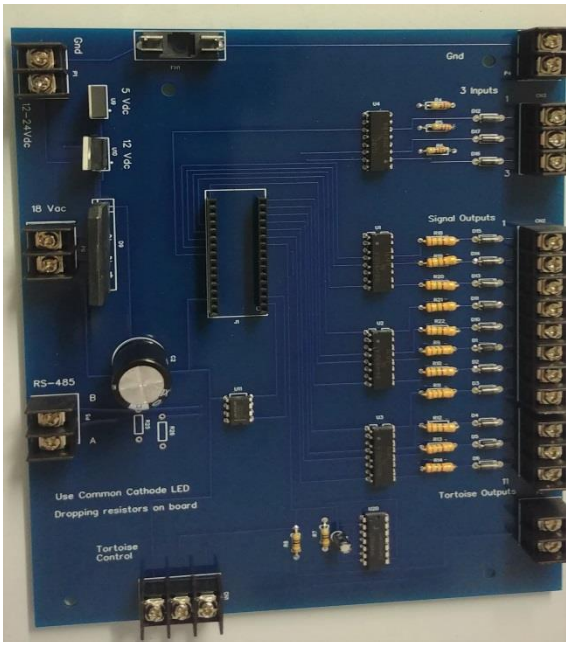
On-Station Signal Siding Card

This card is designed with signal, block detection and turnout control for a siding entrance/exit in mind. It contains control for one Tortoise motor, one block detector and three multiple head signal masts.



Along the left edge the top terminal strip is for 12 to 24 VDC and the middle terminal strip is for 18 VAC. These are the two types of supply power the board can use. **CAUTION: You must choose one of these to supply power, never connect both types of power at the same time.**

Just above these terminal strips (shown horizontally) is the 1.6 Amp fuse. This fuse protects the field devices from drawing excessive current and for short circuits.

The bottom terminal strip (still left edge) is the RS-485 connection. The Arduino is programmed to use this port as a C/MRI node (Computer / Model Railroad Interface) and will connect directly to JMRI (Java Model Railroad Interface) as such.

The top right edge terminal strip is the Signal Ground connection. This board is setup to support common cathode LED connections. Both terminals are GRN and no VCC is required. This is the terminal to be sent out to your device and feed back to an INPUT terminal. More details to follow.

Drop down (right edge) to the next terminal block and you'll find three inputs. The Arduino logic expects two of these inputs to provide status for the turnout position and one for the turnout "occupied" block detector. More details to follow.

The next two terminal strips (middle right) are considered as a block (8 + 3 for a total of 11 outputs) and are used to control the signal aspects. Three multiple head signal masts and one single head dwarf signal are supported as follows.

- 3 + 2 = a three LED head over a two LED head
- 3 + 1 = a three LED head over a one LED head
- 2 = a single head with 2 LEDs

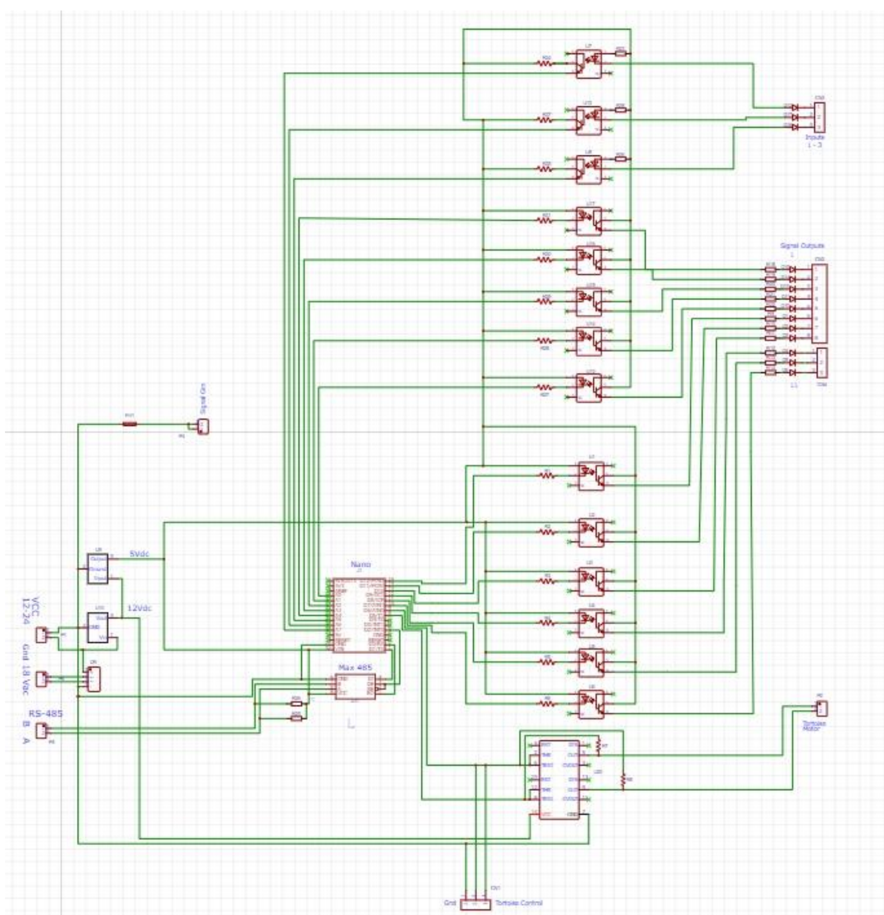
They are also feed from the Signal Ground terminal strip. More details to follow.

The bottom right edge terminal strip and the bottom right edge terminal strip are used for Tortoise motor control. The bottom right edge terminals are connected to the Tortoise motor leads (#1 and #8) and are used to position the points for the proper direction of travel. The bottom edge (right) terminal strip is used for manual control of the turnout points. A GND (pin 3) is provided to be connected as the common connection to the momentary 2 position single pole switch. More details to follow.

The bottom board view has a single point of interest, the “cut here” Label. Silk screened to the bottom of the board is a label called “cut here” and it is used in conjunction with how many cards you are using. If you are using a single card on the RS-485 connection leave the card as delivered, if however you use more that one card you will need to cut this PCB trace on all but the first card. Details to follow.

Schematic Overview:

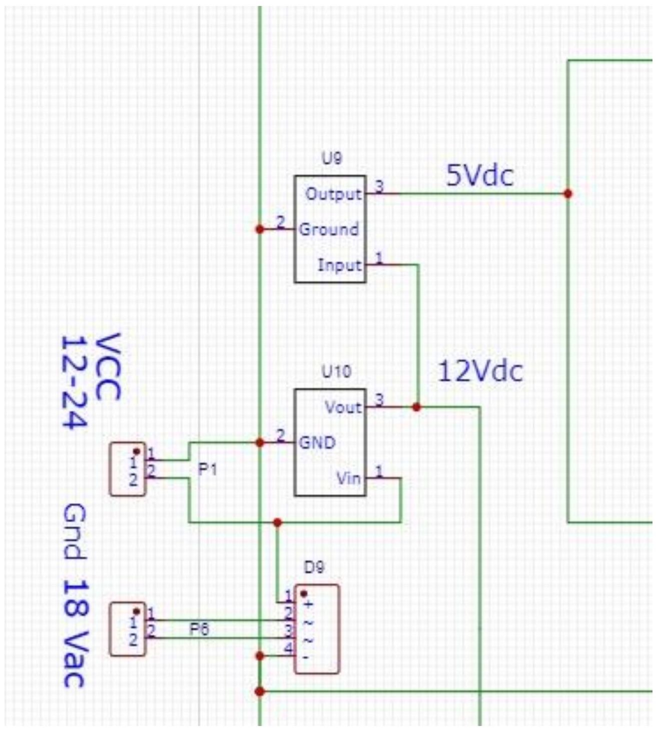
Please don’t be alarmed by the following schematic, mastery is not necessary to use the C/MRI siding card. It is included for a more in-depth understanding of how the card works and will allow advanced users to adapt the card for other purposes.



Blow up the PDF page for more clarity

Power sources:

The board may be run from either a 12-24 VDC or an 18 VAC power source (not both.)



18 VAC – Is applied to terminal strip P6 and connected to a full bridge rectifier D9 (terminals 2 & 3) where it is converted to approximately 25 VDC. The output of the rectifier is applied to a regulator (U10 pin 1) which limits the output (pin 3) to 12 VDC.

The 12 VDC is used in two places:

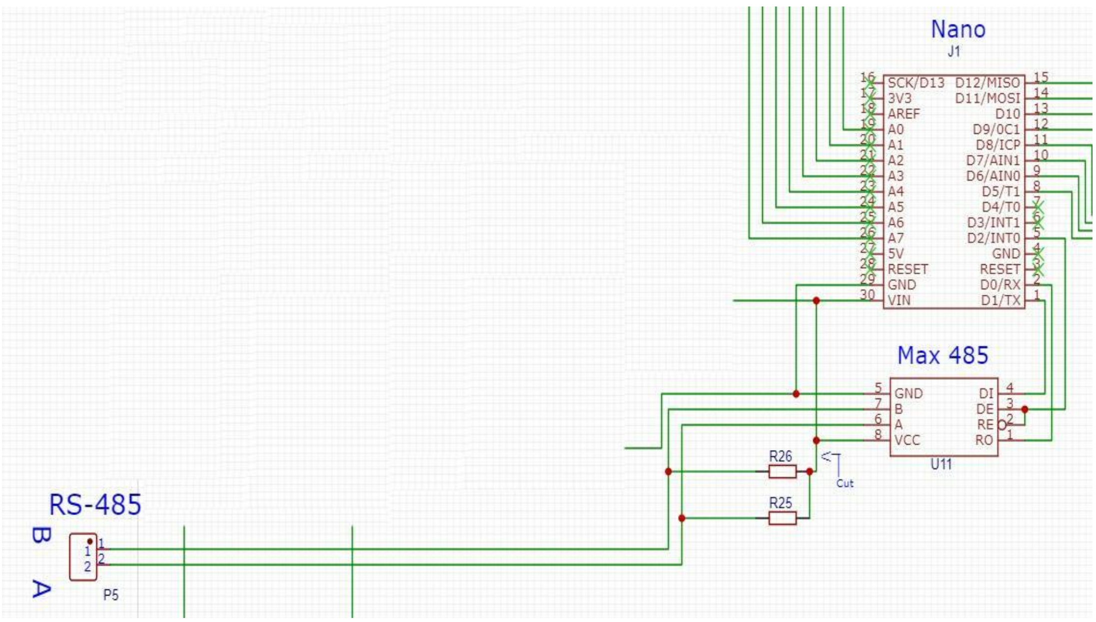
1. Power to control the Tortoise motor
2. The input to U9 (pin 1), the 5 VDC regulator where the output (pin 3) is used for signal LEDs and internal board logic.
3. NOTE: D9-4, U10-2 and U9-2 form a common ground.

12-24 VCC – Is applied to terminal strip P1 and connected to the 12 VDC regulator (U10) which in-turn feeds the 5VDC regulator (U9)

CAUTION: Never connect both 18VAC and 12-24VDC to the same board at-the-same-time as this will damage the rectifier (D9) and 12VDC regulator (U10) and possibly the externally connected power sources.

RS-485 Connection:

This is the connection that allows the board to communicate with and external control device, such as a computer running JMRI (Java Model Railroad Interface.) The Arduino on this card uses libraries to make it communicate as a C/MRI card (Computer / Model Railroad Interface.)

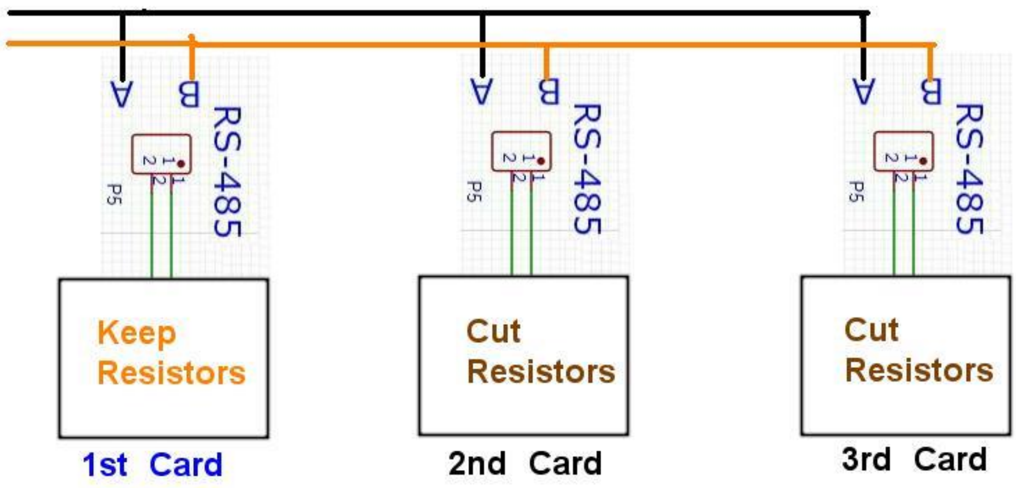


The computer connects at the terminal strip labeled RS-485. The computer will probably have a USB-to-RS-485 adapter. Make sure to follow the “A” and “B” wiring so that “A” goes to “A” and “B” goes to “B”.

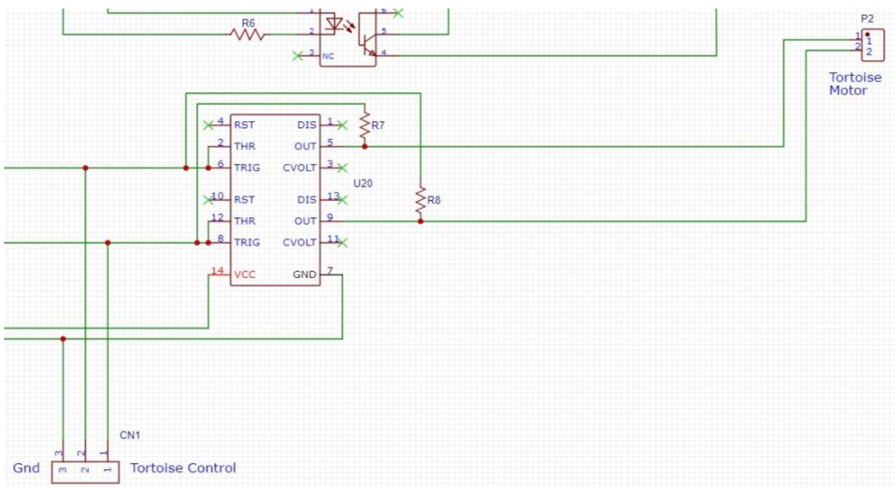
The communications bus needs what are called pull-up resistors to ensure that no noise (unwanted + to - transitions) are seen by the Max 485 chip. Here they are identified as R26 and R25 but the numbers may vary depending on which card you are using. The important thing to know is that the “bus” only needs one set of resistors. So we want to keep the resistors on the first board and disable them on all additional boards. Each card will have a silk trace “Cut” or “Cut Here” point. (Depending on the card this could be labeled on the top-side or the bottom-side of the card.)

If you only have one card on the bus you are done – do nothing with the resistors. If you have two or more cards on the bus then you must,

1. Leave the resistors on the first board untouched (do not cut)
2. On each additional card you must disable these resistors by cutting the copper trace at the point labeled “Cut” or “Cut Here.” An X-Acto knife works well for this and the gap only needs to be wide enough to see that the trace has been cut.

