1 General

The NMRA Communications Standard for Digital Communications (S-9.2) provides a minimal, basic packet format required for interoperability. This STANDARD provides Extended Packet Formats that provide the framework to enable realistic operations to occur.

1.1 Introduction and Intended Use (Informative)

These formats adhere to the general packet format as defined in S-9.2. While the baseline packet has a length of 3 data bytes separated by a "0" bit, a packet using the extended packet format definition may have a length of between 3 and 6 data bytes each separated by a "0" bit.

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-9.1 DCC Electrical Standard
- S-9.2 DCC Communication Standard
- S-9.2.1.1 DCC Advanced Extender Packet Formats
- S-9.3.2 DCC Bi-Directional Communications

1.2.2 Informative

- TN-3.05 Electrical Specifications for Digital Command Control Decoder Transmission
- TN-4.05 Electrical Specifications for Digital Command Control Decoder Transmission
- RCN-210 DCC Protocol Bit Transmission
- RCN-211 DCC Packet Structure
- RCN-212 DCC Operating Commands for vehicle decoders
- RCN-213 DCC Operating commands for accessory decoders
- RCN-214 DCC configuration commands

There has been a concentrated effort to harmonize the Standards and Rail Community Norms listed above.

1.3 Terminology

Term	Definition
Accessory decoders	DCC receiver for controlling stationary device animation.
Broadcast command	Using address 00000000 the command is sent to be available to all decoders.
Consist	Two or more decoders responding to the same commands. See S-9.2.2 CV19 for more information.
Mobile decoders	DCC receiver for controlling vehicle animation.
Multifunction decoders	Commonly called a mobile decoder, used to control multiple functions such a speed, direction, lighting and or sound.
Vehicle	Mobile model railroad device. This includes locomotives and other rolling stock.

2 Format Definitions

Within this Standard, bits within the address and data bytes will be defined using the following abbreviations. Individual bytes within a specific packet format are separated by spaces. Bytes which are within square [ ] brackets can occur one or more times as necessary. Bytes are separated by a framing 0. The last byte is a XOR error detection byte followed by a 1. Bits are numbered from right to left with bit 0 (the right most bit) being the least significant bit (LSB) and bit 7 (the left most bit) being the most significant bit (MSB).

A = Address bit
0 = Bit with the value of "0"
1 = Bit with the value of "1"
U = bit with undefined value either a "1" or a "0" is acceptable
B = Bit Position / Address
C = Instruction Type field
G = Instruction Sub Type
T = Instruction Tertiary Type
F = Flag to determine Instruction Implementation
L = Low Order Binary State Control Address
H = High Order Binary State Control Address
S = Decoder Sub Address
V = CV Address Bit
D = Data
X = Signal Head Aspect Data Bit
E = Error Detection Bit

2.1 Address Partitions

The first data byte of an Extended Packet Format packet contains the primary address. In order to allow for different types of decoders this primary address is subdivided into fixed partitions as follows.

Address 00000000 (0):	Broadcast address
Addresses 00000001-01111111 (1-127)(inclusive):	Multi-Function decoders with 7 bit addresses
Addresses 10000000-10111111 (128-191)(inclusive):	Basic Accessory Decoders with 9 bit addresses and Extended Accessory Decoders with 11-bit addresses. See 2.4 below for further information.
Addresses 11000000-11100111 (192-231)(inclusive):	Multi-Function Decoders with 14 bit addresses. See 2.3 below for further information.
Addresses 11101000-11111100 (232-252)(inclusive):	Reserved for Future Use
Addresses 11111101-11111110 (253-254)(inclusive):	Advanced Extended Packet Formats (refer to S-9.2.1.1)
Address 11111111 (255):	Idle Packet

2.2 Broadcast Command for Multi-Function Digital Decoders

The format for this packet is:

{preamble} 0 00000000 0 {instruction-bytes} 0 EEEEEEEE 1

Instructions addressed to "broadcast address" 00000000 must be executed by all Multi-Function Digital Decoders. The single instruction has the same definition as defined by the Multi-Function Digital Decoder packet and can be one, two, or three bytes in length depending on the instruction. Digital Decoders should ignore any instruction they do not support. The manufacturer must document which broadcast commands a decoder supports.

2.3 Instruction Packets for Multi-Function Digital Decoders

The formats for these packets are:

{preamble} 0 [AAAAAAAA] 0 {instruction-bytes} 0 EEEEEEEE 1

Multi-Function Digital Decoders are used for the purpose of controlling one or more motors and/or accessories. The packet format used to control these devices consists of between 3 and 6 bytes where the first bytes are address bytes followed by one or two instruction bytes and ended by an error control byte.

The primary address byte contains 8 bits of address information. If the most significant bits of the address are between 1100-0000 and 1110-0111 (192-231) (inclusive) then a second address byte must immediately follow. This second address byte will then contain an additional 8 bits of address data. When 2 bytes of address information are present they are separated by a "0" bit. The most significant bit of two byte addresses is bit 5 of the primary address byte, bits 6 and 7 having the value of "1" in this case. The second address byte will contain the address to receive the instructions.

{preamble} 0 [AAAAAAAA] 0 [AAAAAAAA] 0 {instruction-bytes} 0 EEEEEEEE 1

Instruction-bytes are data bytes used to send commands to Multi-Function Digital Decoders. Although it is unlikely that all Digital Decoders will implement all instructions, it is important that if they support packets with more than a single instruction, they can sufficiently parse the packet to be able to recognize if a byte is a new instruction or the second byte of a previous instruction.

Each instruction (indicated by {instruction-bytes}) is defined to be:

{instruction-bytes} = CCCDDDDD,
    CCCDDDDD 0 DDDDDDDD, or
    CCCDDDDD 0 DDDDDDDD 0 DDDDDDDD

Each instruction consists of a 3-bit instruction type field followed by a 5-bit data field. Some 5-bit data fields may contain Instruction Subtypes, Instruction Tertiary-types as well as Flags.

