

# Introduction to the Control nslinktool:

This will be a very basic introduction to the nslinktool application. This is “as is” and is a “work in progress”.

## DESCRIPTION

Comtrols' nslinktool is an X11 based GUI front end that allows the user to perform common administrative tasks related to the DeviceMaster. The Control nslinktool is written in Python using the tkinter package and both must be installed to run nslinktool.

## REQUIREMENTS

1. tkinter package installed on your Linux system
2. The latest Control nslink driver installation. ([ftp://ftp.control.com/dev\\_mstr/rts/drivers/linux/](ftp://ftp.control.com/dev_mstr/rts/drivers/linux/))

## ENVIRONMENT

Comtrols' nslinktool does not use any environment variables other than those used by the Python and tkinter packages.



# Operations

Start by going to your nslink driver installation directory or `usr/sbin` to run "make" then "make install". This will install nslinktool in `/usr/sbin` which is normally on root's PATH after installing tkinter and run `./nslinktool&`

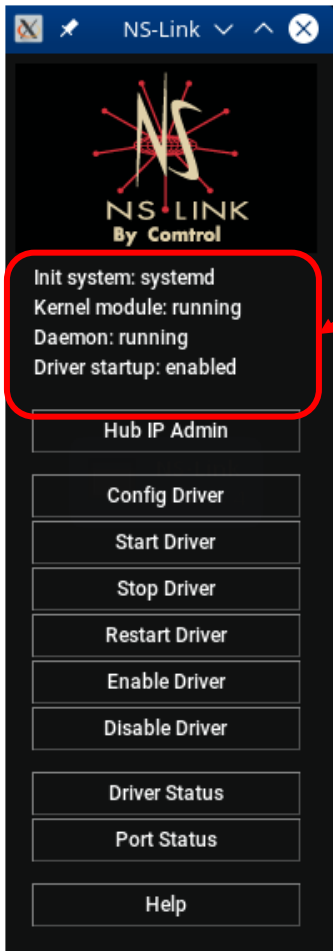
The Control nslinktool performs three major functions:

1. Start/stop/restart the NS-Link device driver and its associated daemon. It also allows the user to control whether the driver will be started automatically at system boot.
2. Edit the configuration file `/etc/nslink.conf`.
3. Examine and configure the IP addressing information in DeviceMasters.

## Driver Administration

The initial window that is seen when nslinktool is started is the driver administration window. It contains four lines of status information and seven buttons:

This should start the Control NS-Link GUI as seen here:



The screenshot shows a window titled "NS-Link" with a logo at the top. Below the logo, the status information is displayed in a red-bordered box:

```
Init system: systemd
Kernel module: running
Daemon: running
Driver startup: enabled
```

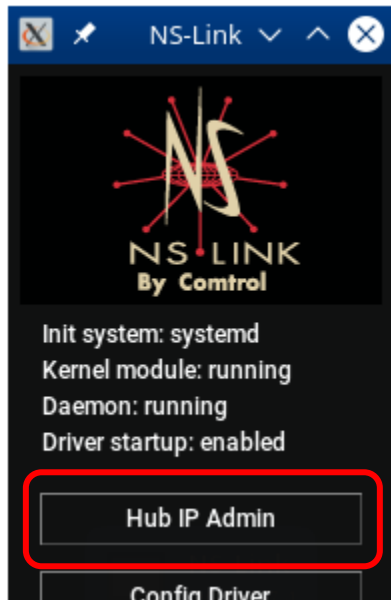
Below the status box are several buttons: Hub IP Admin, Config Driver, Start Driver, Stop Driver, Restart Driver, Enable Driver, Disable Driver, Driver Status, Port Status, and Help.

A red arrow points from the status box to a text box on the right that explains each line of status information.

Status Line	Description
Init system: systemd	Shows what kind of init system is loaded as the system.
Kernel module: running	Shows whether the nslink kernel module is currently running in the kernel. This is based on the information available in <code>/proc/modules</code> .
Daemon: running	Shows whether the daemon nslinkd is currently running. This is based on the file <code>/var/run/nslinkd.pid</code> and the existence of a matching PID in <code>/proc</code> .
Driver startup: enabled	Shows whether the driver and daemon are configured to be loaded/run at system boot. This is determined by querying the init system in use.

