

Location dependent information transmission from the rail to mobile decoders

by means of ZIMO's „signal controlled speed influence“
and “location dependent function control”

and it's use together with “bi-directional communication” by RP NMRA 9.3.1, 9.3.2

This paper is only a description on principle, *not* a specification or a RP draft !

GENERAL:

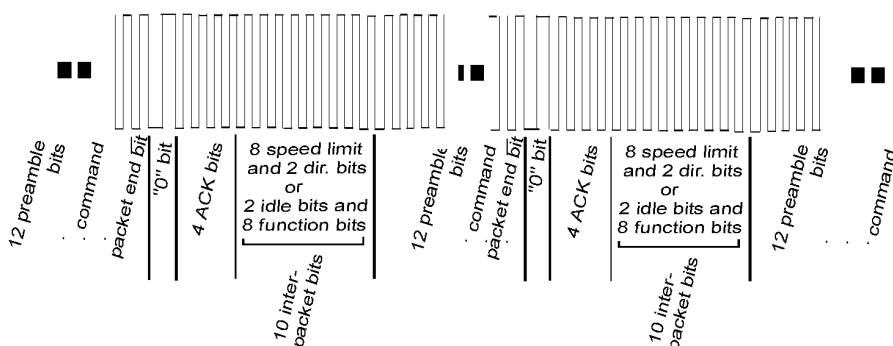
ZIMO systems offer already for many years the special features „signal controlled speed influence“ and „location dependent function control“. Both are applications of a general method transmitting information from isolated track sections to the loco or function decoders. In contrast to the DCC packets this information is not addressed to certain DCC decoder addresses, but is to be evaluated from all decoders which just stay on the track section. The coding of information is done by **cut-out of interpacket bits or parts of interpacket bits**; the so-called interpacket bits are special additional DCC 1-bits (short bits) between packet end and the DCC-conformant 14-bit-preamble, which are produced by the command station (simply by making not 14 preamble-bits, but e.g. 26 preamble bits).

Now we use this existing information channel (details below !) additionally (besides speed influence and function control as before) for transmitting “**position codes**” in order to inform the decoder(s) about their current location (= the track section or a group of track sections defined by a position-code). This information could be used directly by the decoder (by some internal operations, when the decoder is programmable to operations like stopping or going backward after receiving a certain position-code) or – for more general operation – is sent from the decoder by “bi-directional communication” to the “bi-directional detector” (global type, inside or near the command station), most useful (but not necessarily) together with a “routing-code”, stored in the decoder.

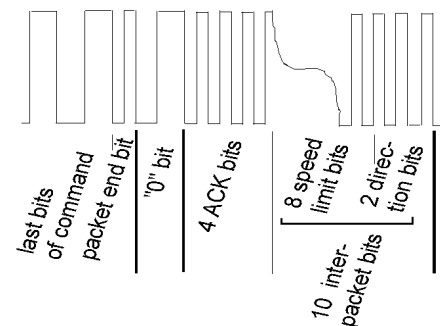
THE EXISTING ZIMO PROTOCOL (10 interpacket bits):

There are 10 interpacket bits (the 4 ACK bits in the drawing are not related to this), which could be cut out bit by bit; this is done from a ZIMO module, which is called “track section module MX9”, in most cases under control of the command station (“system-autonomous”) or under control of a computer software (STP). This protocol is used since ZIMO makes DCC products (1997); a NMRA TN (Technical Note) exists for this protocol.

Details (as it is in the actual ZIMO implementation): Cutting-out 1 .. 5 successive bits beginning with the first interpacket bits means **reduced speed** (various speed limits), cutting-out 7 successive bits means “**stop**” to the decoder. Alternatively cutting-out bits *not* beginning with the first or second interpacket bit (but with the third or fourth, etc.) means **function control** or **position code** (255 possible codes – if one bit is used for function control, there are only 127 codes; if there are two function control bits, 63 codes, etc.). Currently the MX9 makes always 9 times the speed limit code and then 1 time the function/position code.



