
5.1.2 Byte Schreiben	18
5.1.3 Bit Schreiben	19
5.2 ADR	19
5.3 EXT.....	20
5.3.1 Senden der Ortsinformation:	20

5.3.2 Befüllen	21
5.4 DYN.....	21
5.5 subID.....	23
5.5.1 Liste der bisher vergebenen subIDs	23
6 Anwendung für Zubehördecoder (Weichen etc.).....	24
6.1 SRQ - Service Request für Zubehördecoder.....	24
6.2 POM.....	25
6.2.1 Byte Lesen.....	26
6.2.2 Byte Schreiben	27
6.2.3 Bit Schreiben	27
6.3 STAT1.....	27
6.4 ZEIT	28
6.5 FEHLER	29
6.6 DYN.....	30
6.7 STAT2.....	30
Anhang A: Verweise auf andere Normen	31
A.1 Normative Verweise	31
A.2 Informative Verweise.....	31
Anhang B: Historie.....	31

1 Allgemeines

"RailCom" ist eine auf den Namen von Lenz Elektronik für die Klasse 9 "Elektronische Steuerungen" unter der Nummer 301 16 303 eingetragene Deutsche Marke sowie ein für die Klassen 21, 23, 26, 36 und 38 " Electronic Controls for Model Railways" in U.S.A. unter Reg.Nr. 2,746,080 eingetragene Trademark. Das Europäische Patent 1 380 326 B1 wurde aufgehoben. Damit ist RailCom unter Beachtung der Warenzeichen frei verwendbar.

Diese Spezifikation gilt ausschließlich für die Anwendung von RailCom innerhalb des DCC-Datenformats (Protokolls). Die Anwendung von RailCom innerhalb anderer Datenformate ist nicht zugelassen.

2 Physical Layer

Dieses Kapitel beschreibt die physikalische Schicht von RailCom.

2.1 Allgemeines

Der Informationsfluss im DCC-System erfolgt normalerweise von der Zentrale (Booster) über das Gleis zu den Decodern. Für die umgekehrte Übertragungsrichtung ist es erforderlich, diesen Energie- und Datenstrom zu unterbrechen. Dies geschieht durch die Booster, die dazu am Ende eines jeden DCC-Paketes ein sogenanntes RailCom-Cutout erzeugen, indem sie die beiden Track (Gleis)-Leitungen von der Spannungsversorgung trennen und kurzschließen. Diese Funktionsgruppe innerhalb des Boosters wird "Cutout Device" genannt. Ein solches Cutout Device könnte auch als separate Einheit außerhalb des Boosters ausgeführt werden. Die eigentliche Datenübertragung erfolgt mittels einer Stromschleife. Den dazu notwendigen Strom muss der Decoder aus seinem internen Puffer bereitstellen. Abbildung 1 zeigt die Anordnung von Booster, Detector und Decoder während des RailCom-Cutout.

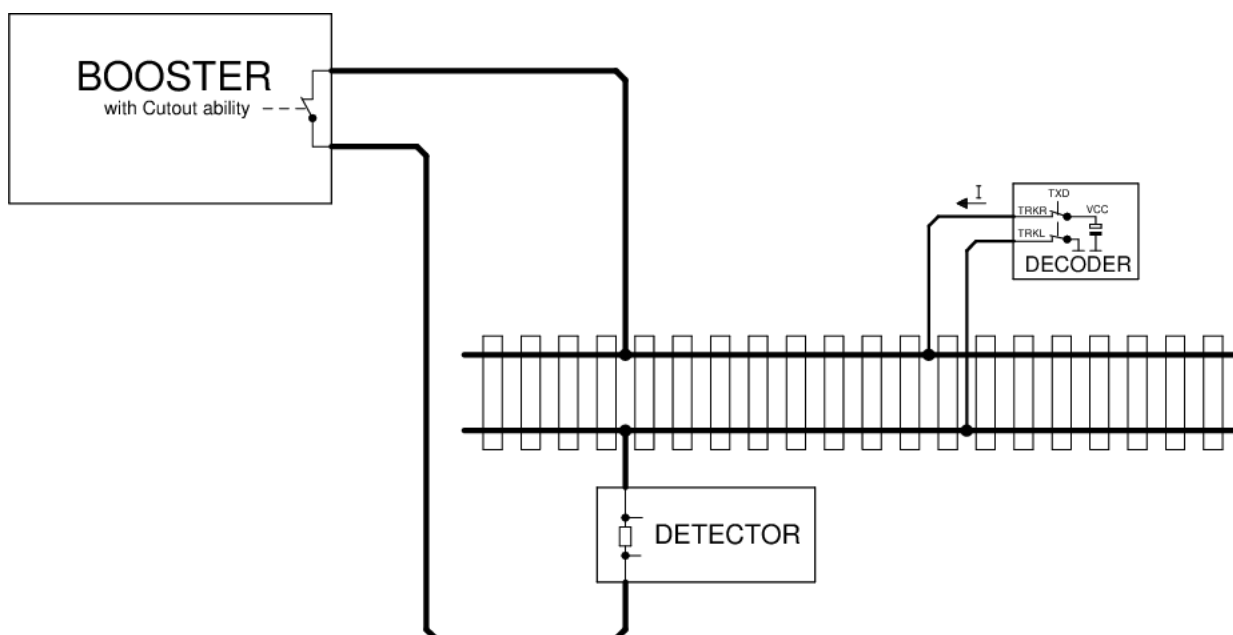


Bild 1: RailCom Prinzipschaltbild

2.2 RailCom – Sender im Decoder

Um eine '0' zu übertragen, muss der Decoder einen Strom I von $30+4/-6$ mA liefern, bei einem Spannungsabfall am Gleis von bis zu 2.2 V. Bei einer '1' darf der Strom I höchstens ± 0.1 mA betragen. Die Stromquelle des Decoders muss gegen unerwartete Fremdspannung am Gleis während des Cutout geschützt sein. Abbildung 2 zeigt eine mögliche Hardware-Realisierung.

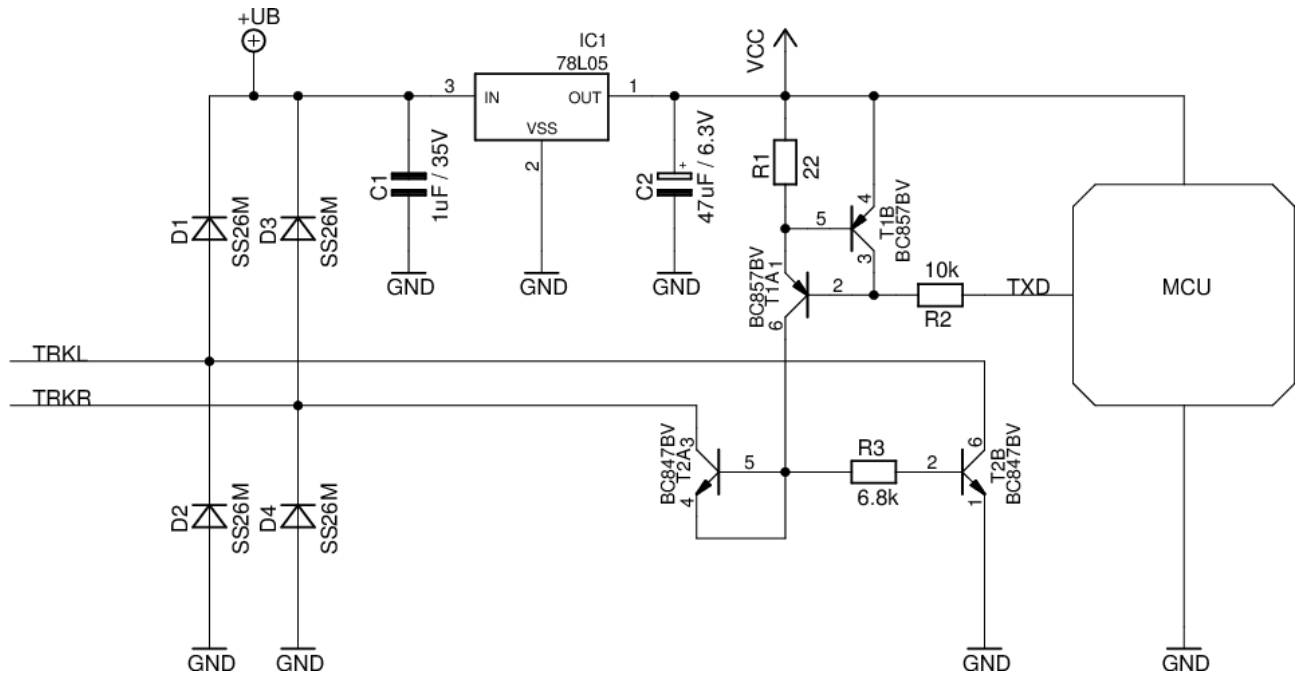


Bild 2: RailCom Decoder Hardware

Erläuterung zum Schaltbild:

Der RailCom-Teil besteht nur aus den Widerständen R1 bis R3 und den Transistoren T1A bis T2B. T1A und T1B bilden eine Stromquelle, T2A ist als Diode geschaltet und schützt die Stromquelle vor positiven Spannungen höher V_{cc} .

