

LocoNet[®] Personal Use Edition 1.0 Extension

Copyright © 2002, 2003, 2004, 2005 Digitrax Inc, Norcross GA 30071
All rights reserved.

This document is an extension to the LocoNet Personal Use Edition 1.0. This information is provided solely for non-commercial private use by Digitrax customers. No rights are conveyed for the commercial use of this information, and Digitrax is not able to provide technical support for private use. Any commercial software or hardware products developed using this information will require a license from Digitrax and also need to be certified by Digitrax. Digitrax conveys no warranty for this information and incurs no obligations for its use or incorrect usage. Possession of this information signifies acceptance of these conditions of usage. LocoNet is a registered trademark of Digitrax Inc.

SV Programming Message Formats Version 13

This note describes how the 16 byte OPC_PEER <0xE5> <0x10> message is used for the purpose of programming System Variables (SVs) over the LocoNet.

This usage of the OPC_PEER <0xE5> <0x10> message is distinguished from others by the value of the 4th byte, and the upper nibbles of both the 6th and 11th bytes as shown below. All devices sending these messages must ensure that those are set appropriately, and all devices interpreting any message with an E5 op-code must check these locations to determine whether the received message is an SV programming command. Two message structures are defined that are distinguished by the value of the 4th byte:

Type 1: E5 10 nn nn 01 0n nn nn nn nn 1n nn nn nn nn cc

Type 2: E5 10 nn nn 02 1n nn nn nn nn 1n nn nn nn nn cc

1) SV programming format 1

This is the message format as implemented by the certain existing devices. New designs should not use this format.

The message bytes are assigned as follows:

<0xE5> <0x10> <SRC> <DST> <0x01> <PXCT1> <D1> <D2> <D3> <D4> <PXCT2> <D5> <D6>
<D7> <D8> <CHK>

The upper nibble of PXCT1 must be 0, and the upper nibble of PXCT2 must be 1. The meanings of the remaining bytes are as defined in the LocoNet Personal Edition specification.

2) SV programming format 2

This is the recommended format for new designs.

The message bytes as assigned as follows:

<0xE5> <0x10> <SRC> <SV_CMD> <SV_TYPE> <SVX1> <DST_L> <DST_H> <SV_ADRL>
<SV_ADRH> <SVX2> <D1> <D2> <D3> <D4> <CHK>

The upper nibble of both SVX1 (PXCT1) and SVX2 (PXCT2) must be 1.

2.1) Message Field Usage

The field designation is defined below in the table:

<SRC>	7bit source address for the device issuing the programming command. 0x0 to 0xF: typically PCs 0x10 to 0x7F: other LocoNet devices
<SV_CMD>	<p>Specifies the SV access type</p> <p>0x01 = SV write: write 1 byte of data from D1 0x02 = SV read: initiate read 1 byte of data into D1 0x03 = SV masked write: write 1 byte of masked data from D1. D2 contains the mask to be used; a 1 bit in D2 means that the corresponding bit should be written. 0x05 = SV write 4 bytes: write 4 bytes of data from D1..D4 0x06 = SV read 4 bytes: initiate read 4 bytes of data into D1..D4 0x07 = Discover: causes all devices to identify themselves by their MANUFACTURER_ID, DEVELOPER_ID, PRODUCT_ID and Serial Number. 0x08 = Identify: causes an individual device to identify itself by its MANUFACTURER_ID, DEVELOPER_ID, PRODUCT_ID and Serial Number. 0x09 = Change Address: Changes the device address to the values specified in <DST_L> + <DST_H> in the device that matches the values specified in <SV_ADRL> + <SV_ADRH> + <D1>..<D4> that we in the reply to the Discover or Identify command issued previously. 0x0F = Reconfigure: initiates a device reconfiguration or reset so that any new device configuration becomes active.</p> <p>read response messages, sent in response to a read command:</p> <p>0x41 = REPLY SV write: transfers a write response in D1 0x42 = REPLY SV read: transfers a read response in D1 0x43 = REPLY SV masked write: transfers a masked write response in D1 0x45 = REPLY SV 4 byte write: transfers a write response in D1..D4 0x46 = REPLY SV 4 byte read: transfers a read response in D1..D4 0x47 = REPLY Discover: transfers an Discover response containing the MANUFACTURER_ID, DEVELOPER_ID, PRODUCT_ID and Serial Number 0x48 = REPLY Identify: transfers an Identify response containing the MANUFACTURER_ID, DEVELOPER_ID, PRODUCT_ID and Serial Number 0x49 = Change Address: transfers a Change Address response. 0x4F = REPLY Reconfigure: Acknowledgement immediately prior to a device reconfiguration or reset.</p>
<SV_TYPE>	<p>This field determines the usage of the other fields. A value of 0x02 is required for this format.</p> <p>All other values reserved</p>
<SVX1>	<p><0 0 0 1 D3 D2 D1 D0> These bits provide the top bits of the next 4 bytes</p> <p>D3: bit 7 of <SV_ADRH> D2: bit 7 of <SV_ADRL> D1: bit 7 of <DST_H> D0: bit 7 of <DST_L></p>
<DST_L>, <DST_H>	<p>16bit destination address of the device being programmed. The <SV_TYPE> = 0x02 message format does not have a broadcast address (<DST_L> = 0 <DST_H> = 0) like other message. The Discover command <SV_CMD> = 0x07 performs a broadcast for all devices to report their identity.</p>
<SV_ADRL>, <SV_ADRH>	<p>16bit destination EEPROM address for read/write. For multi-byte operations this specifies the D1 address: the other bytes go to <Addr+1>, <Addr+2>, <Addr+3></p>
<SVX2>	<p><0 0 0 1 D3 D2 D1 D0> These bits provide the top bits of the next 4 bytes</p> <p>D3: bit 7 of <D4> byte D2: bit 7 of <D3> byte D1: bit 7 of <D2> byte D0: bit 7 of <D1> byte</p>
<D1>...<D4>	<p>Data payload bytes. D1 is used for single byte operations; D1 is LSB for multi-byte operations.</p>

