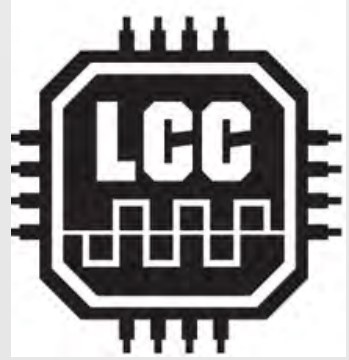


What Is It?

Legal Statement, requested by the NMRA General Counsel



Use, in this clinic, of the LCC logo and LCC information does not mean that the NMRA has reviewed this clinic or approved it in any way.

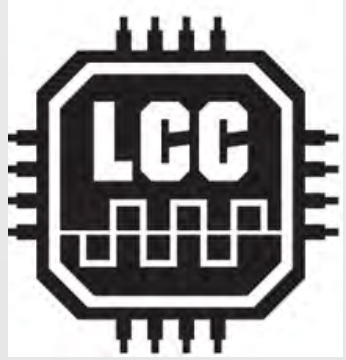
While there may be information in this clinic that appears in the NMRA standards and recommended practices, I have not obtained any of this clinic from that source material.

The LCC logo is a registered trademark belonging to the NMRA.

Its use without permission from the General Counsel is improper.

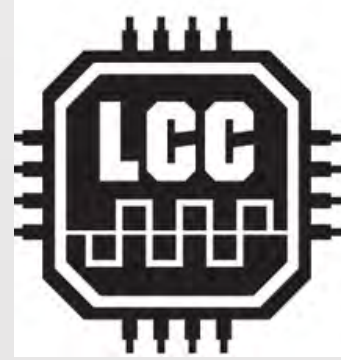
I have received permission for use in this clinic.

LCC is an NMRA-Proposed Solution



The NMRA decided a decade ago to call for the creation of an open (license free) method to interface to your layout.

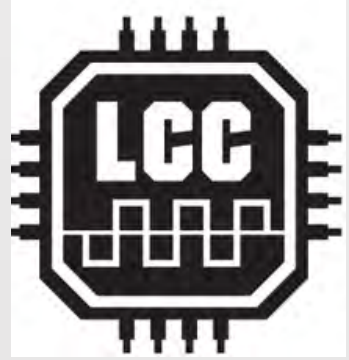
LCC is an NMRA-Proposed Solution



The intent was, like the NMRA DCC standards, that many manufacturers would be able to build layout accessory products that will interchange as freely as is now true for DCC mobile decoders.

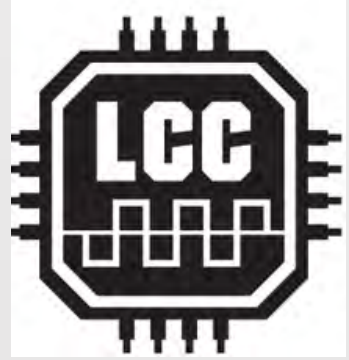
NMRA-Proposed LCC Requirements

- The bus must use license free commercial standards for its communications as much as is possible.

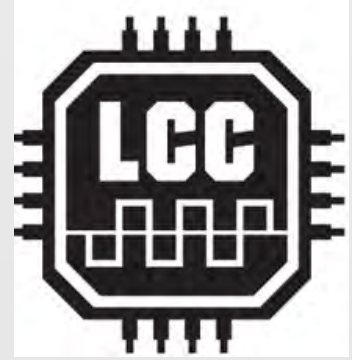


NMRA-Proposed LCC Requirements

- The bus must use license free commercial standards for its communications as much as is possible.
- It should be robust and viable into the next generation of electronic products.

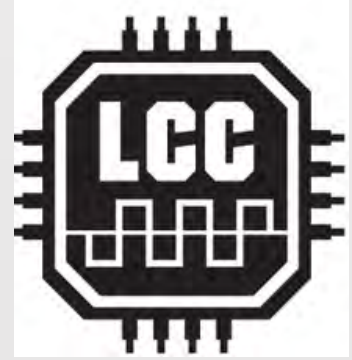


NMRA-Proposed LCC Requirements

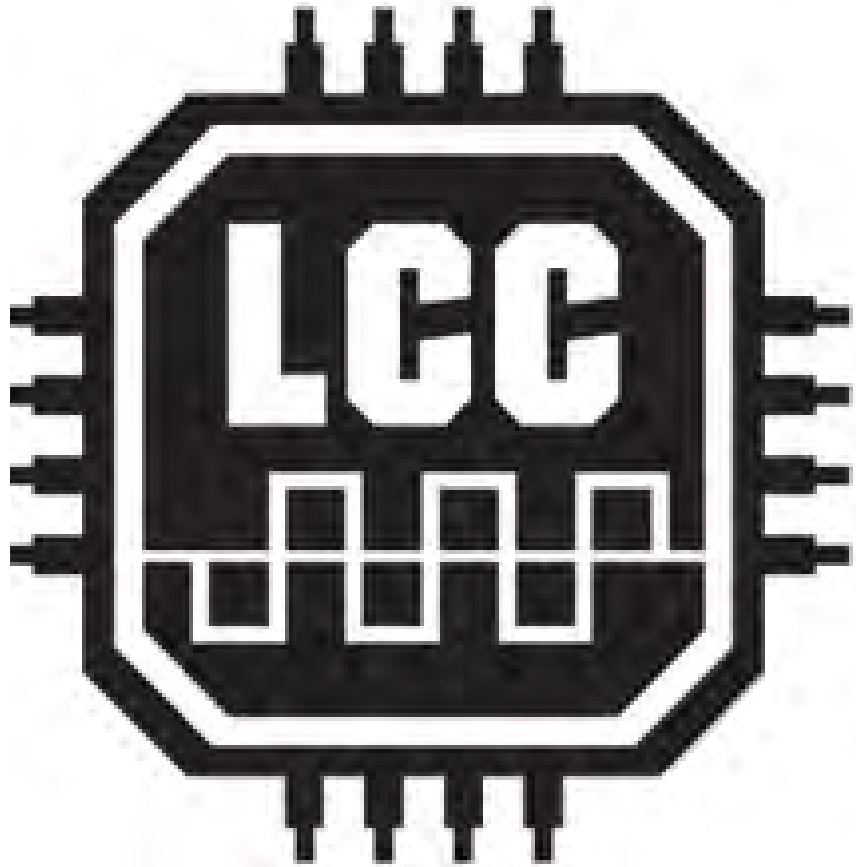


- The bus must use license free commercial standards for its communications as much as is possible.
- It should be robust and viable into the next generation of electronic products.
- **It should be a peer-peer design with no requirements for any central control.**

NMRA-Proposed LCC Requirements



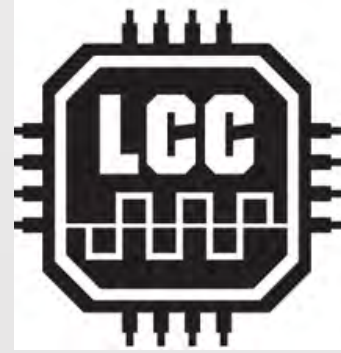
- The bus must use license free commercial standards for its communications as much as is possible.
- It should be robust and viable into the next generation of electronic products.
- It should be a peer-peer design with no requirements for any central control.
- **Any two devices from any manufacturers must be able to exchange data.**



What Is It?



LCC is an Information Superhighway for Accessories and Their Controls on Model Railroad Layouts



LCC provides a common language for Accessories to talk to each other

LCC is Designed for Layout Functions:

Turnout Motors

Block Detection

Signals

Road Crossings

Building Lights

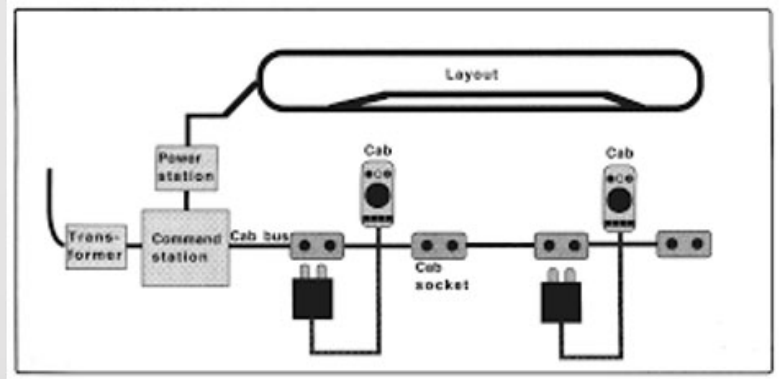
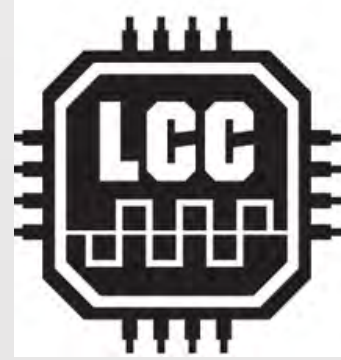
Ambient Lights

Arduinos



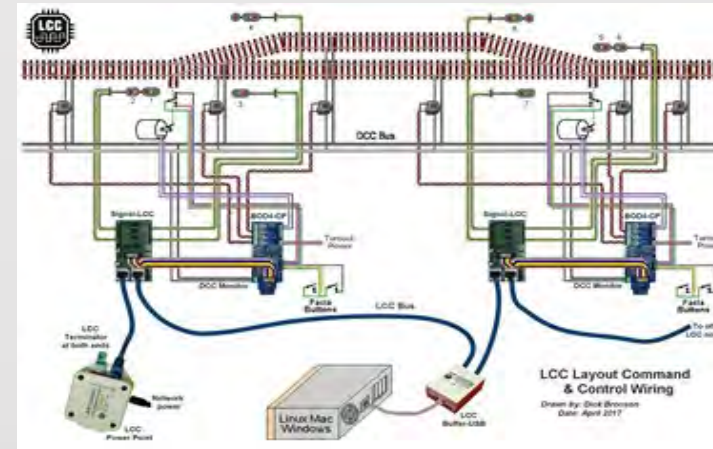


DCC and LCC are Complementary



● DCC Operates the Train

- Powers the track
- One-way signal to decoders
- Each decoder is independent

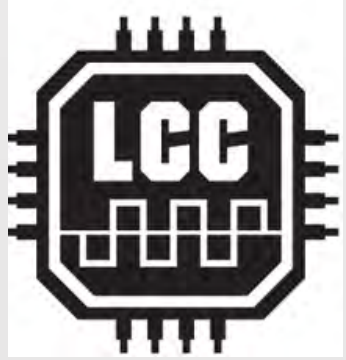


● LCC Operates the Layout

- No connection to track
- Two-way communication
- All “nodes” have all information



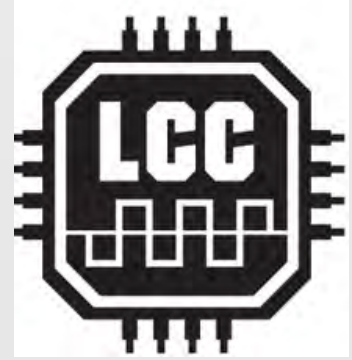
DCC and LCC are Complementary



- **On the Track - DCC**



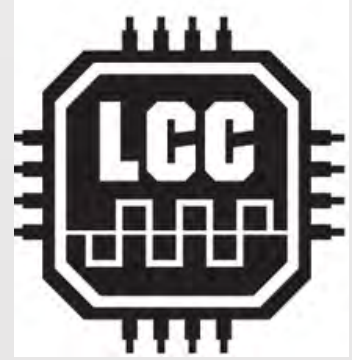
DCC and LCC are Complementary



- On the Track - DCC
- **Beside the Track – LCC**



DCC and LCC are Complementary



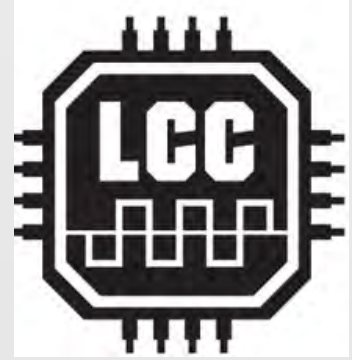
- On the Track - DCC
- Beside the Track – LCC

