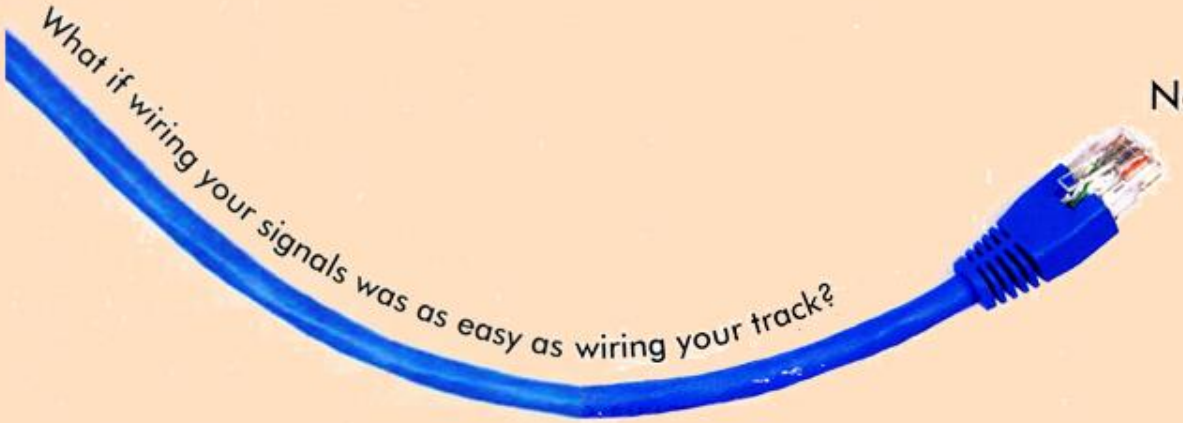
Introduction to LCC



Now it is.

Layout Command Control

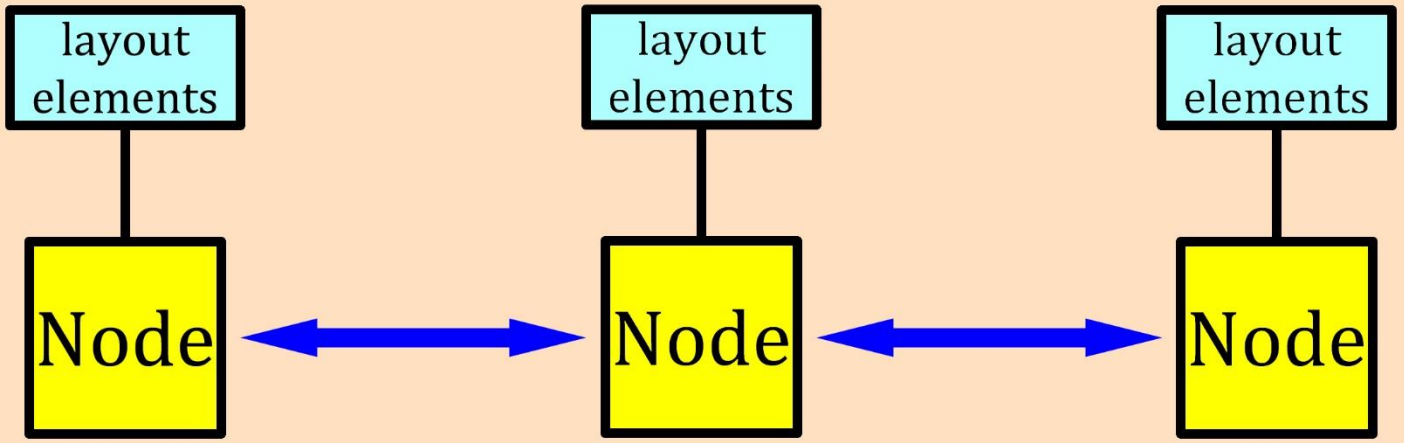
Mark F. Granville

9/17/22

What is LCC?

- **L**ayout **C**ommand **C**ontrol
- It is a set of NMRA standards that defines a peer-to-peer network allowing layout elements to talk to each other.
  - Signals, turnouts, detectors, push buttons, toggle switches, lights, panels, throttles, etc.
- Network components are self-describing and globally unique.
- LCC can be used with any train control method: DCC, DC, AC, Lionel, Marklin, etc.

LCC Architecture



An LCC system consists of a collection of **smart layout element controllers** called “Nodes” that communicate (exchange messages) with each other over a network.

The network can use one or more protocols such as CAN, Ethernet, Wi-Fi, etc.

LCC & DCC compliment each other

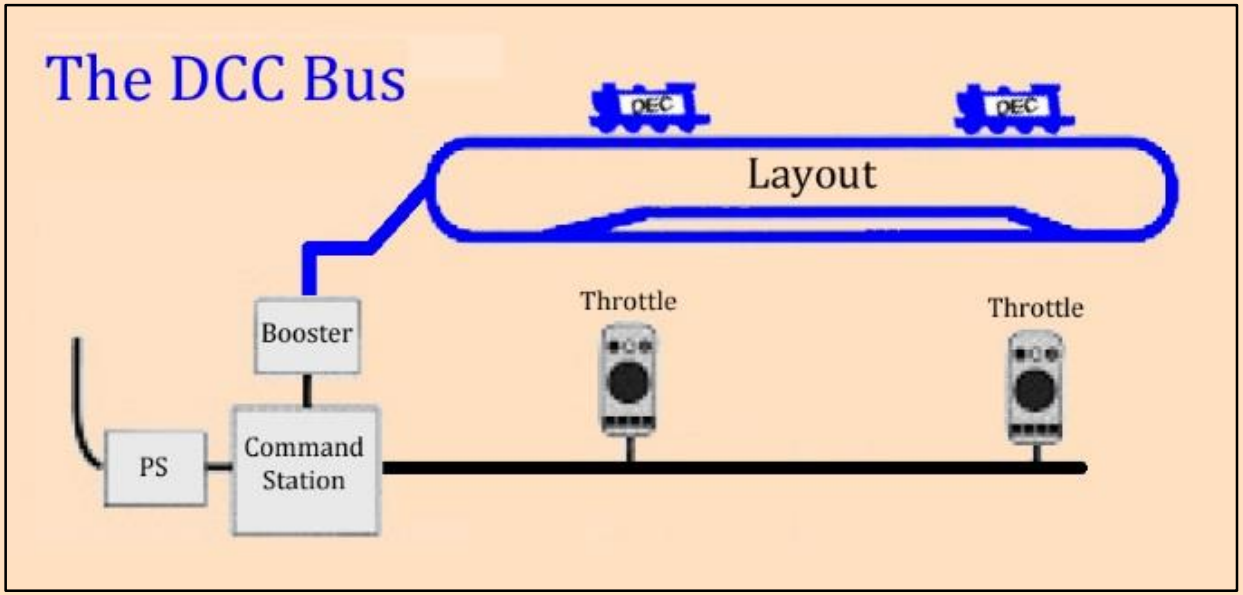
DCC replaced

GE ASTRAC
Keller Onboard
Dynatrol
CTC-16
RailCommand
Zimo Digital
etc.

LCC replaces

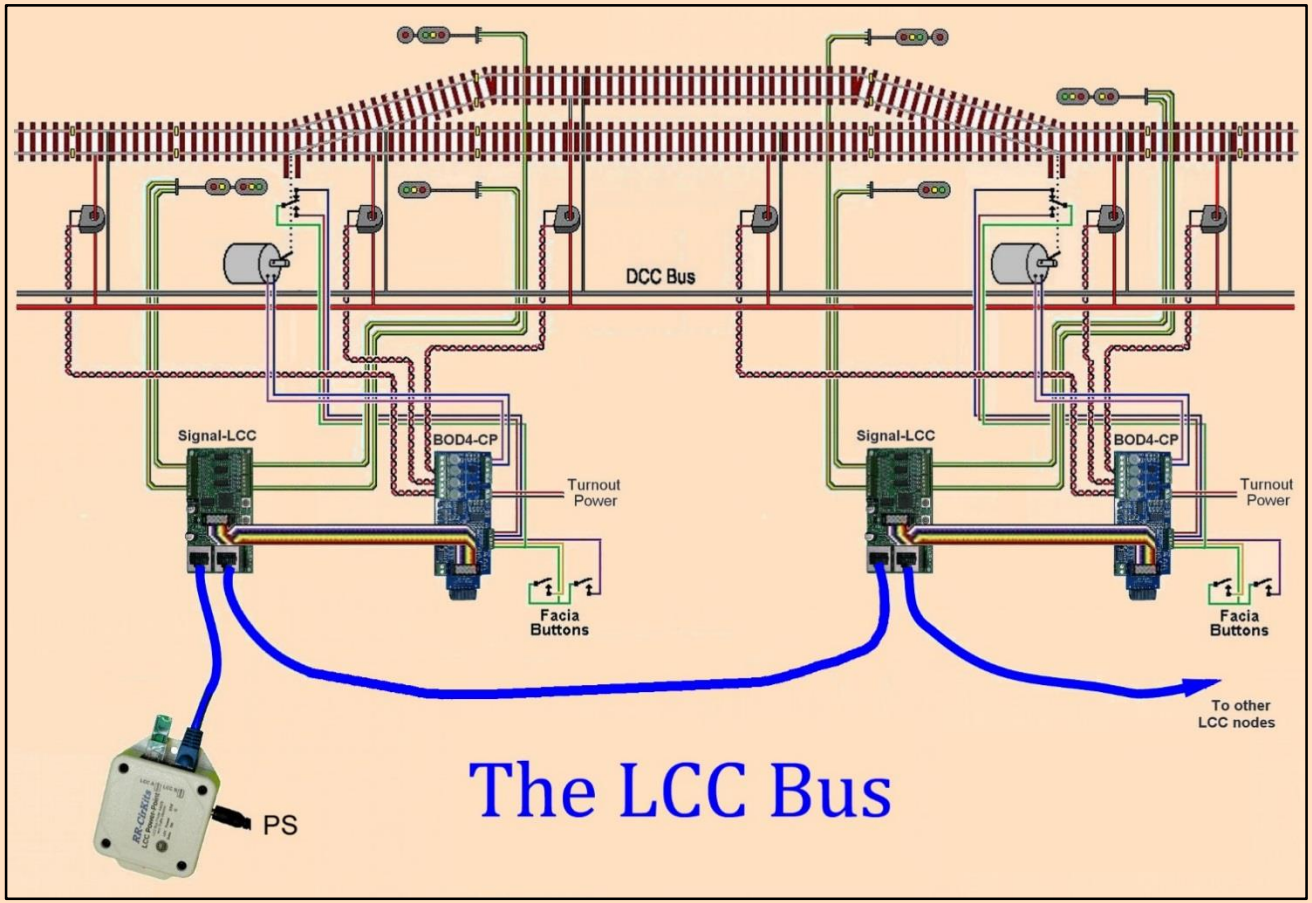
C/MRI (Chubb)
CBUS (MERG)
LocoNet (Digitrax)
XpressNet (Lenz)
AIU (NCE)
Accessory Decoders
etc.

