1 General

1.1 Introduction and Intended Use (Informative)

This standard describes advanced commands that are in the DCC address partitions 253 and 254 as described by S-9.2.1. The 253 address partition provides the following features:

- New packet framing with added integrity checking for packets greater than 6-bytes inclusive of X-OR checksum.
- Bulk data transfers to/from Decoder.
- Ability to address a Decoder by its 7-bit or 14-bit DCC mobile address, or 9-bit or 11-bit DCC accessory address and send it commands

The 254 address partition provides the following features:

- New packet framing with added integrity checking for packets greater than 6-bytes inclusive of X-OR checksum.
- Ability to discover a Decoder by unique 44-bit ID and assign it a 7-bit or 14-bit DCC mobile address, or 9-bit or 11-bit DCC accessory address.
- Ability to address a Decoder by unique 44-bit ID and send it commands.
- Bulk data transfers to/from Decoder.

1.2 References

This standard should be interpreted in the context of the following NMRA Standards, Technical Notes, and Technical Information.

1.2.1 Normative

- S-9.2 DCC Electrical Standard, which specifies the general packet format
- S-9.2.1 DCC Extended Packet Formats, which specifies the available packet address spaces as well as extended DCC packet commands
- S-9.2.2 DCC Configuration Variables
- S-9.2.2, Appendix A, DCC Manufacturer ID Codes
- S-9.3.2 DCC Basic Decoder Transmission, which specifies the technique and basic commands for Decoders to transmit feedback information

1.2.2 Informative

- TN-9.2.1.1 DCC Electrical Standard, which provides commentary on this standard
- RCN-218 DCC-A Automatic Logon, with which applicable parts of this standard are intended to be in harmony

1.3 Terminology

Term	Definition
Decoder	DCC receiver for controlling vehicle or accessory animation.

Term	Definition
System	Device (or combination of devices) that sends DCC commands and processes Decoder feedback. Commonly referred to a Command Station or Central Unit. The system may also comprise separate feedback modules.
Unique ID	Unique 44-bit ID HHHH HHHHHHHH (12-bit Manufacturer ID, represented as big endian) UUUUUUUU UUUUUUUU UUUUUUUU UUUUUUUU (32-bit Unique ID, represented as big endian)
CID	System (Command Station) ID. 16-bit value typically represented by a hash of the system's Manufacturer ID and Unique ID.
DID	Unique ID of a decoder (44-bit).
Session ID	
►	Message sent from System to Decoder
◄	Message sent from Decoder to System (Feedback)
{checksum}	Represents the single X-OR checksum or CRC-8 + X-OR checksum as appropriate for the given packet length.

1.4 Requirements

All messages defined by this standard are optional. For certain interactions, a specific set of messages are required to be supported. If a Decoder implements a message defined by this standard, it must be implemented exactly as described by this standard.

M = Mandatory, O = Optional

Command	Interaction Type		
	Logon	Throttle UI	Bulk Write
Select (ReadShortInfo)	M		
Logon Assign	M		
Logon Enable	M		
Get Data Start		O[1]	
Get Data Continue		O[1]	
POM		M	
XPOM[2]		O[1]	

[1] For the Throttle UI interaction, it is required to support at least one of **POM**, **XPOM**, **Select (ReadBlock)**, **ReadBlock**, or **ReadBackground**. If **Select (ReadBlock)** or **ReadBlock** is supported, it is also required to support **Get Data Start/Continue**.

[2] **POM** and **XPOM** are defined in S-9.2.1

Command	Interaction Type		
	Logon	Throttle UI	Bulk Write
Select (ReadBlock)		O[1]	
ReadBlock		O[1]	
ReadBackground		O[1]	
WriteBlock			M
253 Addressed Continue			M

Data Space	Interaction Type		
	Logon	Throttle UI	Bulk Write
Capabilities	O	O	O
Data Space Info	O	O	O
Short Gui	O	M	O
Configuration Variables	O	O	O

2 Packet Framing

2.1 Error Detection

If total packet length inclusive of the X-OR checksum in bytes (from start of packet bit to packet end bit) is less than or equal to 6-bytes, the normal packet framing and X-OR checksum is used, as described in S-9.2 Baseline Packets.

253 or 254	1-4 bytes payload	X-OR

If the total packet length exceeds 6-bytes, an additional CRC-8 checksum is inserted just before the traditional X-OR byte. A Decoder is required to check that both the CRC-8 and X-OR checksums are valid.

253 or 254	1-4 bytes payload	CRC-8	X-OR

The CRC-8 polynomial used is $x^8 + x^5 + x^4 + 1$, also commonly known as the Dallas/Maxim 1-Wire bus CRC. The initial value is 0, not inverted. On the receiving end, the final result is expected to be 0 when calculated over the entire message including the CRC-8 byte itself and excluding the X-OR byte.

When a CRC-8 is specified for use in Decoder feedback, the same CRC-8 polynomial with initial value 0 and not inverted is used, unless otherwise specified.

All packets defined by this standard (253 and 254 address partitions), use this packet framing and error detection scheme, and therefore the type of error detection that a Decoder must perform on any given packet is fully generalized.

The System shall not send packets with a total length of 7 bytes to the address partitions 253 and 254. A Decoder shall ignore packets received to address partitions 253 and 254 which are 7 bytes long.

The maximum length of a packet in the Address Partition 253 and 254 is 32 bytes unless explicitly noted otherwise, including the 253/254 address and the checksum byte(s).

2.2 Feedback

S-9.3.2 defines two feedback channels 1 & 2.

2.2.1 Address Partition 253

For any addressed message sent to the 253 partition, only the specifically addressed Decoder may send feedback data in channel 2. Feedback channel 1 is reserved for future standards. A mobile Decoder shall not respond in channel 1, even if its CV28, bit 0 setting enables unsolicited Decoder initiated transmission.

The System shall support reception of feedback to addressed packets using the 253 address.

2.2.2 Address Partition 254

All messages sent to the 254 address space combine feedback channels 1 & 2 into a single extended length feedback channel. The channel 1 and 2 start and end timing shall be respected by the Decoder.

