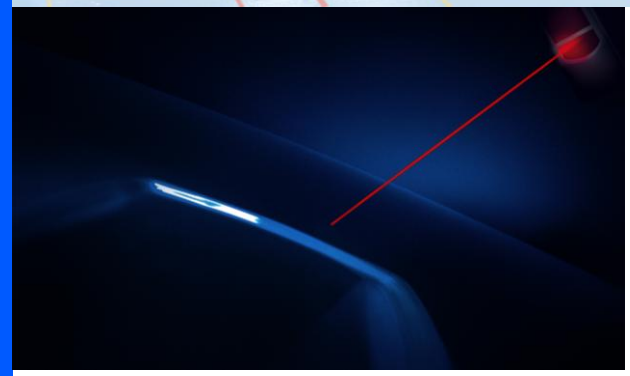
HOWTO: LIDAR SENSOR CONFIGURATIONS WITH LIDAR-LOC

Material Flow Solutions - Mobile Robots
AE Team

March 2026
Version 1.5

SICK Sensor Intelligence



AGENDA

1. Overview
   - → Purpose of this guide
   - → Example cases table of contents
   - → Safety Designer
   - → Safety Designer – Data output
   - → LiDAR-LOC SOPASAir web UI
2. Types of data transmission
   - → Unicasting
   - → Broadcasting
   - → Multicasting
3. Data transmission examples
   - → Unicasting
   - → Broadcasting
   - → Multicasting
   - → TCP Scanner
4. ROS driver configuration

01

OVERVIEW

OVERVIEW

Purpose of this guide

To show the user the different configurations that can be obtained using SICK scanners, LiDAR-LOC and customer's applications.

Examples shown using different types of data transmission:

→ Unicasting

→ Broadcasting

→ Multicasting

With these examples, the user will have the necessary tools to configure LiDAR-LOC and the scanners to meet the requirements.

Note: All supported SICK scanners can be configured with all the different types of data transmission unless stated otherwise.

nanoScan3

picoScan150

microScan3

TiM571

OVERVIEW

Example cases table of contents

Communication	Example #	Used sensor	LiDAR-LOC running on	2nd LiDAR application in IPC, e.g., ROS/C++ driver
Unicasting	1	1x microScan3 (one data channel)	SIM1200	no
	2		IPC(1)	
	3	2x microScan3 (one data channel)	SIM1200	
	4		IPC(1)	
	5	1x nanoScan3 I/O(2) (two data channel)	SIM1200	yes
	6		IPC(1)	
	7	2x nanoScan3 I/O(2) (two data channel)	SIM1200	
	8		IPC(1)	
Broadcasting(3)	9	1x microScan3	SIM1200	
	10	2x microScan3	SIM1200	
Multicasting	11	1x picoScan150	IPC(1)(3)	
	12		SIM1200	
	13	2x picoScan150	IPC(1)(3)	
	14	1x nanoScan3	IPC(1)(3)	
TCP/IP	15	1x TiM571	SIM1200	

(1) In our examples, the **IPC always runs the main AGV application**, e.g., navigation. LiDAR-LOC runs either on IPC or SICK device, e.g., SIM1200

(2) Only nanoScan3 I/O variants have a 2nd data output channel

(3) Unicast and broadcast cannot be used for 2nd receiver of LiDAR data on the same IPC. Use multicast instead (Examples 11, 13 and 14)

OVERVIEW

Safety Designer

Connection of sensor

With the correct configuration, the sensor should be visible in Safety Designer showing connection state.

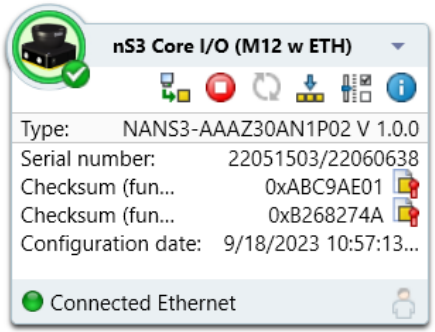
Troubleshooting for missing sensor data:

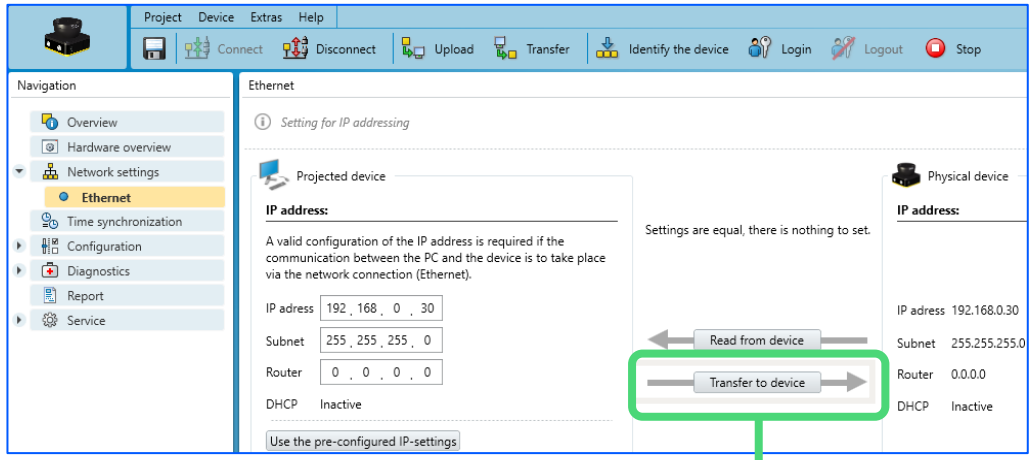
→ Devices need to be in the same subnet

→ Safety Desginer connection to device closed for data in LiDAR-LOC

→ For Windows: Set network category to “private”.

→ **Note**: For troubleshooting, please also check Operating Instructions chapter troubleshooting.





After changing the IP and settings, don’t forget to transfer to device.

OVERVIEW

Safety Designer – Data output

Data output configuration

→ **Deactivated**: No data output (In normal use cases it's fine to set as "Deactivated", as LiDAR-LOC will overwrite the config in SD)

→ **On request**: Data is output when there is a request from a host computer via TCP/IP using CoLa2

→ **On request and also continuously to a target computer**:

- Continuous data output via UDP to a defined target address and for requests from a host computer via TCP/IP using CoLa2
- Scan data for both LiDAR-LOC and other application (multiple receivers)

a. NANS3 I/O (2nd channel is available, no broadcasting needed):

