CBUS 5.0 Specification Rev. A1

Original © Mike Bolton & Gil Fuchs 2007 - 2009
Updates © Mike Bolton, Andrew Crosland, Roger Healey & Pete Brownlow 2009-2015
Updates © Mike Bolton, Pete Brownlow. & Andrew Crosland 2020

Note. To preserve compatibility, no changes to the protocol or additional OpCodes should be made without the approval of the primary author. Such changes, if agreed, will require the issue of an updated specification document. (mikebolton1844@gmail.com)

Update history:

Draft 7c	by Andrew Crosland, Mike Bolton and Roger Healey 22/11/09
Draft 7d	Updates to rev 7c by Mike Bolton 12/01/11
Draft 7e	Update (OPC 0x59 only) by Mike Bolton 05/04/11
Draft 7f	Updates Added OPCs (Pete Brownlow) and one correction 13/04/11
Draft 7g	Added OPCs for short data events and requests. MB. 02/07/11 Added OPCs for RQMN,NAME,DFNON,DFNOF,QNN,renamed FliM setup opcodes to match implementation, reinstated DCC session keep alive. PNB 04/07/11 Updated definition of BOOTM (0x5C) Added Appendix 1, Node Parameter Definitions RKH 04/07/11
Draft 7h	Major changes including added OPCs, changed mnemonics and some OPCs moved. (new values). MPB 03/08/11
Draft 8a	Added opcodes PNN, GLOC, FCLK, some new error messages and description updates by Pete Brownlow 18/2/12
Draft 8b	Minor changes to format. Mike Bolton, 13/07/12
Version 8c	Added OpCodes ENUM and CANID 02/08/12. Dropped use of 'Draft'.
Version 8d	Added OpCode ALOC (0x43). Reinstated OpCode QCON (0x41) Minor typing corrections. Mike Bolton (30/06/15)
Version 8e	Added OpCodes CABDAT (0x C2), DDWS (0xFC), NNRSM (0x4F), NNRST (0x5E) Added error number for DCC Command Station when programming.(9). Added wording for QCON (0x41). Change of email address for Mike Bolton. (14/10/20)
Version 8f	Removed DCC error code 9. Changes to wording in GLOC, RLOC, QLOC & STAT.

Now renamed CBUS 5.0. The revision has reverted to the default which is A1. (Mike Bolton 19th June 2021

License terms:

This document is distributed under the following

Attribution-NoDerivatives 4.0 International (CC BY-ND 4.0)

The full terms of this license must be adhered to. A full license is available via
https://creativecommons.org/licenses/by-nd/4.0/

Communication Protocol

General CAN message format:

[<MjPri><MinPri><ID>] <Opcode><Dat0> ..<DatN>

where:

- **<MjPri>** bits 9 – 10 of the CAN header. Dynamic Priority, elevated by the node to gain access based on a transmit fail count. Values:
  0 – Emergency priority
  1 – High priority
  2 – Normal priority

- <**MinPri>** bits 7 - 8 of the CAN header. Static priority based on message and node type. Values:
  0 – High access
  1 – Above Normal access
  2 – Normal access
  3 – Low access
- **<CANID>** bits 0 – 6 of the CAN header, is a CAN segment-unique ID, assigned via enumeration.
- **<Opcode>** the first data byte is the opcode which includes the length of the message in the upper 3 bits.

In some associated documents, the Opcode is also referred to as the 'command' byte. The abbreviation OPC may also be used. In this document the Opcodes are in hexadecimal.

Packet Definitions (by OPC field) The first column is a decimal OPC reference number. The second column is the actual OPC in hexadecimal.

00 - 1F – 0 Data bytes packets
[<MjPri><MinPri><CAN ID>]<Opcode>

0. 00 General Acknowledgement (*ACK*)
   Format:
   [<MjPri><MinPri=2><CANID>]<00>
   Positive response to query/ request performed or report of availability on-line.

1. 01 General No Ack (*NAK*)
   Format:
   [<MjPri><MinPri=2><CANID>]<01>
   Negative response to query/ request denied.

2. 02 Bus Halt (*HLT*)
   Format:
   [<MjPri><MinPri=0><CANID>]<02>
   Commonly broadcasted to all nodes to indicate CBUS is not available and no further packets should be sent until a BON or ARST is received.

3. 03 Bus ON (*BON*)

Format:
[<MjPri><MinPri=1><CANID>]<03>
Commonly broadcasted to all nodes to indicate CBUS is available following a HLT.

4. 04 Track OFF (*TOF*)
   Format:
   [<MjPri><MinPri=1><CANID>]<04>
   Commonly broadcasted to all nodes by a command station to indicate track power is off and no further command packets should be sent, except inquiries.

5. 05 Track ON (*TON*)
   Format:
   [<MjPri><MinPri=1><CANID>]<05>
   Commonly broadcasted to all nodes by a command station to indicate track power is on.

6. 06 Emergency Stop (*ESTOP*)
   Format:
   [<MjPri><MinPri=1><CANID>]<06>
   Commonly broadcast to all nodes by a command station to indicate all engines have been emergency stopped.

7. 07 System Reset (*ARST*)
   Format:
   [<MjPri><MinPri=0><CANID>]<07>
   Commonly broadcasted to all nodes to indicate a full system reset.

8. 08 Request Track OFF (*RTOF*)
   Format:
   [<MjPri><MinPri=1><CANID>]<08>
   Sent to request change of track power state to “off”.

9. 09 Request Track ON (*RTON*)
   Format:
   [<MjPri><MinPri=1><CANID>]<09>
   Sent to request change of track power state to “on”.

10. 0A Request Emergency Stop ALL (*RESTP*)
    Format:
    [<MjPri><MinPri=0><CANID>]<0A>
    Sent to request an emergency stop to all trains . Does not affect accessory control. See section 9.1.7.

11. 0B Reserved

12. 0C Request Command Station Status (*RSTAT*)
    Format:
    [<MjPri><MinPri=2><CANID>]<0C>
    Sent to query the status of the command station. See description of (STAT) for the response from the command station.

13. 0D Query node number (QNN)
    Format:
    [<MjPri><MinPri=3><CANID>]<0D>

Sent by a node to elicit a PNN reply from each node on the bus that has a node number. See OpCode 0xB6

14. 0E Reserved

15. 0F Reserved

16. 10 Request node parameters(RQNP)
Format:
[<MjPri><MinPri=3><CANID>]<10>
Sent to a node while in 'setup'mode to read its parameter set. Used when initially configuring a node.

17. 11 Request module name (RQMN)
Format:
[<MjPri><MinPri=2><CANID>]<11>
Sent by a node to request the name of the type of module that is in setup mode. The module in setup mode will reply with opcode NAME. See OpCode 0xE2

18. 12 Reserved

19. 13 Reserved

20. 14 Reserved

21. 15 Reserved

22. 16 Reserved

23. 17 Reserved

24 18 Reserved

25 19 Reserved

26 1A Reserved

27 1B Reserved

28 1C Reserved

29 1D Reserved

30 1E Reserved

31 1F Reserved

20 – 3F 1 Data byte packets
[<MjPri><MinPri><CAN ID>]<Opc><Dat1>

32 20 Reserved

33. 21 Release Engine (*KLOC*)
Format:
[<MjPri><MinPri=2><CANID>]<21><Session>
<**Dat1**> is the engine session number as HEX byte.
Sent by a CAB to the Command Station. The engine with that Session number is removed from the active engine list.

34. 22 Query engine (*QLOC*)
Format:
[<MjPri><MinPri=2><CANID>]<22><Session>
<**Dat1**> is the engine session number as HEX byte.
The command station responds with PLOC if the session is assigned.
Otherwise responds with OPC_ERR Message.

35. 23 Session keep alive (DKEEP)
Format:
[<MjPri><MinPri=2><CANID>]<23><Session>
<**Dat1**> is the engine session number as HEX byte.

The cab sends a keep alive at regular intervals for the active session. The interval between keep alive messages must be less than the session timeout implemented by the command station.

36. 24 Reserved

37. 25 Reserved

38. 26 Reserved

39. 27 Reserved

40. 28 Reserved

41. 29 Reserved

42. 2A Reserved

43. 2B Reserved

44. 2C Reserved

45. 2D Reserved

46. 2E Reserved

47. 2F Reserved

48. 30 Debug with one data byte (*DBG1*)
    Format:
    [<MjPri><MinPri=2><CANID>]<30><Status>
    <**Dat1**> is a freeform status byte for debugging during CBUS module development. Not used during normal operation

49. 31 Reserved

50. 32 Reserved

51. 33 Reserved

52. 34 Reserved

53. 35 Reserved

54. 36 Reserved

55. 37 Reserved

56. 38 Reserved

57. 39 Reserved

58. 3A Reserved

59. 3B Reserved

60. 3C Reserved

61. 3D Reserved

62. 3E Reserved

63. 3F Extended op-code with no additional bytes (EXTC)
    Format:
    [<MjPri><MinPri=3><CANID>]<3F><Ext_OPC>
    Used if the basic set of 32 OPCs is not enough. Allows an additional 256 OPCs

40 – 5F 2 data byte packets
[<MjPri><MinPri><CAN ID>]<Opc><Dat1><Dat2>

64. 40 Request engine session (*RLOC*)
Format:
[<MjPri><MinPri=2><CANID>]<40><Dat1><Dat2 >
<**Dat1**> and <**Dat2**> are [AddrH] and [AddrL] of the decoder, respectively.
7 bit addresses have (AddrH=0). 4 bit addresses have bits 6,7 of AddrH set to 1.
The command station responds with (PLOC) if engine is free and is being assigned. Otherwise responds with (OPC_ERR) and the appropriate error number. This command is typically sent by a cab to the command station following a change of the controlled decoder address. RLOC is exactly equivalent to GLOC with all flag bits set to zero, but command stations must continue to support RLOC for backwards compatibility.