Alle anderen Teile der Schaltung gehören zur normal notwendigen Hardware des Decoders. Man beachte den äußerst geringen Hardwareaufwand für den RailCom-Sender.

2.3 Der RailCom Detector

Ein Detector muss einen Strom von größer 10mA während der mittleren 50% der Bitzeit als '0' interpretieren, einen Strom von kleiner 6 mA während der mittleren 50% der Bitzeit als '1'. Der Spannungsabfall über dem Detector darf 200mV bei maximal 34 mA während des Cutout nicht übersteigen. Abbildung 3 zeigt eine mögliche Hardware Realisierung.

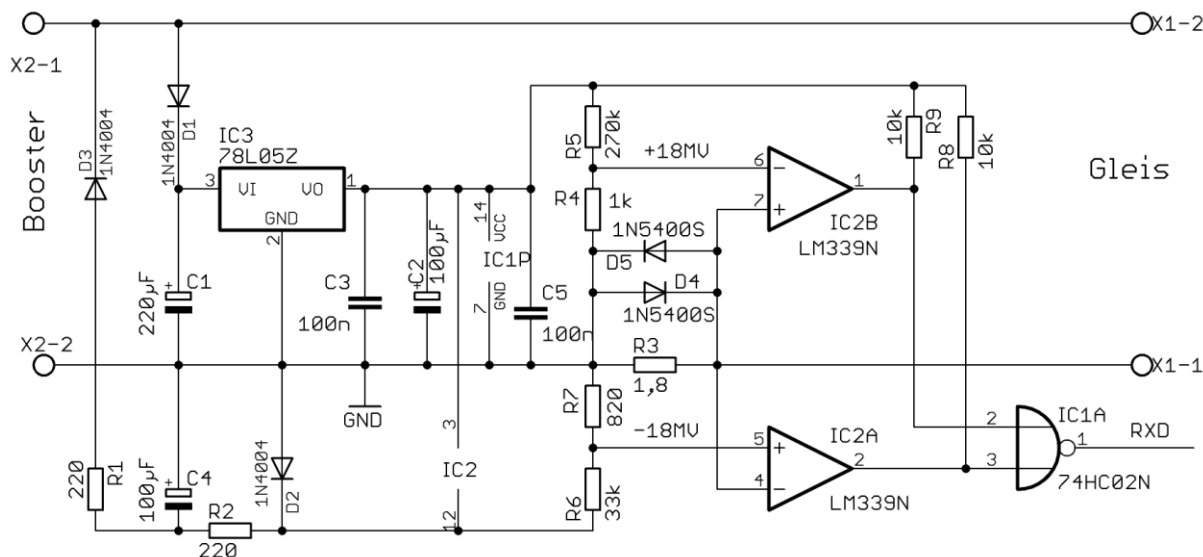


Bild 3: Beispiel einer simplen RailCom Detector Hardware

Der Spannungsabfall über dem Cutout Device darf 10mV bei maximal 34 mA während des Cutout nicht übersteigen.

Es dürfen maximal zwei Detektoren (inkl. des globalen Detectors) in Reihe verwendet werden, wobei der lokale Detector einen Anschluss für externe Auswertung einer Gleisbelegung enthalten sollte. Ist dies nicht der Fall, müssen extern verwendete Belegtmelder für RailCom spezifiziert sein. Erläuterung:

Getestet wurden diese Schaltungen (Sender und Detector) auf großen Clubanlagen bis zu einer Entfernung von 100m. Diese Entfernung wurde problemlos überbrückt. Zugelassen sind dabei nicht vom Gleis durch Brückengleichrichter isolierte Verbraucher von 5 Ohm, die parallel zum Messwiderstand des Detectors liegen. Der Wert von 5 Ohm entspricht bei einer Gleisspannung von 15V einem Strom von 3A. Glühlampen (Kaltleiter) sind immer über einen schnellen Brückengleichrichter (< 500 ns) zu betreiben.

2.4 Timing

In einem Cutout können bis zu 8 Byte Daten übertragen werden. Jedes übertragene Byte beginnt mit einem Startbit ('0') gefolgt von den 8 Datenbits (niederwertigstes Bit zuerst) und endet mit einem Stoppbit ('1'). Die Übertragungsrate ist 250 kbit/s \pm 2%. Die Anstiegszeit (10% \rightarrow 90%) und Abfallzeit (90% \rightarrow 10%) darf 0,5 μ s nicht überschreiten.

Das RailCom Cutout ist in zwei Kanäle unterteilt. Im Kanal 1 können 2 Bytes, im Kanal 2 bis zu 6 Bytes übertragen werden. Abbildung 4 zeigt das Timing Diagramm. Sämtliche Zeiten sind auf den Null-Durchgang der letzten Flanke des Paket-Endebits bezogen.

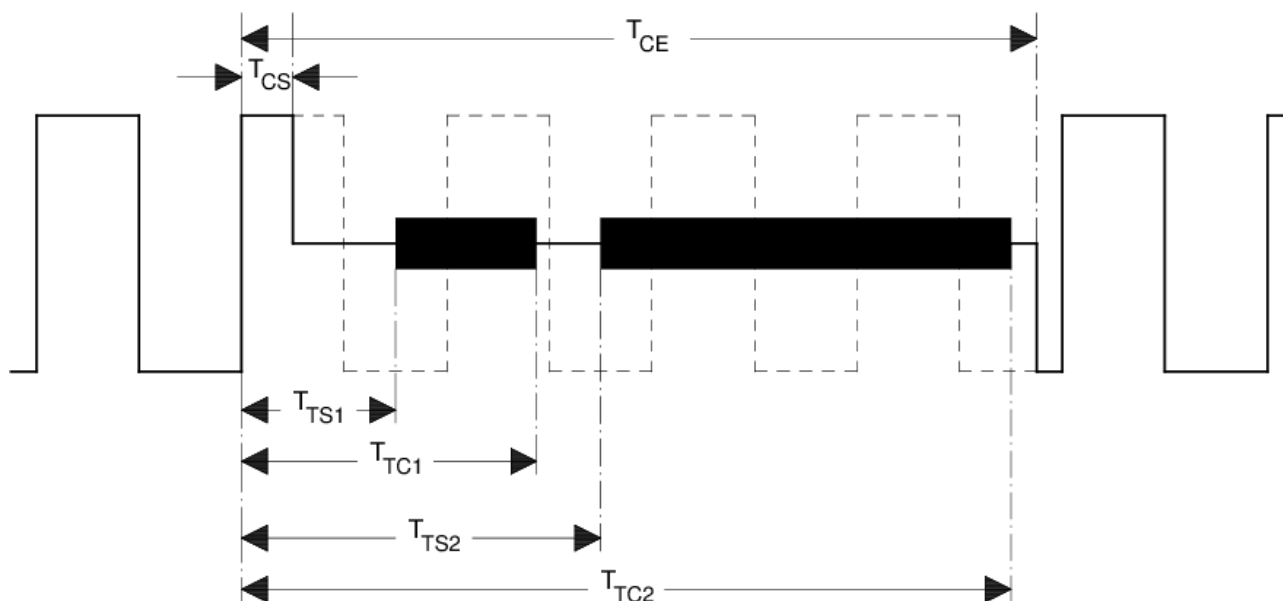


Bild 4: RailCom Timing

Parameter	Name	Min	Max
Cutout Start	TCS	26 μ s	32 μ s
Cutout End	TCE	454 μ s	488 μ s
Start Kanal 1	TTS1	80 μ s	
End Kanal 1	TTC1		177 μ s
Start Kanal 2	TTS2	193 μ s	
End Kanal 2	TTC2		454 μ s

Tabelle 1: Timing Parameter

Bemerkung:

Obige Abbildung zeigt das RailCom Timing mit "1"-Bits von $2 \cdot 58 \mu$ s (Nominalwert des DCC "1"-Bit). Bei kürzeren "1"-Bits ist es möglich, dass das Cutout ins 5. "1"-Bit hineinreicht. Dies ist aber kein Problem, da eine Zentrale mindestens $4 + 12 = 16$ Präambelbits senden muss (Paket-Endebit des vorherigen DCC-Pakets nicht mitgezählt), der Decoder also ausreichend Präambelbits (mindestens 11; 10 sind notwendig) sieht.

Eine Cutout Zeit von ca. 450 μ s darf die Funktion eines Decoders, der RailCom nicht beherrscht, nicht beeinflussen, da auf einer realen Modellbahnanlage Stromunterbrechungen bis 20ms nachgewiesen wurden, d.h. ein Decoder sollte mindestens eine Stromunterbrechung in dieser Größenordnung verarbeiten können.

2.5 Datensicherung

Die Sicherung der Datenübertragung erfolgt via 4/8 Codierung, d.h. jedes übertragene Byte enthält 4 '1'- und 4 '0'-Bits. Ist dieses Verhältnis verletzt, liegt ein Übertragungsfehler vor.

Es gibt 70 verschiedene Bitkombinationen innerhalb eines Bytes, die dieses Verhältnis 4:4 aufweisen. Davon werden 64 für die Übertragung von 6 Nutzbits verwendet, von den 6 übrigen wird eine für kurze Sondermitteilungen genutzt, ACK. Die restlichen fünf Kombinationen werden im Moment nicht benutzt.

