MER 2022
Carolina Special

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/MER-2022-LCC-A.pdf
Part 2 (Applications and the future)
www.rr-cirkits.com/clinics/MER-2022-LCC-B.pdf

What is LCC?

LCC is an information highway for your model railroad layout

What is LCC?

- **LCC is a common language for layout elements to talk to each other**

- Signals
- Turnouts
- Detectors
- Lights
- Panels
- PCs / Smart Phones
- Clocks
- Boosters
- Command Stations
- Throttles
- Power Managers
- Trains
- etc…

What is LCC *NOT*?

LCC does NOT replace DCC.

On the track – DCC

Beside the track – LCC

LCC is not dependent on DCC.

It can run on DC or Märklin layouts, and is not locked to any DCC manufacturer.

Who/what is LCC for?

- A couple of examples:

Who/what is LCC for?

- One of the largest model RR layouts in the USA is the 5,000 square foot Pasadena Model Railroad Museum in Los Angeles, CA. They are in the process of upgrading their control systems to LCC.

Layout Command Control

Who/what is LCC for?

- Obviously my home layout isn't this large. (probably not yours either) Why do I need a system this capable just for a little bit of animation.

- OK, how about this layout, also in HO scale:

Who/what is LCC for?

- This little (8" diameter) layout is a scale model of one particular control point on the Missouri Pacific's Pueblo line. (MP 4566)

- It needs to follow an approximate schedule of passing (ghostly) trains and display the appropriate aspects. Sufficient capabilities are available in an LCC node, including timers, to do this little animation without the need of writing an Arduino program or connecting a computer to do the job.

Why LCC?

- I have heard some say that LCC is a solution looking for a problem, because we already have many ways to control our layouts.

- That is true, but that is part of the problem. We have LocoNet, CMRI, XpressNet, MERG, plus other proprietary methods to connect our layout devices.

- Some folks just don't plan to ever connect anything together. Little islands of control seem OK.

Why not DCC?

- Many of us simply use the DCC itself to control devices. That has two problems.

  - First it is a one way street. Have you ever seen a DCC connected detector? (Railcom .. maybe)

  - Second, DCC is limited in bandwidth, and accessory control messages are in competition with the repetitive locomotive control information. For a small layout with few accessories this is not usually any issue. For larger layouts using signaling it can be a serious issue. Also see above, for signaling you still need some way to read your detectors.
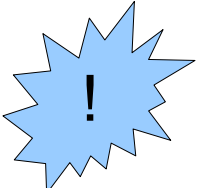
Other Options

- What about LocoNet, CMRI, XpressNet, MERG, plus the many other proprietary methods to connect our devices.

- Other than CMRI which predates DCC, all these other solutions originated due to the difficulty in using the DCC bus for any input information.

Other Options - CMRI

- CMRI was actually the first system to allow for two way communication with the layout. In my opinion its primary drawback is its computer based Master/Slave nature.

- No CMRI node can communicate to any other.

- A CMRI system always requires a (single) master computer to control everything. If that single computer fails, or is turned off, everything fails.

Other Options - LocoNet

- The LocoNet was the first Peer-Peer model railroad network. That does mean that any device may talk to any other device without any master unit or computer being in charge.

- Unfortunately the LocoNet is proprietary and requires licensing for commercial use.

- The LocoNet operates only slightly faster than DCC itself.

- The LocoNet does not overload gracefully. This can cause data loss on busy systems. !

Other Options – Etc.

- The XpressNet is a Master/Slave network with the same limitations as CMRI, and has little US presence or support.

- The UK based MERG group have many excellent designs, including a CAN bus, but they require that the users build their own hardware, and an annual membership fee to access many of the designs. Their addressing is not globally unique, so you must still assign and keep track of node address usage.

A solution is proposed

- The NMRA decided a number of years ago to sponsor an open (license free) method to interface to your layout. The intent is that, like the NMRA DCC standards, many manufacturers will be able to build layout accessory products that will interchange as freely as is now true for DCC mobile decoders.

A solution is proposed

- 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.

- The Open LCB group was chosen to develop this.

- The result was a set of protocols that can be sent over any media. For example, EtherNet, Wi-Fi, CAN (Control Area Network), and others. (some suggest tin cans and string, but don't believe it)

- The NMRA calls the Open LCB standards "LCC". (Layout Command and Control) LCC is NOT a replacement for DCC. (unless you consider replacing DCC accessory decoders)

- LCC can run along side of DCC, AC, DC, DCS, TMCC, RailPro, Battery power, etc. It is not a way to power your trains, it is a way to control your layout.

Why CAN?

- It is important to remember that LCC can be transported over many network technologies.

- When we (RR-CirKits, Inc.) decided to build LCC devices we had to make a choice of which transport to use. Wired Ethernet was one option, but designing a peer-peer network for Ethernet was way above our pay grade. The other problem is that it would require multi port Ethernet Switches with direct cable connections to each device. This would have required more wiring, not less, than current options.

- Wireless has issues with many nodes, and putting a radio in every node seems like a complex and costly solution as well, and would still require power wires. We also didn't want to get on a first name basis with our local FCC.

Why CAN?

