NanoCom Link IS

User Manual

Guidelines for Usage of NanoCom Link IS Radio Solution

MAN 1054052

October 30, 2023

Revision 1.0

NanoCom Link IS User Manual
Guidelines for Usage of NanoCom Link IS Radio Solution

Document reference: MAN 1054052
Source reference: doc-nanocom-link-is-user-manual
Date: October 30, 2023
Revision: 1.0

GomSpace A/S
Langagervej 6, 9220 Aalborg East
Denmark
Phone: +45 71 741 741
www.gomspace.com

Contents

List of Figures

List of Tables

List of Abbreviations

CAN Controller Area Network.

CSP Cubesat Space Protocol.

ESD electrostatic discharge.

GOSH GomSpace Shell.

ICD interface control document.

IP Internet Protocol.

IPA isopropyl alcohol.

IPv4 Internet Protocol version 4.

ISL inter-satellite link.

MTU maximum transmission unit.

OBC on-board computer.

PC personal computer.

PCBA printed circuit board assembly.

PPP Point-to-Point Protocol.

PPS pulse-per-second.

RF radio frequency.

RX receive.

SDR software-defined radio.

TCP/IP Transmission Control Protocol/Internet Protocol.

TDD time-division duplex.

TTY teletypewriter.

TX transmit.

USB universal serial bus.

Changelog

Version	Change
1.0	Initial revision

Table 1: Changelog

1 Introduction

1.1 Purpose

This document contains guidelines for usage of the GomSpace NanoCom Link IS radio solution.

1.2 Scope

This document is applicable within the scope of using the NanoCom Link IS radio communications solution. The following products are covered in this document.

- NanoCom Link IS1 for S-band inter-satellite link (ISL) communications between two spacecrafts.

Note that technical specifications such as performance characteristics are not included in this document as it is focused on usage. Refer to the datasheet for technical specifications [1].

1.3 Structure

The document is structured as follows:

- Section 2 provides a hands-on description of how to use the product.
- Section 3 provides a system overview.
- Section 4 describes configuration of the product.

1.4 Related Documents

The user manual is complemented by the NanoCom Link IS interface control documents (ICDs) [2][3].

2 Getting Started

This section describes the necessary steps for getting started with the NanoCom Link IS1 product.

2.1 Unpacking and Handling Precautions

The NanoCom Link products and their components are electrostatic discharge (ESD) sensitive devices. Proper precautions must be observed when handling the devices. Use an ESD mat and a wrist strap as a minimum. Wear gloves to avoid fingerprints on the anodized aluminium parts, as these are particularly difficult to rinse off. If any cleaning of the parts are required prior to flight, use ESD safe cleaning methods with a neutral, non-reactive, isopropyl alcohol (IPA) solvent.

2.2 Included Components

The NanoCom Link IS1 product consist of the following components.

- NanoCom Link software-defined radio (SDR) unit
- NanoCom ANT2150 antenna

2.2.1 SDR unit

The SDR unit is shown on Figure 2.1.

Figure 2.1: NanoCom Link SDR unit

It consists of a main board where a processing module is installed alongside one or more transceiver modules. The processing module is labelled “Z7000” while the transceiver modules are labelled “TR600”. The main board can hold up to three transceiver modules in so-called “slots”.

The SDR unit is delivered with a cable kit which includes

- Power cable
- Universal serial bus (USB) to serial cable
- Debug breakout printed circuit board assembly (PCBA) with debug cable

The usage of the slots is described on Table 2.1.

Transceiver Slot	Usage	Physical Location
A	Communication with NanoCom ANT2150	Next to the processing module on the same side of the main board.
B	Not used	Directly opposite to slot A on the other side of the main board.
C	Not used	Next to slot B on the same side of the main board.

Table 2.1: NanoCom Link transceiver slot usage and location

2.2.2 NanoCom ANT2150

The NanoCom ANT2150 is shown on Figure 2.2 and on Figure 2.3 with connectors highlighted.

