Application Note AN034

Using redSync – introducing PTP redundancy

Version 1.17.0 - October 2023

Abstract

redSync is a utility allowing to add redundancy to PTP, it is part of the syn1588® Software Suite. redSync allows a PTP end device to process PTP messages from two PTP Grandmasters simultaneously and to dynamically select time information of the better PTP Grandmaster to synchronize its local clock. With redSync the redundancy is significantly enhanced compared to the basic Grandmaster failover mechanism provided by IEEE1588, because several error conditions are taken into account in contrast to the Best Master Clock Algorithm, which is triggered exclusively by the contents and/or absence of PTP Announce messages. redSync is based on the syn1588® PTP Stack as well as the syn1588® PTP hardware. The redSync utility is available starting with the release Q3/2020 (v1.12) of the syn1588® Software Suite. This application note describes the basic usage of this utility.

Primary assumption for operation

The redundancy provided by redSync relies on the following assumptions:

Both PTP Grandmasters use the same main time source. They would typically derive their respective time from a suitable GNSS source, however, they may use different connections to this time source, e.g., different GNSS receivers or lock to different GNSS systems.

redSync will provide seamless switchover, if both PTP Grandmasters provide the same/similar time, or to be more precise the two Grandmaster should differ less than 50ns. This is typically the case for GNSS synchronized PTP Grandmasters. Larger differences between the two PTP Grandmasters will result in delayed switchover.

Scenario A.1 – Hardware Clock Synchronization with syn1588® PCIe NIC

The basic application structure for using redSync to provide redundancy for a single hardware clock synchronization is as follows.

- Two independent instances of the syn1588® PTP Stack are executed on one single end device. Each syn1588® PTP Stack has to be configured to select a specific PTP Grandmaster. This can be easily accomplished by choosing different PTP domains for syn1588® PTP Stack A and B respectively.
- A single syn1588® PCIe NIC that connects to the network over which the two PTP Grandmasters can be accessed

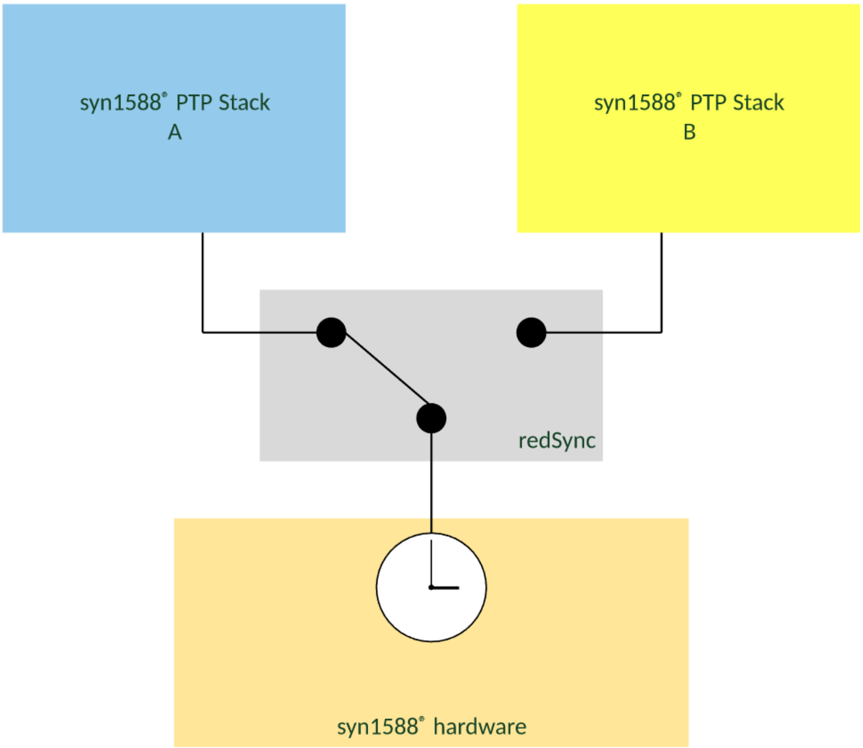

Figure 1: redSync for redundant Hardware Clock Synchronization

redSync observes the synchronization status of these two syn1588® PTP Stack instances and selects on a set of configurable criteria the syn1588® PTP Stack instance that actually shall maintain and control the high-precision PTP hardware clock. If, for example, the connection of the active syn1588® PTP Stack instance to its PTP Grandmaster is lost or dramatically degraded the second syn1588® PTP Stack instance will take over seamlessly without deteriorating the accuracy.

Scenario A.2 – Hardware Clock Synchronization with syn1588® Dual NIC

This sub scenario is similar to Scenario A.1 with the main difference that a syn1588 Dual NIC is used. With this setup it is possible to connect the two redundant ports to two separate physical network connection, providing an additional layer of redundancy.

Scenario B – System Clock Synchronization

The basic application structure for using redSync to provide redundancy for System Clock synchronization is as follows.

- Two independent instances of the syn1588® PTP Stack connect to one PTP Grandmaster each
- Each syn1588® PTP Stack uses a syn1588® PCIe NIC or another IEEE1588 capable network interface
- A degraded operation is also possible with one or both syn1588® PTP Stack instances operating in software mode
- The syn1588® lSync utility is configured for operation together with redSync
- redSync selects the best PTP grandmaster from the two PTP instances and forwards this information to lSync
- lSync uses the selected PTP instance (and the synchronized hardware clock) to synchronize the system clock

