SurfLiner 2024
NMRA National

LCC, What is it, who is it for?

(Layout Command & Control)

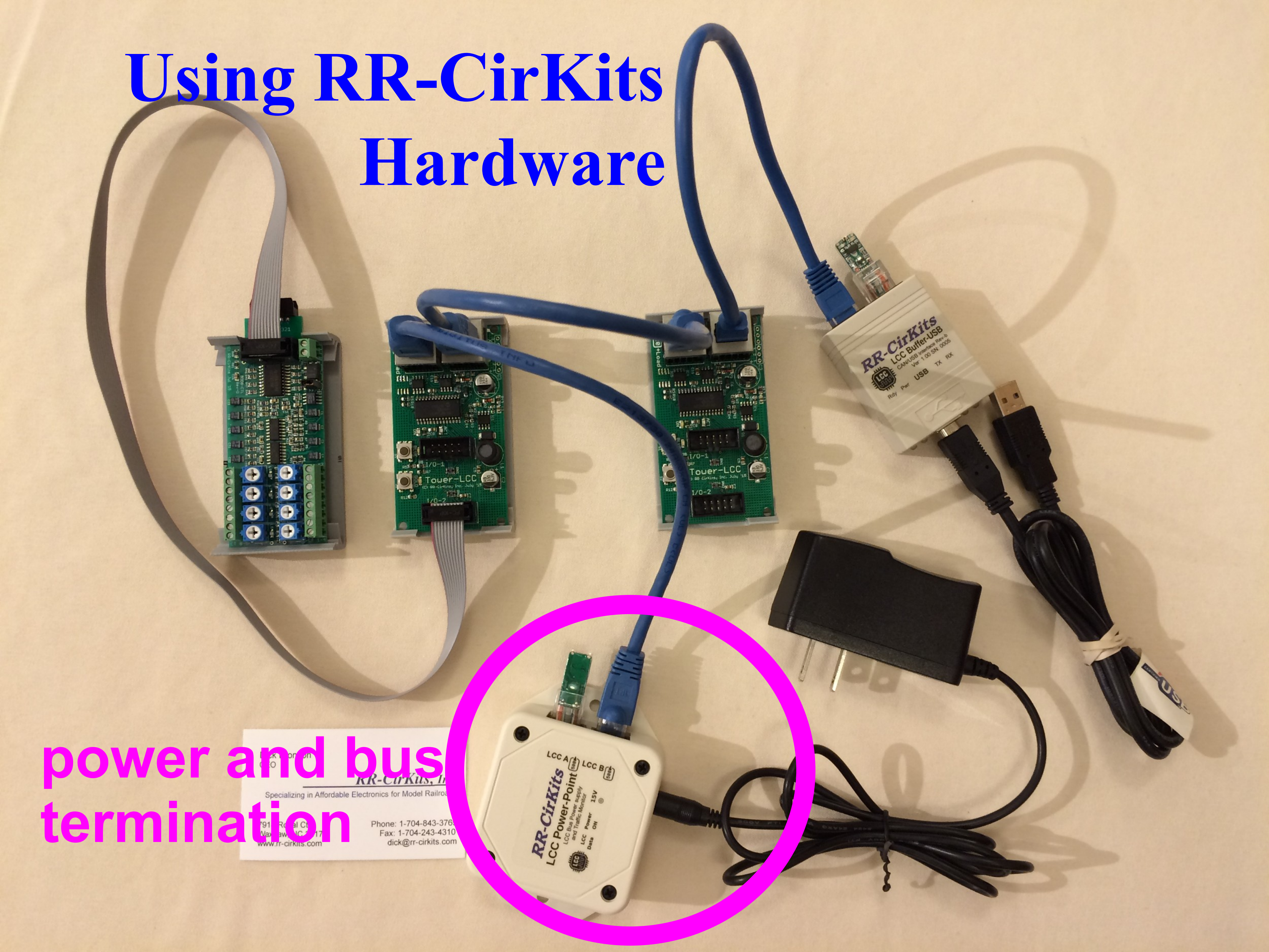
Compiled by: Dick Bronson
RR-CirKits, Inc.

LCC.
Part 1 (What is it, who is it for)
www.rr-cirkits.com/clinics/SL-2024-LCC-A.pdf
Part 2 (Applications and the future)
www.rr-cirkits.com/clinics/SL-2024-LCC-B.pdf

Getting Started

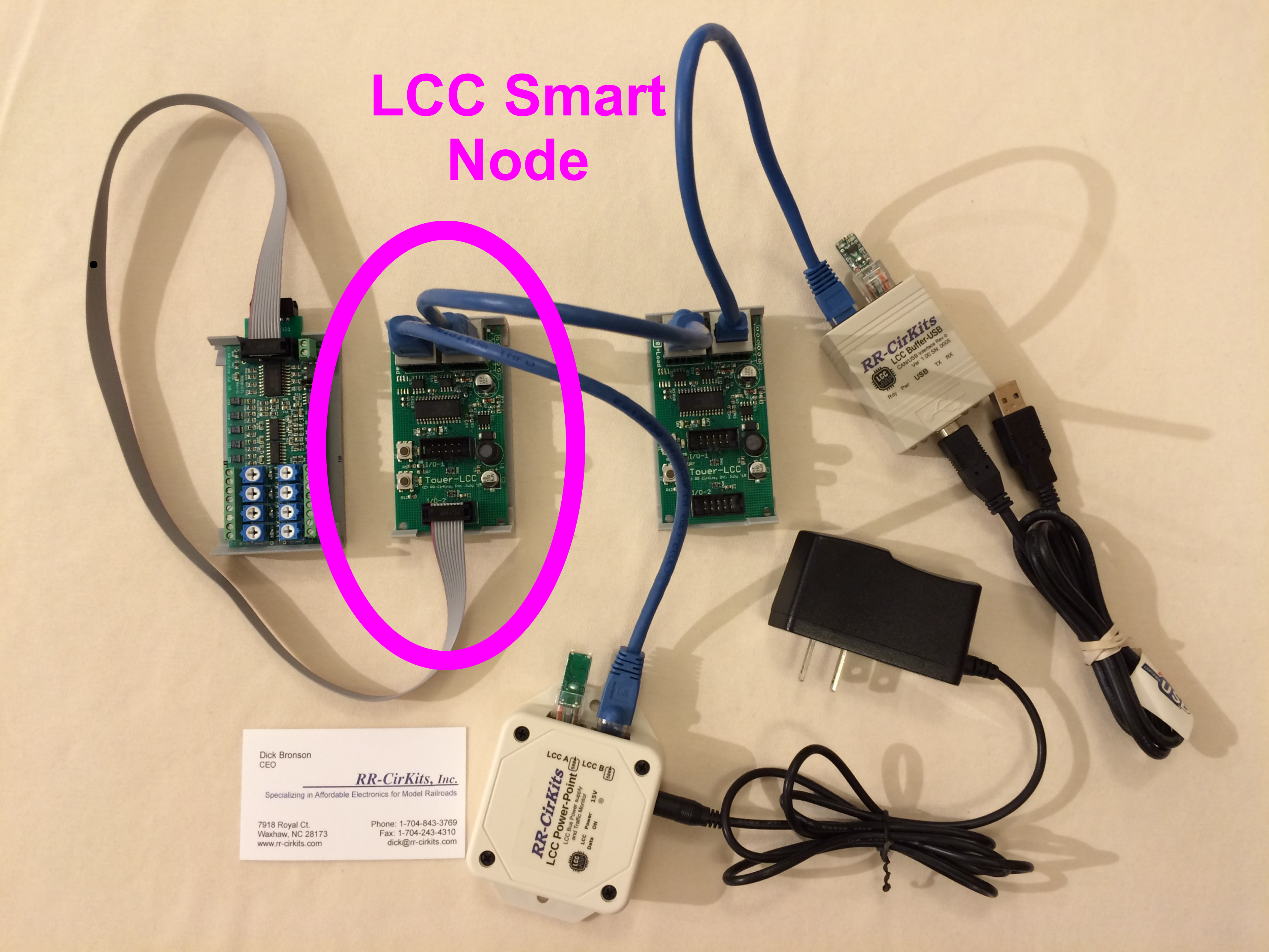
- The first task is to create an LCC network.
  - Power
  - Termination
  - One or more LCC nodes

