N3IX ENGINEERING

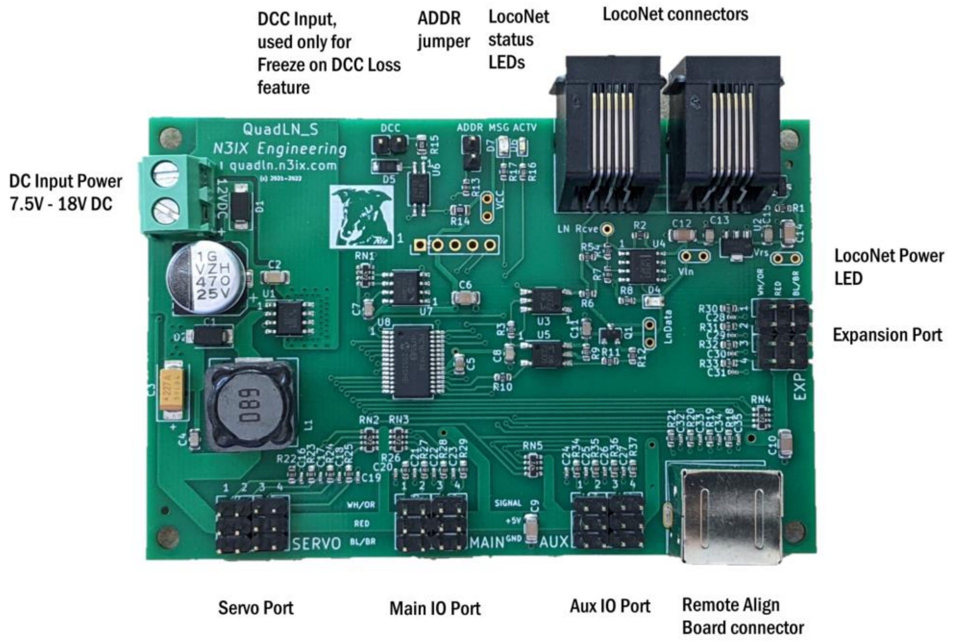
QuadLN_S User Manual

Firmware v3.3

N3IX Engineering
16 Sep 2024

Contents

1 Basics



1.1 Overview

The *QuadLN_S* is a stationary decoder that can control turnouts by driving servos (and stall motors using Driver add-on boards). The turnouts can be actuated using pushbutton or toggle switch inputs connected to **Main** IO or **Aux** IO ports. A single input can control one turnout or multiple turnouts.

The *QuadLN_S* can display the Turnout state on a Fascia Controller or drive user control panel LEDs. Besides showing whether a turnout is Closed or Thrown, the *QuadLN_S* can indicate when a servo is in motion (without requiring Turnout Position Feedback sensors!) and when a turnout is Locked.

The *QuadLN_S* can control turnouts on all ports, meaning it can drive up to **16** independent servo outputs! This capability was added especially for users who employ software panels on tablets and smartphones to control their turnouts, and as a result do not require IO lines to connect pushbuttons or toggle switches. Table 1-1 below shows all the available port configuration combinations. Besides driving servos, optional Driver add-on boards allow any mix of servo outputs and steady 5 volt outputs for stall motors or relays.

The *QuadLN_S* provides realistic signaling effects for Searchlight and Color Position Light signals and for Crossing signals using the optional Signal Board. Both Signal Head/Mast addressing and Turnout addressing for signals are available.

The *QuadLN_S* supports up to 4 current sense detectors using the optional Detector board.

The *QuadLN_S* is a LocoNet device. Turnout outputs (servos and stall motors) can be controlled via LocoNet Switch commands. Turnout position can be reported to software such as JMRI. The *QuadLN_S* even provides local control Lockout capability so that local control of turnouts can be disabled on an individual turnout basis whenever desired by just sending a LocoNet Switch command.

QuadLN_S inputs can be wired to external detectors and then report block occupancy using LocoNet General Sensor messages. Almost any device with a contact closure or 5V logic-level signal output can be wired to an available *QuadLN_S* input and then reported via LocoNet.

The *QuadLN_S* also includes Cascade and programmable Routes/ Selectors features that take advantage of the LocoNet to implement layout control functions such as yard ladder control and track selection.

Port Settings		**Turnouts**				**IO**			**New in Ver 3.0**
Expansion	**Main and Aux IO**	**Total**	**Main**	**Aux**	**Exp**	**Total**	**Detector Board**	**Signal Board**	
Relays	Both IO	4				8			
Signaling	Both IO	4				8		X	
Turnout	Both IO	8			5-8	8			
Relays	Both IO	4				4	X		
Signaling	Both IO	4				4	X	X	
Turnout	Both IO	8			5-8	4	X		
Relays	Main IO, Aux Turnout	8		5-8		4			X
Signaling	Main IO, Aux Turnout	8		5-8		4			X
Turnout	Main IO, Aux Turnout	12		5-8	9-12	4			X
Relays	Main IO, Aux Turnout	8		5-8		0	X		X
Signaling	Main IO, Aux Turnout	8		5-8		0	X	X	X
Turnout	Main IO, Aux Turnout	12		5-8	9-12	0	X		X
Relays	Both Turnout	12	5-8	9-12		0			X
Signaling	Both Turnout	12	5-8	9-12		0		X	X
Turnout	Both Turnout	16	5-8	9-12	13-16	0			X

Table 1-1 Available Port Combinations with Firmware Version 3

1.2 Turnouts - Servos and Stall Motors

The *QuadLN_S* can drive up to 16 SG-90 servos for controlling turnouts or performing other layout animation tasks. Independent control over the travel and speed for each servo make turnout movement realistic. With the addition of Driver Add-on Boards, the QuadLN_S can control a mixture of servos and 5 volt devices such as stall motors or low current relays.

1.3 External Power Supply

The *QuadLN_S* requires an external DC power supply between +7.5V and +18V. Do not exceed +18V DC. Typical current is 100-500 mA from a 12V supply, but heavy servos mechanical loads can increase the current beyond that range. It is strongly recommended that you do not share the *QuadLN_S* power supply with any devices that have non-isolated Loconet interfaces, as this would defeat the built-in isolation. See Section 2.1 for power connection details.

1.4 Onboard LED Indicators

There are 3 LEDs on the *QuadLN_S*.

1.4.1 LN POWER LED

The LN POWER LED indicates that LocoNet Rail Sync is present and the opto-isolated LocoNet interface is powered. This LED should be ON when a LocoNet cable is plugged into the *QuadLN_S*.

1.4.2 LN MESSAGE LED

The MSG LED is normally lit. The LED blinks once when any LocoNet message traffic is seen, other than messages sent by the *QuadLN_S*. If this LED never blinks then there is probably a LocoNet connection issue.

1.4.3 LN ACTIVE LED

The ACTV LED is normally lit. The LED blinks once when any LocoNet message is sent by the *QuadLN_S* or when the *QuadLN_S* acts upon any incoming LocoNet message. This LED also blinks once when the DCC state changes.

1.5 Frog Power

The *QuadLN_S* EXP Port can be configured to drive 5 volt remote relay boards for powering turnout frogs and other uses. The relay state changes at the turnout travel midpoint to reduce problems with electrical shorts during turnout movement.

1.6 Inputs

The *QuadLN_S* MAIN and AUX Ports can each provide 4 signal Inputs. All inputs are fully configurable, including what action to take and what type of input change should trigger the action. Possible actions on an input change include controlling the onboard turnouts, sending Switch commands to other decoders or to trigger routes, and generating general sensor messages that report the input change to any interested LocoNet device.

The *QuadLN_S* has special features that allow the MAIN and AUX input lines to also drive LED indicators or other devices. Since each "input" also has an "output" capability, each connection is called an "IO" line in this manual (IO = Input / Output).

The *QuadLN_S* has an optically isolated DCC input that can optionally monitor the presence or absence of DCC on a power district. This input can be used for freezing input values when there is no DCC signal (see 4.2.7) or for track power status reporting (see 6.4.14).

1.7 Addresses

1.7.1 Device Address

The *QuadLN_S* requires a Device Address so its features can be configured via LocoNet **Ops Mode** programming. This address is just for device programming and is independent from the Turnout and Sensor addresses which are described in Sections 1.7.2 and 1.7.3 below. The valid range for the Device Address is 0-16383 and it must be unique.

Typically a 5 digit extended address is used for the Device Address. This avoids conflict with locomotives and other devices when doing **Ops Mode** programming. The default Device Address is 11000. When first installing *QuadLN_S*'s remember to connect just one unprogrammed unit at a time to LocoNet and change the Device Address to a unique address following the procedure in 7.1. or 2.2.2.

1.7.2 SERVO, MAIN, AUX and EXP Turnout Start Addresses

The *QuadLN_S* Turnout Outputs for each Port respond to LocoNet Switch commands at 4 consecutive addresses starting at the Turnout Start Address for that Port. The Turnout Start Address can be from 1 to 2045. Avoid using addresses 1017-1020 for any Turnout (those addresses are used by the LocoNet special interrogation commands). The Turnout Address ranges must not overlap.

1.7.3 SERVO, MAIN, AUX and EXP Lock Start Address

The *QuadLN_S* Locks for each Port respond to LocoNet Switch commands at 4 consecutive addresses starting at the Lock Start Address for that Port, or to a single address for an entire port if the "One Lock Per Port" feature is enabled (see 6.4.13). The Lock Start Address can be from 1 to 2045. Avoid using addresses 1017-1020 for any Lock (those addresses are used by the LocoNet special interrogation command). Lock Addresses should be unique. If any Turnout and Lock address ranges overlap, any Lock with the same address as a Turnout will not be accessible using LocoNet Switch commands.

1.7.4 Main IO Start Address

The *QuadLN_S* reports the status of its Main inputs using LocoNet Sensor addresses starting at the Main IO Start Address. The *QuadLN_S* uses 4 consecutive sensor addresses for Main IO1 – Main IO4. The Main IO Start Address can be from 1 to 4093. In general the Main IO address range should not overlap the Aux IO address range.

1.7.5 Aux IO Start Address

The *QuadLN_S* reports the status of its Aux inputs using LocoNet Sensor addresses starting at the Aux IO Start Address. The *QuadLN_S* uses 4 consecutive sensor addresses for Aux IO1 – Aux IO4. The Aux IO

Start Address can be from 1 to 4093. In general the Main IO address range should not overlap the Aux IO address range.

1.8 Groups

The *QuadLN_S* has 1 functional Group for each Turnout output. Each Group consists of a Turnout and optionally a Main IO line and/or Aux IO line based on the Port configuration. As shipped, the IO lines in each Group are configured for a momentary pushbutton and change the state of the Turnout on each button press.

Any IO Input can be configured to control any Turnout output. In fact any input can control as many of the *QuadLN_S* Turnout outputs as desired, or it can serve other functions and not control any *QuadLN_S* Turnout output. There are some *QuadLN_S* features (for example LED indication) that only work when an Input is used with the Turnout output in the same Group, so it is best to start with that arrangement when learning about the *QuadLN_S*.

1.8.1 Panel Indicators

The *QuadLN_S* can drive panel indicator LEDs connected to the Main IO and Aux IO lines. The panel indication shown on any specific IO line reflects the state of the Turnout output in the same group. Thus when using panel indicators such as the N3IX Engineering Fascia Controller, it is best to use an IO line in the same Group as the Turnout being controlled so that the position, movement and lock indications are meaningful.

2 Getting Started

2.1 Connecting Power

The *QuadLN_S* requires an external DC power supply between +7.5V and +18V. **Do not exceed +18V DC**. Typical current is 100-200 mA, but heavy servo mechanical loads can increase the current beyond that range. The power input is via the connector with two screw terminals near the upper left corner of the board. Connect the power supply **minus** lead to the **upper** terminal (the one closest to the corner of the board) and connect the power supply **plus** lead to the **lower** terminal (the one furthest from the corner of the board). If the leads are connected backwards, the *QuadLN_S* will not function but will not be damaged.

It is strongly recommended that you do not share the *QuadLN_S* power supply with any devices that have non-isolated Loconet interfaces, as this would defeat the built-in isolation.

2.2 Setup with JMRI

The *QuadLN_S* decoder template in JMRI provides the easiest and quickest way to configure the overall device. This section covers the basics needed to get the decoder up and running. Later sections dive into the details.

2.2.1 Minimum JMRI Version

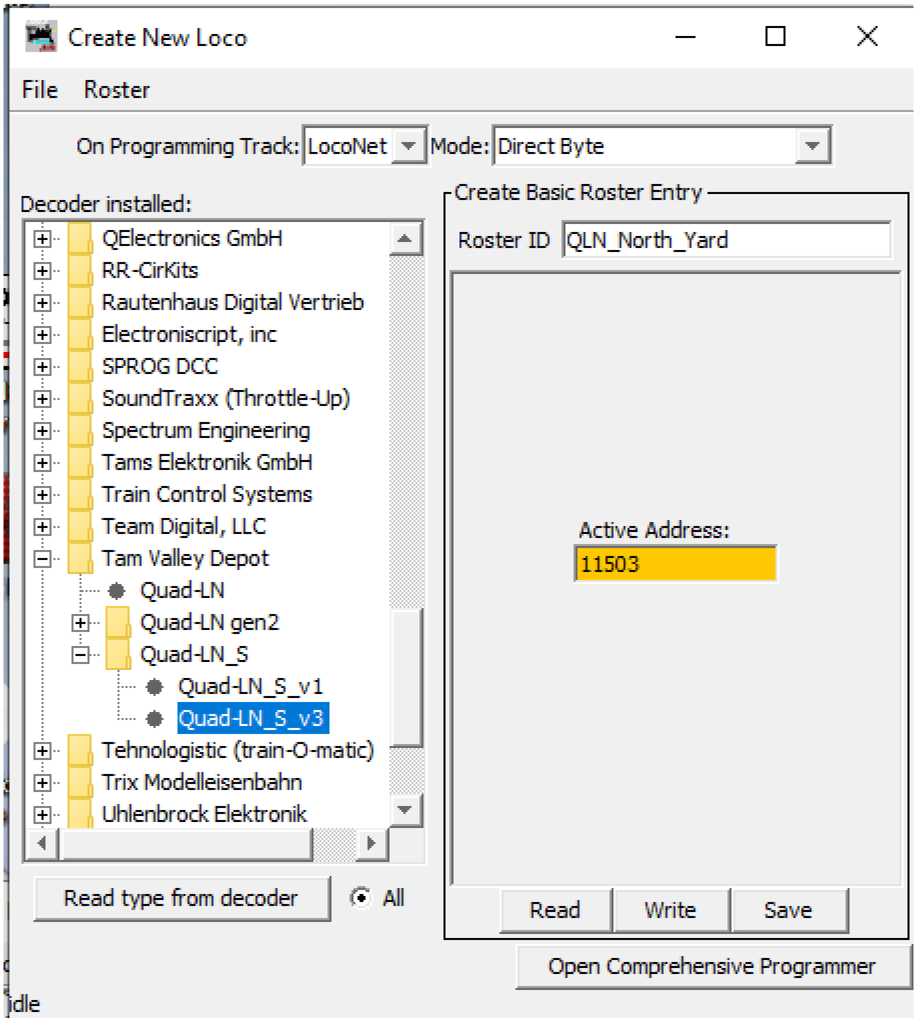
The latest released version of JMRI is recommended. The minimum version of JMRI is 5.2. JMRI version 5.8 and later come with full support for *QuadLN_S* firmware version 3.3. To use JMRI version 5.2 to 5.7, install the available *QuadLN_S* decoder template update (see the note in Section 2.2.3 below).

2.2.2 Create a roster entry and set the Device Address

Each decoder needs its own Roster entry so that there is a place to store the configuration settings that are unique to that decoder. In this step we will create the Roster entry for the decoder. We will also assign a unique Device Address to the decoder. When setting parameters of the *QuadLN_S* we will always use **Ops-Mode / Programming On Main**. The **Service-Mode/Programming-Track** Programmer is only required when performing a Factory Reset via JMRI.

The Device Address is set using programming messages that are received by the *QuadLN_S* over LocoNet. (The *QuadLN_S* is ***never*** connected to the programming track.)

- In DecoderPro click the **New Loco** button.
- Scroll down the decoder list to **Tam Valley Depot**, expand it, select the **Quad-LN_S** family and expand it and then select **Quad-LN_S_v3** decoder. If the **Quad-LN_S_v3** decoder entry is missing, update JMRI per the recommendations in Section 2.2.1.
- Enter a unique **Roster ID** for the device such as QLN_North_Yard.
- Enter the desired Device Address in the **Active Address** field. Choose a unique address between 11001 and 16383. See Section 1.7.1 for information about Device Addresses. A unique address is required to avoid conflicts. Using a 5-digit address is recommended to avoid accidental changes to CVs from throttles.



Create New Loco

On Programming Track: LocoNet

Mode: Direct Byte

Decoder installed: Quad-LN_S_v3

Roster ID QLN_North_Yard

Active Address: 11503

idle

- Click **Save** to add the entry to your Roster.
- Close the Create New Loco dialog by clicking the **X** in the upper right corner.