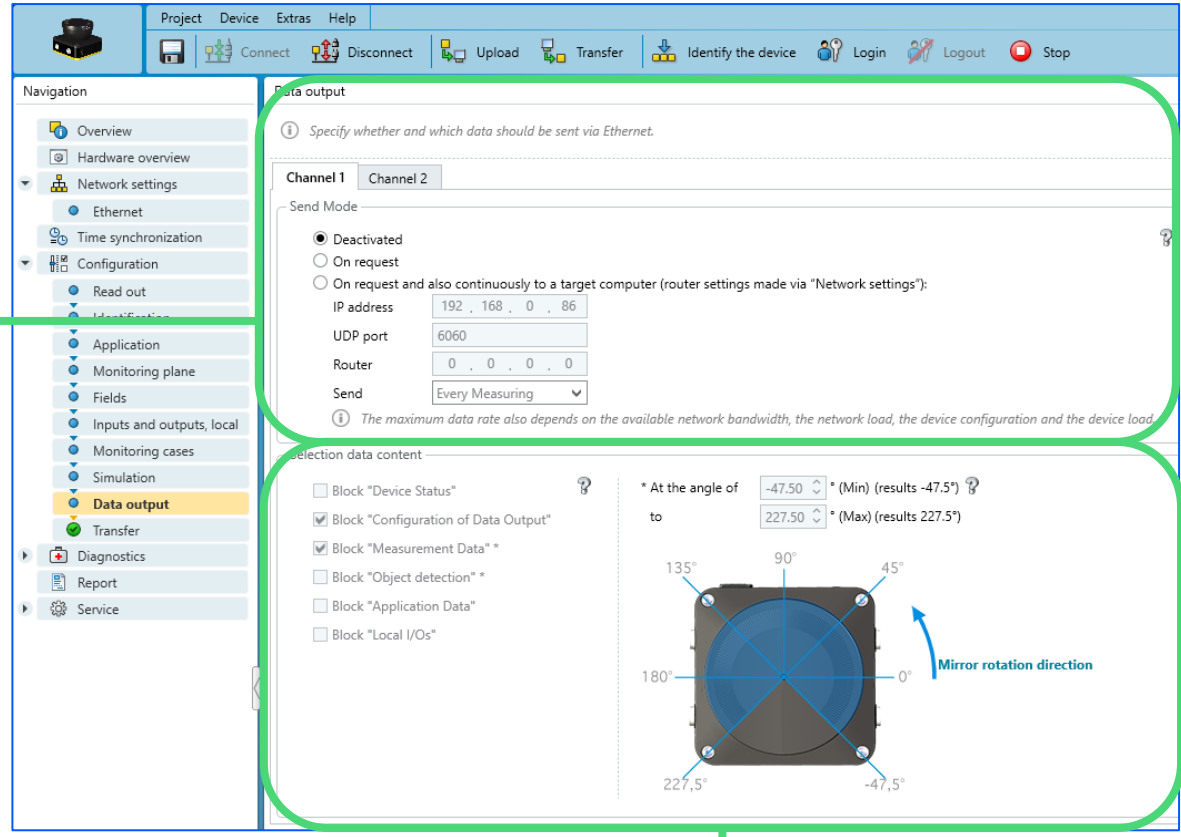
I. channel1: Use for LiDAR-LOC, keep deactivated

II. channel2: manually set IP and UDP port for 2nd application

b. Other scanners (only 1 channel available):

I. set target IP address to X.X.X.**255** for UDP broadcasting

II. If needed, consistent port setting in SD (6060 in screenshot) and SOPASair (LiDARs > Interface > receiverPort > manual)



Only to be configured for option **On request and also continuously to a target computer**

OVERVIEW

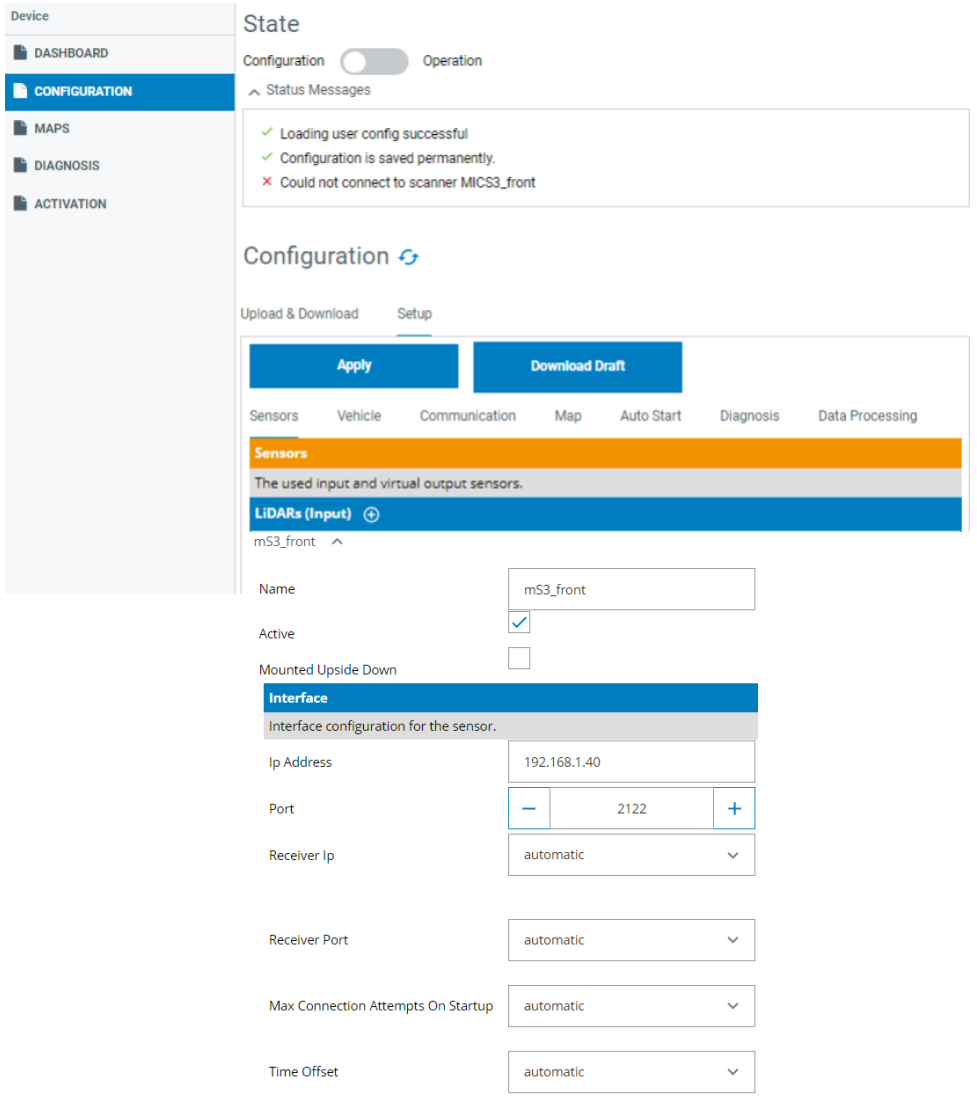
LiDAR-LOC SOPASAir web UI

Usually, LiDAR-LOC **automatically configures** the scan data output of the LiDAR sensors.

Only for **broadcast** and **multicast**, the data output configuration of the LiDAR sensor **must be configured manually**.

Approach

1. In SOPASAir, open tab **Configuration**.
2. Select the **Setup** tab.
3. Select the 2D LiDAR sensor to configure:
   a. *sensors > lidar*
4. To configure the **Interface**, select the **IP address** and **Port** of the sensor:
   a. IP address: sensor IP
   b. 2122: CoLa2 for nanoScan3, microScan3, outdoorScan3, picoScan1xx, multiScan1xx
   c. 2111 or 2112: CoLa A/B for all other LiDAR sensors
5. Confirm the selection by clicking the **Apply** button.
6. Save the selection using the Save Permanent button.



02

TYPES OF DATA TRANSMISSION

TYPES OF DATA TRANSMISSION

UDP or TCP Unicasting

Unicasting is used to transmit data from a single device to **a single device**.

With Unicasting the scan data is provided directly to the LiDAR-LOC device.

LiDAR-LOC **automatically configures** the scan data output of the LiDAR sensor.

Note:

→ If the scanner has two channels, the scan data can be provided to two different devices.

→ First channel is always automatically configured by LiDAR-LOC. The second channel is available for other applications (i.e., ROS drivers).

Unicasting configurations are shown from **examples 1 to 8**.

TYPES OF DATA TRANSMISSION

UDP broadcasting

Broadcasting is used to transmit data from a single device to **all devices in the network**.

Broadcasting is used if:

→ The LiDAR sensor does not have a second output channel

→ LiDAR-LOC runs in one IPC, e.g., SIM1200 and the scan data is additionally required in a second IPC (e.g., ROS drivers).

LiDAR-LOC **does not automatically configure** the scan data output of the LiDAR sensor. All settings from the data output of the LiDAR sensor are kept.

Broadcasting configurations are shown from **examples 9 and 10**.