Crossovers between On and Beside the Track

- LocoNet, CMRI, XpressNet, MERG,
plus others are Proprietary Systems



DCC and LCC are Complementary



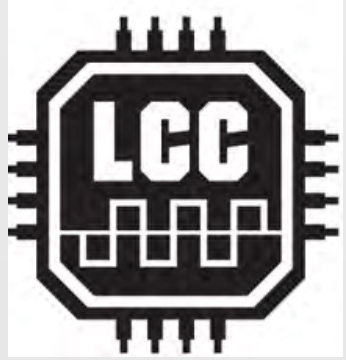
- On the Track - DCC
- Beside the Track – LCC

Crossovers between On and Beside the Track

- LocoNet, CMRI, XpressNet, MERG,
plus others are Proprietary Systems
- **DCC is a Master-Slave system with**
Fixed Address Space



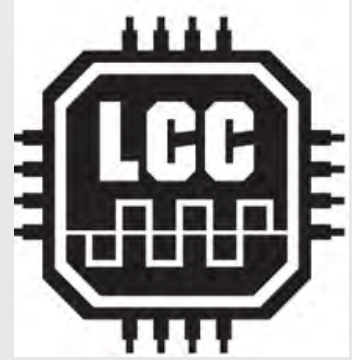
DCC Is Designed for Operating Trains



- DCC provides digital information with the power in the rails



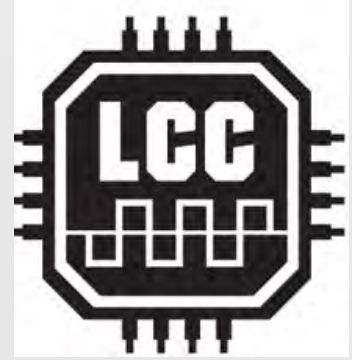
DCC Is Designed for Operating Trains



- DCC provides digital information with the power in the rails
- **Command Station communicates with each throttle (or other input device via a proprietary networks)**



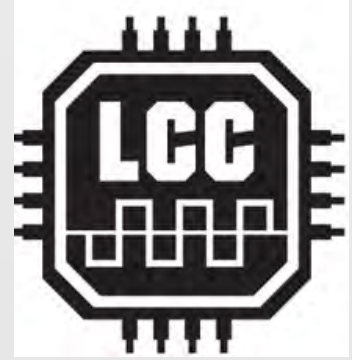
DCC Is Designed for Operating Trains



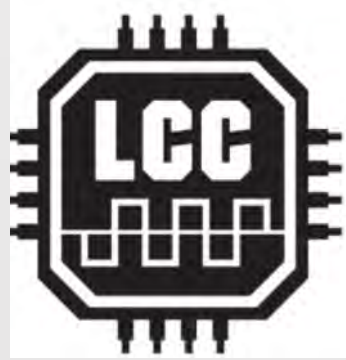
- DCC provides digital information with the power in the rails
- Command Station communicates with each throttle (or other input device via a proprietary Network)
- **Command Station then sends signal to engines (or other DCC Decoder equipment using NMRA standard DCC)**
 - **Each decoder receives all signals and reacts only to those sent to it's address**
 - **Does not provide digital feedback that it has done anything**



DCC Is Designed for Operating Trains



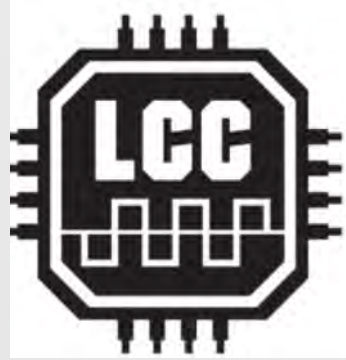
- DCC provides digital information with the power in the rails
- Command Station communicates with each throttle (or other input device via a proprietary Network)
- Command Station then sends signal to engines (or other DCC Decoder equipment using NMRA standard DCC)
 - Each decoder receives all signals and reacts only to those sent to it's address
 - Does not provide feedback that it has done anything
- **Sound equipped engines use more data communications**
- **As more trains (or stationary decoders) are added, the response rate to new commands (think horn/whistle) slows**



LCC Can Run on Layouts Controlled By:

- AC
- DC
- DCC
- DCS
- TMCC
- RailPro
- Battery power
 - etc

Building LCC Is Like Building with LEGO's

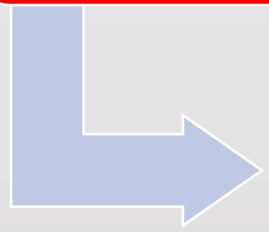


There are three basic building "blocks"

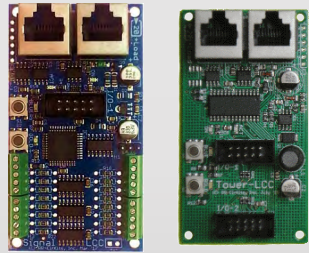
Interface



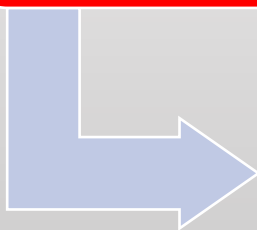
Provides interface to Computer:
Configure LCC Cards
Communication with JMRI
Cat 5 Cable connects to LCC System



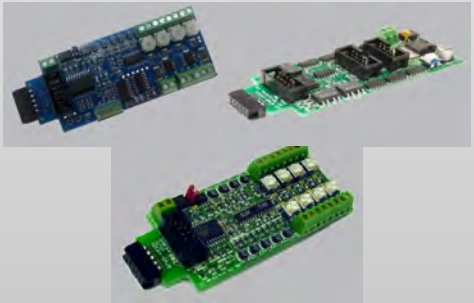
"Brains"



Tower or Signal Cards
provide computation
and communication
daisy chain with Cat 5



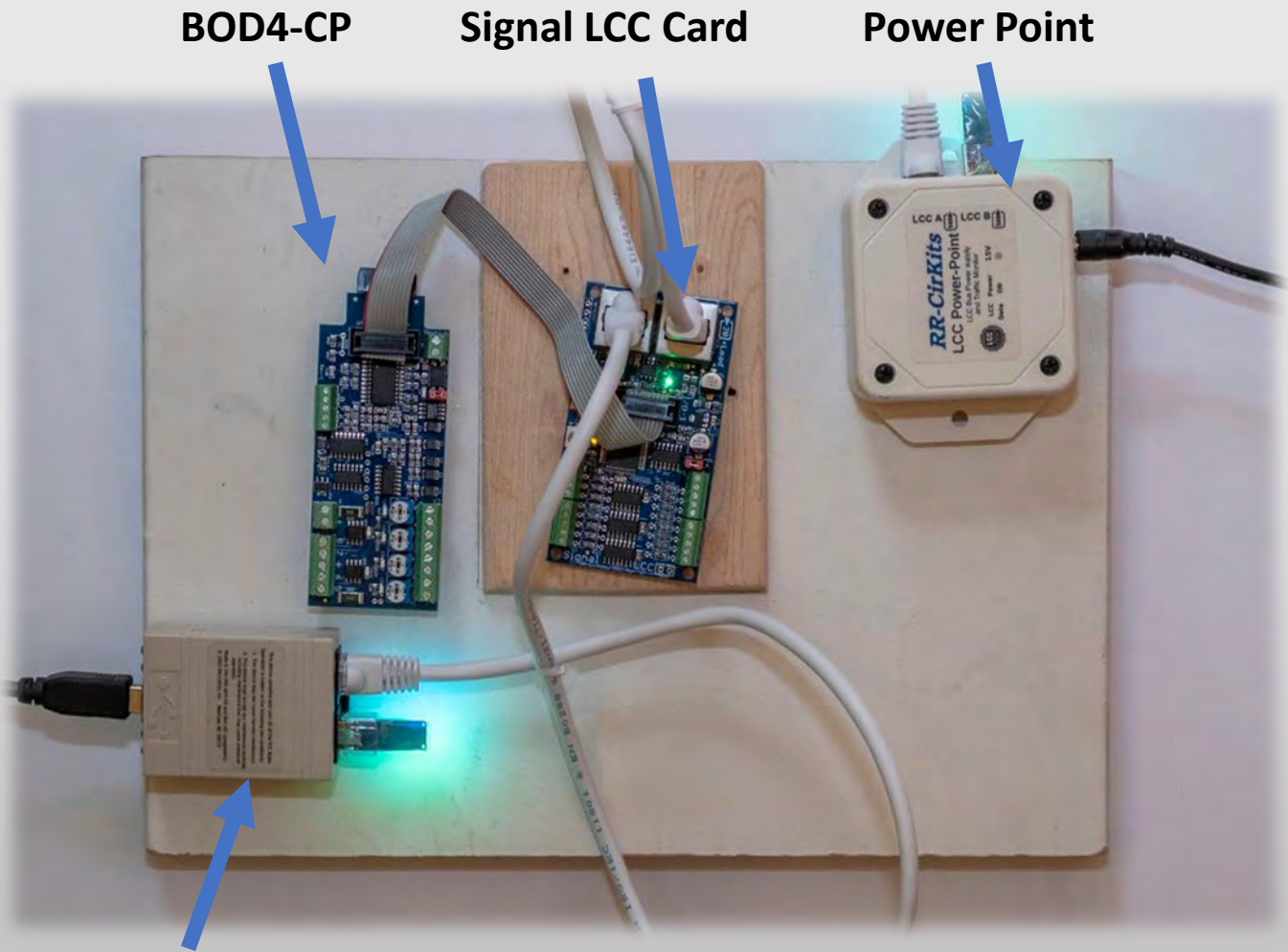
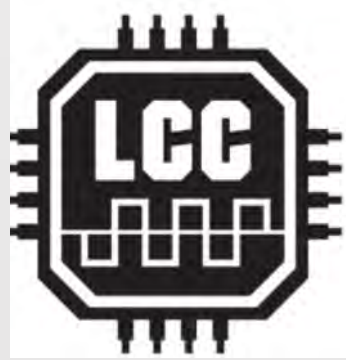
Function



Functions are "Daughter Boards" to Tower or Signal Cards
Operate Turnouts, Detect Occupancy, or Auxiliary Equipment

Connect with 10 wire flat cable to Tower or Signal Card

The World's Simplest LCC System



BOD4-CP

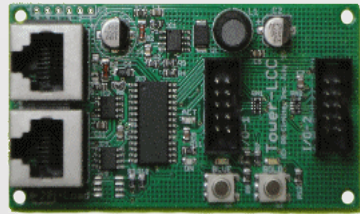
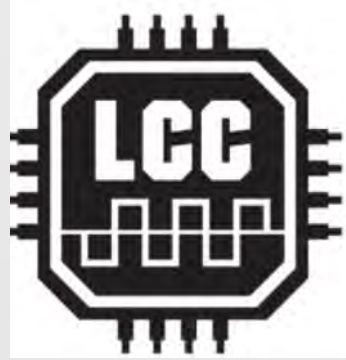
Signal LCC Card

Power Point

LCC Buffer
Computer Interface

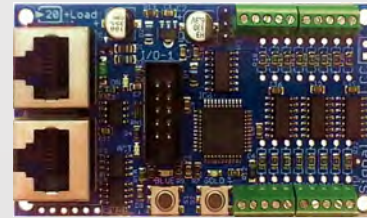


The “Brains” of LCC are in Two Cards



• Tower LCC

- Use this card for most applications
- 16 Line Input/output node for NMRA CAN bus LCC
 - Communicates on LCC Buss via daisy Chain Cat5 Cables
 - It will support two “daughter boards”