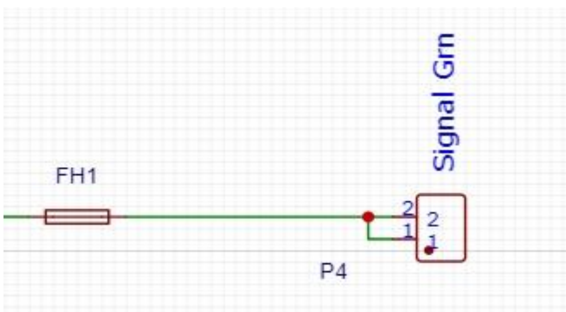


Tortoise Control:



The board will control one Tortoise type turnout motor. Connect Tortoise terminals (1) and (8) to card terminals (1) and (2) of connector P2. Power for the Tortoise comes from the board.
The Diverging/Mainline flip flop action is created by the IC chip. The Nano chip (not shown) sends commands to the IC chip pins (6) or (8) depending on which turnout setting has been selected. The board also allows manual control of the Tortoise using connections from terminal block CN1 to a single pole / double throw (SPDT) **momentary** switch. Terminal (3 GND) is connected to the common connection on the switch. Terminals (1) and (2) will determine which position the points align too. The momentary switch does not have to be held for the entire travel time of the motor, a quick pulse of the switch will be captured by the IC chip and will finish the motor movement.

Signal Ground:



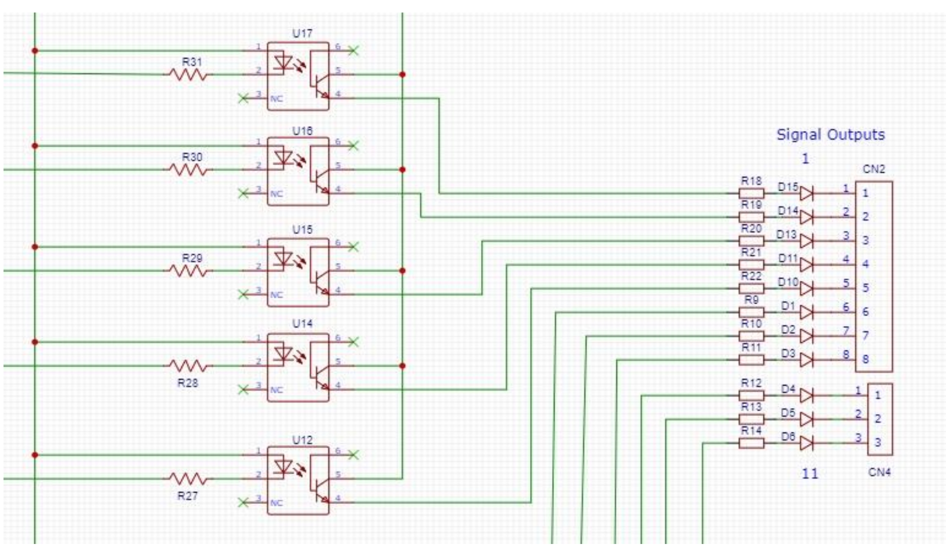
The Signal GRN terminal strip (P4) terminals (1) and (2) are jumpered together on the board. Use these terminals to provide the connection to the common side of the signal LEDs. The board is designed to work with only Common Cathode LED signals. (More details under Signal Outputs section.)

You can also use these terminals to connect the common side of the three input devices.

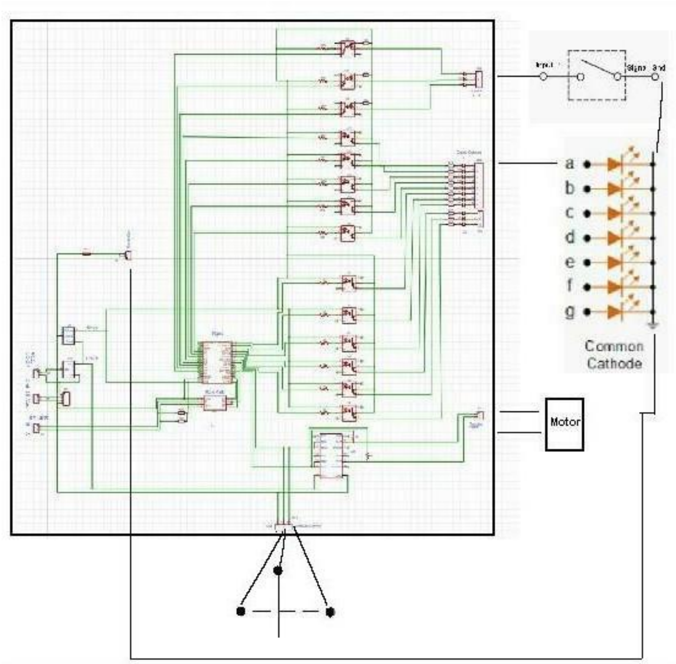
The board comes with a 1.6 Amp fuse but it is safe to use a more common 2 Amp fuse. If this fuse blows then most likely a field device (signal LED or block detector) has been shorted to 5VDC. Nothing on the card will blow this fuse.

Signal Outputs:

There are eleven outputs on the board, eight on terminal strip (CN2) and three on terminal strip (CN4) located right next to each other. The board is designed to work with LEDs (2 Volt (fw) with 20 mA (fw)) and has 150 Ohm resistors built it to reduce the 5 VDC and limit the current.



For signal labeling the two Terminal blocks are treated as one with outputs 1-8 on CN2 and outputs 9-11 on CN4. Examining Output 1 we see D15, these diodes are to protect the circuit from attaching the wrong polarity voltage on the terminal. R18 is the built-in dropping resistor for the LED. IC chip U17 provides galvanic isolation of the output circuit from the rest of the board components. Pin (5) of U17 is connected to 5VDC to source the output when U17 is turned on by the Arduino command on pin (1).



This is an example of a typical OS siding card signal and tortoise motor connection.
Note how the board “Field Ground” connection is on the common side for all connections.

Output Assignments:

While the board output assignments can be assigned many configurations (via JMRI) a suggested standard is show here:

Two Headed Mast: Use outputs 1 thru 5

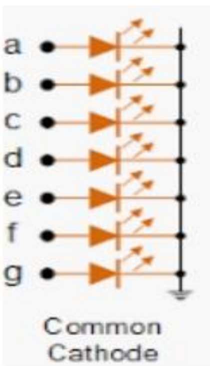
Top head:
Output 1 – Green
Output 2 – Yellow
Output 3 – Red
Bottom head:
Output 4 – Green
Output 5 – Red

Two Headed Mast: Use outputs 6 thru 9

Top head:
Output 6 – Green
Output 7 – Yellow
Output 8 – Red
Bottom head:
Output 9 – Red

Dwarf Mast:
Two Lens:
Output 10 – Green
Output 11 – Red

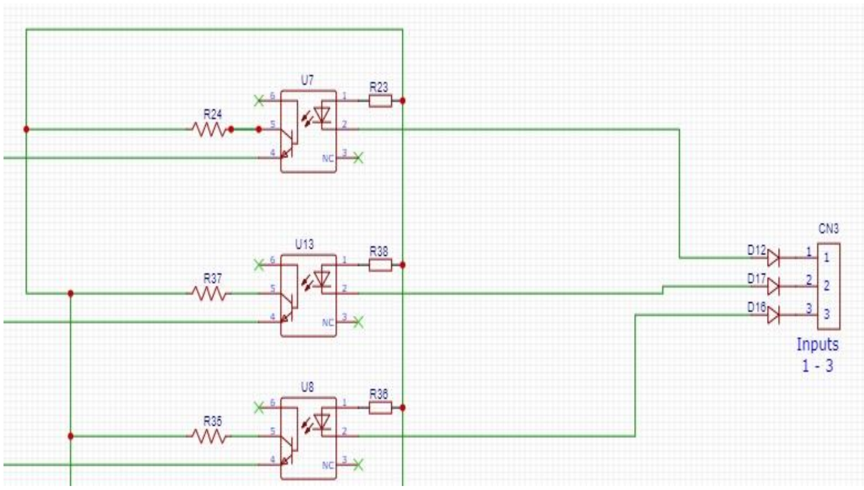
These assignments are not mandatory, but are recommended as they make it very easy to integrate the board with JMRI (Java Model Railroad Interface) signaling features.

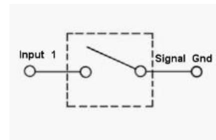


The outputs are designed to support Common Cathode LEDs. The ground symbol in this picture would be connected to the board SIGNAL GRN terminal and points A thru G would each go to a separate output (A = output 1 ….. G = output 7)

Inputs 1-3:

There are three inputs on the board on terminal block CN3. Input 1 is on terminal (1) 2 on (2) and 3 on (3.) Examining Input 1 we see D12, this is a reverse polarity protection diode. D12 is connected to U7 pin (2) this chip provides galvanic isolation for the Nano input terminals. Resistor R23 on pin (1) is a current limiting resistor. Input 1 is active when CN3 pin(1) is connected to Signal GND (active low.)



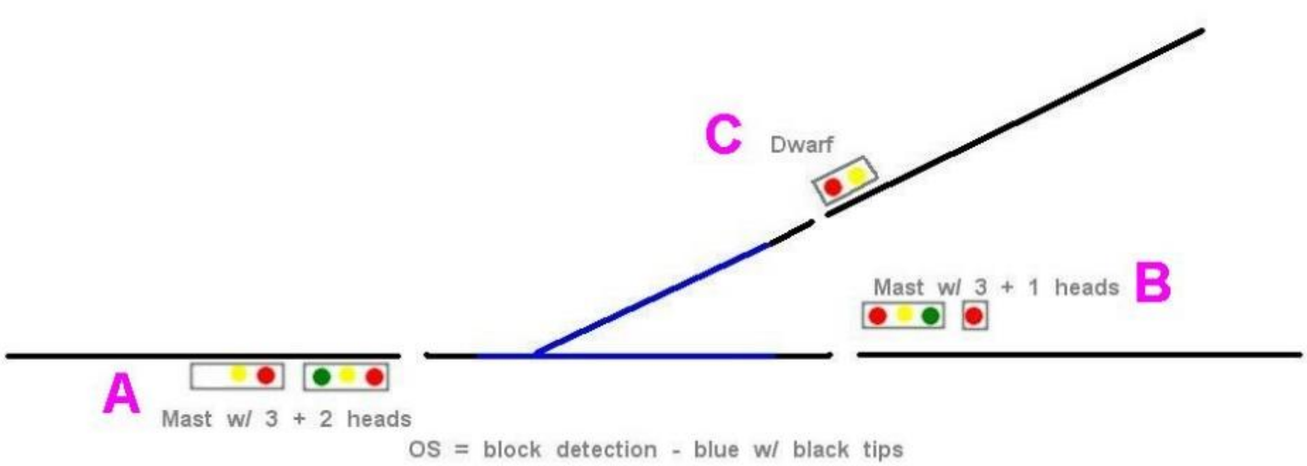


Input Assignments:

While the board input assignments can be assigned many configurations (via JMRI) a suggested standard is show here:

Input 1 – Tortoise Mainline feedback
Input 2 – Tortoise Diverging feedback
Input 3 – OS (On Station) block detector feedback

Siding Entrance Example:



Railroads use a variety of signal arrangements so check your rule book for your railroad. This is an example of a typical “fully loaded” signal layout for model railroading using speed signaling. Note that the OS block detection “gaps” extend far enough past the turnout (blue) to ensure that no rolling stock is close enough to foul the turnout. Using the block detection on a turnout allows logic to “lock” the turnout when it is occupied as well as reporting train location.

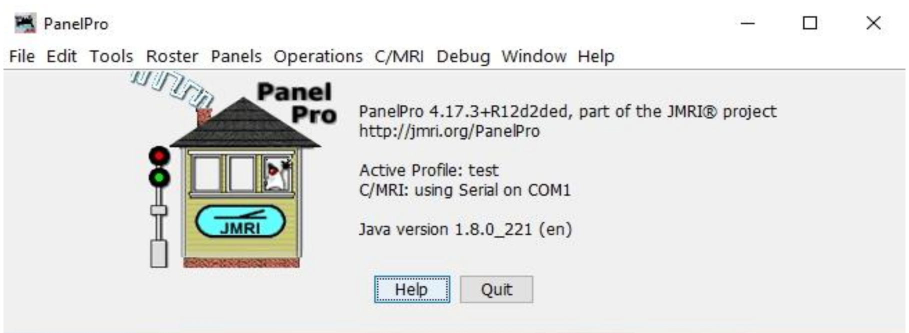
The flexibility of the board will allow many variations of signals but that is beyond the scope of this manual.

NOTE: The magenta A, B and C labels are used to help explain how to name the signals in the JMRI table.

Configuring JMRI to use the board as C/MRI hardware:

The on-board Arduino chip is programmed to act as a C/MRI SMINI card with the default address of zero (0). Though a true SMINI card can address 24 inputs and 48 outputs the Nano chip cannot handle that much I/O. This Siding Signal board can only handle a total of 16 I/O points, so all SMINI node I/O points beyond sixteen (16) will be ignored by the Nano chip.

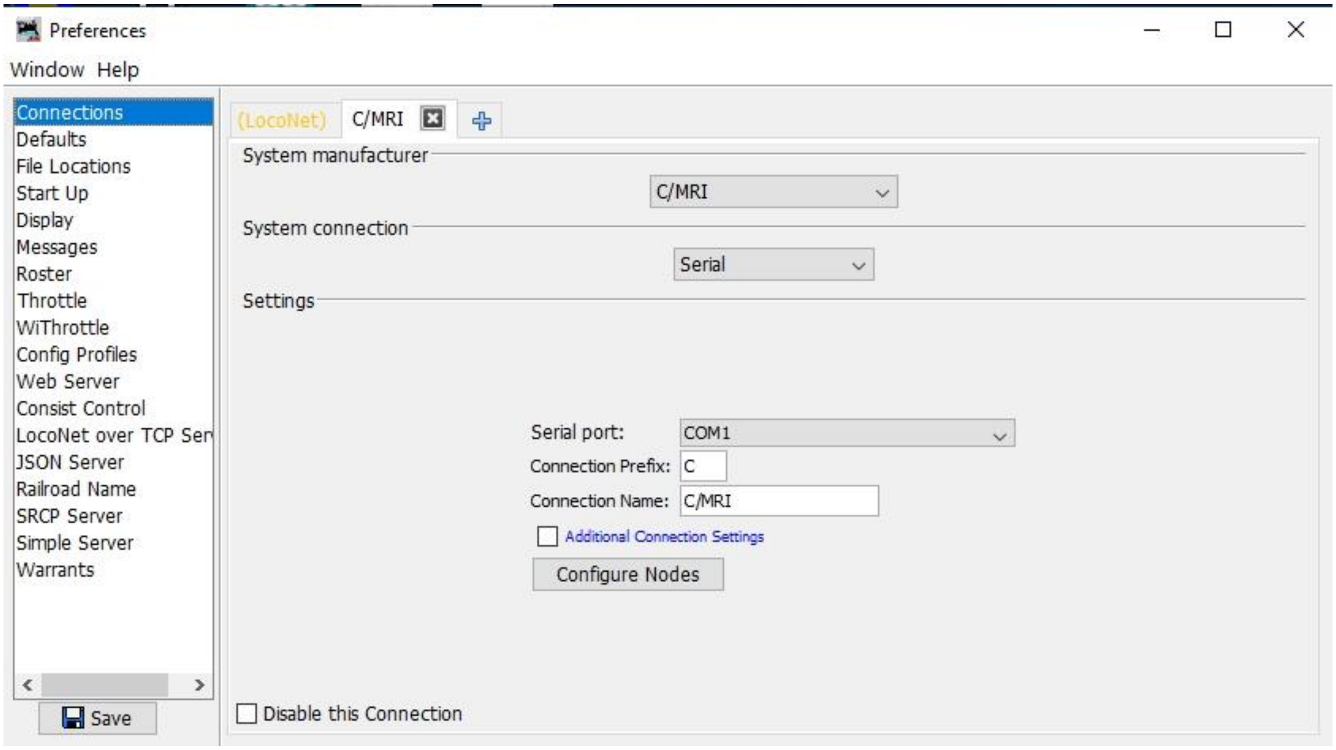
This overview of how to program a C/MRI port is not a comprehensive guide to JMRI but it should be enough information to get you started.



