OpenLCB test plan for the Simple Node Information Protocol Standard

The OpenLCB Group

February 4, 2024

1 Introduction

This note documents the procedure for testing an OpenLCB implementation against the Simple Node Information Protocol Standard.

The tests are traceable to specific sections of the Standard.

The testing assumes that the Device Under Test (DUT) is being exercised by other nodes on the message network, e.g. is responding to enquiries from other parts of the message network.

2 Required Equipment

See the separate "Installing the OpenLCB Test Software" document for initial installation and set up of the test program.

If a direct CAN connection will be used, a supported USB-CAN adapter [1] is required. Connect the adapter to the DUT using a single UTP cable and connect two CAN terminators.

Provide power to the DUT using its recommended method.

3 Set Up

The following steps need to be done once to configure the test program:.

1. Start the test configuration program.
2. Select "Set Up DUT".

[1] See "Installing the OpenLCB Test Software"

3. Get the Node ID from the DUT[2]
4. Enter that Node ID into the program.
5. Configure the test program for the USB-CAN adapter's device address or the TCP hostname and port.
6. Quit the test program and reply "Y" to "Save configuration?" when prompted.

The following steps need to be done at the start of each testing session.

1. Check that the DUT is ready for operation.
2. Start the test program.

4 Simple Node Information Protocol Procedure

Select "SNIP testing" in the test program, then select each section below in turn. Follow the prompts for when to reset/restart the node and when to check outputs against the node documentation.

4.1 SNIP reply testing

This section tests the format of the reply message in Sections 4.2 and 5.1 of the Standard.

It does this by issuing a Simple Node Information Request message, accumulating the reply(s), and then checking:

1. The message indicates its source is the DUT.
2. The message indicates its destination is the testing node.
3. The version byte at the start of the first section is either 1 or 4.
4. The version byte at the start of the second section is either 1 or 2.
5. There are exactly six zero bytes.
6. Each of the six defined strings is no longer than its defined maximum length.
7. There are no data byte(s) after the sixth zero byte.

[2]Where do we require this to be marked on a node?