Figure 2.2: NanoCom ANT2150 antenna

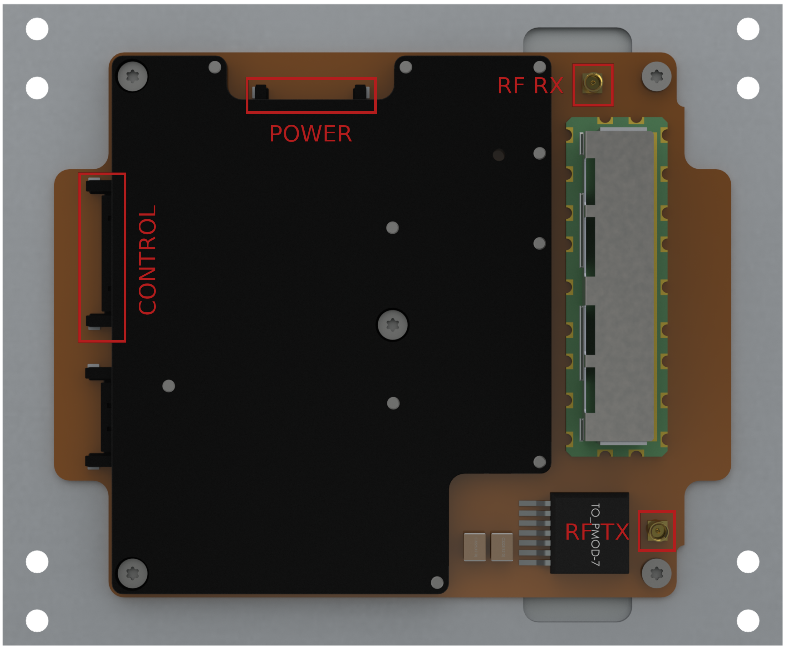

Figure 2.3: NanoCom ANT2150 bottom-view with connectors highlighted

It is delivered with a cable kit which includes

- Power cable
- Controller Area Network (CAN) control cable
- Two radio frequency (RF) cables

The relevant connectors are described on Table 2.2.

Connector	Description
RF receive (RX)	Connected using one RF cable
RF transmit (TX)	Connected using one RF cable
Power	Connected using power cable
Control	Connected using CAN control cable

Table 2.2: NanoCom ANT2150 connector overview

2.3 Connecting the Components

The NanoCom Link products must have an RF connection and a CAN connection between the SDR unit and the RF front-end.

2.3.1 NanoCom ANT2150

Use the SDR transceiver module in slot A for connection between the SDR unit and the NanoCom ANT2150. Use the supplied RF cable to connect the transceiver port labelled "TX1" to the NanoCom ANT2150 J300 connector. Use the other supplied RF cable to connect the transceiver port labelled "RX1" to the NanoCom ANT2150 J102 connector. Use the supplied CAN cable to connect the transceiver connector labelled "AFE CONTROL" to the NanoCom ANT2150 J401 connector.

2.4 Powering the Components

2.4.1 SDR unit

The SDR unit requires a single power supply. For stand-alone testing, use the supplied power cable and a lab power supply with current limiting capabilities. Connect the power cable to the connector labelled "PWR1" on the SDR and the other end to the power supply channel. Configure the power supply channel for e.g. $12\,\mathrm{V}$ with $1\,\mathrm{A}$ current limit. Refer to the NanoCom Link S, X, SX Datasheet[1] for the supported supply voltage range.

The SDR is now ready to be powered on.

2.4.2 NanoCom ANT2150 for S and SX products

The NanoCom ANT2150 requires an $8\,\mathrm{V}$–$18\,\mathrm{V}$ power supply. For stand-alone testing, use a similar setup as the SDR. That is, use the supplied power cable and a lab power supply with current limiting capabilities. Connect the power cable to the J400 connector on the NanoCom ANT2150 and the other end to the power supply channels. Configure the power supply channels for e.g. $12\,\mathrm{V}$ with $1\,\mathrm{A}$ current limit.

The NanoCom ANT2150 is now ready to be powered on.

2.5 Communicating over Debug Serial Interface

The CAN interface is the primary interface for control and telemetry [2]. The serial interface is only used for lab debugging purposes and getting started.

Connect the SDR breakout PCBA with the SDR unit using the supplied debug cable. Use the supplied USB to serial cable to connect your personal computer (PC) to the breakout PCBA. Open the teletypewriter (TTY) device in a serial communication program, e.g., minicom or tio [4, 5]. Configure the serial communication program for $115\,200\,\mathrm{bit/s}$, 8 databits, 1 stopbit and no parity. On Listing 2.1 an example is shown using *tio* [5] from a Linux shell. Note there should be no output at this point, as the board is still powered down.