Note that this manual is written using PanelPro version 4.17.3 Some screens may vary depending on your version of PanelPro.

When starting this example your splash screen will not show the C/MRI on COM1. However as we proceed you will be instructed that PanelPro must be restarted and that is when you'll see the C/MRI listing. Note the menu at the top of this splash screen as all references to menu selections will start from this window.

Go to **Edit/Preferences** on this menu to get the following pop-up.



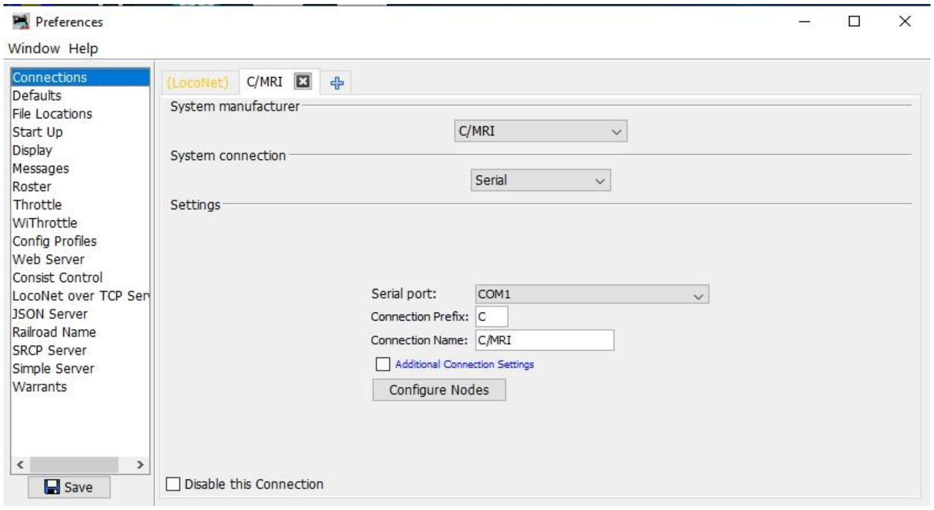
Now select connections from the side-bar menu.
Your window will now have at least two tabs, one is usually named for the DCC system you are using and the Other has a big “plus” sign. Click the “plus” sign to add a new connection for JMRI to use.
From the drop-down list boxes select the following:

- System Manufacturer – C/MRI
- System Connection – Serial
- Serial Port – COM1 (this may vary depending on how many COM ports your computer has used.

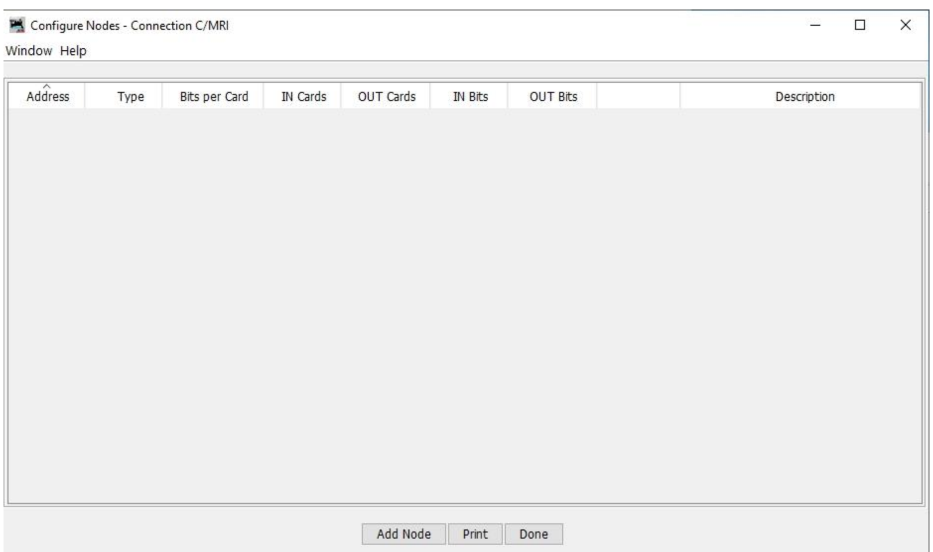
Leave the connection prefix with the default “C.”
The Connection Name can be anything you want it to be, I suggest C/MRI as this is the name that will show up on the tab and in other menu choices making it easy to remember what hardware you are talking too.
Additional Connection Settings: set the baud rate to 9600

Once you fill in the blanks you must click the SAVE button. A pop-up we tell you that PanelPro must be restarted for the changes to take affect. Answer YES and now you’ll see the splash screen as shown above.

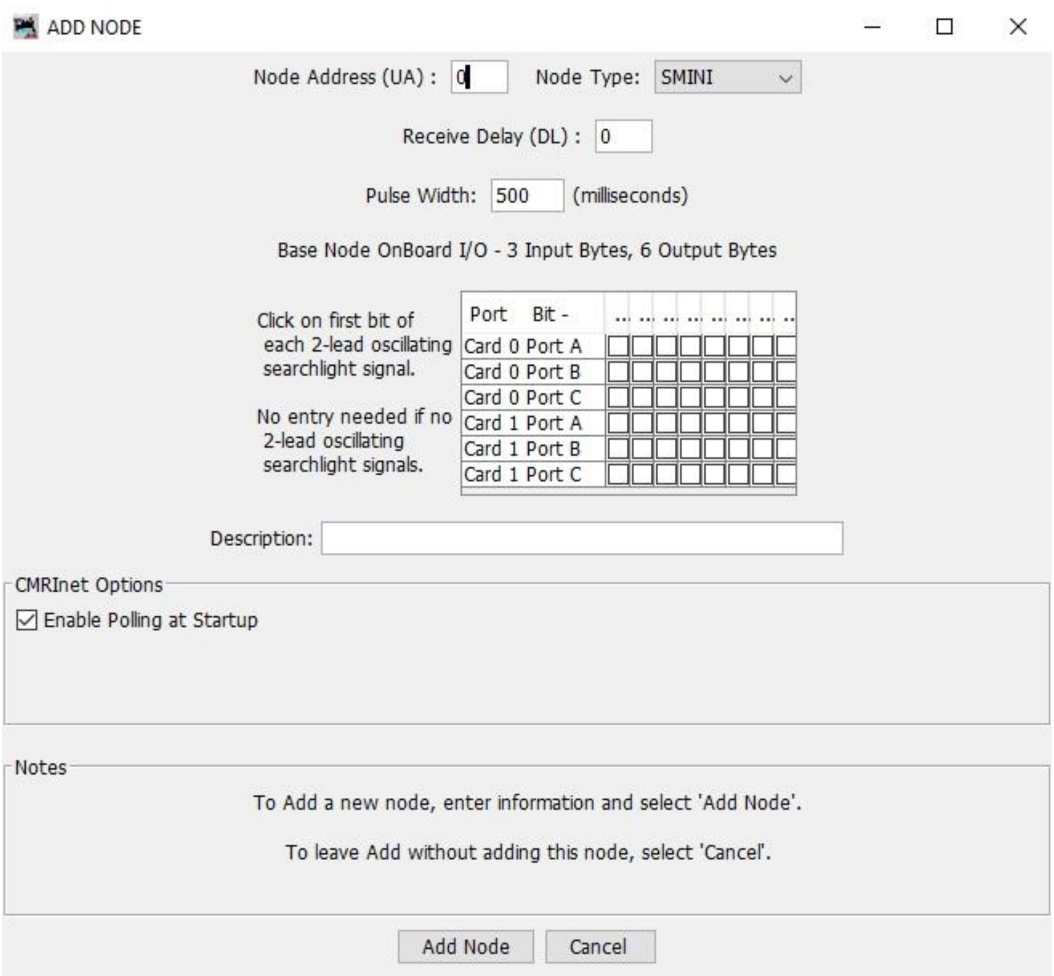
From the splash screen menu go to EDITS/PREFERENCES



Once again select Connections from the side-bar menu. Click the “Configure Nodes” button



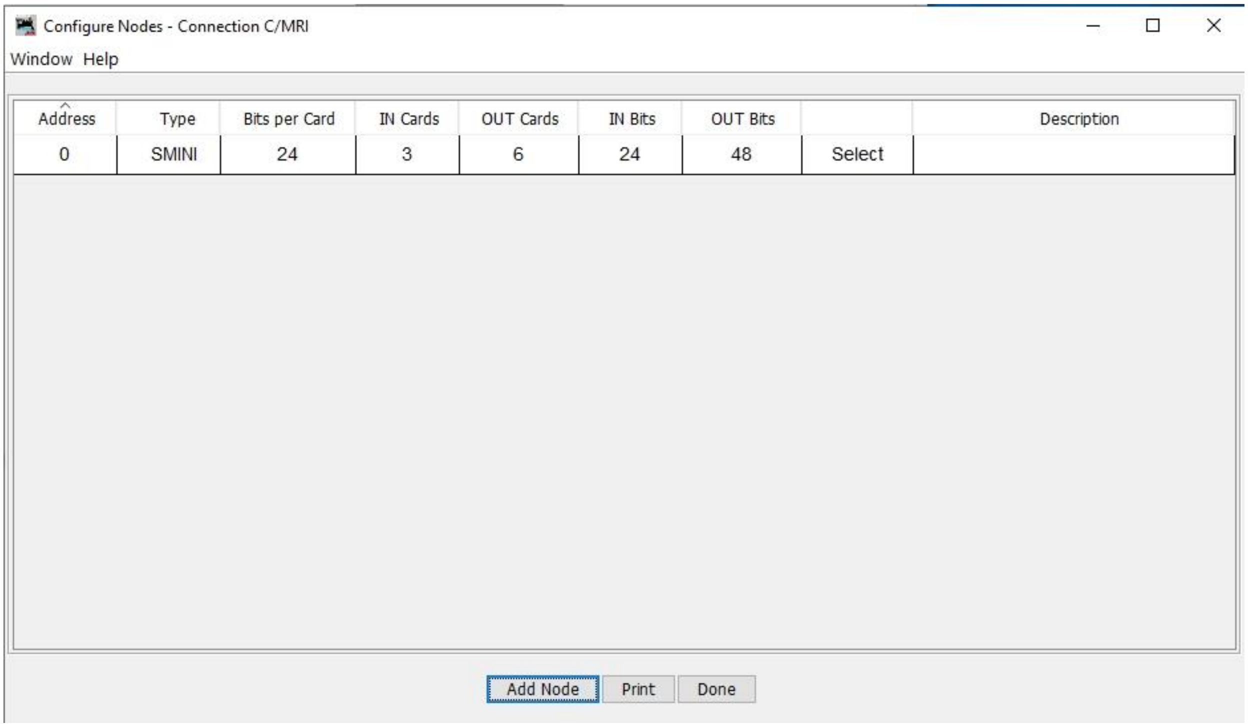
A new blank node list pop-up will be displayed. Click the “Add Node” button to open the next pop-up



C/MRI nodes start at address zero (0) so set that address (the next node if needed would be one (1)) and set the Node Type to SMINI and leave the rest of the boxes set with their default values.

NOTE: You may wish to add a Description to help you remember what this node is used for.

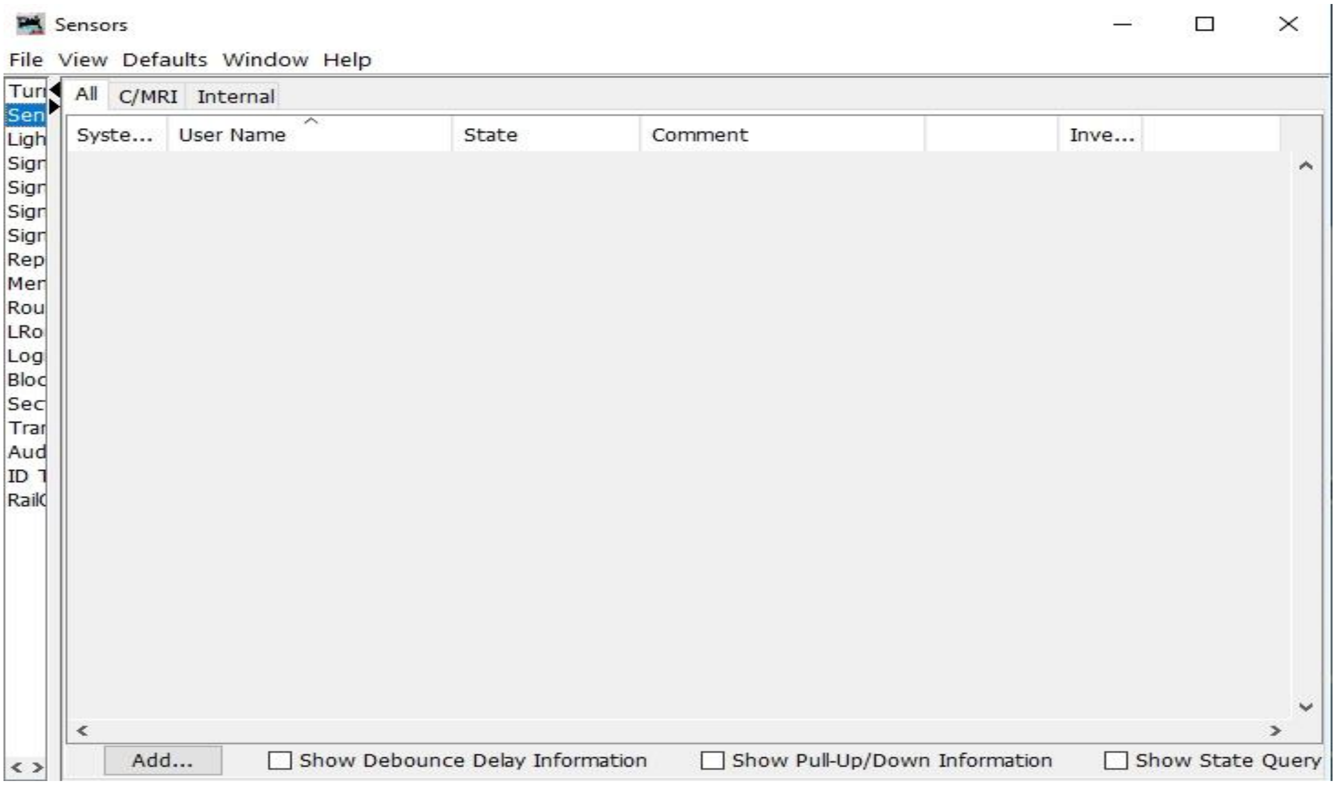
Click to “Add Node” button to close this pop-up



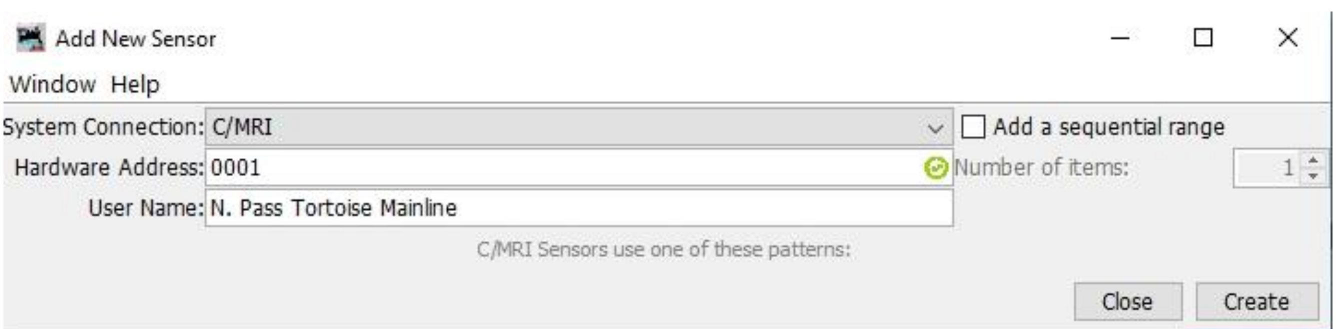
The node configuration pop-up now shows the node you just added. Double check the address (0) and type SMINI before clicking the "Done" button to close this pop-up.

