AN INTRODUCTION TO JMRI® AND CATS

A Caboose Class Presented by Rodney Black on December 22, 2018
Class Goals

- Introduce how to connect JMRI to your layout
- Introduce how CATS can enhance your operations
Outline

- Layout Control Basics – Identifying the Pieces
- Installing and Configuring JMRI
- JMRI Capabilities
- Focus on Using JMRI for Layout Control
- CATS Capabilities
- CATS Demonstration
- References
Connecting to JMRI - Step 1

"Eye of newt, and toe of frog, Wool of bat, and tongue of dog, Adder's fork, and blind-worm's sting, Lizard's leg, and owlet's wing…" William Shakespeare

Collect the components

What components do you want? What is your operating scheme?
Examples of Operating Schemes

Shout and Go

Mother, May I?

TT&TO

TWC/DTC

CTC
Layout Control Basics

Proprietary Protocol

NMRA RP 9.X

Mobile Decoder

Command Station

NMRA

Copyright © 2018 Rodney Black
Layout Control Basics

Mobile Decoder

NMRA

Command Station

Proprietary Protocol

NMRA RP 9.X
Mobile Decoder

NMRA RP 9.X

Proprietary Protocol

Internet

Command Station

Layout Control Basics
JMRI Requirements

Application

Java
Virtual Machine

System Software

Device Drivers

Hardware

Layout

Platform
Layout Connections

- Atlas
- Bachrus
- CMRI
- CTI Acela
- CVP
- DCC++
- Dcc4Pc
- DCC Specialties
- Digi
- Digikeis
- Digitrax
- ESU
- Fleischmann
- Hornby
- Lenz
- Lionel TMCC
- Maple Systems
- Märklin
- MERG CBUS
- Modbus
- MRC
- NAC
- NCE
- Oak Tree
- Open LCB
- ProTrak
- QSI
- Raildriver
- Raspberry Pi
- Roco
- SPROG
- SRCP
- Tams
- Uhlenbrock
- Viessmann
- Wangrow
- X10
- Ximo
- ZTC Controls
JMRI is Downloadable from the Internet

Go to http://jmri.org/download/index.shtml
Connecting to JMRI - Step 2

- How to attach my computer to the layout?
- It depends on the DCC system
- Read carefully! Some systems cannot be connected to a computer.
Example: Installing NCE

Hardware Support: NCE

Supported Hardware

Command Station

JMRI supports three NCE layout connections:
- Power Pro (Powerhouse) via serial cable
- Power Pro (Powerhouse) via network connection
- NCE USB Interface via a USB cable

They are shown in the JMRI Preferences Connection pane as NCE, NCE via network, and NCE USB Interface, respectively.

Limitations

A very small number of early (pre-2002) NCE Power Pro (Powerhouse) command stations have an "optional analog control circuit". If you don't recall special ordering this, you probably don't need to read the rest of this paragraph! In these, pins 5, 6, and 9 are used with an analog voltage as a separate throttle. If you have one of these command stations, a normal 9-pin serial cable will not work! You'll need to make a custom cable that just connects the "send data", "receive data" and "ground" pins. See your NCE manual for details.

Connecting

Settings

ASU/ (Input) Numbering

The NCE "ASU" device allows the NCE command station to check the status of inputs from the layout. These are referenced in JMRI as "Sensors" with names like "NCE Sensor", "No" means "NCE Sensor", and 195 is the number of a specific sensor input.

The sensor number can be found by using the following formula:

Sensor ID = (ASU Cab Address - 1) \times 16 + (Pin Number On ASU) - 1

For example: The ASU cab address is set to 50 and one of the BG20s is connected to pin 2. Therefore the Sensor ID would be...
JMRI is ...

- Open source software
- A “Swiss Army Knife™” of tools for your model railroad
- A set of Lego® building blocks for running your model railroad
JMRI is a Name

“What is in a name?” William Shakespeare

- Officially, “JMRI®” is just a bunch of letters
- Unofficially, Java Model Railroad Interface
- DecoderPro®
- PanelPro™
- DispatcherPro™
- OperationsPro™
- SoundPro™
- Logix™
JMRI is Open Source Software

“It was the best of times. It was the worst of times.” Charles Dickens

■ Licensed under the Free Software Foundation’s “GNU Public License (GPL) version 2 with the classpath exception”

■ The source code is readily available.

■ “Free to modify” – technically, not “free” as in “free beer”.