DCC Runs Trains



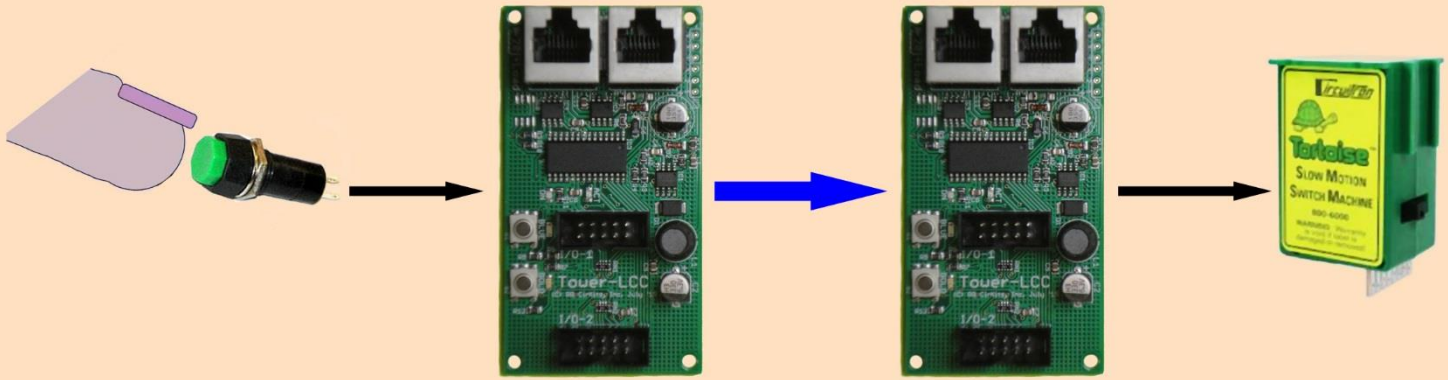
Master-Slave: All communication is between the command station (master) and the decoders (slaves). The slaves do not talk to each other.

LCC Runs Accessories



Peer-to-Peer: No central computer.
Nodes talk directly to each other.

LCC Basic Concept



event	EventID	event
	PRODUCER CONSUMER	

In LCC parlance, one event (pushing a button) causes a Node to Produce an EventID that is Consumed by the same Node or a different Node to cause another event (turnout points move).

LCC Basic Concept

Producers

Pushbutton

Switch

CT Coils

Both

Tortoise
DPDT Switch--Producer
Motor--Consumer

Consumers

Signal

Gate

LEDs

The most important concept for the user is that a "Node" is a piece of hardware that can send (Produce) an "EventID" or can react (Consume) to an "EventID".

A Producer event (such as pushing a button) is connected to a Consumer event (such as throwing turnout points) by associating each with the same EventID.

Commercial LCC Devices

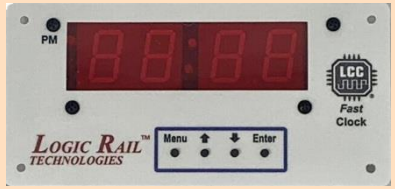
RR-CirKits (www.rr-cirkits.com)



- All purpose Nodes
- Real world I/O modules
- LCC-computer interface
- LCC power supply
- Connectors, flat cable, breakout boards, etc.

➢ Cat5 CAN bus

Logic Rail Technologies (www.logicrailtech.com)



- LCC fast clock

➢ Cat5 CAN bus

Train Control Systems (tcsdcc.com)

- DCC command station that is a Node

➢ Cat5 CAN bus and Wi-Fi

- LCC over Wi-Fi enabled throttles

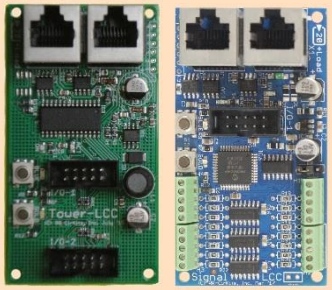
RR-CirKits Building Blocks

Interface

Computer USB to Cat5 CAN bus:

- Communicate with JMRI
- Configure LCC nodes

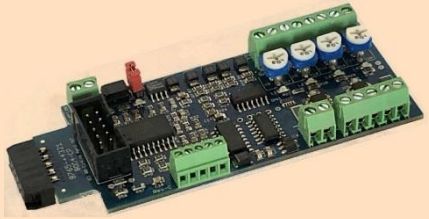
"Brains"



LCC Nodes using Cat5 CAN bus:

- Computation and logic
- Logic level I/O lines

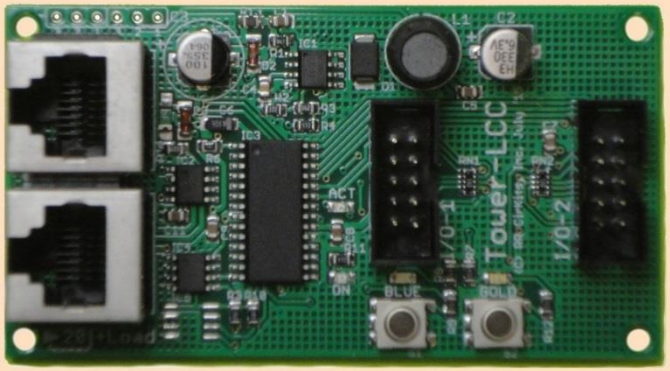
Function

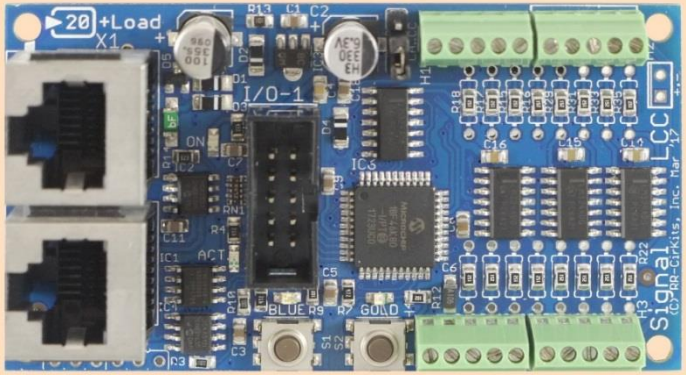


I/O Modules provide "real world" control:

- Turnouts, occupancy detectors, LEDs, etc.
- Connect to Nodes with 10-wire flat cable

RR-CirKits Nodes





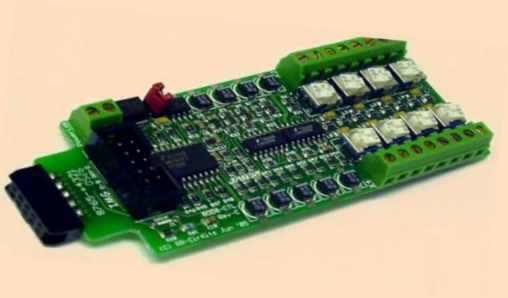
TowerLCC

- General applications
- 16 I/O lines in two 10-pin connectors
- Cat5 CAN bus

SignalLCC

- Signal applications
- 8 I/O lines in one 10-pin connector
- 16 LED drivers
- Cat5 CAN bus

RR-CirKits I/O Modules

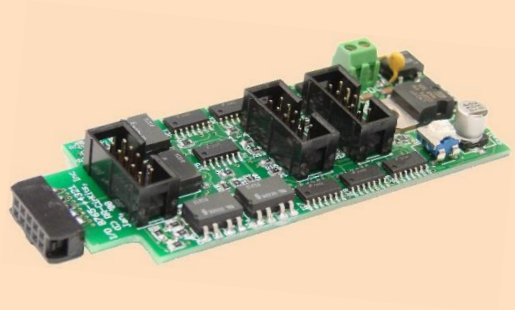


BOD-8

8 Block Occupancy Detector

- Uses remote CT coils

CT Coils

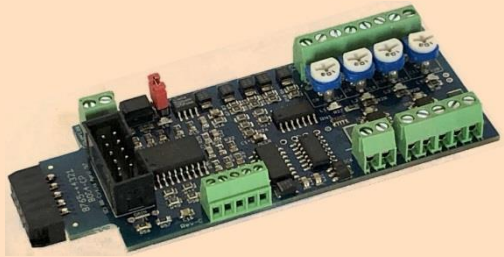


SMD-8

Stall Motor Driver for 8 stall motors

- Can drive up to 100 mA per line (external PS)
- Speed regulated output 4 to 12 VDC

