OpenLCB Standard

OpenLCB-CAN
Frame Transfer

Oct 15, 2010	**Preliminary**

1 Introduction (Informative)

This specification describes the mechanism for sending OpenLCB-CAN messages via frames on a CAN segment. It ensures unique headers to prevent CAN arbitration errors and frame loss, message traceability, node addressing and priority management.

For information on format and presentation conventions, see:

- OpenLCB Common Information Technical Note

2 Intended Use (Informative)

This Standard is intended for use whenever OpenLCB nodes are communicating on a single CAN segment. It is not intended to cover OpenLCB communications over other types of communications links.

3 References and Context (Normative)

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

- The OpenLCB-CAN Physical Layer Standard, which specifies the physical layer for transporting OpenLCB-CAN frames
- The OpenLCB Node Identifier Standard, which specifies that every OpenLCB node has a unique 6-byte identifier

“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.

4 Frame Format (Normative)

OpenLCB-CAN frames shall be sent and received using the CAN extended format (29-bit header) only.

OpenLCB-CAN nodes shall operate properly when the CAN segment carries proper standard-format (11-bit header) frames.

OpenLCB-CAN nodes shall not transmit extended-format remote frames (frames with RTR set). Nodes shall operate properly when the CAN segment carries extended-format remote frames.

Nodes shall operate properly when the CAN segment carries overload frames.

The first (most-significant) bit of each OpenLCB-CAN frame is reserved for future use. It must be transmitted as a 1 bit, and ignored upon receipt.

The second (second-most-significant) bit is the Frame Type indicator. A value of 0 indicates a CAN-specific Control Frame. A value of 1 indicates an OpenLCB Message.

The next 15 bits are the Variable Field. The format and contents of the Variable Field depends on Frame Type and are defined in later sections.

The last twelve bits (least significant) are the Source Node ID Alias value of the source (sending) node.

Bit number:	Bit 0	Bit 1	Bits 2-16	Bits 17-28
Content:	Reserved: Send as 1, ignore upon receipt	Frame Type 1: OpenLCB Message 0: CAN Control Frame	Variable Field	Source NID Alias
Mask:	0x1000,0000	0x0800,0000	0x07FF,F000	0x0000,0FFF
Location:	Solo top bit	Top bit of 6th nibble from right	3 bits, then three nibbles	Right-most three nibbles

Table 1: Frame Format

After the header, the frame shall contain from zero to eight bytes of data. Length and content are defined by specific frame and message definitions elsewhere.

5 States

The frame transfer layer of a node has two states:

- Inhibited
- Permitted

Nodes shall start in the Inhibited state.

A node in the Inhibited state may transmit Check ID, Reserve ID, and Alias Map Definition frames. A node in the Inhibited state shall not transmit any other frame type.

Nodes in Permitted state may transmit any frame type.

6 CAN-specific Control Frames and Interactions (Normative)

OpenLCB CAN control frames shall be carried in frames with a 0 in the Frame Type field.

6.1 Control Frame Format

The format and contents of CAN-specific Control frames are defined in the following table:

Name	Variable Field	Data Bytes
Check ID (CID) frame	MMM,NNNN,NNNN,NNNN MMM is the frame sequence number, with valid values from 0x7 through 0x4 NNNN,NNNN,NNNN is the 12-bit Node ID section being checked	None
Reserve ID (RID) frame	0x0700	None
Alias Map Definition (AMD) frame	0x0701	Full Node ID
Alias Mapping Enquiry (AME) frame	0x0702	Full Node ID
Alias Map Reset (AMR) frame	0x0703	None
Reserved; may not be sent, and must be ignored upon receipt	All others	To be defined

Table 2: Control Frame Format

6.2 Interactions

This section describes the interactions which use the above frames.

6.2.1 Reserving a Node ID Alias

To reserve a Node ID alias while in the Inhibited state, a node shall:

- Generate a tenative source Node ID alias value
- Transmit a Check ID frame (CID) with MMM = 0x7, the least significant 12 bits of the full Node ID in the NNNN, NNNN, NNNN remaining twelve bits of the Variable Field, and the tentative source Node ID alias value in the Source NID Alias field.

- Repeat that three more times with MMM = 0x6, x5 and 0x4, respectively, with each frame carrying the next higher 12 bits of the full Node ID value, and the frames carrying the same tentative source Node ID alias value in the Source NID Alias field.
- Wait at least 50 milliseconds
- Transmit a Reserve ID frame (RID) with the tentative source Node ID alias value n the Source NID Alias field.

The node shall restart the process at the top if, before completion of the process, a frame is received that carries the source Node ID alias value being testing in its source Node ID alias field.