■ Cannot be incorporated into other software.

■ Landmark legal case for open source software (Jacobson Vs Katzer).
JMRI Support

- Software is “as is”. No warranty. No promises of bug fixes or improvements.
- Like public television – “community supported” – financially and technically.
- No helpline, but active volunteer “grass roots” support

- Developer Group
  - 15 to 30 actively working on code at any time
  - 25 to 50 regular contributors and supporters

- User Group
  - Started around October 2002, 189 email addresses
  - July 2004 grown to 1304 addresses
  - Jan 2007 passed 2500 with 2752 addresses
  - Jan 2018 had 9569 addresses

- Extensive context sensitive help
JMRI is a “Swiss Army Knife”

“One ring to rule them all” J. R. R. Tolkien

XtrkCad

PanelEditor
ControlPanelEditor
LayoutEditor
DispatcherPro
SwitchBoardEditor

DecoderPro
PanelPro

SoundPro
OperationsPro

WebServer
Remote Connection

Jython Scripts

Logix
Sensors
Signal Head
Signal Mast
Throttles
Turnouts

Plugins

Manifest Creator
CATS

All or nothing—no à la carte
JMRI Profiles – First Step in Customization

• A profile is a connection configuration

• One profile can share parts with other profiles

• One for layout, one for ‘offline’

• Works with Dropbox and other file sharing

• Can be skipped by picking a default

Slide prepared by Ken Cameron
Profile Options - Preferences

[Image of the Preferences window with options for LocoNet, System manufacturer, System connection, and Settings.]

- System manufacturer dropdown with options for Digitrax, ESU, Easy DCC, Hornby, JMRI (Network), Lenz, Lionel TMCC, and others.
- System connection dropdown with options for LocoNet, JMRI (Network), DCC, and others.
- Connection Prefix field with the value set to "L".
- Connection Name field set to "LocoNet".
- Additional Connection Settings checkbox unchecked.

[Image of a red circle highlighting the Connection Prefix field with the value "L".]
Configuring JMRI Start Up

![Preferences window with Start Up options](image)

- Connections
- Defaults
- File Locations
- Start Up
- Display
- Messages
- Roster
- Throttle
- Winthrottle
- Config Profiles
- Web Server
- LocoNet over TCP Server
- JSON Server
- Railroad Name
- SRCP Server
- Simple Server
- Warrants

Add items to be executed or opened when the application starts. Items will be executed or opened in the order listed.

- Open Block Table
- Open Consist Tool
- Open DCC Decoder Tool
- Open DecoderPro Operations Mode
- Open DecoderPro Service Mode
- Open Dispatcher
- Open Fast Clock Configuration
- Open ID Tag Table

Set the connection for this action.

Scripts should be run last in most cases.

Add  Move  Up  Down  Remove
JMRI “Pros”

- DecoderPro – programming mobile decoders
- SoundPro – controlling sounds
- OperationsPro – car routing/switch lists
- PanelPro
  - Panel Editor/Control Panel Editor
  - Layout Editor
  - DispatcherPro
  - Switchboard Editor

- Building blocks for DIY control
  - Logix (built-in JMRI programming language)
  - Jython scripts (external programming language)
  - JMRI plugins (Java code extensions)
  - JMRI applications
The Rule of Twos

“Always two there are. No more. No less. A master and an apprentice” Yoda

- **Issues in creating a dispatcher panel**
  - Appearance (the presentation of the layout) – Control Panel Editor
  - Functionality (making the panel do something) – Layout Editor

- **Issues in making it work**
  - Connections between elements
  - The logic (rules) controlling signal indications

- **JMRI logic creators**
  - Simple Signal Logic (SSL)
  - Signal Mast Logic (SML)
## JMRI Layout Control Editors