2.2) Message Notes

The following notes contain extra information about the commands described in the table above.

2.2.1) Device Addressing

Destination devices are addressed on the LocoNet by the two fields <DST_L> + <DST_H>. Each device must have a unique address to function correctly. New devices may be delivered with their <DST_L> + <DST_H> fields set to defaults. If several new devices with the same address are connected to the LocoNet, they will need to have their addresses changed before they can be configured. See the Change Address note below for details on how to change a device's address.

2.2.2) Writing to a Device

When a Write command is received, the destination device selected by <DST_L> + <DST_H> programs its EEPROM at the address selected by <SV_ADRL> and <SV_ADRH> with 1 or 4 bytes of data according to the command issued. It then generates a Write Response message with <SV_CMD>.6 = 1 and <D1>..<D4> contain the values that are read back from the locations in the EEPROM that were just written to. By comparing the data values in the response the programming device will be able to determine if the write operation was successful.

2.2.3) Reading from a Device

When a *Read* command is received, the destination device selected by <DST_L> + <DST_H> reads 1 or 4 bytes of data from its EEPROM at the address selected by <SV_ADRL> and <SV_ADRH>. It then generates a "read reply" message with <SV_CMD>.6 = 1 and fields <D1>..<D4> set to the values read from the EEPROM.

2.2.4) Discover and Identify

When the Discover command is issued, all devices will respond with a Discover response. When the Identify command is issued, only the device addressed by <DST_L> + <DST_H> will respond. In either case the response contains the manufacturers assigned MANUFACTURER_ID in <SV_ADRL>, DEVELOPER_ID, in <SV_ADRH>, the PRODUCT_ID in <D1> + <D2> and the Serial Number in <D3> + <D4>. Each 16bit value is transferred LSB first. If there are multiple devices with the same <DST_L> + <DST_H> address then the programming device must detect this and change the addresses of the conflicting devices as described in the Change Address note below.

2.2.5) Change Address

A board address is changed by executing the <SV_CMD> = 0x09 command. For this command the device is NOT address by the fields <DST_L> + <DST_H>, it is addressed by comparing the values in the fields <SV_ADRL> + <SV_ADRH> and <D1>..<D4> fields to its internal values that it would use when it replies to a Identify command. If the fields match then the device changes it's address to the values specified in <DST_L> + <DST_H>.