Command Button graphics will be shown individually

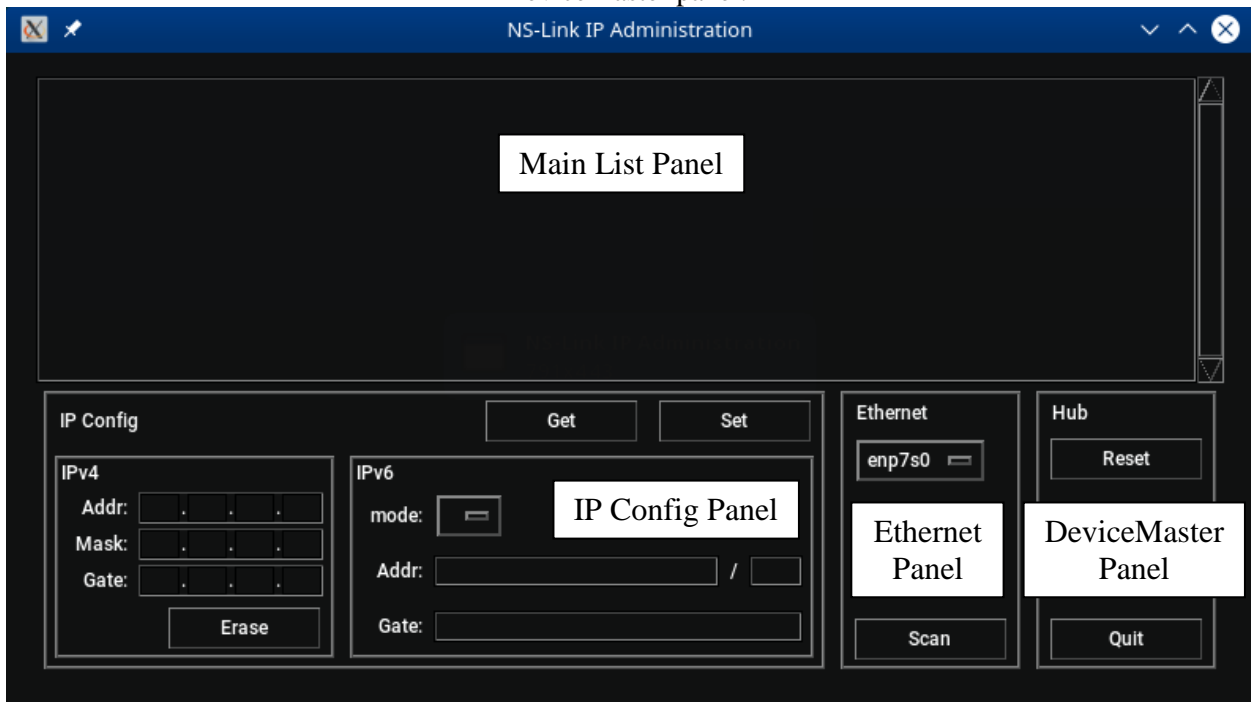
## NS-Link IP Administration



### Hub IP Admin

The HUB IP administration window allows the examination and configuration of static IP addressing information contained in a DeviceMaster's non-volatile memory.

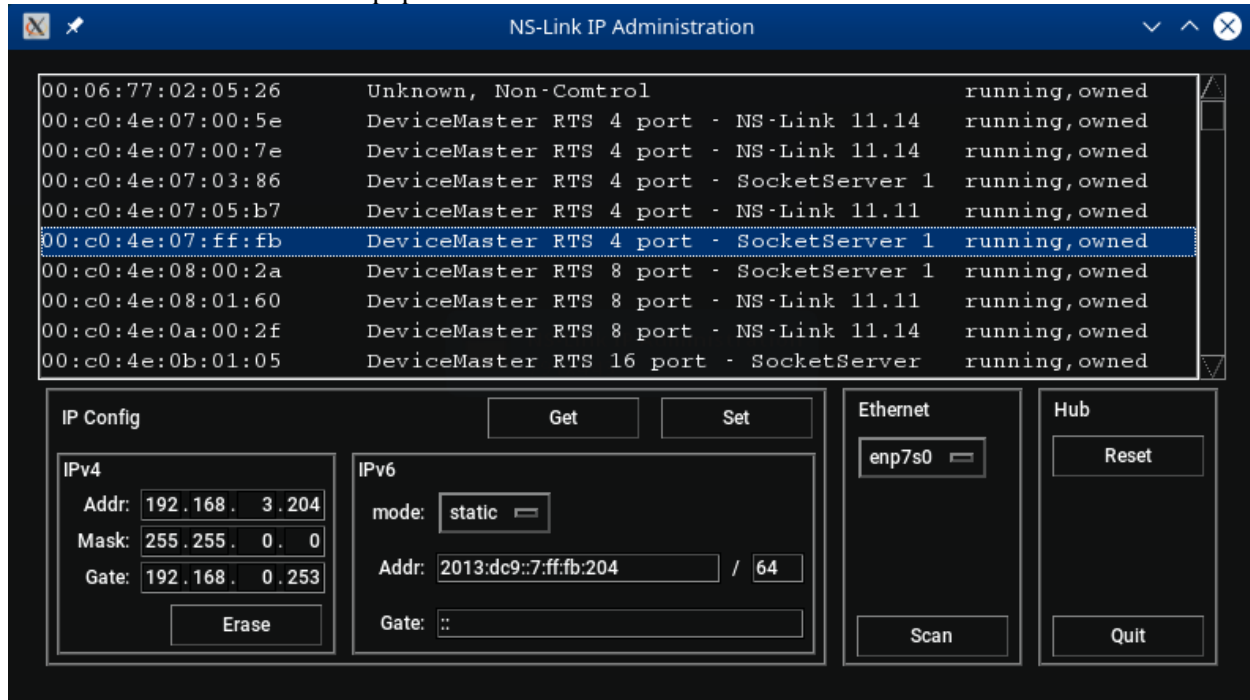
The four main panels in the editor are the main list panel, the IP Config panel, the Ethernet panel and the DeviceMaster panel.