CV28, bit 7 must be set in order to enable address partition 254. Otherwise, a Decoder shall ignore all commands sent to address partition 254.

2.2.3 Encoding, Padding, and Alignment

As defined in S-9.3.2, all feedback data is encoded as 6 bits of actual information into 8-bits over the physical track circuit. S-9.3.2 refers to this as 4/8 coding, and it is more commonly referred to as 6b/8b encoding.

To meet the coding requirement, all feedback data size must fall on a 6-bit boundary. In some cases, the feedback messages defined do not end on a 6-bit boundary. In this case, they shall be padded to the next 6-bit boundary by '0' bits. If after padding '0' bits to the next 6-bit boundary results in remaining space in the feedback channel, the Decoder shall fill the remaining space with ACK bytes.

2.2.4 Variable Length Feedback

Certain interactions defined in this Standard require a variable length message to be transmitted from the Decoder to the System. A variable length message is typically transmitted over multiple cutouts, and has the following format:

{header} {payload-bytes} {CRC-8}.

The header byte is as follows:

```
XRCLLLLL
```

Parameter	Description
X	Signals a special format message that fits into a single cutout **0**: Regular message following the header format **1**: Special format message. The remaing bits of the header byte do not follow the format defined here; see the specific message to which this repsonse is given
R	Response bit **0**: This message is a response to a request sent by the System **1**: This is an unsolicited message generated by the Decoder.
C	Continuation bit **0**: This is the first message from the Decoder in an interaction. **1**: Continuation message, which should be interpreted in the context of the previous message sent from the same Decoder.
LLLLL	Length field, defines the total number of payload bytes that follow. The total number of bytes is Length + 2 (header + payload + CRC-8).

2.3 Acknowledgement

A Decoder shall acknowledge all messages sent to the 253 and 254 address partitions which are specifically addressed to that Decoder, as follows:

- For a message that failed checksum validation, the Decoder shall not transmit in either channel.
- Any response message specified in the standard acts as an acknowledgement.
- For a message in the 254 address partition, if the Decoder does not have a specific message to transmit, then 8 ACK bytes shall be transmitted, filling channel 1 and 2.
- For a message in the 253 address partition, if the Decoder does not have a specific message to transmit, then 6 ACK bytes shall be transmitted, filling channel 2.

2.4 Frequency

In order to make the most efficient use of the limited DCC bandwidth, messages defined by this standard:

1. shall only be repeated if not acknowledged;
2. shall not be sent periodically as part of a background "refresh" loop unless explicitly noted otherwise.

For all of the packets in this standard, a Decoder is required to correctly decode and act upon all packets, including packets sent back-to-back or directly following another previously sent packet addressed to it, with no delay in between.

2.5 Sequenced Messages

Certain interactions defined by this standard require a sequence of DCC messages emitted by the System, and the interpretation and desired behavior of a Decoder depends on the entire sequence (stateful evaluation).

A Decoder shall act on such a sequenced message if and only if the Decoder has successfully decoded an uninterrupted contiguous stream of DCC messages without any checksum errors since

the message that started the stateful sequence. The System, when sending out a sequence of messages for stateful evaluation, shall not interleave packets of any alternative track protocol within the sequence.

2.6 Error Codes

In some cases, as specified in the individual command definitions, a command response might produce an error. These are the generalized error codes that may be used:

Error Code	Description
0x1000	Unspecified permanent error.
0x1040	Unimplemented.
0x1042	Command is not supported by this Decoder.
0x1080	Invalid arguments.
0x1081	Data space number is unknown.
0x1082	Write offset is out of bounds.
0x1083	Data space is read-only.
0x2000	Unspecified temporary error.
0x2010	Timeout.
0x2020	Buffer unavailable or Decoder busy.
0x2040	Unexpected sequence of operations or inconsistency found in the internal state.
0x2041	Decoder's internal state has been reset, for example due to a loss of power.

3 Extended Address Format

Messages within this standard that address a Decoder by traditional DCC address use the following format:

Address Type	Byte 1	Byte 2
14-bit DCC	`xx000000 - xx100111` (CV 17 & 0x3F)	`AAAAAAAA` (CV 18)
11-bit Accessory	`xx101000 - xx101111` ((CV521 & 0x07) | 0x28)	`AAAAAAAA` (CV 1)
9-bit Accessory	`xx110000 - xx110111` ((CV521 & 0x07) | 0x30)	`AAAAAANN` (`NN` == output pair)
7-bit DCC	`xx111000`	`0AAAAAAA`
Broadcast	`xx111000`	`00000000`
Reserved	`xx111001 - xx111111`	

4 Command Types in Address Partition 253

All commands take the following generalized form:

► `{preamble} 0 11111101 0 TTAAAAAA 0 AAAAAAAA 0 {command byte} 0 {payload-bytes} 0 {checksum byte(s)} 1`

Where:

TTxxxxxx	Command Type
11AAAAAA	Addressed
10AAAAAA	Addressed Continue
01AAAAAA	Addressed Control
00AAAAAA	Addressed S-9.2 / S-9.2.1 Chained

For Addressed, Addressed Continue, and Addressed Control, multiple command/payload byte sequences cannot be chained together in a single packet frame.

4.1 Addressed and Addressed Continue

All Addressed and Addressed Continue commands have bit 7 of the first byte set, and the destination Decoder address defined in the first and second byte. Bit 6 of the first byte is the "continue" flag, which is clear if the command is an Addressed Continue command.

Addressed:

► `{preamble} 0 11111101 0 11AAAAAA 0 AAAAAAAA 0 {command-byte} 0 {payload-bytes} 0 {checksum} 1`

Addressed Continue:

► `{preamble} 0 11111101 0 10AAAAAA 0 AAAAAAAA 0 {payload-bytes} 0 {checksum} 1`

An addressed continue message has to be interpreted in the context of the previous addressed message sent to the same Decoder, which shall explicitly allow a continuation message to be sent and the interpretation of said continue message. For more information, please see section 2.5 above.