Using RR-CirKits Hardware



power and bus termination

LCC Smart Node



Dick Bronson
CEO

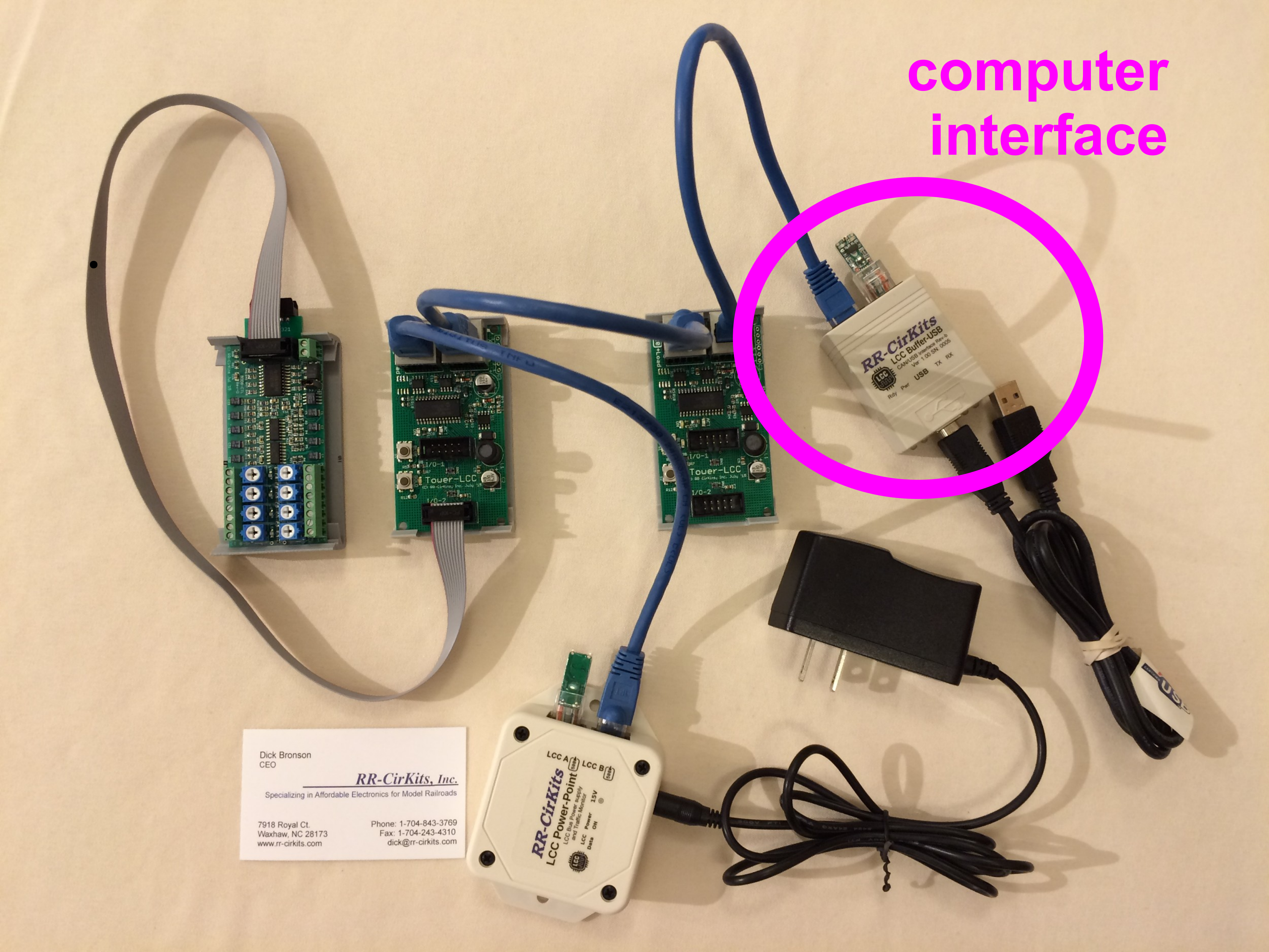
RR-CirKits, Inc.
Specializing in Affordable Electronics for Model Railroads

7918 Royal Ct.
Waxhaw, NC 28173
www.rr-cirkits.com

Phone: 1-704-843-3769
Fax: 1-704-243-4310
dick@rr-cirkits.com

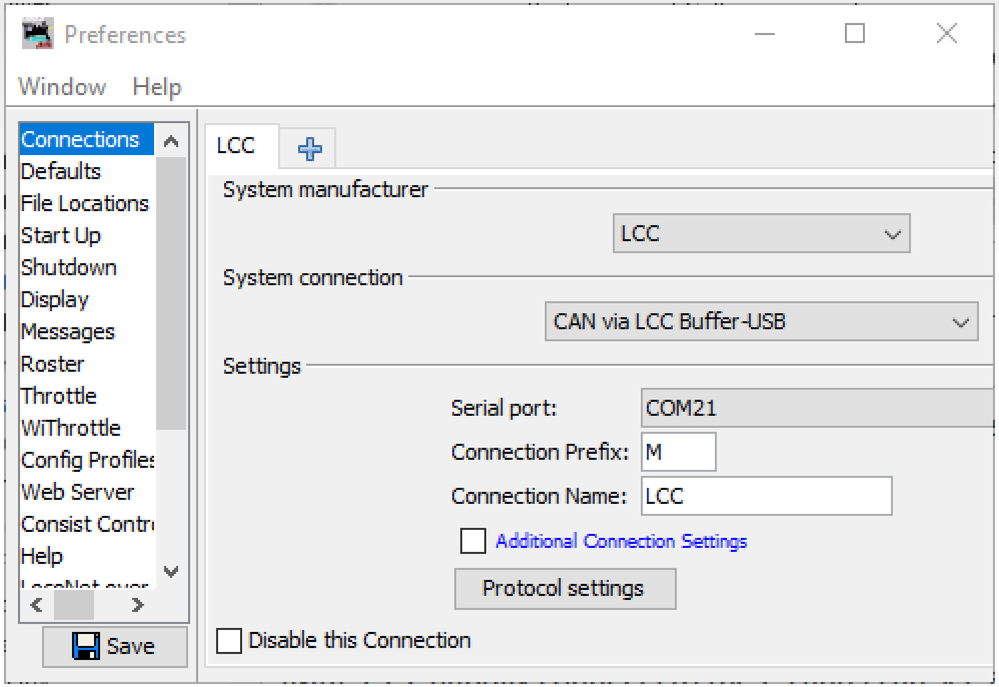
Connecting a Configuration Tool

- The next task is to connect LCC to JMRI, LCCtools, or MRS
  - A LCC Buffer-USB
  - B LCC LocoNet Gateway
  - C TCS Command Station (Using CAN via GC Network Interface)
  - D Some other Wifi Hub such as the OpenMRN Hub.


computer interface
RR-CirKits
LCC Buffer-USB
CAN/USB Interface Rev-b
Ver 1.00 SN 0005
Rdy Pwr USB TX RX
Tower-LCC
I/O-1
I/O-2
RR-CirKits
LCC Power-Point
LCC Bus Power supply and Traffic Monitor
LCC A
LCC B
15V
LCC Power
Data On
Dick Bronson
CEO
RR-CirKits, Inc.
Specializing in Affordable Electronics for Model Railroads
7918 Royal Ct.
Waxhaw, NC 28173
www.rr-cirkits.com
Phone: 1-704-843-3769
Fax: 1-704-243-4310
dick@rr-cirkits.com

Configure JMRI Preferences

- Select JMRI Edit → Preferences → Connections



System manufacturer: LCC

System connection: CAN via LCC Buffer-USB

Settings

Serial port: COM21

Connection Prefix: M

Connection Name: LCC

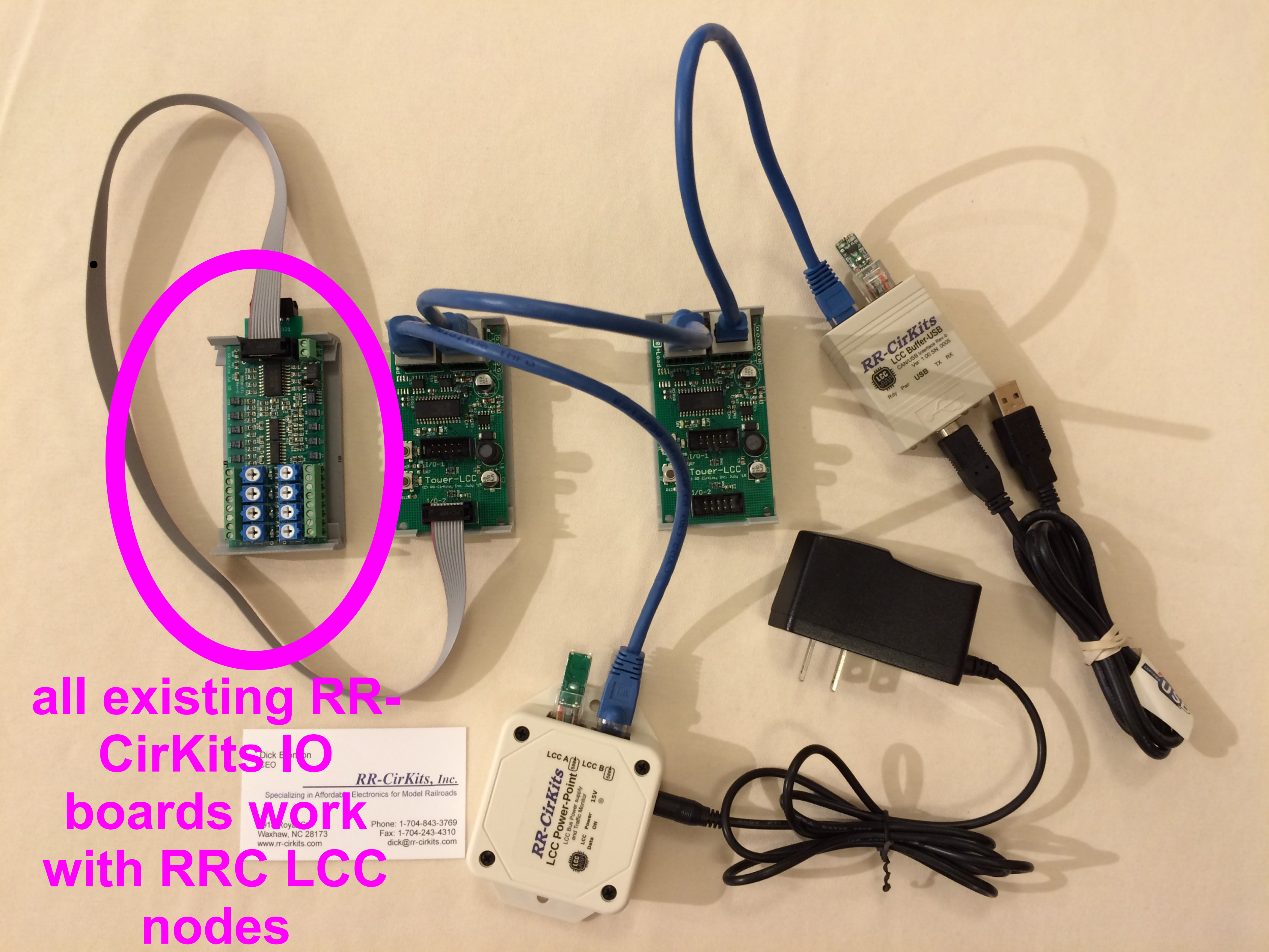
□ Additional Connection Settings

□ Disable this Connection

- System manufacturer: LCC
- System Connection:
  CAN via LCC Buffer-USB
- Serial Port: to match
- Save this configuration
- Restart JMRI