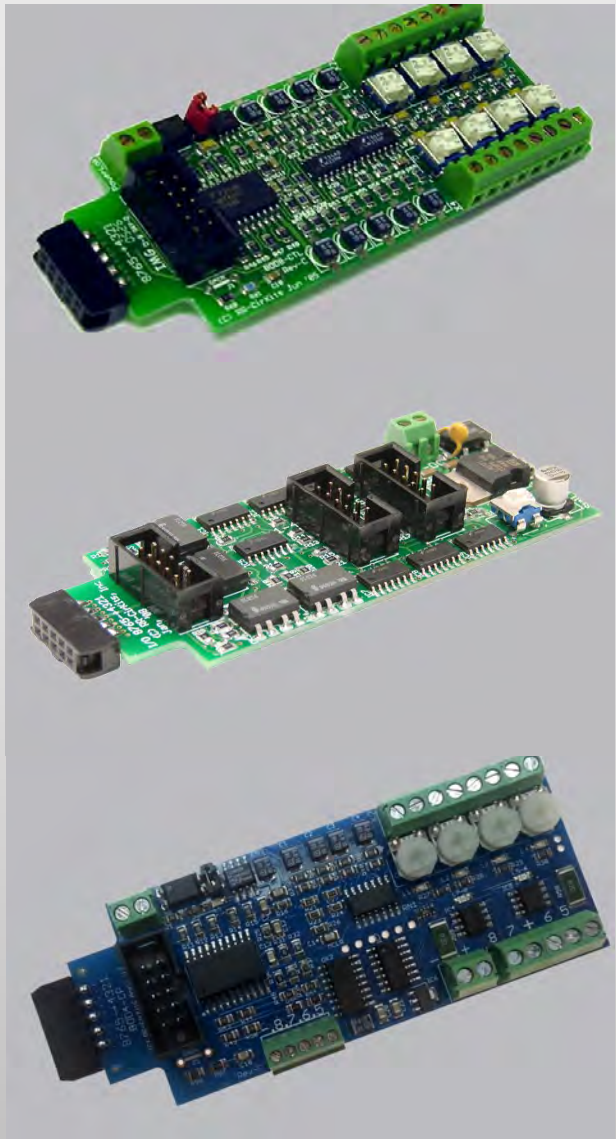
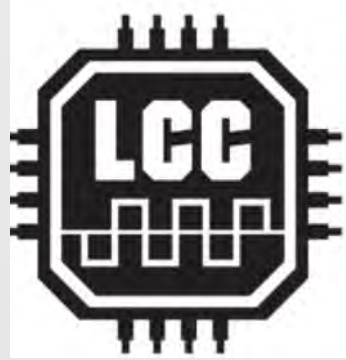


• Signal LCC

- Use this card if you want Signals
- 16 Led drivers plus 8 line Input/Output node for NMRA CAN bus LCC[®]
 - Provides LED control to up to Four Signal Masts
 - It will support one “daughter board”



Functions Happen on "Daughter" Boards



BOD8

Block Occupancy Detector

- This board operates as a DCC occupancy detector for 8 blocks using remote CT coils.
- It outputs logic levels, and has a RR-CirKits standard ribbon connector interface.
- The "Power-Lok" feature monitors the DCC bus power. A power failure latches the detection status of each block until power is restored and re-stabilized.



SMD8

Stall Motor Driver

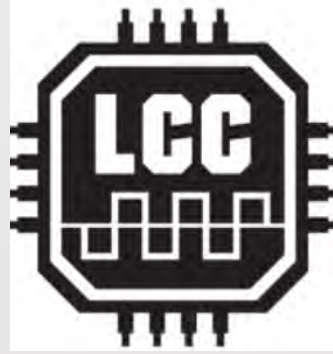
- This 8 output, optically isolated, low current "H" bridge driver is designed for control of 8 stall motor switch machines. (E.g. Tortoises®)
- It can drive up to 100ma. per line, speed regulated output 4 to 12VDC.
- (PTC fuse limited to 200ma total per board for safety)

BOD4CP

Block Occupancy Detector-4

- 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.

The BOD4-CP (Control Point)



Use of "Plug In" is not recommended



Use "Flat Cable" to Connect to either Tower or Signal Card



Common Ground
Sense #4 Gnd 8
Sense #3 7
Sense #2 6
Sense #1 5



Track Power from Same Block as Detection Circuits



All Blocks must be from the same Circuit Breaker

Turnout #2 8
7
-
6
Turnout #1 5

1
2
3
4

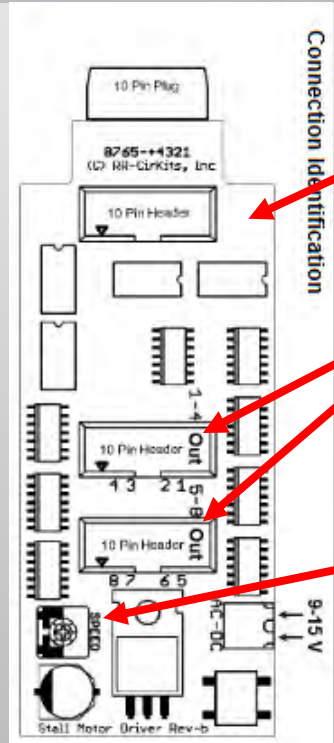
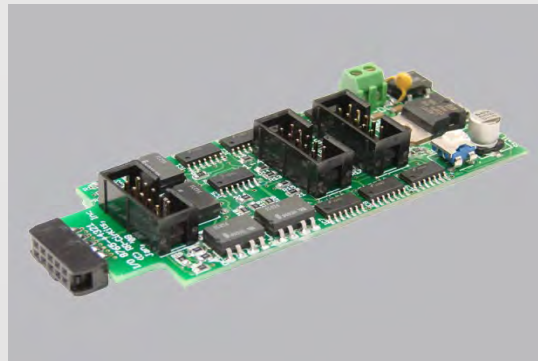
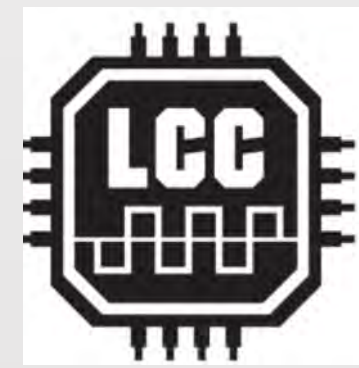
1
2
3
4



Detection Block #1
Detection Block #2
Detection Block #3
Detection Block #4

Numbers "Wrap" Around

SMD8 Card Can be Used for Turnouts or Other Functions



Input – 10 Wire Flat Cable from Tower or Signal Card

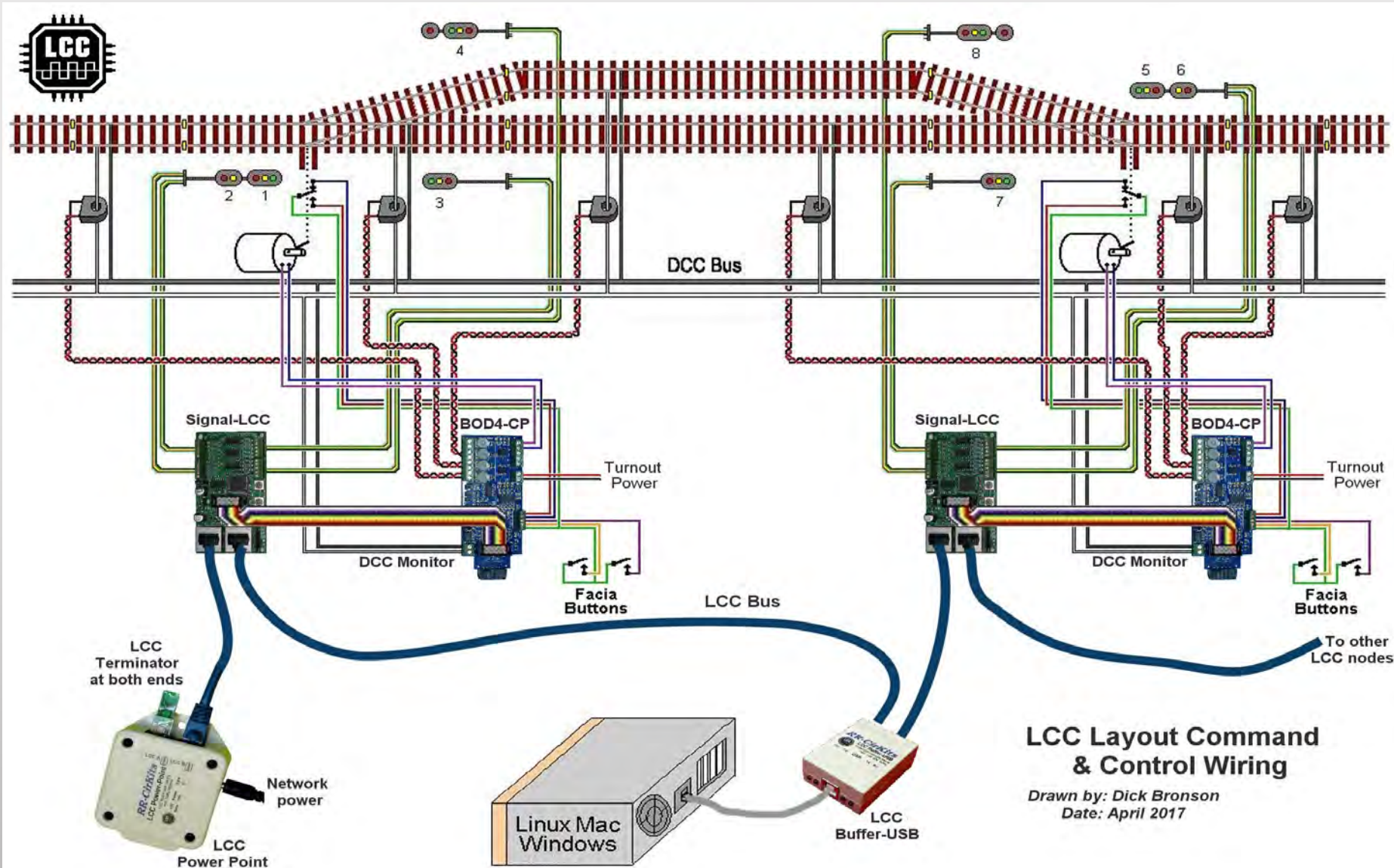
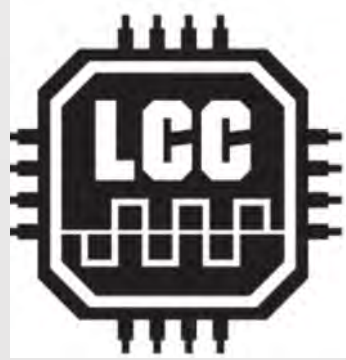
Output – 2 X 10 Wire Flat Cables