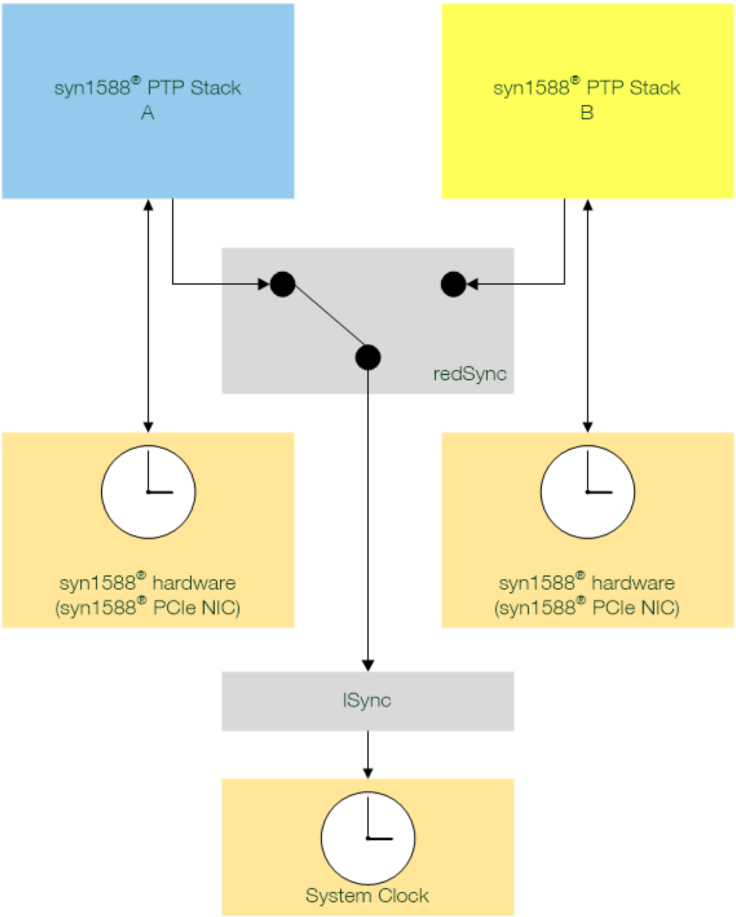

Figure 2: redSync for redundant System Clock Synchronization

Current Support

Note that, redSync is currently only supported on Linux OS. Windows support has not been fully qualified but is possible. Further note, Layer 2 mode is currently not supported.

The supported and tested PTP profiles for redSync are:

- default
- SMPTE (2059.1)
- telecom2 (G.8275.1v2.1)
- G8275_2_s
- enterprise

Support for other PTP profiles will be added step-by-step with the following releases of the syn1588® Software Suite.

Pre-requisites

Scenario A.1

The following pre-requisites are required to redundantly synchronize a PTP node using redSync:

- PTP node with syn1588® hardware offering two independent timestamping units on the same physical network interface
- PTP node with Linux OS (Windows support not fully qualified)
- two syn1588® PTP Stack instances bound to the same (!) physical network interface and thus to the same PTP hardware clock
- each of the two syn1588® PTP Stack instances have to use an independent timestamping unit
- Both syn1588® PTP Stack instances have to run in PTP slave mode
- Both syn1588® PTP Stack instances have to use the identical network mode and PTP profile

Scenario A.2

The following pre-requisites are required to redundantly synchronize a PTP node using redSync:

- PTP node with syn1588® Dual NIC
- PTP node with Linux OS (Dual NIC is currently not supported for Windows)
- Both syn1588® PTP Stack instances have to run in PTP slave mode
- Both syn1588® PTP Stack instances may use different network modes and PTP profiles
- The syn1588® module parameter “syn1588_extime_enable” has to be set to 1 to configure the Dual NIC for redSync operation, for example by manually loading the module:

```
>insmod syn1588nic syn1588_extime_enable=1
```

Scenario B

The following pre-requisites are required to redundantly synchronize a PTP node using redSync:

- PTP node with two syn1588® PCIe NICs
  OR
  with two IEEE 1588 capable NICs
- PTP node with Linux OS (Windows support not fully qualified)
- two syn1588® PTP Stack instances each bound to one of the two NICs
- Both syn1588® PTP Stack instances have to run in PTP slave mode
- The syn1588® PTP Stack instances may use different network modes and PTP profiles
- IPv4 and Linux OS: Both syn1588® PTP Stacks will need to use different IPv4 subnets or VLAN

Important

The current selection modes “simple” and “bmc” do not incorporate the check for a synchronization accuracy boundary. Use a setup with high message rates (>= 8/s) for Sync and DelayReq/Resp messages for a seamless switchover in case of re-establishment of the old, best PTP Master. The boundary/-b parameter of lSync can be utilized to achieve a similar result, refer to the measurement for Scenario B for details.

Configuration

redSync connects to the two syn1588® PTP Stack instances specified on the command line via their respective PTP PortId = ClockID + Port Number. Depending on the selected decision algorithm redSync decides which of the two syn1588® PTP Stack shall be used as time source and thus actually synchronize the node’s clock. This selection is done via the Shared Memory interface. If redSync initiates a switch from one syn1588® PTP Stack instance to another the clock control implemented in the syn1588® PTP Stacks or the syn1588® lSync utility will guarantee a seamless switchover, IFF the primary assumption for redSync operation is satisfied!

redSync may be started in a restrictive “monitored” mode, which will stop operation if certain criteria are not met or in “unmonitored” mode, which will only log unmet criteria.

redSync may be started with different selection modes. “Simple” uses the first port (Port A) by default as source and only switches to the second port (Port B) if the first is no longer available. “BMC” applies the PTP Best Master Clock algorithm as criteria to select between the two PTP Stacks.

Decision mode (compare method) [-m simple | bmc]	simple (default)	redSync sets the first port (port A) active as long as it is alive and in slave state, otherwise the second (port B).	
	bmc	redSync decides based on the IEEE1588 best master clock algorithm (BMCA)	
Environment mode (environment state) [-e unmonitored | monitored]	monitored strict condition checks	start	Both ports must be alive and in slave state.
		runtime	At least one port must be alive and in slave state. Otherwise redSync gives warning and terminates
	unmonitored (default) loose checks	start	redSync starts in all conditions and simply waits for a port alive and in slave state to set it as active port
		runtime	never terminates

Table 1: Mode overview

Parameter: Monitored mode (-e monitored)

In this mode redSync expects that the system on which it is operated is monitored by a supervisor. For that case redSync expects strict start conditions to be fulfilled. If any of them is not met it will not start and return information about the problem. With this information the supervisor can and has to decide what needs to be done to bring the system into an operational state. redSync will also monitor these conditions throughout the operation and will terminate if both redundant PTP ports violate the conditions at the same time.

These conditions are

- syn1588® PTP stack running for both PTP ports.
- Both PTP ports are in slave state.

Parameter: Unmonitored mode (-e unmonitored)

redSync will not enforce the above mentioned conditions. It will start as well as continue to run even if these are violated. Violation of the conditions will only be reported in the log output.

Parameter: Simple Mode (-s simple)

In this mode redSync is instructed to choose the first given PTP port A as default.

If the connection to the PTP Grandmaster or the synchronization to the PTP Grandmaster fails for PTP ort A, redSyc will switch to PTP port B which is now allowed the control and maintain the PTP hardware clock. PTP port B will remain the active control port unless PTP port A synchronizes again to a PTP Grandmaster. redSync will switch back in such an event to PTP port A. PTP port B actually acts as a backup connection for PTP port A in this mode of operation.

