Checking the OpenLCB Configuration Description Information Standard

The OpenLCB Group

February 10, 2024

1 Introduction

This note documents the procedure for checking an OpenLCB implementation against the Configuration Description Information Standard.

The checks are traceable to specific sections of the Standard.

The checking assumes that the Device Being Checked (DBC) is being exercised by other nodes on the message network, e.g. is responding to enquiries from other parts of the message network.

1.1 Required Equipment

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

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

Provide power to the DBC using its recommended method.

2 Set Up

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

1. Start the checker configuration program.
2. Select "Set Up".

[1] See "Installing the OpenLCB Checker Software"

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

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

1. Check that the DBC is ready for operation.
2. Start the checker program.

3 Configuration Description Information Procedure

Select "CDI checking" in the program, then select each section below in turn.

A node which does not self-identify in PIP that it supports Configuration Description Information will be deemed to have passed these checks. [3]

This plan assumes that the Datagram Transport Protocol and the Memory Configuration Protocol have been separately checked. It uses those, but does not do any detailed checking of them.

3.1 Validation checking

This section checks the content of the CDI to make sure that it is valid XML. It reads the information from the 0xFF memory space, then validates it against the 1.3 XML Schema which is stored in a local "schema.xsd" file.

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

[3] Using the -p option or setting the checkpip default value False will skip this check.