<table>
<thead>
<tr>
<th>Type</th>
<th>Images</th>
<th>Unique Features</th>
<th>Track styles</th>
<th>Signal Control</th>
<th>Train Automation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Layout Editor</strong></td>
<td>Vector based track images and raster based background images</td>
<td>Captures the full connectivity of your layout as you draw it. Only single instances of each vector Item are allowed. Predetermined look and feel.</td>
<td>Vector based items may show both animation and occupancy.</td>
<td>SSL Logix Masts</td>
<td>Uses <strong>Dispatcher</strong> and <strong>Transits</strong>.</td>
</tr>
<tr>
<td><strong>Panel Editor</strong></td>
<td>Raster Images used for both track and background images</td>
<td>Multiple instances of any Item are allowed. Uses both supplied and custom images as required for panel visual fidelity.</td>
<td>Raster Images used to show animation. Occupancy info possible, but difficult, to show using track colors.</td>
<td>SSL Logix Masts</td>
<td>None other than by <strong>Scripting</strong>.</td>
</tr>
<tr>
<td><strong>Control Panel Editor</strong></td>
<td>Raster Images used for both track and background images</td>
<td>Multiple instances of any Item allowed. Uses both supplied and custom images as required for panel visual fidelity.</td>
<td>Raster Images to show animation. Occupancy and automated train status info shown.</td>
<td>SSL Logix Masts</td>
<td>Uses <strong>Warrants</strong>, <strong>OBlocks</strong>, and <strong>Portals</strong>. Use of <strong>Masts</strong> allows automated trains to operate at signal indicated speeds. Train recording supported.</td>
</tr>
</tbody>
</table>

Chart prepared by Dick Bronson
Control Panel Editor/Panel Editor

■ Control Panel Editor is designed to create a pictorial CTC machine (Control Panel) - **LOOKS**

■ Uses icons and images for either either a classic or modern look

■ Images do not connect

■ You need to define the connections between images and what they control

■ Does not force you into a way of doing things

■ Panel Editor was the original JMRI panel editor and has been superseded
Control Panel Editor Example
Layout Editor

- A vector view of a layout. I.e. the toe bone connects to the foot bone, the foot bone, connects to the ankle bone, .... - FUNCTION
- Designed to document and edit the connectedness of the layout
- Allows for one definition of each chunk of the layout. (vector) As soon as you have two turnouts that operate as one (e.g. crossover) or two detectors that operate as one you need to fight the system to figure out how to defeat its basic nature.
- It was never designed to make a CTC panel drawing. Doing so is sort of like trying to use a CAD program to paint a landscape.

Source: jmri.org
Layout Editor Example
DispatcherPro

- “provides functions and organizes information relating to dispatching trains on a model railroad layout. Dispatcher is envisioned to work alongside a Panel, constructed in either Layout Editor or Panel Editor, which provides visual feedback of layout status.”
- Support for train start up and termination
- Information to allow easy set up of meets at passing sidings
- Automation of some dispatcher functions
- Set up of automatic running of trains (schedules)
- Support for starting trains according to a Fast Clock
- Support for linking signals to allocation via simple APB signaling support
- Optional automatic setting of turnouts when a section of track is allocated
- Fully automatic running of trains, along with manually run trains
DispatcherPro Example
Switchboard Editor is a graphic, straightforward control interface, showing a range of switch keys to watch and control items.
OperationsPro

- Car routing
- Generates switchlists and manifests (an Excel program adds custom formatting)
- Uses engine roster (from DecoderPro)
- Adds rolling stock roster
- A car is bound to a train from the time the train is “built” until it is terminated
- Supports “batch” mode (generate work reports ahead of time)
- Supports “real time” mode (cars can move multiple times in a session)
- Allows blocks of cars
- Allows Loads and empties as well as custom loads
- Allows schedules
- Layout connection is optional
Networking

- Some web pages (primarily PanelPro and DecoderPro) and throttles can be displayed on a remote computer
- Some layout connections can be distributed across a network
- WiFi throttles (free and enhanced pay for versions)
  - *iPhone WiThrottle*
  - *Android Engine Driver*
Application Programming Interface (API)

An API exposes internal code for controlled uses by external code.

From the JMRI web site:

“The long-term strategy is to:

■ Create a set of publicly available common interfaces for model railroad control code, suitable for use with any type of layout, from any type of computer.

■ Make these useful by providing quality implementations for as many types of layouts as possible.

■ Encourage interoperability of layout hardware and software through the use of common interfaces.”