Plus many others

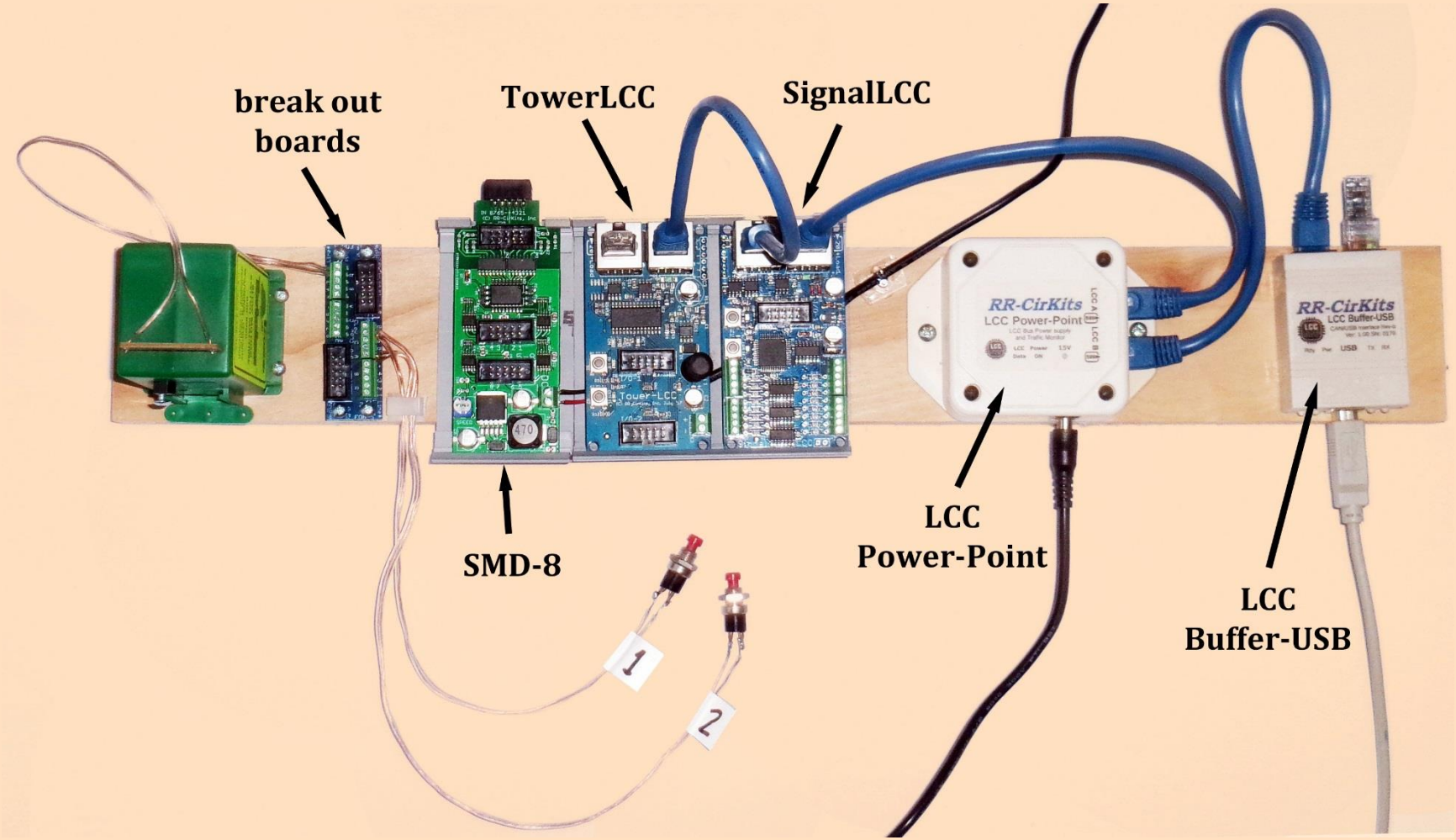


BOD-4-CP

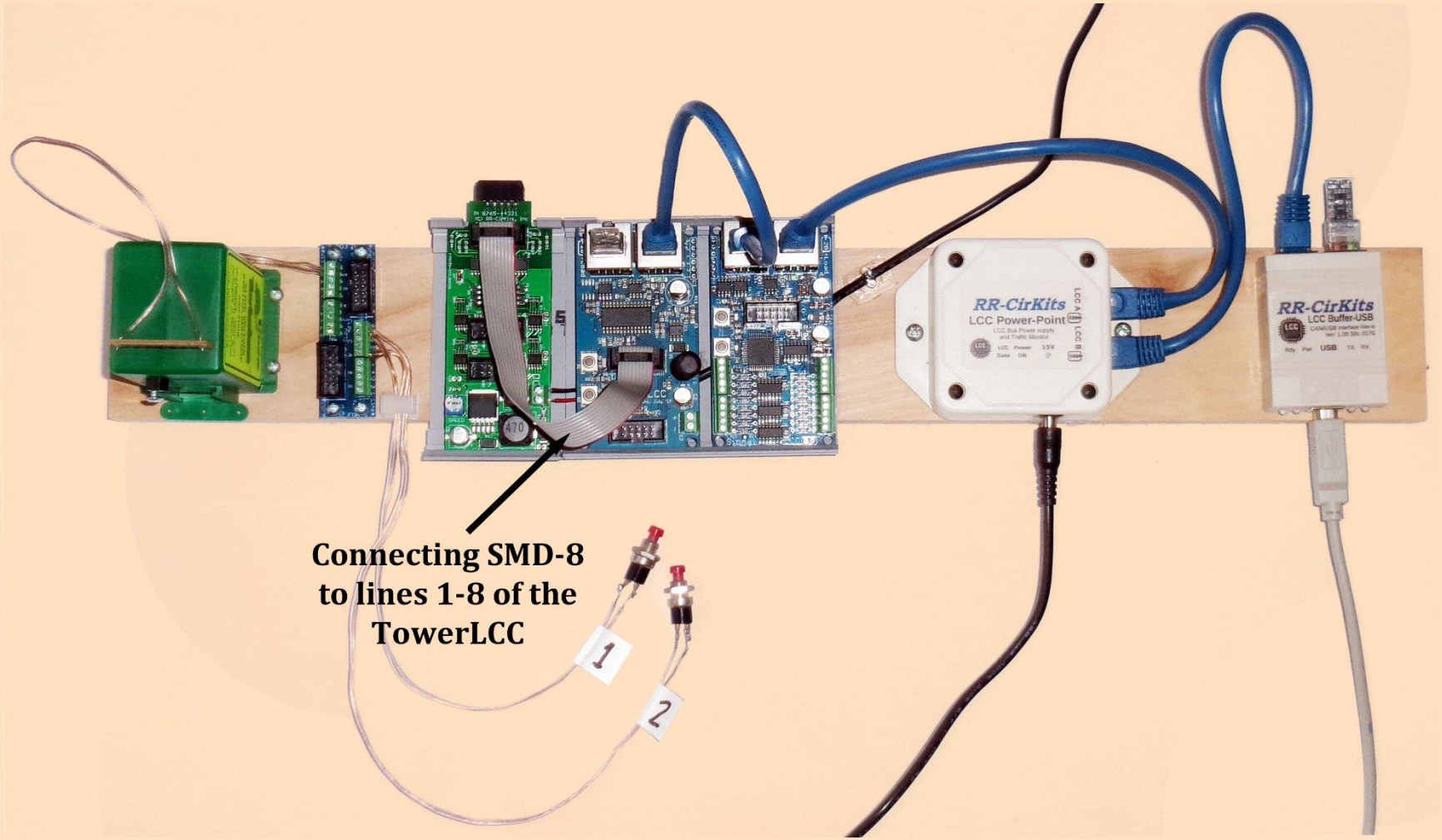
Control Point Functions

- 4 block occupancy detection using CT coils
- 2 turnout motor drivers (external PS)
- 4 input lines