Note: TiM and other TCP based sensors do not support UDP broadcasting. Most sensors with TCP allow multiple receiver from the same sensor.

TYPES OF DATA TRANSMISSION

UDP multicasting

Multicasting is used to for 2 use cases

1. Transmit data from a single device **to a group of devices in the network**.
2. Transmit data from a single device to **a group of applications on a single device**. E.g. send data from a LiDAR sensor to a single IPC running LiDAR-LOC and ROS drivers in parallel on the same device.

Multicasting allows multiple applications to run simultaneously with the same scan data, avoiding the port binding issues that occur with unicasting.

LiDAR-LOC **does not automatically configure** the scan data output of the LiDAR sensor. All settings from the data output of the LiDAR sensor are kept.

Multicasting use case 1 configuration is shown **in example 12.** Multicasting use case 2 configurations are shown **in example 11, 13 and 14**.

Note: TiM and other TCP based sensors do not support UDP broadcasting. Most sensors with TCP allow multiple receiver from the same sensor.

Use case 1

Use case 2

wired connection

Unicasted scan data

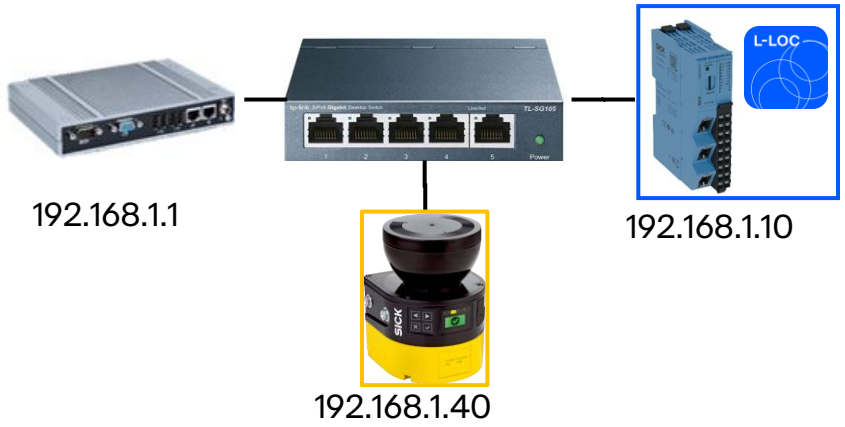
LiDAR-LOC Data

03

DATA TRANSMISSION EXAMPLES UNICASTING

EXAMPLE 1

1 x Scanner, LiDAR-LOC in SIM



Settings in SD

Send Mode

- (•) Deactivated
- ( ) On request
- ( ) On request and also continuously to a target computer (router settings made via "Network settings"):

IP address	0 . 0 . 0 . 0
UDP port	6060
Router	0 . 0 . 0 . 0
Send	Every Measuring

The maximum data rate also depends on the available network bandwidth, the network load, the device

LiDAR settings in SOPASair

mS3_front

Name	mS3_front
Active	☑
Mounted Upside Down	☐

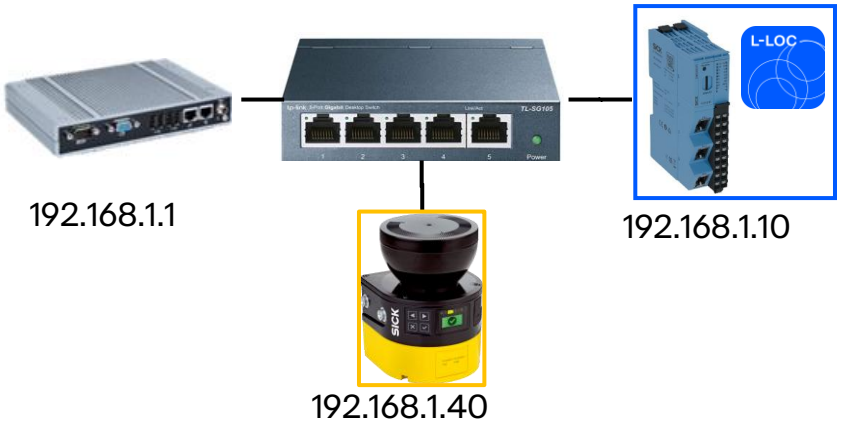
Interface

Interface configuration for the sensor.

Ip Address	192.168.1.40
Port	2122
Receiver Ip	automatic
Receiver Port	automatic
Max Connection Attempts On Startup	automatic
Time Offset	automatic

EXAMPLE 1

1 x Scanner, LiDAR-LOC in SIM



mS3 192.168.1.40		
LiDAR-LOC	ipAddress	192.168.1.40
	Port	2122
	receiverIP	automatic
	receiverPort	automatic

Safety Designer		
	Channel 1	Deactivated
	IP	-
	UDP port	-

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

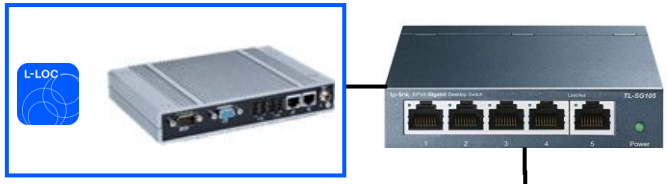
Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
57	0.348437	192.168.1.10	192.168.1.1	LLS2	98	5008 → 5010 Len=56
58	0.348549	192.168.1.10	192.168.1.1	LLS2	75	5008 → 5010 Len=33
59	0.398997	192.168.1.10	192.168.1.1	LLS2	75	5008 → 5010 Len=33
60	0.416714	192.168.1.10	192.168.1.1	LLS2	98	5008 → 5010 Len=56
61	0.416796	192.168.1.10	192.168.1.1	LLS2	75	5008 → 5010 Len=33
71	0.467319	192.168.1.10	192.168.1.1	LLS2	75	5008 → 5010 Len=33
72	0.483074	192.168.1.10	192.168.1.1	LLS2	98	5008 → 5010 Len=56

LLS Pose

EXAMPLE 2

1 x Scanner, LiDAR-LOC in IPC



192.168.1.1

192.168.1.40

Settings in SD

Send Mode

- (•) Deactivated
- ( ) On request
- ( ) On request and also continuously to a target computer (router settings made via "Network settings"):

IP address 0 . 0 . 0 . 0

UDP port 6060

Router 0 . 0 . 0 . 0

Send Every Measuring

The maximum data rate also depends on the available network bandwidth, the network load, the device

LiDAR settings in SOPASair

mS3_front

Name	mS3_front
Active	☑
Mounted Upside Down	☐

Interface

Interface configuration for the sensor.

Ip Address	192.168.1.40
Port	2122
Receiver Ip	automatic
Receiver Port	automatic
Max Connection Attempts On Startup	automatic
Time Offset	automatic

EXAMPLE 2

1 x Scanner, LiDAR-LOC in IPC

mS3 192.168.1.40					
LiDAR-LOC	ipAddress	192.168.1.40	Safety Designer	Channel 1	Deactivated
	Port	2122		IP	-
	receiverIP	automatic		UDP port	-
	receiverPort	automatic			

