NMRA 2019
Salt Lake City
Signaling with LCC - Details
(Layout Command & Control)

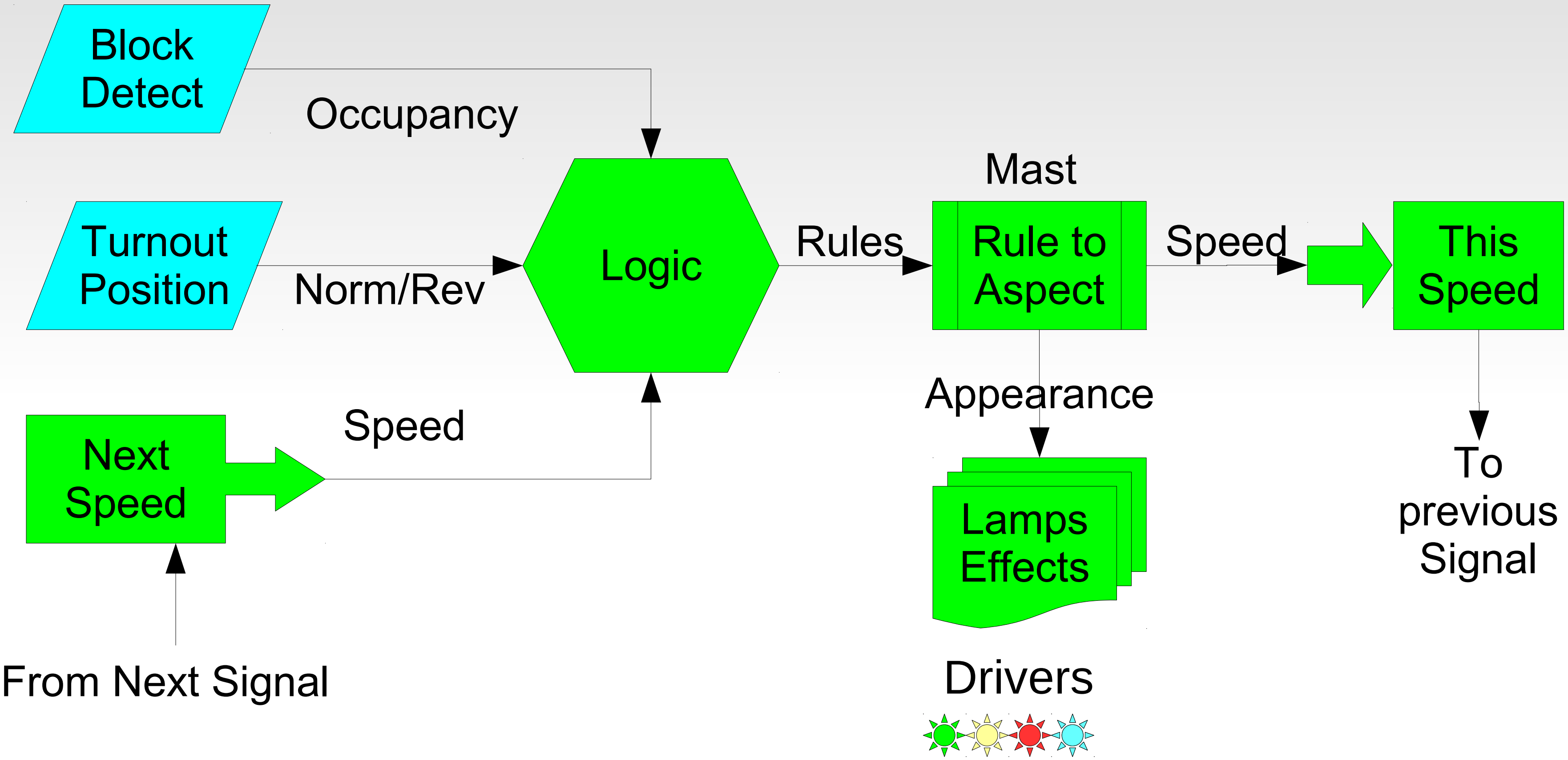
Compiled by: Dick Bronson
RR-CirKits, Inc.

Signaling with LCC.

Part 1 (overview)
www.rr-cirkits.com/clinics/NMRA-2019-Signaling with LCC-A.pdf

Part 2 (details)
www.rr-cirkits.com/clinics/NMRA-2019-Signaling with LCC-B.pdf

Signal Logic Example



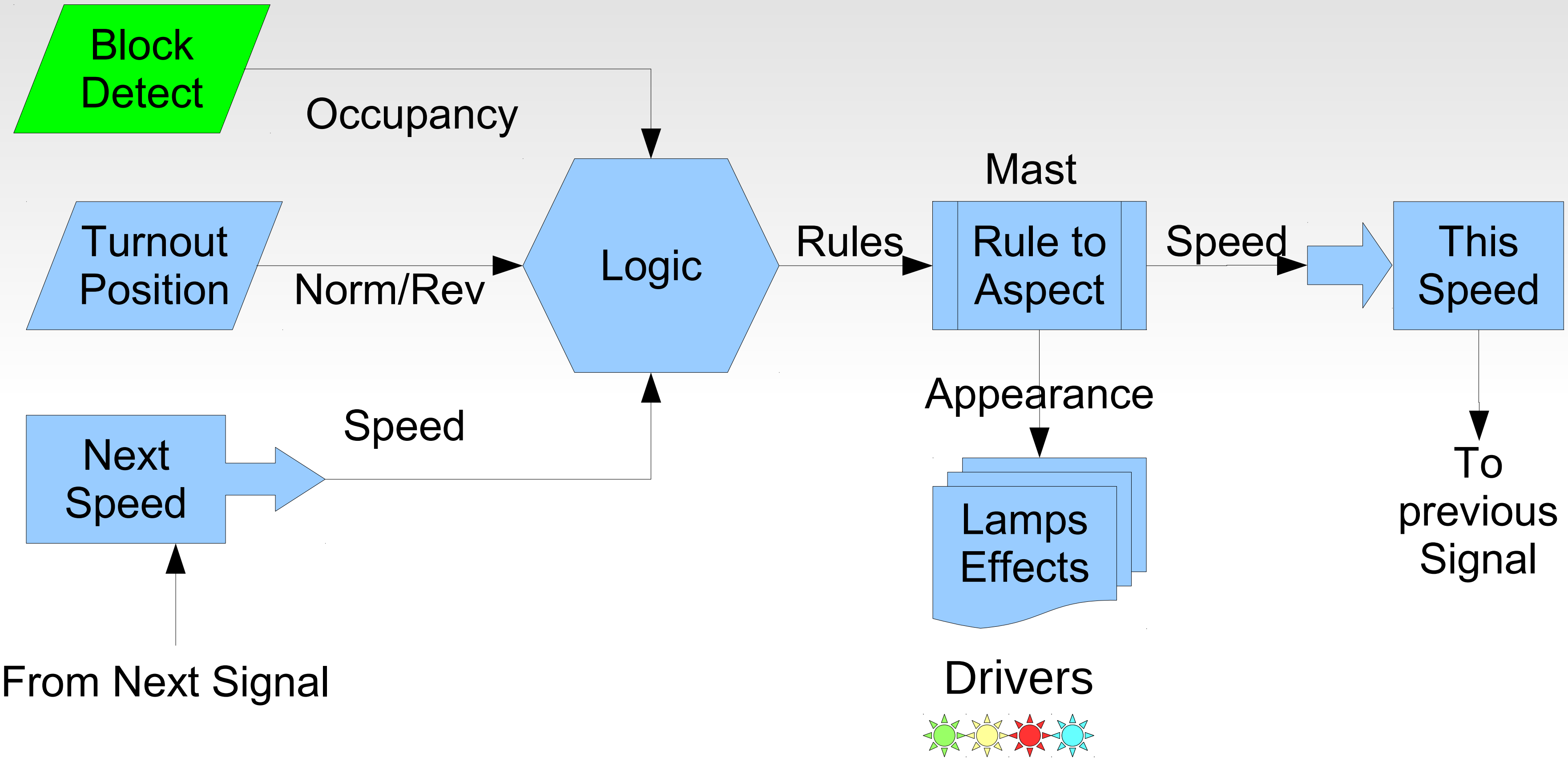
With the Signal LCC all of the control functions required for signaling exist in a single node. Light blue items are taken care of with a daughter card.

If you want to off load (or monitor) any function with a computer you may do so by intercepting the LCC EventIDs that link sections with each other.

Signal Logic

- Signal Logic
- In order to build a signal controller that watches all related status Events from the railroad and CTC panel, and makes independent decisions about the proper signal states and appearances, it must contain internal logic. This logic must either be user controlled or else it must understand all known signaling rules.
- Triggering the evaluation of a conditional is done when any monitored event is seen. There are two trigger options. In the first option evaluation of a conditional is only done if the monitored event actually changes the state of the variable. In the second case the evaluation is done when ever the event is seen, even if there is no resulting change to a variable. This allows repeated single events to trigger a conditional multiple times.

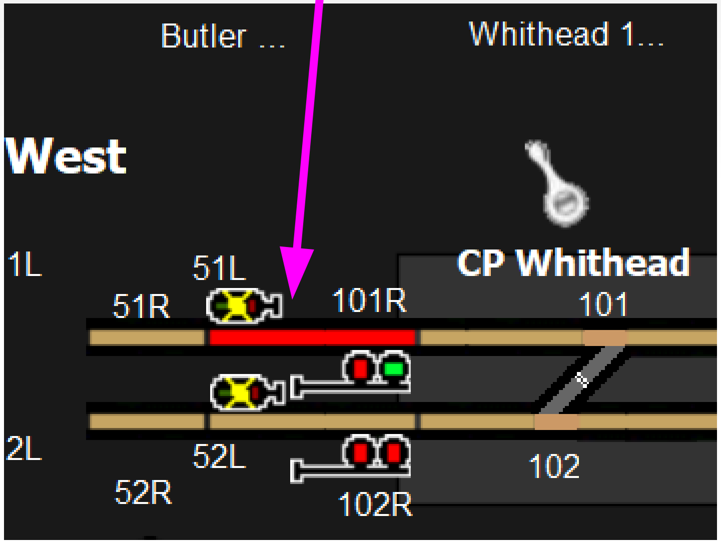
Block Detector Example



The BOD4 and BOD4-CP cards each include 4 block detector circuits for easy connection to the Signal LCC board. These boards use CT coils to prevent track voltage drop and provide 100% isolation.

Block Detector Variables

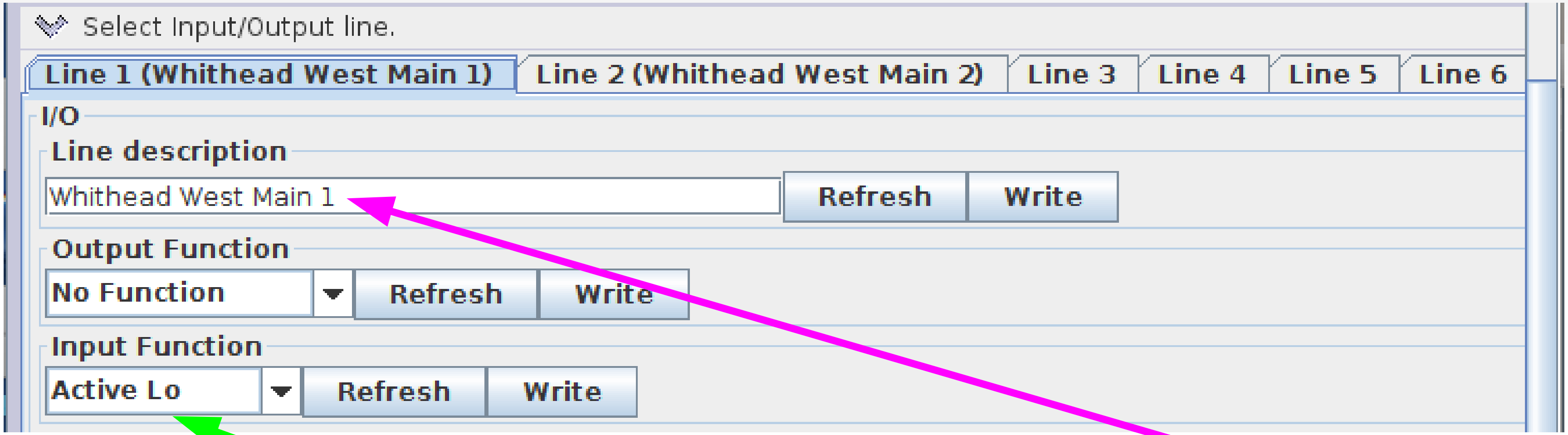
- The block detector watches this section of track. Its name is *Whithead West M1*



Block Detector Variables

- Variables

Variables are used to follow the state of objects of interest such as block detectors, turnout positions, etc. Normally two events will allow the variable to follow the state of some object, true/false, normal/diverging, clear/occupied, etc.



Select Input/Output line.

I/O

Line description: Whithead West Main 1

Output Function: No Function

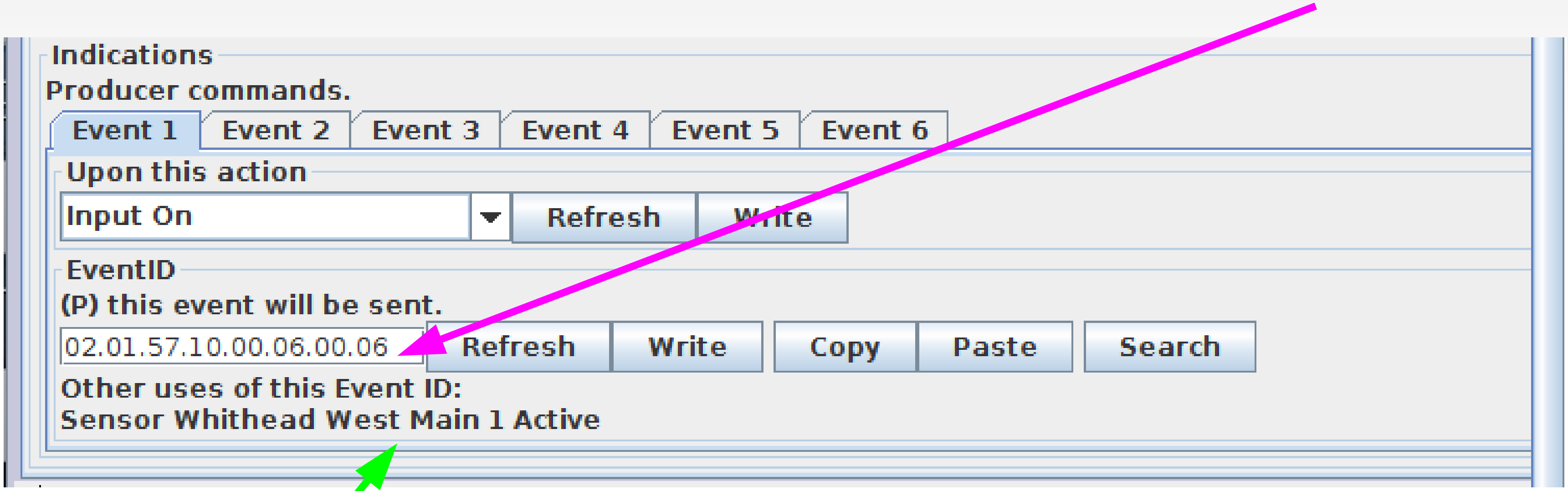
Input Function: Active Lo

Lets start by connecting a block sensor to an input. Its description is 'Whithead West Main 1' so we enter it in the description block and 'Write' it to the node. Detectors are Input Functions with 'Active Lo' so we set that and write it. For a normal Input be sure that the Output Function is set to 'No Function'. Of course 'Line 1' is the one connected to our first block detector.

Block Detector Variables

- Input (Producer) Events

We now go to the Indications (Producers) for this line, and enable two events. The first (Event 1) will be sent when the Input is ’On’ (goes low in our application) When we need to know if the block goes occupied, we will use this EventID.



Indications

Producer commands.

Upon this action

Input On

EventID

(P) this event will be sent.

02.01.57.10.00.06.00.06

Other uses of this Event ID:

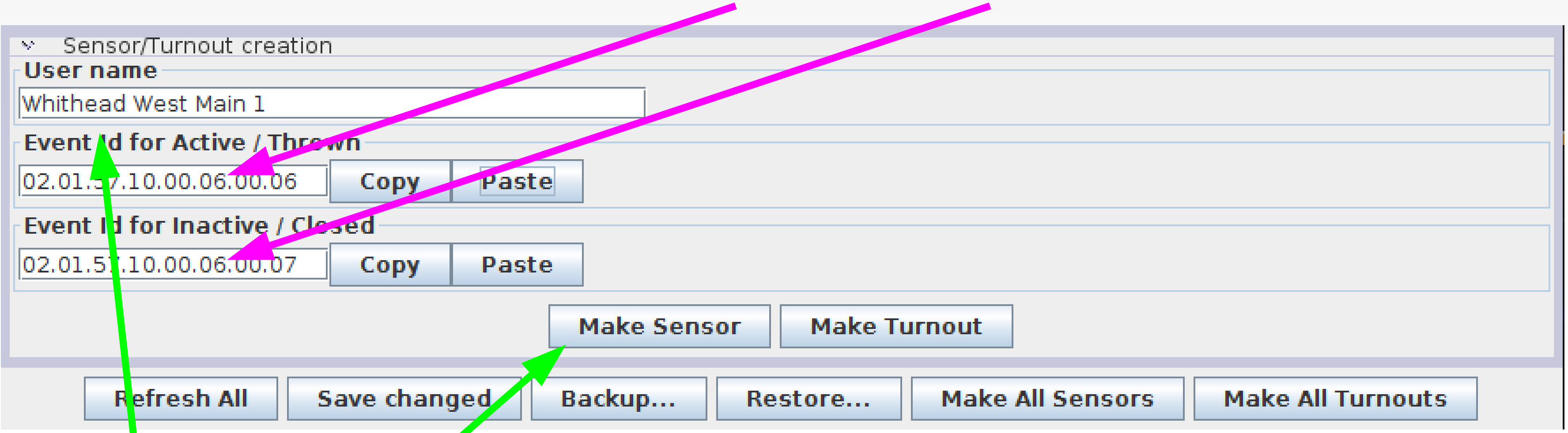
Sensor Whithead West Main 1 Active

Event 2 will be set to ‘Input Off’. Use ‘Copy’ and ‘Paste’ when you need to utilize the magic numbers (EventID) for these events. Its description ‘Whithead West Main 1’ Is noted here to remind you of its function. We can search on this name when we need to use this event in some logic. This information is known because I made a JMRI sensor that follows it. This is a JMRI feature available in the JMRI CDI tool.