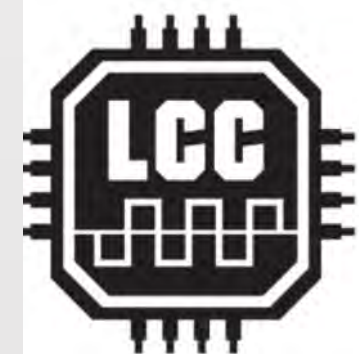
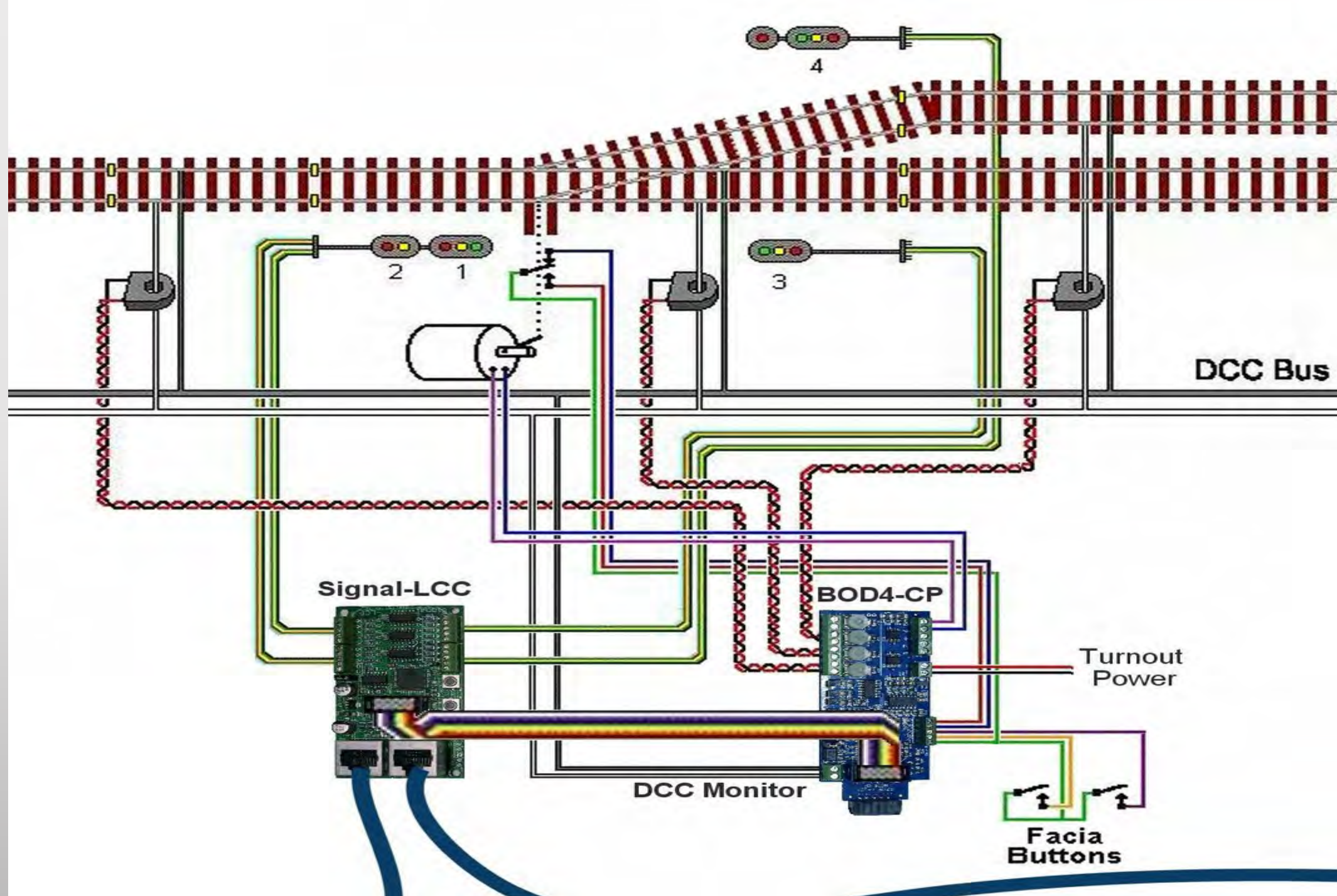
- Up to 100 mA per channel
- Max of 200 mA per board
- Optimized for Tortoise Motors
- Can be used to control other functions—such as LEDs!

Speed Control

- Sets voltage between 4 and 12 Volts
 - Sets Tortoise rate, or
 - Input voltage to LEDs (or other auxiliary)

How It Works at a Single Location!



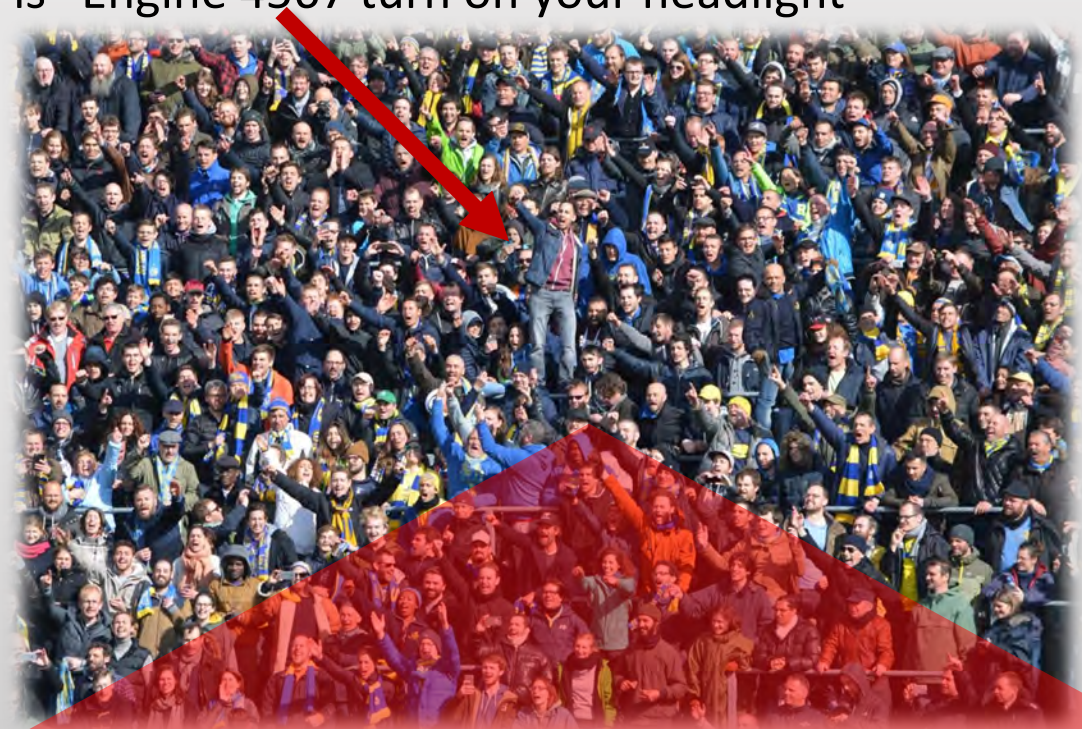


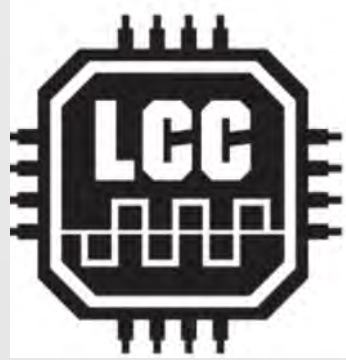
The LCC Bus Structure Provides All Information to All Locations

- DCC sends a message to an individual location
 - *This is analogous to texting an individual in a crowded room*
 - An example is “Engine 4567 turn on your headlight”

LCC announces to the “world”

- *This is analogous to a loudspeaker in an auditorium*
- An example is --A train enters Block 123 which sends a message “I am occupied”
 - All locations hear this message, and several react:
 - JMRI Panel Pro will change the color of the rail on the dispatcher panel to indicate occupancy
 - The signals in this block will change state to a “Red” aspect
 - Previous block signals will also change
 - Perhaps a turnout or other device will also activate if so directed. And the turnout will report its actual position.

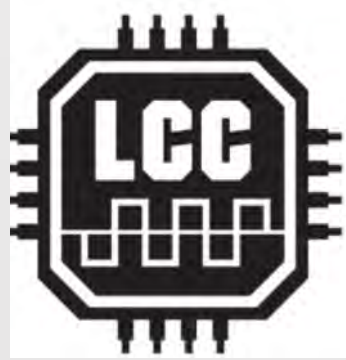




- LCC Basic Concepts include:
Peer-peer

A peer-peer network does not require a controlling computer.

(A computer is required to setup and get the system configured)

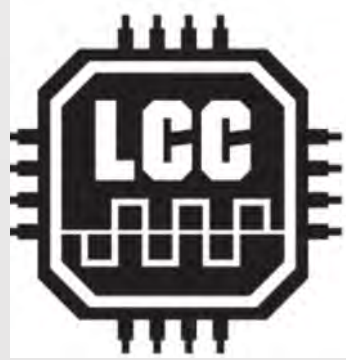


- LCC Basic Concepts include:
Globally unique

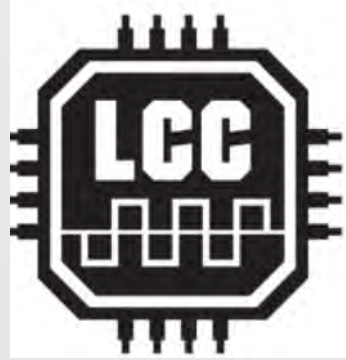
Each node has a globally unique, identifier, one of 281 trillion.

Each NMRA member has 256 node IDs

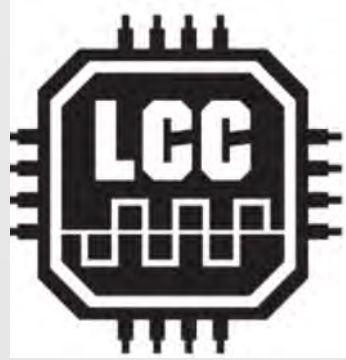
Each node can create 65,536 Events



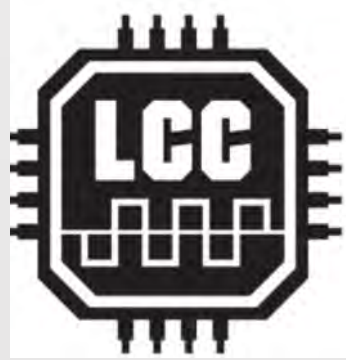
- LCC Basic Concepts include:
Self-describing
- **LCC will provide a complete list of all hardware connected to the Layout**



- LCC Basic Concepts include:
Self-describing
- LCC will provide a complete list of all hardware connected to the Layout
- **With LCC you don't tell the system what hardware you have added, the system will tell you.**

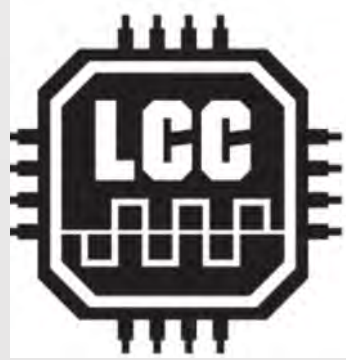


- LCC Basic Concepts include:
Self-describing
- LCC will provide a complete list of all hardware connected to the Layout
- With LCC you don't tell the system what hardware you have added, the system will tell you.
- **You decide what accessories are named.**



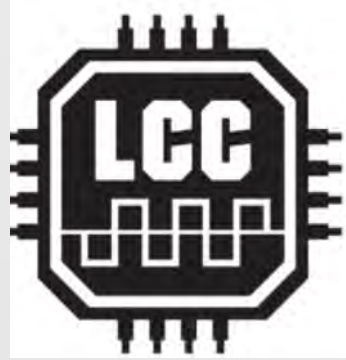
- LCC Basic Concepts include:
Self-describing
- LCC will provide a complete list of all hardware connected to the Layout
- With LCC you don't tell the system what hardware you have added, the system will tell you.
- You decide what accessories are named.
- **The system explains the capabilities elements have.**

LCC is Event Driven in a *Producer-Consumer* Environment



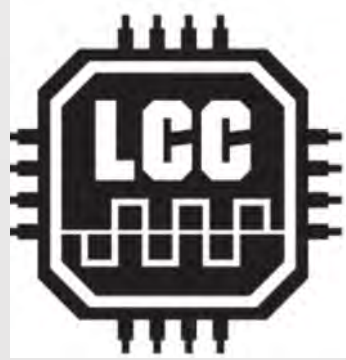
- An LCC Event has no predefined meaning
You define Events during configuration

LCC is Event Driven in a *Producer-Consumer* Environment



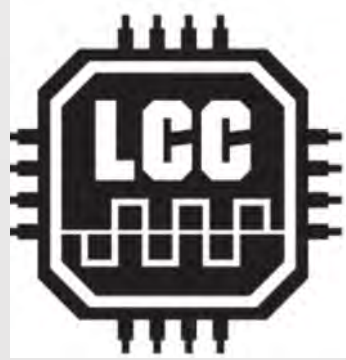
- An LCC Event has no predefined meaning
You define Events during configuration
- **An LCC Event simply says;**
 - **'something has happened'**
 - **'something should happen'**

LCC is Event Driven in a *Producer-Consumer* Environment



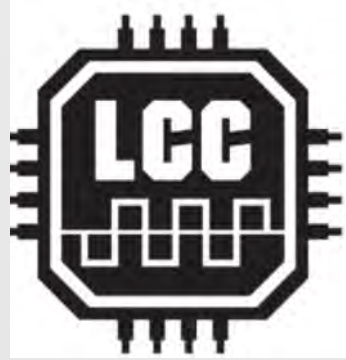
- An LCC Event has no predefined meaning
You define Events during configuration
- An LCC Event simply says;
 - 'something has happened'
 - 'something should happen'
- **Example is “Turnout (your naming) is Set Normal”**

LCC is Event Driven in a *Producer-Consumer* Environment



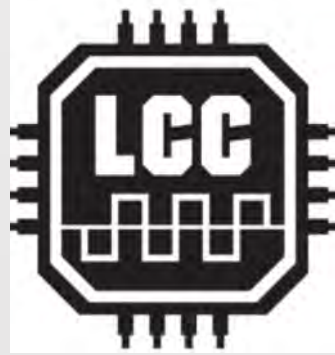
- A *Producer* is some device can create (produce) an Event
Some examples might be a push button or block detector (Current Transformer (CT)) .

