RR-CirKits

Specializing in Affordable Electronics for Model Railroads

Installation Guide

Revision-f

August 2020

Manual for Firmware Rev-C5

Tower LCC

LCC (Layout Command and Control

16 line Input Output board

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

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 software version

Published August 2020.

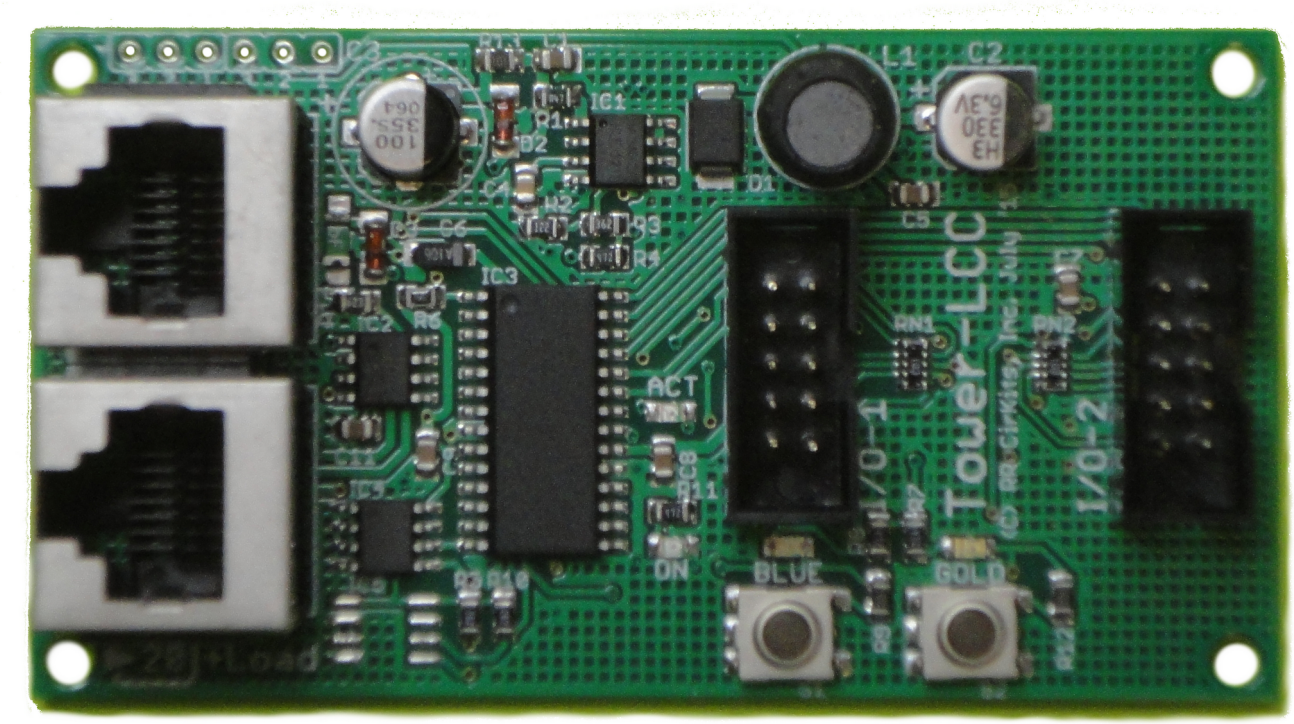
WARNING:	**This product contains a metal known to the state of California to cause cancer, birth defects or other reproductive harm. Do not ingest or otherwise abuse. Keep this product out of the reach of anyone not able to understand this warning, especially if they have a tendency to gnaw on things like printed circuit boards.**

Contents

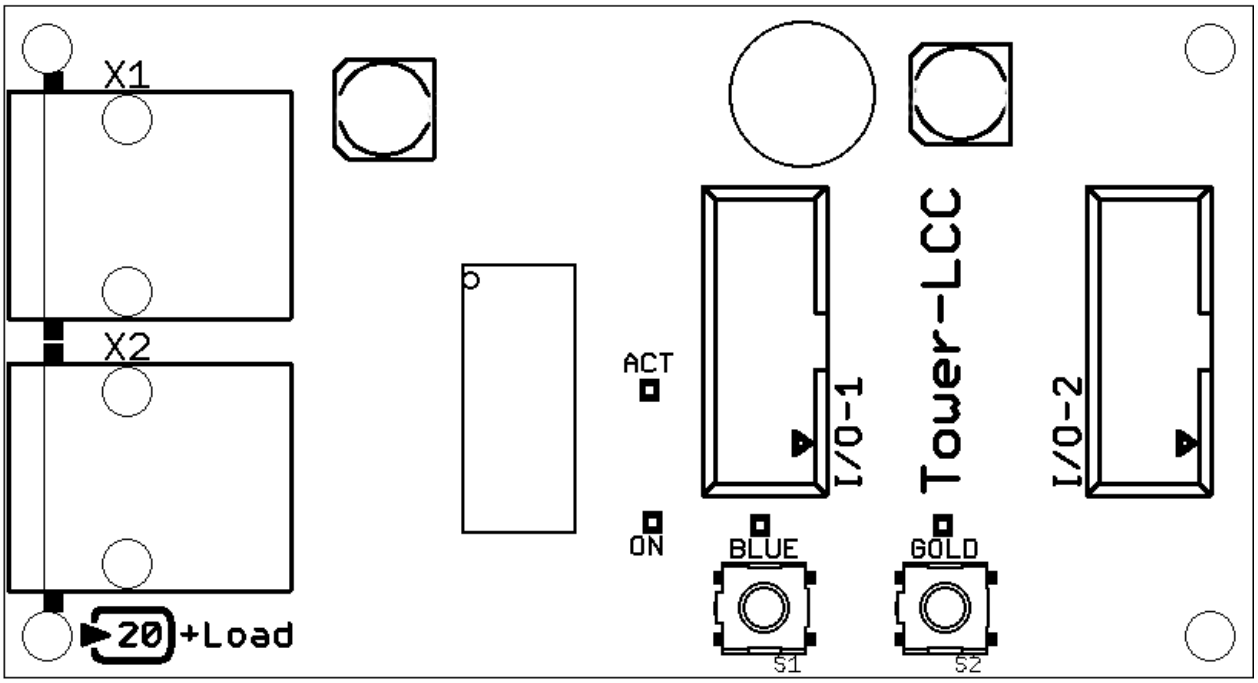
Overview

The Tower LCC (Layout Command & Control) interface provides a simple and easy way to connect between the NMRA LCC® CAN bus and the layout. The Tower LCC may be connected at any convenient point on the NMRA LCC® CAN bus.

LCC® is a registered trademark of the NMRA. www.nmra.org



Tower LCC Image



Tower LCC Connectors

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 events. Each event 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 event is that which you give it.

This event uniqueness is a key differences between the LCC and legacy systems. You can always create a link between any two points in the LCC world without needing to know anything else about the system. No more address conflicts, no more dedicating addresses to specific hardware, no more address space size limits.

The LCC uses 64 bit numbers to represent events, (18,446,744,073,709,551,615 possibilities) so we are not planning to run out of event numbers anytime soon. Events are created by event 'Producers' and used by event 'Consumers'. The same event may be created by one or more Producers, and may be used by any number of Consumers. (or none at all)

Events happen, they are not states nor the status of indicators. The only memory of events past exist in hardware. An event can tell you to turn a light 'on'. A different event can tell you to turn a light 'off'. Different events can tell you to turn the same light 'on'. However there is no event that tells you that the light is 'on'. That is a state, and only resides in the hardware that controls the light, or hardware that watches the state of the light.

1.1 Some Definitions

1.1.1 Node

On a CAN based LCC network there must be a minimum of two nodes. This may consist of a single node plus a computer interface, where the computer acts as a second node, or it may be two different nodes on the same logical segment. This requirement is due to the way that aliases, or unique identifiers, are generated for the limited packet sizes available on a CAN network.

We use the term ‘Node’ to indicate a single device or board that has both an electrical and logical connection to an LCC network. Some nodes may have multiple logical connections to the network, but only count as one node because there is only a single electrical connection. (transceiver) Some devices may have an electrical connection to the network, but not interact with the LCC logically in any way. An example might be a Repeater. It is electrically connected, but other nodes can not interact with it in any way. It does not count as an LCC node, but must be accounted for when counting the number of devices on a segment.