Block Detector Variables

- JMRI Sensors

JMRI includes a handy tool at the bottom of the CDI window to make sensors or turnouts from events. LCC Nodes may use two (or more) EventIDs to control sensors and turnouts, so you must use cut/paste to choose the pair that you want for JMRI. For this sensor we use Event 1 and Event 2 that we just defined.



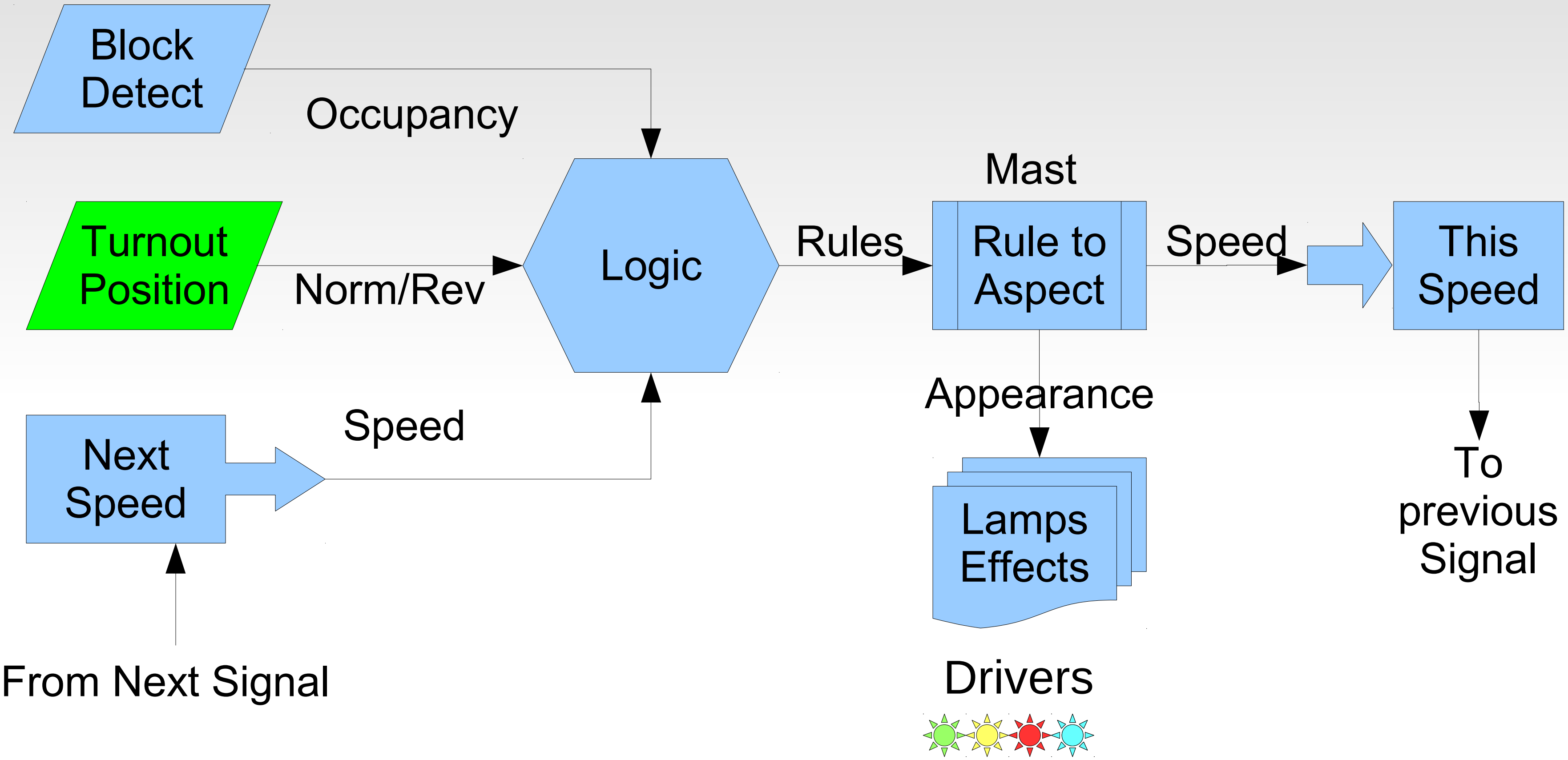
User name: Whithead West Main 1

Event Id for Active / Thrown: 02.01.57.10.00.06.00.06

Event Id for Inactive / Closed: 02.01.57.10.00.06.00.07

Enter the JMRI user name for this sensor, (or turnout) then click on the Make Sensor button. This item will automatically be added to your JMRI Sensor (or turnout) table. Be sure to save the table for future use. Normally this data will become part of a ‘Panels’ file, and be synchronized with the node when loaded.

Turnout Control Example

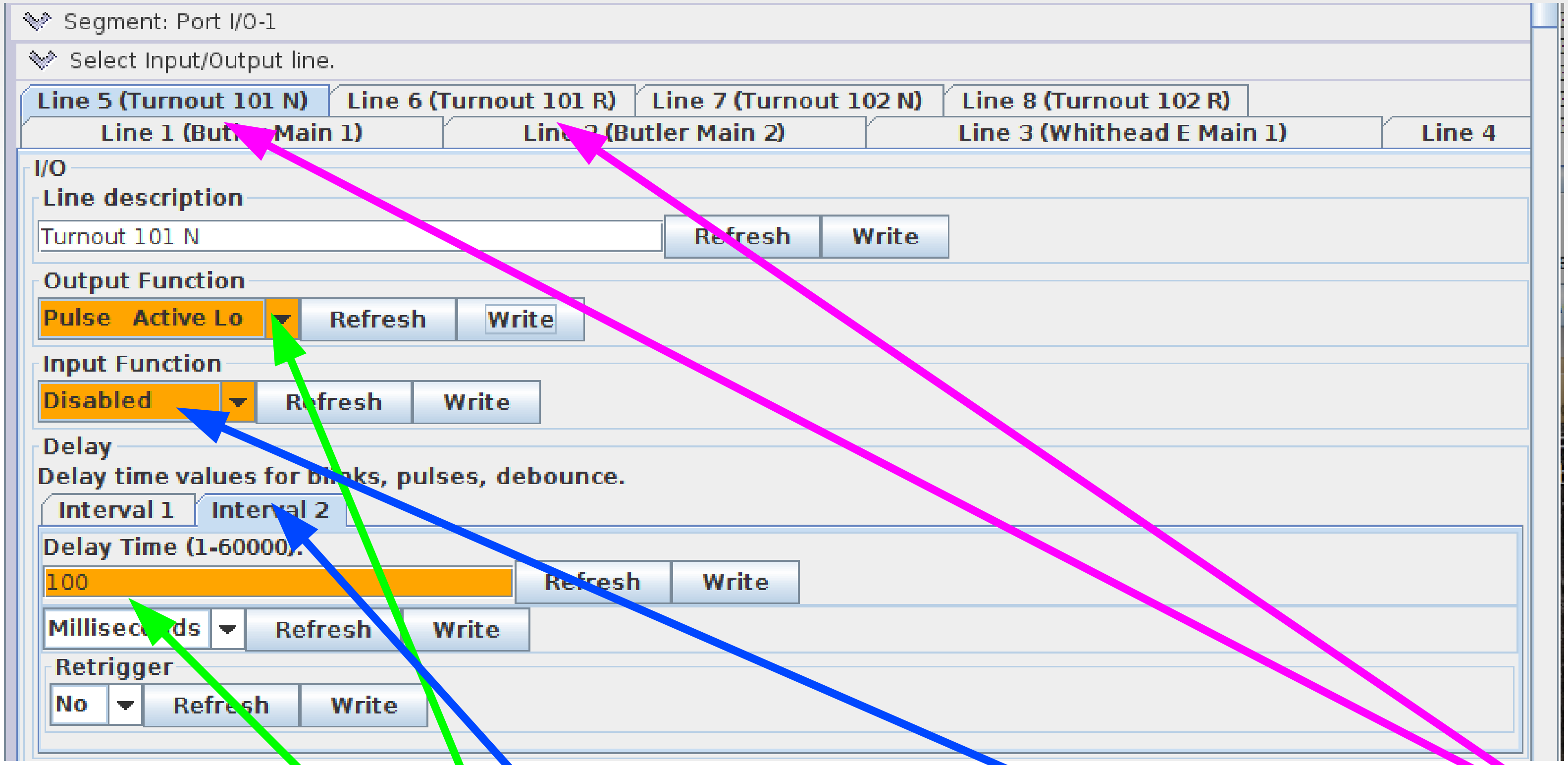


The BOD4-CP cards also include 2 ‘H’ Bridge drivers controlled by the Signal LCC board. These drivers are isolated from the LCC bus to prevent any power supply issues.

Turnout Variables

- Output (Consumer) Events

We now go to the Indications (Producers) for a line. (on board ...07)



Segment: Port I/O-1

Select Input/Output line.

I/O

Line description: Turnout 101 N

Output Function: Pulse Active Lo

Input Function: Disabled

Delay

Delay time values for blinks, pulses, debounce.

Delay Time (1-60000): 100

Milliseconds

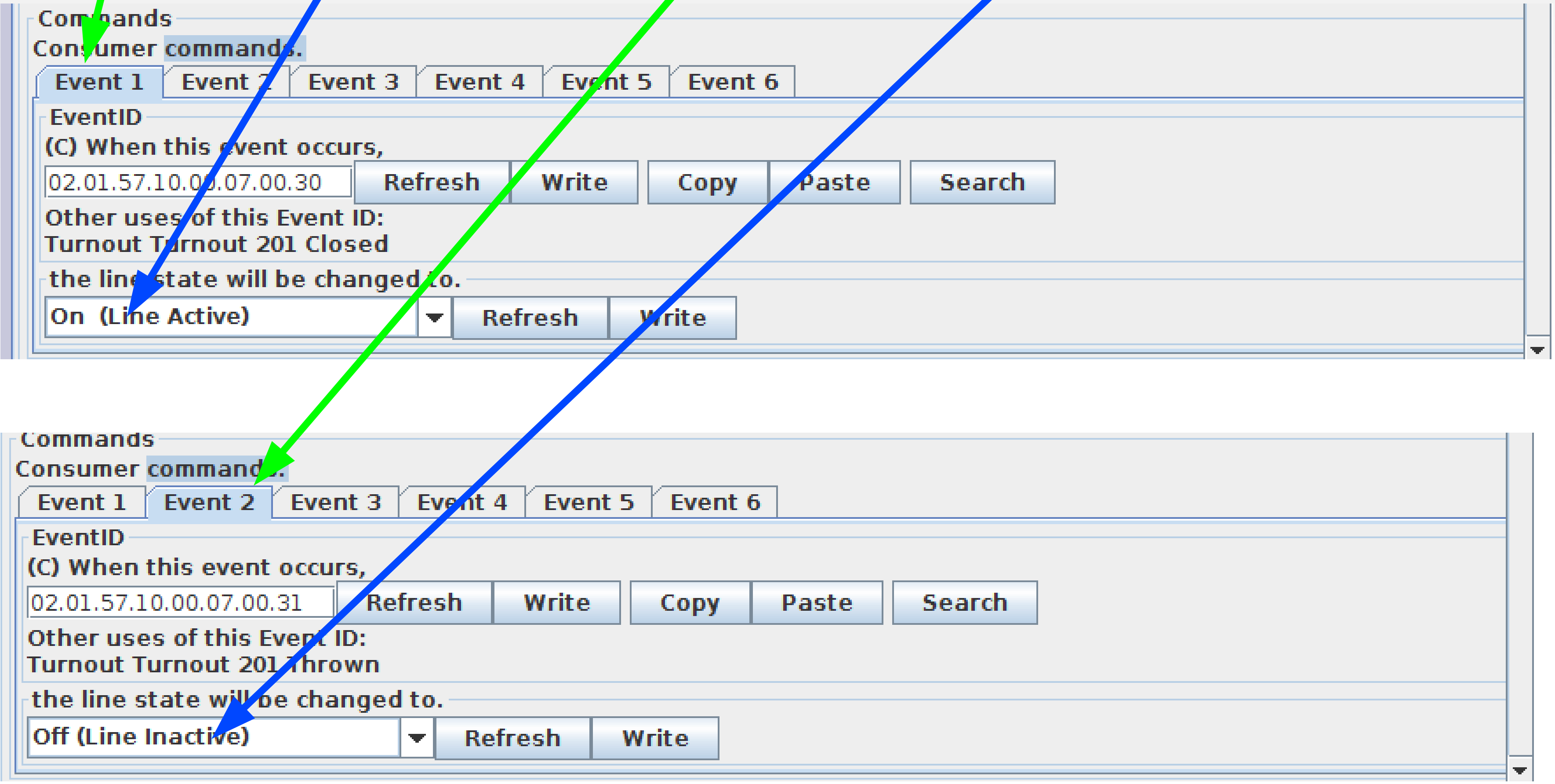
Retrigger: No

Our turnouts are controlled by Kato dual coil solenoids. This requires dual line drivers and 100mS pulse outputs. Normally inputs are disabled for Output Functions. Note: Use Interval 2 for pulse length. Interval 1 is the pulse delay.

Turnout Variables

- Output (Consumer) Events

Event 1 will turn ‘On’ the line and event 2 will turn it ‘Off’. Remember we already specified that ’On’ just sends a 100mS pulse, so our coils are safe.



Commands

Consumer commands.

EventID

(C) When this event occurs,

02.01.57.10.00.07.00.30

Other uses of this Event ID:

Turnout Turnout 201 Closed

the line state will be changed to.

On (Line Active)

Commands

Consumer commands.

EventID

(C) When this event occurs,

02.01.57.10.00.07.00.31

Other uses of this Event ID:

Turnout Turnout 201 Thrown

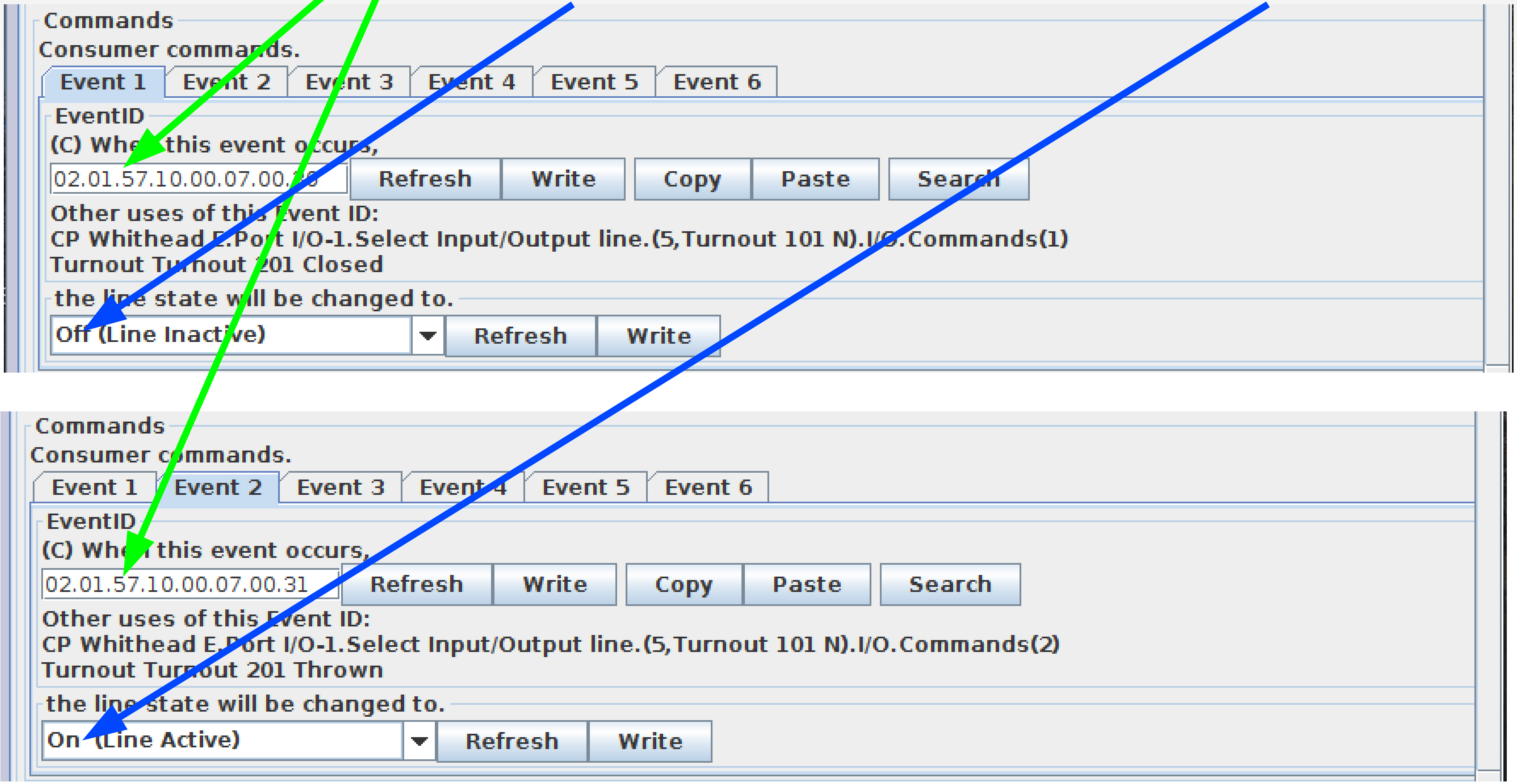
the line state will be changed to.

Off (Line Inactive)

Turnout Variables

- Output (Consumer) Events

To configure the second coil we will do two tricks with events. First we copy and paste the two events from the first line to the second line. Next we reverse their actions. Event 1 will turn ‘Off’ the line and event 2 will turn it ‘On’. Done!