Adding Applications

- Finally we make some connection to simple layout hardware such as:
    - Detectors
    - Buttons
    - Turnouts
    - Cross bucks
    - Indicators
    - Signals
    - Building Lights
    - Etc.



all existing RR-CirKits IO boards work with RRC LCC nodes

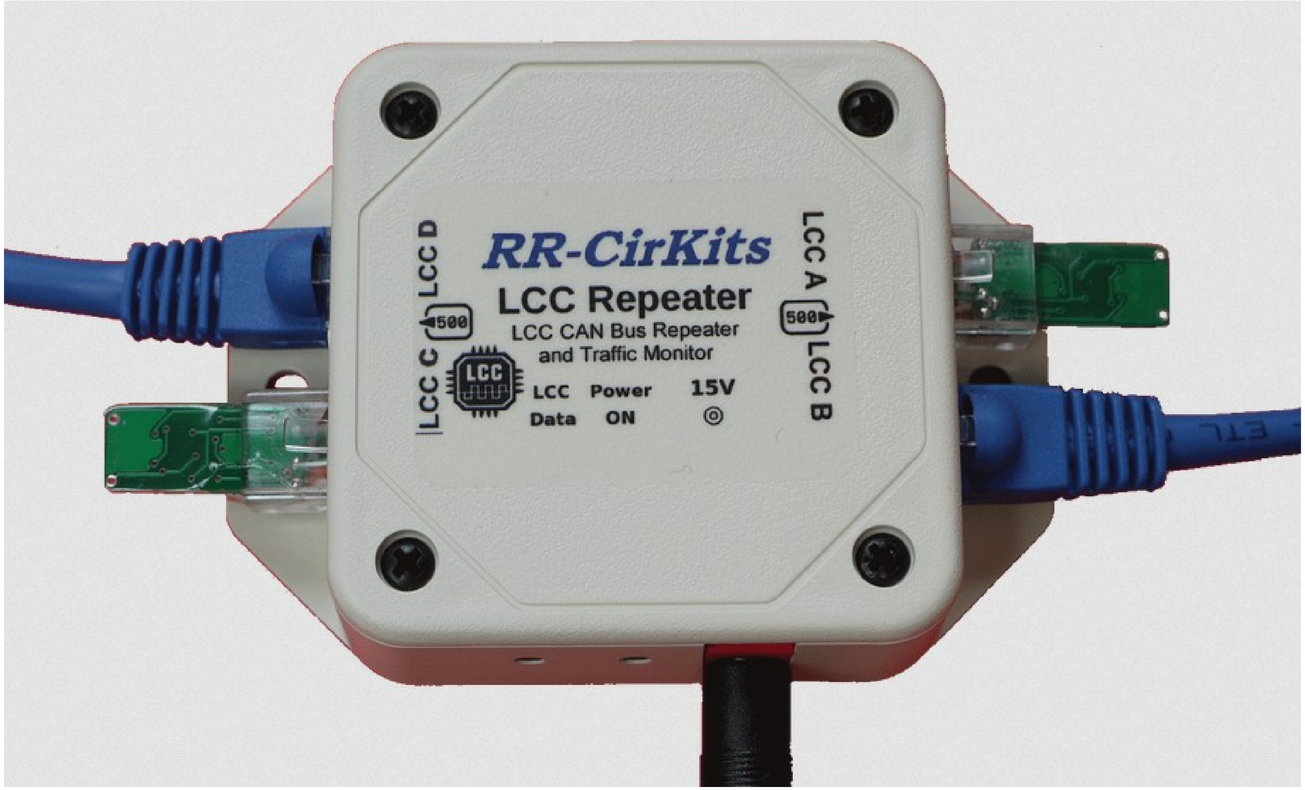
More Applications

- Or add more complex devices such as:
  - Repeaters
  - Fast clocks
  - Gateways
  - Command Stations
  - Boosters
  - Throttles
  - Power Managers
  - Trains
  - Computers

Applications

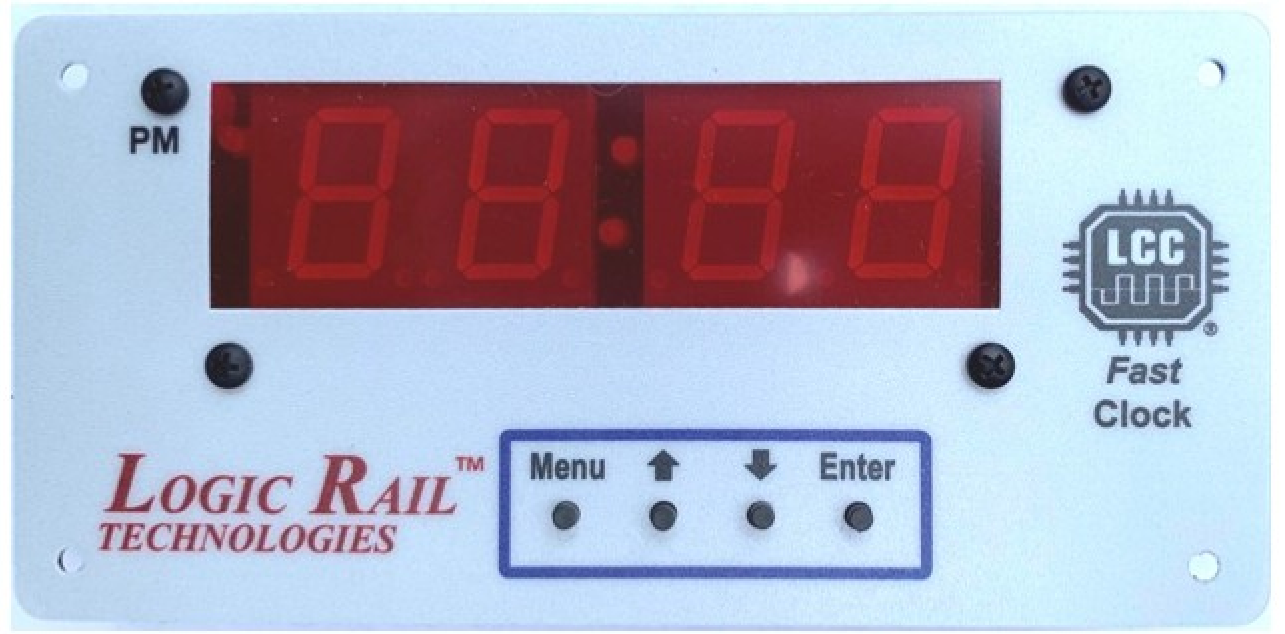
- Repeater

  A repeater is useful if you need to exceed the limits of a single CAN bus segment. Specifically, more than 40-50 nodes, or more than a single bus section. (e.g. a layout requiring a branching bus connection to follow a branch line)



Applications

- Fast Clock



Logic Rail Technologies has an LCC Fast Clock. This unit takes full advantage of the LCC protocol to support vastly improved capabilities over their similar products for legacy systems. Their NCE version is a repeater for the NCE cab fast clock. Their LocoNet version adds the capability to be configured to trigger up to four different LocoNet commands. The LCC version turns that option on its head, and allows any LCC node to be configured to be triggered at any of the 1,440 possible fast minutes in a fast day. Any LCC Fast Clock attached to the LCC bus may be a master or a slave. The LCC Fast Clock is configured over the bus with any CDI tool.

Applications

- Gateways
- A Gateway is some device that converts one protocol into another. In this example it can change a few types of LCC commands (EventIDs) into similar LocoNet messages. This allows LocoNet devices to be added onto an LCC network, and also allows a limited subset of LCC messages to be sent to a LocoNet network. We use a gateway to convert LocoNet throttle outputs into LCC Train Control Protocol.

- Just remember that the entire LocoNet device address space is about the same as that available in a couple of LCC nodes.

Applications

- LCC - DCC Command Station

The primary purpose of an LCC command station is to act as the proxy for the LCC Train Control Protocol for other train control methods such as DCC. Other than wireless, most over the rail control methods have limited feedback options that limit trains from acting as full LCC nodes. As RailCom improves many of these DCC limits will be removed.

Applications

- Wireless Throttles (LCC over Wifi)
- The TCS UWT-100 can operate in two different WiFi modes. Most current users use it to communicate with their existing command stations using the JMRI WiThrottle protocol. I suspect that many current owners don’t even realize that it also supports direct LCC traction protocol over WiFi, and that it can be easily configured over the LCC network.
- TCS is also supplying other throttle options and command station choices for LCC layouts.

Applications

- LCC - Gateway

  Expanding on the Gateway concept is another way to take advantage of the LCC standards. In this example MRC chose to create a gateway between the LCC and their existing DCC standard products.