Listing 2.1: Example of SDR serial communication setup with *tio*

```
user@HostPC:~$ tio -b 115200 /dev/ttyUSB0
[tio 11:15:57] tio v1.20
[tio 11:15:57] Press ctrl-t q to quit
[tio 11:15:57] Connected
```

2.6 Bootstrapping the Components

2.6.1 SDR unit

Bootstrap the SDR by enabling the power supply channel and wait for output to appear on the serial connection. The bootstrapping process includes a u-boot console which automatically continues to a Linux startup process. When the bootstrapping process is done a Linux prompt is available on the serial interface. Note that additional time is necessary for all software components to start before the SDR is operational.

Listing 2.2 shows the output on the serial interface that appears when a Linux prompt is available.

Listing 2.2: Output on serial interface when Linux prompt is reached

```
GomSpace Linux 4.1.4 nanomind-z7030-nv3 ttyPS1

nanomind-z7030-nv3 login: root (automatic login)


   ____  ___  __  __
  / ___|/ _ \|  \/  |___ _ __   __ _  ___ ___
 | |  _| | | | |\/| / __| '_ \ / _` |/ __/ _ \
 | |_| | |_| | |  | \__ \ |_) | (_| | (_|  __/
  \____|\___/|_|  |_|___/ .__/ \__,_|\___\___|
                        |_|

GomSpace NanoCom SGL Image

root@nanomind-z7030-nv3:~#
```

PPS Signal

The time-division duplex (TDD) timing used by the NanoCom Link IS1 product is based on a pulse-per-second (PPS) signal. A PPS *must* be provided to the SDR units. Refer to NanoCom Link S, X, SX Datasheet [1] for details on the physical and electrical interface of the PPS signal.

To verify the PPS signal is connected and working, run the following command on the SDR unit Linux terminal:

```
$ pps_test /dev/pps0
```

The comand should print a line every second, with information like the following:

```
source 0 - assert 1667896609.583869308, sequence: 647 - clear  0.000000000, sequence: 0
source 0 - assert 1667896610.583864731, sequence: 648 - clear  0.000000000, sequence: 0
source 0 - assert 1667896611.583858936, sequence: 649 - clear  0.000000000, sequence: 0
source 0 - assert 1667896612.583855031, sequence: 650 - clear  0.000000000, sequence: 0
```

2.6.2 NanoCom ANT2150

Bootstrap the NanoCom ANT2150 by simply enabling the power supply channels associated with it.

2.7 Enabling RF Transmission

The following section uses the SDR debug serial interface for commanding the unit. In production the CAN interface should be used as described in [2].

To enable RF transmission access the Linux prompt on the SDR and run

```
gosh localhost 5005
```

This opens the GomSpace Shell (GOSH) for the link control software. The NanoCom Link products use the GomSpace parameter system for control and telemetry. This is described in detail in [2].

2.7.1 Enabling transmission

Access the parameter table for configuration using the `param select` GOSH command. Set the `enable` parameter to `true` using the `param set` GOSH command. This procedure is illustrated on Listing 2.3.

Listing 2.3: Example of enabling transmission over SDR debug serial interface

```
root@nanomind-z7030-nv3:~# gosh localhost 5005
connected to 127.0.0.1:5005
LINK # param select is1_base_cfg
LINK # param set enable true
```

Note that the `tdd_rx_first` parameter must be configured opposite on the two units. The default value is `false`. On one of the units, set it to `true` using the `param set` GOSH command.

Listing 2.4: Example of configuring TDD timing

```
LINK # param select is1_base_cfg
LINK # param set tdd_rx_first true
```

The transmission is now active using the default configuration. The configuration can be seen using the `param list` GOSH command as illustrated on Listing 2.5.