02.01.57.10.00.07.00.30

Other uses of this Event ID:
CP Whithead E.Port I/O-1.Select Input/Output line.(5,Turnout 101 N).I/O.Commands(1)
Turnout Turnout 201 Closed

the line state will be changed to.
Off (Line Inactive)

02.01.57.10.00.07.00.31

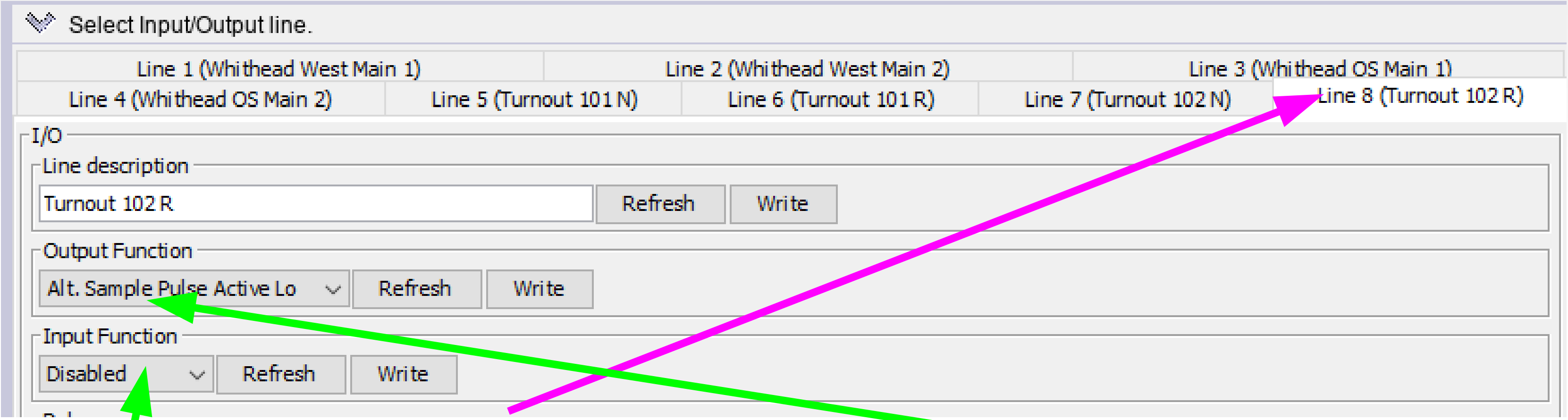
Other uses of this Event ID:
CP Whithead E.Port I/O-1.Select Input/Output line.(5,Turnout 101 N).I/O.Commands(2)
Turnout Turnout 201 Thrown

the line state will be changed to.
On (Line Active)

Turnout Control

- Input (Producer) Events.

- We now get really fancy. To be compatible with the Berrett Hill Touch Triggers we added a ’Sample’ option to our I/O lines. We take advantage of that on the BOD4-CP. Each driver output has a corresponding input line. (lines 5, 6, 7, and 8)



Select Input/Output line.
Line 1 (Whithead West Main 1)
Line 2 (Whithead West Main 2)
Line 3 (Whithead OS Main 1)
Line 4 (Whithead OS Main 2)
Line 5 (Turnout 101 N)
Line 6 (Turnout 101 R)
Line 7 (Turnout 102 N)
Line 8 (Turnout 102 R)
I/O
Refresh
Write
Refresh
Write
Refresh
Write

Line description: Turnout 102 R

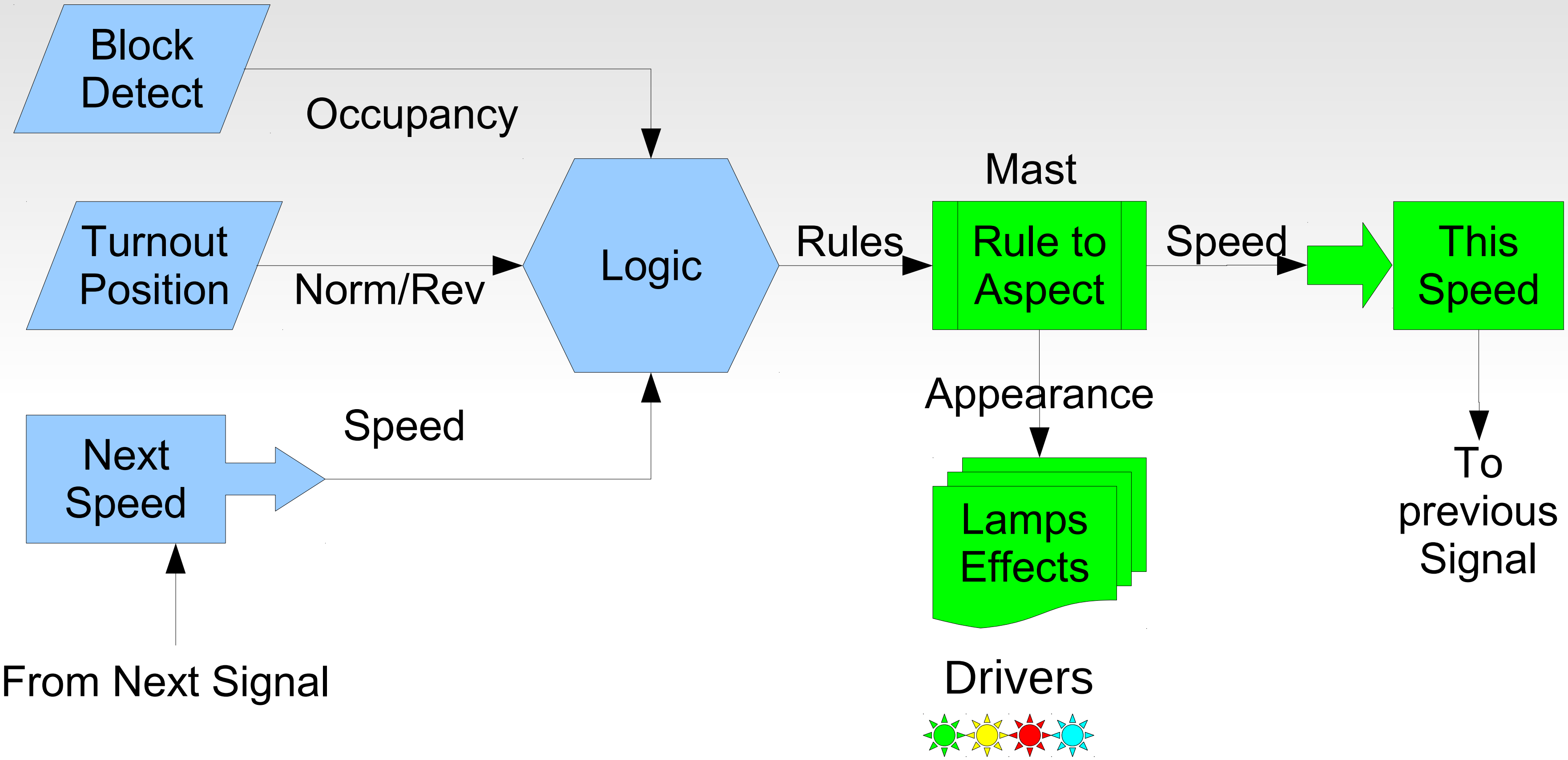
Output Function: Alt. Sample Pulse Active Lo

Input Function: Disabled

- We use the input for Line 8 and connect it to a push button. We leave the ‘Input Function’ as ‘disabled’, but we prefix the output function ‘Pulse Active Lo’ with ‘Alt. Sample’. ‘Sample’ means that now the input state is also watched. ‘Alt.’ means each time it goes ‘low’ it will alternate the input function state. The line still sends its output to drive the turnout as before, but now we can also use the same line (physical wire) as an input by sampling it. BOD4-CP resistors prevent the button from shorting the output drivers.

- For simplicity we just have the line send the turnout control events directly. For realism, combine the control events with occupancy and/or panel information that prevents any turnout movement when occupied, or locked.

Rule to Aspect



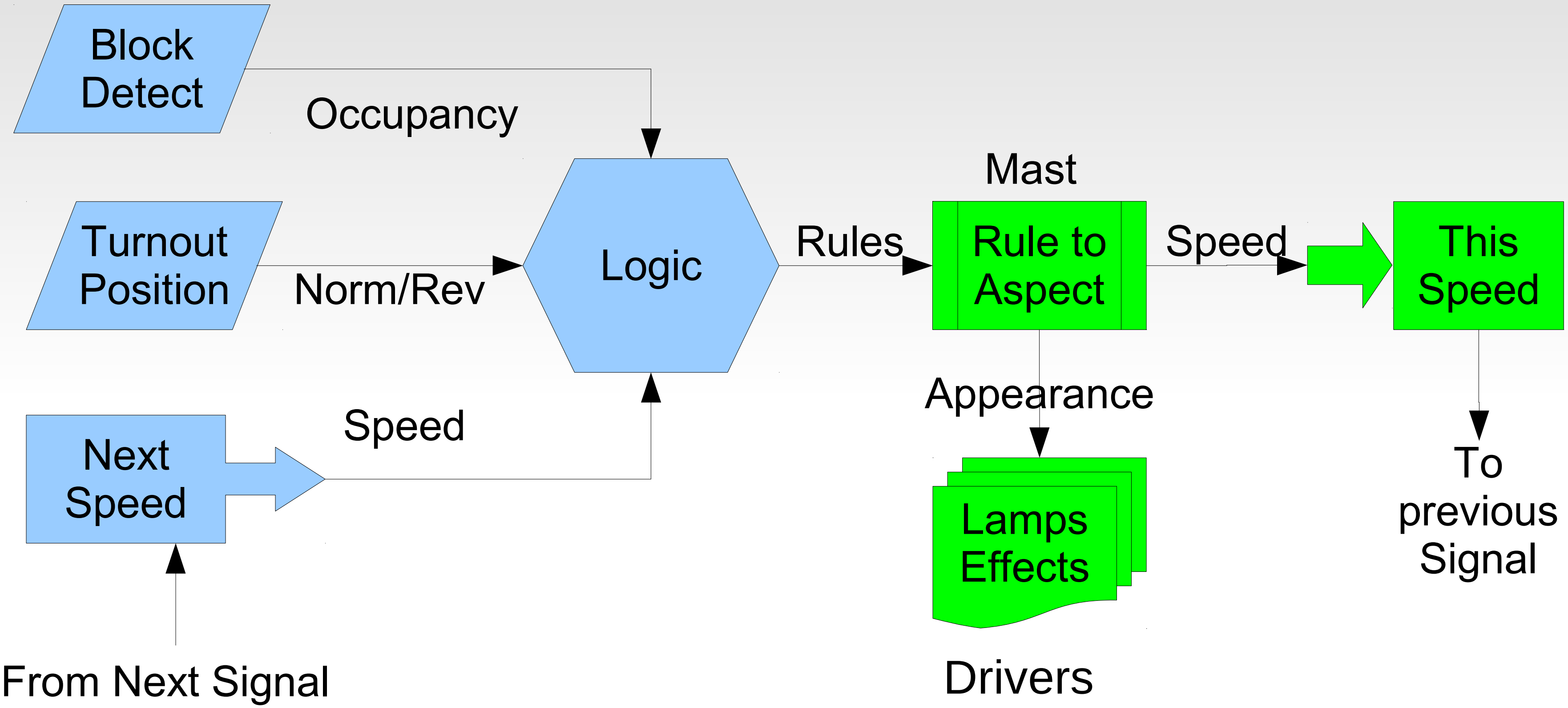
Signal Masts

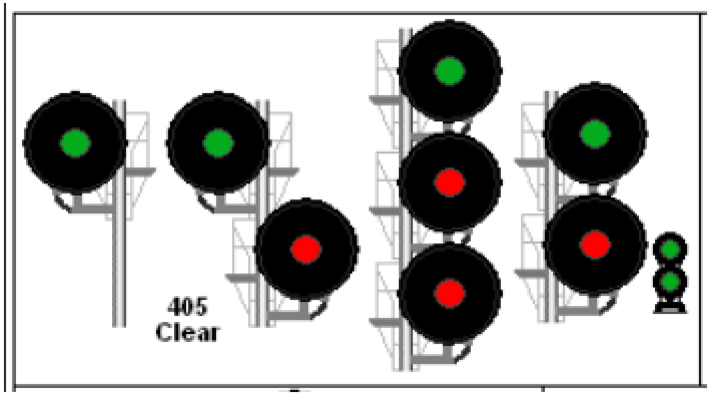
- The EventIDs sent and responded to by each rule are also controlled by the ‘Rule to Aspect’ segment. Because this Rule to Aspect conversion is actually the links between the ’Rule’ events and the actual hardware we call it all ’Masts’ and treat it as one segment in the CDI.
- The Signal LCC supports up to 8 Masts, each of which supports 8 indications (Rules). If a single mast requires more than 8 aspects, then another mast may be logically linked with a previous one.
- A ’Signal Mast’ definition makes two assumptions that simplify things.

  1) Only one signal aspect may be shown at a time. Setting any aspect automatically cancels any previous aspect.

  2) A mast may only have a single speed limit at a time. This ’Speed’ is the speed allowed when passing the mast.
- Making a mast ’Linked to Previous’ carries the above assumptions over from any previous mast/masts. Speed is always taken from the first mast.

Rule to Aspect



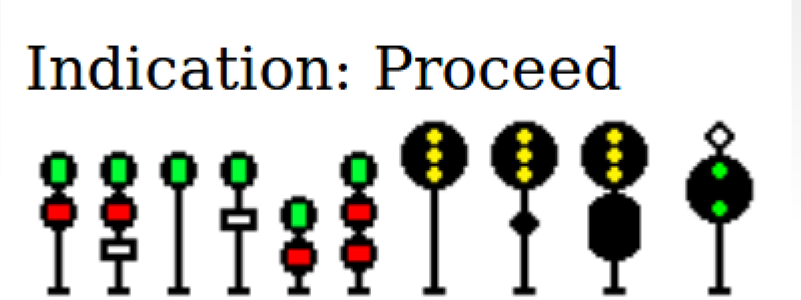


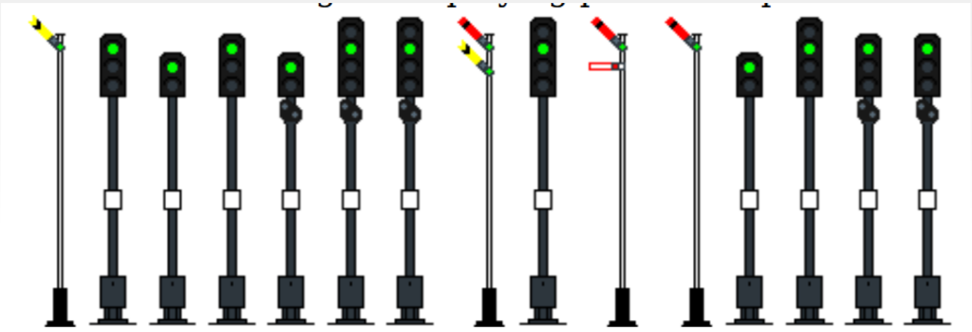
A quick look at any railroad rule book will reveal that the same rule may be displayed in many ways. This means that we need a 'Rule' to 'Aspect' conversion process.

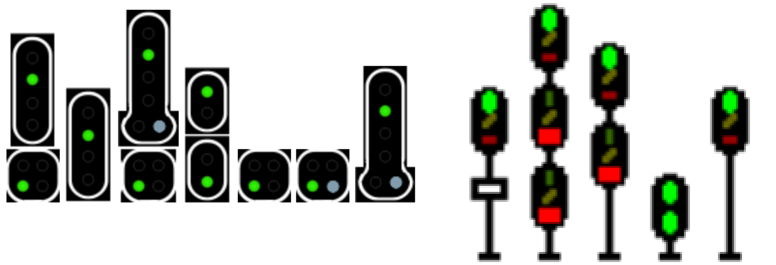
Rule to Aspect

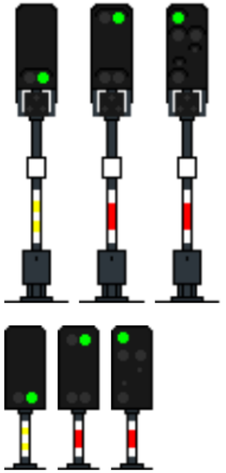
- Rule to Aspect conversion

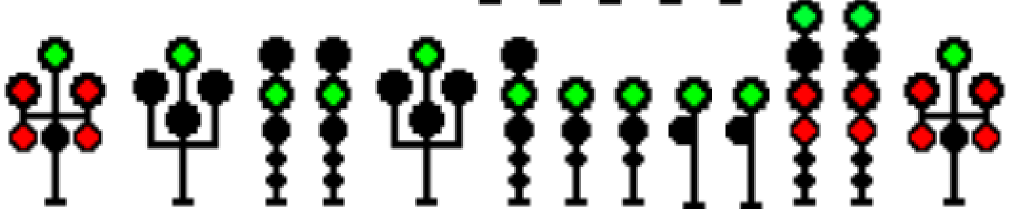
Signal rules such as ’Stop’, ‘Approach’, ’Clear’, Etc. are displayed differently on different types of signals. A simple way to make these conversions is needed.

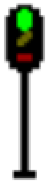






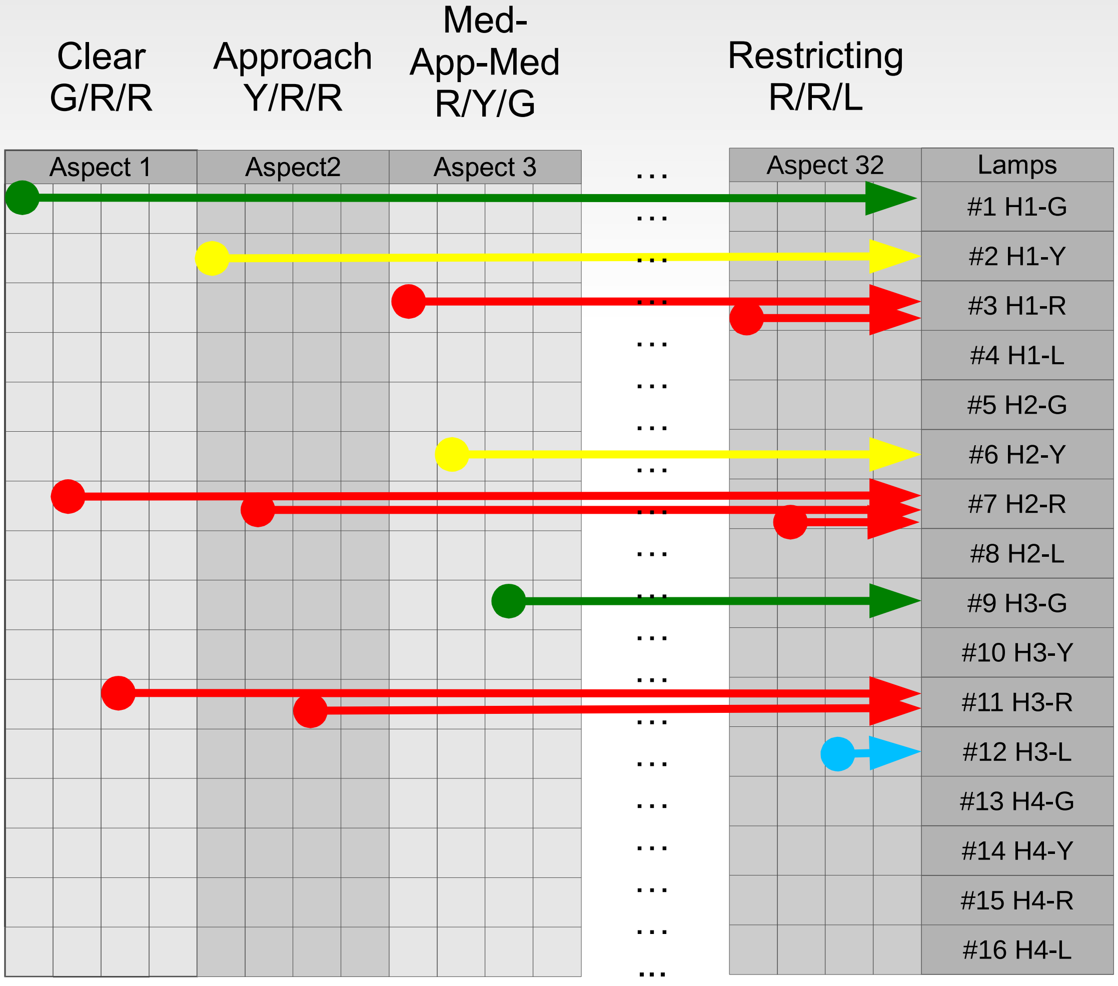




- My point of course is that a signal driver designed for  may not work for you.