CCCDDDDD = CCCGGGGG or CCCGTTTT
See specific instruction for details. Some instructions have one or two or three bytes of data. The 3-bit instruction type field is defined as follows where CCC is equal to the following 3 bits:

000 Decoder and Consist Control Instruction (see 2.3.1)
001 Advanced Operation Instructions ~~(CCC=001) 001GGGGG~~ (see 2.3.1)
010 Speed and Direction Instruction for reverse operation (see 2.3.3)
011 Speed and Direction Instruction for forward operation (see 2.3.3)
100 Function Group One Instruction (see 2.3.4)
101 Function Group Two Instruction (see 2.3.5)
110 Feature Expansion (see 2.3.6)
111 Configuration Variable Access Instruction (long and short forms see 2.3.7)

The last byte of the packet is the Error Detection Byte, which is calculated the same as is done in the baseline packet using all address, and all instruction bytes (see S-9.2).

2.3.1 Decoder and Consist Control Instruction (CCC=000)

With the exception of the decoder acknowledgment function (00001111), only a single decoder and consist control instruction may be contained in a packet.

2.3.1.1 Decoder Control (CCCG = 0000)

The decoder control instructions are intended to set up or modify decoder configurations.

This instruction has the format of:
{instruction byte} = 0000TTTF, or {instruction byte} = 0000TTTF 0 DDDDDDDD

This instruction (0000TTTF) allows specific decoder features to be set or cleared as defined by the value of F ("1" indicates set). When the decoder has decoder acknowledgment enabled, receipt of a decoder control instruction shall be acknowledged with an operations mode acknowledgment.

TTT = 000 F = "0": Digital Decoder Reset - A Digital Decoder Reset shall erase all volatile memory (including and speed and direction data), and return to its initial power up state as defined in S-9.2.4 section A. Command Stations shall not send packets to addresses 112-127 for 10 packet times following a Digital Decoder Reset. This is to ensure that the decoder does not start executing service mode instruction packets as operations mode packets (Service Mode instruction packets have a short address in the range of 112 to 127 decimal.)
F = "1": Hard Reset - Configuration Variables 29, 31 and 32 are reset to its factory default conditions, CV#19 is set to "00000000" and a Digital Decoder reset (as in the above instruction) shall be performed.
TTT = 001 Factory Test Instruction - This instruction is used by manufacturers to test decoders at the factory. It must not be sent by any command station during normal operation. This instruction may be a multi-byte instruction.
TTT = 010 Reserved for future use
TTT = 011 Reserved for future use
TTT = 100 Reserved for future use
TTT = 101 Set Advanced Addressing (CV#29 bit 5=F) (see 2.3.1.2)
TTT = 110 Reserved for future use
TTT = 111 F=1 Decoder Acknowledgment Request (see 2.3.1.3)

2.3.1.2 Set Advanced Addressing (TTT = 101) 000GTTTF

This command is one byte long and has the format of: {instruction bytes} = 0000101F
This command is to select either 14 bit (long) addressing or 7 bit (short) addressing, where F=Bit 5 of CV29 and must be set to 0 for a short address (stored in CV1) or 1 for a long address (stored in CV17 & CV18). CV17 contains the most significant bits of the two byte address and must have a value between 11000000 and 11100111 inclusive in order for this two byte address to be valid. CV 18 contains the least significant bits of the address and may contain any value. CV17 and CV18 should be written at the same time to avoid problems. Refer to S-9.2.2 for additional information.

2.3.1.3 Decoder Acknowledgement Request (TTT = 111) 000GTTTF

This command is one byte long and has the format of: {instruction bytes} = 00001111
Only an acknowledgment of the command is expected.

2.3.1.4 Consist Control (CCCG = 0001)

This instruction controls consist setup and activation or deactivation.

When Consist Control is in effect, the decoder will ignore any speed or direction instructions addressed to its normal locomotive address (unless this address is the same as it's consist address). Speed and direction instructions now apply to the consist address only.

Functions controlled by Function Group One (100) and Function Group Two (101) will continue to respond to the decoders baseline address. Functions controlled by instructions 100 and 101 also respond to the consist address if the appropriate bits in CVs 21 and 22 have been activated. See S-9.2.2 for more information.

By default all forms of Bi-directional communication are not activated in response to commands sent to the consist address until specifically activated by a Decoder Control instruction. Operations mode acknowledgement and Data Transmission applies to the appropriate commands at the respective decoder addresses.

The format of this instruction is: {instruction bytes} = 0001TTTT 0 0AAAAAAA

A value of "1" in bit 7 of the second byte is reserved for future use. Within this instruction TTTT contains a consist setup instruction, and the AAAAAAA in the second byte is a seven bit consist address between 1-127. If the address is "0000000" then the consist is deactivated. If the address is non-zero, then the consist is activated. See S-9.2.2 CV19 for more information.

If the consist is deactivated (by setting the consist to '0000000'), the Bi-Directional communications settings are set as specified in CVs 26-28.

When operations-mode acknowledgement is enabled, all consist commands must be acknowledged via operations-mode acknowledgement.

The format for TTTT shall be:

TTTT=0010 Set the consist address as specified in the second byte, and activate the consist. The consist address is stored in bits 0-6 of CV19 and bit 7 of CV19 is set to a value of 0. The direction of this unit in the consist is the normal direction. If the consist address is 0000000 the consist is deactivated.

TTTT=0011 Set the consist address as specified in the second byte and activate the consist. The consist address is stored in bits 0-6 of CV 19 and bit 7 of CV19 is set to a value of 1. The direction of this unit in the consist is opposite its normal direction. If the consist address is 0000000 the consist is deactivated.

All other values of TTTT are reserved for future use.

2.3.2 Advanced Operations Instruction (CCC=001)

These instructions control advanced decoder functions. Only a single advanced operations instruction may be contained in a packet.

