RR-CirKits

Specializing in Affordable Electronics for Model Railroads

Installation Guide
Revision-a.2
July 2020

LCC-LocoNet Gateway

LCC (Layout Command and Control)
1024 function LCC – LocoNet translator

This PDF is designed to be read on screen, two pages at a time. If you want to print a copy, your PDF viewer should have an option for printing two pages on one sheet of paper, but you may need to start with page 2 to get it to print facing pages correctly. (Print this cover page separately.)

Copyright

This document is Copyright © June 2020 by **RR-CirKits, Inc.**. You may distribute it under the terms of either the GNU General Public License, version 3 or later (http://www.gnu.org/licenses/gpl.html), or the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/), version 3.0 or later.

All trademarks within this guide belong to their legitimate owners.

Authors

Dick Bronson

Feedback

Please direct any comments or suggestions about this document to:

dick@rr-cirkits.com

Contact Information

RR-CirKits, Inc.
7918 Royal Ct.
Waxhaw, NC USA 28173

http://www.rr-cirkits.com
sales@rr-cirkits.com
service@rr-cirkits.com
1-704-843-3769
Fax: 1-704-243-4310

Publication date and firmware version

Published July 2020.

Firmware Rev B6b

You can download an editable version of this document from
http://www.rr-cirkits.com/manuals/LCC-LocoNet-Gateway-manual-a2.odt

Contents

Overview

LCC–LocoNet Gateway Image

LCC–LocoNet Gateway Connectors

The LCC®-LocoNet® Gateway interface provides a simple and easy way to connect between the NMRA LCC® bus and the Digitrax LocoNet bus. The LCC-LocoNet Gateway may be connected at any convenient point on the NMRA LCC® bus.LCC® is a registered trademark of the NMRA. www.nmra.org
LocoNet® is a registered trademark of the Digitrax corporation. www.digitrax.com

1 About LCC

The NMRA LCC® is a subset of the OpenLCB specifications created by the OpenLCB group for Layout Command and Control. http://www.openlcb.org/

NMRA LCC® devices are controlled by EventIDs. Each EventID has a unique value that will never be repeated by any other LCC® event in use anyplace on your system, nor even on anyone else's system. The only meaning given to any specific EventID is that which you give it. It might represent a turnout command, (Rimrock East Reverse) a block detection message, (Rimrock East Main Occupied) or even a time such as 'Crack of Dawn' or 'High Noon'. Internally the LCC uses a 64 bit number to represent each event (18,446,744,073,709,551,615 possible EventIDs) so we are not planning to run out of EventIDs anytime soon. EventIDs are created by event 'Producers' and used by EventID 'Consumers'. The same EventID may be created by one or more Producers, and may be used by any number of Consumers. (or none at all)

EventIDs happen, they are not states nor the status of indicators. The only memory of EventIDs past exist in hardware. In other words an EventID can tell you to turn a light 'on'. A different EventID can tell you to turn a light 'off'. Many different EventIDs can tell you to turn the same light 'on'. However there is no EventID that tells you that the light is 'on'. That is a state, and only resides in the hardware that controls the light. (actually LCC defines other messages that can query nodes for their state during initialization)

Of course you are free to add another piece of hardware that watches for the same EventIDs and controls its own light to duplicate the original. You could even have a light that turns on when it sees the EventID 'Crack of Dawn' and turns off when it sees the EventID 'Evening'. You could label it "Daylight" and use it to tell you that the room lights should be 'on', or better yet, turn on the room lights itself.

1.1 Some Definitions

Event

Events are the real world changes on our layouts:

- the change in occupancy state of a Block
- the intent (pressing a button) to control the position of a turnout
- the feedback of the change in position of a turnout
- the triggering an animation as another physical thing happens
- the control of various signaling schemes based on sensed track positions and train movements etc.

EventID

Events on our layout are represented in LCC by EventIDs which are the actual messages sent over the system. Because events can represent anything that happens on our layouts there is no presumed context nor meaning given to EventIDs. They are simply globally unique 64 bit numbers. Their only definition is what you give them.

Well Known EventID

Some EventIDs are given a meaning by the LCC developers for common purposes. This simply means that someone else has defined a particular EventID or group of EventIDs and published what it is. You are free to use it only if you accept its definition for use on your layout. One example might be the time related EventIDs used by system clocks. If a clock manufacturer uses the Well Known time EventIDs, then you can simply look up each EventID in a list and use it to cause something to happen at that time without copying the EventID from the clock node itself. If a second manufacturer decides to build a clock node and uses the same Well Known time EventIDs, then your nodes will work with his clock just as they did before.

Line (channel)

Each LCC-LocoNet Gateway contains 1024 I/O lines. (or channels) Each line has the ability to watch for 2 events (consumers) and to send out 2 events. (producers) Each line translates its events into the opposite protocol and queues them for sending out to the appropriate bus.

Consumer

Each consumer event can be configured to control the line's output in one of two ways. These are; 'on' (activate), 'off' (inactivate).

Producer

Each producer event can be configured to trigger in one of 2 ways: These are function 'on'/'off'.

2 LCC-LocoNet Gateway Features

- The LCC-LocoNet Gateway uses the CAN bus implementation of the NMRA LCC.
- Communicates over the LCC CAN bus at 125Kb.
- Support for a total of 1024 Input/Output lines. (channels)
- CDI (Configuration Description Information) controlled programming via Software.
- Boot Loader allows contact less user firmware upgrades over the LCC® (Layout Command & Control) connection.
- Power is supplied from the LCC® bus. The LCC-LocoNet Gateway requires 20ma.
- Efficient switcher regulated 5VDC.
- Power may be supplied to the LCC Bus. (500mA per LCC connector)
- Option to create a stand alone LocoNet for systems without a Digitrax command station. (this option bypasses the LocoNet isolation)
- Option to connect the Digitrax Railsync lines to the DCC pair of the LCC cable to allow the command station to feed boosters over the LCC cable.
- Each segment of LCC® cable requires a terminator at each end.