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

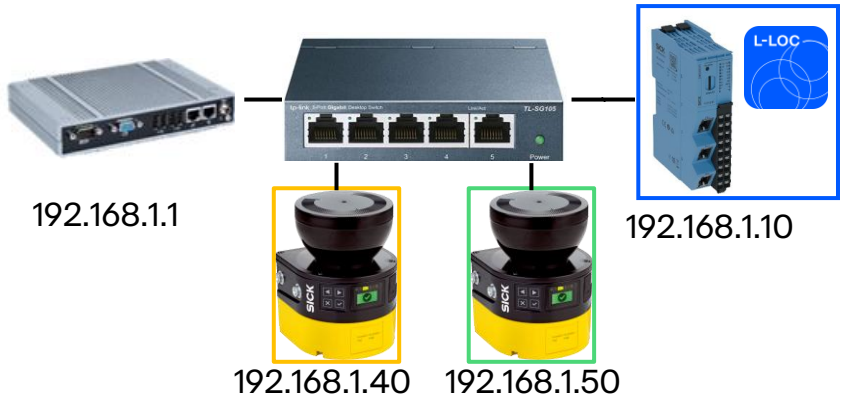
No.	Time	Source	Destination	Protocol	Length	Info
3065	29.610176	192.168.1.40	192.168.1.1	UDP	1502	51003 → 51003 Len=1460
3066	29.615266	192.168.1.40	192.168.1.1	UDP	1502	51003 → 51003 Len=1460
3067	29.619380	192.168.1.40	192.168.1.1	UDP	454	51003 → 51003 Len=412
3068	29.623576	192.168.1.1	192.168.1.230	LLS2	98	5008 → 5010 Len=56
3069	29.645251	192.168.1.40	192.168.1.1	UDP	1502	51003 → 51003 Len=1460
3070	29.650324	192.168.1.40	192.168.1.1	UDP	1502	51003 → 51003 Len=1460
3071	29.654355	192.168.1.40	192.168.1.1	UDP	454	51003 → 51003 Len=412
3072	29.658049	192.168.1.1	192.168.1.230	LLS2	98	5008 → 5010 Len=56

mS3 ← A UDP port is automatically assigned

LLS Pose

EXAMPLE 3

2 x Scanners, LiDAR-LOC in SIM



Settings in SD

Send Mode

- Deactivated
- On request
- On request and also continuously to a target computer (router settings made via "Network settings"):

IP address 0 . 0 . 0 . 0
UDP port 6060
Router 0 . 0 . 0 . 0
Send Every Measuring

Send Mode

- Deactivated
- On request
- On request and also continuously to a target computer (router settings made via "Network settings"):

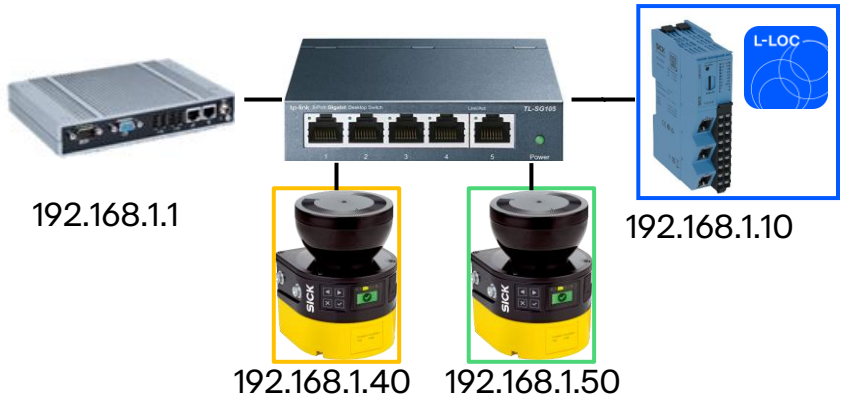
IP address 0 . 0 . 0 . 0
UDP port 6060
Router 0 . 0 . 0 . 0
Send Every Measuring

LiDAR settings in SOPASair

mS3_front		mS3_back	
Name	mS3_front	Name	mS3_back
Active	✓	Active	✓
Mounted Upside Down	☐	Mounted Upside Down	☐
Interface		**Interface**	
Interface configuration for the sensor.		Interface configuration for the sensor.	
Ip Address	192.168.1.40	Ip Address	192.168.1.50
Port	2122	Port	2122
Receiver Ip	automatic	Receiver Ip	automatic
Receiver Port	automatic	Receiver Port	automatic
Max Connection Attempts On Startup	automatic	Max Connection Attempts On Startup	automatic
Time Offset	automatic	Time Offset	automatic

EXAMPLE 3

2 x Scanners, LiDAR-LOC in SIM



mS3 front					
LiDAR-LOC	ipAddress	192.168.1.40	Safety Designer	Channel 1	Deactivated
	Port	2122		IP	-
	receiverIP	automatic		UDP port	-
	receiverPort	automatic			

mS3 back					
LiDAR-LOC	ipAddress	192.168.1.50	Safety Designer	Channel 1	Deactivated
	Port	2122		IP	-
	receiverIP	automatic		UDP port	-
	receiverPort	automatic			

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

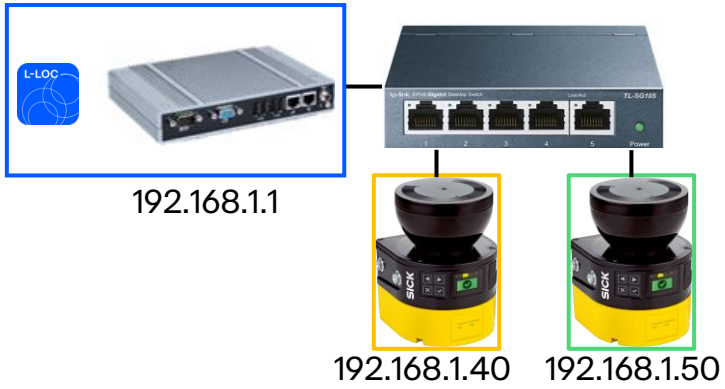
Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
57	0.348437	192.168.1.10	192.168.1.1	LLS2	98	5008 → 5010 Len=56
58	0.348549	192.168.1.10	192.168.1.1	LLS2	75	5008 → 5010 Len=33
59	0.398997	192.168.1.10	192.168.1.1	LLS2	75	5008 → 5010 Len=33
60	0.416714	192.168.1.10	192.168.1.1	LLS2	98	5008 → 5010 Len=56
61	0.416796	192.168.1.10	192.168.1.1	LLS2	75	5008 → 5010 Len=33
71	0.467319	192.168.1.10	192.168.1.1	LLS2	75	5008 → 5010 Len=33
72	0.483074	192.168.1.10	192.168.1.1	LLS2	98	5008 → 5010 Len=56