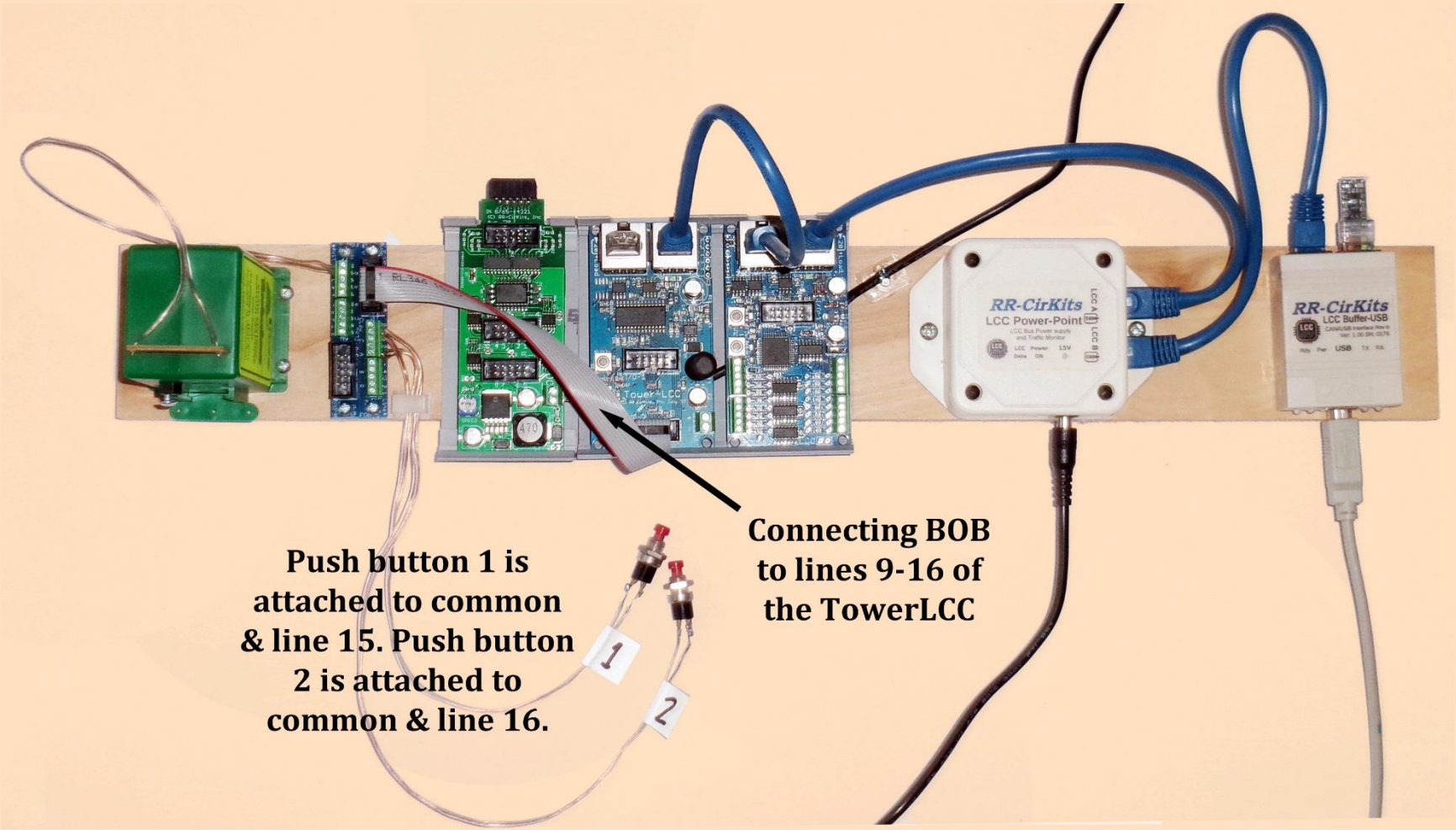
Demonstration Test Bed



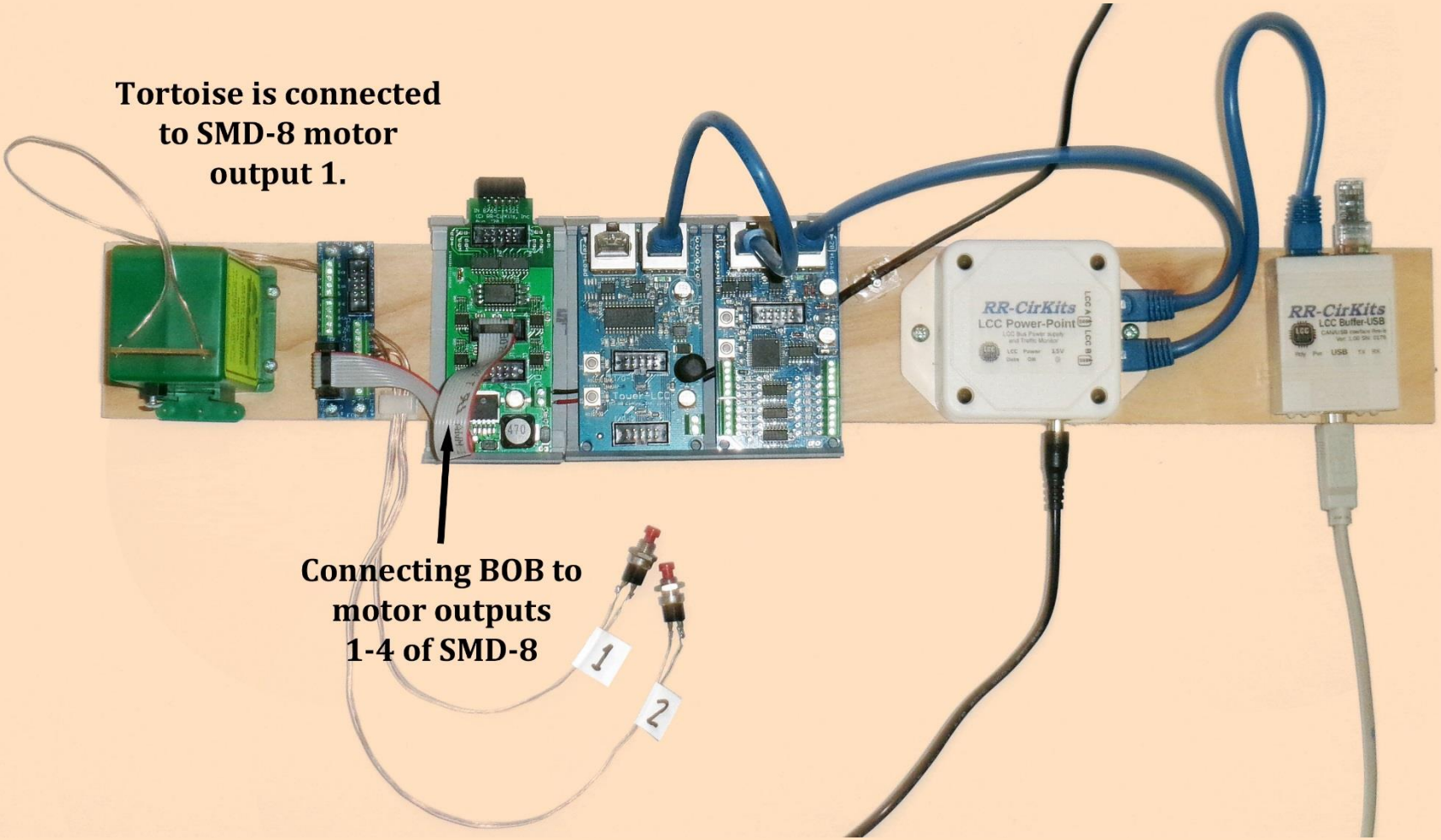
Demonstration Test Bed



Demonstration Test Bed



Demonstration Test Bed

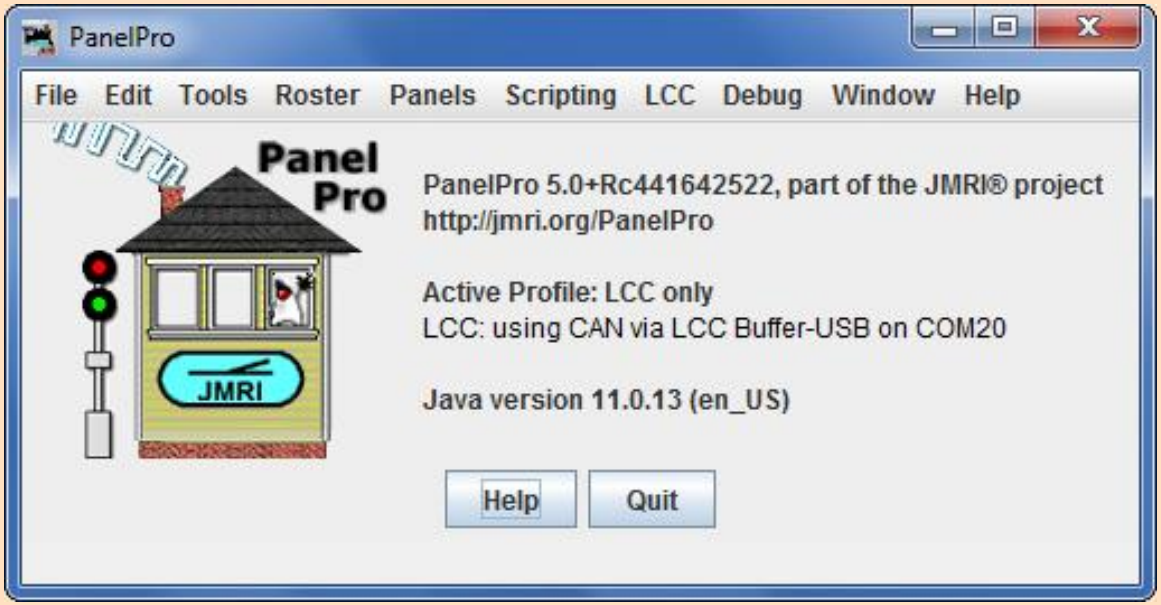


Configuring Nodes

Nodes are configured using a configuration tool from Java Model Railroad Interface or from Deepwoods Software.

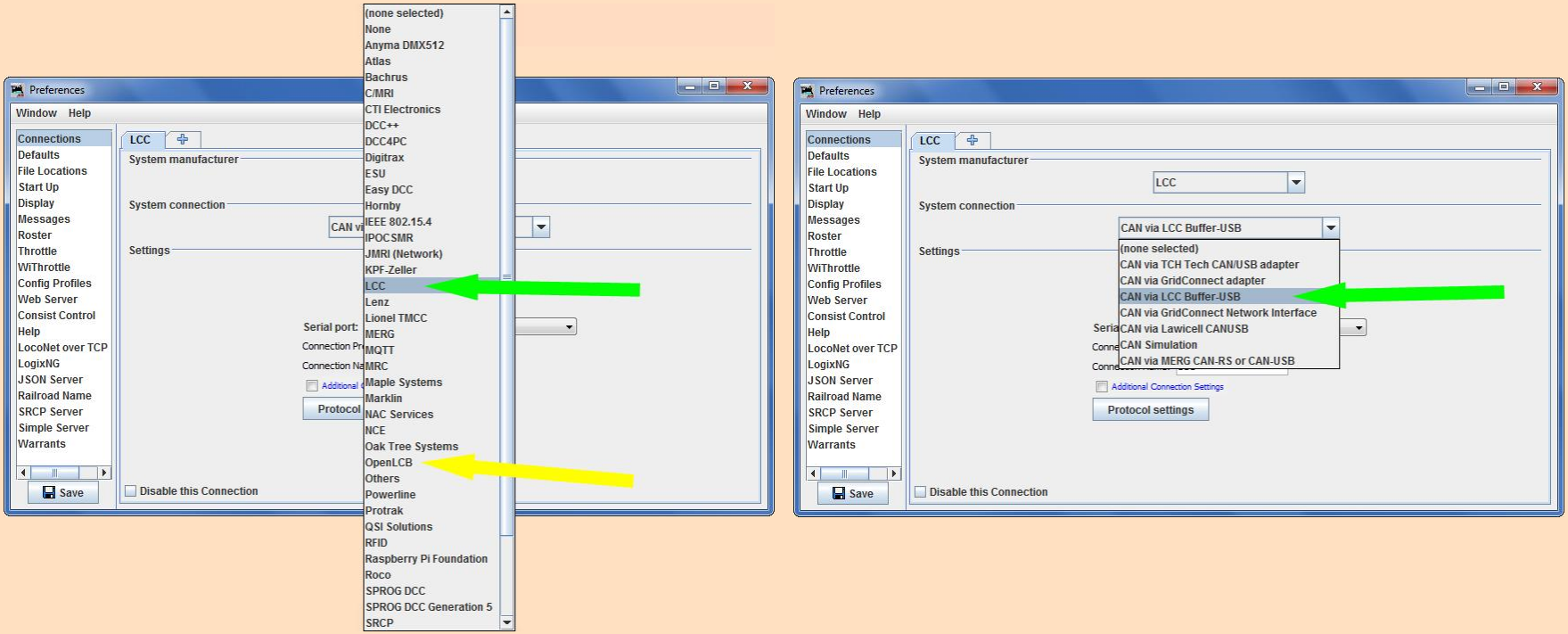
(https://www.jmri.org/)

(https://www.deepsoft.com/home/products/modelrailroadsystem/downloadmr/)