## Main List Panel

This window displays a list of DeviceMaster's found on the network and allows the user to select which one is to be accessed.

Click on the 'SCAN' button to populate list as seen.



## IP Config Panel

This window contains the controls used to examine and change the IP addressing information in the DeviceMaster.

- **Get**
  - Reads the IP configuration from the selected DeviceMaster. This is the static IP configuration information in non-volatile memory. Blank values indicate that the DeviceMaster will attempt to use DHCP to obtain an IP address, netmask, and gateway address. The information obtained by the DeviceMaster using DHCP is not shown.
- **Set**
  - Writes the currently shown IP configuration to nonvolatile memory in the selected DeviceMaster.
- **Erase**
  - Erase the IP configuration in the selected DeviceMaster. This results in the DeviceMaster attempting to use DHCP to obtain an IP address the next time it is reset.
- **IP Address, Net Mask, Gateway**
  - These values should be entered in decimal, dotted-quad notation (e.g. 192.168.1.15 or 255.255.255.0). Make the appropriate changes, and click on the "Set" button to apply the changes to the Device Master.

- **Mode**
  - **Static**
    - The DeviceMaster uses a static IP Address
  - **DHCP**
    - The DeviceMaster will get its IP Address from a DHCP server
  - **Disabled**
    - No IP Address is assigned to the DeviceMaster

## **Ethernet Panel**

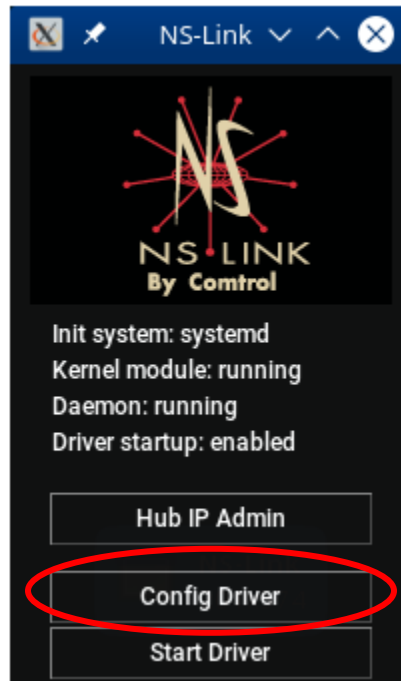
- **Scan**
  - This button queries the Ethernet network to construct a list of the DeviceMaster's present on the network. This is a drop-down list of the network interfaces found on the host where nslinktool is running.