Command Byte		Brief Description
0000HHHH	Manufacturer Specific	Command space available to manufacturers using their 12-bit DCC manufacturer ID
00010000 – 11111011	Reserved	All undefined commands are reserved for future use
11111100	WriteBlock	Data space write
11111101	ReadBackground	Background data space read
11111110	ReadBlock	Fast data space read
11111111	Reserved	All undefined commands are reserved for future use.

4.2 Addressed Control

All Addressed control commands have bit 7 clear and bit 6 set in the first byte. Currently there are no messages defined for Addressed Control.

► {preamble} 0 11111101 0 01AAAAAA 0 AAAAAAAA 0 {control byte} 0 {payload-bytes} 0 {checksum} 1

4.3 Addressed S-9.2 / S-9.2.1 Chained

All Addressed S-9.2 / S-9.2.1 chained commands have bits 7 and 6 of the first byte clear, and the destination Decoder address defined in the first and second byte.

► {preamble} 0 11111101 0 00AAAAAA 0 AAAAAAAA 0 {instruction-bytes} 0 {instruction-bytes} 0 … {instruction-bytes} 0 {checksum} 1

◄ [ch2]: ack or other feedback response

Each set of {instruction-bytes} is a separate S-9.2 / S-9.2.1 Multi-Function Decoder command. A chained packet may contain multiple commands up to a maximum allowable packet length of 16 bytes, including the 253 address and the checksum byte(s).

Upon successful reception, the Decoder shall respond in feedback channel 2 with either an ACK or another appropriate feedback response.

5 Command Types in Address Partition 254

The general format is:

► {preamble} 0 11111110 0 {command byte} 0 {parameter bytes(s)} 0 {checksum} 1

Command Byte		Brief Description
00000000	Get Data Start	First of sequenced messages for Decoder to send data
00000001	Get Data Cont	Further sequenced messages for Decoder to send data
00000010	Get Data Start	First of sequenced messages for Decoder to receive data
00000011	Get Data Cont	Further sequenced messages for Decoder to receive data
00000100 – 10111111	Reserved	All undefined commands are reserved for future use.
1101HHHH	Select	Requests metadata such as desired address from Decoder
1110HHHH	Logon Assign	Assigned a DCC address to a given Decoder
11110000 – 10111011	Reserved	All undefined commands are reserved for future use.
111111GG	Logon Enable	Requests registration from one or more Decoders

6 Logon

Logon is an automatic registration procedure for DCC. The aim is to significantly increase the user-friendliness of model railway controls. Its application relieves the user from the burden of assigning addresses and functions. The aim is, for example, to have a vehicle immediately available to the DCC System with its name and all its properties automatically discovered upon placement onto the rails.

6.1 Procedure

A unique Decoder Identifier (DID) is used to distinguish between different Decoders. Based on this DID, the Decoders are assigned a shortened (session) address (non-conflicting regular DCC address) which enables Decoders to receive regular DCC commands. If possible, the previous Decoder address is used for the session address. For this purpose, a registration procedure is carried out at the beginning in order to assign a DCC address and optionally make known the Decoder properties.

With respect to the registration procedure a Decoder may be in one of two states: Unselected or Selected.

The registration is divided into the following steps:

6.1.1 Enumeration

This is where the existing Decoders are determined and any access conflicts that may arise are resolved. At the end of the enumeration phase, the System knows the DID's of the existing Decoders.

The System sends requests to the Decoders to Logon (**Logon Enable**). These Logon requests contain a unique identifier (CID) for the System and a Session ID. The Decoders can use the combined ID to recognize the System after a power cycle. If the System is not recognized, the Decoder shall start in the Unselected state. The Decoders in Unselected state respond to the **Logon Enable** command according to certain rules with a Logon feedback response containing the DID of the Decoder.

If many of the Decoders are already known in the System, or if local feedback detectors are used, this phase will be short. In the event of a collision of simultaneous responses from several Decoders, the detection is not reliable. A separation is then carried out by means of dynamic back-offs in the Decoders. The separation takes place (identified by the coding of the **Logon Enable** command) separately for accessory Decoders and mobile Decoders.

6.1.2 Confirmation

The System confirms the enumeration by addressing the Decoder via its DID using the **Select** (**ReadShortInfo**) command. The **ReadShortInfo** feedback to the **Select** command tells the System important information about the Decoder such as the last assigned DCC address, protocol capabilities, and function capabilities.

When a Decoder receives a command addressed to its unique ID, such as a **Select** command, it will transition to Selected state for the current session.

6.1.3 Assignment

The System assigns a DCC address to the Decoder which is to be used in this session using the **Logon Assign** command. The Decoder replies to this message with feedback containing information about whether its configuration has changed since last discovered by the System.

When a Decoder receives an assigned address, after a Decoder restart, upon receiving a **Logon Enable**, the Decoder shall start in Selected state if the CID is the same and the Session ID is the same or has been incremented by less than four.

6.1.4 Configuration Discovery

Lastly, the System may optionally discover additional configuration information about the Decoder, such as user-visible name, function assignments, etc… The most commonly used information is

mapped to standardized CVs. Existing CV read/write commands (**POM / XPOM**) may be used to retrieve this information. Implementation of the Data Space commands are not required; however, the Decoder may optionally implement the Data Space commands for more efficient transfer of this information.

6.1.5 CID

The CID is a 16-bit value chosen by the manufacturer to have a reasonable probability of uniqueness among all systems produced by a given manufacturer.

6.2 Logon Commands

6.2.1 Logon Enable

This is a registration request. It shall be sent out periodically at least once every 300 milliseconds. More frequent transmission is recommended after the system is started.