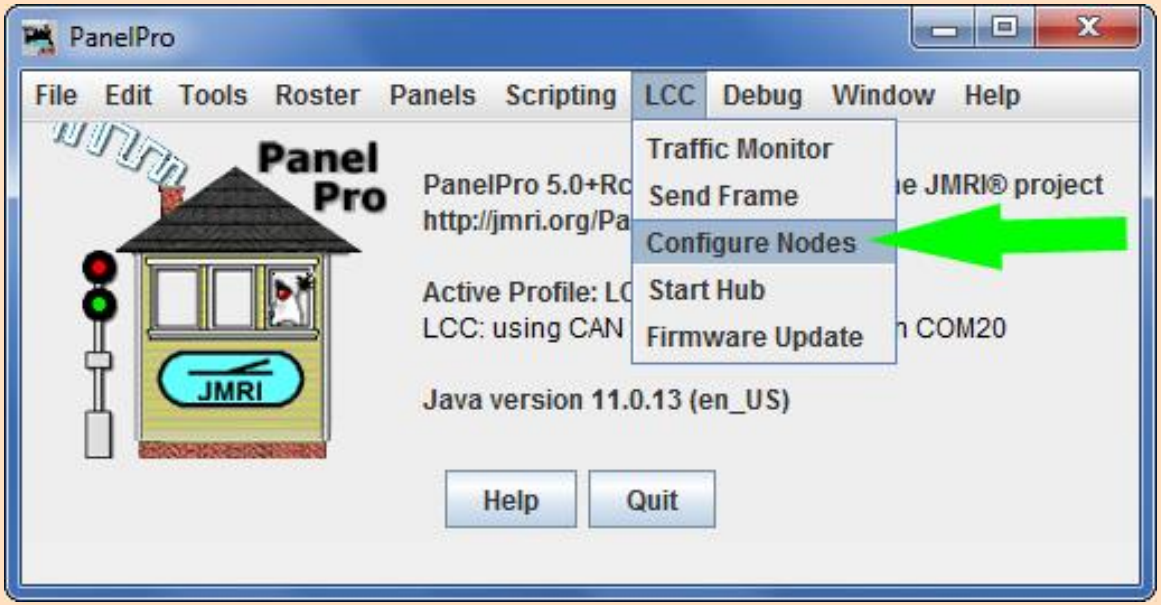
Configuring Nodes

Add a connection using either “LCC” or “OpenLCB” (no functional difference). Choose the hardware connection, in this case “CAN over LCC-Buffer (USB)”



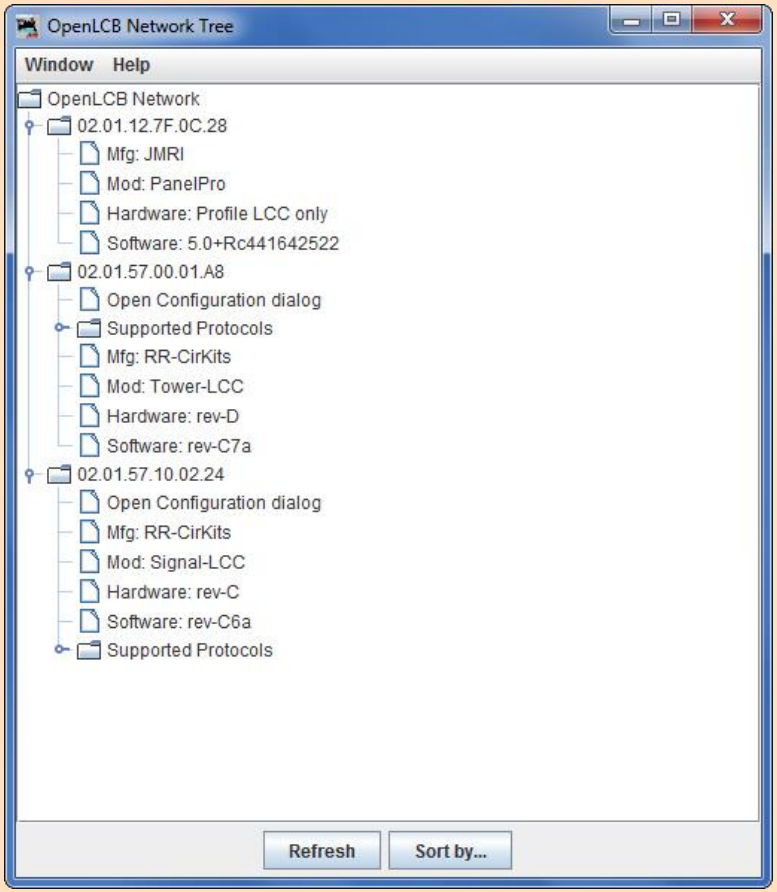
Configuring Nodes

Open the “LCC” (or “OpenLCB”) menu and click on “Configure Nodes”.



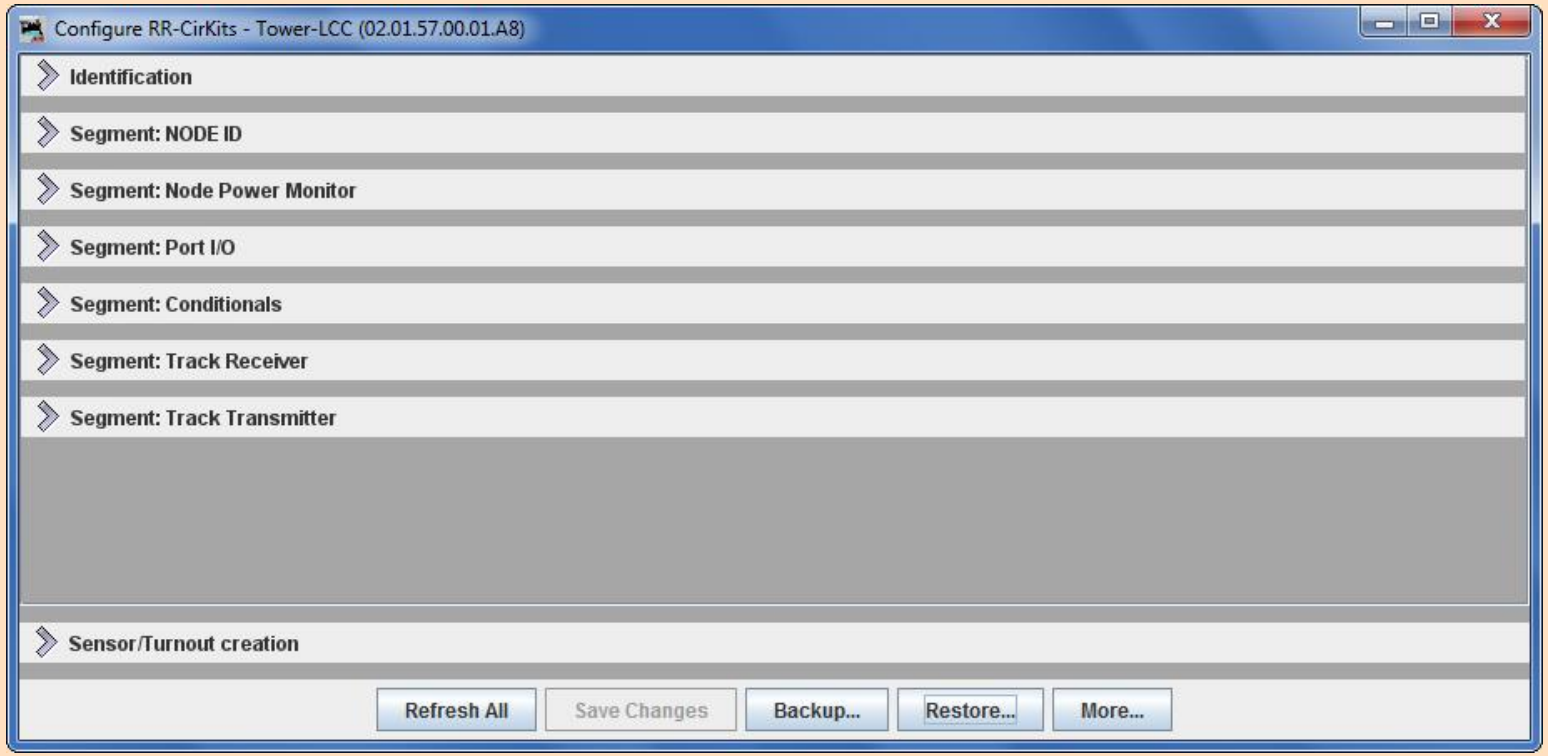
Configuring Nodes

Nodes are self-describing and globally unique.
Note the JMRI computer is also a Node.