## **Hub (DeviceMaster Panel)**

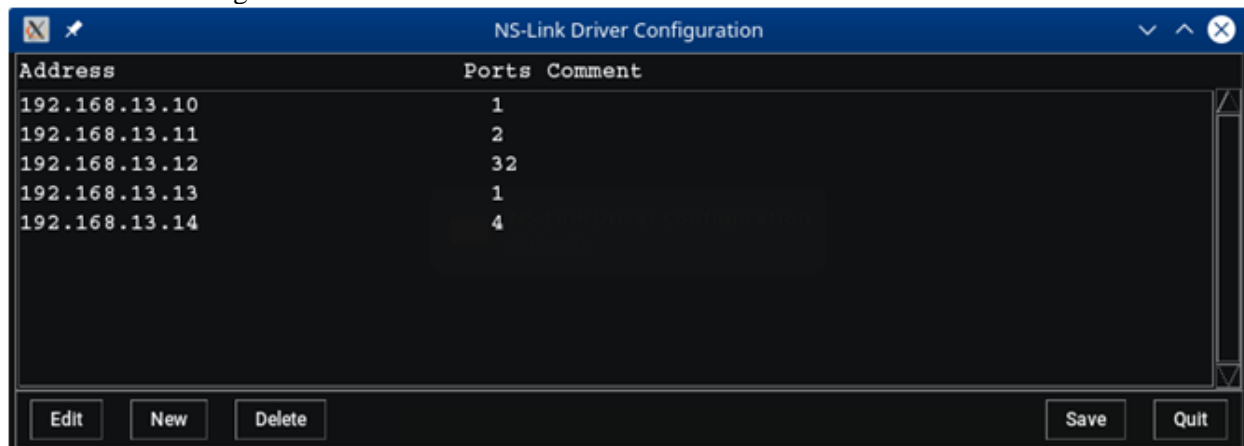
This window contains buttons to perform general functions

- **Reset**
  - This sends a reset request to the DeviceMaster. A DeviceMaster must be reset before any IP configuration changes will take effect.
- **Quit**
  - Closes the IP administration window.

## Config Driver



The Config Driver button will open the NS-Link Driver Configuration window which edits the information in the `nslink.conf(5)` file. The main list window shows a one-line summary for each DeviceMaster configuration found.



- **Edit**
  - Opens a hub-editor window that allows the user to change the configuration of the DeviceMaster selected in the main list window.
- **New**
  - Creates a new DeviceMaster configuration and opens the Edit Hub Configuration window. To add a new Device Master, click on the "New" button and another GUI will be displayed as seen in the Edit Hub Configuration section.
- **Delete, Save and Quit** require no explanation

## Edit Hub Configuration

Edit Hub Configuration
⌵ ⌶ ⌵

Comment:

TCP/IP

Ethernet

Number of Ports:

Link timeout:  seconds (0 == Never)

Scan period:  milliseconds (0 == Use Default)

ReTransmit Timer:  milliseconds (0 == Use Default)

Port	Type	Timeout	LL	FD	TTR	RTR	Port	Type	Timeout	LL	FD	TTR	RTR
0:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	16:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	17:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	18:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	19:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	20:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	21:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	22:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	23:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	24:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	25:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	26:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	27:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
12:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	28:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	29:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	30:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	31:	232	<input style="width: 50px;" type="text" value="0"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

### Main options:

- **Comment** Place any comments here such as the description or location of the DeviceMaster

- **TCP/IP** Enter the IP Address of the DeviceMaster. You can also enter a hostname instead of an IPv4 or IPv6 address. [Assuming you have DNS or hosts file set up to resolve that hostname to the hub's address.]
- **Ethernet** Enter the MAC address of the DeviceMaster
- **Number of Ports** Select the number of serial ports on this DeviceMaster
- **Link Timeout**
  - This is a Keep Alive Timer (see appendix A for full details) that controls how long the link between the NS-Link driver and the DeviceMaster can go down before the DeviceMaster is considered "gone" and the ports will report errors.
  - The value for timeout is specified in seconds. A connection check command is sent to the DeviceMaster every timeout/2 seconds. If more than timeout seconds passes between receiving connection check responses, the link will timeout and any open ports on that DeviceMaster reports errors. A value of 0 disables link timeout. If no value is specified, 120 seconds is used as the default.
- **Scan Period**
  - Typically, you should leave the Scan period set to the default value (0 = 10 ms) for most applications. To adjust latency for time-critical applications, enter a longer or shorter interval in the range of 2 to 50. If a value larger than 50 is entered, the maximum of 50 is implemented.
- **Retransmit Timer**
  - This controls how long the MAC mode protocol handlers will wait for a packet to be acknowledged before they retransmit it. It's called "retransmit timer" in the nslink.conf man page. The retransmit-timer value controls the length of time the DeviceMaster waits before retransmitting packets. It is specified in milliseconds and defaults to 200. Changing the retransmit timer, will increase CPU usage and network traffic.