Es lassen sich in Kanal 1 netto 12 Bit, in Kanal 2 netto bis zu 36 Bit Nutzdaten übertragen.

Die möglichen Codierungen:

Wert	4/8 Code	Wert	4/8 Code	Wert	4/8 Code	Wert	4/8 Code
0x00	10101100	0x10	10110010	0x20	01010110	0x30	11000110
0x01	10101010	0x11	10110100	0x21	01001110	0x31	11001100
0x02	10101001	0x12	10111000	0x22	01001101	0x32	01111000
0x03	10100101	0x13	01110100	0x23	01001011	0x33	00010111
0x04	10100011	0x14	01110010	0x24	01000111	0x34	00011011
0x05	10100110	0x15	01101100	0x25	01110001	0x35	00011101
0x06	10011100	0x16	01101010	0x26	11101000	0x36	00011110
0x07	10011010	0x17	01101001	0x27	11100100	0x37	00101110
0x08	10011001	0x18	01100101	0x28	11100010	0x38	00110110
0x09	10010101	0x19	01100011	0x29	11010001	0x39	00111010
0x0A	10010011	0x1A	01100110	0x2A	11001001	0x3A	00100111
0x0B	10010110	0x1B	01011100	0x2B	11000101	0x3B	00101011
0x0C	10001110	0x1C	01011010	0x2C	11011000	0x3C	00101101
0x0D	10001101	0x1D	01011001	0x2D	11010100	0x3D	00110101
0x0E	10001011	0x1E	01010101	0x2E	11010010	0x3E	00111001
0x0F	10110001	0x1F	01010011	0x2F	11001010	0x3F	00110011
reserviert ¹	00001111						
ACK ¹	11110000	"Befehl ist verstanden und wird ausgeführt werden" oder "JA"					
reserviert	11100001						
reserviert	11000011						
reserviert	10000111						
reserviert	00111100						

Tabelle 2: Codierungen im 4/8 Code

¹ In früheren Versionen dieser Spezifikation wurde für ACK auch die Bitkombination 11110000 verwendet.

3 Packet Layer

Dieses Kapitel beschreibt den Aufbau von RailCom-Paketen.

RailCom Pakete (im Folgenden mit Datagramm bezeichnet) haben eine Länge von 6, 12, 18 oder 36 Nutz-Bits. Somit ergeben sich für die beiden Kanäle folgende Übertragungsmöglichkeiten:

Kanal 1	Kanal 2
12 Bit Datagramm	6 Bit Datagramm (ACK)
	12 Bit Datagramm
	12 Bit Datagramm + 12 Bit Datagramm
	12 Bit Datagramm + 12 Bit Datagramm + 12 Bit Datagramm
	18 Bit Datagramm
	18 Bit Datagramm + 12 Bit Datagramm
	18 Bit Datagramm + 18 Bit Datagramm
	24 Bit Datagramm
	36 Bit Datagramm

Tabelle 3: Übertragungsmöglichkeiten

Andere Möglichkeiten sind reserviert. Optional kann der Datenkanal mit ACK auf 36 Bit aufgefüllt werden.

Datagramme (außer ACK) beginnen, wenn nicht anders erläutert, mit einem 4 Bit Identifier, gefolgt von 8, 14 oder 32 Bit Nutzdaten, die wie folgt übertragen werden:

Datagramm	Bytes
12 Bit	ID[3-0]D[7-6]+D[5-0]
18 Bit	ID[3-0]D[13-12]+D[11-6]+D[5-0]
24 Bit	ID[3-0]D[19-18]+D[17-12]+D[11-6] +D[5-0]
36 Bit	ID[3-0]D[31-30]+D[29-24]+D[23-18]+D[17-12]+D[11-6] +D[5-0]

Tabelle 4: Datagramm-Aufbau

Die Länge des Datagramms ist durch den Identifier bestimmt. Die Identifier werden weiter unten definiert (Ausnahme: ID0 hat zwei verschiedene Längen).

Mobile Decoder (Fahrzeugdecoder) und stationäre Decoder (Zubehördecoder) haben unterschiedliche Rückmeldeanforderungen. Entsprechend werden die Kanäle für beide Decodertypen unterschiedlich genutzt. Die Bedeutung der Datagramme ist somit abhängig von der Adresse des vorangestellten DCC Pakets. Daneben gibt es Systemanforderungen, die alle Decoder gleichermaßen erfüllen müssen. Für diesen Zweck wird die DCC-Adresse 255

als "Systemadresse" festgelegt. Entsprechend werden die folgenden RailCom-Befehlsarten MOB (mobil) und STAT (stationär) anhand der DCC Adresse unterschieden:

DCC Adresse		RailCom Befehlsart
1. Byte	2. Byte	
1-127		MOB (Fahrzeugdecoder)
128-191		STAT (Zubehördecoder)
192-231	ADR low	MOB (Fahrzeugdecoder)
253		Programmieradresse ²
255		MOB/STAT 255 ist die Adresse des Leerlauf-Paketes (0xFF - 0x 00 - 0xFF) Das 1. Datenbyte ungleich 0 kennzeichnet die Verwendung als Systemadresse. In jedem Fall besteht das Paket dann aus mehr als 3 Byte.

Tabelle 5: Befehlsarten und Systemadressen

Auf andere Adressen sowie auf Service Mode Pakete dürfen Decoder keine Rückmeldung senden.

3.1 RailCom Befehlsart MOB

Kanal 1 nutzen mobile Fahrzeugdecoder zur schnellen Lokalisierung auf der Anlage (siehe app:adr). Dazu müssen sie nach jedem an einen Fahrzeugdecoder gerichteten DCC Paket ihre DCC-Adresse senden, die dann von lokalen Detectoren auf der Anlage empfangen wird. Ausgenommen sind die Programmiermodus-Pakete von dem Zeitpunkt an, zu dem der Decoder den Programmiermodus erkennt.

Kanal 2 darf nur vom adressierten Decoder benutzt werden und dient zur Übermittlung von Decoderinformationen. Ein adressierter Decoder muss stets eine Rückmeldung in Kanal 2 senden (gegebenenfalls ACK), um den fehlerfreien Empfang des DCC-Paketes zu bestätigen.

Eine Rückmeldung im Kanal 2 signalisiert, dass der Decoder den Befehl fehlerfrei empfangen hat, nicht jedoch, dass der Befehl vom Decoder auch akzeptiert und ausgeführt wird.

² gemäß RailCommunity Beschluss vom 16./17.09.2011

Die folgenden Identifier (Datagramme) sind für Fahrzeugdecoder definiert:

ID	Kanal 1	Kanal 2	Bemerkung
0		app:pom	Zwingend, 1 Byte
1	app:adr_high		zwingend
2	app:adr_low		zwingend
3		app:ext	optional
4			reserviert
5			reserviert
6			reserviert
7		app:dyn	optional
8			reserviert
9			reserviert
10			reserviert
11			reserviert
12		app:subID	optional
13			reserviert
14			reserviert
15			reserviert

Tabelle 6: Befehlsart MOB Identifier (Datagramme)

"zwingend" → bedeutet vollständige Implementierung

"optional" → entweder vollständige Implementierung oder: teilweise Implementierung mit den unter 4.1 genannten Bedingungen

Bemerkung:

Ältere Decoder haben während einer Testphase verschiedene Identifier belegt, die jetzt als "reserviert" gekennzeichnet sind. Neuere Decoder müssen in einer speziellen CV (siehe Abschnitt "RailCom CVs") die RailCom-Versionsnummer eingetragen haben. Dies kann zur Unterscheidung benutzt werden.

Ältere Decoder ohne Versionsnummer sollte man updaten (lassen).

3.2 RailCom Befehlsart STAT

Die RailCom Spezifikation für Zubehördecoder ist noch nicht abgeschlossen. Alle Angaben dazu sind als vorläufig zu betrachten.

Zubehördecoder nutzen Kanal 1 zur Meldung von Service Request Anforderungen (siehe app:srq). Dazu können sie nach jedem an einen Zubehördecoder gerichteten DCC Paket ihre Identität (12-Bit Adresse) senden (12 Bit Wert ohne Identifier !!) (Nicht bei Adressierung via Decoder ID). Melden sich mehrere Decoder gleichzeitig, muss eine Suche gestartet werden.

Kanal 2 darf nur vom adressierten Decoder benutzt werden und dient zur Übermittlung von Decoderinformationen. Ein adressierter Decoder muss stets eine Rückmeldung in Kanal 2 senden (gegebenenfalls ACK), um den fehlerfreien Empfang des DCC-Pakets zu bestätigen.

Eine Rückmeldung im Kanal 2 signalisiert, dass der Decoder den Befehl fehlerfrei empfangen hat, nicht jedoch, dass der Befehl vom Decoder auch akzeptiert und ausgeführt wird.