Listing 2.5: Example of listing configuration over SDR debug serial interface

```
LINK # param list is1_base_cfg
Table is1_base_cfg (1):
  0x0000 freq             DBL 2200.000000
  0x0008 symrate          FLT 66.000000
  0x000C rs_depth         U8  3
  0x000D tx_pwr_lvl       U8  0
  0x000E tx_inhibit       BL  false
  0x000F enable           BL  true
  0x0010 tdd_rx_first     BL  false
```

A detailed list of configuration parameters and descriptions is available in [2]. To access telemetry use the `param list` GOSH command as illustrated on Listing 2.6.

Listing 2.6: Example of listing telemetry over SDR debug serial interface

```
LINK # param list is1_base_telem
Table ccsds_base_telem (4):
  0x0000 bb_running       BL  true
  0x0001 afe_running      BL  true
  0x0002 afe_steady       BL  true
  0x0003 afe_temp_off     BL  false
  0x0004 afe_temp_pa      FLT 27.800001
  0x0008 afe_temp_mcu     FLT 22.700001
  0x000C afe_uptime       U32 68
  0x0010 afe_power_in     FLT -12.426064
  0x0014 rx_gain          FLT 61.000000
  0x0018 tx_gain          FLT -14.250000
  0x001C rx_lock          BL  false
  0x001D rx_sig_pres      BL  false
  0x001E tr_running       BL  true
  0x001F rs_depth         U8  3
```

A detailed list of telemetry parameters and descriptions is available in [2].

3 System Overview

The NanoCom Link IS1 product is depicted on Figure 3.1 as a general block-diagram.