The format of this instruction is: {instruction bytes} = 001GGGGG 0 DDDDDDDD
The 5-bit sub-instruction GGGGG allows for 32 separate Advanced Operations Sub-Instructions.

2.3.2.1 GGGGG = 11111: 128 Speed Step Control

Instruction "11111" is used to send one of 126 *Digital Decoder* speed steps. The subsequent single byte (DDDDDDDD) shall define speed and direction with bit 7 being direction ("1" is forward and "0" is reverse) and the remaining bits used to indicate speed. The most significant speed bit is bit 6. A data-byte value of U0000000 is used for stop, and a data-byte value of U0000001 is used for emergency stop. This allows up to 126 speed steps. When operations mode acknowledgment is enabled, receipt of a 128 Speed Step Control packet must be acknowledged with an operations mode acknowledgement.

2.3.2.2 GGGGG = 11110: Reserved for Zimo East-West Direction Proposal

Information will be provided prior to implementation of this proposed function which is under development.

2.3.2.3 GGGGG = 11101: Analog Function Group

The format of this instruction is: {instruction bytes} = 00111101 0 VVVVVVVV 0 DDDDDDDD where;

VVVVVVVV - Analog Function Output (0-255)
DDDDDDDD - Analog Function Data (0-255)

Analog Output Use: 00000001 Volume Control

All other values of VVVVVVVV are reserved. This function must not be used to control the speed of a mobile decoder.

When operations mode acknowledgment is enabled, receipt of an Analog Function Group Instruction must be acknowledged with an operations mode acknowledgement.

2.3.2.4 GGGGG = 11100 thru 00000

The remaining 29 instructions are reserved for future use.

2.3.3 Speed and Direction Instructions (CCC=010 and CCC=011)

These two instructions have these formats:
for Reverse Operation: {instruction bytes} = 010DDDDD
for Forward Operation: {instruction bytes} = 011DDDDD

A speed and direction instruction is used to send information to motors connected to Multi-Function *Digital Decoders*. Instruction "010" indicates a Speed and Direction Instruction for reverse operation and instruction "011" indicates a Speed and Direction Instruction for forward operation. In these instructions, the data is used to control speed with bits 0-3 being defined exactly as in S-9.2 Section B. If Bit 1 of CV29 has a value of one (1), then bit 4 is used as an intermediate speed step, as defined in S-9.2, Section B. If Bit 1 of CV29 has a value of zero (0), then bit 4 shall be used to control FL[1]. In this mode, Speed =U0000 is stop, Speed =U0001 is emergency stop, Speed =U0010 is the first speed step and Speed =U1111 is full speed. This provides 14 discrete speed steps in each direction.

If a decoder receives a new speed step that is within one step of current speed step, the Digital Decoder may select a step half way between these two speed steps. This provides the potential to control 56 individual speed steps should the command station alternate speed packets. Decoders may ignore the direction information transmitted in a broadcast packet for Speed and Direction commands that do not contain stop or emergency stop information.
When operations mode acknowledgment is enabled, receipt of any speed and direction packet must be acknowledged with an operations mode acknowledgement.

[1] FL is used for the control of the headlights.

2.3.4 Function Group One Instruction (CCC=100)[2]

The format of this instruction is: {instruction bytes} = 100DDDDD

Up to 5 auxiliary functions (functions FL and F1-F4) can be controlled by the Function Group One instruction. Bits 0-3 shall define the value of functions F1-F4 with function F1 being controlled by bit 0 and function F4 being controlled by bit 3. A value of "1" shall indicate that the function is "on" while a value of "0" shall indicate that the function is "off". If Bit 1 of CV29 has a value of one (1), then bit 4 controls function FL, otherwise bit 4 has no meaning.

When operations mode acknowledgment is enabled, receipt of a function group 1 packet must be acknowledged according with an operations mode acknowledgement.

2.3.5 Function Group Two Instruction (CCC=101)[3]

This instruction has the format: {instruction bytes} = 101SDDDD

Up to 8 additional auxiliary functions (F5-F12) can be controlled by a Function Group Two instruction. Bit 4 defines the use of Bits 0-3. When bit 4 (S) is '1', Bits 0-3 (DDDD) shall define the value of functions F5-F8 with function F5 being controlled by bit 0 and function F8 being controlled by bit 3. When bit 4 (S) is '0', Bits 0-3 (DDDD) shall define the value of functions F9-F12 with function F9 being controlled by bit 0 and function F12 being controlled by bit 3. A value of "1" shall indicate that the function is "on" while a value of "0" shall indicate that the function is "off".

When operations mode acknowledgment is enabled, receipt of function group 2 packet shall be acknowledged according with an operations mode acknowledgement.

2.3.6 Feature Expansion Instruction (CCC=110)

The instructions in this group provide for support of additional features within decoders. (See TN-3-05)

The format of two byte instructions in this group is:
{instruction bytes} = 110GGGGG 0 DDDDDDDD

The format of three byte instructions in this group is:
{instruction bytes} = 110GGGGG 0 DDDDDDDD 0 DDDDDDDD

The 5-bit sub-instruction GGGGG allows for 32 separate Feature Expansion Sub-instructions.

2.3.6.1 GGGGG = 00000: Binary State Control Instruction long form[4]

Sub instruction "00000" is a three byte instruction and provides for control of one of 32767 binary states within the decoder. The two bytes following this instruction byte have the format DLLLLLLL 0 HHHHHHHH".

[2] Any function in this packet group may be directionally qualified.

[3] Any function in this packet group may be directionally qualified.

[4] Binary States 1-15 are reserved for NMRA Bidirectional communication see S-9.3.2.

Bits 0-6 of the first data byte (LLLLLLL) shall define the low order bits of the binary state address; bits 0-7 of the second data byte (HHHHHHHH) shall define the high order bits of binary state address. The addresses range from 1 to 32767. Bit 7 of the second byte (D) defines the binary state. A value of "1" shall indicate that the binary state is "on" while a value of "0" shall indicate that the binary state is "off". The value of 0 for the address is reserved as broadcast to clear or set all 32767 binary states. An instruction of
"11000000 0 00000000 0 00000000" sets all 32767 binary states to off.

