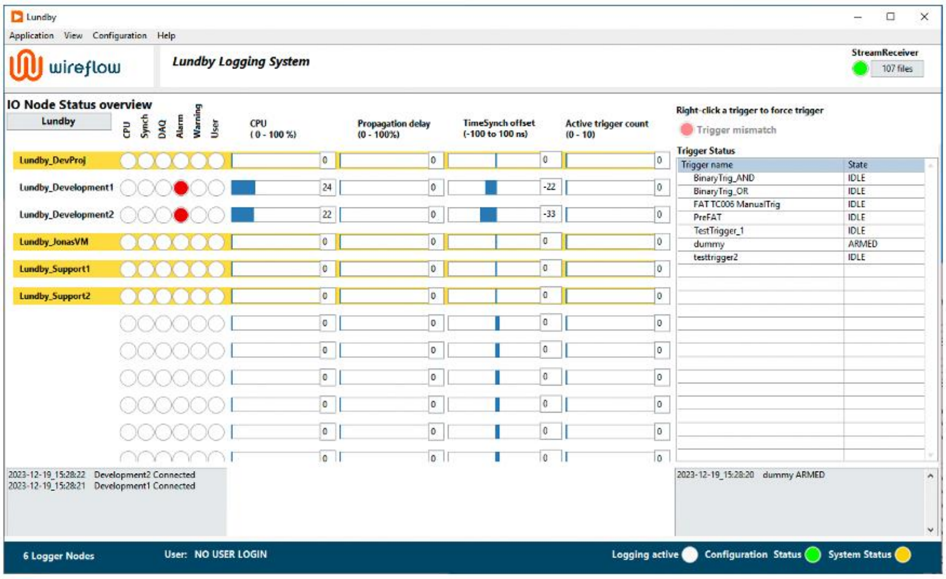
AC0255-001, rev B

# Lundby User Manual



# Contents

# 1 Support information

## 1.1 Technical support and Product information

www.wireflow.se

## 1.2 WireFlow headquarters

WireFlow AB
Krokslätts Fabriker 18
SE-431 37 Mölndal

# 2 Important information

## 2.1 Warranty

## 2.2 Copyright

The Lundby system is Copyright 2023, WireFlow AB (SE556851155301)

## 2.3 Trademarks

## 2.4 Patents

## 2.5 Licenses

The Lundby system uses a modified version of the JSONtext LabVIEW add-on released under BSD license

### 2.5.1 JSONtext


Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

### 2.5.2 JDP Science Common Utilities

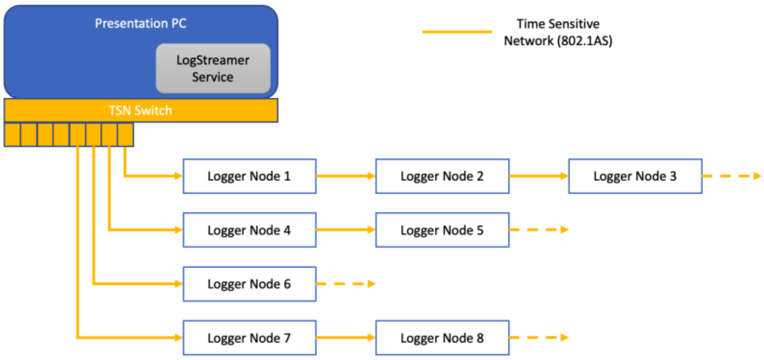


Redistribution and use in source and binary forms, with or without

## 2.6 Warning regarding use of WireFlow products

# 3 Lundby Logger System components

The Lundby Logger system is a distributed data acquisition system, with high time synchronization and the possibility to log massive amount of data. The system consists of the following components




- Lundby System
  - When talking about the **Lundby System** it means the complete setup, including GUI, Streamer Service and one or more Logger nodes connected through TSN switches.
- Lundby Logger Node
  - A Lundby Logger Node is a network node that can be configured from the GUI to perform trigger checks as well as logging data to disk. The **Lundby Logger Node** is both the DAQ hardware and the Real Time software that runs on the node controller
- Lundby GUI
  - The **Lundby GUI** is the glue that brings all Lundby Logger Nodes together into one system, by configuring the system in terms of network, name, channels logging etc. It is the monitor application as well as the configuration application for the Logging system
- Lundby StreamerService
  - The **Lundby_StreamerService** is a background service running 24/7 on a Windows PC, continuously monitoring and fetching ready data from all Logger Nodes. The reason is that the data logged from distributed nodes only makes sense once all data has been consolidated.

## 3.1 Terminology

| | |
|---|---|
| DAQ | Data Acquisition |
| DAQ block | A chunk of data read from the DAQ process and handled in the Lundby Server |
| TDMS | The logfile format used in Lundby (TDM for Streaming applications) |
| IEEE 802.1AS | The time synchronization standard used in Lundby |
| GUI | Graphical User Interface |

# 4 Quick Start

To use the Lundby System the system must be configured in the following steps

- General System Configuration
  Setup general system configuration in the Lundby GUI. Includes network configuration as well as configuring the name of the System (see 9.3)
- Adopting Nodes (see 9.4.1)
  This step adds Lundby Logger Nodes to the current system by setting the system name and the ownership of the node.
- Configure Channels (see 9.4)
  Once the nodes have been added it is time to configure the hardware channels
- Create Calculated Channels (see 9.6)
  If necessary, calculated channels can be created to get more detailed channel characteristics in real-time, e.g. statistics, waveform attributes etc.
- Configure Logging (See 9.8)
  Finally it is time to create logging conditions, including continuous logging.

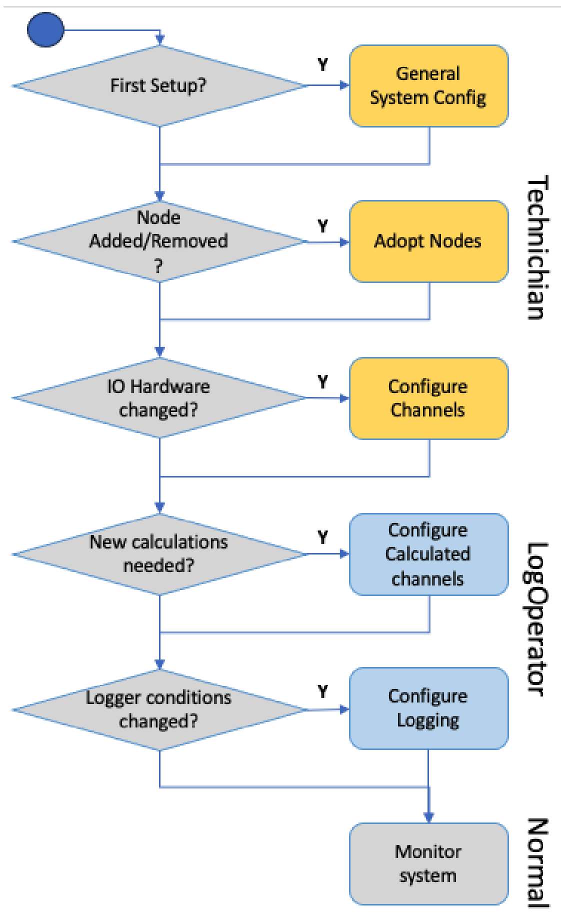



*Figure 1. Configuration flow*

# 5 Hardware

The Lundby system is based on network distributed Logger Nodes executing on real-time hardware and a PC running the monitoring and data collection tasks.

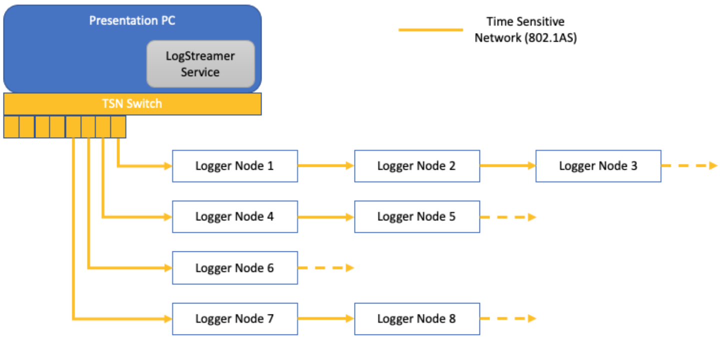



*Figure 2. Hardware overview*

Because of the time-synchronization requirements all LoggerNodes must be interconnected via network switches that can handle TSN (IEEE 802.1AS) traffic.

The system then relies on the Best Master Clock Algorithm to select a timing master in the system – This means the selection of timing master relies on configured priorities as well as the quality of the clock devices.

Normally an external clock device, e.g. a GPS clock, is connected to the top-level TSN-Switch and supplies the system with a high-quality clock source.

Each Logger Node can use different combinations of IO modules and are adopted and configured using the Lundby GUI.

# 6 Communication

The Lundby system relies on several network communication "links" to perform all tasks.

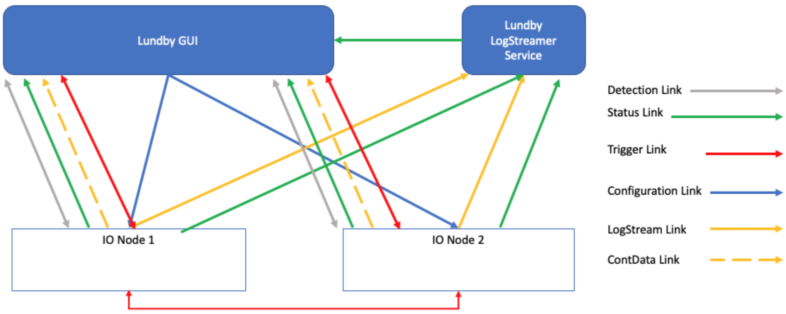



*Figure 3. Lundby communication links*

In Figure 3 these links are outlined, and the usage is:

- Detection Link
    - Used by the GUI to detect new Nodes to be adopted, using UDP Multicast
- Status Link
    - Each Node creates a status publisher that publisher status to any subscriber once per second. This link is using ZMTP over TCP/IP.
    - The status is used by both GUI and StreamerService to detect if a node is alive
    - The status link is also used in a special mode to let the StreamerService signal its status to the GUI
- TriggerLink
    - Used by the nodes to signal to other Nodes that a locally detected trigger condition is active. This link uses UDP multicast and the trigger data is resent every second as long as the trigger is active.
- Configuration Link
    - The GUI connects to each Node using TCP/IP to manage configuration and to get more detailed status.
- LogStream Link
    - Once the Node status indicates that a node is connected, the LogStreamer Service uses this link to fetch new logfiles as indicated by the status message (using TCP/IP).
- Continuous Data Link
    - In the GUI it is possible to view raw channel values wither in the channel overview, in a history graph or as the raw waveform graph. This link is used in the background to fetch these values using ZMTP over TCP/IP.

NOTE. For best performance Wireflow recommends that the Lundby system is separated from other networks.

## 6.1 Network configuration

Since the IP configuration is normally not changed, the configuration is “hidden” in the config file “ConfiguredStations.ini” (see 11), and is not available in the GUI.

In case the communication is limited by firewalls it might be necessary to configure the firewall to allow the following ports and IP addresses used in the Lundby system (this is the default values)

Note that the setting key AdapterIP specifies the local network adapter by its IPv4 address, and if not specified the auto detection of nodes is limited to the primary network interface of the computer running the Lundby GUI.

### 6.1.1 TCP ports

The TCP communications are used for management, status monitoring and LogStream

- Status.Port = 53010
- Management.Port = 53020
- LogStream.Port = 53030
- ContStream.Port = 53050

### 6.1.2 UDP ports

UDP ports are usd only on the Multicast addresses to “broadcast” information, one-to-many, e.g. The Detection.port is used to detect new Nodes, and the Trigger.Port is used when a Node detects a new trigger.

- Detection.Port = 52000
- Trigger.Port = 53040

### 6.1.3 Multicast addresses

The Multicast addresses are used together with the UDP port to create on-to-many signaling links

- Trigger.MulticastAddr = "224.0.0.255"
- Detection.MulticastAddr = "224.0.1.255"

# 7 Software overview

The Lundby system consists of three main components

- Lundby GUI
  - The main monitor and configuration application
- Lundby StreamerService
  - Background service to continuously get data from the Logger Nodes
- Lundby Server
  - Real time software executing on the Lundby Logger Nodes – controlled and managed by the Lundby GUI and the Lundby StreamerService

This document describes the usage of Lundby GUI and partly the Lundby LogStreamer service.

These applications should be installed on a Windows PC, and once installed the Lundby GUI can be launched from the Start menu (or search for LundbyGUI) and the Lundby LogStreamer starts as a windows service when the PC starts.

Since the Lundby Streamer Service runs in the background as a windows service it has no specific GUI but is configured and monitored from the Lundby GUI application.

**NOTE.** The Lundby Server has no user interface and is treated/configured from the Lundby GUI as a hardware component in the system.

**NOTE.** The PC running the GUI is regarded as the master in the system, and the MAC address of the PC is used to check if Nodes belongs to the system or not.