2.2.3 Set the Device Address in the QuadLN_S

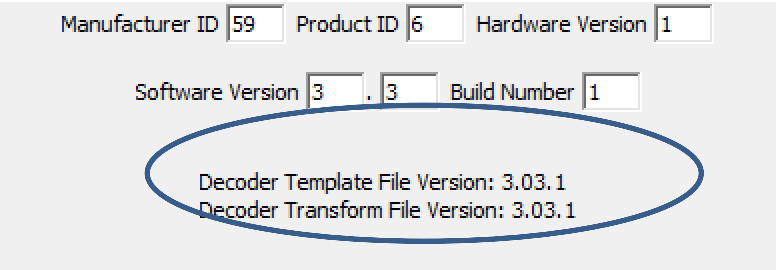
The Device Address of the QuadLN_S is set to 11000 by default and it needs to be changed now to match the value in the Roster Entry you just created.

- In DecoderPro click on the **Programming On Main** option.
- Select the roster entry you just created and then click the “Program…” button. The roster entry will open after a few seconds.
- Go to the **CVs** tab.
- Install the **ADDR** jumper on your *QuadLN_S*. The **ACTV** and **MSG** LEDs will start flashing alternately.
- **Write** CV17 and then **Write** CV18. (Ignore the “Programmer error” message, this is normal.) You must write the CVs in that order.

- The LEDs will stop flashing if the address was successfully changed. (If the LEDs are still flashing, check the Loconet connection to your computer and to the *QuadLN_S*. Then retry the previous step.)
- Remove the **ADDR** jumper.

REMINDER: Remember to **remove** the **ADDR** jumper when finished. Failure to remove the **ADDR** jumper can result in unintended changes to the Board Address.

When using *QuadLN_S* Firmware Version 3.3 or later, if the Decoder Template File Version is less than 3.03.1 or the Decoder Transform File version is less than 3.03.1 (shown towards the bottom of the Quad-LN_S tab) then please follow these steps to download and install the necessary template file.



Manufacturer ID 59 Product ID 6 Hardware Version 1

Software Version 3 . 3 Build Number 1

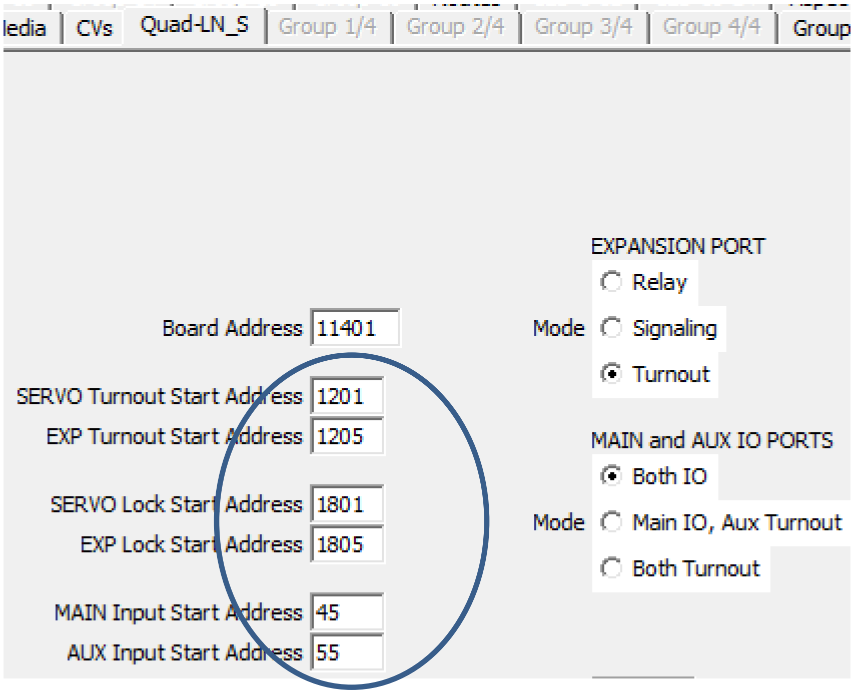
Decoder Template File Version: 3.03.1
Decoder Transform File Version: 3.03.1

- Download the latest *QuadLN_S* update package from https://quadln.n3ix.com.
- Following the instructions in the readme file, copy the new decoder files and new tvd folder (if provided) into the appropriate locations.
- **Close** the Roster Entry if it is still open.
- Re-open the Roster Entry and recheck the Decoder File Versions.

2.2.4 Turnout and Lock Start Addresses, Main and Aux IO Start Addresses

Now we are ready to enter other parameters of the *QuadLN_S*.

- If the *QuadLN_S* Roster Entry is not already open for Programming On Main:
  - In DecoderPro click on the **Programming On Main** option.
  - Select the new *QuadLN_S* entry from the Roster and then click the "Program…" button. The roster entry will open after a few seconds.
- Go to the *QuadLN_S* tab.
- Click **Read full sheet** to read the firmware version information from the *QuadLN_S* into JMRI. The JMRI roster entry will adjust based on the *QuadLN_S* firmware version, so his step is important!



Board Address 11401

SERVO Turnout Start Address 1201
EXP Turnout Start Address 1205

SERVO Lock Start Address 1801
EXP Lock Start Address 1805

MAIN Input Start Address 45
AUX Input Start Address 55

EXPANSION PORT
Mode ○ Relay ○ Signaling ● Turnout

MAIN and AUX IO PORTS
Mode ● Both IO ○ Main IO, Aux Turnout ○ Both Turnout

- Enter the desired SERVO port turnout start address in the **SERVO Turnout Start Address** field.
- Enter the desired EXP port turnout start address in the **EXP Turnout Start Address** field.
- (optional) Enter the desired SERVO lock start address in the **SERVO Lock Start Address** field.
- (optional) Enter the desired EXP lock start address in the **EXP Lock Start Address** field.
- (optional) Enter the desired Main IO start address in the **Main IO Start Address** field.
- (optional) Enter the desired Aux IO start address in the **Aux IO Start Address** field.
- Click **Write changes on sheet** to update the *QuadLN_S*.
- (recommended) Click **Read full sheet** to read back the address values and confirm the settings.
- Click **Save** on the **File** menu to update the roster entry.

2.2.5 Travel and Speed

You can follow the procedure in Section **Error! Reference source not found.** below to manually adjust your servo Travel and Speed using the Remote Align Board. You can also click on one of the Group tabs in the roster entry and begin exploring the full capabilities of the *QuadLN_S* using the information starting in Chapter 3.

2.3 Setup without JMRI

All *QuadLN_S* behavior is controlled via CV values so it is possible to fully configure the device manually. CV tables are provided in Chapter 8 below for that purpose. Please keep in mind that manually calculating the proper CV values can be tedious and error prone. Using JMRI for configuring the *QuadLN_S* is strongly recommended.

2.4 Adjustment of Servo Travel and Speed using the Remote Align Board

The travel endpoints and speed for each servo can be set manually using the **Select**, **Up** and **Down** buttons on the Remote Alignment Board. Here is the adjustment procedure.

1. Connect the Remote Align Board to the *QuadLN_S*.
2. Hold the **Select** button down until a numeric selection LED turns on (about 1 second), indicating that you are in Turnout Selection mode and that Turnout is selected. (The initial Turnout selection will be the same as the one selected the last time alignment was performed.)
3. An LED on the Remote Align Board indicates the selected Turnout using the following method:

Turnout	Remote Align LED indication
1-8	0-7 steady
9-16	0-7 flashing

   Use the **Up** and **Down** buttons to select the desired Turnout. Once the desired Turnout is indicated, press the **Select** button briefly.
4. The Closed LED is now flashing to indicate that the Turnout is in the CLOSED Position. For Servo outputs the Closed Position can be adjusted using the **Up** and **Down** buttons to move the Servo to the desired position. The servo position will change very slowly at first, so hold down the Up or Down button continuously when doing rough adjustment. When satisfied with the Closed Position, press the **Select** button briefly.
5. The Thrown LED is now flashing to indicate that the Turnout is in the THROWN Position. For Servo outputs the Thrown Position can be adjusted using the **Up** and **Down** buttons to move the Servo to the desired position. The servo position will change very slowly at first, so hold down the Up or Down button continuously when doing rough adjustment. When satisfied with the Thrown Position, press the Select button briefly.
6. The Speed LED is now flashing to indicate that the Speed is being adjusted. For Servo outputs the **Up** and **Down** buttons can be used to change the speed. The Servo will change position to demonstrate the new Speed. You can continue to adjust the speed while the Servo is moving if desired. When satisfied with the Speed, press the **Select** button briefly.
7. To readjust any of the parameters, press the **Select** button briefly to return to step 4.
8. For Servo outputs a brief press of both the **Select and Up** buttons simultaneously will center a servo at any point in the Alignment process. LED 7 (Ctr) will flash while the servo is centered. Press **Select** when finished to resume normal Align modes.

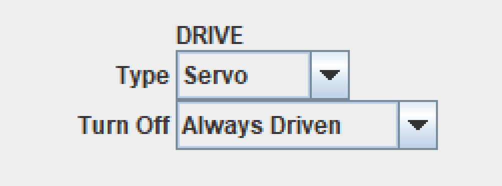
To exit the Adjustment process and save the settings at any point, hold the **Select** button until the Remote Align Board LEDs turn off (about 1 second). The new settings are now saved and in use.

3 Turnout (Servo and Stall Motor) Control

Each Turnout Output can be configured individually. To access the settings, open the Roster Entry and click on the Group tab for the desired Turnout Output.

3.1 Drive

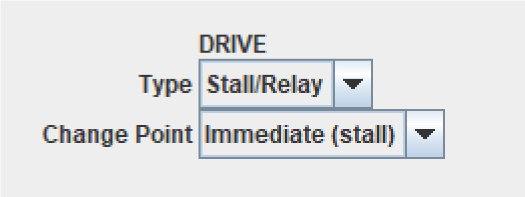
The Drive parameters allow the output behavior to be set for each Turnout output.



DRIVE

Type: Servo

Turn Off: Always Driven



DRIVE

Type: Stall/Relay

Change Point: Immediate (stall)

3.1.1 Type

The Type parameter is used to specify whether the connected output device is a Servo (pulse output drive), or a Stall Motor or Relay (steady output drive). The available settings are **Servo** and **Stall/Relay**. When Type is set to **Servo**, the JMRI template will display the “Turn Off” setting, and when Type is set to **Stall/Relay** the template will display the “Switch Point” setting .

3.1.2 Turn Off

The Turn Off setting is available when the Turnout Type is set to **Servo**. Some servos may buzz when not moving. In this case turning off the drive to the servo will stop the buzzing. If the force on the servo is modest then it will hold position even with the drive turned off.

The Turn Off parameter allows you to set whether the drive pulses to the servo stop when the servo is not moving. The available Turn Off settings are **Always Driven** and **Off When Stopped**.

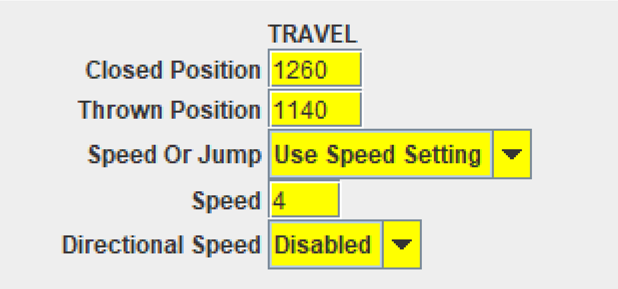
3.1.3 Switch Point

The Switch Point setting is available when the Turnout Type is set to **Stall/Relay**. When driving a stall motors it is generally desirable to have the motor start to move immediately when a turnout command is received. When driving relays you may want it to move immediately or you may want to delay the relay change. This parameter offers the option to delay for about the time a stall motor would take to reach mid-travel.

The Switch Point parameter allows whether the Turnout output changes immediately following a turnout command or after a delay. The available Turn Off settings are **Immediate (stall)** and **Delayed (relay)**.

3.2 Travel

The Travel parameters allow the Servo Closed and Thrown Position endpoints, and Speed features to be set for each Servo.



TRAVEL

Closed Position: 1260

Thrown Position: 1140

Speed Or Jump: Use Speed Setting

Speed: 4

Directional Speed: Disabled

3.2.1 Closed Position

The servo Closed Position endpoint can be set from 0 to 2400, where 0 is fully counter-clockwise and 2400 is fully clockwise.

3.2.2 Thrown Position

The servo Thrown Position endpoint can be set from 0 to 2400, where 0 is fully counter-clockwise and 2400 is fully clockwise.

3.2.3 Speed Or Jump

Normally the QuadLN_S moves the servo slowly when changing from one position to the other. For some animation effects, like a pile driver, it may be desirable for the servo to move immediately to the new position.

This parameter allows you to select whether the servo moves slowly per the Speed setting or jumps directly to the new position. The available settings are **Use Speed setting** and **Jump to Setpoint**. When Speed Or Jump is set to **Use Speed setting** the JMRI template will display the "Speed" setting.

3.2.4 Speed

The Speed can be set from 0 to 63, where 0 is very slow and 63 is very fast. A typical Speed setting for turnouts is 4. The servo position is controlled continually during movement so full torque is always applied. This makes movement smooth even at very slow speed settings.

3.2.5 Directional Speed

Normally the *QuadLN_S* drives a servo at the same speed whether the servo is moving towards the Closed position or towards the Thrown position. The *QuadLN_S* also allows each servo to have independent Speed (and Speed Or Jump) settings for each direction of movement. This feature might be useful when using a servo to drive a semaphore blade or an animation function like a pile driver, where the speed in one direction is different than the other.

The Directional Speed parameter allows you to set independent Speed and Speed Or Jump values for each direction of travel. The available Directional Speed settings are **Disable** and **Enable**. When Directional Speed is set to **Enable**, the JMRI template will display two additional settings: "Thrown Speed Or Jump" and "Thrown Speed".

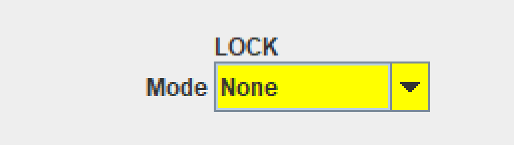
3.2.6 Thrown Speed Or Jump

When Directional Speed is enabled, the Thrown Speed or Jump parameter allows you to select the desired movement speed behavior in the Thrown direction. The Speed or Jump parameter in Section 3.2.3 always sets the behavior in the Closed direction.

3.2.7 Thrown Speed

When Directional Speed is enabled and Thrown Speed Or Jump is set to Use Speed Setting, the Thrown Speed parameter sets the speed when moving in the Thrown direction only. The parameter in Section 3.2.3 and 3.2.4 always governs the speed when moving in the Closed direction.

3.3 Lock Mode



LOCK

Mode None

The *QuadLN_S* has three optional Lock modes - **Local Lockout**, **Midpoint and Midpoint/Oscillate -** which are described below. The LocoNet Switches specified by the **Lock Start Address** for each IO Port control the locks on the Turnouts.

The available Lock Mode settings for each Turnout are **None, Local Lockout, Midpoint and Midpoint/Oscillate**. When set to **None**, the LocoNet Switch at the Lock Address has no effect. When set to **Local Lockout**, the local controls for the turnout will be disabled (locked) when the Lock Control Switch is set to Thrown. LocoNet control of the turnout will also be disabled if the **Lock Disables LocoNet Control** option has been set. When set to **Midpoint**, a Servo it will move to the mid-travel position when the Lock Control Switch is set to Thrown and will return to the last commanded position when the Lock Control is set to Closed. When set to **Midpoint/Oscillate**, the Turnout will continually move back and forth between the Closed and Thrown setpoints when the Lock Control is set to Closed, and a Servo will move to the mid-travel position when the Lock Control Switch is set to Thrown.

3.3.1 Local Lockout

In systems like CTC, some turnouts are controlled remotely by the dispatcher. These remotely control turnouts are normally “locked “, that is they cannot be operated by a local train crew member. When a train crew wishes to operate one of these controlled turnouts such as during local switch moves, they are required to first obtain permission from the dispatcher before the turnout is “unlocked” and then operated locally.

Besides supporting realistic operation under CTC, the ability to lock a turnout is useful just to prevent it from changing inadvertently. For example, if running a train continuously during an Open House it is desirable to lock the mainline route so that someone bumping into a fascia control doesn’t cause a derailment or worse.

The *QuadLN_S* **Local Lockout** feature permits local control of a turnout to be disabled (locked out). It also has the ability to provide a local indication of the state of the turnout lock. Locking a *QuadLN_S* Turnout means that local control actions via the Main and Aux IO lines (i.e. buttons, toggles and secondary triggers) can no longer directly change the Turnout position. The Turnout still responds normally to LocoNet Switch commands at its address, including commands originated by secondary messages or by routes, unless the **Lock Disables LocoNet Control** option has been set.

Example – Local Lockout

Servo 1 Lock Mode is set to **Local Lockout** and the *QuadLN_S* Lock Start Address is set to 13. This means the Lock Control Switch for Turnout 1 is at LocoNet Switch address 13. When Switch 13 is

Thrown, local control of Turnout 1 is locked, and when Switch 13 is Closed local control of Turnout 1 is unlocked.

Example – No Lock Action
Servo 1 Lock Mode is set to **None** and the *QuadLN_S* Lock Start Address is set to 13. Changing Switch 13 to Thrown or Closed has no effect on local control of Turnout 1.

The *QuadLN_S* Lock mechanism supports unlocking either by the local crew or by the Dispatcher as desired. The local crew could control the lock by sending the appropriate Switch command to the Lock Control Switch using a handheld throttle. A more interesting approach might be for the local crew to control the lock via a keyswitch or other special electrical interlock device that is connected to a *QuadLN_S* input pin. There is an example in Section 4.5.5 showing how to accomplish this. Having the local crew control the lock in this manner is similar to prototype practice but requires hardware like a keyswitch for every turnout to be unlocked.