Parameter: BMC Mode (-s bmc)

The BMC algorithm is normally used by the syn1588® PTP Stack to determine which PTP Grandmaster in a PTP clock domain shall be chosen. redSync uses this decision algorithm to decide between two PTP Grandmasters that are active in different PTP clock domains. redSync decides based on the Grandmaster properties and will allow the PTP port connected to the better PTP Grandmaster to be used as source. If the active PTP port loses the connection to its Grandmaster or the Grandmaster's performance is degraded the other PTP port will be selected.

Parameter: Port A (-A <clockID,portNumber>) and Port B (-B <>)

These two parameters are mandatory and set the two ports that are used for redundant synchronization.

A port is identified by its Port ID, as specified by the IEEE1588 standard. Hence, a clockID and a port number have to be given as value to this parameter.

For example:

"-A 8c:a5:a1:ff:fe:00:10:20,1 -B 8c:a5:a1:ff:fe:33:33:33,1"

This will select a syn1588® PTP Stack that is currently controlling the PTP Clock identified by its clockID 8c:a5:a1:ff:fe:00:10:20 and connected to the PTP port number 1 for Port A. Whereas Port B is identified by a different clock ID = 8c:a5:a1:ff:fe:33:33:33 and port number 1.

This example setup is a basic configuration for a redSync application suited for Scenario B, i.e., using two syn1588® interfaces in parallel (different clockIDs).

Parameter: System Clock Synchronization (-l)

This flag parameter (takes no value) has to be used to setup redSync for choosing the source to synchronize the system clock = Scenario B. If this parameter is not set, redSync will chose the source for synchronizing the hardware clock = Scenario A.

Parameter: Verbosity (-v 1|2|3|4)

This parameter can be used to get more verbose log output. A higher number will increase the verbosity level of redSync.

Parameter: help (-h)

This will print additional help for redSync usage into the shell.

Example – Scenario A.1

The setup of the system is as depicted in Figure 1.

Two syn1588® PTP Stack instances have to be invoked in the background and have to synchronize to two different PTP grandmaster instances over the network.

A single syn1588® PCIe NIC is required for this setup.

The different programs should be invoked in the following order.

syn1588® PCIe NIC

For this example a syn1588 PCIe NIC is mounted in the PC and named “enp1s0”.

The MAC address of this NIC is: 8c:a5:a1:00:00:c0 and the clockID for PTP operation is derived from it.

clockID: 8c:a5:a1:ff:fe:00:00:c0

syn1588® PTP Stack for Port A

```
> ptp ./portA.cfg
```

Where the PTP Stack configuration file contains the following:

```
port1:
interface enp1s0
domain 50
```

… to synchronize to a PTP Grandmaster (PTP domain: 50) over the local Ethernet interface “enp1s0”

This PTP instance can be identified by its clockID (derived from the interface “enp1s0” and the port number.

PortID: 8c:a5:a1:ff:fe:00:00:c0, 1

syn1588® PTP Stack for Port B

```
> ptp ./portB.cfg
```

Where the PTP Stack configuration file contains the following:

```
port1:
interface 127.0.0.1
port2:
interface enp1s0
domain 100
timestamper userTS
```

… to synchronize to a PTP Grandmaster (PTP domain: 100) over the same local Ethernet interface “enp1s0” and using a different timestamper unit (userTS) for PTP Packet timestamps.

Note that “port1” links to the local loopback to create a dummy port for the PTP Stack for the second instance. The PTP Stack instance for Port B is actually setup to operate as “port2”.

This PTP instance can be identified by its clockID (derived from the interface “enp1s0” and the port number.

PortID: 8c:a5:a1:ff:fe:00:00:c0, 2

syn1588® redSync

```
> sudo ./redSync -A 8c:a5:a1:ff:fe:00:0d:2b,1 -B  8c:a5:a1:ff:fe:00:0d:2b,2 -v
info -m simple -e unmonitored
```

After invocation redSync checks the port states of both PTP ports and gives the control of the clock to one of PTP port; in this case to PTP port A. The following log output shows a successful invocation of redSync:

```
2022-01-10 12:55:40.130359 [INFO    ] [              ]  syn1588(R) redSync -
Redundant Synchronization Engine
2022-01-10 12:55:40.130397 [INFO    ] [              ]  Build date: 2022-01-
05T16:10:27 - v1.14-12-g36e5ee73
2022-01-10 12:55:40.130408 [INFO    ] [              ]  Copyright (c) Oregano
Systems - Design & Consulting GesmbH 2005-2022
2022-01-10 12:55:40.130416 [INFO    ] [              ]  Confidential unpublished
data - All rights reserved
2022-01-10 12:55:40.130424 [INFO    ] [              ]  Command line: ./redSync -A
8c:a5:a1:ff:fe:00:0d:3b -B 8c:a5:a1:ff:fe:00:0d:3c -m bmc -v info -l -e
unmonitored -f /media/oregano/LIVESYN1588/redsync_an034.txt
2022-01-10 12:55:40.130625 [INFO    ] [1.sharedmemory.api.io]  Init shared mem
2022-01-10 12:55:40.130876 [INFO    ] [1.sharedmemory.api.io]  Init shared mem
2022-01-10 12:55:40.131051 [INFO    ] [CredSyncEngine]  =======================
2022-01-10 12:55:40.131065 [INFO    ] [CredSyncEngine]  Current Iteration: 0
2022-01-10 12:55:40.131075 [INFO    ] [CredSyncEngine]  =======================
2022-01-10 12:55:40.131194 [INFO    ] [CredSyncEngine]       [Port A] is active!
2022-01-10 12:55:40.131323 [INFO    ] [CredSyncEngine]       [Port B] is available
but not active!
2022-01-10 12:55:40.131339 [INFO    ] [CredSyncEngine]  [RESULT] Port A : active
2022-01-10 12:55:40.131350 [INFO    ] [CredSyncEngine]  [RESULT] Port B :
available
2022-01-10 12:55:41.131904 [INFO    ] [CredSyncEngine]  =======================
2022-01-10 12:55:41.131935 [INFO    ] [CredSyncEngine]  Current Iteration: 1
2022-01-10 12:55:41.131945 [INFO    ] [CredSyncEngine]  =======================
2022-01-10 12:55:41.132003 [INFO    ] [CredSyncEngine]      [Port A] is active!
2022-01-10 12:55:41.132048 [INFO    ] [CredSyncEngine]      [Port B] is available
but not active!
2022-01-10 12:55:41.132062 [INFO    ] [CredSyncEngine]  [RESULT] Port A : active
2022-01-10 12:55:41.132076 [INFO    ] [CredSyncEngine]  [RESULT] Port B :
available
2022-01-10 12:55:42.132166 [INFO    ] [CredSyncEngine]  =======================
2022-01-10 12:55:42.132193 [INFO    ] [CredSyncEngine]  Current Iteration: 2
2022-01-10 12:55:42.132203 [INFO    ] [CredSyncEngine]  =======================
2022-01-10 12:55:42.132252 [INFO    ] [CredSyncEngine]      [Port A] is active!
2022-01-10 12:55:42.132296 [INFO    ] [CredSyncEngine]      [Port B] is available
but not active!
2022-01-10 12:55:42.132310 [INFO    ] [CredSyncEngine]  [RESULT] Port A : active
2022-01-10 12:55:42.132321 [INFO    ] [CredSyncEngine]  [RESULT] Port B :
available
2022-01-10 12:55:43.132753 [INFO    ] [CredSyncEngine]  =======================
2022-01-10 12:55:43.132782 [INFO    ] [CredSyncEngine]  Current Iteration: 3
2022-01-10 12:55:43.132815 [INFO    ] [CredSyncEngine]  =======================
2022-01-10 12:55:43.132872 [INFO    ] [CredSyncEngine]      [Port A] is active!
2022-01-10 12:55:43.132920 [INFO    ] [CredSyncEngine]      [Port B] is available
but not active!
2022-01-10 12:55:43.132934 [INFO    ] [CredSyncEngine]  [RESULT] Port A : active
2022-01-10 12:55:43.132946 [INFO    ] [CredSyncEngine]  [RESULT] Port B :
available
```