**NOTE.** If the Lundby StreamService fails to be installed - Make sure the C:\Windows\Windows32 path is specified in the environmental variables of the system, in order for the Lundby Streamer Service to be installed:

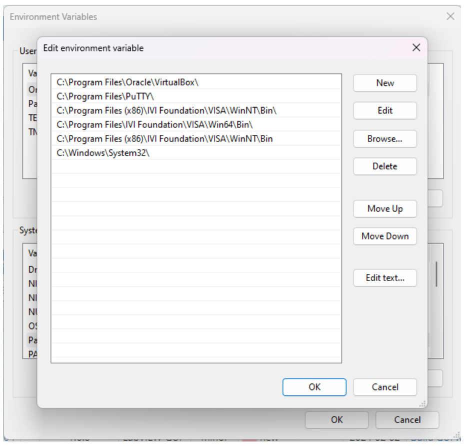



*Figure 4. StreamerService environment variable fix.*

## 7.1 Configuration files

All related configurations are stored in the folder

<ProgramData>\WireFlow\LundbySystem

The configuration is normally managed from the user interface, but for completeness the different files are outlined in chapter 11

**NOTE. It is the end user's responsibility to backup this folder!**

# 8 Lundby GUI

The main functionality is performed by the Logger Nodes, while the purpose of the Logger GUI is to monitor and configure the nodes to perform as expected.

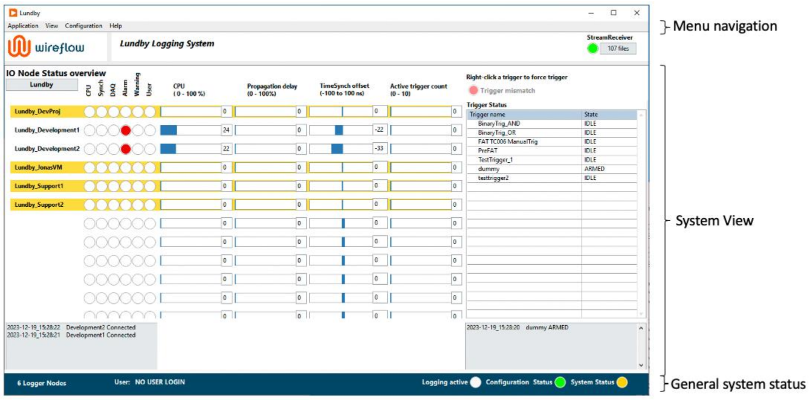



*Figure 5. Lundby GUI Overview*

The GUI displays the status of the application in a number of selectable view.

Views are selected in the “View” menu and are displayed in the “System View” parts of the panel (see Figure 5).

Available views are:

1. System Status
2. Channel Monitor (see 8.2)

The menu system also gives access to tasks like system configuration, IO Node backup handling, accessing help documentations etc.

The General system status displays

- Total number of known LoggerNodes (connected + disconnected)
- Current logged in username
- Logging status: white if no logging is active, green if Logging is active
- Configuration status: Green if all OK, Red if any node has a configuration mismatch, i.e. running config different from disk
- System Status: Red (No nodes connected), Yellow (some nodes connected, Green (All nodes connected)

## 8.1 Menu navigation

From the main menu much of the functionality is accessed from the menu selections. The available Lundby GUI menu selections are listed below

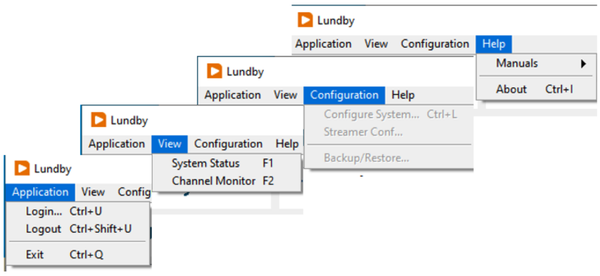



*Figure 6. Lundby GUI menus*

- Application - Login...
    - Open a Login dialog to allow a user to access features requiring elevated rights. If wrong username/password combination is entered the OK button is disabled and the Unknown user/pwd LED turns red.


*Figure 7. Login dialog*

    - **NOTE.** If the Login is cancelled the current user is logged out
- Application – Logout
    - Logout current user
- Application – Exit
    - Closes the Lundby GUI application
- View – System Status
    - Switches to System status view- displays the status of the system from a Logger Node perspective. Also displays the configured triggers and corresponding status. See 8.2.
- View – Channel Monitor
    - Displays all configured channels in the system and continuously updates values. See 8.5
- Configuration – Configure system...
    - Opens a configuration dialog for system node configuration (see 9)
- Configuration – Streamer conf...
    - Opens a configuration dialog for Streamer service specific configuration. See 9.1
- Configuration - Backup/Restore

  - Opens a dialog allowing backup and restore of configurations for Logger Nodes. See 8.2
- Help - Manuals
  - Lists the available manuals, the first manual is accessible with the shortcut [Ctrl-H]
- Help – About
  - Displays the current application version and dependencies used to build the application. See 8.3

## 8.2 Node backup/restore

Opening the Backup/Restore menu selection brings up a dialog to backup LoggerNodes to the local PC disk, and allows for quick restore of a configuration, e.g. in case a Node is replaced.

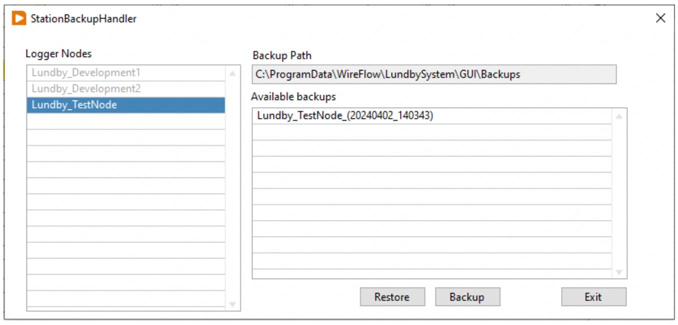



*Figure 8. Backup/Restore dialog*

Pressing “Restore” brings up a file dialog to select the backup folder to be used, pressing Backup saves the Node configuration backup to “C:\ProgramData\WireFlow\LundbySystem\GUI\Backups\XYZ”, where XYZ is the node name.

NOTE. Backup means copying data from the Logger Node to the local PC disk. The backup of the local folder (“C:\ProgramData\WireFlow\LundbySystem\GUI”) is the responsibility of the user of the Lundby system.

## 8.3 About dialog

The about dialog gives information about the version of the components in the system. Pressing "More..." or "...Less" to show more or less information.

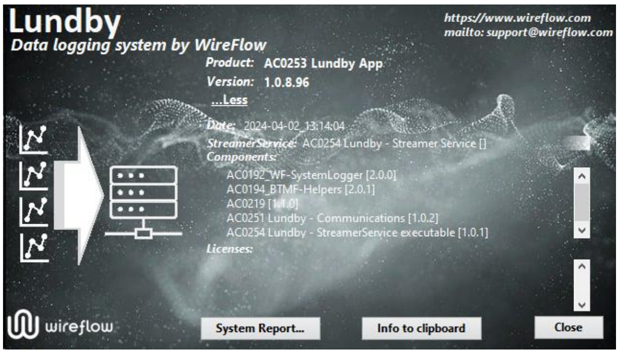



*Figure 9. Lundby GUI About dialog*

- Pressing System Report collects information of all connected LoggerNodes in addition to the version info of the PC application and saves this to a report file.
- Pressing Info to clipboard copies the PC information only to the clipboard.

Both these options are very valuable when requesting support.

## 8.4 System Status

The System Status View displays the status of Logger nodes as well as the status of the configured triggers.

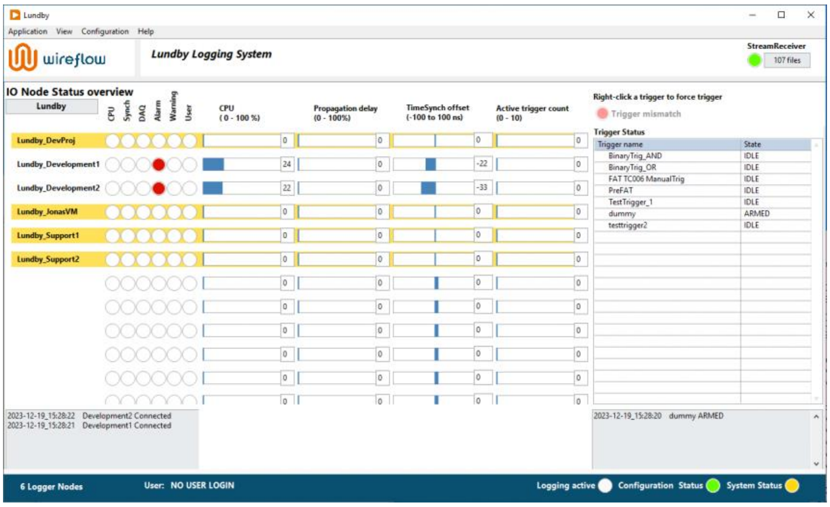



*Figure 10. Lundby System status overview*

Each node is listed and displaying any active alarms as well as some status parameter values. A disconnected node is listed with a yellow background.

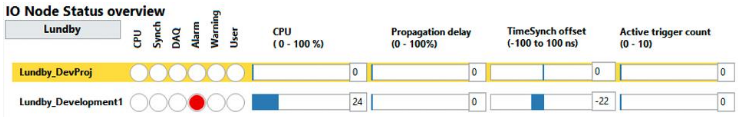


*Figure 11. Node quick status*

- CPU
  - CPU alarm LED
- Synch
  - Time synch Alarm LED
- DAQ
  - DAQ Alarm LED
- Alarm
  - Any active alarm LED
- Warning
  - Any active warning LED
- User
  - Any active User Alarm LED (from trigger configuration)
- CPU
  - bar showing CPU usage between 0 and 100%
- Propagation delay
  - bar showing the percentage ( 0-100 ) of sampling time used for calculations and distribution.
- TimeSynch Offset
  - Synchronization offset from current time master, -100 to 100ns
- Active Trigger count
  - The current number of running triggers on the Node

## 8.4.1 Node status details

Double clicking a Node brings up the Node status details, see Figure 12.

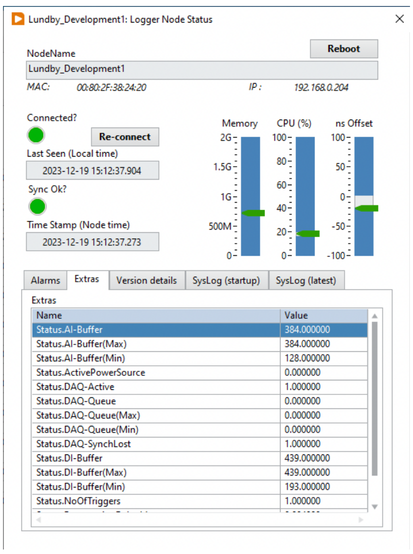


*Figure 12. Node status details*

This view gives more details about the running status of the Node, including IP and MAC address.

- Last Seen (Local time)
  - PC timestamp of the last received status message
- Time Stamp (Node time)
  - Timestamp in Node time in last status message
- Extras TAB
  - Displays internal status parameter values, useful for support
- Version details TAB
  - Displays the version details of the running software of the node
- SysLog (startup) TAB
  - Shows the startup log, in case there is a problem with the current configuration
- SysLog (latest) TAB
  - Shows the latest log generated by the system
- Alarms TAB
  - Displays more details about alarms and warnings, i.e name of alarm bits and a yellow LED for warning and a Red LED for alarm. Some alarms (e.g. CPU level) will light both Red and Yellow in case of an alarm (see 12.3)

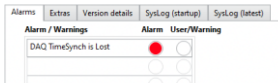



*Figure 13. Alarm details*

### 8.4.2 Node event logs

Lower left corner of the Status View displays Logger node events that happens while the Lundby GUI is running. A Node can be connected, disconnected and removed. A node is Removed if the Node reports another master MAC address than the running PC.

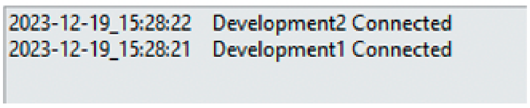



*Figure 14. Node status events*

### 8.4.3 Trigger events

The lower right corner displays the Trigger events detected while the Lundby GUI is running. The trigger events that are recorded are :

ARMED = trigger is waiting for trig condition to start running

RUNNING(x) = trigger is currently running on x Logger Nodes

| | |
|---|---|
| 2023-12-20_16:20:34 | testtrigger2 ARMED |
| 2023-12-20_16:20:24 | testtrigger2 RUNNING (1) |
| 2023-12-20_16:20:12 | testtrigger2 ARMED |
| 2023-12-20_16:20:02 | testtrigger2 RUNNING (1) |
| 2023-12-20_16:17:00 | BinaryTrig_AND RUNNING (1) |
| 2023-12-20_16:16:58 | BinaryTrig_AND ARMED |

*Figure 15. Trigger events*

### 8.4.4 Trigger status

The trigger status lists all currently configured triggers and the state. The state column can show one of the states

- IDLE
    - Trigger is configured but not applied to any of the connected nodes
- ARMED
    - Trigger is active and applied to one or more connected nodes
- RUNNING ( x )
    - Trigger is running on x nodes

In addition a running trigger displays a blinking bullet

| Trigger name | State |
|---|---|
| o BinaryTrig_AND | RUNNING (1) |
| BinaryTrig_OR | IDLE |
| FAT TC006 ManualTrig | IDLE |
| PreFAT | IDLE |
| TestTrigger_1 | IDLE |
| testtrigger2 | ARMED |

*Figure 16. Trigger status list*

Right clicking the trigger list brings up a contextual menu to manage triggers

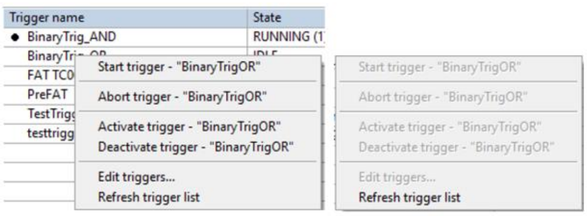



*Figure 17. Trigger contextual menu*

Most selections in this menu are disabled for a normal user, but enabled for LogOperator or higher, see Figure 17. The available options are

- Start trigger XXXX
    - Manually starts trigger XXXX on all configured Logger nodes
- Abort trigger XXXX
    - Aborts a running trigger XXXX on corresponding Logger Nodes
- Activate trigger XXXX
    - Quick access to activate trigger XXXX
- Deactivate trigger XXXX
    - Quick access to deactivate trigger XXXX
- Edit triggers...