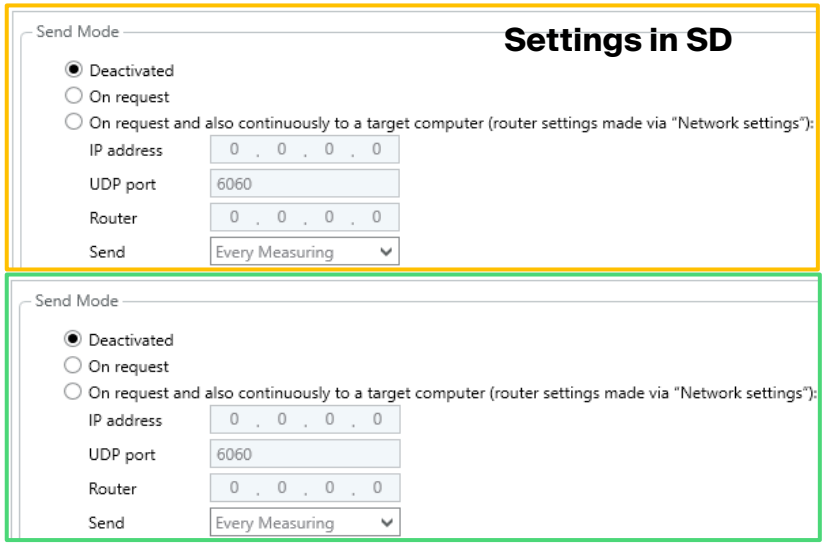
LLS pose

EXAMPLE 4

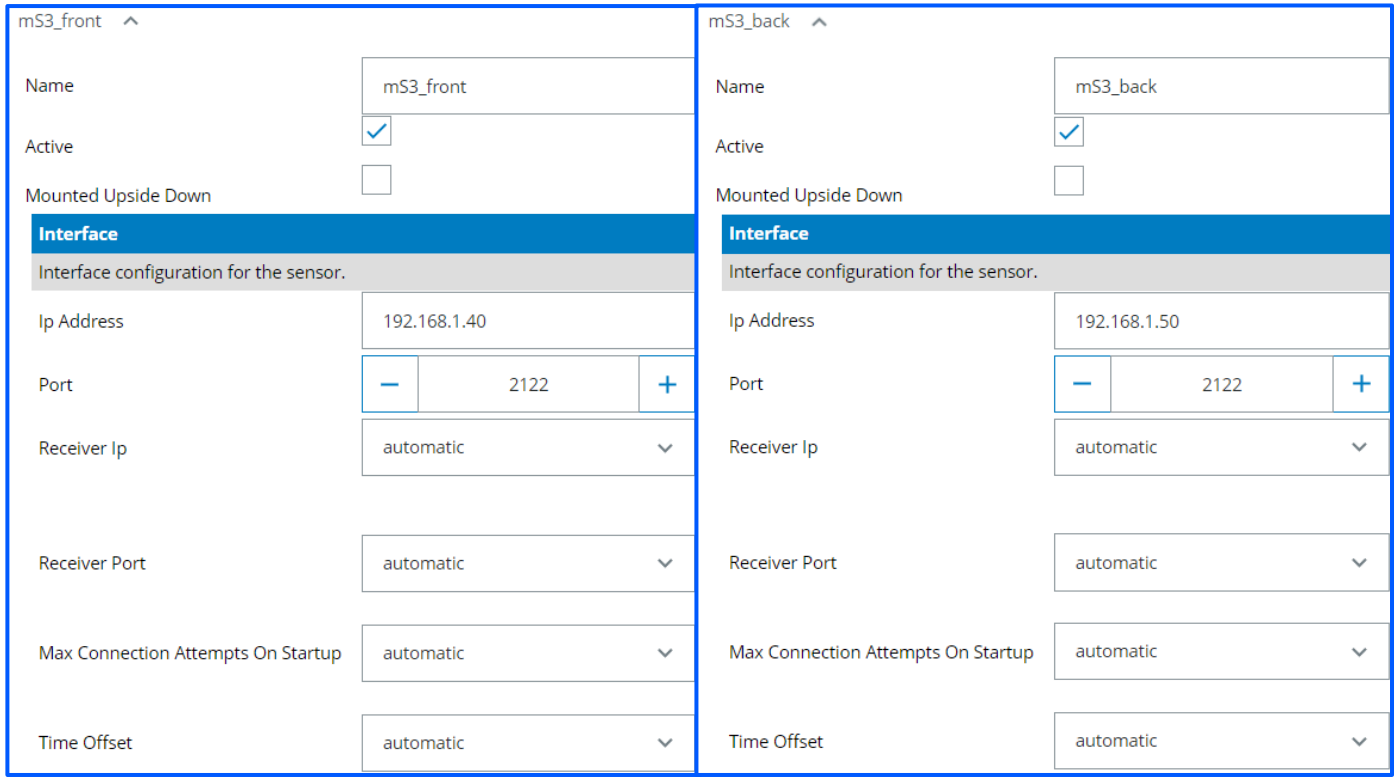
2 x Scanners, LiDAR-LOC in IPC



Settings in SD

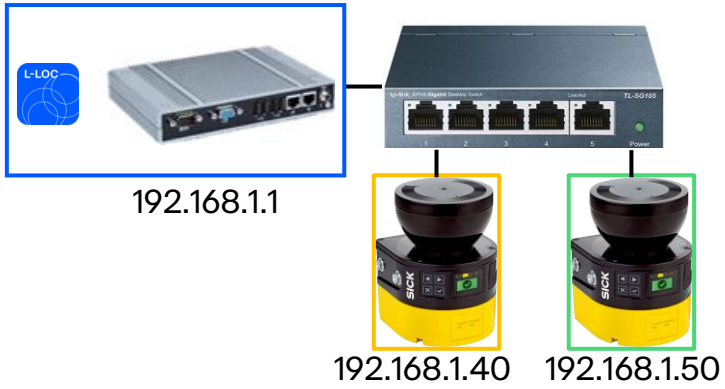


LiDAR settings in SOPASair



EXAMPLE 4

2 x Scanners, LiDAR-LOC in IPC



mS3 front					
LiDAR-LOC	ipAddress	192.168.1.40	Safety Designer	Channel 1	Deactivated
	Port	2122		IP	-
	receiverIP	automatic		UDP port	-
	receiverPort	automatic			

mS3 back					
LiDAR-LOC	ipAddress	192.168.1.50	Safety Designer	Channel 1	Deactivated
	Port	2122		IP	-
	receiverIP	automatic		UDP port	-
	receiverPort	automatic			

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
2198	12.231781	192.168.1.1	192.168.1.230	LLS2	98	5008 → 5010 Len=56
2199	12.242722	192.168.1.50	192.168.1.1	UDP	1502	51007 → 51007 Len=1460
2200	12.247822	192.168.1.50	192.168.1.1	UDP	1502	51007 → 51007 Len=1460
2201	12.252059	192.168.1.50	192.168.1.1	UDP	454	51007 → 51007 Len=412
2202	12.258370	192.168.1.40	192.168.1.1	UDP	1502	51006 → 51006 Len=1460
2203	12.263372	192.168.1.40	192.168.1.1	UDP	1502	51006 → 51006 Len=1460
2204	12.267451	192.168.1.40	192.168.1.1	UDP	454	51006 → 51006 Len=412
2205	12.276368	192.168.1.1	192.168.1.230	LLS2	98	5008 → 5010 Len=56
2206	12.282730	192.168.1.50	192.168.1.1	UDP	1502	51007 → 51007 Len=1460
2207	12.287886	192.168.1.50	192.168.1.1	UDP	1502	51007 → 51007 Len=1460