We now have a C/MRI node configured and ready for JMRI to use.

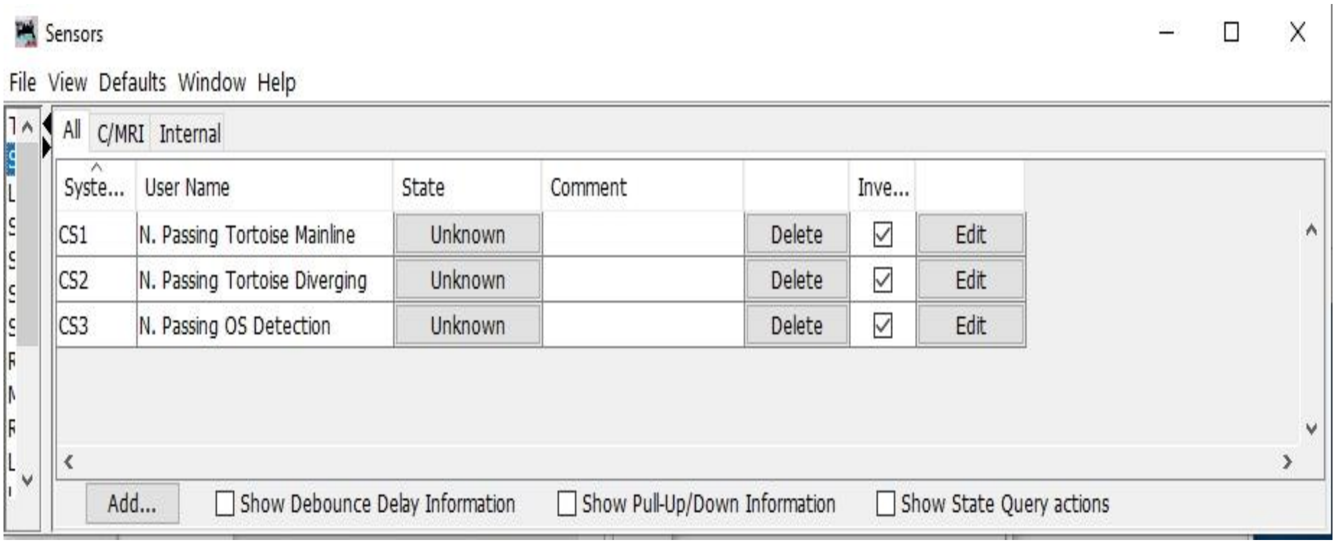
From the splash screen menu select TOOLS/TABLES/SENSORS



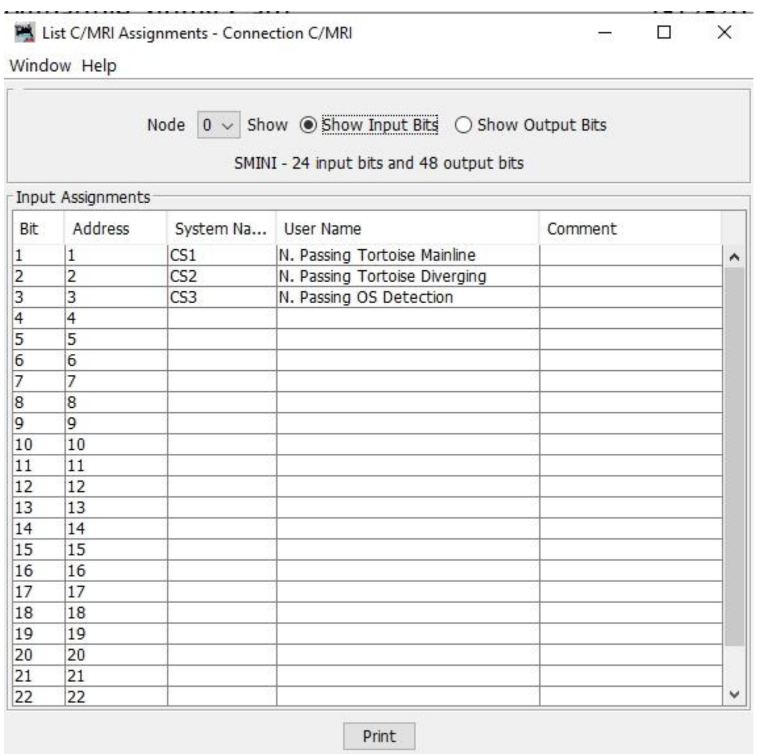
Sensors are input signals for JMRI to use for status displays and logic decisions. We have three (3) inputs on the signal card to define. Click the “Add” button to get started.



Make sure that the System Connection is set to C/MRI. As with outputs the input Hardware Addresses start with one (1) and again we’ll use only node zero (0) so our addresses will be 0001 – 0xxx. We start with address 0001 and give it a User name of our choice. Press the “Create” button and the pop-up will automatically close. Repeat this process to add the other two inputs as shown below.



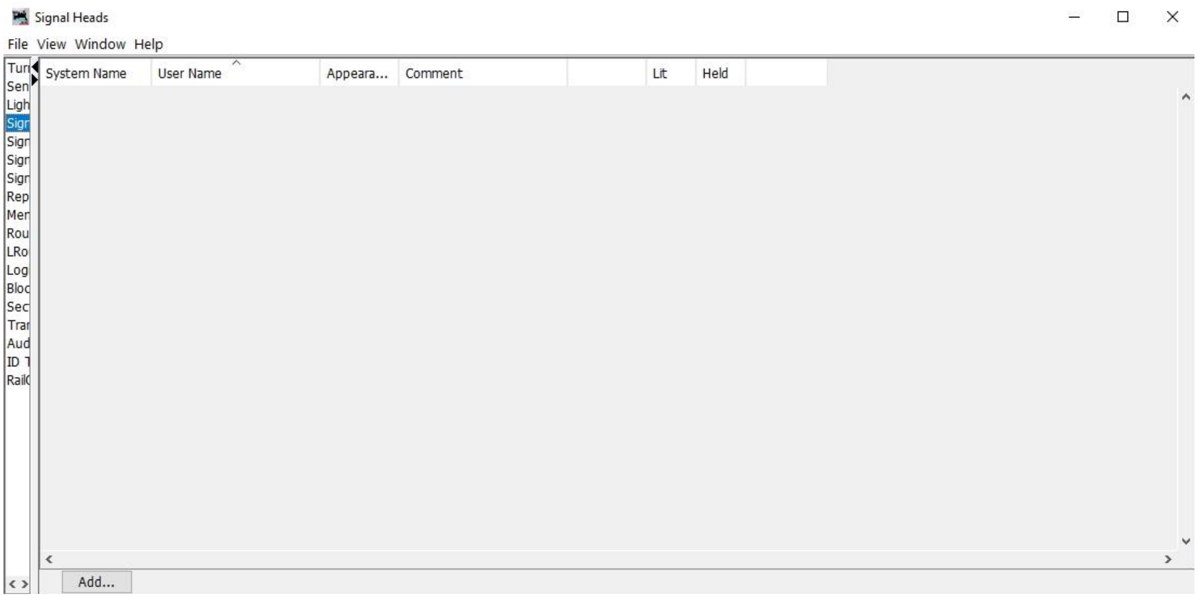
There is one last step to our sensor configuration. In the “State” column when using these sensors it will show either ACTIVE or INACTIVE but these are the opposite of what we expect to see so we check the Invert (Inve..) box to get it to act the way we expect.
From the Splash Screen got to C/MRI/List Manager



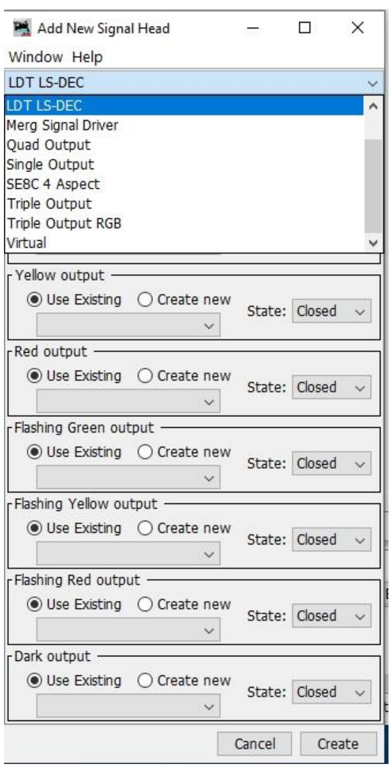
Check that you’re on “Node 0”
Click “Show Input Bits”

Here we can see that JMRI has linked its internal “System Name” and “User Name” to bits in the C/MRI hardware.

From the splash screen select TOOLS/TABLES/SIGNALS/HEADS



Once again out table is empty and we click the “Add” button



The top most drop-down lists all the possible head types. We will use the Single Output, Double Output and Triple Output types to create our heads.

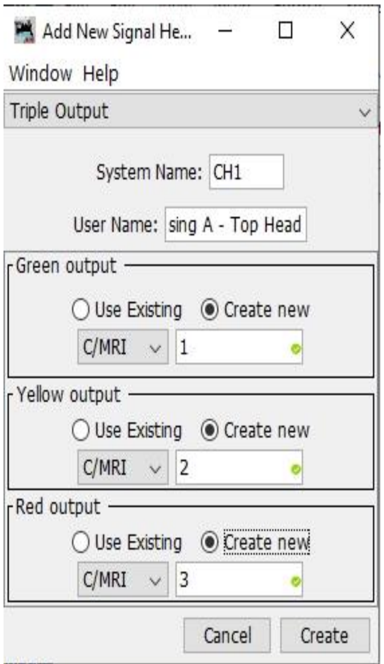
Refer back to the Signal Assignments (page 7) and Siding Entrance Example (page 9) to review the heads we need to create,

A This is the facing signal with two heads. The top head will be a Triple Output head and the bottom will be a Double Output head.

B This is the trailing signal with two heads. The top head will be a Triple Output head and the bottom will be a Single Output head.

C This is the diverging dwarf signal with one head. This will be a Double Output head.

Select the Triple Output from the drop-down list.



System Name:
I chose C (C/MRI) and H (mandatory) one (CH1) to keep things related.

NOTE: use capital letters or JMRI will reject the System Name.

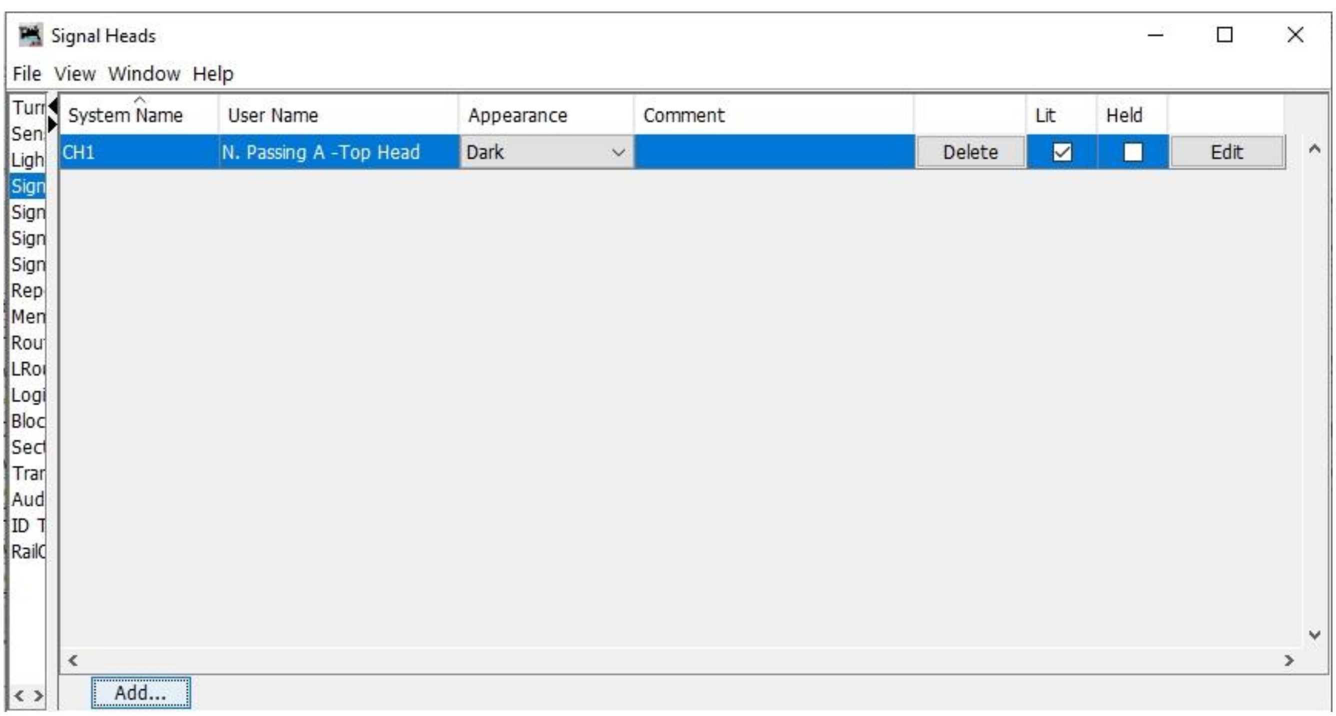
User Name:
I chose = N. Passing Siding A – Top Head

Select "Create new" for all outputs and place the correct output number (page 7) assignment in the right color box.
For this examples