If redSync has to switch the PTP port controlling the PTP hardware clock it will output these log messages:

```
2022-01-10 12:56:06.143437 [INFO    ] [CredSyncEngine]  =======================
2022-01-10 12:56:06.143463 [INFO    ] [CredSyncEngine]  Current Iteration: 26
2022-01-10 12:56:06.143472 [INFO    ] [CredSyncEngine]  =======================
2022-01-10 12:56:06.143522 [INFO    ] [CredSyncEngine]      [Port A] is active!
2022-01-10 12:56:06.143570 [INFO    ] [CredSyncEngine]      [Port B] is available
but not active!
2022-01-10 12:56:06.143582 [INFO    ] [CredSyncEngine]  [RESULT] Port A : active
2022-01-10 12:56:06.143593 [INFO    ] [CredSyncEngine]  [RESULT] Port B :
available
2022-01-10 12:56:07.144068 [INFO    ] [CredSyncEngine]  =======================
2022-01-10 12:56:07.144097 [INFO    ] [CredSyncEngine]  Current Iteration: 27
2022-01-10 12:56:07.144107 [INFO    ] [CredSyncEngine]  =======================
2022-01-10 12:56:07.144157 [INFO    ] [CredSyncEngine]      [Port A] No new data
can be fetched from ptp. Check if it is still running !
2022-01-10 12:56:07.144172 [WARNING ] [CredSyncEngine]      [Port A] Port is not
available !
2022-01-10 12:56:07.144214 [INFO    ] [CredSyncEngine]      [Port B] is available
but not active!
2022-01-10 12:56:07.144228 [INFO    ] [CredSyncEngine]  [RESULT] Port A : not
available
2022-01-10 12:56:07.144238 [INFO    ] [CredSyncEngine]         [CHANGE] Detected!
At Port B
2022-01-10 12:56:07.144280 [INFO    ] [1.redsync     ]  Port B sets system clock
adjustment to 1
2022-01-10 12:56:07.144294 [INFO    ] [CredSyncEngine]         [CHANGE] New
Active Port B
2022-01-10 12:56:07.144303 [INFO    ] [CredSyncEngine]         [CHANGE] Old
Active Port A
2022-01-10 12:56:07.144311 [INFO    ] [CredSyncEngine]  [RESULT] Port B : active
2022-01-10 12:56:08.144961 [INFO    ] [CredSyncEngine]  =======================
2022-01-10 12:56:08.144989 [INFO    ] [CredSyncEngine]  Current Iteration: 28
2022-01-10 12:56:08.145000 [INFO    ] [CredSyncEngine]  =======================
2022-01-10 12:56:08.145052 [INFO    ] [CredSyncEngine]      [Port A] No new data
can be fetched from ptp. Check if it is still running !
2022-01-10 12:56:08.145068 [WARNING ] [CredSyncEngine]      [Port A] Port is not
available !
2022-01-10 12:56:08.145111 [INFO    ] [CredSyncEngine]      [Port B] is active!
2022-01-10 12:56:08.145124 [INFO    ] [CredSyncEngine]  [RESULT] Port A : not
available
2022-01-10 12:56:08.145133 [INFO    ] [CredSyncEngine]  [RESULT] Port B : active
```

Example – Scenario B

The setup of the system is as depicted in Figure 2.

Two syn1588® PTP Stack instances have to be invoked in the background and have to synchronize to two different PTP grandmaster instances over the network.

Two syn1588® PCIe NIC are required for this setup.

In addition to redSync, an instance of the syn1588® lSync utility has to be invoked.

For the sake of demonstration, Port B will be synchronizing to a free-running PTP Grandmaster that provides a PTP time that is offset at the time of this test to a few hundred ns to the PTP Grandmaster on Port A, which is synchronized to GPS.

syn1588® PCIe NIC Port A

For this example a syn1588® PCIe NIC is mounted in the PC and named “enp1s0”.

The MAC address of this NIC is: 00:1e:c0:f6:c7:3d and the clockID for PTP operation is derived from it.

clockID: 00:1e:c0:ff:fe:f6:c7:3d

syn1588® PTP Stack for Port A

```
> ptp -I enp1s0 -d 51
```

… to synchronize to a PTP Grandmaster (PTP domain: 51) over the local Ethernet interface “enp1s0”

This PTP instance can be identified by its clockID (derived from the interface “enp1s0” and the port number (default: 1).

PortID: 00:1e:c0:ff:fe:f6:c7:3d, 1

syn1588® PCIe NIC Port B

For this example a syn1588 PCIe NIC is mounted in the PC and named “enp2s0”.