Binary states accessed with all high address bits set to zero would be the same as accessed by the short form of the binary state control. Command stations shall use the short form in this case, i.e. Binary State Controls 1 to 127 should always be addressed using the short form. Decoders supporting the long form shall support the short form as well.

2.3.6.2 GGGGG = 00001: Time and Date Command

The command station should transmit model time at most once every (model) minute. The command station may skip the time command if other packets need the bandwidth. Skipped or missing times must be tolerated by the decoders and can be replaced (internal to the decoder) with the clock ratio. The date command is transmitted (at least three times) only if changed. Time and date commands are a broadcast to address 0.

The general format is;
{preamble} 0 [00000000] 0 [110-00001] 0 [CCxxxxxx] 0 [xxxxxxxx] 0 [xxxxxxxx] 0 EEEEEEEE 1

The Packet is always sent to broadcast short address 0. See the first bracket [00000000]. The command is four bytes. The value in CC determines the information in the packet.

When CC=00, Time Command. The format of the instruction is:
{preamble} 0 [00000000] 0 [110-00001] 0 [00MMMMMM] 0 [WWWHHHHH] 0 [U0BBBBBB] 0 EEEEEEEE 1

MMMMMM = minutes. Value range: 0..59

WWW = Day of the week. Value range: 0 = Monday, 1 = Tuesday, 2 = Wednesday, 3 = Thursday, 4 = Friday, 5 = Saturday, 6 = Sunday 7=not supported

HHHHH = hours. Value range: 0..23
U = Update. If U = 1 the time has changed significantly since the last update. This bit is normally 0.

BBBBBB = acceleration factor. Value range 0..63. 0 = clock has stopped, 1 = real time, 2 = clock runs 2x real time, 3 = 3x real time, 4= 4x real time etc.

When CC = 01, Date Command. The format is;
{preamble} 0 [00000000] 0 [110-00001] 0 [010TTTTT] 0 [MMMMYYYY] 0 [YYYYYYYY] 0 EEEEEEEE 1

TTTTT = Day of the month. Value range: 1..31

MMMM = Month. Value range: 1..12. 1 = January, 2 = February, 3 = March etc.

YYYYYYYYYYYY = year. Value range: 0..4095. Least significant bits in the 5th byte. Most significant bits in the 4th byte with the months.

CC = 10 reserved
CC = 11 reserved

2.3.6.3 GGGGG=00010: System time

The command for the system time is three bytes long and has the format:
{preamble} 0 [00000000] 0 [110-00010] 0 [MMMMMMMM] 0 [MMMMMMMM] 0 EEEEEEEE 1

The bits marked 'M', indicate milliseconds since system startup. The maximum value is 0xFFFF = 65535 and corresponds to about 65.5 seconds. The third byte contains the most significant bits, the fourth byte contains the least significant bits. When the maximum value is reached, the counter starts again at 0. When determining relative times of up to one minute can easily be worked with a 16 bit integer without an error due to an overflow.

This timestamp refers to the beginning of the start bit. If this feature is implemented it is recommended the command station send this packet once approximately every 30 seconds to ensure adequate synchronization.

2.3.6.4 GGGGG = 11101: Binary State Control Instruction short form[5]

Sub-instruction "11101" is a two byte instruction and provides for control of one of 127 binary states within the decoder. The single byte following this instruction byte has the format: {instruction bytes} = DLLLLLLL.

{preamble} 0 [AAAAAAAA] 0 [110-11101] 0 [DLLLLLLL] 0 EEEEEEEE 1

Bits 0-6 of the second byte (LLLLLLL) shall define the number of the binary state starting with 1 and ending with 127. Bit 7 (D) defines the binary state. A value of "1" shall indicate the binary state is "on" while a value of "0" shall indicate the binary state is "off". The value of 0 for LLLLLLL is reserved as broadcast to clear or set <u>all</u> 127 binary states accessible by the short form of the binary state control. An instruction "11011101 0 00000000" sets all 127 binary states accessed by this instruction to off.

Binary State Controls are quite similar to Functions, as they may control any output, sound or any other feature of digital nature within a decoder in direct response to a packet received. However, Binary State Controls do have a different access method and function space. Therefore they have a different name.

Binary state control packets – both short and long form – will not be refreshed. Therefore non-volatile storage of the function status is recommended. When operations mode acknowledgment is enabled, receipt of a Binary State Control packet shall be acknowledged accordingly with an operations mode acknowledgment. Consult the Technical Note(s) for additional information on this instruction. (See TN-4-05)

---

[5] Binary States 1-15 are reserved for NMRA Bidirectional communication see S-9.3.2.

2.3.6.5 GGGGG = 11110: F13-F20 Function Control

Sub-instruction “11110” is a two byte instruction and provides for control of eight (8) additional auxiliary functions F13-F20.

{preamble} 0 [AAAAAAAA] 0 [110-11110] 0 [DDDDDDDD] 0 EEEEEEEE 1

The single byte following this instruction byte indicates whether a given function is turned on or off, with the least significant bit (Bit 0) controlling the lower function. In this case F13, and the most significant bit (bit 7) controlling the higher function. In this case F20. A value of “1” in F for a given function shall indicate the function is “on” while a value of “0” in F for a given function shall indicate a given function is “off”. It is recommended, but not required, that the status of these functions be saved in decoder storage such as non~~e -~~-volatile random~~-~~-access memory (NVRAM). It is not required, and should not be assumed that the state of these functions is constantly refreshed by the command station. Command Stations that generate these packets, and which are not periodically refreshing these functions, must send at least two repetitions of these commands when any function state is changed. When operations mode acknowledgment is enabled, receipt of an F13-F20 Function Control packet shall be acknowledged accordingly with an operations mode acknowledgement. Consult the Technical Note(s), TN-4-05, for additional information on this instruction.

