CBUS© 4.0 Specification Rev. 7 fe

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.

Sessions for Loco Control

Messages for loco control include a session number (sometimes referred to a session, session handle or just handle). This is a unique identifier assigned by the command station when a cab (or throttle) first requests control of a loco. The session number is used in all further messages referring to that loco

Cabs that are capable of service mode programming must have a valid session, but this needn’t refer to the loco on the service mode programming track. The session is required for error handling, in case two or more cabs attempt overlapping service mode operations.

DCC Addresses

When CBUS is used for DCC loco control, the DCC address is always sent as two bytes. For short addresses, the MSB will be zero. For long addresses the MSB will have bit 7 and 6 set.

DCC Programming

DCC programming modes are selected by unique op-codes or mode bytes, depending upon the operation. For those operations that use a mode byte, the modes are defined as follows:

0 Direct Byte
1 Direct Bit
2 Page Mode#
3 Register Mode
4 Address Only Mode

Not all CBUS DCC command stations will support all modes.

Programming operations, other than a successful service mode read, will return SSTAT when complete. The associated status byte is defined as follows:

0 Reserved
1 No Acknowlegde
2 Overload on service mode programming track
3 Write Acknowledge
4 Busy
5 CV out of range

Successful service mode read operations return the data with PCVS, rather than SSTAT

DCC Error Codes

Error codes for OPC ERR are defined as follows:

0 Reserved
1 Loco stack full
2 Loco address taken by another cab
3 Session not present
4 No more engines
5 Engine not found

Packet Definitions (by OPC field)

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

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

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

02 Bus Halt (*HLT*) Format:
[<MjPri><MinPri=0><CANID>]<02>
Commonly broadcasted to all devices to indicate CBUS is not available and no further packets should be sent until a BON or ARST is received.
Currently only sent by the command station when its input queue is full. Can be ignored by all modules other than those that need to send to the command station (e.g. cabs)

03 Bus ON (*BON*) Format:
[<MjPri><MinPri=1><CANID>]<03>
Commonly broadcasted to all devices to indicate CBUS is available.
Currently only sent by the command station when it has space in its input queue again

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

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

06 Track Stopped (*ESTOP*) Format:
[<MjPri><MinPri=1><CANID>]<06>
Commonly broadcasted to all devices by a command station to indicate all engines have been emergency stopped.

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

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

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

0A Request Emergency Stop ALL (*RESTP*) Format:
[<MjPri><MinPri=0><CANID>]<0A>
sent by a device to request an emergency stop of the entire layout.

0B Request firmware Version (*RVER*) Format:
[<MjPri><MinPri=2><CANID>]<0B>
Sent by any device to request firmware version information from a node in ‘setup’ mode

0C Request node Status (*RSTAT*) Format:
[<MjPri><MinPri=2><CANID>]<0C>
Sent by a node to query the status of the command station or another device. See description of (STAT) for the response from the command station/ other devices.

0D Request node number (RQNN) Format:
[<MjPri><MinPri=3><CANID>]<0D>
Sent by a device joining the bus that is in setup/configuration mode and does not have a node number (NN) assigned. The device allocating node numbers responds with (SNN) which contains the newly assigned node number.

0E – 0F reserved

10 Request node parameters(RQNP) Format:
[<MjPri><MinPri=3><CANID>]<10>

Sent by device to a node while in ‘setup’mode to read its parameter set. Used when initially configuring a node.

11 – 1F reserevd

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

20 Command Station status report(*STAT*) Format:
[<MjPri><MinPri=2><CANID>]<20><SSSSSSSS>
<**Dat1**> is status,
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.

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

22 Query engine (*QLOC*) Format:
[<MjPri><MinPri=2><CANID>]<22><Session>
<**Dat1**> is the engine session number.
The command station responds with PLOC if the session is assigned. Otherwise responds with ERR: engine not found.

23 – 2F reserved

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

31 - 3E reserved

3F Extended op-code with no data byte packet (EXTC)
[<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>

40 Request engine assignment(*RLOC*) Format:
[<MjPri><MinPri=2><CANID>]<40><AAAAAAAA><AAAAAAAA>
<**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. The command station responds with (PLOC) if engine is free and is being assigned. Otherwise responds with (ERR): engine in use or (ERR:) stack full. This command is typically sent by a cab to the command station following a change of the controlled decoder address.

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: no more engines.

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 device to assign a node number to a requesting node in response to a *RQNN* message.
The target node must be in ‘setup’ mode.

