

Introduction to PanelPro - Logix™

Dick Bronson - *R R - C irK its*, *Inc*.

Intro to Logix[™]

Indirect layout control (PP-clinic-3)





Why LogiX?

- When Dave Duchamp first started adding a graphical logic package to JMRI we wondered about what to call it. "Logic" seemed to be a logical name for logic, but Dave had already added 'Lights' as a function, therefore 'L' was no longer available as an item name, so he just used 'X' instead. The logic function was 'Internal' to JMRI, so its system name was 'I'. This means the the proper identifier for the logic function became 'IX' and we jokingly started calling them Logix in our e-mail discussions because of the 'IX'. The name has stuck.
- Logix are functionally similar to industrial ladder logic in that they do not have any parenthetical structure. Therefore there is no logical 'OR' function. To do an OR you simply create multiple different Logix for each different conditional, or else invert the sense of items and use the 'NOT AND' instead.

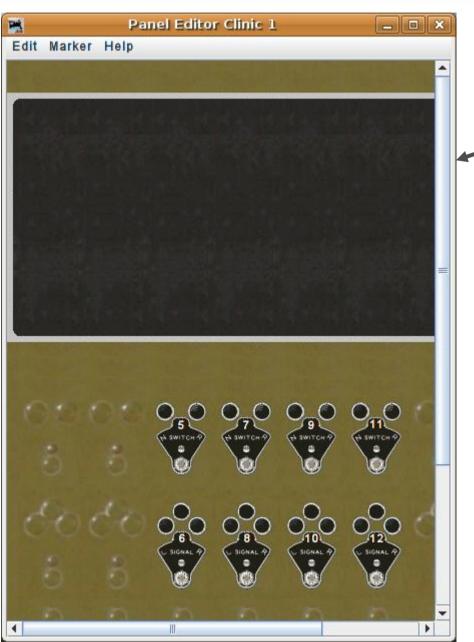


Layout control

- In our previous clinic we simply tied our active icons directly to the layout commands that we needed to send. This is no more sophisticated that drilling some holes in a piece of Masonite, spray painting some lines, mounting some switches and lamps, and then connecting them to our switch machines. Granted a computer can usually be found for not very much money, but a few switches or push buttons, a chopped up string of Christmas tree lights, and some paint would be cheaper.
- On the prototype railroads it is not allowable to have remote control of turnouts without some fairly reliable method of knowing the position of the points and preventing them from ever being changed while a train is crossing them. (or about to) Now that we mention it, these are pretty good things to do for our models as well, even if the life hazard is less. (not counting what might happen to the dispatcher when he accidentally sends that new brass onto the floor through that place with no scenery)
- All this to say, maybe just flipping a turnout with a remote switch isn't the best idea after all, especially if you can't see it from the panel.



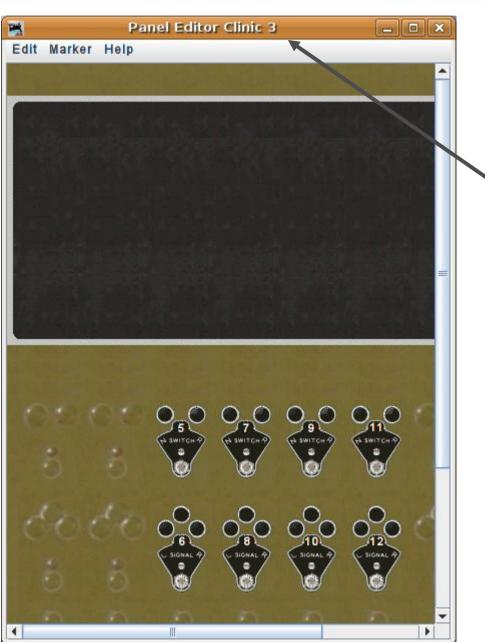
Fixed images



- First lets load in the basic
- panel background we made in clinic #1 then rename it and save it as clinic #3



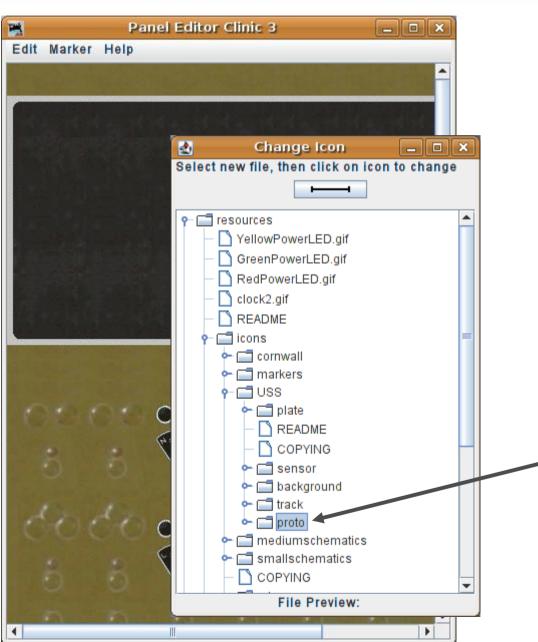
Fixed images



- First lets load in the basic panel background we made in clinic #1 then rename it and save it as clinic #3
- You will be expected to know how to do the basic operations already covered in previous sessions, so I am not going to repeat the detail of each operation as we move along.



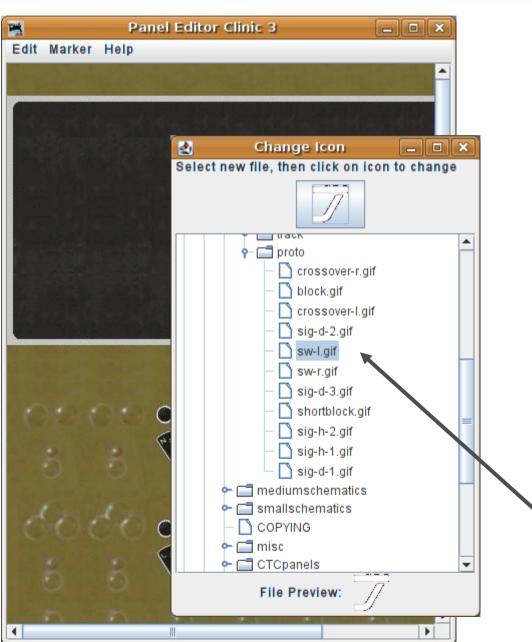
Fixed images



- First lets load in the basic panel background we made in clinic #1 then rename it and save it as clinic #3
- You will be expected to know how to do the basic operations already covered in previous sessions, so I am not going to repeat the detail of each operation as we move along.
- Navigate to the 'proto' folder where we have a set of images created from photographs of an original (unrestored) classic era prototype US&S CTC machine.



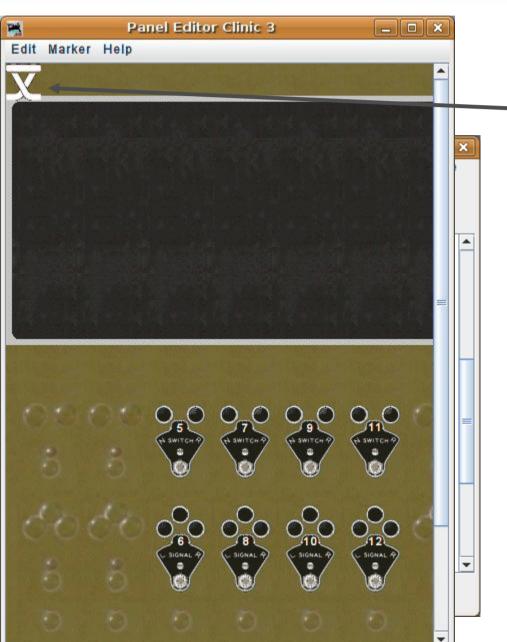
Fixed images



- First lets load in the basic panel background we made in clinic #1 then rename it and save it as clinic #3
- You will be expected to know how to do the basic operations already covered in previous sessions, so I am not going to repeat the detail of each operation as we move along.
- Navigate to the 'proto' folder where we have a set of images created from photographs of an original (unrestored) classic era prototype US&S CTC machine.
- These few images are not designed for animation, but for making a more realistic panel.



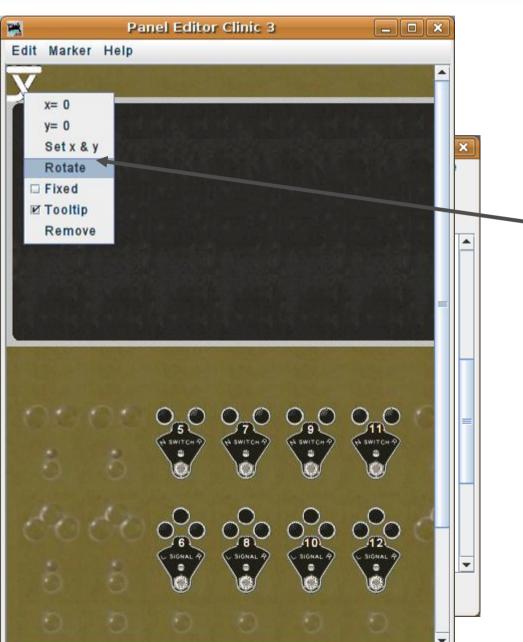
Fixed images



- Use the 'Add icon:' button to add two left (sw-l.gif) and two
 - right (sw-r.gif) turnout icons to our panel.



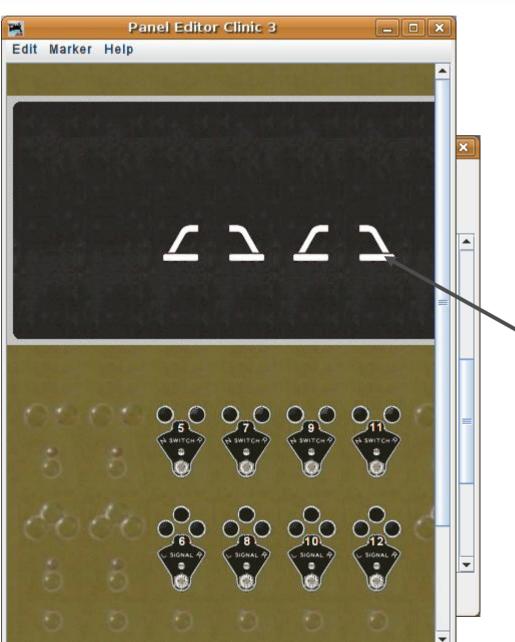
Fixed images



- Use the 'Add icon:' button to add two left (sw-l.gif) and two right (sw-r.gif) turnout icons to our panel.
- These images only face in one direction, so they will need to be rotated for our use on this panel. Right click (meta for Mac) to bring up the tools, then click on 'Rotate' to rotate 90°.



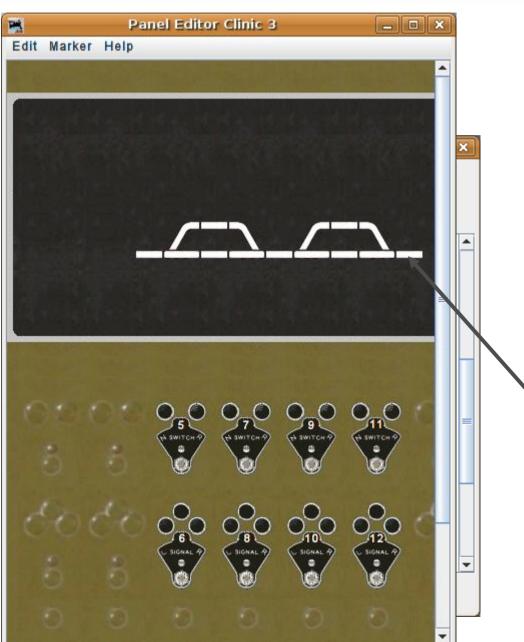
Fixed images



- Use the 'Add icon:' button to add two left (sw-l.gif) and two right (sw-r.gif) turnout icons to our panel.
- These images only face in one direction, so they will need to be rotated for our use on this panel. Right click (meta for Mac) to bring up the tools, then click on 'Rotate' to rotate 90°.
- Each icon is 'rotated' twice to face up, and then positioned on the panel.



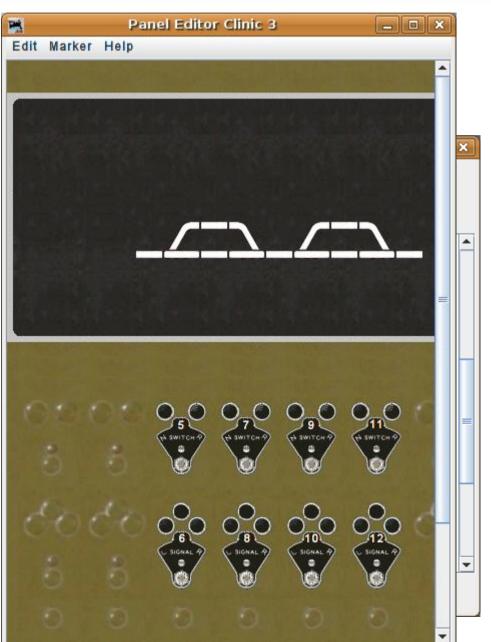
Fixed images



- Use the 'Add icon:' button to add two left (sw-l.gif) and two right (sw-r.gif) turnout icons to our panel.
- These images only face in one direction, so they will need to be rotated for our use on this panel. Right click (meta for Mac) to bring up the tools, then click on 'Rotate' to rotate 90°.
- Each icon is 'rotated' twice to face up, and then positioned on the panel.
- Add in some straight track images (block.gif) and we have our basic track plan drawn.



Sensor images



Occupancy sensors

One of the 'rules' we have for remote operation is that we do not throw a switch under a train. To accomplish that we need to know when a train is on the switch or 'OS' (On Switch) section. (OS can mean other things such as 'On Sheet')



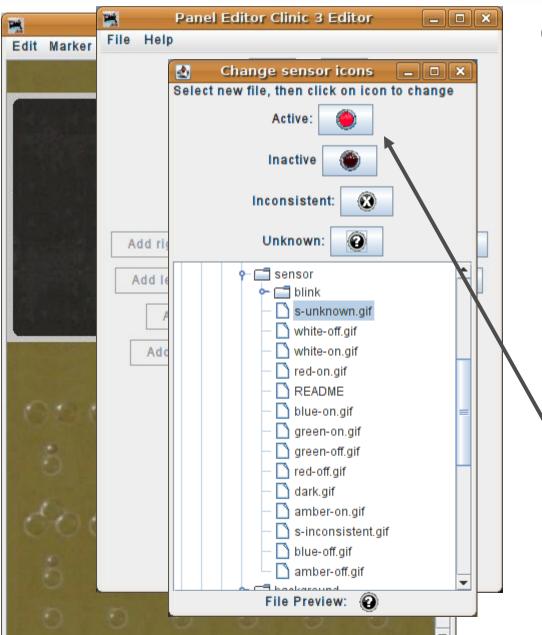
Sensor images

PR.	Panel Editor Clinic 3 Editor 📃 🗆 🗙
Edit Marker	File Help
	x: 0 y: 0
-	Set panel name
1.000	Pick background image
	Fick background image
	Add text:
	Add icon: Change icon
	Add right-hand turnout: Change icon
	Add left-hand turnout: Change icon
	Add sensor: Change icon
	Add signal head: Change icon
000	Add memory:
C.E.I	Add reporter:
	Add RPS reporter:
	Add multi-sensor
200	
(MAK)	Add Fast clock:
	Panel items popup menus active

- One of the 'rules' we have for remote operation is that we do not throw a switch under a train. To accomplish that we need to know when a train is on the switch or 'OS' (On Switch) section. (OS can mean other things such as 'On Sheet')
- In clinic 2 we added active icons for our turnouts. Now we will do the same for our occupancy sensors. Start by changing the icons.