2.3.6.6 GGGGG = 11111: F21-F28 Function Control

Sub-instruction “11111” is a two byte instruction and provides for control of eight (8) additional auxiliary functions F21-F28.

The format of this instruction byte in this group is:
{instruction bytes} = 11011111 0 DDDDDDDD

The single byte indicates whether a given function is turned on or off, as described above, with the least significant bit (Bit 0) controlling the lower function. In this case F21, and the most significant bit (bit 7) controlling the higher function. In this case F28.

2.3.6.7 GGGGG = 11000: F29-F36 Function Control

Sub-instruction “11000” is a two byte instruction and provides for control of eight (8) additional auxiliary functions F29-F36.

The format of this instruction byte in this group is:

{instruction bytes} = 11011000 0 DDDDDDDD

The single byte indicates whether a given function is turned on or off, as described above, with the least significant bit (Bit 0) controlling the lower function. In this case F29, and the most significant bit (bit 7) controlling the higher function. In this case F36.

2.3.6.8 GGGGG = 11001: F37-F44 Function Control

Sub-instruction “11001” is a two byte instruction and provides for control of eight (8) additional auxiliary functions F37-F44.

The format of this instruction byte in this group is:

{instruction bytes} = 11011001 0 DDDDDDDD

The single byte indicates whether a given function is turned on or off, as described above, with the least significant bit (Bit 0) controlling the lower function. In this case F37, and the most significant bit (bit 7) controlling the higher function. In this case F44.

2.3.6.9 GGGGG = 11010: F45-F52 Function Control

Sub-instruction “11010” is a two byte instruction and provides for control of eight (8) additional auxiliary functions F45-F52.

The format of this instruction byte in this group is:

{instruction bytes} = 11011010 0 DDDDDDDD

The single byte indicates whether a given function is turned on or off, as described above, with the least significant bit (Bit 0) controlling the lower function. In this case F45, and the most significant bit (bit 7) controlling the higher function. In this case F52.

2.3.6.10 GGGGG = 11011: F53-F60 Function Control

Sub-instruction “11011” is a two byte instruction and provides for control of eight (8) additional auxiliary functions F53-F60.

The format of this instruction byte in this group is:

{instruction bytes} = 11011011 0 DDDDDDDD

The single byte indicates whether a given function is turned on or off, as described above, with the least significant bit (Bit 0) controlling the lower function. In this case F53, and the most significant bit (bit 7) controlling the higher function. In this case F60.

2.3.6.11 GGGGG = 11100: F61-F68 Function Control

Sub-instruction “11100” is a two byte instruction and provides for control of eight (8) additional auxiliary functions F61-F68.

The format of this instruction byte in this group is:

{instruction bytes} = 11011100 0 DDDDDDDD

The single byte indicates whether a given function is turned on or off, as described above, with the least significant bit (Bit 0) controlling the lower function. In this case F61, and the most significant bit (bit 7) controlling the higher function. In this case F68.

The remaining 23 sub-instructions are reserved by the NMRA for future use.[6]

2.3.7 Configuration Variable Access Instruction (CCC=111)

The Configuration Variable Access instructions are intended to set up or modify Configurations Variables either on the programming track or on the main line. There are ~~two~~three forms of this instruction. The short form is for modifying selected frequently modified Configuration Variables. The long form is for verifying or modifying any selected Configuration Variable. The XPOM form is for modifying or reading up to four CVs at one time using indexed addressing. Only a single configuration variable access instruction may be contained in a packet.

2.3.7.1 Configuration Variable Access Acknowledgment

If a configuration variable access acknowledgment is required, and the decoder has decoder operations-mode acknowledgment enabled, the decoder shall respond with an operations mode acknowledgment.

---

[6] The NMRA shall not issue a NMRA Conformance Warrant for any product that uses an instruction or sub-instruction that has been reserved by the NMRA.

2.3.7.2 Configuration Variable Access Instruction - Short Form

This instruction has the format of:

{instruction bytes} = 1111GGGG 0 DDDDDDDD 0 DDDDDDDD

The 8 bit data DDDDDDDD is placed in the configuration variable identified by GGGG according to the following table.

Table 2.1

GGGG	Description	CVs affected	Two Identical Packets Required
0000	Not available for use	N/A	N/A
0010	Acceleration Adjustment Value	CV23	NO
0011	Deceleration Adjustment Value	CV24	NO
0100	Long Address	CV17, CV18, CV29	YES
0101	Indexed CVs	CV31, CV32	YES
1001	See S-9.2.3 Appendix B		

NOTE: The 8-bit data in the second and possibly third command byte DDDDDDDD are stored configuration variables, which are defined by bits 0-3 in the first command byte GGGG. The configuration variables contain the data for the CV with the smaller number (least significant bits) in the second command byte, the data for the CV with the larger number (most significant bits) in the third command byte. For example: CV17 is represented by the second command byte followed by CV18 in the third command byte

The remaining values of GGGG are reserved and will be selected by the NMRA as need is determined.[7] Paired CVs must be written at the same time to avoid problems. If the decoder successfully receives both packets, it shall respond with an operations mode acknowledgment.

2.3.7.3 Configuration Variable Access Instruction - Long Form

The long form allows the direct manipulation of all CVs[8]. This instruction is valid both when the Digital Decoder has its long address active and short address active. Digital Decoders shall not act on this instruction if sent to its consist address. The format of the instructions using Direct CV addressing is:

{instruction bytes}= 1110GGVV 0 VVVVVVVV 0 DDDDDDDD

The actual Configuration Variable desired is selected via the 10-bit address with the 2-bit address (VV) in the first data byte being the most significant bits of the address. The Configuration variable being addressed is the provided 10-bit address plus 1. For example, to address CV1 the 10 bit address is “00 00000000”.

The defined values for Instruction type (~~CC~~GG) are:

GG=00 Reserved for future use
GG=01 Verify byte
GG=11 Write byte

[7] The NMRA shall not issue a NMRA Conformance Warrant for any product that uses an instruction or sub-instruction that has been reserved by the NMRA.