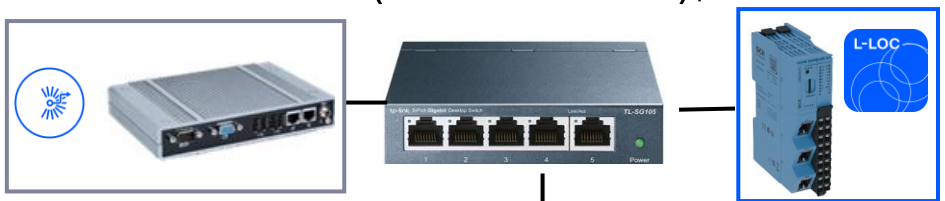
mS3_back

mS3_back

LLS pose

EXAMPLE 5

1 x Scanner (CH1&CH2), LiDAR-LOC in SIM & 2nd App in IPC

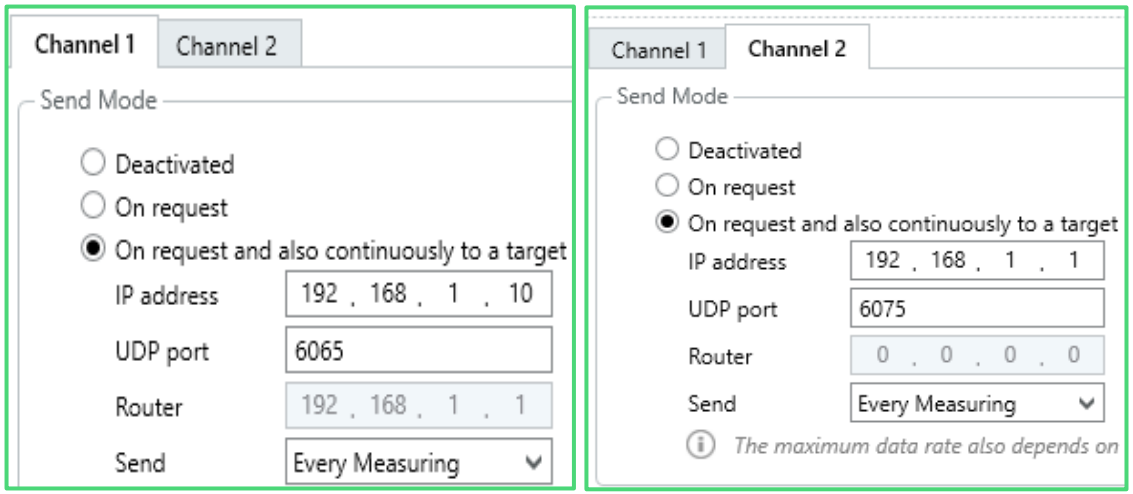


192.168.1.1

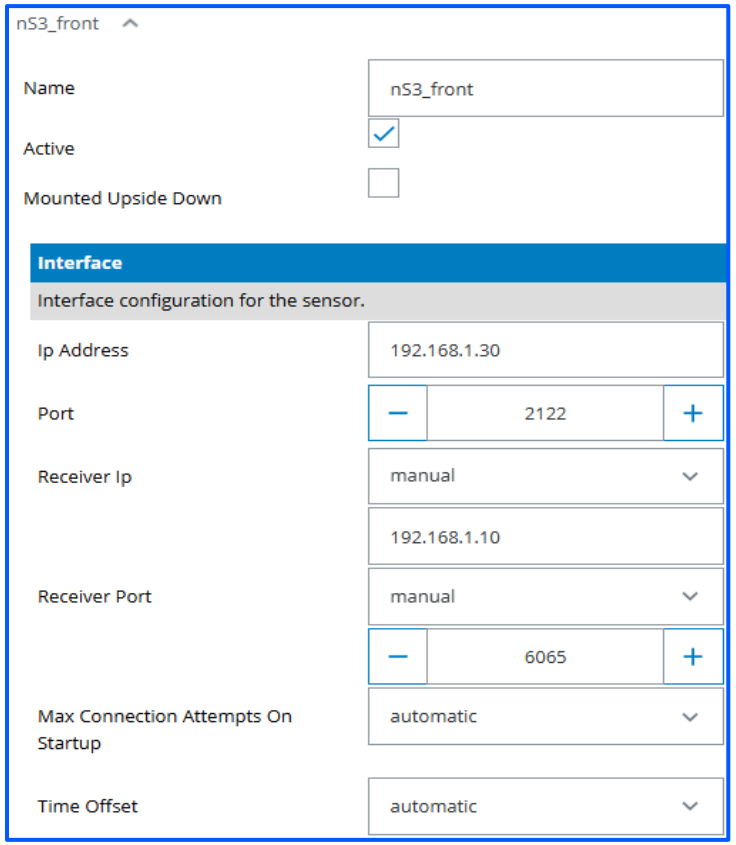
192.168.1.10

192.168.1.30

Settings in SD



LiDAR settings in SOPASair



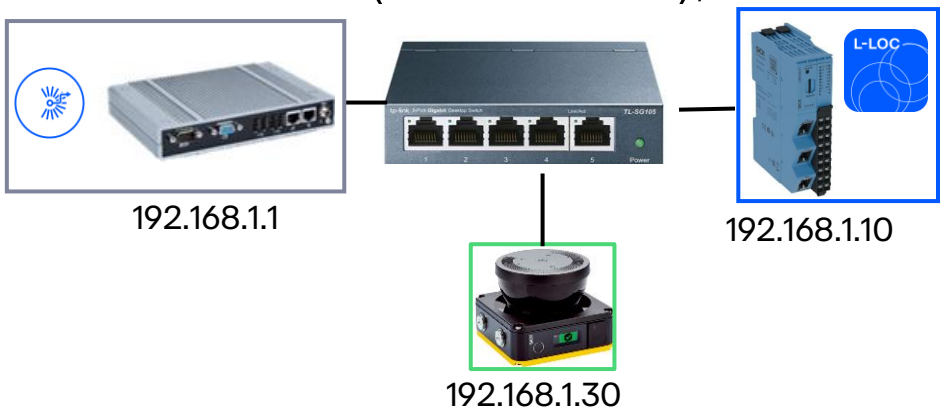
NOTE:

→ LiDAR-LOC can only get data from “Channel 1”

→ Setup can also work with automatic settings

EXAMPLE 5

1 x Scanner (CH1&CH2), LiDAR-LOC in SIM & 2nd App in IPC

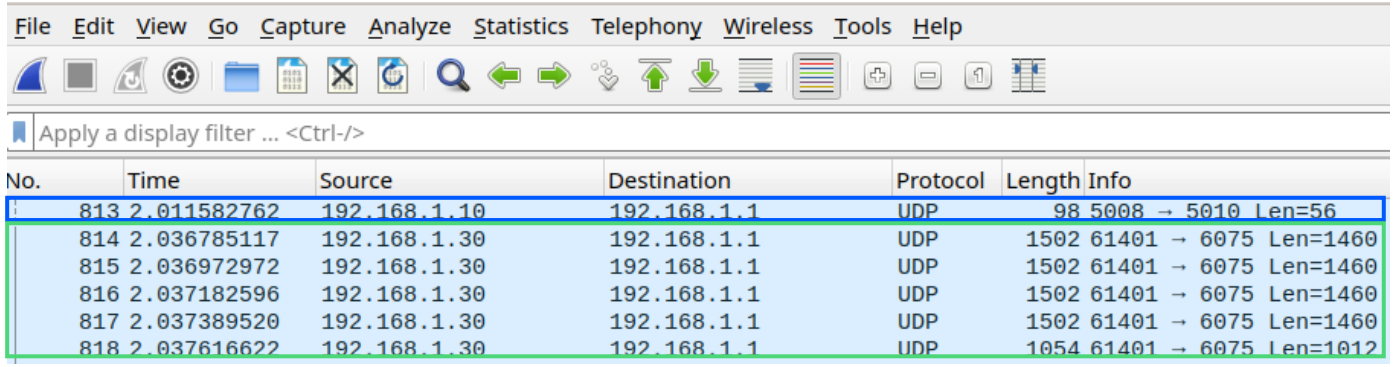


nS3 192.168.1.30					
LiDAR-LOC	ipAddress	192.168.1.30	Safety Designer	Channel 1	On request and continuously
	Port	2122		IP	192.168.1.10
	receiverIP	192.168.1.10		UDP port	6060
	receiverPort	6065		Channel 2	On request and continuously
				IP	192.168.1.1
				UDP port	6075

NOTE:

→ LiDAR-LOC can only get data from “Channel 1”

→ Setup can also work with automatic settings



LLS Pose

nS3_front

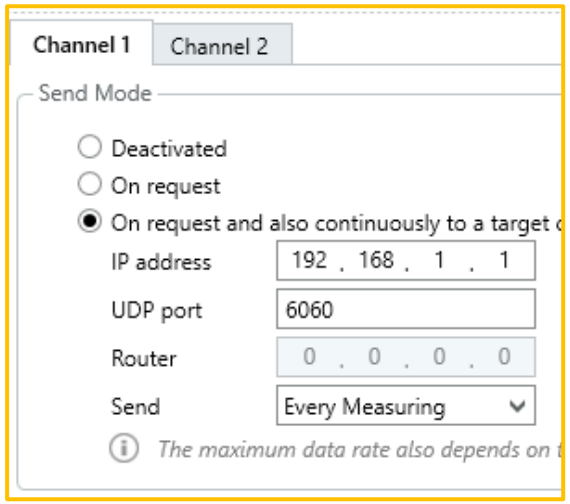
EXAMPLE 6

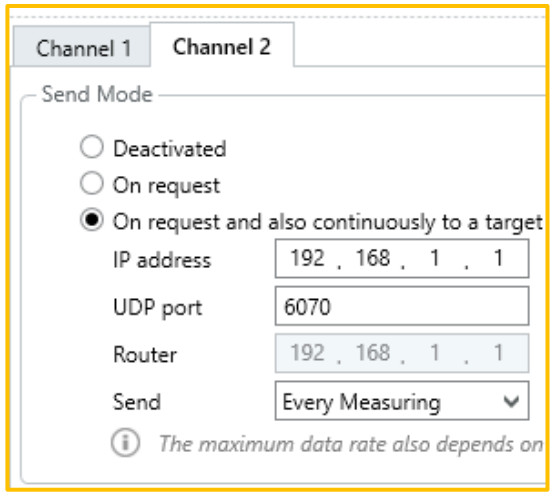
1 x Scanner (CH1&CH2), LiDAR-LOC & 2nd App in IPC