65. 41 Query Consist (*QCON*)
Format:
[<MjPri><MinPri=2><CANID>]<41><ConID><Index>
<**Dat1**> is consist address.
<**Dat2**> is engine index in the consist.
Allows enumeration of a consist. Command station responds with PLOC if an engine exists at the specified index, otherwise responds with
ERR: code no. 5. Loco not found.
Note that a command station need not support this opcode if it uses advanced consisting and has no way of reading back the CV currently containing the consist address in a loco.

66. 42 Set Node Number (*SNN*)
Format:
[<MjPri><MinPri=3><CANID>]<42><NNHigh><NNLow>
<**Dat1**> is high byte of the node number.
<**Dat2**> is low byte of the node number.
Sent by a configuration tool to assign a node number to a requesting node in response to a *RQNN* message.
The target node must be in 'setup' mode.

67. 43 Allocate loco to activity. (ALOC)
Format:
[<MjPri><MinPri=2><CANID>]<43><Session ID><Allocation code >
<**Dat1**> is Session ID.
<**Dat2**> is application specific allocation code. (one byte)

68. 44 Set CAB session mode (STMOD)
Format:
[<MjPri><MinPri=2><CANID>]<44><Session><MMMMMMMM>
<**Dat1**> Session number
<**Dat2>** contains mode bits:
0 – 1: speed mode
00 – 128 speed steps
01 – 14 speed steps
10 – 28 speed steps with interleave steps
11 – 28 speed steps
2: service mode
3: sound control mode

69. 45 Consist Engine (*PCON*)

Format:
[<MjPri><MinPri=2><CANID>]<45><Session><Consist#>
<**Dat1**> Session number
<**Dat2**> is consist address (8 bits).
Adds a decoder to a consist.
Dat2 has bit 7 set if consist direction is reversed.

70. 46 Remove Engine from consist (*KCON*)

Format:
[<MjPri><MinPri=2><CANID>]<46><Session><Consist#>
<**Dat1**> loco session number
<**Dat2**> is consist address.
Removes a loco from a consist.

71. 47 Set Engine Speed/Dir (*DSPD*)

Format:
[<MjPri><MinPri=2><CANID>]<47><Session><Speed/Dir>
<**Dat1**> session number
<Dat2> is speed/dir value, where the most significant bit is direction and the 7ls bits are the unsigned speed value. Sent by a CAB or equivalent to request an engine speed/dir change.

72. 48 Set Engine Flags (*DFLG*)

Format:
[<MjPri><MinPri=2><CANID>]<48><Session><DDDDDDDD>
<**Dat1**> Session number
<**Dat2>** is the flags:
Bits 0-1: Speed Mode
00 – 128 speed steps
01 – 14 speed steps
10 – 28 speed steps with interleave steps
11 – 28 speed steps
Bit 2: Lights On/OFF
Bit 3: Engine relative direction
Bits 4-5: Engine state (active =0 , consisted =1, consist master=2, inactive=3)
Bits 6-7: Reserved.
Sent by a cab to notify the command station of a change in engine flags.

73. 49 Set Engine function on (*DFNON*)

Format:
[<MjPri><MinPri=2><CANID>]<49><Session><Fnum>
<Dat1> is the engine session number.
<**Dat2**> is the function number – 0 to 27.

Sent by a cab to turn on a specific loco function. This provides an alternative method to DFUN for controlling loco functions. A command station must implement both methods.

74. 4A Set Engine function off (*DFNOF*)

Format:
[<MjPri><MinPri=2><CANID>]<4A><Session><Fnum>
<Dat1> is the engine session number.
<**Dat2**> is the function number – 0 to 27.

Sent by a cab to turn off a specific loco function. This provides an alternative method to DFUN for controlling loco functions. A command station must implement both methods.

75. 4B Reserved

76. 4C Service mode status. (SSTAT)
Format:
[<MjPri><MinPri=3><CANID>]<4C><Session><Status>
Status returned by command station/programmer at end of programming operation that does not return data.

77. 4D Reserved

78. 4E Reserved

79. 4F Reset to manufacturers defaults (NNRSM)
Format:
[<MjPri><MinPri=3><CANID>]<4F><NN hi><NN lo>

Causes the module to reset settings to manufacturers defaults. The module should retain any node number and remain in FLiM mode. What the manufacturers defaults are will be defined for each module, but should be equivalent to putting a new module into FLiM, with no events taught, only default events defined (if any) and all NVs returned to their default values

80. 50 Request node number (RQNN)
Format:
[<MjPri><MinPri=3><CANID>]<50><NN hi><NN lo>

Sent by a node that is in setup/configuration mode and requests assignment of a node number (NN). The node allocating node numbers responds with (SNN) which contains the newly assigned node number. <NN hi> and <NN lo> are the existing node number, if the node has one. If it does not yet have a node number, these bytes should be set to zero.

81. 51 Node number release (NNREL)
Format:
[<MjPri><MinPri=3><CANID>]<51><NN hi><NN lo>
Sent by node when taken out of service. e.g. when reverting to SLiM mode.

82. 52 Node number acknowledge. (NNACK)
Format:
[<MjPri><MinPri=3><CANID>]<52><NN hi><NN lo>
Sent by a node to verify its presence and confirm its node id. This message is sent to acknowledge an SNN.

83. 53 Set node into learn mode (NNLRN)
Format:
[<MjPri><MinPri=3><CANID>]<53><NN hi><NN lo>
Sent by a configuration tool to put a specific node into learn mode.

84. 54 Release node from learn mode (NNULN)
Format:
[<MjPri><MinPri=3><CANID>]<54><NN hi><NN lo>
Sent by a configuration tool to take node out of learn mode and revert to normal operation.

85. 55 Clear all events from a node (NNCLR)
Format:
[<MjPri><MinPri=3><CANID>]<55><NN hi><NN lo>
Sent by a configuration tool to clear all events from a specific node. Must be in learn mode first to safeguard against accidental erasure of all events.

86. 56 Read number of events available in a node (NNEVN)
Format: [<MjPri><MinPri=3><CANID>]<56><NN hi><NN lo>
Sent by a configuration tool to read the number of available event slots in a node. Response is EVLNF (0x70)

87. 57 Read back all stored events in a node (NERD)
Format:
[<MjPri><MinPri=3><CANID>]<57><NN hi><NN lo>
Sent by a configuration tool to read all the stored events in a node. Response is 0xF2.

88. 58 Request to read number of stored events (RQEVN)
Format:
[<MjPri><MinPri=3><CANID>]<58><NN hi><NN lo>
Sent by a configuration tool to read the number of stored events in a node. Response is 0x74(NUMEV).

89. 59 Write acknowledge (WRACK)
Format:
[<MjPri><MinPri=3><CANID>]<59><NN hi><NN lo>
Sent by a node to indicate the completion of a write to memory operation. All nodes must issue WRACK when a write operation to node variables, events or event variables has completed. This allows for teaching nodes where the processing time may be slow.

90. 5A Request node data event (RQDAT)
Format:
[<MjPri><MinPri=3><CANID>]<5A><NN hi><NN lo>
Sent by one node to read the data event from another node.(eg: RFID data). Response is 0xF7 (ARDAT).

91. 5B Request device data – short mode (RQDDS)
Format:
[<MjPri><MinPri=3><CANID>]<5B><DN hi><DN lo>
To request a ‘data set’ from a device using the short event method.
where DN is the device number. Response is 0xFB (DDRS)

92. 5C Put node into bootload mode (BOOTM)
Format:
[<MjPri><MinPri=3><CANID>]<5C><NN hi><NN lo>

For SliM nodes with no NN then the NN of the command must be zero. For SLiM nodes with an NN, and all FLiM nodes, the command must contain the NN of the target node. Sent by a configuration tool to prepare for loading a new program.

93. 5D Force a self enumeration cycle for use with CAN (ENUM)
Format:
[<MjPri><MinPri=3><CANID>]<5D><NN hi><NN lo>

For nodes in FLiM using CAN as transport.. This OPC will force a self-enumeration cycle for the specified node. A new CAN_ID will be allocated if needed. Following the ENUM sequence, the node should issue a NNACK to confirm completion and verify the new CAN_ID. If no CAN_ID values are available, an error message 7 will be issued instead.

94. 5E Restart node (NNRST)

Format:
<MjPri><MinPri=3><CANID>]<5E><NN hi><NN lo>

Causes module to carry out a software reset to restart the firmware. No settings are affected.

95. 5F Extended op-code with 1 additional byte (EXTC1)

Format:
[<MjPri><MinPri=3><CANID>]<5F><Ext_OPC><byte>

Used if the basic set of 32 OPCs is not enough. Allows an additional 256 OPCs

60-7F 3 data byte packets
[<MjPri><MinPri><CAN ID>]<OPC><Dat1><Dat2><Dat3>

96. 60 Set Engine functions (*DFUN*)
Format:
[<MjPri><MinPri=2><CANID>]<60><Session><Fn1><Fn2>
<Dat1> is the engine session number.
<**Dat2**> is the function range.
1 is F0(FL) to F4
2 is F5 to F8
3 is F9 to F12
4 is F13 to F20
5 is F21 to F28
<**Dat3>** is the NMRA DCC format function byte for that range in corresponding bits. Sent by a CAB or equivalent to request an engine Fn state change.