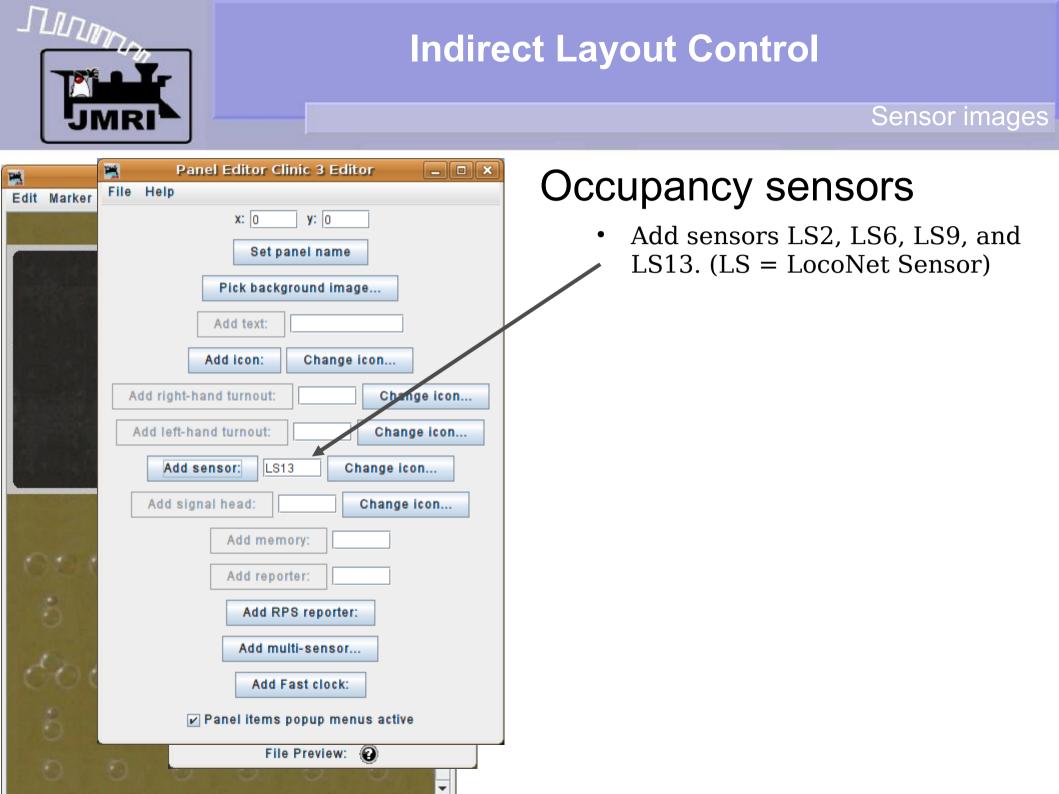


Sensor images



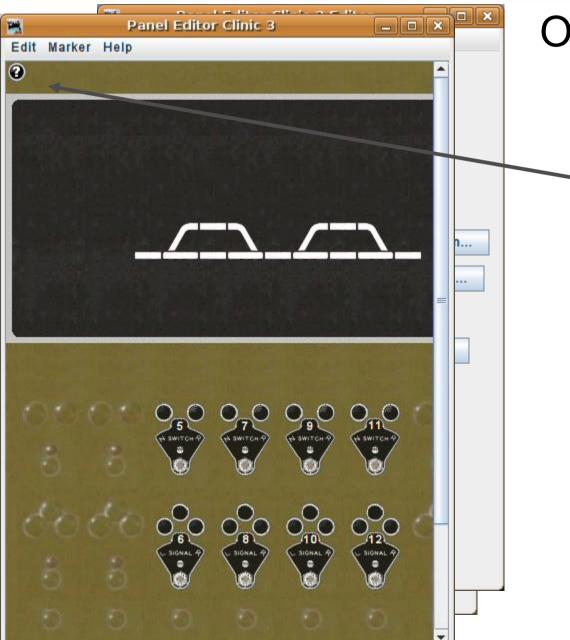
- One of the 'rules' we have for remote operation is that we do not throw a switch under a train. To accomplish that we need to know when a train is on the switch or 'OS' (On Switch) section. (OS can mean other things such as 'On Sheet')
- In clinic 2 we added active icons for our turnouts. Now we will do the same for our occupancy sensors. Start by changing the icon images.
- Navigate to 'resources' 'icons'

 'USS' 'sensor'. Many railroads used red indicator lamp jewels for the OS sections. We will do the same.





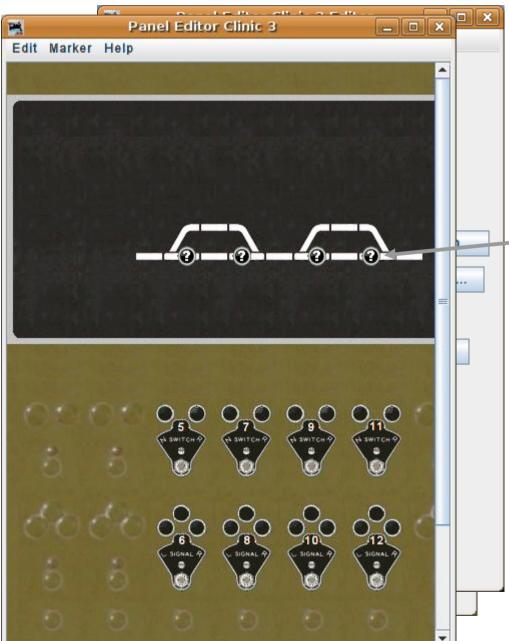
Sensor images



- Add sensors LS2, LS6, LS9, and LS13. (LS = LocoNet Sensor)
- This piles our images in the usual place.



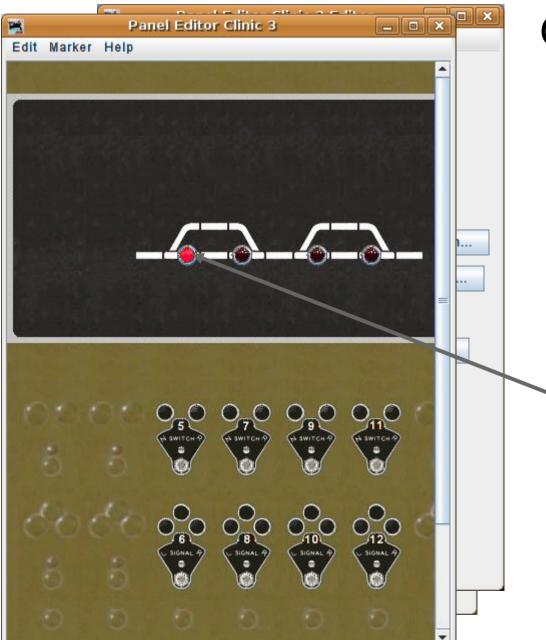
Sensor images



- Add sensors LS2, LS6, LS9, and LS13. (LS = LocoNet Sensor)
- This piles our images in the usual place.
- Move them down into their proper track locations.



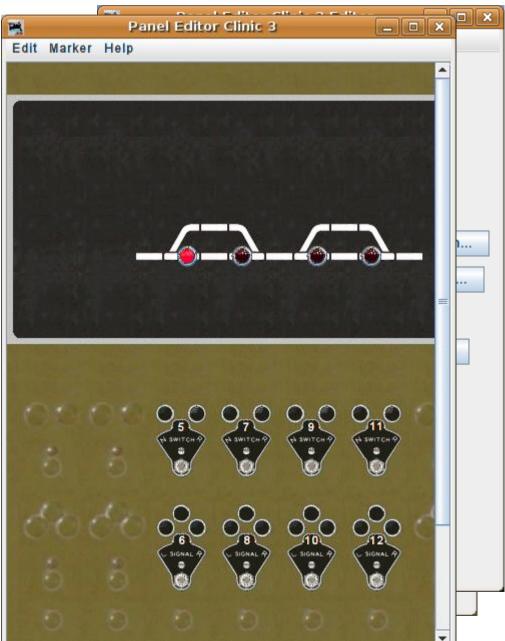
Sensor images



- Add sensors LS2, LS6, LS9, and LS13. (LS = LocoNet Sensor)
- This piles our images in the usual place.
- Move them down into their proper track locations.
- Normally we would 'disable' the sensor images so that they would only respond to our occupancy detectors. However we don't actually have any sensors attached, so we will simulate detection by clicking on our images to activate them.



Sensor images

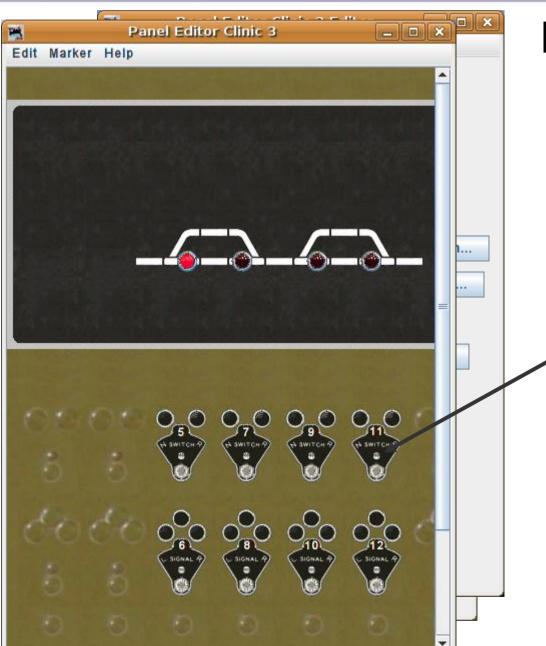


Internal sensors

• Our next concept is that of 'Internal' sensors. These are really just single bit memory devices. They react with the images just as if they were hardware, but only exist internally to JMRI.



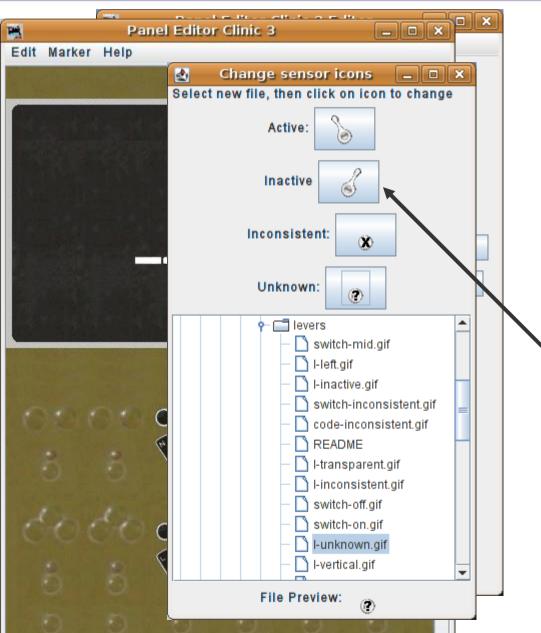
Sensor images



- Our next concept is that of 'Internal' sensors. These are really just single bit memory devices. They react with the images just as if they were hardware, but only exist internally to JMRI.
- We need some levers that are not directly attached to the turnouts like we did originally.



Sensor images



- Our next concept is that of 'Internal' sensors. These are really just single bit memory devices. They react with the images just as if they were hardware, but only exist internally to JMRI.
- We need some levers that are not directly attached to the turnouts like we did originally.
- This time we will use the lever images for our sensors.



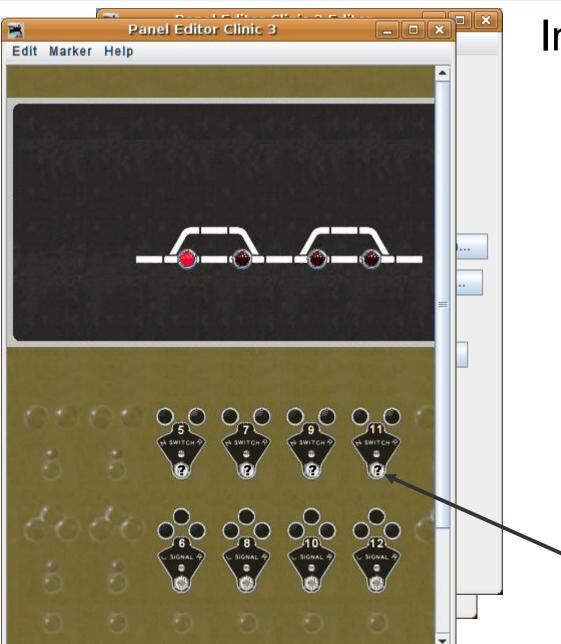
Sensor images

8	Panel Editor Clinic 3 Editor 📃 🗆 🗙
a Edit Marker	File Help
Section 21	x: 0 y: 0
_	Set panel name
	Pick background image
	Add text:
	Add icon: Change icon
	Add right-hand turnout: Change icon
	Add left-hand turnout: Change icon
	Add sensor: IS11 Change icon
And Street Street	Add signal head: Change icon
	Add memory:
	Add reporter:
	Add RPS reporter:
	Add multi-sensor
	Add Fast clock:
	Panel items popup menus active
	File Preview.
	O DO OLO OLO OL

- Our next concept is that of 'Internal' sensors. These are really just single bit memory devices. They react with the images just as if they were hardware, but only exist internally to JMRI.
- We need some levers that are not directly attached to the turnouts like we did originally.
- This time we will use the lever images for our sensors.
- Add sensors named IS5, IS7,
 IS9, and IS11 to match our plate numbers.