- CAN was initially developed as a solution for automotive networking. This means that it is an industry standard, noise tolerant, and designed for the 12-24V world. CAN can be operated over a wide speed range, with a linear trade off between bus speed and bus length.

- The OpenLCB engineers picked a 125Kb rate and 1000' length as a good compromise for model railroad use. This is an order of magnitude faster than DCC.

- Also, unlike the other popular Peer-Peer system, the CAN bus will operate continuously at 100% speed, with error free collision detection and resolution.

Why not CAN?

- Disadvantages: The relatively high CAN bus speed does not allow for free form network designs. A CAN network segment requires a linear bus with a fixed termination at each end.

- Due to timing and other electrical limitations a single CAN segment is limited to 40 or fewer physical nodes. There are fairly simple ways to expand a CAN network into multiple segments, so this is not a serious concern.

- With today's hardware the system can not be expanded to over 1,000' length, end to end.

Why CAN?

- CAN has several different cabling and connector standards. Some of these use large and costly connectors. Industrial CAN uses the same DB-9 connectors used by RS-232 serial cables. These are still relatively large and no longer so very easy to locate, especially in longer lengths.

- Another CAN connector option uses the same RJ45 connectors and cables as wired Ethernet does. The OpenLCB engineers opted to use these RJ45 connectors because of the relatively low cost and their common availability world wide. The 4 pairs of a standard Ethernet cable additionally allow for optional power and other signals in addition to the CAN data pair itself.

LCC Basic Concepts

- How many here read the August 2017 NMRA Magazine article on Events in LCC?

- How many understood it?

- I will repeat its bottom line again here. EventIDs are simply magic numbers that represent your information on the bus, or over the air. There is no reason that you should ever need to enter one manually. There is little if any reason that you would ever need to know the details of what they mean. (which isn't a whole lot anyway)

LCC Basic Concepts

- Its the Event ma'am, just the Event.

- In previous control systems using a bus and events, (e.g. LocoNet and in a lesser sense CMRI) the events or messages sent on the bus have two parts, first an identifier number (address), and second the message type. This follows the original RR signal code line concept where each event was a station number plus one or more commands. For example: *control point #23 turnout set normal*.

LCC Basic Concepts

- This is:

  1. a Control Point #23 command
  2. for the turnout position
  3. to be set to normal

- A matching command with a predefined one bit different would mean *Control Point #23, turnout, set to reverse.* Another one bit change would create *Control Point #24, turnout, set to normal* etc.

- The size of the command space and the protocol design limits the number of possible selections to a predefined set. e.g. CTC 35 control points, DCC 2048 turnouts, LCC 72,057,594,037,927,935 EventIDs, etc.)

LCC Basic Concepts

- For example model turnouts normally only have two options, normal and reverse. That requires two messages.

- If you have a prototype turnout, then the system also has feedback information to confirm its movement. That requires another two messages.

- There is really no reason to distinguish one type of message from another, and the OpenLCB protocol does not do so. This means that an output message from a block detector can be an input message for an indicator.

LCC Basic Concepts

- In the LCC world an event has no predefined meanings. None, Keiner, Nada! An LCC event simply says; 'something has happened', or 'something should happen.' How it is defined is 100% up to you, the user. In our previous example it could still mean *Control point #23 set turnout normal.* However with LCC 'control point #23' is just what you call it on your layout, not that it was pin 23 on some brand of hardware controller. *Set turnout normal* just means that the event moves the turnout to normal. Undoubtedly you will want another event to move the turnout back, however that will be a completely different event with a different meaning. (e.g. *Control point #23 set turnout reverse)*

LCC Basic Concepts

- Maybe you want all turnouts to move to normal when you first start up. With our conventional control bus you need some way to send the proper commands to each turnout. (sometimes called Routes) With the LCC system you could simply define a new Event that says '*set all turnouts normal*' and then configure each turnout to also respond to that command (by moving in the appropriate direction)

LCC Terminology

- **Producer - Consumer** You will probably hear LCC folks throwing around terms like Producer and Consumer. They aren't talking about a big business takeover. They are just trying to sound educated. <G>

  - *Producer* simply means that some device can create (produce) an Event. Some examples might be a push button or block detector.

  - *Consumer* just means that some device can respond to (consume) an Event. It could be a lamp, a turnout driver, or anything else that you can control.

  - *Events* can have from 1 to many *Producers*. Events can have from 0 to many *Consumers*.

  http://openlcb.org/trunk/documents/notes/ProducerConsumerModel.html

LCC Basic Concepts

To elaborate a little bit. For an event to happen something must have created it. Therefor there has to 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. Thus the statement that every Event has one or more *producers*.