► `{preamble} 0 11111110 0 111111GG 0 CCCCCCCC 0 CCCCCCCC 0 SSSSSSSS 0 {checksum} 1`

◄ `[ch1+ch2]: {ID15}HHHH HHHHHHHH UUUUUUUU UUUUUUUU UUUUUUUU UUUUUUUU`

Parameter	Description
`GG`	The address group determines which Decoders shall respond **00** ALL: All Decoders respond **01** LOCO: Mobile Decoders only **10** ACC: Accessory Decoders only **11** NOW: All Decoders (regardless of back-off)
`CCCC..CCCC`	CID (System Identifier), two bytes, MSB first followed by LSB
`SSSSSSSS`	Session ID number
`HHHH..HHHH`	Manufacturer ID, 12-bits, four MSb first followed by eight LSb
`UUUU..UUUU`	Decoder Unique ID, four bytes, MSB first, LSB last

A Decoder, upon receiving a **Logon Enable** message, shall evaluate if it is in Selected or Unselected state according to Section 6.4.1 below. Decoders in Selected state shall never respond to Logon Enable commands. Decoders in Unselected state respond based on the GG parameter and their back-off algorithm according to Section 6.4.2 below:

- When GG is NOW, all Decoders respond with their Unique ID (DID) and reset their back-off algorithm to initial state.
- Otherwise, the Decoders matched by GG respond when their back-off counter is expired.

6.2.2 Select

This command addresses a Decoder by its DID, and transitions the Decoder to Selected state for the current session.

► {preamble} 0 1101HHHH 0 HHHHHHHH 0 UUUUUUUU 0 UUUUUUUU 0 UUUUUUUU 0 UUUUUUUU 0 CCCCCCCC 0 (NNNNNNNN 0 VVVVVVVV 0 VVVVVVVV
245 0 VVVVVVVV 0 SSSSSSSS 0) {checksum} 1

Parameter	Description
HHHH..HHHH	Manufacturer ID, 12-bits, four MSb first followed by eight LSb
UUUU..UUUU	Decoder Unique ID, four bytes, MSB first, LSB last
CCCCCCCC	Command Byte. **11111111**: ReadShortInfo **11111110**: ReadBlock **11111101**: Reserved **11111100**: WriteBlock - Not currently defined **11111011**: Set Decoder internal status **11111010..00000000**: Reserved
NNNNNNNN	[optional] Data space number
VVVV..VVVV	[optional] CV address, three bytes MSB first (CV31 followed by CV32), LSB last
SSSSSSSS	[optional] Number of CV's requested

6.2.2.1 ReadShortInfo

If CCCCCCCC is 11111111 (**ReadShortInfo**), the response is inline:

◄ [ch1+ch2]: 1RAAAAAA AAAAAAAA FFFFFFFF PPPPPPPP PPPPPPPP CRC-8

Parameter	Description
R	Reserved – Send as 0, check upon receipt[3]
AAAA..AAAA	Suggested Decoder address, see Section 3 above.
FFFFFFFF	**Mobile Decoder**: Highest configured function assignment (N-0: F0 = 0, 0 - 255) **Standard Accessory Decoder**: Highest number of output pairs (N-1, 0 - 255)[4] **Extended Accessory Decoder**: Highest possible position (N-1, 0 - 255)

[3] Check upon receipt means that a receiving component has to verify that this bit has the prescribed value (0 in this case), and if it has a different than expected value, then reject the message. In this case, the address assignment will fail and trigger an error condition if this bit is 1.

[4] A normal four output pair accessory decoder would use value = 3 for this field. An accessory decoder that has more than four output pairs with sequential addressing would have a value > 3.

Parameter	Description
PPPP..PPPP	Protocol Support Flags - overlaps with capabilities space bytes 0 and 1 as defined in Section 7.4.1 below. PPPPPPPP PPPPPPPP: Byte 0 PPPPPPPP PPPPPPPP: Byte 1

6.2.2.2 ReadBlock

If CCCCCCCC = 11111110 (**ReadBlock**) a simple ack response shall be sent. The **Select** message shall be immediately followed (back-to-back) by a **Get Data Start** message.

◄ [ch1+ch2]: ACK

After the **Get Data Start**, the System shall emit enough **Get Data Continue** messages for the Decoder to transmit the requested amount of data. This sequence uses variable length feedback format as described in Section 2.2.4 above.

6.2.2.3 Set Decoder Internal Status

If CCCCCCCC = 11111111 (**Set Decoder Internal Status**), a simple ack response shall be sent.

► {preamble} 0 1101HHHH 0 HHHHHHHH 0 UUUUUUUU 0 UUUUUUUU 0 UUUUUUUU 0 UUUUUUUU 0 11111011 0 NNNNNNNN 0 {checksum} 1

◄ [ch1+ch2]: ACK

Parameter	Description
HHHH..HHHH	Manufacturer ID, 12-bits, four MSb first followed by eight LSb
UUUU..UUUU	Decoder Unique ID, four bytes, MSB first, LSB last
CCCCCCCC	**11111011**: Set Decoder internal status
NNNNNNNN	**11111111**: clear all change flags to 0 **11111110..00000000**: Reserved

6.2.3 Get Data

6.2.3.1 Get Data Start

A Decoder is allowed to use the cutout after this packet if and only if the immediately preceding DCC packet was addressed to that Decoder, received without error, and recognized by the Decoder as a packet which a standard defines as a command with a Sequence of **Get Data** messages, such as **Select** or **ReadBlock**.

► {preamble} 0 11111110 0 00000000 0 {checksum} 1

◄ [ch1+ch2]: {upstream data chunk}

6.2.3.2 Get Data Continue

► {preamble} 0 11111110 0 00000001 0 {checksum} 1

◄ [ch1+ch2]: {upstream data chunk}

6.2.3.3 Get Data Feedback

The data sent by the Decoder begins in the **Get Data Start** feedback channel and may continue in subsequent **Get Data Continue** feedback channels. The upstream data chunks are transmitted using the Variable Length Feedback format (Section 2.2.4 above). The Decoder shall use the Data space number as the seed value for the CRC-8 computation of the response.

The System shall send as many **Get Data Continue** messages as needed for the addressed Decoder to transmit the payload and a concluding CRC-8 byte as indicated by the header. If a Decoder receives additional **Get Data Continue** messages after all payload bytes and the CRC-8 byte is transferred, the Decoder shall reply with an ACK.