Aspects to Appearances

- One way to visualize a solution to the problem is with a grid. Each (of 32) aspects has 4 columns that may each control any one of the 16 lamp drivers. The G, Y, R, L in the lamp names is arbitrary and may drive any color LED or pair of LEDs.



Clear
G/R/R

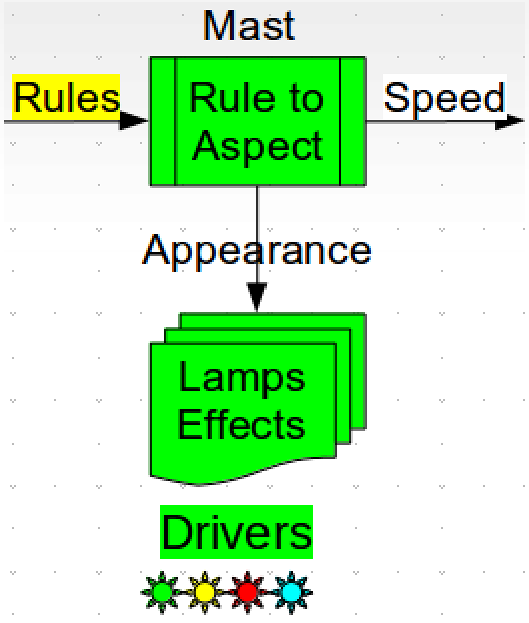
Approach
Y/R/R

Med-
App-Med
R/Y/G

Restricting
R/R/L

Aspect 1 | Aspect2 | Aspect 3 | … | Aspect 32 | Lamps

#1 H1-G
#2 H1-Y
#3 H1-R
#4 H1-L
#5 H2-G
#6 H2-Y
#7 H2-R
#8 H2-L
#9 H3-G
#10 H3-Y
#11 H3-R
#12 H3-L
#13 H4-G
#14 H4-Y
#15 H4-R
#16 H4-L



Mast

Rules

Rule to Aspect

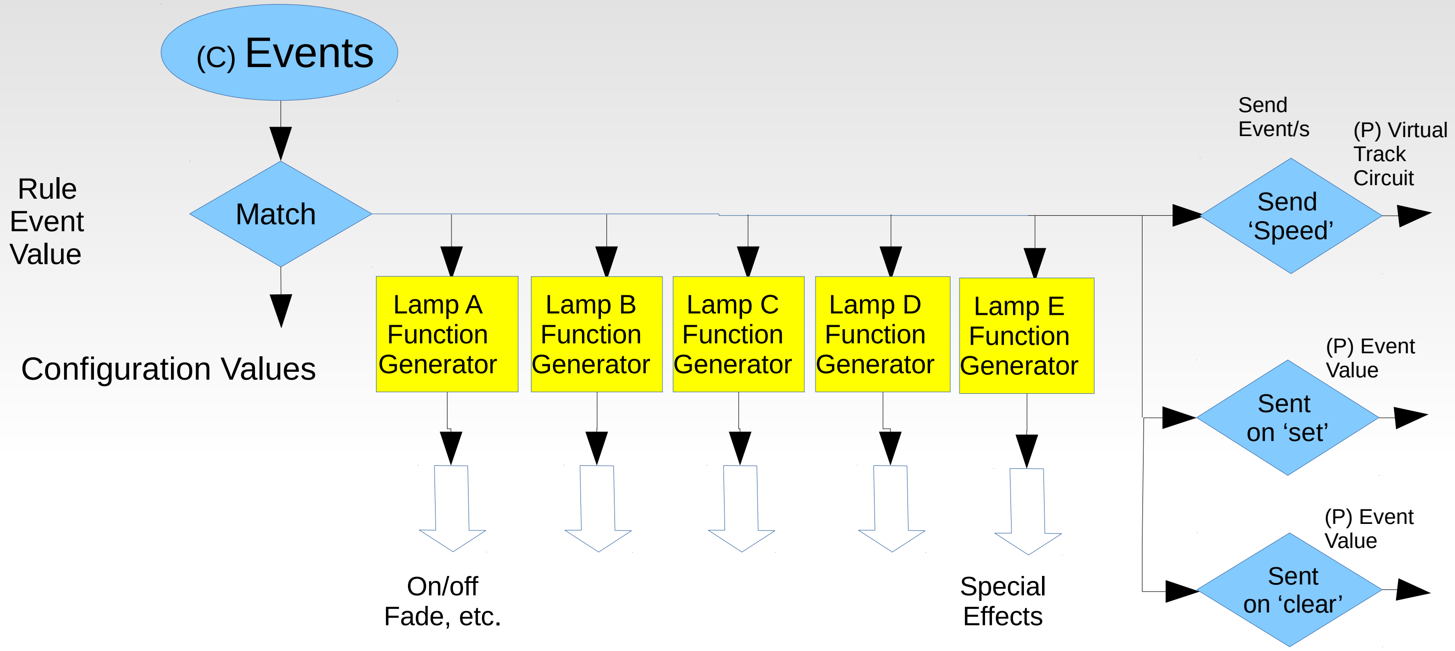
Speed

Appearance

Lamps Effects

Drivers

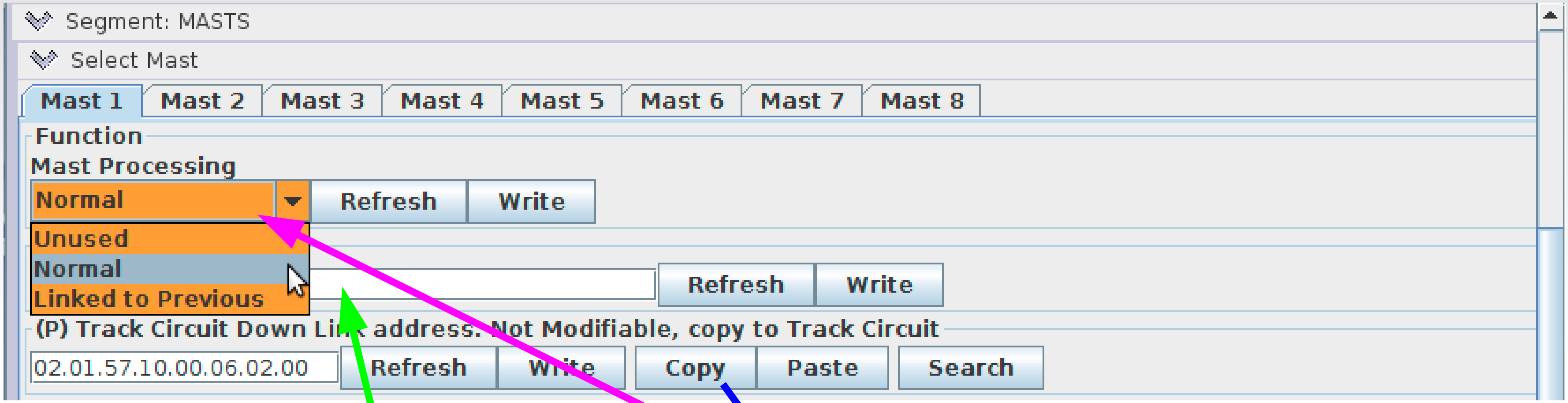
Aspect to Appearance



- Signal Masts

This flow chart shows the Mast functionality in a different way. Any signal rule that is seen (matched) can send up to 4 lamp control messages, plus an optional special effect. Speed is sent by the virtual track circuit, which also sends 2 optional events. These optional events may be used to send indications back to a CTC panel, or allow some other process to monitor aspect changes.

Signal Mast Setup



Mast Processing: Normal

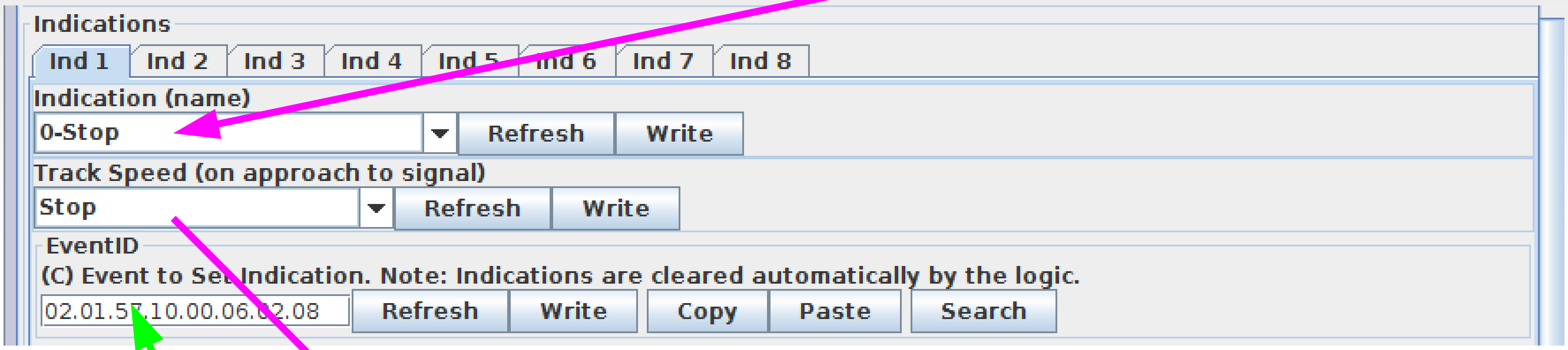
(P) Track Circuit Down Link address: Not Modifiable, copy to Track Circuit

02.01.57.10.00.06.02.00

- Function
- To use a mast you must first change it to ’Normal’ or ’Linked to Previous’.
- Next give it a Mast ID so you can easily find it again later. This could be a CTC panel number, a mile marker, a control point name, etc.
- Track Circuit Down Link Address. This fixed EventID is used as a pointer to the current track speed setting for this mast. Copy this number to any track circuit receive (RX) table to make it easy for logic to follow speed.

Indications (Name)

- Indications tell the crew what to do at a signal. The ’Rule’ or ’Name’ is the shorthand for the Indication. The options include common names.



Indications

Indication (name): 0-Stop

Track Speed (on approach to signal): Stop

EventID

(C) Event to Set Indication. Note: Indications are cleared automatically by the logic.

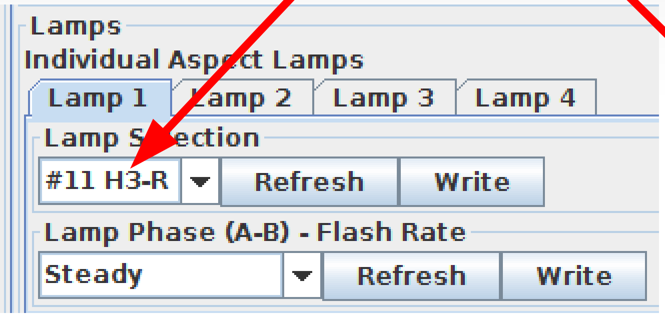
02.01.57.10.00.06.02.08

- The selected ’Track Speed’ (one of eight possible) is the value that will be sent back to the previous signal over the Virtual Track Circuit. If the names don’t match your rule book, simply pick something similar. Its actually just a code number to the track circuit.

- EventID to Set Indication. This is the EventID used by the signal logic to activate this signal rule.

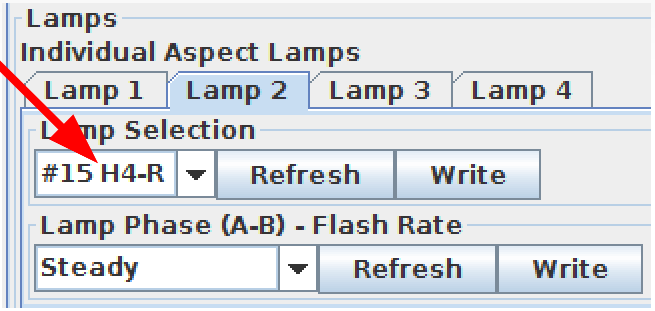
Lamps

- The bottom line in displaying each aspect is to choose what lamps are lit. After all, that is what the crew (and the visitors) actually see.
- This mast on the signal bridge has dual head searchlights. This means ’Stop’ will display as Red over Red. Pick the appropriate lamps to show this.



Lamps
Individual Aspect Lamps
Lamp 1 | Lamp 2 | Lamp 3 | Lamp 4
Lamp Selection
#11 H3-R

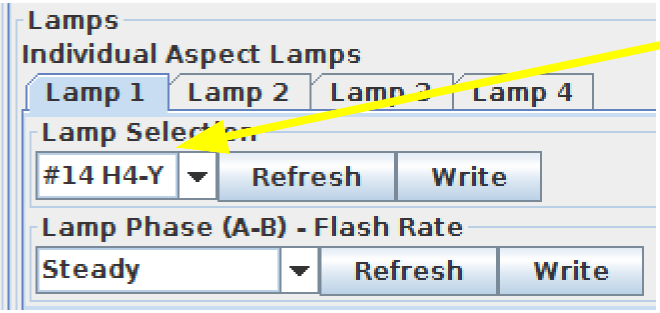
Lamp Phase (A-B) - Flash Rate
Steady



Lamps
Individual Aspect Lamps
Lamp 1 | Lamp 2 | Lamp 3 | Lamp 4
Lamp Selection
#15 H4-R

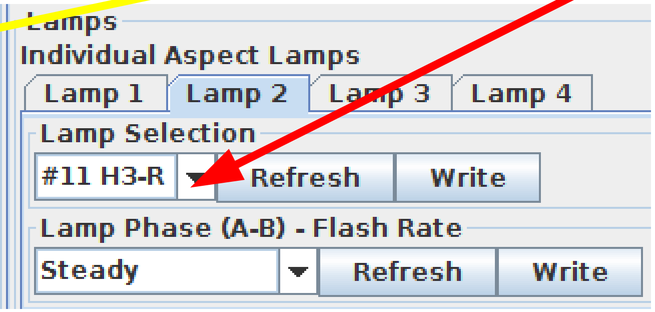
Lamp Phase (A-B) - Flash Rate
Steady

- To show Indication 2 - ’Approach’ display Yellow over Red.



Lamps
Individual Aspect Lamps
Lamp 1 | Lamp 2 | Lamp 3 | Lamp 4
Lamp Selection
#14 H4-Y

Lamp Phase (A-B) - Flash Rate
Steady



Lamps
Individual Aspect Lamps
Lamp 1 | Lamp 2 | Lamp 3 | Lamp 4
Lamp Selection
#11 H3-R

Lamp Phase (A-B) - Flash Rate
Steady

- Continue in like manner until you have entered each possible aspect.

Signal Lamp Drivers

- Each Indication (Aspect) can be displayed with as many as four lamps. If you have a rare signal aspect that can not be shown with just 4 lighted lamps you can make a duplicate mast to light any additional lamps. Remember dual lamps that light together only count as a single lamp. (e.g. in Position Lights and Color Position lights) Only lighted lamps count. Only controlled lamps count. A marker that is always lighted can simply be powered full time.
- Lamp Phase – Flash Rate may be used to flash signals automatically. One common example is ’Advance Approach’ which is commonly displayed with a flashing yellow lamp. Setting an appropriate Flash Rate means that the signal logic doesn’t need to worry about flashing the signal or overloading the bus with unecessary traffic. Providing both A and B phase options is handy for grade crossing flashers or other alternating lamp situations. Phase A is ’on’, ’off’, ’on’, etc. Phase B is ’off’, ’on’, ’off’, etc. Both are off when the aspect is inactive.