3 Power, Serial Connections, and Indicators

The LCC-LocoNet Gateway has six connectors and four status indicators. Two of these connectors are for connections to the LCC bus network. Two are used as connections to the LocoNet lines. One is a power connection, and the remaining connector is for the USB connection. This section covers the system connections consisting of the CAN bus port connectors, LocoNet connectors, power connection, USB port connection, and Status indicators.

3.1 CAN LCC® Compatible Connector

The LCC data connection is made to the LCC-LocoNet Gateway via an LCC cable connected to either of the two RJ-45 connectors LCC1 and LCC2. The LCC wiring passes straight through both connectors. The LCC specification requires a minimum of 1' of cable length between connectors. Slightly shorter cables (10") should not significantly impact operation.

Pin outs for the CAN LCC RJ-45 data connector:

Pin	Description
1	CAN H
2	CAN L
3	GND
4	DCC RailSync
5	DCC RailSync
6	GND
7	GND
8	+Power 12-27V

LCC power is supplied on Pin 7 and Pin 8. Power can be from +12VDC to +27VDC. The RR-CirKits LCC-LocoNet Gateway delivers approximately 15VDC to the bus when powered by the included power supply.

The LCC connectors accept standard Ethernet style CAT5 (or better) cables. 4 pair cables are required by the LCC-LocoNet Gateway. For any but the smallest networks it is recommended that you choose AWG 24 CAT-5 wiring. The use of AWG 26 wiring reduces the maximum length of your network to approximately 40% of its specified length. Especially avoid using copper clad aluminum wire or AWG 28 low profile wiring as they have even higher than normal resistance at the relatively low frequencies used by the LCC. This higher resistance shortens the maximum distance for reliable communications even more than using AWG 26 wiring does.

A note on connectors: RJ-45 crimp connectors are made with three blade styles. (the end that crimps into/onto the wire) Single 'U', double 'UU', and triple 'VVV' points. Stranded cables may be made with any of the three blade styles because the points crimp into and between the individual wire strands. However if you are using solid wire, then you must only use the three point style of blade. It is

designed to trap the solid wire between the three points, two on one side, and the center one on the other side, for a corrosion tight connection. The single or double pointed blades simply press against the side of the solid wire, and will fail in time. (usually the morning of your open house)

3.2 Power Connections

The LCC-LocoNet Gateway requires a power source of between 7.5 and 27 volts DC connected. This power may be provided through the 5.5mm by 2.1mm coaxial connector and/or through the CAN bus power from another power source, such as an LCC Power-Point, or the RR-CirKits LCC Repeater.

If powered via the coaxial power input, then the LCC-LocoNet Gateway can act as the power source for a segment of LCC cable. The LCC-LocoNet Gateway acts as a convenient way to power a small LCC network including other LCC boards over standard RJ45 cables.

3.3 USB

The LCC-LocoNet Gateway includes an internal LCC to USB interface. The USB connection is optional and the Gateway function will operate with or without the USB connection. The USB interface is only to the LCC, and does not serve as a LocoNet interface. The USB interface can serve as the interface for the LCC network instead of an LCC Buffer-USB or other interface device.

3.4 LocoNet Connector Wiring

The LocoNet® connection is made to the LCC-LocoNet Gateway via the provided cable connected to one of the RJ-12 (6 pin) modular jacks. (LocoNet1 or LocoNet2)

LocoNet® cables are wired straight through, not reversed like phone cables.

Pin outs for the RJ-12 connector:

Pin	Description	Color
1	Rail_Sync-	white
2	Signal Ground	black
3	LocoNet-	red
4	LocoNet+	green
5	Signal Ground	yellow
6	Rail_Sync+	blue

LocoNet® pins 2 and 5, and pins 3 and 4 are connected together internally.

3.4.1 LocoNet configuration options

The LCC-LocoNet Gateway contains several internal option switches and jumpers

---

related to the LocoNet operation.
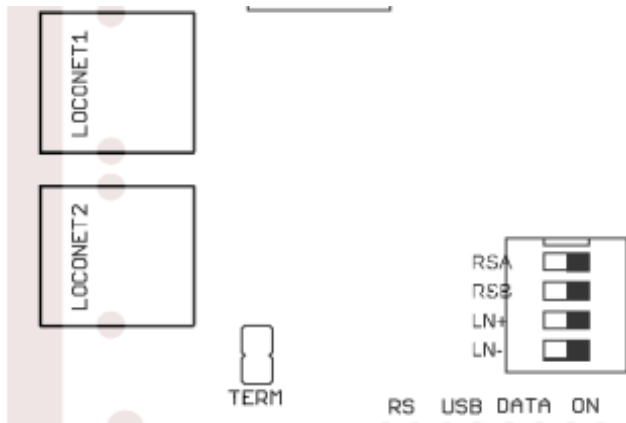
Loconet Options

TERM – The Term jumper allows the LCC-LocoNet Gateway to provide the required LocoNet 15ma. pull up current for its stand alone operation. Do NOT connect this jumper if the LocoNet is being provided by a command station or other source.

LN+ and LN- - These two switches provide DC power from the LCC-LocoNet Gateway to the Railsync lines in order to power throttles or isolated LocoNet devices such as a LocoBuffer-USB that normally receive their power from the Railsync booster drive. These switches will bypass the normal isolation between the LCC and LocoNet networks. Do NOT activate these switches if the RailSync lines are part of a normal DCC booster drive circuit as it will cause a short circuit that may damage equipment.