[8] Because of the length of this instruction, care must be taken to ensure that the maximum time between packets is not exceeded.

GG=10 Bit manipulation

Type = "01" VERIFY BYTE
The contents of the Configuration Variable as indicated by the 10-bit address are compared with the data byte (DDDDDDDD). If the decoder successfully receives this packet and the values are identical, the Digital Decoder shall respond with the contents of the CV as the Decoder Response Transmission, if enabled.

Type = "11" WRITE BYTE
The contents of the Configuration Variable as indicated by the 10-bit address are replaced by the data byte (DDDDDDDD). Two identical packets are needed before the decoder shall modify a configuration variable[9]. These two packets need not be back to back on the track. However any other packet to the same decoder will invalidate the write operation. (This includes broadcast packets.) If the decoder successfully receives this second identical packet, it shall respond with a configuration variable access acknowledgment.

Type = "10" BIT MANIPULATION
The bit manipulation instructions use a special format for the data byte (DDDDDDDD): 111FDBBB, where BBB represents the bit position within the CV, D contains the value of the bit to be verified or written, and F describes whether the operation is a verify bit or a write bit operation.

F = "1" WRITE BIT
F = "0" VERIFY BIT

The VERIFY BIT and WRITE BIT instructions operate in a manner similar to the VERIFY BYTE and WRITE BYTE instructions (but operates on a single bit). Using the same criteria as the VERIFY BYTE instruction, an operations mode acknowledgment will be generated in response to a VERIFY BIT instruction if appropriate. Using the same criteria as the WRITE BYTE instruction, a configuration variable access acknowledgment will be generated in response to the second identical WRITE BIT instruction if appropriate.

2.3.7.4 Configuration Variable Access Instruction XPOM

The Extended Program on the Main, or XPOM, form allows the direct access of CVs by their full 24-bit indexed address[10]. Up to four bytes can be written or read at one time. This instruction is valid both when the Digital Decoder has its long address active and short address active. Digital Decoders shall not act on this instruction if sent to its consist address. The format of the instructions using Direct CV addressing is:
{instruction bytes}= 1110GGSS 0 VVVVVVVV 0 VVVVVVVV 0 VVVVVVVV 0 {DDDDDDDD 0 {DDDDDDDD 0 {DDDDDDDD 0 {DDDDDDDD}}}}[11]

---

[9] See S-9.2.2 for more information on paired CVs.

[10] Because of the length of this instruction, care must be taken to ensure that the maximum time between packets is not exceeded.

[11] The XPOM command intentionally violates the maximum packet length restriction of 6 bytes inclusive of the X-OR byte. The decoder is required to receive two identical packets before it commits any CV write as an added integrity check for this otherwise excessively long packet.

SS is a sequence number that is required for bi-directional feedback when accessing many CV's in blocks. Without bi-directional feedback, these bits are meaningless. Therefore, the use of the sequence number is defined in S-9.3.2. Systems that do not support bi-directional feedback shall set these bits to 00.

VVVVVVVV 0 VVVVVVVV 0 VVVVVVVV is the 24-bit index CV address. The first byte corresponds to CV31, the second byte corresponds to CV32, and the third byte corresponds to the second byte of the "Configuration Variable Access Instruction – Long Form".

The defined values for Instruction type (GG) are:
GG=00 Reserved for future use
GG=01 Read bytes
GG=11 Write bytes
GG=10 Write bits

Type = "01" READ BYTES
There are no data bytes transferred. The contents of the selected CV and three following CVs is transmitted via bi-directional feedback.

Type = "11" WRITE BYTE
Up to four sequential CVs may be written by this instruction. The 24-bit address points to the first CV in the up to four-byte sequence. The values of the four CVs are always reported back via bi-directional feedback. Two identical packets are needed before the decoder shall modify a configuration variable. These two packets need not be back to back on the track. However any other packet to the same decoder will invalidate the write operation. (This includes broadcast packets.) If the decoder successfully receives this second identical packet, it shall respond with values of the four CV's via bi-directional feedback.

Type = "10" WRITE BITS
The fifth byte has the format 1111-DBBB as in the "Configuration Variable Access Instruction – Long form", whereby BBB represents the bit position within the CV and D contains the value of the bit to be written. Two identical packets are needed before the decoder shall modify a configuration variable. These two packets need not be back to back on the track. However any other packet to the same decoder will invalidate the write operation. (This includes broadcast packets.) If the decoder successfully receives this second identical packet, the values of the four CVs are reported back via bi-directional feedback.

2.4 Accessory Digital Decoder Packet Formats

Accessory Digital Decoders are intended for control of a number of simple functions such as switch machine control or turning on and off lights. It is permissible to develop *Digital Decoders* that respond to multiple addresses so that more devices can be controlled by a single *Digital Decoder*.

2.4.1 Basic Accessory Decoder Packet Format

The format for packets intended for Accessory Digital Decoders is:

{preamble} 0 10AAAAAA 0 1AAADAAR[12]~~CDDD~~ 0 EEEEEEEE 1

Accessory Digital Decoders can be designed to control momentary or constant-on devices, the duration of time each output is active being controlled by configuration variables CVs #515 through 518. ~~-~~Bit 3 of the second byte "D~~C~~" is used to activate or deactivate the addressed device. (Note if the duration the device is intended to be on is less than or equal the set duration, no deactivation is necessary.) Since most devices are paired, the convention is that bit 0 of the second byte "R" is used to distinguish between which of a pair of outputs the accessory decoder is activating or deactivating. By convention, R = 0 means diverging, direction of travel to the left, or signal to stop and R = 1 means normal, direction of travel to the right, or signal to proceed. ~~Bits 1 and 2 of byte two are used to indicate which of 4 pairs of outputs the packet is controlling.~~ The most significant bits of the ~~9~~11-bit address are bits 4~~-~~ to 6 of the second ~~data~~ byte. By convention these bits (bits 4~~-~~ to 6 of the second ~~data~~ byte) are in ones' complement[13]. This is followed by bits 0 to 5 of the first byte. The least significant bits of the 11-bit address are bits 1 to 2 of the second byte.[14]