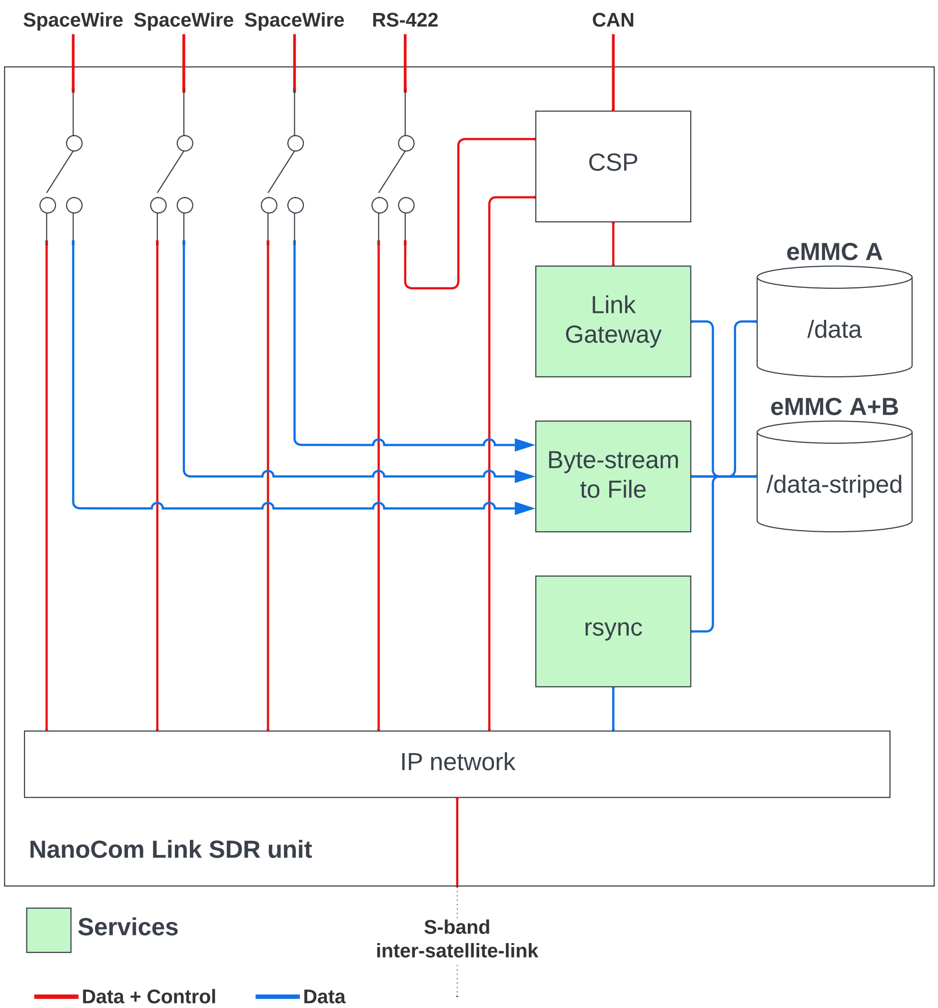

Figure 3.1: NanoCom Link System Overview

3.1 Data Plane

3.1.1 Data Exchange with Spacecraft Bus

The SpaceWire and RS-422 external interfaces are primarily used for data. The details of these interfaces are described in [3]. In summary, these interfaces support data exchange as Transmission Control Protocol/Internet Protocol (TCP/IP). Additionally, the SpaceWire interfaces supports data exchange as a byte-stream to file. On the SDR unit, TCP/IP is realized as a Linux Internet Protocol (IP) TUN interface for ease of integration with third-party Linux tools.

3.1.2 Data Exchange via S-band Communication

The interfaces mentioned in Section 3.1.1 can be used for direct, instantaneous transmission of data or, alternatively, to store data for future transmission.

When data is stored for future transmission it is put into files and stored on the SDR file system. These files can e.g. be transferred with the Linux `rsync` application. These use-cases are described in more detail in the following sections.

Direct transmission is enabled by another IP TUN interface named `rf0`. The IP packets sent on this interface are transmitted over the S-band RF link. The data-plane is thus composed of multiple interfaces that can all be represented by IP TUN interfaces and interconnected using standard Linux IP routing capabilities.

3.2 Control Plane

The CAN and RS-422 interfaces are the primary control interfaces, although other interface may also be used. This is described in detail in [6]. In summary, the NanoCom Link products use Cubesat Space Protocol (CSP) and the GomSpace parameter system. Consequently, the payloads and/or on-board computer (OBC) must support CSP to control and retrieve telemetry from a NanoCom Link product.

3.3 Storage

Two storage locations are available for data storage. The details of the storage is shown in Table 3.1.

Mount Point	Capacity	Physical Media	Read/Write Performance
`/data`	10 GB	eMMC A	low
`/data-striped`	115 GB	eMMC A + eMMC B	high

Table 3.1: Data storage properties

The `/data-striped` storage has the highest performance and capacity. The drawback compared to `/data` is that the data is distributed across two separate physical devices. In case either one of the physical mediums malfunction, the `/data-striped` storage in unavailable.

4 Configuring the Radio Parameters

It is recommended to configure and save the NanoCom Link IS radio parameters prior to flight. Follow the steps in this section to configure the radio using the SDR debug serial interface.

The following configuration is used as an example.

- $2200\,\mathrm{MHz}$ frequency
- $32\,\mathrm{dBm}$ TX power

Access the SDR unit Linux terminal using the serial debug interface as described in Section 2. Open the GOSH for the link control application:

```
gosh localhost 5005
```

While in the link control application GOSH access the parameter table for basic IS1 radio configuration:

```
param select is1_base_cfg
```

Set the parameters:

```
param set freq 2200
param set tx_pwr_lvl 12
```

Save the changes:

```
param save is1_base_cfg
```

The changes are now stored in persistent storage. The parameters can be changed any time during the mission including on-orbit. During operation, however, the dedicated control interfaces should be used as described in [2].

5 IP Networking

The NanoCom Link IS1 product have multiple data interfaces that support IP based communication. This section describes how to leverage this to build an end-to-end IP network that interconnects all nodes.

Figure 5.1 shows an overview of the IP network.