RSA and RSB The Railsync A and Railsync B switches connect the DCC drive lines of the LCC bus to the DCC drive lines of the LocoNet bus. This allows a LocoNet command station to drive LCC connected boosters, or an LCC command station to drive LocoNet connected boosters. Please note that if the booster drive connection is not isolated then it may be required to install a common connection between the command station and any boosters.

3.5 Status Indicators

The LCC-LocoNet Gateway has four status indicators located near to the coaxial power connector. The green **ON** status indicator shows the power status of the LCC-LocoNet Gateway itself. The yellow **Data** indicator normally shows all data activity related to the LCC-LocoNet Gateway, and also any activity/error status during power up or a boot loader firmware upgrade. The red **USB** indicator shows activity over the USB connection. The green **Sync** indicator shows power on the RailSync lines. (see section 3.4.1)

3.6 Prog Button

The Prog Button may be used to force the unit into boot loader mode for firmware upgrades. Normally this is not required.

4 Getting Started

We suggest that you use a computer program such as the JMRI DecoderPro (4.20 or later) <http://www.jmri.org/> or Deepsoft LCC-CDI <http://www.deepsoft.com/> to configure the LCC-LocoNet Gateway. These "point and click" interfaces will save you much time and frustration while setting the many possible options that you will need to configure, and in fact are the only way that we offer for configuring the LCC-LocoNet Gateway node.

Node Address: Each LCC-LocoNet Gateway has a single node address that is used for CDI programming on the layout. Each individual LCC-LocoNet Gateway has its node address imprinted on the label on the front of the case.

## 4.1 CDI (Configuration Description Information)

The CDI is the way that LCC nodes store their configuration options internally. Instead of relying on printed manuals or volunteer created files to present the various decoder options, (like DCC devices have for the past 20 years) the LCC specification expects the manufacturer of the LCC node itself to present its capabilities and options in a standardized manner from an internal file. This allows any LCC configuration tool to be used interchangeably, and not need to be updated to support new hardware or firmware upgrades.

CDI Window

Start up the CDI tool. Once the tool is monitoring the LCC network you will be presented with a list of nodes similar to this.
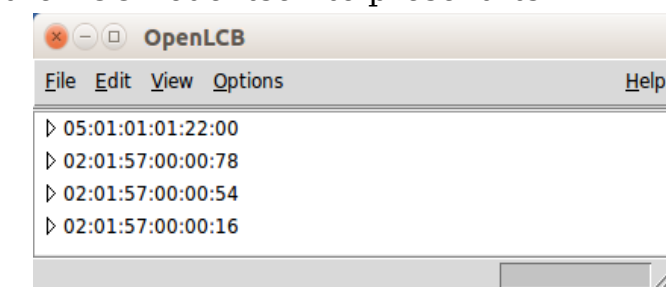
The list includes all the nodes that are currently visible on your LCC network. In this example the first entry in the list (05:01:01:01:22:00) is the configuration program itself as seen through the interface.
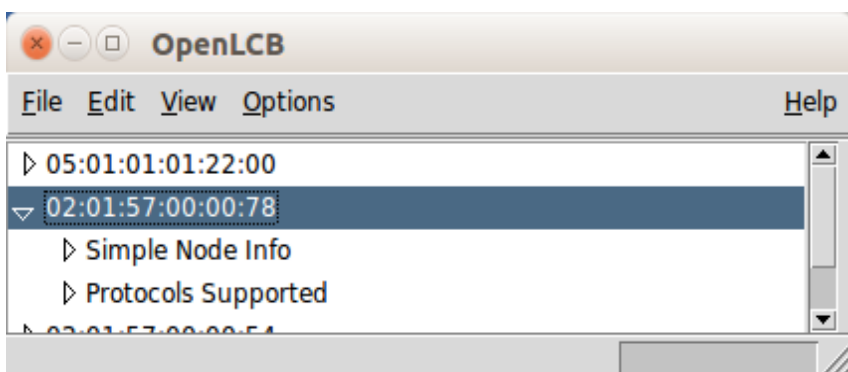
Open Node

Select the node that you desire to configure and click on its arrow to open it. This will open up further options for that node. In this example there is an option for 'Simple Node Information' and a list of 'Protocols Supported'.

The Simple Node Information will help you to be sure that you have chosen the correct node.

With the Deepsoft OpenLCB program you simply highlight a Protocol item to open it in a new window.

The JMRI CDI tool is similar.

CDI is the supported Protocol that you will need for configuration purposes.

Remember that with the LCC this display information is provided by the manufacturer and stored in the node itself rather than in some piece of paper, external file or program. This means that if you use JMRI instead of Deepsoft OpenLCB to open your node for configuration, the layout of the tool windows may be different, but the content will remain unchanged.

Once the CDI window opens up you can modify its contents by using the 'Read' and 'Write' buttons found near to each item.

Read button will fill in the field with any contents that were previously stored in the node. Using 'Read' for an EventID will present a new, unused, value if one has not been previously stored.

There is also a 'Refresh All' button located at the bottom of the window.

Write button will store the currently displayed value or selection into the node's memory. If you have changed any value you must always then do a 'Write' to store it into the node.

There is also a 'Save Changed' button located at the bottom of the window. This will save all changes that have not already been written back into the node.

The newest JMRI CDI tool will highlight the entry with orange until it has been written to the board. This is a helpful reminder that the change has not been stored into the board where it can take effect.

