Recommended Practice Proposal (version 2):

Service Mode Streaming

A. Purpose

The intention of this proposal is to extend the programming capabilities of DCC decoders to include streaming data. Possible uses for this programming mode is to allow firmware upgrades of decoders via the rails, or loading high volume data into designated memory in the decoder.

B. Environment

The decoder/engine is assumed to be connected to a special programming track, similar to the configuration used in RP 9.2.3 – Service Mode.

C. Initialization

CV{TBD-1} is designated as the streaming mode switch. This CV is programmed with a special value, hex ‘AA’ or 170 decimal, to signal to the decoder entry to streaming mode programming. The CV is programmed with this value using service mode or operations mode programming. A reset packet is then sent to allow the decoder to enter streaming mode programming. It is permissible to disconnect power to the decoder following the programming of CV{TBD-1}, and the decoder remains in streaming mode programming following the next power-up cycle. The decoder shall not exit streaming mode programming until it successfully receives an end-of-transmission packet (packet type ‘01’), with no error indication. Upon reception of such a packet, the decoder shall set CV{TBD-1} to 0, flash the headlights 3 times for 1 second to signal a completed successful reception, and restart normal DCC operations.

Upon entry to streaming mode programming, the decoder shall listen for incoming bytes. The command station/ programmer shall send a test data packet, containing no data, and verify detection of a service mode acknowledgement within {TBD-2} milliseconds from end of transmission. If no acknowledgement is received, the command station shall reverse the polarity of its track signal and retry the test packet. This sequence shall be repeated at least 30 times, unless an acknowledgement is received from the decoder. Upon reception of an acknowledgement, the command station shall start sending data packets, using the same signal polarity of the test packet for which the acknowledgement was received. If no acknowledgement is received following 30 or more attempts with both signal polarities, it is allowed for the command station to report an error and halt transmission.

D. Streaming Mode Bit and Packet Transmission

Bit encoding in streaming mode programming follows RS-232 format. The steady-state signal on the rails is +V in the positive rail and –V in the negative rail. This state also marks a ‘1’ bit. A ‘0’ bit is marked by opposite polarity, -V in the positive rail and +V in the negative rail. The positive rail is identified by the rail to which there exists a connection from the decoder’s red wire, and similarly the negative rail is identified by the rail to which there exists a connection from the decoder’s black wire. Rails can change designation based on the decoder orientation. During initialization as described in section (C), polarity is automatically matched with the decoder orientation such that the signal can be properly read by the decoder.

Voltage levels correspond to the permissible ranges for DCC for the various scales, as specified in S 9.1. Bit timing corresponds to RS-232 serial transmission at 19.2 KBPS.

Every packet received and verified by the decoder is acknowledged by service-mode ACK, a 5 milliseconds current pulse of >60mA. Typically the decoder turns on the motor to signal acknowledgement. If an acknowledgement is not detected within {TBD-2} milliseconds following transmission of the end of the packet, the command station will re-send the data packet. The command station shall retry sending the data packet for 5 times. If, following 5 attempts, an acknowledgement is not received, the command station shall halt transmission and report an error. In this case the transmission has to be restarted. If the decoder detects an error in transmission, due to a bad checksum or timeout of transmission, it shall not acknowledge reception of the packet. If the decoder detects more than 5 errors in a transmission it shall halt reception and turn on the headlights to signal an error.

E. Byte Encoding

Bytes are encoded following the RS-232 standard, 1-start bit which is a ‘0’ bit, 8 Data Bits (Least Significant Bit first) and 1 Stop bit (‘1’ bit).

F. Streaming Mode Programming Packets

Packets in streaming mode follow the Extended Intel Hex Format. In this format, every packet contains a header, up to 31 octets of data encoded in ASCII (2 hex digit characters, which are 2 bytes in transmission, represent one octet), and a checksum ASCII encoded octet.

Except for the start character, all byte values are encoded as 2-ASCII characters representing a hex value. This format, compliant with ASCII code, supports downloading hex data files to a decoder via a PC terminal emulation program.

The header contains the character ‘:’, followed by 2 characters representing the hex byte count (“00” to “20”), followed by 4 characters representing the target address of the first data byte in the packet, followed by 2 characters representing the packet type. The target address is used by the decoder to properly write the data into the expected location.

Data portion of the packet contains 0 – 32 ASCII encoded byte values, according to the byte count in the header.

The checksum value is the 2’s complement of the sum of all bytes in the packet up to but not including the checksum, ASCII encoded.

ASCII representation:

“: **nn AAAA TT DD…DD CC”**

nn – Data Byte count, 0 - 31

AAAA – Starting memory bank address

TT - Packet type

DD…DD - Data bytes

CC - Checkusm

Fig. 1: General Packet Format in ASCII text

Note: Spaces are introduced for clarity and are not actually part of the packet.

G. Types of packets in Streaming Mode Programming

Streaming mode packets are recognized by the packet type octet (4th octet from the beginning of the packet).

Packet ‘00’ – Data: Format “: nn AAAA TT DD…DD CC”

Where: nn – data byte count

AAAA – address within a data bank

TT – Packet type

DD…DD – 0 – 32 Data bytes

CC – checksum. 2’s complement of the sum of all preceding bytes.

this packet contains 0 – 31 octets of data, as described in (F). The address range for data bytes is 0 – 65536 or hex 0 - ‘FFFF’. This range forms a 64KB data bank and constitutes one destination memory bank.

Packet ‘01’ – End of Transmission: Format “: 00 0000 01 FF”

This packet contains no data octets and signals the end of transmission to the decoder. Upon successful reception of this packet the decoder shall clear CV{TBD-1} and exit Streaming Mode Programming.

Packet '04' – Extended Address: Format ": 02 0000 04 XXXX CC"

Where: XXXX - the extended (memory bank) address.

CC – checksum. 2's complement of the sum of all preceding bytes.

This packet contains 2 data octets which form an extended part of the address. The extended part serves as a selector of a memory bank. Following reception of this packet, all subsequent data shall be programmed into the designated addresses within the selected memory bank, until another Extended Address packet is received or until end of transmission.