OpenLCB Standard	
Simple Time Protocol	
Sep 11, 2018	Preliminary

1 Introduction (Informative)

“Fast clock” is a model railroading concept, where the layout is modeling a particular time, which can then be used for various purposes, such as running trains according to a timetable, or displaying lighting effects based on time of day. A layout control bus can do a number of useful things with the modeled time information:

- Connect a number of clock displays to keep them synchronized.
- Provide time displays on plug-in devices, e.g. throttles.
- Provide cueing for time-based occurrences, such as lights turning on and off at specific modeled times.

Generally, existing fast clock systems have one unit that produces time information, here called a clock generator, and one or more units that consume it. It is typically sufficient to report minutes, not seconds or finer time divisions. Some existing fast clock systems track a day/date in addition to time.

Fast clocks run at various rates, and can be controlled by the user either at the clock generator or from other locations. Some fast clock systems broadcast run/stop and rate information, which can also be useful when interpolating within a fast-minute or between time broadcasting messages.

The OpenLCB protocol presented in this document broadcasts time information by producing events with a specific structure of IDs, corresponding to specific times within the day, for example “08:10”. This allows consumers to be taught to react to time-of-day. The year and date are handled separately for those installations that choose to use it.

2 Intended Use (Informative)

The primary use of this information is to display time on clock faces around the layout.

Since remote control of the fast clock is desired, a protocol for setting up the time producer using produced and consumed events is defined. This makes it possible for throttles and other nodes with user interface to have a general fast-clock-control capability built in.

In addition, simple nodes can use specific EventIDs to trigger their actions at specific times. For example, lights in buildings in a model town can be sequenced to come on at various times by configuring consumers in a node to react to time events by changing output lines. Most importantly, in this use case, the consuming node does not need to be aware that the event is related to time.

3 References and Context (Normative)

This specification is in the context of the following OpenLCB-CAN Standards:

- OpenLCB Event Transport Standard, which defines messages for transporting Event IDs and identifying producers and consumers.
- The OpenLCB Event Identifiers Standard, which defines the format and content of Event IDs including the class of Well-Known Event IDs and Automatically-Routed Event IDs.
- OpenLCB Unique Identifiers Standard, which defines the allocation of OpenLCB 48-bit unique identifiers.

For more information on format and presentation, see:

- OpenLCB Common Information Technical Note

4 Message Formats (Normative)

This Standard defines a number of Event IDs.

The upper six bytes of the event IDs defined in the following subsections shall be one of the following:

- 01.01.00.00.01.00 – referred to as “Default Fast Clock”
- 01.01.00.00.01.01 – referred to as “Default Real-time Clock”
- 01.01.00.00.01.02 – referred to as “Alternate Clock 1”
- 01.01.00.00.01.03 – referred to as “Alternate Clock 2”
- a valid unique ID under the control of the manufacturer of the clock generator node
- a valid unique ID under the control of the person or organization configuring the clock generator node

The upper six bytes are referred to as the “Specific Upper Part” in the subsections below.

4.1 Report Time Event ID

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Specific Upper Part						Hours 0x00-0x17	Minutes 0-59

The upper nibble of byte 6 is only 0 or 1, which can be used to distinguish this format.

These event IDs represent time in 24-hour format, i.e. byte 6 values of 0–11 represent a.m. time, and values of 12–23 represent p.m. time.

4.2 Report Date Event ID

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Specific Upper Part						0x20+Month 0x21-0x2C	Day 1-31

The upper nibble of byte 6 is 2, which can be used to distinguish this format.

4.3 Report Year Event ID

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Specific Upper Part						0x3000+Year 0x3000-0x3FFF	

The upper nibble of byte 6 is 3, which can be used to distinguish this format.

The lower twelve bits are the year, 0AD to 4095AD.

4.4 Report Rate Event ID

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Specific Upper Part						0x4000+Rate	

The upper nibble of byte 6 is 4, which can be used to distinguish this format.

Rate is a 12 bit signed fixed point rrrrrrrrrr.rr, -512.00, -511.75, …, -1.00, …, -0.25, 0.0, 0.25, 0.50, ..., 511.75

4.5 Set Time Event ID

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Specific Upper Part						Hours 0x80-0x97	Minutes 0-59

The upper nibble of byte 6 is only 8 or 9, which can be used to distinguish this format.

These event IDs represent time in 24-hour format, i.e. byte 6 values of 0x80+(0 to 11) represent a.m. time, and values of 0x80+(12 to 23) represent p.m. time.