Source: jmri.org
JMRI Internal Items Exposed

- Aspects for signals
- Blocks (track segments)
- Conditionals (for logiX, although these are not entirely named beans)
- signal mast Groups
- signal Heads
- signal masts
- Lights (a type of output)
- eNtry/exit destiNation points
- Memories
- Outputs
- Power districts and subdistricts
- Reporters
- Routes
- input Sensors
- Turnouts
- logiX
- Sections
JMRI Names - Again

“To summon a thing that is not there at all, to call it by speaking its true name, that is a great mastery, not lightly used.” Ursula K. Le Guin

All devices controlled by JMRI must have a name:

- **System Name** – constructed from the System Prefix (connection), Type Letter, and connection bus identifier (e.g. address).
  - *For example, LS192 designates a Sensor (S) on the Loconet bus (L) with address 192*

- **User Name** – a name which makes sense to you in identifying the device on your layout
  - *For example, “Staging_1” for the first track in staging*

- **Why the complexity?** Many devices refer to other devices.
  - *For example, Simple Signal Logic needs to know the “name” of an occupancy detector that it is protecting. You can create it and test by defining an Internal Sensor (e.g. System Name IS26), then change the System Name to a real name when attaching to the layout.*
Default System Prefixes

A. CTI Acela, Bachrus Speedometer
B. Direct DCC control
C. C/MRI serial
D. SRCP, Anyma DX512, DCC++ (previously written DCCpp), CC4PC
E. EasyDCC
F. RFID tag readers
G. ProTrak Grapevine
H. Internal, e.g. objects with no associated hardware
I. JMRI network connections
J. Maple Systems
K. LocoNet
L. Model railroad layout control buses, including OpenLCB and MERG CBUS, Model Rectifier Corp (MRC), Marklin CS2
M. NCE (also Wangrow currently)
N. Oak Tree Systems
O. P Powerline transmission, e.g. X10 and Insteon, Raspberry Pi native pins
P. QSI programmer interface
Q. RPS system
R. SPROG
S. SPROG
T. Lionel TMCC, TAMS
U. ESU ECoS
V. TracTronics SECSI
W. (reserved for Wangrow, but that’s currently still combined with NCE)
X. XpressNet used by a number of Lenz, Atlas, Hornby and other connections
Y. ZimoMX-1, IEEE802.15.4 and Z21 connections
# Type Letters

<table>
<thead>
<tr>
<th>A.</th>
<th>Audio</th>
<th>L.</th>
<th>Light</th>
<th>W.</th>
<th>Unused</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.</td>
<td>Block</td>
<td>M.</td>
<td>Memory</td>
<td>X.</td>
<td>logiX</td>
</tr>
<tr>
<td>C.</td>
<td>Unused</td>
<td>N.</td>
<td>eNtry/exit</td>
<td>Y.</td>
<td>Sections</td>
</tr>
<tr>
<td>D.</td>
<td>iDentity</td>
<td>O.</td>
<td>Unused</td>
<td>Z.</td>
<td>Unused</td>
</tr>
<tr>
<td>E.</td>
<td>Unused</td>
<td>P.</td>
<td>Power manager</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F.</td>
<td>Signal Mast</td>
<td>Q.</td>
<td>Unused</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G.</td>
<td>Signal Group</td>
<td>R.</td>
<td>Reporter, Route</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H.</td>
<td>Signal Head</td>
<td>S.</td>
<td>Sensor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>I.</td>
<td>Unused</td>
<td>T.</td>
<td>Turnout</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J.</td>
<td>Unused</td>
<td>U.</td>
<td>Unused</td>
<td></td>
<td></td>
</tr>
<tr>
<td>K.</td>
<td>Unused</td>
<td>V.</td>
<td>Unused</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Select JMRI Types

- Sensor – Typically an input for status, feedback, etc., as from an electrical switch
- Turnout – a historic misnomer; an output to control an electrical switch
- Light – not a signal light, but control of a scenery element
- Signal Head (Arm) – one component of a signal; typically a color
- Signal Mast – a group of Signal Heads; typically identified by an indication (e.g. Clear, Advance Approach); can reference prototype rules
- Logix – a simple programming language consisting of “if” conditionals and “then” actions (e.g. if Block_1 is Active and Block_2 is not active then throw CT 111).
Example of LED addressing on a Digitrax SE8C

<table>
<thead>
<tr>
<th>SE8C Factory Default Settings for Beard ID 01</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SE8C Driver Socket</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>DRV&lt;sub&gt;1&lt;/sub&gt;</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>DRV&lt;sub&gt;2&lt;/sub&gt;</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>DRV&lt;sub&gt;3&lt;/sub&gt;</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>DRV&lt;sub&gt;4&lt;/sub&gt;</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>DRV&lt;sub&gt;5&lt;/sub&gt;</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>DRV&lt;sub&gt;6&lt;/sub&gt;</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<sup>a</sup> Fi Yell = Fiery Yellow
System Name Example