Die folgenden Identifier (Datagramme) sind für Zubehördecoder definiert:

ID	Kanal 1	Kanal 2	Bemerkung
	app:srq		zwingend
0		app:pom	zwingend
1			reserviert
2			reserviert
3			reserviert
4		app:stat1	zwingend
5		app:zeit	optional
6		app:fehler	zwingend
7		app:dyn	optional
8		app:stat2	nicht für Neuentwicklungen
9			reserviert
10			reserviert
11			reserviert
12		app:subID	optional
13			reserviert
14			reserviert
15			reserviert

Tabelle 7: Befehlsart STAT Identifier (Datagramme)

"zwingend" → bedeutet vollständige Implementierung

"optional" → entweder vollständige Implementierung oder: teilweise Implementierung mit den unter 4.1 genannten Bedingungen

4 RailCom CVs und DCC-Befehle

4.1 Systemanforderungen

Diese RailCom Spezifikation ist so aufgebaut, dass sie rückwärtskompatibel ist, d.h. nicht RailCom-fähige Decoder können weiter betrieben werden und nicht RailCom-fähige Zentralen können weiter RailCom-fähige Decoder ansteuern.

Das Ansteuern analoger Fahrzeuge (Lokomotiven ohne Decoder) mit Hilfe der verlängerten "0" (stretched zero) ist nicht zugelassen.

Ein Decoder mit RailCom Implementierung muss folgende Eigenschaften zwingend unterstützen:

APP.ADR; APP.POM; ACK

Auf jede Fahrzeug- bzw. Weichen- Adressierung muss der Decoder im Kanal 2 der Austastlücke antworten (siehe Applikationen).

4.2 CVs

4.2.1 CV28 RailCom Konfiguration

Bit	=	Funktion
0	1	Kanal 1 freigegeben für Adress-Broadcast
	0	
1	1	Kanal 2 freigegeben für Daten und Acknowledge
	0	
2-3		reserviert
4	1	Programmieradresse 253 freigegeben ³
	0	Programmieradresse 253 sperren
5-7		reserviert

Tabelle 8 RailCom Konfiguration.

4.2.2 CV29

Verwendung nach [RCN225].

4.2.3 CV31, CV32

Verwendung als Pointer nach [RCN225].

³ gemäß RailCommunity Beschluss v. 16./17.09.2011

4.2.4 RailCom Page

CV31 = 0 und CV32 = 255 adressieren eine Page von 256 CVs, die für RailCom-Anwendungen nach folgender Tabelle belegt sind:

Byte	Belegung	z/o	
0	Manufacturer ID (nach NMRA [S-9.2.2 Appendix A])	z	R
4-7	Produkt ID (Herstellerspezifische Produktkennung, um die einzelnen Produkte auseinanderhalten zu können. Little Endian)	o	R
8-11	MUN (Manufacturer Unique Number. Little Endian) 4 Byte Seriennummer über alle Geräte eines Herstellers.	o	R
12-15	Produktionsdatum (Anzahl der Sekunden seit dem 1.1.2000, Little Endian, Unsigned)	o	R
16-63	Herstellerspezifische Belegung möglich	o	R/W
64-127	dynamische Variablen nach APP DYN, Seite 25 72 – Behälter 1 ... 83 – Behälter 12	o	R/W
128	RailCom Versionsnummer "vor Punkt" , binär	z	R
129	RailCom Versionsnummer "nach Punkt", binär	z	R
130	Feature-Nummer für subID (siehe 5.5)	o	R
131	reserviert		
132	spezifischer Verbrauch, Behälter 1	o	R/W
133	spezifischer Verbrauch, Behälter 2	o	R/W
134	spezifischer Verbrauch, Behälter 3	o	R/W
135	spezifischer Verbrauch, Behälter 4	o	R/W
136	spezifischer Verbrauch, Behälter 5	o	R/W
137	spezifischer Verbrauch, Behälter 6	o	R/W
138	spezifischer Verbrauch, Behälter 7	o	R/W
139	spezifischer Verbrauch, Behälter 8	o	R/W
140	spezifischer Verbrauch, Behälter 9	o	R/W
141	spezifischer Verbrauch, Behälter 10	o	R/W
142	spezifischer Verbrauch, Behälter 11	o	R/W
143	spezifischer Verbrauch, Behälter 12	o	R/W
144	Füllstand aller Behälter (0...255), ein Schreibbefehl in diese CV setzt den Inhalt aller Behälter auf den angegebenen Wert.	o	W
145-255	reserviert		

Tabelle 9: Belegung Page 255

z = Zwingend, o = Optional, R = Read (Lesen), W = Write (Schreiben)

4.3 DCC-Befehle

Die erweiterte Funktionalität durch RailCom bedingt auch zusätzliche DCC-Befehle. Dazu gehören:

4.3.1 Systemkommandos

Systemkommandos werden über die Adresse 255 (0xFF) übertragen. Diese Adresse ist nach NMRA dem Leerlauf-Paket vorbehalten. Als Systemkommando wird aber das 2. Byte des Leerlauf-Paketes ungleich 0 verwendet.

4.3.2 Zusätzliche Funktionskommandos

Durch RailCom ergeben sich zusätzliche Funktionsmöglichkeiten wie z.B. Suchfunktion, Befüllen, etc. (siehe auch unter Abschnitt "Anwendungen (Application/APP) Layer für Fahrzeugdecoder"). Hierfür werden die Binärzustandssteuerungsbefehle kurze Form nach [RCN212] benutzt:

Instruktion: Binärzustandssteuerungsbefehl kurze Form ([RCN212])	
▶	1101-1101 DLLL-LLLL LLL-LLLL kennzeichnet die Funktionsnummer von 1 bis 127 LLL-LLLL = 0 schaltet alle o.g. Funktionen aus D = 1: Funktion aktiviert = 0: Funktion deaktiviert

Diese Befehle werden mit XF1 bis 127 bezeichnet.

Von diesen Befehlen sind in [RCN212] die ersten 28 für besondere Anwendungen wie RailCom reserviert. Für RailCom sind die ersten 15 Befehle vorgesehen.

XF =	Funktion:
1	Suchen
2 - 15	reserviert
16 - 28	nicht von RailCom belegt

Tabelle 10: Funktionsnummern

4.3.3 Erweiterte Programmierbefehle

Die [RCN214] kennt für die Programmierung der CVs auch noch den Konfigurationsvariablen-Zugriffsbefehl - Kurze Form Dieser Befehl ist zwei oder drei Byte lang und hat das Format **1111-KKKK DDDD-DDDD** bzw. **1111-KKKK DDDD-DDDD DDDD-DDDD** wobei **KKKK** festlegt, welche CV(s) beschrieben werden sollen.

In die so definierte(n) CV(s) werden dann die folgenden Datenbytes **DDDD-DDDD** bzw. **DDDD-DDDD DDDD-DDDD** in das bzw. geschrieben wobei bei zwei Datenbytes das erste Datenbyte in die adressierte CV geschrieben wird und das zweite Datenbyte in die darauf folgende CV geschrieben wird.

Die Zuordnung erlaubt uns, statt nur einer auch zwei CVs gleichzeitig zu beschreiben. Dies erhöht die Sicherheit und vereinfacht das Umprogrammieren z.B. der Erweiterte Adresse.

KKKK = 0000, 0010, 0011 und **1001** sind reserviert bzw. für einzelne CVs belegt.

Für den Zugriff auf zwei CVs sind folgende Codes festgelegt:

KKKK = 0100 = Schreiben von CV17 (erstes Datenbyte) und CV18 (zweites Datenbyte) (Erweiterte Adresse) gleichzeitig, und setzen des Bit 5 in CV29.

Rückmeldung erfolgt auf der alten Adresse über zwei aufeinanderfolgende Datagramme mit ID0, zuerst das erste Datenbyte, danach das zweite Datenbyte. Beide Datagramme müssen im selben Kanal 2 gesendet werden, also 12 Bit Datagramm + 12 Bit Datagramm.

KKKK = 0101 = Schreiben von CV31 (erstes Datenbyte) und CV32 (zweites Datenbyte) (Zeiger-Wert für erweiterten Bereich, höherwertiges Byte (31) und niederwertiges Byte (32)).

Rückmeldung erfolgt über zwei aufeinanderfolgende Datagramme mit ID0, zuerst das erste Datenbyte, danach das zweite Datenbyte. Beide Datagramme müssen im selben Kanal 2 gesendet werden, also 12 Bit Datagramm + 12 Bit Datagramm.

4.3.4 NOP für Zubehördecoder

Zubehördecoder melden sich nach einem Schaltbefehl, wenn sie der Zentrale etwas mitteilen wollen (SRQ).

Üblicherweise werden Schaltbefehle aber nur sporadisch gesendet, also könnten sich Zubehördecoder nur ebenso selten bemerkbar machen. Deshalb ist in [RCN213] der NOP-Befehl definiert, der regelmäßig gesendet wird, aber zunächst nichts bewirkt, außer Zubehördecodern ein SRQ zu ermöglichen.

Dieser muss von allen nicht RailCom-fähigen Einfache und Erweiterte Zubehördecodern als ungültig erkannt und damit ignoriert werden. Er dient zum einen dazu, es Zubehör-Decodern zu ermöglichen, in der nachfolgenden Austastlücke einen SRQ abzusetzen, zum anderen erlaubt er für den Fall, dass sich mehrere Decoder gleichzeitig mit einem SRQ melden, auch die Suche nach den beteiligten Decodern.