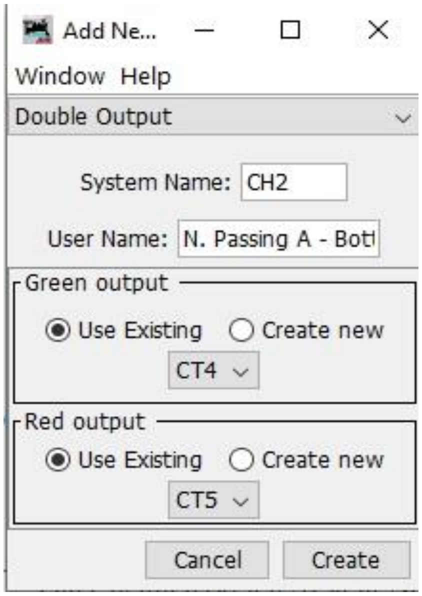
- Green 1
- Yellow 2
- Red 3

Click the "Create" button

The head is added to the table



Click the "Add" button to return to the general pop-up and select a double head (scroll up if not visible)



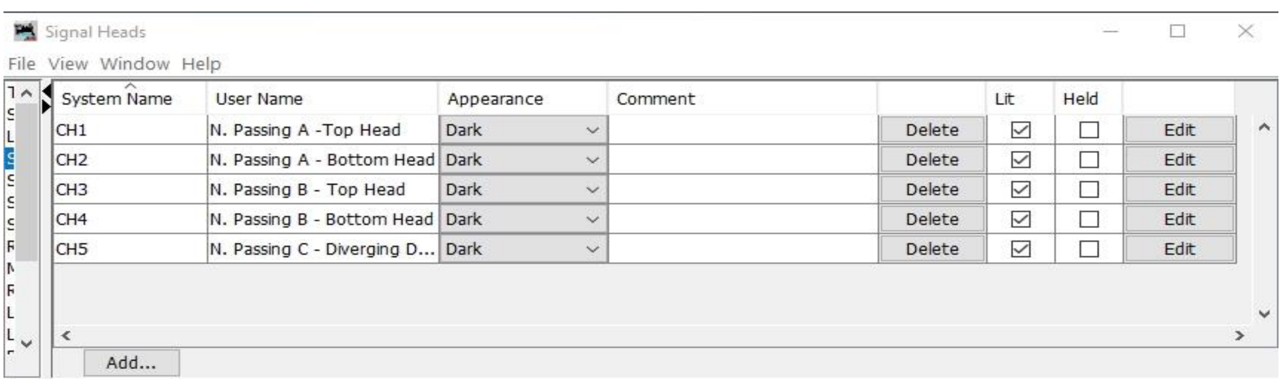
Once again assign a System Name and User Name and refer to the page 7 for the correct outputs to use and select “create new” outputs.

NOTE: Our example uses Yellow and Red lens and your rule book may call for Green and Red. Place the Red output in the red box and the alternate color in the green box. The logic works, you simple terminate whatever color LED you want to the hardware.

Click the “Create” button and the head is added to the table.

NOTE: On pop-ups with more data blocks leave them at the default settings.

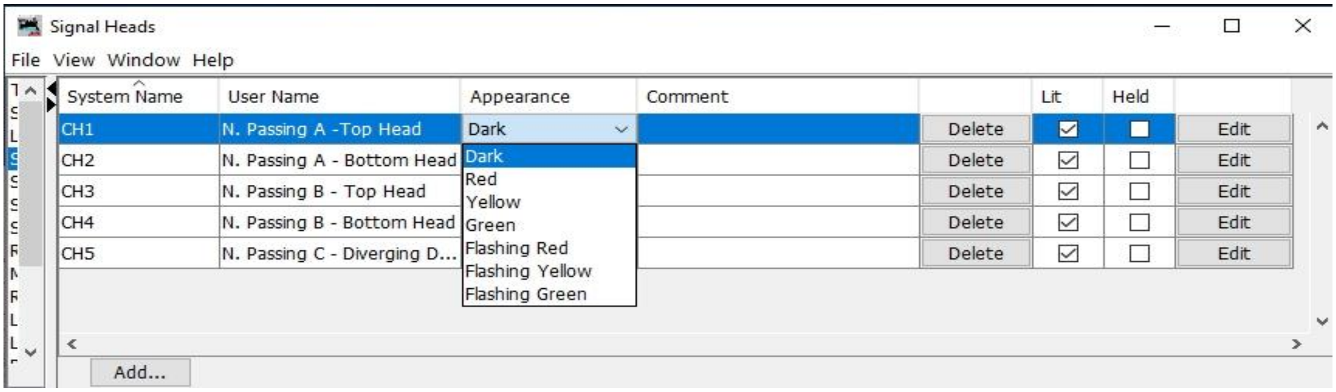
For brevity not all signal head creations are illustrated, but all the information needed is contained in the above examples to complete the rest.



System Name	User Name	Appearance	Comment		Lit	Held	
CH1	N. Passing A -Top Head	Dark		Delete	☑	☐	Edit
CH2	N. Passing A - Bottom Head	Dark		Delete	☑	☐	Edit
CH3	N. Passing B - Top Head	Dark		Delete	☑	☐	Edit
CH4	N. Passing B - Bottom Head	Dark		Delete	☑	☐	Edit
CH5	N. Passing C - Diverging D...	Dark		Delete	☑	☐	Edit

Use the above table to complete the remaining heads.

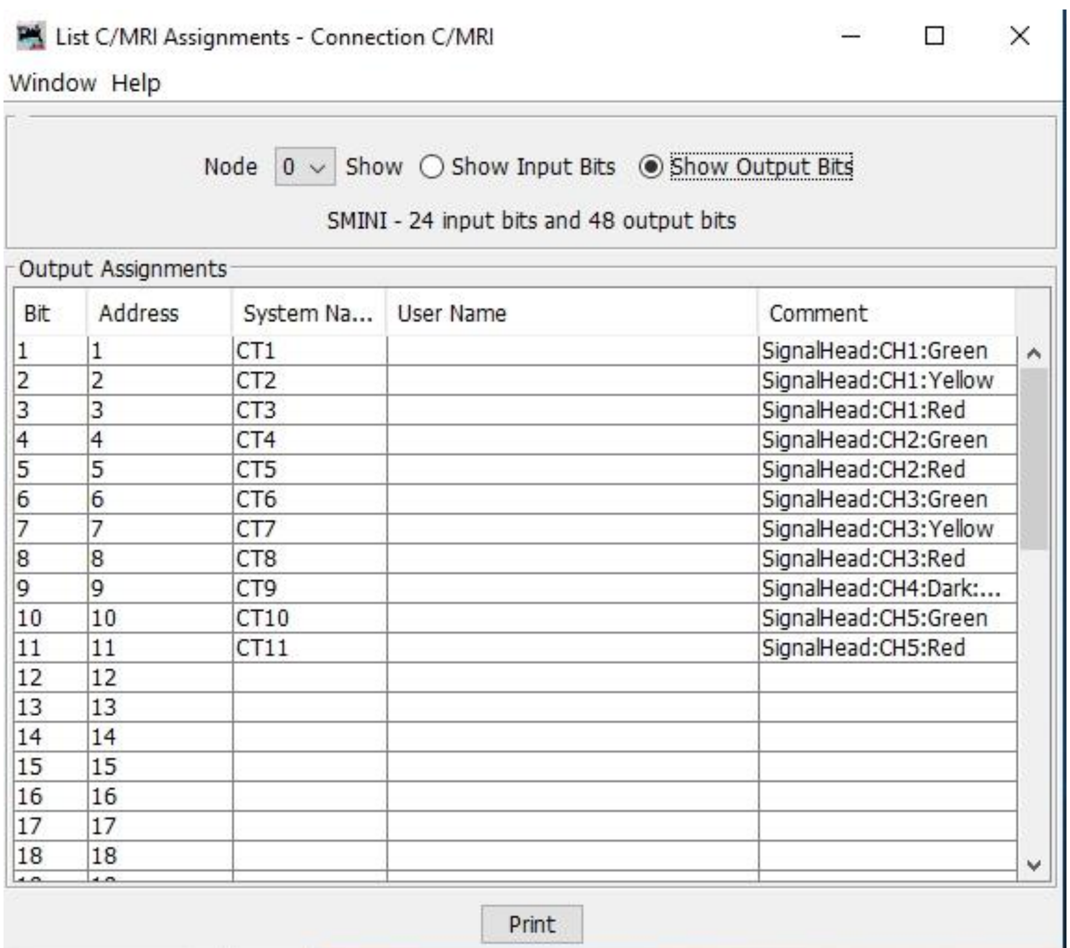
Once entered into the table any head may be selected and tested by using the “Appearance” drop-down list to select the output setting.



System Name	User Name	Appearance	Comment		Lit	Held	
CH1	N. Passing A -Top Head	Dark		Delete	☑	☐	Edit
CH2	N. Passing A - Bottom Head			Delete	☑	☐	Edit
CH3	N. Passing B - Top Head			Delete	☑	☐	Edit
CH4	N. Passing B - Bottom Head			Delete	☑	☐	Edit
CH5	N. Passing C - Diverging D...			Delete	☑	☐	Edit

Lit – light the LEDs in the head
Held – Set Signal to red until released (Think CTC)

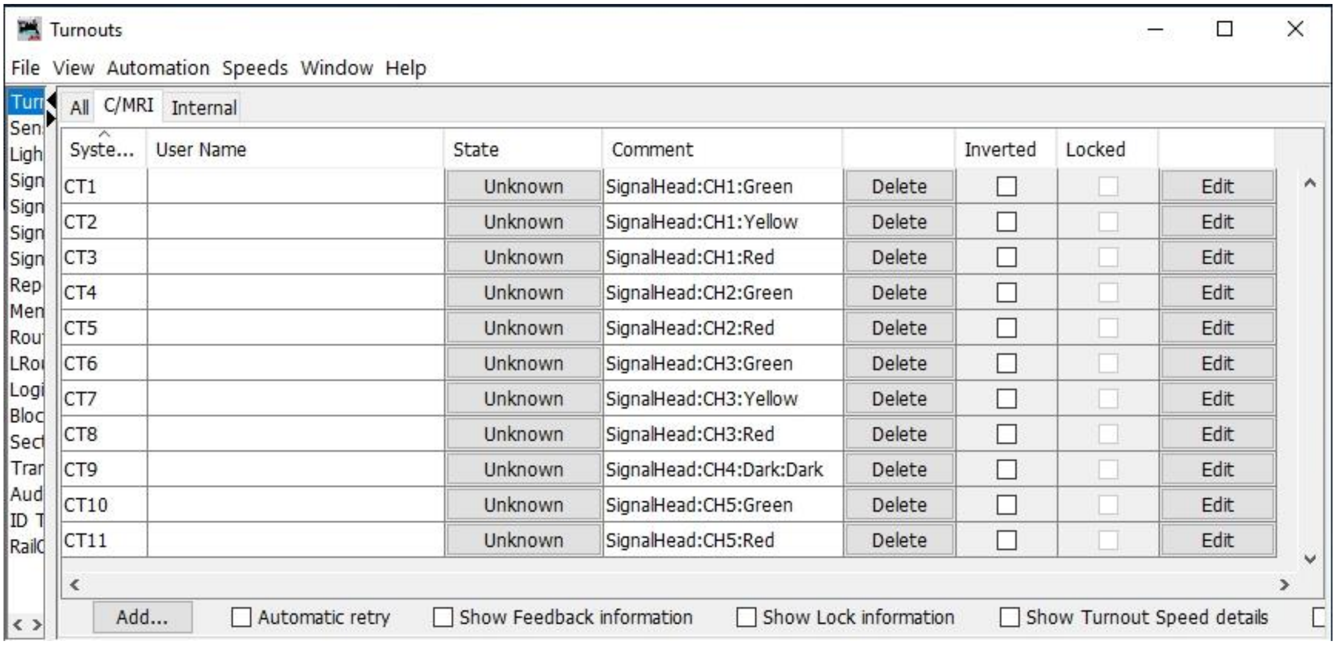
If you closed the List manager then go to the Splash Screen got to C/MRI/List Manager. If you left it open you need to toggle the pop-up between Inputs and Outputs to refresh the list.



Bit	Address	System Na...	User Name	Comment
1	1	CT1		SignalHead:CH1:Green
2	2	CT2		SignalHead:CH1:Yellow
3	3	CT3		SignalHead:CH1:Red
4	4	CT4		SignalHead:CH2:Green
5	5	CT5		SignalHead:CH2:Red
6	6	CT6		SignalHead:CH3:Green
7	7	CT7		SignalHead:CH3:Yellow
8	8	CT8		SignalHead:CH3:Red
9	9	CT9		SignalHead:CH4:Dark:...
10	10	CT10		SignalHead:CH5:Green
11	11	CT11		SignalHead:CH5:Red
12	12			
13	13			
14	14			
15	15			
16	16			
17	17			
18	18			

The C/MRI node now has eleven outputs linked to the signals we just created. But there is an important step you need to know about to avoid confusion.

From the Splash Screen go to TOOLS/TABLES/TURNOUTS



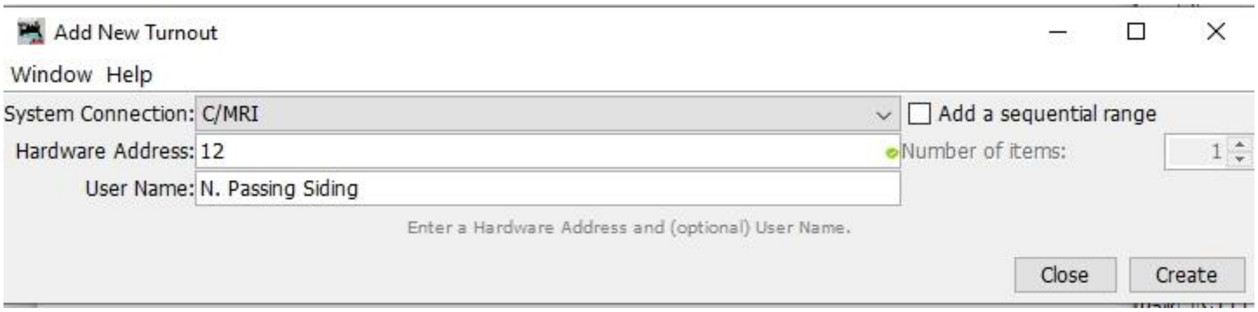
Syste...	User Name	State	Comment		Inverted	Locked	
CT1		Unknown	SignalHead:CH1:Green	Delete			Edit
CT2		Unknown	SignalHead:CH1:Yellow	Delete			Edit
CT3		Unknown	SignalHead:CH1:Red	Delete			Edit
CT4		Unknown	SignalHead:CH2:Green	Delete			Edit
CT5		Unknown	SignalHead:CH2:Red	Delete			Edit
CT6		Unknown	SignalHead:CH3:Green	Delete			Edit
CT7		Unknown	SignalHead:CH3:Yellow	Delete			Edit
CT8		Unknown	SignalHead:CH3:Red	Delete			Edit
CT9		Unknown	SignalHead:CH4:Dark:Dark	Delete			Edit
CT10		Unknown	SignalHead:CH5:Green	Delete			Edit
CT11		Unknown	SignalHead:CH5:Red	Delete			Edit

JMRI automatically created these entries for use. Remember when we create the signal heads we said to "Create New" and then we typed in a System Name. Look at the table above and notice that the System Names correspond to the same names you typed into the signal heads.