An alternative approach is to have the Dispatcher control the lock by sending the appropriate LocoNet Switch command, which could be done via a control panel lever for instance. This approach doesn’t require extra hardware and the Fascia Controller can indicate the state of the Lock to the local crew to avoid confusion (see Section 4.3.1).

To lock specific turnouts to protect a continuously running train during an open house, the Lock Control Switch for any turnout can be set using a handheld throttle. Lock Control Switches can be part of a Route, making it possible to lock or unlock a series of turnouts with just a single action. The Lock or Unlock Routes could even be fired by a pushbutton or toggle switch connected to one of the *QuadLN_S* inputs, making the process of locking or unlocking key fascia turnout controls very simple. There is an example in Section 4.5.5 showing how to configure a pushbutton press to fire a Route. Fascia Controllers can display the state of the Lock so it is easy to see that the layout is secured (see Section 4.3.1).

3.3.2 Midpoint

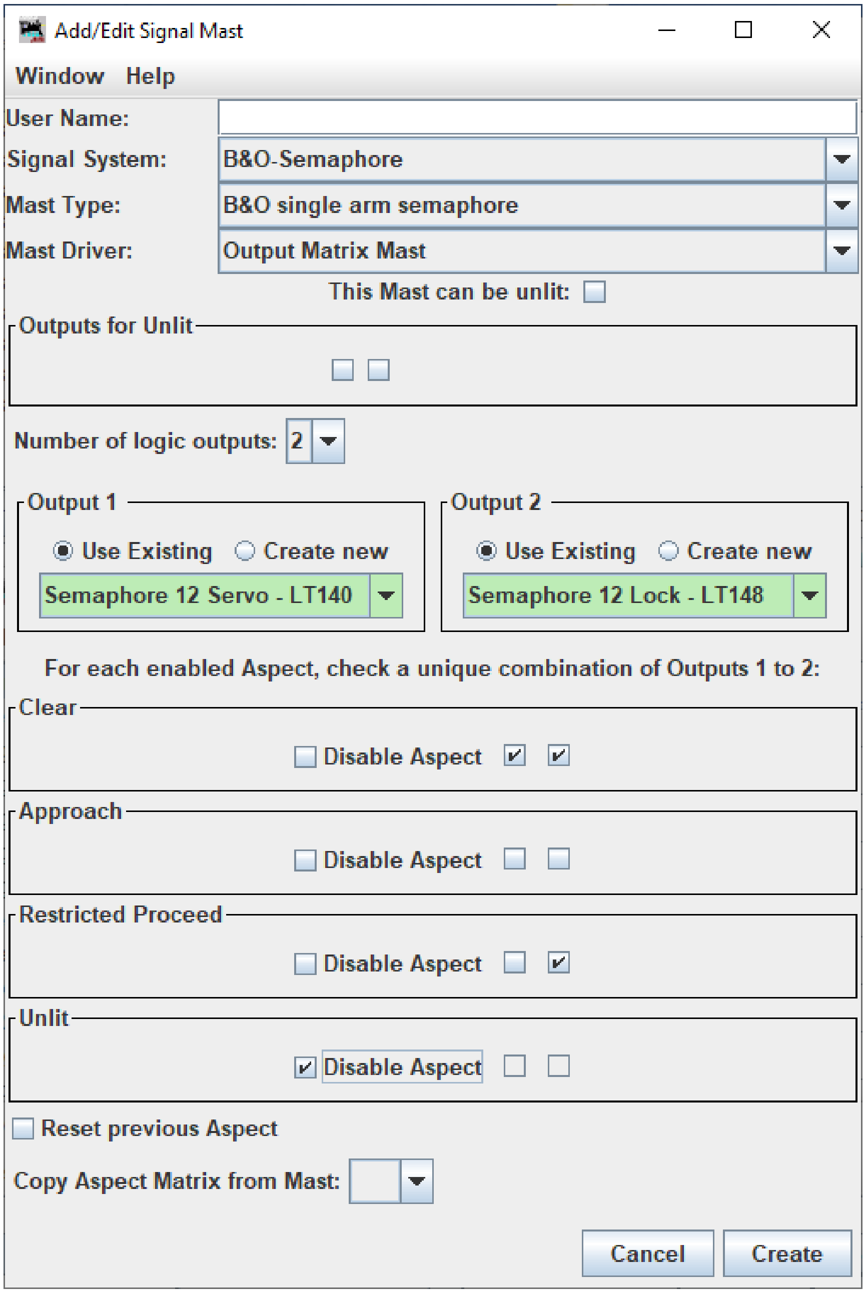
When **Midpoint** mode is selected, setting the Lock Control Switch to Thrown causes the servo to move midway between the Closed and Thrown setpoints. The servo will return to the last commanded position when the Lock Control Switch is Closed. This mode may be useful for control of 3 position semaphores, 3-way stub turnouts, and special animation effects.

Both LEDs on a Fascia Controller will be lit when the corresponding servo is at the midpoint position. If using bicolor LEDs, the Red and Green elements are switched on and off rapidly so that each is on half the time, producing a shade of Yellow. The switch rate is faster than the eye can detect so no flicker is visible. If using a custom circuit to drive an indicator, the Yellow shade can be varied by changing the Red and Green LED current levels (often more Red current / less Green current is required).

3.3.3 Midpoint Example – Semaphore Mast using Output Matrix Mast Driver

Suppose that a semaphore is controlled by Servo turnout LT140 and the associated Lock turnout is LT148. To set up the *QuadLN_S*, enable **Midpoint** mode for Servo LT140 as described above. Now whenever LT148 is THROWN the servo goes to the Midpoint regardless of the state of LT140. When LT148 is CLOSED the servo will follow the state of LT140.

To enable control of the Semaphore in JMRI, create a new Signal Mast in JMRI and set the **Mast Driver** to **Output Matrix Mast**. Next set the **Number of logic outputs** to **2** and configure the rest of the mast as shown in the following screenshot.



Add/Edit Signal Mast

Window Help

User Name:

Signal System: B&O-Semaphore

Mast Type: B&O single arm semaphore

Mast Driver: Output Matrix Mast

This Mast can be unlit: □

Outputs for Unlit

□ □

Number of logic outputs: 2

Output 1

● Use Existing ○ Create new

Semaphore 12 Servo - LT140

Output 2

● Use Existing ○ Create new

Semaphore 12 Lock - LT148

For each enabled Aspect, check a unique combination of Outputs 1 to 2:

Clear

□ Disable Aspect ☑ ☑

Approach

□ Disable Aspect □ □

Restricted Proceed

□ Disable Aspect □ ☑

Unlit

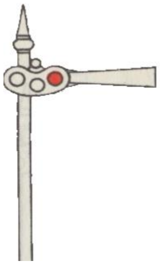
☑ Disable Aspect □ □

□ Reset previous Aspect

Copy Aspect Matrix from Mast:

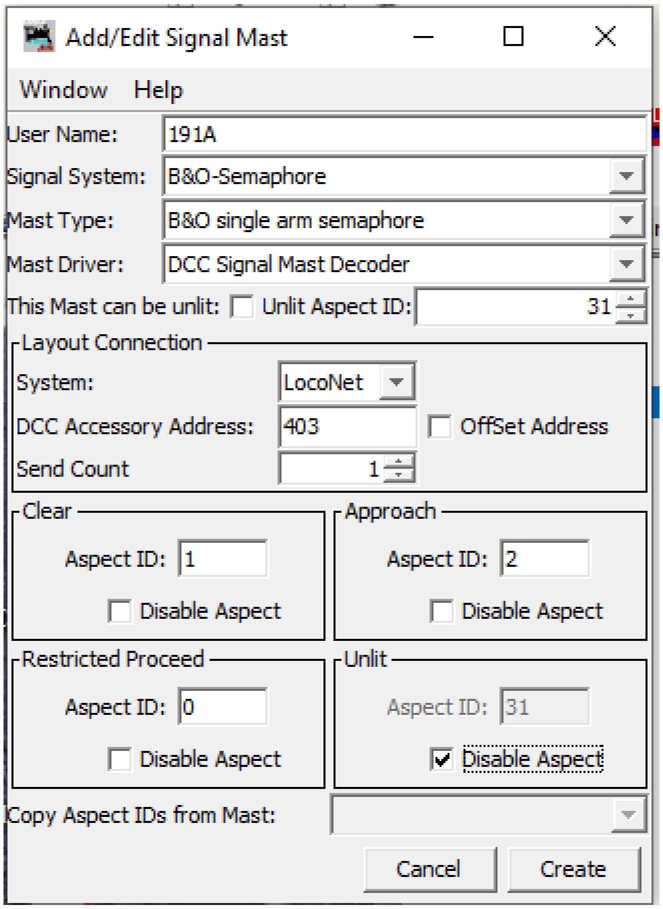
There are 2 checkboxes in each Aspect: the first is for LT140 (Output 1) and the second is for LT148 (Output 2). When one of these boxes is checked JMRI sets associated turnout to CLOSED for that Aspect, and when the box is not checked JMRI sets the associated turnout to THROWN for that Aspect. The key is the second checkbox, which controls Lock turnout LT148. Only the Approach Aspect sets LT148 to THROWN since that forces the servo to the Midpoint. Both the Clear and Stop Aspects set LT148 to CLOSED, which allows LT140 to govern the servo position per the first checkbox.

3.3.4 Midpoint Example – Semaphore Mast using Signal Aspects



To make semaphore control simpler, *QuadLN_S* firmware 3.2 added support for controlling Turnouts using Signal Aspects whether or not a Signal Board is present. In this example we have a Semaphore blade controlled by a servo connected to the Turnout 1 output of the *QuadLN_S*. To set up the *QuadLN_S*, enable **Midpoint** mode for Turnout 1 as described above. Now whenever the Lock for Turnout 1 is THROWN the servo goes to the Midpoint regardless of the state of the Turnout 1 output. When the Lock for Turnout 1 is CLOSED the servo follows the state of the Turnout 1 output.

To enable control of the Semaphore in JMRI, create a new Signal Mast in JMRI and set the **Mast Driver** to **DCC Signal Mast Decoder**. Configure the mast like the following screenshot.



User Name: 191A
Signal System: B&O-Semaphore
Mast Type: B&O single arm semaphore
Mast Driver: DCC Signal Mast Decoder
This Mast can be unlit: ☐ Unlit Aspect ID: 31

Layout Connection
System: LocoNet
DCC Accessory Address: 403 ☐ OffSet Address
Send Count 1

Clear
Aspect ID: 1
☐ Disable Aspect

Approach
Aspect ID: 2
☐ Disable Aspect

Restricted Proceed
Aspect ID: 0
☐ Disable Aspect

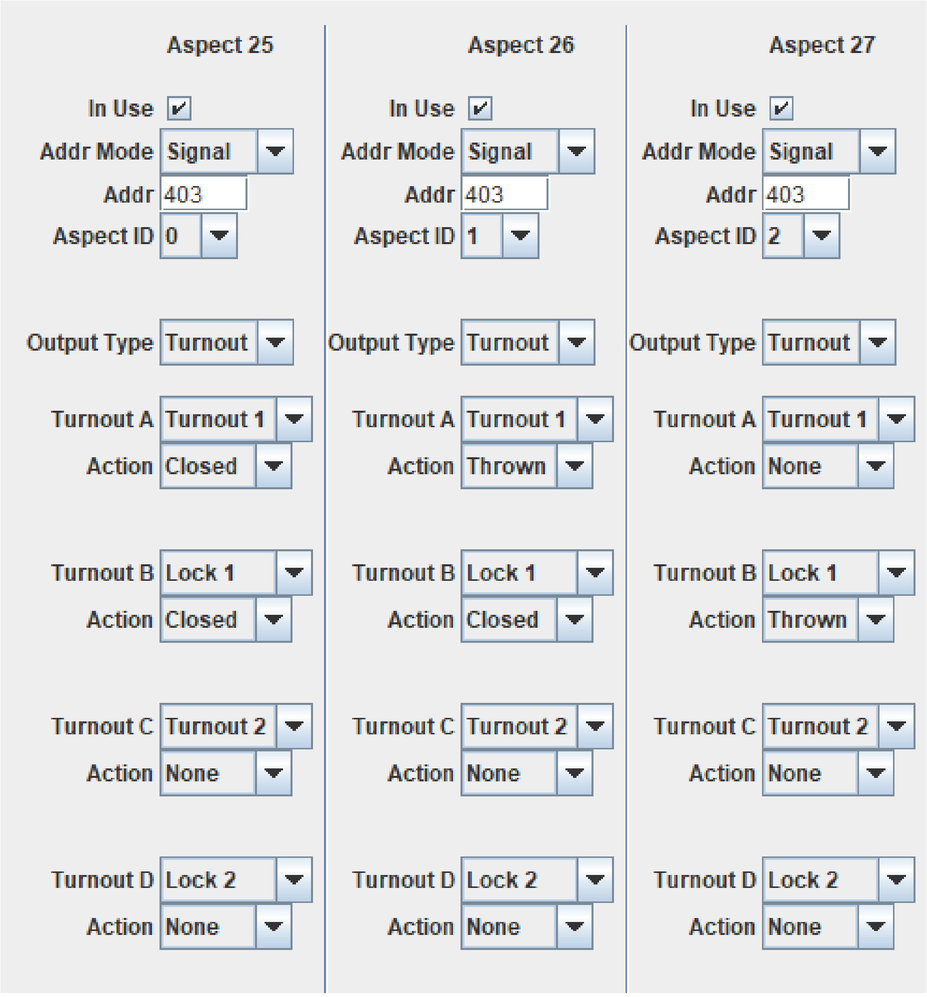
Unlit
Aspect ID: 31
☑ Disable Aspect

Copy Aspect IDs from Mast:

For this example the Signal Address is set to 403 and the Aspects in use are as follows:

Aspect ID	Aspect	Appearance
0	Restricted	Horizontal
1	Clear	Vertical
2	Approach	Angled

Here are the corresponding settings in the QuadLN_S Aspect Table.



	Aspect 25	Aspect 26	Aspect 27
In Use	☑	☑	☑
Addr Mode	Signal	Signal	Signal
Addr	403	403	403
Aspect ID	0	1	2
Output Type	Turnout	Turnout	Turnout
Turnout A	Turnout 1	Turnout 1	Turnout 1
Action	Closed	Thrown	None
Turnout B	Lock 1	Lock 1	Lock 1
Action	Closed	Closed	Thrown
Turnout C	Turnout 2	Turnout 2	Turnout 2
Action	None	None	None
Turnout D	Lock 2	Lock 2	Lock 2
Action	None	None	None

Stop. The first column has Aspect ID set to **0** which is the Stop Aspect. Turnout A Out is set to **Turnout 1**, the *QuadLN_S* Turnout output that drives the servo that controls the Semaphore. The Action is set so that the blade is horizontal. Turnout B Out is set to **Lock 1**, and the Action is set to **Closed** so that the servo is not forced to the midpoint. The Mode for Turnout C and Turnout D is **None** so they have no effect.

Clear. The second column has Aspect ID set to **1** which is the Clear Aspect. Turnout A Out is set to **Turnout 1** and the Action is set so that the blade is vertical. Turnout B Out is set to **Lock 1** and the

Action is set to **Closed** so that the servo is not forced to the midpoint. The Mode for Turnout C and Turnout D is **None** so they have no effect.

Approach. The last column has Aspect ID set to 2 which is the Approach Aspect. We are going to use the Lock to move the servo midway between the Closed and Thrown positions. Turnout B Out is set to **Lock 1** and the Action is set to **Thrown** so that the servo is forced to the travel midpoint. The Mode for Turnouts A, C and D is **None** so they have no effect.

Only the Turnout A and Turnout B settings were needed for the single blade Semaphore in this example. For a double blade Semaphore the Turnout C and Turnout D settings would be used for the second blade and additional aspects would be added as desired.

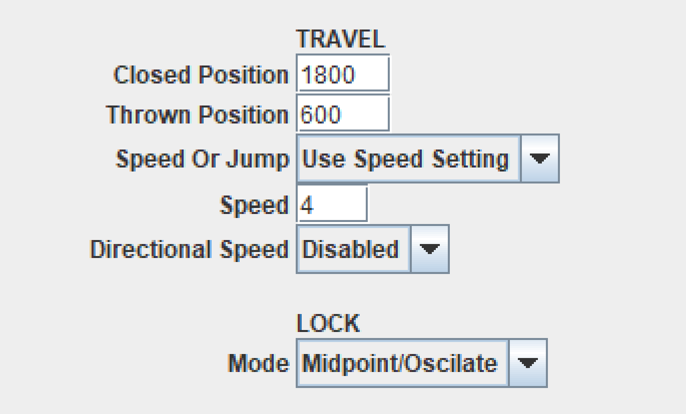
3.3.5 Midpoint/Oscillate

When **Midpoint/Oscillate** mode is selected, setting the Lock Control Switch to Thrown causes the servo to move midway between the Closed and Thrown setpoints. The Turnout output will continually cycle between the Closed and Thrown setpoints when the Lock Control Switch is Closed. The setting of the Turnout Switch position has no effect in this mode. Midpoint/Oscillate may be useful for control of wig-wag crossing signals, pile drivers, walking beam (horse head) oil pumps and other special animation effects.