{preamble} 0 $10A_7A_6A_5A_4A_3A_2$ 0 $1\bar{A}_{10}\bar{A}_9\bar{A}_8DA_1A_0R$ 0 EEEEEEEE 1

By convention, the first address, known to the user as address "1" starts at:

{preamble} 0 10000001 0 1111D00R 0 EEEEEEEE 1

For address bits $A_7..A_2$, there exists two addressing conventions for when the value "rolls over". The two conventions are called "Linear" and "Non-Linear":

{preamble} 0 10000000 0 $1\bar{A}_{10}\bar{A}_9\bar{A}_8DA_1A_0R$ 0 EEEEEEEE 1

For existing designs, both conventions are considered compliant with the standard. However, the "Linear" convention is required for new designs to be considered compliant:

User Address	**Linear**			**Non-Linear**		
	Byte 1	**Byte 2**	**$A_{10}..A_0$**	**Byte 1**	**Byte 2**	**$A_{10}..A_0$**
1	10000001	1111D00R	4	10000001	1111D00R	4
…	…	…	…	…	…	…
252	10111111	1111D11R	255	10111111	1111D11R	255
253	10000000	1110D00R	256	10000000	1111D00R	0
254	10000000	1110D01R	257	10000000	1111D01R	1
255	10000000	1110D10R	258	10000000	1111D10R	2
256	10000000	1110D11R	259	10000000	1111D11R	3

[12] Prior versions of this Standard use the notation 1AAACDDD. This has been updated to harmonize with the notation used in RCN-213.

[13] The ones' complement form is where every bit is inverted. ~~E.G.~~e.g. the ones' complement of 000 is 111, ones' complement of 001 is 110, of 010 is 101 etc.

[14] 'Ā' identifies an inverted address bit.

User Address	Linear			Non-Linear		
	Byte 1	Byte 2	$A_{10}..A_0$	Byte 1	Byte 2	$A_{10}..A_0$
257	10000001	1110D00R	260	10000001	1110D00R	260
…	…	…	…	…	…	…
508	10111111	1110D11R	511	10111111	1110D11R	511
509	10000000	1101D00R	512	10000000	1110D00R	256
510	10000000	1101D01R	513	10000000	1110D01R	257
511	10000000	1101D10R	514	10000000	1110D10R	258
512	10000000	1101D11R	515	10000000	1110D11R	259
513	10000001	1101D00R	516	10000001	1110D00R	516
…	…	…	…	…	…	…
2041	10111111	1000D00R	2044	10111111	1000D00R	2044
2042	10111111	1000D01R	2045	10111111	1000D01R	2045
2043	10111111	1000D10R	2046	10111111	1000D10R	2046
2044	10111111	1000011R	2047	10111111	1000011R	2047
2045	10000000	1111D00R	0	10000000	1000D00R	1789
2046	10000000	1111D01R	1	10000000	1000D01R	1790
2047	10000000	1111D10R	2	10000000	1000D10R	1791
2048	10000000	1111D11R	3	10000000	1000D11R	1792

If operations-mode acknowledgement is enabled, receipt of a basic accessory decoder packet must be acknowledged with an operations-mode acknowledgement. Refer to S-9.3.2 Bi-Directional Communications.

2.4.2 Extended Accessory Decoder Control Packet Format

The Extended Accessory Decoder Control Packet is included for the purpose of transmitting aspect control to signal decoders or data ~~bytes~~ to more complex accessory decoders. Each signal head can display one aspect at a time.

{preamble} 0 10AAAAAA 0 0AAA0AA1 0 ~~000~~XXXXXXXX 0 EEEEEEEE 1

The most significant bits of the 11-bit address are bits 4 to 6 of the second byte. By convention these bits (bits 4 to 6 of the second byte) are in ones' complement[15]. This is followed by bits 0 to 5 of the first byte. The least significant bits of the 11-bit address are bits 1 to 2 of the second byte.[16]

[15] The ones' complement form is where every bit is inverted. E.G. the ones' complement of 000 is 111, ones' complement of 001 is 110, of 010 is 101 etc.

[16] 'Ā' identifies an inverted address bit.

{preamble} 0 10$A_7A_6A_5A_4A_3A_2$ 0 0$\bar{A}_{10}\bar{A}_9\bar{A}_8$0$A_1A_0$1 0 XXXXXXXX 0 EEEEEEEE 1

By convention, the first address, known to the user as address “1” starts at:

{preamble} 0 10000001 0 01110001 0 XXXXXXXX 0 EEEEEEEE 1

Linearly increasing addressing is used for successive addresses.

The 256 possible states are transmitted via bits 0 to 7 in the third byte (XXXXXXXX).

When used for a signaling systems, XXXXXXXX is for a single head. A value of 00000000 for XXXXXXXX indicates the absolute stop aspect. All other aspects represented by the values for XXXXXXXX are determined by the signaling system used and the prototype being modeled.

When used in a time-based application, the third byte defines the output to be controlled and the time for which the output should be active. The format of the third byte is RZZZZZZZ. ZZZZZZZ defines the output on time with a resolution of 100 milliseconds. The value of 0000000 means that the output is switched off. The value of 1111111 means that the output is switched on continuously, with no timeout, until the next command to the address. Bit 7 “R” of the third byte is used to select the output within a a pair of outputs linked to the same address.

If operations-mode acknowledgement is enabled, receipt of an extended accessory decoder packet must be acknowledged with an operations-mode acknowledgement.