43 Reserved

44 Set Throttle 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

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.
If engine is consisted already, an error is reported: engine in use.???

46 Remove Engine from consist (*KCON*) Format:
[<MjPri><MinPri=2><CANID>]<46><Session><Consist#>
<**Dat1**> Session number
<**Dat2**> is consist address.
Removes a decoder from a consist.

If engine is not consisted, an error is reported:
Engine not found.

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 7 lsb are unsigned speed value.
Sent by a device to request an engine speed/dir change.

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.

49- 4B Reserved

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

4D- 4F Reserved

50 Node number acknowledge. (NNACK) Format:
[<MjPri><MinPri=3><CANID>]<50><NN hi><NN lo>
Sent by device to verify presence of a specific node.

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

52 Node ‘keepalive’. (NNREF) Format:
[<MjPri><MinPri=3><CANID>]<52><NN hi><NN lo>

Sent by node to indicate it is still active.

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

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

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

56 Read number of events available in a node (NNEVN)
Format: [<MjPri><MinPri=3><CANID>]<56><NN hi><NN lo>
Sent by device to read the number of available event slots in a node.

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

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

59 Write acknowledge. (WRACK)
Format: [<MjPri><MinPri=3><CANID>]<59><NN hi><NN lo>
Sent by device to indicate the completion of a write to memory operation.

Used when teaching nodes where the processing time may be slow.

5A Request node data event (RQDAT)
Format: [<MjPri><MinPri=3><CANID>]<5A><NN hi><NN lo>
Sent by device to read the node data event (eg: RFID data). Response is 0xE4 (ARDAT).

5B Reserved

5C Put node into bootload mode. (BOOTM) For SLiM consumers with no NN, the command is executed but the NN is ignored.

Format: [<MjPri><MinPri=3><CANID>]<5C><NN hi><NN lo>
Sent by a device to prepare a node for loading a new program.

5D – 5E Reserved

5F Extended op-code with 1 data byte packet. (EXTC1)
[<MjPri><MinPri=3><CANID>]<5F><Ext_OPC><param1>

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>

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 F19
5 is F20 to F28
<Dat3> is the DCC format function byte for that range in corresponding bits. Sent by a device to request an engine Fn state change.

61 – 62 Reserved

63 Command Station Error report(*ERR*) Format:
[<MjPri><MinPri=2><CANID>]<63><Adr-hi><Adr_lo><Error>
<Dat1> is high byte of loco address
<Dat2> is low byte of loco address
<Dat3> is error code –
Error 1. Loco stack full
Error 2. Loco address taken
Error 3. Session number not present.
Other error numbers to be allocated
Sent in response to an error situation.

64 – 6E reserved

6F Error messages from nodes during configuration (CMDERR)
[<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. Error numbers to be defined.

70 Event space left reply from 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.

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.

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 F2 (ENRSP)

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.
Response is 9B~~3~~ (PARAN)

74 Number of events stored in node (NUMEV)
Format: [<MjPri><MinPri=3><CANID>]<74><NN hi><NN lo><No.of events>
Response to request 58 (RQEVN)

75 – 7E reserved.

7F Extended op-code with 2 data byte packets. (EXTC2)
[<MjPri><MinPri=3><CANID>]<7F><Ext_OPC><param1><param2>
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>

80 Request 3-byte DCC Packet (*RDCC3*) Format:
[<MjPri><MinPri=2><CANID>]<80><REP><Byte0>..<Byte2>
<**Dat1(REP)**> is # of repetitions in sending the packet.
<**Dat2>..<Dat4>** 3 bytes of the DCC packet.
Allows a device 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.

81 Reserved

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)

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 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)

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.

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.

86 to 8F reserved

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.

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.

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. 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 or to trigger an event from a ‘combi’ node.

93 Accessory Response Event (ARSPO) 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’ response event. A response event is a reply to a status request (AREQ or ASRQ) without producing an ON or OFF event.

94 Accessory Response Event (ARSPN) 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 or ASRQ) without producing an ON or OFF event.

95 Unlearn an event in learn mode. (EVULN) Format:
[<MjPri><MinPri=3><CANID>]<95><NN hi><NN lo><EN hi>
<EN lo>
Sent by device to remove an event from a node.

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

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.

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 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.