Note: When a Turnout is in Midpoint/Oscillate mode its Cascade Actions are not triggered and no Output Messages are generated.

3.3.6 Midpoint/Oscillate Examples

Wig-Wag Crossing Signal. A Wig-Wag crossing signal is controlled by Servo turnout LT140 and the associated Lock turnout LT156. To set up the QuadLN_S, enable Midpoint/Oscillate mode for Servo LT140 as described above. When LT156 set to Closed, the signal will continuously swing back and forth. When LT156 is set to Thrown, the signal will stop in the mid-travel position just like the prototype signal.



TRAVEL

Closed Position 1800

Thrown Position 600

Speed Or Jump Use Speed Setting

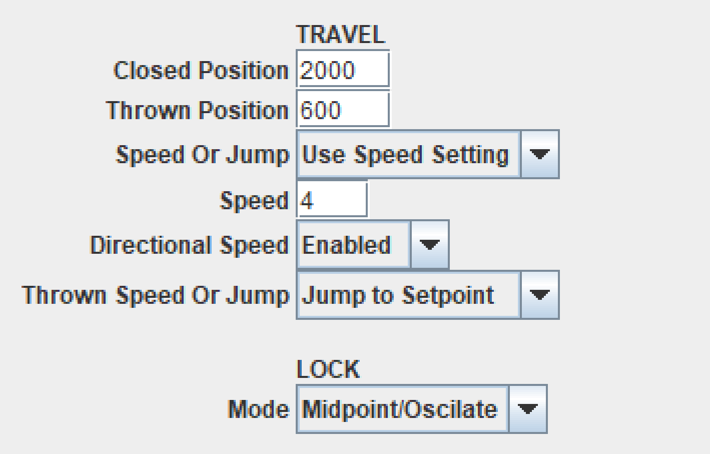
Speed 4

Directional Speed Disabled

LOCK

Mode Midpoint/Oscilate

Pile Driver. Suppose that a pile driver hammer is controlled by Servo turnout LT140 and the associated Lock turnout is LT156. To set up the QuadLN_S, enable Midpoint/Oscillate mode for Servo LT140 as described above. Set Directional Speed to Enabled, set the downward Speed Or Jump to **Jump to Setpoint** and set the upward speed to an appropriately slow value. With LT156 set to Closed, the hammer will continuously move up the leader slowly and then fall rapidly, simulating driving of a pile.



TRAVEL

Closed Position 2000

Thrown Position 600

Speed Or Jump Use Speed Setting

Speed 4

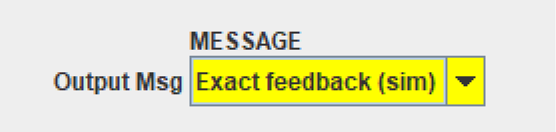
Directional Speed Enabled

Thrown Speed Or Jump Jump to Setpoint

LOCK

Mode Midpoint/Oscilate

3.4 Output Message - Servo Position Feedback without sensors



MESSAGE

Output Msg Exact feedback (sim)

The *QuadLN_S* sends continuous position commands to each servo, so it always knows the servo position. As a result the *QuadLN_S* can provide Turnout Position Feedback messages based on the servo position. This feature is a very effective way to provide visual indication of the turnout state on your dispatcher panel without any sensors!

The available Output Message choices are **No Output Feedback** and **Exact feedback (sim)**. If you are using software such as JMRI to create control panels, **Exact feedback (sim)** will allow you to incorporate field device feedback into those panels. If you are not using software that monitors turnout position feedback or you just don't require turnout position feedback for your layout, then select **No Output Feedback** to eliminate the unneeded LocoNet messages.

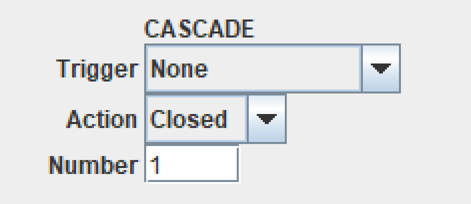
3.4.1 Exact feedback (sim)

Exact feedback normally utilizes two limit switches that are mounted on the turnout: one limit switch turns on when the turnout reaches the Closed position and the other limit switch turns on when the turnout reaches the Thrown position. Feedback messages are sent when either limit switch changes state.

In Exact feedback (sim) mode, the *QuadLN_S* sends the exact feedback messages but bases them on the current servo position. When the Servo first starts to move the limit switch corresponding to the initial position is reported as Low and when the Servo reaches its endpoint the limit switch corresponding to the final position is report as High.

Exact feedback (sim) mode also provides feedback reports when the Turnout output is set to Stall/Relay. In this case the *QuadLN_S* does not know the exact position of the output device so fixed internal travel and speed settings are used to approximate the time it takes a Tortoise to complete its movement when driven at 5 volts.

3.5 Cascade



CASCADE

Trigger None

Action Closed

Number 1

Cascade action refers to a condition where command to one turnout also triggers a command to a second turnout. If the second turnout triggers a command to a third turnout, etc, the command is said to “Cascade” down. Cascade action provides a very handy way to align a route through a yard ladder or crossover by sending only one LocoNet Switch command.

The details of the Cascade settings are shown below, followed by examples to illustrate the basics.

3.5.1 Trigger

The Trigger is the specific Switch command that initiates the Cascade event. When the Trigger command is received, the Cascade Switch command is sent.

The available Cascade Trigger settings for each Turnout are **None, Closed, Thrown** and **Closed or Thrown**. When set to **Closed**, the Cascade action is triggered whenever the Turnout receives a Closed Switch command. When set to **Thrown**, the Cascade action is triggered whenever the Turnout receives a Thrown Switch command. When set to **Closed or Thrown**, the Cascade action is triggered whenever the Turnout receives any Switch command.

3.5.2 Action

The Action is the type of Switch command that is sent to the Cascade turnout when the Trigger condition is met.

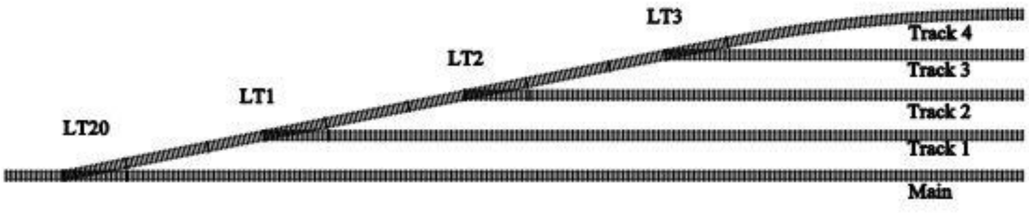
The available Cascade Action settings for each Turnout are **Closed, Thrown, Follow** and **Invert**. When set to **Closed**, a Closed Switch command is sent. When set to **Thrown**, a Thrown Switch command is sent. When set to **Follow**, a Switch command matching the state of the Turnout is sent. When set to **Invert**, a Switch command opposite to the state of the Turnout is sent

3.5.3 Number

The Number is the LocoNet Switch number for the turnout that is to receive the Cascade Switch command.

3.5.4 Cascade Examples

Yard ladder alignment. We have a number of yard tracks accessed via a simple ladder.



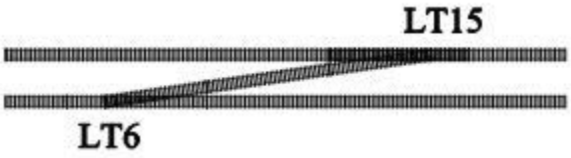
The entry to the yard from the main is controlled by Switch 20: Closed for the main and Thrown for the ladder. Switches on the yard ladder are Closed for the ladder and Thrown to select their specific track, so Switch 1 is Thrown for track 1, Switch 2 is Thrown for track 2, etc. Suppose we wish to fully line the route from the main to any yard track by only issuing one Switch command.

First set Switch 1 Cascade Trigger to **Closed or Thrown**, Cascade Action to **Thrown**, and Cascade Number to **20**. Now whenever Switch 1 receives any command, it will send a message to Switch 20 to move to the Thrown position. To line a train into Track 1, all we need to do is set Switch 1 to Thrown: Switch 20 will get Thrown by the Cascade action.

Next set Switch 2 Cascade Trigger to **Closed or Thrown**, Cascade Action to **Closed**, and Cascade Number to **1**. Now whenever Switch 2 receives any command, it will send a message to Switch 1 to move to the Closed position. Switch 1 will in turn send a command to Switch 20 to move to the Thrown position. To line a train into Track 2, all we need to do is set Switch 2 to Thrown: Switch 1 will get Closed and Switch 20 will get Thrown by the Cascade action.

Repeat this process for each of the remaining tracks. The key is that any change to the turnout that directly selects the track sends a Cascade message to the preceding turnout that aligns it as required.

Crossover. We have a crossover controlled by two turnouts at addresses 6 and 15.



For the normal routes, both 6 and 15 are Closed. For the crossover route, both 6 and 15 are Thrown. Now suppose we want to just send Switch 6 Closed and Switch 6 Thrown to select the routes.

One simple way for this to happen is if Switch 6 sends a Cascade message to Switch 15 to Follow it anytime Switch 6 is commanded to a position. Set Switch 6 Cascade Trigger to **Closed or Thrown**, Cascade Action to **Follow**, and Cascade Number to **15**. Now whenever Switch 6 receives a command, it will send a message to Switch 15 to move to the same position.

3.6 Handling 16 Turnouts

Firmware Version 3.0 supports up to 16 Turnout outputs while previous versions supported a maximum of 8 outputs. The additional Turnout output capability was added especially for users who have moved away from using local pushbutton or toggle switches and instead use software panels on tablets, smartphones and computers. These users no longer require some/all of the IO capability of the Main and Aux IO Ports and they can now utilize those Ports to drive additional Turnout outputs.

4 Main and Aux IO lines in IO Mode

There are 8 digital Input / Output (IO) lines: 4 Main IO lines and 4 Aux IO lines. Each IO line can be used as an Input, an Output, or both. Simple uses of the input capability of an IO line include reporting input state over LocoNet or controlling a Turnout output. More complex actions are also possible, for instance a single IO line can control all of the *QuadLN_S* servos. The output capability of an IO line is typically used to drive an LED panel indicator but can even be used to drive a Relay Board.

The specific behavior of each IO line is configured using the parameters described in this section. To access the IO parameters in JMRI always use **Ops-Mode / Programming On Main**.

4.1 Turnout State Indication on IO line output

Each MAIN or AUX IO line that is configured for IO operation can provide an output signal that indicates the state of one particular Turnout output. The association between an IO line and a specific Turnout state is determined by the EXPANSION PORT and MAIN and AUX IO PORT settings. (See Section 6.1 for information on how to adjust these settings.) Each Group tab in the roster entry shows the appropriate Turnout column label and IO column based on the port settings. Table 4-1 lists the Turnout output state that is assigned to each IO line LED output drive for all relevant configurations.

Port Setting		**LED Indication**		**New in Ver 3.0**
Expansion	**Main and Aux IO**	**MAIN IO**	**AUX IO**	
Relays	Both IO	Turnout 1-4	Turnout 1-4	
Signaling	Both IO	Turnout 1-4	Turnout 1-4	
Servo	Both IO	Turnout 5-8 (Exp 1-4)	Turnout 1-4	
Relays	Main IO, Aux Turnout	Turnout 5-8 (Aux 1-4)		X
Signaling	Main IO, Aux Turnout	Turnout 5-8 (Aux 1-4)		X
Servo	Main IO, Aux Turnout	Turnout 5-8 (Aux 1-4)		X
Relays	Main Turnout, Aux Turnout			X
Signaling	Main Turnout, Aux Turnout			X
Servo	Main Turnout, Aux Turnout			X

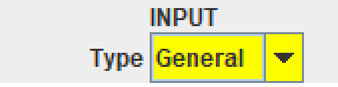
Table 4-1 IO LED Indication to Turnout Output State Association

To configure an IO line, first select the Group tab that corresponds to its associated Turnout output as indicated in Table 4-1 above. Then proceed to adjust the settings.

4.2 Input Response

When an IO line is used as an input, the *QuadLN_S* samples the line 50 times per second and debounces (filters) input changes to suppress electrical noise. This helps to prevent multiple actions from single button pushes or multiple events caused by detector chattering.

4.2.1 Type – Aux Inputs



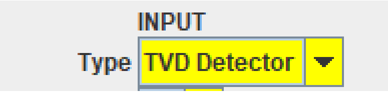
INPUT

Type General

The available Input Types for each Aux Input are **General** and **Detector**. For **General** inputs, a debounce time of 100 milliseconds is used for both Positive and Negative edges. Trigger and Indication options are available. Typical devices connected to **General** inputs are pushbuttons, contact closures, and open collector outputs.

For **Detector** inputs, Trigger is always set to **Both Edges** and Trigger options are not available. The debounce time for positive edges is extended to 2 seconds to ensure that the occupied (low) indication is maintained during momentary loss of contact (current detector) or momentary change in light level (optical detector). Indication options are available.

4.2.2 Type – Main Inputs

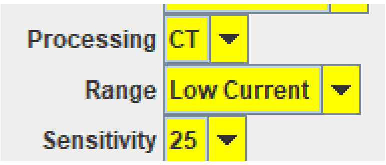


INPUT

Type TVD Detector

The available Input Types for each Main Input are **General** and **TVD Detector**. For **General** inputs, a debounce time of 100 milliseconds is used for both Positive and Negative edges. Trigger and Indication options are available. Typical devices connected to **General** inputs are pushbuttons, contact closures, and open collector outputs.

The **TVD Detector** input is used when connected to an N3IX Engineering Detector Board. For **TVD Detector** inputs, Trigger is always set to **Both Edges** and the debounce time for positive edges is extended to 2 seconds to ensure that the occupied (low) indication is maintained during momentary loss of contact (current detector) or momentary change in light level (future optical detector). Detector Type, Range and Sensitivity options are available for **TVD Detector** inputs, while Trigger and Indication options are not available.


Processing CT
Range Low Current
Sensitivity 25

4.2.3 Detector Processing (Main IO pins only)

When using an N3IX Engineering Detector Board, Current Transformer (CT) or Infrared (IR) detection processing can be individually assigned to each Main Input pin. Leave this set to CT since the detector board only supports Current Transformer detection at the present time.

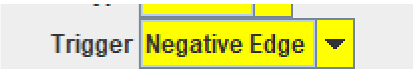
4.2.4 Detector Range (Main IO pins only)

When using a Current Detector Board the **Range** may be set for either **Low Current** or **High Current**. The **Low Current** range allows the sensitivity to be adjusted between approximately 0.3mA and 4mA. The **High Current** range allows the sensitivity to be adjusted between approximately 4mA and 20mA.

4.2.5 Sensitivity (Main IO pins only)

When a Main IO pin is set for **TVD Detector**, the input is processed using the Sensitivity value to determine if the signal represents the Occupied or Unoccupied state. The Occupied state is reported as a 0 and the Unoccupied state is reported as a 1. The higher the Sensitivity value, the more sensitive the input and the lower the signal value required for detection. Set the Sensitivity value just high enough to ensure reliable detection. If false detection occurs, decrease the Sensitivity value.

4.2.6 Trigger


Trigger Negative Edge

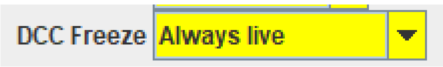
Input actions are initiated only when an input change occurs that matches the Trigger condition.

For **General** inputs the available Input Trigger settings for each Input are **None, Positive Edge, Negative Edge** and **Both Edges**. When set to **Positive Edge**, the input action is triggered whenever the Input changes from the Low level to the High level. When set to **Negative Edge**, the input action is triggered whenever the Input changes from the High level to the Low level. When set to **Both Edges**, the input action is triggered whenever any Input change occurs.   When set to **None**, no action is triggered by the input and the input line is not sampled.

For **Detecto**r inputs the Input Trigger is fixed at **Both Edges** and the Trigger selection is not available.

When a sampled input line changes state and then remains stable for the debounce time, a LocoNet Sensor message is generated.

4.2.7 DCC Freeze

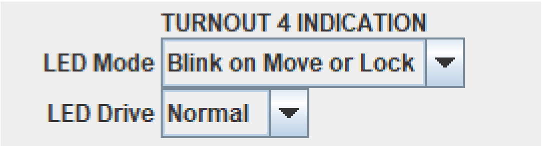


DCC Freeze Always live

Some input devices give false indications when DCC is not present.   For example, a short in a DCC power district may cause all block occupancy detectors in that district to indicate that their blocks are unoccupied whether or not trains are present.   The *QuadLN_S* has the ability to monitor the DCC signal and to freeze an input at its previous level when the DCC signal is lost, thus preventing false reports from detectors during short circuit conditions.

The available settings are **Always Live** and **Freeze on DCC loss**.   When set to **Always Live**, input action follows the Trigger setting independent of the DCC state.   This is generally the appropriate setting when the input is used for Turnout control.   When set to **Freeze on DCC lost**, input action is disabled when no DCC signal is present on the DCC input pins.   This is generally the appropriate setting when the input is used for a block occupancy detector.