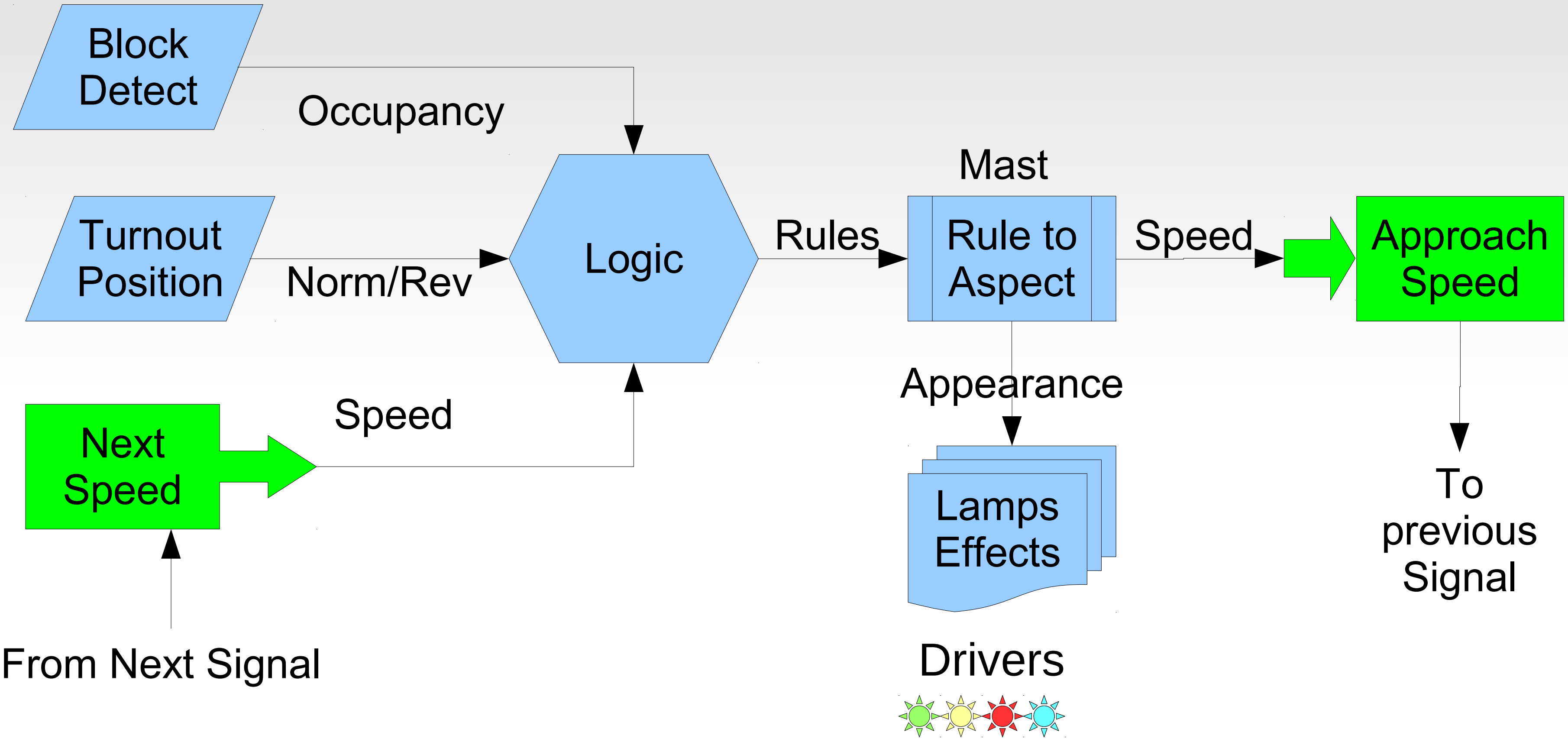
Lamps – Special Effects

- Incandecent fade. Prototype signal lamps are wired differently than standard household lamps. They include a ballast resistor in series with the lamp. This ballast serves two purposes. One is simply to set the brightness of the lamp. More importantly, when a cold lamp is first powered up it prevents the normal surge of high current by dropping most of the voltage across the ballast until the lamp warms up. The visual result of this is that a signal does not blink on instantly. In fact signals lamps fade on slowly enough to be noted. Of course even houshold incandecents fade off slowly as the lamps cool down again.
- Transition effects. The B&O signal clip we saw earlier showed an interesting transition between Clear and Stop. Not only does it show the fade up and down, but it interjects a brief 'Approach' into the change. I can not remember why this is done, but selecting 'Transition Down' as a special effect on 'Stop' will allow you to do this. (and wow that rivet counter in your crew)

Lamps – Special Effects

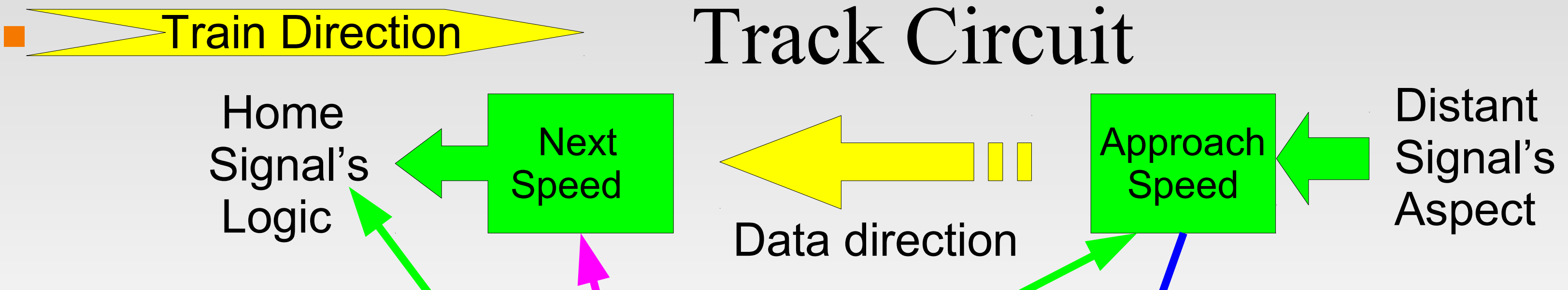
- H2 Red Flicker. Many of you probably know that many searchlight signals (not just the H2) have an internal arm that swings back and forth in front of the lamp. It hangs by gravity with its red roundel in center position. To show either green or yellow requires swinging the arm one way or the other out of its center position with electromagnets. Not quite as obvious is the fact that you can not change between the yellow and green roundels without going past the red roundel between them. This is what causes the red flash. The other part of the effect is that the arm is free swinging and during a change it will often actually overshoot its position before it settles down in its new position. This swings the color roundels past their normal positions which causes the signal to flicker off briefly.
- Strobe lights can be found around the layout. Sometimes it is nice to be able to utilize unused signal outputs for other purposes.

Track Circuits



Paste the next masts ‘Up Link EventID’ from the mast to a local track circuit. This creates a virtual link directly into the logic variables by virtual name rather than by using actual event numbers. The logic for a mast can be setup, or mass produced, without knowing any actual mast IDs ahead of time.

Signal Masts – Track Circuit

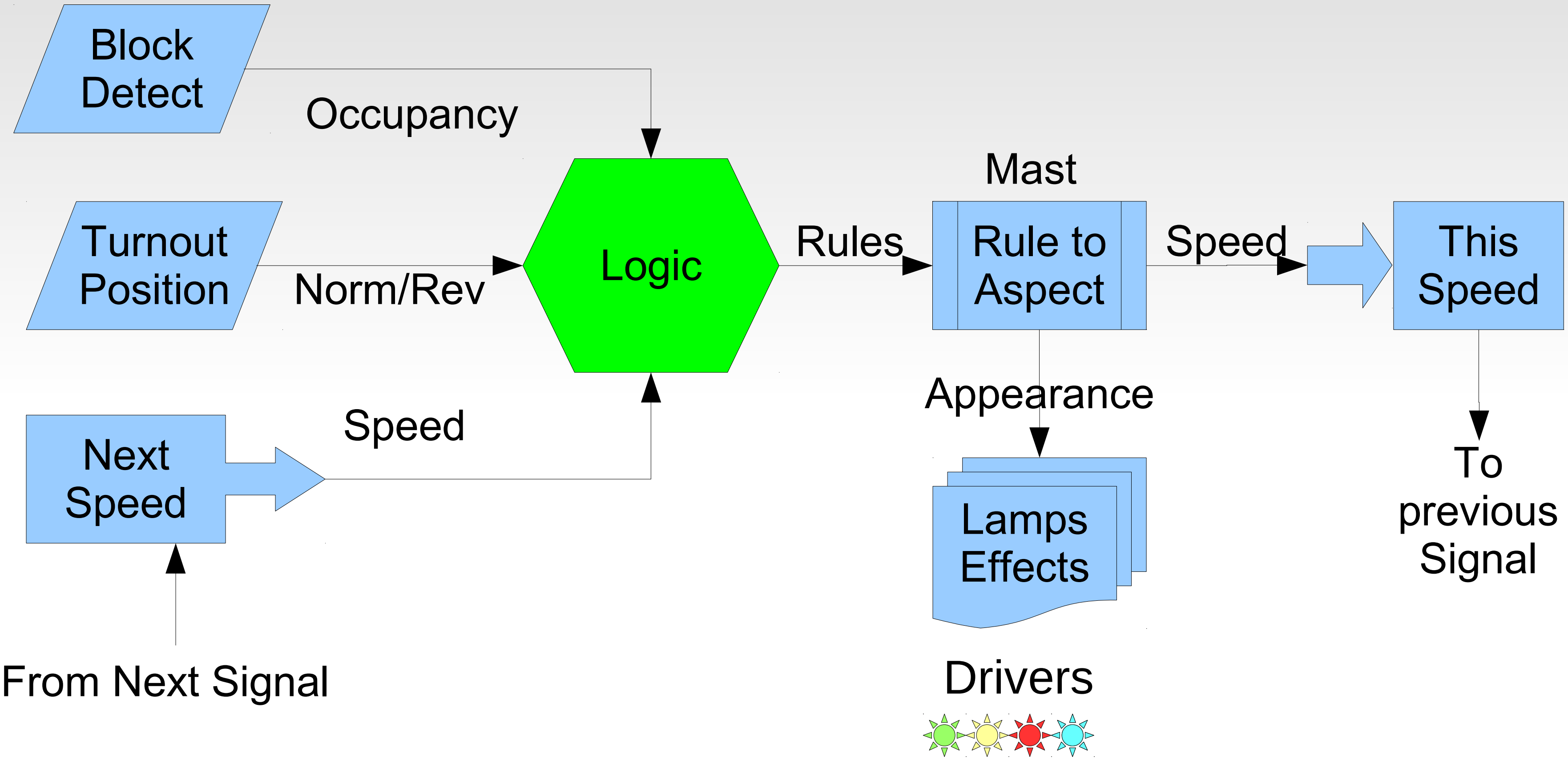


- When calculating signal rules, the most important information from the next signal is the required track speed on approach to that signal. In many cases this information actually is a part of the rule name.

- In modular layout setup, getting this information easily from module to module is the single biggest roadblock to installing authentic signaling. Our Virtual Track Circuit concept was designed to help simplify this.

- To link the speed selected on a mast to the logic of a previous mast, simply copy the 'Track Circuit Down Link address' from the next (distant) mast, and paste it into the 'Remote Mast Up Link address' of this mast. This automatically makes the speed information from one mast available to the logic of another mast without requiring the entry of each specific EventID for every speed change into the appropriate logic conditionals.

Signal Logic



We have covered all the edges. Now we can talk about the central subject, Signal Logic itself.

Signal Logic

- Signal Logic may be done by JMRI just as it is done for simpler hardware. In that case JMRI listens to the various events related to turnouts positions, occupancy detection, and signal aspects. It then calculates the proper aspect and sends that information to the signal head drivers.
- Some LCC nodes contain internal logic that may be used to calculate signal aspects. In that case the signals can operate properly even without JMRI running. This requires a full understanding of your signal system’s rules and more node configuration up front, but the reward is a more robust signal system that always runs whenever the LCC and layout power is turned on. Configuration errors are limited to the area where they have been made, and do not impact other parts of the system that are already running properly. The system may be expanded or corrected while it is in operation, an important consideration for club layouts where some members may be running trains while others are working on the layout.
- A third option, and the one we see here in the Signal Demo layout, is that the LCC nodes completely control the signals, but a JMRI panel exists that can monitor the layout and even simulate inputs to it.

Signal Logic

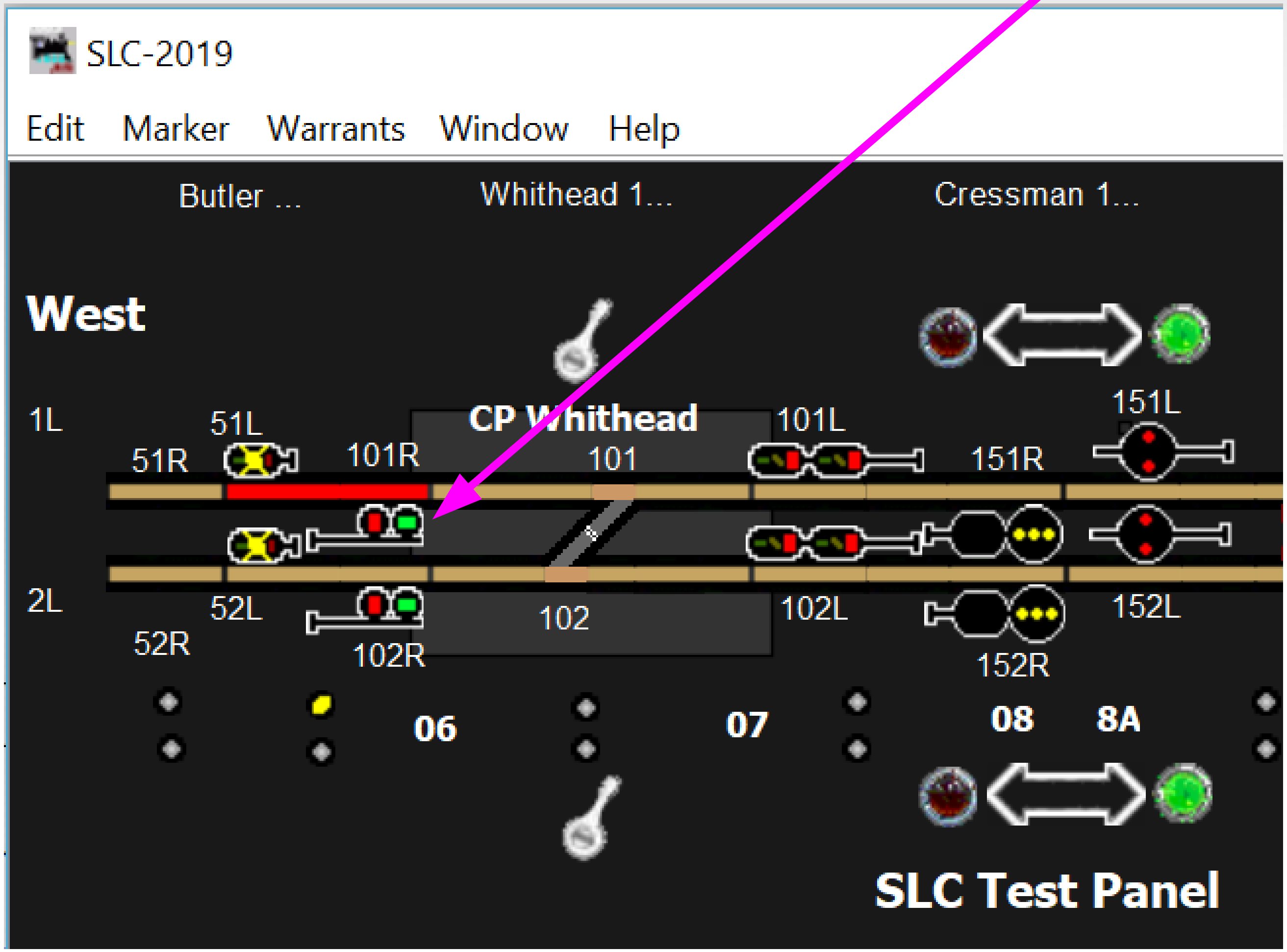
- Signal Logic is just a series of conditions (called conditionals) that are checked to see what signal rule should currently be in effect. These conditions simply execute the rule book’s descriptions of how the signals must work.
- Logic conditionals should be easy to cascade with the calculations for the most restrictive rules having priority over less restrictive rules. We do this by checking each conditional in order from top down. Any rule that is found to be true first checks to see if any more restrictive rule is still in effect. (which exits processing if found) Then it sends its appropriate events, and finally skips over any less restrictive rules for that mast.
- Built in logic conditionals may directly send events representing signal rules (or anything else) when it is found to be true. (or false) A cascade option allows even more events to be sent in special situations. Note: this logic may be used for many other purposes other than just calculating signal aspects.

Internal Signal Logic

- Logic Functions consist of the usual AND, OR, XOR operators. In addition there are two ’change’ operators. These change the true/false sense of a conditional based on the AND and OR of the variables.
- Additionally we have added a non-standard logic operator called ’AND Then’. This makes it very easy to keep track of train direction. You can simply watch two block detectors and determine train direction by the order in which they are activated.
- The processing options for each conditional are to ’Send then Exit Group’, ‘Send then Evaluate Next’, ‘Exit Group’, and ’Evaluate Next’.
- A recent addition is the ability to control the ‘Send’ action associated with both true and false evaluations of a conditional. e.g. ‘Send then Exit Group if True’, or ‘Send then Exit Group if false’. In addition you can send one event if ‘true’ and a different event if ‘false’.

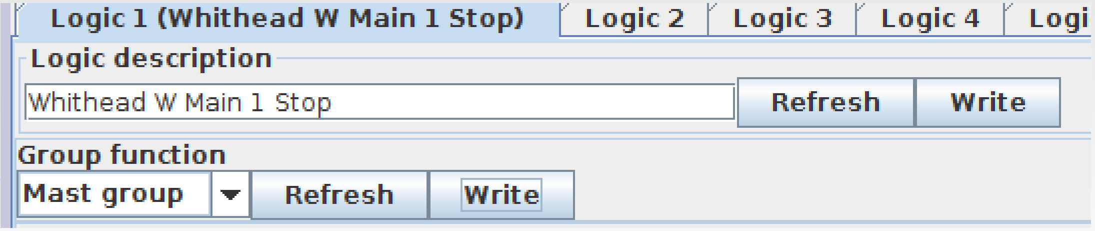
Internal Signal Logic

- First we will configure a very simple signal. 101R is pretty easy to figure out.



Logic Conditionals

- Normally Signal Logic Conditionals will have a Group function of ’Mast Group’ or else ’Last’.



Logic description: Whithead W Main 1 Stop

Group function: Mast group

- The function of a conditional ’Group’ is to pick the most restrictive rule for a mast and send it to the mast table for conversion to the proper aspect.

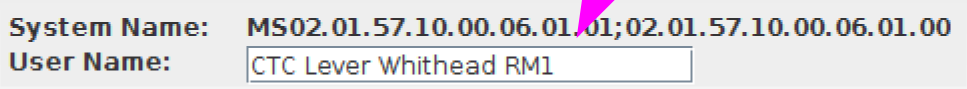
Trailing Signal Logic

Comments	Variable 1	Funct	Variable 2	This Aspect	This Speed
Not CTC-Right / OS occupied	**Not** CTC-Right	OR	OS BOD	Stop	Stop
Siding selected / Main occupied	Turnout Reverse	OR	Main BOD	Stop	Stop
Next speed Stop	Main Mast Stop	null		Approach	Medium
Next speed Medium	Main Mast Medium	null		Approach Medium / Advance Approach	Clear
Next speed Clear	Main Mast Clear	null		Clear	Clear

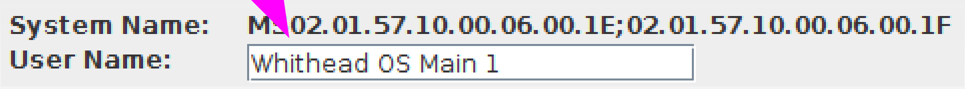
- To create ’**Not** CTC-Right’ simply reverse the events controlling ’Variable 1’ for that conditional. This data is from the direction lever on the CTC panel.
- We check for wrong CTC direction, the turnout against us, the OS occupied, and the track past the turnout occupied. Any of these will set the signal to Stop.
- If the signal has not been set to Stop, then we check to see if the next signal’s speed is ’Stop’. (Main Mast Stop) If so we set this signal to ’Approach’ with a speed of ’Medium’. (or ’Approach’) Sometimes it is helpful to realize that ’Approach’ when used by itself is short hand for ’Approach Stop’. (you are approaching a stop signal)
- If not Stop, then we check for next signal’s speed of ’Medium’ (or ’Approach’) and set our aspect appropriately. (Approach Medium or Advance Approach)
- Finally, finding nothing more restrictive, we can set our signal to ’Clear’.

