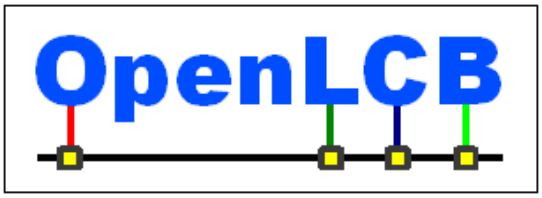

OpenLCB Technical Note	
OpenLCB Unique Identifiers	
Oct 24, 2010	Preliminary

1 Introduction

This Technical Note contains informative discussion and background for the corresponding "OpenLCB Unique Identifiers Specification". This explanation is not normative in any way.

Originally, OpenLCB was defined in terms of 48-bit unique "node IDs". Other uses were found for these IDs that didn't associate them with a specific node, which demonstrated the need for unique IDs beyond just node IDs. Therefore, the Standard and this Technical Note are generally written in terms of unique IDs, which includes use as node IDs.

2 Annotations to the Specification

This section provides background information on corresponding sections of the Specification document. It's expected that two documents will be read together.

2.1 Introduction

2.2 Intended Use

The "globally unique" requirement only refers to the universe of connected nodes; nodes that never need to communicate with each other need not have separate Node IDs. In general, however, nodes can move: They can be sold or loaned for use on another layout, nodes on modular layouts can be connected to other arbitrary modules, and few assumptions can be made. It's best if nodes are given a completely unique ID when manufactured, so there's no need to ever detect and resolve a conflict.

2.3 References and Context

2.4 Content and Format

The Standard doesn't require any particular human-readable format, but hex-pairs with separators are strongly suggested, e.g. 01.AB.34.01.CD.E3; decimal pairs could also be used, but in that case it's important to provide a way to know which is use.

There are many methods to store a unique identifier, and that's not constrained by this Standard. It could be stored in a non-volatile memory, as jumpers on a board, etc.

2.5 Allocation

Unique IDs are assigned via a delegation process. At the highest level, ranges are assigned to people and organizations, within which they are responsible for assigning unique identifiers to separate devices. These ranges can be subdivided and delegated further, as needed. Additional

ranges can then be requested, which will be recorded here or eventually in some online system accessible to anybody.

One of the reasons for having a long, 48-bit unique identifier space is to make it easier to use a delegation system like this. Because there are a lot of possible unique identifiers, large ranges can be delegated to groups without having to ensure that the range be extensively used. For example, most NMRA members will not design their own OpenLCB nodes and need to assign node IDs, but assigning a range to every member makes it easy for those who want to, at a cost of only 0.000016 of the total unique identifier space.

In these delegated assignments, the lower order byte(s) are self-assigned. The value of zero should be reserved, indicating that a number within the range hasn't been assigned.

The high byte of each range is different for each type of assignment, making it easy to determine the allocation pattern in use for a particular unique identifier.

2.5.1 Unique identifiers assigned by manufacturers

As an initial simplification, and to encourage manufacturers to adopt OpenLCB, a group of 2^{24} identifiers will be assigned to each manufacturer with an NMRA manufacturer ID number. Manufacturers can assign any unique identifier from within their range to a board so long as each identifier is only assigned to at most one node. There's no requirement that they be assigned sequentially, or by type, or in any other order.

The NMRA has defined ID number 238 for future expansion with in the existing 8-bit ID space, but has not determined how to use it. When 238 is finally used to identify a new set of manufacturers, those manufacturers can be assigned OpenLCB ranges with a different set of high-order bits.

Software manufacturers who use software keys can generate the low-order 24 bits from some unique part of the software key. Hardware manufacturers need to ensure that resetting a device to “factory defaults” doesn't lose the self-assigned part of the Node ID, e.g. the serial number.

2.5.2 Unique identifiers assigned by members of organized groups

MERG kit builders and others would like to assign their own identifiers without going through a complicated process. To make this possible without any interaction with anybody, these people are assigned identifier ranges that involve their member number in the organization. A member may assign any identifier from this range to the node(s) they produce, provided that each identifier is assigned to at most one node. A range of 255 identifiers per member is sufficient for hobby usage; after building that many, the hobbyist can get another, larger assignment. It's also convenient to give hobbyists a byte as their range.