    - Opens the trigger overview for trigger editing
    - For trigger information see 9.8 (information about trigger overview and editing are found in 9.8.1 - 9.8.4)
- Refresh trigger list
    - Refreshes the list of triggers.

## 8.5 Channel View

The Channel View is a quick view into the current channel values of the system. It can be used to make a sanity check of the input values as well as aid in channel calibration.

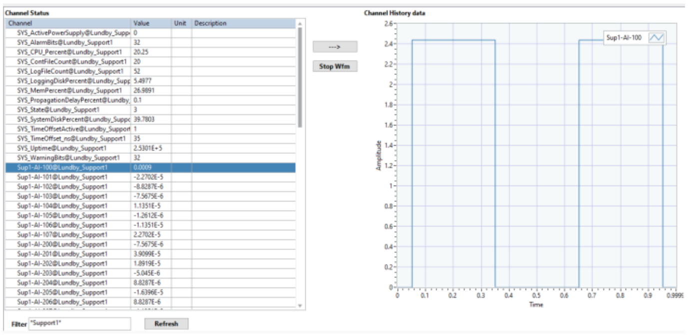



*Figure 18. Channel View*

### 8.5.1 Channel status list

The left part of the Channel View lists the channels in a table, with values updated at roughly 1Hz. Channel names are listed with Node name added as a suffix (@<NodeName>) if the name exists more than once.

Specify a Filter string to only show a subset of the channels, , e.g. Filter *Support1* to only show channels from Node “Support1”.

Double click a channel to open a channel history window (see 8.5.3).

### 8.5.2 Channel Waveform data

The right-hand side of the Channel View contains a graph where the waveform of a single channel can be viewed. Mark the channel and press ---> to start the waveform, the current waveform channel is visible in the graph.

This waveform contains the raw data in its native SampleRate, and is updated each BlockSize(s), default 1second of data, and introduces a lot more traffic on the network than just viewing the last updated channel values.

**NOTE.** When leaving the Channel View the waveform update will automatically be disabled to save network reseources.

### 8.5.3 Channel History window

A double-click on a channel brings up a floating graph window where the channel history can be monitored over several updates, covering at least one minute of data. Multiple simultaneous floating windows can be open, and these windows stay open even if the Main View changes to Node Overview or if any other windows is opened.

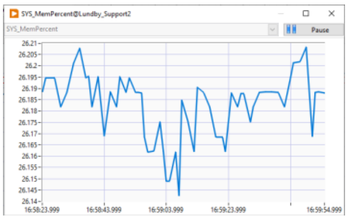



*Figure 19. Channel History window*

The x-axis on a Channel history window represents the actual timestamp of the updates. Pressing "Pause" freezes the graph updates but data will still be updated in the background.

NOTE. Multiple floating windows can be opened at the same time and each window keeps starts the historical data from when it was opened (and stops historical data when closed).

# 9 Configuration

All configuration except the streamer and trigger configuration is accessed from the Main Window menu, or by pressing “Ctrl-L”

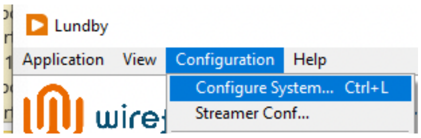
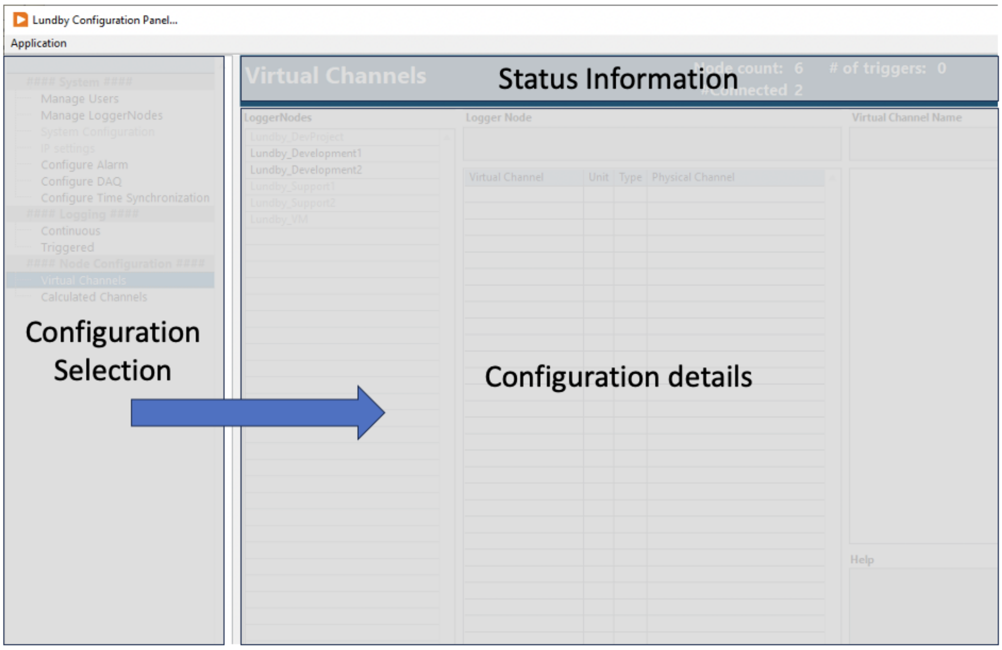



*Figure 20. Accessing the Lundby configuration panel*

The configuration panels adapts to the current user level, limiting certain actions to elevated user rights. The panel layout is basically split in three different areas.




- Configuration Selection
  The available configuration options
- Configuration Details
  Once an option in the Configuration Selection is selected, the corresponding details are shown in this area
- Status Information
  Displays the current selection as well as the Node count information

Legend

In all configuration panels where nodes are listed specific symbols are used to indicate the status of a node

## 9.1 Streamer Configuration

The streamer service uses the same configuration as the LundbyGUI, but also some streamer specific configuration, to specify the local logfolder (on the GUI PC).

To configure the service select “Steamer conf...” from the Configuration menu.

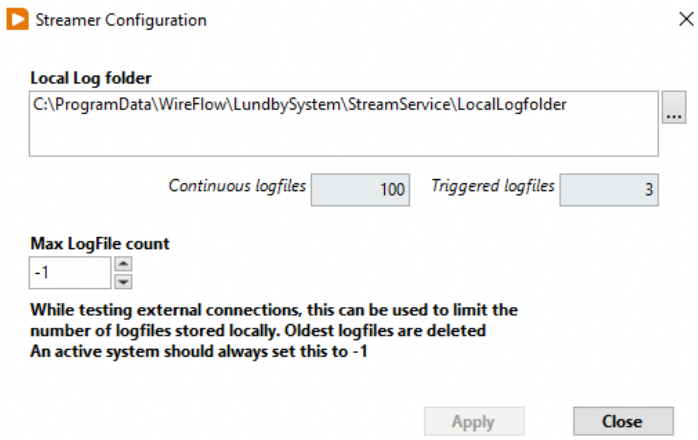



*Figure 21. Streamer specific configuration*

Specify the Local Log folder to specify the top folder for local storage.

The streamer continuously fetches new triggered logfiles and continuous logfiles from all connected Nodes. Depending on the type of logfile the files are placed in one of the subfolders

- triggered
- continuous

During testing of external access, it is possible to limit the number of logfiles on the local folder by specifying the Max Logfile Count

NOTE. During transfer of files from the Nodes, the files have the extension *.part, and is renamed to TDMS once completed. The *.part files should not be accessed by any external log file handler.

## 9.2 User configuration

The Lundby GUI uses the following User Levels to manage different levels of the Lundby System.

- Normal
  - Can only view data on the system, cannot configure or perform any changes
- LogOperator
  - A LogOperator is allowed to change the Logging configurations as well as perform backups
- Technichian

  - A Technichian can perform configurations on the DAQ channels, Calculated channels etc.
- SuperUser
  - Have full access, and is required to create new Technichians

The Manage User configuration lets an Technichian or SuperUser manage the users on the LundbyLogger. Once logged in - press refresh to update the list of current users in the system.

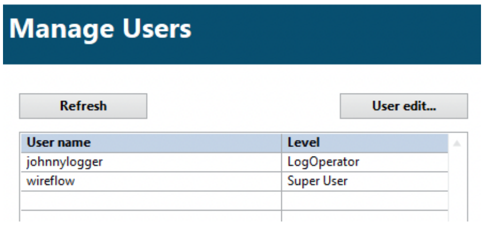



*Figure 22. User list*

Each user is listed with name (lower case) and the associated access level (only users with access level lower or equal to the current user is visible).

To add new users or edit an existing user, press “User edit…”. This brings up a new dialog where a current user can be edited or a new user added.

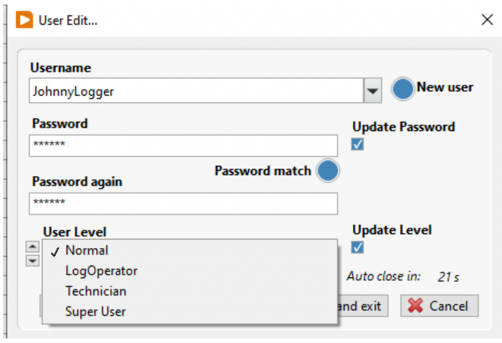



*Figure 23. User editing dialog.*

#### *9.2.1.1 Editing an existing user*

To edit an existing user select the user in the drop-down menu.

## 9.3 System Configuration

System configuration is the low-level configuration of the Nodes, this includes Logging output location, DAQ setup, TSN setup and Alarm configuration.

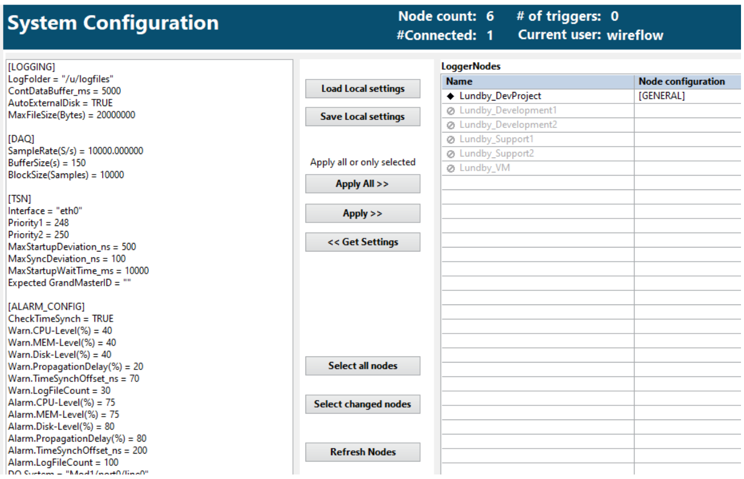



*Figure 24. System Configuration*

To change the system configuration the user must have "Super User" access level.

In most cases all Logger Nodes can have the same configuration, but it is not required by the system. There is always a system default configuration stored on the PC, and all Nodes with configurations that doesn't match this configuration is marked with a "diamond" symbol.

## 9.3.1 Buttons

The following buttons are used:

- Load Local Settings
  Loads the stored setting from the disk of the Lundby GUI
- Save Local Settings
  Saves current setting to the Lundby GUI disk
- Apply All >>
  Applies the configuration to all nodes
- Apply >>
  Applies the configuration to the selected nodes only
- << Get Settings
  Get setting from the specified node
- Select All nodes
  Select all nodes
- Select Changed nodes
  Select the nodes with not matching the current configuration
  NOTE. Some configuration requires reboot before the node reports the new setting
- Refresh nodes
  Refresh node listing

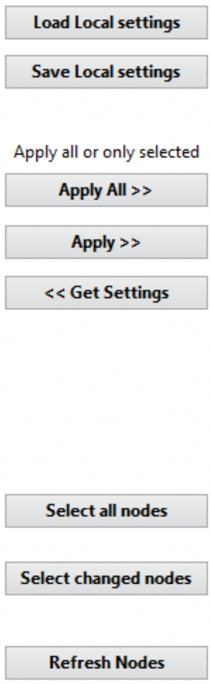



*Figure 25. Action Buttons*

## 9.3.2 LOGGING

The logging section configures the logging output on the Logger Node

- LogFolder = "/u/logfiles"
  - Specifies the output folder for the triggered logfiles, normally on an external disk /u/
  - Continuous logfiles are handled automatically
- ContDataBuffer_ms = 5000
  - Specifies the memory buffering of the continuous logging before written to file.
  - A too big values risks missing data points in case of a crash, and a too small value generates too much overhead in the logfiles.
- AutoExternalDisk = TRUE
  - Currently not used
- MaxFileSize(Bytes) = 20000000
  - The maximum triggered logfile size before automatic segmentation.
  - Default is 20MB, since this is a size that is reasonable for file transfer and it also means that data is not kept too long on the Logger Node

### 9.3.3 DAQ

DAQ section determines how the system collects data from the hardware interfaces.