Turnouts and Signals are so integrated that much of the same logic is used to control them, so they are entered into the same table.

Click the "Add" button

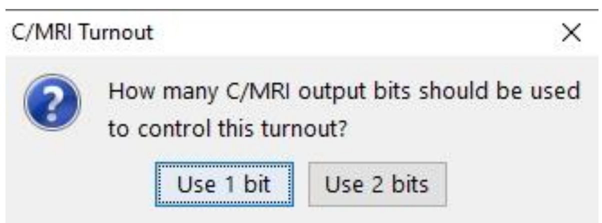
Since we clicked the add button from the "Turnout Table" you will see the pop-up for adding a new turnout.



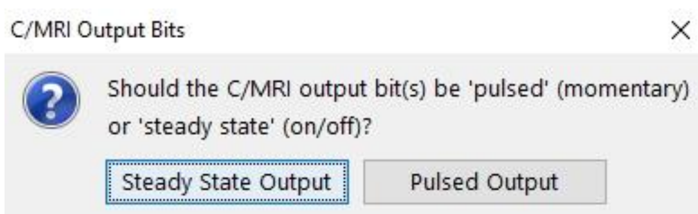
Check that the System Connection setting is C/MRI.
Add a meaningful User Name – such as N. Passing Siding
Hardware Address: The next available address is 12

Click the “Create” button

You then greeted by this pop-up



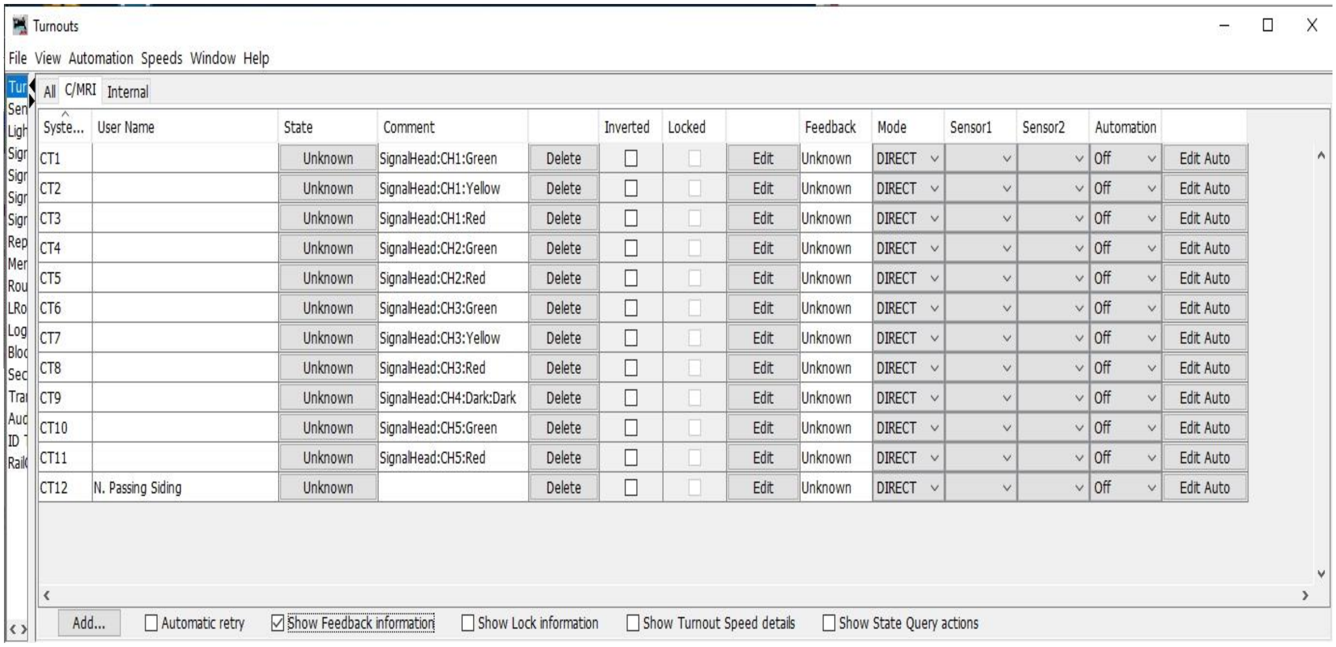
Since we are using Tortoise motors we need two (2) outputs to be able to reverse the motor direction so we select “Use 2 bits.” (That assigns our bit 15 to this turnout)



If we select “Steady State Output” the output is set to either On or Off and left in that state until you command it to change. Since we want to allow the operator to choose between using a local fascia mounted switch or a computer output we must release the output state so that the Siding Board Flip/Flop IC chip can be controlled from either device.

Using the pulsed output setting leaves both bits Off and sending no commands to the turnout (The Siding Board “remembers” the last commanded state.) This allows the use of the momentary fascia mounted turnout switch to command the turnout and return to the Off state when released.

Select “Pulsed Output”



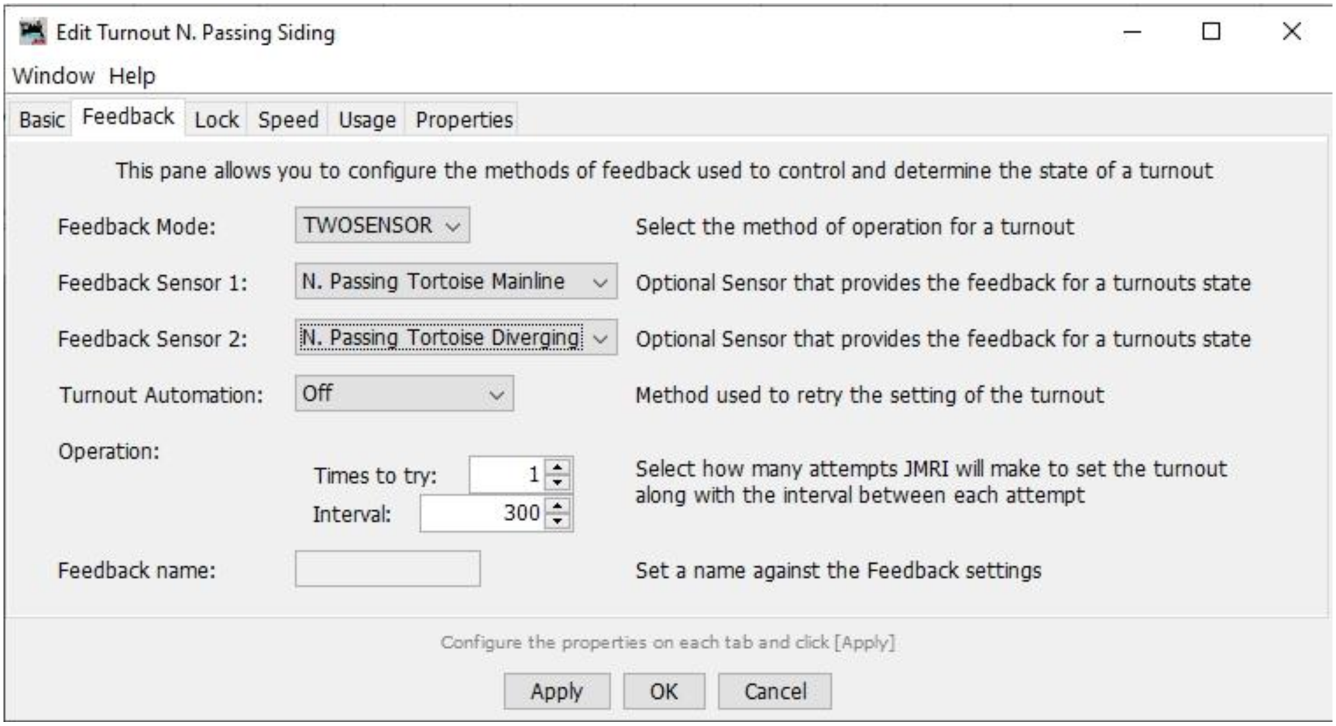
Syste...	User Name	State	Comment		Inverted	Locked		Feedback	Mode	Sensor1	Sensor2	Automation	
CT1		Unknown	SignalHead:CH1:Green	Delete			Edit	Unknown	DIRECT			Off	Edit Auto
CT2		Unknown	SignalHead:CH1:Yellow	Delete			Edit	Unknown	DIRECT			Off	Edit Auto
CT3		Unknown	SignalHead:CH1:Red	Delete			Edit	Unknown	DIRECT			Off	Edit Auto
CT4		Unknown	SignalHead:CH2:Green	Delete			Edit	Unknown	DIRECT			Off	Edit Auto
CT5		Unknown	SignalHead:CH2:Red	Delete			Edit	Unknown	DIRECT			Off	Edit Auto
CT6		Unknown	SignalHead:CH3:Green	Delete			Edit	Unknown	DIRECT			Off	Edit Auto
CT7		Unknown	SignalHead:CH3:Yellow	Delete			Edit	Unknown	DIRECT			Off	Edit Auto
CT8		Unknown	SignalHead:CH3:Red	Delete			Edit	Unknown	DIRECT			Off	Edit Auto
CT9		Unknown	SignalHead:CH4:Dark:Dark	Delete			Edit	Unknown	DIRECT			Off	Edit Auto
CT10		Unknown	SignalHead:CH5:Green	Delete			Edit	Unknown	DIRECT			Off	Edit Auto
CT11		Unknown	SignalHead:CH5:Red	Delete			Edit	Unknown	DIRECT			Off	Edit Auto
CT12	N. Passing Siding	Unknown		Delete			Edit	Unknown	DIRECT			Off	Edit Auto

The turnout table now shows the turnout just created, but you’re not done yet.

Click the “Show Feedback information” check-box.

Everything I’m about to explain can be done from the drop-down boxes on the turnout entry (CT12) but it is easier to see and explain if I use the “Edit” button and open a pop-up that shows everything at once.

Click CT12 Edit Button



This discussion will be limited to the Feedback tab (it's the one we need to make the Siding Board work the other tab defaults will work.)

Set the Feedback Mode to "TWOSENSOR"

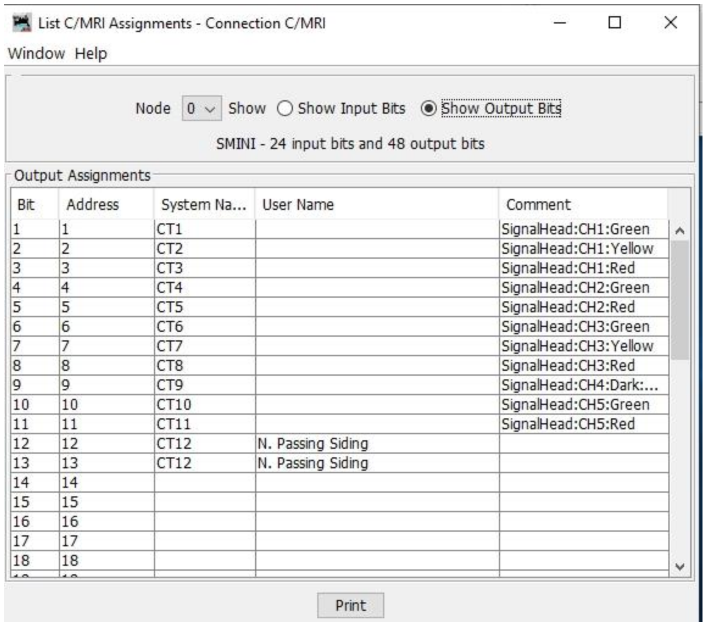
Now we use the two sensors that we created on the sensor table from the drop-down list.

Feedback Sensor 1: = N, Passing Tortoise Mainline
Feedback Sensor 2: = N, Passing Tortoise Diverging

Leave the remaining parameters at the default settings

Click OK

From the Splash Screen go to C/MRI/List Manager.



List C/MRI Assignments - Connection C/MRI

Window Help

Node 0 Show ○ Show Input Bits ◉ Show Output Bits

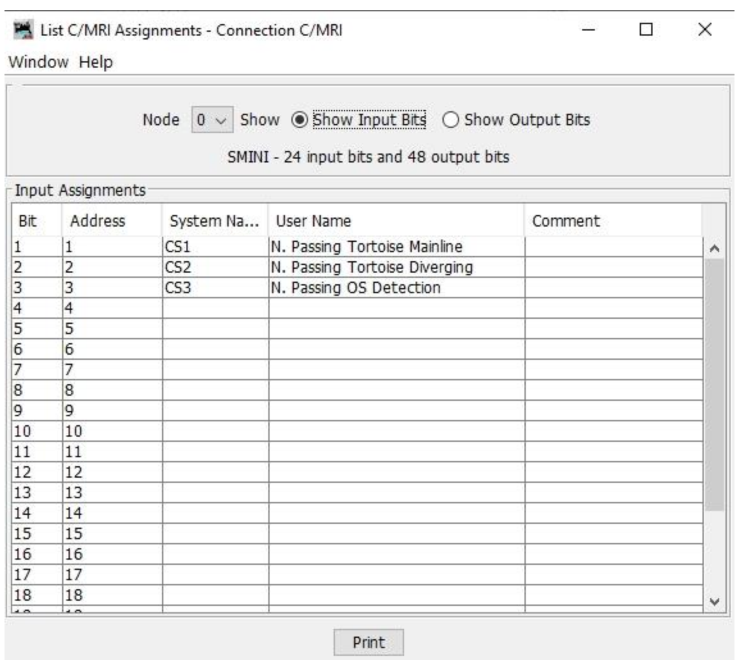
SMINI - 24 input bits and 48 output bits

Output Assignments

Bit	Address	System Na...	User Name	Comment
1	1	CT1		SignalHead:CH1:Green
2	2	CT2		SignalHead:CH1:Yellow
3	3	CT3		SignalHead:CH1:Red
4	4	CT4		SignalHead:CH2:Green
5	5	CT5		SignalHead:CH2:Red
6	6	CT6		SignalHead:CH3:Green
7	7	CT7		SignalHead:CH3:Yellow
8	8	CT8		SignalHead:CH3:Red
9	9	CT9		SignalHead:CH4:Dark:...
10	10	CT10		SignalHead:CH5:Green
11	11	CT11		SignalHead:CH5:Red
12	12	CT12	N. Passing Siding	
13	13	CT12	N. Passing Siding	
14	14			
15	15			
16	16			
17	17			
18	18			

Print

Note that CT12 has been assigned to bits 12 and 13, that’s because 2 bit control has been selected for the turnout.