Send 31 data bytes Example:

► `{preamble} 0 11111110 0 00000000 0 {checksum} 1` – **Get Data Start**

◄ `[ch1+ch2]: {header: length = 31} {5 data bytes}`

► `{preamble} 0 11111110 0 00000001 0 {checksum} 1` – **Get Data Continue**

◄ `[ch1+ch2]: {6 data bytes}`

► `{preamble} 0 11111110 0 00000001 0 {checksum} 1` – **Get Data Continue**

◄ `[ch1+ch2]: {6 data bytes}`

► `{preamble} 0 11111110 0 00000001 0 {checksum} 1` – **Get Data Continue**

◄ `[ch1+ch2]: {6 data bytes}`

► `{preamble} 0 11111110 0 00000001 0 {checksum} 1` – **Get Data Continue**

◄ `[ch1+ch2]: {6 data bytes}`

► `{preamble} 0 11111110 0 00000001 0 {checksum} 1` – **Get Data Continue**

◄ `[ch1+ch2]: {2 data bytes} {CRC-8} ACK ACK ACK ACK`

In the last feedback it is shown how to stuff the feedback cutout with ACK datagrams to the full length.

Send 2 data bytes Example:

► `{preamble} 0 11111110 0 00000000 0 {checksum} 1` – **Get Data Start**

◄ `[ch1+ch2]: {header: length = 2} {2 data bytes} {CRC-8} 0000 ACK ACK`

In the feedback it is shown how to pad a partial codepoint in the feedback cutout and stuff with ACK datagrams to the full length.

6.2.4 Set Data

Reserved for future definition.

6.2.5 Logon Assign

Assigns a session specific address to a Decoder. The address is only valid for the current session. For permanent assignment, a separate CV programming operation must be completed. This message shall not modify the value of CV1, CV17, CV18, or CV19 for mobile Decoders and CV1(513) or CV9(521) for accessory Decoders.

Please see Section 6.4.3 below for CV19 specific behavior following this assignment.

► 11111110 0 1110HHHH 0 HHHHHHHH 0 UUUUUUUU 0 UUUUUUUU 0 UUUUUUUU 0 UUUUUUUU 0 11AAAAAA 0 AAAAAAAA 0 {checksum} 1

◄ [ch1+ch2]: {ID13}FFFF FFFFCCCC CCCCCCCC PPPPPPPP PPPPPPPP CRC-8

Parameter	Description
HHHH..HHHH	Manufacturer ID, 12-bits, four MSb first followed by eight LSb
UUUU..UUUU	Decoder Unique ID, four bytes, MSB first, LSB last
AAAA..AAAA	Decoder address, see Section 3 above.
FFFFFFFF	Change Flags[5] FFFFFFFF: Reserved, send as 0, ignore upon receipt FFFFFFFF: Decoder is in a CV19 consist, CV19 != 0 FFFFFFFF: GUI data (such as locomotive name/image) have been changed FFFFFFFF: The function assignment of the Decoder has changed FFFFFFFF: The driving behavior of the Decoder has changed[6] FFFFFFFF: The firmware of the Decoder has changed FFFFFFFF: The change flags were last reset on a System with different ID
CCCC..CCCC	Change counter. This value is changed every time the Decoder is programmed with new configuration, including if its firmware is updated. A value of 0xFFF corresponds to a Decoder, which is new, factory reset, or firmware updated, and is always regarded as a "changed" value.
PPPP..PPPP	Protocol Support Flags - overlaps with capabilities space bytes 2 and 3 as defined in Section 7.4.1 below. PPPPPPPP PPPPPPPP: Byte 0 PPPPPPPP PPPPPPPP: Byte 1

6.3 System Behavior

Each time a System is restarted, it shall increment its session ID. After Session ID 255, it shall roll over to Session ID 0.

6.3.1 Registration

When the System starts, it sends **a Logon Enable (Now)** to register all Decoders. Depending on the type of feedback system (local versus global detectors), the reaction will be a series of **ID15** messages or detected collisions.

The Decoders identified by a correct reception of an **ID15** feedback message are immediately addressed by DID with a **Select** message. Once the Decoder receives a **Select** message addressed to its DID, it stops responding to **Logon Enable** messages.

[5] See Section 6.2.2.3 Set Decoder Internal Status for instructions on clearing these flags.

[6] Motor control parameters such as min/mid/max speed, speed tables, braking rate/distance, acceleration/deceleration rate, etc…

If collisions are detected, the back off phase is started. The System sends a sequence of **Logon Enable (All)**. Each Decoder not yet addressed by **Select** will execute the back-off algorithm described in Section 6.4.2 below, and send replies to only some of these **Logon Enable** messages, thereby increasing the chance of a successful reception of the **ID15** message.

If no more new Decoders are received, and no more collisions are detected, the System can switch back to **Logon Enable (Now)** in order to enable new Decoders to be registered quickly.

6.3.2 Configuration Discovery

Reading of Decoder configuration can be accomplished using different methods depending on what method(s) the Decoder supports. The **Short Info** feedback to the Select (**Read Short Info**) command determines what methods are supported by the Decoder.

All Decoders that implement Logon shall support **POM** commands. Support for **XPOM** is highly encouraged because it is significantly more efficient for reading out configuration data. Support for **Select** + **Read**, **ReadShort**, and **ReadBackground** is optional, but offers even greater read efficiency.

6.4 Decoder Behavior

6.4.1 Startup

A startup of the Decoder can either be a fresh start or just a temporary loss of connectivity (dirty track for example). In addition, the Decoder does not know whether it is being controlled by a System that implements Logon.

If no message with **Logon Enable** is received within 700 milliseconds after startup, the Decoder shall proceed by using its permanent DCC address (CV1/17/18) and consist address if set (CV19).

If a Decoder detects a motor that is already turning when it starts, it may assume that there is lack of contact and can accept control commands at the previous DCC address (which may be permanent or an assigned address).

Upon receiving the first **Logon Enable**, the Decoder checks the CID and Session ID values against the stored values from the last successful address assignment. If the CID matches and the Session ID is the same or has been incremented by less than 4, the Decoder shall start in Selected state, with the assumption that the Decoder is known to the System. The Decoder shall accept control commands at the previously assigned DCC address and skip the Logon procedure. Otherwise the Decoder shall start in Unselected state.

The Decoder shall remain at zero speed and display an error condition in all of the following cases:

- it is assigned the address 0 with **Logon Assign**.
- it responded to three consecutive[7] **Logon Enable (Now)** and never received a valid Logon Assign.