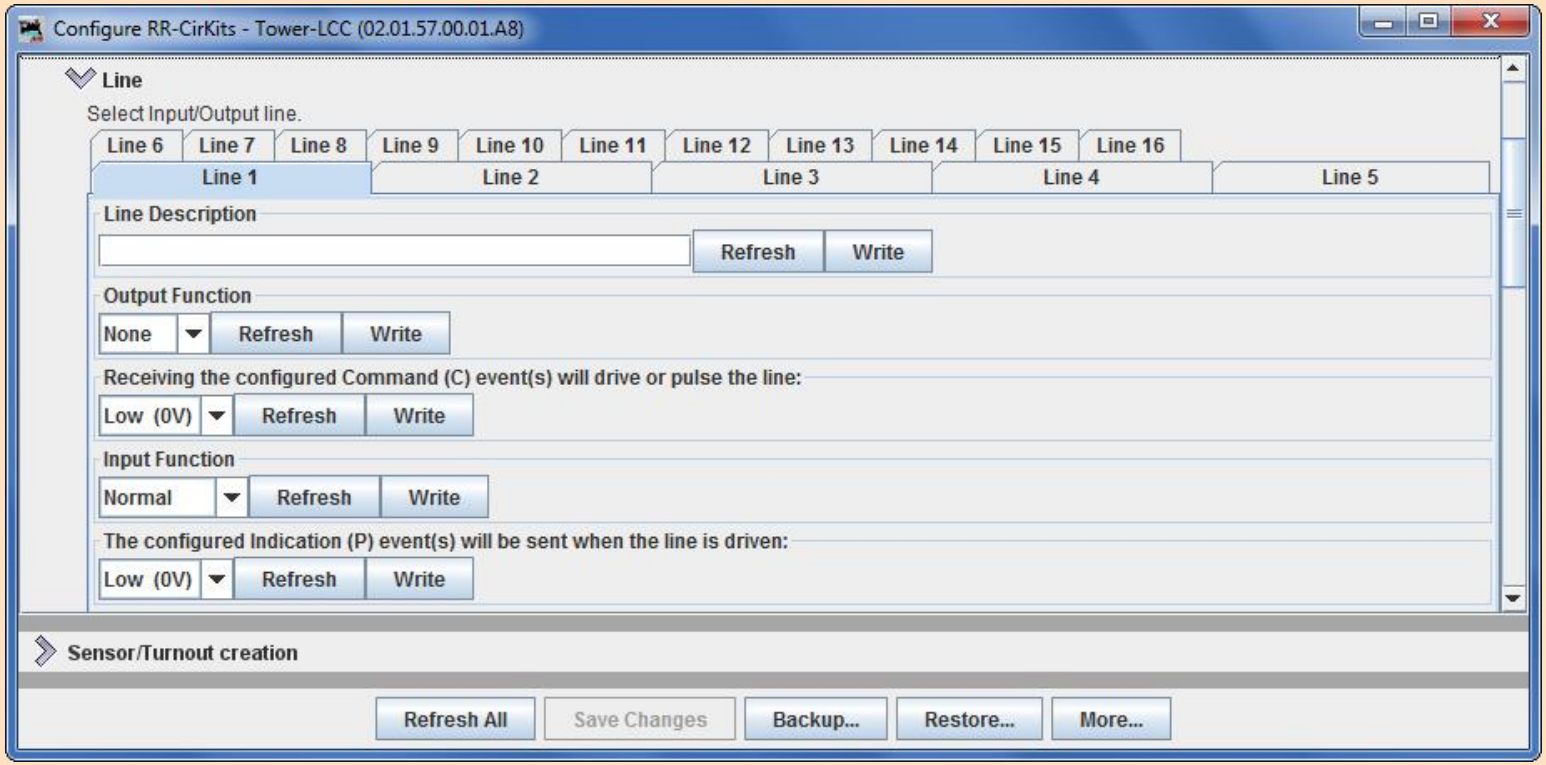
Configuring Nodes

Nodes are self-describing. All the information about how the Node can be configured is contained in its CDI.



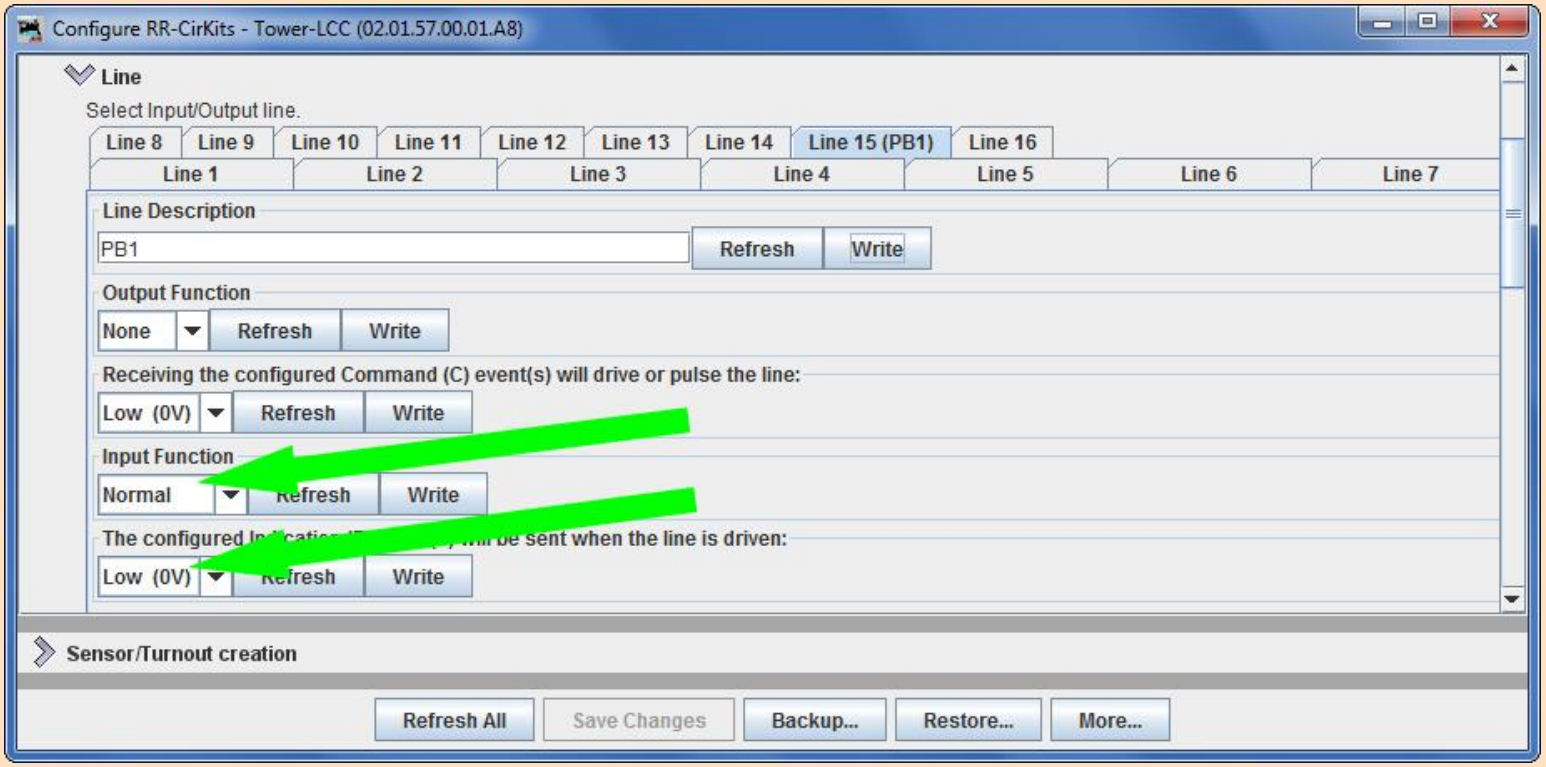
Configuring Nodes

Each of the TowerLCC I/O lines can be configured as an output, input, or both. I/O lines are logic level and each line can be configured so its “On” state is either 5V or 0V.



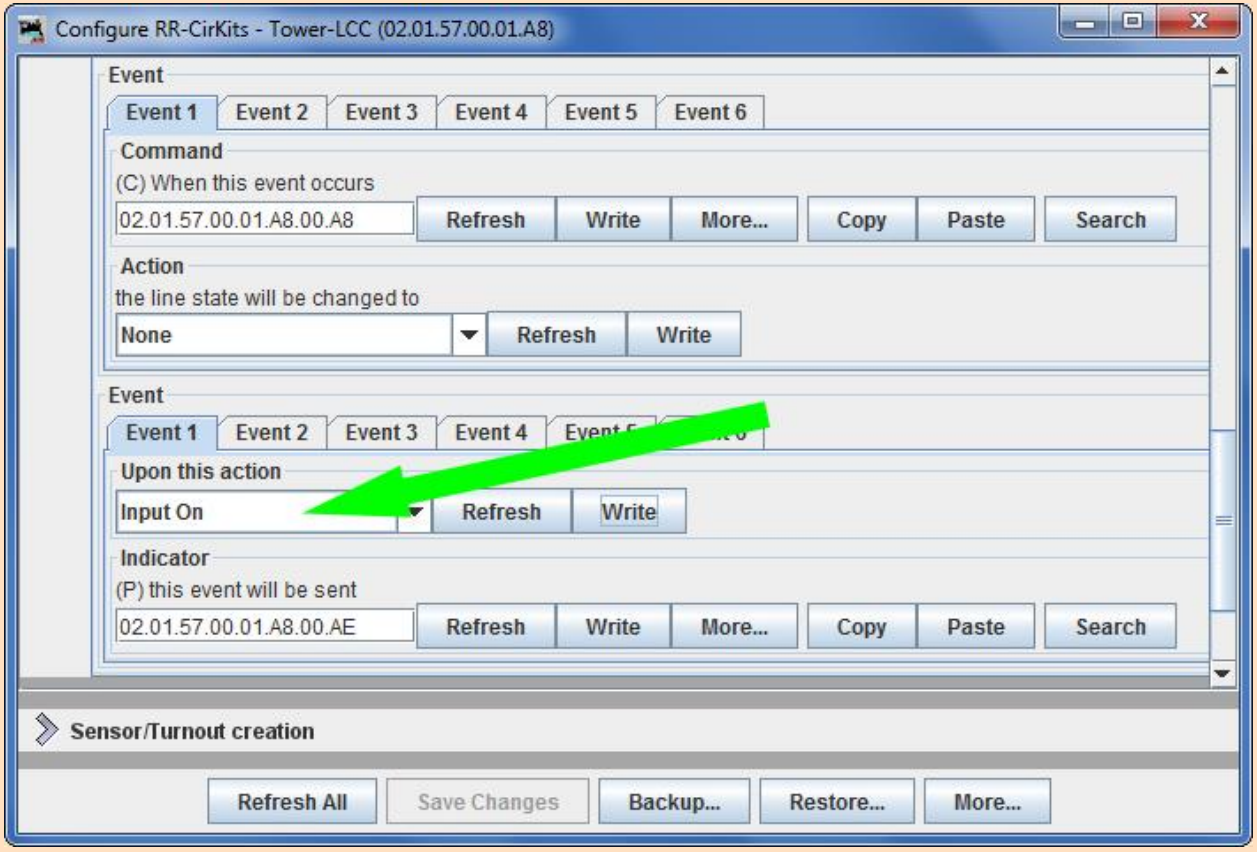
Configuring Nodes

We have two pushbuttons attached to lines 15 & 16 of the TowerLCC. The buttons are normally open so the line input is 5V until the button is pressed. Then it becomes 0V.