Logic

Comments	Variable 1	Funct	Variable 2	This Aspect	This Speed
Not CTC-Right / OS occupied	**Not** CTC-Right	OR	OS BOD	Stop	Stop

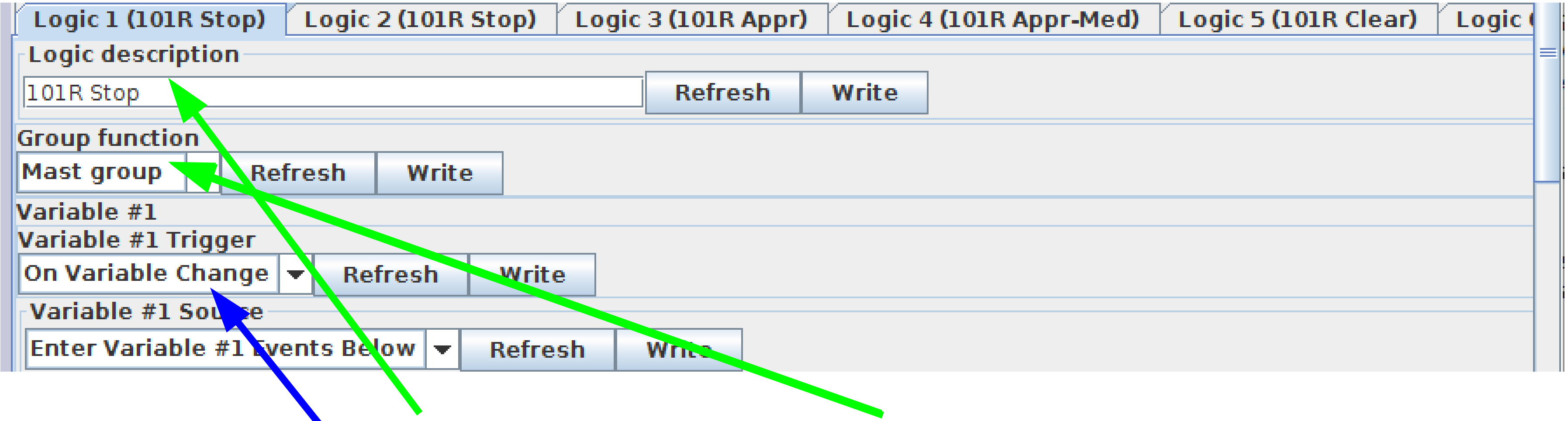


System Name: MS02.01.57.10.00.06.01.01;02.01.57.10.00.06.01.00
User Name: CTC Lever Whithead RM1



System Name: MS02.01.57.10.00.06.00.1E;02.01.57.10.00.06.00.1F
User Name: Whithead OS Main 1

- These two variables as seen in JMRI. I used the Sensor/Turnout creation tool to enter them.



Logic 1 (101R Stop) | Logic 2 (101R Stop) | Logic 3 (101R Appr) | Logic 4 (101R Appr-Med) | Logic 5 (101R Clear) | Logic

Logic description: 101R Stop

Group function: Mast group

Variable #1

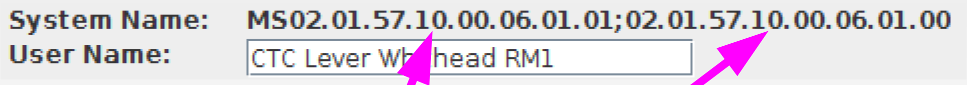
Variable #1 Trigger: On Variable Change

Variable #1 Source: Enter Variable #1 Events Below

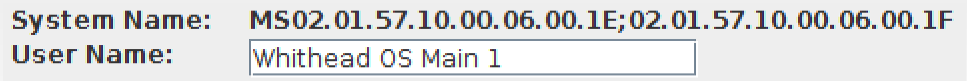
- Enter the logic description and set the function to Mast Group. Logic defaults to watching variable changes.

Logic

Comments	Variable 1	Funct	Variable 2	This Aspect	This Speed
Not CTC-Right / OS occupied	**Not** CTC-Right	OR	OS BOD	Stop	Stop

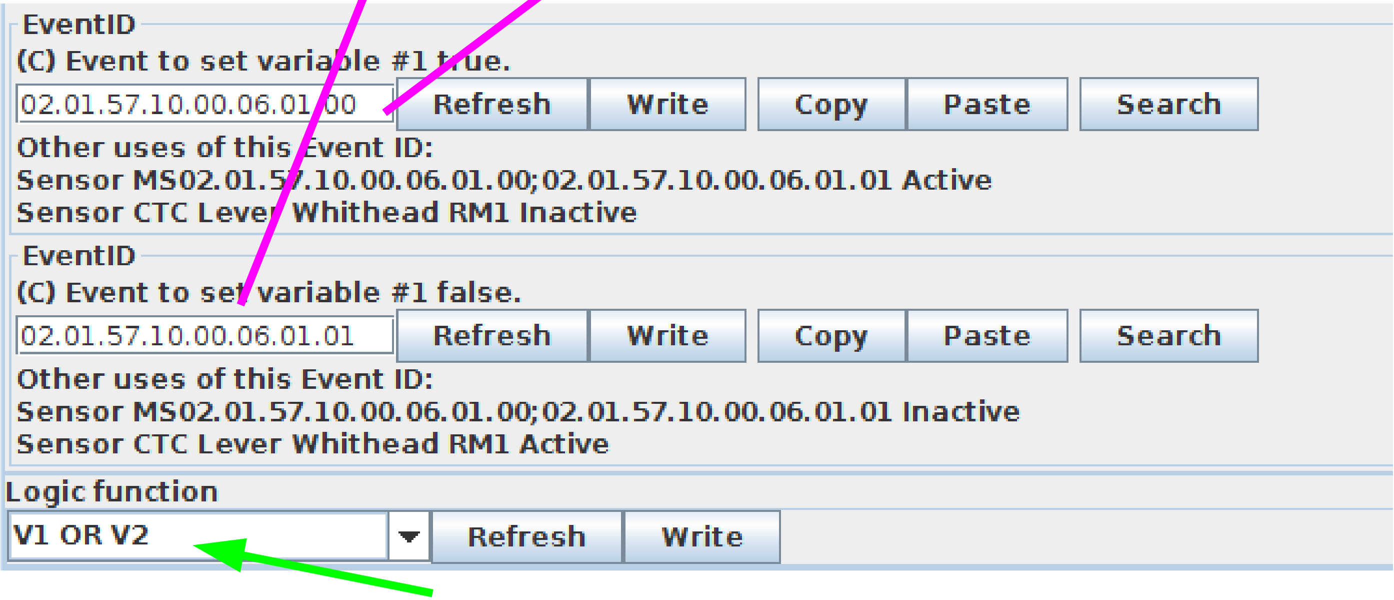


System Name: MS02.01.57.10.00.06.01.01;02.01.57.10.00.06.01.00
User Name: CTC Lever Whithead RM1



System Name: MS02.01.57.10.00.06.00.1E;02.01.57.10.00.06.00.1F
User Name: Whithead OS Main 1

- I actually used the default EventIDs found in variable #1 to create my lever. EventIDs are globally unique so I had no worry about conflicts in meanings.



EventID
(C) Event to set variable #1 true.
02.01.57.10.00.06.01.00
Other uses of this Event ID:
Sensor MS02.01.57.10.00.06.01.00;02.01.57.10.00.06.01.01 Active
Sensor CTC Lever Whithead RM1 Inactive

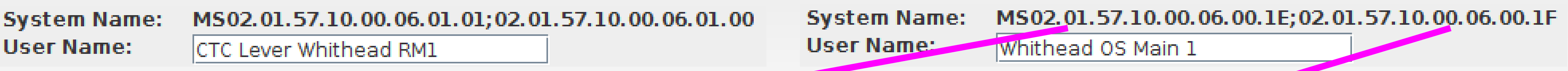
EventID
(C) Event to set variable #1 false.
02.01.57.10.00.06.01.01
Other uses of this Event ID:
Sensor MS02.01.57.10.00.06.01.00;02.01.57.10.00.06.01.01 Inactive
Sensor CTC Lever Whithead RM1 Active

Logic function
V1 OR V2

- Enter the logic function. In this case it is ’OR’.

Logic

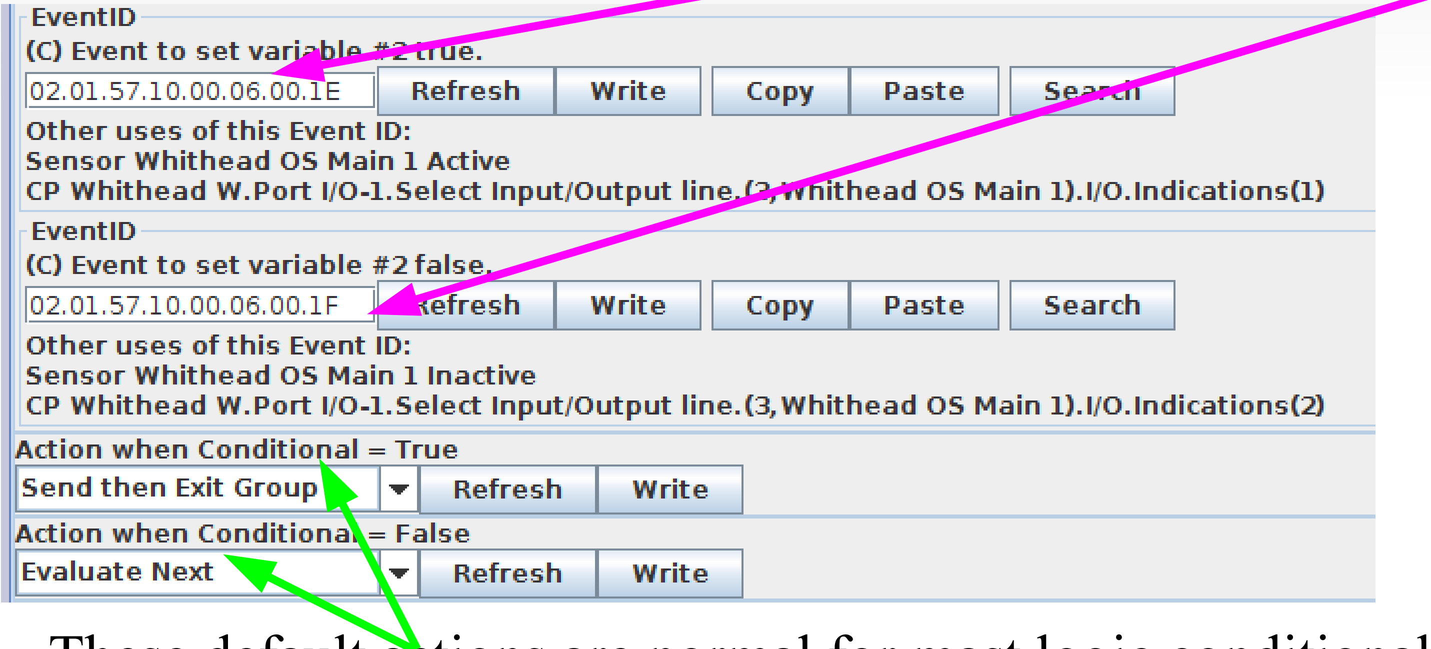
Comments	Variable 1	Funct	Variable 2	This Aspect	This Speed
Not CTC-Right / OS occupied	**Not** CTC-Right	OR	OS BOD	Stop	Stop



System Name: MS02.01.57.10.00.06.01.01;02.01.57.10.00.06.01.00
User Name: CTC Lever Whithead RM1

System Name: MS02.01.57.10.00.06.00.1E;02.01.57.10.00.06.00.1F
User Name: Whithead OS Main 1

- For the block detector I copy/pasted from the I/O line into Variable #2.



EventID
(C) Event to set variable #2 true.
02.01.57.10.00.06.00.1E

Other uses of this Event ID:
Sensor Whithead OS Main 1 Active
CP Whithead W.Port I/O-1.Select Input/Output line.(3,Whithead OS Main 1).I/O.Indications(1)

EventID
(C) Event to set variable #2 false.
02.01.57.10.00.06.00.1F

Other uses of this Event ID:
Sensor Whithead OS Main 1 Inactive
CP Whithead W.Port I/O-1.Select Input/Output line.(3,Whithead OS Main 1).I/O.Indications(2)

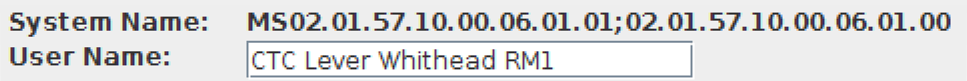
Action when Conditional = True
Send then Exit Group

Action when Conditional = False
Evaluate Next

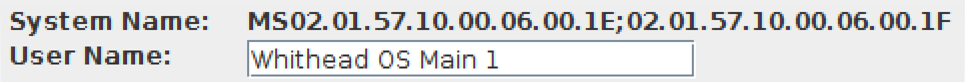
- These default actions are normal for mast logic conditionals. If the condition is true, then any actions are sent, and all less restrictive aspects are skipped.

Logic

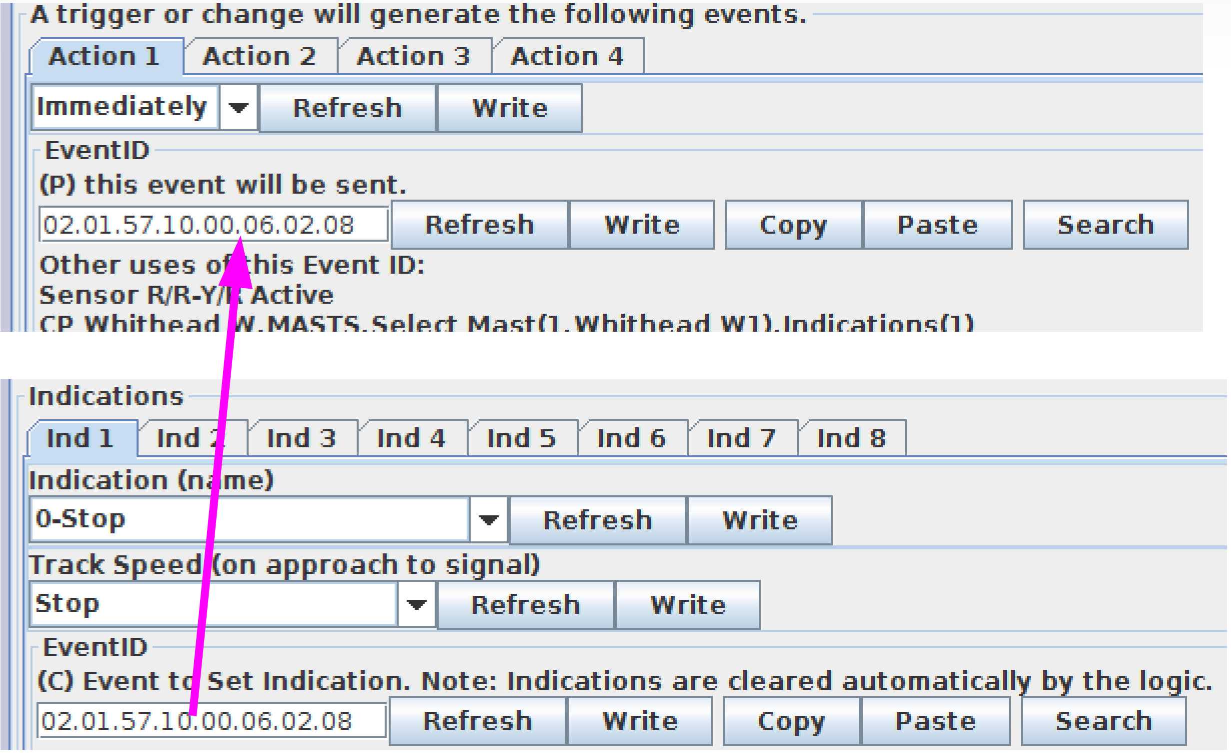
Comments	Variable 1	Funct	Variable 2	This Aspect	This Speed
Not CTC-Right / OS occupied	**Not** CTC-Right	OR	OS BOD	Stop	Stop



System Name: MS02.01.57.10.00.06.01.01;02.01.57.10.00.06.01.00
User Name: CTC Lever Whithead RM1



System Name: MS02.01.57.10.00.06.00.1E;02.01.57.10.00.06.00.1F
User Name: Whithead OS Main 1



A trigger or change will generate the following events.

Immediately

EventID

(P) this event will be sent.

02.01.57.10.00.06.02.08

Other uses of this Event ID:
Sensor R/R-Y/R Active
CP Whithead W.MASTS.Select Mast(1.Whithead W1).Indications(1)

Indications

Indication (name)
0-Stop

Track Speed (on approach to signal)
Stop

EventID

(C) Event to Set Indication. Note: Indications are cleared automatically by the logic.

02.01.57.10.00.06.02.08

- I then copied the event that sets the Signal rule to ’Stop’ into ’Action 1’ of the logic. Therefore anytime the CTC direction lever is not ’Traffic Right’ or if the OS section is occupied, then the signal will be set to ’Stop’.

JMRI Signal Control

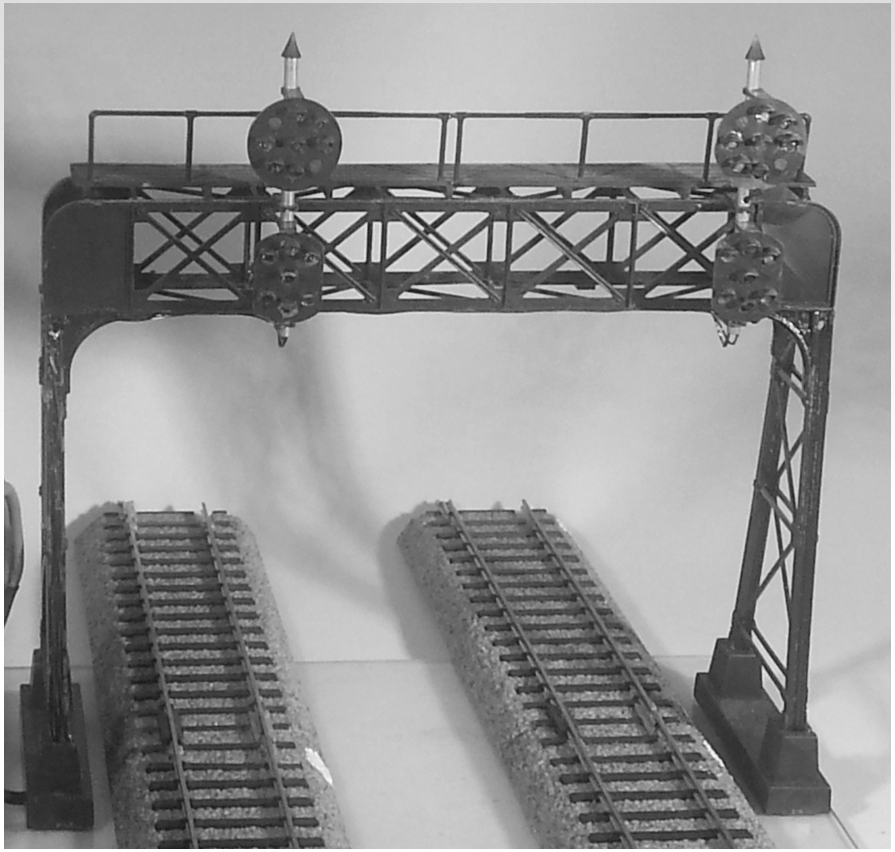
- A recent addition to JMRI is an OpenLCB controlled signal mast. We will show how to use that tool which allows you to take advantage of the built in JMRI signal rules for many popular railroads.
- The disadvantage of JMRI for some users such as museums and clubs is that a computer running JMRI becomes a permanent and necessary part of the signaling system, and usually the signals don’t run while the system is under development. This nullifies some key LCC advantages, which include updates and configuration changes to a running system.