Please note that this is not exactly the same usage of the term ‘node’ as is documented in the NMRA LCC specifications.

1.1.2 Segment

On a CAN based LCC network there are electrical limitations on the total number of devices connected to the same cable, or ‘Segment’. These limitations take several forms. Electrical limits may be overcome by the use of a repeater.

- Electrical current limits. The CAT5 cable used has a limitation of 1A per conductor. The user is responsible to assure that sufficient power is supplied to the cable to supply all nodes within 20’ of a power injection point without exceeding this amount. Each node is marked with the amount of current required or supplied to assist the user in this calculation. Be sure to count external loads such as signal lamps, etc.
- Propagation delay. The speed of any CAN network is inversely proportional to its total length. The LCC CAN network was chosen to run a maximum of 1000’/300m at 125K bits per second. This is a good match to most layouts.
- Further, the maximum cable length is reduced by 20’/6m for each physical node attached to the segment. This limits any segment to about 48 nodes. The maximum cable length is shortened by the use of a Repeater.
- Each CAN segment should be a single serial string of nodes with a termination at each end. Short branches are allowed, but they count as double their length when subtracted from the segment total.

1.1.3 Line

Each Tower LCC contains 16 I/O lines. Each line has the ability to watch for 6 events (consumers) and to send out 6 events. (producers) Each line has two registers. One register remembers the 'State' of the line. (on or off) The second register remembers the state of the 'Veto' option. The veto option allows or disallows some events used to control or respond to the line.

1.1.4 Consumer

Each consumer event can be configured to control the line's output or veto register state in one of several ways. These are; 'on' (activate), 'off' (inactivate), 'change', 'veto on', and 'veto off'.

The hardware has an internal function generator that may be configured to create different types of actual outputs. These are; steady (output line follows the output state), blink (output line alternates when output state is 'on'), and pulse. (output line alternates one time when the output state is first ‘on’)

The function generator also includes delay settings for both 'on' and 'off' transitions. These delays can be used to control the blink rate and pulse length, or to simply delay the output action for some interval after the controlling event is seen. (e.g. to simulate “running time” on a CTC panel)

The consumer events may also control a Veto state. If the veto state is 'on', then the consumer 'Gated on' (activate) and 'Gated off' (inactivate) events are ignored unless in 'sample' mode. The consumer veto action is not active in 'Sample' mode because 'sample' mode implies that the output is always active.

The producer 'on' (not vetoed input), and 'off' (not vetoed input) are also ignored (blocked) when the veto state is 'on'.

1.1.5 Producer

Each producer event can be configured to trigger in one of 10 ways:

- ‘Consumer on' (activate), 'Consumer off' (inactivate), are trigger options that allow you to create a new producer event based on a command to activate or inactivate the output. E.g. to cascade a yard ladder.
- 'on' (function), 'off' (function), create a new event when the output of the function generator changes. This might be used to build a realistic traffic light controller. Be careful with this option because it can create a lot of traffic continuously, especially if the function output is blinking rapidly.
- 'on' (input), 'off' (input), allow you to create events simply based on a change of the input line. This is the normal use for a producer.
- Input 'on' (vetoed), or input 'off' (vetoed), allow events to respond to, or ignore, any input changes based on the veto state. For example this would allow you to enable/disable fascia buttons for local control of a turnout by using the output events from a panel switch to control the veto.
- Input 'on' (held), or input 'off' (held), allow you to delay any input change events based on the veto state. When the 'hold' is released the 'held' event will be produced. Note that the output state is not otherwise used when a line is configured as an 'input'.

The 'on'-'off' time delays are used as function output delay or input debounce delay depending on the line status.

The input line may be configured for normal response, alternate action response, or Sample.

- Normal response is used when an input change directly controls the sending of events.
- Alternate action response is used for a single line that produces alternating control events. (e.g. turnout normal, reverse)
- Sample is used for Touch Toggles or other dual mode situations where the input and output states of a line may not necessarily be the same. In Sample mode the line is normally driven by its output state, but briefly it disables the output drive and reads the un-driven state of the line. The input must be current limited for the case where the input and output are not the same. The normal output load must also be tolerant of brief changes during the input sample interval.

2 Tower-LCC Features

- The Tower-LCC uses the CAN bus implementation of the NMRA LCC.
- Communicates over the LCC CAN bus at 125Kb.
- Support for a total of 16 Input/Output lines:
  - Up to 16 Input Lines. (internal pull-up termination on all lines)
  - Up to 16 Output Lines.
- Internal Logic Blocks with up to 32 conditional statements.

- Support for up to 8 virtual code lines. (Track Circuits)
- CDI (Configuration Description Information) controlled programming via Software.
- Lines may be configured as TC-64 compatible ports or as individual lines.
- Automatically saves input/output and logic states during power down.
- Boot Loader allows contact less user firmware upgrades over the LCC® (Layout Command & Control) connection.
- Power is supplied over the LCC® bus. The TowerLCC requires 20mA. plus whatever load may be imposed by the I/O modules that you choose.
- Efficient switcher regulated 5VDC is available on each I/O port connector to power external modules or lamps.

2.1 Electrical Specifications

I/O Port 1:

Pin 1 – 5 volt logic level at 2mA. *
Pin 2 - 5 volt logic level at 25mA.
Pin 3 - 5 volt logic level at 2mA. *
Pin 4 - 5 volt logic level at 25mA.
Pin 7 - 5 volt logic level at 2mA. *
Pin 8 - 5 volt logic level at 25mA.
Pin 9 - 5 volt logic level at 2mA. *
Pin 10 - 5 volt logic level at 25mA.

I/O Port 2:

Pin 1 - 5 volt logic level at 25mA.
Pin 2 - 5 volt logic level at 25mA.
Pin 3 - 5 volt logic level at 25mA.
Pin 4 - 5 volt logic level at 25mA.
Pin 7 - 5 volt logic level at 25mA.
Pin 8 - 5 volt logic level at 25mA.
Pin 9 - 5 volt logic level at 25mA.
Pin 10 - 5 volt logic level at 25mA.

Maximum current to be supplied by all I/O lines combined is 200mA. This 200mA total limit means that not over 8 lines may supply their maximum current at any one time.

*note: Pins 1, 3, 7, and 9 of Port 1 have a greatly reduces output drive capability and should normally be used as inputs when ever possible, or else be used to control external drivers.

3 Power and Serial Connections

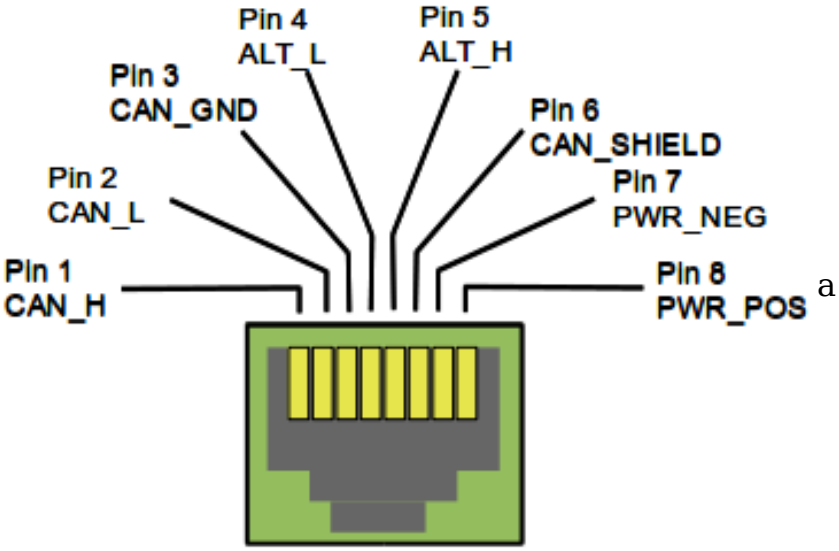
The Tower LCC (16 Line I/O Board) has four connectors and four status indicators. Two of these connectors are for connections to the LCC bus network. The other two are used as connections to the I/O lines. This section covers the system connections consisting of the CAN bus port connectors, power connections, I/O port connections and Status indicators.

3.1 CAN LCC® Compatible Connector

The data connection is made to the Tower LCC via a standard RJ-45 CAT5 cable connected to either of the two RJ-45 connectors. The LCC wiring passes straight through both connectors.