LCC is Event Driven in a *Producer-Consumer* Environment



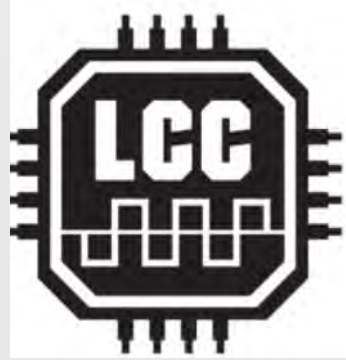
- A *Producer* is some device can create (produce) an Event
Some examples might be a push button or block detector (Current Transformer (CT)) .
- **A *Consumer* is some device can respond to (consume) an Event**
It could be a lamp, a turnout driver, or anything you can control.

LCC is Event Driven in a *Producer-Consumer* Environment

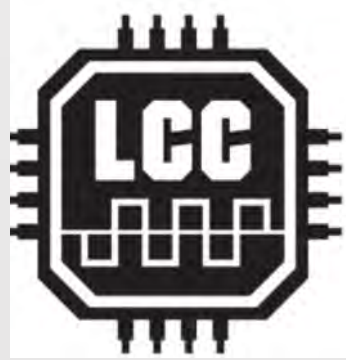


- A *Producer* is some device can create (produce) an Event
Some examples might be a push button or block detector (Current Transformer (CT)) .
- A *Consumer* is some device can respond to (consume) an Event
It could be a lamp, a turnout driver, or anything you can control.
- ***An Event can have from 1 to many Producers***

LCC is Event Driven in a *Producer-Consumer* Environment

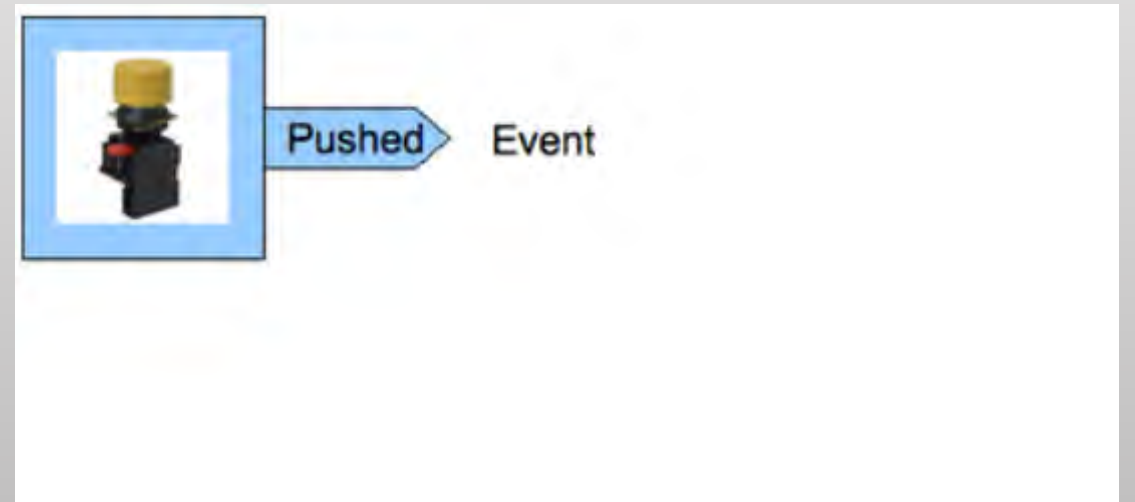


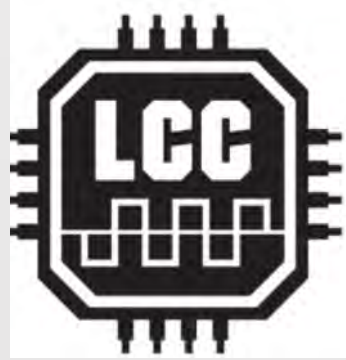
- A *Producer* is some device can create (produce) an Event
Some examples might be a push button or block detector (Current Transformer (CT)) .
- A *Consumer* is some device can respond to (consume) an Event
It could be a lamp, a turnout driver, or anything you can control.
- *An Event* can have from 1 to many *Producers*
- *An Event* can have from 0 to many *Consumers*



To Elaborate a Little *Bit* on Events

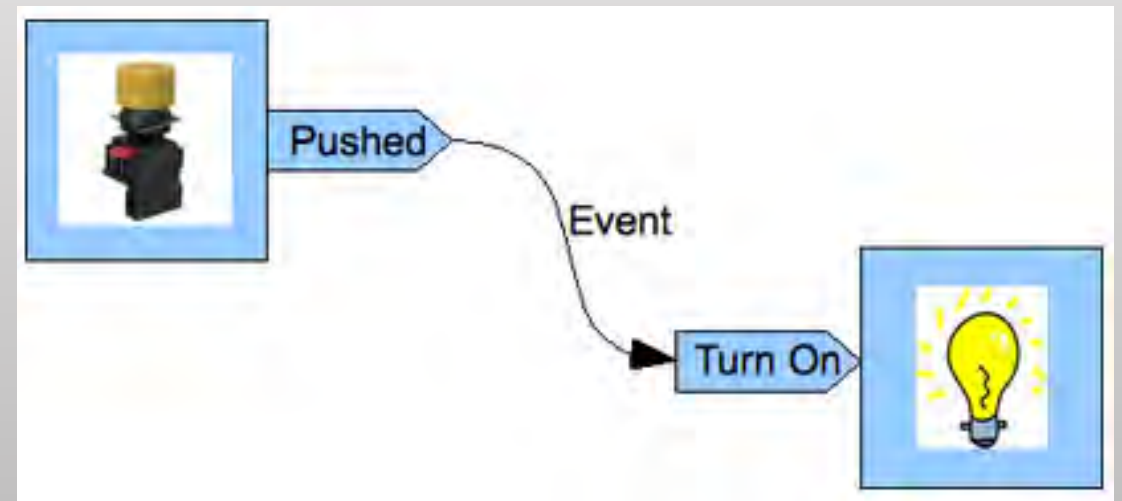
- For an event to happen something must have occurred.

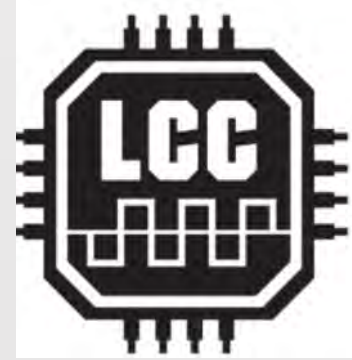




To Elaborate a Little *Bit* on Events

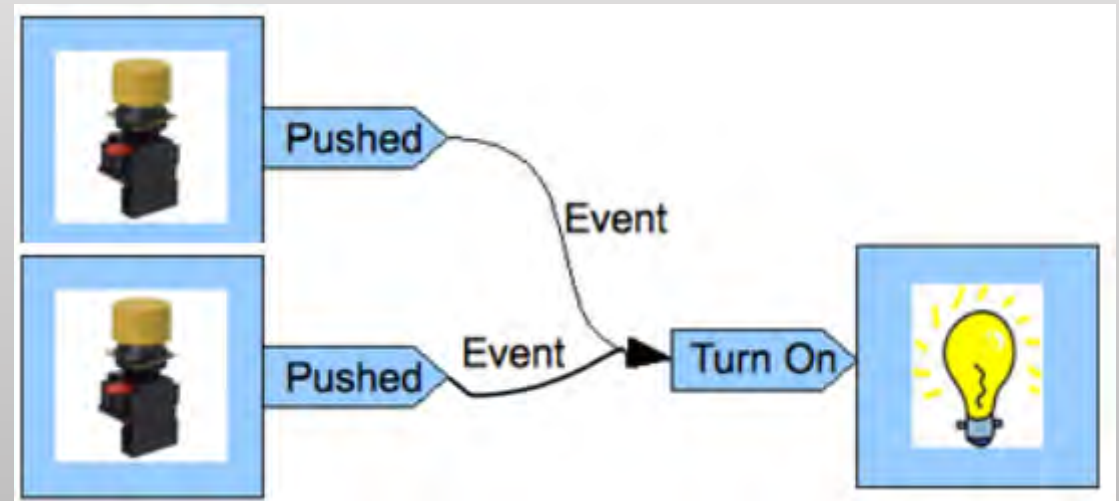
- For an event to happen something must have occurred.
- **Therefore, there must be at least one *Producer*.**

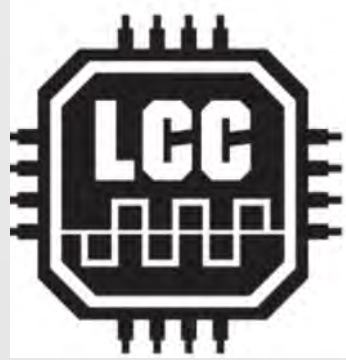




To Elaborate a Little *Bit* on Events

- For an event to happen something must have occurred.
- Therefore, there must be at least one *Producer*.
- **In the LCC world it is possible for many different *Producers* to create the same event.**
- **For example you might want to have turnout control buttons track side and on a remote panel.**

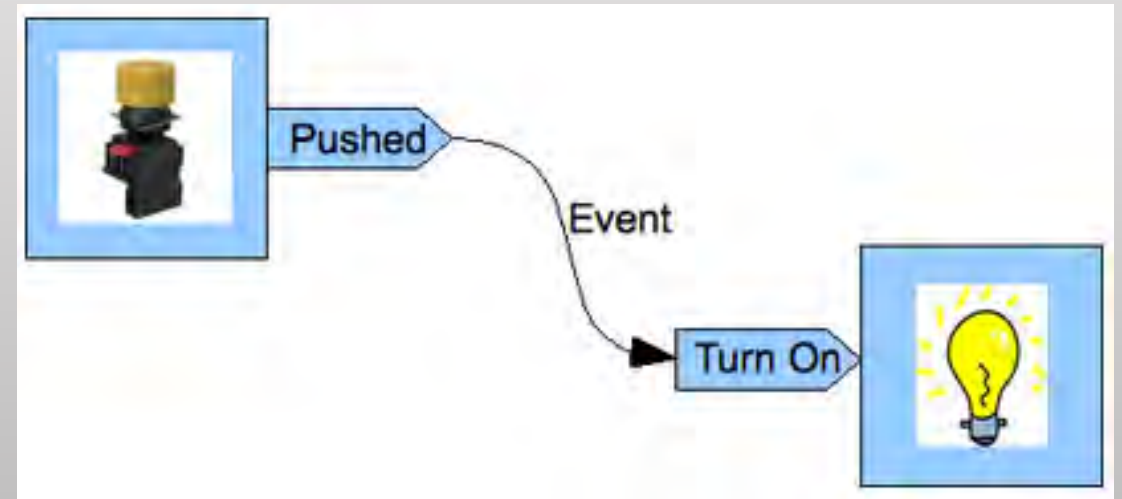


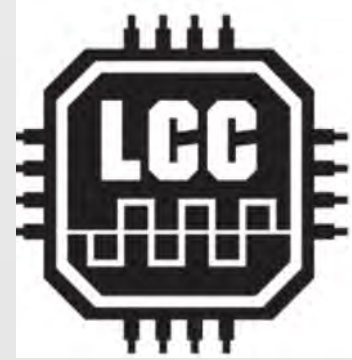


To Elaborate a Little *Bit* on Events

- For an event to happen something must have occurred.
- Therefore, there must be at least one *Producer*.
- In the LCC world it is possible for many different *Producers* to create the same event.
- For example you might want to have turnout control buttons track side and on a remote panel.