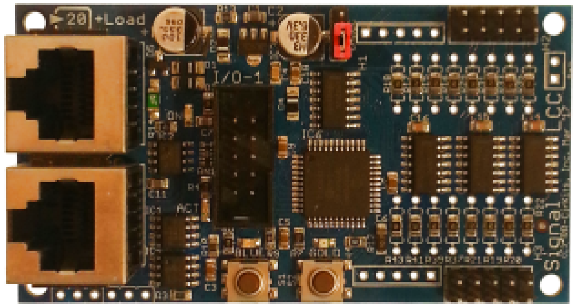
JMRI OlcbSignalMast

- The new JMRI OlcbSignalMast was first added to JMRI 4.11.2 in January 2018. The GUI for this mast was not available until JMRI 4.14.
- This new mast object is a true event driven mast. Previous masts have been based on control by ‘turnouts’ or decoders. These did not play well with external logic options like will be common with LCC.

JMRI OlcbSignalMast

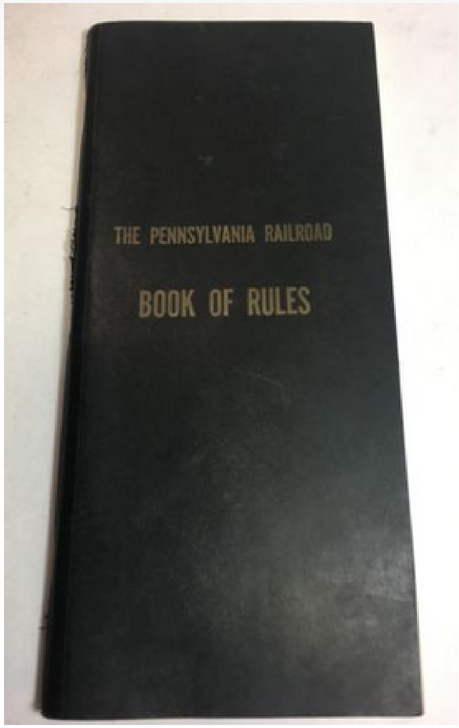
- The first step is to determine what signal system you are using.
- In this example it is a pair of double head PRR signals. These are found in JMRI under the PRR 1957 rules.
- Next determine the control system. To match the clinic title we will use an LCC signal driver node.

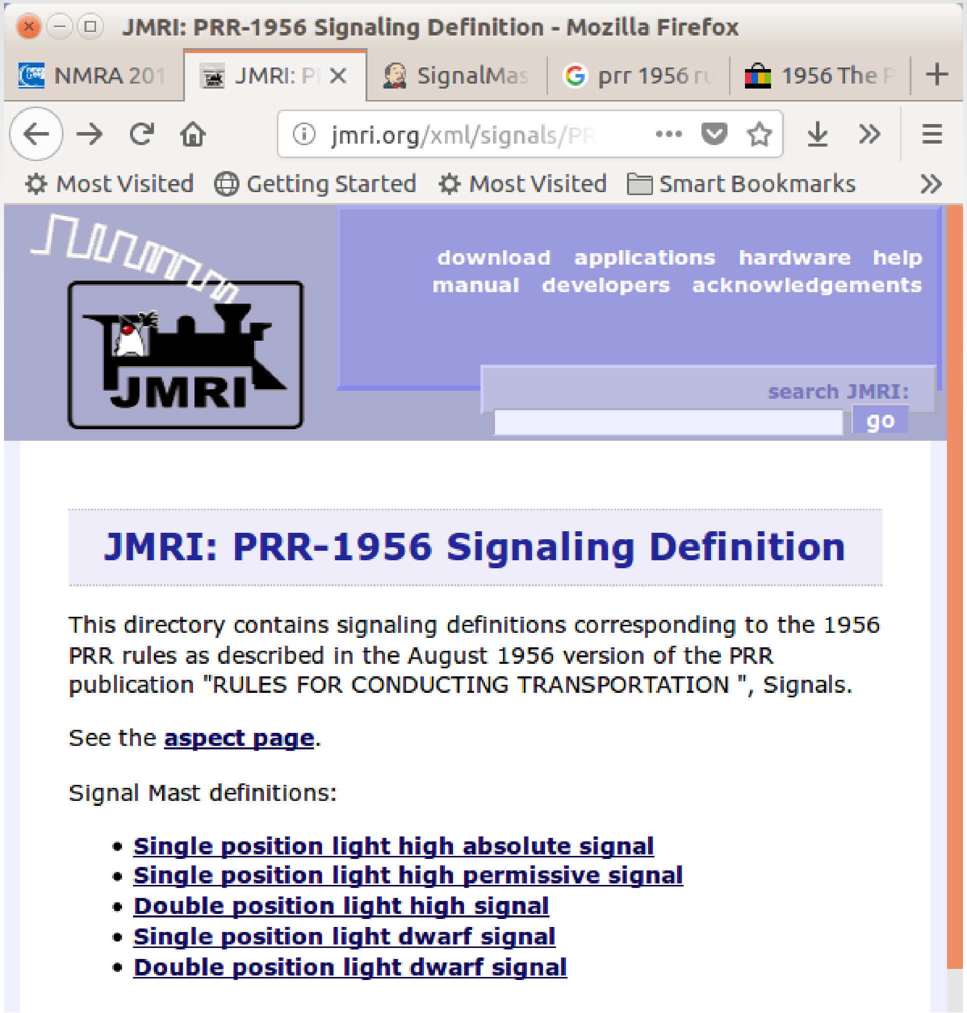




JMRI OlcbSignalMast

- Then
  - a) Locate a rule book.
    - or
  - b) Locate a rule set in JMRI.





JMRI: PRR-1956 Signaling Definition

This directory contains signaling definitions corresponding to the 1956 PRR rules as described in the August 1956 version of the PRR publication "RULES FOR CONDUCTING TRANSPORTATION ", Signals.

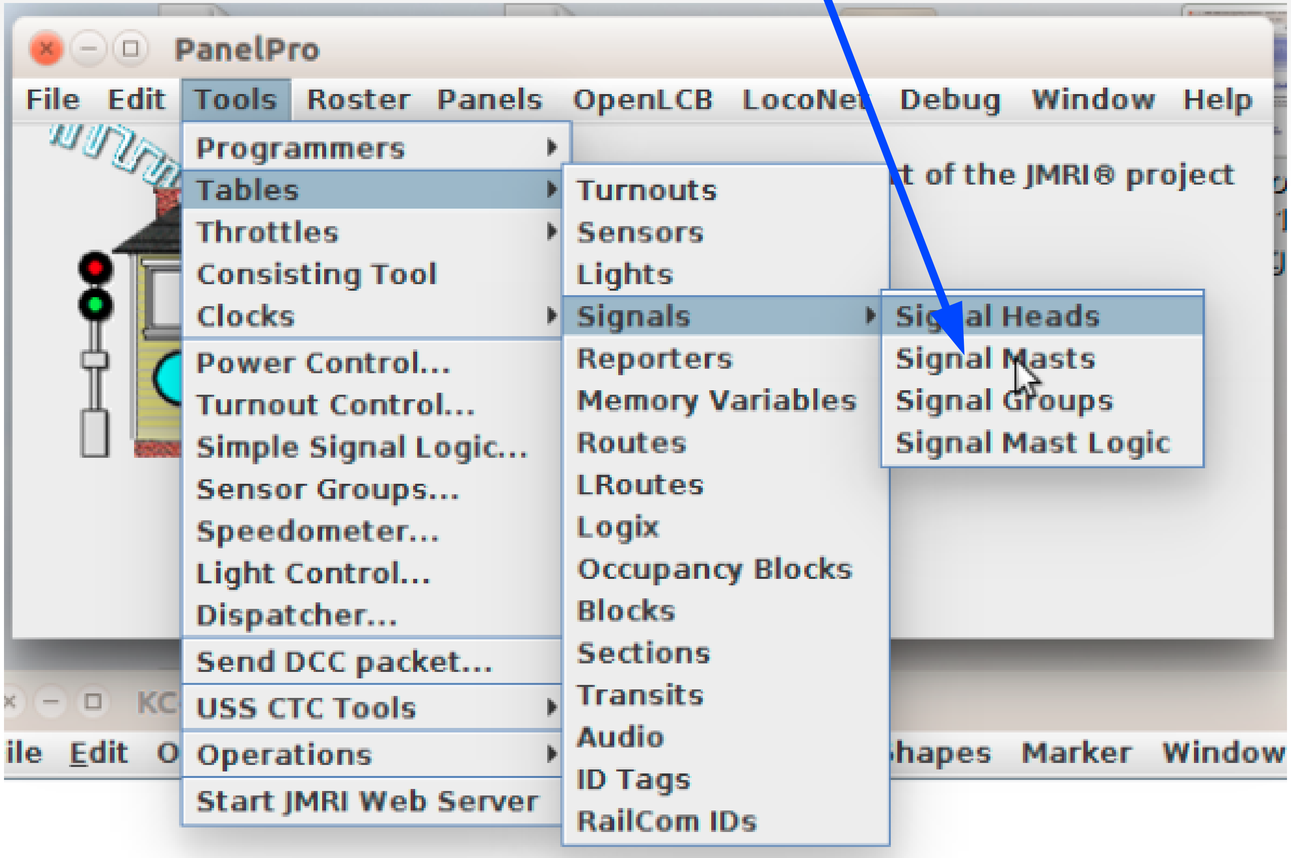
See the **aspect page**.

Signal Mast definitions:

- **Single position light high absolute signal**
- **Single position light high permissive signal**
- **Double position light high signal**
- **Single position light dwarf signal**
- **Double position light dwarf signal**

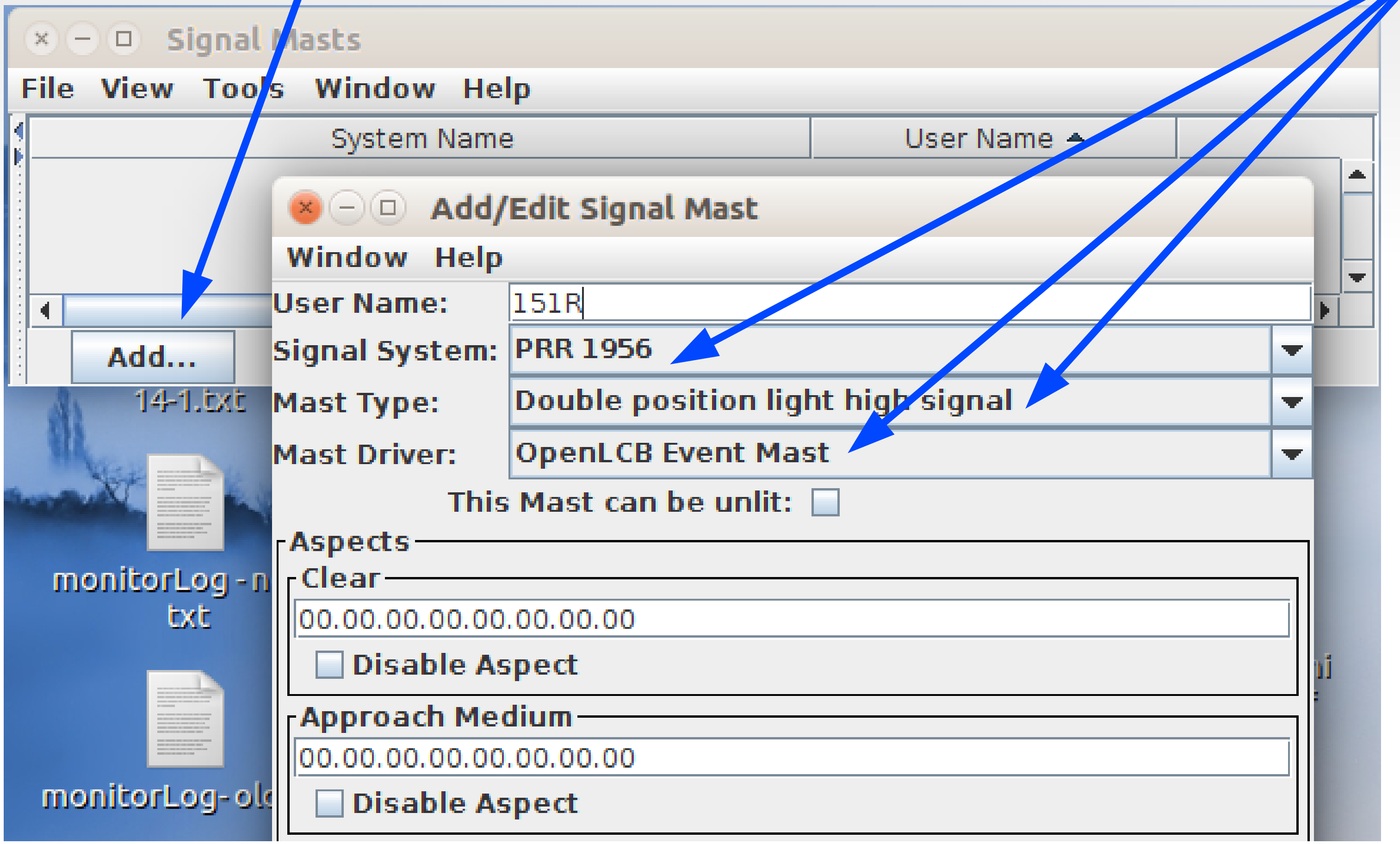
JMRI OlcbSignalMast

- Open the JMRI Signal Masts creation tool.



JMRI OlcbSignalMast

- Click on Add.
- Fill in the required information to select the OLCB mast.

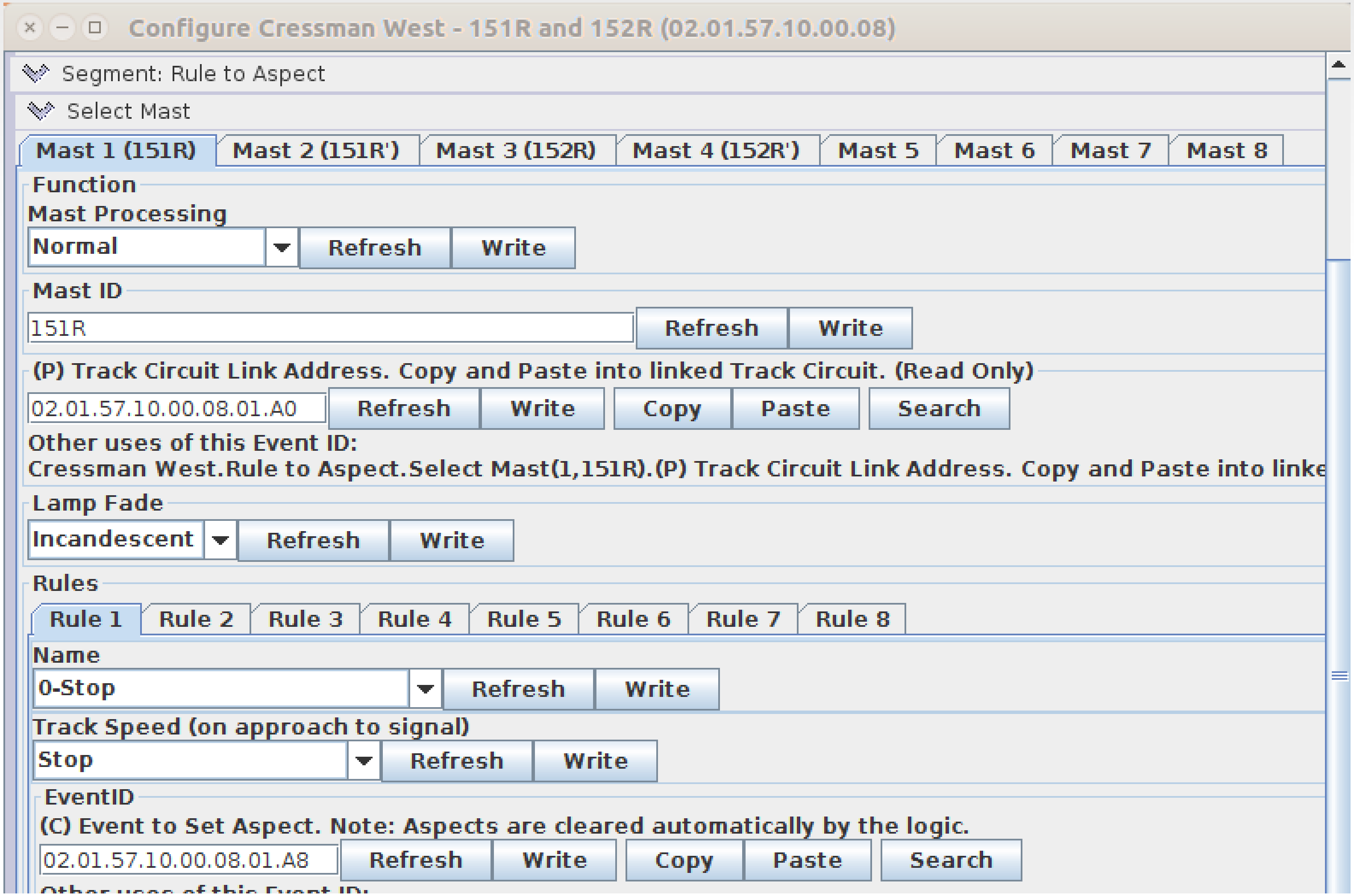


User Name: 151R
Signal System: PRR 1956
Mast Type: Double position light high signal
Mast Driver: OpenLCB Event Mast
This Mast can be unlit: □

Aspects
Clear
00.00.00.00.00.00.00.00
□ Disable Aspect
Approach Medium
00.00.00.00.00.00.00.00
□ Disable Aspect

JMRI OlcbSignalMast

- Open the CDI for the controller node.
- Configure each rule to match the required appearance.



