Introduction

Didrik Voss has done a good job introducing Scale Rail readers to some of the issues involved in developing an accessory bus for model railroad layouts. He outlined the history of DCC and his wish to develop a similar standard local control bus (LCB), and the initial mandate of the NMRA Net Working Group to be used as a guide to he development of such a bus.

He also described one of the proposals, S9.5, in the article. This article is directed at filling in some of the gaps in the history and progress of the design process, and to describe how S9.6 differs from other proposed LCBs.

The planning for a open LCB started before the NMRAnet mandate ....

The NMRAnet group developed some other documents to guide the development process and to aide in the evaluation of any subsequent proposals. These were titled "Goals and Acceptance Criteria" and are as follows:

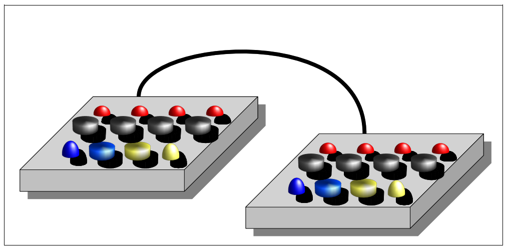
1. **Interoperable** Products from one company can work with products from any other company, and with legacy systems.
2. **No Central Control** Products can interact with other products without the need for a central processor.
3. **Optional PC Control** A PC can provide higher-level functionality, such as a CTC Interlocking, and system functions such as monitoring, testing, and debugging.
4. **Simple** Make it easy for a novice customers to install and configure without technical knowledge (should be easier than DCC systems).
5. **Expandable** Allow very large layouts (a higher-level of technical knowledge required is OK for large layouts).
6. **Flexible** Customers can easily connect many-to-many devices, and connect to legacy systems.
7. **Extensible** Allow additional functionality to be added easily, whether by the NMRA, or manufacturers.
8. **Easy to Implement** Easy and inexpensive to implement products that are reliable and compatible for vendors wishing to create products.
9. **Bi-directional** Support bi-directional exchange of information.
10. **Free IP** The Standards and/or RPs should be free from intellectual-property restrictions from parties other than the NMRA.
11. **Transport Agnostic** Allow different transport mechanisms for messages.
12. **Train Agnostic** Support controlling trains when the network is connected to a train-control system.
13. **Discoverable** Allow a user to find out what devices are connected and how they're configured on a layout.
14. **Self Describing** Devices should describe themselves.
15. **Testable** Allow components to be easily tested for compliance.
16. **Compliance** A name protected by a trademark that NMRA can use to help ensure compliance.
17. **First-Time Use** A new user can buy a device and have it work.

A longer description of these criteria is available online at http://nmranet.sourceforge.net/NMRANetGoalsAndMeasures-2007-10-22.pdf.

S9.6 / OpenLCB Proposal

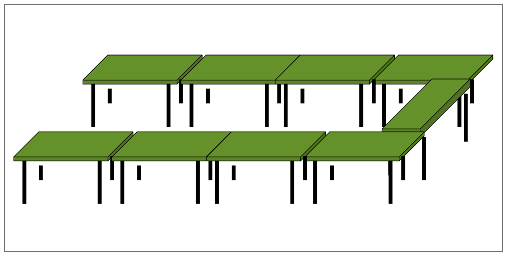
We have put a lot of thought about the hard problems, and how to design an integrated solution that will serve novices to experts, and small to large layouts. We have concentrated on these areas:

1. Simplicity for the first time user: A novice can buy two nodes, hook them together and start to teach them to interact.
2. Support for the large modular layout: S9.6 prevents conflict between modules and automates set-up. Meets do not need to pre-assign or keep lists of number.
3. Automatic traffic control and filtering: S9.6 has this built into the protocol and can use fast buses to interconnect multiple bus segments.
4. Cost effective: S9.6 efficiently bridges to legacy equipment maintaining your prior investments.
5. Controlling complexity with tools.



Simplicity when things are new or small:

S9.6 simplifies life for the novice. Nodes come pre-loaded with serial numbers from the factory. The novice can simply plug nodes together, program them, and immediately control his accessories. Nodes can be programmed with as few as two push-buttons using what we call the Blue-Gold method. The diagram shows two simple nodes. *(insert real picture of a 4channel node).*

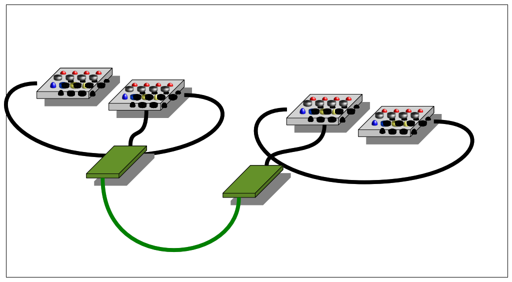


Support when things are big:

S9.6 has ample room for growth, and can easily handle large layouts including museum layouts and large modular layouts, such as occur at annual meets of Ntrak in North America and Fremo in Europe., see the picture to the right). Clubs nodes cannot conflict, even when multiple clubs get together with modules that were programmed at home, because all nodes have guaranteed different serial and event numbers. In addition, none of the modules will need to be re-programmed since there is no chance of a conflict. The exception to this is, of course, teaching some new interactions between the modules. The meet's organizers will not have to keep lists of numbers, nor pre-allocate module numbers, channels numbers, or events. S9.6 also automates and simplifies the set-up and configuration of the modules, allowing the meet organizers to drill down into the individual modules, if necessary.