- **Producers create Events**

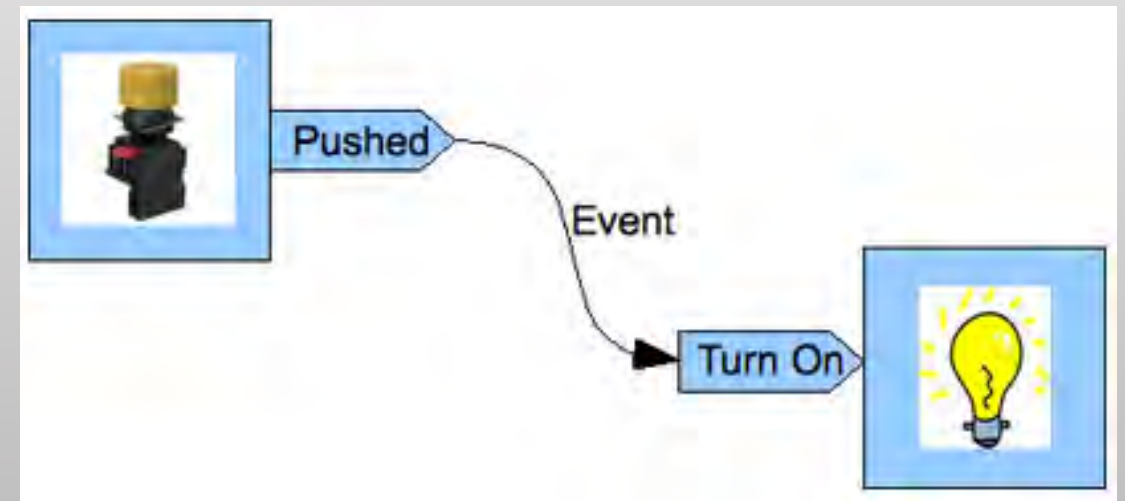


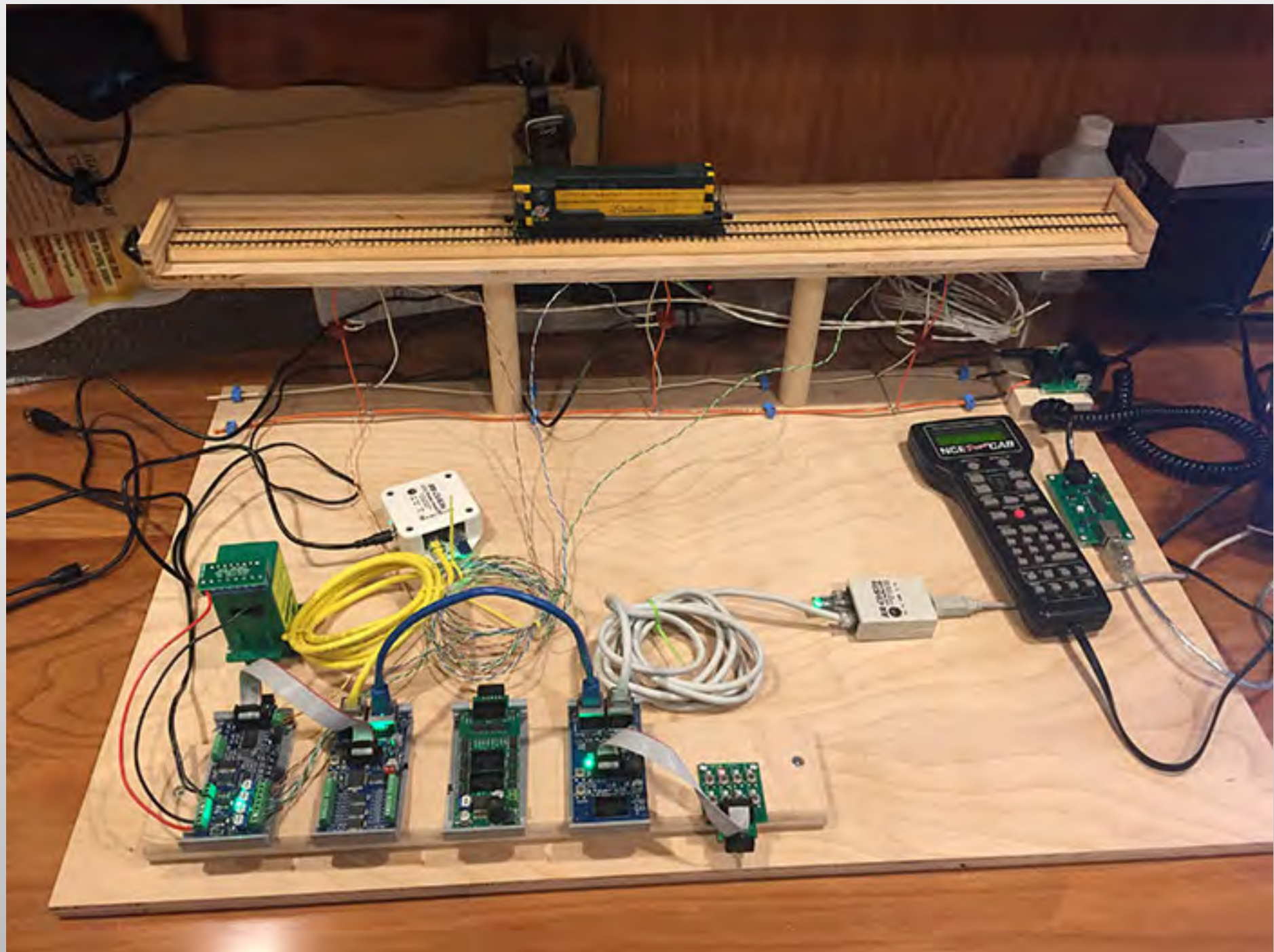


To Elaborate a Little *Bit* on Events

- For an event to happen something must have occurred.
- Therefore, there must be at least one *Producer*.
- In the LCC world it is possible for many different *Producers* to create the same event.
- For example you might want to have turnout control buttons track side and on a remote panel.

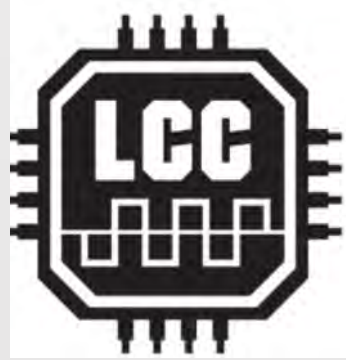
- Producers create Events
- **Consumers act on Events**







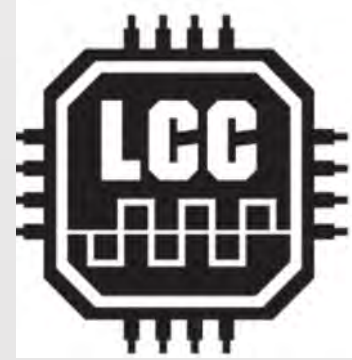
Coffee Break Before:



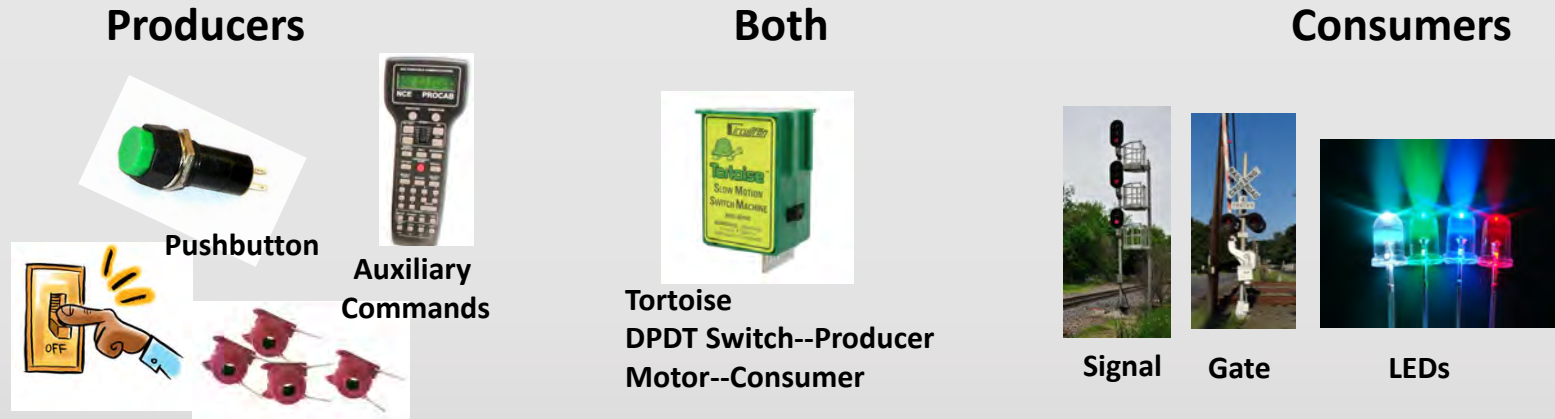
- **Configuring Can be Tedious at First**
- **Lessons Learned on Planning**
- **LCC configuration and JMRI Panel Pro**
- **Experience with Layout Command Control,
LCC At the Arizona Railroad Historical Society**



Configuring Can be Tedious at First, But Then it Becomes Cut and Paste!

