Use Cases

The table below describes the Use Cases that were and are being considered for the S-9.6/OpenLCB NMRAnet design. NMRAnet refers to the S-9.6/OpenLCB proposal.

Entry Level User	A model railroader wants to learn about NMRAnet. He buys two inexpensive boards at his local hobby shop. He connects them on his layout to two pushbuttons and a turnout motor, does a simple configuration, and is able to control the turnout with the pushbutton.
	Alice wants to control her three track station. She has goes to her local store and buys one 8-button, four turnout-controller nodes, and a power supply. When she gets home, she unpacks the nodes and connects each of the turnout-controllers to her four turnouts, E1, E2, W1, and W2. Then she connects the nodes together with the cables. Following the directions, she then uses the blue button on each of the turnout-controllers and ensures the attached turnout changes position. Now she is ready to make the buttons control the turnouts. The buttons are labelled A-H. To connect A to turnout E1, she pushes the blue button on the E1 controller until the turnout is in the position she wants, and then the gold button – this tells the controller to learn this position. She then goes back to the button node and pushes the gold button, button A, and then the gold button again. This teaches the turnout controller to respond to button A. She then sets E1 to the other position and trains it with button B. (In more detail, on E1 she pushed blue repeatedly to position the turnout, then pressed gold, then pressed gold, B, gold on the button node). She repeats this with turnouts E2, W1, and W2 and C-H. However, she found that she still forgot to compatibly position the turnouts at both ends of the station occasionally. However, he realized that she could improve her scheme. She reset the nodes to factory settings as directed and then taught button A to make all the turnouts select track 1. (In detail: on E1 set diverge and gold, on E2 set diverge and gold, on W1 set diverge and gold, on W2 set diverge and gold, and finally on the button node pushed gold, A, and gold). She also made buttons B and C select tracks 2 and 3. Now she could control her station tracks with only three buttons, and there was never a chance that they were misaligned! Later, she changed it again so that A-C selected tracks 1-2 for entrance, and D-F selected them for exit.
Expanding a Small Layout	A user has four NMRAnet boards working on his layout. He buys one more input board to provide extra pushbuttons to operate his yard ladder from the other end of the layout. He's able to configure the new board and put it into operation.
	Alice decides that it would be nice to be able to control her station tracks from another location. She buys another button node (lets label these M-T), and hooks it to the other nodes. She then copies the learning from the first button node to this new one. (In detail: on the original set of buttons she pushed A and gold, and then on the new buttons she pushed gold, M and gold, and repeated the same process with the other buttons).

Mid-size Layout Problem	A user has twelve NMRAnet boards operating. Something changes, and the system is no longer reliable. Certain operations no longer work, and others only work sometimes. The user attempts some debugging steps using features of the existing hardware, but is unable to locate the problem, so buys/borrows diagnostic device(s).
	Mike is at a loss, although he kept good records things have gone south on his layout. He calls up his friend and borrows his laptop with the NMRAnet configuration software and a NMRAnet-USB adapter. He plugs them into his layout LCB and turns it on. The program asks him to select the layout name or choose new. He chooses new. The program lists his twelve boards. He can drill down to each part of each node and see what other nodes receive its events. He can also see whose events it receives. There is also a diagram of the nodes. He pushes some buttons and watches as the program animates the events travelling out of one node and into the receiving nodes. He finds his mistake, and makes changes to his configuration and tests again. Problem solved – he vows to buy a laptop.
Large Layout Upgrade	A model railroad has 40 NMRAnet boards controlling a large layout. To upgrade a major yard, the user wants to replace 12 boards with new ones that offer new (non-NMRAnet) features. He captures the existing setup, replaces old boards with new ones, configures the equipment and is back in operation.
	Joe has been assigned the non-trivial task of upgrading the yard with the new boards the planning committee bought. He tells the NMRAnet configuration program to make a fixed record of the configuration of the twelve boards he is replacing, and saves it to disk, twice, he's paranoid. He then proceeds to disconnect the original twelve boards and replace them with the new ones. Sometime later ... he brings up the configuration program, and its informs him there are twelve new boards on the layout. “Tell me something I don't know” he mutters. He brings up another copy and retrieves his stored configuration. He notes that some of the names of things have changed, but he can manage. He uses the old configuration to find nodes from which he can teach the old events from – this is easy, on the screen he just grabs the node's consumer or producer entry and drags it to one or more of the new nodes consumers and/or producers. The nodes are linked and using the old events. “Sweet” he thinks. Now he sets about to use all those new capabilities that these new nodes have – “Wow, it was worth it, these babies are hot”.
Distant Control Panel	A model railroad has turnouts and signals controlled via NMRAnet. The owner wants to put a physical control panel in his house, a separate building, and operate it via NMRAnet.
	Mike has been operating his layout from afar using a 'glass' control panel on his computer connected to his layout via his Wifi router. However, it doesn't have the same satisfaction as having a real control panel. He decides to build one, and he installs a number of CAN NMRAnet nodes to interface it. The big day arrives and now he needs to configure it. He connects a CAN-to-Ethernet bridge and plugs it into his networks. He boots up his configuration program, links to the bridge and sets it up to talk to his layout. He then links to is layout, the program announces there are lots of new nodes. Using drag-and-drop he links his new nodes into the existing layout. This is relatively easy, as he is just duplicating his old controls. In fact he can just drag one whole old node onto a new one and all its events are duplicated. The whole process only took 5 minutes. Mike thinks “Now, let's get down to some serious operating”.