The LCC specification requires minimum of 1’ of cable length between connectors. Slightly shorter cables (10”) should not significantly impact operation.

These cables are commonly sold for wired Ethernet use.



a

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

Pin	Description
1	CAN H
2	CAN L
3	CAN GND
4	Alt L (DCC negative)
5	Alt H (DCC positive)
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 Power-Point delivers approximately 15VDC to the bus.

The LCC connectors accept standard Ethernet style CAT5 (or better) cables. 4 pair cables are required by the Tower-LCC. 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 Tower LCC requires an external power source of between 7.5 and 27 volts DC from the LCC cable.

The LCC Power-Point unit is a convenient way to supply the required power to the Tower LCC and other LCC boards over standard RJ45 cables.

Each segment of LCC® cable requires a terminator at each end. Power can also be supplied by other powered LCC modules, or with the RR-CirKits LCC Repeater.

LCC Power-Point shown with Terminator

3.3 Status Indicators

The Tower LCC has two status indicators located near to the LCC connectors. The green ON status indicator shows the power status of the Tower LCC itself. The red ACT (activity) indicator normally shows all data activity on the bus, and also any activity/error status during a boot loader firmware upgrade. (see section 10.0)

3.4 Blue/Gold Buttons and LEDs

A limited amount of configuration may be accomplished on some manufacturer's nodes by using the Blue and Gold push buttons and indicators. The primary use is to link up producer and consumer lines. The Tower LCC does not support this option.

The Gold LED can indicate two different error messages. If it is flashing (10% duty cycle) it indicates that it is idling in forced boot loader mode. If the Gold LED is blinking (50% duty cycle) it indicates that the board was unable to initialize itself on the network, most likely because it could not establish an alias.

3.4.1 Setting up Virtual Code Lines

Use the CDI tools to setup links. The EventID for the TX code set will always come from the transmitting node and be entered into the receiving node to avoid accidental reuse of EventID numbers from show to show.

Each node can setup one or more virtual code lines to any other node. For simplicity these virtual links are named 'Track 1', 'Track 2', etc. It is incumbent upon the user to keep track of which 'Coded' virtual links are created between nodes. Be sure to record which block (1-8) is used for each side of the virtual links

if you have not standardized these connections. There is no need to use the same 'block' number on both sides of any virtual coded track circuits, and in fact they will not normally be matching. Normally these virtual links will follow along with the rails, but there is no actual requirement that they do so.

3.4.2 Master Clock Adjustment

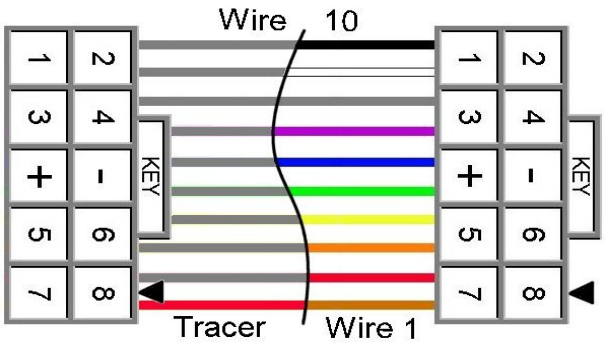
The other current use for the blue/gold buttons is to put the node into clock calibrate mode. Hold both of the buttons down for 10 seconds until the blue and gold LEDs start to flash rapidly. A 100Hz signal will appear on the Aux port pin 1. Use a calibrated frequency meter or digital scope to measure it. If it is off frequency you may ‘tune’ the master clock by pressing either the Blue or Gold buttons to raise or lower the master frequency. The frequency should be within 0.1% or better for optimal CAN bus length. This frequency is temperature corrected from -10C to 75C.

3.5 Tower LCC I/O Connector Wiring

The two port connector's wiring is as follows. Note that the pin numbers and I/O line numbers are NOT the same, and actually run opposite to each other.

Pin number	Connection name
1	line 8
2	line 7
3	line 6
4	line 5
5	Ground
6	+5VDC
7	line 4
8	line 3
9	line 2
10	line 1

10 position IDC cable



4 Getting Started

We suggest that you use a computer program such as the JMRI DecoderPro (4.14 or later) <http://www.jmri.org/> or Deepsoft LCC-CDI <http://www.deepsoft.com/> MRS to configure the Tower LCC. 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 suggest for configuring the Tower LCC node.

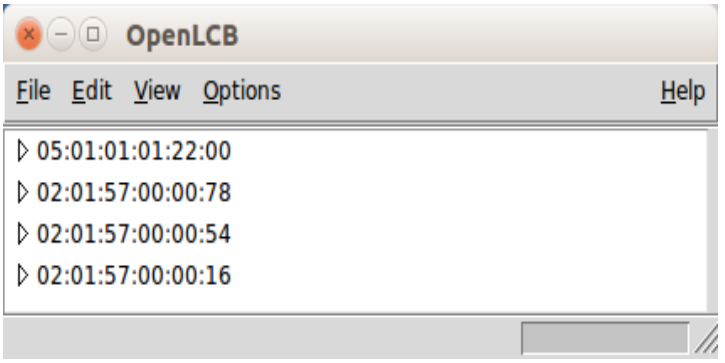
Node Address: Each Tower LCC has a single node address that is used for CDI programming on the layout. Each individual Tower LCC has its node address imprinted on a label on the back side of the board. It is recommended that you name your node as the first step in configuration.

4.1 CDI (Configuration Description Information)

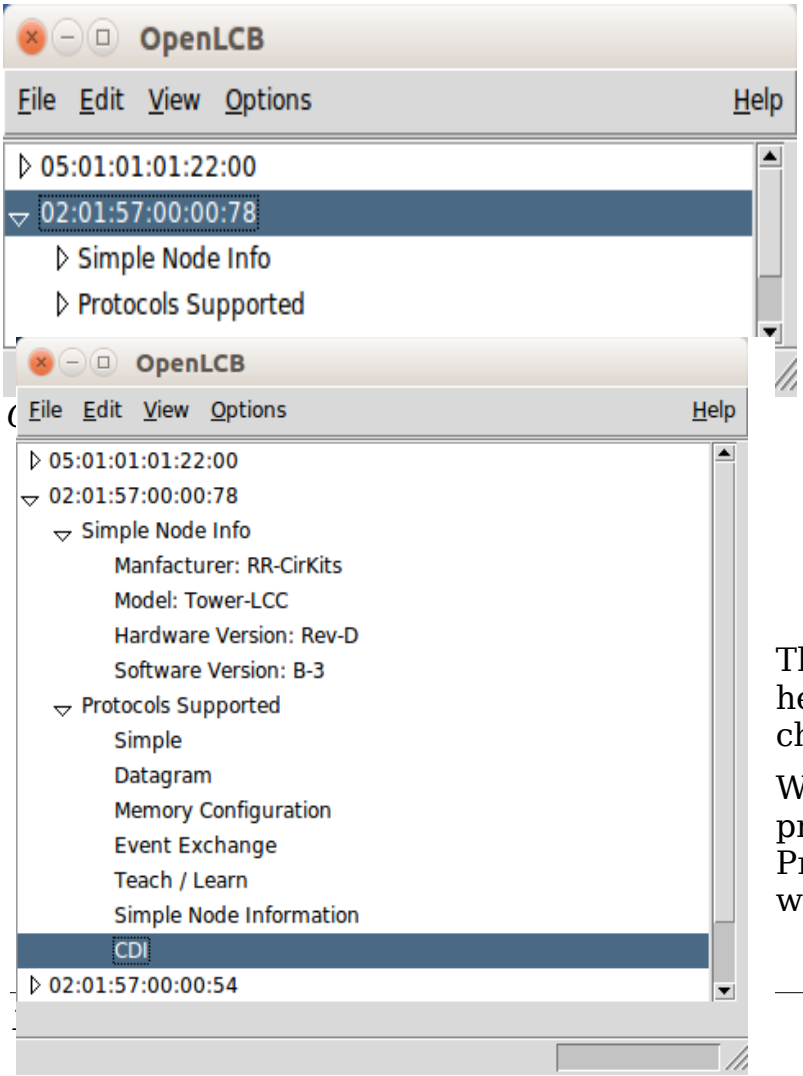
The CDI is the way that LCC nodes store their configuration options internally. Instead of relying on printed manuals or volunteer created files to present the various decoder options, (like DCC devices have for the past 20 years) the LCC specification expects the manufacturer of the LCC node itself 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.