Producer

Pushed

Event

Turn On
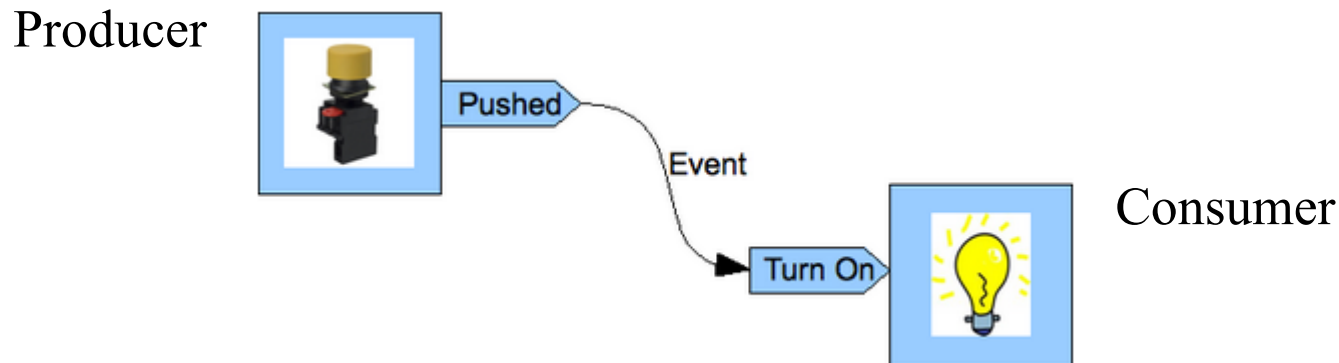
Consumer

LCC Basic Concepts

For *consumers* the picture is a bit different. There is nothing in the specification that says that any device has to respond to an Event. You may have built a panel for a passing siding that doesn't yet exist. If you press its turnout control button an Event message gets sent out. (*producer*) However there is nothing on the layout to respond. (*consumer*) Later you might add a turnout controller and a computer based CTC machine and have several *consumers* for that Event. Thus the statement that every Event has zero or more *consumers*.

Configuration of LCC nodes

- One of the key new concepts in the LCC protocol is that, not only the configuration, but the 'decoder file' (in JMRI terms) itself should reside in the LCC node. This is an important change from the status quo.

- Originally hardware had a fixed purpose. Each required its own dedicated connections. Lionel crossing gates flashed with direct wiring connected to contacts triggered by the passing wheels. (blink-blink….blink-blink….)

Configuration of LCC nodes

- Then some devices were connected to a bus. (or track for train control) This required assigning addresses or channels. The usual solution for addressing was to include a set of jumpers or switches for the selection. In some cases it was a plug with different component values.

- As electronics improved the selection of addresses was moved into the device code itself. An example that we are all familiar with is modern DCC mobile decoders.

Configuration of LCC nodes

- One of downsides of this new method is that our decoders now each need to be configured with a specific (non default) address. That itself was automated by some manufacturers, but it soon became evident that something more was needed than simple interactions through a hand held throttle. Some new decoders currently have 1000 or more values to configure. (called CVs or Configuration Values)

Configuration of LCC nodes

- JMRI and other programs have come to the rescue, but the decoders are now so complex that a 'decoder file' is required for each locomotive and is stored on a computer to help keep track of these changes. The DCC specification does not include an easy way to read information from a decoder except very laboriously and slowly over a special connection. (called a programming track)

Configuration of LCC nodes

- This manual address assignment was deemed to be too slow and inflexible for the new LCC equipment.

- Two key changes were required. The first was that any LCC node could be configured in place on the layout at any time with no need to access it for jumper changes or button presses.

- The second was that any information required to configure a node should reside in the node itself, and be available to any configuration tool connected to the network. Now any node could be configured in one place and moved to another with all the information moving with the node itself. This means not only configuration values but any user names and comments as well.

Configuration of LCC nodes

- Another key design choice of LCC was that the manufacturer would assign a default node ID during manufacturing, in a manner that prevents any duplication of addresses…. Ever, …. anywhere! (similar to Ethernet's MAC addresses)

- This manufacturer based address assignment has another unforeseen benefit. Any automatic or user linking of two LCC nodes no longer needs to know anything at all about the rest of the layout in order to prevent unintended conflicts.

- Adding a new node to the layout will never conflict with any already installed devices.

LCC Configuration Tools

- Because all the configuration information as well as the values, user names, and coments reside permanently in the nodes themselves, it is easy to use different configuration tools interchangeably. There is no need to synchronize them externally or move files around from computer to computer.

- One of the latest tools available (Sept '22) for configuration of LCC nodes is an i-Phone app. Of course it is only useful on an LCC network with WiFi. (e.g. the TCS CS-105)

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
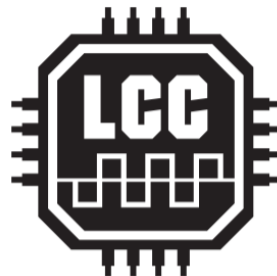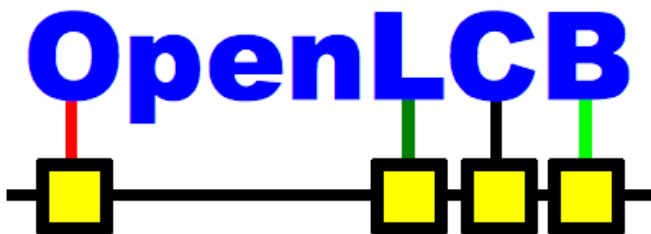   In October 2022 we had over 440 members

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

Questions

- ?