4.2 Input/Output Configuration
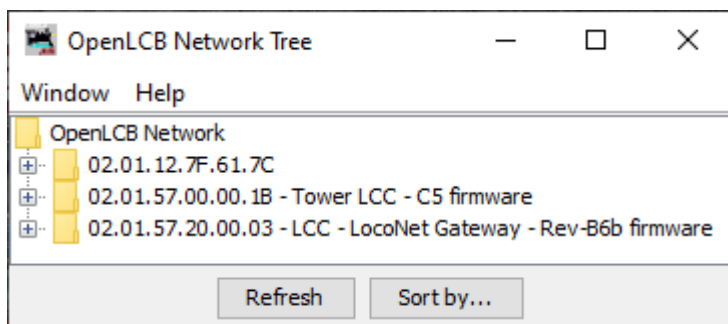
Open the JMRI CDI Window
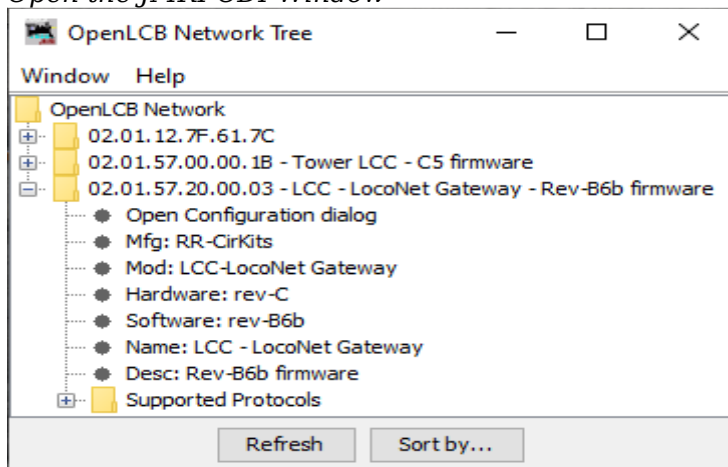
Expand the node to select 'Open Configuration dialog'

We suggest that the user take advantage of the JMRI CDI tool or a similar program to set the LCC-LocoNet Gateway configuration values.

The following examples were made using the JMRI CDI tool for the LCC-LocoNet Gateway.

Select the 'LCC' drop down list and click on 'Configure Nodes'. When the node selection window opens, expand the node to configure, then click on 'Open Configuration dialog' to open the CDI.

This will open the CDI tool and automatically read in the basic information for the node.

This information is presented in a tabular format to allow a reasonably

compact display but still have easy access to the vast amount of configuration information.

4.3 Initialization

The LCC – LocoNet Gateway will only forward those LocoNet messages for which the gateway has established a consumer on the LCC bus, and the gateway will mark those consumers as 'active' for the current session. All LocoNet messages for which no consumer exists on the LCC bus will simply be ignored. This list of 'active' LCC consumers will be rebuilt every session using the Consumer Inquiry command.

5 Segment:

The CDI is divided up into a number of segments to make it more manageable to configure. The LCC-LocoNet Gateway CDI consists of four main segments.
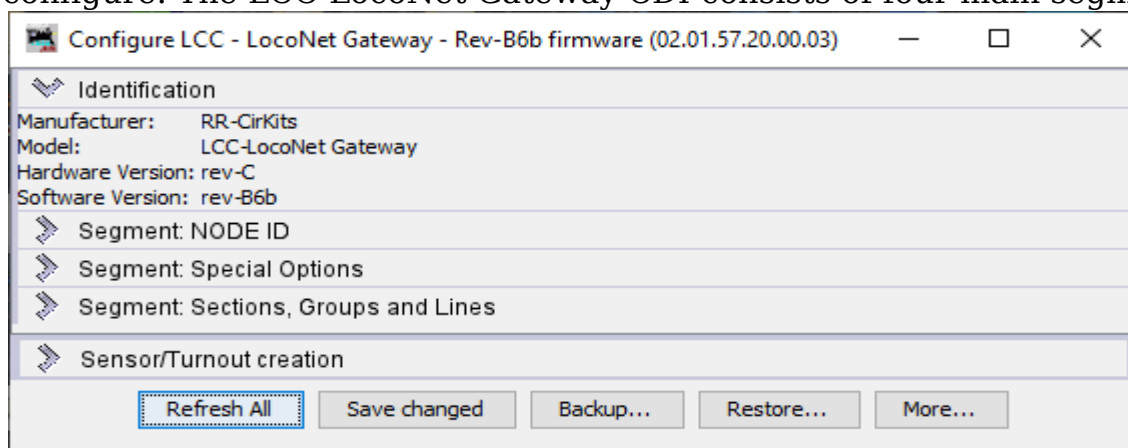

5.1 Segment: Identification

The first segment shown will be the Identification. It includes the manufacturers name and node model plus any version information.