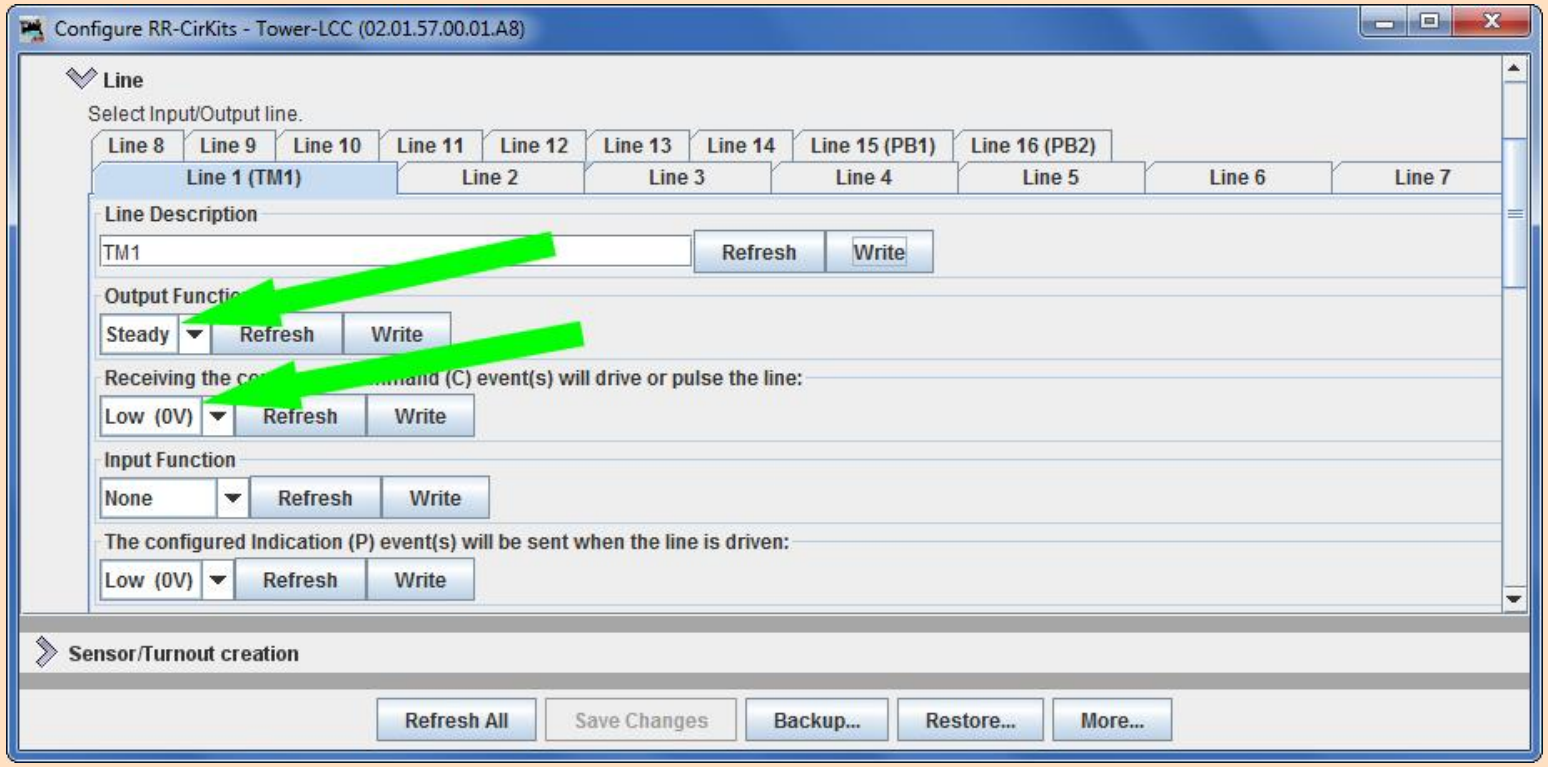
Configuring Nodes

We only need to define one event for each push button so an EventID is sent out when a button is pushed.



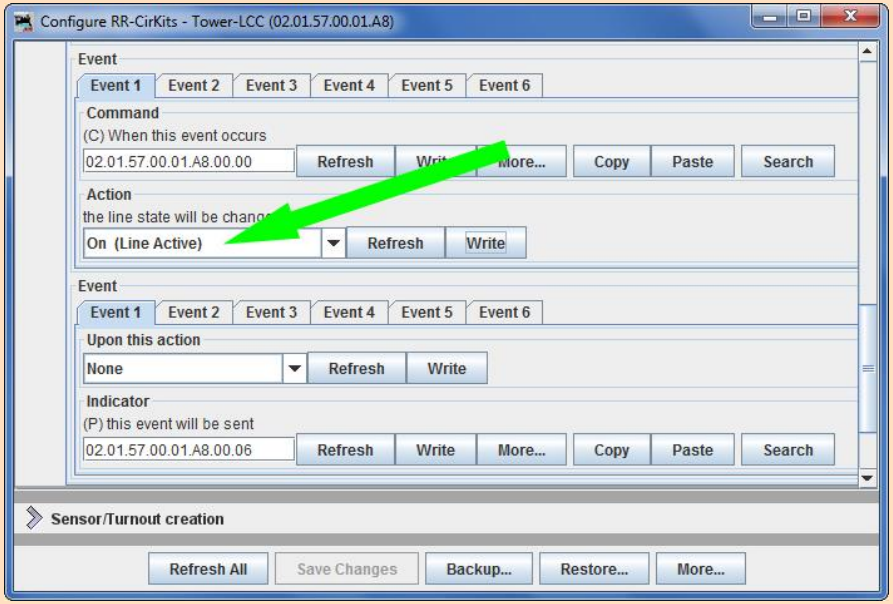
Configuring Nodes

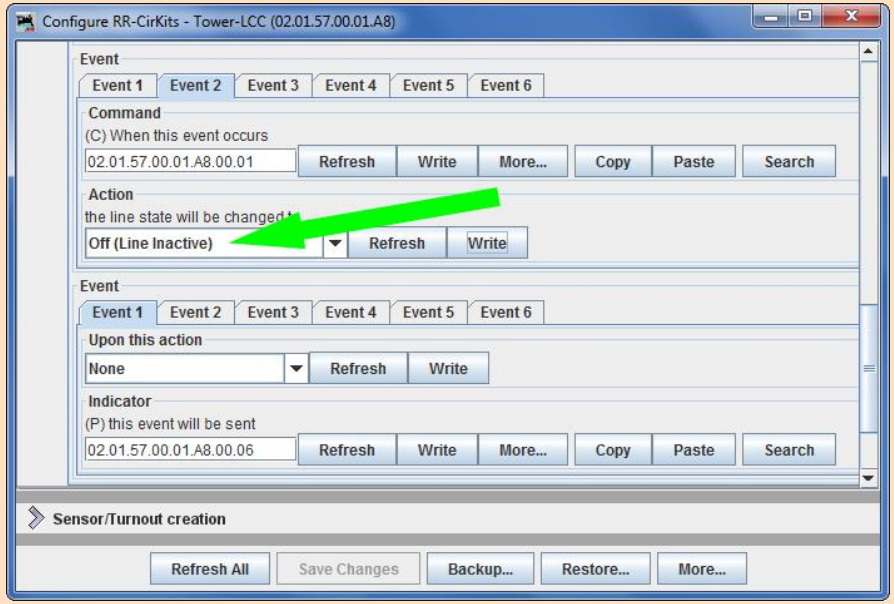
We have a tortoise switch machine attached to a SMD-8. The SMD-8 output for our tortoise is attached to TowerLCC line 1. We have chosen the “On” state of this line to be 0V.



Configuring Nodes

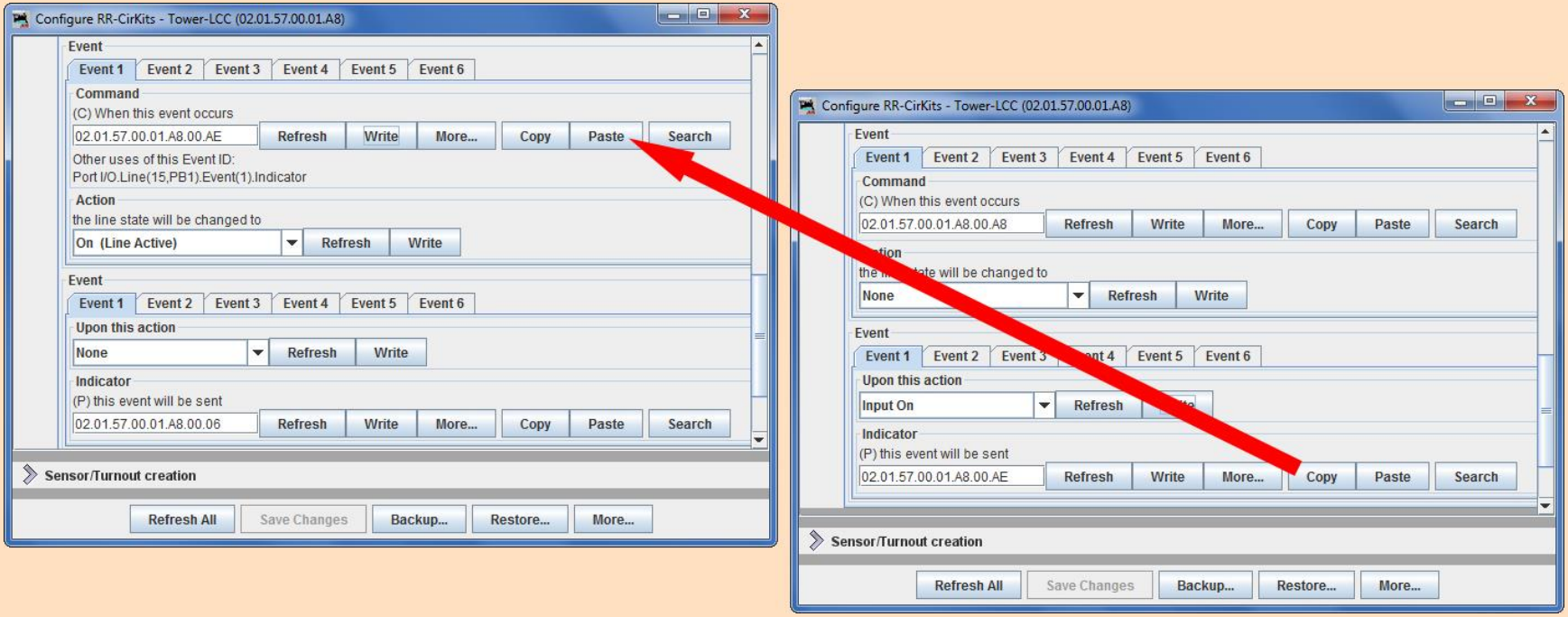
We need to configure two events for this line so one EventID will make the line “On” and another will make the line “Off”, resulting in the two tortoise positions.





