

Checking the OpenLCB Train Control Protocol Standard

July 13, 2024

1 Introduction

This note documents the procedure for checking an OpenLCB implementation against the Train Control Protocol 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.

2 Train Control Protocol Procedure

A node which does not self-identify in PIP that it supports the Train Control Protocol should be considered to have passed these checks.

Note that this commands the speed and functions of a locomotive node. Although the commanded speed is quite low and only for a short time, the checks should be run in a way that the associated physical locomotive does not run away.

2.1 Defined Event ID checking

This section checks that the node supports the isTrain event defined in section 4.1 and 6.4 of the Train Control Protocol Standard.

It does this by issuing an Identify Events to the node, and then checking for a Producer Identified reply carrying the isTrain Event ID.

2.2 Check set and query speeds

This checks the ability to set and query the speed and direction of the train. Note that it checks that forward/reverse is independent of the speed setting, particularly at zero.

1. Set the speed and direction to 0.75 reverse.
2. Query the speed and direction. Check for 0.75 reverse.
3. Set the speed and direction to 0 reverse.
4. Query the speed and direction. Check for 0 reverse.
5. Set the speed and direction to 0.75 forward.
6. Query the speed and direction. Check for 0.75 forward.
7. Set the speed and direction to 0 forward.
8. Query the speed and direction. Check for 0 forward.

In any case, even if the earlier checks failed, end by setting the speed and direction to 0 forward.

2.3 Check set and query of functions

This section assumes that F0 is available on the train. The FDI is not used to confirm that in this section.

1. Set F0 to on.
2. Query F0 and check for an "on" response.
3. Set F0 to off.
4. Query F0 and check for an "off" response.

In any case, even if the earlier checks failed, end by setting F0 to "off".

2.4 Check Emergency Stop

1. Set the speed and direction to 0.75 reverse.
2. Query the speed and direction. Check for 0.75 reverse.
3. Send an emergency stop to the train.
4. Query the speed and direction. Check for 0 reverse.
5. Set the speed and direction to 0.75 forward.
6. Query the speed and direction. Check for 0.75 forward.
7. Set the speed and direction to 0 forward.

In any case, even if the earlier checks failed, end by setting the speed and direction to 0 forward.

2.5 Check Global Emergency Stop

2.6 Check Global Emergency Off

2.7 Check memory spaces

This section checks the existence and properties of memory spaces defined in section 7 of the Train Control Protocol Standard.

1

2.8 Checking function memory space connection

This section assumes that F0 is available on the train. The FDI is not used to confirm that in this section.

1. Set function 0 "on"
2. Check byte 0 in the 0xF9 memory space for a non-zero value.
3. Write byte 0 in the 0xF9 memory space to zero.
4. Check that function 0 is "off".

2.9 Check Controller Configuration command and response

2.10 Check Listener Configuration command and response

2.11 Check Train Control Management command and response

¹This does not check the information defined by the Function Definition Information Standard.