The MAC address of this NIC is: 00:1e:c0:85:d0:0c and the clockID for PTP operation is derived from it.

clockID: 00:1e:c0:ff:fe:85:d0:0c

syn1588® PTP Stack for Port B

```
> ptp -I enp2s0 -d 50
```

… to synchronize to a PTP Grandmaster (PTP domain: 50) over the local Ethernet interface “enp1s0”

This PTP instance can be identified by its clockID (derived from the interface “enp2s0” and the port number (default: 1).

PortID: 00:1e:c0:ff:fe:85:d0:0c, 1

syn1588® redSync

```
> ./redSync -A 00:1e:c0:ff:fe:f6:c7:3d,1 -B 00:1e:c0:ff:fe:85:d0:0c,1 -m bmc -v 3 -l -e unmonitored
```

After invocation redSync checks the port states of both PTP ports and gives the control of the clock to one of the PTP ports; in the example shown below to PTP port A. The following log output shows a successful invocation of redSync:

```
2022-01-17 15:59:20.567500 [INFO    ] [             ]  syn1588(R) redSync - Redundant Synchronization Engine
2022-01-17 15:59:20.567689 [INFO    ] [             ]  Build date: 2022-01-11T15:47:05 - v1.14-35-g20f13f6e
2022-01-17 15:59:20.567701 [INFO    ] [             ]  Copyright (c) Oregano Systems - Design & Consulting GesmbH 2005-2022
2022-01-17 15:59:20.567710 [INFO    ] [             ]  Confidential unpublished data - All rights reserved
2022-01-17 15:59:20.567717 [INFO    ] [             ]  Command line: ./redSync -A 8c:a5:a1:ff:fe:00:0d:3c,1 -B 8c:a5:a1:ff:fe:00:0d:3b,1
-v info -m bmc -l -e unmonitored
2022-01-17 15:59:20.567765 [INFO    ] [             ]  Port 0: adding config "A" = "8c:a5:a1:ff:fe:00:0d:3c"
2022-01-17 15:59:20.567787 [INFO    ] [             ]  Port 0: adding config "B" = "8c:a5:a1:ff:fe:00:0d:3b"
2022-01-17 15:59:20.567802 [INFO    ] [             ]  Port 0: adding config "v" = "info"
2022-01-17 15:59:20.567814 [INFO    ] [             ]  Port 0: adding config "m" = "bmc"
2022-01-17 15:59:20.567825 [INFO    ] [             ]  Port 0: adding config "l" = ""
2022-01-17 15:59:20.567835 [INFO    ] [             ]  Port 0: adding config "e" = "unmonitored"
2022-01-17 15:59:20.567845 [WARNING ] [             ]  More then one clock id was given!
2022-01-17 15:59:20.568029 [INFO    ] [1.sharedmemory.api.io]  Init shared mem
2022-01-17 15:59:20.568268 [INFO    ] [1.sharedmemory.api.io]  Init shared mem
2022-01-17 15:59:20.568410 [INFO    ] [CredSyncEngine]  =======================
2022-01-17 15:59:20.568424 [INFO    ] [CredSyncEngine]  Current Iteration: 0
2022-01-17 15:59:20.568434 [INFO    ] [CredSyncEngine]  =======================
2022-01-17 15:59:20.568542 [INFO    ] [CredSyncEngine]  [Port A] is not in slave but in  'Listening'  state. Waiting for slave state
...!
2022-01-17 15:59:20.568558 [WARNING ] [CredSyncEngine]       [Port A] Port is not available !
2022-01-17 15:59:20.568670 [INFO    ] [CredSyncEngine]  [Port B] is not in slave but in  'Uncalibrated'  state. Waiting for slave state
...!
2022-01-17 15:59:20.568685 [WARNING ] [CredSyncEngine]       [Port B] Port is not available !
2022-01-17 15:59:20.568695 [ERROR   ] [CredSyncEngine]       [CRITICAL] There is no available port left!
2022-01-17 15:59:20.568704 [INFO    ] [CredSyncEngine]  [RESULT] Port A : not available
2022-01-17 15:59:20.568713 [INFO    ] [CredSyncEngine]  [RESULT] Port B : not available
2022-01-14 10:12:25.759482 [INFO    ] [CredSyncEngine]  [RESULT] Port B : not available
…
```

```
…
2022-01-17 15:59:23.570414 [INFO    ] [CredSyncEngine]  =======================
2022-01-17 15:59:23.570444 [INFO    ] [CredSyncEngine]  Current Iteration: 3
2022-01-17 15:59:23.570454 [INFO    ] [CredSyncEngine]  =======================
2022-01-17 15:59:23.570504 [INFO    ] [CredSyncEngine]  [Port A] is not in slave but in  'Uncalibrated'  state. Waiting for slave state
...!
2022-01-17 15:59:23.570520 [WARNING ] [CredSyncEngine]       [Port A] Port is not available !
2022-01-17 15:59:23.570561 [INFO    ] [CredSyncEngine]       [Port B] is available but not active!
2022-01-17 15:59:23.570575 [INFO    ] [CredSyncEngine]  [RESULT] Port A : not available
2022-01-17 15:59:23.570585 [INFO    ] [CredSyncEngine]          [CHANGE] Detected! At Port B
2022-01-17 15:59:23.570627 [INFO    ] [1.redsync    ]  Port B sets system clock adjustment to 1
2022-01-17 15:59:23.570640 [INFO    ] [CredSyncEngine]          [CHANGE] New Active Port B
2022-01-17 15:59:23.570649 [INFO    ] [CredSyncEngine]  [RESULT] Port B : active
2022-01-17 15:59:24.571083 [INFO    ] [CredSyncEngine]  =======================
2022-01-17 15:59:24.571112 [INFO    ] [CredSyncEngine]  Current Iteration: 4
2022-01-17 15:59:24.571122 [INFO    ] [CredSyncEngine]  =======================
2022-01-17 15:59:24.571173 [INFO    ] [CredSyncEngine]       [Port A] is available but not active!
2022-01-17 15:59:24.571218 [INFO    ] [CredSyncEngine]       [Port B] is active!
2022-01-17 15:59:24.571232 [INFO    ] [CredSyncEngine]  [RESULT] Port A : available
2022-01-17 15:59:24.571244 [INFO    ] [CredSyncEngine]  [RESULT] Port B : active
```