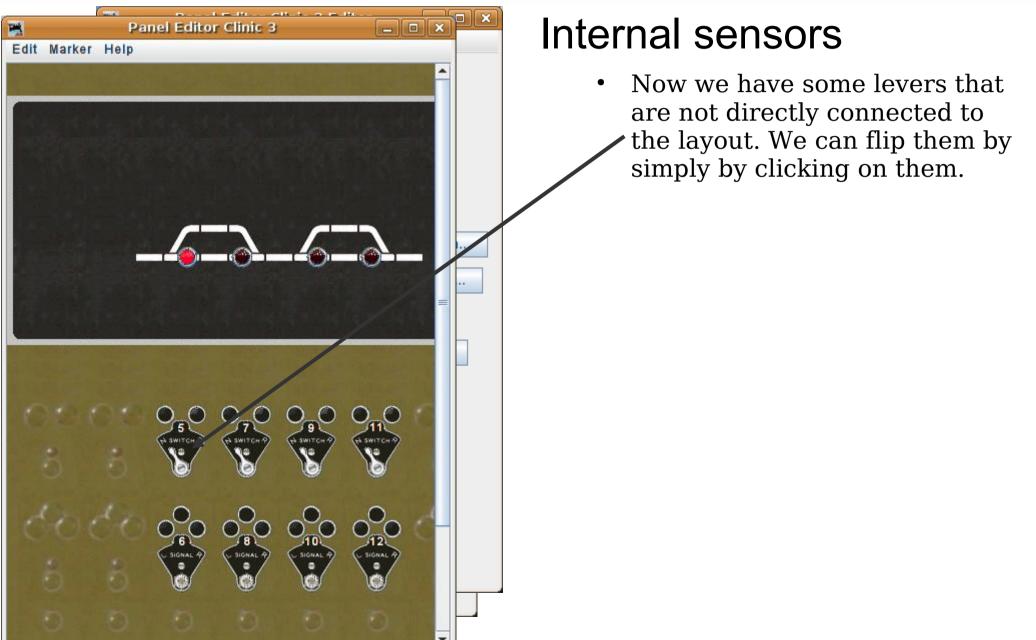
Sensor images



- Our next concept is that of 'Internal' sensors. These are really just single bit memory devices. They react with the images just as if they were hardware, but only exist internally to JMRI.
- We need some levers that are not directly attached to the turnouts like we did originally.
- This time we will use the lever images for our sensors.
- Add sensors named IS5, IS7, IS9, and IS11 to match our plate numbers.
- As before, move the new icons into their proper locations.

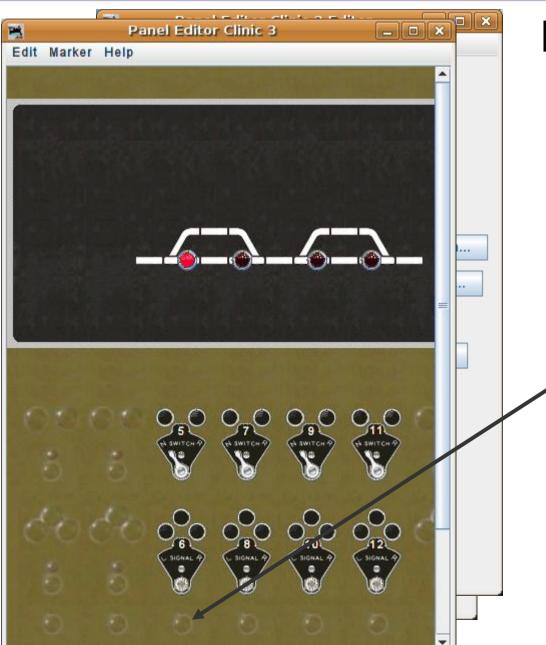


Sensor images





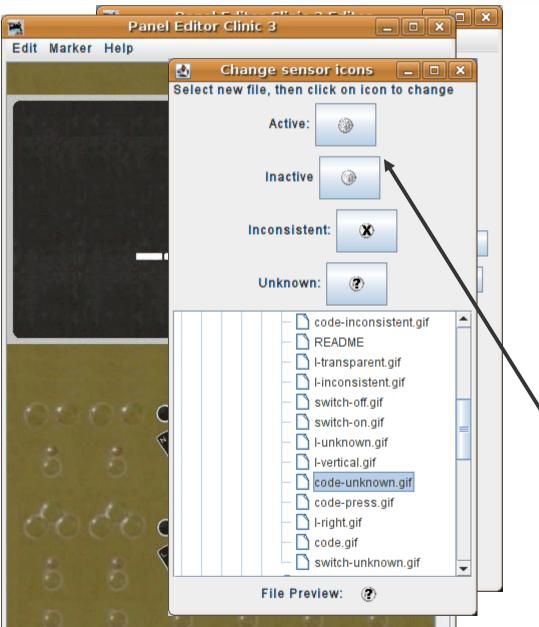
Sensor images



- Now we have some levers that are not directly connected to the layout. We can flip them by simply by clicking on them.
- The prototype CTC panel did not directly connect the lever to the switch machine. The operator moved a lever and then pressed a 'Send Code' button that encoded and sent the commands over the track side wires in a serial format using short and long pulses. (I bet you thought DCC was a new concept)



Sensor images



Internal sensors

- Now we have some levers that are not directly connected to the layout. We can flip them by simply by clicking on them.
- The prototype CTC panel did not directly connect the lever to the switch machine. The operator moved a lever and then pressed a 'Send Code' button that encoded and sent the commands over the track side wires in a serial format using short and long pulses. (I bet you thought DCC was a new concept)

Change our icons to the 'code' button images.



Sensor names

*	Panel Editor Clinic 3 Editor 📃 🗆 🗙
Edit Marker	File Help
	x: 0 y: 0
	Set panel name
PERCE	Pick background image
	Add text:
	Add icon: Change icon
0.2.1.20	Add icon. Change icon
	Add right-hand turnout: Change icon
	Add left-hand turnout: Change icon
	Add sensor: IS:P6:CB Change icon
a she was a state of the	Add signal head: Change icon
	Add memory:
ser.	
	Add reporter:
	Add RPS reporter:
	Add multi-sensor
Et.	Add Fast clock:
	Add Fast clock.
	Panel items popup menus active
	File Preview.
	O O O O

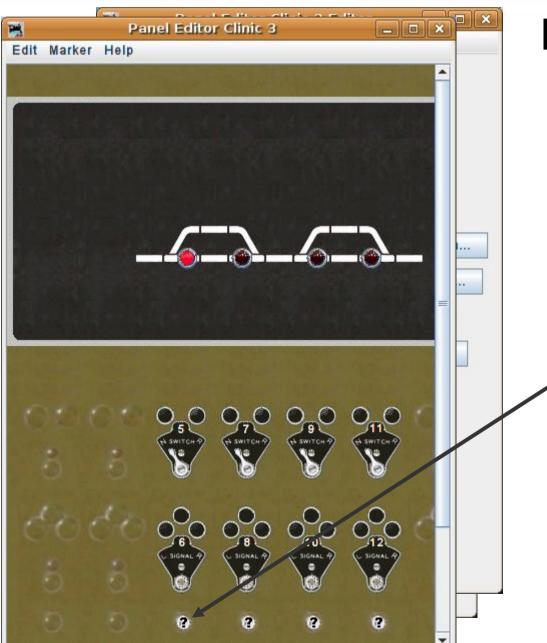
Internal sensor names

When it comes to naming our new buttons we hit a snag. We already used IS5. We could use IS6, but what about the lever for the signals in the next clinic? Maybe we should use a more descriptive name. We are not attaching to real hardware, so any name is allowed. I chose 'IS:P6:CB'. Normally system generated names use the ":" and user names should not. The plan is that a tool will generate these names, so I use that as an excuse for including the ":".

IS = Internal Sensor, :P6 = Plant 6, :CB = Code Button.



Sensor names

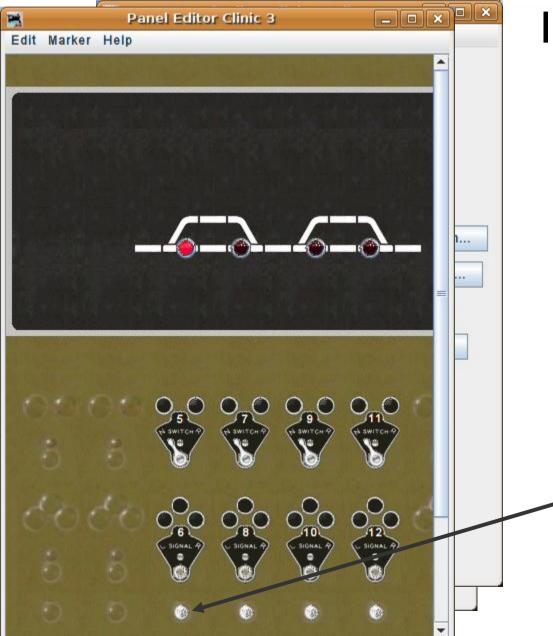


Internal sensor names

- Add 'IS:P6:CB', 'IS:P8:CB', 'IS:P10:CB', and 'IS:P12:CB' for our new buttons.
- Actually the code button was normally at the bottom of each column of switches and levers. However that is off the bottom of this portion of our image, so I cheated and placed the
 buttons here. You should do it right and reserve these for 'maintenance call' or 'call on' switches or lamps.



Sensor names



Internal sensor names

- Add 'IS:P6:CB', 'IS:P8:CB', 'IS:P10:CB', and 'IS:P12:CB' for our new buttons.
- Actually the code button was normally at the bottom of each column of switches and levers. However that is off the bottom of this portion of our image, so I cheated and placed the buttons here. You should do it right and reserve these for 'maintenance call' or 'call on' switches or lamps.
- Clicking on these images reveals that the first click pushes the button and the next click releases it. The actual code button is a spring return.



Sensor names

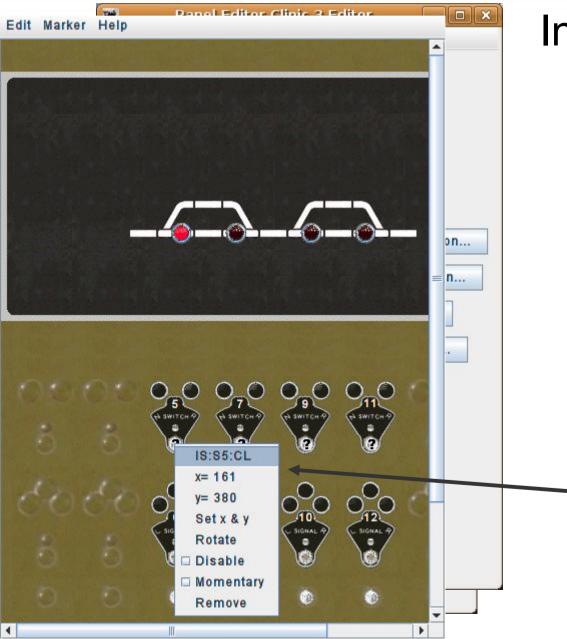


Internal sensor names

To solve the 'momentary' problem by simply right clicking on each icon to bring up its tools, then check the box called 'Momentary'. Further testing will show that the buttons will now directly follow your mouse clicks. Note, if the button pops up when you click on it, you have simply reversed the images. To correct them change the image icons and reenter the buttons.



Sensor names

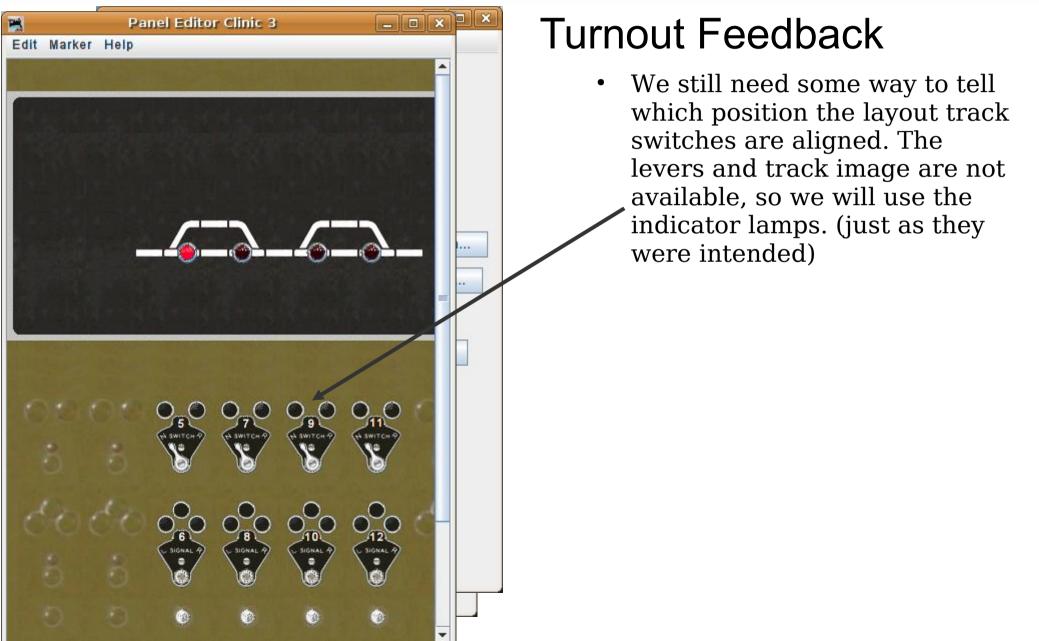


Internal sensor names

- To solve the 'momentary' problem by simply right clicking on each icon to bring up its tools, then check the box called 'Momentary'. Further testing will show that the buttons will now directly follow your mouse clicks. Note, if the button pops up when you click on it, you have simply reversed the images.
- While we are correcting things lets also use some better names for the levers. I used **IS:S5:CL**.
 - **IS** = **I**nternal **S**ensor,
 - **:S5** = **S**witch **5**,
 - **:CL** = **C**ontrol Lever.

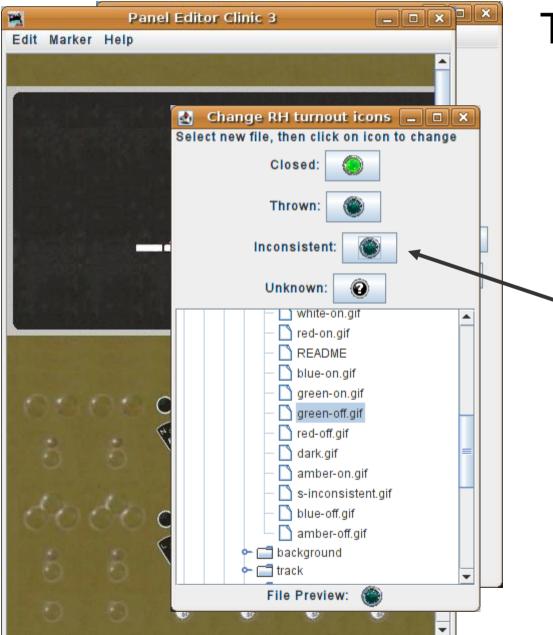
'Remove' the originals and add in the new replacements.







Turnout Feedback



- We still need some way to tell which position the layout track switches are aligned. The levers and track image are not available, so we will use the indicator lamps. (just as they were intended)
- Change our turnout icons to be green jewels. Note: Use the 'green-off.gif' for the 'Inconsistent:' position.