- We see this option in the recently announced MRC NEXXT Gateway. It will allow a user with an existing Prodigy command station to add Train Control Protocol throttles, either WiFi or wired. These include MRC's own NEXXT WiFi throttle, the TCS LCC WiFi throttles, or even app based LCC throttles such as provided in the Apple LCC Tools, all into the mix of available throttle options.

- In this MRC product we see the gateway option taken even further by the inclusion of a JMRI Wi Throttle server to allow for even more throttle options for your users.

- MRC also has plans for a new LCC based command station similar in function to the TCS CS-105.

Applications

- Wireless Throttles (LCC over Wifi)
- The MRC-NEXXT throttle is also planned to operate in two different WiFi modes. LCC Train Control Protocol and the JMRI WiThrottle protocol.
- From what I understood it will include a browser based configuration tool, but I didn't understand how that would differ from a CDI tool.
- The NEXXT Gateway is planned to support bi-directional translation of Prodigy Cab Bus to LCC Train Control Protocol.

Applications

- JMRI and MRS
- We already mentioned using a computer running JMRI or MRS to configure an LCC network using their available CDI tools and we will explore that in depth in the next few slides. Here we are referring to these program’s capabilities as very smart nodes on the LCC network.
- In a peer-peer network such as the LCC no computer is required to operate the network, but a computer is certainly capable of participating in the network operations. Some examples might be to simulate a CTC panel, or even to automate train control.
- JMRI has supported the development of LCC from its infancy, and in many case was available even prior to the hardware supported to aid in its development.

Applications

- Like JMRI, the Deepwoods Software package includes a CDI tool as well as control panel tools.
- In addition to their free LCC related software, Deepwoods Software also offers some ESP32 based kits for building your own LCC nodes.
- LCC does not expect its users to do any programming, but for those that do want to create their own code for LCC nodes there is the OpenMRN. OpenMRN (Open Model Railroad Network) is a set of software libraries that are designed to support the NMRA's LCC (Layout Command Control) bus.
- OpenMRN is a collection of C++ code that is designed to make it easier to implement support for LCC. This might be in accessory decoders, in a command station, in a throttle, or any other device. The code is designed to be able to run on micro-controllers

The CDI

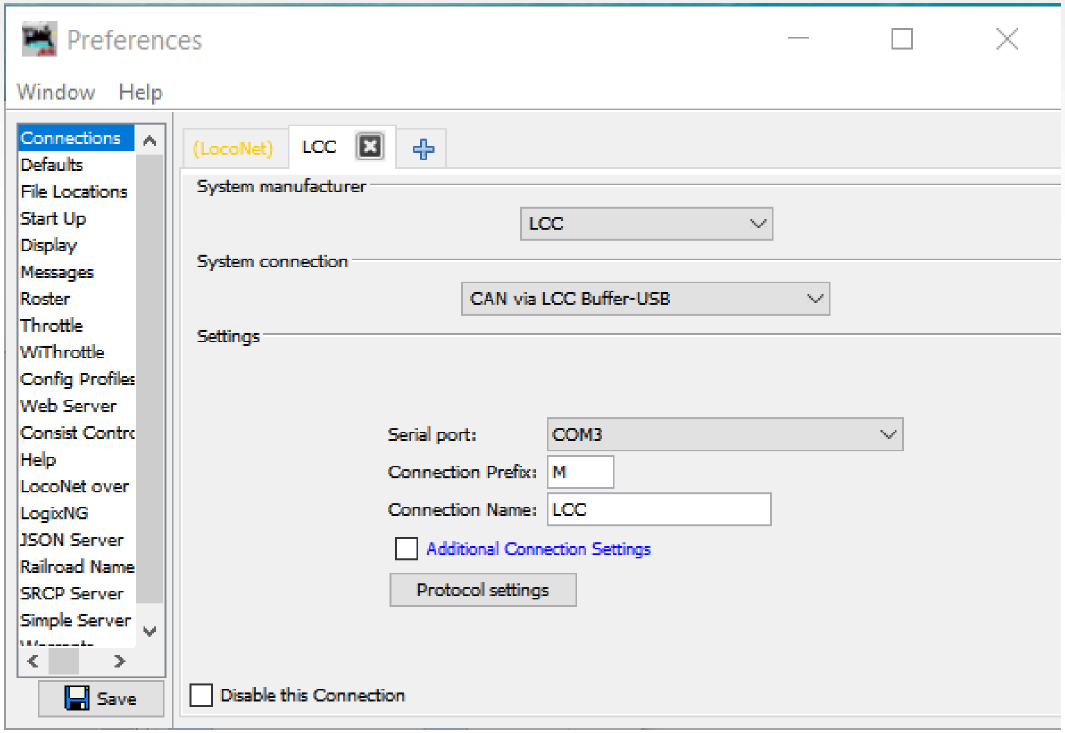
- One of the key concepts of the LCC protocol is that a node is self describing. Everything required for the node to initialize and operate in any system is contained within the node itself. No extra information is required from you, the user or from other sources to safely attach it to a network.
- Each node contains a globally unique address that prevents data conflicts with any other nodes on the network.
- Each node contains information about what it is and how to configure it. Yes, printed manuals are allowed. :)
- The node stores its own specific settings once the user has entered them. This allows a node to be moved to another network and carry all of its information with it.

The CDI

- The ”CDI” is the name of a tool or program that allows you, the user to access and change the configuration information stored in each LCC node.
- First the CDI reads the network to find all attached nodes and let you select the one for configuration.
- Once chosen, the CDI reads the node configuration data that defines the required presentation format.
- Once the proper format is available, then any stored information is entered into the data fields for user access, and optional modification.

The JMRI CDI

- To use the CDI in JMRI go into the Edit → Preferences → Connections and add a new connection.
- Choose whatever connection method is appropriate for your hardware. In this case it is an LCC Buffer-USB on COM3.
- Click Save and you should find JMRI immediately connected to the LCC network.



System manufacturer: LCC

System connection: CAN via LCC Buffer-USB

Settings

Serial port: COM3

Connection Prefix: M

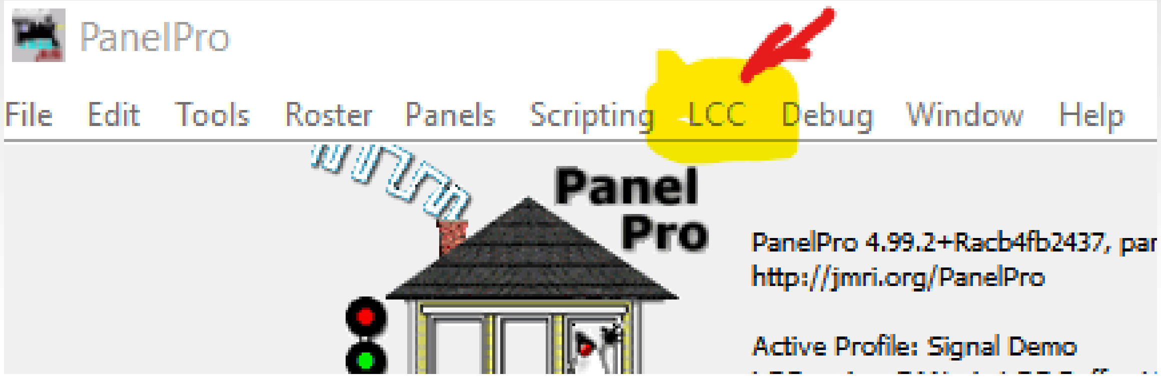
Connection Name: LCC

□ Additional Connection Settings

□ Disable this Connection

The JMRI CDI

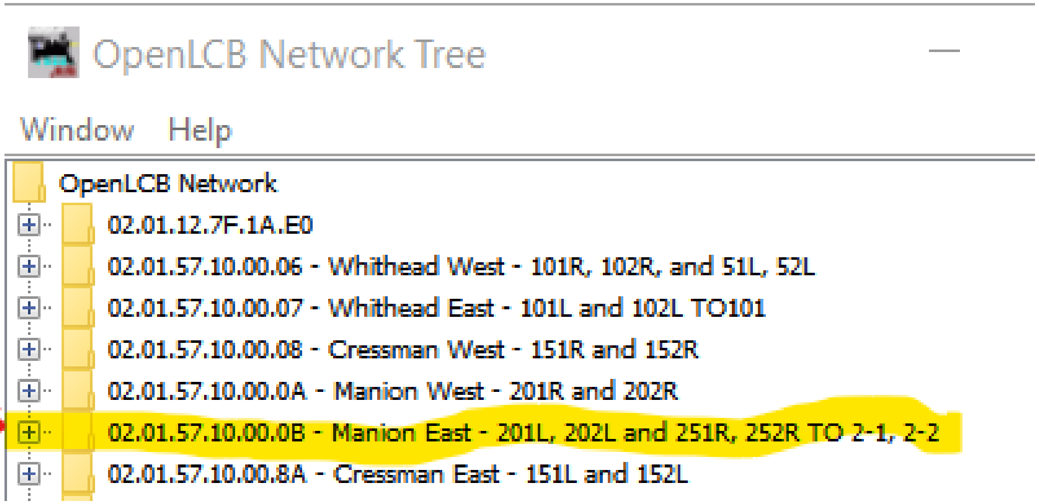
- The next time that you open JMRI it should have an "LCC" entry.



PanelPro 4.99.2+Racb4fb2437, par
http://jmri.org/PanelPro

Active Profile: Signal Demo

- Click on "LCC" and select "Configure Nodes" to see your current node list and choose the node you want to configure.