97. 61 Get engine session (*GLOC*)
Format:
[<MjPri><MinPri=2><CANID>]<61><Dat1><Dat2><Flags>
<**Dat1**> and <**Dat2**> are [AddrH] and [AddrL] of the decoder, respectively.
7 bit addresses have (AddrH=0).
14 bit addresses have bits 6,7 of AddrH set to 1.
<Flags> contains flag bits as follows:
Bit 0: Set for "Steal" mode
Bit 1: Set for "Share" mode

Both bits set to 0 is exactly equivalent to an RLOC request
Both bits set to 1 is invalid, because the 2 modes are mutually exclusive
The command station responds with (PLOC) if the request is successful.
Otherwise responds with (ERR). See OpCode ERR. (0x63).
GLOC with all flag bits set to zero is exactly equivalent to RLOC, but command stations must continue to support RLOC for backwards compatibility.

98. 62 Reserved

99. 63 Command Station Error report (*ERR*)
Format:
[<MjPri><MinPri=2><CANID>]<63><Dat 1><Dat 2><Dat 3>

Sent in response to an error situation by a command station. See DCC Error codes for values.

100. 64 Reserved

101. 65 Reserved

102. 66 Reserved

103. 67 Reserved

104. 68 Reserved

105. 69 Reserved

106. 6A Reserved

107. 6B Reserved

108. 6C Reserved

109. 6D Reserved

110. 6E Reserved

111. 6F Error messages from nodes during configuration (CMDERR)
Format:
[<MjPri><MinPri=3><CANID>]<6F><NN hi><NN lo><Error number>
Sent by node if there is an error when a configuration command is sent.
See Accessory Module Error Codes section for values.

112. 70 Event space left in a node (EVNLF)
Format:
[<MjPri><MinPri=3><CANID>]<70><NN hi><NN lo><EVSPC>
EVSPC is a one byte value giving the number of available events left in that node.

113. 71 Request read of a node variable (NVRD)
Format:
[<MjPri><MinPri=3><CANID>]<71><NN hi><NN lo><NV#>
NV# is the index for the node variable value requested. Response is NVANS.

114. 72 Request read of stored events by event index (NENRD)
Format:
[<MjPri><MinPri=3><CANID>]<72><NN hi><NN lo><EN#>
EN# is the index for the stored event requested.
Response is 0xF2 (ENRSP)

115. 73 Request read of a node parameter by index (RQNPN)
Format:
[<MjPri><MinPri=3><CANID>]<73><NN hi><NN lo><Para#>
Para# is the index for the parameter requested. Index 0 returns the number of available parameters
Response is 0x9B (PARAN) .

116. 74 Number of events stored in node (NUMEV)
Format:
[<MjPri><MinPri=3><CANID>]<74><NN hi><NN lo><No.of events>
Response to request 0x58 (RQEVN). Maximum number if events is 255.

117. 75 Set a CAN_ID in existing FLiM node (CANID)
Format:
[<MjPri><MinPri=3><CANID>]<75><NN hi><NN lo><CAN_ID >
Used to force a specified CAN_ID into a node. Value range is from 1 to 0x63 (99 decimal)
This OPC must be used with care as duplicate CAN_IDs are not allowed.. Values outside the permitted range will produce an error 7 message and the CAN_ID will not change.

118. 76 Reserved

119. 77 Reserved

120. 78 Reserved

121. 79 Reserved

122. 7A Reserved

123. 7B Reserved

124. 7C Reserved

125. 7D Reserved

126. 7E Reserved

127. 7F Extended op-code with 2 additional bytes (EXTC2)
Format:
[<MjPri><MinPri=3><CANID>]<7F><Ext_OPC><byte1><byte2>
Used if the basic set of 32 OPCs is not enough. Allows an additional 256 OPCs

80-9F 4 data byte packets
[<MjPri><MinPri><CAN ID>]<Opc><Dat1><Dat2><Dat3><Dat4>

128. 80 Request 3-byte DCC Packet (*RDCC3*)
Format:
[<MjPri><MinPri=2><CANID>]<80><REP><Byte0>..<Byte2>
<**Dat1(REP)**> is number of repetitions in sending the packet.
<**Dat2>..<Dat4>** 3 bytes of the DCC packet.
Allows a CAB or equivalent to request a 3 byte DCC packet to be sent to the track. The packet is sent <REP> times and is not refreshed on a regular basis.
Note: a 3 byte DCC packet is the minimum allowed.

129. 81 Reserved

130. 82 Write CV (byte) in OPS mode *(WCVO)*
Format:
[<MjPri><MinPri=2><CANID>]<82><Session><High CV#><Low CV#><Val>
<**Dat1**> is the session number of the loco to be written to
<**Dat2>** is the MSB # of the CV to be written (supports CVs 1 - 65536)
<**Dat3>** is the LSB # of the CV to be written
<**Dat4>** is the byte value to be written
Sent to the command station to write a DCC CV byte in OPS mode to specific loco.(on the main)

131. 83 Write CV (bit) in OPS mode (WCVB)
Format:
[<MjPri><MinPri=2><CANID>]<83><Session><High CV#><Low CV#><Val>
<**Dat1**> is the session number of the loco to be written to
<**Dat2>** is the MSB # of the CV to be written (supports CVs 1 - 65536)
<**Dat3>** is the LSB # of the CV to be written
<**Dat4>** is the value to be written

The format for Dat4 is that specified in NMRA RP 9.2.1 for OTM bit manipulation in a DCC packet.
This is '111CDBBB' where C is here is always 1 as only 'writes' are possible OTM. (unless some loco ACK scheme like RailCom is used). D is the bit value, either 0 or 1 and BBB is the bit position in the CV byte. 000 to 111 for bits 0 to 7.
Sent to the command station to write a DCC CV in OPS mode to specific loco.(on the main)

132. 84 Read CV (*QCVS*)
Format:
[<MjPri><MinPri=2><CANID>]<84><Session><High CV#><Low CV#><Mode>
<**Dat1>** is the session number of the cab
<Dat2> is the MSB # of the CV read (supports CVs 1 - 65536)
<**Dat3>** is the LSB # of the CV read
<**Dat4>** is the programming mode to be used

This command is used exclusively with service mode.
Sent by the cab to the command station in order to read a CV value. The command station shall respond with a PCVS message containing the value read, or SSTAT if the CV cannot be read.

133. 85 Report CV (*PCVS*)

Format:
[<MjPri><MinPri=2><CANID>]<85><Session><High CV#><Low CV#><Val>
<**Dat1>** is the session number of the cab
<Dat2> is the MSB # of the CV read (supports CVs 1 - 65536)
<**Dat3>** is the LSB # of the CV read
<**Dat4>** is the read value
This command is used exclusively with service mode.
Sent by the command station to report a read CV.

134. 86 Reserved

135 87 Reserved

136. 88 Reserved

137. 89 Reserved

138 8A Reserved

139. 8B Reserved

140. 8C Reserved

141. 8D Reserved

142. 8E Reserved

143. 8F Reserved

144. 90 Accessory ON (ACON)

Format:
[<MjPri><MinPri=3><CANID>]<90><NN hi><NN lo><EN hi><EN lo>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the event number
<**Dat4>** is the low byte of the event number

Indicates an 'ON' event using the full event number of 4 bytes. (long event)

145. 91 Accessory OFF (ACOF)

Format:
[<MjPri><MinPri=3><CANID>]<91><NN hi><NN lo><EN hi><EN lo>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the event number
<**Dat4>** is the low byte of the event number

Indicates an 'OFF' event using the full event number of 4 bytes. (long event)

146. 92 Accessory Request Event (AREQ)

Format:
[<MjPri><MinPri=3><CANID>]<92><NN hi><NN lo><EN hi><EN lo>
<**Dat1>** is the high byte of the node number (MS WORD of the full event #)
<**Dat2>** is the low byte of the node number (MS WORD of the full event #)
<**Dat3>** is the high byte of the event number
<**Dat4>** is the low byte of the event number

Indicates a 'request' event using the full event number of 4 bytes. (long event)
A request event is used to elicit a status response from a producer when it is required to know the 'state' of the producer without producing an ON or OFF event.

147. 93 Accessory Response Event (ARON)

Format:
[<MjPri><MinPri=3><CANID>]<93><NN hi><NN lo><EN hi><EN lo>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the event number
<**Dat4>** is the low byte of the event number

Indicates an 'ON' request event. A response event is a reply to a status request (AREQ) without producing an ON or OFF event.

148. 94 Accessory Response Event (AROF)

Format:
[<MjPri><MinPri=3><CANID>]<94><NN hi><NN lo><EN hi><EN lo>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the event number
<**Dat4>** is the low byte of the event number

Indicates an 'OFF' response event. A response event is a reply to a status request (AREQ) without producing an ON or OFF event.

149. 95 Unlearn an event in learn mode (EVULN)

Format:
[<MjPri><MinPri=3><CANID>]<95><NN hi><NN lo><EN hi><EN lo>
Sent by a configuration tool to remove an event from a node.

150. 96 Set a node variable (NVSET)

Format:
[<MjPri><MinPri=3><CANID>]<96><NN hi><NN lo><NV# ><NV val>
Sent by a configuration tool to set a node variable. NV# is the NV index number.