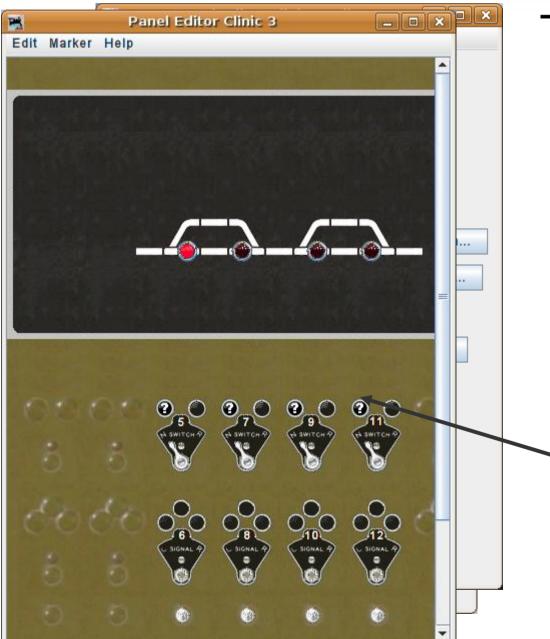
Turnout Feedback

*	🎽 🛛 Panel Editor Clinic 3 Editor 📃 🗖 🗙
Edit Marker	File Help
3	x: 0 y: 0
	Set panel name
1 BROKE	Pick background image
	Add text:
	Add icon: Change icon
	Add right-hand turnout: LT4 Change icon
	Add left-hand turnout:
	Add left-hand turnout:
	Add sensor: IS:S11:CL Change icon
	Add signal head: Change icon
-	Add memory:
OBU	Add reporter:
	Add RPS reporter:
	Add multi-sensor
600	Add East slask
	Add Fast clock:
8	Panel items popup menus active
	File Preview:

- We still need some way to tell which position the layout track switches are aligned. The levers and track image are not available, so we will use the indicator lamps. (just as they were intended)
- Change our turnout icons to be green jewels. Note: Use the 'green-off.gif' for the 'Inconsistent:' position.
- Add LT1, LT2, LT3, and LT4.



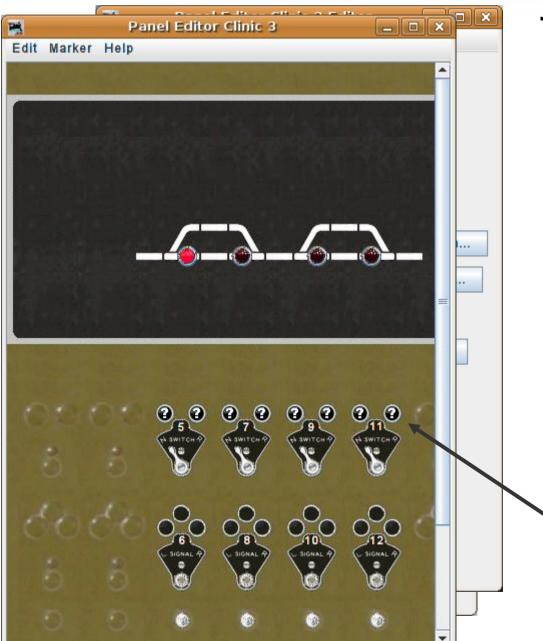
Turnout Feedback



- We still need some way to tell which position the layout track switches are aligned. The levers and track image are not available, so we will use the indicator lamps. (just as they were intended)
- Change our turnout icons to be green jewels. Note: Use the 'green-off.gif' for the 'Inconsistent:' position.
- Add LT1, LT2, LT3, and LT4.
- Move them into place on the 'N' (Normal) side of the plates.



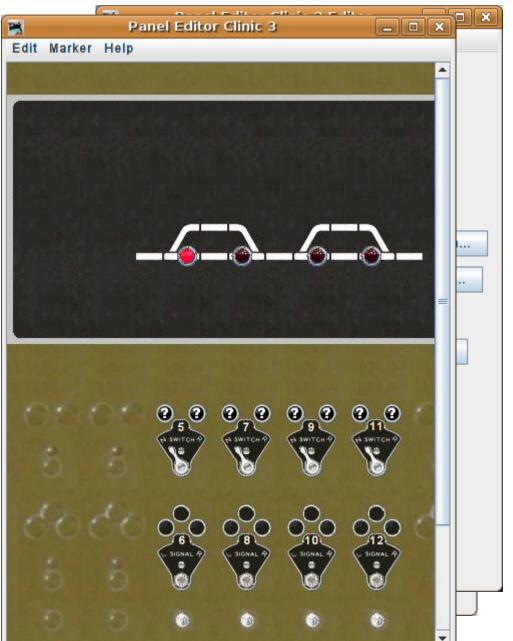
Turnout Feedback



- We still need some way to tell which position the layout track switches are aligned. The levers and track image are not available, so we will use the indicator lamps. (just as they were intended)
- Change our turnout icons to be green jewels. Note: Use the 'green-off.gif' for the 'Inconsistent:' position.
- Add LT1, LT2, LT3, and LT4.
- Move them into place on the 'N' (Normal) side of the plates.
- Change the jewel colors to amber and do the same for the 'Reverse' lamps. (remember 'Thrown' is lit for them)







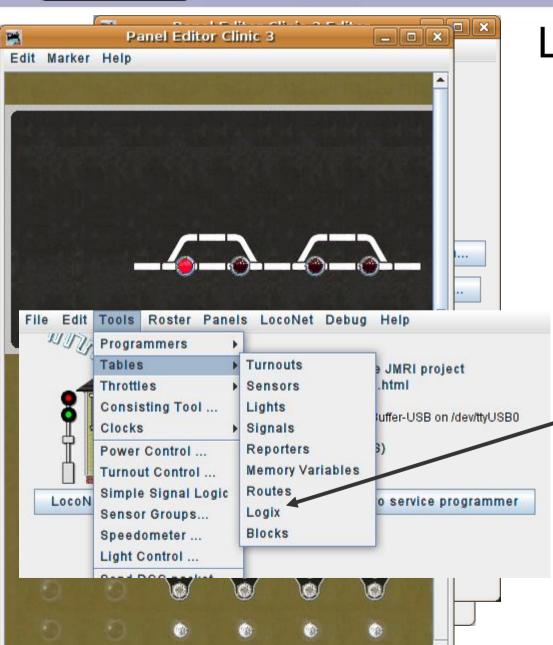
Logix

We now have all our required inputs and outputs on the panel. All that is missing is the logic to make it work. Our first example will be simple:

If the Control Lever is changed And the OS is NOT occupied And the Code Button is pressed Then send a turnout command.



Logix



Logix

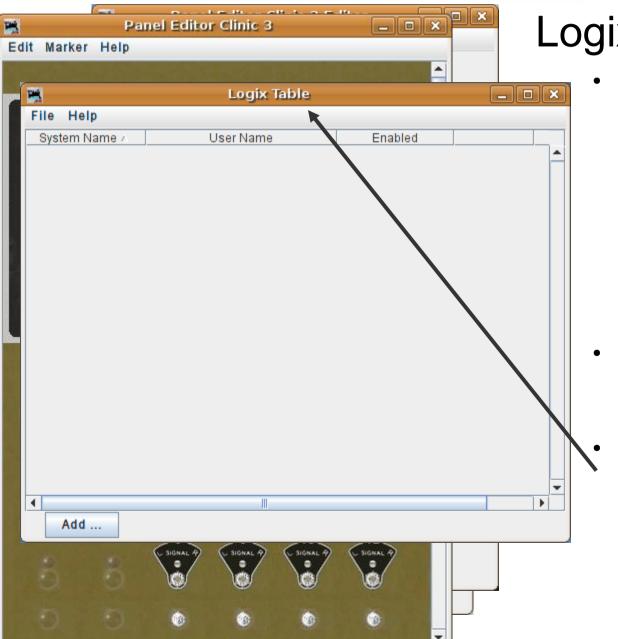
We now have all our required inputs and outputs on the panel. All that is missing is the logic to make it work. Our first example will be simple:

If the Control Lever is changed And the OS is NOT occupied And the Code Button is pressed Then send a turnout command.

 To open the Logix tools navigate from the main window via 'Tools' – 'Tables' – 'Logix'.



Logix



Logix

- We now have all our required inputs and outputs on the panel. All that is missing is the logic to make it work. Our first example will be simple:
 - If the Control Lever is changed And the OS is NOT occupied And the Code Button is pressed Then send a turnout command.
- To open the Logix tools navigate from the main window via 'Tools' - 'Tables' - 'Logix'.
- This brings up an empty Logix table.



Logix

We now have all our required

panel. All that is missing is the

logic to make it work. Our first

If the Control Lever is changed

And the Code Button is pressed Then send a turnout command.

navigate from the main window

via 'Tools' - 'Tables' - 'Logix'.

Click on 'Add ...' to create a

new Logix entry.

This brings up an empty Logix

And the OS is NOT occupied

inputs and outputs on the

example will be simple:

To open the Logix tools

table.

Ed	it Marker Help	anel Editor Clinic 3		Logix
	*	Logix Table	_	
	File Help			• V in p le
	System Name 🔺	User Name	Enabled	p p
				e
				A
				Г • П
				n
				v
				г • Г
				▼
	•			• (
	Add			• C
		SIGNAL A	SIGNAL P	1.



Logix

Edit Ma	Pa Pa arker Help	nel Editor Clinic 3		🖹 Lo	ogi	X
	Help tem Name ×	Logix Table User Name	Enabled		•	We now have all our required inputs and outputs on the panel. All that is missing is the logic to make it work. Our first example will be simple:
						If the Control Lever is changed And the OS is NOT occupied And the Code Button is pressed Then send a turnout command.
		Add Logix System Name			•	To open the Logix tools navigate from the main window via 'Tools' – 'Tables' – 'Logix'. This brings up an empty Logix
•		ter system name and user n reate Logix, then add Condi Create Logix Cancel		•	•	table. Click on 'Add' to create a new Logix entry.
	0				•	Fill in the required information.

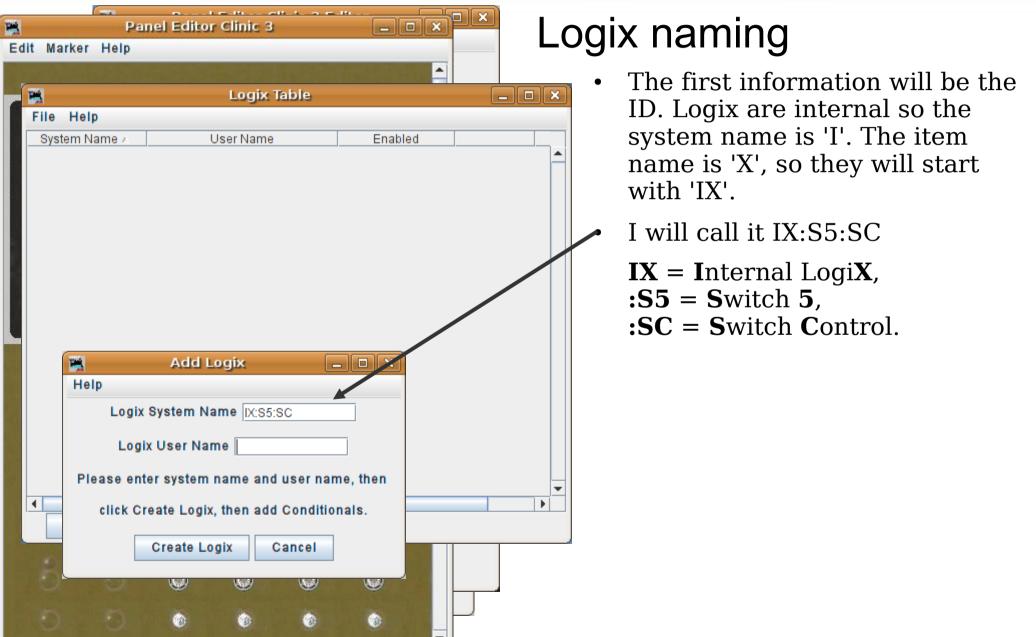


Logix naming

Panel Editor Clinic 3	Logix naming
► Logix Table File Help System Name ▲ User Name Enabled	 The first information will be the ID. Logix are internal so the system name is 'I'. The item name is 'X', so they will start with 'IX'.
Add Logix Help Logix System Name Logix User Name Please enter system name and user name, then click Create Logix, then add Conditionals. Create Logix Cancel	



Logix naming





Logix naming

Panel Editor Clinic 3	Logix naming
Logix Table - File Help System Name ▲ User Name Enabled	• The first information will be the ID. Logix are internal so the system name is 'I'. The item name is 'X', so they will always start with 'IX'.
	Lets call it IX:S5:SC
	IX = Internal LogiX, :S5 = Switch 5, :SC = Switch Control.
Add Logix Help Logix System Name Logix User Name Switch 5 Control Please enter system name and user name, then click Create Logix, then add Conditionals. Create Logix Cancel	 The 'Logix User Name' may be any description we choose to use. 'Switch 5 Control' for example.



Logix naming

		-	Pa	nel Edito	r Clinic :	3	m.10.		X			\mathbf{c}
Ed	it Ma	rker He	lp									J
									^			•
r	1				Logix	Table				L	. • ×	
	File	Help										
	Syst	em Name	Δ		User Nam	e		Enable	ed		_	
												•
		-		Add I	ogix			I X				•
		Help										
		L	.ogix	System N	ame IX:S	5:SC						
			Logi			h E O antro	ul l					
			Logi	(User Na	me Switc	n 5 Contro						
		Please	e ent	er system	name ar	nd user r	name, t	hen			-	
	•	clic	ck Cr	eate Logi)	x, then ad	ld Condi	tionals					
b												
	-			Create Lo	ogix	Cancel						4
		100	0	w/	W	W		e l				
				0	-	e la						
				۲	۲	۲		0				
										1		

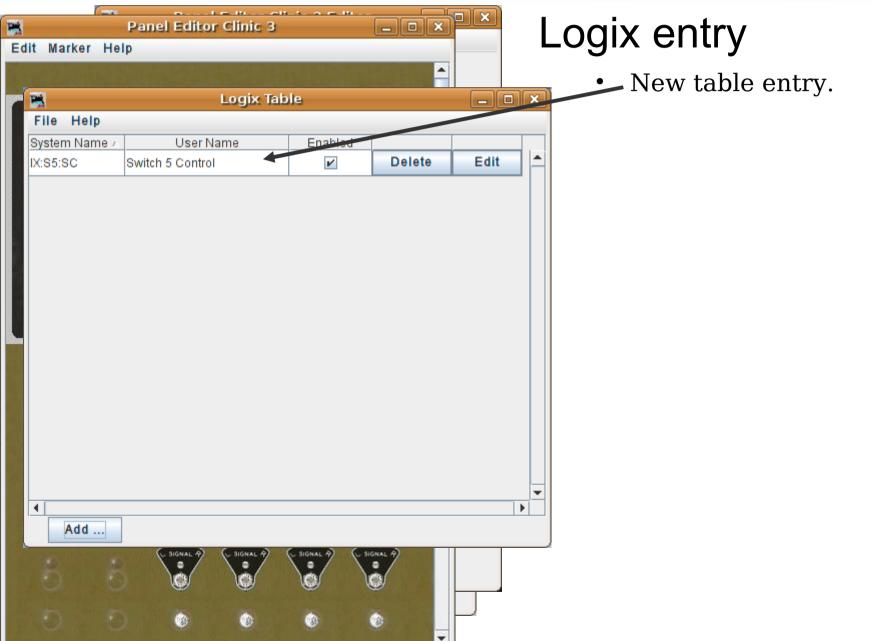
_ogix naming

- The first information will be the ID. Logix are internal so the system name is 'I'. The item name is 'X', so they will always start with 'IX'.
- Lets call it IX:S5:SC

IX = Internal LogiX, :S5 = Switch 5, :SC = Switch Control.