4.3 Indication



TURNOUT 4 INDICATION

LED Mode Blink on Move or Lock

LED Drive Normal

When the Input line is not being sampled, the *QuadLN_S* has the capability to treat the line as an Output and can drive an LED indicator to display the state of the Turnout.   Since the input sampling takes place very quickly and at a rate higher than the eye can detect, the LED indicator will appear to be on continuously with no discernible flicker.   (If using input devices other than Fascia Controllers, please see the general Quad datasheet for information on simultaneously wiring switches and LEDs to the Input lines.)

For **TVD Detecto**r inputs the LED Indication feature is not available.

4.3.1 LED Mode

When the LED indicator is enabled, the steady LED indication shows whether the Turnout is Closed or Thrown.   The LED can be set to blink rapidly while the Turnout is moving, thus providing feedback that travel is complete when the indicator again becomes steady.   When the associated Turnout is Lockable, the LED can be set to blink slowly to provide feedback on the whether local control of the

Turnout is Locked or Unlocked. (The global Lock Indicator setting determines whether blinking indicates the Locked or the Unlocked state – see Section 6.4.5.)

For **TVD Detecto**r inputs the LED Mode feature is not available.

The available LED Mode settings for each Input are **Disable – Input only, Steady Drive, Blink on Move, Blink on Move or Lock, Blink on Lock, Lock State, Position** and **Blink on Thrown**. When set to **Steady Drive**, the output indication shows the Turnout position. If the Turnout is in motion the output indication changes when the Turnout passes the midpoint. Choose this mode when the IO pin is driving a 5 Volt relay board for frog power. When set to **Blink on Move**, the output indication shows the commanded Turnout position and blinks rapidly while the Turnout is moving. When set to **Blink on Move or Lock**, the output indication shows the commanded Turnout position, blinks rapidly while the Turnout is moving and blinks slowly when the Turnout is **Lockable** and the state of the lock matches the Lock Indicator setting. When set to **Blink on Lock**, the output indication shows the commanded Turnout position and blinks slowly when the Turnout is **Lockable** and the state of the lock matches the Lock Indicator setting. When set to **Lock State**, the output indication shows the commanded Lock position instead of the commanded Turnout position. When set to **Position**, the output indication shows the actual Turnout position instead of the commanded position, and only changes when the Turnout reaches the commanded position and stops. When set to **Blink on Thrown**, the output indication blinks when the Turnout is commanded to the Thrown position. In this mode the Main and Aux IO indications blink on opposite phases, which allows them to be used to animate grade crossing flashers when the Turnout is controlling a crossing gate. When set to **Disable – Input only**, no output drive is provided and the line is always left as an input. This is appropriate when the input device is a detector for example.

Except in the **Steady** and **Position** modes, the LED output changes gradually (fades) from one state to another to simulate the appearance of an incandescent panel lamp turning on or off. In the **Steady** and **Position** modes the output changes abruptly in one step, so that the output signal can more easily be used to trigger other types of devices if desired.

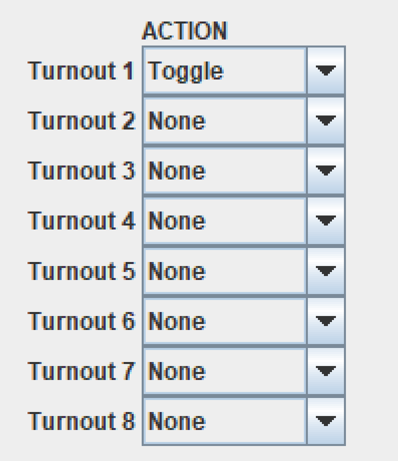
4.3.2 LED Drive

If the LED panel wiring results in the opposite indication from what is desired, the *QuadLN_S* can simply invert the output drive to the LED. This eliminates the need to rewire the LED indicators or remount the Fascia Controller.

For **TVD Detecto**r inputs the LED Drive feature is not available.

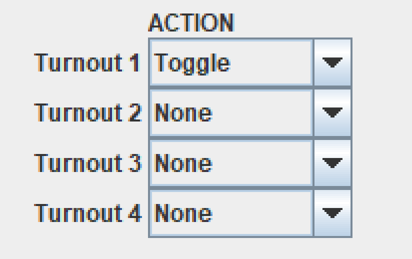
The available LED Sense settings for each Input are **Normal** and **Inverted**. When set to **Normal**, the LED output drive is High when the Turnout is Closed and Low when Thrown. When set to **Inverted**, the LED output drive is Low when the Turnout is Closed and High when Thrown.

4.4 Action



ACTION

Turnout 1	Toggle
Turnout 2	None
Turnout 3	None
Turnout 4	None
Turnout 5	None
Turnout 6	None
Turnout 7	None
Turnout 8	None



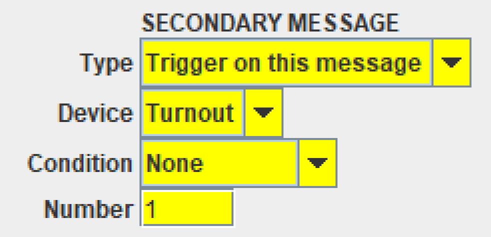
ACTION

Turnout 1	Toggle
Turnout 2	None
Turnout 3	None
Turnout 4	None

Each *QuadLN_S* input can control any or all of the Turnout outputs. This allows inputs to trigger routes without requiring a LocoNet connection. The local action of any input on each of the *QuadLN_S* Turnouts can be set independently. (The local action on a Turnout is only performed if that Turnout is not locked.)

The available Turnout action settings for each Input are **None, Toggle, Follow, Invert, Closed, Thrown, Lock Toggle, Lock Follow, Lock Invert, Lock Closed and Lock Thrown**. When set to **None**, the state of the corresponding Turnout is unaffected by the input. When set to **Toggle**, the state of the corresponding Turnout is set to the opposite of its current state. When set to **Follow**, the state of the corresponding Turnout is set to Closed if the Input is High and set to Thrown if the Input is Low. When set to **Invert**, the state of the corresponding Turnout is set to Closed if the Input is Low and set to Thrown if the Input is High. When set to **Closed**, the state of the corresponding Turnout is set to Closed. When set to **Thrown**, the state of the corresponding Turnout is set to Thrown. When one of the **Lock** settings is selected, the state of the corresponding Lock is set accordingly.

4.5 Secondary Message



SECONDARY MESSAGE

Type	Trigger on this message
Device	Turnout
Condition	None
Number	1

Each Input can receive or send a Secondary message. This can be used to implement more complex logic for yards, panels or other functions. If set to send a Secondary message, when the Input Trigger condition set in Section 4.2.1 is met the Secondary message is sent. If set to receive a Secondary message, a LocoNet sensor or turnout message that matches the Secondary Trigger fires the Turnout Actions that are configured for the Input.

4.5.1 Type

The available Secondary Type settings for each Input are **Trigger on this message** and **Send this message**. When set to **Trigger on this message**, the specified Actions for the input are executed when the Condition below is met. When set to **Send this message**, the Secondary message is sent when the Input Trigger condition is met, and the message content is determined by the Condition below.

4.5.2 Device

The available Secondary Device settings for each Input are **Turnout** and **Sensor**. When set to **Turnout**, the Secondary message is a Switch command message. When set to **Sensor**, the Secondary message is a General Sensor message.

4.5.3 Condition

The available Secondary Condition settings for each Input are **None, Hi / Closed, Lo / Thrown** and **Both / Follow**.

If Type is **Trigger on this message:** When set to **Hi / Closed**, the trigger condition is a Sensor Hi or Switch Closed message. When set to **Lo / Closed**, the trigger condition is a Sensor Lo or Switch Thrown message. When set to **Both / Follow**, the trigger condition is a Sensor or Switch message in either state. When set to **None**, the Secondary message function is disabled.

If Type is **Send this message:** When set to **Hi / Closed**, the message sent is a Sensor Hi or Switch Closed message. When set to **Lo / Closed**, the message sent is a Sensor Lo or Switch Thrown message. When set to **Both / Follow**, the message sent is a Sensor or Switch message with the state matching the Input state. When set to **None**, the Secondary message function is disabled.

4.5.4 Number

The Switch or Sensor number for the Secondary message.

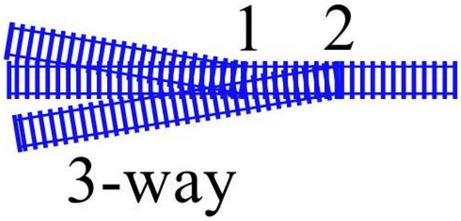
4.5.5 Secondary Message Examples

All Mainline Turnouts to Closed. Suppose we have many mainline turnouts controlled by *QuadLN_S* boards. Each turnout is controlled by a pushbutton on its associated Main or Aux IO line. We would like to be able to set all the turnouts to their Closed position with a single Switch command, say Switch 200 Closed. For each Main and AUX IO line, set the Secondary Message Type to **Trigger on this message**, the Device to **Turnout**, the Condition to **Hi / Closed** , and the Number to **200**. Now sending a single Switch command lines all the turnouts to the Closed position.

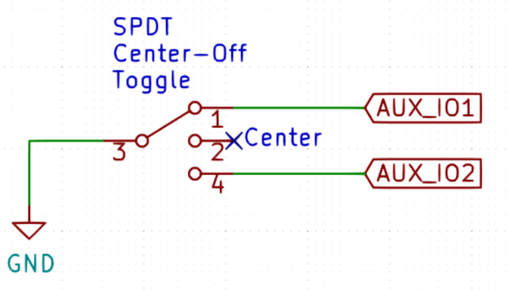
Unlock a Mainline Turnout using a keyswitch. We have a local fascia pushbutton set up for control of Turnout 17 but wish to Lockout (disable) the pushbutton unless a keyswitch is actuated. Let’s suppose the Lock Address for Turnout 17 is set to 33 (see Section 6.2.3 for information on the Lock Address). Wire the keyswitch to an available IO line on a *QuadLN_S* board, then adjust the settings for the IO line as follows: Trigger to **Both Edges**; Action to **None** for each Turnout; Secondary Message Type to **Send this message**, Device to **Turnout**, Condition to **Both/Follow** and Number to 33 which is the Lock Address for Turnout 17. Now opening the keyswitch sends a LocoNet Turnout 33 Closed command which unlocks Turnout 17, and closing the keyswitch sends a LocoNet Turnout 21 Thrown command which locks Turnout 17. A suitable indicator on the Dispatcher panel could monitor Turnout 33 and provide an indication when Turnout 17 is unlocked.

4.6 3-Way Turnout Settings

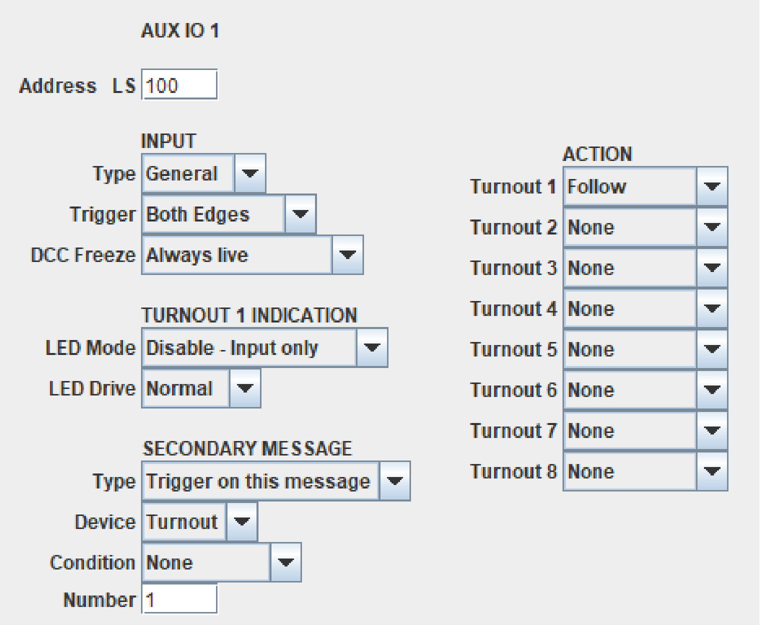
4.6.1 Example – Standard 3-Way Turnout



Suppose that a standard 3-way turnout is driven by Servo 1 and Servo 2. The 3-way turnout is controlled using a 3-position, SPDT center-off toggle switch wired to *QuadLN_S* inputs Aux IO1 and Aux IO2.



Wire the SPDT center-off toggle switch as shown in this schematic. Next configure the *QuadLN_S*, Aux IO1 and Aux IO2 as shown below.


AUX IO 1

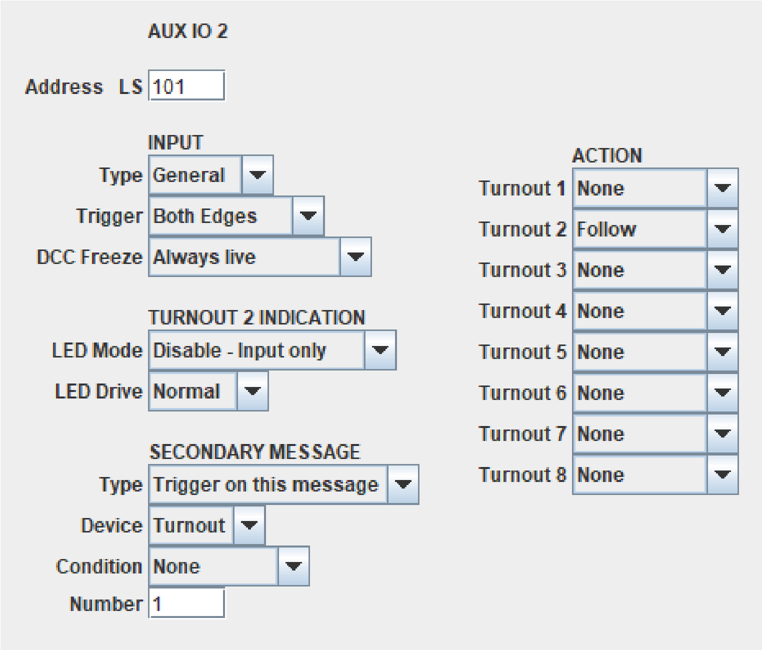
Address LS 100

INPUT
Type General
Trigger Both Edges
DCC Freeze Always live

TURNOUT 1 INDICATION
LED Mode Disable - Input only
LED Drive Normal

SECONDARY MESSAGE
Type Trigger on this message
Device Turnout
Condition None
Number 1

ACTION
Turnout 1 Follow
Turnout 2 None
Turnout 3 None
Turnout 4 None
Turnout 5 None
Turnout 6 None
Turnout 7 None
Turnout 8 None



AUX IO 2

Address LS 101

INPUT

Type: General

Trigger: Both Edges

DCC Freeze: Always live

TURNOUT 2 INDICATION

LED Mode: Disable - Input only

LED Drive: Normal

SECONDARY MESSAGE

Type: Trigger on this message

Device: Turnout

Condition: None

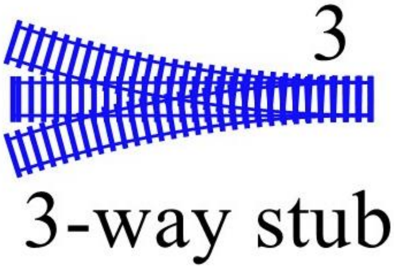
Number: 1

	ACTION
Turnout 1	None
Turnout 2	Follow
Turnout 3	None
Turnout 4	None
Turnout 5	None
Turnout 6	None
Turnout 7	None
Turnout 8	None

Set the Trigger to **Both Edges** for Aux IO1 and for Aux IO2. Set the LED Mode for Aux IO1 and for Aux IO2 to **Disable – Input only**. Set Aux IO1 Turnout 1 ACTION to **Follow**. Set Main IO2 Turnout 2 ACTION to **Follow**.

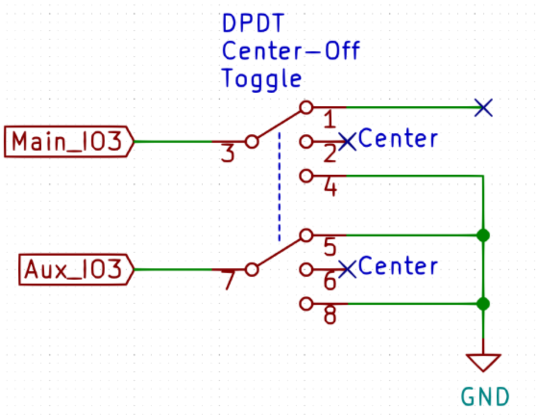
Explanation. When the toggle switch is in the upper position, Servo 1 will be THROWN, Servo 2 will be CLOSED, and the upper diverging route will be selected. When the toggle switch is in the center positions, both Servo 1 and Servo 2 will be CLOSED and the center route will be selected. When the toggle switch is in the lower position, Servo 2 will be THROWN and the lower diverging route will be selected.