Dies wird dadurch erreicht, dass mit dem NOP eine Zubehördecoder-Adresse übertragen wird. Es melden sich daraufhin nur diejenigen Decoder, deren Adresse kleiner oder gleich ist wie die im NOP enthaltene. Dadurch kann bei Meldungen von mehreren Decodern mittels sukzessiver Approximation der Decoder mit der jeweils niedrigeren Adresse ausfindig gemacht und bedient werden. Die Suche wird solange wiederholt, bis sich kein Decoder mehr meldet.

Befehl: NOP für Zubehördecoder	
▶	1 0AA-AAAA 0AAA-1AAT A = Zubehördecoder-Adresse wie in [RCN213] definiert T = 0 Adresse eines Einfachen Zubehördecoders = 1 Adresse eines Erweiterten Zubehördecoders T wird für die Suchfunktion als höchstwertiges Adressbit betrachtet (A11)

Für die regelmäßige Abfrage aller Zubehördecoder sendet die Zentrale sinnvollerweise ein NOP mit der höchstmöglichen Adresse, damit alle Decoder angesprochen werden.

Solange sich hierauf immer nur ein Decoder mit einem SRQ meldet, kann es bei diesem NOP mit der höchsten Adresse bleiben. Erst, wenn sich mehrere Decoder gleichzeitig melden, muss die Zentrale eine Suche starten, indem sie die im NOP enthaltene Adresse geeignet variiert.

Eine RailCom-fähige Zentrale muss zur Abfrage der Zubehördecoder in bestimmten Zeitabständen einen NOP-Befehl senden. Der zeitliche Abstand zwischen zwei NOPs ist ein Kompromiss aus Bandbreiteneinschränkung beim DCC-Signal und Reaktionszeit auf SRQ-Meldungen. Es wird ein Intervall von ca. 0,5 Sekunden empfohlen.

Bei Suchvorgängen nach Mehrfachmeldungen müssen die zur Suche verwendeten NOPs natürlich schnellstmöglich hintereinander gesendet werden.

5 Anwendungen (Application/APP) Layer für Fahrzeugdecoder

Die nachfolgenden Abschnitte beschreiben die Befehle für die RailCom-Funktionalität.

Die Befehle von der Zentrale zum Decoder (►) sind jeweils ohne die Adressierungs-Daten notiert. Die Adressierung erfolgt nach dem DCC-Standard.

(◄) kennzeichnet die gesendeten RailCom-Daten.

Wenn nicht anders angegeben, beziehen sich Werte immer auf ein 8-Bit-Feld. Binäre Werte sind durch ein vorangestelltes 0b gekennzeichnet. Hexadezimale Werte sind durch ein vorangestelltes 0x gekennzeichnet.

5.1 POM

POM bedeutet "Programming On the Main", also die Programmierung auf dem Gleis zum Fahren. Es dient dem Lesen und Schreiben von Konfigurationsvariablen im Betriebsmodus entsprechend [RCN214].

Betriebsbefehle: POM	
►	1110- KKVV VVVV - VVVV DDDD - DDDD (Zugriff auf Bytes)
►	1110-10 VV VVVV - VVVV 111 K - DBBB (Zugriff auf Bits)
◄	Kanal 2: 0b (ID0) DDDD - DDDD

mit **V** = CV-Adresse, **D** = CV-Daten und **B** = Bitposition (0-7).

Wenn die CV-Adresse im Bereich 257 bis 512 ist, wird immer und automatisch die indizierte CV-Adressierung mit Hilfe von CV31 und 32 benutzt.

Folgende Modi sind definiert

KK	Beschreibung
0b	reserviert
0b	Bytes lesen
0b	Bytes schreiben
0b	Bit Zugriff: K = 0 Lesen, K = 1 Schreiben,

Tabelle 11: POM Modi

Den Zusammenhang zwischen über den Index aus CV31 und CV32 adressierten und den bisherigen 1024 CVs liefert folgende Tabelle:

Standard POM-Befehl	Indizierter POM Befehl [CV31,32]
CV{1-256}	CV{257-512} [0]
nur mit Index	CV{257-512} [1]
CV{513-768}	CV{257-512} [2]
CV{769-1024}	CV{257-512} [3]
	CV{257-512}[4]
	CV{257-512}[5]
	...
	CV{257-512} [65535]

Tabelle 12: Zusammenhang Standard POM-Befehl zu indiziertem POM-Befehl

5.1.1 Byte Lesen

Betriebsbefehl: POM Byte Lesen	
▶	1110-01VV VVVV-VVVV 0000-0000
◀	Kanal 2 (12Bit): 0b (ID0) DDDD-DDDD

mit **V** = CV-Adresse und **D** = CV-Daten.

Das zugehörige Antwortdatagramm (ID0) muss nicht im selben Paketrahmen gesendet werden, sondern kann zu einem späteren Zeitpunkt erfolgen. Eine Zentrale muss also dafür sorgen, dass der Decoder nochmals adressiert wird und vorher kein anderer Programmierbefehl gesendet wird (der gleiche ist erlaubt).

Wenn der Lesevorgang beendet ist, sendet der Decoder auf den zugehörigen, erneuten Lesebefehl das Ergebnis. Gibt der Decoder nicht innerhalb von 0,5s die Daten zurück dann gilt der Lesevorgang als fehlgeschlagen.

5.1.2 Byte Schreiben

Betriebsbefehl: POM Byte Schreiben	
▶	1110-11VV VVVV-VVVV DDDD-DDDD
◀	Kanal 2 (12Bit): 0b (ID0) hier wird der Wert zurückgeschickt, der nach der POM Operation in der CV vorhanden ist.

mit **V** = CV-Adresse und **D** = CV-Daten.

Ein Schreibbefehl darf vom Decoder nur akzeptiert werden, wenn der Befehl 2-mal gesendet wurde. Gibt der Decoder nicht innerhalb von 0,5s einen Wert zurück dann gilt der Schreibvorgang als fehlgeschlagen.

Bei einer CV, die nur gelesen werden kann, liefert der Decoder den aktuellen Wert der CV.

5.1.3 Bit Schreiben

Betriebsbefehl: POM Bit Schreiben	
▶	1110-10VV VVVV-VVVV 1111-DBBB
◀	Kanal 2 (12Bit): 0b (ID0) DDDD-DDDD

mit **V** = CV-Adresse, **D** = Bitwert und **B** = Bitposition (0-7), **DDDD-DDDD** sind die rückgelesene CV-Daten

Ein Schreibbefehl darf vom Decoder nur akzeptiert werden, wenn der Befehl 2-mal gesendet wurde.

Antworten wie bei "Byte Schreiben". Der Befehl Bit lesen ist für RailCom uninteressant, da immer das ganze Byte zurückgemeldet wird.

5.2 ADR

Diese Rückmeldung dient der Adressfeststellung von Fahrzeugdecodern auf der Anlage. Mit Hilfe von ortsfesten Detectoren kann sie zur Lokalisierung verwendet werden.

Fahrzeugdecoder nutzen Kanal 1 als Broadcast Kanal für die eigene Adresse. Im Cutout nach jedem an einen Fahrzeugdecoder gerichteten DCC Paket senden sie ihre aktive Adresse (Basis, Erweiterte oder Mehrfachtraktion). Hierfür werden die folgenden 12-Bit Datagramme mit ID1 und ID2 definiert:

ADR1 (ID1)	ADR2(ID2)	Adresse
0 0 0 0 0 0 0 0	0 A6 A5 A4 A3 A2 A1 A0	Basis-Adresse (CV1)
0 1 1 0 0 0 0 0	R A6 A5 A4 A3 A2 A1 A0	Mehrfachtraktionsadresse (CV19)
1 0 A13 A12 A11 A10 A9 A8	A7 A6 A5 A4 A3 A2 A1 A0	Erweiterte Adresse (CV17+CV18)

Tabelle 13: ADR-Zuordnung Adresse

Ein Decoder muss abhängig von seiner aktiven Adresse die beiden Datagramme ADR1 und ADR2 alternierend senden. Die "aktive Adresse" ist die, unter der der Decoder seine Fahrbefehle erhält.

Betriebsbefehls-Antwort: ADR	
▶	Betriebsbefehl an Decoderadresse
◀	Kanal 1: 0b (ID1) ADR 1 / 0b (ID2) ADR 2

Mögliche Anwendung:

Lokale Detectoren (Local Detector) auf der Anlage werten diese Datagramme zeitnah aus und senden diese Informationen weiter (Zentrale, Computer etc.).

Auf diese Weise kann die Zentrale erfahren, welcher Decoder sich auf einem bestimmten Gleisabschnitt befindet. Abbildung 5 zeigt dies schematisch.

Dieses Verfahren eignet sich besonders zur Zugbeeinflussung.