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 should include 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.



CDI Window



Highlight 'CDI' to open it in a new window.

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.

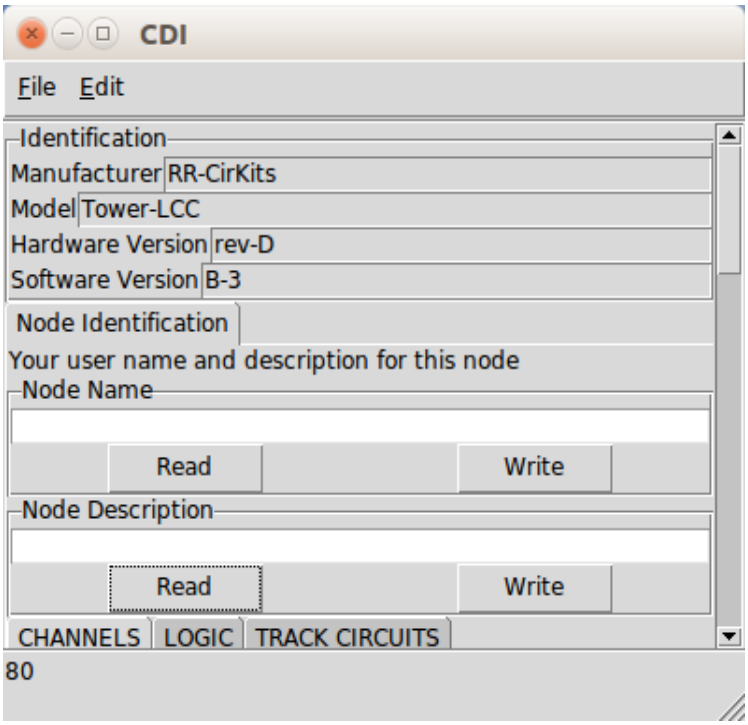
With the Deepsoft OpenLCB program you simply highlight a Protocol item to open it in a new window.

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.



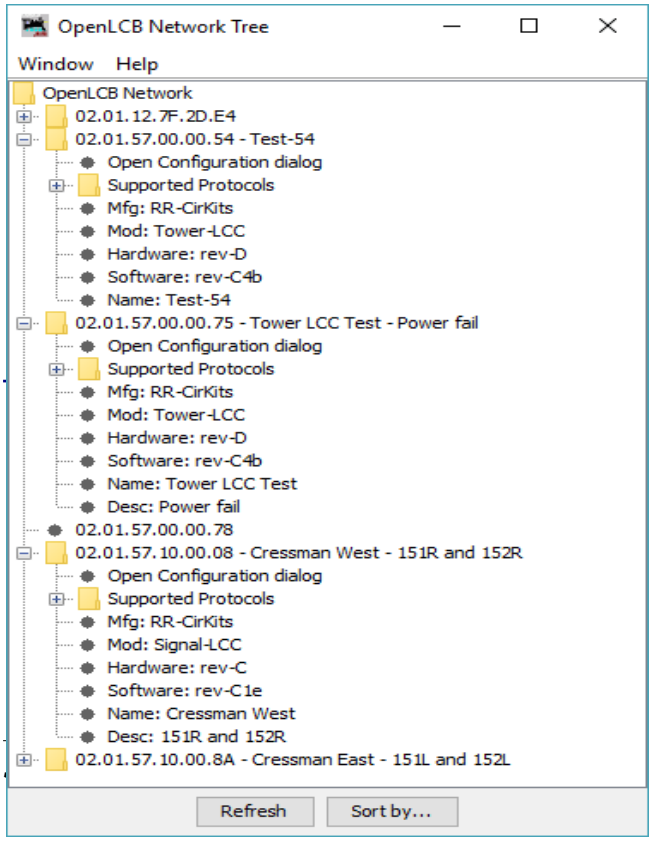
Deepsoft CDI Window

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 ‘Read All’ button located at the bottom of the window. Be forewarned that it can take a long time to fill in all of the options and values stored in the node, so you may want to use discretion with this option.

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.

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.



Open the JMRI CDI Window

5 Input/Output Configuration

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

The following examples are using the JMRI CDI tool for the Tower LCC. Select the ‘LCC’ drop down list and click on

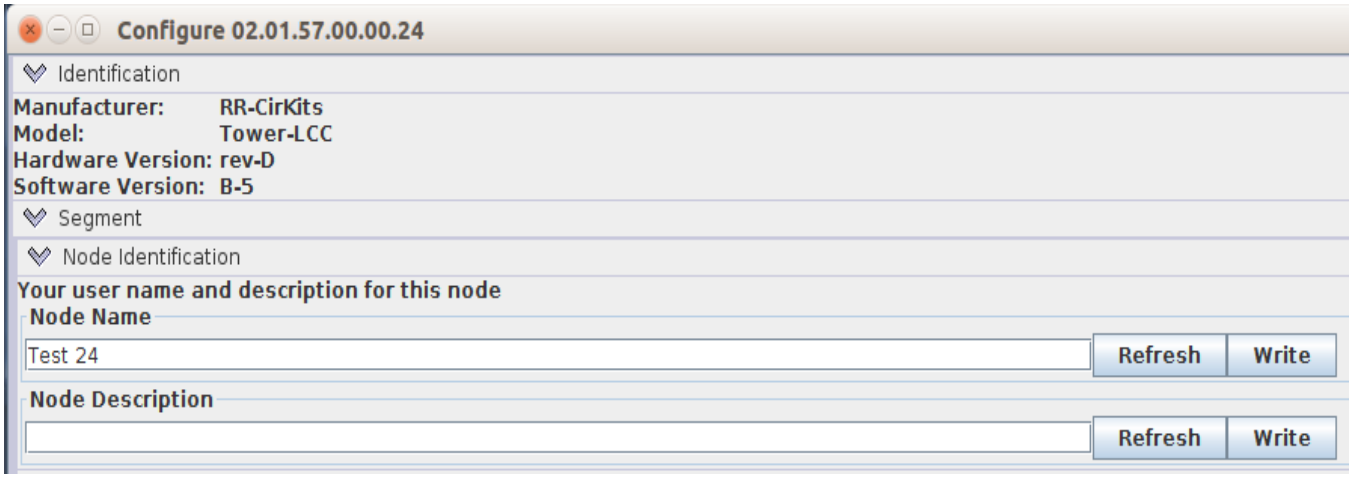
‘Configure Nodes’. When the node selection window opens, choose the node to configure, click on ‘Open Configuration dialog’.

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.

5.1 Identification

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



JMRI CDI Window

5.2 Node Identification

The next item is the Node Identification. It contains the Name and Description that you give to the node. The name of this node is ‘TEST 24’. This name will appear in the node selection window to make it easier to select the correct node for configuration. There is a 64 character limit to the node name and description items.

5.3 Line (I/O Ports)

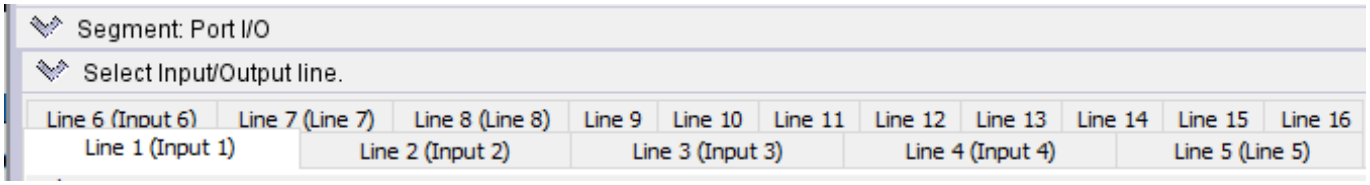
The Tower LCC has two 8 bit Input/Output ports for a total of 16 lines. Each port is normally configured to be either all Inputs, or all Outputs, to be compatible with the various RR-CirKits I/O modules. However each line may be individually set as either input or output for special purposes.

For the special case of one wire I/O a line may even be configured as both input and output at one time. (Sample Mode section 1.2) The Berrett Hill Touch Trigger is an example of a one line device. The Tower LCC can both control the indicator's color using consumers and report the output using producers in the same channel.