- SampleRate(S/s) = 10000.000000
  - The sample rate used to collect data from the hardware (Samples per second)
  - NOTE. This should be the same on all Nodes
- BufferSize(s) = 150
  - This specifies the in memory buffer size (in seconds) used for pre-trigger- and software trigger operations. Maximum value is 150s, any value bigger than this will be truncated.
  - The bigger the number the more memory has to be allocated,
  - NOTE. To get a pre-trigger of X seconds this BufferSize has to be at least X+10s to ensure SW trigger operations to allow for network jitter and software trigger delays.
- BlockSize(Samples) = 10000
  - This is the data size for each read from the DAQ hardware.
  - NOTE The system is designed and tested to have this specified to the same values as SampleRate, meaning 1s between reads.

### 9.3.4 TSN

The time synchronization feature of the Lundby Logger System is the very core to the Distributed data logging system. It relies on the clock being forwarded from a precise clock source over the network depending on the IEEE 802.1AS time synchronization. In this setup the priorities of network devices are important to ensure the right clock master is selected.

Looking in Figure 26, the devices in the bottom layer should never act as masters.

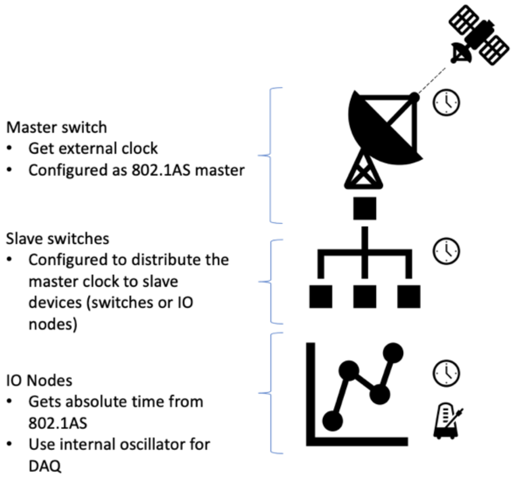



*Figure 26. Time synchronization in Lundby*

The TSN section defines the TimeSycnhronization parameter used by the IEEE 802.1AS.

Normally these parameters can be left untouched, except for the GrandMasterID that might be worth changing. To match the GrandMaster clock of the system.

NOTE. If multiple reduntant time masters are used, specifying the GrandMasterID will result in TimeSynch Alarms.

- Interface = "eth0"
  - Specifies the ethernet interface to use for time synch. For a standard Lundby system this should be left at eth0.
  - NOTE. Specify "" to disable time synch on the target
- Expected GrandMasterID = ""
  - To be able to detect a change of GrandMaster, it is possible to specify the GrandMaster clock ID. This means that when the Node detects a change in GrandMaster clock it sets the TimeSynch Alarm
- Priority1 = 248
  - Specifies the IEEE 802.1AS Priority 1 value
  - The Logger Node should have a value higher than the intermediate switches on the network (i.e. lower priority)
- Priority2 = 250
  - Specifies the IEEE 802.1AS Priority 2 value
  - The Logger Node should have a value higher than switches on the network (i.e. lower priority)
- MaxStartupDeviation_ns = 500
  - At startup a LoggerNode waits until the TimeSynchronization is within this time (specified in nano seconds).
  - NOTE. Since the DAQ system relies on good synchronization the startup synch is important
- MaxStartupWaitTime_ms = 10000
  - Maximum time to wait until the Node reaches the specified MaxStartupDeviation

- MaxSyncDeviation_ns = 100
  - This value is used when the Node checks the time synchronization for the Status handling

### 9.3.5 ALARM_CONFIG

The Alarm configuration section specifies the Alarm and warning levels to be used within the nodes. Alarms and Warnings are signaled to the user both in the Status messages from the nodes and optionally with LED outputs (See 10.1)

Alarms and Warnings are also available in the system as a bitpattern in the system channels SYS-Warnings and SYS- Alarms, with the following weights

- Bit0, 0x01 = CPU Level %
- Bit1, 0x02 = Memory Level %
- Bit2, 0x04 = Disk Level %
- Bit3, 0x08 = Propagation Delay %
- Bit4, 0x10 = TimeSynchOffset %
- Bit5, 0x20 = LogFileCount

The following configurations are possible.

- CheckTimeSynch
  specifies if the TimeSynch monitoring should be active. Normally this should always be active, but in the case of a mini system where only one node is active, this can be set to FALSE to avoid Warnings/Alarms due to missing time source.
- Warning levels, Alarm Levels
  - CPU-Level(%)
    Specifies the maximum CPU level in %
  - MEM-Level(%)
    Specifies the maximum Memory level in %
  - Disk-Level(%)
    Specifies the maximum Disk level in % (Logging disk)
  - PropagationDelayl(%)
    Specifies the maximum time from a packet is received until it has been forwarded to all trigger checks, continusous logging etc. Measured in percent of the time it takes to sample a block of data.
  - TimeSynchOffset_ns
    Specifies the maximum offset from the GrandMasterClock in nano seconds. This item is only active if CheckTimeSynch is TRUE. NOTE. TimeSynch alarm/warning can also be set if the correct time master is not present or if the TimeSynch system is not active.
  - LogFileCount
    Specifies the maximum count of LogFiles. This item is basically an alarm feature to detect that the StreamerService is not connected to the Node.

#### 9.3.5.1 *Alarm Levels*

It is also possible to specify digital outputs that can be used from triggers as well as

- System
  This digital output mirrors the Controller LED, see 10.1for information regarding blinkpattern
- UserLevel1 – UserLevel7
  These digital outputs are controlled from the Logging configurations and is activated one a trigger is active.
- UserLedInverted
  If TRUE the digital UserLevel outputs is inverted, i.e., normally ON

## 9.4 Managing Logger nodes

The logger system must have at least one Logger Node, and managing the nodes currently in the system is done in the configuration panel under "Manage LoggerNodes"

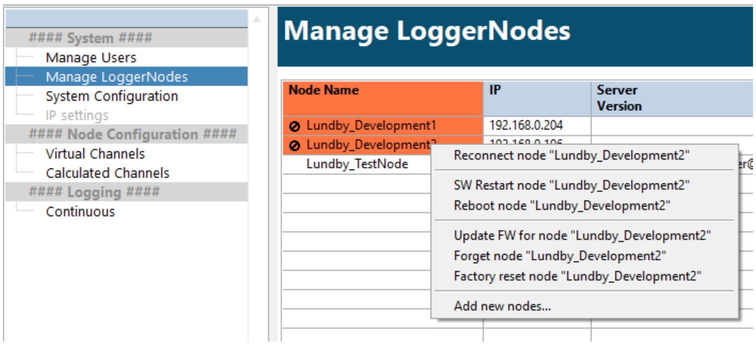



*Figure 27. Manage Logger Nodes panel*

Rightclicking a node allows the user to modify an existing node by:

- Reconnect
  - Reconnects all links to the node
- SW Restart
  - Performs a SW restart, i.e. restart the Lundby Logger Node SW but not the Real time OS
- Reboot
  - Performs a hardware reboot, i.e. shuts down and restarts the RealTime OS
- Update FW
  - Updates the firmware of the node
- Forget
  - Forgets the node, i.e., releases the node for other systems to adopt
- Factory reset
  - Resets all configuration items to default

### 9.4.1 Adopting Nodes

To add a new node simply select

- "Add new nodes…"

This will bring up a dialog with all nodes detected by the UDP multicast messaging

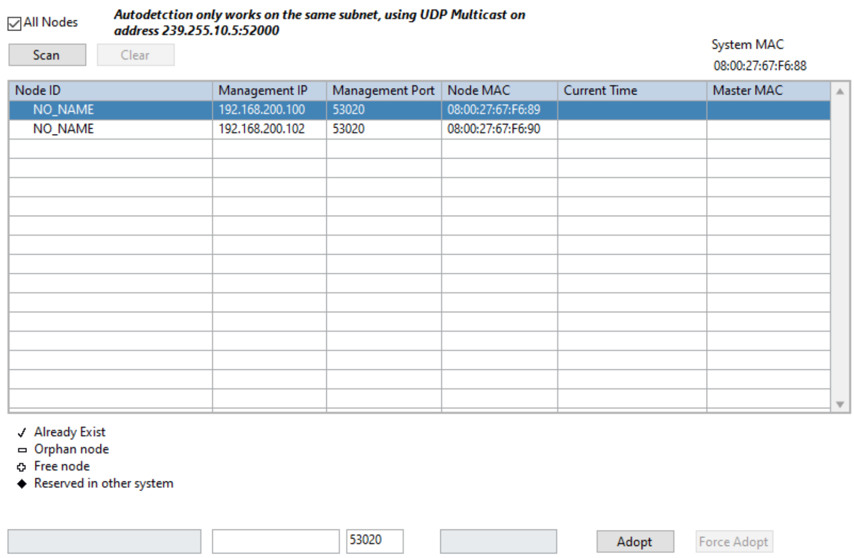



*Figure 28. List of detected nodes*

Selecting a node and then press "Adopt" displays the dialog to name the new node.

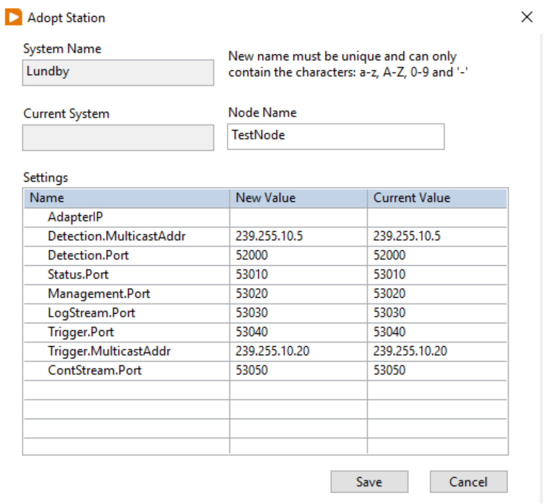



*Figure 29. Naming the new node*

Pressing save in the dialog updates the network settings on the node according to the GUI settings, and finally restarts the node.

If everything is correct the node should show up in the Logger node management list.

## 9.5 Virtual Channels

Virtual channels are connected to a physical channel and converts the raw channel input to a physical value using scaling etc.

The Main configuration view for Virtual channels shows the Logger Nodes to the left and an overview of the Virtual channels for the selected node in the middle. On the right the current configuration of the selected channel is displayed, see Figure 30.

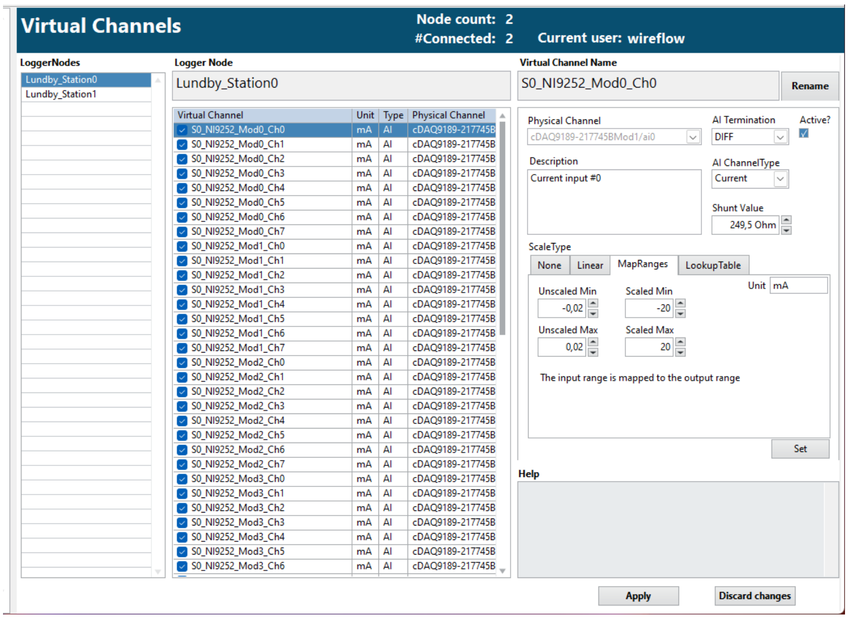



*Figure 30. Virtual channel main view*

Once all edits for the channels on a node is completed press Apply to save the values, this will bring up a dialog prompting to restart. Please note that it is possible to skip restart and continue editing but that a restart of each edited Node is required to activate the configuration.

### 9.5.1 Add, Duplicate, Delete...

Adding, duplicating, and deleting channels are done by right-click in the channel overview list.

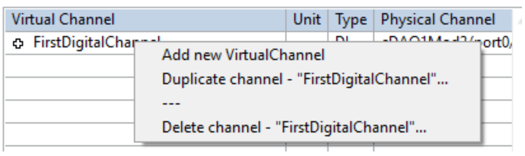



*Figure 31. Virtual channel list contextual menu*

Adding a new channel or duplicating an existing channel brings up a dialog to enter the name of the new channel. Channel name must be unique and start with a letter (a-z or A-Z) and can in addition contain the characters 0-9 '-' and '_'. If the name is not OK it is not possible to press the OK button.