4.6.2 Example - 3-Way Stub Turnout

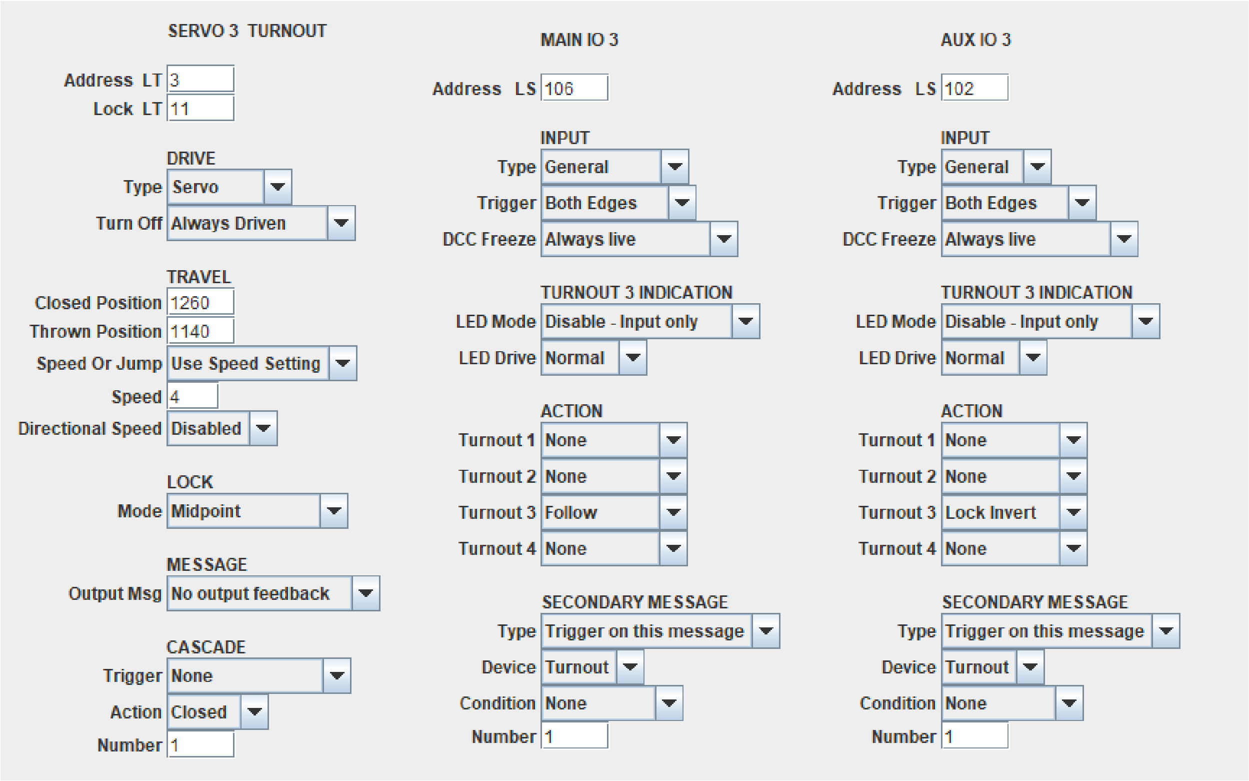


Suppose that a 3-way stub turnout is driven by Servo 3. We use **Midpoint** mode for Servo 3 so that whenever the Lock is THROWN the servo moves to the travel Midpoint, selecting the center route through the turnout. When the Lock is CLOSED the servo moves to a travel endpoint, selecting one of the diverging routes through the turnout.

In this example the 3-way stub turnout is controlled using a 3-position, DPDT center-off toggle switch wired to QuadLN_S inputs Main IO3 and Aux IO3.



Wire the DPDT center-off toggle switch as shown in this schematic. Next configure the *QuadLN_S*, Main IO3 and Aux IO3 as shown below.



SERVO 3 TURNOUT

Address LT 3
Lock LT 11

DRIVE
Type Servo
Turn Off Always Driven

TRAVEL
Closed Position 1260
Thrown Position 1140
Speed Or Jump Use Speed Setting
Speed 4
Directional Speed Disabled

LOCK
Mode Midpoint

MESSAGE
Output Msg No output feedback

CASCADE
Trigger None
Action Closed
Number 1

MAIN IO 3

Address LS 106

INPUT
Type General
Trigger Both Edges
DCC Freeze Always live

TURNOUT 3 INDICATION
LED Mode Disable - Input only
LED Drive Normal

ACTION
Turnout 1 None
Turnout 2 None
Turnout 3 Follow
Turnout 4 None

SECONDARY MESSAGE
Type Trigger on this message
Device Turnout
Condition None
Number 1

AUX IO 3

Address LS 102

INPUT
Type General
Trigger Both Edges
DCC Freeze Always live

TURNOUT 3 INDICATION
LED Mode Disable - Input only
LED Drive Normal

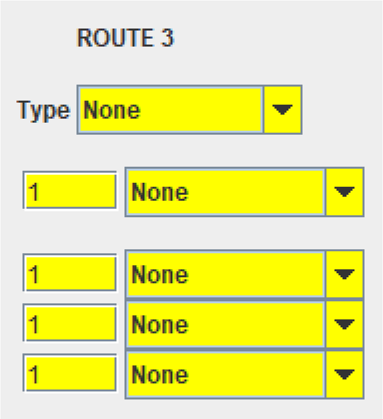
ACTION
Turnout 1 None
Turnout 2 None
Turnout 3 Lock Invert
Turnout 4 None

SECONDARY MESSAGE
Type Trigger on this message
Device Turnout
Condition None
Number 1

Set the Lock for Servo 3 to **Midpoint** mode. Set the Trigger to **Both Edges** for Main IO3 and for Aux IO3. Se the LED Mode for Main IO3 and for Aux IO3 to **Disable – Input only**. Set Aux IO3 Turnout 3 ACTION to **Lock Invert**. Set Main IO3 Turnout 3 ACTION to **Follow**.

Explanation. When the toggle switch is in the upper or lower position, the Servo 3 Lock will be CLOSED and the servo will move to the endpoint that is selected by the toggle switch position. When the toggle switch is in the center position the Servo 3 Lock will be THROWN and the servo will move to the midpoint.

5 Routes



ROUTE 3

Type None

1 None

1 None

1 None

1 None

Routes are arbitrary sequences of Switch and Sensor messages that are sent when a trigger condition is met. The *QuadLN_S* has 8 routes, each containing up to 4 entries consisting of Switch commands, Sensor state messages or Signal aspects. Routes can be set to Expand the prior route, so it is possible to have a single route with as many as 32 entries. This makes it easy to do things like create an arbitrary track alignment by sending a single command. To access the following parameters in JMRI always use **Ops-Mode / Programming On Main**.

5.1 Type

The available Route Types are **None, Normal Route, Selector** and **Expand (Prior) Route**. A **Normal Route** is only triggered by the first entry and always sends the remaining entries exactly as entered. A **Selector Route** is triggered by any of the entries and sends the remaining entries in the opposite state from that entered. This action implements a "1-of-n" selection. An **Expanded Route** simply adds its entries to the prior route.

5.2 Entries

Each Route entry consists of an Address and Device Type / Action. For Switch messages the available Actions are **Closed** and **Thrown** and for Sensor messages the available Actions are **Sensor Hi** and **Sensor Lo.** For these messages, the Address is the LocoNet device number. For Signals the available action is **Aspect**, and the Address represents the number of an Aspect on this *QuadLN_S*.

5.3 Route Examples

Align a Route when a button is pressed. We would like to be able to line a route that involves many turnouts by pressing a pushbutton. Wire the pushbutton to an available Aux IO line on a *QuadLN_S* board, then adjust the settings for the Aux IO line so that the Input Type is set to **General**. Set the Action to **None** for each Servo. Pick a Route in any *QuadLN_S* and set the Type to **Normal Route**, set the first entry to the **Aux IO address** and **Sensor Hi**, and then enter the **Switch address** and **Closed** or **Thrown** in the following entries for each turnout in the route. Use Expanded Routes as needed. Now pressing the button lines the entire route.

Lock All Mainline Turnouts. We have many mainline turnouts controlled by *QuadLN_S* boards. Each turnout is controlled by a pushbutton on its Main IO line. We would like to be able to lock out all the pushbuttons (for continuous running during an Open House for example) by sending a single Switch command, say Switch 200 Closed. Pick a Route in any *QuadLN_S* and set the Type to **Normal Route**, set the first entry to **200 Closed**, and then enter the **Lock address** and **Thrown** in the following entries for each mainline turnout lock. Use Expanded Routes as needed. Now sending a single Switch command locks all the mainline turnouts.

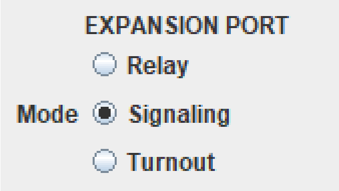
Panel track selection indicator. We have a yard panel with an LED for each track. The LEDs are controlled via a stationary decoder(s) using Switch commands. Setting a particular Switch address to Closed turns an LED on, and setting it to Thrown turns the LED off. We would like to be able to send a single Switch command to light the LED for the selected track and have all the other LEDs go out. Pick a Route in any *QuadLN_S*, set the Type to **Selector** and then enter the **Switch address** and **Closed** in the route entries for each Switch address that controls an LED. Use Expanded Routes as needed. Now sending a single Switch command turns on the LED for the selected track and turns off the LEDs for the non-selected tracks.

Set signals for a turnout. We have a number of signals protecting Switch 200 and would like to set their Aspects just based on the position of a turnout. Pick a Route in the *QuadLN_S* that controls the signals and set the Type to **Normal Route**, set the first entry to **200 Closed**, and then enter the desired **Aspect Number** for each signal and **Aspect** in the following entries in the route. Use Expanded Routes as needed. Next repeat these steps for another **Normal Route**, this time with **200 Thrown** as the first entry and the desired Aspect Number in each of the following entries. Now sending any command to Switch 200 will cause each of the signals to display the appropriate Aspect.

6 The QuadLN_S Pane

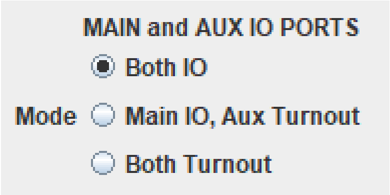
The *QuadLN_S* has a number of address settings and special configuration settings. Most users will set the Turnout Start Address, Main IO Start Address and Aux IO Start Address along with the Expansion Port and Main and Aux IO Ports configuration (the Board Address is shown for information only and should have already been set if required). The remaining special configuration settings can generally be left at their default values. To access the following parameters in JMRI always **use Ops-Mode / Programming On Main**.

6.1 Port Configuration



EXPANSION PORT

Mode
○ Relay
● Signaling
○ Turnout



MAIN and AUX IO PORTS

Mode
● Both IO
○ Main IO, Aux Turnout
○ Both Turnout

The *QuadLN_S* supports multiple configurations for the **Main** IO, **Aux** IO and **Expansion** Ports to fit your application. The **Servo** Port always supports Turnouts 1-4. Table 6-1 shows the assignments for the other Ports in each possible configuration.

Port Settings		Turnouts				IO			New in Ver 3.0
Expansion	**Main and Aux IO**	**Total**	**Main**	**Aux**	**Exp**	**Total**	**Detector Board**	**Signal Board**	
Relays	Both IO	4				8			
Signaling	Both IO	4				8		X	
Turnout	Both IO	8			5-8	8			
Relays	Both IO	4				4	X		
Signaling	Both IO	4				4	X	X	
Turnout	Both IO	8			5-8	4	X		
Relays	Main IO, Aux Turnout	8		5-8		4			X
Signaling	Main IO, Aux Turnout	8		5-8		4			X
Turnout	Main IO, Aux Turnout	12		5-8	9-12	4			X
Relays	Main IO, Aux Turnout	8		5-8		0	X		X
Signaling	Main IO, Aux Turnout	8		5-8		0	X	X	X
Turnout	Main IO, Aux Turnout	12		5-8	9-12	0	X		X
Relays	Both Turnout	12	5-8	9-12		0			X
Signaling	Both Turnout	12	5-8	9-12		0		X	X
Turnout	Both Turnout	16	5-8	9-12	13-16	0			X

Table 6-1 Firmware Version 3.0 Port Configurations

6.1.1 Expansion Port

The Expansion Port may be used to directly drive four 5 Volt Relay Boards, 4 additional Turnout outputs, or 24 LEDs using the N3IX Engineering Signal board. This setting determines which signals are supplied to the expansion port. When set to **Relay**, relay board drive signals are supplied that change state when the servo is at mid travel. When set to **Signaling**, LED drive signals are supplied to control 24 LEDs using the N3IX Engineering Signal Board. When set to **Turnout**, drive signals are supplied for 4 additional Turnout outputs.

Note: After writing a new Port setting to the *QuadLN_S*, it is a good idea to **Read** each of the Group tabs. This will ensure that the JMRI template you are viewing reflects the values in your *QuadLN_S*.

6.1.2 Main and Aux IO Ports

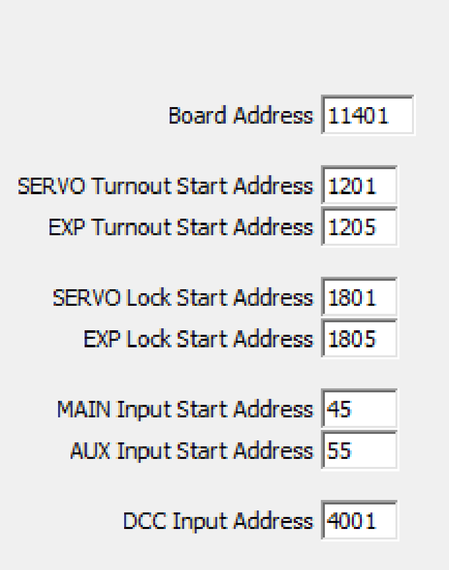
The Main and Aux IO Ports may be used as Input/Output ports or Turnout ports. This setting determines the configuration of each of the ports. When set to **Both IO**, both the Main and Aux IO Ports are configured as Input/Output ports. When set to **Main IO, Aux Turnout**, the Main IO Port is configured as an Input/Output port and the Aux IO Port is configured to supply turnout drive signals. When set to **Both Turnout**, both the Main IO and Aux IO Ports are configured to supply turnout drive signals.

The Main IO Port must be configured as an IO port when using the Detector board.

Note: After writing a new Port setting to the *QuadLN_S*, it is a good idea to **Read** each of the Group tabs. This will ensure that the JMRI template you are viewing reflects the values in your *QuadLN_S*.

6.2 Addresses

Firmware version 3.3 added more flexibility in setting addresses for each port.



Board Address 11401

SERVO Turnout Start Address 1201

EXP Turnout Start Address 1205

SERVO Lock Start Address 1801

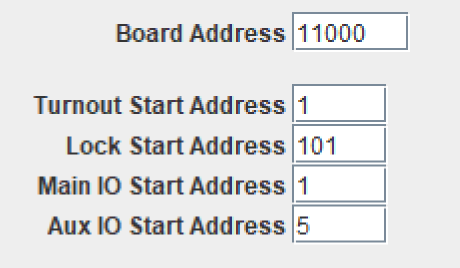
EXP Lock Start Address 1805

MAIN Input Start Address 45

AUX Input Start Address 55

DCC Input Address 4001

Version 3.3



Board Address 11000

Turnout Start Address 1

Lock Start Address 101

Main IO Start Address 1

Aux IO Start Address 5

Pre-Version 3.3

6.2.1 Board Address

The Board Address is the device address used when programming the device in **Ops Mode**. See Section 1.7.1 for more info. A special procedure is required to change the device address, and this procedure is described in Section 7.1. This procedure is intended to prevent inadvertent device address changes.

6.2.2 SERVO, MAIN, AUX and EXP Turnout Start Address

The LocoNet Switch addresses for the *QuadLN_S* Turnout outputs on each IO Port are set using these fields. Each *QuadLN_S* IO Port uses 4 consecutive Switch addresses for Turnout control. The Ranges must not overlap. Avoid using addresses 1017-1020 for any Turnout (those addresses are used by

LocoNet special interrogation commands). The address for each Turnout is displayed on its Group tab. Turnout Addresses ranges for each port should be unique. See Section 1.7.2 for more info on the Turnout Start Address.

6.2.3 SERVO, MAIN, AUX and EXP Lock Start Address

The LocoNet Switch addresses for the *QuadLN_S* Locks on each IO Port are set using these fields. Each The *QuadLN_S* IO Port uses 4 consecutive Switch addresses for Lock control, or a single address for the entire port if the “One Lock Per Port” feature is enabled (see 6.4.13). Avoid using addresses 1017-1020 for any Lock (those addresses are also used by LocoNet special interrogation commands). The address for each Lock is displayed on its Group tab. Lock Addresses for each port should be unique. If Turnout and Lock address ranges overlap, any Lock with the same address as a Turnout will not be accessible via a LocoNet Switch command. See Section 1.7.3 for more info on the Lock Start Address.

6.2.4 Main IO Start Address

The LocoNet Sensor addresses for the *QuadLN_S* Main IO are set using this field. The *QuadLN_S* uses 4 consecutive Sensor addresses for the Main IO. See Section 1.7.4 for more info on the Main IO Start Address. This parameter is only displayed when the Main IO port is set for IO.

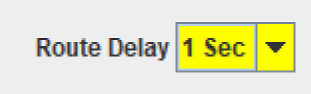
6.2.5 Aux IO Start Address

The LocoNet Sensor addresses for the *QuadLN_S* Aux IO are set using this field. The *QuadLN_S* uses 4 consecutive Sensor addresses for the Aux IO. See Section 1.7.5 for more info on the Aux IO Start Address. This parameter is only displayed when the Aux IO port is set for IO.

6.2.6 DCC Input Address

The *QuadLN_S* can report changes in the state of the DCC input using Loconet Sensor messages. This field specifies the Sensor Address to use for those reports. This parameter is only displayed when the **Report DCC Change** option Is enabled. See 6.4.14 below for more details on this feature.