Example: cut-out of some interpacket bits

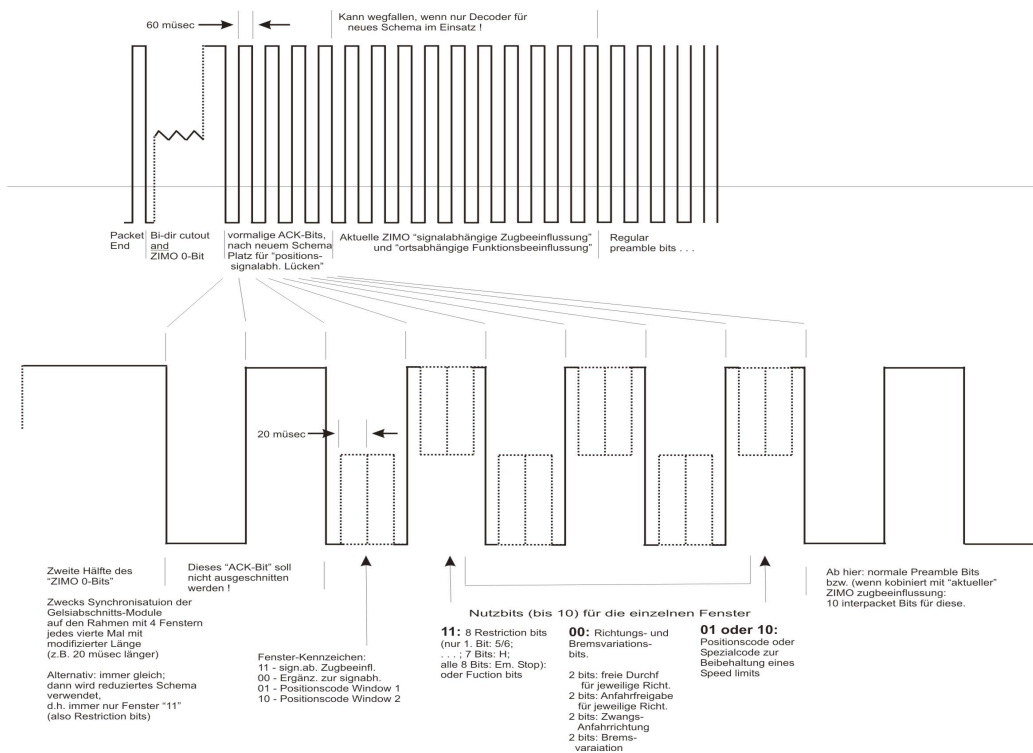


THE FUTURE ZIMO PROTOCOL (long form - 4 interpacket bits):

The existing protocol (preceding chapter) works well, but the today state-of-the-art microcontrollers allow to develop a new protocol, which needs less bits (for a better DCC bandwidth) and offers more capacity at the same time.

The drawing below shows it together with the existing protocol, but of course the existing protocol can be left, if no "old" decoders are on the track. The new (future protocol) allows to transmit

- speed limits (5 steps, stop, emergency stop), very similar as with the existing protocol,
- function bits for control of up to 8 function outputs, similar to existing protocol,
- additional information for speed limits and stopping (speed limit or stop only for one direction, starting allowed in only one direction, forcing direction change, special instructions for braking behaviour), much more possibilities than in the existing protocol,
- Position code window 1 and
- Position code window 2 (we use 2 windows for position code in order to minimize the problem of bridging isolation gaps); in contrast to the existing protocol non overlapping with function control.



THE FUTURE ZIMO PROTOCOL (short form - 1 interpacket bit):

This is a proposal to use this method also with command stations, which are not able to produce additional preamble bits for use as interpacket bits. It seems that every command station today makes at least 15 preamble bits, so one of these (the first) could be used as interpacket bit.