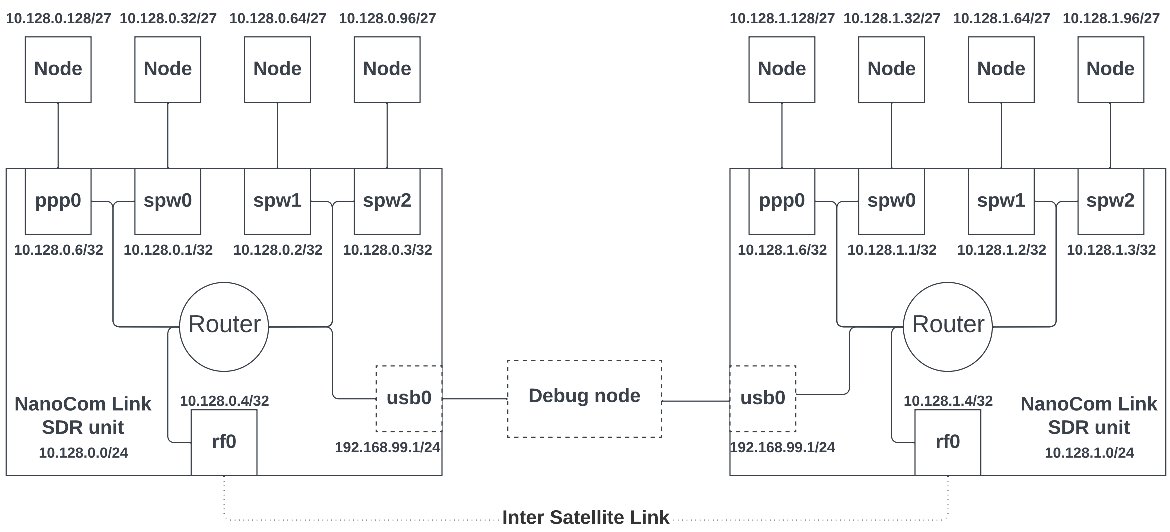

Figure 5.1: IP network of the NanoCom Link system

5.1 Interfaces with IP Capabilities

Any interface in the NanoCom Link products that supports IP is accessible through a Linux IP TUN network interface on the SDR unit. This means any Linux utility that uses an IP connection such as `rsync` [7, 8] and `ssh` [9, 10] can operate over any given interface seamlessly.

Interface Name	Description
spw0	Interface using SpaceWire on PL DIFF IO-1 connector
spw1	Interface using SpaceWire on PL DIFF IO-2 connector
spw2	Interface using SpaceWire on PL DIFF IO-3 connector
ppp0	Interface using RS-422 on COM1/COM2 connector
rf0	Ingress and egress IP data to/from the radio link

Table 5.1: IP interfaces available on NanoCom Link

Table 5.1 lists the IP TUN interfaces available on NanoCom Link products. The SpaceWire interfaces automatically detect if there is a working connection to the other end, and set the TUN network interface up/down accordingly. The `rf0` interface can operate in both one-way and two-way links. With one-way links, the interface has no way of knowing if the link is established on the receiver in the other end. To accommodate one-way links, the `rf0` interface is always reported as up.

5.2 Addresses and Routing in Space Segment

All interfaces are configured with a default Internet Protocol version 4 (IPv4) address and routing. The interfaces are configured to act like bridges by default thus forwarding any IPv4 packets with a destination address in their subnet. The default addresses and routes are listed on Table 5.2.

Interface Name	IP address	Routing	MTU
spw0	10.128.0.1/32	10.128.0.32/27	9000
spw1	10.128.0.2/32	10.128.0.64/27	9000
spw2	10.128.0.3/32	10.128.0.96/27	9000
ppp0	10.128.0.6/32	10.128.0.128/27	1500
rf0	10.128.0.4/32	10.128.1.0/24	9000

Table 5.2: IP configuration

Consider, as an example, the `spw0` interface which is configured with an IPv4 address of `10.128.0.1`. This interface is, by default, configured to route any IPv4 packet in the `10.128.0.32/27` subnet. This means any IPv4 packet with a destination in the range `10.128.0.32` to `10.128.0.63` is forwarded over this interface. Any IPv4 enabled payload, connected to the SpaceWire interface via the PL DIFF IO-1 connector, should be configured with an IPv4 address in the `10.128.0.32/27` network.

Note that the `rf0` interface is configured to route any IPv4 packet with a destination in the `10.128.1.0/24` subnet. This subnet denotes another spacecraft. Each spacecraft is assigned a `/24` subnet in the `10.128.0.0/16` network.

5.3 Configuring Addresses and Routing

The IPv4 addresses and routing is configured through configuration files on the root file system. Table 5.3 shows the location of the file used to configure a certain interface.

Interface Name	Configuration File
spw0	`/etc/systemd/network/spw0.network`
spw1	`/etc/systemd/network/spw1.network`
spw2	`/etc/systemd/network/spw2.network`
ppp0	`/etc/ppp/peers/psrs422srv`
rf0	`/etc/systemd/network/rf0.network`
usb0	`/etc/systemd/network/80-wired.network`

Table 5.3: IP configuration file locations

5.3.1 Systemd Config Files

The IP configuration files located under `/etc/systemd/network/` looks like Listing 5.1.