- The 'Logix User Name' may be any description we choose to use. 'Switch 5 Control' for example.
- Once we have named our new
 creation click on 'Create Logix' to add it to the table and bring up a blank entry window.







Edit Marker H	Panel Editor Clinic 3 Telp		La La	ogix entry	
	Edit Logix		_ _ X	• New table entry.	
Help	Logix System Name I Logix User Name Switch 5 Control	X:85:8C		Edit Logix windo	W.
	Conditionals (in Order of Cal	ulation, max 50)			
System Name	User Name	State			
	New Conditional Reords	r Calculate			
	Done Delete	Logix			



Logix entry

R Edit	Marker	1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 -	anel Editor Clinic 3		Log
1			Edit Logix		- • •
Help					-
			Logix System Name I	X:85:8C	
		Logix	User Name Switch 5 Control		•
			onditionals (in Order of Calc		
Sy	stem Nam	ie	User Name	State	
		N	ew Conditional Reorde	r Calculate	
			Done Delete	Logix	

- New table entry.
 - Edit Logix window.
 - Each Logix will contain one or more 'Conditionals' or things that may be true or false. A 'Conditional' may optionally do one or two actions when it becomes true or becomes false or simply changes state.



Logix entry

R	Panel Editor Clinic 3		
Edit Marker	r Help		LOG
1	Edit Logix		• • × •
Help			
	Logix System Name I	X:85:8C	•
	Logix User Name Switch 5 Control		
	Conditionals (in Order of Calo	culation, max 50)	
System Nar	me User Name	State	
	New Conditional Reorde	er Calculate	
	Done Delete	Logix	

- New table entry.
 - Edit Logix window.
 - Each Logix will contain one or more 'Conditionals' or things that may be true or false. A 'Conditional' may optionally do one or two actions when it becomes true or becomes false or simply changes state.
 - Click the 'New Conditional' button to bring up the 'Edit Conditional' window.

Logix entry

22	JUNICI	209.70
1	Edit Conditional	
Help		
	Conditional System Name IX:S5:SCC1	
-Logi	Conditional User Name	e entry.
LUgi		k window.
		ix will contain one or nditionals' or things be true or false. A nal' may optionally do o actions when it true or becomes false changes state.
Actio	Add State Variable Check State Variables	'New Conditional' bring up the 'Edit al' window.
	Action 1 - Type None Action 2 - Trigger Action On Change To True On Change To False On Change Action 2 - Type None	• Note that JMRI automatically added 'C1' to the name we gave this item.
	Update Conditional Cancel Delete Conditional	

SUUMIN

JMIN	
Edit Conditional	_ _ ×
Help	
Conditional System Name IX:S5:SCC1	
Conditional User Name Switch 5 Normal	
Logical Expression	
State Variables (max 20)	
Variable Type Name Data 1 Data 2 State T	riggers Calculation
Add State Variable Check State Variables	
Actions	
Action 1 - Trigger Action 💿 On Change To True 🔵 On Change To False 🔵 O	On Change
Action 1 - Type None 💌	
	Dr. Change
Action 2 - Trigger Action 🖲 On Change To True 🔾 On Change To False 🔾 O	on change
Action 2 - Type None 👻	
Update Conditional Cancel Delete Conditional	

SUUMAN

Logix entry

Logix entry

We will create two conditionals, one to throw switch 5 and one to close it.
Name this first one "Switch 5 Normal".



Logix entry

Edit Conditional
Help
Conditional System Name IX:S5:SCC1
Conditional User Name Switch 5 Normal
Logical Expression
State Variables (max 20)
Variable Type Name Data 1 Data 2 State Triggers Calculation
Add State Variable Check State Variables
Actions
Action 1 - Trigger Action 💿 On Change To True 🔘 On Change To False 🔘 On Change
Action 1 - Type None 🗸
Action 2 - Trigger Action 🖲 On Change To True \ominus On Change To False \ominus On Change
Action 2 - Type None 💌
Update Conditional Cancel Delete Conditional

- We will create two conditionals, one to throw switch 5 and one to close it. Name this first one "Switch 5 Normal".
- We call the various items that will be checked 'Variables' because they vary between one value and another. In this case between being true and being false. Click here to add our first one.

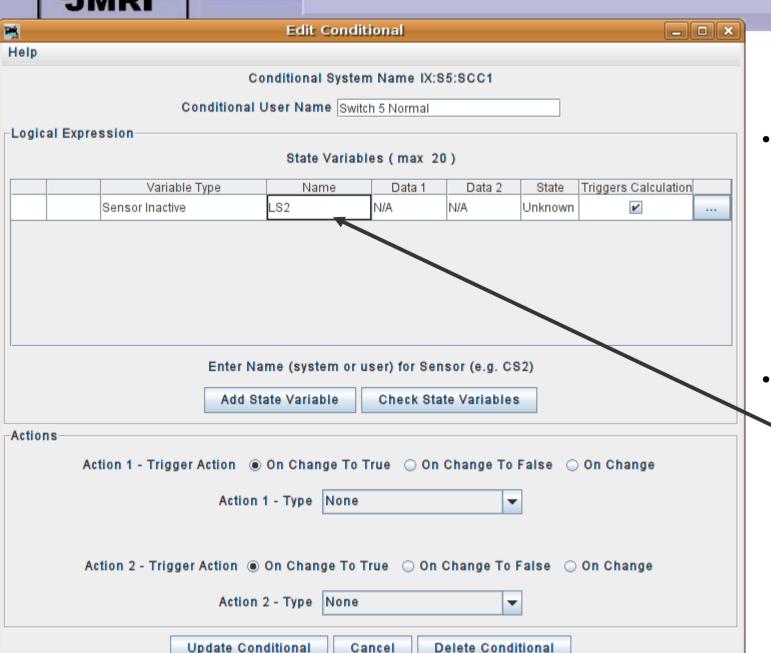


*			Edit Condi	ional		_	
Help							
		Co	nditional Syster	n Name IX:8	5:SCC1		
	Condition	al I	Jser Name Swite	h 5 Normal			
			Sel Name Switt	in 5 Normai			
Logical E	xpression						
			State Variab	les (max 2	0)		
	Variable Type	_	Name	Data 1	Data 2	State Triggers Calculation	1
	Sensor Inactive	•		N/A	N/A	Unknown	
	Sensor Active						
	Sensor Inactive						
	Turnout Thrown Turnout Closed						
	Conditional True						
	Conditional False						
	Light On						
	Light Off	-					
	Enter	iva	me (system or u	iser) for Ser	isor (e.g. C	82)	
	Add	St	ate Variable	Check Sta	te Variable	s	
Actions—							
	Action 1 - Trigger Action	۲	On Change To ⁻	Frue 🔾 On	Change To	False 🔾 On Change	
	Acti	on	1-Type None			•	
	Action 2 - Trigger Action	۲	On Change To T	"rue 🔾 On	Change To	False 🔾 On Change	
	Acti	on :	2 - Type None		`		
]
	Update C	on	ditional Ca	incel D	elete Cond	litional	

Logix entry

Logix entry

 Click in the variable box to call up a list of possible options. Choose 'Sensor Inactive' because we don't want this to happen if a train is on the OS.



<u>IIIImm</u>

Logix entry

- Click in the variable
 box to call up a list
 of possible options.
 Choose 'Sensor
 Inactive' because
 we don't want this
 action to happen if
 a train is on the OS.
- Enter the OS sensor ID. In this case it is 'LS2'.

	Edit Conditional
elp	
	Conditional System Name IX:S5:SCC1
	Conditional User Name Switch 5 Normal
	pression
Ugical Ex	
	State Variables (max 20)
	Variable Type Name Data 1 Data 2 State Triggers Calculation
	Sensor Inactive LS2 N/A N/A Unknown 🗹
	Enter Name (system or user) for Sensor (e.g. CS2) Add State Variable Check State Variables
ctions—	
	Action 1 - Trigger Action 💿 On Change To True 🔾 On Change To False 🔍 On Change
	Action 1 - Type None 🔽
	Action 2 - Trigger Action () On Change To True () On Change To False () On Change
	Action 2 - mgger Action (e) on onange to rule () on onange to raise () on onange
	Action 2 - Type None 👻
	Update Conditional Cancel Delete Conditional

SUUMAN

Logix entry

- Click in the variable
 box to call up a list
 of possible options.
 Choose 'Sensor
 Inactive' because
 we don't want this
 action to happen if
 a train is on the OS.
- Enter the OS sensor ID. In this case it is 'LS2'.
- Click on 'Check State Variables' to let the system read the newly added sensor item.

	JMIKI
1	Edit Conditional
lelp	
	Conditional System Name IX:S5:SCC1
	Conditional User Name Switch 5 Normal
ogica	Il Expression
	State Variables (max 20)
	Variable Type Name Data 1 Data 2 State Triggers Calculation
	Sensor Inactive LS2 N/A N/A False 🗹
	All state variables are OK.
	Add State Variable Check State Variables
ction	s
	Action 1 - Trigger Action 💿 On Change To True 🔘 On Change To False 🔘 On Change
	Action 1 - Type None
	Action 1 - Type None 🔽

TUTMIN

Logix entry

- Click in the variable
 box to call up a list
 of possible options.
 Choose 'Sensor
 Inactive' because
 we don't want this
 action to happen if
 a train is on the OS.
- Enter the OS sensor ID. In this case it is 'LS2'.
- Click on 'Check State Variables' to let the system read the newly added sensor item.
- Note that it is False.

	F		In	dire	ct L	ayoı	ut Co	on	trol
JMR	XI 🏊 🛌								Logix entry
*		Edit Condi	ional					×	
Help		conditional System User Name Swite		5:8001					Logix entry
-Logical Expression		State Variab	les (max 20						Add the second
	Mariahla Tura		-	-	Otata	Trianana Oak			— variable for the
Senso	Variable Type or Inactive	LS2	Data 1 N/A	Data 2 N/A	State False	Triggers Cak			lever. (IS:S5:CL)
AND					Unknown	L.			
		ariable Type, ther State Variable	n enter requir Check Stat						
Actions									
Action 1	- Trigger Action () Action	On Change To ⁻	「rue ⊖ On (oFalse 🔾	On Change	•		
Action 2	- Trigger Action (Action) On Change To T n 2 - Type None	'rue ⊖ On C	_	False 🔾	On Change			
	Update Co	nditional Ca	ancel De	elete Con	ditional				

ן ר ן			lr	ndire	ct L	ayout C	onti	
	'JMRI-							Logix entry
Help		Edit Cond	ditional					
TICIP	c	onditional Syst	em Name IX	:S5:SCC1				Logix entry
		User Name Sw						0 ,
-Logica	al Expression		ibles (max	20)			•	Add the second variable for the
	Variable Type	Name	Data 1	Data 2	State	Triggers Calculation		lever. (IS:S5:CL)
AND	Sensor Inactive Sensor Active	LS2	N/A N/A	N/A N/A	False Unknown			
		lame (system ol State Variable	1	ensor (e.g. C tate Variable				the 'N' (Normal) position.
-Action	Action 1 - Trigger Action) On Change To 1 - Type Non		n Change To) On Change		

Delete Conditional

Update Conditional

Cancel

1			Edit Cond	ditional				
lelp								
			Conditional Syst	em Name IX:	S5:SCC1			
		Condition	al User Name Sw	vitch 5 Normal				
Logic	al Expr	ression						
			State Varia	ables (max 💈	20)			
		Variable Type	Name	Data 1	Data 2	State	Triggers Calculation	
		Sensor Inactive	LS2	N/A	N/A	False	2	
AND		Sensor Active	IS:S5:CL	N/A	N/A	True	×	
AND	NOT	Turnout Closed	LT1	N/A	N/A	Unknown	×.	
	· · · · · · · · · · · · · · · · · · ·	Enter	Name (system or	user) for Tur	mout (e.g. L1	[12]		
		Enter	Name (system or	user) for Tur	mout (e.g. L1	Γ12)		
			Name (system or I State Variable	- 1 [nout (e.g. Lī ate Variable			
				- 1 [
Actio		Add	l State Variable	Check St	ate Variable	S		
Actio			l State Variable	Check St	ate Variable	S	On Change	
Actio		Add	l State Variable	Check St	ate Variable	s False O	On Change	
Action		Add	State Variable	Check St	ate Variable n Change To	s False O	On Change	
Actio	A	Add Action 1 - Trigger Action Acti	• On Change To on 1 - Type Non	Check St	ate Variable n Change To	s False O		
Actio	A	Add	• On Change To on 1 - Type Non	Check St	ate Variable n Change To	s False O		
Actio	A	Add Action 1 - Trigger Action Action Action 2 - Trigger Action	• On Change To on 1 - Type Non	Check St o True O Or e o True O Or	ate Variable n Change To	s False O False O		
Actio	A	Add Action 1 - Trigger Action Action Action 2 - Trigger Action	State Variable On Change To on 1 - Type Non On Change To	Check St o True O Or e o True O Or	ate Variable n Change To n Change To	s False O False O		

Logix entry

- Add the second variable for the lever. (IS:S5:CL)
- It is 'Active' when in the 'N' (Normal) position.
- We only want to send a command if the turnout is NOT already in position. This is not the same as being 'Thrown' because it could be moving or unknown.