~~2.4.3 Broadcast Command for Basic Accessory Decoders~~

~~The format for the broadcast instruction is:~~

~~{preamble} 0 10111111 0 1000CDDD 0 EEEEEEEE 1~~

~~This packet shall be executed by all accessory decoders. CDDD is defined as specified in the paragraph on Basic Accessory Decoder Packet Format.~~

~~2.4.4 Broadcast Command for Extended Accessory Decoders~~

~~The format for the broadcast instruction is:~~

~~{preamble} 0 10111111 0 00000111 0 000XXXXX 0 EEEEEEEE 1~~

~~All extended accessory decoders must execute this packet. XXXXX is defined as specified in the paragraph on Extended Accessory Decoder Packet Format.~~

~~2.4.5~~2.4.3 Accessory Decoder Configuration Variable Access Instruction

Accessory decoders can have their Configuration variables changed in the same method as locomotive decoders using the **Configuration Variable Access Instruction -** Long Form instruction defined above. ~~For the purpose of this instruction, the accessory decoders’ address is expanded to two bytes in the following method.~~ If operations-mode acknowledgement is enabled, the receipt of an Accessory Decoder Configuration Variable Access instruction must be acknowledged in the same manner as the Configuration Variable Access Instruction – Long Form.

The decoder can only be addressed for configuration on the addresses for which it is programmed to react to commands. It is permissible to configure all outputs via one address.

~~2.4.6~~2.4.3.1 Basic Accessory Decoder ~~Packet address for o~~Operations ~~m~~Mode P~~p~~rogramming

{preamble} $10A_7A_6A_5A_4A_3A_2$ 0 $1\bar{A}_{10}\bar{A}_9\bar{A}_8 1A_1A_0 0$ 0 … ~~10AAAAAA 0 1AAACDDD~~

New programming devices must by default set bit 3 of the second byte to 1. Prior versions of this standard allowed bit 3 of the second byte to be 0 in order to support decoders with four consecutive addresses. Programming devices may provide a user option for backwards compatibility with older decoders that support this prior convention.

~~Where DDD is used to indicate the output whose CVs are being modified and C=1.~~
~~If CDDD= 0000 then the CVs refer to the entire decoder. The resulting packet would be~~

{preamble} 10AAAAAA 0 1AAA1AA0~~CDDD~~ 0 **(**1110CCVV 0 VVVVVVVV 0 DDDDDDDD**)** 0 EEEEEEEE 1[17]

~~Accessory Decoder Address (Configuration Variable Access Instruction) Error Byte~~

~~2.4.7~~2.4.3.2 Extended Accessory Decoder Control ~~Packet address for o~~Operations ~~m~~Mode ~~p~~Programming

{preamble} $10A_7A_6A_5A_4A_3A_2$ 0 $0\bar{A}_{10}\bar{A}_9\bar{A}_8 0A_1A_0 1$ 0 … ~~10AAAAAA 0 0AAA0AA1~~

Please note that the use of 0 in bit 3 of the second byte ~~2~~ is to ensure that this packet cannot be confused with the legacy accessory-programming packets. The resulting packet ~~would be~~is:

{preamble} 10AAAAAA 0 0AAA0AA1 0 **(**1110CCVV 0 VVVVVVVV 0 DDDDDDDD**)** 0 EEEEEEEE 1[18]
~~Signal Decoder Address (Configuration Variable Access Instruction) Error Byte~~

2.4.4 NOP Command for Basic and Extended Accessory Decoders

The format for the No Operation, or NOP, command is:
{preamble} 0 10AAAAAA 0 0AAA1AAT 0 EEEEEEEE 1

T = 0: Basic accessory decoder
T = 1: Extended accessory decoder

The NOP command enables Bi-Directional capable accessory decoders to send a service request (SRQ) to the command station without changing the current status This command is recognized as invalid by all non-Bi-Directional capable basic and extended accessory decoders and is therefore ignored.

---

[17] A distinction between operating commands and configuration variable access commands can only be made via the length of the packet.

[18] A distinction between operating commands and configuration variable access commands can only be made via the length of the packet.

For more information, see S-9.3.2.

2.5 Operations Mode Acknowledgment

725 The operations-mode acknowledgment mechanism as defined in S-9.3.2 are the only valid acknowledgement in operations mode. Whenever an acknowledgment is requested, the decoder shall respond using this mechanism described in S-9.3.2.

3 Document History

Date	Description
July 1995	First Release
March 1997	Revisions approved by NMRA BOD
July 2003	Revisions approved by NMRA BOD
January 2006	Revisions approved by NMRA BOD
July 2009	Edited to agree with S-9.2.2
July 2012	Became a Standard
8-Feb-2022	Migrated to new template. Error corrections. Added time clock Standards. Added instruction types G and T for clarity. Added Function Groups F29-F68. Added information to harmonize with S-9.2.1.1, S-9.3.2, RCN-214 & RCN-212. Removed section of set decoder flags. This had many errors and conflicts. It is not used by any manufacturer.
15-May-2022	Revisions approved by NMRA BOD

4 Appendix A.

This Appendix contains additional useful information and/or legacy instructions. A DCC product need not implement any items described in this appendix.

4.1 Accessory Decoder Configuration Variable Access Instruction[19]

The following command is included for backward compatibility for some older accessory decoders. Its use is discouraged in new decoder designs.

The format for Accessory Decoder Configuration Variable Access Instructions is:

{preamble} 0 10AAAAAA 0 0AAA11VV 0 VVVVVVVV 0 DDDDDDDD 0 EEEEEEEE 1

Where:

A = Decoder address bits
V = Desired CV address - (CV 513 = 10 00000000)
D = Data for CV

The bit patterns described by VV VVVVVVVV in the second and third bytes and DDDDDDDD in the fourth byte are also identical to the corresponding bits in the Configuration Variable Access Instruction - Long Form (see 2.3.7.3).

The purpose of this instruction is to provide a means of programming all parameters of an accessory decoder after it is installed on the layout. It is recommended that Command Stations exercise caution if changes to the address (CV 513 and CV 521) are allowed.

[19] For backward compatibility, decoders should check the length of instruction packets when bit 7 of byte 2 is zero.

5 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.