In this one bit there would be 4 cut-out positions, so stop and 2 speed limits could be transmitted.

USE of “SIGNAL CONTROLLED SPEED INFLUENCE” and POSITION CODES together with BI-DIRECTIONAL COMMUNICATION by NMRA RP 9.3.1, 9.3.2

This is a way, to implement block control, automatic routing, station control, staging yards, etc. by already existing means together with the “bi-directional communication” according to NMRA RP drafts 9.3.1, 9.3.2.

The necessary hardware is: a “global bi-directional detector” (*no* “local detectors necessary), and a product like the ZIMO track section module MX9 or the future (in development) HLU track decoder. Both (the MX9 and the HLU track decoder) are devices, which have some track section outputs which transmit speed limits and position codes by means of cut-out of preamble bits (as explained above).

Typical way of use:

The train arrives at a red signal in front of a group of turnouts (the entrance of a station); it is stopped by “signal controlled speed influence”, The track section (which transmits the stop information to the loco decoder) also transmits its position code to the loco decoder. The loco decoder transmits this position code together with its own routing code (the content of a new CV in the decoder) with “bi-directional communication” to the global detector (which is in the command station or between command station and track). The command station gets the information from the global detector and decides based on the combination of position code and routing code (according to pre-defined user orders): e.g. choosing a certain track (switching turnouts), make a signal green, releases the train (= removing the speed limit “stop”). Of course also a computer connected to the command station can handle the proceedings.

MORE DETAILED SPECIFICATION FOR THE FUTURE ZIMO PROTOCOL currently only in German available !

Die Codierung der vom Gleisabschnitt zu den Fahrzeugen erfolgt durch „Cutouts“ („Lücken“), die vom „Gleisabschnitts-Modul“ (o.ä.) in das DCC-Schienensignal geschnitten werden. Als Platz stehen dafür 4 1-Bits (also je 120 – oder normmäßig genauer – 116 müsec, also insgesamt ca. 0,5 ms) zur Verfügung, welche nach dem Packet-End-Bit und dem „ZIMO 0-Bit“ (in welche geg.falls das Cutout der „bi-directional communication eingebettet ist) zu Verfügung. Es handelt sich bei diesem Platz um die bisherigen „ZIMO ACK-Bits“ (laut NMRA TI-9.2.1), welche nicht benötigt werden, wenn deren Funktion der Zugnummernerkennung durch die „bi-directional communication“ wahrgenommen wird.

Für ZIMO-eigene Anwendungen in der Übergangsphase ist jedoch auch Kombination möglich, weil derzeit ZIMO in der Praxis nur das erste ACK-Bit für ACKs intensiv benützt; deswegen ist diese bei dem vorgeschlagenen Schema für die neuen Cutouts ausgelassen. Dieses Nicht Verwenden des ersten ACK-Bits ist außerdem gut für den Decoder-Kondensator (der noch von der vorangehenden „bi-dir“ Lücke ausgesaugt sein könnte)

In diesen ehemaligen ACK-Bits gibt es 12 Cutout-Plätze (siehe Bild), welche jeweils durch nicht ausschneidbare Flanken (20 müsec Bereiche) von einander getrennt sind. Diese fixen Flanken zwischen den Fenstern dienen zu Nach-Synchronisation der Decoder und zur sicheren Nachladung des Decoder-Kondensators. 4 aufeinanderfolgende Interpacket-Phasen bilden zusammen einen Rahmen, in dem 4 verschiedene Fenster übertragen werden können.

Diese 4 Fenster müssen jedoch von allen „Gleisabschnitts-Modulen“ (bei ZIMO „MX9“) synchron gesendet werden (damit bei Überbrücken der Gleistrennungen immer die zusammenpassenden Fenster überlagern, insbesondere für die Speed limits wichtig). Diese kann geschehen, indem das zweite Halbit des „ZIMO 0-Bits“ jedes 4. Mal etwas länger gemacht wird.