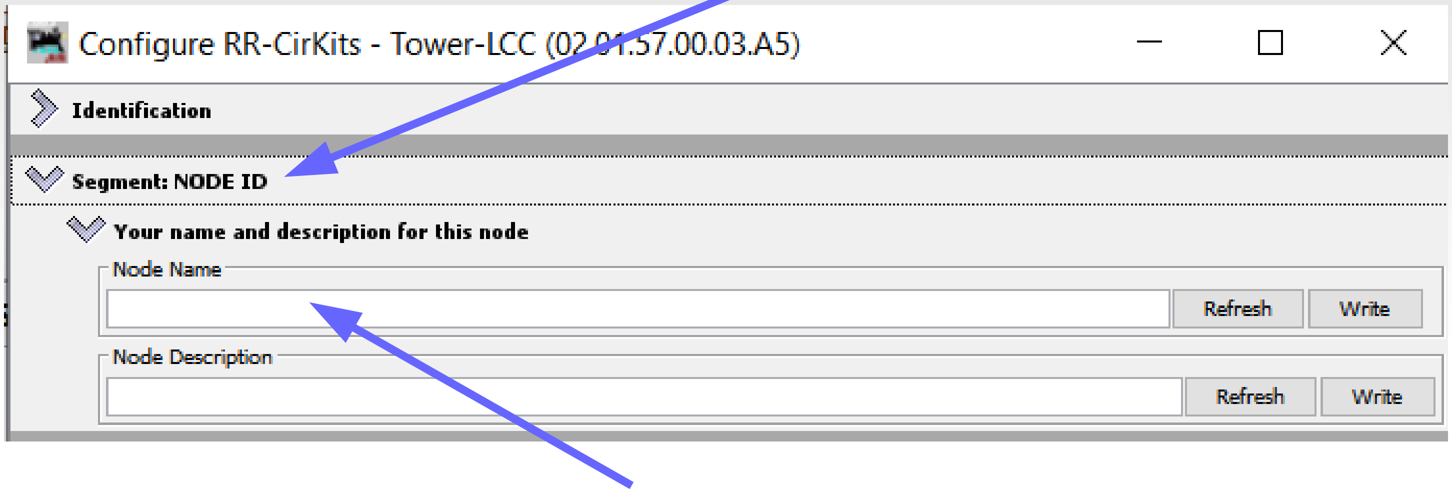
OpenLCB Network
02.01.12.7F.1A.E0
02.01.57.10.00.06 - Whithead West - 101R, 102R, and 51L, 52L
02.01.57.10.00.07 - Whithead East - 101L and 102L TO101
02.01.57.10.00.08 - Cressman West - 151R and 152R
02.01.57.10.00.0A - Manion West - 201R and 202R
02.01.57.10.00.0B - Manion East - 201L, 202L and 251R, 252R TO 2-1, 2-2
02.01.57.10.00.8A - Cressman East - 151L and 152L

The JMRI CDI

- Click on the ”+” to open the node, and then select ”Open Configuration dialog” to open the JMRI CDI tool for that node.
- Because the CDI tool reads its setup info from the node itself, the information that gets displayed will depend on which node you have opened. Obviously the option settings for a fast clock are going to be different from a signal driver.
- The positive aspect is that to configure a new device you do not need to install the most recent release of the CDI in order to be able to configure your new device. The truth is that the JMRI CDI tool was created before any existing hardware was even available to be configured.

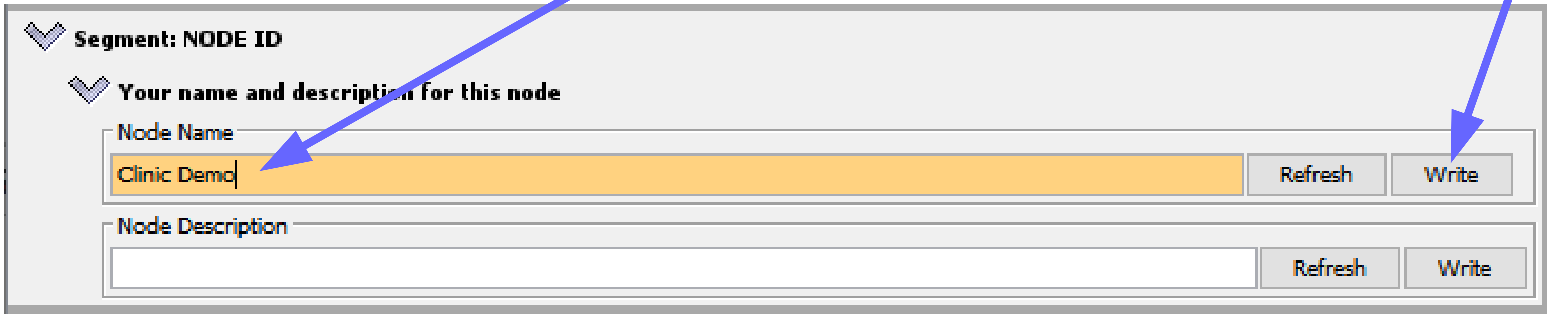
The JMRI CDI

- Lets open the first segment on the Demo node.



Configure RR-CirKits - Tower-LCC (02.01.57.00.03.A5)

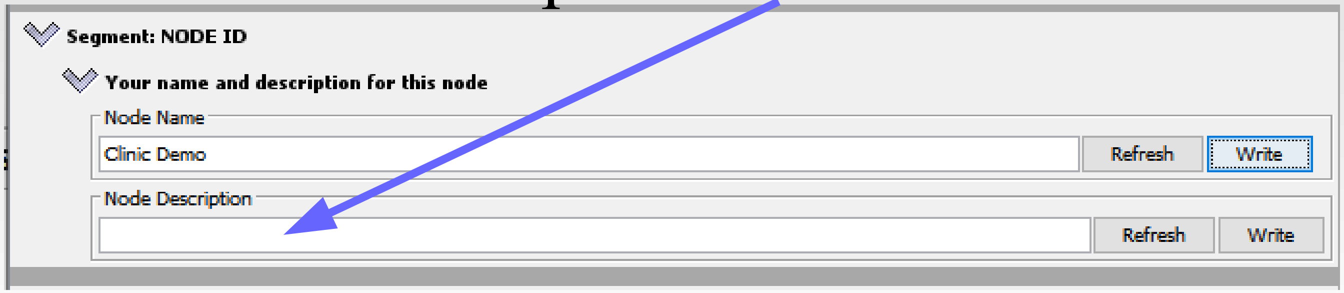
- We will give it a name. (e.g. Clinic Demo) Then click on 'Write' to save it to the node.



Node Name: Clinic Demo

The JMRI CDI

- We could add descriptions if we want to.



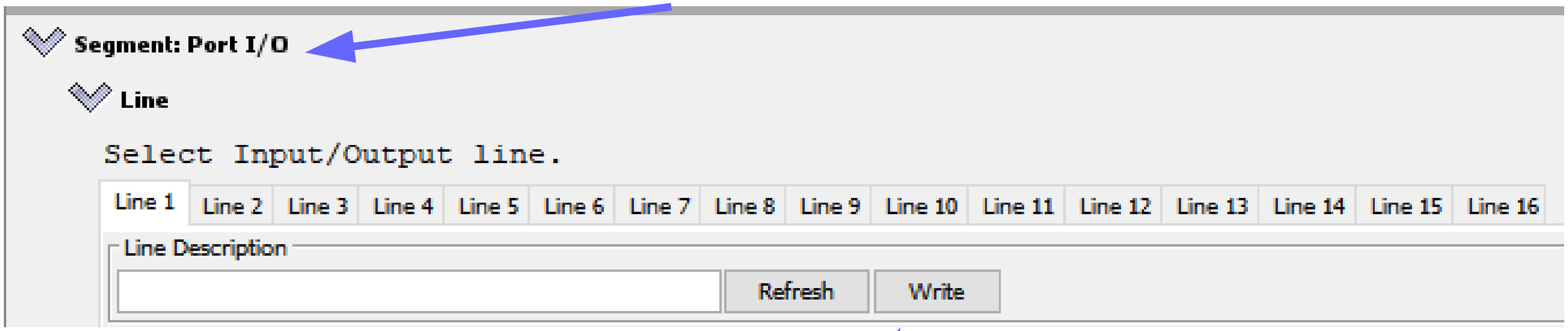
Segment: NODE ID

Your name and description for this node

Node Name: Clinic Demo

Node Description:

- Now open the Segment Port I/O to do something with the node.



Segment: Port I/O

Line

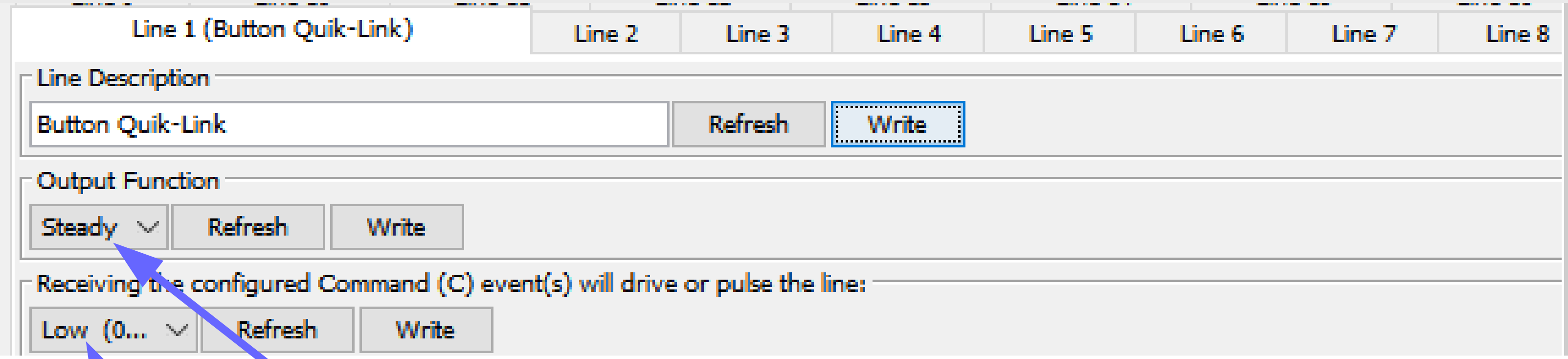
```
Select Input/Output line.
```

Line Description:

- This node has 16 I/O lines and currently we have ’Line 1’ selected. We will connect a ’Button Quik-Link’ currently configured to show Red and Green.
- Label it and Write the changes.