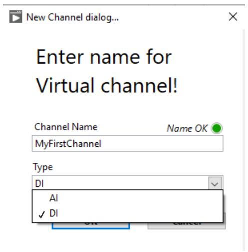



*Figure 32. Virtual channel name dialog*

In case it is a new channel the user can also select the type of the channel (e.g. AI or DI).

When multiple channels are selected it is possible to delete more than one channel at once.

### 9.5.2 Overview Symbols

The overview of the Virtual channels displays at minimum Name, Unit, Type and Physical channel, but expanding the window horizontally shows more data.

The list displays a symbol to the left of each Virtual Channel name to quickly get an overview of changes currently made.

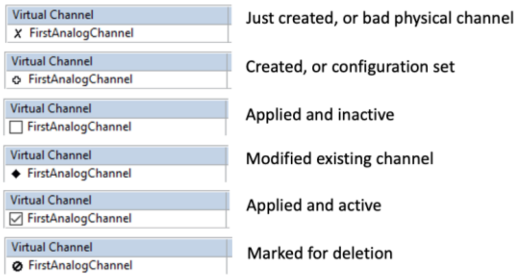



*Figure 33. Virtual Channel Symbols*

All changes are either applied or discarded by pressing one of the two buttons in the lower right corner.

**NOTE. The changes must be applied before selecting a new Node**

### 9.5.3 Editing a Virtual channel

Once a Virtual channel is selected it is possible to edit the configuration of the channel. Depending on if the connected physical channel is analog or digital different options are available. Regardless of the type of channel it is always possible to select a new (unused) physical channel of the same type, change the description, change unit and set the channel as active or inactive.

NOTE. After the changes are complete it is important to press Set to get the configuration change – otherwise the changes will be discarded.

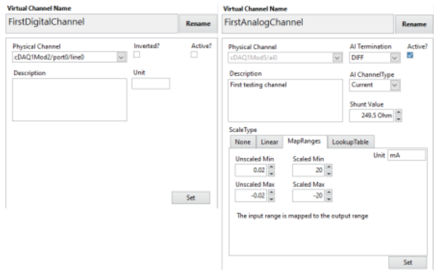


*Figure 34. Virtual Channel edit, Digital vs Analog*

For analog channels it is possible to additionally specify signal-conditioning (see Figure 34)

- AI Termination
  - Specifies how the hardware is connected to the measurement point. The available choices depends on the actual hardware but typical values are
    - DIFF - differential connections
    - NRSE - Non-Referenced Single Ended
    - RSE – Referenced Single Ended
- AI Channel type
  - Specifies what kind of measurement the channel is used for
- Shunt Value
  - Specifies the shunt resistor value in Ohms
  - Available only if AI channel type specifies a Current measurement and the hardware reports shunt-resistor support.
- Scale Type
  - Specifies how the data should be scaled to physical units.
  - Select from None (no scaling), Linear, MapRanges and Lookup table
  - NOTE. This is applied after any shunt resistor conversion
- Scale properties
  - Depending on the Scale Type different properties is used to define the scaling.

Digital channels are much simpler and only allows the user to specify if the digital input should be inverted or not.

#### 9.5.3.1 *Analog scaling example*

To perform analog scaling it is necessary to specify both type of scaling and the values defining the scale.

As an example lets define the linear scale y = 5 + 2 * x in the system using Linear, Map Ranges and Lookup scale types

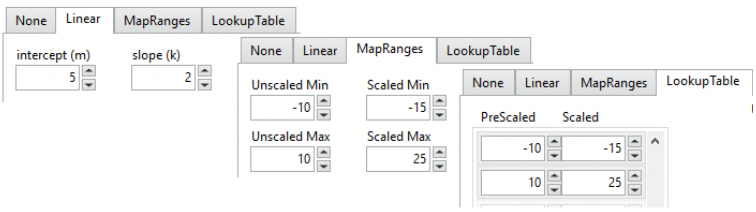



*Figure 35. Scaling configuration example*

Defining the scale as a linear scale is straightforward and only involves setting m, an k-values. Using Map ranges we enter the the raw input range and maps that to the scaled range. In this case the lookup table is basically the same as Map range.

The usage of the lookup table comes when the scaling is not purely linear, e.g. if we know that input 0 should be -0.1 in the output.

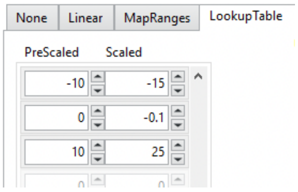



*Figure 36. Adding a segment to the lookup*

## 9.6 Calculated channels

Calculated channels are intended to be used to get channel characteristics in real time, e.g. to define triggering conditions based on a channel characteristic like RMS or mean value.

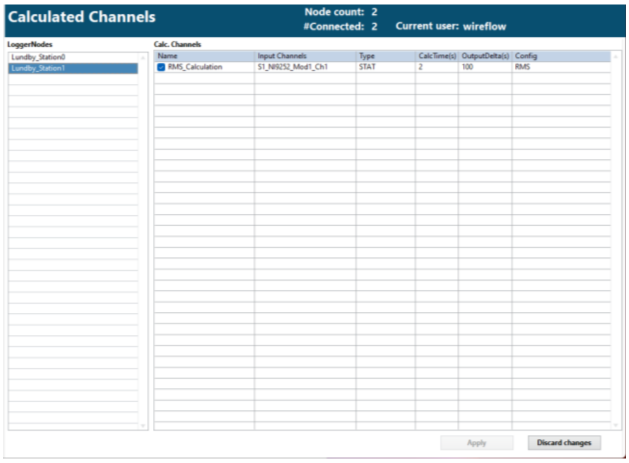



*Figure 37. Calculated Channel Overview*

The main configuration view for the Calculated channels shows the Logger Nodes to the left and an overview of the defined Calculated channels for the selected node to the right.

### 9.6.1 Add, Duplicate, Delete…

Adding, duplicating, and deleting channels are done by right-click in the channel overview list.

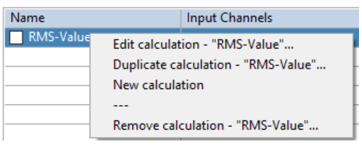



*Figure 38. Calculated channel list contextual menu*

Adding a new channel or duplicating an existing channel brings up a dialog to enter the name of the new channel. Channel name must start with a letter (a-z or A-Z) and can in addition contain the characters 0-9 '-' and '_'

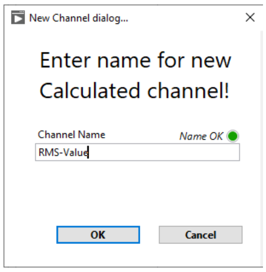



*Figure 39. Calculated channel name dialog*

When multiple channels are selected it is possible to delete more than one channel at once.

### 9.6.2 Overview Symbols

The overview of the Calculated channels displays at minimum Name, Input channels, Type, CalcTime(s), Output rate and, but expanding the window horizontally might show more data if available.

The list displays a symbol to the left of each channel name to quickly get an overview of changes currently made.

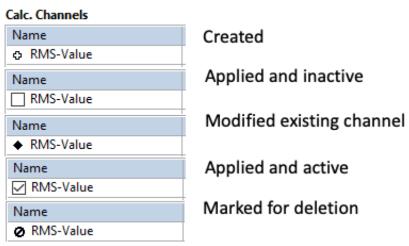



*Figure 40. Calculated Channel Symbols*

All changes are either applied or discarded by pressing one of the two buttons in the lower right corner.

**NOTE. The changes must be applied before selecting a new Node**

### 9.6.3 Editing Calculated channels

Editing a calculated channel brings up a new window that allows us to specify how the calculation should be performed

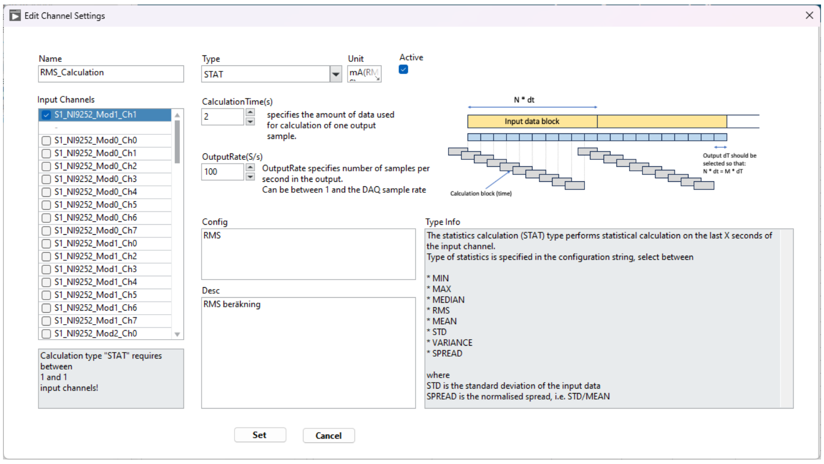



*Figure 41. Calculated channel editor*

The work flow defining a calculation as:

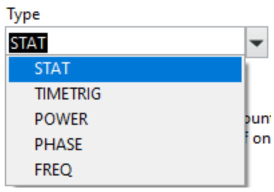



*Figure 42. Calculation types*

1. Select calculation type
    a. STAT – for statistics for a channel
    b. TIMETRIG – to get periodic measurements
    c. POWER – Calculates the power based on Current and Voltage input channels
    d. PHASE – Measures the phase between two input channels
    e. FREQ – Measures the frequency of an input channel
2. Specify input channels
    a. Number of channels needed depends on the selected type
3. Specify CalculationTime(s)
    a. This specifies the amount of data (specified in seconds) used for the calculation.
    b. NOTE. The longer the time the more CPU/Memory a calculation requires
4. Specify OutputRate(S/s)
    a. Data normally arrives in blocks of 1s with a data rate of 10kS/s (see configuration in 9.3.3), The OutputRate must therefore be in the range of 1 to 10kS/s
    b. NOTE. A higher OutputRate means more CPU and more Memory to be used.
5. Specify the configuration in the Config field, using the information in the Type info field as a help.
    a. Make sure that spelling gets correct as the channel fails to load otherwise
6. Add optional Description and Unit
7. Set the channel active to be able to use it.
8. Press Set to save the channel to be applied or cancel to discard editing.

### 9.6.4 Calculation Types

#### 9.6.4.1 STAT

The statistics type supports the following statistical calculations. The type is specified in the Config field

- MIN
  - Returns the minimum value of the last CalcTime(s) of data
  - NOTE. Tracking minimum value requires the system to hold all CalcTime(s) of previous values sorted in memory for each output sample
- MAX
  - Returns the maximum value of the last CalcTime(s) of data
  - NOTE. Tracking maximum value requires the system to hold all CalcTime(s) of previous values sorted in memory for each output sample
- MEDIAN
  - Returns the median value of the last CalcTime(s) of data
  - NOTE. Tracking median value requires the system to hold all CalcTime(s) of previous values sorted in memory for each output sample
- RMS
  - Returns the true RMS value of the last CalcTime(s) of data
  - Calculated as: $RMS = \sqrt{\frac{\sum_{k=1}^{N} x_k^2}{N}}$
- MEAN
  - Returns the mean value of the last CalcTime(s) of data
  - Calculated as: $X_{mean} = \frac{\sum_{k=1}^{N} x_k}{N}$
- VARIANCE
  - Returns the statistical variance of the last CalcTime(s) of data
  - Calculated as: $Var = \frac{\sum_{k=1}^{N}(x_k - x_{mean})^2}{N}$
- STD
  - Returns the statistical Standard Deviation of the last CalcTime(s) of data
  - Calculated as: $Std = \sqrt{Var}$
  - NOTE. STD is the same as $RMS_{AC}$
- SPREAD
  - Returns the spread of the input signal, calculated as Std/Mean.
  - Normalized STD

Please note that the RMS calculation is a True RMS calculation, meaning it considers both offset and waveform type.

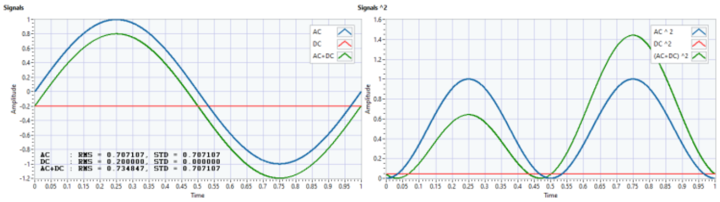


*Figure 43. RMS for AC, DC and combined*

To get $RMS_{AC}$ please use the STD as this removes the offset component. If we expand the Variance calculation as follows

$$Var = \frac{\sum_{k=1}^{N}(X_k - X_{mean})^2}{N} = \frac{1}{N}\left(\sum_{k=1}^{N}(X_k^2 - 2X_kX_{mean} + X_{mean}^2)\right) =$$

$$Var = \frac{\sum_{k=1}^{N} X_k^2}{N} - \frac{2X_{mean}\sum_{k=1}^{N} X_k^1}{N} + \frac{X_{mean}^2 \sum_{k=1}^{N} 1}{N}$$

$$Var = \frac{\sum_{k=1}^{N} X_k^2}{N} - X_{mean}^2$$

Looking at this final form, this can be interpreted as

$$RMS_{AC}^2 = RMS_{true}^2 - RMS_{DC}^2$$

I.e., where the Standard deviation is the $RMS_{AC}$ and the mean value is the $RMS_{DC}$.

#### *9.6.4.2 TIMETRIG*

Time trig generates a time pulse on the defined time.