Eine „Spar-Version“ dieses Schemas könnte darin bestehen, dass nur das Fenster „11“ verwendet wird (also keine Positionscodes und kein komplexeres Richtung- und Brems-Handling); dort kann die „MX9-Synchronisation“ wegfallen.

Das **Fenster „11“** dieses Schema entspricht der heutigen „ZIMO signalabhängigen Zugbeeinflussung“ (nur eben effizienter übertragen – 12 mal so schnell), also 8 Bits, die je nach gewünschtem Geschwindigkeitslimit (in 5 Stufen, und Halt, Schnellhalt) blockweise (also nicht binär codiert) ausgeschnitten werden (1 Bits ausschneiden für 5/6-Limit, 2 Bits für 4/6-Limit, usw. bis 7 Bits für Halt und 8 Bits ausschneiden für Schnellhalt). Alternative Funktionsbeeinflussung wie bisher.

Das **Fenster „00“** ergänzt die „signalabhängige Zugbeeinflussung“ auf Grund der langjährigen Erfahrung, die bei ZIMO diesbezüglich entstanden ist. Mit Hilfe dieser Ausschneide-Bits (siehe Bild) kann eine freie Durchfahrtsrichtung (Signal von hinten soll unwirksam sein) definiert werden, außerdem aber auch eine Sicherung gegen das Losfahren eines Zuges in verkehrter Richtung nach Grün-Werden des Signals, weiters eine Zwangs-Richtungsschaltung der Lok (Ost-West, nicht lokbezogen !) in die von der Fahrstraße bestimmten Fahrtrichtung (eine Funktion, die uns bisher in vielen Anwendungen sehr abgegangen ist), und eine Variation des Bremsverhaltens gegenüber den normalen in den Decoder-CVs eingestellten Bremszeiten/-strecken (ist uns bisher auch abgegangen, weil z.B. in verdeckten Anlagengebieten ohne weiteres schneller gebremst werden könnte als im sichtbaren Bereich, oder auch im Gefälle).

Fenster „01“ und „10“ dienen zur Übermittlung eines Positions-Codes *) von der Schiene zu den Decodern im Gleisabschnitt. Der Grund für das Vorhandensein zweier Fenster für den gleichen Inhalt ist das „Grundproblem“ jeder gleisabschnitts-orientierten Datenübertragung (wie man es auch von der „bi-directional“ Diskussion kennt) – das Überbrücken der Trennstelle zwischen benachbarten Abschnitten beim Überrollen durch Loks und Wagen. Die beiden Fenster sollen nun so verwendet werden, dass ein Abschnitt das jeweils andere Fenster als sein Nachbar verwendet und das jeweils nicht verwendete Fenster einen Total-Cutout aller 10 Bits enthält; dann bewirkt das Überbrücken zweier Abschnitte, dass der Decoder wie gültige Codes in den beiden Fenstern erhält.

*) Der Positionscode muss nicht unbedingt ein fixer geographisch gebundener Wert sein, er kann auch variable „gleisabschnitts-gebundene Befehle an alle“ enthalten; denkbar wäre z.B. eine Aufteilung der 10 Bits in ein 2-Bit-Kennzeichen, welches sagt, ob die dahinterliegenden 8 Bits eine Position oder einen Befehl darstellen (flexible Aufteilung, z.B. $3 \times 256 = 768$ Positions-Codes und 256 Befehle; in Wirklichkeit etwas weniger, weil Werte wie „0“, „255“ und Spezialcodes nicht verwendbar sind).

