OpenLCB Working Note

DCC Detection Protocol

Sep 8, 2023 | Preliminary

1 Introduction

A Working Note is an intermediate step in the documentation process. It gathers together the content from various informal development documents, discussions, etc into a single place. One or more Working Notes form the basic for the next step, which is one or more Standard/TechNote pairs.

This working note describes a OpenLCB protocol for a detector to report its detection of a train, including the DCC address of an associated DCC decoder.

“Detector” refers to a fixed-location electronic device that can interact with a DCC decoder to ascertain the address of the DCC decoder while the decoder is in a specific area or at a specific point. Possible implementations include but are not limited to RailCom detectors, Digitrax detectors, RFID and QR readers.

1.1 Served Use Cases

A train enters a current-detected block. A detector recognizes this, successfully queries the decoder for its address, and reports that information on the OpenLCB bus. That information results in a signal controller sending a speed command such as Slow or Stop to that particular detector.

A train crosses an RFID reader which determines the train’s DCC address and reports that address on the OpenLCB bus.

A train exits a current-detected block. A detector recognizes this, and reports that interaction on the OpenLCB bus.

A train enters a current detected block. A detector recognizes this, and attempts unsuccessfully to query the decoder for its address. The detector emits a report without decoder address information.

1.2 Unserved Use Cases

Being able to report information on a native OpenLCB (non-DCC) train is an unserved use case. This protocol doesn’t provide a way to report the Unique ID of a fully OpenLCB train. All devices being reported need to have a DCC address.

Having a monitor tool that can show all detection information without configuration is a unserved use case. The Unique IDs of the detectors, hence the upper bytes of Event IDs to be consumed, are determined by a pre-configuration process in the consumer. They can’t be determined automatically.

2 Specified Sections

This is the usual section organization for a Technical Note, to accumulate the Standard and Technical Note content in its eventual order.

2.1 Introduction

Note that this section of the Standard is informative, not normative.

2.2 Intended Use

Note that this section of the Standard is informative, not normative.

This protocol is intended to provide a standard way to transfer train location and DCC decoder identification information on OpenLCB DCC layouts.

2.3 Reference and Context

This protocol builds on the Event Transport Protocol as defined by that Standard and Technical Note.

2.4 Message Formats

Detection events are produced via PCER messages with an Event ID which contains

1) 6 byte unique ID of the detector,
2) Two bits to represent entry/exit and forward/reverse. These are coded:

0b00	unoccupied, ie exit
0b01	occupied, forward, ie entry
0b10	occupied, reverse, ie entry
0b11	occupied, direction unknown, ie entry

3) 14 bits of DCC address
   A) If the high bits are 0x38 this represents a DCC short address
   B) If the high bits are 0x39 this represents a DCC consist address
   C) An address of 0x3800 (short address 0) represents “unknown”.
   D) Otherwise it’s a DCC long address

The encoding of DCC addresses into 14 bits follows the S-9.2.1.1 and RCN-218 standards.

The 0x3800 value is used for unknown instead of 0x0000 because 0x0000 is long address zero, which is a valid (though uncommon) long address.

2.5 States

This protocol has no states of its own.

An “exit Event ID can be sent at most once after a corresponding entry Event ID” state requirement was discussed but not accepted. Note that some hardware detectors determine only that a decoder has left the monitored section, not its address, so those scanners will be producing “NNN entry” followed by “(none) exit”.

2.6 Interactions

A detector sends a PCER with an Event ID as described above when it detects entry of a decoder into the monitored location. It may also send a PCER when the exit of the decoder from the monitored location is detected.

A detector, acting as a producer, must reply to Verify Events and Identify Producers messages with the appropriate event(s) or range.

Consumers can consume the entire 16-bit range of the lower EventID bytes. Or they can consume just the specific EventIDs that correspond to the particular decoder(s) that they’re tracking. In either case, they must respond to Verify Events and Identify Consumers messages with the appropriate event(s) or range.

There is no PIP bit allocated to this protocol.

3 Background Information

This protocol has been implemented in several products. It is known to work for its served use cases.

Table of Contents