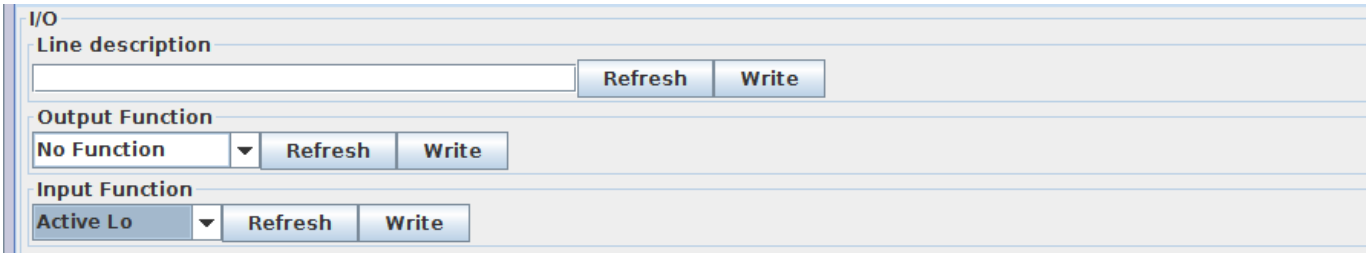
Any special effects may be applied differently for each line. E.g. one line may be held steady while another sends a pulse or is blinking.

5.3.1 Lines

Each I/O line is called a 'Line' and is selected by tabs and presented separately.



5.3.2 I/O Configuration



For example in the above snapshot I have selected the Line #1 'Output Function' as 'No Function' and its 'Input Function' as 'Active Lo'. The Output function options are:

- No Function – no output action
- Steady Active Hi – the output voltage follows the output state
- Steady Active Lo – the output voltage is the inverse of the output state
- Sample Steady Active Hi – the input is sampled and the output voltage follows the output state
- Sample Steady Active Lo – the input is sampled and the output voltage is the inverse of the output state
- Alt. Sample Steady Active Hi – the input is Alt sampled and the output voltage follows the output state
- Alt. Sample Steady Active Lo – the input is Alt sampled and the output voltage is the inverse of the output state
- Pulse Active Hi – the output voltage pulses + when the output state is true
- Pulse Active Lo – the output voltage pulses - when the output state is true
- Sample Pulse Active Hi – the input is sampled and the output voltage pulses + when the output state is true

- Sample Pulse Active Lo – the input is sampled and the output voltage pulses - when the output state is true
- Alt Sample Pulse Active Hi – the input is Alt sampled and the output voltage pulses + when the output state is true
- Alt Sample Pulse Active Lo – the input is Alt sampled and the output voltage pulses - when the output state is true
- Blink A Active Hi – the output voltage blinks phase A + when true
- Blink A Active Lo – the output voltage blinks phase A - when true
- Blink B Active Hi – the output voltage blinks phase B + when true
- Blink B Active Lo – the output voltage blinks phase B - when true

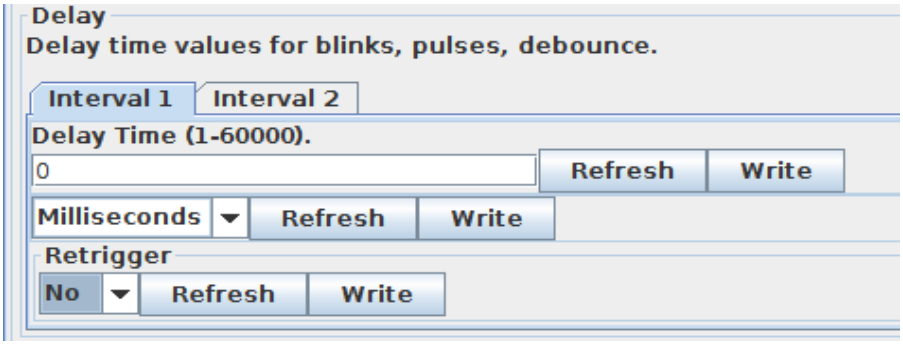
The input function options are:

- Disabled – no input action
- Active Hi – the input is set true when it goes high
- Active Lo – the input is set true when it goes low
- Alt Action Hi – the input alternates true/false when it goes high
- Alt Action Lo – the input alternates true/false when it goes low
- Sample Hi – the input is set true when it samples high
- Sample Lo – the input is set true when it samples low
- Alt Sample Hi – the input alternates true/false when it samples high
- Alt Sample Lo – the input alternates true/false when it samples low

A line must only have output functions or input functions enabled. When one of the 'Sample' options is selected it will automatically enable the opposite function. Note that these configuration options relate to the true or false status of the line, not the actual output voltage levels on the lines or events consumed or produced. These are controlled in the individual event settings.

For safety, any input function will over ride an output function so you must always be sure to set an input to 'Disabled' in order to use it as an output.

5.3.3 Delay



Each line includes two delay timers that are used to control blinks, pulses, and input debounce times.

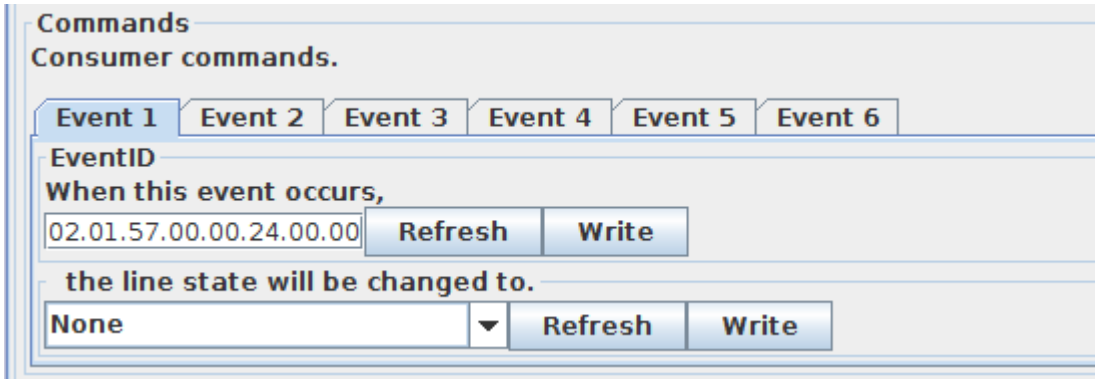
Interval 1 controls the 'on' delay or time, and Interval 2 controls the 'off' delay or time. The count may be set from 0-60,000 and the base interval may be set to Milliseconds, Seconds, or Minutes. This allows for delays from 1ms. to over 41 days. Probably the extremes will never be required, but this gives you a good range to choose from. Actual accuracy is ½% or better. Millisecond times are calculated to the nearest 8 ms.

Retrigger allows the time interval to be reset if the line state is set to ‘true’ again prior to the end of the delay time.

If the line state is set to ‘false’ prior to the end of the delay time, then the output does not occur.

5.3.4 Commands

Commands are consumer events that are sent to the line. A command can directly control the line, for example by turning it on or off. A command may also indirectly control the line, for example by controlling its veto status. Each line has 6 consumer events associated with it. Each Consumer event is associated with a single task or action.



For example event 1 could be used to turn an output ‘on’, and event 2 could be used to turn an output ‘off’. The consumer events are:

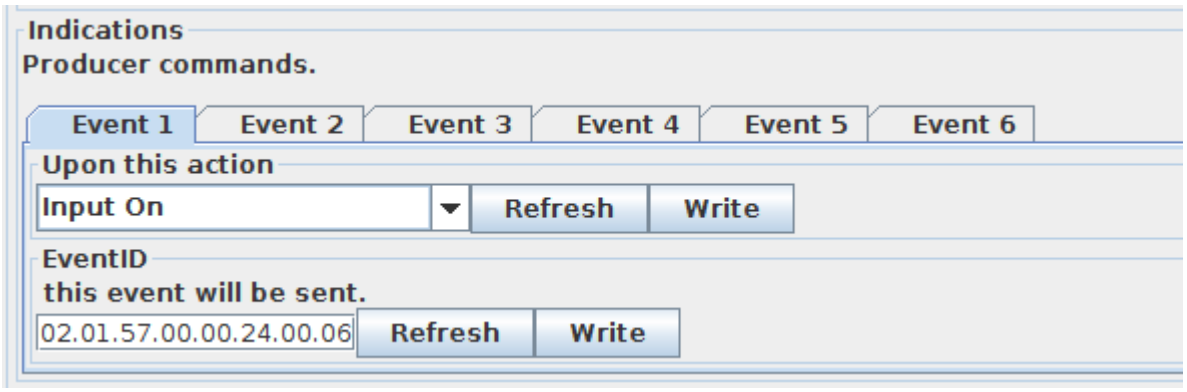
- None –
- On (Line Active) – Sets the output state to ‘true’
- Off (Line Inactive) – Sets the output state to ‘false’
- Change (Toggle) – Changes the output state to the opposite state
- Veto On (Active) – Sets the veto state to ‘true’
- Veto Off (Inactive) – Sets the veto state to ‘false’
- Gated On (Non Veto Output) – Sets the output state to ‘true’ if veto is ‘false’
- Gated Off (Non Veto Output) – Sets the output state to ‘false’ if veto is ‘false’

The last two items probably need some clarification. An event set to ‘On’ will always set the output state to ‘true’. However an event set to ‘Gated On’ will only set the output to ‘true’ if the veto state is ‘off’. If the veto state is ‘on’ then the ‘Gated On’ and ‘Gated Off’ events will be ignored.