Configure Cressman West - 151R and 152R (02.01.57.10.00.08)

Segment: Rule to Aspect

Select Mast

Function

Mast Processing: Normal

Mast ID: 151R

(P) Track Circuit Link Address. Copy and Paste into linked Track Circuit. (Read Only): 02.01.57.10.00.08.01.A0

Other uses of this Event ID:
Cressman West.Rule to Aspect.Select Mast(1,151R).(P) Track Circuit Link Address. Copy and Paste into linke

Lamp Fade: Incandescent

Rules

Name: 0-Stop

Track Speed (on approach to signal): Stop

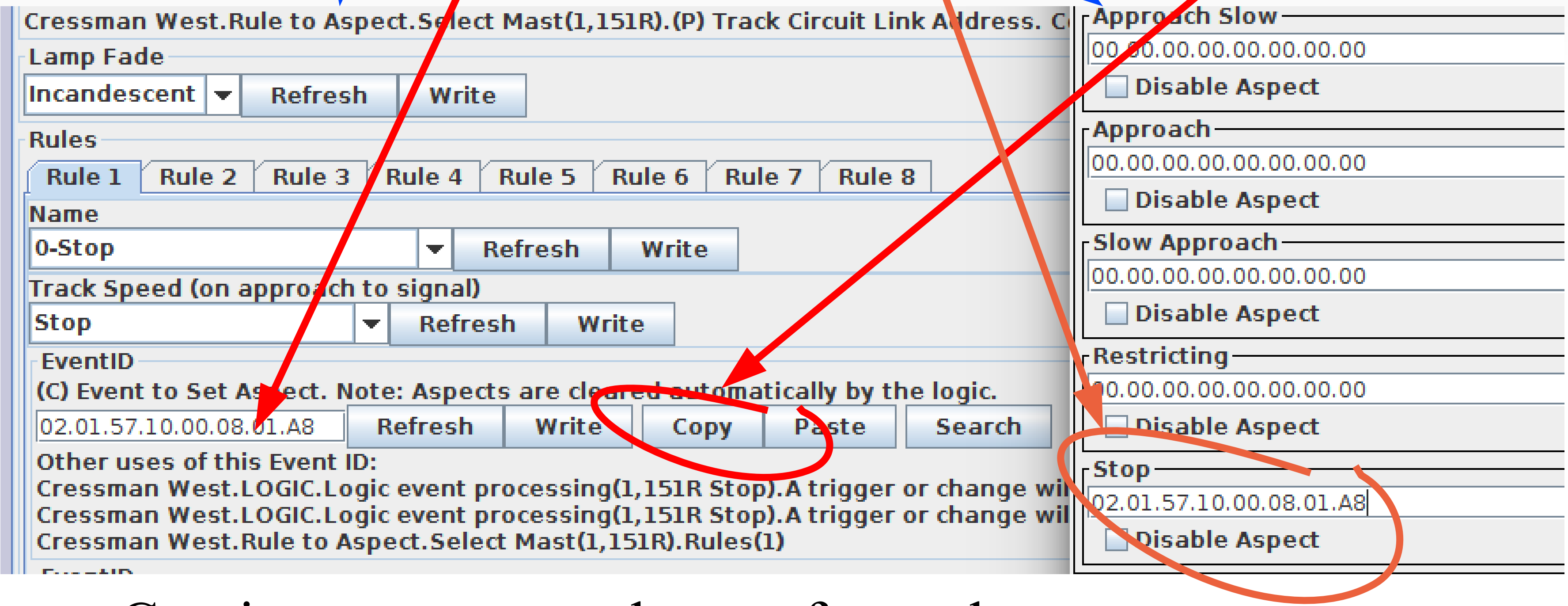
EventID

(C) Event to Set Aspect. Note: Aspects are cleared automatically by the logic.: 02.01.57.10.00.08.01.A8

Other uses of this Event ID:

JMRI OlcbSignalMast

- Place the CDI and mast next to one another on your screen.
- In the CDI at the EventID that sets each aspect, Click [Copy].
- Then double-click to select the aspect event, and Ctrl-V to paste the EventID into the Add Mast Window.



Cressman West.Rule to Aspect.Select Mast(1,151R).(P) Track Circuit Link Address. C

Lamp Fade

Incandescent

Rules

Name

0-Stop

Track Speed (on approach to signal)

Stop

EventID

(C) Event to Set Aspect. Note: Aspects are cleared automatically by the logic.

02.01.57.10.00.08.01.A8

Other uses of this Event ID:

Cressman West.LOGIC.Logic event processing(1,151R Stop).A trigger or change wi

Cressman West.LOGIC.Logic event processing(1,151R Stop).A trigger or change wil

Cressman West.Rule to Aspect.Select Mast(1,151R).Rules(1)

Approach Slow

00.00.00.00.00.00.00.00

□ Disable Aspect

Approach

00.00.00.00.00.00.00.00

□ Disable Aspect

Slow Approach

00.00.00.00.00.00.00.00

□ Disable Aspect

Restricting

00.00.00.00.00.00.00.00

□ Disable Aspect

Stop

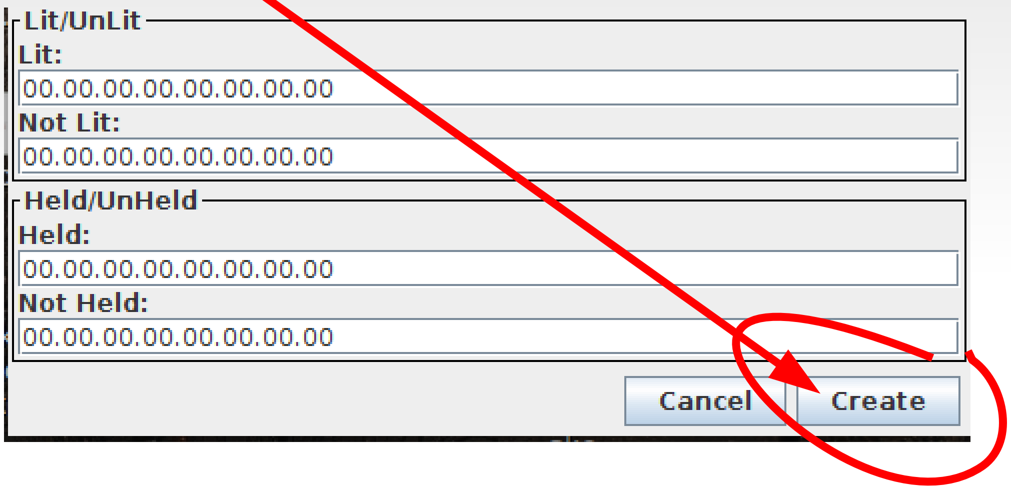
02.01.57.10.00.08.01.A8

□ Disable Aspect

Continue to copy and paste for each aspect.

JMRI OlcbSignalMast

- Once each EventID has been added to the mast click on [Create] to enter it into the Mast Table.



Lit/UnLit

Lit: 00.00.00.00.00.00.00.00

Not Lit: 00.00.00.00.00.00.00.00

Held/UnHeld

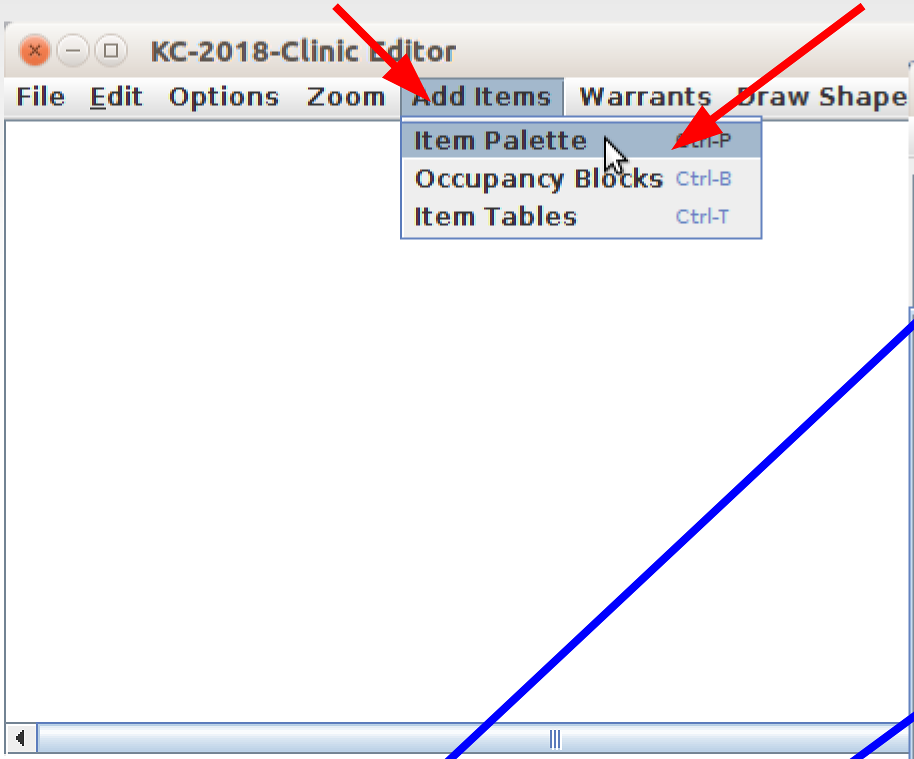
Held: 00.00.00.00.00.00.00.00

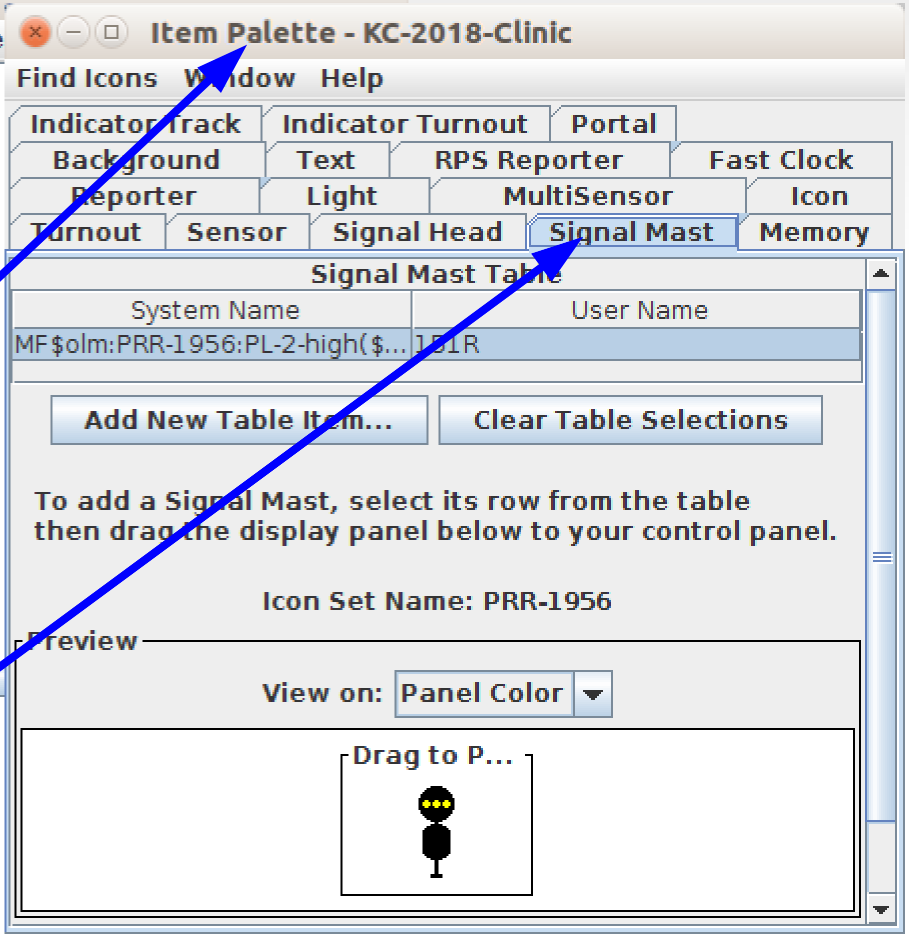
Not Held: 00.00.00.00.00.00.00.00

- With LCC as soon as you click [Create] the mast appears in the Mast Table and is live on the layout.
- Be sure to save your panel before closing down JMRI. Your LCC mast will retain its information, but JMRI will not if you fail to save it.

JMRI Panel

- Now open a Panel in JMRI.
- Select ’Add Items’ then ’Item Palette’.





Signal Mast Table

System Name	User Name
MF$olm:PRR-1956:PL-2-high($...	151R

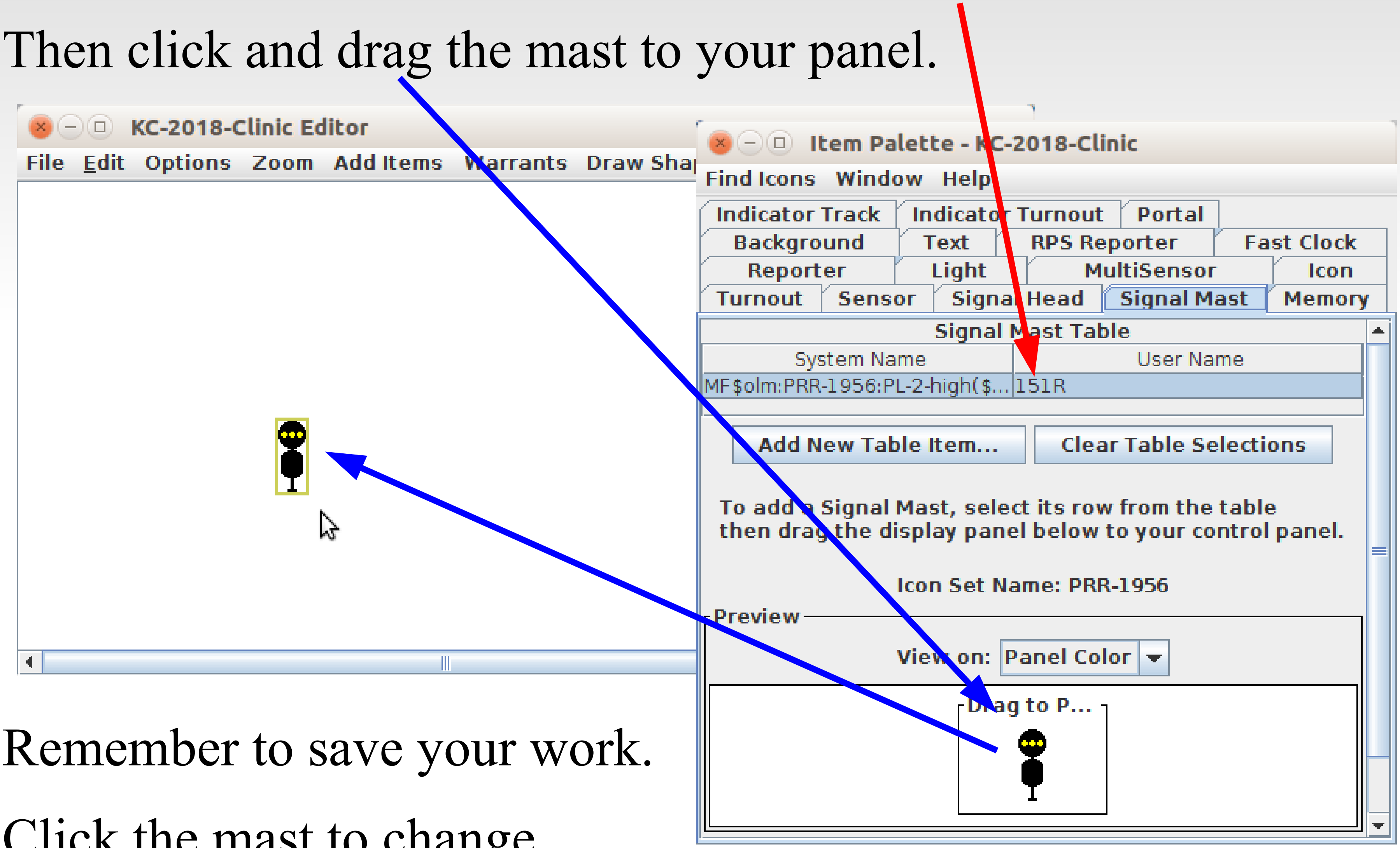
To add a Signal Mast, select its row from the table then drag the display panel below to your control panel.

Icon Set Name: PRR-1956

- An Item Palette window opens.
- Select Signal Mast.

JMRI Panel

- Select the new mast from the Signal Mast Table.
- Then click and drag the mast to your panel.



KC-2018-Clinic Editor

Item Palette - KC-2018-Clinic

Signal Mast Table

System Name	User Name
MF$olm:PRR-1956:PL-2-high($...	151R

To add a Signal Mast, select its row from the table then drag the display panel below to your control panel.

Icon Set Name: PRR-1956

- Remember to save your work.
- Click the mast to change.

JMRI Panel

- The new OpenLCB signal mast object in JMRI has the important feature of being an equal node with the actual mast on your layout. If JMRI logic changes the aspect it changes on both the panel and on the layout. That much is the same as other signals.
- Unlike some other signal types, external logic, be it another copy of JMRI, or internal logic in the signals themselves, may also be used to change both the layout and panel signals. This is a very important change from other signal control options.

Other Layout Animation

- Signaling is normally the most complex animation applied to a model railroad layout.
- Crossing gates and flashers with or without sound is another closely related animation that is often attempted by modelers. Commercial gate animators have various levels of sophistication, from simple on – off, control to reasonably accurate operation. I have seen designers twist themselves into knots trying to figure out how to do it accurately in both directions. However if you think in terms of Events it is actually very simple. Define two blocks. The first covers the entire gate A*pproach* area. The second covers just the highway portion. We call it the I*sland*.

  The Logic:
  1. Approach clear AND Island clear = gates up This requires memory of the two events plus AND logic, or else running both the island and approach feeders through the 'approach' block coil.
  2. Approach occupied event = gates down
  3. Island occupied event = gates down
  4. Island clear event = gates up

- 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 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.
- For those that have visited the LCC tables in the SIG room you may have noticed the Logic Rail LCC fast clock unit on display there. That of course is an important key to accomplishing the above.

- Signaling is way too complex a subject to cover in much further 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

Key OpenLCB Contributors: Bob Jacobsen, Alex Shepherd, David Harris, Stuart Baker, Balazs Racz, Jim Kueneman, Don Goodman-Wilson, John Plocher

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 are supporting JMRI

User Group

Started November 2009

- July 2019 we have over 290 members

NMRA liaison: Brian Barnt

- NMRA w.g. chairman: Karl Kobel

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

http://nmra.org, choose S&RP scroll to 9.7

Book: Introduction to Layout Command Control (Amazon.com)
by Dana Zimmereli PhD

Questions

- ?