Spezialverwendung der Positionsfenster zur „Ökonomisierung“ der signalabhängigen Zugbeeinflussung:

Die „signalabhängige Zugbeeinflussung“, wie sie von ZIMO seit langem praktiziert wird, ist relativ aufwändig, da sämtliche Bereiche einer Anlage (wo es Fahrstrassen gibt) aus Gleisabschnitten mit jeweils eigener Ansteuerlektronik bestehen müssen; z.B. auch die Weichenharfen eines Bahnhofes (da ja dort der Zug seine Langsamfahrt beibehalten muss und nicht plötzlich wieder auf volle Geschwindigkeit gehen darf) – bei Bahnhöfen mit zwei- oder mehrgleisigen Einfahrten ergibt diese eine große Zahl kleiner Gleisabschnitte, oftmals fast für jede Weiche ein eigener. Durch Spezialcodes (z.B. 10101..) in den Positionscodes-Fenstern (in der Gleisharfe wird ohnedies keine Positionserkennung gebraucht) könnte man eine „Speedlimit-Beibehalte-Funktion“ einführen, und einen einzigen Ausgang eines Gleisabschnitts-Moduls für eine Vielzahl solcher kleiner Abschnitte nutzen.

Beispiel: 2 Poscode-Abschnitte auf freier Strecke; Erkennung spätestes dann, wenn Zug komplett innerhalb der Strecke, weil beide Poscodes lesbar; wenn Lok vorne, meistens frühere Erkennung; wenn Lok hinten, Erkennung, wenn hintere Lok einfährt.

NOTES to the ZIMO DEMONSTRATION LAYOUT at DCC SPRING MEETING 2005 currently only in German available !

Durch „ortsabh. Zugbeeinflussung“ wird dem Decoder (den Decodern) am Gleisabschnitt ein Positions-Code übermittelt. Im Decoder selbst gibt es eine CV für einen „Routing-Code“. Der Decoder schickt (bei Bedarf priorisiert) im Quittungsverfahren Positions-Code und Routing-Code zum globalen „bi-directional Detektor“. Das Basisgerät schaltet (laut „ARA“ Definition in CVs, siehe unten, in späterer SW-Version durch Musterfahrten programmierbar) Weichen, Signals, und MX9.

Decoder-Software:

Empfangs-Fähigkeit für „ortsabhängige Zugbeeinflussung“ und damit auch „Positions-Code“, Hinweis: Grundsätzlich gilt jedes empfangenes „ortsabh. Funktions“-Byte (Verifizierung durch Doppelpfang ??) auch als Positions-Code, auch dann, wenn ein oder mehrere Bits in den Orts-CVs als Funktionsbits deklariert sind, außer wenn durch Bit 2 im Routing-Sende-Modus ausgeschlossen.

Neue CVs:

- Routing Code: 0 – keiner (daher auch keine Aussendung), 1 – 255
- Routing Delay: 0 – Gemeinsames Aussenden von Routing Code und Positions Code soll sofort nach Erhalt eines Positions-Codes beginnen, 1 – 255 mit 1 – 255 sec Verzögerung.
- Routing Sende-Modus: Bit 0: im Broadcast nein/ja
Bit 1: im Quittungsmodus nur zusammen mit Positions-Code / immer
Bit 2: „ortsabh. Byte“ mit deklarierten Funktionsbits gelten als Pos-Code / nicht
Bit 3, ...: ev. Details über Aussende-Häufigkeit unter bestimmten Bedingungen.

Detektor-Software:

Korrekturen, Empfang und Weiterleitung Positions- und Routing-Code (vorerst nur interessant, wenn beide zusammen, später Positions-Code auch allein, Routing-Code allein hingegen global uninteressant).

MX1-Software:

Vorerst sollen die „ARA's“ (Automatische Routing Abläufe) in Form von CVs im MX1 realisiert werden, mit eingeschränkten Möglichkeiten) später durch Musterabläufe.