5.2 Segment: NODE ID

The next item is the Node Identification. It contains the Node Name and Node Description that you give to the node. The name of this node is 'LCC – LocoNet Gateway'. (not very original) This name will appear in the node selection window to make it easier to select the correct node for configuration. The Node Description in this example shows the firmware version, but normally would be more related to your layout. There is a 64 character limit to both the node name and description
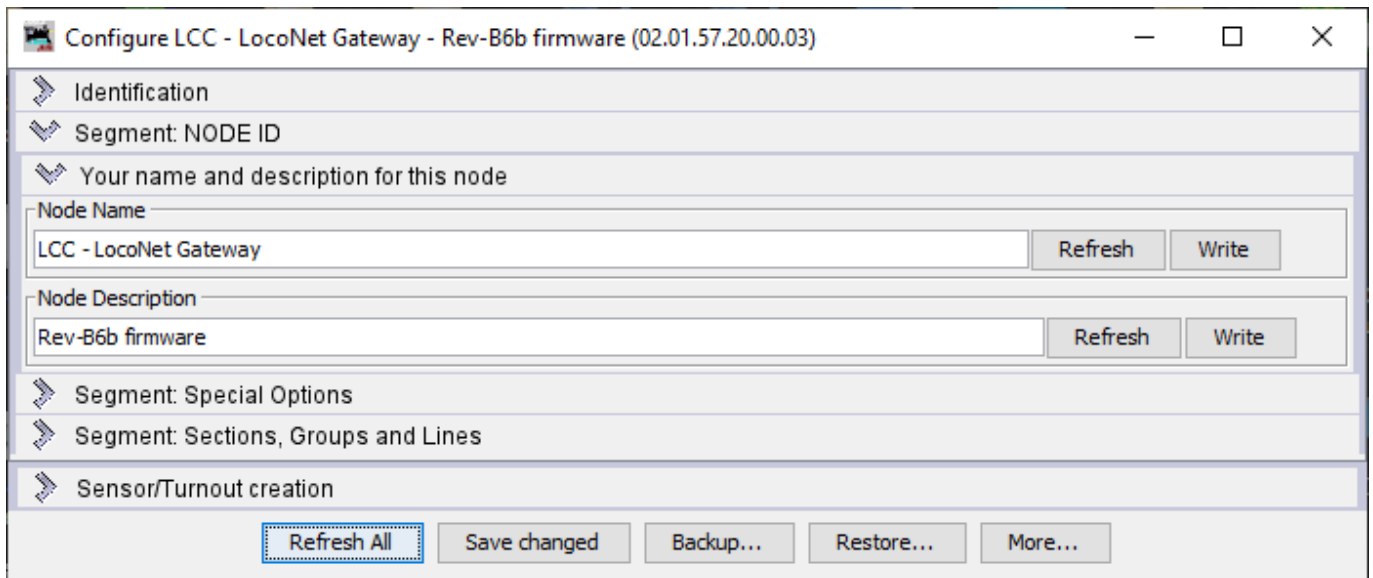
items. These will show next to the node number in the node tree, so short and descriptive is good.

5.3 Segment: Special Options

The LCC-LocoNet Gateway has a number of supported special options. Some of these are node related options, and others are specific LocoNet commands that may be useful on the main LCC network.
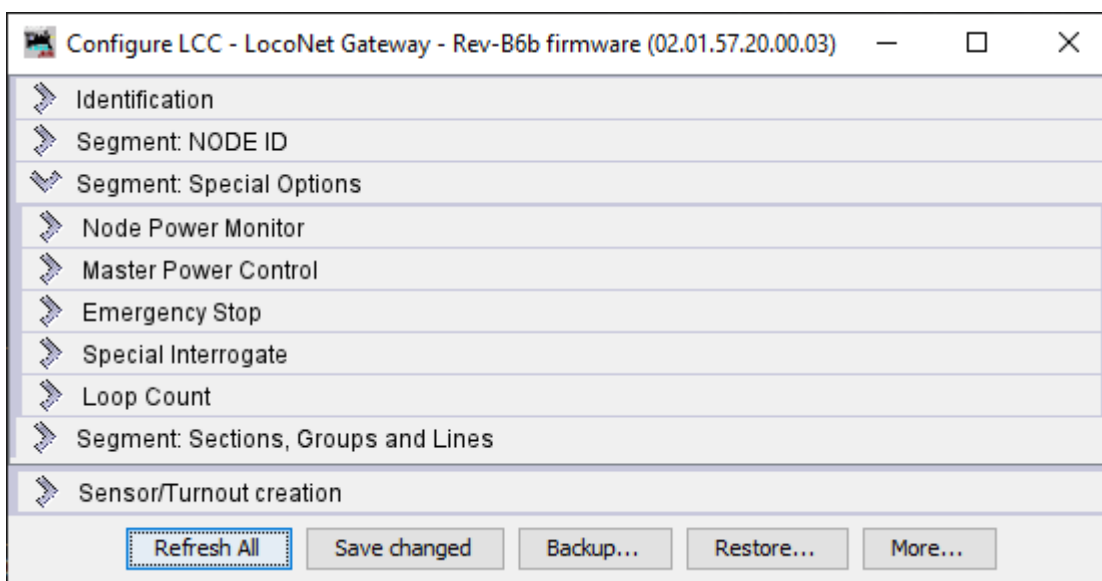
5.3.1 Node Power Monitor

LCC nodes may monitor their own power input and send notifications to the bus if they are in or out of specification. You may choose the EventID or use the defaults.

5.3.2 Master Power Control

Master Power Control relates to the LocoNet messages GPON and GPOFF. (General Power On, and General Power Off) You can either select the default LCC well known *power on* and *power off* EventIDs, use the node defaults, or define your own.

5.3.3 Emergency Stop

Emergency Stop is simply what it says. It is an Event IDthat will send the LocoNet Emergency Stop command. (85 7A Emergency STOP) The command station puts all selected throttles into Emergency Stop. (speed 1)

5.3.4 Special Interrogate

This sends the LocoNet special interrogate series of commands. LocoNet devices are expected to respond with messages indicating their current states.

5.3.5 Loop Count

If the LocoNet fails in a manner that causes rapidly repeated messages this watchdog will disable the forwarding of those messages to the LCC network. Any (non zero) loop count value sets the maximum number of any specific LocoNet messages allowed to be translated per second. If this number is exceeded then the line (channel) is disabled until the 1 second timer is reset.

5.4 Segment: Sections, Groups, and Lines

The LCC-LocoNet Gateway has 1024 bi-directional translation lines. (channels) To prevent presenting 1024 different tab choices we chose to organize them as four sections of 256. Each Section is further divided into sixteen groups of sixteen lines each. With the option to label each section, group, and line it should be possible to configure your gateway in a way that matches your LocoNet layout hardware.

5.4.1 Section

Each section includes 1/4th of the gateway lines. (channels) If it is convenient to divide your LocoNet into sections this can help you. If not, then give them some arbitrary names.