## Port options

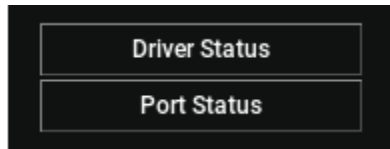
- **Type**
  - RS232
  - RS422
  - RS485 2-Wire
  - RS485 4-Wire Slave
  - RS485 4-Wire Master
- **Timeout (Intercharacter Timeout)**
  - Input Timeout: If the Serial Rx Timeout feature is enabled (Input Timeout), the DeviceMaster buffers receive serial data until a gap in the serial receive data stream has been detected. This gap can be configured to values from 1 ms to 4,294,967,295 ms (approximately 49 days). The default is 0ms.
- **LL**
  - Enables the low-latency flag (low\_latency)
  - Enable low-latency operation for received data. Rather than buffering receive data and transferring it to the line-discipline layer once every 10ms, receive data will be transferred immediately upon arrival.



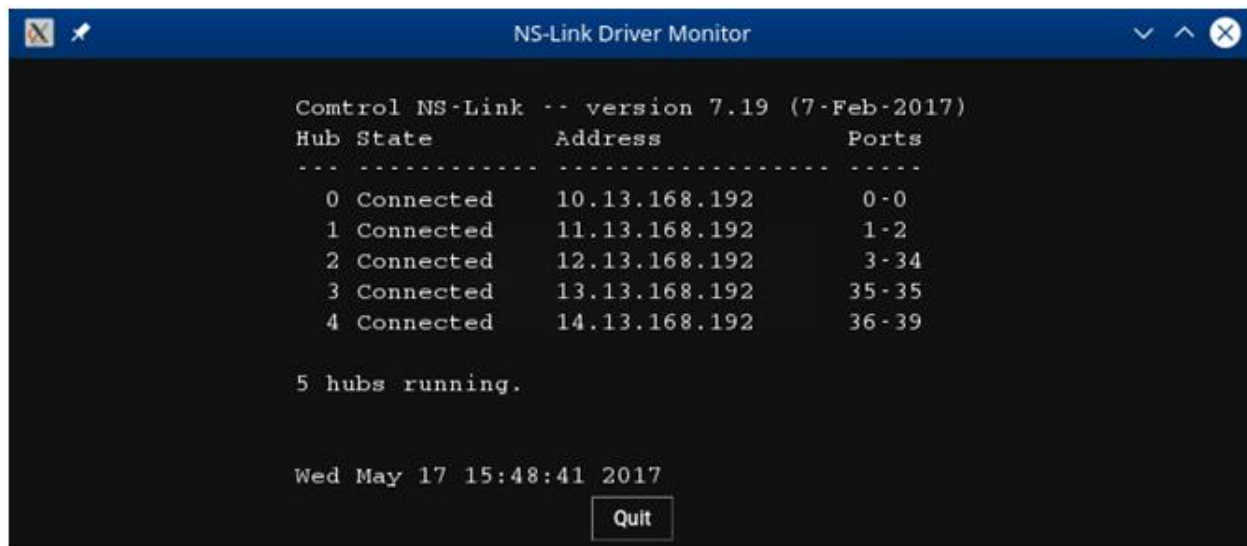
- **FD**
  - Disable the rx FIFO. (`rx_fifo_disable`)
  - Disable the use of the UART's receive FIFO. This is currently only implemented for the 1-port models. When the 1-port models have the receive FIFO enabled, data bytes received with invalid parity bits may be discarded. Disabling the receive FIFO on the 1-port models will cause bytes received with invalid parity to be passed to the line discipline layer. Not recommended at baud rates above 19200.
  
- **TTR**
  - Enables the tx terminating resistor (`term_res_tx_en`)
  - Enable the termination resistor between the RS422 TX+ and TX- signals (in RS422 mode) or across the 2-wire RS485 signals (when in RS485 mode). This option is only available for certain models (at time of writing, only the LT16).
  
- **RTR**
  - Enables the rx terminating resistor
  - Enable the termination resistor between the RS422 RX+ and RX- signals (when in RS422 mode). This option is only available for certain models (at time of writing, only the LT16).
  
- **OK**
  - Save and Close
  
- **Cancel**
  - Close without saving changes

Verify the entries are in place by doing a `'cat /etc/nslink.conf'` and examine the output.