The node shall restart the process at the top if, before completion of the process, any error is encountered during frame transmission.

6.2.2 Transition to Permitted State

To transition from the Inhibited state to the Permitted state, a node shall, in order:

- Have or obtain a valid reserved Node ID alias
- Transmit a Alias Map Definition (AMD) frame with the node's reserved Node ID alias and Node ID

6.2.3 Node ID Alias validation

A node in Permitted state receiving a Alias Mapping Enquiry frame shall compare the full Node ID in the CAN data segment to the node's own Node ID. If and only if they match in length and content and the receiving node is in Permitted state, the node shall reply with a Alias Map Definition frame carrying the node's full Node ID in the data segment of the frame.

A node in Permitted state receiving a Alias Mapping Enquiry frame with no data content shall reply with a Alias Map Definition frame carrying the node's full Node ID in the data segment of the frame.

A node in Inhibited state shall not reply to a Alias Mapping Enquiry frame.

6.2.4 Transition to Inhibited State

To transition from the Permitted state to the Inhibited state, a node shall successfully transmit an Alias Map Reset frame with the node's reserved Node ID alias and Node ID.

6.2.5 Node ID Alias Collision Handling

A node shall compare the source Node ID alias in each received frame against all reserved Node ID aliases it currently holds. In case of a match, the receiving node shall

If the frame is a Check ID (CID) frame, send a Reserve ID (RID) frame in response.

If the frame is a any other type, the node is in Permitted state, and the received source Node ID alias is the current Node ID alias of this node, this node shall immediately transition to Inhibited state and release the current Node ID alias.

If the frame is another other type, and the node is not in Permitted state or the received source Node ID alias is not the current Node ID alias of this node, the node shall immediately release the matching node ID alias.

6.2.6 Duplicate Node ID Handling

Each node shall compare the node ID in each received Alias Map Definition frame with its own Node ID. Should they match, in addition to any other actions that may be required by the incoming message, the node

- may, but is not required to, signal the user that duplicate Node ID values exist using a lamp or other directly-visible indicator
- if in Permitted state, shall emit a PCER message with the reserved Event ID “Duplicate Node ID Detected”.

6.3 Node ID Alias Generation

Alias values shall not be zero.

The first alias values generated by nodes of the same type with node ID values within 255 of each other shall not be identical.

When two different nodes generate the same alias value at some point in their sequence, there shall be more than a 99% probability that the next alias values generated by the two nodes are different.

A node may, but need not, save the current alias generation state so that it restarts the sequence at the same point, hence the same alias value, after a reset or power cycle.

7 OpenLCB Message Frame Format

OpenLCB messages shall be carried in frames with a 1 in the Frame Type field. They shall contain message type information and/or address information in the 15-bit variable field, and zero to eight CAN data bytes.

For OpenLCB messages, the variable field shall be used in one of two forms:

- Unaddressed messages – messages that don't have a destination address – shall be formatted:

Variable Field Bit 0 **Header Bit 2** **0x0400,0000**	**Variable Field Bits 1-14** **Header Bits 3-16** **OpenLCB Variable Header Content** **0x03FF,F000**
0	OpenLCB message information

Table 3: Unaddressed Message Format

- Addressed messages – messages that have a specific destination address – shall have the address alias in the low 12 bits of the variable field. Two upper bits can carry part of the OpenLCB message.

Variable Field Bit 0 **Header Bit 2** **0x0400,0000**	**Variable Field Bits 1-2** **Header Bits 3-4** **OpenLCB Variable Header Content** **0x0300,0000**	**Variable Field Bits 3-14** **Header Bits 5-16** **OpenLCB Variable Header Content** **0x00FF,F000**
1	OpenLCB message information	Destination Node ID Alias

Table 4: Addressed message format

Table of Contents

1 Introduction (Informative).......... 1
2 Intended Use (Informative).......... 1
3 References and Context (Normative).......... 1
4 Frame Format (Normative).......... 1
5 States.......... 2
6 CAN-specific Control Frames and Interactions (Normative).......... 3
6.1 Control Frame Format.......... 3
6.2 Interactions.......... 3
6.2.1 Reserving a Node ID Alias.......... 3
6.2.2 Transition to Permitted State.......... 4
6.2.3 Node ID Alias validation.......... 4
6.2.4 Transition to Inhibited State.......... 4
6.2.5 Node ID Alias Collision Handling.......... 4
6.2.6 Duplicate Node ID Handling.......... 5
6.3 Node ID Alias Generation.......... 5
7 OpenLCB Message Frame Format.......... 5