151. 97 Response to a request for a node variable value (NVANS)

Format:
[<MjPri><MinPri=3><CANID>]<97><NN hi><NN lo><NV# ><NV val>
Sent by node in response to request. (NVRD)

Short events. (Device addressing)

Although the producer will send the complete 4 byte event number, the consumer will ignore the producer's node number bytes. This allows a "many to many" situation where producers like DCC handsets can activate the same accessories even though they will have unique node numbers. Clearly this limits the number of 'short' events to 64K-1. For short events, the lower two bytes define the 'Device Number' or DN. The DN can also be considered as a 'device address'.

For these short events, the full 4 byte event is still sent, both to keep the format the same and to allow identification of the producer when required.

152. 98 Accessory Short ON (ASON)

Format:
[<MjPri><MinPri=3><CANID>]<98><NN hi><NN lo><DN hi><DN lo>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the Device Number
<**Dat4>** is the low byte of the Device Number

Indicates an 'ON' event using the short event number of 2 LS bytes.

153. 99 Accessory Short OFF (ASOF)

Format:
[<MjPri><MinPri=3><CANID>]<99><NN hi><NN lo><DN hi><DN lo>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the Device Number
<**Dat4>** is the low byte of the Device Number

Indicates an 'OFF' event using the short event number of 2 LS bytes.

154. 9A Accessory Short Request Event (ASRQ)

Format:
[<MjPri><MinPri=3><CANID>]<9A><NN hi><NN lo><DN hi><DN lo>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the Device Number
<**Dat4>** is the low byte of the Device Number

Indicates a 'request' event using the short event number of 2 LS bytes. A request event is used to elicit a response from a producer 'device' when it is required to know the 'state' of the device without producing an ON or OFF event .

155 9B Response to request for individual node parameter (PARAN)

Format:
[<MjPri><MinPri=3><CANID>]<9B><NN hi><NN lo><Para#><Para val>
NN is the node number of the sending node. Para# is the index of the parameter and Para val is the parameter value.

156 9C Request for read of an event variable (REVAL)

Format:
[<MjPri><MinPri=3><CANID>]<9C><NN hi><NN lo><EN#><EV#>
This request differs from B2 (REQEV) as it doesn't need to be in learn mode but does require the knowledge of the event index to which the EV request is directed.
EN# is the event index. EV# is the event variable index. Response is B5 (NEVAL)

157. 9D Accessory Short Response Event (ARSON)
Format:
[<MjPri><MinPri=3><CANID>]<9D><NN hi><NN lo><DN hi><DN lo>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the device number
<**Dat4>** is the low byte of the device number

Indicates an 'ON' response event. A response event is a reply to a status request (ASRQ) without producing an ON or OFF event.

158. 9E Accessory Short Response Event (ARSOF)
Format:
[<MjPri><MinPri=3><CANID>]<9E><NN hi><NN lo><DN hi><DN lo>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the device number
<**Dat4>** is the low byte of the device number

Indicates an 'OFF' response event. A response event is a reply to a status request (ASRQ) without producing an ON or OFF event.

159. 9F Extended op-code with 3 additional bytes (EXTC3)
Format:
[<MjPri><MinPri=3><CANID>]<9F><Ext_OPC><byte1><byte2><byte3>
Used if the basic set of 32 OPCs is not enough. Allows an additional 256 OPCs

A0-BF 5 data byte packets
[<MjPri><MinPri><CAN ID>]<Opc><Dat1><Dat2><Dat3><Dat4><Dat5>

160. A0 Request 4-byte DCC Packet (*RDCC4*)
Format:
[<MjPri><MinPri=2><CANID>]<A0><REP><Byte0>..<Byte3>
<**Dat1(REP)**> is number of repetitions in sending the packet.
<**Dat2>..<Dat5>** 4 bytes of the DCC packet.
Allows a CAB or equivalent to request a 4 byte DCC packet to be sent to the track. The packet is sent <REP> times and is not refreshed on a regular basis.

161. A1 Reserved

162. A2 Write CV in Service mode (*WCVS*)
Format:
[<MjPri><MinPri=2><CANID>]<A2><Session><High CV#><LowCV#><Mode>
<CVval>
<**Dat1>** is the session number of the cab
<Dat2> is the MSB # of the CV to be written (supports CVs 1 - 65536)
<**Dat3>** is the LSB # of the CV to be written
<**Dat4>** is the service write mode
<**Dat5>** is the CV value to be written
Sent to the command station to write a DCC CV in service mode.

163. A3 Reserved

164. A4 Reserved

165. A5 Reserved

166. A6 Reserved

167. A7 Reserved

168. A8 Reserved

169. A9 Reserved

170. AA Reserved

171. AB Reserved

172. AC Reserved

173. AD Reserved

174. AE Reserved

175. AF Reserved

176. B0 Accessory ON (ACON1)

Format:
[<MjPri><MinPri=3><CANID>]<B0><NN hi><NN lo><EN hi>
<EN lo><data>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the event number
<**Dat4>** is the low byte of the event number
<**Dat5**> is an additional data byte

Indicates an 'ON' event using the full event number of
4 bytes with one additional data byte.

177. B1 Accessory OFF (ACOF1)

Format:
[<MjPri><MinPri=3><CANID>]<B1><NN hi><NN lo><EN hi>
<EN lo><data>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the event number
<**Dat4>** is the low byte of the event number
<**Dat5**> is an additional data byte

Indicates an 'OFF' event using the full event number of 4 bytes with one additional data byte.

178 B2 Read event variable in learn mode (REQEV)

Format:
[<MjPri><MinPri=3><CANID>]<B2><NN hi><NN lo><EN hi>
<EN lo><EV# >
Allows a configuration tool to read stored event variables from a node. EV# is the EV index. Reply is (EVANS)

179. B3 Accessory Response Event (ARON1)

Format:
[<MjPri><MinPri=3><CANID>]<B3><NN hi><NN lo><EN hi>
<EN lo><data>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the event number
<**Dat4>** is the low byte of the event number
<**Dat5**> is the additional data byte 1

Indicates an 'ON' response event with one additional data byte. A response event is a reply to a status request (AREQ) without producing an ON or OFF event.

180. B4 Accessory Response Event (AROF1)

Format:
[<MjPri><MinPri=3><CANID>]<B4><NN hi><NN lo><EN hi>
<EN lo><data>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the event number
<**Dat4>** is the low byte of the event number
<**Dat5**> is the additional data byte 1

Indicates an 'OFF' response event with one additional data byte. A response event is a reply to a status request (AREQ) without producing an ON or OFF event.

181. B5 Response to request for read of EV value (NEVAL)

Format:
[<MjPri><MinPri=3><CANID>]<B5><NN hi><NN lo><EN#>
<EV#><EVval>
NN is the node replying. EN# is the index of the event in that node. EV# is the index of the event variable. EVval is the value of that EV. This is response to 9C (REVAL)

182. B6 Response to Query Node (PNN)

Format:
[<MjPri><MinPri=3><CANID>]<B6><NN Hi><NN Lo><Manuf Id><Module Id><Flags>
<NN Hi**>** is the high byte of the node number
<NN Lo**>** is the low byte of the node number
<Manuf Id**>** is the Manufacturer id as defined in the node parameters
<Module Id**>** is the Module Type Id id as defined in the node parameters
<Flags> is the node flags as defined in the node parameters.

The Flags byte contains bit flags as follows:

Bit 0: Set to 1 for consumer node
Bit 1: Set to 1 for producer node
Bit 2: Set to 1 for FLiM mode
Bit 3: Set to 1 for Bootloader compatible

If a module is both a producer and a consumer then it is referred to as a "combi" node and both flags will be set.

Every node should send this message in response to a QNN message.

183. B7 Reserved

184. B8 Accessory Short ON (ASON1)

Format:
[<MjPri><MinPri=3><CANID>]<B8><NN hi><NN lo><DN hi><DN lo><data 1>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the Device Number
<**Dat4>** is the low byte of the Device Number
<**Dat5**> is the additional data byte 1

Indicates an 'ON' event using the short event number of 2 LS bytes with one added data byte.

185. B9 Accessory Short OFF (ASOF1)

Format:
[<MjPri><MinPri=3><CANID>]<B9><NN hi><NN lo><DN hi><DN lo><data 1>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the Device Number
<**Dat4>** is the low byte of the Device Number
<**Dat5**> is the additional data byte 1

Indicates an 'OFF' event using the short event number of 2 LS bytes with one added data byte.

186. BA Reserved

187. BB Reserved

188. BC Reserved

189. BD Accessory Short Response Event (ARSON1) with one data byte

Format:
[<MjPri><MinPri=3><CANID>]<BD><NN hi><NN lo><DN hi><DN lo><data 1>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the device number
<**Dat4>** is the low byte of the device number
<**Dat5**> is the additional data byte 1

Indicates an 'ON' response event with one added data byte. A response event is a reply to a status request (ASRQ) without producing an ON or OFF event.

190. BE Accessory Short Response Event (ARSOF1) with one data byte

Format:
[<MjPri><MinPri=3><CANID>]<BE><NN hi><NN lo><DN hi><DN lo><data 1>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the device number
<**Dat4>** is the low byte of the device number
<**Dat5**> is the additional data byte 1

Indicates an 'OFF' response event with one added data byte. A response event is a reply to a status request (ASRQ) without producing an ON or OFF event.