## Driver and Port Status Options



**Driver Status** will display the NS-Link Driver Monitor



```
Control NS-Link -- version 7.19 (7-Feb-2017)
Hub State      Address          Ports
-----
 0 Connected    10.13.168.192   0-0
 1 Connected    11.13.168.192   1-2
 2 Connected    12.13.168.192   3-34
 3 Connected    13.13.168.192   35-35
 4 Connected    14.13.168.192   36-39

5 hubs running.

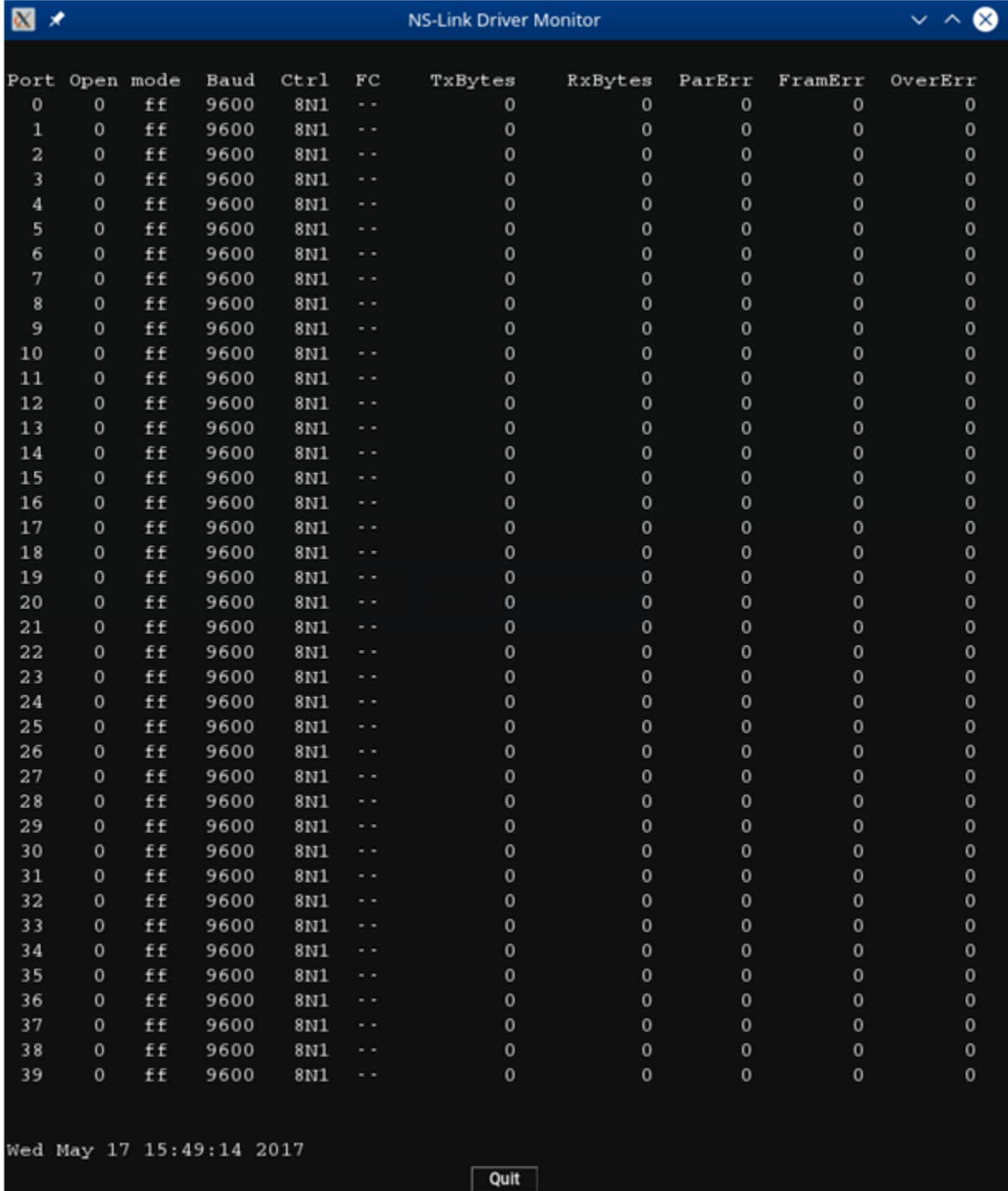
Wed May 17 15:48:41 2017
Quit
```

This shows the details of the DeviceMaster itself.