Listing 5.1: Example of Systemd IP interface configuration file

```
[Match]
Name=spw0

[Network]
Address=10.128.0.1/32

[Route]
Destination=10.128.0.32/27
```

The `Address` attribute configures the IPv4 address and subnet. An interface can have multiple addresses.

The `Destination` attribute configures a route via the given interface, given as IPv4 address and subnet. An interface can have multiple routes.

Any changes to the configuration is applied on a reboot, or using

```
networkctl reload
```

5.3.2 PPP Config Files

The configuration file for Point-to-Point Protocol (PPP) interfaces in `/etc/ppp/peers/` looks like Listing 5.2.

Listing 5.2: Example of PPP interface configuration file

```
10.128.0.6:10.128.0.129
/dev/ttyPS0
```

In the first line, two IPv4 addresses appear, separated by a ':'. The first address, is the address given to the local interface, when someone connects. The second address, is the address given to the PPP client connecting.

Any changes to the configuration requires a reboot.

5.3.3 Editing the Files

Change the file according to your needs. Using the serial debug interface, the files can be edited using e.g. *nano* [11] or *vim* [12].

In order to change the configuration files, the root file system must be writeable. You can make the filesystem writeable using:

```
mount -o remount,rw /
```

After the files are updated, the filesystem can be mount read-only again using:

```
mount -o remount,ro /
```

6 Storing SpaceWire Data

The following example describes how to store data from the SpaceWire data link layer directly to the NanoCom Link SDR unit. The example is depicted on Figure 6.1.

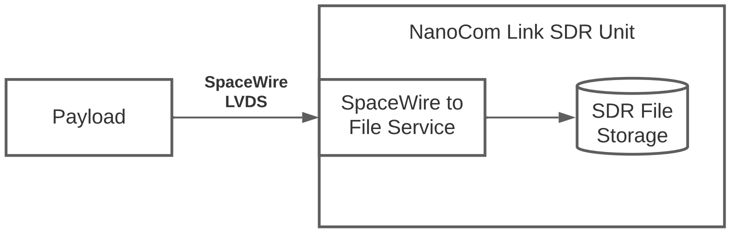

Figure 6.1: Overview of SpaceWire data storage example

The payload data in the data link layer frames is stored as files on the SDR unit filesystem with configurable name, size, and location. The frames are never split between files but the boundaries between frames are not preserved by the NanoCom Link SDR unit. That is, the payload must provide the necessary means for recovering frames boundaries if these are needed for decoding. This can be done by adding additional protocol layers on the top of the SpaceWire data link layer.

6.1 Payload Prerequisites

The payload must be compatible with the SpaceWire standard [13] for the physical and data link layers. There are no requirements to the format of the data link layer frames except the frame size which must not exceed $65\,535\,\mathrm{B}$. For more details regarding the SpaceWire interface protocol stack refer to [14].

6.2 SpaceWire Configuration

The NanoCom Link SpaceWire interface must be configured to operate in *file* mode. To configure the mode, access the SpaceWire parameter table and change the *mode* parameter. In the following, this is illustrated using the debug serial interface as shown in Section 2.

On the SDR unit, open the GOSH for the link control application.

```
gosh localhost 5005
```

Access the SpaceWire configuration parameter table

```
param select 3
```

Set the mode for the SpaceWire interface to *file*. In the following SpaceWire interface 0 is assumed. Use `s1_mode` or `s2_mode` instead of `s0_mode` depending on the target SpaceWire interface.

```
param set s0_mode file
```

The SpaceWire interface is now operating in file mode. See Listing 6.1 for an example of how to configure file names, sizes, and output directory. This example configures the SpaceWire data link layer frames to be stored in `/data-striped/spw-data/`. The files are named as `spw0-data-<date>-<counter>.bin` where `<date>` is e.g. 24-12-2022 for the 24th of December, 2022. The `<counter>` is a 9-digit zero-padded number running from 000000001 and upwards. This number is preserved across all configurations and multiple runs. If no configuration values are provided the values in this example are used by default.