4.6 Set Date Event ID

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Specific Upper Part						0xA0+Month 0xA1-0xAC	Day 1-31

The upper nibble of byte 6 is A, which can be used to distinguish this format.

4.7 Set Year Event ID

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Specific Upper Part						0xB000+Year 0xB000-0xBFFF	

The upper nibble of byte 6 is B, which can be used to distinguish this format.

The lower twelve bits are the year, 0AD to 4095AD.

4.8 Set Rate Event ID

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Specific Upper Part						0xC000+Rate	

The upper nibble of byte 6 is C, which can be used to distinguish this format.

4.9 Query Event ID

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Specific Upper Part						0xF000	

4.10 Stop/Start Event ID

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Specific Upper Part						Stop 0xF001 Start 0xF002	

4.11 Date Rollover Event ID

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Specific Upper Part						0xF003	

4.12 Undefined/Reserved Event IDs

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Specific Upper Part						All byte 6 and byte 7 values not defined above	

All undefined byte 6 and byte 7 values are reserved for future use. They shall not be sent. If received, they shall be ignored.

5 States (Normative)

Each clock has an independent current time, independent running/stopped state and an independent rate.

When the clock is in stopped state, it’s time is not changing.

When the clock is in running state, it’s internal time is advancing <rate> times as fast as normal time.

6 Interactions (Normative)

6.1 Startup

When a clock generator starts to operate, it shall send a Producer Range Identified and a Consumer Range Identified message covering the whole 16-bit range of the clock’s 6-byte “Specific Upper Part”. Clock consumers shall send either a Consumer Range Identified message covering the whole 16-bit range of the clock’s 6-byte “Specific Upper Part” or individual Consumer Identified messages for each individual time event consumed.

When a clock generator starts to operate, it shall also send the sequence specified in section 6.3 below as if a Query Event had been received.

6.2 Clock Report

While running, a clock generator shall send a Report Time Event <u>no more frequently than</u> once per real world minute and <u>no less frequently than</u> once per real world hour with the following exceptions:

1. The interaction described in section 6.3 below requires it.
2. The clock generator has previously received a Consumer Identified message for a specific Report Time Event, in which case it will always generate this event when appropriate.

It is the responsibility of clock consumers to keep track of intermediate time between Report Time Events. The standard purposely does not define how a clock consumer internally implements time.

While running, a clock generator shall send a Date Rollover Event immediately prior to a rollover in the progression of time in either direction through hour 0 and minute 0 and three real seconds later send Report Year and Report Date Events.

6.3 Clock Query

If a clock generator receives a Query Event, it shall respond with the following sequence of messages with the last valid events of the given group, in this specific order:

1. Producer Identified Valid for Start or Stop Event ID,
2. Producer Identified Valid for Report Rate event,
3. Producer Identified Valid for Report Year event,
4. Producer Identified Valid for Report Date event,
5. Producer Identified Valid for Report Time event,
6. Producer/Consumer Event Report for Report Time event for next minute, when it becomes valid.

6.4 Clock Set

Any node may configure a clock generator.

If a Set Rate Event is received by the clock generator, the clock generator's rate may be, but is not required to be, set to the rate embedded in the event. If the clock producer does not support the requested rate, it shall move to the closest non-zero supported rate. The rate may be set while the clock is running or stopped.

If a Start or Stop, Set Rate, Set Year, Set Date, or Set Time Event is received, the clock generator shall make the change effective immediately, and produce the effective Report Rate, Report Year, Report Date, or Report Time Event. Three seconds after the last Start or Stop, Set Rate, Set Year, Set Date, and/or Set Time has been received, the sequence of messages defined in section 6.3 above shall be sent.

Table of Contents

1 Introduction (Informative)....1
2 Intended Use (Informative)....1
3 References and Context (Normative)....2
4 Message Formats (Normative)....2
4.1 Report Time Event ID....2
4.2 Report Date Event ID....3
4.3 Report Year Event ID....3
4.4 Report Rate Event ID....3
4.5 Set Time Event ID....3
4.6 Set Date Event ID....3
4.7 Set Year Event ID....4
4.8 Set Rate Event ID....4
4.9 Query Event ID....4
4.10 Stop/Start Event ID....4
4.11 Date Rollover Event ID....4
4.12 Undefined/Reserved Event IDs....5
5 States (Normative)....5
6 Interactions (Normative)....5
6.1 Startup....5
6.2 Clock Report....5
6.3 Clock Query....6
6.4 Clock Set....6