5.4.2 Group

Each group includes 16 gateway lines. This conveniently supports the number of lines in some of the Digitrax hardware items. There is no actual requirement that the LocoNet addresses be consecutively numbered, but it can be helpful when setting up your gateway to do so. This is especially true if you name your groups to match your hardware.

5.4.3 Line (Channel)

Each line (or channel) controls the translation of one LocoNet item such as a turnout (output) or sensor. (input) The LCC-LocoNet Gateway also supports the NMRA extended accessory messages that are normally used for controlling signal aspects. The extended accessory messages support as many as 32 possible states for each message number.

Configuration Items

- Line name – Your identification for the line being converted.
- LocoNet Address – The address portion of the message.
- Message type - Select the specific type of LocoNet message for this translation.
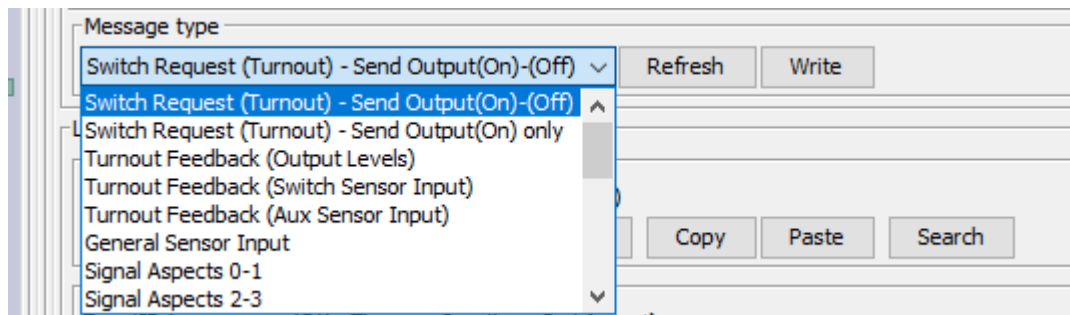
- EventID for message (On – Closed – Active – 1st Aspect) – The first EventID for the LCC side of the translation. You may use the globally unique default EventID or enter your own.
- EventID for message (Off – Thrown – Inactive – 2nd Aspect) – The second EventID for the LCC side of the translation. You may use the globally unique default EventID or enter your own.

For signal aspects you only need to enter aspects that are actually used. However this can cause still create unused items if, for example, you have 'Aspect 0' in your system, but not 'Aspect 1'. Selecting 'Signal Aspects 0-1' will give you both translations but only one will be used.

5.5 Sensor/Turnout Creation

Sensor/Turnout Creation is not actually part of the CDI, but is a feature of JMRI to make it easy to enter LCC EventIDs into the JMRI Sensor or Turnout tables.
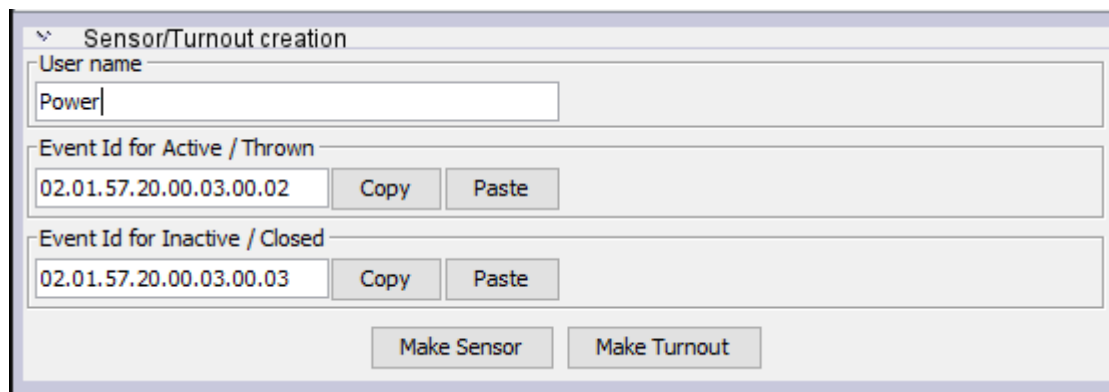
Simply use the Copy and Paste features to select the appropriate EventID values from each line and move them into the tool. You may either type a User Name or cut and paste it from the item. (using Ctrl+C and Ctrl+V) Click on the appropriate [Make Sensor] or [Make Turnout] button and a matching entry will automatically be entered in JMRI. Be sure to save panels in order for JMRI to save this information for future use.

6 Backup and Restore

This ASCII text representation of the CDI data contents should be useful for all sorts of interesting stuff.

7 Trouble shooting

7.1 Sanity Test

To perform a very basic LCC-LocoNet Gateway sanity test perform the following steps:

- Power up the LCC-LocoNet Gateway by plugging it into a powered LCC network, or by connecting a power supply to the unit.
- The green power **ON** LED should come on.
- If no other node is present on the network, then the yellow **Data** light will begin to flash a 4 blink error code while the board seeks to establish an alias.
- Alternatively connect the USB jack to your computer. The device should appear in your active USB device list.
- No power light will show, but the same 4 flash error code will begin to appear on the yellow **Data** light.

If the green power LED does not light, be sure that a power supply is connected to the LCC network segment, or provide at least 9V to the LCC-LocoNet Gateway power jack. The green power LED will initially light at much lower voltages, so it is not a reliable indicator of suitable power.

7.2 Activity Test

The LCC-LocoNet Gateway's USB circuit and code sends data directly to the unit's processor, so if you send any command to the unit over its USB connection, then it should immediately be seen on the red **USB** LED. This test uses the free software available from the JMRI project to watch the test commands. (www.jmri.org)

Steps:
- Open the JMRI LCC® Monitor window. Using the JMRI turnout control send a command to any configured lines on this LCC-LocoNet Gateway. The command should appear in the LCC® monitor window and the LCC-LocoNet Gateway yellow **Data** LED should blink.
- The mapped LocoNet command should be sent out to the LocoNet.