The following log snippet displays the behavior of redSync when it detects an issue on the currently selected port. The issue is resolved by switching to the other port and, in this scenario, trigger a switch to Port A also for the connected utility ISync.

```
2022-01-17 16:10:42.966637 [INFO    ] [CredSyncEngine]  =======================
2022-01-17 16:10:42.966666 [INFO    ] [CredSyncEngine]  Current Iteration: 682
2022-01-17 16:10:42.966676 [INFO    ] [CredSyncEngine]  =======================
2022-01-17 16:10:42.966726 [INFO    ] [CredSyncEngine]       [Port A] is available but not active!
2022-01-17 16:10:42.966771 [INFO    ] [CredSyncEngine]  [Port B] is not in slave but in  'Listening'  state. Waiting for slave state
...!
2022-01-17 16:10:42.966826 [INFO    ] [1.redsync    ]  Port B sets system clock adjustment to 0
2022-01-17 16:10:42.966841 [WARNING ] [CredSyncEngine]       [Port B] Port is not available !
2022-01-17 16:10:42.966852 [INFO    ] [CredSyncEngine]          [CHANGE] Detected! At Port A
2022-01-17 16:10:42.966894 [INFO    ] [1.redsync    ]  Port A sets system clock adjustment to 1
2022-01-17 16:10:42.966908 [INFO    ] [CredSyncEngine]          [CHANGE] New Active Port A
2022-01-17 16:10:42.966917 [INFO    ] [CredSyncEngine]          [CHANGE] Old Active Port B
2022-01-17 16:10:42.966925 [INFO    ] [CredSyncEngine]  [RESULT] Port A : active
2022-01-17 16:10:42.966934 [INFO    ] [CredSyncEngine]  [RESULT] Port B : not available
2022-01-17 16:10:43.967172 [INFO    ] [CredSyncEngine]  =======================
2022-01-17 16:10:43.967201 [INFO    ] [CredSyncEngine]  Current Iteration: 683
2022-01-17 16:10:43.967211 [INFO    ] [CredSyncEngine]  =======================
2022-01-17 16:10:43.967260 [INFO    ] [CredSyncEngine]       [Port A] is active!
2022-01-17 16:10:43.967305 [INFO    ] [CredSyncEngine]  [Port B] is not in slave but in  'Listening'  state. Waiting for slave state
...!
2022-01-17 16:10:43.967330 [INFO    ] [1.redsync    ]  Port B sets system clock adjustment to 0
2022-01-17 16:10:43.967343 [WARNING ] [CredSyncEngine]       [Port B] Port is not available !
2022-01-17 16:10:43.967353 [INFO    ] [CredSyncEngine]  [RESULT] Port A : active
2022-01-17 16:10:43.967363 [INFO    ] [CredSyncEngine]  [RESULT] Port B : not available
```

syn1588® lSync

The lSync utility is used to synchronize the system clock to the PTP controlled hardware clock. Together with redSync, lSync chooses the better PTP Port, hence, the more accurate PTP Grandmaster. With the support of redSync, lSync switches automatically to the other PTP Grandmaster, if the connection to the original Grandmaster is lost or degrades in performance.

```
> ./lSync -v info -r -b 2.5us
```

The parameter “-b 2.5us” configures lSync to switch to the alternate Port selected by redSync, only if it is within a certain boundary, in this case 2500 ns.

Important: Note that this boundary configuration is higher than recommended and will result in obvious switchover behavior as is demonstrated in the measurement for Scenario B below.