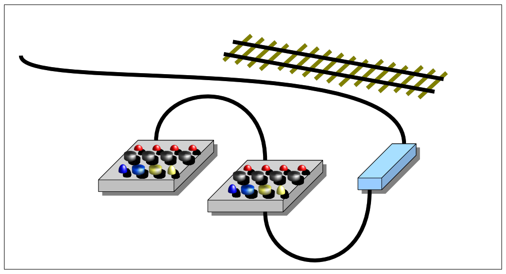
Illustration 1: Fremo Meet in Europe

Correcting when things are too full or too busy:



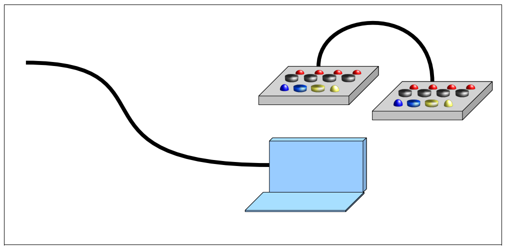
S9.6 can start as one segment consisting of two nodes or multiple segments connected by a fast S9.6 bus. If a segment gets too many nodes, or too much traffic, it can be split into two segments. Multiple segments can be joined by repeater- or bridge-nodes that can automatically filter and route traffic to just those segments and nodes that need it.

Protect your investment:



S9.6 let's you keep using your legacy equipment. Intelligent *bridges* connect to it and integrate its devices right into the S9.6 protocol by including the it inside S9.6's messages. For example, DCC accessory commands are converted to S9.6 events and they can address accessories on the LCB. Other buses expected to be supported are Loconet, Xpressnet, NCR's throttle bus and C\MRI.

Control when things are complex:



Layouts have a habit of growing, both in extent and in complexity. The S9.6 group has designed S9.6 to allow multiple computers to connect to your layout to help design, configure, debug, and if you want, to operate your layout.

How does it all work?

Big Node Numbers

Each S9.6 node has an identifying number called its ID, which is essentially its serial number, which is unique to it in the whole world. This means that you do not need to supply an ID for it, and will never have to deal directly with it. S9.6 node IDs are 48-bit numbers, and each anyone that builds S9.6 nodes is given a large batch of serial numbers that they can assign to their nodes. Each manufacturer is assigned IDs, and even you, if you want to build your own nodes, has reserved series of IDs.

Big Events

The main workhorse on a S9.6 LCB is event, which is represented by a number that is unique to that event (yes, in the whole world). One or more nodes can send this event number in a message (producers), and all the nodes with the matching event-number will listen (consumers). For example, two buttons on either side of a peninsula might set the turnouts in a station to the main line, or a BOD might turn a signal red.

One of the main differences between S9.6 and other buses is that we use very big event numbers which are 64-bits. This has some distinct advantages to the user: nodes can automatically assign a unique event number whenever one is needed, you do not need to think one up nor record it. Any node can teach its events to any other node(s), whether it or they be consumers or producers, in one teaching session.

Another advantage is we can place such things as fast-time, RFID numbers, and other bus's messages right inside S9.6 events, letting nodes respond to these just like any other event, without needing any extra programming to make it happen.

A message for all sizes of messages

Besides events, S9.6 also uses datagrams and streams to carry larger messages. Datagrams can send up to 72 bytes of information between two nodes in a single operation. Streams can send messages from one node to another continuously.

Control of size and of traffic

Layouts will grow to the point where there is too many nodes on it or too much traffic. S9.6 is designed to alleviate both of these problems. The solution to too many nodes is to split the bus into segments. S9.6 was designed from the ground up run efficiently on a segmented bus. It use of a large flat ID-space simplifies the problems inherent in this situation, without the need to extend the node or event IDs, as in other schemes.

While CAN will likely be used in small and medium sized layouts, we believe that a faster bus will be required. For this reason, S9.6 is also implemented on Ethnernet. This allows the connection of multiple CAN segments through gateways. To help with traffic overload, S9.6 implements automatic analysis, filtering and routing of message traffic such that messages only go to nodes and segments that need them – this is called *interest-based routing* – and reduces segment traffic to mostly local messages.