If there is activity at the LCC Terminator blue LED, but no activity light at the LCC-LocoNet Gateway when events are sent, check the LCC wiring. If the command is seen in the LCC® monitor, but not in the command light, be sure that the command you are sending is configured to respond on this LCC-LocoNet Gateway. If there is no activity shown in the LCC® monitor window, check that you have the correct interface selected in the JMRI preferences, and that you have the correct COM port selected.

8 Boot Loader

8.1 Firmware Upgrade

If an update to your LCC-LocoNet Gateway firmware is needed, a program such as "Firmware Update" in JMRI version 4.6 or later is required.

To enter Firmware upgrade mode:

1) Start JMRI and select "LCC".

2) Select 'Firmware Update' from the LCC drop down list.

3) Click 'Select' to pick a firmware file.

4) From the file menu, select: 'LCC_Gateway_UPDATE_B6b.hex' or the latest upgrade available.

5) Optionally you may check the 'Lock Node' check box to take it off line during the upgrade.

6) Click the 'Load' button to initiate the upgrade to LCC-LocoNet Gateway revision B6b.

7) Wait untill 'updating device firmware..' is complete.

8) Switch back to the OpenLCB Network Tree window.

9) It should now show 'Mod: LCC-LocoNet Gateway' and 'Software: rev-B6b'.

10)   Any errors will be shown in the lower window ticker tape display.

If the node does not automatically enter boot mode and start the upgrade it may be forced into boot mode by un-powering it, then holding down the 'Gold' button as you power it up again. The gold LED should start flashing to indicate that it is in forced boot mode. This will likely be required after a failed upgrade attempt.

Note: When upgrading from the built in interface you need to provide external power during the entire process. However the USB connection will be reset before and after the firmware upgrade, causing JMRI to loose its connection during the upgrade process. You will need to restart JMRI and continue the process from the point where JMRI disconnected each time.

9 Grounding and Isolation

The LCC-LocoNet Gateway provides 2,500 VAC electrical isolation between the computer, the LCC bus, and the LocoNet® by using two high speed digital isolators. This prevents possible ground loop problems between the LocoNet®, the LCC, and your computer. For example, many systems are installed without properly bonding the booster to power ground nor to each other.

Some interface devices may connect the LocoNet® ground directly to the USB cable ground. This provides a ground path from the power line through your computer and its USB cable to the LocoNet® and then to the booster and all the railroad wiring. At best this indirect grounding causes electrical noise. At worst it could create a fault path via the small gauge LocoNet® wiring to ground.

By isolating the LCC bus ground, the LocoNet ground, and the USB ground from each other the LCC-LocoNet Gateway prevents the inadvertent introduction of additional ground loop paths between your different networks.

Note: for the LCC and LocoNet isolation to be effective you should not use the option for creating a stand alone LocoNet using the LCC-LocoNet Gateway, as this will use the same power supply for both networks and connect their grounds. For normal LocoNet connections to a command station this isolation is maintained. For connecting isolated LocoNet devices on a stand alone LocoNet network there is no concern with connecting the LCC and LocoNet networks.

10 Warranty Information

We offer a one year warranty on the LCC-LocoNet Gateway. This device contains no user serviceable parts.

If a defect occurs, please contact RR-CirKits at: service@rr-cirkits.com for a replacement.

11 FCC Information

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions:

1. This device may not cause harmful interference, and

2. this device must accept any interference received, including interference that may cause undesired operation.

Note: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the

following measures:
   --Reorient or relocate the receiving antenna.
   --Increase the separation between the equipment and receiver.
   --Connect the equipment into an outlet on a circuit different from that to
    which the receiver is connected.
   --Consult the dealer or an experienced radio/TV technician for help.

Any modifications to this device voids the user's authority to operate under and be in compliance with these regulations. The actual measured radiation from the LCC-LocoNet Gateway is much lower than the maximum that is permitted by the FCC Rules, so it is unlikely that this device will cause any RFI problems.

RR-CirKits, Inc.
7918 Royal Ct.
Waxhaw, NC USA 28173

http://www.rr-cirkits.com
sales@rr-cirkits.com
service@rr-cirkits.com
1-704-843-3769
Fax: 1-704-243-4310

Appendix:
Theory of operation

Note:

This information is from the OpenLCB web site and covers the operation of Gateways in general, not the LCC-LocoNet Gateway specificly.

Gateways

Gateways can route PCERs only to segments with nodes expressing interest by processing the other event messages, including Consumer and Producer Identified. The automatically-routed event range always has to be routed. This can be done by making a permanent entry in routing tables, or any other implementation method.

Examples

Consider a OpenLCB installation with one CAN segment A and one LOCONET segment B connected via a gateway. In addition, a program is attached via a USB link and LocoBuffer-USB "GB2" to LOCONET segment B.
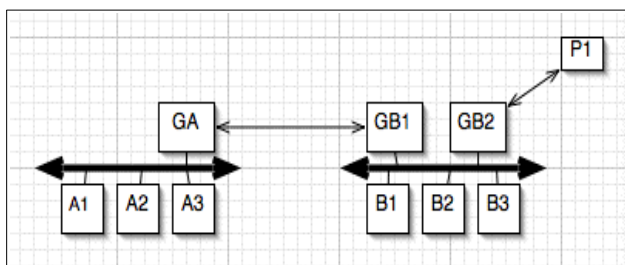

Figure 1 – Example OpenLCB Network

The program has to be able to fully communicate with all OpenLCB nodes on segment A and all LocoNet nodes on segment B. These examples shows how the gateway can make that happen. For simplicity, the labels on the figure will be used as if they were the full NID of the node, e.g. "GA" for the gateway on segment A.