The following snippet shows the first few seconds of operation.

```
2022-01-17 15:59:21.094517 [INFO    ] [             ]  syn1588(R) lSync - Local Synchronization engine
2022-01-17 15:59:21.094557 [INFO    ] [             ]  Build date: 2022-01-11T15:47:05 - v1.14-35-g20f13f6e
2022-01-17 15:59:21.094570 [INFO    ] [             ]  Copyright (c) Oregano Systems - Design & Consulting GesmbH 2005-2022
2022-01-17 15:59:21.094578 [INFO    ] [             ]  Confidential unpublished data - All rights reserved
2022-01-17 15:59:21.094586 [INFO    ] [             ]  Command line: ./lSync -r -v info -b 2.5us -f lSync_test9.txt
2022-01-17 15:59:21.094609 [INFO    ] [syn1588      ]  Syn1588Ifc requires at least:
2022-01-17 15:59:21.094621 [INFO    ] [syn1588      ]  - linux driver version 1.4-15-g05b7283
2022-01-17 15:59:21.094630 [INFO    ] [syn1588      ]  - windows driver version 10/05/2017, 10.9.16.182
2022-01-17 15:59:21.094684 [INFO    ] [syn1588      ]  Device /dev/syncD0 found
2022-01-17 15:59:21.094740 [INFO    ] [syn1588      ]  syn1588(R) Hardware Clock M 2.3.5 f=125000000 Hz
2022-01-17 15:59:21.094764 [INFO    ] [syn1588      ]  Found Ultimate TS support
2022-01-17 15:59:21.094775 [INFO    ] [syn1588      ]  Found stop clock support
2022-01-17 15:59:21.094847 [INFO    ] [syn1588      ]  Using MAC TS Version 3160
2022-01-17 15:59:21.094862 [INFO    ] [syn1588      ]  Using programmable 1-step TS
2022-01-17 15:59:21.094916 [INFO    ] [syn1588      ]  syn1588(R) PCIe NIC Revision 2.1, Build 876 with HQ Oscillator
2022-01-17 15:59:21.094949 [INFO    ] [syn1588      ]  Board revision 2.1.2.0
2022-01-17 15:59:21.094985 [INFO    ] [syn1588      ]  Serial: 012108343
2022-01-17 15:59:21.095049 [INFO    ] [clock        ]  Spike M2S: Init with ival 0, buffer size 16
2022-01-17 15:59:21.095310 [INFO    ] [clock        ]  Spike Path: Init with ival 0, buffer size 16
2022-01-17 15:59:21.095400 [INFO    ] [syn1588      ]  Device /dev/syncD1 found
2022-01-17 15:59:21.095450 [INFO    ] [syn1588      ]  syn1588(R) Hardware Clock M 2.3.5 f=125000000 Hz
2022-01-17 15:59:21.095472 [INFO    ] [syn1588      ]  Found Ultimate TS support
2022-01-17 15:59:21.095482 [INFO    ] [syn1588      ]  Found stop clock support
2022-01-17 15:59:21.095507 [INFO    ] [syn1588      ]  Using MAC TS Version 3160
2022-01-17 15:59:21.095518 [INFO    ] [syn1588      ]  Using programmable 1-step TS
2022-01-17 15:59:21.095566 [INFO    ] [syn1588      ]  syn1588(R) PCIe NIC Revision 2.1, Build 876 with HQ Oscillator
2022-01-17 15:59:21.095597 [INFO    ] [syn1588      ]  Board revision 2.1.2.0
2022-01-17 15:59:21.095629 [INFO    ] [syn1588      ]  Serial: 012108342
2022-01-17 15:59:21.095686 [INFO    ] [clock        ]  Spike M2S: Init with ival 0, buffer size 16
2022-01-17 15:59:21.095914 [INFO    ] [clock        ]  Spike Path: Init with ival 0, buffer size 16
2022-01-17 15:59:21.096023 [INFO    ] [clock        ]  Spike M2S: Init with ival 0, buffer size 16
2022-01-17 15:59:21.096267 [INFO    ] [clock        ]  Spike Path: Init with ival 0, buffer size 16
2022-01-17 15:59:21.096297 [INFO    ] [             ]  Found 3 clocks
2022-01-17 15:59:21.096452 [INFO    ] [clock        ]  Clk: Using Oregano Systems; syn1588(R) PCIe NIC Revision 2.1; 8C:A5:A1:00:0D:3C
2022-01-17 15:59:21.096466 [INFO    ] [clock        ]    with ClockId 00:00:00:00:00:00:00:00
2022-01-17 15:59:21.096495 [INFO    ] [clock        ]  Clk: Resetting servos
2022-01-17 15:59:21.096502 [INFO    ] [clock        ]  Clk: Resetting filters
2022-01-17 15:59:21.096508 [INFO    ] [clock        ]  Spike M2S: Init with ival 0, buffer size 16
2022-01-17 15:59:21.096514 [INFO    ] [clock        ]  Spike Path: Init with ival 0, buffer size 16
2022-01-17 15:59:21.096542 [INFO    ] [clock        ]  Clk: Using Oregano Systems; syn1588(R) PCIe NIC Revision 2.1; 8C:A5:A1:00:0D:3B
2022-01-17 15:59:21.096549 [INFO    ] [clock        ]    with ClockId 00:00:00:00:00:00:00:00
2022-01-17 15:59:21.096565 [INFO    ] [clock        ]  Clk: Resetting servos
```

```
2022-01-17 15:59:21.096572 [INFO    ] [clock        ]  Clk: Resetting filters
2022-01-17 15:59:21.096577 [INFO    ] [clock        ]  Spike M2S: Init with ival 0, buffer size 16
2022-01-17 15:59:21.096612 [INFO    ] [clock        ]  Spike Path: Init with ival 0, buffer size 16
2022-01-17 15:59:21.096630 [INFO    ] [clock        ]  Clk: Using Linux System Clock;;
2022-01-17 15:59:21.096638 [INFO    ] [clock        ]    with ClockId 00:00:00:00:00:00:00:00
2022-01-17 15:59:21.096653 [INFO    ] [clock        ]  Clk: Resetting servos
2022-01-17 15:59:21.096660 [INFO    ] [clock        ]  Clk: Resetting filters
2022-01-17 15:59:21.096665 [INFO    ] [clock        ]  Spike M2S: Init with ival 0, buffer size 16
2022-01-17 15:59:21.096670 [INFO    ] [clock        ]  Spike Path: Init with ival 0, buffer size 16
…
…
2022-01-17 15:59:36.000803 [INFO    ] [ClSyncEngine ]  ptp stack synced with boundary 18(configured: 13), Clock 0 selected = 0
2022-01-17 15:59:36.000856 [INFO    ] [ClSyncEngine ]  ptp stack synced with boundary 17(configured: 13), Clock 1 selected = 1
2022-01-17 15:59:36.000954 [WARNING ] [ClSyncEngine ]  syn1588 0:  1642420918.481401516 2022-01-17 15:59:36.000970 [WARNING ]
[ClSyncEngine ]
2022-01-17 15:59:36.000980 [WARNING ] [ClSyncEngine ]  syn1588 1:  1642435213.000752471 2022-01-17 15:59:36.000990 [WARNING ]
[ClSyncEngine ]   (Origin)
2022-01-17 15:59:36.000998 [WARNING ] [ClSyncEngine ]  Local:      1642435213.000751127 2022-01-17 15:59:36.001007 [WARNING ]
[ClSyncEngine ]
2022-01-17 15:59:36.001015 [INFO    ] [ClSyncEngine ]  current UTC offset: 37 sec
2022-01-17 15:59:36.001024 [WARNING ] [ClSyncEngine ]
2022-01-17 15:59:36.001043 [INFO    ] [ClSyncEngine ]  sys delta: c:168, m:180.000000, stdev:18.330303 2022-01-17 15:59:36.001066 [INFO
] [ClSyncEngine ]  hw delta: c:120, m:112.000000, stdev:11.313708
2022-01-17 15:59:36.001096 [INFO    ] [ClSyncEngine ]  not adjusting clock syn1588 0 in slave state..
2022-01-17 15:59:36.001110 [WARNING ] [ClSyncEngine ]  Adjusting Local:
2022-01-17 15:59:36.001120 [INFO    ] [clock        ]  Update M2S-Delay -1295 ns
2022-01-17 15:59:36.001134 [INFO    ] [clock        ]  Adjusting clock at -1295.00 ns offset
2022-01-17 15:59:36.001149 [INFO    ] [clock        ]  changed boundary to     2500 ns(13)
2022-01-17 15:59:36.001162 [INFO    ] [clock        ]  Spike M2S: mean: -1194, median: -1193, variance: 3861
2022-01-17 15:59:36.001175 [INFO    ] [clock        ]  Spike Path: mean: 0, median: 0, variance: 0
```

ISync detects two ®syn1588 hardware units, “syn1588 0” and “syn1588 1”, it uses “syn1588 1” = “Port B” as clock source, which has been qualified by redSync.

As can be seen from the last line, after a few seconds ISync starts to synchronize the system clock to the “Origin”.

At this point the system clock is offset by -1295 ns to the source clock. This offset will differ form run to run and can be up to 100 ms. If you are testing this scenario keep this in mind that a high initial offset will require some time (up to three minutes) for ISync to be fully compensated.