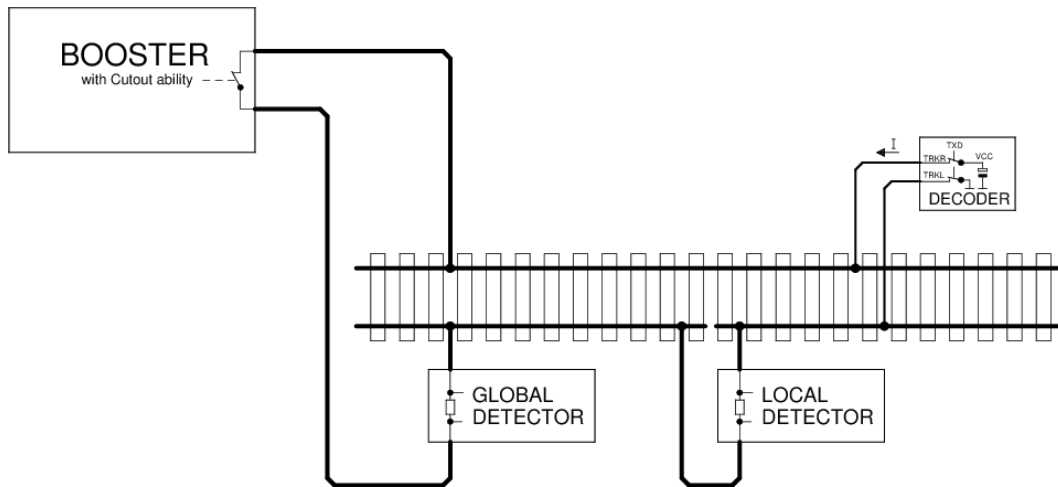


Bild 5: Lokalisierung Fahrzeugdecoder

Die Lokalisierung eines Decoders nach obigem Muster kann nur funktionieren, wenn der Decoder sich alleine auf dem vom Local Detector überwachten Gleisabschnitt befindet. Problematisch ist dies jedoch im Multitraktionsbetrieb. Hierfür wird empfohlen, dass nur die führende Lok die ADR-Datagramme auf Kanal 1 sendet, während bei den folgenden Loks diese Funktion via CV28 deaktiviert ist. Dies kann beim Zusammenstellen des Fahrzeugverbands via POM geschehen.

5.3 EXT

Diese Rückmeldung dient der Übertragung einer Ortsinformation. Damit kann der Ort des Fahrzeugdecoders einer gegebenen Adresse festgestellt und ggf. ein Befüllen von Vorräten ausgelöst werden.

5.3.1 Senden der Ortsinformation:

Die Ortsinformation kann durch den Decoder oder den Detector gesendet werden, je nachdem, wo die Information vorliegt.

Fall 1: Ortsinformation liegt im Decoder vor (z.B. durch Infrarotübertragung).

Fall 2: Ortsinformation liegt im Detector vor.

Betriebsbefehls-Antwort: EXT			
▶	Befehl 1101-1101 0000-0001 = Binärzustandssteuerungsbefehl "XF1 aus" an Fahrzeugdecoder		
	Fall 1: Ortsinformation liegt im Decoder vor (z.B. durch Infrarotübertragung)		
◀	Kanal 2 von Decoder 0b 0011 (ID3) 0b 00 0bZZZZ-0000-0000 ZZZZ = 0000 – 0111 : Ortsinformation ZZZZ = 1000 – 1111 : reserviert ZZZ-0000-0000 : 11 Bit Ortsadresse		
	Fall 2: Ortsinformation liegt im Detector vor		
◀	<table border="0"> <tr> <td style="vertical-align: top;">Kanal 2 von Decoder: 0b0011 (ID3) 0b01</td> <td style="vertical-align: top;">Kanal 2 von Detector: 0bTTTT-0000-0000 TTTT = 0000 – 0111: nur Ortsinformation TTTT = 1000: reserviert TTTT = 1001: reserviert TTTT = 1010: Dieseltankstelle TTTT = 1011: Kohlebansen TTTT = 1100: Wasserkran TTTT = 1101: Besandungsanlage TTTT = 1110: Ladestation (Akku) TTTT = 1111: Allgemeine Füllstation 0000-0000: 8 Bit Ortsadresse</td> </tr> </table>	Kanal 2 von Decoder: 0b 0011 (ID3) 0b 01	Kanal 2 von Detector: 0b TTTT-0000-0000 TTTT = 0000 – 0111 : nur Ortsinformation TTTT = 1000 : reserviert TTTT = 1001 : reserviert TTTT = 1010 : Dieseltankstelle TTTT = 1011 : Kohlebansen TTTT = 1100 : Wasserkran TTTT = 1101 : Besandungsanlage TTTT = 1110 : Ladestation (Akku) TTTT = 1111 : Allgemeine Füllstation 0000-0000 : 8 Bit Ortsadresse
Kanal 2 von Decoder: 0b 0011 (ID3) 0b 01	Kanal 2 von Detector: 0b TTTT-0000-0000 TTTT = 0000 – 0111 : nur Ortsinformation TTTT = 1000 : reserviert TTTT = 1001 : reserviert TTTT = 1010 : Dieseltankstelle TTTT = 1011 : Kohlebansen TTTT = 1100 : Wasserkran TTTT = 1101 : Besandungsanlage TTTT = 1110 : Ladestation (Akku) TTTT = 1111 : Allgemeine Füllstation 0000-0000 : 8 Bit Ortsadresse		

Der Detector funktioniert wie in Abbildung 5, wird aber um einen RailCom-Sender ergänzt. Da die Ortsinformation über das Gleis zurück gesendet wird, ist kein weiterer Anschluss an z.B. ein Bussystem notwendig.

5.3.2 Befüllen

Das Befüllen eines Behälters oder aller Behälter erfolgt mit dem Befehl "Byte Schreiben", Abschnitt 5.1.2, auf die entsprechenden CVs in der RailCom-Page (CV31=0 und CV32=255).

5.4 DYN

Diese Rückmeldung dient der Übertragung dynamischer Informationen von Fahrzeugdecodern. Unter "Dynamischen Informationen" werden CV-Inhalte (RailCom-CVs 64-127) verstanden, die sich während des Betriebes ändern (z.B. Geschwindigkeit, Tankinhalt etc.)

Betriebsbefehls-Antwort: DYN	
▶	Betriebsbefehl an Decoderadresse
◀	Kanal 2 18Bit [+ 18Bit]: 0b 0111 (ID7) 0b DDDD-DDDD-XXXX-XX [0b 0111 (ID7) 0b DDDD-DDDD-XXXX-XX]

Die Übertragung von dynamischen Variablen (DV) (z.B. Geschwindigkeit, Behälter, ...) erfolgt in einem 18 Bit Datagramm (ID7), welches den 8 Bit Wert der DV (D) sowie einen 6 Bit Subindex (X) enthält, der eine von 64 möglichen DVs selektiert. Die Bedeutung der DV ist durch den Subindex festgelegt.

2 beliebige DVs können in einem Rückmeldeframe übertragen werden. Welche DVs ein Decoder wann sendet bestimmt er selbst.

X	Bedeutung
0	Echte Geschwindigkeit Teil 1 * Es werden Werte zwischen 0 und 255 vom Decoder zurück gesendet. 255 ist maximale Geschwindigkeit. Die Angaben sind in km/h (oder m/h). Bei höheren Geschwindigkeiten als 255 wird in Teil 2 die Differenz abgelegt.
1	Echte Geschwindigkeit, Teil 2 *
2	reserviert
3	reserviert
4	reserviert
5	Flag Register, Inhalte noch festzulegen
6	Input Register, Belegung noch festzulegen
7	Empfangsstatistik: Der Fahrzeugdecoder führt eine Statistik über alle empfangenen DCC-Pakete und übermittelt die Zahl fehlerhafter Pakete / Gesamtzahl in %. (Wertebereich 0-100).
8	Inhalt des Behälter 1 in % (Wertebereich 0-100)
9	Inhalt des Behälter 2 in % (Wertebereich 0-100)
10	Inhalt des Behälter 3 in % (Wertebereich 0-100)
11	Inhalt des Behälter 4 in % (Wertebereich 0-100)
12	Inhalt des Behälter 5 in % (Wertebereich 0-100)
13	Inhalt des Behälter 6 in % (Wertebereich 0-100)
14	Inhalt des Behälter 7 in % (Wertebereich 0-100)
15	Inhalt des Behälter 8 in % (Wertebereich 0-100)
16	Inhalt des Behälter 9 in % (Wertebereich 0-100)
17	Inhalt des Behälter 10 in % (Wertebereich 0-100)
18	Inhalt des Behälter 11 in % (Wertebereich 0-100)
19	Inhalt des Behälter 12 in % (Wertebereich 0-100)
20-63	reserviert

Tabelle 14: Dynamische Informationen Fahrzeugdecoder

* Hinweis zur Übertragung der echten Geschwindigkeit:

Der Decoder berechnet z.B. die wirklich auf der Anlage gefahrene Geschwindigkeit. Diese wird mit dem Maßstab multipliziert, der im Decoder einstellbar ist (z.B. für H0 der Wert 87). Ergebnis ist die echte Geschwindigkeit. Sie wird als km/h übertragen. Bis zum Wert 255 erfolgt die Übertragung in ID 7 DYN0. Ist die Geschwindigkeit höher, wird ausschließlich(!) die Differenz (also berechnete Geschwindigkeit in km/h minus 256) in ID 7 DYN 1 übertragen. ID 7 DYN 0 wird dann nicht übertragen, da es keine relevanten Informationen mehr enthält.