Driver Monitor Status

- **Hub #**
- **State**
  - Connected
  - Off-line
- **Address**
  - IP Address assigned to the DeviceMaster
- **Ports**
  - Number of ports on the DeviceMaster ttySIx
    - 0-0 = 1 serial port ttySI0
    - 1-2 = 2 serial ports ttySI1 and ttySI2
    - 3-34 is a 32 port DeviceMaster ttySI3 thru ttySI34

Port Status will open the NS-Link Driver Monitor that displays the status of the serial ports



The screenshot shows a window titled "NS-Link Driver Monitor" with a dark blue header and a black background. The window contains a table with 11 columns: Port, Open, mode, Baud, Ctrl, FC, TxBytes, RxBytes, ParErr, FramErr, and OverErr. The table lists 40 rows, one for each port from 0 to 39. All values in the table are consistent: Open is 0, mode is ff, Baud is 9600, Ctrl is 8N1, FC is --, TxBytes is 0, RxBytes is 0, ParErr is 0, FramErr is 0, and OverErr is 0. At the bottom left, the date and time "Wed May 17 15:49:14 2017" are displayed. At the bottom center, there is a "Quit" button.

Port	Open	mode	Baud	Ctrl	FC	TxBytes	RxBytes	ParErr	FramErr	OverErr
0	0	ff	9600	8N1	--	0	0	0	0	0
1	0	ff	9600	8N1	--	0	0	0	0	0
2	0	ff	9600	8N1	--	0	0	0	0	0
3	0	ff	9600	8N1	--	0	0	0	0	0
4	0	ff	9600	8N1	--	0	0	0	0	0
5	0	ff	9600	8N1	--	0	0	0	0	0
6	0	ff	9600	8N1	--	0	0	0	0	0
7	0	ff	9600	8N1	--	0	0	0	0	0
8	0	ff	9600	8N1	--	0	0	0	0	0
9	0	ff	9600	8N1	--	0	0	0	0	0
10	0	ff	9600	8N1	--	0	0	0	0	0
11	0	ff	9600	8N1	--	0	0	0	0	0
12	0	ff	9600	8N1	--	0	0	0	0	0
13	0	ff	9600	8N1	--	0	0	0	0	0
14	0	ff	9600	8N1	--	0	0	0	0	0
15	0	ff	9600	8N1	--	0	0	0	0	0
16	0	ff	9600	8N1	--	0	0	0	0	0
17	0	ff	9600	8N1	--	0	0	0	0	0
18	0	ff	9600	8N1	--	0	0	0	0	0
19	0	ff	9600	8N1	--	0	0	0	0	0
20	0	ff	9600	8N1	--	0	0	0	0	0
21	0	ff	9600	8N1	--	0	0	0	0	0
22	0	ff	9600	8N1	--	0	0	0	0	0
23	0	ff	9600	8N1	--	0	0	0	0	0
24	0	ff	9600	8N1	--	0	0	0	0	0
25	0	ff	9600	8N1	--	0	0	0	0	0
26	0	ff	9600	8N1	--	0	0	0	0	0
27	0	ff	9600	8N1	--	0	0	0	0	0
28	0	ff	9600	8N1	--	0	0	0	0	0
29	0	ff	9600	8N1	--	0	0	0	0	0
30	0	ff	9600	8N1	--	0	0	0	0	0
31	0	ff	9600	8N1	--	0	0	0	0	0
32	0	ff	9600	8N1	--	0	0	0	0	0
33	0	ff	9600	8N1	--	0	0	0	0	0
34	0	ff	9600	8N1	--	0	0	0	0	0
35	0	ff	9600	8N1	--	0	0	0	0	0
36	0	ff	9600	8N1	--	0	0	0	0	0
37	0	ff	9600	8N1	--	0	0	0	0	0
38	0	ff	9600	8N1	--	0	0	0	0	0
39	0	ff	9600	8N1	--	0	0	0	0	0

Open a window to monitor the port status present in `/proc/driver/nslink/ports`. This window does the equivalent of the shell command: `while sleep 1; do cat /proc/driver/nslink/ports; date; done`