Measurement for Scenario B

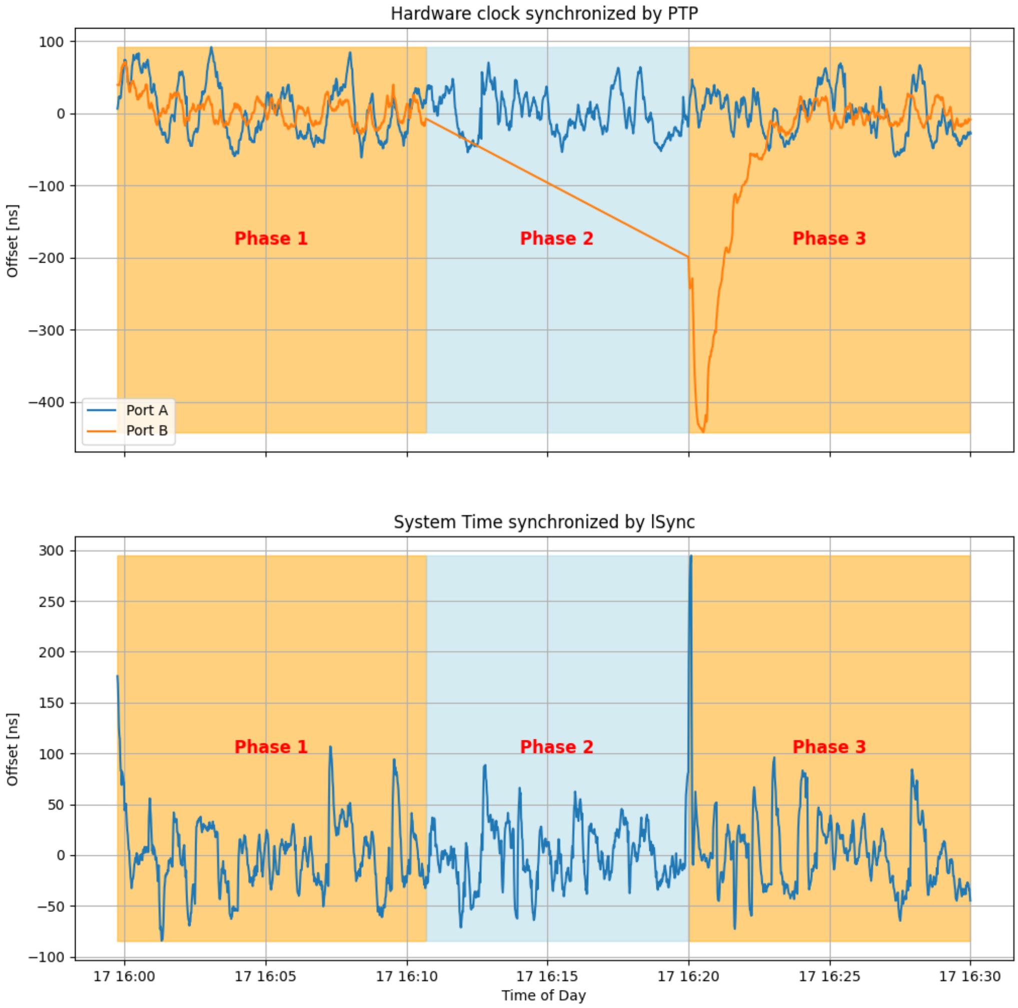

Figure 3: Scenario B, full timeline

The full timeline for this example for Scenario B is depicted in Figure 3.

1. lSync starts to synchronize the system clock to the syn1588 NIC Hardware clock controlled by the PTP Stack for Port B in Phase 1.
2. During Phase 2 this PTP Stack looses its connection to the PTP Grandmaster. This is detected by redSync and Port A (which still is connected to its PTP Grandmaster) is chosen. As both PTP Stacks have been continuously synchronizing to their respective PTP Grandmasters the switchover from Phase 1 to Phase 2 is seamless.
3. At the end of Phase 2, the PTP Stack on Port B reconnects to its PTP Grandmaster and starts synchronization. In this example lSync is configured to use an additional qualifier, the current accuracy, to decide if a selected PTP Port is used as new time source. This qualifier is configured to a higher value than usual, in this case a PTP Stack already qualifies, if it is below 2500 ns. For Port B this is

the case right after reconnecting with its PTP Grandmaster, as can be seen in the upper subfigure representing this offset. Therefore, the switchover from Port A to Port B from Phase 2 to Phase 3 is not as seamlessly as from Phase 1 to Phase 2, which is to be expected. ISync allows narrower configuration of this qualifier which allows good optimization for different applications.

4. As can be seen over the course of the runtime, the System time stays within +/- 100 ns of the PTP Grandmaster time even during switchover between the two redundant ports. Only excemption is the switchover from Phase 2 to 3, which can be easily optimized as described above.

Table 2: Scenario B, Phase 1

	ISync	PTP_PortA	PTP_PortB
min	-84,36	-61,25	-29,05
std	35,12166	37,32018	18,48091
max	168	91,87	71,35
mean	0,30662	5,457003	5,02284
count	645	654	655

Table 3: Scenario B, Phase 2

	ISync	PTP_PortA	PTP_PortB
min	-71,3	-54,25	NaN
std	28,94446	28,91877	NaN
max	88,61	70,32	NaN
mean	0,58435	-0,23334	NaN
count	554	557	0

Table 4: Scenario B, Phase 3

	ISync	PTP_PortA	PTP_PortB
min	-72,65	-60,23	-442,56
std	42,47698	30,24652	115,2156
max	294,62	69,3	27,33
mean	4,105094	-0,37858	-60,0314
count	583	598	599

Version history

Version 1.14.1 – January 2022

Added example measurement for Scenario B

Version 1.14.0 – January 2022

Update for release v1.14

Version 1.13.2 - October 2021
Editorial Review

Version 1.13.1 - July 2021
Added description for Scenario B, changed version number scheme

Version 1.0 - October 2020
Initial version (released with syn1588® Software Suite v1.12)

Version 1.17.0 - October 2023

Oregano Systems
A Meinberg Company

Franzosengraben 8
A-1030 Vienna
Austria
http://oregano.at
contact@oregano.at

Oregano Systems does not assume any liability arising out of the application or use of any product described or shown herein nor does it convey any license under its patents, copyrights, or any rights of others.

Licenses or any other rights such as, but not limited to, patents, utility models, trademarks or tradenames, are neither granted nor conveyed by this document, nor does this document constitute any obligation of the disclosing party to grant or convey such rights to the receiving party.

Oregano Systems reserves the right to make changes, at any time without notice, in order to improve reliability, function or design. Oregano Systems will not assume responsibility for the use of any circuitry described herein.