191. BF Extended op-code with 4 data bytes (EXTC4)

Format:
[<MjPri><MinPri=3><CANID>]<BF><Ext-OPC><byte1><byte2><byte3 > <byte4>

Used if the basic set of 32 OPCs is not enough. Allows an additional 256 OPCs

C0-DF 6 data byte packets
[<MjPri><MinPri><CAN ID>]<Opc><Dat1><Dat2><Dat3><Dat4><Dat5><Dat6>

192. C0 Request 5-byte DCC Packet (*RDCC5*)

Format:
[<MjPri><MinPri=2><CANID>]<C0><REP><Byte0>..<Byte4>
<**Dat1(REP)**> is # of repetitions in sending the packet.
<**Dat2>..<Dat6>** 5 bytes of the DCC packet.
Allows a CAB or equivalent to request a 5 byte DCC packet to be sent to the track. The packet is sent <REP> times and is not refreshed on a regular basis.

193. C1 Write CV (byte) in OPS mode by address (WCVOA)

Format:
[<MjPri><MinPri=2><CANID>]<C1><AddrH><AddrL><High CV#>
<Low CV#><Mode><Val>
<**Dat1**> and <**Dat2**> are [AddrH] and [AddrL] of the decoder, respectively.
7 bit addresses have (AddrH=0).
14 bit addresses have bits 7,8 of AddrH set to 1.
<Dat3> is the MSB # of the CV to be written (supports CVs 1 - 65536)
<**Dat4>** is the LSB # of the CV to be written
<**Dat5>** is the programming mode to be used
<**Dat6>** is the CV byte value to be written

Sent to the command station to write a DCC CV byte in OPS mode to specific loco (on the main). Used by computer based ops mode programmer that does not have a valid throttle handle.

194. C2 Cab Data (CABDAT)

Format:
[<MjPri><MinPri=2><CANID>]<0xC2><addrH><addrL><datcode><aspect1><aspect2>
<speed>

Transmitted by a layout control system to send data to a cab controlling a specific loco. <addrH> and <addrL> are the loco address in the same format as RLOC and GLOC.7 bit (short) addresses have (addrH=0).14 bit (long) addresses have bits 6,7 of addrH set to 1. <datcode> defines the meaning of the remaining 3 bytes. Values of <datcode> may be defined as required. The following value has currently been defined:01 - CABSIG - Transmitted by a layout control system to send signal aspects to be displayed on a cab handset as cab signalling. Where <datcode> is set to 01, CABSIG, the remaining 3 bytes are defined as follows:

<aspect1> is is signalling system independent, and is defined as follows (colours in brackets correspond to UK colour light signalling, the given aspect names may be displayed differently in other signalling systems):Bits 0-1:2 bit aspect code 00=danger (red), 01=caution (yellow), 10=preliminary caution (double yellow), 11=proceed (green)Bit 2:1 = calling on aspect (bits 0-1 are set to 00 for danger when calling on)Bit 3: 0 indicates upper nibble is feather location, 1 for upper nibble is theatre type route indicator. Bits 5-8:0 = no route indicated, 1 to 6 = feather position or 1 to 15 for theatre indication <aspect1> should be set to 0xFF if no signal information is available. This can be used, for example, to indicate leaving a cab signalling area. A cab should extinguish any currently showing aspect on receipt of this code. Note that because bits 0 and 1 should be set to zero when bit 2 is set, the code 0xFF is not otherwise a valid aspect. <aspect2> may be used as required for other signalling systems. For UK signalling, bit 0 set indicates a flashing aspect, applicable to caution, preliminary caution or proceed.<speed> is a speed limit indication that a cab may optionally display to the driver. If <speed> is not implemented by a layout control system, or whenever speed limit information is not available, this byte should be set to 0xFF (255).

195. C3 Reserved

196. C4 Reserved

197. C5 Reserved

198. C6 Reserved

199. C7 Reserved

200. C8 Reserved

201. C9 Reserved

202. CA Reserved

203. CB Reserved

204. CC Reserved

205. CD Reserved

206. CE Reserved

207. CF Fast Clock (FCLK)
Format:
[<MjPri><MinPri=3><CANID>]<CF><mins><hrs><wdmon> <mday><temp>
<mins**>** is the minutes of the fast clock
<hrs**>** is the hours of the fast clock
<wdmon**>** bits 0-3 are the weekday (1=Sun, 2=Mon etc)
bits 4-7 are the month (1=Jan, 2=Feb etc)
<**div**> Set to 0 for freeze, 1 for real time
<mday> is day of the month 1-31
<temp> Temperature as twos complement -127 to +127

Used to implement a fast clock for the layout.

Note: This definition is at variance with the NMRA Addendum to RP 9.2.1

This addendum defines a time encoding as follows

The data bytes contains CCDDDDDD , there are four bytes in a time packet

CC = 00 DDDDDD = minutes in the range 0 – 59
CC = 10 DDDDDD = 0HHHHHH (sic) the hour in the rang 0-23 (note there is an extra H in the NMRA document)
CC = 01 DDDDDD = 000WWW the day of the week, 0 = Monday etc.
CC = 11 DDDDDD = 00FFFFF (sic) the acceleration factor, 1 means real time, 2 means real time * 2 etc (note there is an extra F in the NMRA document)

There is clearly some redundancy in the codes, particularly CC = 01. There is no way to determine a date, so we could do something like the following

CC = 01 DDDDDD = 1ddddd where ddddd = the day of the month, range 1 to 31
CC = 01 DDDDDD = 01mmmm where mmmm = the month, January = 1

208. D0 Accessory ON (ACON2)

Format:
[<MjPri><MinPri=3><CANID>]<D0><NN hi><NN lo><EN hi><EN lo>
<data1><data2>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the event number
<**Dat4>** is the low byte of the event number
<**Dat5**> is an additional data byte 1
<**Dat6**> is additional data byte 2

Indicates an 'ON' event using the full event number of 4 bytes with two additional data bytes.

209. D1 Accessory OFF (ACOF2)

Format:
[<MjPri><MinPri=3><CANID>]<D1><NN hi><NN lo><EN hi><EN lo>
<data1><data2>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the event number
<**Dat4>** is the low byte of the event number
<**Dat5**> is an additional data byte 1
<**Dat6**> is additional data byte 2

Indicates an 'OFF' event using the full event number of 4 bytes with two additional data bytes.

210. D2 Teach an event in learn mode (EVLRN)

Format:
[<MjPri><MinPri=3><CANID>]<D2><NN hi><NN lo><EN hi><EN lo>
<EV#><EV val>
Sent by a configuration tool to a node in learn mode to teach it an event. Also teaches it the associated event variables (EVs) by the EV index (EV#). This command is repeated for each EV required.

211. D3 Response to a request for an EV value in a node in learn mode (EVANS)

Format:
[<MjPri><MinPri=3><CANID>]<D3><NN hi><NN lo><EN hi><EN lo>
<EV#><EV val>
A node response to a request from a configuration tool for the EVs associated with an event (REQEV). For multiple EVs, there will be one response per request.

212. D4 Accessory Response Event (ARON2)

Format:
[<MjPri><MinPri=3><CANID>]<D4><NN hi><NN lo><EN hi><EN lo>
<data1><data2>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the event number
<**Dat4>** is the low byte of the event number
<**Dat5**> is an additional data byte 1
<**Dat6**> is additional data byte 2

Indicates an ‘ON’ response event with two added data bytes. A response event is a reply to a status request (AREQ) without producing an ON or OFF event.

213. D5 Accessory Response Event (AROF2)

Format:
[<MjPri><MinPri=3><CANID>]<D5><NN hi><NN lo><EN hi><EN lo>
<data1><data2>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the event number
<**Dat4>** is the low byte of the event number
<**Dat5**> is an additional data byte 1
<**Dat6**> is additional data byte 2

Indicates an ‘OFF’ response event with two added data bytes. A response event is a reply to a status request (AREQ) without producing an ON or OFF event.

214. D6 Reserved

215. D7 Reserved

216. D8 Accessory Short ON (ASON2)

Format:
[<MjPri><MinPri=3><CANID>]<D8><NN hi><NN lo><DN hi><DN lo>
<data 1><data 2>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the Device Number
<**Dat4>** is the low byte of the Device Number
<**Dat5**> is the additional data byte 1
<**Dat6**> is additional data byte 2

Indicates an ‘ON’ event using the short event number of 2 LS bytes with two added data bytes.

217. D9 Accessory Short OFF (ASOF2)

Format:
[<MjPri><MinPri=3><CANID>]<D9><NN hi><NN lo><DN hi><DN lo>
<data 1><data 2>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the Device Number
<**Dat4>** is the low byte of the Device Number
<**Dat5**> is the additional data byte 1
<**Dat6**> is additional data byte 2

Indicates an ‘OFF’ event using the short event number of 2 LS bytes with two added data bytes.

218. DA Reserved

219. DB Reserved

220. DC Reserved

221. DD Accessory Short Response Event (ARSON2) with two data bytes
Format:
[<MjPri><MinPri=3><CANID>]<DD><NN hi><NN lo><DN hi><DN lo>
<data 1><data 2>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the device number
<**Dat4>** is the low byte of the device number
<**Dat5**> is the additional data byte 1
<**Dat6**> is the additional data byte 2

Indicates an 'ON' response event with two added data bytes.
A response event is a reply to a status request (ASRQ) without producing an ON or OFF event.