For a mobile Decoder, an error condition shall be indicated by a double blink of both the front and rear lights. The total period shall be between one and two seconds, (on - off - on - off - blank). This requirement is waived if the model does not have the appropriate hardware necessary to support it, and the manufacturer is recommended to fully document an alternative method of displaying an error.

[7] Consecutive does not mean back-to-back DCC packets, rather without any other type of **Logon Enable** between the three **Logon Enable (Now)** packets.

6.4.2 Back-off

If a Decoder does not receive a **Select** confirmation after an attempted registration, it no longer replies to a certain number of **Logon Enable** messages. The number of **Logon Enable** messages to be ignored is based on a random number, which may be generated by the DID. The first random number is to be chosen from a range of 0 to 7. If the Logon is not activated again, the number is selected from a range of 0 to 15. If the Logon is not activated again, the number is selected from a range of 0 to 31. If the Logon is not activated again, the number is selected from a range of 0 to 63, and no further.

If a **Logon Enable (Now)** is received, the Decoder resets the current back-off value and immediately tries to register again.

If the Decoder contains a True Random Number Generator (TRNG), it may be used to generate a random value used in Back-off.

[The random number algorithm requirements are currently under reevaluation.]

6.4.3 CV19 Consisting Behavior

If a Decoder reverts to using its permanent address it shall also honor and act upon its programmed CV19 value. Otherwise, a Decoder shall ignore its CV19 consist address even if set until the System can reaffirm it with the Decoder. If reaffirmed, the Decoder shall honor the affirmed CV19 consist address until the Decoder gets back into Unselected state.

Affirmation is accomplished by reprogramming CV19 using either a **POM** or **Consist Control** instruction, which the Decoder is shall acknowledge once successful.

The feedback response to Logon Assign provides a flag indicating to the system if the current CV19 consist address is set (!= 0).

7 Data Spaces

Data Spaces are used for the efficient transfer of large amounts of data between a Decoder and the System.

Normally, the CRC-8 seed value used by this Standard is always 0, as described in section 2.1 above. For **ReadBackground** and **ReadBlock**, the CRC-8 seed value used in the feedback response as defined in section 2.2.4 above corresponds to the Data Space number, as defined in Section 0 below.

7.1 WriteBlock

This message can be used to perform writes into a Data Space.

► `{preamble} 0 11111101 0 11AAAAAA 0 AAAAAAAA 0 11111100 0 NNNNNNNN 0 VVVVVVVV 0 VVVVVVVV 0 VVVVVVVV 0 {payload byte(s)} 0 {checksum} 1`

◄ `[ch2]: ACK`

Parameter	Description
`AAAA..AAAA`	Decoder address, see Section 3 above.
`NNNNNNNN`	Data space number
`VVVV..VVVV`	Dataspace offset, three bytes MSB first, LSB last
payload byte(s)	Data that will be written from the given offset

7.1.1 WriteBlock Feedback

The Acknowledgement returned on the **WriteBlock** message indicates the error-free reception of the packet by the addressed Decoder. To notify the System about the **WriteBlock** operation, the Decoder sends response messages in channel 2 on any DCC packet addressed to that Decoder:

► Any addressed DCC message

◄ `[ch2]: ID13 1CCCCCCC 11111100 {optional argument word, MSB-first}`

1CCCCCCC	Argument Word	Description
10000001	optional	Executed Successfully.
	0x8000	Written data differs from data sent by the System[8].
10000010	{optional error code}	Failed with a permanent error.
10000011	{optional error code}	Decoder busy or write encountered a temporary error and may be re-tried later.
10000000	n/a	The write operation is still in progress.

Optional error codes are defined in Section 2.6 above.

If the Decoder does not send any write feedback to the System for 700 msec, the System shall assume that the write operation was lost. The Decoder shall send an operation still in progress feedback at least every 500 msec (or the next available opportunity) until the **WriteBlock** is completed or fails. The System shall provide the Decoder with frequent enough addressed packets to have an opportunity to meet the feedback requirement.

7.1.2 WriteBlock Continue

After a **WriteBlock** command is completed successfully according to the **WriteBlock** feedback, the System may send an Addressed Continue message to continue writing additional data from the offset where the previous write ended. The System shall not send any other 253-Addressed command to the given Decoder between the **WriteBlock** and the Addressed Continue command.

Long write sequence:

► `{preamble} 0 11111101 0 11AAAAAA 0 AAAAAAAA 0 11111100 0 NNNNNNNN 0 VVVVVVVV 0 VVVVVVVV 0 VVVVVVVV 0 {payload byte(s)} 0 {checksum} 1` - **WriteBlock**

◄ `[ch2]: ACK`

► Any addressed DCC message

◄ `[ch2]: ID13 10000001 11111100` - **Write Successful**

► `{preamble} 0 11111101 0 10AAAAAA 0 AAAAAAAA 0 {payload byte(s)} 0 {checksum} 1` - **Addressed Continue** (for **WriteBlock**)

◄ `[ch2]: ACK`

[8] The Decoder is not required to implement this feature.

► Any addressed DCC message

◄ `[ch2]: ID13 10000001 11111100` - **Write Successful**

► `{preamble} 0 11111101 0 10AAAAAA 0 AAAAAAAA 0 {payload byte(s)} 0 {checksum} 1` - **Addressed Continue** (for **WriteBlock**)

◄ `[ch2]: ACK`

► Any addressed DCC message

◄ `[ch2]: ID13 10000001 11111100` - **Write Successful**

etc…

If the System sends out a **WriteBlock Continue** command and receives no acknowledgement, or an error feedback, the System shall not re-try the **WriteBlock Continue** command. Instead, the **WriteBlock** command shall be issued with an explicit specification of the write offset.

If the Decoder receives an **Addressed Continue** command without having internal state about the previous command, the command shall be rejected with temporary error, with the recommended [optional] error code of 0x2040 or 0x2041.

7.2 ReadBackground

ReadBackground data is transmitted in feedback channel 2 of addressed DCC packets, including addressed packets using the 253 address.