The JMRI CDI

- We add descriptions so we can remember what it is by the time the clinic is finished.



Line Description: Button Quik-Link

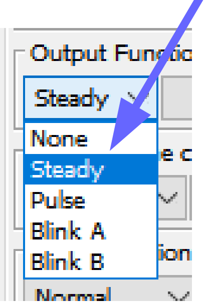
Output Function: Steady

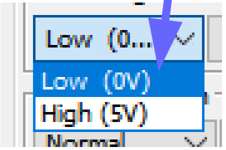
Receiving the configured Command (C) event(s) will drive or pulse the line: Low (0...

- The Button Quik-Link uses a single wire interface (plus power) to both control its color and monitor its button presses. This means we will configure this first line as both an output and an input.

- We want the output to be ”Steady” (not a pulse or blink) and to be active (on) when the voltage is low.

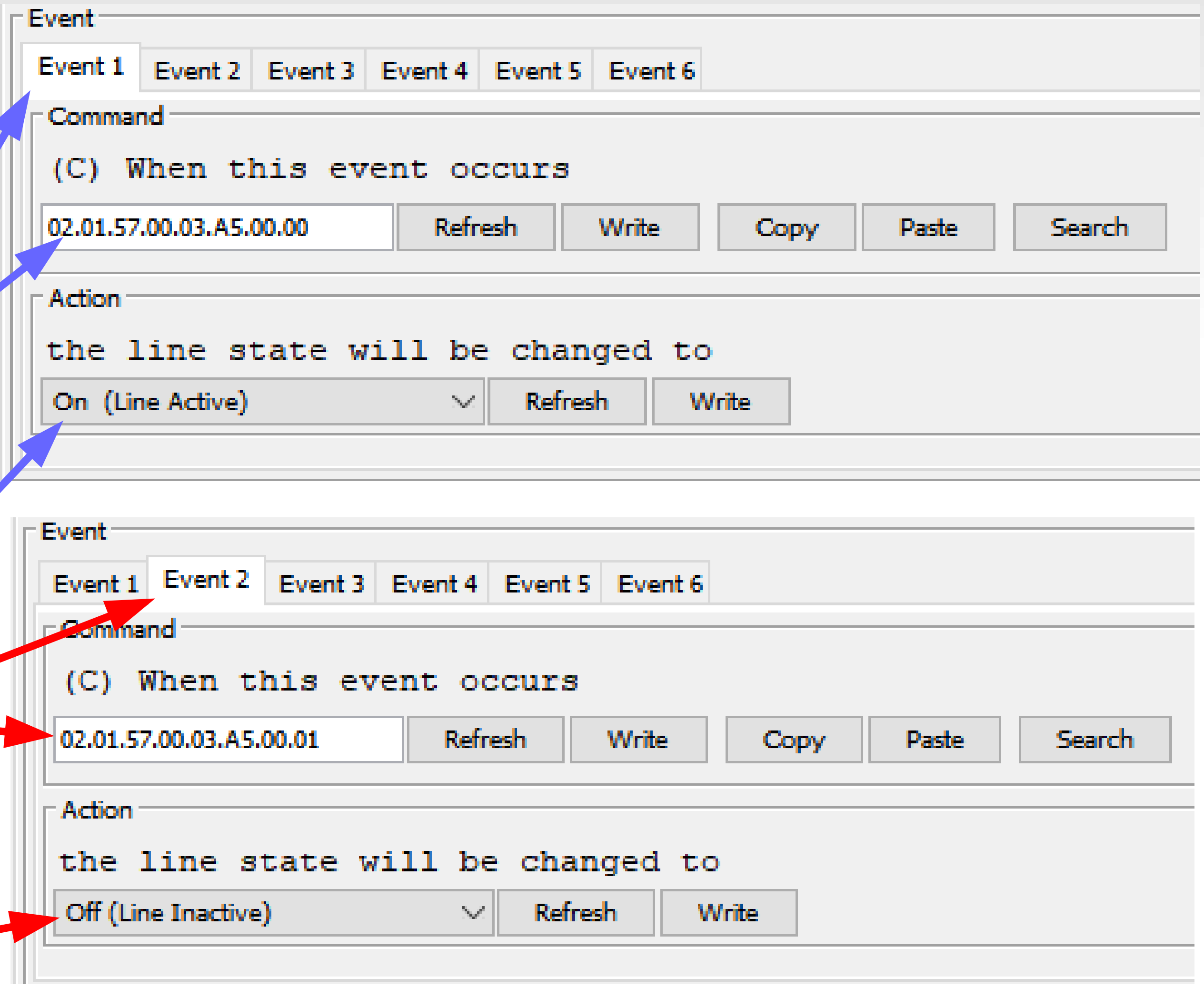
- Our Options:


Output Functio: Steady


Low (0...

The JMRI CDI

- Now we need to consider what needs to happen to control the color of the button LED.
- Event 1 and Event 2 control the on/off of the output. I.e. the lamp color.
- This first EventID Commands the output line. It changes it to ”On”.
- The next EventID also Commands the same output line. It changes it to ”Off”.



Event

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

Command

(C) When this event occurs

02.01.57.00.03.A5.00.00

Action

the line state will be changed to

On (Line Active)

Event

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

Command

(C) When this event occurs

02.01.57.00.03.A5.00.01

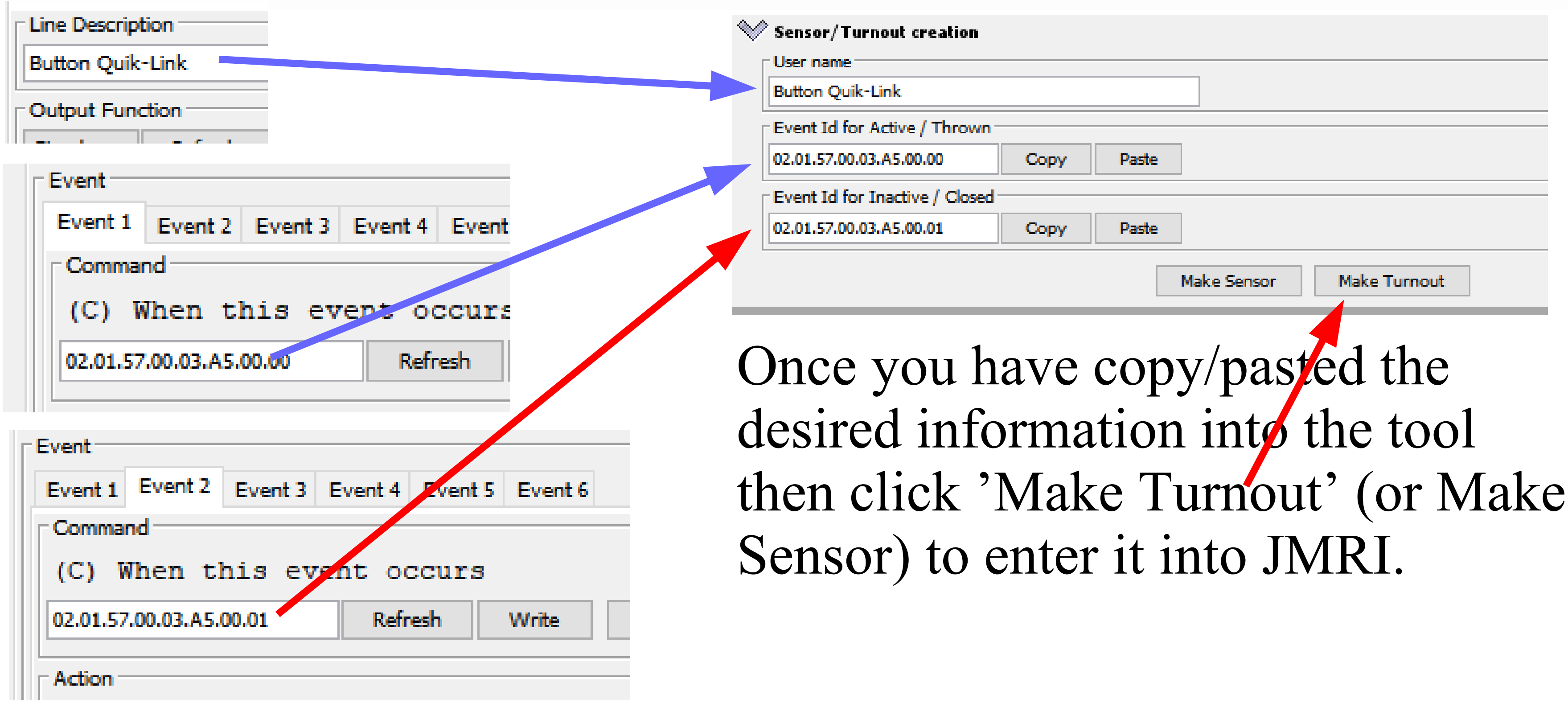
Action

the line state will be changed to

Off (Line Inactive)

The JMRI CDI

- Next we can add our lamp to a JMRI output table. JMRI calls all outputs ’Turnouts’ and inputs ’Sensors’. This is historical. Don’t argue, just go with the vocabulary.
- At the bottom of the CDI is a segment that will automatically add this node information into a JMRI table. Use Copy→ Paste to fill in.



