Minutes – Toronto, ON – July 17, 2003

The DCC Working Group meeting was held in the convention hotel from 8:00 am –12:00 pm on Thursday, July 17th, in the Wild Rose Night Club.

Karl Kobel	K2 Engineering	Rutger Friberg	NMRA # 101462
Nathan Oxhandler	NMRA #L534	Bernd Lenz	Lenz Elektronik
Reinhard Mueller	NMRA #121885	Gil Fuhs	MERG
Stefan Bormann	NMRA #128599	Mark Gurries	NMRA #95575
Bob Jacobsen	NMRA #121997	Didrik Voss	NMRA #103404
Richard Daniels	NMRA #22380	David Parks	NMRA #36925
Peter Ely	NMRA #L5035 ?	Allen Pollock	NMRA # ??
Stan Ames	NMRA #L5357	Brian Barnt	NMRA #L5192
Walter Nauorann	NMRA #119885		

Note: Because of the low manufacturer attendance (due to SARs), all votes will be considered straw votes; a binding vote will be cast at the fall meeting.

1. Review of Spring Meeting

Attendance list was corrected.

2. Fall Meeting

A fall meeting will be scheduled during the HobbyVisions show in Las Vegas, tentatively Friday, October 10th at 7:30 pm.

2.1. Topic 9910241 – Analog Output Instruction

Steve did not attending. Tabled till Fall Meeting.

2.2. Topic 0107122 – User Defaults

Waiting for demonstration.

2.3. Topic 0104191 – CV Table Structure Proposal

Waiting for demonstration.

2.4. Topic 0304231 - Additional Mfg CVs for Accessory Decoders

David Nickolson of ZTC has proposed adding CVs 542-544 to the Manufacturer Unique area starting at CV 545. There was a general consensus to accept this proposal.

2.5. Topic – Medium Plug Ehancement

Roco has several different proposals to expand the Medium connector in locomotives to support more functions. S-Helper Service has also enhanced the medium plug, which needs to be investigated.

- A counter-proposal was discussed to include a function interface (ala SUSI)
- More information is needed about pin assignments
- A suggestion was made to use a second connector for functions
- A further suggestion was made to use a JST connector for additional functions (<u>not</u> the 9-pin in current use), and that this connector should include a serial interface

2.6. Topic 0307121 – New Decoder Register Matrix

Rutger Friberg has proposed changing the way CVs are accessed and organized. Reinhard Mueller to coordinate discussion.

2.7. Topic 0305051 – Decoder Lock Proposal

AJ Ireland of Digitrax posted a proposal (attached) for a new method of programming multiple decoders in a single locomotive. This topic was discussed, and several modifications are interspersed with AJ original proposal (see the *italic* text)

2.8. Topic 0307122 – Service Mode Decoder Lock Proposal

Bernd Lenz, on behalf of a new manufacturer, is proposing an additional Service Mode instruction to prevent the programming of decoders in low-end systems where there is no programming track. Using this method would eliminate the problem of re-programming every decoder on the layout to the same address. This proposal also eliminates a conflict between RP-9.2.1 and RP-9.2.3.

The service mode instruction (0xF9) would be modified to match the format defined in RP-9.2.1.

RP-9.2.3, Appendix A defines the 0xF9 command as follows:

long-preamble 0 AAAAAAAA 0 11111001 0 EEEEEEEE 1

RP-9.2.1 defines all Configuration Variable access instructions as having the overall format:

preamble 0 AAAAAAAA 0 1111CCCC 0 DDDDDDDD 0 EEEEEEEE 1

To resolve both the problem and the document conflict, Bernd proposes the following broadcast version of the packet:

Upon receiving an 0xF9 command, a decoder which implements this feature will check it's short address against the address contained in the packet (see below). If the address **does not** match the address in the packet, the decoder will ignore further service mode commands (even after a power cycle).

If the decoder receives any packet that is not a valid service mode packet, it will stop ignoring service mode packets.

Existing decoders that implement the existing 0xF9 would reject the packet because the checksum will not match.

The proposed sequence of events would be:

1. <User selects an existing locomotive address & enters programming mode>
2. The system broadcasts the 0xF9 instruction along with the address entered, and turns off power to the track.
3. The command station indicates that it is ready for a new locomotive to be placed on the track, or that an existing locomotive can be re-programmed.
4. <User places new locomotive on the track.>
5. <User selects new address to program (or reprogram) the locomotive to.>
6. The command station sends the normal service mode sequence to program the address.

The format of the instruction is:

1111111111111 0 00000000 0 11111001 0 0aaaaaaa 0 EEEEEEEE 1

where:

aaaaaaa – short address of a decoder to be programmed
EEEEEEEE -- Error Byte

2.9. Topic 0307161 - Clarification of sequence of Multiple Protocols

An additional comment was received about command stations and decoders supporting more than two protocols (i.e. DCC, Bi-Directional, Motorola), and what order that they should be in.