List C/MRI Assignments - Connection C/MRI

Window Help

Node 0 Show ◉ Show Input Bits ○ Show Output Bits

SMINI - 24 input bits and 48 output bits

Input Assignments

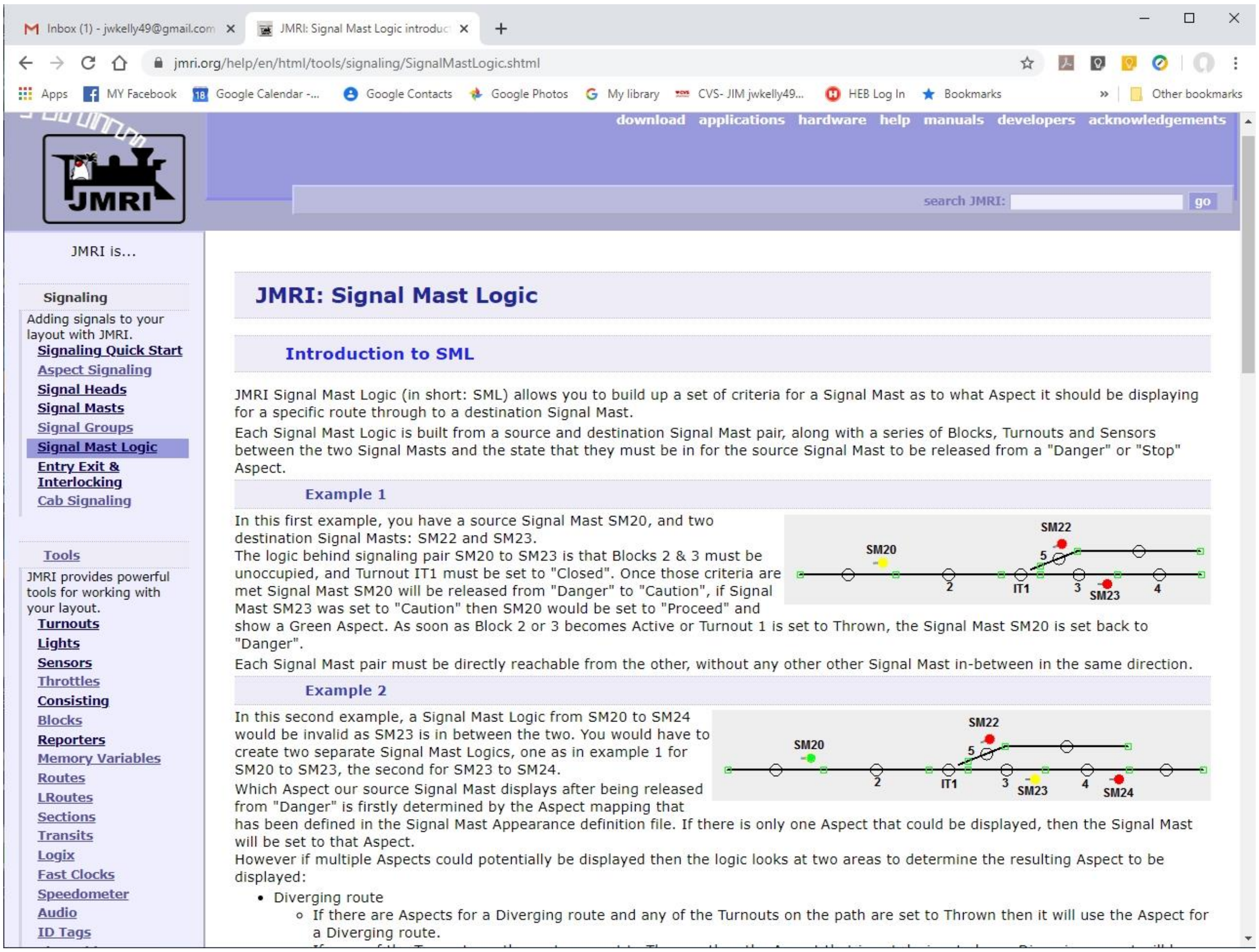
Bit	Address	System Na...	User Name	Comment
1	1	CS1	N. Passing Tortoise Mainline	
2	2	CS2	N. Passing Tortoise Diverging	
3	3	CS3	N. Passing OS Detection	
4	4			
5	5			
6	6			
7	7			
8	8			
9	9			
10	10			
11	11			
12	12			
13	13			
14	14			
15	15			
16	16			
17	17			
18	18			

Print

Toggle between the “Show Input Bits” and “Show Output Bits” We see 13 outputs and 3 inputs have been assigned to C/MRI Node 0. This is all the I/O that the Signal Siding Card can control. The next siding would be added to Node 1 and start with “System Names” CS4 for sensors (inputs) and CT14 for signal heads (outputs.)

Before going any further a brief introduction the JMRI Signal philosophy is needed.

https://www.jmri.org/help/en/html/tools/signaling/SignalMastLogic.shtml



download applications hardware help manuals developers acknowledgements

search JMRI: go

JMRI is...

Signaling

Adding signals to your layout with JMRI.

Signaling Quick Start
Aspect Signaling
Signal Heads
Signal Masts
Signal Groups
Signal Mast Logic
Entry Exit & Interlocking
Cab Signaling

Tools

JMRI provides powerful tools for working with your layout.

Turnouts
Lights
Sensors
Throttles
Consisting
Blocks
Reporters
Memory Variables
Routes
LRoutes
Sections
Transits
Logix
Fast Clocks
Speedometer
Audio
ID Tags

JMRI: Signal Mast Logic

Introduction to SML

JMRI Signal Mast Logic (in short: SML) allows you to build up a set of criteria for a Signal Mast as to what Aspect it should be displaying for a specific route through to a destination Signal Mast.

Each Signal Mast Logic is built from a source and destination Signal Mast pair, along with a series of Blocks, Turnouts and Sensors between the two Signal Masts and the state that they must be in for the source Signal Mast to be released from a "Danger" or "Stop" Aspect.

Example 1

In this first example, you have a source Signal Mast SM20, and two destination Signal Masts: SM22 and SM23.

The logic behind signaling pair SM20 to SM23 is that Blocks 2 & 3 must be unoccupied, and Turnout IT1 must be set to "Closed". Once those criteria are met Signal Mast SM20 will be released from "Danger" to "Caution", if Signal Mast SM23 was set to "Caution" then SM20 would be set to "Proceed" and show a Green Aspect. As soon as Block 2 or 3 becomes Active or Turnout 1 is set to Thrown, the Signal Mast SM20 is set back to "Danger".

Each Signal Mast pair must be directly reachable from the other, without any other other Signal Mast in-between in the same direction.

Example 2

In this second example, a Signal Mast Logic from SM20 to SM24 would be invalid as SM23 is in between the two. You would have to create two separate Signal Mast Logics, one as in example 1 for SM20 to SM23, the second for SM23 to SM24.

Which Aspect our source Signal Mast displays after being released from "Danger" is firstly determined by the Aspect mapping that has been defined in the Signal Mast Appearance definition file. If there is only one Aspect that could be displayed, then the Signal Mast will be set to that Aspect.

However if multiple Aspects could potentially be displayed then the logic looks at two areas to determine the resulting Aspect to be displayed:

- Diverging route
  - If there are Aspects for a Diverging route and any of the Turnouts on the path are set to Thrown then it will use the Aspect for a Diverging route.

This is must reading before making the final decision of your signal system selection. Up to this point this manual has shown you how to connect the maximum I/O from the Signal Siding Board without referencing a specific signaling prototype (or Basic Model Signals.)

This example will continue using CSX-1998 as the Signal System.

Here is an example of the aspects for the first mast we will create.

Reference https://www.jmri.org/xml/signals/CSX-1998/index.shtml for aspects used in these examples.

JMRI "CSX-1998: Double head 3-2 (approach) color light high signal" Appearance Table

For aspect table: CSX-1998

Name: Double head 3-2 (approach) color light high signal

Appearance reference: As described in the CSX Transportation Signal Rules - 281-298 January 1998.

Appearances for a three over two lamp signal head (lower head approach) without other badging

Rule 281: Clear

Indication: Proceed.

Show: green
Show: red

Appearance reference: 281 (b)

Rule 281-D: Limited Approach

Indication: Limited Speed through turnouts, crossovers, sidings and over power-operated switches; then proceed, prepared to stop at next signal.

Show: red
Show: flashyellow

Appearance reference: 281-D (b)

Rule 282-A: Advance Approach

Indication: Proceed prepared to stop at second signal.

Show: yellow
Show: yellow

Appearance reference: 282-A (b)

Rule 285: Approach

Indication: Proceed prepared to stop at the next signal. Train exceeding medium speed must immediately begin reduction to Medium Speed as soon as the engine passes the Approach Signal.

Show: yellow
Show: red

Appearance reference: 285 (b)

Rule 286: Medium Approach

Indication: Medium Speed through turnouts, crossovers, sidings and over power-operated switches, then proceed, prepared to stop at next signal.

Show: red
Show: yellow

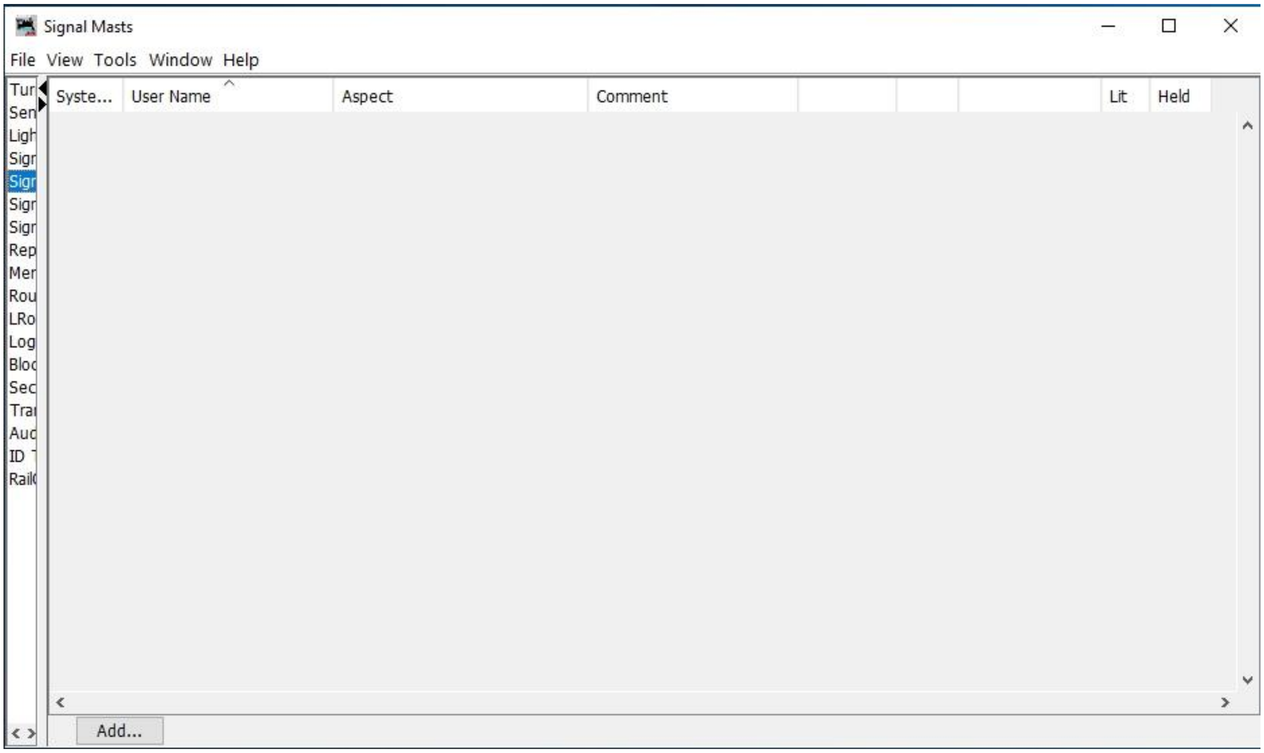
Appearance reference: 286 (b)

Rule 292: Stop

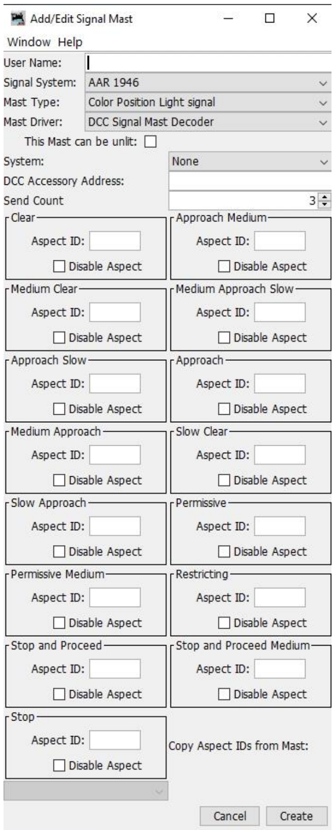
Indication: Stop.

Show: red
Show: red

From the Splash Screen select TOOLS/TABLES/SIGNALS/SIGNAL MASTS

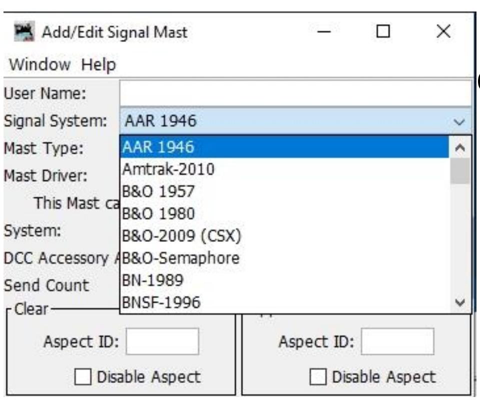


Click the “Add” button

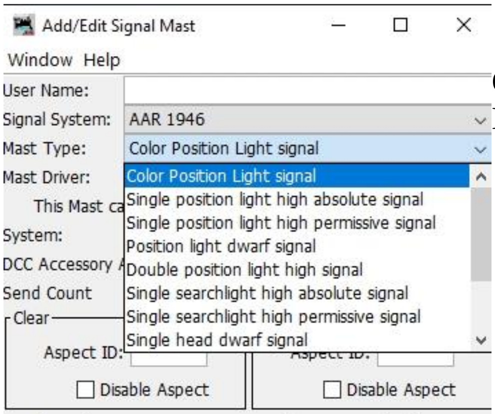


This pop-up changes shape and content as you select items from the drop-down boxes.