Line Description
Button Quik-Link
Output Function

Event
Event 1 | Event 2 | Event 3 | Event 4 | Event
Command
(C) When this event occurs
02.01.57.00.03.A5.00.00 Refresh

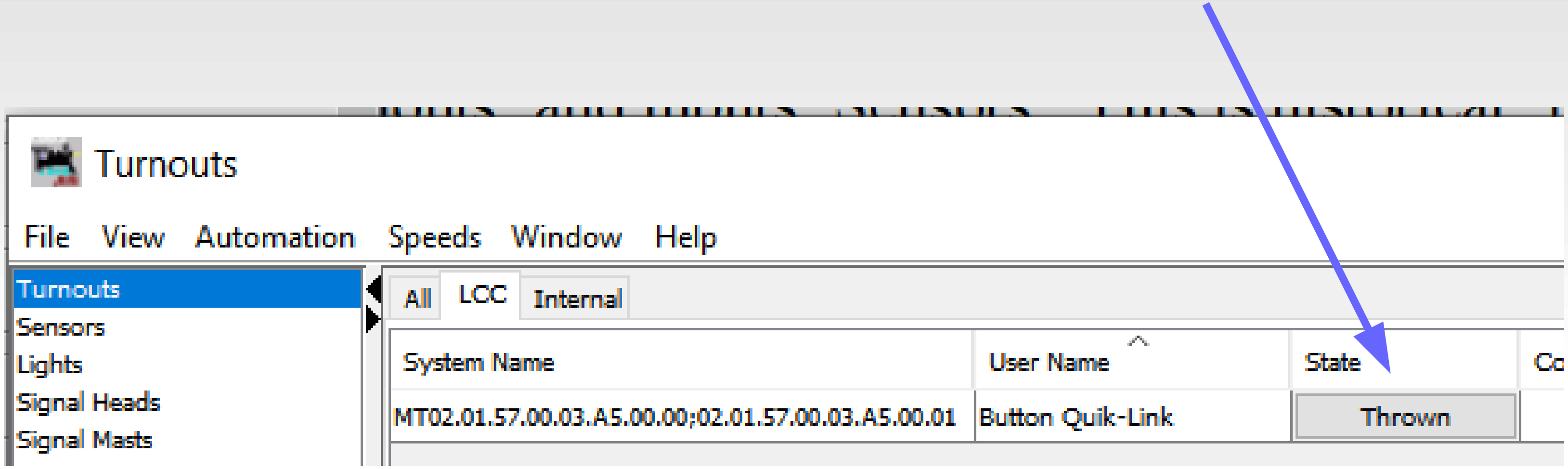
Event
Event 1 | Event 2 | Event 3 | Event 4 | Event 5 | Event 6
Command
(C) When this event occurs
02.01.57.00.03.A5.00.01 Refresh Write
Action

Sensor/Turnout creation
User name: Button Quik-Link
Event Id for Active / Thrown: 02.01.57.00.03.A5.00.00 Copy Paste
Event Id for Inactive / Closed: 02.01.57.00.03.A5.00.01 Copy Paste
Make Sensor | Make Turnout

Once you have copy/pasted the desired information into the tool then click ’Make Turnout’ (or Make Sensor) to enter it into JMRI.

The JMRI CDI

- From the JMRI table you can now control the lamp.

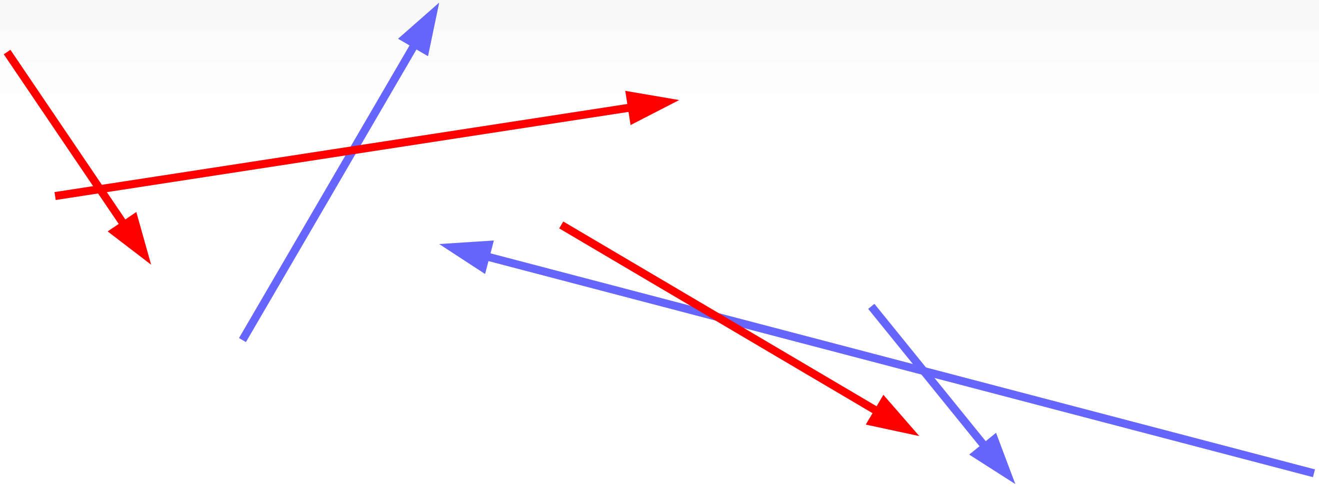


System Name	User Name	State	Co
MT02.01.57.00.03.A5.00.00;02.01.57.00.03.A5.00.01	Button Quik-Link	Thrown	

- Be sure to save your JMRI table by creating a panel file.
- If you are a JMRI user you know that you can use these table entries for panel graphics. This is not a JMRI clinic, so we will leave that to others.

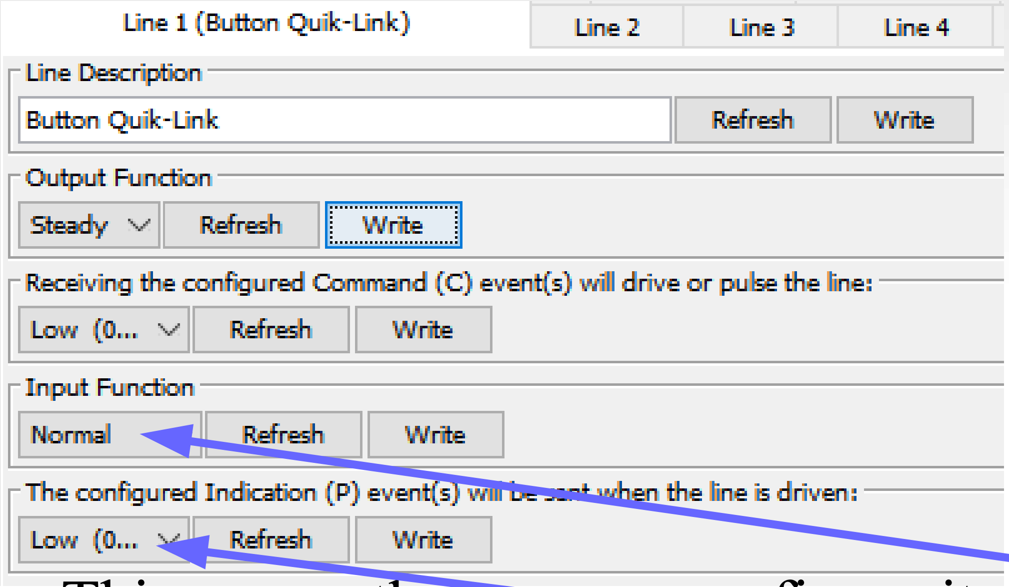
The JMRI CDI

- These Control ”EventIDs” are called ”Consumers” because they ’consume’ or do something with the EventID. These consumers can come from various places, other I/O lines, JMRI, control devices, or even from the same line used in ”input” mode.



The JMRI CDI

- Now we turn our attention to the Input Function called a ”Producer”. This button happens to be configured as ”On – Off”, like a switch.



Line Description: Button Quik-Link

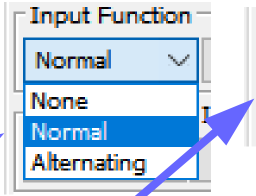
Output Function: Steady

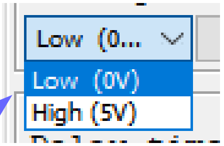
Receiving the configured Command (C) event(s) will drive or pulse the line: Low (0...

Input Function: Normal

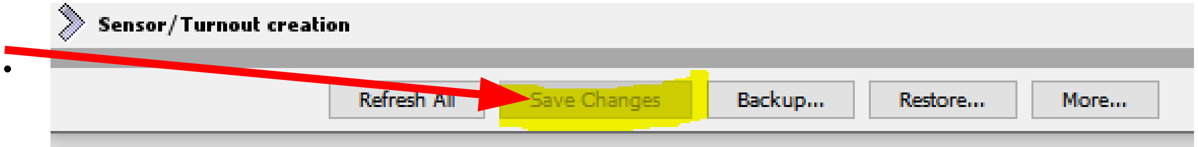
The configured Indication (P) event(s) will be sent when the line is driven: Low (0...