The configuration specifies the start time with three parameters separated by commas ','

(D) Day in month (1-31), (H) Hour on day(0-23), (M) Minutes in hour (0-59)

Each value kan be either just a value, just a asterisk "*" or a value prefixed by *.Time is defined with the input string

E.g. a configuration string of "*, *12, 30" means output is active (output is 1) any day of month, every 12:th hour at minute = 30

#### *9.6.4.3 POWER*

Returns the average of the product of two input signals (current and voltage)

$$Power = \frac{\sum_{k=1}^{N} I_k * U_k}{N}$$

#### *9.6.4.4 PHASE*

Returns the phase between two input signals in degrees

**NOTE, since this is a FFT based operation it can be quite CPU intensive so adjust CalculationTime and OutputRate accordingly.**

##### *9.6.4.5 FREQ*

Returns the frequency of the input signal in Hz

**NOTE, since this is a FFT based operation it can be quite CPU intensive so adjust CalculationTime and OutputRate accordingly.**

### 9.6.5 OutputRate vs CalcTime(s)

The combination of output rate and CalcTime can have a high impact on the CPU and Memory. This section aims at describing how the combination affects the number of calculations necessary for each update.

OutputRate basically determines how many samples per second the output should contain. And the CalcTime specifies how much data is used for the calculation of each of these output samples.

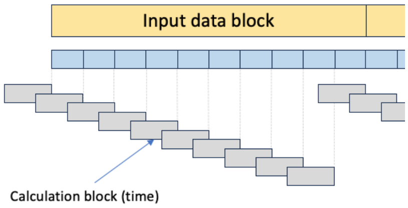



*Figure 44. OutputRate vs CalcTime(s)*

**Example 1:**

We have an input containing a 230V/50Hz signal, sampled at 10kS/s. On this signal we'd like to monitor the frequency continuously.

50Hz corresponds to a period 0.02s

We setup the calculation to use 10 waveforms (0.2s) and that the output should contain 5 Samples/s.

This means that the system will perform 5 frequency calculations (FFT) on 0.2s seconds of data, and since this is a perfect match 0.2 * 5 = 1s, the system uses 10kS for the calculation.

**Example 2:**

The same input as Example 1 but now the user defines that each calculation should be based on 1s of data and that the output should be 100S/s

In this case the system will perform 100 frequency calculations on 100 kS data (10s * 10kS/s), in total a 10 MS of data for each new input data.

The system allows both these configurations but it is the user responsibility to check that CPU, Memory and Propagation Delay is still in a valid range (see 8.2)

Some calculations are heavier than others and to set an exact number of the limit is therefore impossible since it also depends on other factors like SampleRate, Number of triggers and number of channels in the system.

## 9.7 Continuous Logging

The system supports continuous logging of data regardless of trigger states.

Continuous data is limited to log data between one and 200 S/s, and is most useful to monitor trends or calculated entities like RMS values.

Each Logger Node logs data to files named according to the Scheme:

Cont_<SYSTEM>_<NODE-NAME>_<TIMESTAMP>Z.tdms

- SYSTEM is the name of the system owning the LoggerNode
- NODE-NAME is the name of the individual node
- TIMESTAMP is a timestamp in UTC with 'Z' indicating the UTC+0, e.g. Cont_Lundby_S1_20231105_121624Z

Editing the configuration for continuous logging requires editing the included channels as well as the lograte etc.

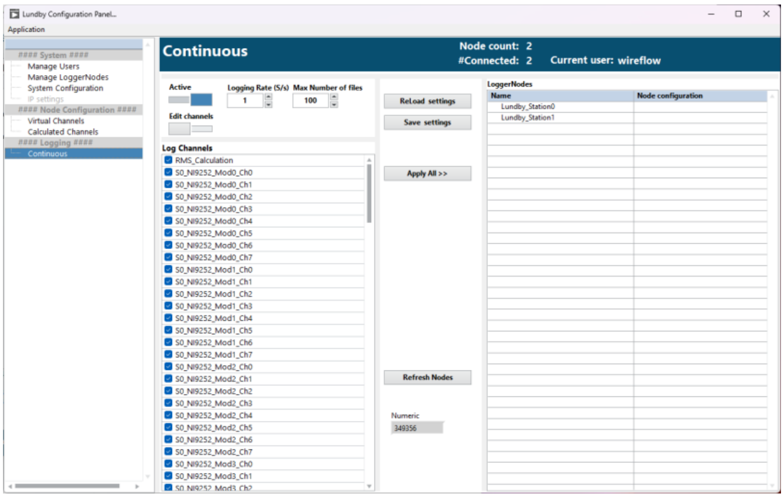



*Figure 45. Continuous logging configuration*

### 9.7.1 Buttons

**ReLoad settings**, Reload settings from the local PC disk

**Save settings**, save settings to local PC disk

**Apply All >>**, apply current visible configuration to all active nodes

**Refresh Nodes**, refresh node listing and update LoggerNodes list status

### 9.7.2 Edit channels

To edit the channels enable the switch Edit channels, then select and deselect channels from the list.

Once editing is done, disable the switch again.

The symbol to the left of the channel name indicates the state of the channel.

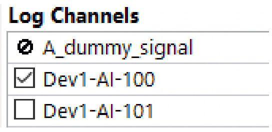



*Figure 46. Continuous logchannel symbols*

The symbols in Figure 46 means

- The channel is not recognized on any connected Node
- The channel is present and is currently used
- The channel is present but not currently used.

### 9.7.3 Logging rate (S/s)

The Logging Rate (S/s) defines the maximum number of samples per second to save to file.

Channels with slower data rates will still log data at their native rate

### 9.7.4 Max Number of files

Since the Continuous data files are stored on a RAM disk on the Node to prevent degrading the logdisk over time.

The Max number of files defines the maximum number of files to keep until the oldest files will be discarded. This is only invoked in cases where the Streamer Service is not connected for some reason.

Each logfiles normally contains 5s of data (see 9.3.2), and the StreamerService fetches data every 10-20 s depending on the number of logfiles generated.

### 9.7.5 Logger Node list

The logger node list displays the status of each node configuration compared to the GUI configuration

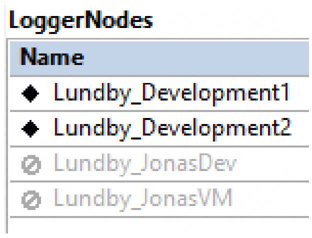



*Figure 47. Continuous Logger node symbols*

The symbols in Figure 47 means

- No symbol, – the node is active with a matching configuration
- Diamond, ◆, – the node is active but with a configuration mismatch
- Warning, ⊘ ,– the node is currently not active.

## 9.8 Triggers

The definition of a trigger can span multiple LoggerNodes, e.g. one node has the trigger conditions and other nodes are only logging data, and the configuration is stored locally on the PC in the GUI configuration folder. A trigger existing only on the Lundby Server cannot be used and will be deleted when triggers are applied.

C:\ProgramData\WireFlow\LundbySystem\GUI\Triggers

Using the GUI triggers can be applied to the LoggerNodes from the Overview panel (see 9.8.1). Once triggers are applied and activated they are evaluated whenever new data is received from the DAQ system, determined by the Sample Rate and the Block Size, see 9.8.9.

Each Logger Node logs data to files (see 9.8.10) named according to the scheme

<TRIGGER>_<TIMESTAMP>Z _<NODE-NAME>.tdms

- TRIGGER is the name of the trigger writing to the file
- SYSTEM is the name of the system owning the LoggerNode
- NODE-NAME is the name of the individual node
- TIMESTAMP is a timestamp of the block where the trigger was detected, formatted in UTC with 'Z' indicating the UTC+0

A trigger is active on a node if any Log channels or any Trig conditions exists on the node. A Trigger can be setup as

- Manual trigger
    - If a trigger is configured without any trig conditions, it can still be started manually and log data with pre-trigger etc.
- Trigger signaling
    - If a trigger is configured without LogChannels it is still active and can be used signal trigger conditions to other nodes
    - It can also be used to monitor specific conditions and have these conditions present on the User LEDs.
- Normal trigger
    - A trigger defined with both Log Channels and Trig conditions will start logging when a trig conditions is met.
    - NOTE. Some Logger Nodes might only have trigger conditions, and are then used for trig detection only, while other nodes might only have Log channels and only used for channel logging.

### 9.8.1 Overview

Right clicking the Trigger status list in the Main window (see 8.4.4) and selecting Edit triggers, brings up the trigger overview.

| | | BinaryTrig_AND | BinaryTrig_OR | FAT TC006 ManualTrig |
|---|---|---|---|---|
| | | *Active = TRUE*<br>*TrigCount 1, LogCh 0* | *Active = TRUE*<br>*TrigCount 1, LogCh 8* | *Active = FALSE*<br>*TrigCount 1, LogCh 2* |
| **Lundby_DevProject** | *Active : 3 (6)*<br>*Orphans : 0* | Trig 1, Ch 0 | Trig 1, Ch 8 | |
| **Lundby_Development1** | -- ?? -- | -- ?? -- | -- ?? -- | -- ?? -- |

*Figure 48. Trigger overview*

The trigger overview lists one Logger Node per row, and the defined triggers in columns.

- Inactive on Node
    - If a trigger is not active on a specific Node the cell background is white and there is no text
- Disconnected Node
    - If a Node is not connected all its cells are white background with the text "-- ?? --"
- Node configuration doesn't match
    - If the applied configuration on the Logger Node doesn't match the system configuration, the cell background is yellow. The text indicates number of active trig conditions and number of active logging channels on the node.
- Node configuration is correct
    - If the applied configuration on the Logger Node matches the system configuration, the cell background is green. The text indicates number of active trig conditions and number of active logging channels on the node.

The first cell under each trigger indicates if the trigger configuration is active and the total number of conditions and logging channels that are defined.

First cell to the right of the Logger Node indicates the number of active triggers (out of X applied), as well as the number of orphans.

An orphan is a trigger definition that doesn't exist in the system configuration, this configuration is automatically removed when triggers are applied to the node.

From the trigger overview double click a trigger column to edit the configuration.

If the external configuration changed, e.g. new nodes connected or if Virtual channels has been reconfigured, refresh the list to get new data in to configurator.

When editing is done, remember to apply configuration to the Logger Nodes.

NOTE. If configuration is not applied the configuration is still saved locally on the PC, and can be applied later.

## 9.8.2 Trigger editing

After a double-click on the trigger we enter the trigger configuration

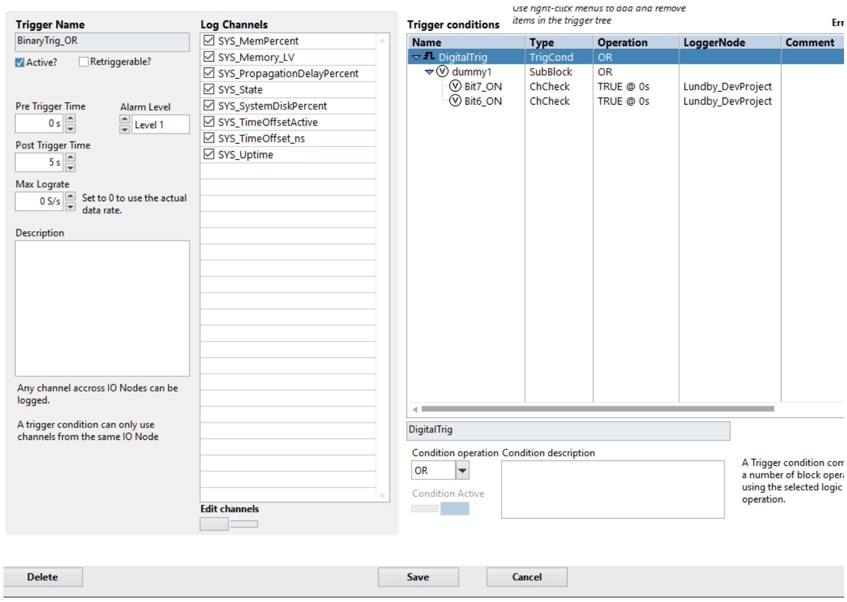


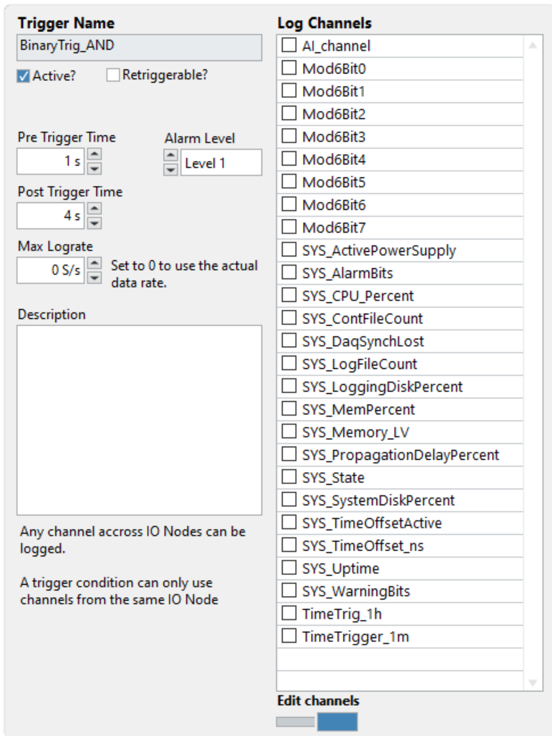

*Figure 49. Trigger configuration view*

From this view we configure general trigger configurations, as well as the trigger conditions. Once all editing is done press either Save or Cancel to save the trigger to the PC local disk.