LT289

Type Letters

- A. Audio
- B. Block
- C. Unused
- D. Identity
- E. Unused
- F. Signal Mast
- G. Signal Group
- H. Signal Head
- I. Unused
- J. Unused
- K. Unused
- L. Light
- M. Memory
- N. Entry/exit
- O. Unused
- P. Power manager
- Q. Unused
- R. Reporter, Route
- S. Sensor
- T. Turnout
- U. Unused
- V. Unused
- W. Unused
- X. logiX
- Y. Sections
- Z. Unused
Example of Simple Signal Logic
Example of Signal Mast Logic (SML)

- Used in conjunction with Layout Editor and Logix to build complex signal rules
- JMRI can discover much of the contents automatically
Areas of Confusion

■ Polarity – its all one or the other
  – 1 or 0
  – Open or closed
  – High or low
  – Active or inactive
  – On or off

■ JMRI does not know how you wired things up
  – The stationary decoder wire may turn on/off a lunar rather than a red LED
  – Two Tortoises may be wired identically, yet flip one 180 degrees and normal becomes reverse

■ The moral – don’t try too hard to overthink things. Sometimes the best course of action is to wire it up and try it. You can reverse wires (or polarity) after trying it. “We will fix it in software” – Dan Delany
CATS – a reminiscence

“Memories are like mulligatawny soup in a cheap restaurant. It is best not to stir them.” P. G. Wodehouse

- The first record that I have of CATS successfully using JMRI is dated April 5, 2004.
- I wanted to learn some new programming skills and Pat Lana wanted a CTC panel for his CRANDIC layout.
- After developing a magnet board, I needed to figure out how to tie it to the railroad; thus, I had to figure out how to talk to Loconet.
- About that time, I read about JMRI and was thrilled to discover that it already could talk to Loconet and other layout systems.
- Knowing how much work was going to be involved in this, I decided to add a little more effort and make it available to a wider audience.
Things That I Did Not Like About Panel Editor

- It did not look like a prototypical CTC panel.
- It required a lot of behind the scenes work to connect the pieces together and to create the signaling logic.
- Editing was next to impossible.
- It was frustrating to work with.
- Model railroading is getting too complicated for the average model railroader. It should not require a degree in computer programming to create a CTC panel.
- PE required a layout connection.
- Not Invented Here.
How CATS is Different From PE

CATS did not do the things I disliked in PE – introduced its own problems

■ Presents a modern looking CTC panel – modeled after Digicon, with a healthy dose of Train Dispatcher 2

■ Discovers connections between elements automatically

■ Generates signal logic rules automatically – follows the tracks

■ Layout design is a separate program from layout running

■ Layout design does not need a connection to a layout

■ Follows a more conventional “document” model, rather than a “workbook” model of layout development
CATS Adds

- Built-in track coloring
- Built-in train tracking
  - By detection
  - By arrow keys
  - By mouse drag and drop
- Train management
- Crew management
  - Who
  - Job
  - Train
  - Time on job
  - Dead on law
- Jobs management

- Track and time
- Session logging, including replay
- Selectable ABS, APB, CTC, DTC signal safety rules
- Integration with OperationsPro
  - Real time updates of a train’s consist
  - Real time updates on train length and weight
- Separate train lineup display screen
  - Customizable as to information and format
  - Remote presentation
  - Remote crew management
The CATS Program Suite

- Designer – to draw the layout
- CATS – to run the layout
- Trainstat – to share train status
A Digicon Screen Shot
CATS Compared to the Prototype

Screenshot of the Digicon Prototype

CATS Rendering of the Prototype

Photo courtesy of Dick Johannes, HUB Division
A CATS Screen Shot
A Trainstat Screen Shot
Examples of CATS Usage

Minimal (magnet board)

Typical

Tricked out
CATS Demonstration

Lets Play!
References and Acknowledgements

- [http://www.jmri.org](http://www.jmri.org)
- [http://www.cats4dcc.org](http://www.cats4dcc.org)
- [http://www.hubdiv.org/signalComm.htm](http://www.hubdiv.org/signalComm.htm) Lot’s of good stuff (some videos) on using CATS and signaling a modular railroad
- Thanks to Bob Jacobson for keeping JMRI on track
- Thanks to Ken Cameron for material from his Power Point presentation
- Thanks to Dick Bronson for material from his Power Point presentations on layout control panel editors
The End
Layout Control Basics