222. DE Accessory Short Response Event (ARSOF2) with two data bytes
Format:
[<MjPri><MinPri=3><CANID>]<DE><NN hi><NN lo><DN hi><DN lo>
<data 1><data 2>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the device number
<**Dat4>** is the low byte of the device number
<**Dat5**> is the additional data byte 1
<**Dat6**> is the additional data byte 2

Indicates an 'OFF' response event with two added data bytes.
A response event is a reply to a status request (ASRQ) without producing an ON or OFF event.

223. DF Extended op-code with 5 data bytes (EXTC5)
Format:
[<MjPri><MinPri=3><CANID>]<DF><Ext-OPC><byte1><byte2><byte3 >
<byte4><byte5>
Used if the basic set of 32 OPCs is not enough. Allows an additional 256 OPCs

E0-FF 7 data byte packets
[<MjPri><MinPri><CAN ID>]<OPC><Dat1>..<Dat7>

224. E0 Request 6-byte DCC Packet (*RDCC6*)

Format:
[<MjPri><MinPri=2><CANID>]<E0><REP><Byte0>..<Byte5>
<**Dat1(REP)**> is number of repetitions in sending the packet.
<**Dat2>..<Dat7>** 6 bytes of the DCC packet.
Allows a CAB or equivalent to request a 6 byte DCC packet to be sent to the track. The packet is sent <REP> times and is not refreshed on a regular basis.

225. E1 Engine report (*PLOC*)

Format:
[<MjPri><MinPri=2><CANID>]<E1><Session><AddrH><AddrL>
<Speed/Dir><Fn1><Fn2><Fn3>

<**Dat1**> Session for engine assigned by the command station. This session number is used in all referenced to the engine until it is released.
<**Dat2**> is the MS byte of the DCC address. For short addresses it is set to 0.
<**Dat3>** is the LS byte of the DCC address. If the engine is consisted, this is the consist address.
<**Dat4>** is the Speed/Direction value. Bit 7 is the direction bit and bits 0-6 are the speed value.
<**Dat5**> is the function byte F0 to F4
<**Dat6>** is the function byte F5 to F8
<**Dat7>** is the function byte F9 to F12

A report of an engine entry sent by the command station. Sent in response to QLOC or as an acknowledgement of acquiring an engine requested by a cab (RLOC or GLOC).

226. E2 Response to request for node name string (NAME)

Format:
[<MjPri><MinPri=3><CANID>]<E2><char1><char2><char3><char4>
<char5><char6><char7>

A node response while in ‘setup’ mode for its name string. Reply to (RQMN). The string for the module type is returned in char1 to char7, space filled to 7 bytes. The Module Name prefix , currently either CAN or ETH, depends on the Interface Protocol parameter, it is not included in the response.

227. E3 Command Station status report (*STAT*)

Format:
[<MjPri><MinPri=2><CANID>]<E3><NN hi><NN lo><CS num><flags>
<Major rev><Minor rev><Build no.>
<NN hi> <NN lo> Gives node id of command station, so further info can be got from parameters or interrogating NVs
<CS num> For future expansion - set to zero at present
<flags> Flags as defined below
<Major rev> Major revision number
<Minor rev> Minor revision letter
<Build no.> Build number, always 0 for a released version.

<flags> is status defined by the bits below.

bits:
0 - Hardware Error (self test)
1 - Track Error
2 - Track On/ Off
3 - Bus On/ Halted
4 - EM. Stop all performed
5 - Reset done
6 - Service mode (programming) On/ Off
7 – reserved

Sent by the command station in response to RSTAT or by any device when an error state occurs.

228. E4 Reserved

229. E5 Reserved

230. E6 Reserved

231. E7 Reserved

232. E8 Reserved for streaming protocol See Appendix A for details.

233. E9 Reserved for streaming protocol

234. EA Reserved for streaming protocol

235. EB Reserved for streaming protocol

236. EC Reserved for streaming protocol

237. ED Reserved for streaming protocol

238. EE Reserved for streaming protocol

239. EF Response to request for node parameters (PARAMS)
Format:
[<MjPri><MinPri=3><CANID>]<EF><PARA 1><PARA 2><PARA 3>
<PARA 4><PARA 5><PARA 6><PARA 7>
A node response while in ‘setup’ mode for its parameter string. Reply to (RQNP)

240. F0 Accessory ON (ACON3)
Format:
[<MjPri><MinPri=3><CANID>]<F0><NN hi><NN lo><EN hi><EN lo>
<data1><data2><data3>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the event number
<**Dat4>** is the low byte of the event number
<**Dat5**> is an additional data byte 1
<**Dat6**> is additional data byte 2
<**Dat7**> is additional data byte 3

Indicates an ‘ON’ event using the full event number of 4 bytes with three additional data bytes.

241. F1 Accessory OFF (ACOF3)

Format:
[<MjPri><MinPri=3><CANID>]<F1><NN hi><NN lo><EN hi><EN lo>
<data1><data2><data3>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the event number
<**Dat4>** is the low byte of the event number
<**Dat5**> is an additional data byte 1
<**Dat6**> is additional data byte 2
<**Dat7**> is additional data byte 3

Indicates an ‘OFF’ event using the full event number of 4 bytes with three additional data bytes.

242. F2 Response to request to read node events (ENRSP)

Format:
[<MjPri><MinPri=3><CANID>]<F2><NN hi><NN lo>
<EN3><EN2><EN1><EN0><EN#>

Where the NN is that of the sending node. EN3 to EN0 are the four bytes of the stored event. EN# is the index of the event within the sending node. This is a response to either 57 (NERD) or 72 (NENRD)

243. F3 Accessory Response Event (ARON3)

Format:
[<MjPri><MinPri=3><CANID>]<F3><NN hi><NN lo><EN hi><EN lo>
<data1><data2><data3>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the event number
<**Dat4>** is the low byte of the event number
<**Dat5**> is an additional data byte 1
<**Dat6**> is additional data byte 2
<**Dat7**> is additional data byte 3

Indicates an ‘ON’ response event with three added data bytes. A response event is a reply to a status request (AREQ) without producing an ON or OFF event.

244. F4 Accessory Response Event (AROF3)

Format:
[<MjPri><MinPri=3><CANID>]<F4><NN hi><NN lo><EN hi><EN lo>
<data1><data2><data3>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the event number
<**Dat4>** is the low byte of the event number
<**Dat5**> is an additional data byte 1
<**Dat6**> is additional data byte 2
<**Dat7**> is additional data byte 3

Indicates an ‘OFF’ response event with three added data bytes. A response event is a reply to a status request (AREQ) without producing an ON or OFF event.

245. F5 Teach an event in learn mode using event indexing (EVLRNI)

Format:
[<MjPri><MinPri=3><CANID>]<F5><NN hi><NN lo><EN hi><EN lo>
<EN#><EV#><EV val>
Sent by a configuration tool to a node in learn mode to teach it an event. The event index must be known. Also teaches it the associated event variables.(EVs). This command is repeated for each EV required.

246. F6 Accessory node data event (ACDAT)

Format:
[<MjPri><MinPri=3><CANID>]<F6><NN hi><NNlo>
<data1><data2><data3><data4><data5>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the first node data byte
<**Dat4>** is the second node data byte
<**Dat5**> is the third node data byte
<**Dat6**> is the fourth node data byte
<**Dat7**> is the fifth node data byte

Indicates an event from this node with 5 bytes of data.
For example, this can be used to send the 40 bits of an RFID tag. There is no event number in order to allow space for 5 bytes of data in the packet, so there can only be one data event per node.

247. F7 Accessory node data Response (ARDAT)

Format:
[<MjPri><MinPri=3><CANID>]<F7><NN hi><NN lo>
<data1><data2><data3><data4><data5>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the first node data byte
<**Dat4>** is the second node data byte
<**Dat5**> is the third node data byte
<**Dat6**> is the fourth node data byte
<**Dat7**> is the fifth node data byte

Indicates a node data response. A response event is a reply to a status request (RQDAT) without producing a new data event.

248. F8 Accessory Short ON (ASON3)

Format:
[<MjPri><MinPri=3><CANID>]<F8><NN hi><NN lo><DN hi><DN lo>
<data 1><data 2><data 3>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the Device Number
<**Dat4>** is the low byte of the Device Number
<**Dat5**> is the additional data byte 1
<**Dat6**> is additional data byte 2
<**Dat7**> is additional data byte 3

Indicates an ‘ON’ event using the short event number of 2 LS bytes with three added data bytes.

249. F9 Accessory Short OFF (ASOF3)

Format:
[<MjPri><MinPri=3><CANID>]<F9><NN hi><NN lo><DN hi><DN lo>
<data 1><data 2><data 3>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the Device Number
<**Dat4>** is the low byte of the Device Number
<**Dat5**> is the additional data byte 1
<**Dat6**> is additional data byte 2
<**Dat7**> is additional data byte 3

Indicates an ‘OFF’ event using the short event number of 2 LS bytes with three added data bytes.

250. FA Device data event (short mode) (DDES)

Format:
[<MjPri><MinPri=3><CANID>]<FA><DN hi><DN lo>
<data1><data2><data3><data4><data5>
<**Dat1>** is the high byte of the device number
<**Dat2>** is the low byte of the device number
<**Dat3>** is the first device data byte
<**Dat4>** is the second device data byte
<**Dat5**> is the third device data byte
<**Dat6**> is the fourth device data byte
<**Dat7**> is the fifth device data byte

Function is the same as F6 but uses device addressing so can relate data to a device attached to a node. e.g. one of several RFID readers attached to a single node.