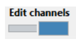
Pressing delete, removes the trigger from PC disk

## 9.8.3 General Trigger configuration

The general configuration of a trigger includes the following configuration items:




- Active ?
  - This checkbox indicates if the trigger is active in the system.
- Retriggerable
  - If checked the trigger accepts new trigs while the previous trigger is still active. Essentially prolongs the trigger active state
- Alarm Level
  - This selector indicates if the trigger should be indicated on the User LEDs, and which level to use.
  - NOTE. The LED output is signaled on all LoggerNodes with the trigger active.

- Pre-Trigger time
  - Specifies amount of time, in seconds, before the trigger to be recorded (max 120s)
  - NOTES.
    - For performance reasons the pre-trigger time is rounded so that the complete data block (see Figure 63) containing the pre-trigger time is saved to file.
    - The system tracks the time between triggers, and if the previous trigger time is greater than the Pre-trigger time, only the new data is saved to disk.
- Post-Trigger Time
  - Specifies amount of time, in seconds, the trigger is active after the time it was detected. (max 3600s)
  - NOTE. Both Pre-Trigger and Post-Trigger cannot be 0
- Max Lograte
  - Specifies the maximum rate (Samples/seconds) for the logdata. This means that it is possible to log data at a slower rate than the DAQ system is actually running.
  - Lograte can never be greater than the DAQ rate, and a value of less than or equal to 0 uses maximum rate possible.
- Description
  - Text describing the Trigger
- Log Channels
  - Contains a list of log channels that should be saved to disk when the trigger is active.
  - Enable list-editing by switching on the “Edit channels” switch, 

. The mark the channels to use, and mark the checkbox.
  - NOTE. This can be empty in cases where a trigger is only used for monitoring.

### 9.8.4 Trigger conditions

A trigger conditions is based on a combination of Channel checks, i.e. checking that a channel enters a defined state.

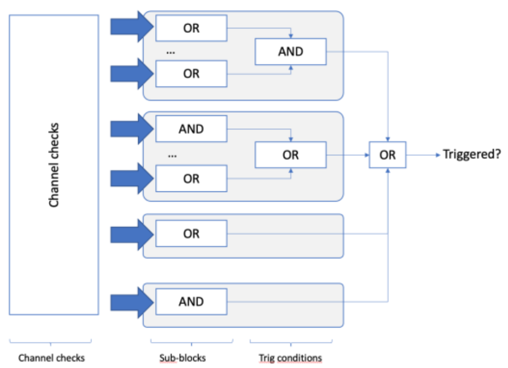



*Figure 50. Trigger evaluation*

The basic operation of the trigger engine is:

1. Evaluate all Channel checks
2. Combine Channel checks in Subblocks
3. Combine Subblocks in Trigger conditions
4. The first Trig condition that has an active range is used as the detected trigger

NOTE. A trigger is only detected if it has an inactive region first, e.g., swithing on a digital input only returns one trigger.

NOTE2. Only Subblocks and channel checks with all steps defined on a single node is evaluated, i.e. define separate trig conditions on different Nodes and let the system trigger between nodes using SW triggering.

### 9.8.5 Navigating the Trigger condition Tree

Editing the trigger evaluation is done in the Trigger conditions tree. Right-clicking to add, delete and use items.

It is possible to have unused SubBlocks and Channel checks, and these are presented at the bottom of the tree under a separator.

**Trigger conditions** *items in the trigger tree*

| Name | Type | Operation | LoggerNode |
|---|---|---|---|
| DigitalTrig | TrigCond | OR | |
| BitAND | SubBlock | AND | |
| Bit1_ON | ChCheck | FALSE @ 0s | Lundby_DevProject |
| Bit6_ON | ChCheck | FALSE @ 0s | Lundby_DevProject |
| BitAND2 | SubBlock | AND | |
| Bit1_ON | ChCheck | FALSE @ 0s | Lundby_DevProject |
| Bit6_ON | ChCheck | FALSE @ 0s | Lundby_DevProject |
| AnalogChannel | ChCheck | [ 0 , 0 ] @ 0s | Lundby_DevProject |
| Bit7_ON | ChCheck | TRUE @ 0s | Lundby_DevProject |

*Figure 51. Trigger condition tree*

There are three element types: TrigCond, SubBlock and ChCheck. Depending on the type of element that is right-clicked different options are available.

Some options are globally available:

- New -> Trigger Condition
  - Create new Trigger Condition
- New -> Sub Block
  - Create new Subblock
- New -> Channel Check
  - Create new Channel check

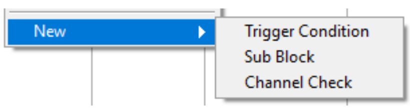



*Figure 52. Global options*

For item specific options see the other sections.

The tree view displays items with the following symbols

, Channel used in Channel Check is not on the same LoggerNode as other channels, TrigCondition is disabled

, ChannelCheck or SubBlock is unattached to a parent

, Indicates a trigger condition

, SubBlock or Channel check is part of a AND operation (specified by the parent item)

, SubBlock or Channel check is part of a OR operation (specified by the parent item)

### 9.8.6 Editing TrigCond type

The tree right-click options for TrigCond items are

- Duplicate
  - Duplicates the Trig condition
- Delete
  - Deletes the Trig condition
- Add Subblock
  - Makes it possible to add SubBlocks not currently under the TrigCond

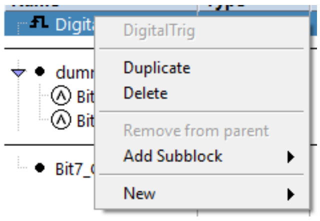



*Figure 53. TrigCond options*

Selecting a TrigCond item in the tree makes it possible to select the type of operation and to add a description of the Trig condition. Supported operations are AND or OR.

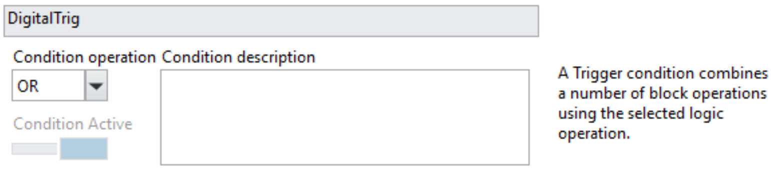



*Figure 54. Trigger condition specification*

### 9.8.7 Editing SubBlock type

The tree right-click options for SubBlock items are

- Duplicate
  - Duplicates the SubBlock
- Delete
  - Deletes the SubBlock
- Remove from parent
  - Remove the SubBlock check from the parent
  - NOTE. This is only available if the clicked item is under another item in the tree
- Add Channel Check
  - Makes it possible to add Channel checks not currently under the Subblock.

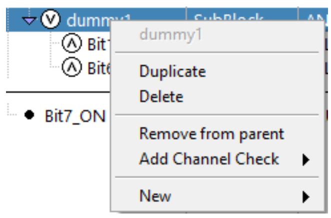



*Figure 55. Sub-block options*

Selecting a SubBlock item in the tree makes it possible to select the type of operation and to add a description of the subblock. Supported operations are AND or OR.

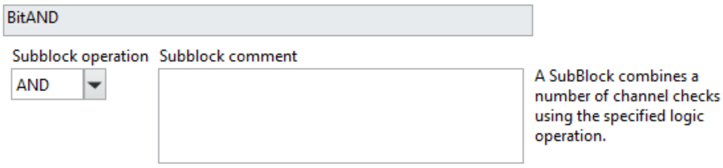



*Figure 56. SubBlock specification*

### 9.8.8 Editing a Channel Check item

The tree right-click options for Channel checks are

- Duplicate
    - Duplicate the Channel check
- Delete
    - Deletes the channel check and removes it from parent items
- Remove from parent
    - Remove the Channel check from the parent
    - NOTE. This is only available if the Channel check is under another item in the tree

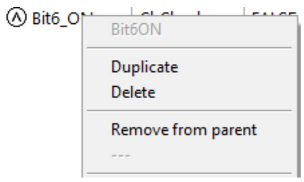



*Figure 57. Channelo check options*

Selecting and editing a Channel check depends on the type of channel selected. Digital input channels only specify the digital level to check, TRUE or FALSE, while an analog channel specifies a range.

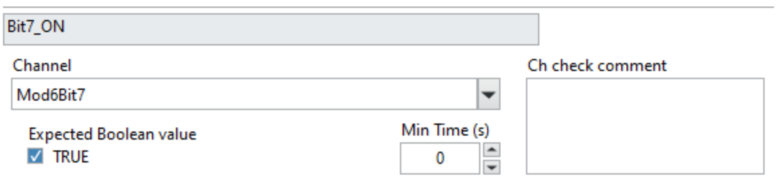



*Figure 58. Digital channel check*

For digital channels, the channel check either checks if the value is ON (1) or OFF (0). To check for the OFF state, the check is inverted.

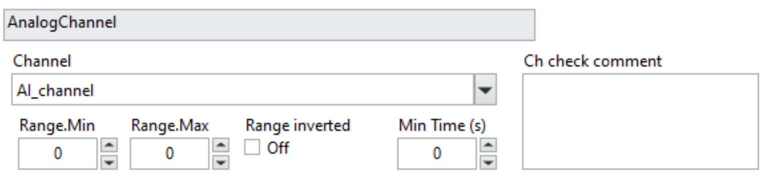



*Figure 59. Analog channel check*

For analog channels (including SYS- and calculated channels), the values are checked against a defined range. The range is defined with Min-and Max values with the option to invert the range.

The "Min Time(s)" is used to reject noise and requires the channel to be within range at least X seconds to accept the transition. Normally this value is very small

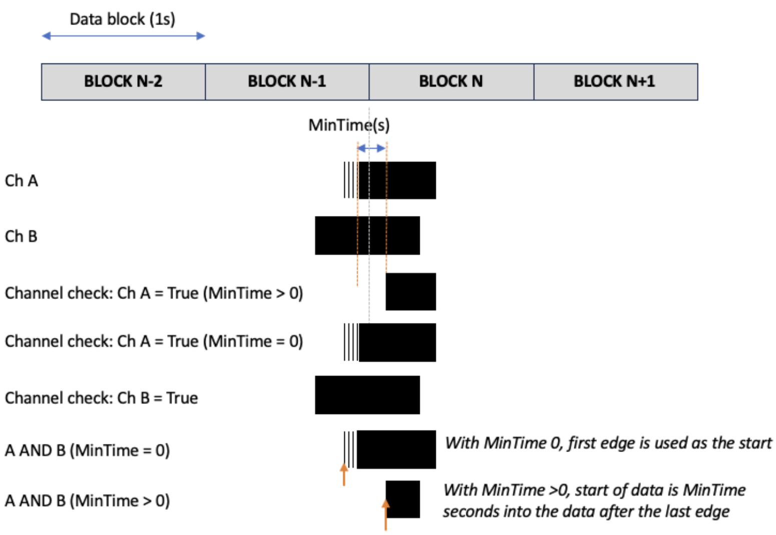



*Figure 60. The effect of MinTime(s) on ChannelCheck*

NOTE. If Min Time(s) is used on a channel the used edge from the Channel Check is the actual edge time plus the MinTime(s), See Figure 60.

#### 9.8.8.1 *Analog range check*

The analog range check is defined with a range with minimum and maximum values, but can also be inverted. Inverting the range means that values outside of the specified range is the active region.

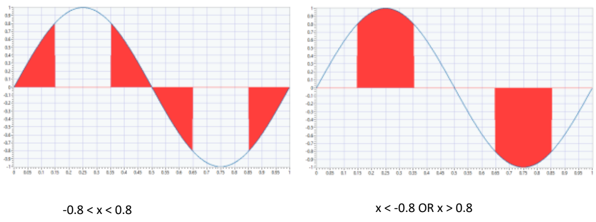



*Figure 61. Analog range, normal vs inverted*

This is illustrated in Figure 61, the graphs illustrate the active regions (filled with red) for normal range (left) and inverted range (right).

To detect a trigger the trigger must first be outside the range at least one sample, then enter the range. Entering the range from above or below doesn't matter.

### 9.8.9 Trigger evaluation

Since the triggered logging is such a crucial part of the system, this section outlines the process of evaluating triggers. The trigger evaluation runs on every new block of data received from the data acquisition (normally 1s of data), including SYS- and Calculated channels.

During evaluation the system takes the following actions

1. Perform all Channel checks on raw channel data. Channel data can have different data rates (e.g. System channels)
  - The output is a number of sections with start- and stop times where the Channel checks evaluate to TRUE
2. Next all SubBlocks are calculated, i.e. one or more Channel checks are combined into new sections with start- and stop times.
   **NOTE. A Channel check can be part of more than one SubBlock**
3. The final stage is to combine one or more SubBlocks into TriggerConditions. The output from each TrigConditions is the first time section where the Trigger condition is True.
4. Lastly the trigger checks if any TrigCondition has an active section, and returns the timestamps when a section is activated.