The sequence for doing this is as follows:

1. Issue a <SV_CMD> = 0x07 Discover or <SV_CMD> = 0x08 Identify command. This will all or the specified devices to respond with their MANUFACTURER_ID, DEVELOPER_ID, PRODUCT_ID and Serial Number.
2. The programming device (probably a PC) will collect all the responses and will detect devices with the same <DST_L> + <DST_H>.
3. For each device with a conflicting address or for the specific device you want to change, the programming device will execute a <SV_CMD> = 0x08 Change Address command, using the information returned by the previous Identify response.

Important: For the Change Address command to work correctly, each device must be factory or user programmed with a unique 16bit serial number which is least unique to the layout.

If board sales exceed 64k units, to maintain serial number uniqueness a new PRODUCT_ID will need to be assigned for the same device. Software suppliers should consider this possibility in their designs.

Some manufacturers (e.g. Digitrax) may anticipate board sales >64k units from the start and pre-allocate several PRODUCT_ID's for a new product?

2.2.6) Standard SV/EEPROM Locations

To ensure consistency across devices, a number of SVs are predefined to contain useful system information. These SV locations are defined as follows:

SV 1 = EEPROM Size

0 = 256 bytes
1 = 512 bytes
2 = 1024 bytes
3 = 2048 bytes
4 = 4096 bytes

SV 2 = Software Version Number

The software version number is a value in the range 0 – 255.

SV 3 = Serial Number Low & SV 4 = Serial Number High

The 16bit serial number is accessible via SVs so that it can be user configurable. Some vendors may choose to disable this feature.

2.3) Manufacturer and Developer ID's

It is intended that the both the Initial Program Loader (IPL) protocol and the SV protocol should use the same MANUFACTURER_ID and DEVELOPER_ID numbers.

The MANUFACTURER_ID is an 8 bit value that uniquely identifies all devices from a device manufacturer. If a manufacturer has a *NMRA DCC Manufacturer ID* Number, this should be used. For a list of numbers see: <http://www.nmra.org/standards/DCC/mfgnumbers.html>

Manufacturers who use their own *NMRA DCC Manufacturer ID* may choose to assign and manage their own set of DEVELOPER_ID numbers or leave the DEVELOPER_ID set to 0.

LocoNet developers who do not have a *NMRA DCC Manufacturer ID* should apply for a unique DEVELOPER_ID from Digitrax. This DEVELOPER_ID would be allocated by Digitrax under one of the following special reserved *NMRA DCC Manufacturer ID* numbers: 0, 254 & 255.

DIY LocoNet Developers (LocoNet Hackers) should allocate and manage their own DEVELOPER_ID list under the *DIY NMRA DCC Manufacturer ID* of 13. This list will most likely be contained in a file in the files section of the LocoNet Hackers Yahoo list.

1.1 Change Log

Version 5 22-Dec-2002 Alex Shepherd

- Added Laurence's comments about serial number uniqueness and possible Serial Number wraparound, requiring multiple PRODUCT_ID's to be allocated for the same devices when sales exceed 64k units.
- Standard SVs now are numbered from 1
- Assigned Serial Number to standard SVs 3 & 4 to allow for it to be user configurable via the normal SV write mechanism if vendor allow it.

Version 6 31-Dec-2002 Alex Shepherd

- Added Bob's comments about there not being a broadcast address anymore.
- Added Vendor Id Register

Version 7 01-Jan-2003 Bob Jacobsen

- Added explicit discussion of how to recognize these messages, described the older format, removed masked-read operation, specified mask location for masked write. Minor editorial changes for consistency.

Version 8 02-Jan-2003 Bob Jacobsen

- Following suggestions by David Harris, reformatted to remove some italics & bolding, added parallel comments to section 2 on format constraints.

Version 9 02-Jan-2003 Bob Jacobsen

- Fixed typo in section 2 on constraints.

Version 10 11-Nov-2003 Stefan Bormann

- Fixed typo in page 2, section of <SV_CMD>
- Added FREMO to vendor list.

Version 11

- Reformatted

Version 12 3-Dev-2005 Alex Shepherd

- Reformatted
- Added Personal Use Prefix and Digitrax Copyright Notice
- Generated PDF

Version 13 5-May-2006 Alex Shepherd

- Changed meaning of the 16 bit VENDOR_ID to be an 8 bit MANUFACTURER_ID and an 8 bit DEVELOPER_ID. This is to make the SV protocol consistent with the Initial Program Loader (IPL) protocol usage of what was similar but not identical terms.
- Removed VENDOR_ID table as these will now be managed separately.