For example when a CTC operator controls a turnout he would send events configured as ‘On’ and ‘Off’. However a local operator button would send (different) events configured as ‘Gated On’ and ‘Gated Off’. The CTC operator could then send a ‘Veto On’ event that would block the local operator from controlling the turnout, but still allow normal operation from his own panel.

5.3.5 Indications

Indications are producer events that are sent to the bus. Each line has 6 producer events associated with it.



Each Producer event is associated with one action. For example event 1 could be used to indicate that the input line is ‘active’, and event 2 could be used to indicate that the input line is ‘inactive’. The consumer events are:

- Output State On command – Responds to an output state change to ‘true’
- Output State Off command – Responds to an output state change to ‘false’
- Output On (Function hi) – Responds to an output level change to ‘high’
- Output Off (Function lo) – Responds to an output level change to ‘low’
- Input On – Responds to an input level change to ‘high’
- Input Off – Responds to an input level change to ‘low’
- Gated On (Not Veto Input) – Gated response to an input level change to ‘high’
- Gated Off (Not Veto Input) – Gated response to an input level change to ‘low’

5.3.6 Tower LCC Secondary Messages

In these examples there are no second or third messages being sent in the sense of our previous products. However additional messages may be sent or responded to if desired by utilizing unused events. For example to activate a 'Next' turnout whenever the 'first' turnout event is sent, in order to sequence a yard ladder, you could use the ‘Output State On command’ to send an event to the next turnout in the ladder. These additional messages, if enabled, are sent whenever the primary event occurs. Another simple example would be to allow two or even three different messages to be sent when ever a single button is pressed. For example a single event could be the master reset for many turnouts simply by adding it as an additional ‘On’ or ‘Off’ event to each turnout in the group.

6.0 Tower LCC Logic

In addition to its 16 I/O lines, and virtual track code lines, the Tower LCC also includes 32 logic conditionals. This logic is event driven, not state driven. The logic blocks may be used to create signaling logic or other animations such as control of grade crossings. It may also be used to create NX (eNtry Exit) routing.

6.1 Logic Conditionals

Each conditional statement of this logic table consists of an identifier and two memory locations (called variables 6.2.1) that remember the last command given for any items that they are 'watching', a logic operator or function, and a delay timer. Each conditional may generate (produce) up to four events when it evaluates to true, when it evaluates to false, or optionally after any delay is complete. These ‘Variables’ allow decision making, even over power failures.

To calculate this logic, the Tower LCC keeps a list of all events that can change the state of any variable. Any time one of these events is seen on the bus, a pass is made through the Logic table. Each variable is checked for a match to the new event, and changed if appropriate. Unless set to ‘None’, any change, or optionally a match, will reevaluate each logic conditional containing the variable. If the conditional is found to be ‘true’ then up to 4 action (producer) events will be sent.

Each conditional is assigned a group type.

- (0) – Blocked
- (1) - Group
- (2) – Last (Single)

Groups of one or more conditionals, terminated by a 'Last (Single)', form a logic block. A logic 'block' consists of one, or several, statements all related to the same mast or logic calculation.

Each logic block is treated as if it were a group of statements independent of any other groups., but in fact each group is checked, but if no events match the new event it will be passed over.

6.1.1 Mast Conditionals

‘Mast conditional’ logic statements are designed to make the calculation of signal aspects easy to do. Groups of mast conditionals are always configured with the most restrictive aspects first, then in increasing speed order to the least restrictive aspect.

Evaluation always begins starting from the first to the last conditional in the group.

Actions (indication messages) only get sent when an evaluation causes a change in mast aspect, either to a more restrictive or a less restrictive one.

When ever any conditional in the mast group evaluates to ‘true’, but without any aspect change, then the processing is skipped forward past the next ‘End Item’, and no changes are sent.

Starting with Rev C1 firmware the exit format of each conditional may be specified as:

Action when True	**Action when False**
Send then Exit group (default)	Send then Exit group
Send then Evaluate Next	Send then Evaluate Next
Exit group	Exit group
Evaluate Next	Evaluate Next (default)

If a default 'Mast Group' conditional evaluates to 'Send' and causes an action, then its associated delay is triggered. The Action itself may be conditioned in several ways:

Action
None
Immediately
After Delay
Immediate if True
Immediate if False
Delayed if True
Delayed if False

By evaluating as ‘Send’ on both true and false exit actions and then applying true or false conditions to the actions, it becomes possible to send different actions based on the True/False exit conditions.

If a conditional evaluates to ‘Exit group’, then the evaluation process is ended for that group by skipping forward past the next ‘Last (Single)’ item. Once the delay runs to completion then its associated events are sent. (produced) From one through four events may be sent. This allows for the creation of basic multiple lamp signal head control.

If no conditional in a 'Mast Group' evaluates as ‘Send’, then the evaluation simply proceeds to the next logic item with no resulting event being produced.

A default indication message may be sent by placing a ‘null’ (true) conditional at the end of a ‘Mast Group’. It will be triggered if no previous conditional in the group was true. Note: an adverse effect of this may result in the repeated sending of a default indication whenever any event for another mast in the same node is received. To reduce generation of spurious events, avoid using this option if possible.

6.1.2 Last (Single) Item

This group function marks the end of a block of mast conditionals containing one or more items to be evaluated in order. Once a conditional in a mast group is evaluated as ‘Exit’ then processing skips ahead past the next ‘Last (Single) Item’. If any conditional is the last one in a block of conditionals then it should be identified with this function to allow this to happen.

6.1.3 Ladder Group

Unlike a ‘Mast Group’, where all conditionals are always checked for a match to any variables that match the current event, a Ladder group may be created by setting each conditional to ‘Send then Evaluate Next’. Any matching variable can then trigger its actions. This allows for the easy creation of routes with more than 4 output events by including additional identical 'true' conditionals following the first evaluated one.

6.2 Variables

Each variable has two events (consumers) associated with it. The first event enables the variable, (true) and the second disables it. (false)

Any time a variable changes state, the evaluation of its conditional may result in conditional action events. Optionally the evaluation of any conditional may be disabled.

Each conditional statement contains one or two variables for identifying the required events in order to do simple signal interlocking or route generation. Any logic statement evaluates to true only if both its function and variables are true. A ‘null’ function entry is considered to be true if evaluated. (but will not itself trigger any evaluation) Thus a ‘null’ entry placed at the end of a mast logic block will always cause a default indication message to be sent if no previous conditional statement within the same group has been evaluated to 'true'.

A single variable may be used to trigger a conditional by setting the Logic Function to ‘V1 Only’ or ‘V2 Only’.

The state of all variables is remembered internally in order to compare them. An event or change in either variable's state may optionally trigger a new comparison. This dual variable capability makes it easy to calculate a signal aspect or train direction based on turnout position and occupancy.

6.2.1 Variables and 'Not'

Each variable is controlled by a pair of events, one that sets it 'true' and the other which sets it 'false’. To invert the sense of any variable in a logic function, reverse the positions of the two controlling events for that variable. If you have 2 events, 'Crack of Dawn' and 'High Noon' that are used to control the logic 'Its morning', then use 'Crack of Dawn' to control 'true' and 'High Noon' to control 'false'. However if your logic needs to be 'Its **not** morning' then use 'High Noon' to control 'true' and 'Crack of Dawn' to control 'false'.