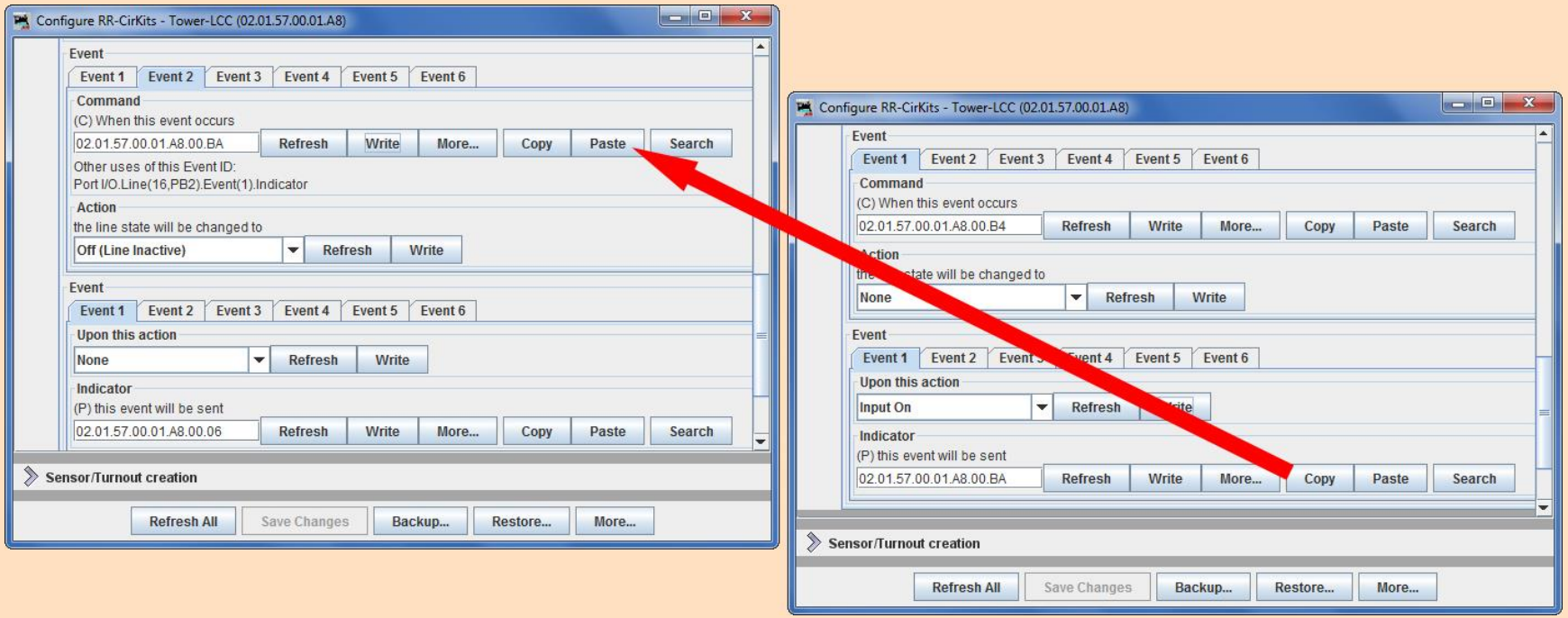
Configuring Nodes

To link push button 1 to the “On” position of the tortoise, we copy and paste the Producer event 1 EventID from push button 1 to the Consumer event 1 of the line controlling the tortoise.



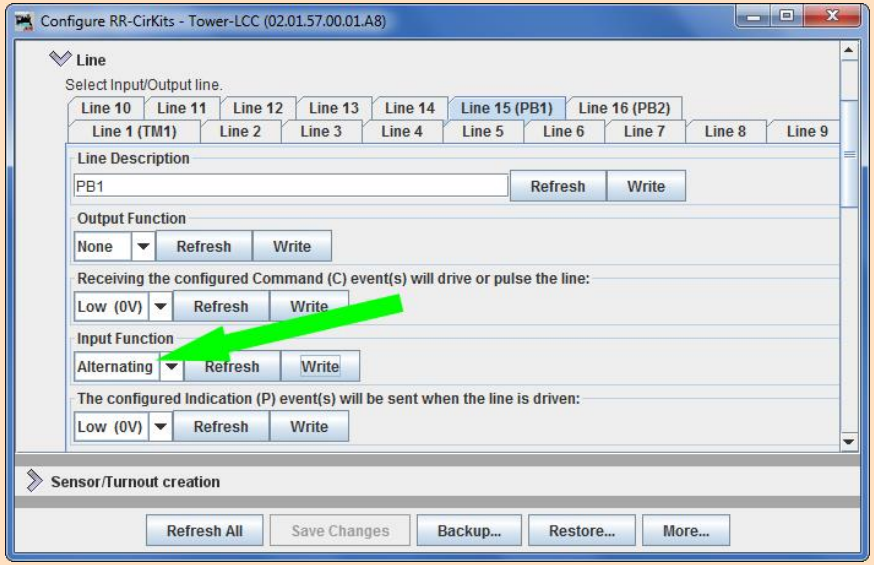
Configuring Nodes

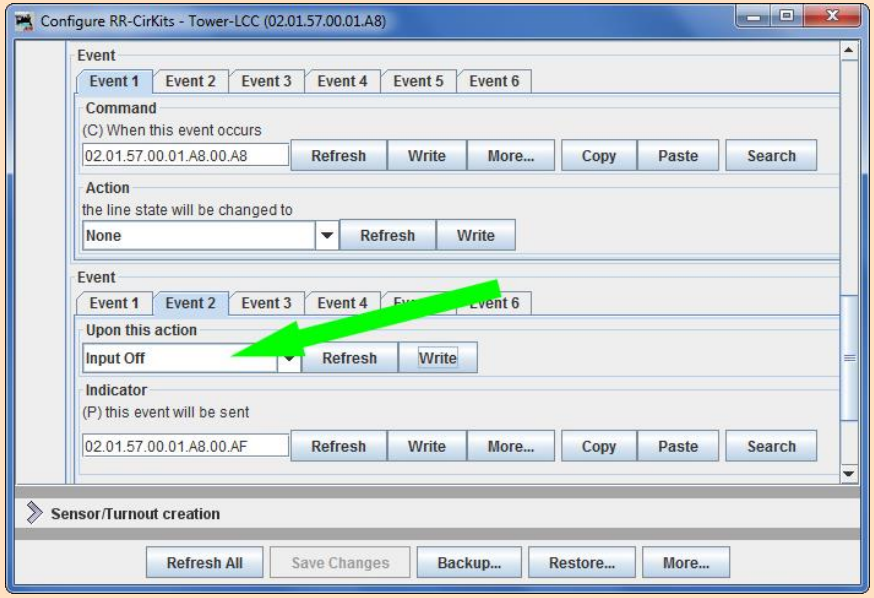
To link push button 2 to the “Off” position of the tortoise, we copy and paste the Producer event 1 EventID from push button 2 to the Consumer event 2 of the line controlling the tortoise.



Configuring Nodes

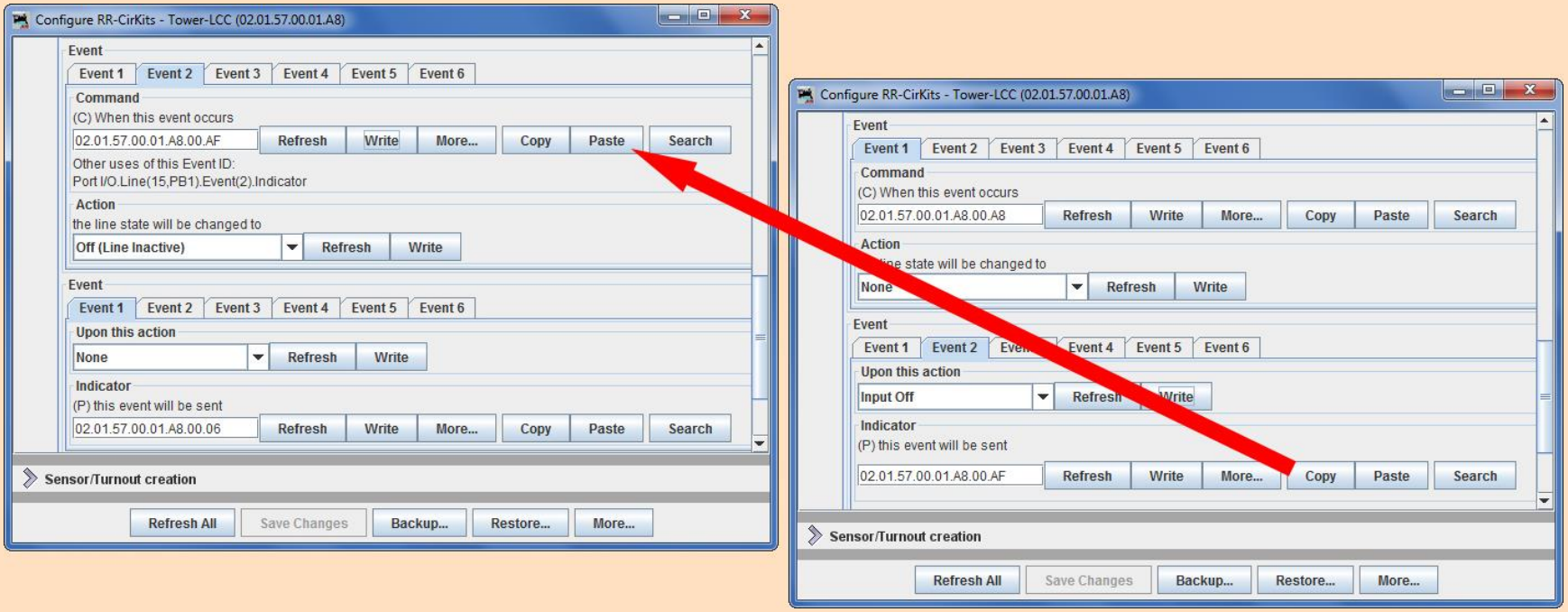
We can use one push button instead of two if we use an “Alternating” input. We just need to make use of a second event for the “off” state of the line.





Configuring Nodes

Just copy and paste event 2 EventID for push button 1 into event 2 of the line controlling the tortoise to replace PB2.



Questions?