Each organization is assigned a unique high-order two bytes. The organization member number is given 24 bits. Byte 2 of the unique identifier advances by 4 between groups (NMRA is 0x00, MERG is 0x04, etc) to allow a little more headroom on group membership numbers; this space can be reclaimed later if needed.

Other groups have defined mechanisms to ensure that their node numbers or equivalent constructs are uniquely assigned. They may have e.g. non-technical reasons for wishing to use those same mechanisms to assign OpenLCB unique identifiers. Ranges of OpenLCB unique identifier ranges can

be assigned to these, so that when people then use the group's mechanism to select a value within that range, the result will be a properly unique OpenLCB unique identifier.

The first example of this is MERG CBUS developers. MERG CBUS has defined a “no cost” way of identifying unique 16-bit Node Number (NN) for CBUS use, perhaps with an optional 16-bit Layout Number (LN). People who wish to use this mechanism to allocate unique OpenLCB Node ID identifiers can, without having to consult anybody, generate an OpenLCB Node ID from the unique CBUS number(s) as described in the Standard.

If the user is involved in determining the unique identifier for a node (the Node ID), e.g. by setting switches, the possibility of duplicated node IDs must be considered. Users make mistakes. To reduce user frustration, the node should provide a user-visible way to indicating a duplicate has been seen, and should fully implement the relevant wire-protocol-specific methods for detecting duplicate node IDs.

2.5.3 Unique identifiers assigned by software at run-time

Programs that act as one or more OpenLCB nodes need to associate unique identifiers with them. For licensed software, where a unique key can be associated with each instance of the program, this is easy: Use the manufacturer space defined above, and generate the lower bits of specific ID from the license key.

Free, open and unlicensed software can't use a license-key-based method. Unfortunately, the 48-bit address space is too small to use the IP-address-plus-signature GIDs that would otherwise make this a simple problem, or the even larger MAC-address-plus-signature GIDs.

If the computer has a global IP address (not part of one of the four non-global IP ranges), that can be used as described in this section. If the program implements multiple nodes, or wants to handle multiple invocations of the program, it needs to have a method to provide a unique value for the lowest bytes.

Non-global IP addresses (called out as specific ranges in the Standard) will require another algorithm. It may be as simple as requesting that the user get an identifier from one of the above spaces and provide it. If the software determines it has an internet connection, it could also request a unique allocation from a central source. Future work may support a unique identifier made from e.g. use of time in nanoseconds.

2.5.4 Globally defined unique identifiers

In addition to use as node identifiers (node IDs), OpenLCB unique identifiers are used to ensure uniqueness of specific global event identifiers and for other purposes. These numbers must be allocated so that they are kept unique. The identifiers specified in this section are of that type.

Note that the actual use of these identifiers is actually specified elsewhere. In some cases, the protocols are still being developed, and the entry here is just reserving a range for a specific future use.

2.5.5 Unique identifiers assigned for use with locomotive control systems

Locomotive control was initially beyond the scope of OpenLCB development, but later work stated to define OpenLCB methods for working with existing locomotive control systems. This section specifies ranges of unique identifiers that are reserved for the purpose of interfacing with locomotive control.

The details of how there are to be used are specified elsewhere, but sufficient range has been reserved to allow providing decoders with OpenLCB unique identifiers, etc.

2.5.6 Unique identifier ranges assigned by request

110 Users can request blocks of identifiers of various sizes. The small (256) and medium (65536) blocks are not scarce resources. The 24-bit blocks are slightly scarcer, but there are still almost 2^{16} of them available by using additional values for byte 2. Requests for these blocks should be routinely granted once the requestor has identified himself.

2.5.7 Reserved unique identifiers

115 For error detection and future expansion, we permanently reserve all identifiers that start with either a 0x00 or 0xFF value. OpenLCB implementations can treat those as errors without causing any problems with future expansion.

Table of Contents

120