Input Options:



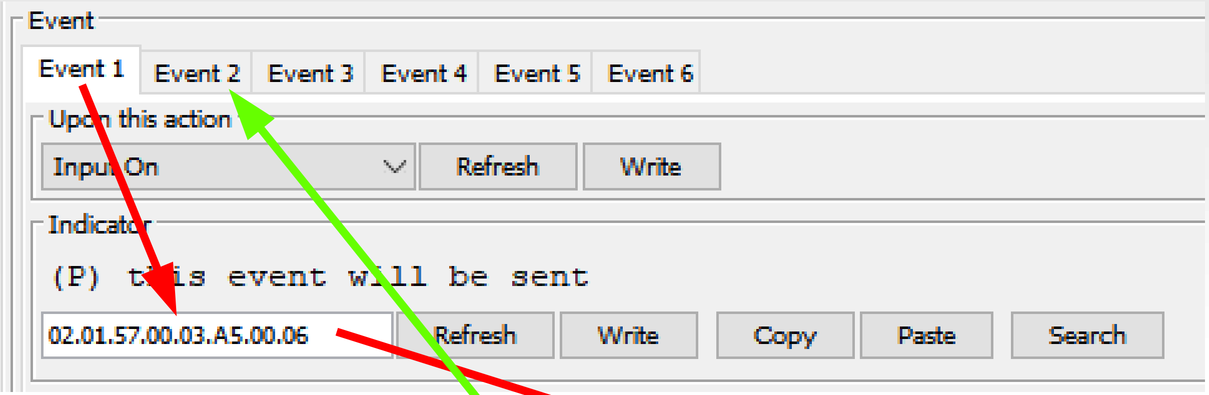


- This means that we can configure it as a ”Normal” switch or contact.
- It is also set to be active (on) when the voltage is low.
- Be sure to ”Write” any changes, or else go to the bottom of the CDI and choose ”Save Changes” to save any open items.



The JMRI CDI

- The producers (Inputs) have a default setting as shown here.



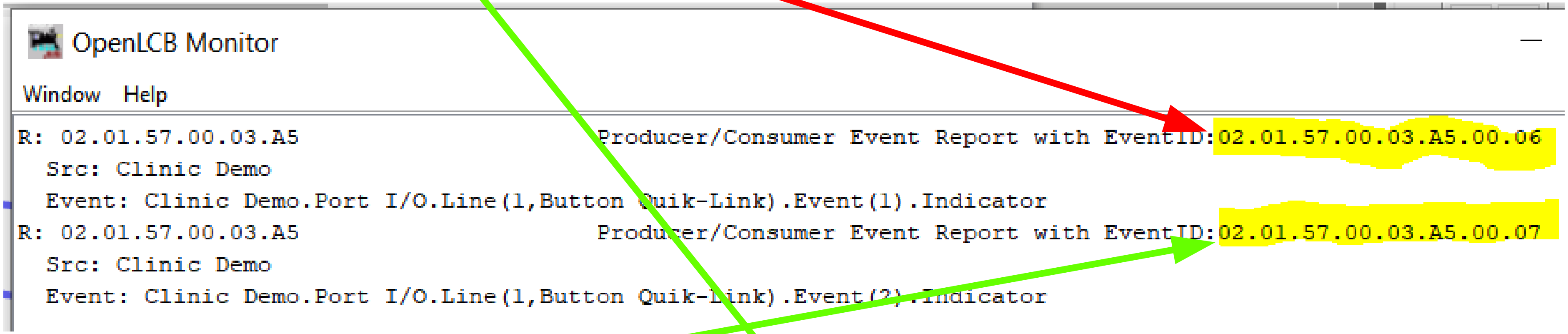
Upon this action: Input On

Indicator

(P) this event will be sent

02.01.57.00.03.A5.00.06

- Pressing/Releasing the button produces the following alternating commands:



```
R: 02.01.57.00.03.A5                         Producer/Consumer Event Report with EventID:02.01.57.00.03.A5.00.06
  Src: Clinic Demo
  Event: Clinic Demo.Port I/O.Line(1,Button Quik-Link).Event(1).Indicator
R: 02.01.57.00.03.A5                         Producer/Consumer Event Report with EventID:02.01.57.00.03.A5.00.07
  Src: Clinic Demo
  Event: Clinic Demo.Port I/O.Line(1,Button Quik-Link).Event(2).Indicator
```

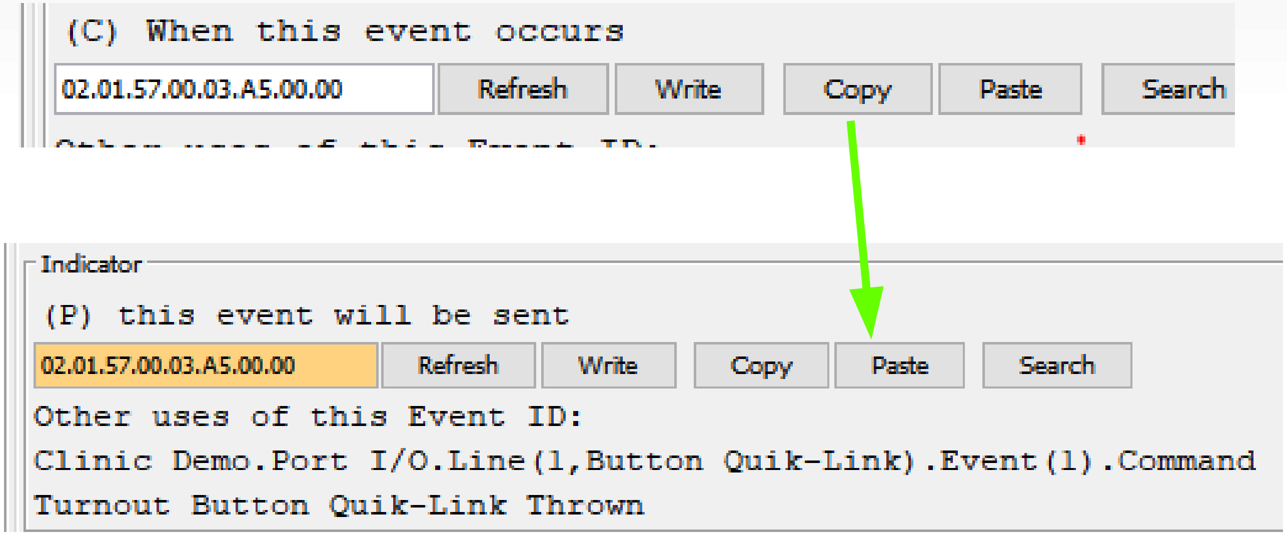
- The second EventID comes from Event2 above.

The JMRI CDI

- Now we have two pairs of EventIDs, one pair that controls the output. (lamp color for the button), and another pair that tells us if the button is pressed. We can extend this as far as we want to, but because all LCC EventID defualt values are globally unique we just have a pile of controllable outputs and inputs that use and generate a different bunch of EventIDs.
- To do useful things we need to pair these up. Only when a consumer EventID matches a producer EventID does anything happen in a syncronized manner.
- The nodes all come with unique EventIDs, but the CDI allows you to change them as desired to match them up in interesting ways. If we add these EventIDs into JMRI tables we can give them useful ”User Names”. Once we do that we can even search by the name that you have called it.

The JMRI CDI

- In order to have our button control the color of the indicator we need to change the events of one to match the other. Again use Copy→ Paste to make these changes. I prefer to use the default events from the primary function, but it is your layout and your standards.



(C) When this event occurs

02.01.57.00.03.A5.00.00

Other uses of this Event ID:

Indicator

(P) this event will be sent

02.01.57.00.03.A5.00.00

Other uses of this Event ID:

Clinic Demo.Port I/O.Line(1,Button Quik-Link).Event(1).Command

Turnout Button Quik-Link Thrown

- Be sure to do the same for Event 2 and then ’Write’ the changes.

Left to the Student

- Now that you know the basics your imagination is in charge.

- Traffic signals. Simple flashers to full four or six cycle control.
- Building lighting and signage.
- Day – Night lighting.
- Street and parking lot lighting.
- Operating bridge spans.
- Warehouse doors.
- Mine skips.
- All of the above could be controlled by individual devices, or centrally controlled for even more realism. Building lights could follow room lighting, bright in the evening, off late at night, then on again early in the morning. Traffic signals go to flashing mode late at night. Warehouse doors open when trains arrive. Etc.

- Signaling is way too complex a subject to cover in much detail in the short time we have here today.
- Lets open the floor to questions and comments to find out more about your expectations of LCC.

Acknowledgements

Developer Group

10 to 15 actively working on code at any time

- 25 to 50 regular contributors and supporters
- Many of the same people as supporting JMRI

OpenLCB User Group

Started November 2009

- In July 2024 we had 387 members

Layout Command Control User Group

- https://groups.io/g/layoutcommandcontrol

Started July 2012

- In July 2024 we had 512 members

Info

Users Groups:

https://groups.io/g/openlcb

https://groups.io/g/layoutcommandcontrol

To Join: openlcb+subscribe@groups.io
layoutcommandcontrol+subscribe@groups.io

Useful Links:

http://openlcb.org or http://openlcb.com

https://www.nmra.org/lcc

This is the official LCC information site, but its not always the most up to date.

Questions

- ?