2.10. Topic 0307162 - Transition Sequence for Multiple Protocols

As a corollary to Topic 0307161, a comment was received about adding a possible transition sequence before changing to an alternative protocol in multi-protocol systems. An separate proposal was received to S-9.2 for multi-protocol systems.

2.11. Topic 0307163 - Glossary

A proposal was received to convert the Draft Glossary document to a Technical Note.
There was a short discussion and general consensus to do this.

2.12. Topic 0307164 – RP-9.3.2 Format Revision

?

2.13. Topic 0307165 – Clarification / Rewording of RP-9.2.1 and RP-9.2.2

Many comments were received about the complexity of the new instructions in RP-9.25.1. This topic will review the wording.

DCC Decoder Lock mechanism: Monday, May 05, 2003

Summary:

DCC users sometimes want to install more than one decoder in a single locomotive. Common cases are: using separate decoders for motor control and sound generation, and also additional function only decoders. Because these decoders often need to have their CVs adjusted separately, a mechanism is needed to communicate with only one of multiple decoders installed in a locomotive shell, at a time. As a common workaround, it is possible to have several decoders at different CV1 or CV17/18 addresses, but this is cumbersome to operate and is problematic with service mode programming.

This Lock mechanism must be able to:

1) Do service-mode read and write from a specific decoder, with no need to disconnect other decoders electrically.

2) Discover the type of decoder(s) present inside a locomotive, without disassembly, and despite multiple decoders being present.

3) Work with existing DCC systems without modification

Proposal: Using CV15 and CV16 formally assigned by NMRA to Digitrax for Experimental Lock feature development.

CV16 will carry a number from 0 to 7 inclusive. This is called the "**ID number**". This number identifies a single decoder, so a unique value must be assigned to each decoder in a particular locomotive.

Because most users will use this mechanism with decoders that provide different functions, the recommended coding is:

0: Reset value, as shipped

255: Broadcast ID so all decoders can be programmed to the same address at the same time.

De-facto Definitions:

1: Motor decoder

2: Sound decoder

3: Function-only decoder (e.g. for additional lights)

4-7: Reserved for other user choices

CV15 is the "**unlock number**" , 0 to 7 inclusive, and is used to select the decoder that will respond to standard NMRA programming commands in Service and Operations mode programming.

Lock Rule:

When the values in CV15 and CV16 are equal, all CVs in the decoder can be read or written. When the values in CV15 and CV16 are not equal, only CV15 can be written or read.

Some manufacturers may want to provide a bit in a CV to enable/disable this feature. If so, that should be in a CV reserved for the manufacturers use, as no NMRA CV has been reserved for this purpose.

[The NMRA will define a CV bit for this purpose.]

Note that a command to reset the decoder to default CV values must not be acted upon unless the CV15 and CV16 values are equal. This prevents the user from accidentally resetting multiple decoders at the same time, and losing the ability to separately address them.

CV 1 cannot be locked for writing.

To configure decoders at new installation:

Before installing each decoder in the locomotive:

a) Connect it to the programming track all by itself

b) Write 8 to CV8 to ensure CV15 and CV16 are both zero

c) If desired, read CV15 to confirm that the decoder can be addressed (This is a read of 0, hence fast)

d) Write CV16 to the desired ID number, e.g. 2

e) Write CV15 to the ID number

f) Configure the rest of the decoder

g) Disconnect it and install it in the locomotive

Repeat in turn for each decoder to be installed

Note that if the user is installing only one decoder, there is no need to modify the default values of zero in CV15 and CV16.

To access a decoder after installation:

a) Write the ID of the desired decoder into CV15

b) Read or write other CVs as needed, that will now be executed by the decoder with a matching value in CV16

To identify the decoder(s) present in a locomotive:

a) Write a 0 to CV15

b) Attempt to read a 0 from CV16.

c) If no acknowledge is received, there is no decoder with ID 0.
If an acknowledge is received, a decoder with ID 0 exists.

Repeat this process for ID 1 through 7 to check for those.

Because the ID values are low and only use the 3 least significant bits, a successful read is quite fast. An unsuccessful read, i.e. when the decoder is not present, will take some time on many existing command stations, but eventually give an answer. This can be made faster in the future by having

command stations attempt to read just the expected value, instead of 0 to 255, in the absence of an acknowledge.

When a mixture of lockable and “old” non-lockable decoders are present, simply program the lockable decoder(s) and then finally program the “old” decoder. Operations mode programming is effected with the same logic and makes configuration and programming very easy.

Conclusion: This proposal meets all the required and desired properties of a usable decoder lock mechanism. Of importance is the fact that it works with existing decoders and command stations. Many other methods with greater levels of complexity, flags, sophistication and problems were evaluated, but did not compare to the practicality of this minimalist approach.

Additionally, this has been field-tested in many thousands of decoders of different types over the last 18-months and is separately supported by more than one decoder and software manufacturer. The firmware impact in the decoders is minimal, and is compact and easily added to existing programming algorithms in memory-limited CPU’s.

AJ Ireland @ Digitrax Inc.

PS Special thanks to all the users and others who helped to extensively field-test this over the past 18 months.