Large Layout Expands	A large model using CAN, but not any attached computers, grows enough that it needs to have another CAN segment. This must be possible without reconfiguring all the existing nodes, event IDs, etc.
	Hugh realizes he has a problem – the NMRAnet-CAN bus running his layout is not big enough, he's maxed out the number of nodes. He talks to the local guru, and the guru says – no problem, just cut your CAN into two parts and join the parts with a CAN-to-CAN bridge. Hugh is doubtful, it can't be that simple, won't he need to change node ids and change his event programming? The guru reassures him – it is that simple – no need to change anything, the nodes will still see alll the other nodes events, jyst as before. Hugh take shim at his word, picks up a bridge and, after five minutes deciding where to break the CAN bus, inserts the bridge and is running trains again.
Connect Multiple Programs	The user wants to run multiple programs from multiple vendors in his home computer that simultaneously connect to the layout NMRAnet and control/monitor it.
	Mike visits Hugh and brings some new NMRAnet software on his laptop. “This software will let us make a mock-up of your layout showing the location and connections between your nodes. I just need to plug it into your layout”.
Remote Dispatcher	A model railroad has turnouts and signals controlled via NMRAnet. NMRAnet is used to install a CTC panel at a distant location. The panel could be either a physical panel or on a computer screen.
	See Distant Control Panel.
Modular Layouts	A modular club has fifty modules, each of which as a CAN NMRAnet with two or three nodes controlling the module. These modules are separately built, with no central administration. They are brought to a central location for a meet, where they are all connected together in some pre-planned orientation. The NMRAnets are connected in some fashion, and used to operate the entire layout from both central and distributed locations.
	John is in charge of the meet. He tells the attendees to connect the modules together into multiple CAN segments, observing the limitations of length and drive capabilities of the nodes. He assigns segment leaders and instructs them to configure and debug each segment in isolation by connecting a computer with configuration software. XML files describing the modules that some attendees brought help in this, but the nodes themselves contain enough information to allow them to be connected and integrate their existing block detection and signalling systems. If the number of modules was smaller, then John would have connected the segments together using CAN-to-CAN relay nodes, but he had decided to connect the segments together using an Ethernet backbone, using Ethernet-to-CAN bridges. This will reduce the bandwidth requirements on each of the segments because the bridges will automatically only pass events between the segments that are actually required, rather than all packets. Together with his segment leaders, they quickly integrate the segments onto the backbone. Attendees can observe and control the layout from disparate points on teh layout.

Remote Diagnostics	A club layout is operated via NMRAnet. There's something not quite right about the signalling, and one of the members wants to check the operation of the signals. He makes a remote connection to the layout and checks the status and configuration of the hardware from his home computer.
	Bob receives a call from his friend Hugh at the clubhouse that there's some trouble on the layout. He doesn't have time to drive across town and back, but fortunately, his club has an Ethernet connection to the layout, and even better they have it attached to the local area network (LAN) and to the wider internet through their router – he can just log in from home! He opens his copy of the configuration program on his home computer and connects to teh layout. It appears that one of the signals is not responding, and this has lead to a cascade of failures downstream from the signal. He phones Hugh back and tells him to check the connections on that signal. Hugh crawls under the layout and discovers a plug has pulled out – it must have been that work party mid-week. Everything seems back to normal. Hugh thanks John. John replies “Its all in a day's work for a NMRAnet repairman”.
Aggregation of Modular Clubs	Dozens of clubs put together dozens of NMRAnet segments and hundreds of modules that have been separately configured. A large FREMO meet would be an example. Collision avoidance: EventIds, NodeIds that are already configured into the modules should already be unique. It must be possible to build automated tools for translating, disambiguating Event IDs. Conventions and/or tools for connecting across the boundaries are needed so that e.g. signaling systems can work with adjacent modules they've never met before. It must be possible to build automated tools for health monitoring across the layout.
	See Modular Layouts. In this case, thanks to prior planning, each club has provided an XML file describing the events that must be connected to allow the integration of the signalling and other systems. Hugh and John use the configuration program to build a schematic diagram of the entire layout. They then drill down to each interface between layout segments and connect the edge nodes of each segment together. They quite quickly repeat the exercise on each adjacent pair of segments, with the help of the leads of the pairs of segments. Since every node has a unique ID and unique Events, they are not worried about conflicts or having to change ranges of events on any modules – reprogramming hundreds of nodes would not be inviting.