9 CV-Gruppen; jede Gruppe umfasst 10 CV's (also insgesamt CVs 110 bis 199):

109: Generelles Time-out (für alle ARA's)

110: Positions-Code

111: Routing-Code (# 110 und # 111 bilden zusammen die Kennung für den definierten « ARA », bei deren Empfang der ARA aktiviert wird)

112: Start-MX9-Hauptabschnitt, der normalerweise auf U-H Bremsung steht, und auf dem der Positions-Code (siehe oben) ausgeschickt wird.

Nach Empfang des bi-directional Routing code zusammen mit dem bi-directional Positions-Code von einer Lok wird der „ARA“ aktiviert (= Stellen der „Fahrstrasse“), d.h. wenn die sonstigen Bedingungen (Nicht-Besetzt und Nicht-Belegt des Zieles) erfüllt sind, werden die Weichen und Signale geschaltet, dieser Gleisabschnitt auf „F“ gesetzt, solange bis Besetzt-Meldung weg oder Time-out (30 sec); Die Belegung des Ziel-Abschnitts wird durch dessen Erreichen aufgehoben (oder durch doppeltes Time-out – erstes Time-out Freiwerden des Start-Abschnitts, zweites Time-out zum Erreichen des Zieles)

113: Ziel-MX9-Hauptabschnitt, der nicht besetzt und nicht belegt sein darf als „sonstige Bedingung“ für Ablauf des „ARA“. Als „belegt“ gilt dieser

- # 114: Weiche oder Signal + Stellung 1.
- # 115: Weiche oder Signal + Stellung 2.
- # 116: Weiche oder Signal + Stellung 3.
- # 117: Weiche oder Signal + Stellung 4.
- # 118: Weiche oder Signal + Stellung 5.
- # 119: Weiche oder Signal + Stellung 6; Schalten der Weiche # Signale in Zeitintervallen 0,5 sec.

MX31-Software:

Anzeige des Routing- und Positions-Code im Bi-Di Balken (ev. anstelle des Häufigkeitsregisters wg. Platzmangel).

ZIMO TRACK SECTION MODULES and HÜBSCH TRACK SECTION ENCODERS

For using the methods mentioned in this paper (position dependent transmission of information for “signal controlled speed influence” and sending of position codes”) a product is needed, which carries this out:

TRACK SECTION MODULE MX9V (already existing):

The ZIMO track section module MX9V is able to carry out the transmission of position dependent information (signal controlled speed influence and position-codes); besides this it makes occupancy detection. Therefore it is used also in the demonstration layout – described in the chapter before. MX9 has 16 track section outputs, and it is a good solution for applications, which are rather big, and where the system bus (in case of ZIMO - the CAN bus - is available on-site.

TRACK SECTION ENCODER (future product):

The MX9 is somewhat too big for small applications; and for outdoor layouts it has the disadvantage of needing the ZIMO system bus (CAN bus) for control and reporting back its information. Therefore there will be an alternative solution, which just now (Mar/April 2005) is in development by Arnold Hübsch (in cooperation with ZIMO): the “Track Section Encoder”.

This “Track Section Encoder” has an accessory decoder address (so there are 512 possible track sections or less, if some of these addresses are used for normal accessory decoders). It sets speed limit information and position codes onto the track. It is able to use both ZIMO’s HLU and the asymmetric DCC signal method, also known as LENZ ABC (It is currently not known how Lenz handles the licensing issue, so maybe we will only implement the stop function).

The Track Section Encoder also does the occupancy detection for its track output, and is able to send this information back via “bi-directional communication” according to NMRA RP 9.3.1, 9.3.2. to the global detector (command station).

Function outputs of the Track Section Encoder allow to control a signal (preferably if it is tied together with the speed limit for the track section; this means a little “normal” accessory decoder is included in every Track Section Encoder”).

Additionally the “Track Section Encoder” has input lines which allow setting the speed limits with simple switches by pulling one of the input lines to ground. This helps existing model railroaders to convert over to DCC. If both input lines and DCC commands are active the result may be AND,OR, XOR combined. The switch may be a contact inside a semaphore, or part of a turnout.

