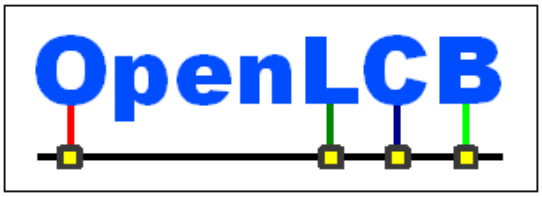
NMRA Standard	
Layout Command Control® (LCC) **Datagram Transport**	
Feb 17, 2015	**9.7.3.2-S**

Adopted as a NMRA Standard

The OpenLCB Standard document appended to this cover sheet has been formally adopted as a NMRA Standard by the NMRA Board of Directors on the date shown in the *Adopted* column in the *Version History* table below.

Version History

Date	Adopted	Summary of Changes
Feb 17, 2015		Initial version



OpenLCB Standard	
Datagram Transport	
Feb 17, 2015	Adopted

1 Introduction (Informative)

This standard defines the protocol for transporting OpenLCB datagrams.

2 Intended Use (Informative)

The datagram transport protocol is intended to efficiently transfer small amounts (0-72 bytes) of data reliably between two OpenLCB nodes. It allows for management of overlapping independent transmissions.

The datagram transport protocol relies on the underlying OpenLCB message transport protocol for reliable sequenced communications.

This document describes the required message formats for datagram transport. Section 4 gives an overview of the message types with an abstract numeric description intended as a normative guide to the construction of concrete message types over specific physical transport media. Section 7 describes, in concrete detail, the implementation of the datagram transport message formats for the specific physical transport media that have been adopted as normative standards.

3 References and Context (Normative)

This is in the context of the following OpenLCB Standards:

- The OpenLCB Message Network Standard, which defines the basic messages and how they interact. Higher-level protocols are based on this message network, but are defined elsewhere. The Message Network Standard defines the global error codes which are referenced here.
- The OpenLCB CAN Frame Transfer Standard, which specifies the use and format of CAN frames for OpenLCB communications.

4 Message Formats (Normative)

In the following, the “Common MTI” column specifies the the MTI value to be used when communicating in OpenLCB common format. The Common MTI is an abstract numeric description intended as a normative guide to the construction of concrete message formats over specific physical transport media.

4.1 Datagram Content

Name	Simple Node	Dest ID	Event ID	Common MTI	Data Content
Datagram Content	N	Y	N	0x1C48	0-72 bytes

The first byte of the data content defines the datagram type and is designated the Datagram Content ID. The values for that byte are documented in the Standard for the protocol that defines the type.

4.2 Datagram Received OK

Name	Simple Node	Dest ID	Event ID	Common MTI	Data Content
Datagram Received OK	N	Y	N	0x0A28	Flags (1 byte)

The flag bits are defined as:

- MSB 0x80 – Reply Pending – Use is defined by higher-level protocols.
- Low four bits 0x0F – Timeout Value – Zero indicates no timeout value. A value N of 0x01 through 0x0F indicates that the pending reply will be transmitted before 2^N seconds have elapsed; if not, an error has occurred.
- All others are reserved, shall be sent as zero and ignored upon receipt.

Datagram Received OK messages without a Flags byte shall be treated as if they contained a byte with a zero value.

4.3 Datagram Rejected

Name	Simple Node	Dest ID	Event ID	Common MTI	Data Content
Datagram Rejected	N	Y	N	0x0A48	Error Code (2 bytes)

Nodes shall accept and process Datagram Rejected messages that do not contain the full error code. Missing error code bits are to be interpreted as zero.

4.3.1 Error Codes

The Error Code field contains 16 bits and shall use the codes defined in the OpenLCB Message Network Standard.

Nodes may, but are not required to, use 0-3 of the error code field to define specific error codes in concert with the flag bits defined above.

5 States (Normative)

The common OpenLCB datagram protocol has no formal states.

6 Interactions (Normative)

A node that receives a valid Datagram Content message shall send either a Datagram Received OK or Datagram Rejected message in reply. A node that receives a Datagram Content message that does not comply with this Standard may, but is not required to, reply with a Datagram Rejected message.

6.1 Normal Transmission

Normal transmission consists of the transmitting node sending a Datagram Content message to the receiving node, followed by the receiving node sending a Datagram Received OK message to the transmitting node.

6.2 Rejected Transmission

After the transmitting node sends a Datagram Content message to the receiving node, the receiving node may send a Datagram Rejected message to the transmitting node.

If a receiving node receives a 2nd Datagram Content message before sending a reply to the the 1st Datagram Content message, it may, but is not required to, reject the 2nd Datagram by sending a Datagram Rejected message with the Transport error and Resend OK error flag bits set.

Upon receipt of a Datagram Rejected message with the Resend OK bit set, the original transmitting node may resend the same Datagram Content message, or may abandon the transmission attempt.

Upon receipt of a Datagram Rejected message with the Resend OK bit clear, the original transmitting node shall abandon the transmission attempt and not resend the original Datagram Content message.

7 Adaptation to CAN Transport (Normative)

This section describes the CAN implementation of the datagram transport message formats.

7.1 CAN Message Formats

The OpenLCB CAN Frame Transport Standard and OpenLCB Message Network Standard define how OpenLCB messages are carried across CAN networks. Following those standards, the Datagram Transport messages used on CAN are as defined in the following table.

Name	CAN-MTI	Can Header	Data Content
Datagram Content	0xddd[1]	0x1Add,dsss[2] – Single[3] 0x1Bdd,dsss – First 0x1Cdd,dsss – Middle 0x1Ddd,dsss – Last	0–8 bytes
Datagram Received OK	0xA28	0x19A2,8sss	0xfddd[4], Flags
Datagram Rejected	0xA48	0x19A4,8sss	0xfddd, Error Code

7.2 CAN States

A node implementing the OpenLCB-CAN datagram protocol shall maintain a Datagram-started state for each datagram that it is receiving as a sequence of frames. If the node receives multiple overlapping datagrams, the states shall be independent.

7.3 CAN Interactions

7.3.1 Normal Transmission

Normal transmission of a datagram over CAN consists of the transmitting node sending the Datagram Content message using one of two sequences of Datagram frames:

- One Datagram Content Single Frame
- One Datagram Content First Frame, followed by zero or more Datagram Content Middle Frame, followed by one Datagram Content Last Frame

A node shall not interleave transmission of frames from more than one datagram to the same target node. A node shall not transmit frames with lower CAN priority between the frames making up a datagram. A node may, but is not required to, transmit frames with higher CAN priority between the frames making up a datagram.

A receiving node receiving either of the above sequences shall send either a Datagram Received OK or Datagram Rejected message in reply.

7.3.2 Rejected Transmission

If a receiving node receives a sequence of Datagram frames other than one of

- One Datagram Content Single Frame

[1] ddd – The 12-bit destination alias field

[2] sss – The 12-bit source alias field

[3] Because CAN frames are limited to 8 bytes, datagrams larger than 8 bytes must be broken up among multiple messages. Thus, four distinct message types are defined to aid in flow control.

[4] fddd — First two bytes of the data-part, representing the 4-bit flag field and 12-bit destination Alias. See the OpenLCB-CAN Frame Transport Standard.

- One Datagram Content First Frame, followed by zero or more Datagram Content Middle Frame, followed by one Datagram Content Last Frame[5]

100 the receiving node shall send a Datagram Rejected message with the Transport error and Resend OK bits set.

[5]The total payload bytes sent, including any First, Middle, and Last Frames, cannot exceed 72 bytes.

Table of Contents