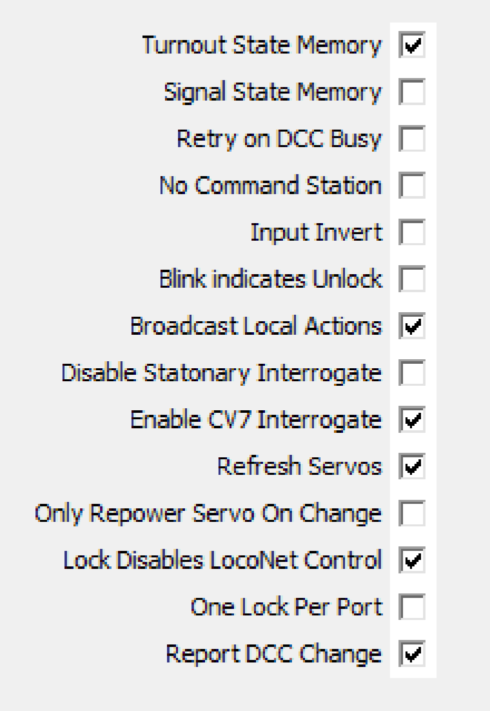
6.3 Route Delay



Route Delay 1 Sec

Each action in a local route can occur immediately after the route is triggered or it can occur after a fixed delay. The length of the delay is controlled by this setting. The available **Route Delay** settings are **1 Sec, 2 Sec, 4 Sec** and **8 Sec**. The default is **1 Sec**.

6.4 Options



Turnout State Memory ☑
Signal State Memory □
Retry on DCC Busy □
No Command Station □
Input Invert □
Blink indicates Unlock □
Broadcast Local Actions ☑
Disable Statonary Interrogate □
Enable CV7 Interrogate ☑
Refresh Servos ☑
Only Repower Servo On Change □
Lock Disables LocoNet Control ☑
One Lock Per Port □
Report DCC Change ☑

6.4.1 Turnout State Memory

This setting controls whether the *QuadLN_S* records the Turnout positions when they change and then restores all Turnouts to their last known position at power-up. By default the box is **Checked** meaning that the state of each Turnout will be **Restored on Power-up**. If the box is **Unchecked** then each Turnout will be set to **Closed at Power-up**.

6.4.2 Signal State Memory

This setting controls whether the *QuadLN_S* records the Signal aspects when they change and then restores all Signals to their last known aspect at power-up. By default the box is **Unchecked** meaning all Signal outputs will be **Off at Power-up**. If the box is **Checked** then the state of each Signal output will be **Restored on Power-up**. This parameter is only displayed when the Expansion IO port is set for Signaling.

6.4.3 Retry on DCC Busy

This setting controls whether the *QuadLN_S* attempts to retry Switch Requests when the Command Station reports that the DCC bus is busy. The default value is **Unchecked** meaning no **Retry Switch Requests** will be sent.

Note: This section is somewhat technical. Users that don't need any Switch Request sent by the *QuadLN_S* to reach DCC-only stationary decoders can simply ignore this section. The information below is included so that those users with complex configurations that include DCC-only stationary decoders can better understand the system performance tradeoffs.

When the *QuadLN_S* sends a Switch Request on LocoNet as part of a Cascade, Secondary Message or Route, the Command Station attempts to send the command over the DCC bus. This is done so that DCC-only stationary decoders will also receive the command. However due to the difference in bandwidths, LocoNet Switch Requests can arrive at the Command Station faster than the Command Station can send them out over DCC. When this happens the Command Station internally buffers the Switch Requests in order to give the DCC bus a chance to catch up. When the Command Station receives a Switch Request and its internal buffer is already full (or with some Command Stations if track power is turned off so that DCC commands cannot be sent), the Command Station responds with a special LocoNet message to indicate that the Switch Request did not make it to the DCC bus.

When a Switch Request does not get to the DCC bus, the *QuadLN_S* has two options: ignore the failure or send the Switch Request again. Sending the Switch Request again usually works out fine, but if there are a large number of Switch Requests being are generated on LocoNet (from one or more devices) then the retries can start to pile up. When this happens performance degrades until the Command Station is able to get the DCC side caught up, and under most circumstances things get resolved without any noticeable impact.

If you need the *QuadLN_S* to be able to send Switch Requests to DCC-only stationary decoders then **Check** this option and the *QuadLN_S* will **Retry Switch Requests** when required. If you have not set up the *QuadLN_S* to send Switch Requests to DCC-only stationary decoders, then **Uncheck** this option since in this case it doesn't matter whether a command sent by the *QuadLN_S* makes it to the DCC bus and thus there is no reason to create additional LocoNet traffic by resending Switch Requests.

Finally, if you are using a Digitrax command station and have no DCC-only stationary decoders at all, you can set OPSW 27 to Closed (Bushby bit). Closing OPSW 27 tells the command station not to send any Switch Requests over the DCC bus. This eliminates the entire DCC backlog issue and allows Switch Requests to be sent at full LocoNet speed under all conditions.

6.4.4 No Command Station

This setting is applicable to standalone **LocoNet** installations where no command station is present. The *QuadLN_S* can expand the responses to certain programming messages to also provide the replies normally sent by a Command Station. When no Command Station is present, these responses will suppress error messages in JMRI or other software that expects to see the command station replies. The default value is **Unchecked** meaning that a Command Station is present. **Check** this option if **No Command Station** is present on **LocoNet**.

6.4.5 Input Invert

This setting allows the Sensor Input Polarity to be inverted for use with certain layout control programs that require Active Low inputs to be reported as High. The default value is **Unchecked** meaning inputs are not inverted. When **Checked** the signal level is **Inverted** at the pin prior to any other processing. Note that input settings may need to be adjusted whenever this option is changed. This parameter is only displayed when the Main IO port or both the Main and Aux IO ports are set for IO.

6.4.6 Blink Indicates Unlock

This setting determines whether the IO line LED indicators configured to display their Turnout Lock status will blink when the Turnout is in the Locked state or when it is in the Unlocked state. The default value is **Unchecked** and a blinking LED indicates the Locked state. When **Checked** a blinking LED indicates the **Unlocked** state. This parameter is only displayed when the Main IO port or both the Main and Aux IO ports are set for IO.

6.4.7 Broadcast Local Actions

This setting controls whether the *QuadLN_S* makes local Turnout and Lock actions visible on LocoNet. The default value is **Checked** meaning **Local Actions are Broadcast**. This parameter is only displayed when the Main IO port or both the Main and Aux IO ports are set for IO.

The *QuadLN_S* can perform the local Turnout or Lock actions that are triggered by input changes (see Section 4.4 for information on local actions) in two slightly different ways. In the first approach the *QuadLN_S* directly changes the Turnouts and Locks to the new values. In the second approach the *QuadLN_S* generates LocoNet Switch Request commands just as would happen if some other LocoNet device were sending the commands. The second approach is used by default and provides LocoNet visibility to the state changes caused by the local actions the same as with any LocoNet Switch Request. The second approach also allows the *QuadLN_S* input changes to trigger routes. If LocoNet visibility is not needed and local routes are not in use, then you can **Uncheck** option and eliminate the unnecessary LocoNet message traffic. Regardless of which setting you choose, the local actions will be performed whether or not the LocoNet is connected.

6.4.8 Disable Stationary Interrogate

(Advanced) This setting controls whether the *QuadLN_S* responds normally to LocoNet Stationary Interrogate messages or ignores those messages. The default value is **Unchecked** meaning that the board provides **Normal Interrogate Responses**.

JMRI queries the LocoNet using the Stationary Interrogation method when it starts, and Command Stations do it every time layout power is turned on. Interrogation can generate a great deal of LocoNet traffic on large layouts, resulting in slow response and missed messages. The **Disable Stationary Interrogate** option can speed system response on large layouts when used in concert with the **Enable CV7 Interrogate** option discussed in Section 6.4.9. When the **Disable Stationary Interrogate** option is **Checked** the *QuadLN_S* will ignore Stationary Interrogate messages, reducing the LocoNet traffic during Interrogation.

For most layouts the Stationary Interrogation method works acceptably and this option should be left **Unchecked**. If your layout is large and you have decided to implement a manual query using the CV7 Interrogate feature, then you may wish to **Check** this option to **Disable Stationary Interrogate** responses.

6.4.9 Enable CV7 Interrogate

(Advanced) This setting controls whether the *QuadLN_S* responds to Ops Mode writes to CV7 with an Interrogate Response. The default value is **Unchecked** meaning Writes to CV7 are ignored.

It is possible for a PC monitoring the LocoNet to get out of sync with the state of the physical Turnouts or Sensors. There is a standard method provided by LocoNet for obtaining device status called "Interrogation". Interrogation queries the entire layout at once and on large layouts many LocoNet devices may attempt to respond at the same time. This can result in one or more responses getting lost. The *QuadLN_S* provides an alternative approach that allows the state of individual devices to be queried one-by-one. To query an individual *QuadLN_S* directly requires a series of 8 Ops Mode Writes to CV7. Start by writing a value of 0 to CV7 on a specific device. The addressed QuadLN_S will respond by reporting the state of 1-2 turnouts, 1-2 Lock turnouts (if used), and one Sensor input (if used). Repeat the Ops Mode CV7 write with values 1-7 to receive the full state of the device. A simple JMRI script can be written to perform these steps automatically or on demand.

For most layouts the normal Interrogation method works acceptably and this option can be left **Unchecked**. If your layout is large and JMRI sometimes is not in sync with your layout, **Check** this option and then you can use a script to query all your boards one at a time.

6.4.10 Refresh Servos

This setting causes servos set to **Off When Stopped** to be briefly pulsed every ten seconds. Some brands of servos are very sensitive to static discharge and it can cause these servos to move away from the setpoint. If that were to happen, the Refresh pulse will bring them back to the setpoint. The only downside is a quiet tick might be heard once every 10 seconds when the Refresh happens. The default value is **Unchecked** meaning that no Refresh is performed. When **Checked** all servos are briefly pulsed once every 10 seconds.

6.4.11 Only Repower Servo On Change

This setting determines whether a servo set to **Off When Stopped** is repowered when a position command is received and it is already set to the commanded position. Previously setting a servo to the current position did not have any effect. This meant that to restore a servo that was off setpoint, due to manual movement or electrical disturbance, the commanded position had to be cycled back and forth. The default behavior is now to always repower (pulse) a servo for 1 second when a position command is received locally or via LocoNet, whether or not the setpoint was changed. This optional setting has been provided in case the old behavior is desired for some reason. The default value is **Unchecked** meaning that a servo is always repowered when a position command is received. When **Checked** a servo set to **Off When Stopped** is only repowered if it is commanded to move to a new position.

6.4.12 Lock Disables LocoNet Control

This setting determines whether the locking a Turnout also disables LocoNet control of the turnout (by throttles, touchscreens, etc). The default value is **Unchecked** meaning that locking a Turnout only disables Local control via the Main and Aux IO. When **Checked** both local control and LocoNet control of the Turnout are disabled.

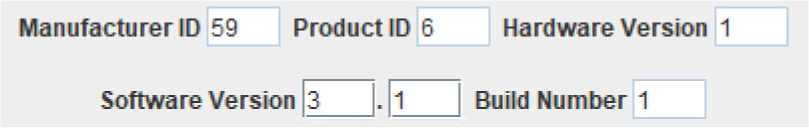
6.4.13 One Lock Per Port

This setting determines whether each Turnout in an IO Port has a unique address are shares the same address. The default value is **Unchecked** meaning that each Turnout has its own Lock Address. When **Checked** the Lock for all 4 Turnouts is controlled using a single Lock Address.

6.4.14 Report DCC Change

This setting determines whether the QuadLN_S sends a Sensor message when it detects a change in the state of the DCC input (from present to absent, or from absent to present). The default value is **Unchecked** meaning that changes in the DCC input are not reported. When **Checked** changes in the state of the DCC input are reported using Sensor messages at the specified DCC Input Address. This feature can be used to display DCC power status and provide short circuit indications on LEDs or JMRI panels.

6.5 Device Information



Manufacturer ID 59 Product ID 6 Hardware Version 1

Software Version 3 . 1 Build Number 1

This section contains information mainly useful for debugging problems and performing updates.

6.5.1 Manufacturer ID

The QuadLN_S uses 59 as the Manufacturer ID. Writing special values to this parameter when in Service Mode can reset the device – see Section 7.5.

6.5.2 Product ID

This value identifies the product and helps prevent installing incorrect firmware updates. The Product ID for the *QuadLN_S* is 6.

6.5.3 Hardware Version

This value identifies the product hardware revision and helps prevent installing incorrect firmware updates.

6.5.4 Software Version

This value identifies the product software revision and helps prevent installing incorrect firmware updates.

6.5.5 Build Number

This value is an additional software identifier and enables better factory support.

7 Maintenance

7.1 Change the Device Address

The Device Address can be changed whenever required. (The *QuadLN_S* is *never* connected to the programming track.)

- If there is no existing Roster Entry for this *QuadLN_S*, create a new *QuadLN_S* Roster Entry with the desired Device Address.

  ---or ---

  If there is an existing Roster Entry for this *QuadLN_S,* open the Roster Entry and change the Board Address on the QuadLN_S tab to the desired address. Now Save the Roster Entry and then Close the Roster Entry.
- Select the **Programming On Main** option.
- Select the desired *QuadLN_S* entry from the Roster and then click the “Program…” button. The roster entry will open after a few seconds.
- Go to the **Quad-LN_S** tab
- Enter the desired address in the **Board Address** field. Choose a unique address between 11001 and 16383. A unique 5-digit address is required to avoid conflicts. See Section 1.7.1 for information about Device Addresses.
- Go to the **CVs** tab.
- Install the **ADDR** jumper on your *QuadLN_S*. The **ACTV** and **MSG** LEDs will start flashing alternately.
- **Write** CV17 and then **Write** CV18. (Ignore the “Programmer error” message, this is normal.) The LEDs should stop flashing.
- Remove the **ADDR** jumper.

REMINDER: Remember to **remove** the **ADDR** jumper when finished. Failure to remove the **ADDR** jumper can result in unintended changes to the Board Address.

7.2 Identify the Device Address

If you have lost track of the Device Address for a *QuadLN_S.* the address can be identified using the following simple procedure.

- Open the Monitor Loconet window in JMRI
- Install the **ADDR** jumper on the *QuadLN_S* that has the unknown Device Address. The **ACTV** and **MSG** LEDs will start flashing alternately.
- Remove the **ADDR** jumper.
- Examine the Monitor Loconet window for a new message similar to this:
  Programming Response: Byte Read on Main Track (Ops Mode) Was Successful: **Decoder address 11000**: CV1 value 0 (0x00, 00000000b).

7.3 Update the Firmware

The *QuadLN_S* design incorporates a bootloader that permits the firmware to be updated in the field using JMRI. The bootloader portion of the firmware is protected and cannot be accidentally overwritten during the update process. Normally the *QuadLN_S* user settings are not affected by the firmware update process. However it is always best to ensure your *QuadLN_S* roster entry is up to date before performing a firmware update so that your settings can be easily restored later if required.

To perform a firmware update, follow the steps below.

1. Download the new *QuadLN_S* firmware update package from quadln.n3ix.com and place the ".hex" file in your JMRI User Files Location folder.
2. Select **Download Firmware** from the JMRI **LocoNet** menu.
3. For older versions of JMRI, the **File format** must be manually set by clicking the **16 bit** File format button if it isn't already selected. If you have a newer version of JMRI and the **File format** buttons are not selectable, just proceed to the next step.
4. Click the **Select** button and then open the firmware file you downloaded in Step 1.
5. For older versions of JMRI there will be a **Read file** button. If that button is present, click it to load the file into the downloader. If you have a newer version of JMRI and there is no **Read file** button, just proceed to the next step.
6. Connect the Remote Alignment Board to the *QuadLN_S*.
7. Simultaneously press and hold down all three buttons on the Remote Alignment Board.
8. After about 2 seconds when the **MSG LED** dims, release the buttons and the **MSG LED** should go dark. The *QuadLN_S* is now in bootloader mode.
9. In JMRI, click **Download** to begin downloading the file to the *QuadLN_S*. The **MSG LED** will toggle as each message is received. The **ACTV LED** will flash briefly as each block of memory is written.
10. If the download completes successfully, the *QuadLN_S* will resume normal operation.
11. If an error occurs, the **ACTV LED** will flash continuously about once/second. The **MSG LED** will indicate the error code.
    - **MSG LED** On = communication error - bad message received.
    - **MSG LED** Off = configuration error. This can be one of the following:
        - firmware file error - wrong manufacturer code or wrong product code. The file you are attempting to download is not for this product.
        - hardware version error – the firmware is not for this version of the board. Obtain the proper firmware file for the hardware version of your board.
    - **MSG LED** Flashing once/second = software version error – the firmware isn't newer than the existing firmware. The current firmware is as new or newer than the firmware you are attempting to download. If the download is still desired, select **Don't check software version** in the JMRI Firmware Downloader, cycle power to the board and repeat the process starting with step 7 above.

Notes:

The *QuadLN_S* software is overwritten as the download progresses. Once the download starts, if it is interrupted or if a communication error occurs then the *QuadLN_S* software will probably be corrupted. However the bootloader will be intact, so just click **Abort,** remove power to the board, press and hold the Select button then reapply power to the board. The **ACTV LED** should be lit and the **MSG LED** should be dark. Now repeat the download process starting with step 9 above.

