Standards for European Model Railroads

Digital Control Signal DCC

Baseline Data Packets

NEM

671

Binding Standard

Edition 2014
(First english Edition)

Note 1: NEM 671 content matches NMRA-Standard S 9.2 (as of July 2004).
NEM 671 follows the NMRA-Standard S 9.2.

Note 2: With this standard there is no backwards compatibility to older decoders with a 14 speed step mode and supplemental function, or likewise to older decoders whose internal timing system isn't compatible with the timings given here.

Note 3: Detailed technical data and foundations for conformance tests are found in the following standards of RailCommunity (www.railcommunity.org):
RCN-210 DCC Bit transmission
RCN-211 DCC Packet structure, fundamental construction of all packets and the address range.
RCN-212 DCC Operational commands for vehicle decoders, Commands for vehicle decoders.
RCN-213 DCC Operational commands for accessory decoders
RCN-214 DCC Configuration commands

1. Purpose of the Standard

This standard describes fundamental data packets (baseline data packets) that are transmitted to the DCC decoder.

2. Explanations

- A DCC data packet is a defined sequence of bits, which are described as track signal in NEM 670.
- The DCC baseline data packet is composed of a minimum of bits and bit groups, also referred to as a data packet.
- The bit groups, composed each of eight bits, are called bytes. Each bit is weighted relative to its position; the first, left most bit, has the highest value and is called MSB (most significant bit). The bits of a byte are numbered from left to right beginning with 7 down to 0. The last bit is called the LSB (least significant bit).
- The following symbology is used to identify the meaning of the bits:

0	bit value 0
1	bit value 1
A	address bit
D	data bit (not relevant here)
P	check bit – the eight bits of the last byte of a packet
x	placeholder for a bit, whose value is dependent on the type of packet and command and isn't of further concern at this point.

3. The Components of the General DCC Baseline Data Packet

The following components of a data packet composed of bits and bytes define the general valid composition of a DCC baseline data packet used to activate a decoder[1)].
Parts 4 and 5 described below occur as pairs one or more times!

The DCC Baseline Data Packet is composed out of the following parts:

[1)] It is permissable that a decoder may also recognize controll formats other than DCC (see also 6.).

1. **Sync Bits (a.k.a. Preamble):**
   The identification of a DCC data packet and the synchronization along byte boundaries is accomplished with a sequence of one bits.
   A command station must transmit at least 16 sync bits (the preamble).
   A decoder shall be capable of receiving a packet with 12 or more sync bits.
   A decoder shall not accept as valid, any packet that has less than 10 sync bits.
2. **Packet Start Bit:**
   The packet start bit is a zero bit that immediately follows the sync bits. The start bit completes the synchronization and indicates to the decoder that the following bits belong to the first byte of the packet.
3. **First Byte:**
   The first byte of a packet is an address byte in operational mode, or a command byte in programming mode.
4. **Data Byte Start Bit:**
   This zero bit leads in the following data byte.
5. **Data Byte:**
   The 8 bits of every data byte are used for addresses, instructions, data, or for error detection as check bytes.
   The last byte of a DCC packet is always the check byte.
6. **Packet End Bit:**
   The packet end bit is a one bit and marks the end of the data packet.
   If the next DCC packet will follow without interruption, then the packet end bit can be counted as part of the following packet's sync bits.
   If a packet in a different format or an interruption follows a DCC packet, then the DCC bit sequence must be continued after the packet end bit for at least 26 µs, i.e. during that time period there can be no polarity change or turning off of voltage.

The check byte is constructed by performing an exclusive or (XOR) operation on all of the prior byes. It can be checked by performing an XOR across all of the bytes and the check byte, the result of which should be a null value. All decoders must ignore any packet that fails the check byte test.

A DCC packet is at least three bytes long. This results in a bit sequence for a packet of three bytes:

1111111111111111 0	**xxxxxxxx 0**	**xxxxxxxx 0**	**PPPPPPPP 1**
Sync bits & Start	Byte 1	Byte 2	Byte 3 = Check byte and packet end bit

An example of a base speed and direction command for a 7 bit address, 55, forward with speed step 14, results in the following bit sequence:

1111111111111111 0 00110111 0 01100111 0 01010000 1

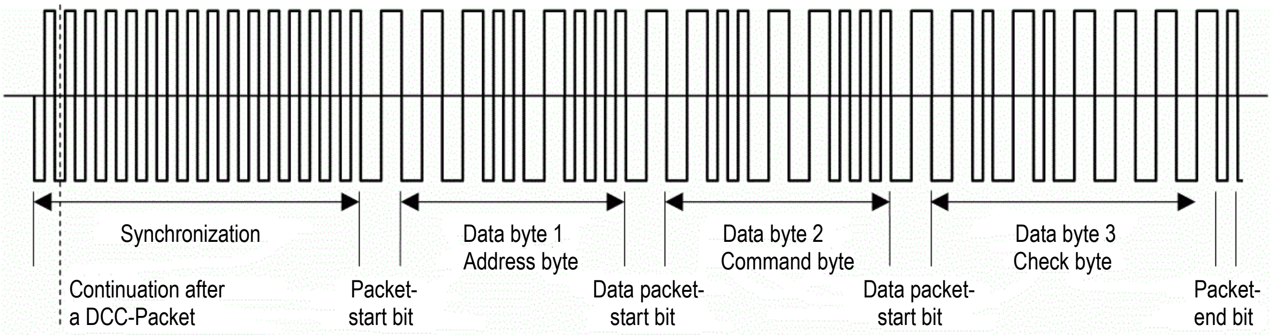

Figure 1: Example of a three byte DCC packet (1 address byte, 1 command byte, 1 check byte), encoded for address 55 and forward with speed step 14. The minimally required transmission of 16 sync bits is also depicted.

The bit sequence for a six byte packet is:
{sync bits} 0 xxxxxxxx 0 xxxxxxxx 0 xxxxxxxx 0 xxxxxxxx 0 xxxxxxxx 0 PPPPPPPP 1

4. The Format of the DCC Baseline Data Packet

This standard specifies that for DCC compatible parts, that every DCC controller is capable of encoding the operator input into conforming DCC baseline data packets, and that any vehicle decoder can recognized and decode said packets and provide the corresponding electrical control output signals. DCC baseline data packets are defined to provide a minimum of interoperability between controllers/decoders of different DCC control systems. More complex data packets that enable decoders to support additional functions, addresses, etc. are beyond the scope of this standard.

4.1 DCC Baseline Data Packet to Control the Speed and Direction of a Locomotive

Format of DCC baseline data packet:

1111111111111111	**0**	**0AAAAAAA**	**0**	**01DCSSSS**	**0**	**PPPPPPPP**	**1**
Synchronization		Data byte 1		Data byte 2		Data byte 3	

Data byte 1 – Address byte: The address byte transmits the address, the number, of the intended recipient of the data packet.

The first byte of the DCC packet format contains, in operational mode, the primary address. To support differing types of decoders, this primary address is subdivided into the following fixed blocks.

- Address 0000-0000 (0):
  Broadcast address for communications to all vehicle decoders.
- Addresses 0000-0001 to 0111-1111 (1-127)(inclusive):
  Vehicle decoders with 7 bit addresses 0AAA-AAAA
- Addresses 1000-0000 to 1011-1111 (128-191)(inclusive):
  Simple accessory decoders with 11 bit addresses 10AA-AAAA 1AAA-DAAR and
  Expanded accessory decoders with 11 bit addresses 10AA-AAAA 0AAA-0AA1
- Addresses 1100-0000 to 1110-0111 (192-231)(inclusive):
  Vehicle decoders with 14 bit addresses 11AA-AAAA AAAA-AAAA
- Addresses 1110-1000 to 1111-1110 (232-254)(inclusive):
  Reserved for future use
- Address 1111-1111 (255):
  Neutral or also idle packet

Commands sent to the broadcast address 0000-0000 must be executed by all vehicle decoders. The reset packet, which is a reset command 0000-0000 (see section 4.2) to the broadcast address 0000-0000, must also be observed by accessory decoders.

Data byte 2 – Command byte: The command byte transmits the information for the control of the speed and direction functions of the addressed locomotive.

The bits 7 (zero bit) and 6 (one bit) mark the data byte as a command byte.[2)]

Bit 5 (D) defines the direction of travel; a value of „1“ indicating the locomotive must move forwards[3)], while a value of „0“ indicates reverse travel.

[2)] Other bit patterns in bits 7 and 6 are reserved for specific instructions within the command byte.
[3)] Forwards means that the vehicle end designated as 1 is at the front in direction of travel.

Bit 4 (C) has a special function and is generally the last bit (LSB) of the speed control.
Bit 3 – 0 (SSSS) as a group along with Bit 4, form a binary encoding of the speed level. Table 1 describes the connection between binary encoding and speed level.

Data byte 3 – Check byte: The check byte enables digital decoders to recognized transmission errors.

Table 1: Correlation between encoding of the S bits 3-0, the C bit 4, and the speed steps [4]

$S_3S_2S_1S_0$ C	Speed Step	$S_3S_2S_1S_0$C	Speed Step	$S_3S_2S_1S_0$C	Speed Step	$S_3S_2S_1S_0$C	Speed Step
0 0 0 0 0	Stop	0 1 0 0 0	5	1 0 0 0 0	13	1 1 0 0 0	21
0 0 0 0 1	Stop**	0 1 0 0 1	6	1 0 0 0 1	14	1 1 0 0 1	22
0 0 0 1 0	EStop*	0 1 0 1 0	7	1 0 0 1 0	15	1 1 0 1 0	23
0 0 0 1 1	EStop**	0 1 0 1 1	8	1 0 0 1 1	16	1 1 0 1 1	24
0 0 1 0 0	1	0 1 1 0 0	9	1 0 1 0 0	17	1 1 1 0 0	25
0 0 1 0 1	2	0 1 1 0 1	10	1 0 1 0 1	18	1 1 1 0 1	26
0 0 1 1 0	3	0 1 1 1 0	11	1 0 1 1 0	19	1 1 1 1 0	27
0 0 1 1 1	4	0 1 1 1 1	12	1 0 1 1 1	20	1 1 1 1 1	28

* Emergency Stop, stop the locomotive as quickly as possible!
** Optionally, bit 5, the direction of travel, may be ignored.