	JMR							
1			Edit Cond	litional				
Help								
			Conditional System	em Name IX	(:S5:SCC1			
		Condi	itional User Name Sw	itch 5 Normal				
Logi	al Expression							
			State Varia	bles (max	20)			
		Variable Type	e Name	Data 1	Data 2	State	Triggers Calculation	
	Senso	or Inactive	LS2	N/A	N/A	False	v	
AND	Senso	or Active	IS:S5:CL	N/A	N/A	True	r	
AND	NOT Turno	ut Closed	LT1	N/A	N/A	False	~	
			Add State Variable	Check S	tate Variable	S		
Actio	ns							
	Action 1	- Trigger Acti	ion 💿 On Change To	True 🔾 O	n Change To	False (On Change	
			Action 1 - Type None			_		
			Action 1 - Type None	,				
	A ship n O			True 🔾 O	n Change To	False (On Change	
	Action 2	- Trigger Act	tion 🖲 On Change To					
	Action 2		Action 2 - Type None			-		

SUUM

Logix entry

- Add the second variable for the lever. (IS:S5:CL)
- It is 'Active' when in the 'N' (Normal) position.
- We only want to send a command if the turnout is NOT already in position.
 This is not the same as being 'Thrown' because it could be moving or unknown.
- A state check shows that 'Not closed' is 'False' because it is closed. (not not)

Edit Conditional	x entry
Halp	
Help Logix ontr	1
Conditional System Name IX:S5:SCC1 LOgix entr	у
Logical Expression • Now add our	code
Variable Type Name Data 1 Data 2 State Triggers Calculation Sensor Inactive LS2 N/A N/A False	
AND Sensor Active IS:S5:CL N/A N/A Trate M	
AND NOT Turnout Closed LT1 N/A N/A False	
AND Sensor Active IS:P6:CB	
Error found, please correct and try again. Add State Variable Check State Variables	
Actions Action 1 - Trigger Action On Change To True On Change To False On Change Action 1 - Type None Action 2 - Trigger Action On Change To True On Change To False On Change Action 2 - Type None Update Conditional Cancel Delete Conditional	



		JMRI							Logix entry
-			Edit Cond	itional			_ - ×		
He	lp							1	l agiv aptru
			Conditional Syste	m Name IX:	S5:SCC1				Logix entry
		Cor	nditional User Name Swi	tch 5 Normal					
-Lo	gical	Expression	·						Norrada orrado
	-		State Varial	oles (max 2	20)			•	Now add our code button.
		Variable T	ype Name	Data 1	Data 2	State	Triggers Calculation		
		Sensor Inactive	LS2	N/A	N/A	False	₽	•	Note: If we attempt
AN	ID	Sensor Active	IS:S5:CL	N/A	N/A	True	₽		to check the state
AN		OT Turnout Closed	LT1	N/A	N/A	False	✓ …		before moving out
AN	ID	Sensor Active	IS:P6:CB	N/A	N/A	Unknown	✓ …		of the entry box we
		· · ·		_					5
									will get an error.
			Error found, please	e correct and	l try again.				
			Add State Variable	Check St	ate Variable	s			
A	-				Error				×
	X	The name "" does	not match an existing se reate as required, and tr		ə: lf you were	editing in	n table, click in State colur	nn and	d try again.
				[ОК				
		Up	odate Conditional C	ancel	Delete Cond	itional			



	J							
<u>í</u>			Edit Cond	litional				
lelp								
			Conditional Syst	em Name IX:	S5:SCC1			
		Conditio	nal User Name Sw	vitch 5 Normal				
_ogi	cal Expr	ession						
			State Varia	ibles (max 🔅	20)			
		Variable Type	Name	Data 1	Data 2	State	Triggers Calculation	
		Sensor Inactive	LS2	N/A	N/A	False	2	
AND		Sensor Active	IS:S5:CL	N/A	N/A	True	~	
AND	NOT	Turnout Closed	LT1	N/A	N/A	False	~	
AND		Sensor Active	IS:P6:CB	N/A	N/A	Unknown		
			Error found, pleas		,, again.			
		Ad	d State Variable	Check St	ate Variable	s 🔶		
Actio	ns							
	A	action 1 - Trigger Action	On Change To	o True 🔾 Or	n Change To	False 🤇) On Change	
						_		
		Act	tion 1 - Type None	9		•		
	,	Action 2 - Trigger Action) 🖲 On Change To	True 🔾 Or	Change To	False C	On Change	
						_		
		Act	tion 2 - Type None	e		•		
		Undata	Conditional	Cancel	Delete Cond	ditional		
		Opdate	Conditional	Cancer	Delete Cont	annonal		

Logix entry

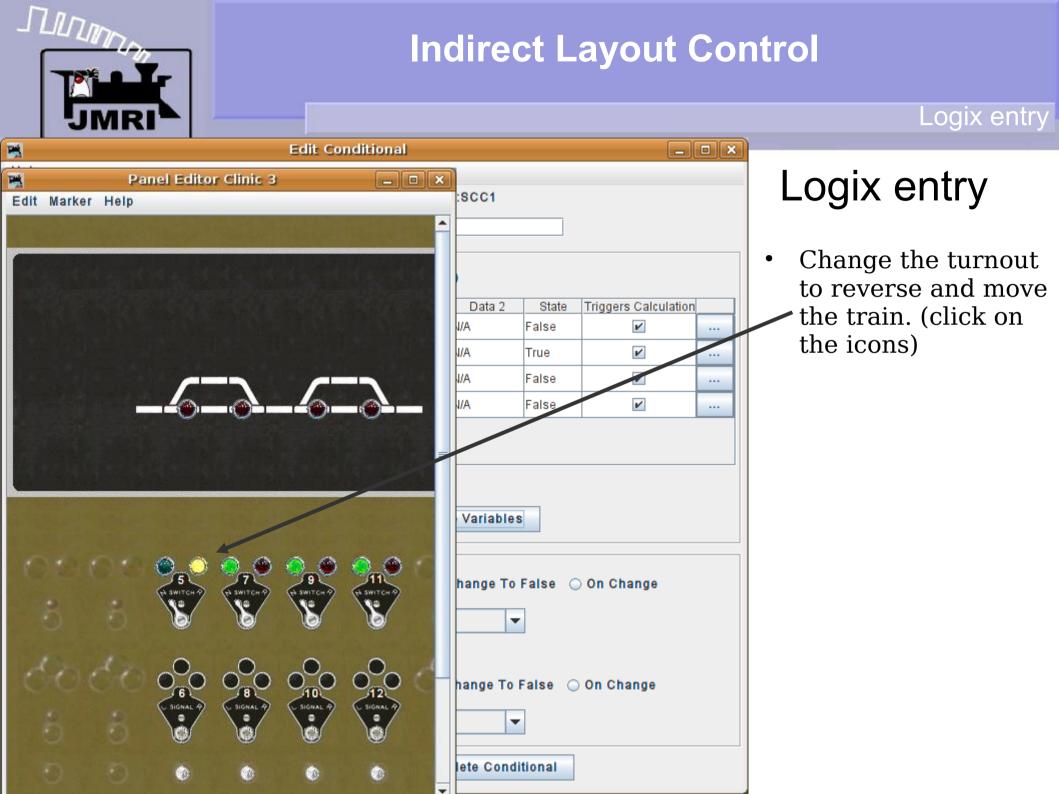
- Now add our code button.
- Note: If we attempt to check the state before moving out of the entry box we will get an error.
- Simply click in the state column box first then check it.



	J							
1			Edit Cond	litional				
Help								
			Conditional Syst	em Name IX:	S5:SCC1			
		Condition	al User Name Sw	vitch 5 Normal				
Logi	cal Expr	ession						
-	-		State Varia	ables (max 2	20)			
		Variable Type	Name	Data 1	Data 2	State	Triggers Calculation	
		Sensor Inactive	LS2	N/A	N/A	False	~	
AND		Sensor Active	IS:S5:CL	N/A	N/A	True	N	
AND	NOT	Turnout Closed	LT1	N/A	N/A	False	Ľ	
AND		Sensor Active	IS:P6:CB	N/A	N/A	False		
			All state va	ariables are (DK.			
		Add	State Variable	Check St	ate Variable	S		
Actio	ns							
	А	ction 1 - Trigger Action	On Change To	o True 🔾 Or	Change To	False (On Change	
						_		
		Actio	on 1 - Type None	e	•	-		
	4	Action 2 - Trigger Action	On Change To		Change To	False () On Change	
	,	ingger/info			change to		, on onango	
		Actio	on 2 - Type None	е	•	-		
				Domost 1	Delete Ca			
		Update C	onditional	Cancel	Delete Conc	litional		

Logix entry

- Now add our code button.
- Note: If we attempt to check the state before moving out of the entry box we will get an error.
- Simply click in the state column box first then check it.
- As you can see we have piled up many reasons that we will not send a turnout command at this time, even if we do press the button.





	J							
1			Edit Conc	ditional				
Help								
			Conditional Syst	tem Name IX:	S5:SCC1			
		Condition	al User Name Sw	vitch 5 Normal				
Logi	cal Expr		·					
-			State Varia	ables (max 2	20)			
		Variable Type	Name	Data 1	Data 2	State	Triggers Calculation	
		Sensor Inactive	LS2	N/A	N/A	True		
AND		Sensor Active	IS:S5:CL	N/A	N/A	True		
AND	NOT	Turnout Closed	LT1	N/A	N/A	True		
AND		Sensor Active	IS:P6:CB	N/A	N/A	False		
			All state va	ariables are (рк.			
		Add	State Variable	Check St	ate Variable	S		
Actio	ns							
		ation 1 Trigger Action	© On Change T			Ealaa (On Change	
		ction 1 - Trigger Action	On Change It		i change i o	raise () On Change	
		Acti	on 1 - Type None	e	•	-		
	1	Action 2 - Trigger Action	On Change To	o True 🔾 Or	Change To	False 🤇) On Change	
		Acti	on 2 - Type None	e		-		
		Update C	onditional (Cancel	Delete Cond	litional		

Logix entry

Change the turnout to reverse and move the train. (click on the icons)

Logix entry

Now a check of the states shows that only the button press is False.



Edit Conditional Help Conditional System Name IX:S5:SCC1 Conditional User Name Switch 5 Normal Logical Expression State Variables (max 20) Variable Type Name Data 1 Data 2			
Conditional System Name IX:S5:SCC1 Conditional User Name Switch 5 Normal Logical Expression State Variables (max 20)			
Conditional User Name Switch 5 Normal Logical Expression State Variables (max 20)			
Logical Expression State Variables (max 20)			
State Variables (max 20)			
			_
Variable Type Name Data 1 Data 2			
	State Trig	ggers Calculation	
Sensor Inactive LS2 N/A N/A T	ue		
AND Sensor Active IS:S5:CL N/A N/A T	ue		
AND NOT Turnout Closed LT1 N/A N/A T	ue		
AND Sensor Active IS:P6:CB N/A N/A F	alse		
All state variables are OK.	7		
	7		
Add State Variable Check State Variables			
Actions			
Action 1 - Trigger Action 💿 On Change To True 🕓 On Change To Fa	lse 🔾 On	Change	
Action 1 - Type None 👻			
Action 2 - Trigger Action 💿 On Change To True 🛛 On Change To Fa	lse 🔾 On	Change	
Action 2 - Type None 👻			
	nal		

Logix entry

Change the turnout to reverse and move the train. (click on the icons)

- Now a check of the states shows that only the button press is False.
 - Actually the button press is the only item we want to attempt to trigger the conditional, so uncheck the others.



Logix entry

Help														
	Conditional	System Name IX	(:85:8001											
Conditional User Name Switch 5 Normal														
Logical Expression														
State Variables (max 20)														
Variable T	Variable Type Name Data 1 Data 2					ion								
Sensor Inactive	LS2	N/A	N/A	True										
AND Sensor Active	IS:S5:CL	N/A	N/A	True										
AND NOT Turnout Closed	LT1	N/A	N/A	True										
AND Sensor Active	IS:P6:CB	N/A	N/A	False	~									
Add State Variable Check State Variables														
	All sta	ate variables are	0К.											
Astisms														
Actions														
Action 1 - Trigger A	iction (e) On Chan	ge to true 🔾 C	n Change To	False 🤇) On Change									
Action 1 - Type None			•	-										
	None	-												
Action 2 - Trigger Action On Chan Set Signal Appearance alse On Change														
Action 2 - Trigger A	Set Signal Held		aise (/ on onange										
	ACTION Z - TYPE	Clear Signal Hel Set Signal Dark	d											
		Set Signal Lit		ional		Set Signal Lit								
Up	date Conditional	Trigger Route												

- Change the turnout to reverse and move the train. (click on the icons)
- Now a check of the states shows that only the button press is False.
- Actually the button press is the only item we want to attempt to trigger the conditional, so uncheck the others.
- Now we choose our action. Select 'Set Turnout'.



20	J	MRI 🐂 📔 🗕							
-			Edit Cond	litional					
Help									Logiv
			Conditional Syst	em Name IX	:S5:SCC1				Logix
		Conditio	nal User Name Sw	itch 5 Normal					
Log	ical Exp	ression							• This gi
			State Varia	ibles (max 🗄	20)				box an
		Variable Type	Name	Data 1	Data 2	State	Triggers Calculation		
		Sensor Inactive	LS2	N/A	N/A	True			enter t
AND		Sensor Active	IS:S5:CL	N/A	N/A	True			and de
AND	NOT	Turnout Closed	LT1	N/A	N/A	True			
AND		Sensor Active	IS:P6:CB	N/A	N/A	False			
			All state va	ariables are	ок.				
	Add State Variable Check State Variables								
Actions									
Action 1 - Trigger Action									
	Action 1 - Type Set Turnout Closed								
	Action 2 - Trigger Action 💿 On Change To True \ominus On Change To False \ominus On Change								
Action 2 - Type None									
Update Conditional Cancel Delete Conditional									

Logix entry

Logix entry

• This gives us a new box and selection to enter the turnout ID and desired action.



	J							
1			Edit Cond	litional				
Help								
			Conditional Syste	em Name IX:	S5:SCC1			
		Condi	tional User Name Sw	itch 5 Normal				
Logi	cal Expr	ession						
			State Varia	bles (max 2	20)			
		Variable Type	Name	Data 1	Data 2	State	Triggers Calculation	
		Sensor Inactive	LS2	N/A	N/A	True		
AND		Sensor Active	IS:S5:CL	N/A	N/A	True		
AND	NOT	Turnout Closed	LT1	N/A	N/A	True		
AND		Sensor Active	IS:P6:CB	N/A	N/A	False	2	
			Add State Variable	Check St	ate Variable	s		
Actio	ne							
Avin		dian di Talanan Anti		T	Ohenne Te			
	A	cuon 1 - Trigger Acti	on 🖲 On Change To	illue O Or	i Change To	Faise (j on change	
		Action 1 - Type	e Set Turnout	-	LT1	Close	ed 👻	
		ation 2 - Trigger Acti	on 🛞 On Change Te	True O Or	Change Te	Folgo (On Change	
	A	ction 2 - Trigger Acti	on 🖲 On Change To		change 10	raise (y on change	
			Action 2 - Type None)	•	-		

Logix entry

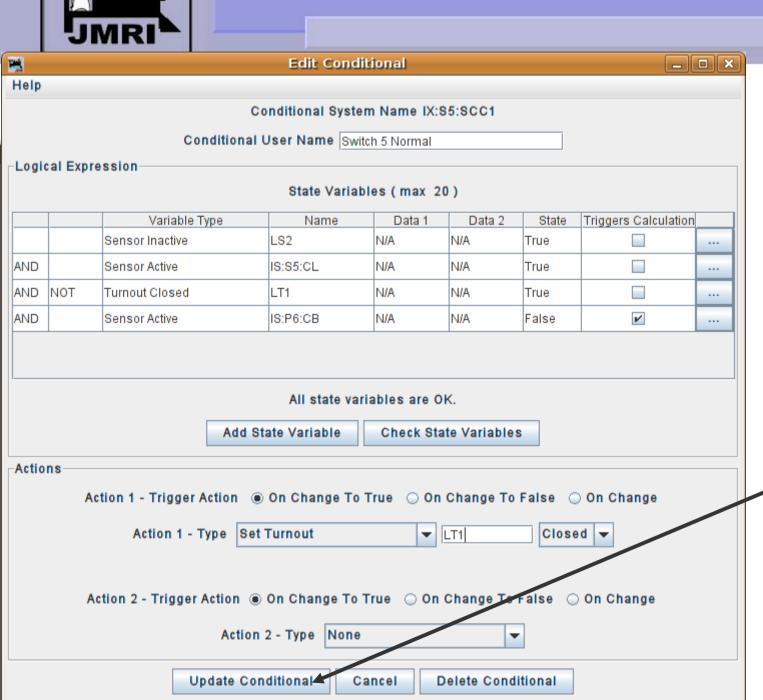
- This gives us a new box and selection to enter the turnout ID and desired action.
- Enter LT1 as the ID and the action is already defaulted to 'Closed'.