- **Port**
  - /dev/ttySI#
- **Open**
  - 0 = Open and in use
  - 1 = Available, not currently in use
- **Mode (Interface Mode)**
  - Value      Definition
  - 32          RS232
  - 22          RS422
  - 85          RS485
  - 87          RS485 4-wire master
  - 86          RS465 4-wire slave
  - de          Off
  - ff          Unconfigured (may be off or RS232 depending on model)
- **Baud**
  - Baud rate currently configured
- **Ctrl**
  - Serial port settings (8N1)
    - 8 = Word size (data bits)
    - N= Parity
    - 1 = Stop Bits
- **FC Flow Control**
  - When the port is in use, this displays SW for software or HW for hardware
- **TxBytes**
  - Number of bytes Transmitted
- **RxBytes**
  - Number of bytes Received
- **ParErr**
  - Number of Parity Errors
  - A parity bit, or check bit, is a bit added to a string of binary code to ensure that the total number of 1-bits in the string is even or odd. Parity bits are used as the simplest form of error detecting code.
- **FramErr**
  - Number of Framing Errors
  - Generally, a framing error is the result of starting to read a sequence of data at the wrong point. In serial communications, a framing error is the result of reading a string of symbols which are grouped in blocks at the wrong starting point. The symbols are bits and the blocks are bytes, ten bits in asynchronous transmission and eight in synchronous. A framing error in an asynchronous stream usually recovers quickly, but a framing error in a synchronous stream produces gibberish at the end of the packet. Framing errors can be detected with parity bits.

- **OverErr**
  - Number of Overrun Errors
  - Typically occurs in devices and applications when they receive more data than they anticipate, usually because the allocated or physical memory buffer is not big enough.



#### **Config Driver**

Opens the driver configuration editor window which allows the user to edit the contents of the nmlink.conf file.

#### **Start Driver**

Loads the driver kernel module and starts the daemon using the nmlink script installed

#### **Stop Driver**

Stops the daemon and unloads the driver kernel module using the nmlink script installed

#### **Restart Driver**

Restarts the driver/daemon using the nmlink script installed. The driver must be restarted in order for changes to the nmlink.conf file to take effect.

#### **Enable Driver**

Configures the system so that the driver is started when the system enters run states 2, 3, or 5.

#### **Disable Driver**

Configures the system so that the driver is not started automatically.

## NOTES

Please be aware, you are **not** warned if you quit with unsaved changes. You are not asked for confirmation when you delete something. Closing the initial driver administration window (the one with the logo at the top) will close all of the other windows also.

Please be aware that with the high number of variations of windowing systems in the Linux Operating System some displays may behave differently than others and may not present all of the optional selections correctly.

If check marks are not displayed. Please try a different desktop theme.

## Additional resources

### Man Pages

nslinkd(8), nslink(8), nslinkadmin(8), nslinkrelease(8), nslink.conf(5)

### Release Notes

Check the release notes file for an up-to-date list.

### Web site

[www.comtrol.com](http://www.comtrol.com)

### FTP downloads

<ftp://ftp.comtrol.com/>

## Appendix

### A: Link Timeout (Keep Alive)

Use the Keep Alive Timeout option to set the amount of time in seconds that this DeviceMaster waits until it closes this connection and frees all the ports associated with it. The Keep Alive feature works in the following way.

Values entered may be from 0 to 999999. Values less than 5 will cause the driver to FAIL to maintain communications. Values greater than 120 will require greater time periods for recovery and are not generally recommended. A setting of 10 is generally more affective in re-establishing communications the quickest.

During normal operation the driver periodically sends a connection check to the DeviceMaster, and the DeviceMaster then returns a response. There are two timers: one in the driver, and one in the DeviceMaster. These timers are reset when a connection check signal is received. If a connection is broken, that is, a check is not received, the data is stored in the computer and/or the DeviceMaster. Depending on the amount of time that has expired since the connection was lost, the following happens:

- When the computer loses its connection to the DeviceMaster but re-establishes it before the time-out period expires, any data transmitted during this period is queued and sent when the connection resumes.
- When the computer loses its connection to the DeviceMaster and does not re-establish it before the time-out period expires; the driver then purges any pending I/O data for ports on that connection and returns all pending, and future I/O commands, with the exception of the Close command, to the application with an error indicating the disconnected status.

This is similar to the processing which occurs when the computer receives a notification from the DeviceMaster that a port release request was processed on a port it owns with the exception that a different status is returned. When the connection is re-established, the computer will attempt to re-acquire the ports that were open when the connection was lost. If the attempt is successful, normal operations resume for the port. If any port fails to be acquired, then the computer will continue to fail all further I/O operations, with the exception of a Close request. When the Close request is received, the port can then be re-opened.

- If the computer loses its connection and the time period expires, the DeviceMaster places the port into a state that another computer can establish a connection, locking out the original driver when a connection is made. The driver will respond to all I/O commands, with the exception of the Close command, with an error indicating the disconnected status. If the port is still available when the driver re-establishes a connection, then it will claim the port and allow I/O to resume.