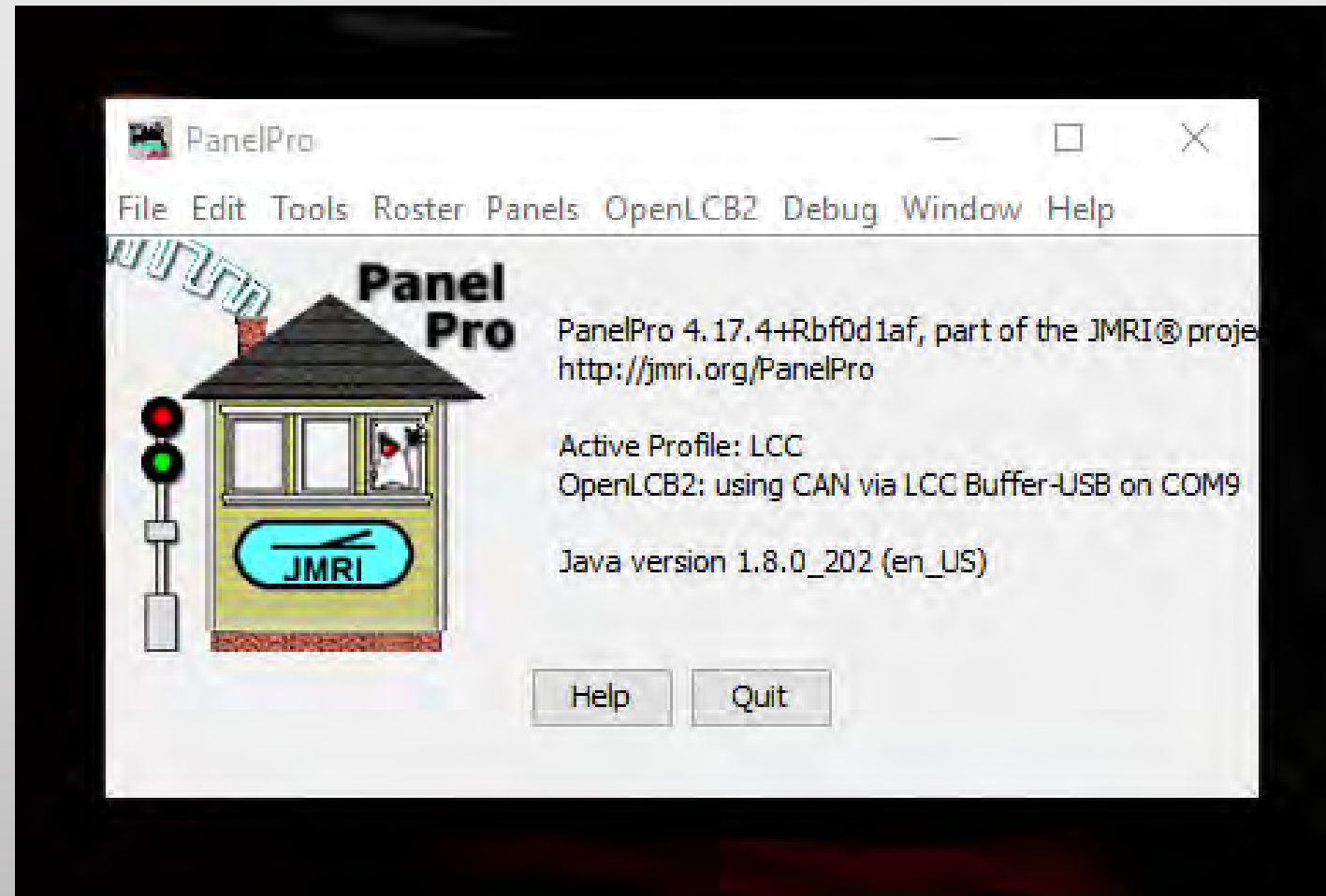
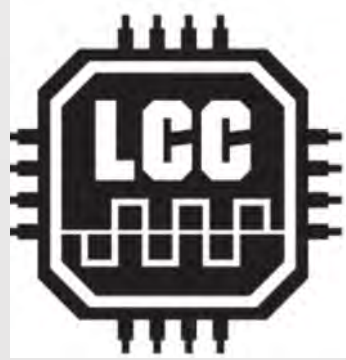


- Most Texts begin by describing in detail “Nodes”, “Producers”, and “Consumers”

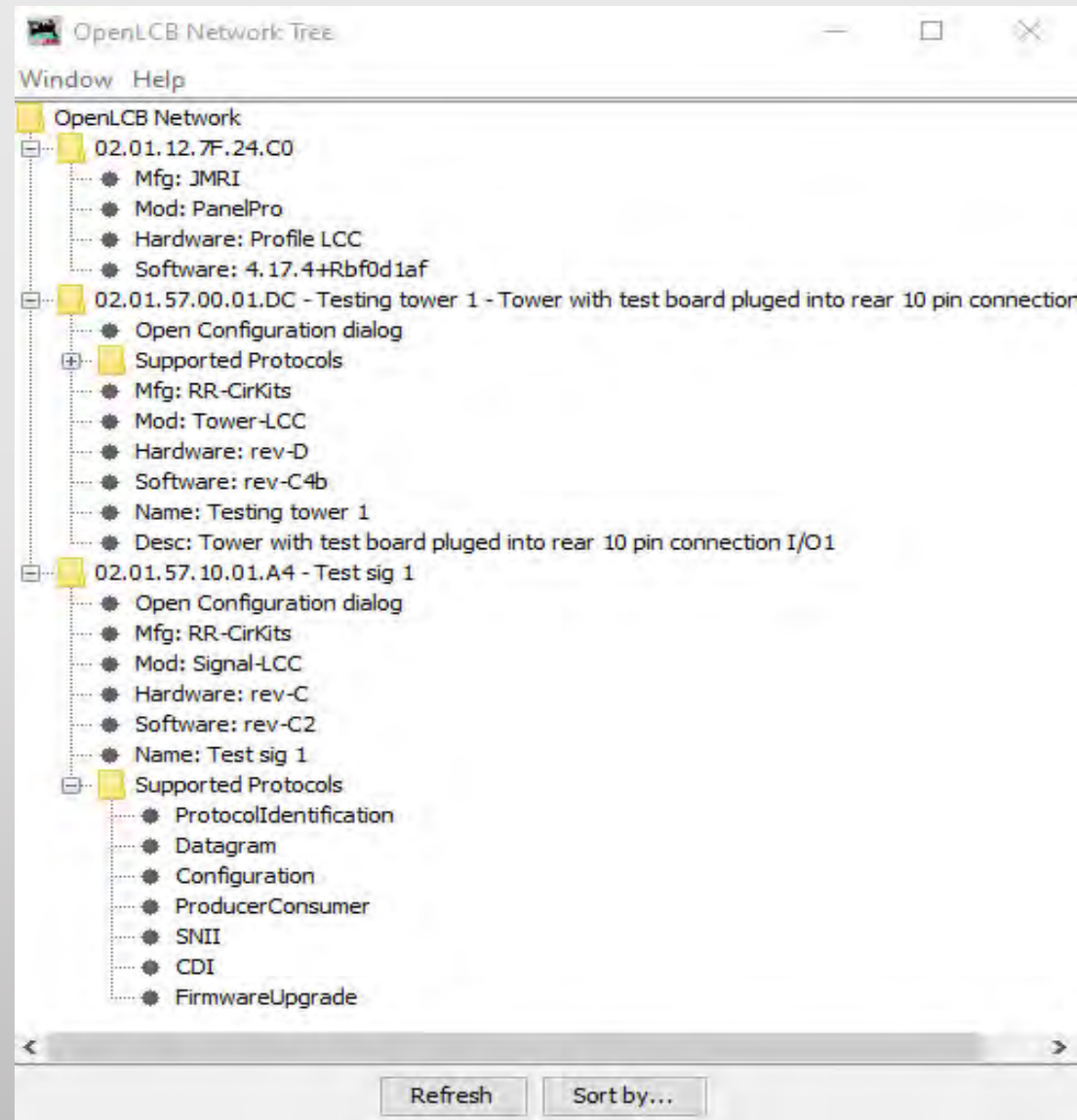
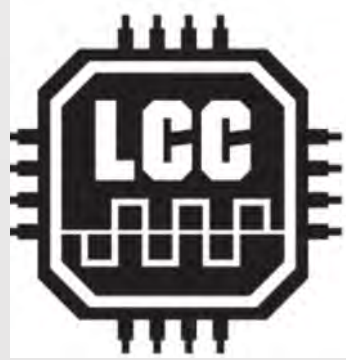


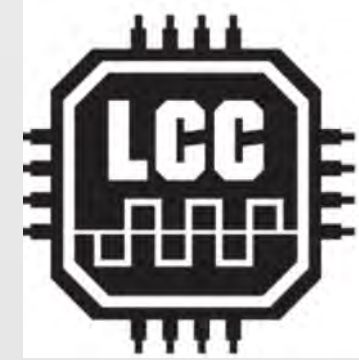
- The Signal and Tower Cards are Configured—not Programmed!
 - The cards are preloaded with a default configuration which covers many cases
 - The cards contain all of the possible choices for operations which are selected from a drop-down menu
 - Once configured, the result can be copied to other cards as appropriate
 - The “daughter” boards are not configured, but work off the commands of the Signal or Tower Card

Configuration Tool Using JMRI Panel Pro



JMRI Configuration Network Tree

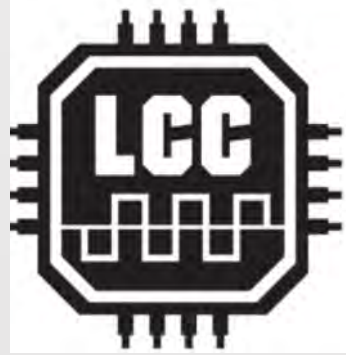




JMRI Configuration Network Tree

- 02.01.57.00.01.DC - Testing tower 1 - Tower with test board plugged into rear 10 pin connection
 - Open Configuration dialog
 - Supported Protocols
 - Mfg: RR-CirKits
 - Mod: Tower-LCC
 - Hardware: rev-D
 - Software: rev-C4b
 - Name: Testing tower 1
 - Desc: Tower with test board plugged into rear 10 pin connection I/O 1

JMRI Configuration Testing Tower



Configure Testing tower 1 - Tower with test board plugged into rear 10 pin connection I/O1 (02...

Identification
Manufacturer: RR-CirKits
Model: Tower-LCC
Hardware Version: rev-D
Software Version: rev-C4b

Segment: NODE ID

Your name and description for this node

Node Name
Testing tower 1

Node Description
Tower with test board plugged into rear 10 pin connection I/O1

Segment: Port I/O

Select Input/Output line.

Line 7	Line 8	Line 9	Line 10	Line 11	Line 12	Line 13
Line 1 (Button 1 pressed)			Line 2 (LED 2 Lights on button 1 press)		Line 3	Line 4

I/O

Line description
Button 1 pressed

Output Function
No Function

Input Function
Active Lo

Delay
Delay time values for blinks, pulses, debounce.
Interval 1 Interval 2
Delay Time (1-60000).
0
Milliseconds

Retrigger
No

Commands

Consumer commands.

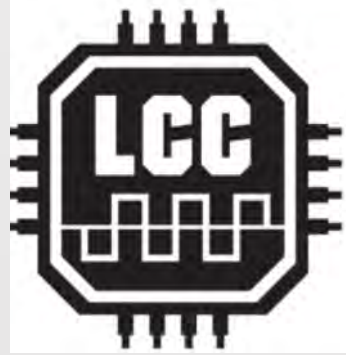
Event 1 Event 2 Event 3 Event 4 Event 5 Event 6

EventID
(C) When this event occurs,
02.01.57.00.01.DC.00.00

Other uses of this Event ID:
Testing tower 1.Port I/O.Select Input/Output line.(1,Button 1 pressed).I/O.Commands(1)
the line state will be changed to.
None

Sensor/Turnout creation

JMRI Configuration Testing Tower



Configure Testing tower 1 - Tower with test board plugged into rear 10 pin connection I/O1 (02.01.57.00.01.DC)

Node Name
Testing tower 1] Refresh Write

Node Description
Tower with test board plugged into rear 10 pin connection I/O1 Refresh Write

Segment: Port I/O

Select Input/Output line.

Line 6 (Light LED 6 on button 8)	Line 7	Line 8	Line 9	Line 10	Line 11
Line 1 (Button 1 pressed)	Line 2 (LED 2 Lights on button 1 press)				Line 3

I/O

Line description
Button 1 pressed Refresh Write

Output Function
No Function Refresh Write

Input Function
Active Lo Refresh Write

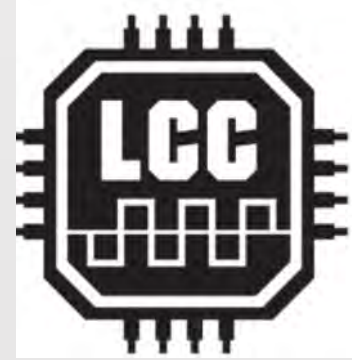
Delay
Delay time values for blinks, pulses, debounce.
Interval 1 Interval 2

Delay Time (1-60000).
0 Refresh Write

Milliseconds Refresh Write

Retrigger
No Refresh Write

JMRI Configuration Testing Tower Line 1



Select Input/Output line.