Listing 6.1: Example of SpaceWire data file configuration

```
param set s0_file_pre spw0-data-
param set s0_file_size 5
param set s0_file_ext bin
param set s0_file_out /data-striped/spw-data/
```

When SpaceWire data is received, it is stored in a temporary file until that file reaches the configured size of e.g. $5\,\mathrm{MB}$. At this point the temporary file is renamed according to the configuration and naming scheme. In case the SpaceWire interface is idle for a period of time the temporary file is renamed regardless of its size to avoid data being stuck in the temporary file. The period of time to wait while idle is also configurable. Refer to [6] for details regarding all configuration parameters.

6.2.1 Storage Full Protection

The default configuration protects the file storage from becoming full. If the usage of the file storage goes above a limit, configured by the *sX_file_du_t* parameters, data is automatically discarded.

Only files matching any of the configured data file configurations (see Listing 6.1) are considered for removal.

Whether newest or oldest data files are removed is configurable through the *sX_file_rm_n* parameters.

The default configuration is shown in Listing 6.2.

Listing 6.2: Default configuration for disk storage full protection

```
param set s0_file_du_t 0.9
param set s0_file_rm_n false
```

6.2.2 Persisting Configuration

To save the configuration to persistent storage run the following.

```
param save 3
```

This ensure the SpaceWire interface is configured correctly when the SDR unit is powered on.

6.3 SpaceWire Telemetry

The SpaceWire interfaces offer various telemetry parameters. To see these, access parameter table 9 in the link control application. In the following, this is illustrated using the debug serial interface as shown in Section 2.

On the SDR unit, open the GOSH for the link control application.

```
gosh localhost 5005
```

Access the SpaceWire telemetry parameter table

```
param select 9
```

To see the current mode for e.g. SpaceWire interface 0 run

```
param get s0_mode
```

To see if SpaceWire interface 0 has an active connection run

```
param get s0_up
```

For a complete list of all telemetry parameters in the table run

```
param list
```

Refer to [2]for details regarding all telemetry parameters.

6.4 Operation

When the SpaceWire file storage is needed, turn on the NanoCom Link SDR unit. When the SDR unit is booted it is ready to accept SpaceWire data link layer frames. Turn on the payload and keep the SDR unit running while the payload is producing SpaceWire data. Verify by observing the telemetry parameters as described in Section 6.3.

When the payload operation is finished the SDR may be turned off.

7 References

[1] **GomSpace**
DS 1046020
NanoCom Link S, X, SX Datasheet
Cited on pages 2, 5, 6

[2] **GomSpace**
ICD
NanoCom Link IS Control Interface
Cited on pages 2, 6–8, 11, 17

[3] **GomSpace**
ICD
NanoCom Link IS Data Interface
Cited on pages 2, 10

[4] **minicom**
Software tool
Available at `https://github.com/Distrotech/minicom`
Cited on page 6

[5] **tio**
Software tool
Available at `https://github.com/tio/tio`
Cited on page 6

[6] **GomSpace**
ICD 1045207
NanoCom Link S, X, SX Control Interface
Cited on pages 10, 16

[7] **rsync(1)**
Unix manual page
Available at `https://man7.org/linux/man-pages/man1/rsync.1.html`
Cited on page 12

[8] **rsync**
Software tool
Available at `https://rsync.samba.org/`
Cited on page 12

[9] **ssh(1)**
Unix manual page
Available at `https://man7.org/linux/man-pages/man1/ssh.1.html`
Cited on page 12

[10] **OpenSSH**
Software tool
Available at `https://www.openssh.com/`
Cited on page 12

[11] **nano**
Software tool
Available at `https://www.nano-editor.org/`
Cited on page 14

[12] **vim**
Software tool
Available at `https://www.vim.org/`
Cited on page 14

[13] **European Cooperation for Space Standardization (ECSS)**
Standard ECSS-E-ST-50-12C
SpaceWire - Links, nodes, routers and networks
May 2019, Revision 1
Available at `https://tinyurl.com/yx8k4pzt`
Cited on page 15.

[14] **GomSpace**
ICD 1045208
NanoCom Link S, X, SX Data Interface
Cited on page 15