The default **Delay (msec)** value is 200. Most systems can perform the download at a faster rate without any problems. You might want to try a setting 100 or even 50 and see if it works on your layout.

If a configuration error occurs, the *QuadLN_S* did not accept the download and the previous software is intact. If you still want to perform the update, resolve the issue and then repeat the download process. When errors occur repeatedly you may want to try increasing the **Delay** value as described in the previous paragraph.

Multiple *QuadLN_S* boards can be updated at the same time. It is recommended that one board be updated and tested first to be sure everything is working properly before updating the remaining boards.

7.4 Update the Decoder Template

If the firmware update contains new features, you will need the latest JMRI decoder template file to take advantage of those features. The latest version of JMRI should contain the latest decoder template. If you are not running the latest version of JMRI and a *QuadLN_S* feature appears to be missing when you open a roster entry, please follow these steps to download and install the new decoder template files.

- Download the latest *QuadLN_S* JMRI update package from quadln.n3ix.com.
- Follow the instructions in the readme file to copy the new decoder files and new tvd folder (if provided) into the appropriate locations.

7.5 Perform a Factory Reset

The *QuadLN_S* settings can always be restored to their factory defaults. Factory Reset is performed by writing a special value to CV 8 (Manufacturer ID) using Service Mode/Programming Track **Programmer**. (The *QuadLN_S* is *never* connected to the programming track.) Menu choices have been provided in the JMRI *QuadLN_S* template file to make this easy to do. Alternate manual methods that do not require JMRI are also described below.

REMINDER: Remember to **remove** the **ADDR** jumper when finished. Failure to remove the **ADDR** jumper can result in unintended changes to the Board Address.

7.5.1 Restore Selective Parameters to Factory Defaults using JMRI

- Remove all locomotives from the programming track.
- Open the Service-Mode(Programming Track) **Programmer** or open the Roster and select the Programming Track option.
- Select any *QuadLN_S* roster entry and open it.
- Select the **Reset** menu and then click **Factory Reset…**
- Select **Restore All CVs…**, **Restore Addresses…** or **Restore Servo and Input CVs...** to specify which parameters to restore.
- Install the **ADDR** jumper.   The **ACTV** and **MSG** LEDs should start flashing alternately.
- Click **OK** to perform the reset.
- Remove the **ADDR** jumper.
- **Close** the Roster Entry and close the Service Mode Programmer if open.
- If you chose "Restore Addresses..." above, cycle power to the *QuadLN_S* so the changes can take effect.

7.5.2 Restore All Parameters to Factory Defaults using the Remote Align Board

- Remove power from the *QuadLN_S*.
- Install the **ADDR** jumper.
- Connect the remote alignment board to the *QuadLN_S*.
- Press and hold both the Up and Down buttons.
- Reapply power to the *QuadLN_S*.
- Continue holding both the Up and Down buttons until the **ACTV** and **MSG** LEDs flash alternately for 1 second and then stop flashing.
- Release the buttons.
- Remove the **ADDR** jumper.

8 CV Tables

8.1 Full CV Table

CV #	Description	Range	Default
1-3	Reserved		
4	Software Version	0-255	-
5	Software Build Number	0-255	-
6	Hardware Revision	0-255	-
7	Product ID	6	6
8	Manufacturer ID	59	59
9-10	Reserved		
11	Configuration	Table 8-3	134
12	Configuration2	Table 8-4	8
13-16	Reserved		
17,18	Board Long Address	Table 8-2	11000 (234, 248)
19-23	Reserved		
24	Configuration3	Table 8-5	0
25-32	Reserved		
33-34	SERVO Turnout Start Address	0-2044	0 (0,0)
35-36	MAIN Turnout Start Address	0-2044	4 (4,0)
37-38	AUX Turnout Start Address	0-2044	8 (8,0)
39-40	EXP Turnout Start Address	0-2044	12 (12.0)
41-42	SERVO Lock Start Address	0-2044	16 (16,0)
43-44	MAIN Lock Start Address	0-2044	20 (20,0)
45-46	AUX Lock Start Address	0-2044	24 (24,0)
47-48	EXP Lock Start Address	0-2044	28 (28,0)
49-50	Reserved		
51-52	MAIN Input Start Address	0-4092	0 (0,0)
53-54	AUX Input Start Address	0-4092	4 (4,0)
55-56	Reserved		
57-58	DCC Input Address	0-4092	4000(159,15)
59-250	Signal Aspect 1-48	Table 8-17	
251-256	Reserved		
257-264	Servo 1-8 Speed	Table 8-6	4
265-272	Servo 1-8 Speed Thrown	Table 8-7	4
273-288	Servo 1-8 Closed Position	Table 8-8	1260 (4,236)
289-304	Servo 1-8 Thrown Position	Table 8-9	1140 (4,116)
305-320	Servo 1-8 Cascade Address	Table 8-10	0 (0,0)
321-324	Aux IO 1-4 Option	Table 8-11	13
325-328	Main IO 1-4 Option	Table 8-11	13
329-332	Main IO 1-4 Detector	Table 8-12	16
333-340	Aux IO 1-4 Secondary	Table 8-13	0 (0,0)
341-348	Main IO 1-4 Secondary	Table 8-13	0 (0,0)
349-364	Aux IO 1-4 Action	Table 8-14	Table 8-14
365-380	Main IO 1-4 Action	Table 8-14	Table 8-14

CV #	Description	Range	Default
381-388	Servo 9-16 Speed	Table 8-6	4
389-396	Servo 9-16 Speed Thrown		4
397-412	Servo 9-16 Closed Position	Table 8-8	1260 (4,236)
413-428	Servo 9-16 Thrown Position	0-2400	1140 (4,116)
429-444	Servo 9-16 Cascade Address	Table 8-10	0 (0,0)
445-512	Reserved		
513-536	Signal Lamp 1-24	Table 8-21	160
537-632	Signal Address 1-48	Table 8-19 Table 8-20	0
633-656	Signal LED 1-24	Table 8-22	159
657-704	Signal Aspect Config 1-48	Table 8-18	0
705-768	Reserved		
769-776	Route 1 Entry 1-4	Table 8-15	0 (0,0)
777-784	Route 2 Entry 1-4	Table 8-15	0 (0,0)
785-792	Route 3 Entry 1-4	Table 8-15	0 (0,0)
793-800	Route 4 Entry 1-4	Table 8-15	0 (0,0)
801-808	Route 5 Entry 1-4	Table 8-15	0 (0,0)
809-816	Route 6 Entry 1-4	Table 8-15	0 (0,0)
817-824	Route 7 Entry 1-4	Table 8-15	0 (0,0)
825-832	Route 8 Entry 1-4	Table 8-15	0 (0,0)
833-896	Reserved		
897-904	Route 1-8 Control	Table 8-16	0
905-	Reserved		

For address entries, the associated LocoNet device number is (address + 1).

Table 8-1 Full CV Table

8.2 Board Long Address CV Format

CV	17								18							
Bit	1	1	h	h	h	h	h	h	l	l	l	l	l	l	l	l
Default	1	1	1	0	1	0	1	0	1	1	1	1	1	0	0	0

h = High address bits
l = Low address bits

Table 8-2 Board Long Address CV Format

8.3 Configuration CV Format

CV	11							
Bit	b	lb	i	i	si	se	e	e
Default	0	0	0	0	0	1	1	0

b = Busy retry: 0 = Disabled, 1 = Enabled

lb = Lock blink state: 0 = Locked, 1 = Unlocked

i = IO mode: 0 = Both IO, 1 = Main IO/Aux Turnout, 2=Reserved, 3=Both Turnout

si = Signal State Memory: 0 = Off, 1 = On

se = Turnout State Memory: 0 = Off, 1 = On

e = Expansion Port: 0 = Relay Drive, 1 = Signal Drive, 2 = Turnout Drive, 3 = Reserved

Table 8-3 Configuration CV Format

8.4 Configuration2 CV Format

CV	12							
Bit	v	i	rd	rd	l	c	ll	p
Default	0	0	0	0	1	0	0	0

v = CV7 Interrogate: 0 = Ignore, 1 = Respond

i = Interrogate request: 0 = Normal response, 1 = Ignore

rd = Route Delay: 0 = 1 Sec, 1 = 2 Sec, 2 = 4 Sec, 3 = 8 Sec

l = Local Action: 0 = Internal, 1 = Visible on LocoNet

c = No Command Station: 0 = Disabled, 1 = Enabled

ll = Lock Disables LocoNet Control: 0 = no, 1 = yes

p = Invert Input: 0 = Normal, 1 = Inverted

Table 8-4 Configuration2 CV Format

8.5 Configuration3 CV Format

CV	12							
Bit	r	r	r	r	rd	ls	cp	sr
Default	0	0	0	0	0	0	0	0

sr = Servo Periodic Refresh : 0 = Disable, 1 = Enable

cp = Servo Repower Upon Command: 0 = Always, 1 = Only on Change

ls = Lock Span: 0 = 1 Turnout, 1 = 4 Turnouts

rd = Report DCC: 0 = Disabled, 1 = Enabled

r = Reserved

Table 8-5 Configuration3 CV Format

8.6 Servo Speed CV Format

CV								
Bit	d	j	s	s	s	s	s	s
Default	0	0	0	0	0	1	0	0

d = Drive Type: 0 = pulse, 1 = steady

j = Speed Or Jump mode: 0 = Speed, 1 = Jump

s = Speed: 0 = slowest, 63 = fastest

Table 8-6 Servo Speed CV Format

8.7 Servo Thrown Speed CV Format

CV								
Bit	d	j	s	s	s	s	s	s
Default	0	0	0	0	0	1	0	0

d = Drive Off: 0 = never, 1 = when stopped

j = Speed Or Jump mode: 0 = Speed, 1 = Jump

s = Speed: 0 = slowest, 63 = fastest

Table 8-7 Servo Thrown Speed CV Format

8.8 Servo Closed Position CV Format

CV																
Bit	d	l	l	m	h	h	h	h	l	l	l	l	l	l	l	l
Default	0	0	0	0	0	1	0	0	1	1	1	0	1	0	0	0

d = Directional Speed: 0 = Disabled, 1 = Enabled

l = Lock mode: 0 = None, 1 = Local Lockout, 2 = Midpoint, 3 = Midpoint/Oscillate

m = Message: 0 = None, 1 = Exact Feedback

h = High position bits

l = Low position bits

Table 8-8 Servo Closed Position CV Format

8.9 Servo Thrown Position CV Format

CV																
Bit	a	a	a	a	h	h	h	h	l	l	l	l	l	l	l	l
Default	0	0	0	0	0	1	0	0	1	1	1	0	1	0	0	0

a = IO Action (turnout 9-12 in 12 turnout mode only, see Table 8-14 for bit values)

m = Message: 0 = None, 1 = Exact Feedback

h = High position bits

l = Low position bits

Table 8-9 Servo Thrown Position CV Format

8.10 Servo Cascade Address CV Format

CV																
Bit	t	t	a	a	x	h	h	h	l	l	l	l	l	l	l	l
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

t = Trigger: 0 = None, 1 = Closed, 2 = Thrown, 3 = Closed or Thrown

a = Action: 0 = Closed, 1 = Thrown, 2 = Follow, 3 = Invert

h = High address bits

l = Low address bits

Table 8-10 Servo Cascade Address CV Format

8.11 IO Option CV Format

CV								
Bit	f	ty	s	l	l	l	tr	tr
Default	0	0	0	0	1	1	1	0

f = DCC Freeze: 0 = Always live, 1 = Freeze on DCC loss

ty = Input Type: 0 = General, 1 = Detector

s = LED Sense: 0 = Normal, 1 = Inverted

l = LED Mode: 0 = Disable, 1 = Steady, 2 = Blink on Move, 3 = Blink on Move or Lock, 4 = Blink on Lock, 5 = Lock State, 6 = Position, 7 = Blink on Thrown

tr = Trigger: 0 = None, 1 = Positive Edge, 2 = Negative Edge, 3 = Both Edges

Table 8-11 IO Option CV Format

8.12 Main IO Detector Option CV Format – Detector Input

CV								
Bit	t	r	g	s	s	s	s	s
Default	0	0	0	1	1	0	1	0

t = Input Type: 0 = CT, 1 = IR

r = Reserved

g = range: 0 = Low, 1 = High

s = Sensitivity: 0-31

Table 8-12 Main IO Option CV Format – Detector Input

8.13 IO Secondary CV Format

CV																
Bit	t	d	c	c	h	h	h	h	l	l	l	l	l	l	l	l
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

t = Type: 0 = Trigger on this message, 1 = Send this message

d = Device: 0 = Turnout, 1 = Sensor

c = Condition: 0 = None, 1 = Hi/Closed, 2 = Lo/Thrown, 3 = Both/Follow

h = high address bits

l = low address bits

Table 8-13 IO Secondary CV Format

8.14 IO Action CV Format

CV																
Bit	h	h	h	h	g	g	g	g	f	f	f	f	e	e	e	e
Default	H	H	H	H	G	G	G	G	F	F	F	F	E	E	E	E
CV																
Bit	d	d	d	d	c	c	c	c	b	b	b	b	a	a	a	a
Default	D	D	D	D	C	C	C	C	B	B	B	B	A	A	A	A

h = Turnout 8 (8+ Turnout mode only)
g = Turnout 7 (8+ Turnout mode only)
f = Turnout 6 (8+ Turnout mode only)
e = Turnout 5 (8+ Turnout mode only)
d = Turnout 4
c = Turnout 3
b = Turnout 2
a = Turnout 1

0 = None, 1 = Toggle, 2 = Follow, 3 = Invert, 4 = Closed, 5 = Thrown, 9 = Lock Toggle, 10 = Lock Follow, 11 = Lock Invert, 12 = Lock Closed, 13 = Lock Thrown

Default: For the Turnout in the same group, Default = 1; for all other Turnouts, Default = 0.

Table 8-14 IO Action CV Format

8.15 Route Entry CV Format

CV	(even)								(odd)							
Bit	r	a	a	a	h	h	h	h	l	l	l	l	l	l	l	l
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

r = Reserved
a = Action: 0= None, 1 = Closed Turnout, 2 = Thrown Turnout, 3 = Hi Sensor, 4 = Lo Sensor
h = high address bits
l = low address bits

Table 8-15 Route Entry CV Format

8.16 Route Control CV Format

CV								
Bit	r	r	r	r	r	r	t	t
Default	0	0	0	0	0	0	0	0

r = Reserved

t = Type: 0 = None, 1 = Basic, 2 = Selector, 3 = Expand

Type = 3 is not used on Route 1.

Table 8-16 Route Control CV Format

8.17 Signal Aspect CV Format

CV								
Bit	b	m	m	p	p	p	p	p
Default	0	0	0	0	0	0	0	0

b = Bicolor: 0 = No, 1 = Yes

mm = Mode: 0 = None, 1 = Steady, 2 = Flash Phase A, 3 = Flash Phase B

ppppp = Pin: 0-31

i = In Use: 0 = No, 1 = Yes

a = Address Mode: 0 = Signal Mode, 1 = Turnout Mode

r = Reserved

Table 8-17 Signal Aspect CV Format

8.18 Signal Aspect Config CV Format

CV								
Bit	i	m	tS	r	r	r	r	r
Default	0	0	0	0	0	0	0	0

i = In Use: 0 = No, 1 = Yes

a = Address Mode: 0 = Signal Mode, 1 = Turnout/Turnout Mode

tS = Turnout/Sensor Selector: 0 = Turnout, 1 = Sensor

r = Reserved

Table 8-18 Signal Aspect Config CV Format

8.19 Signal Aspect Address CV Format – Signal Mode

CV	(even)								(odd)							
Bit	a	a	a	a	a	h	h	h	l	l	l	l	l	l	l	l
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

a = Aspect: 0-31

h = Mast high address bits

l = Mast low address bits

Table 8-19 Signal Aspect Address CV Format – Signal Mode

8.20 Signal Aspect Address CV Format – Turnout Mode

CV	(even)								(odd)							
Bit	m	m	m	m	s	h	h	h	l	l	l	l	l	l	l	l
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

a = Mast: 0-15

s = Turnout Sense: 0 = Thrown, 1 = Closed

h = Turnout high address bits

l = Turnout low address bits

Table 8-20 Signal Aspect Address CV Format – Turnout Mode

8.21 Signal Lamp CV Format

CV								
Bit	f	fr	t	r	r	r	r	r
Default	1	0	1	0	0	0	0	0

r = Reserved

t = Type: 0 = Non-Searchlight, 1 = Searchlight

fr = Flash Rate: 0 = Signal, 1 = Crossing

f = Fade: 0 = No, 1 = Yes

Table 8-21 Signal Lamp CV Format

8.22 Signal LED CV Format

CV								
Bit	g	e	e	b	b	b	b	b
Default	1	0	0	1	1	1	1	1

g = Group Start: 0 = No, 1 = Yes

e = Effect: 0 = Off, 1 = On, 2 = Bicolor, 3 = Reserved

b = Brightness: 0-31

Table 8-22 Signal LED CV Format

9 FCC Information

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and the receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

Changes and Modifications not expressly approved by N3IX Engineering voids the user's authority to operate this equipment under and be in compliance with Federal Communication Commission rules. N3IX Engineering believes this device would be unlikely to cause any RFI problems if properly installed.