251. FB Device data response (short mode) (DDRS)

Format:
[<MjPri><MinPri=3><CANID>]<FB><DN hi><DN lo>
<data1><data2><data3><data4><data5>
<**Dat1>** is the high byte of the device number
<**Dat2>** is the low byte of the device number
<**Dat3>** is the first device data byte
<**Dat4>** is the second device data byte
<**Dat5**> is the third device data byte
<**Dat6**> is the fourth device data byte
<**Dat7**> is the fifth device data byte

The response to a request for data from a device. (0x5B)

252. FC Device data write (short mode) (DDWS)

Format:
[<MjPri><MinPri=3><CANID>]<FC><DN hi><DN lo>
<data1><data2><data3><data4><data5>
<**Dat1>** is the high byte of the device number
<**Dat2>** is the low byte of the device number
<**Dat3>** is the first device data byte

<**Dat4>** is the second device data byte
<**Dat5**> is the third device data byte
<**Dat6**> is the fourth device data byte
<**Dat7**> is the fifth device data byte

Uses device addressing so can relate data to a device attached to a node. e.g. one of several RFID readers attached to a single node. This is a complement to DDES for tag writing.

253. FD Accessory Short Response Event (ARSON3)
Format:
[<MjPri><MinPri=3><CANID>]<FD><NN hi><NN lo><DN hi><DN lo>
<data 1><data 2><data 3>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the device number
<**Dat4>** is the low byte of the device number
<**Dat5**> is the additional data byte 1
<**Dat6**> is the additional data byte 2
<**Dat7**> is the additional data byte 3

Indicates an 'ON' response event with with three added data bytes. A response event is a reply to a status request (ASRQ) without producing an ON or OFF event.

254. FE Accessory Short Response Event (ARSOF3)
Format:
[<MjPri><MinPri=3><CANID>]<FE><NN hi><NN lo><DN hi><DN lo>
<data 1><data 2><data 3>
<**Dat1>** is the high byte of the node number
<**Dat2>** is the low byte of the node number
<**Dat3>** is the high byte of the device number
<**Dat4>** is the low byte of the device number
<**Dat5**> is the additional data byte 1
<**Dat6**> is the additional data byte 2
<**Dat7**> is the additional data byte 3

Indicates an 'OFF' response event with with three added data bytes. A response event is a reply to a status request (ASRQ) without producing an ON or OFF event.

255. FF Extended op-code with 6 data bytes (EXTC6)
Format:
[<MjPri><MinPri=3><CANID>]<FF><Ext-OPC><byte1><byte2><byte3 >
<byte4><byte5><byte6>
Used if the basic set of 32 OPCs is not enough. Allows an additional 256 OPCs

CBUS OpCode list with brief description and cross referencing. Applies to Rev 8e. (14/10/20)

OPC number	OPC HEX value	Mnemonic	Reference	Ref. OPC	Acc.	DCC	Config.	General	Description
0	00	ACK						x	General acknowledgement - affirmative
1	01	NAK						x	General acknowledgement - negative
2	02	HLT						x	CAN bus not available / busy
3	03	BON						x	CAN bus available
4	04	TOF				x			DCC track off
5	05	TON				x			DCC Track on
6	06	ESTOP				x			Emergency stop all
7	07	ARST						x	System reset
8	08	RTOF				x			Request track off
9	09	RTON				x			Request track on
10	0A	RESTP				x			Request emergency stop all
11	0B								
12	0C	RSTAT	STAT	E3			x		Query status of command station
13	0D	QNN	PNN	B6			x		Query node status
14	0E								
15	0F								
16	10	RQNP	PARAMS	EF			x		Request node parameters
17	11	RQMN	NAME	E2			x		Request module name
18	12								
19	13								
20	14								
21	15								
22	16								
23	17								
24	18								
25	19								
26	1A								
27	1B								
28	1C								
29	1D								
30	1E								
31	1F								

OPC number	OPC HEX	Mnemonic	Reference	Ref. OPC	Acc.	DCC	Config	General	Description
32	20								
33	21	KLOC				x			Release engine
34	22	QLOC	PLOC/ERR	E1 / 63		x			Query engine
35	23	DKEEP				x			Session keepalive from CAB
36	24								
37	25								
38	26								
39	27								
40	28								
41	29								
42	2A								
43	2B								
44	2C								
45	2D								
46	2E								
47	2F								
48	30	DBG1						x	Debug. For development only
49	31								
50	32								
51	33								
52	34								
53	35								
54	36								
55	37								
56	38								
57	39								
58	3A								
59	3B								
60	3C								
61	3D								
62	3E								
63	3F	EXTC						x	Extended OPC with no added bytes

OPC number	OPC HEX	Mnemonic	Reference	Ref. OPC	Acc.	DCC	Config	General	Description
64	40	RLOC	PLOC / ERR	E1 / 63		x			Request engine session
65	41	QCON	PLOC / ERR	E1 / 63		x			Query consist
66	42	SNN	RQNN	50			x		Set node number (node in 'setup')
67	43	ALOC				x			Allocate loco to ‘assignment or activity’
68	44	STMOD				x			Set CAB session mode
69	45	PCON				x			Set loco into consist (advanced)
70	46	KCON				x			Remove loco from consist
71	47	DSPD				x			Set engine speed / direction
72	48	DFLG				x			Set engine (session) flags
73	49	DFNON				x			Set engine function ON
74	4A	DFNOF				x			Set engine function OFF
75	4B								
76	4C	SSTAT				x			Service mode status
77	4D								
78	4E								
79	4F	NNRSM						x	Reset a node to the manufacturers defaults
80	50	RQNN	SNN	42			x		Request node number
81	51	NNREL					x		Node number release
82	52	NNACK	SNN	42 / 5D			x		Node number acknowledge (node in 'setup')
83	53	NNLRN					x		Set node into learn mode
84	54	NNULN					x		Release node from learn mode
85	55	NNCLR					x		Clear all events from a node
86	56	NNEVN	EVLNF	70			x		Read number of events available
87	57	NERD	ENRSP	F2			x		Read back all events in a node
88	58	RQEVN	NUMEV	74			x		Read number of stored events in node
89	59	WRACK					x		Write acknowledge. (Handshake)
90	5A	RQDAT	ARDAT	F7	x				Request node data event
91	5B	RQDDS	DDRS	FB	x				Request device data (short)
92	5C	BOOTM					x		Put node into 'bootloader' mode
93	5D	ENUM					x		Force self enumeration of CAN_ID
94	5E	NNRST						x	Software reset a node
95	5F	EXTC1						x	Extended OPC with one added byte

OPC number	OPC HEX	Mnemonic	Reference	Ref. OPC	Acc.	DCC	Config	General	Description
96	60	DFUN				x			Set engine functions (DCC format)
97	61	GLOC				x			Get engine session – used in dispatching
98	62								
99	63	ERR				x			Command station error report
100	64								
101	65								
102	66								
103	67								
104	68								
105	69								
106	6A								
107	6B								
108	6C								
109	6D								
110	6E								
111	6F	CMDERR					x		Error message during configuration
112	70	EVNLF	NNEVN	56			x		Event space left
113	71	NVRD	NVANS	97			x		Request read of node variable
114	72	NENRD	ENRSP	F2			x		Request read of events by index
115	73	RQNPN	PARAN	9B			x		Request read of node parameter by index
116	74	NUMEV	RQEVN	58			x		Number of events stored in node
117	75	CANID					x		Force a specific CAN_ID
118	76								
119	77								
120	78								
121	79								
122	7A								
123	7B								
124	7C								
125	7D								
126	7E								
127	7F	EXTC2						x	Extended OPC with two added bytes

OPC number	OPC HEX	Mnemonic	Reference	Ref. OPC	Acc.	DCC	Config	General	Description
128	80	RDCC3				x			Request 3 byte DCC packet
129	81								
130	82	WCVO				x			Write CV in OPS mode (byte)
131	83	WCVB				x			Write CV in OPS mode (bit)
132	84	QCVS	PCVS	85		x			Request read CV (service mode)
133	85	PCVS	QCVS	84		x			Report CV (sevice mode)
134	86								
135	87								
136	88								
137	89								
138	8A								
139	8B								
140	8C								
141	8D								
142	8E								
143	8F								
144	90	ACON			x				Accessory ON event (long)
145	91	ACOF			x				Accessory OFFevent (long)
146	92	AREQ			x				Accessory status request (long)
147	93	ARON	AREQ	92	x				Accessory response ON (long)
148	94	AROF	AREQ	92	x				Accessory response OFF (long)
149	95	EVULN					x		Unlearn an event in learn mode
150	96	NVSET					x		Set a node variable
151	97	NVANS	NVRD	71			x		Node variable value response
152	98	ASON			x				Accessory ON event (short)
153	99	ASOF			x				Accessory OFFevent (short)
154	9A	ASRQ			x				Accessory status request (short)
155	9B	PARAN	RQNPN	73			x		Parameter readback by index
156	9C	REVAL	NEVAL	B5			x		Request read of event variable
157	9D	ARSON	ASRQ	9A	x				Accessory response ON (short)
158	9E	ARSOF	ASRQ	9A	x				Accessory response OFF (short)
159	9F	EXTC3						x	Extended OPC with three added bytes