6.3 Logic function

The following are the available logic operators:

Logic Function
V1 AND V2 = 'V1' AND 'V2' is true
V1 OR V2 = 'V1' OR 'V2' is true
V1 XOR V2 = 'V1' XOR 'V2' is true
V1 AND V2 => change 'V1' AND 'V2' is true has changed state (any change triggers a single evaluation of the group)
V1 OR V2 => change 'V1' OR 'V2' has changed state (any change triggers a single evaluation of the group)
V1 AND Then V2 => First 'V1' is true AND THEN 'V2' becomes true. This special operator makes it easy to determine train direction. A V2 change to true triggers a single evaluation of the group only if V1 is already true. The V1 state is retained so that further changes to V2 true may re-trigger the evaluation.
V1 Only => V1 is True
V2 Only => V2 is True
null => True => This entry can be used to send a default event anytime the evaluation of a group is triggered, but no other entry evaluates as true. Of course it must always be the last entry in a group if it is used.

6.4 Delay

The delay time may be set with a resolution of minutes, seconds, or milliseconds. (0-60,000) That gives you resolution enough to create real time day/night lighting controls for your buildings. (or layout room) The 'Re-trigger' option controls if a running time delay gets reset when it is re-triggered before completion. If so, then the time delay period starts over again. If not, then the re-trigger event is ignored, and the timer continues to run to its original completion. To 'kill' the output of a timer already in progress you disable a timer's output by setting the triggering function to no longer be 'true'. Millisecond values are rounded to the nearest 8 ms value.

6.5 Producers

Any logic conditional may generate up to four events when it is true. Each of these four events may be; ignored with ‘none’, sent ‘Immediately’, or sent ‘After delay’. Each may be be conditioned with the True or False condition that sent it.

7 Track Circuits

The prototype railroads have the need to send signal indication information from one signal to the next in order to calculate the proper aspects to display. These calculations are done locally, not at the dispatcher's office nor by some central computer. This indication information can be sent over local wires from one signal to the next, but that means lots of infrastructure to maintain. Fortunately there are always one pair of conductors that need power on them for detection circuits and that automatically go to the right place. That is the rails themselves. Even from the earliest semaphore days some basic indication information was passed over the rails simply by switching the polarity of the battery feeding the DC track circuit being fed from one end of a block to the other.

As signaling became more complex and speed signaling was introduced, there was a need to pass more information over the same two rails. Genrakode® and Electrocode® are two brands of equipment that do this by sending a series of pulses rather than just DC polarity from one end of the block to the other. These circuits are normally bi-directional, so the transmit and receive circuits switch from one direction to the other every few seconds.

7.1 Simulating a Code Line with Events

To send this indication information from signal to signal over the usual EventIDs is difficult because each link needs to be specifically made using different EventID numbers for each indication. That makes it difficult or impossible to determine ahead of time what all the EventID numbers will be for a modular setup.

One solution to this dilemma is to use events in a way similar to those used in coded track circuits and then allow nodes to learn these groups of internally generated events simply by using the CDI tools.

These internal event groups are then translated into normal user programmable EventIDs for use in Logic and connections with other nodes and panels. To link up these connections at a modular meet the operators would simply link the signal nodes at each side of a module boundary to create these virtual groups of links automatically.

7.2 Linking Virtual Code Lines

To link these virtual code circuits use the CDI to copy the key from one track circuit to another. The EventID for the TX code set will always come from the transmitting node and be entered into the receiving node to avoid accidental reuse of EventID numbers from show to show.

Each node can setup one or more virtual code lines to any other node. For simplicity these virtual links are named 'Track 1', ‘Track 2’, etc. It is incumbent upon the user to keep track of which 'Coded' virtual links are created between nodes. Be sure to record which block (1-8) is used for each side of the virtual links if you have not standardized these connections. There is no need to use the same 'block' number on both sides of any virtual coded track circuits, and in fact they will not normally be matching. Normally these virtual links will follow along with the rails, but there is no actual requirement that they do so.

7.3 Prototype Code Line

In our solution to this problem we follow the prototype with a few changes. When each prototype code is sent onto the rails there are four data items simultaneously sent in each packet. They are the ‘Start’ bit, (1) the 'Speed' or 'Indication' code, (2, 3, 4, 6, 7, 8, or 9), the '5' code, (5) and the 'M' code. (M) In order to make this solution more 'EventID' friendly we represent each code with two events, one to turn it on and the other to turn it off. This allows us to ignore the ‘Start’ bit.

TABLE 5-1 CODE RATE AND ASPECT

CODE RATE	ASPECT
7	Clear
4	Advance Approach
3	Approach Limited
8	Approach Medium
2	Approach
9	Approach Slow
6	Accelerated Tumble Down
5	Non-Vital code indicating track occupancy, or a hand-throw switch in the block out of normal correspondence
M	Non-Vital code indicating power off in the block, or a lamp out condition in the block. Power Off will indicate from the east end CP, lamp out from the west end CP

Note: the prototype can only send indications via an unoccupied block, because the train itself is shorting the rails going toward the signal ahead when it is occupied. On the model we do not have that restriction.

We can send either ‘Stop’ (5) or ‘Tumble down’ (6) as an indication. This simplifies the coding of APB signals.

To limit the number of indication EventIDs to 16 we have omitted the ‘M’ code as being unnecessary.

7.4 ABS and APB Signal examples

For examples of using logic to control signals see the Signal LCC manual. The Tower LCC logic may be used to expand a Signal LCC logic table if more statements are required.

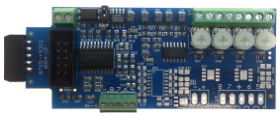
8 Tower LCC compatible Input/Output Cards

The RR-CirKits Tower LCC and its compatible I/O modules are designed to be clipped into Tyco 3-1/4" Snap-Track® mounted to the bench work. (Snap-Track® is a plastic channel designed to mount PC cards to a chassis, not something to run trains on.)

A single Tower LCC or compatible I/O module fits into the 3TK2-1 (single) mounting track. Other widths are available for compact installations using multiple boards.

Each I/O module is equipped with two connectors to facilitate these I/O board connections. Use IDC connectors and ribbon cables to connect the Tower LCC to the I/O cards.

8.1 BOD-4 (DCC Block Occupancy Detector - 4 block plus 4 I/O)



This board operates as a DCC block occupancy detector for 4 blocks using remote CT coils. It outputs logic levels, and has a RR-CirKits standard ribbon connector interface. The "Power-Lok" feature optionally monitors the DCC bus power. A power failure latches the detection status of each block until power is restored and re-stabilized. There are also 4 general purpose I/O connections fed through to the driver board.

8.2 BOD4-CP (DCC BOD 4 block, 4 Inputs, plus 2 turnout drivers)

This board operates as a DCC occupancy detector for 4 blocks using remote CT coils. It outputs logic levels, and has a RR-CirKits standard ribbon connector interface. The "Power-Lok" feature optionally monitors the DCC bus power. A power failure latches the detection status of each block until power is restored and re-stabilized. The CP version also includes dual turnout drivers. When used with the Tower LCC or Signal LCC boards there are also 4 general purpose I/O connections available using the Sample options.

8.3 BOD-8 (DCC Block Occupancy Detector - 8 block)

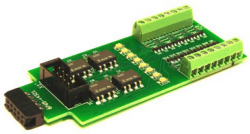
The BOD-8 does not expect you to re-wire your layout to bring track feeders to the detector cards. The small CT (Current Transformer) detection coils are placed directly on the track feeders where they belong. Simple lengths of Cat-5 cable are the usual way to run the signals back to the detector boards. Use of CT coils means that there are no track voltage losses associated with the detectors. Normal detection levels are 1mA. but may be adjusted to higher levels with on board pots.

During a DCC bus power failure the Power-Lok input on the BOD-8 instantly locks the current state of each block detector. I.e. the state of the layout does NOT change during a DCC power outage, neither to all occupied, nor to all vacant. It just suspends sending any occupancy changes until after power is restored and things have stabilized again. If you do not want the feature there is a jumper to disable it.

The BOD-8 outputs are low during detection so the Tower LCC should be configured accordingly.

It is planned to build a ‘Detector LCC’ board that will combine the LCC interface and a detector card.

8.4 OIB-8 (Opto Isolator Board - 8 input)



This 8 input board is used when a non-isolated source of voltage needs to be monitored and input to the Tower LCC. One example would be to monitor the DCC voltage on a set of points to determine the position of a turnout without using auxiliary contacts.