Line 6 (Light LED 6 on button 8)	Line 7	Line 8
Line 1 (Button 1 pressed)	Line 2 (LED 2 Lights)	

I/O

Line description

Button 1 pressed

Output Function

No Function

Input Function

Active Lo

Delay

Delay time values for blinks, pulses, debounce.

Interval 1 Interval 2

Delay Time (1-60000).

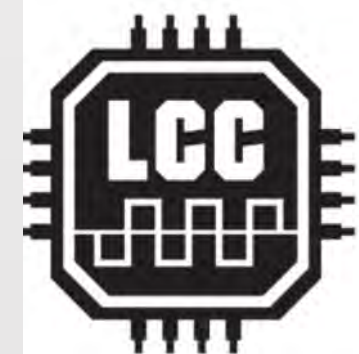
0

Milliseconds

Retrigger

No

JMRI Configuration Testing Tower Line 1 Producer of Event 1 (02.01.57.00.01.DC.00.06)



Indications

Producer commands.

Event 1 Event 2 Event 3 Event 4 Event 5 Event 6

Upon this action

Input On Refresh Write

EventID

(P) this event will be sent.

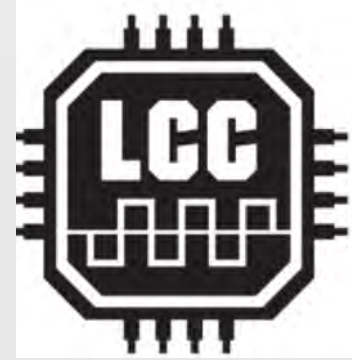
02.01.57.00.01.DC.00.06 Refresh Write Copy Paste Search

Other uses of this Event ID:

Testing tower 1.Port I/O.Select Input/Output line.(2,LED 2 Lights on button 1 press).I/O.Commands(2)

Testing tower 1.Port I/O.Select Input/Output line.(4,LED 4 Lights on button 1 press).I/O.Commands(2)

JMRI Configuration Testing Tower Line 2



Select Input/Output line.

Line 6 (Light LED 6 on button 8)	Line 7	Line 8	Line 9	Line 10
Line 1 (Button 1 pressed)	Line 2 (LED 2 Lights on button 1 press)			

I/O

Line description

LED 2 Lights on button 1 press

Output Function

Blink A Active Hi

Input Function

Disabled

Delay

Delay time values for blinks, pulses, debounce.

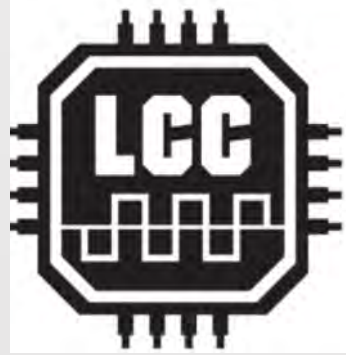
Interval 1 Interval 2

Delay Time (1-60000).

500

Milliseconds

JMRI Configuration Testing Tower Line 2 Consumer Event 2 (02.01.57.00.01.DC.00.06)



Commands

Consumer commands.

Event 1 Event 2 Event 3 Event 4 Event 5 Event 6

EventID

(C) When this event occurs,

02.01.57.00.01.DC.00.06 Refresh Write Copy Paste Search

Other uses of this Event ID:

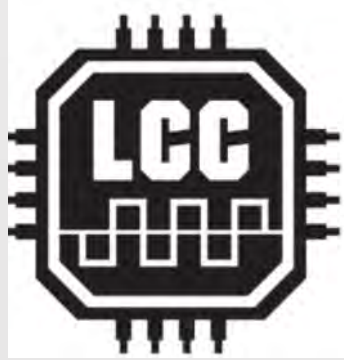
Testing tower 1.Port I/O.Select Input/Output line.(1,Button 1 pressed).I/O.Indications(1)

Testing tower 1.Port I/O.Select Input/Output line.(4,LED 4 Lights on button 1 press).I/O.Commands(2)

the line state will be changed to.

Off (Line Inactive) Refresh Write

JMRI Configuration Testing Tower



Indications

Producer commands.

Event 1 Event 2 Event 3 Event 4 Event 5 Event 6

Upon this action

Input On Refresh Write

EventID

(P) this event will be sent.

02.01.57.00.01.DC.00.06 Refresh Write Copy Paste Search

Other uses of this Event ID:

Testing tower 1.Port I/O.Select Input/Output line.(2,LED 2 Lights on button 1 press).I/O.Commands(2)

Testing tower 1.Port I/O.Select Input/Output line.(4,LED 4 Lights on button 1 press).I/O.Commands(2)

Commands

Consumer commands.

Event 1 Event 2 Event 3 Event 4 Event 5 Event 6

EventID

(C) When this event occurs,

02.01.57.00.01.DC.00.06 Refresh Write Copy Paste Search

Other uses of this Event ID:

Testing tower 1.Port I/O.Select Input/Output line.(1,Button 1 pressed).I/O.Indications(1)

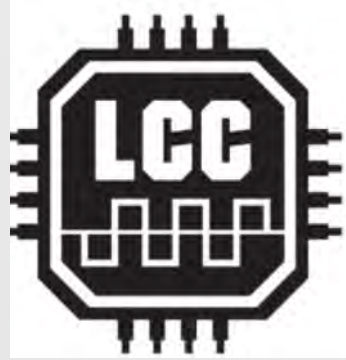
Testing tower 1.Port I/O.Select Input/Output line.(4,LED 4 Lights on button 1 press).I/O.Commands(2)

the line state will be changed to.

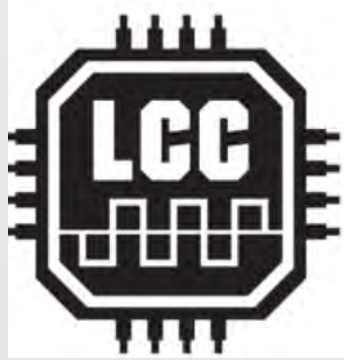
Off (Line Inactive) Refresh Write



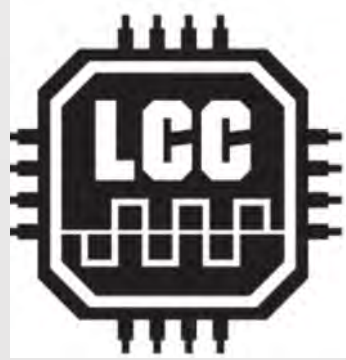
Experience with Layout Command Control, LCC At the Arizona Railroad Historical Society



Arizona Railroad Historical Society, ARHS, Display at The Arizona Capitol Museum

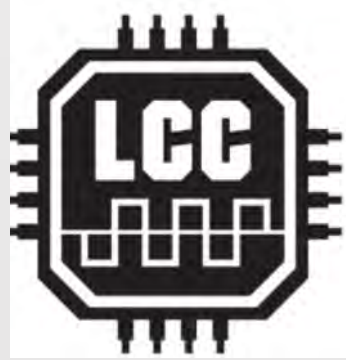


Lessons Learned: LCC configuration and JMRI Panel Pro



- **Yes, there is a Learning Curve: Take it Step by Step**
 - Creating a test bed to learn the basic building blocks was essential
- **Limited documentation to guide the beginner**
 - An Intro to Layout Command Control by Dana Zimmerli, PhD
 - LCC and JMRI user forums at [Groups.io](https://groups.io), [JMRI.org](https://jmri.org)
 - ARHS has gotten lots of help from manufacturers (RRCirkits) and JMRI developers
 - Don't be afraid to ask questions
- **Take it slow, once you learn the configuration process, it is repetitive for each card/node.**
- **Recommend creating all LCC-JMRI elements on the computer that will run the railroad.**
 - ARHS had different members working on separate computers and then attempting to integrate them on the main computer. This has created challenges—especially with integrating JMRI.

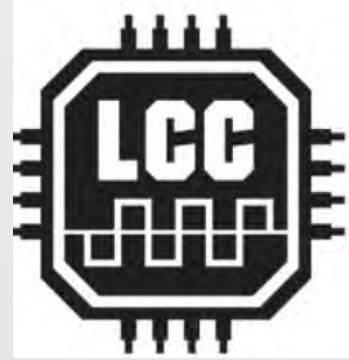
Lessons Learned on Planning



- **Up Front Planning is Important since this loads into all LCC Cards**
 - Pre-plan the names and numbers for all Block Locations, Turnouts and Signals
 - This seems like a big waste of time until the configuring begins, and then this becomes the most important investment of time!

- **Positioning of the Cards on the layout must be done in advance**
 - It is important to get the cards near where the “node” located which means “scattering them around the layout—but they must be closer together to lower voltage drop.”
 - Bus must be “daisy chained”
 - Need to be close to function
 - Must inject power within 10 feet

Lessons Learned About System Design

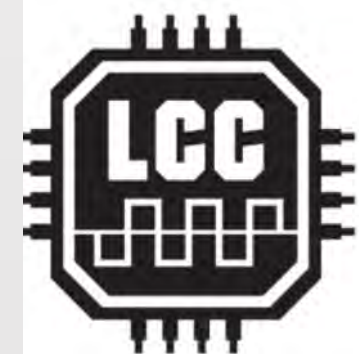


- The **“Rule of Twos”** in LCC Design
 - All cards support 2, 4 or 8 components—having an odd number of tracks/functions makes the design logic more difficult. (A four or six track yard is easier to layout than a 3 or 5 track yard.)
- **Current Transformers can be located either on a panel or near the point of “use” or anywhere in between**
- **Blocks must be completely isolated before the Current Transformers work properly.**
- **A separate “sub-block” can be used for auto-throw of a turnout or activating a crossing signal etc.**
- **Future applications boggle the mind**
 - Arduinos
 - Lighting
 - Special Effects....





The System Works Great!!



- **ARHS has installed 66 Signal and 55 BOD4-CP Cards without letting the “Magic Smoke” out of any of them! ***

- Over 100 Turnouts are operational
- Occupancy Detectors are sensitive and reliable
- Starting to experiment with Signals

- **Would we recommend this system?—Absolutely!**

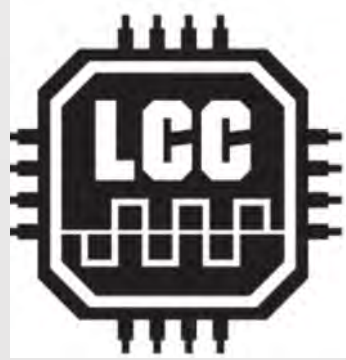
- **The National Convention in Salt Lake City convinced us that LCC was the right decision—All of the system developers were present, they were taking feedback and all suggestions for improvements very seriously**

- “Debugging” quirks in the LCC cards—such as a power up issue
- Working the interface to JMRI—assuring JMRI and LCC talk which is a challenge
- Taking suggestions to improve user interface—No Push-back!!

- LCC is good today—and is getting better.....



**Remember when DCC was new and people asked
“Why would I ever convert?”
That is where LCC is today!!**



Special Thanks

**The Arizona Railroad Historical Society
Jess Poole, Founding President**

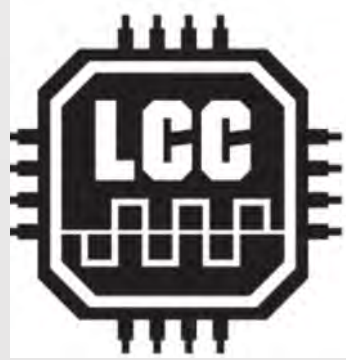
And

Dick and Karen Bronson of





Additional Information



- Users Group

[openlcb@ groups.io](mailto:openlcb@groups.io)

LayoutCommandControl@groups.io

- Useful Links

<http://openlcb.org>

<http://openlcb.com>

<http://nmra.org>

www.rr-cirkit.com/clinics

The Basic Signal Logic Overview

Next Time