98 Accessory Short ON (ASON) Format:
[<MjPri><MinPri=3><CANID>]<98><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 short event number
<Dat4> is the low byte of the short event number

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

99 Accessory Short OFF (ASOF) Format:
[<MjPri><MinPri=3><CANID>]<99><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 short event number
<Dat4> is the low byte of the short event number

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

9A Accessory Short Request Event (ASRQ) Format:
[<MjPri><MinPri=3><CANID>]<9A><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 short event number
<Dat4> is the low byte of the short event 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 when it is required to know the ‘state’ of the producer without producing an ON or OFF event or to trigger an event from a “combi” node.

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

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)

9D-9E Reserved

9F Extended op-code with 3 data byte packets. (EXTC3)
[<MjPri><MinPri=3><CANID>]<9F><Ext_OPC><param1>

<param2><Param3>
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>

A0 Request 4-byte DCC Packet (*RDCC4*) Format:
[<MjPri><MinPri=2><CANID>]<A0><REP><Byte0>..<Byte3>
<**Dat1(REP)**> is # of repetitions in sending the packet.
<**Dat2>..<Dat5>** 4 bytes of the DCC packet.
Allows a device 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.

A1 Reserved

A2 Write CV in Service mode (*WCVS*) Format:
[<MjPri><MinPri=2><CANID>]<A2><Session><High CV#>
<Low CV#><Mode><Val>
<**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 value to be written
Sent to the command station to write a DCC CV in service mode.

A3 AF Reserved

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

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

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 device to read stored event variables from a node. EV# is the EV index. Reply is (EVANS)

B3 Reserved

B4 Accessory Response Event (ARSP1O) 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 'ON' response event. A response event is a reply to a status request (AREQ or ASRQ) without producing
an ON or OFF event.

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)

B6 Accessory Response Event (ARSP1N) Format:
[<MjPri><MinPri=3><CANID>]<B6><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. A response event is a reply to a status request (AREQ or ASRQ) without producing

an ON or OFF event.

B7 to BE Reserved

BF Extended op-code with 4 data byte packets. (EXTC4)
[<MjPri><MinPri=3><CANID>]<BF><Ext-OPC><Param1 >
<Param2><Param3 ><param4>
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>

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 device 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.

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 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.

C2-CF Reserved

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 Hi byte
<**Dat6**> is additional data Lo byte

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

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 Hi byte
<Dat6> is additional data Lo byte

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

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 device to a node in learn mode to teach it an event. Also teaches it the associated event variables. (EVs). This command is repeated for each EV required.

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 device for the EVs associated with an event (REQEV). For multiple EVs, there will be one response per request.

D4 Accessory Response Event (ARSP2O) 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. A response event is a reply to a status request (AREQ or ASRQ) without producing
an ON or OFF event.

D5 Accessory Response Event (ARSP2N) Format:

[<MjPri><MinPri=3><CANID>]<D5><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

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

D6 to DE Reserved

DF Extended op-code with 5 data bytes. (EXTC5)
[<MjPri><MinPri=3><CANID>]<DF><Ext-OPC><Param1 >
<Param2><Param3 ><param4><param5>
Used if the basic set of 32 OPCs is not enough. Allows an additional 256 OPCs
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>

E0 Request 6-byte DCC Packet (*RDCC6*) Format:
[<MjPri><MinPri=2><CANID>]<E0><REP><Byte0>..<Byte5>
<**Dat1(REP)**> is # of repetitions in sending the packet.
<**Dat2>..<Dat7>** 6 bytes of the DCC packet.
Allows a device 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.

E1 Engine report (*PLOC*) Format:
[<MjPri><MinPri=2><CANID>]<E1><Session><AddrH>
<AddrL><Speed/Dir> | Consist#><Fn1><Fn2>

<**Dat1**> Session for engine assigned by the command station. This session number is used in all referenced to the engine until it is released.
engine.
<**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).

E2 reserved

E3 Accessory node data event (ACDAT) Format:
[<MjPri><MinPri=3><CANID>]<E3><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.

E4 Accessory node data Response (ARDAT) Format:
[<MjPri><MinPri=3><CANID>]<E4><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.

E5-EE reserved

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)

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 3
<Dat6> is additional data byte 2
<Dat7> is additional data byte 1

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

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 3
<Dat6> is additional data byte 2
<Dat7> is additional data byte 1

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

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)

F3 Accessory Response Event (ARSP3O) 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 3
<Dat6> is additional data byte 2
<Dat7> is additional data byte 1

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

F4 Accessory Response Event (ARSP3N) 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 3
<Dat6> is additional data byte 2
<Dat7> is additional data byte 1

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

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 device 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.

F6 to F7 Reserved

F8 to FF Reserved for streaming protocol

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