► `{preamble} 0 11111101 0 11AAAAAA 0 AAAAAAAA 0 11111101 0 NNNNNNNN 0 (VVVVVVVV 0 VVVVVVVV 0 VVVVVVVV 0 SSSSSSSS 0) {checksum} 1` - **ReadBackground**

◄ `[ch2]: ACK`

► Any addressed DCC message

◄ `[ch2]: ID13 {upstream data chunk}`

► Any addressed DCC message

◄ `[ch2]: ID14 {upstream data chunk}`

etc…

Parameter	Description
`AAAA..AAAA`	Decoder address, see Section 3 above.
`NNNNNNNN`	Data space number
`VVVV..VVVV`	Data space number, three bytes MSB first, LSB last
`SSSSSSSS`	[optional] Number of requested bytes.

When the Data Space offset is not present, it is assumed to be 0. When the number of requested bytes is not specified, the Decoder continues responding until the total size of the Data Space is exhausted.

The first upstream data chunk is transmitted using **ID13**, all further chunks using **ID14**. The upstream data chunks are transmitted using the Variable Length Feedback format (Section 2.2.4

above). The Decoder shall use the Data Space number as the seed value for the CRC-8 computation of the response.

7.3 ReadBlock

► `{preamble} 0 11111101 0 11AAAAAA 0 AAAAAAAA 0 11111110 0 NNNNNNNN 0 (VVVVVVVV 0 VVVVVVVV 0 VVVVVVVV 0 SSSSSSSS 0) {checksum} 1` - **ReadBlock**

◄ `[ch2]: ACK`

► `{preamble} 0 11111110 0 00000000 0 {checksum} 1` – **Get Data Start**

◄ `[ch1+ch2]: {header} {5 data bytes}`

► `{preamble} 0 11111110 0 00000001 0 {checksum} 1` – **Get Data Continue**

◄ `[ch1+ch2]: {6 data bytes}`

etc...

Parameter	Description
`AAAA..AAAA`	Decoder address, see section 3 above.
`NNNNNNNN`	Data space number
`VVVV..VVVV`	Data space number, three bytes MSB first, LSB last
`SSSSSSSS`	[optional] Number of requested bytes.

When Data space offset is not present, it is assumed to be 0. When the number of requested bytes is not present, the Decoder continues responding until the total size is exhausted.

The upstream data chunks are transmitted using the Variable Length Feedback format (Section 2.2.4 above). The Decoder shall use the Data space number as the seed value for the CRC-8 computation of the response.

The Decoder shall respond to **Get Data Continue** with an **ACK** if no more data is available.

7.3.1 ReadBlock Errors

The **ReadBlock** request shall be rejected if CV28 bit 7 is clear, as described in Section 2.2.2 above.

► `{preamble} 0 11111101 0 11AAAAAA 0 AAAAAAAA 0 11111110 0 NNNNNNNN 0 (VVVVVVVV 0 VVVVVVVV 0 VVVVVVVV 0 SSSSSSSS 0) {checksum} 1` - **ReadBlock**

◄ `[ch2]: ID13 1CCCCCCC 11111110 {optional argument word, MSB-first}`

1CCCCCCC	Argument Word	Description
`10000010`	{optional error code}	Failed with a permanent error.

Optional error codes are defined in Section 2.6 above.

7.4 Data Space Definitions

Data Space	Description
0	Capabilities
1	Data Space Info
2	Short Gui
3	Configuration Variables
4 – 255	Reserved for future use

7.4.1 Capabilities

Bytes 0 and 1 are also transmitted in response to a **Select (ReadShortInfo)** command as defined in Section 6.2.2.1 above. Bytes 2 and 3 are also transmitted in response to a **Logon Assign** command as defined in Section 6.2.5 above.

Byte(s)	Bits(s)	Description
0	7	Reserved, send as 0, ignore upon receipt
	6	XPOM supported
	5	Select + ReadBlock supported
	4	ReadBlock supported
	3	ReadBackground supported
	2..0	Reserved, send as 0, ignore upon receipt
1	7..4	Reserved, send as 0, ignore upon receipt
	3	CV Read/Write data space supported
	2	Short GUI data space supported
	1	Data Space Info data space supported
	0	Capabilities data space supported
2	7	Addressed S-9.2 / S-9.2.1 Chained supported
	6..0	Reserved, send as 0, ignore upon receipt
3	7..1	Reserved, send as 0, ignore upon receipt
	0	Extended capabilities supported
4..30		Extended capabilities, reserved for future expansion. Bytes beyond the last supported extended capability feature supported shall be omitted.

7.4.2 Data Space Info

This Data Space contains one bit for every possible Data Space number (256 total). A bit is 1 if the given space number is supported, 0 if not supported. Bytes beyond the last supported Data Space shall be omitted.

Byte(s)	Bits(s)	Description

0	7	Data Space 7
	6..1	Data Spaces 6..1
	0	Data Space 0
1	7	Data Space 15
	6..1	Data Spaces 4..9
	0	Data Space 8
2..30		etc…

7.4.3 Short GUI

Byte(s)	Bits(s)	Description
0..7		User defined [short] name (UTF-8 format, 0x00 padded)
8..9		Image index, to be defined…
10	6..7	Function information f0
	5..4	Reserved, send as 0, check upon receipt
	3..0	Principal Symbol, see table below
11..28		F1..F68 function information, two bits per function, F1 starts with least significant byte and bits, see table below. Bytes beyond the last "configured" function shall be omitted.