Gateway "GB1" on the LOCONET side doesn't physically exist, but lives inside gateway "GA" and is represented by a routing table. Frames and messages on CAN are represented by ["source address", "content", "destination address"] triplets, those on LOCONET by ["opcode", "destination address", "content"]. Not all messages are shown. Note that there are many ways that gateways can manage traffic, and this is just one approach.

Before the cases below, both segments come up, assign the numbers 1,2,3 as aliases to their local nodes. For simplicity, A1 produces event X, which A2 and B2 want to consume. A1 also produces Y, which no other node wants to consume. B1 produces event Z, which A3 wants to consume.

Case 1: All nodes initialized before link established, event X is sent

The combined GA-GB1 gateway initializes its internal tables.

A1 sends X. GA receives it. Because GA hasn't seen X before, X is (internally) forwarded to GB1 where it is looked up in the routing table. If it exists and has a message type assigned to it, X is translated into a LOCONET message of that type.

GB1 puts the translated X message on the LOCONET segment, and B2 consumes it.

Since we have no way of knowing whether message X exists on B2, we have to rely on the entries in the routing table, which are put in by the user.

GB1 notes message sent, forwards to GA, which marks its routing table to continue forwarding these. GA forwards "Consumer of X Identified" on the CAN segment in case there are other listeners for it. If the LOCONET is disconnected or not powered, GB1 cannot send message X and will not forward "Consumer of X Identified" back to GA.

Case 2: Node B2 comes up, after event X has been sent in the example above

The gateway is up, but doesn't know about B2. GA has seen X sent by A1, but has marked it as "do not forward" if it exists and has a message type assigned in the routing table, because there was no consumer for it on B.

B2 comes up, but the gateway still doesn't know about B2. Note that B2 can only come up if the LOCONET segment is connected and powered.

GB1 detects that the LOCONET segment is up, which changes all "do not forward" markers in the routing table into "forward" markers.

GA now remembers that event X must be forwarded across this link if seen.

A1 sends X, GA receives it, forwards it across link.

Case 3: Node A1 comes up, then event X is sent

The gateway is up, but doesn't know about A1 or X.

A1 comes up, gets a NID alias assigned, sends "AMD" and "Initialization Complete".

A1 sends "Producer of X Identified". GA forwards that to GB1.

GB1 notifies GA that event X, if it exists and has a message type assigned in the routing table, currently must be forwarded across this link if/when seen.

A1 sends X. GA receives it. Because GA hasn't seen X before, X is forwarded across the link to GB1.

The rest of this proceeds like Case 1.

Case 4: Node A1 comes up, then event Y is sent

The gateway is up, but doesn't know about A1 or Y.

A1 comes up, gets a NID alias assigned, sends "AMD" and "Initialization Complete".

A1 sends "Producer of Y Identified".

GA remembers that event Y currently must be forwarded across this link if seen.

A1 sends Y. GA receives it. Because GA hasn't seen Y before, Y is forwarded across the link to GB1.

If Y doesn't exist in the routing table or has no message type assigned to it, no "Consumer of Y Identified" message is received back from GB1, so after a short timeout GA marks it's tables to not forward Y. (If a node later comes up that consumes Y, a sequence similar to Case 2 occurs.)

Case 5: All nodes initialized before link established, event Z is sent

The combined GA-GB1 gateway initializes its internal tables.

B1 sends Z. GB1 receives it. Because GB1 hasn't seen Z before, Z is translated and forwarded across the link to GA if Z exists in the routing table.

GA puts the Z message on the CAN segment, and A3 consumes it.

GA puts the "Identify Consumers of Z" message on the CAN segment.

A3 responds with "Consumer of Z Identified".

GA receives "Consumer of Z Identified", forwards to GB1, which marks its routing

table to continue forwarding these.

Case 6: Node A3 comes up, after event Z has been sent in the example above

The gateway is up, but doesn't know about A3. GB1 has seen Z sent by B1, but has marked it as "do not forward" because there was no consumer for it on A.

A3 comes up, gets a NID alias assigned, sends "AMD" and "Initialization Complete".

A3 sends "Consumer of X Identified". GA forwards that to GB1.

GB1 remembers that event X must be forwarded across this link if seen.

B1 sends Z, GB1 receives it, forwards it across link.

Case 7: Node B1 comes up, then event Z is sent

The gateway is up, but doesn't know about B1 or Z.

B1 comes up, but the gateway still doesn't know about B1. Note that B1 can only come up if the LOCONET segment is connected and powered.

GB1 detects that the LOCONET segment is up, which changes all "do not forward" markers in the routing table into "forward" markers.

GB1 now remembers that event Z currently must be forwarded across this link if seen.

B1 sends Z. GB1 receives it. Because GB1 hasn't seen Z before, Z is forwarded across the link to GA, followed by an "Identify Consumers of Z" message.

The rest of this proceeds like case 5.

Case 8: Special Interrogate Command is sent

The gateway is up, but doesn't know anything about segment B.

P1 comes up, and sends the Special Interrogate Command (8 Interrogate Switch Requests). The gateway can also issue this command by itself if GA receives the "Special Interrogate" event.

Nodes on segment B respond to this command by sending Turnout Sensor State messages <0xB1>.

GB1 receives each sensor message and proceeds like Case 5.

GB1 receives the first "Interrogate Switch Request", then checks the routing table for all "forward" markers GA →

Appendix: Theory of operation   *25*

GB1. For each "forward" marker that has message type "Turnout Feedback" or "Sensor Message", a message is generated on segment B. All further "Interrogate Switch Requests" will be ignored. A maximum of 10 messages will be sent every 100 milliseconds, in order to not overload the LOCONET bus.