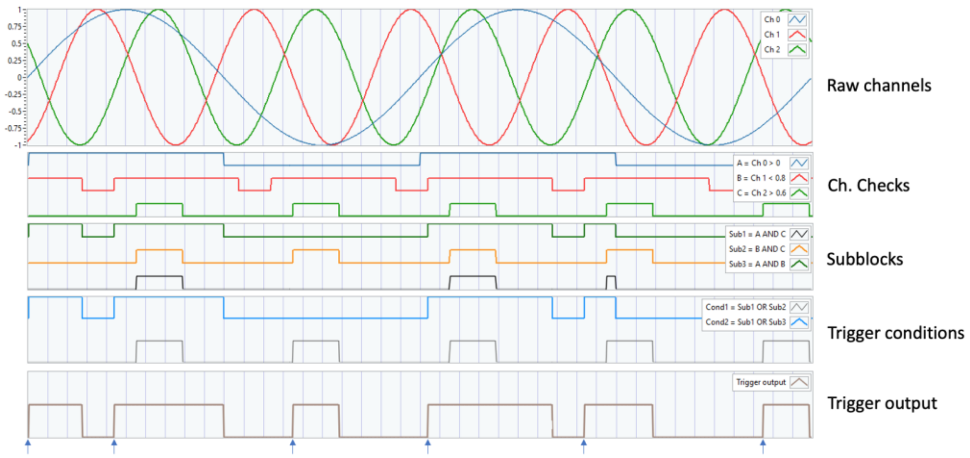



*Figure 62. Trigger evaluation example*

This process is illustrated in Figure 62, where the final trigger output detected multiple active regions.

#### *9.8.9.1 Pre-Trigger data*

When the trigger is inactive and detects an active trigger section, the Pre-Trigger data is saved relative to the first section detected.

#### *9.8.9.2 Retrigger*

When the trigger is active and detects a new trig it will record the new trigger to the active logfile as a record with Trigger reason (name of the condition) and the timestamp when it was detected.

In addition, if Retrigger is active, the recording time will be reset, making the logging run for a longer time

### 9.8.10 TDMS file

The data format for both Continuous logging as well as Triggered logging is TDMS. TDMS is a binary data format with META data. TDMS file format is developed by NI, and is described at

https://www.ni.com/en/support/documentation/supplemental/06/the-ni-tdms-file-format.html

The benefits of TDMS is the ability to add META data to individual channels, channel groups or the file itself, meaning that system properties, channel properties as well as trigger properties can be written to file.

There are plugins for various applications to enable them reading the TDMS file format, e.g. Excel and MatLab.

For performance reasons the data are written as Waveforms, meaning that instead of creating and writing sample timestamps, data are written with properties on the channel defining the timing and the sample length

- wf_samples
  number of samples written each time in the waveform
- wf_increment
  time between samples in seconds
- wf_start_time
  timestamp of the first sample of the data
- NI_ChannelLength
  total number of samples written to the file

This is a much more efficient way of storing data, as the option is to add time data for each channel in the file (because they can have different sample rates).

Since continuous logging focus on “slow” data and generates many smaller logfiles, it only writes channel names and channel values to the TDMS files, no META data is added.

The triggered logging on the other hand adds system information to the file properties

- Node.ID = Identification of the Logger Node
- Node.ServerVersion = Version of the Logger Node software
- Node.ServerBuildTime = Build time of the Logger Node software

It also adds the first detected trigger in each data block to the file properties.

These properties are named

- Txxxx_<TrigReason>

Where xxx is just an ordinal number, and TrigReason is the Trigger condition causing the trigger. To find the sample index (0-indexed) of a specific time T (e.g. A trigger), use the calculation

Index = ( T – wf_start_time ) / wf_increment

Each channel gets the following extra properties added:

- Type
- Unit
- InputChannels
- Scaling
- Configuration
- Description

For performance reasons the system always saves data in complete DAQ blocks, meaning that some additional data is saved to the logfile if Pre- or Post-Trig is in another DAQ block.

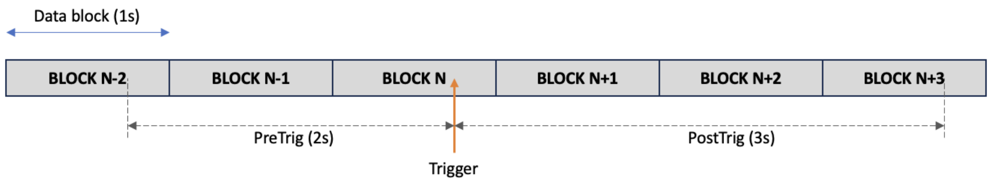



*Figure 63. Trigger data vs. DAQ blocks*

This is illustrated in Figure 63, where the trigger is found in Block N, the PreTrigger is found in Block N-2 and the PostTrig is found in Block N+3.

The total number of blocks written is 6 (from Block N-2 to Block N+3), but the actual time between PreTrig and Post trig is 5s, resulting in 1s extra time written to file.

# 10 System Status signaling

The Lundby System overview displays the individual status of each Lundby Logger Node. If connection is lost or during maintenance

## 10.1 System LED

The System supports status signaling using several LEDs

The User LED on the controller and the defined DO.System digital (see 9.3.5.1) output both signal the status of the running software according to Table 1 and Figure 64

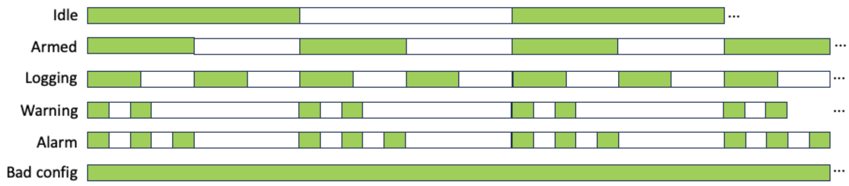



*Figure 64. LED patterns*

Symmetric patterns are indicating a normal state, while asymmetric patterns indicate something is not OK.

*Table 1. System status signaling patterns*

| State | LED pattern | Description |
|---|---|---|
| Normal - Idle | Slowly blinking:<br>2s OFF, 2s ON, … | The system is running normally but no trigger is active |
| Normal – Armed | Slowly blinking:<br>1s OFF, 1s ON, … | System has one or more armed triggers |
| Normal – Logging | Fast blinking:<br>0.5s OFF, 0.5s ON | The system is running normally and is currently logging data to disk |
| Warning | Blinks two times then pause:<br>1.4s OFF, 0.2s ON, 0.2s OFF, 0.2s ON | One or more defined warning levels has been reached |
| Alarm | Blinks three times the pause:<br>1s OFF, 0.2s ON, 0.2s OFF, 0.2s ON, 0.2s OFF, 0.2s ON | One or more defined alarm levels has been reached |
| Bad configuration | Steady ON | The system could not start because of a bad configuration |

## 10.2 Trigger LEDs

Each trigger can be defined to set a specific Alarm Level when active.

The trigger configuration (see 9.8.3) then allows these alarms to be sent through to a Digital output channel. The digital channel used for each level is defined in the general system configuration, see 9.3.5.1.

# 11 GUI files

The GUI stores the configuration under the program data folder ( %PROGRAMDATA% ).

%PROGRAMDATA%
- \WireFlow
  - \LundbySystem
    - \GUI
      - ConfiguredStations.ini = list of configured station and the network configuration
      - GUIConfiguration.ini = GUI specific config, including the system name
      - LundbyStreamer.ini = storage of streamer specific config
      - \Backups = holds the backups of the LoggerNodes
      - \GlobalCfg
        - RawConfig.ini = the default Node configuration
      - \Manual = contains user manuals for the LundbySystem
      - \SysLog = contains logfiles for debugging
      - \Triggers = contains trigger configurations
    - \StreamService
      - \LocalLogfolder = default location for the StreamerService local storage

# 12 Troubleshooting

This chapter helps in troubleshooting issues and also some insight in how the GUI can be used to prevent issues from arising.

## 12.1 Checking Lundby Performance

The Lundby system has no limitations on the number of VirtualChannels, Calculated channels or triggers. While this adds flexibility it also means that the user must check that a configuration is able to execute on the RealTime server.

The CPU specific measurements are shown on the IO Node Status Overview page (see 8.2), show as CPU (%) and Propagation Delay (%). A well balanced system should not have the CPU load running continuously above 50% in the idle state, and at full load the recommendation is to average below 70% CPU.

Propagation delay is mostly affected by the calculation channels and the number of triggers in the system. If this value gets closer to 100% the system cannot keep up, and the configuration must be changed not to loose any data. to The propagation delay is normally very stable and should not be above 60-70%, as this leaves to little time.

Memory is checked in the Node status detail window (see 8.4.1), and should never be above 70% of installed memory.

NOTE. When the Lundby Server starts memory will start growing until all Pre-trigger buffers has been filled, and this is perfectly normal.

*Table 2. System impact by configuration change*

| Configuration change | CPU impact | Memory impact | Comment |
|---|---|---|---|
| Virtual channels | Low | High | Each virtual channel must hold data for pretriggering, so both the sample rate and the pre-buffering size affects the impact. |
| Calc channels | Medium to high | Low | It depends on the configuration of the calculation, i.e. the output rate and the calculation size |
| Triggers | Low to medium | Low | It depends on how many and how complex the trig conditions are. |

## 12.2 General troubleshooting

This chapter should list the most common problems that a user might encounter and how to get more details

| Fault | Possible Reason | Solution |
|---|---|---|
| Local logfile count isn't updated | Streamer is not running. | Open services on windows and verify that Lundby-StreamerService is running |
| Channels are not seen after a reboot | Problem connecting to the timesynch master | Perform a SW restart of the node |
| Lundby-StreamerService is not connected | Installation failed due to user rights or missing info | Check if the Lundby-StreamerService is running as a service.<br>Verify that environment variables are correct (Windows 11), see 7 |

## 12.3 Alarms

The alarms displayed in the Node status window, see 8.4.1, might require user interaction to be resolved.

*Table 3. Node alarm explanations*

| Alarm name | Reason | Resolution |
|---|---|---|
| DAQ not running | DAQ process failed to start | Check Virtual channel configuration and DAQ general configuration and restart Node |
| UserAlarm1 | By Log configuration | User specific |
| UserAlarm2 | By Log configuration | User specific |
| UserAlarm3 | By Log configuration | User specific |
| UserAlarm4 | By Log configuration | User specific |
| UserAlarm5 | By Log configuration | User specific |
| UserAlarm6 | By Log configuration | User specific |
| UserAlarm7 | By Log configuration | User specific |
| CPU Level | CPU is above alarm limit | Decrease CPU usage by changing calculated channels, log configuration etc. |
| MEM Level | Memory is above alarm limit | Decrease Mem usage by changing calculated channels, log configuration etc. |
| Log disk Level | To many logfiles on the disk | Verify streamer service is running and that the disk is healthy |
| Propagation delay | Time for calculations and forwarding to logging tasks is taking too long | Check configuration |

| | | |
|---|---|---|
| TimeSynch | System detected a synchronization issue to the GrandMaster | In n some cases this can be resolved automatically, if not check network setup |
| LogFileCount | Too many logfiles on Logdisk | Verify streamer service is running and check logfile limit |
| DAQ SynchError | DAQ timestamp differs from system time | Restart DAQ, if the problem persists, restart Node |
| DAQ OutputError | Problem accessing DO channels | Verify DO module is in place and that the output configuration is correct. More details in Extras TAB, look for DAQ OutputError |
| DAQ AnalogError | Problem accessing AI channels | Verify AI modules are in place and that the Virtual Ch configuration is correct. More details in Extras TAB, look for DAQ AnalogError |
| DAQ DigitalError | Problem accessing DI channels | Verify DI modules are in place and that the Virtual Ch configuration is correct. More details in Extras TAB, look for DAQ DigitalError |
| DAQ FatalError | Generic DAQ error | Check Extras TAB, look for DAQ FatalError.<br>Restart Node, if the problem persists conatact support |
| DAQ ConfigError | Problem with the DAQ configuration | Check the Extras TAB for more info, verify VirtualCh configuration |
| Bad LogFolder Cfg | The logfolder specified isn't accessible | Change System configuration and restart Node |
| StreamService connection | Node has lost connection to Streamer Service for more than 120s | Make sure StreamService is running and that the PC has network connection to the Node. |

## 12.4 Error codes

*Table 4. Error codes*

| Code | Description | Source |
|---|---|---|
| | User privilege error, i.e. current user are not allowed to perform the action | GUI user level checks |
| | Virtual channel configuration error | |

## 12.5Network issues

Since the Lundby system is based on network communication and relies on these for the synchronization, a number of network specific issues might affect the performance.

*Table 5. Network issues*

| Symptom | Resolution |
|---|---|
| Nodes cannot be detected by the Lundby GUI | The Lundby nodes uses UDP Multicast messaging to be adopted and to send SW triggers between nodes.<br>• Make sure the Multicast addresses is the same on all nodes<br>• Make sure the switches are configured to allow multicast traffic |
| SW triggers between nodes are not working | The Lundby nodes uses UDP Multicast messaging to be adopted and to send SW triggers between nodes.<br>• Make sure the Multicast addresses is the same on all nodes<br>• Make sure the switches are configured to allow multicast traffic |
| Connecting or disconnecting a node cause drop-outs on other nodes | This issue is probably caused by the switches configured to use SpanningTree to protect and handle ethernet ring topologies.<br>• Disable Spanning Tree on the switches (OK if a start topology is used.<br>• Change topology or disable only a single port<br>NOTE. This doesn't affect logging and since the drop-outs are short, the time should still be in synch. |
| A node reports wrong time | This is probably caused by the node failing to connect to the 802.1AS time master.<br>• Check the TSN setting for the Node<br>• Check switch settings so that the actual ports are in the same PTP domain<br>NOTE. While booting a switch might report the system time until PTP time is synchronized. In an attempt to avoid timing issues the Lundby Node waits (max 60s) until the time received is no longer the UNIX Epoch |