



OpenLCB Standard	
OpenLCB-CAN Message Network	
13. Oct. 2010	Preliminary

1 Introduction (Informative)

OpenLCB is based on a global exchange of individual messages. This specification describes how messages and/or parts of messages are transported across CAN segments as CAN frames.

2 Intended Use (Informative)

- 5 For moving messages on CAN...

3 References and Context (Normative)

This specification is in the context of the following OpenLCB-CAN Specifications:

- The OpenLCB Frame Transfer Specification, which specifies ...
- The OpenLCB Node Identifier Specification, which specifies ...

- 10 “CAN” refers to the electrical and protocol specifications as defined in ISO 11898-1:2003 and ISO 11898-2:2003 and their successors.

External certification of parts shall be accepted for conformance to these standards. Conformance with a later version of a standard shall be accepted as conformance with the referenced versions.

- 15 For more information on format and presentation, see:
- OpenLCB Common Information Technical Note

4 States

The message network layer in an OpenLCB-CAN network has two states:

- Uninitialized
- 20 • Initialized

Nodes shall start in the Uninitialized state.

A node in the Uninitialized state may transmit an Initialization Complete message. A node in the Uninitialized state shall not transmit any other message type.

A node in the Initialized state may transmit any message type.

25

5 Message Format (Normative)

OpenLCB-CAN messages are sent in single CAN frames using the transfer mechanism and format described in the “OpenLCB-CAN Frame Transfer Standard”. This defines a 15 bit variable field in the frame, the format and contents of which is defined in this section. There are two formats, indicated by the most significant bit of the variable field.

5.1 Global (non-addressed) messages

Global messages do not carry a destination Node ID alias, and therefore are directed at all nodes receiving the particular message.

5.2 Addressed messages

Addressed messages carry the Node ID alias of their destination as the low 12 bits of the variable field.

6 Messages (Normative)

6.1.1 Initialization Complete

Indicates that the node initialization is complete, and once the message is delivered, reachable on the network.

40 Message Type Indicator: 0x
 Destination address present: No
 Simple subset:
 Priority group:
 Content: Full NID of the sending node
 45 CAN frame format:

6.1.2 Verify Node ID

 Message Type Indicator: 0x
 Destination address present: Both yes and no
 Simple subset:
 50 Priority group:
 Content:
 CAN frame format:

6.1.3 Verified Node ID

 Message Type Indicator: 0x

55 Destination address present: No (space reasons originally, but these no longer apply?)
 Simple subset:
 Priority group:
 Content: NID of the sending node – this is sent in full 48 bit format in all wire protocols, even if an alternate form or alias is available elsewhere in the message

60 CAN frame format:

6.1.4 Optional Interaction Rejected

Message Type Indicator: 0x
 Destination address present: Yes
 Simple subset:
 65 Priority group:
 Content:

- Mandatory most recent MTI (2 bytes)
- Mandatory error code (TBD)
- Optional data content (TBD)

 70 CAN frame format:

6.1.5 Terminate Due to Error

Message Type Indicator: 0x
 Destination address present: Yes
 Simple subset:
 75 Priority group:
 Content:

- Mandatory most recent MTI (2 bytes)
- Mandatory error code (TBD)
- Optional data content (TBD)

 80 CAN frame format:

7 Interactions (Normative)

All nodes shall take part in all standard interactions defined in this section.

7.1 Node Initialization

85 Newly functional nodes, once their start-up is complete and they are fully operational, shall send an "Initialization Complete" message and enter Initialized state.

- There is no guarantee that any other node is listening for this. No reply is possible.
- Nodes must not emit any other OpenLCB message before the "Initialization Complete" message.

90 Sending the IC message is required to insure that higher-level tools are notified that they may start to work with the node.

7.2 Duplicate Node ID Detection

95 OpenLCB nodes must have unique node IDs. To detect this across the entire connected OpenLCB, all OpenLCB nodes must indicate an error if they detect an incoming message with a Source Node ID equal to their own. If possible, they should indicate it at the board itself using a light or similar. If possible, they should emit a PCER message with the "Duplicate Source ID detected" global event, which will carry the duplicate event ID in the Source Node ID field.

100 After sending the "Duplicate Source ID detected" global event, the node should not transmit any further messages until reset because this message will be received at the other duplicate-ID node(s), resulting in additional "Duplicate Source ID detected" global events and causing a possible message loop.

To further improve the reliability of this detection, OpenLCB nodes should, but need not, emit a Verified Node ID message every 30 to 90 seconds. As an implementation detail, it's recommended that CAN-attached nodes use their NIDa to pick that interval so that messages don't bunch up.

105 7.3 Node ID Discovery

Upon receipt of a Verify Node ID Number message addressed to it, or an unaddressed Verify Node ID Number message, a node will reply with an unaddressed Verified Node ID Number.

This can be used as check that a specific node is still reachable. When wire protocols compress the originating and/or destination NID, this can be used to obtain the full NID.

110 The standard Verify Node ID Number interaction can be used to get the full 48-bit NID from a node for translation. At power up each node must obtain a alias that is locally unique. Gateways will also have to obtain unique aliases for remote nodes they are proxying on to the segment.

7.4 Error Handling

There are multiple mandatory error-handling scenarios defined.

115 (Need to explain "optional" here)

7.4.1 Reject Addressed Optional Interaction

- Node A receives an addressed message from Node B that carries Node A's NID.
- The MTI indicates the start of an optional interaction.
- If Node A does not want to take part in the optional interaction, it may send an Optional Interaction Rejected message addressed to Node B with the original MTI in the message content. There is no requirement that OIR be sent; the node may silently ignore the incoming message.

The message content also contains an optional reason code and an optional data value. The use of these fields is to be defined.

7.4.2 Reject Unaddressed Optional Interaction

- Node A receives an unaddressed message from Node B.
- The MTI indicates the start of an optional interaction.

If Node A does not want to take part in the optional interaction, it silently drops the message without reply.

7.4.3 Reject Addressed Standard Interaction Due to Error

- Node A is taking part in an addressed interaction with Node B. Either node may be able to send the next message.
- Some error condition prevents Node A from continuing the interaction.
- To terminate the interaction, Node A sends a Terminate Due to Error message to Node B. It then resets its state so as to no longer be taking part in the addressed interaction.

The message content contains the most recent MTI received in this interaction, a mandatory reason code and an optional data value. The use of these fields is to be defined.

Table of Contents

1 Introduction (Informative).....	1
2 Intended Use (Informative).....	1
3 References and Context (Normative).....	1
4 States.....	1
5 Message Format (Normative).....	2
5.1 Global (non-addressed) messages.....	2
5.2 Addressed messages.....	2
6 Messages (Normative).....	2
6.1.1 Initialization Complete.....	2
6.1.2 Verify Node ID.....	2
6.1.3 Verified Node ID.....	2
6.1.4 Optional Interaction Rejected.....	3
6.1.5 Terminate Due to Error.....	3
7 Interactions (Normative).....	3
7.1 Node Initialization.....	4
7.2 Duplicate Node ID Detection.....	4
7.3 Node ID Discovery.....	4
7.4 Error Handling.....	4
7.4.1 Reject Addressed Optional Interaction.....	5
7.4.2 Reject Unaddressed Optional Interaction.....	5
7.4.3 Reject Addressed Standard Interaction Due to Error.....	5