Index	Principal Symbol (mobile Decoder)	Principal Symbol (accessory Decoder)
0000	Steam Locomotive	Turnout/Switch
0001	Diesel Locomotive	Signal
0010	Electric Locomotive	Turntable
0011	Diesel Railcar	Lighting
0100	Electric Railcar	Traffic Light
0101	Cab Car, FRED (End of Train)	
0110	Passenger Car	
0111	Caboose	
1000	Maintenance of Way	
1001	Generic Function Decoder	
1010	Automobile (Car)	
1011	Automobile (Truck)	
1100	Automobile (Bus)	
1101	Automobile (Other)	
1110	Other Mobile Decoder	

Index	Principal Symbol (mobile Decoder)	Principal Symbol (accessory Decoder)
1111	Other (non-mobile) Decoder[9]	Other

495

[9] A “mobile” decoder used in a stationary applications such as building lighting, stationary crane, etc…

Bits 1..0	Function Information
00	Function undefined
01	Function available and latching
10	Function available and momentary, where the effect stops immediately upon releasing of the button (e.g. horn)
11	Function available and trigger, which cannot be re-triggered for a certain period of time (e.g. announcement)

7.4.4 Configuration Variables (CV's)

The offset (24-bits) is always defined as the indexed CV address whereby the first byte corresponds to CV31 and the second byte corresponds to CV32.

7.4.5 [Indexed] CV Space Overlay Mapping

Data Spaces are also (dual) mapped to the CV Space and therefore can be read out by using POM or XPOM. When this mapping occurs, the Data Space length in bytes is prepended in the CV mapping.

CV31	CV32	Offset	Max Size (bytes)	Data Space
2	0	0	1	Capabilities Length (4 - 31)
		1	31	Capabilities Data
	1	0	1	Data Space Info Length (1 - 30)
		1	31	Data Space Info Data
	2	0	1	Short GUI Length (11 - 29)
		1	28	Short GUI Data
	3	-	-	Overlaps with CV Data Space, unimplemented
	4..255			Reserved for future use

8 Manufacturer Specific Command Space

Manufacturers that wish to define their own custom command(s) without seeking permission from the NMRA DCC WG shall use the following packet format. It is left entirely up to the manufacturer on how to define the payload bytes. The total packet length, including checksum, shall be kept to 16 bytes or less. This results in the number of manufacturer defined payload bytes being 9 or less.

The feedback shall contain an ack or any other valid feedback message, as defined in Section 2.2.1 above.

► `{preamble} 0 11111101 0 11AAAAAA 0 AAAAAAAA 0 0000HHHH 0 HHHHHHHH 0 {manufacturer defined payload} 0 {checksum} 1`

◄ `[ch2]: ACK`

Parameter	Description
`HHHH..HHHH`	Manufacturer ID, 12-bits, four MSb first followed by eight LSb

9 Document History

Date	Description
	First Revision

10 Table of Contents

Important Notices and Disclaimers Concerning NMRA Standards Documents

The Standards (S), Recommended Practices (RP), Technical Note (TN), and Technical Information (TI) documents of the National Model Railroad Association ("NMRA Standards documents") are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page, appear in all standards and may be found under the heading "Important Notices and Disclaimers Concerning NMRA Standards Documents."

Notice and Disclaimer of Liability Concerning the Use of NMRA Standards Documents

NMRA Standards documents are developed within the Standards and Conformance Department of the NMRA in association with certain Working Groups, members, and representatives of manufacturers and sellers. NMRA develops its standards through a consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. NMRA Standards documents are developed by volunteers with modeling, railroading, engineering, and industry-based expertise. Volunteers are not necessarily members of NMRA, and participate without compensation from NMRA.

NMRA does not warrant or represent the accuracy or completeness of the material contained in NMRA Standards documents, and expressly disclaims all warranties (express, implied and statutory) not included in this or any other document relating to the standard or recommended practice, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; and quality, accuracy, effectiveness, currency, or completeness of material. In addition, NMRA disclaims any and all conditions relating to results and workmanlike effort. In addition, NMRA does not warrant or represent that the use of the material contained in NMRA Standards documents is free from patent infringement. NMRA Standards documents are supplied "AS IS" and "WITH ALL FAULTS."

Use of NMRA Standards documents is wholly voluntary. The existence of an NMRA Standard or Recommended Practice does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the NMRA Standards documents. Furthermore, the viewpoint expressed at the time that NMRA approves or issues a Standard or Recommended Practice is subject to change brought about through developments in the state of the art and comments received from users of NMRA Standards documents.

In publishing and making its standards available, NMRA is not suggesting or rendering professional or other services for, or on behalf of, any person or entity, nor is NMRA undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any NMRA Standards document, should rely upon their own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given NMRA Standards document.

IN NO EVENT SHALL NMRA BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: THE NEED TO PROCURE SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD OR RECOMMENDED PRACTICE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

NMRA's development of NMRA Standards documents involves the review of documents in English only. In the event that an NMRA Standards document is translated, only the English version published by NMRA is the approved NMRA Standards document.

Official Statements

A statement, written or oral, that is not processed in accordance with NMRA policies for distribution of NMRA communications, or approved by the Board of Directors, an officer or committee chairperson, shall not be considered or inferred to be the official position of NMRA or any of its committees and shall not be considered to be, nor be relied upon as, a formal position of NMRA.

Comments on Standards

Comments for revision of NMRA Standards documents are welcome from any interested party, regardless of membership. However, **NMRA does not provide interpretations, consulting information, or advice pertaining to NMRA Standards documents.**

Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since NMRA standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, NMRA, its departments, Working Groups or committees cannot provide an instant response to comments, or questions except in those cases where the matter has previously been addressed. For the same reason, NMRA does not respond to interpretation requests. Any person who would like to participate in evaluating comments or in revisions to NMRA Standards documents may request participation in the relevant NMRA working group.

Laws & Regulations

Users of NMRA Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any NMRA Standards document does not constitute compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. NMRA does not, by the publication of NMRA Standards documents, intend to urge action that is not in compliance with applicable laws, and NMRA Standards documents may not be construed as doing so.

Copyrights

NMRA Standards documents are copyrighted by NMRA under US and international copyright laws. They are made available by NMRA and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of modeling, structural and engineering practices and methods. By making NMRA Standards documents available for use and adoption by public authorities and private users, NMRA does not waive any rights in copyright to the NMRA Standards documents.

IMPORTANT NOTICE

NMRA Standards documents do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other systems, devices or networks. NMRA Standards documents development activities consider research and information presented to the standards development group in developing any safety recommendations. Other information about safety practices, changes in technology or technology implementation, or impact by peripheral systems also may be pertinent to safety considerations during implementation of the standard. Implementers and users of NMRA Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.