This board may be configured to monitor the absence or presence of an AC or DC signal. This board requires 10mA. for reliable operation and includes built in current limiters.

8.5 SCSD-8 (Single Coil Solenoid Driver)

The SCSD-8 Output Module is designed to drive individual solenoid coils or other high voltage high power devices. Normally the input voltage should not exceed 27VDC. The SCSD-8 board is optically isolated from the driving circuitry to protect the Tower LCC or other control device from the high power outputs. When driving single coils or high power loads configure the line as a steady output.

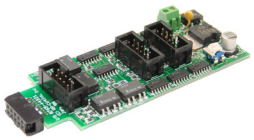
By using the proper options on the Tower LCC the SCSD-8 may also be used to control dual coil momentary switch machines. In 'Dual Coil' mode the output lines must be paired such that the pair of lines requires just single address pair. However reverse the two EventIDs. This action will normally require a 0.1 second pulse when driving solenoids.

The lines are paired and only the primary event of the first line of each pair will be used to trigger a pulse.

Dual coil operation should not be attempted if the switch machine power supply is not of the capacitive discharge type that will limit the long term current to a low value in case of hardware or configuration errors.

Failure to observe this precaution may result in destruction of equipment and be a fire hazard!

8.6 SMD-8 (Stall Motor Driver – 8 line)



The SMD-8 board contains 8 individual, optically isolated, H-Bridge drivers. This allows the board to be powered from any supply between 9 Volts and 27 Volts. It is primarily designed to drive stall motor turnout machines such as those found in Tortoise® and Switchcraft® machines . Do not exceed 20VAC or 27VDC at the power input.

This board includes an adjustable buck switching regulator to allow you to control the speed of your switch machine motors. This regulator can not boost the drive voltage above the supply voltage.

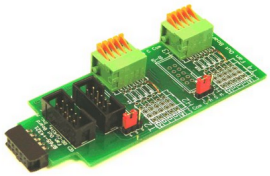
8.7 RB-4 (Relay Board - 4 x SPDT)

Relay Board - 4 is a Quad 10A SPDT relay board with logic level drivers. It is suitable for use with Tower LCC or other logic level output devices. It requires 12V auxiliary power to drive the relay coils. Auxiliary power is optically isolated from the logic inputs for double isolation. LED indicators for each relay make it easy to monitor activity.

Includes dual ribbon connectors with offset lines to allow easy connection as output 1-4, or output 5-8, of the Tower LCC, or other driver.

The RB-4 input lines are active low so all lines on this Tower LCC port should be configured appropriately. This inverted input mode matches most types of driver outputs, and the drive polarity may be easily switched either in the Tower LCC configuration or by reversing the RB-4 output contacts.

8.8 FOB-A (Fan Out Board)



This board is a convenient way to convert from 10 pin ribbon cable to compression terminals. It may be used for inputs or outputs, and includes pads for mounting resistors such as may be required for connecting LEDs used for direct drive signals or panels.

For input lines jumper wires will need to be installed for each line. Do not exceed 5V on any input or the Tower LCC will be damaged.

8.9 BOB-S (Break Out Board – Screw Terminal)

This board is a convenient way to convert from 10 pin ribbon cable to screw terminals. It may be used for inputs or outputs.

Do not exceed 5V on any input or output or the Tower LCC will be damaged.

The BOB-S may be mounted to a panel or stringer using #4 or smaller screws and spacers.

9 Trouble shooting

9.1 Sanity Test

To perform a very basic Tower LCC sanity test perform the following steps:

- Power up the Tower LCC by plugging it into a powered network.
- The green power LED should come on.
- If no other board is present on the network, then the gold light will begin to flash at 50% while the board seeks to establish an alias.
- The previous output states should be automatically be restored.

If the green power LED does not light, be sure that a power supply is connected to the LCC network segment, and provides at least 7.5V to the Tower LCC. The green power LED will initially light at much lower voltages, so it is not a reliable indicator of suitable power.

9.2 Activity Test

The Tower LCC's input circuit and code sends data directly to the unit's processor, so if you send any command to the unit it should immediately be seen on the command (COM) 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 output line on this Tower LCC. The command should appear in the LCC® monitor window and the Tower LCC command (Y) LED should blink.
- The connected output should respond.

If there is activity at the LCC Terminator blue LED, but no activity light at the Tower LCC 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 Tower LCC. 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.

A blinking yellow lamp on a single board may indicate the failure to complete the alias check. Note that a minimum network will have two nodes, or a single node plus a computer connection.

The Tower LCC is initially configured as simple input lines. You may use a RR-CirKits I/O test board to send events simply by connecting it to either input port and pressing the test buttons.

10 Boot Loader

10.1 Boot Loader Upgrade

Note: The Boot Loader upgrade is only required for nodes #02.01.57.00.00.76 or earlier that have not yet had their boot loader upgraded. The upgrades make the firmware update process work better with JMRI and other programs.

!! IMPORTANT !!: If you have JMRI open, please close it.

1) Restart JMRI V4.20 or later and do not open the 'OpenLCB > Configure Nodes' menu under any circumstance.
2) Select OpenLCB > 'Firmware Update'.
3) From the 'Target Node ID' drop down box, select the TowerLCC node to be updated.
4) Click Select to pick a firmware file.
5) From the file menu, select this file: 'bootloader_V11_fix_for_B3.hex'.
6) Click "Open' and leave 'Address Space' at '239'; do not check 'Lock Node'.

7) Now click 'Load' to download the new boot loader V1.1.

During the download the Gold led will blink to show that the node is in 'Boot Loader State'. (10% flash)

The progress bar on screen will now fill up to 100%, on the node the Blue and Red led show bus activity.

After some time the message 'Download completed successfully' should appear.

The Gold led will continue to blink, because the node will remain in 'Boot Loader State'.

Do not close the 'Firmware Downloader' window at this time, leave it open.

From the OpenLCB menu, click 'Configure Nodes' and select the (partially) updated node.

It should now show 'Mod: Tower-LCC Bootloader' and 'Software: V1.1'.

Do not close the 'OpenLCB Network Tree' window at this time, leave it open and proceed to 10.2 (3).

10.2 Firmware Upgrade

Note: Version C and later CDI files store the Virtual Track Circuit information in a different format than version B. You will need to re configure this section after an upgrade. Do NOT restore a configuration saved from an earlier firmware version. It will destroy your virtual track circuit information.

If an update to your Tower LCC firmware is needed, a program such as "Firmware Update" in JMRI version 4.14 or later is required.

To enter Firmware upgrade mode:

1) Start JMRI and select "OpenLCB".
2) Select ‘Firmware Update’ from the OpenLCB drop down list.
3) Select your 'Target Node ID'. If you have just completed the boot loader upgrade it should still be selected.
4) Click ‘Select’ to pick a firmware file.
5) From the file menu, select: 'TowerLCC_C4b_UPDATE.hex' or the latest upgrade available.
6) Optionally you may check the ‘Lock Node’ check box to take it off line during the upgrade.
7) Click the 'Load' button to initiate the upgrade to Tower-LCC revision C4b.
8) Wait until 'updating device firmware..' is complete.
9) Switch back to the OpenLCB Network Tree window.
10) It should now show 'Model: Tower-LCC' and 'Software Version: rev-C4b'.
11) 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.

You may need to adjust the master clock rate if you are upgrading the boot loader and firmware from nodes #02.01.57.00.00.76 and earlier for the first time. (see section 3.4.2)

11 Grounding and Isolation

Unlike the LCC Buffer-USB, the Tower LCC is not optically isolated from the LCC bus. This allows for possible ground loop problems between the LCC® and your layout accessory power supplies, so be sure to keep the ground connection to the Power-Point either isolated, or else in common with your layout power source.

Normally all Tower LCC connections will originate or reference to the Tower LCC board itself, so there is no danger of ground loops with these connections. RR-CirKits High power output boards are optically isolated from the Tower LCC ports and use their own power sources.

If you are building your own I/O boards or using third party units be sure to observe the common/isolated ground rules, and never exceed 5V on any I/O pin.

Properly ground your boosters, your power supplies, and your desktop computer through a 3 wire cable, and isolate them from each other via isolated equipment where necessary.

12 Warranty Information

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

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

13 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 Tower LCC 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