192.168.1.1

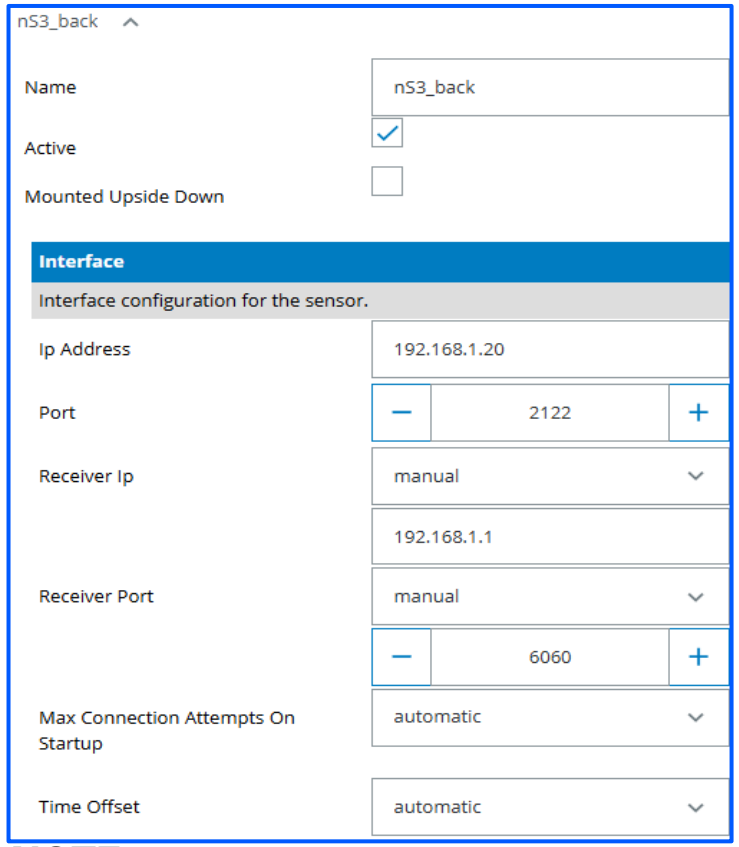
192.168.1.20

Settings in SD





LiDAR settings in SOPASair



NOTE:

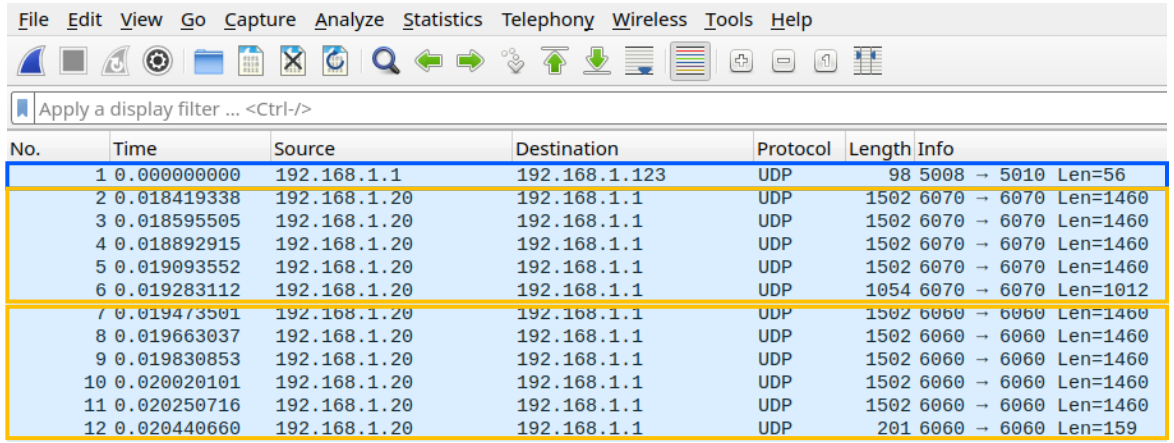
→ LiDAR-LOC can only get data from “Channel 1”

→ Setup can also work with automatic settings

EXAMPLE 6

1 x Scanner (CH1&CH2), LiDAR-LOC & 2nd App in IPC

nS3 192.168.1.20					
LiDAR-LOC	ipAddress	192.168.1.20	Safety Designer	Channel 1	On request and continuously
	Port	2122		IP	192.168.1.1
	receiverIP	192.168.1.1		UDP port	6060
	receiverPort	6060		Channel 2	On request and continuously
				IP	192.168.1.1
				UDP port	6070