5.5 subID

Die subID gibt einzelnen Herstellern die Möglichkeit, neue RailCom Features zu entwickeln, zu testen und in Produkte zu implementieren, die nicht mit den in der Spezifikation vorhandenen Befehlssatz zu realisieren sind.

Dazu bekommt der Hersteller von der RailCommunity) eine Nummer zugeteilt, die in die CV130 der RailCom Page einzutragen ist. Durch diese Nummer lassen sich verschiedene Applikationen voneinander unterscheiden. Voreinstellung der CV130 ist "0".

Der Hersteller verpflichtet sich der RailCommunity das Entwicklungsziel zu nennen und bei Fertigstellung der Entwicklung eine detaillierte Beschreibung (wie in dieser Spezifikation für die anderen IDs praktiziert) zur Verfügung zu stellen.

Die RailCommunity wird die Nummernvergabe und die Ziele, wie auch die Beschreibung des fertigen Produktes in Absprache mit dem Hersteller (spätestens zur Markteinführung) publizieren. Dadurch wird es anderen Herstellern ermöglicht, diese Entwicklung ebenfalls zu nutzen.

5.5.1 Liste der bisher vergebenen subIDs

In dieser Liste werden die vergebenen SubIDs mit dem jeweiligen Verwendungszweck aufgeführt.

SubID (dec)	Zweck	erteilt an / am
0	Voreinstellung	
10	Synchronisierung Fahrzeug-Sounddecoder mit Sounderzeugern auf der Modellbahn	28.03.2013

Tabelle 15: subIDs

6 Anwendung für Zubehördecoder (Weichen etc.)

Die Befehle von der Zentrale zum Decoder (→) sind jeweils ohne die Adressierungs-Daten notiert. Die Adressierung erfolgt nach dem DCC-Standard.

6.1 SRQ - Service Request für Zubehördecoder

Der Kanal 1 des Cutouts wird von Zubehördecodern benutzt, um die Zentrale zu einer Kommunikation zu veranlassen. Diese Aufforderung wird im Folgenden als SRQ (Service Request) bezeichnet.

Der SRQ kann entweder nach einem beliebigen Zubehördecoder-Paket gesendet werden (gleichgültig welche Adresse damit angesprochen wird und gleichgültig, ob es ein Einfaches oder ein Erweitertes Zubehörsteuerungspaket ist) oder nach einem NOP, wenn die eigene Adresse kleiner oder gleich der im NOP enthaltenen ist. (Vgl. Kapitel 4.3.4)

Bei einem SRQ nach einem NOP ist die entsprechende Meldung in Kanal 2 gleich mitzusenden, um etwas Zeit zu sparen. Erfolgt der SRQ jedoch nach einem regulären Zubehördecoder-Befehl, so darf die Meldung nicht gesendet werden, damit die Meldungen des adressierten Decoders nicht unkenntlich gemacht werden.

Beim SRQ handelt es sich um ein 12-Bit-Datagramm. Im Gegensatz zu allen anderen Datagrammen enthält der SRQ keine Kennung, vielmehr werden die 12 Nutzbits zur Übertragung der kompletten Zubehöradresse herangezogen.

Bei den Betriebsbefehlen für Zubehördecoder sind Adresse und Daten so kombiniert, dass die Adresse Teil des angegebenen Befehls ist. Daher wird hier das gesamte Befehlspaket (aber ohne Präambel und Prüfbyte) dargestellt

Betriebsbefehl: Einfache Zubehörsteuerung	
▶	10AA-AAAA 1AAA-DAAR
oder Betriebsbefehl: Erweiterte Zubehörsteuerung	
▶	10AA-AAAA 0AAA-0AA1 DDDD-DDDD
oder Betriebsbefehl: NOP	
▶	10AA-AAAA 0AAA-1AAT
◀	Kanal 1: 0b 0 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0 (Einfache Zubehördecoder)
◀	Kanal 1: 0b 1 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0 (Erweiterte Zubehördecoder)
◀	Kanal 2: den SRQ betreffende Meldung, z. B. Fehler (nur bei Antwort auf NOP)

R ist die Auswahl des Ausgangs innerhalb einer Weichenadresse. **D** besagt, ob der Ausgang ein-(**1**) oder aus-(**0**) geschaltet werden soll Die Bitwertigkeiten innerhalb des DCC-Befehls sind wie in [RCN213] definiert.

Durch die Beschränkung der Adressauflösung bei Einfachen Zubehördecodern auf Ausgangspaare erhält man ebenso wie bei den Erweiterten Zubehördecodern eine 11-Bit-Adresse und kann dadurch mit dem 12. Bit des SRQ-Datagramms zwischen diesen beiden Kategorien unterscheiden.

Hat ein Decoder einen SRQ abgesetzt, muss er diesen wiederholen, bis er behandelt wurde. Der Decoder darf in dieser Zeit auf keine an sich gerichteten Stellbefehle reagieren.

Ein SRQ gilt als behandelt, wenn der Decoder auf der eigenen Adresse einen Löschbefehl erhalten hat. Als Löschbefehl dient der "Coil Off"-Befehl bzw. der "absolute stop"-Befehl. In diesem Zustand werden die genannten Befehle als solche nicht ausgeführt, sondern bewirken lediglich, dass der SRQ nicht mehr gesendet wird.

Löschbefehl für Einfache Zubehördecoder: Coil off

Format: **10AA-AAAA 1AAA-0AA0**

Löschbefehl für Erweiterte Zubehördecoder: Absolute stop (=Aspect 0)

Format: **10AA-AAAA 0AAA-0AA1 0000-000**

Hinweis: Eine RailCom-fähige Zentrale sendet regelmäßig einen NOP, um SRQs zu ermöglichen. Empfängt ein Zubehördecoder in den ersten 5 Sekunden nach dem ersten Empfang des DCC-Formates keine NOPs, so darf er davon ausgehen, dass die Zentrale nicht RailCom-fähig ist, somit keine SRQs bearbeitet werden können und er auch keine senden braucht. In diesem Fall wird die Funktion des Decoders nicht blockiert.

6.2 POM

POM bedeutet "Programming On the Main", also die Programmierung auf dem Gleis zum Fahren. Da Zubehördecoder meistens fest mit dem Digitalsignal für den Betrieb verbunden sind erlauben diese Befehle das Lesen und Schreiben von Konfigurationsvariablen im Betriebsmodus entsprechend [RCN214].

Bei diesem Befehlen wird die Adresse nicht mit angegeben und bezüglich der Adressierung auf [RCN214] verwiesen.

Betriebsbefehle: POM	
▶	1110-KKVV VVVV-VVVV DDDD-DDDD (Zugriff auf Bytes)
▶	1110-10VV VVVV-VVVV 111K-DBBB (Zugriff auf Bits)
◀	Kanal 2: 0b (ID0) DDDD-DDDD

mit **V** = CV-Adresse, **D** = CV-Daten und **B** = Bitposition (0-7).

Wenn die CV-Adresse im Bereich 257 bis 512 ist, wird immer und automatisch die indizierte CV-Adressierung mit Hilfe von CV31 und 32 benutzt.

Folgende Modi sind definiert

KK	Beschreibung
0b	reserviert
0b	Bytes lesen
0b	Bytes schreiben
0b	Bit Zugriff: K = 0 Lesen, K = 1 Schreiben,

Tabelle 11: POM Modi

Den Zusammenhang zwischen über den Index aus CV31 und CV32 adressierten und den bisherigen 1024 CVs liefert folgende Tabelle:

Standard POM-Befehl	Indizierter POM Befehl [CV31,32]
CV{1-256}	CV{257-512} [0]
nur mit Index	CV{257-512} [1]
CV{513-768}	CV{257-512} [2]
CV{769-1024}	CV{257-512} [3]
	CV{257-512}[4]
	CV{257-512}[5]
	...
	CV{257-512} [65535]

Tabelle 16: Zusammenhang Standard POM-Befehl zu indiziertem POM-Befehl

6.2.1 Byte Lesen

Betriebsbefehl: POM Byte Lesen	
▶	1110-01VV VVVV-VVVV 0000-0000
▶	0000} (ID0) DDDD-DDDD

mit **V** = CV-Adresse und **D** = CV-Daten.

Das zugehörige Antwortdatagramm (ID0) muss nicht im selben Paketrahmen gesendet werden, sondern kann zu einem späteren Zeitpunkt erfolgen. Eine Zentrale muss also dafür sorgen, dass der Decoder nochmals adressiert wird und kein anderer Befehl an diesen Decoder gesendet wird (der gleiche ist erlaubt).

Wenn der Lesevorgang beendet ist, sendet der Decoder auf den zugehörigen, erneuten Lesebefehl das Ergebnis. Gibt der Decoder nicht innerhalb von 0,5s die Daten zurück dann gilt der Lesevorgang als fehlgeschlagen.

6.2.2 Byte Schreiben

Betriebsbefehl: POM Byte Schreiben	
▶	1110-11VV VVVV-VVVV DDDD-DDDD
◀	Kanal 2 (12Bit): 0b 0000 (ID0) DDDD-DDDD hier wird der Wert zurückgeschickt, der NACH der POM Operation in der CV vorhanden ist.

mit **V** = CV-Adresse und **D** = CV-Daten.