Logix entry Logix entry This gives us a new box and selection to Data 2 State Triggers Calculation enter the turnout ID N/A True ... and desired action. N/A True N/A True Enter LT1 as the ID • V N/A False ... and the action is already defaulted to 'Closed'

٠

When you are done click on 'Update Conditional'. Unlike images, Logix will not operate until they are closed.





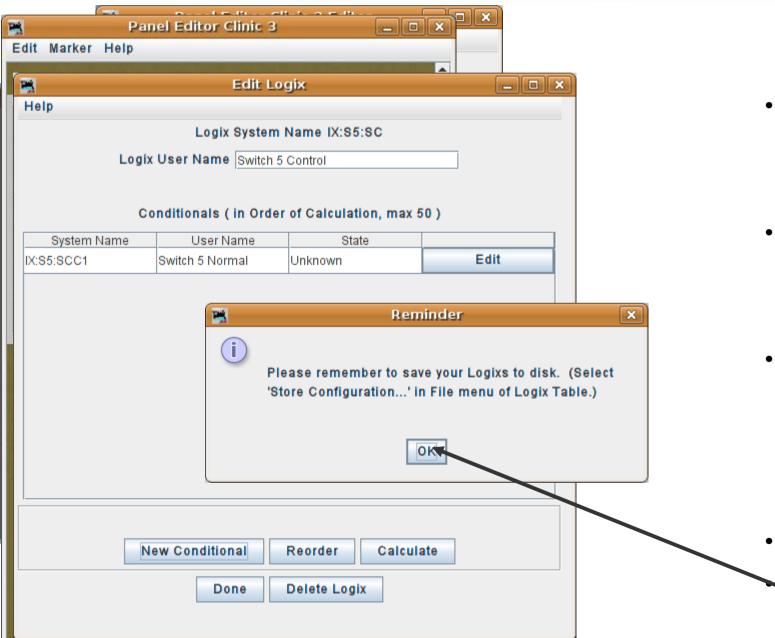
Logix entry

Edit Markar		litor Clinic 3		×	L
Edit Marker	нер		to a di los manaliti		_
1		Edit Lo	gix	-0	
Help					• 1
			Name IX:S5:SC		h
	Logix User	Name Switch 5	Control		e
-					6
	Conditio	onals (in Order	of Calculation, max 5	0)	
System I	Vame	User Name	State		• F
IX:S5:SCC1	Switc	h 5 Normal	Unknown	Edit	6
					a la
					1
					• \
					0
					i
					r
					t
	New Co	onditional	Reorder Calcula	ate	• (
		Done	Derete Logix		

- This gives us a new box and selection to enter the turnout ID and desired action.
- Enter LT1 as the ID and the action is already defaulted to 'Closed'.
- When you are done click on 'Update Conditional'. Unlike images, Logix will not operate until they are closed.
- Click 'Done' to close

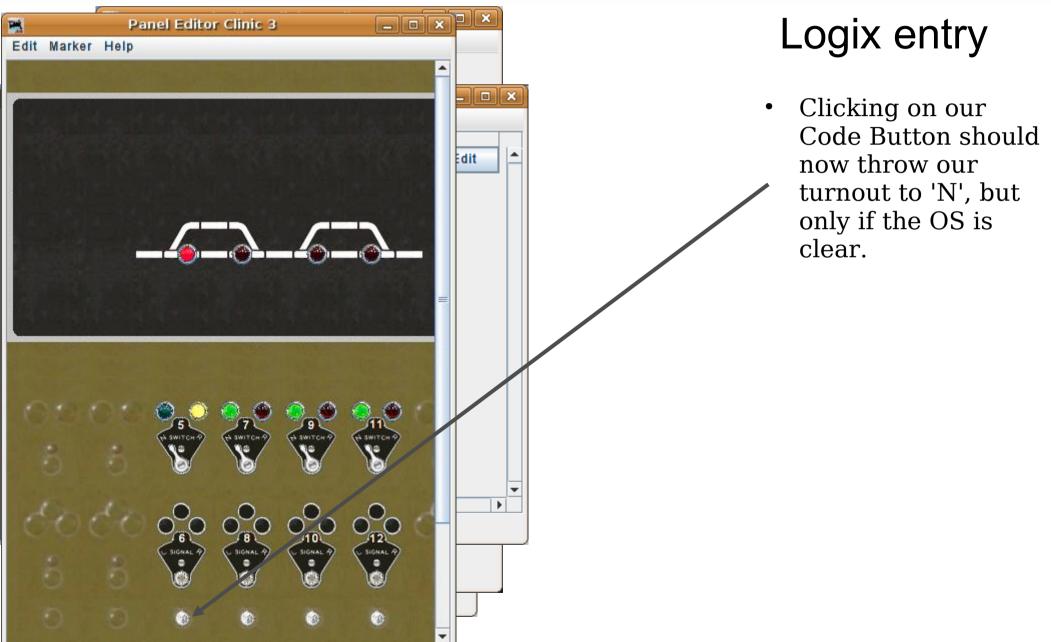


Logix entry



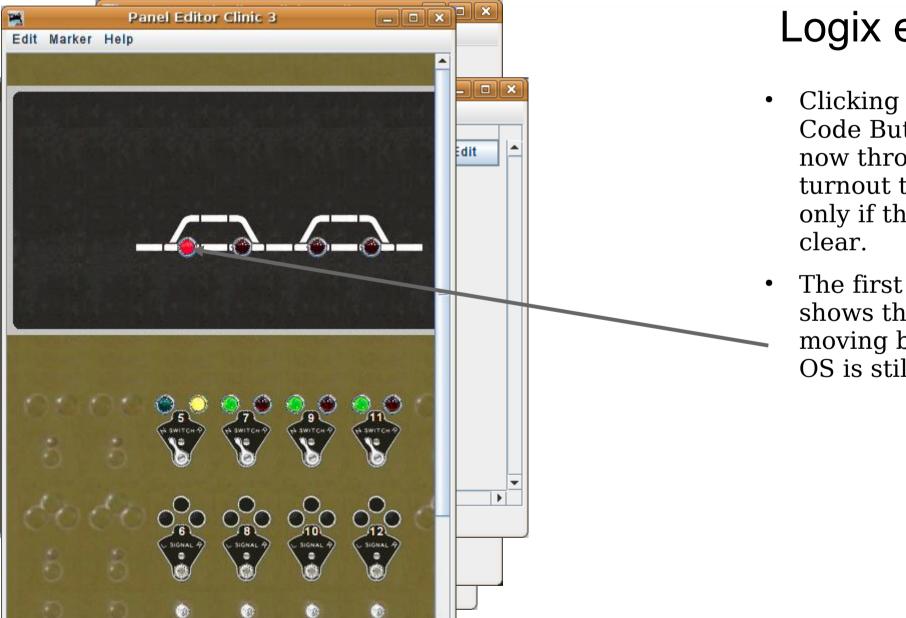
- This gives us a new box and selection to enter the turnout ID and desired action.
- Enter LT1 as the ID and the action is already defaulted to 'Closed'.
- When you are done click on 'Update Conditional'. Unlike images, Logix will not operate until they are closed.
- Click 'Done' to close
- Saved Panels will include their Logix.







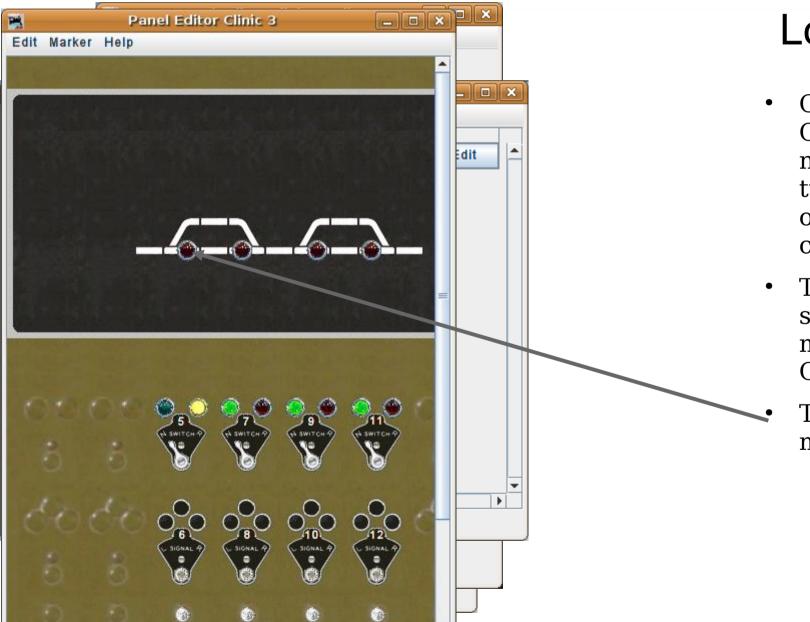
Logix entry



- Clicking on our Code Button should now throw our turnout to 'N', but only if the OS is
- The first test only shows the button moving because the OS is still occupied.



Logix entry



- Clicking on our Code Button should now throw our turnout to 'N', but only if the OS is clear.
- The first test only shows the button moving because the OS is still occupied.
- Try it again after moving the train.



Logix entry

Panel Editor Clinic 3 P Edit Marker Help . Edit SWITC \mathbf{T} • SIGNAL

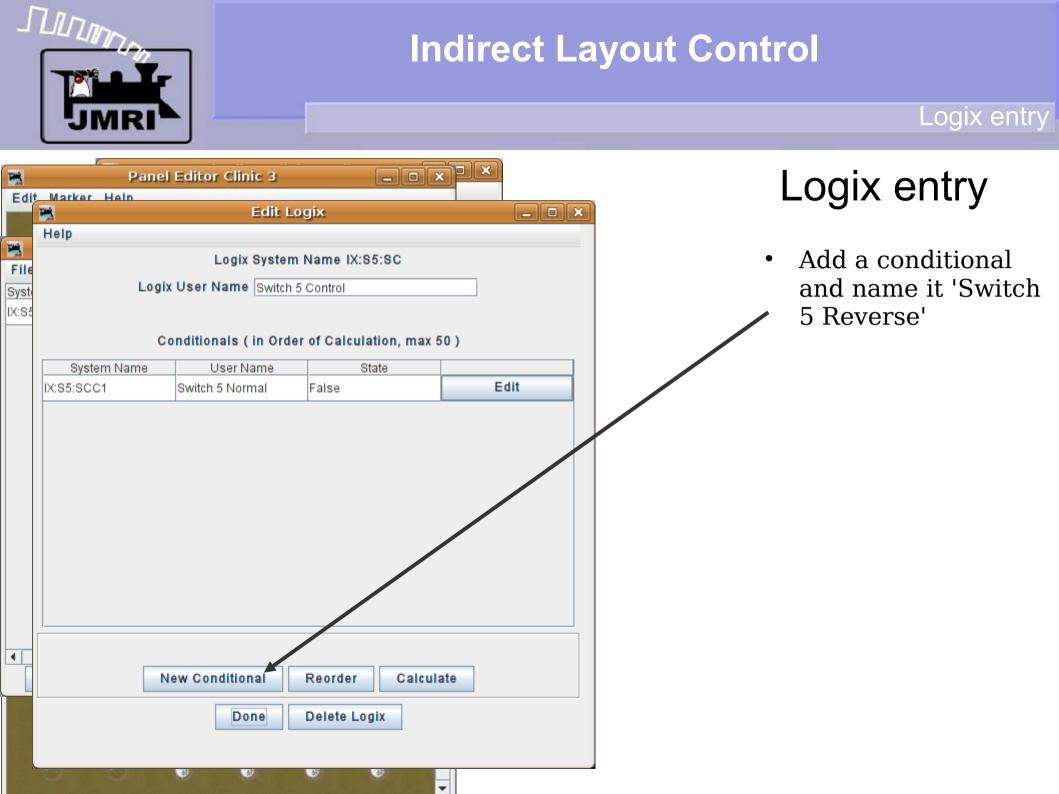
- Clicking on our Code Button should now throw our turnout to 'N', but only if the OS is clear.
- The first test only shows the button moving because the OS is still occupied.
- Try it again after moving the train.
- This time it workedas it should.

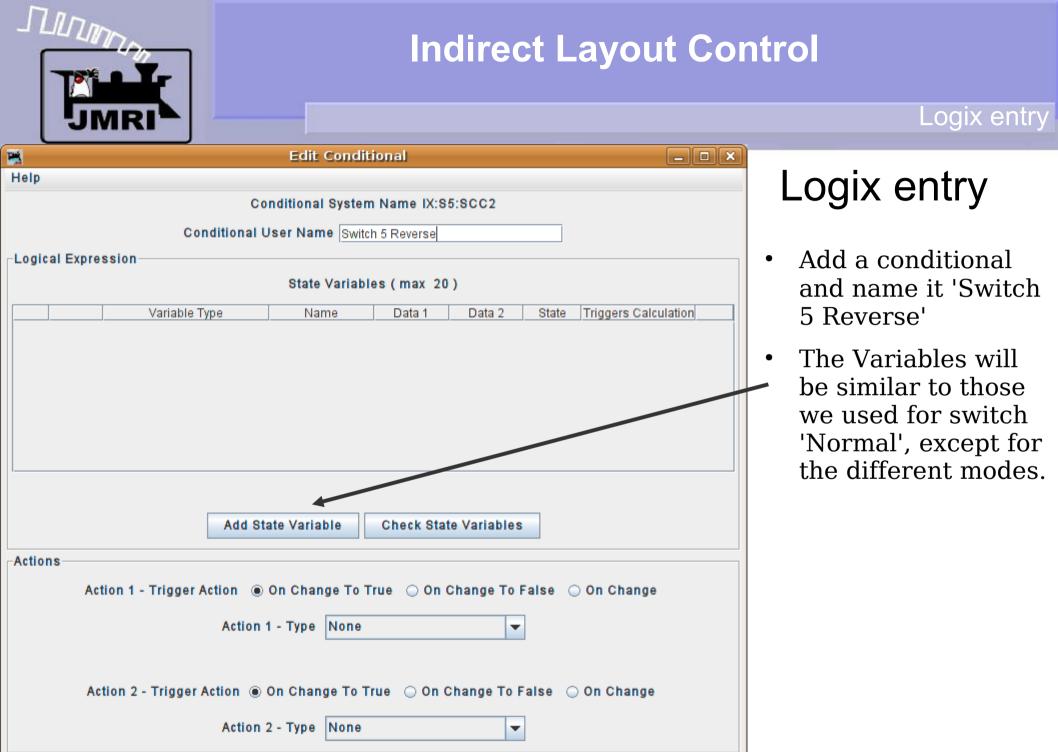


Logix entry

- o x Pl P Panel Editor Clinic 3 Edit Marker Help _ O X PÅ. Logix Table File Help System Name User Name Enabled Delete Edit IX:S5:SC r Switch 5 Control ٠ . Add

- Clicking on our Code Button should now throw our turnout to 'N', but only if the OS is clear.
- The first test only shows the button moving because the OS is still occupied.
- Try it again after moving the train.
- This time it worked as it should.
- Now click on the 'Edit' button to open up our Logix again.