OPC number	OPC HEX	Mnemonic	Reference	Ref. OPC	Acc.	DCC	Config	General	Description
160	A0	RDCC4				x			Request 4 byte DCC packet.
161	A1								
162	A2	WCVS				x			Write CV in service mode
163	A3								
164	A4								
165	A5								
166	A6								
167	A7								
168	A8								
169	A9								
170	AA								
171	AB								
172	AC								
173	AD								
174	AE								
175	AF								
176	B0	ACON1			x				Accessory ON event with one added byte (long)
177	B1	ACOF1			x				Accessory OFF event with one added byte (long)
178	B2	REQEV	EVANS	D3			x		Read event variable in learn mode
179	B3	ARON1	AREQ	92	x				Accessory response event ON with one added byte (long)
180	B4	AROF1	AREQ	92	x				Accessory response event OFF with one added byte (long)
181	B5	NEVAL	REVAL	9C			x		Read of EV value response
182	B6	PNN	QNN	0D				x	Response to query node
183	B7								
184	B8	ASON1			x				Accessory ON event with one added byte (short)
185	B9	ASOF1			x				Accessory OFF event with one added byte (short)
186	BA								
187	BB								
188	BC								
189	BD	ARSON1	ASRQ	9A	x				Accessory response ON with one added data byte (short)
190	BE	ARSOF1	ASRQ	9A	x				Accessory response OFF with one added data byte (short)
191	BF	EXTC4						x	Extended OPC with four added bytes

OPC number	OPC HEX	Mnemonic	Reference	Ref. OPC	Acc.	DCC	Config	General	Description
192	C0	RDCC5				x			Request 5 byte DCC packet.
193	C1	WCVOA				x			Write CV in OPS mode by address
194	C2	CABDAT				x			Send layout information to a cab.
195	C3								
196	C4								
197	C5								
198	C6								
199	C7								
200	C8								
201	C9								
202	CA								
203	CB								
204	CC								
205	CD								
206	CE								
207	CF	FCLK			x				Fast clock
208	D0	ACON2			x				Accessory ON event with two added bytes (long)
209	D1	ACOF2			x				Accessory OFF event with two added bytes (long)
210	D2	EVLRN					x		Teach event in learn mode
211	D3	EVANS	REQEV	B2			x		Response to request for EV value in learn mode
212	D4	ARON2	AREQ	92	x				Accessory response event ON with two added bytes (long)
213	D5	AROF2	AREQ	92	x				Accessory response event OFF with two added bytes (long)
214	D6								
215	D7								
216	D8	ASON2			x				Accessory ON event with two added bytes (short)
217	D9	ASOF2			x				Accessory OFF event with two added bytes (short)
218	DA								
219	DB								
220	DC								
221	DD	ARSON2	ASRQ	9A	x				Accessory response ON with two added data bytes (short)
222	DE	ARSOF2	ASRQ	9A	x				Accessory response OFF with two added data bytes (short)
223	DF	EXTC5						x	Extended OPC with five added bytes

OPC number	OPC HEX	Mnemonic	Reference	Ref. OPC	Acc.	DCC	Config	General	Description
224	E0	RDCC6				x			Request 6 byte DCC packet.
225	E1	PLOC	RLOC	40		x			Engine report from command station
226	E2	NAME	RQMN	11			x		Response to request for node name
227	E3	STAT	RSTAT	0C		x			Command station status report
228	E4								
229	E5								
230	E6								
231	E7								
232	E8	?							Streaming protocol frame
233	E9								Reserved for future streaming protocol
234	EA								Reserved for future streaming protocol
235	EB								Reserved for future streaming protocol
236	EC								Reserved for future streaming protocol
237	ED								Reserved for future streaming protocol
238	EE								Reserved for future streaming protocol
239	EF	PARAMS	RQNP	10			x		Response to request for node parameters (in setup)
240	F0	ACON3			x				Accessory ON event with three added bytes (long)
241	F1	ACOF3			x				Accessory OFF event with three added bytes (long)
242	F2	ENRSP	NERD/NENRD	57 / 72			x		Response to request to read node events
243	F3	ARON3	AREQ	92	x				Accessory response event ON with three added bytes (long)
244	F4	AROF3	AREQ	92	x				Accessory response event OFF with three added bytes (long)
245	F5	EVLRNI					x		Teach event in learn mode using event indexing
246	F6	ACDAT			x				Accessory node data event. 5 data bytes (long)
247	F7	ARDAT	RQDAT	5A	x				Accessory node data response. 5 data bytes (long)
248	F8	ASON3			x				Accessory ON event with three added bytes (short)
249	F9	ASOF3			x				Accessory OFF event with three added bytes (short)
250	FA	DDES			x				Accessory node data event. 5 data bytes (short)
251	FB	DDRS	RQDDS	5B	x				Accessory node data response. 5 data bytes (short)
252	FC	DDWS			x				Write accessory node data event. 5 data bytes (short)
253	FD	ARSON3	ASRQ	9A	x				Accessory response ON with three added data bytes (short)
254	FE	ARSOF3	ASRQ	9A	x				Accessory response OFF with three added data bytes (short)
255	FF	EXTC6							Extended OPC with six added bytes

Alphabetic sort of OPCs

Mnemonic		OPC number	OPC HEX value		Mnemonic		OPC number	OPC HEX value
ACDAT		246	F6		EXTC5		223	DF
ACK		0	0		EXTC6		255	FF
ACOF		145	91		FCLK			
ACOF1		177	B1		GLOC		97	61
ACOF2		209	D1		HLT		2	2
ACOF3		241	F1		KCON		70	46
ACON		144	90		KLOC		33	21
ACON1		176	B0		NAK		1	1
ACON2		208	D0		NAME		226	E2
ACON3		240	F0		NENRD		114	72
ALOC		67	43		NERD		87	57
ARDAT		247	F7		NEVAL		181	B5
AREQ		146	92		NNACK		82	52
AROF		148	94		NNCLR		85	55
AROF1		180	B4		NNEVN		86	56
AROF2		213	D5		NNLRN		83	53
AROF3		244	F4		NNREL		81	51
ARON		147	93		NNRSM			4F
ARON1		179	B3		NNRST			5E
ARON2		212	D4		NNULN		84	54
ARON3		243	F3		NUMEV		116	74
ARSOF		158	9E		NVANS		151	97
ARSOF1		190	BE		NVRD		113	71
ARSOF2		220	DE		NVSET		150	96
ARSOF3		254	FE		PARAMS		239	EF
ARSON		157	9D		PARAN		155	9B
ARSON1		189	BD		PCON		69	45
ARSON2		221	DD		PCVS		133	85
ARSON3		253	FD		PLOC		225	E1
ARST		7	7		QCON		65	41
ASOF		153	99		QCVS		132	84
ASOF1		185	B9		QLOC		34	22
ASOF2		217	D9		QNN		13	0D
ASOF3		249	F9		RDCC3		128	80
ASON		152	98		RDCC4		160	A0
ASON1		184	B8		RDCC5		192	C0
ASON2		216	D8		RDCC6		224	E0
ASON3		248	F8		REQEV		178	B2
ASRQ		154	9A		RESTP		10	0A
BON		3	3		REVAL		156	9C
BOOTM		92	5C		RLOC		64	40
CABDAT		194	C2		RQDAT		90	5A
CANID		117	75		RQDDS		91	5B
CMDERR		111	6F		RQEVN		88	58
DBG1		48	30		RQMN		17	11
DDES		250	FA		RQNN		80	50
DDRS		251	FB		RQNP		16	10
DDWS		252	FC		RQNPN		115	73
DFLG		72	48		RSTAT		12	0C
DFNOF		74	4A		RTOF		8	8
DFNON		73	49		RTON		9	9
DFUN		96	60		SNN		66	42

DKEEP		35	23		SSTAT		76	4C
DSPD		71	47		STAT		227	E3
ENRSP		242	F2		STMOD		68	44
ENUM		93	5D		TOF		4	4
ERR		99	63		TON		5	5
ESTOP		6	6		WCVB		131	83
EVANS		211	D3		WCVO		130	82
EVLRN		210	D2		WCVOA		193	C1
EVLRNI		245	F5		WCVS		162	A2
EVNLF		112	70		WRACK		89	59
EVULN		149	95					
EXTC		63	3F					
EXTC1		95	5F					
EXTC2		127	7F					
EXTC3		159	9F					
EXTC4		191	BF					

Error Codes

Accessory Module – Error codes

Error codes for CBUS accessory modules, these error codes are returned by OPC CMDERR 0x6F

1	Command Not Supported	- see note 1.
2	Not In Learn Mode	
3	Not in Setup Mode	- see note 1
4	Too Many Events	
5	Reserved	
6	Invalid Event variable index	
7	Invalid Event	
8	Reserved	- see note 2
9	Invalid Parameter Index	
10	Invalid Node Variable Index	
11	Invalid Event Variable Value	
12	Invalid Node Variable Value	

Note 1: Accessory modules do not return this error
Note 2: Currently used by code that processes OPC REVAL 0x9C but this code should be updated to use codes 6 & 7.

DCC Error codes

These codes are returned by OPC ERR 0x63

1	Loco stack full	First two bytes are loco address, third is error number.
2	Loco address taken	- First two byes are loco address, third is error number.
3	Session not present	- First byte session id, second byte zero, third is error number.
4	Consist empty	- First byte consist id, second byte zero, third is error number.
5	Loco not found	- First byte session id, second byte zero,, third is error number.
6	CAN bus error	- Two data bytes set to zero (not used), third is error number. - This would be sent out in the unlikely event that the command station buffers overflow.
7	Invalid request	- First two bytes are loco address, third is error number. Indicates an invalid or inconsistent request. For example, a GLOC request with both steal and share flags set.
8	Session cancelled	- First byte session id, second byte zero, third is error number. Sent to a cab to cancel the session when another cab is stealing that session.