Ein Schreibbefehl darf vom Decoder nur akzeptiert werden, wenn der Befehl 2-mal gesendet wurde. Gibt der Decoder nicht innerhalb von 0,5s die Daten dann gilt der Schreibvorgang als fehlgeschlagen.

Bei einer CV, die nur gelesen werden kann, liefert der Decoder den aktuellen Wert der CV.

6.2.3 Bit Schreiben

Betriebsbefehl: POM Bit Schreiben	
▶	1110-10VV VVVV-VVVV 1111-DBBB
▶	Kanal 2 (12Bit): 0b 0000 (ID0) DDDD-DDDD

mit **V** = CV-Adresse, **D** = Bitwert und **B** = Bitposition (0-7), **DDDD-DDDD** sind die rückgelesene CV-Daten

Ein Schreibbefehl darf vom Decoder nur akzeptiert werden, wenn der Befehl 2-mal gesendet wurde.

Antworten wie bei "Byte Schreiben".

6.3 STAT1

Diese Rückmeldung dient der Übertragung von Statusnachrichten von Zubehördecodern, Teil 1

Betriebsbefehls-Antwort: STAT1	
▶	Betriebsbefehl an Decoderadresse
◀	Kanal 2 12Bit: 0b 0100 (ID4) 0b DDDD-DDDD

Der Statuscode kann nach Zubehördecoderbefehlen als Quittung zurückgesendet werden. Auf ein "ACK" kann dann verzichtet werden.

1. Datagramm

Bit	Bedeutung
11 ... 8	Identifizier 0x4
7	reserviert
6	0: Ausgangszustand stimmt nicht mit dem letzten empfangenen Befehl überein. 1: Ausgangszustand stimmt mit dem letzten empfangenen Befehl überein.
5	0: der zurückgemeldete Aspect ist der Sollwert 1: der zurückgemeldete Aspect ist der Istwert anhand echter Rückmeldung
4 ... 0	Aktueller Aspect. Z.B. Weichendecoder haben 2 Ausgangszustände, Signaldecoder (sog. Erweiterte Zubehördecoder) bis zu 31

Tabelle 17: Statusmeldungen, Teil 1

6.4 ZEIT

Diese Rückmeldung dient der Übertragung der prognostizierten Umlaufzeit

Betriebsbefehls-Antwort: ZEIT	
▶	Betriebsbefehl an Decoderadresse
◀	Kanal 2 12Bit: 0b 0101 (ID5) 0b DDDD-DDDD

Diese Befehlsbestätigung kann nach Zubehördecoderbefehlen als Quittung zurückgesendet werden. Auf ein "ACK" kann dann verzichtet werden.

Die 7 niederwertigen Bits der Restlaufzeit kennzeichnen die Laufzeit bis zum Erreichen des Ende-Zustands dieses Begriffes (prognostizierte Umlaufzeit). Die Zeit wird abhängig vom MSB in 1/10 Sekunden (MSB = 0) oder 1 Sekunde (MSB = 1) angegeben. Eine Zeit von 0 bedeutet keine Schaltzeit - z.B. bei Signaldecodern ohne Glühlampensimulation. Damit ergibt sich ein Wertebereich 0 ... 12,7 Sekunden bzw. 0 ... 127 Sekunden.

1. Datagramm

Bit	Bedeutung
11 ... 8	Identifizier 0x5
7	0: Auflösung 1/10 Sekunde 1: Auflösung 1 Sekunde
6 ... 0	Prognostizierte Umlaufzeit

Tabelle 18: prognostizierte Umlaufzeit

6.5 FEHLER

Diese Rückmeldung dient der Übertragung von Fehlerinformationen

Betriebsbefehls-Antwort: FEHLER	
▶	Betriebsbefehl an Decoderadresse (einschließlich NOP)
◀	Kanal 2 12Bit: 0b 0110 (ID6) 0b DDDD-DDDD

Der Fehlercode kann nach jedem Befehl, welcher den Decoder adressiert, zurückgesendet werden, einschließlich NOP. Auf ein "ACK" kann dann verzichtet werden.

Die Fehlermeldungsübertragung wird durch einen erneuten Schaltbefehl gelöscht. Liegt ein permanenter Fehler vor, darf der Decoder keinen neuen SRQ auslösen, solange der selbe Fehler vorliegt. Die Zentrale muss deshalb nach der Löschung den Decoder erneut ansprechen um festzustellen, ob es sich um einen permanenten Fehler handelt.

Bit	Bedeutung	
11 ... 8	Identifizier 0x6	
7	reserviert	
6	0: Es liegt nur der in den folgenden 6 Bits angegebene Fehler vor. 1: Es liegen neben dem angegebenen noch weitere Fehler vor.	
5 ... 0	Fehlercode	Bedeutung
	0x00	Kein Fehler (mehr)
	0x01	Befehl konnte nicht ausgeführt werden, unbekannter Befehl / ungültiger Aspect.
	0x02	Stromaufnahme des Antriebs zu hoch.
	0x03	Versorgungsspannung zu gering, die Funktion ist nicht sichergestellt.
	0x04	Sicherung defekt.
	0x05	Temperatur zu hoch.
	0x06	Rückmeldefehler (ungewollte Verstellung festgestellt)
	0x07	Handverstellung (z.B. per Taster am Decoder)
	0x10	Weichenlaterne oder Signallaterne defekt
	0x20	Servo defekt.
0x3F	Interner Decoderfehler, z.B. Selbsttest Prozessor Prüfsumme fehlerhaft.	

Tabelle 19: Fehlermeldungen

6.6 DYN

Diese Rückmeldung dient der Übertragung dynamischer Informationen von Zubehördecodern. Unter "Dynamischen Informationen" werden CV-Inhalte (RailCom-CVs) verstanden, die sich während des Betriebes ändern.

Betriebsbefehls-Antwort: DYN	
▶	Betriebsbefehl an Decoderadresse
◀	Kanal 2 18Bit+[18Bit]: 0b 0111 (ID7) 0b DDDD-DDDD-XXXX-XX [0b 0111 (ID7) 0b DDDD-DDDD-XXXX-XX]

Die Übertragung von dynamischen Variablen (DV) erfolgt in einem 18 Bit Datagramm (ID7), welches den 8 Bit Wert der DV (D) sowie einen 6 Bit Subindex (X) enthält, der eine von 64 möglichen DVs selektiert. Die Bedeutung der DV ist durch den Subindex festgelegt.

2 beliebige DVs können in einem Rückmeldeframe übertragen werden. Welche DVs ein Decoder wann sendet, bestimmt er selbst.

X	Bedeutung
0	Flag Register, Inhalte noch festzulegen
1-63	reserviert

Tabelle 20: Dynamische Informationen

6.7 STAT2

Dient nicht für neue Designs, wird aber von auf dem Markt befindlichen Komponenten schon benutzt. Bei anderer Nutzung der ID8 wird die Länge des Datagrams als Unterscheidungsmerkmal verwendet.

Diese Rückmeldung dient der Übertragung von Statusnachrichten von Zubehördecodern Teil 2

Betriebsbefehls-Antwort: STAT2	
▶	Betriebsbefehl an Decoderadresse
◀	Kanal 2 12Bit: 0b 1000 (ID8) 0b DDDD-DDDD

Speziell zugeschnitten für mechanische Stellvorgänge.

1. Datagramm

Bit	Bedeutung
11 ... 8	Identifizier 0x8
7 ... 4	Konfiguration, bisher festgelegt: 0000 — Entkupppler 0001 — Weiche 0010 — Dreiwegweiche 0011 — Doppel-Kreuzungsweiche 1000 — Gleisperrsignal 1001 — Formsignal Hp0/Hp1 1010 — Formsignal Hp0/Hp1/Hp2 1011 — Vorsignal Vr0/Vr1 1100 — Vorsignal Vr0/Vr1/Vr2 1101 — Bahnschranke
3	0: Der in den Bits 2 ... 0 gemeldete Zustand entspricht dem Sollwert, bzw. "Stellvorgang dauert noch an". 1: Der gemeldete Zustand entspricht dem Istwert anhand echter Rückmeldung.
2 ... 0	Aktueller Zustand

Tabelle 21: Statusmeldungen, Teil 1

Anhang A: Verweise auf andere Normen

A.1 Normative Verweise

Um diese Norm zu erfüllen, müssen keine anderen Normen eingehalten werden.

A.2 Informative Verweise

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

[RCN211] [RCN-211](#) DCC Paketstruktur

[RCN212] [RCN-212](#) DCC Betriebsbefehle für Fahrzeugdecoder

[RCN213] [RCN-213](#) DCC Betriebsbefehle für Zubehördecoder

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

[RCN225] [RCN-225](#) DCC Konfigurationsvariablen

[S922AA] NMRA: [S-9.2.2 Appendix A](#) DCC Manufacturer ID codes

Anhang B: Historie

Datum	Änderungen gegenüber der jeweils vorhergehenden Version	Version
18.12.2016	Erste Version basierend auf Spezifikation RailCom V 1.4 vom April 2015 der Fa. Lenz	1.0

Copyright 2016 RailCommunity – Verband der Hersteller Digitaler Modellbahnprodukte e.V.