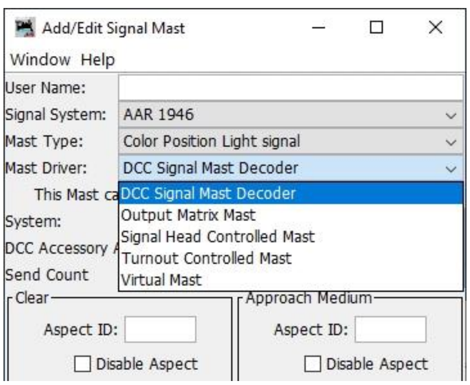
Enter the User Name N. Passing Siding (Facing)



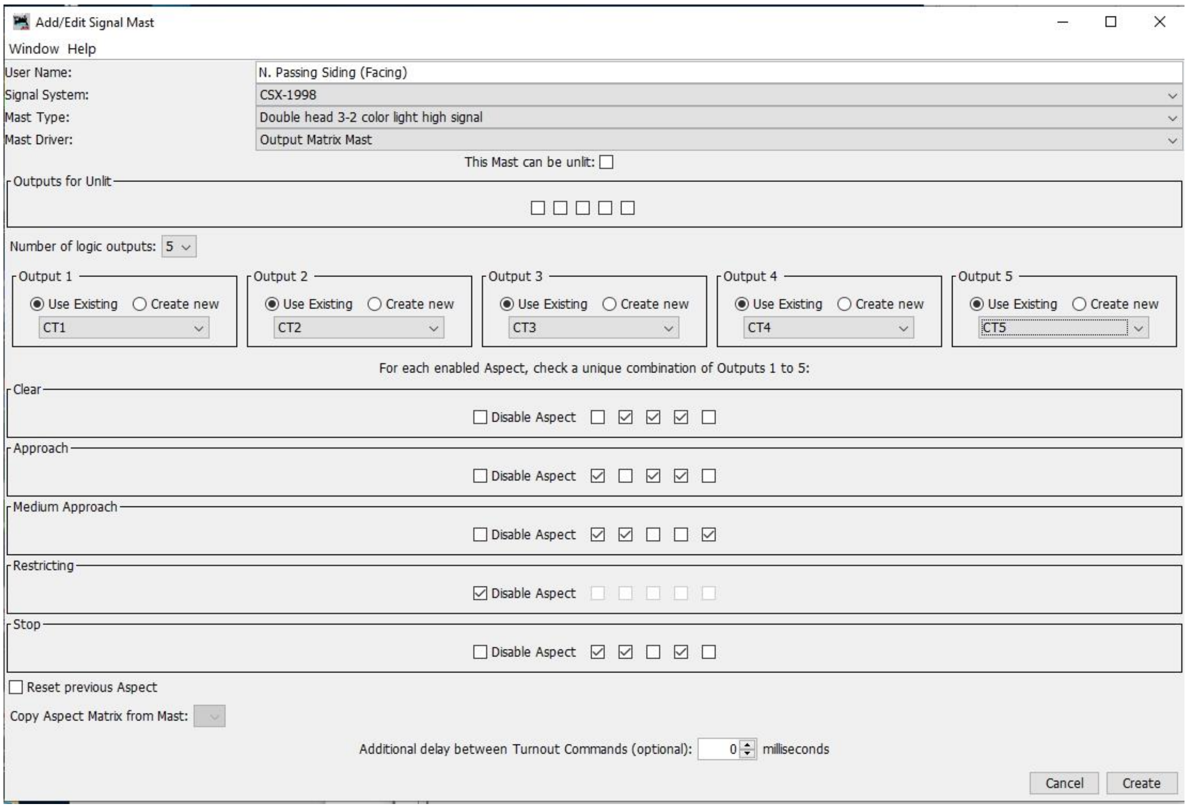
Open the signal System drop-down list-box and select CSX-1998



Open the Mast Type drop-down list and select “Double Head ...3-2 color light head high signal.”



Open the Mast Driver drop-down list and select “Output Matrix Logic.”



Make sure the "Number of Logic Outputs" is set to 5

Remember from the Siding Entrance Example that we had 2 heads (3 over 2 [3-2]) at the facing end of the turnout and labeled it as "A." Next you create 2 heads for "A" using CT1 through CT5

CT1 = Green
CT2 = Yellow
CT3 = Red

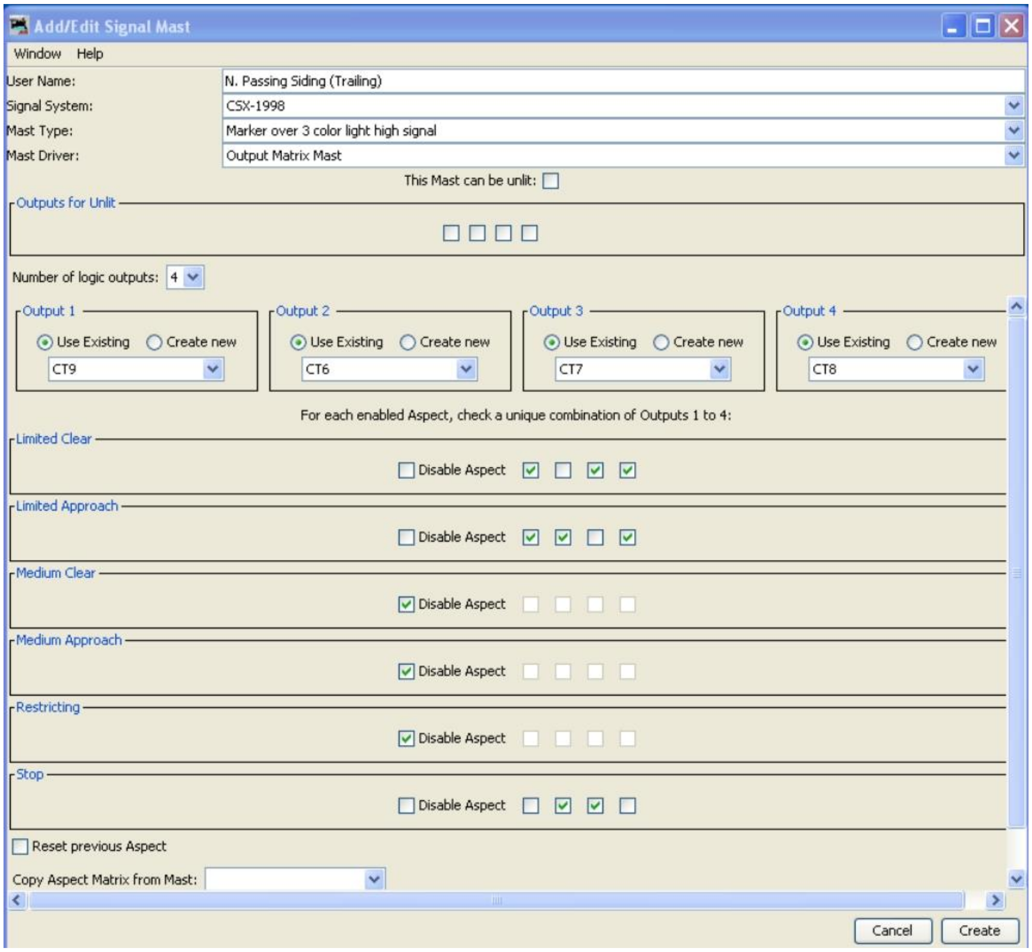
CT4 = Yellow
CT5 = Red

Enter them from left to right for the Outputs.

Under the Aspects section place a check for each lamp you want turned off and disable any you chose not to use.

Click Create
The new signal mast now appears in the Signal Mast Table
Click Add to enter the next Mast

Select Mast Type – Marker Over 3 Color High Light Signal
Set User Name – N. Passing Siding (Trailing)



Make sure the “Number of Logic Outputs” is set to 4

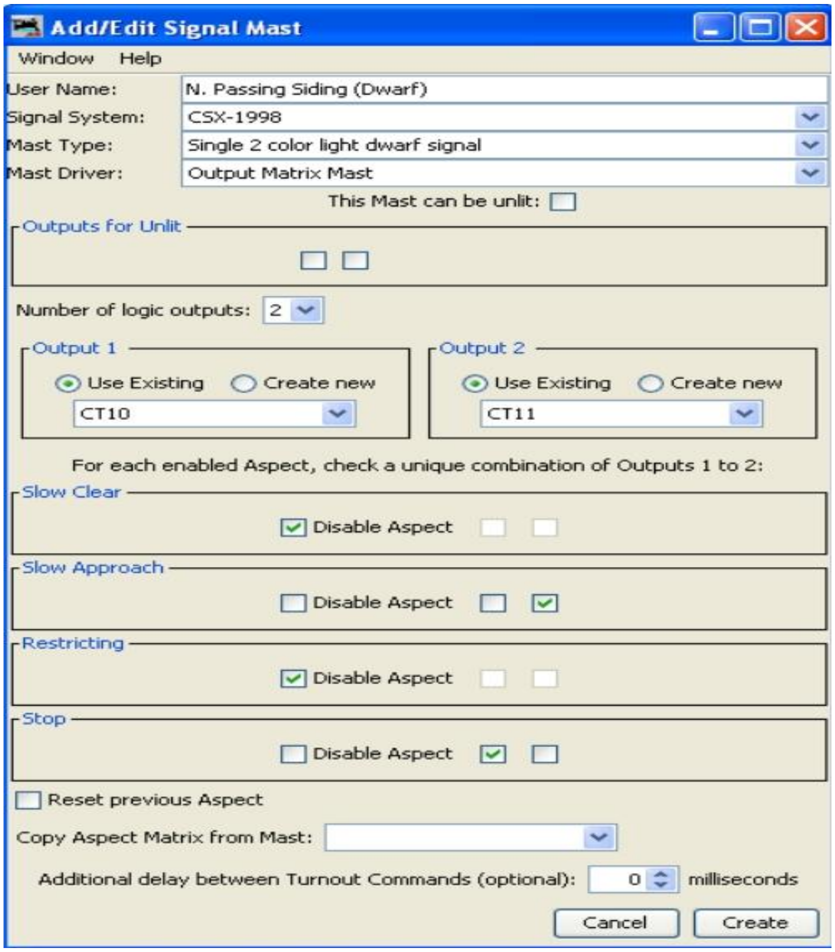
Remember from the Siding Entrance Example that we had 2 heads (3 over 1 [3-1]) at the trailing end of the turnout and labeled it as “B.” Next you create 2 heads for “B” using CT6 through CT9

CT9 = Red Note: CT9 has been moved in order to place it as the “TOP” marker.
CT6 = Green
CT7 = Yellow
CT8 = Red

Enter them from left to right for the Outputs.

Under the Aspects section place a check for each lamp you want turned off and disable any you chose not to use.

Click Create
The new signal mast now appears in the Signal Mast Table
Click Add to enter the next Mast
Select Mast Type – Single 2 Color Light Dwarf Signal
Set User Name – N. Passing Siding (Dwarf)



Make sure the “Number of Logic Outputs” is set to 2

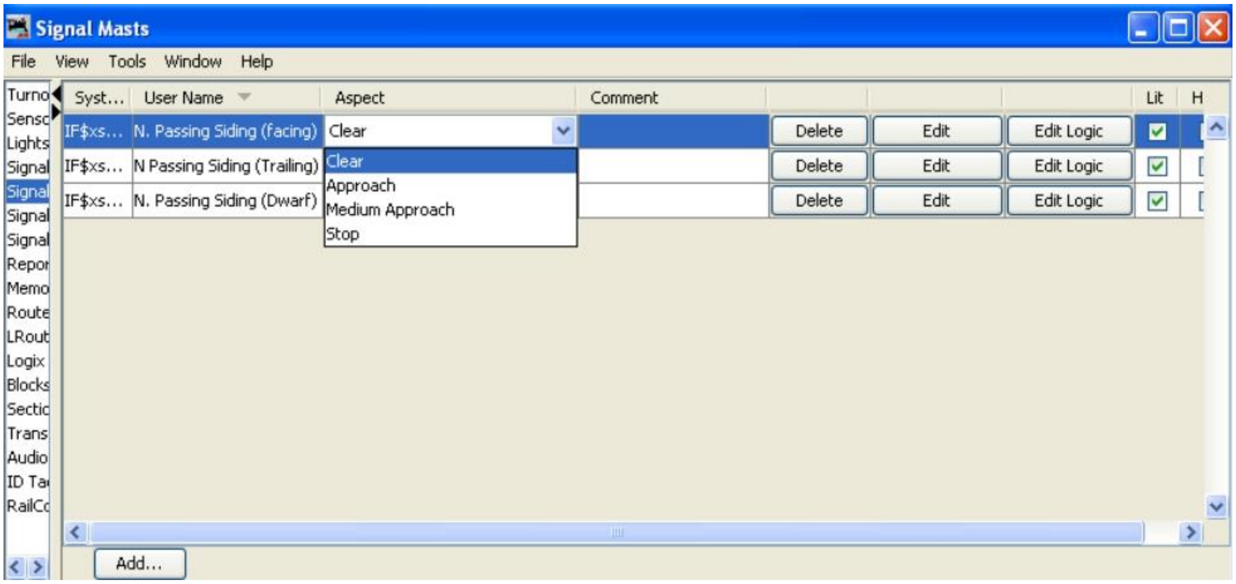
Remember from the Siding Entrance Example that we had a dwarf signal at the diverging end of the turnout and labeled it as “C.” Next you create the head for “C” using CT10 and CT11

CT10 = Yellow
CT11 = Red

Enter them from left to right for the Outputs.

Under the Aspects section place a check for each lamp you want turned off and disable any you chose not to use.

Click Create
The new signal mast now appears in the Signal Mast Table



Now from the Signal Mast Table you can control the signal heads combined as "MASTS" and control them as a single unit to display an "Aspect"

JMRI is now configured to work with C/MRI node 0 and will now poll COM port 1 to send and receive I/O commands and status.

Arduino Code

```
#include <CMRI.h>
#include <Auto485.h>
//Author: Michael Adams (<http://www.michael.net.nz>)
// Copyright (C) 2012 Michael D K Adams.
// Released under the MIT license.

// three inputs
#define Mainline A4
#define Diverging 4
#define OSdetection 3

// signal head CH1  -- A N. Passing siding facing top head
#define Agreen A3
#define Ayellow A2
#define Ared A1

// signal head CH2 -- A N. Passing siding facing bottom head
#define AbottomGreen A0
#define AbottomRed 13

// signal head CH3 -- B N passing siding trailing top head
#define Bgreen 12
#define Byellow 11
#define Bred 10

// signal head CH4 -- B N passing siding trailing bottom head
#define BbottomRed 9

// signal head CH5 -- C N. passing siding dwarf signal
#define Cgreen 8
#define Cred 7

// outputs to Tortoise pins 1 and 8
#define Tmainline 6
#define Tdiverging 5

#define DE_PIN 2
Auto485 bus(DE_PIN);

#define CMRI_ADDR 0
CMRI cmri(CMRI_ADDR, 24, 48, bus);
```

```
void setup() {
  bus.begin(9600,SERIAL_8N2);

  pinMode(Mainline, INPUT_PULLUP);
  pinMode(Diverging, INPUT_PULLUP);
  pinMode(OSdetection, INPUT_PULLUP);

  pinMode(Agreen, OUTPUT);
  pinMode(Ayellow, OUTPUT);
  pinMode(Ared, OUTPUT);

  pinMode(AbottomGreen, OUTPUT);
  pinMode(AbottomRed, OUTPUT);

  pinMode(Bgreen, OUTPUT);
  pinMode(Byellow, OUTPUT);
  pinMode(Bred, OUTPUT);

  pinMode(BbottomRed, OUTPUT);

  pinMode(Cgreen, OUTPUT);
  pinMode(Cred, OUTPUT);

  pinMode(Tmainline, OUTPUT);
  pinMode(Tdiverging, OUTPUT);
}

void loop() {
  // Inputs to C/MRI
  cmri.process();
  cmri.set_bit(0, digitalRead(Mainline));
  cmri.set_bit(1, digitalRead(Diverging));
  cmri.set_bit(2, digitalRead(OSdetection));

  // Output from C/MRI
  digitalWrite(Agreen, cmri.get_bit(0));
  digitalWrite(Ayellow, cmri.get_bit(1));
  digitalWrite(Ared, cmri.get_bit(2));

  digitalWrite(AbottomGreen, cmri.get_bit(3));
  digitalWrite(AbottomRed, cmri.get_bit(4));

  digitalWrite(Bgreen, cmri.get_bit(5));
  digitalWrite(Byellow, cmri.get_bit(6));
  digitalWrite(Bred, cmri.get_bit(7));
```

```
  digitalWrite(BbottomRed, cmri.get_bit(8));

  digitalWrite(Cgreen, cmri.get_bit(9));
  digitalWrite(Cred, cmri.get_bit(10));

  digitalWrite(Tmainline, cmri.get_bit(11));
  digitalWrite(Tdiverging, cmri.get_bit(12));
}
```