Update Conditional	Cancel	Delete Conditional



PÅ. Edit Conditional Help Conditional System Name IX:S5:SCC2 Conditional User Name Switch 5 Reverse Logical Expression State Variables (max 20) Variable Type Data 2 Triggers Calculation Name Data 1 State 1.82 N/A Sensor Inactive N/A True ... • IS:S5:CL N/A N/A False AND Sensor Inactive 🧹 ... AND NOT Turnout Thrown ┥ N/A N/A True _T1 ... IS:P6:CB V AND Sensor Active False ... All state variables are OK. Add State Variable Check State Variables Actions Action 1 - Trigger Action 💿 On Change To True 🔘 On Change To False 🔘 On Change Action 1 - Type Set Turnout LT1 Thrown 😽 Action 2 - Trigger Action On Change To True On Change To False On Change Action 2 - Type None Ŧ Update Conditional Cancel Delete Conditional

Logix entry

Add a conditional and name it 'Switch 5 Reverse'

- The Variables will be similar to those we used for switch 'Normal', except for the different modes.
 - Note these three differences due to the inverted logic required to move the turnout in the opposite direction.



PÅ. Edit Conditional Help Conditional System Name IX:S5:SCC2 Conditional User Name Switch 5 Reverse Logical Expression State Variables (max 20) Variable Type Data 2 Triggers Calculation Name Data 1 State 1.82 N/A N/A Sensor Inactive True ... • IS:S5:CL N/A N/A False AND Sensor Inactive ... AND NOT Turnout Thrown LT1 N/A N/A True ... IS'P6'CB N/A V AND Sensor Active N/A False ... All state variables are OK. • Add State Variable Check State Variables Actions Action 1 - Trigger Action 🔘 On Change To True 🔾 On Change To False 🔾 On Change Action 1 - Type Set Turnout LT1 Thrown 🔫 Action 2 - Trigger Action On Change To True On Change To False On Change Action 2 - Type None Update Conditional Cancel Delete Conditional

Logix entry

Add a conditional and name it 'Switch 5 Reverse'

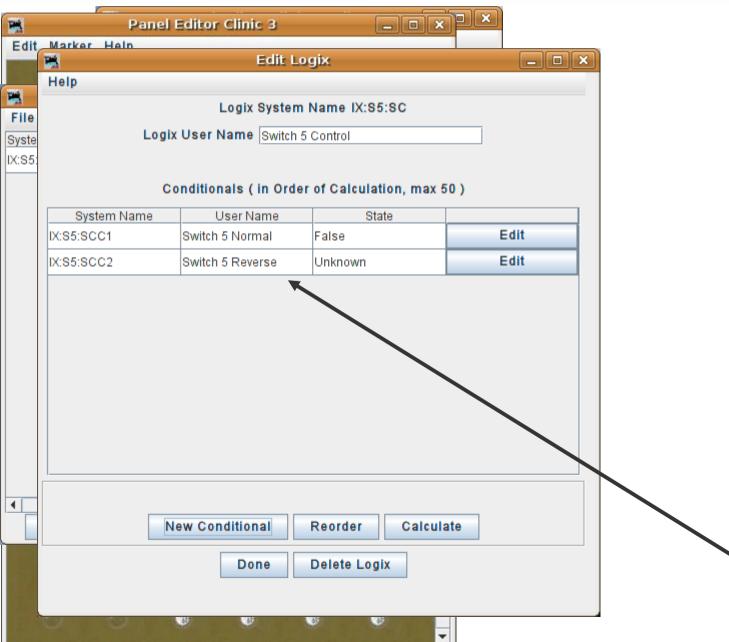
Logix entry

- The Variables will be similar to those we used for switch 'Normal', except for the different modes.
- Note these three differences due to the inverted logic required to move the turnout in the opposite direction.

• Update the entry.



Logix entry



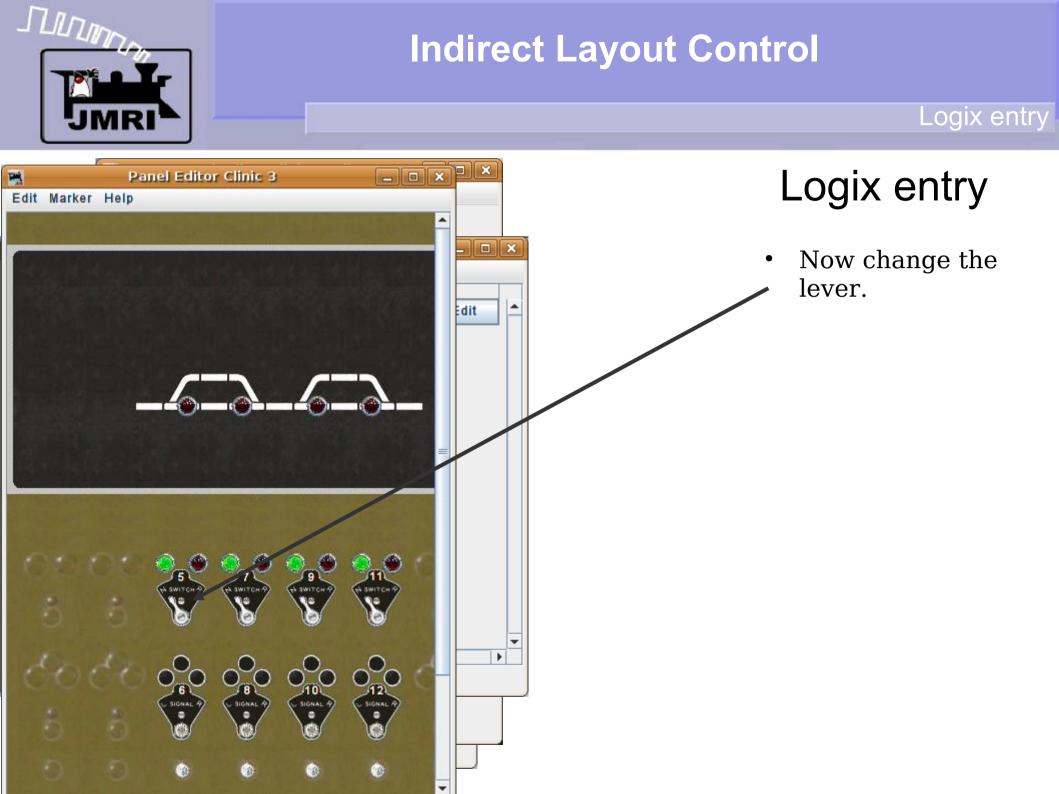
- Add a conditional and name it 'Switch 5 Reverse'
- The Variables will be similar to those we used for switch 'Normal', except for the different modes.
- Note these three differences due to the inverted logic required to move the turnout in the opposite direction.
- Update the entry.
- New item shown.



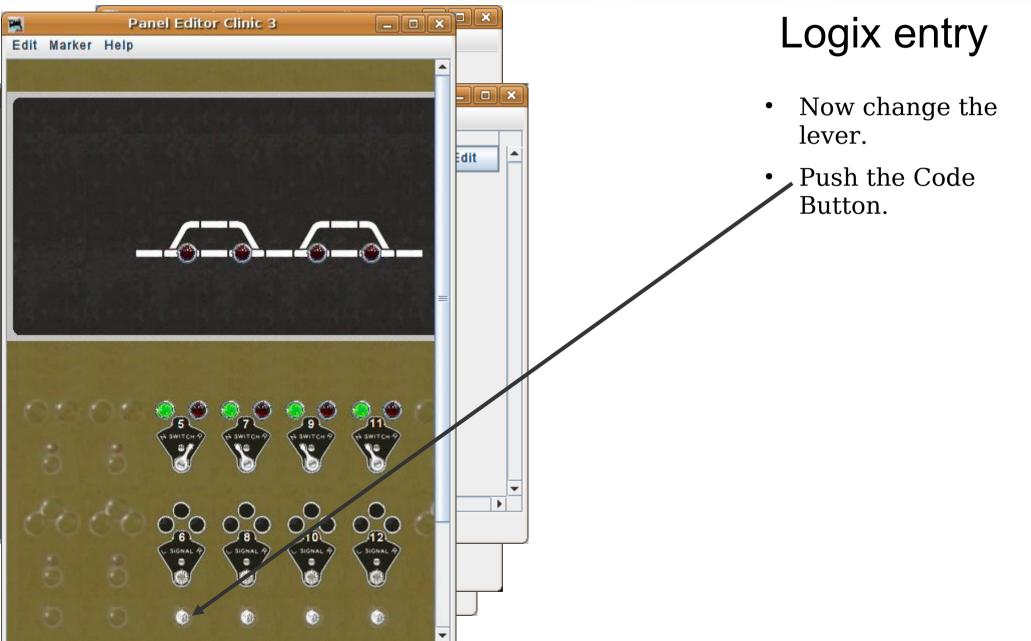
Logix entry

- O X D X P Panel Editor Clinic 3 Edit Marker Help _ O X Edit Logix PŔ Help P<u>i</u> Logix System Name IX:S5:SC File Logix User Name Switch 5 Control Syste IX:S5 Conditionals (in Order of Calculation, max 50) User Name State System Name Edit IX:S5:SCC1 Switch 5 Normal False Edit IX:S5:SCC2 Switch 5 Reverse Unknown . New Conditional Calculate Reorder Delete Logix Done

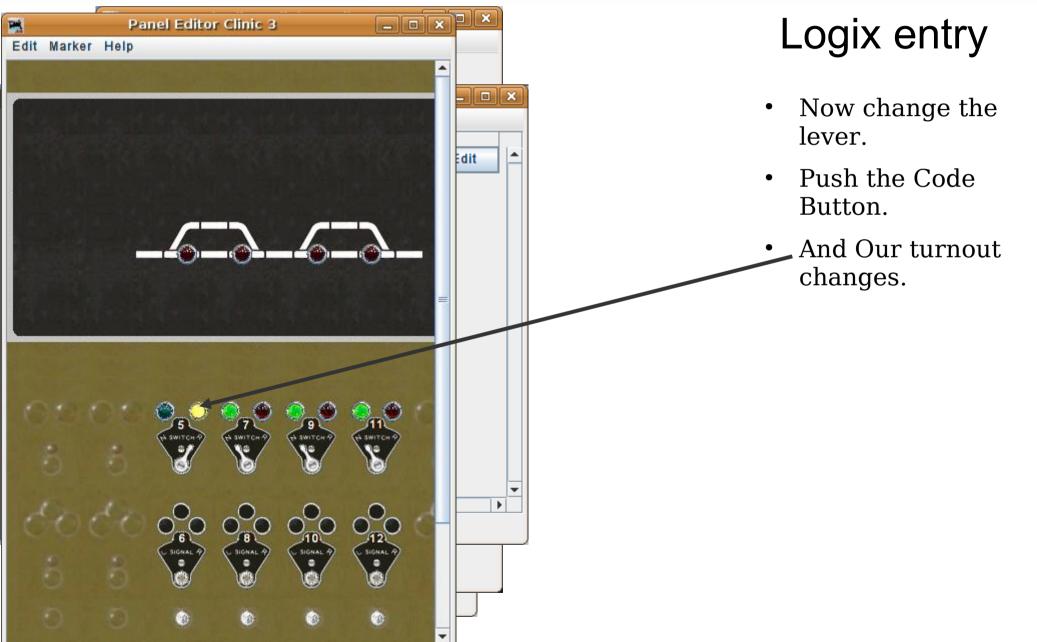
- Add a conditional and name it 'Switch 5 Reverse'
- The Variables will be similar to those we used for switch 'Normal', except for the different modes.
- Note these three differences due to the inverted logic required to move the turnout in the opposite direction.
- Update the entry.
- New item shown.
- Click Done.



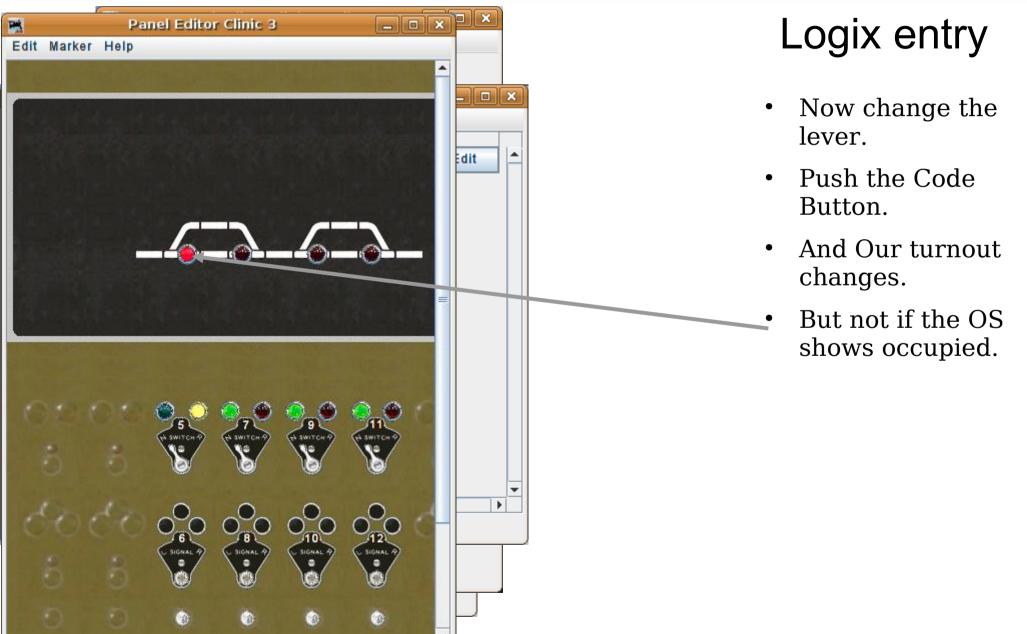














Logix entry

PR Panel Editor Clinic 3 Logix entry Marker Help Edit File Help New panel... y: 0 0 Store panels... Now change the Set panel name Delete this panel... lever. Pick background image ... Push the Code • Add text: Button. Add icon: Change icon... And Our turnout • Add right-hand turnout: LT4 Change icon... changes. Add left-hand turnout: Change icon... But not if the OS Add sensor: IS:S11:CL Change icon... shows occupied. Add signal head: Change icon... Save our work. Add memory: Add reporter: Add RPS reporter: Add multi-sensor... Add Fast clock: Panel items popup menus active



Logix entry

R	1.11	anel Edi	tor Clinic 3				ogix entry
Edit Marker	Help	File H New pa					.ogix criti y
		New pa	anels anels Set nanel Save In:	Save	PanelEditorClinic1.xml PanelEditorClinic2.xml PanelEditorClinic3.xml PanelEditorClinic3.xml PanelProConfig2.xml	• • •	Now change the lever. Push the Code Button. And Our turnout changes. But not if the OS shows occupied. Save our work. As Clinic-3.xml
õ			🗹 Panel items popu	p menus active			

100



- This completes Clinic 3. In the next session we will talk about how we may manually edit the Panel.xml file to easily duplicate our Switch 5 Logix to extend it to Switches 7, 9, and 11.
- The remainder of the next clinic will cover basic ABS signaling using SSL. (Simple Signal Logic)
 These clinic files will be available at our web site.
 http://www.rr-cirkits.com/Clinics/Clinics.html
 Versions from previous years clinics are also available there for your convenience.