4.2 DCC Baseline Data Packet for Resetting of Decoders

Format of the DCC Baseline Data Packet:

1111111111111111	**0**	**00000000**	**0**	**00000000**	**0**	**00000000**	**1**
Synchronization		Data byte 1		Data byte 2		Data byte 3	(Check byte)

The data packet within which all bytes have only bit values of „0“, is the data packet for the resetting of a decoder. It clears all of the decoder's volatile storage including the speed and direction of travel data. After receiving the data packet, the decoder returns to it power up state; any moving vehicles are to immediately stop.

Within 20 milliseconds of resetting, no data packets shall be sent to addresses in the range 01100100 (address 100) and 01111111 (address 127) inclusive, unless intending to put the decoder into service mode[5].

4.3 DCC Baseline Data Packet for Idling of Decoders

Format of the DCC Baseline Data Packet:

1111111111111111	**0**	**11111111**	**0**	**00000000**	**0**	**11111111**	**1**
Synchronization		Data byte 1		Data byte 2		Data byte 3	(Check byte)

The data packet whose first and third bytes are composed of all 1 bits, and whose second byte is composed of all 0 bits, is the idle packet.

After receiving this data packet a decoder shall take now new action. The decoder shall behave as if this was a normal packet addressed to a different decoder.

4.4 DCC Baseline Data Packet for Stopping of Decoders

Format of the DCC Baseline Data Packet:

1111111111111111	**0**	**00000000**	**0**	**01DC000S**	**0**	**PPPPPPPP**	**1**
Synchronization		Data byte 1		Data byte 2		Data byte 3	(Check byte)

The data packet whose first byte is 0 bits, whose second byte is a specific stop command and whose third byte is the check byte, is a data packet for the stopping of a decoder (Speed = 0).
If bit 0 of byte 2 (bit S) is a 0 bit, then the decoder shall bring the locomotive to a stop normally,

[4] The corresponding speed step is produced by subtracting 3 from the binary value of the bit group.
[5] Decoder configuration changes may commence immediately after the decoder reset data packet.

under the decoder configured delay settings. If the S bit is a 1, decoder's controlling a locomotive's motor shall stop providing power to the motor.

5. Repeating the DCC Basis Data Packets

5.1 Time Between Two Data Packets

Data packets sent to decoders should be repeated as often as possible, because interruptions or poor electrical conductivity between rails, wheels and power pickups can lead to data loss.

The track power may be interrupted between the end bit of a packet and the synchronization bits (preamble) of a subsequent packet, to enable the transmission of an alternative command control format (bi-directionality).

Decoders must be operationally ready when multiple data packets are addressed to them so long as there is a time span of a minimum of 5 milliseconds between the stop bit of the first packet and the start bit of the second packet[6].

Whenever a decoder receives a data packet with missing or invalid data byte start or end bits, or incorrect check bytes, it must recognize the next valid synchronization (preamble) as the beginning of a new data packet. A new type of control signal is only allowed to be transmitted on the track signal after the stop bit of a packet and before the synchronization sequence (preamble) of a subsequent packet.

Minimum time between two DCC data packets:	$\mathbf{t_D} > \mathbf{5\ ms}$	Spacing Time

5.2 Repetition of Identical Data Packets

DCC controllers shall be capable of repeating the same data packet at least every 30 milliseconds, as measured between the start bits of two data packets.

Maximum time to repeat a DCC data packet:	$\mathbf{t_R} \leq \mathbf{30\ ms}$	Repetition Time

6. Decoder Behavior Under Automatic Conversion to Varying Control Systems

Form the manufacturers of decoders that include automatic control format change over which include the NEM-DCC control format (multisystem decoders), it is required that one can disable this automatic format change functionality such that the decoder reacts exclusively to DCC control.

With enabled automatic conversion, decoders must remain in DCC mode for at least 30 milliseconds. With disabled automatic conversion, decoders must remain in DCC mode regardless of whether start bits of other control formats are received[7].

Dwell time of digital decoder in DCC mode:	$\mathbf{t_w} \geq \mathbf{30\ ms}$	Wait Time

[6] Care is to be taken when transmitting two signal packets within 5 milliseconds of each other. If the addresses of these packets lie in the range of 112 (binary 01110000) and 127 (01111111), as older DCC decoders may interpret these packets as service mode packets.

[7] Several older DCC deocders require the reception of a valid DCC data packet within 30 milliseconds in order to avoid transitioning into analog mode. A repetition step of greater than 30 milliseconds can result in a drop in the performance of a decoder.