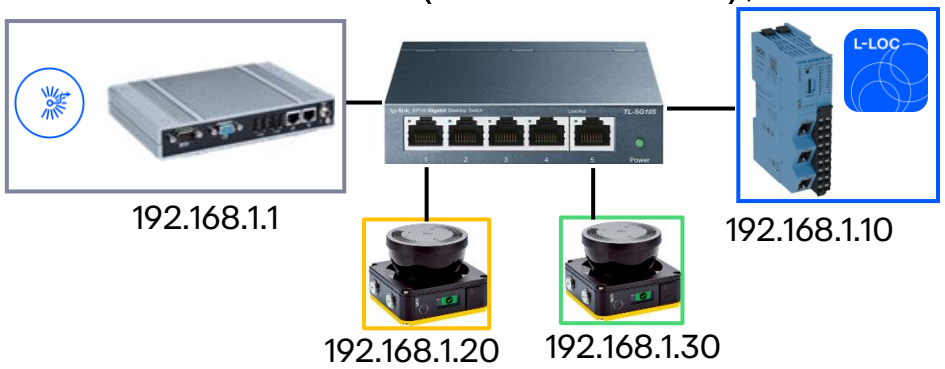


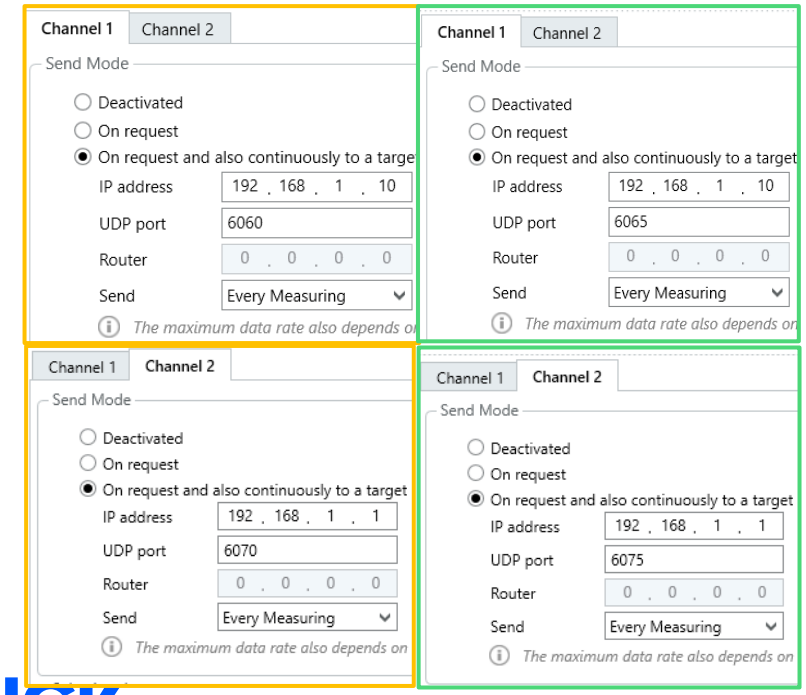
LLS pose

nS3 back

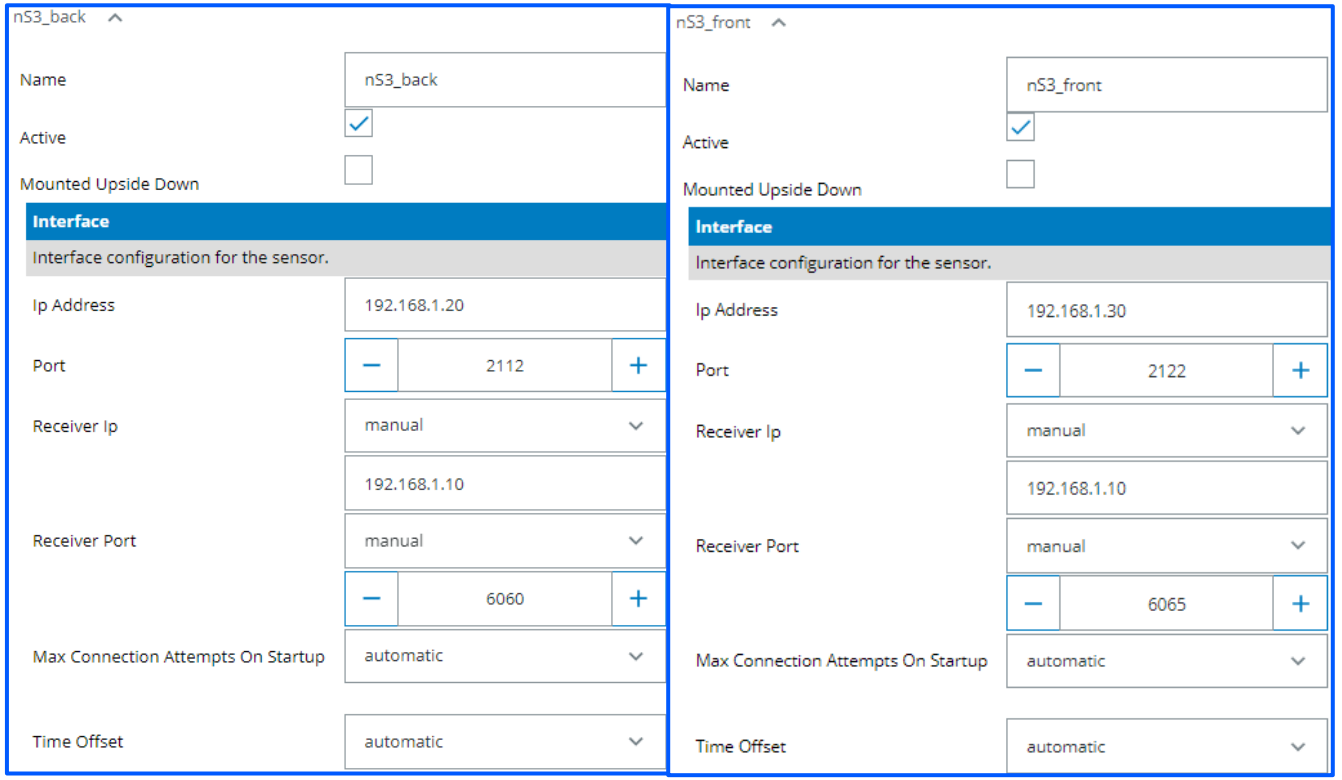
EXAMPLE 7

2 x Scanners (CH1&CH2), LiDAR-LOC in SIM & 2nd App in IPC



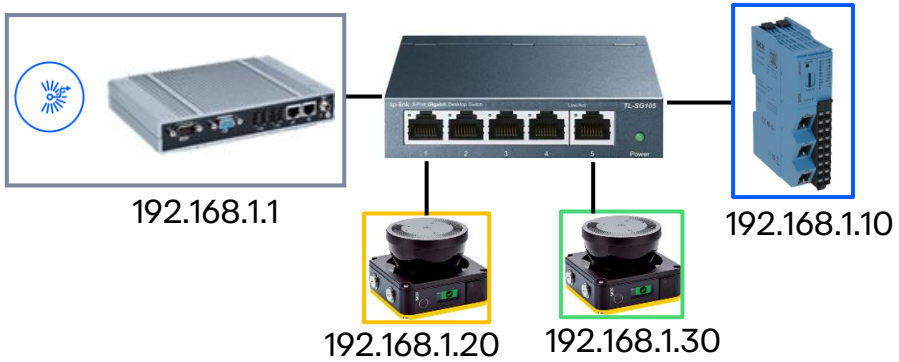


LiDAR settings in SOPASair



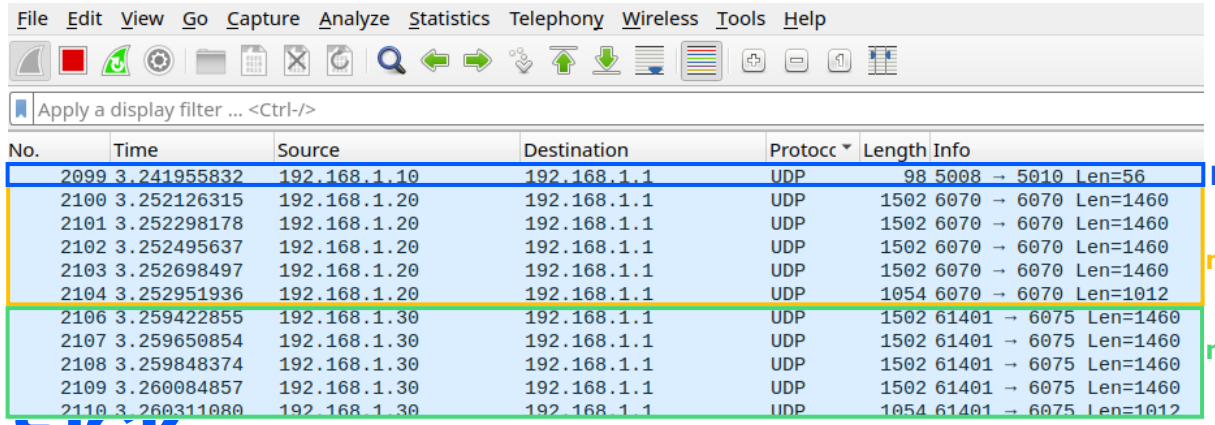
EXAMPLE 7

2 x Scanners (CH1&CH2), LiDAR-LOC in SIM & 2nd App in IPC



nS3 back 192.168.1.20					
LiDAR-LOC	ipAddress	192.168.1.20	Safety Designer	Channel 1	On request and continuously
	Port	2122		IP	192.168.10
	receiverIP	192.168.1.10		UDP port	6060
	receiverPort	6060		Channel 2	On request and continuously
				IP	192.168.1.1
				UDP port	6070

nS3 front 192.168.1.30					
LiDAR-LOC	ipAddress	192.168.1.30	Safety Designer	Channel 1	On request and continuously
	Port	2122		IP	192.168.1.10
	receiverIP	192.168.1.10		UDP port	6065
	receiverPort	6065		Channel 2	On request and continuously
				IP	192.168.1.1
				UDP port	6075



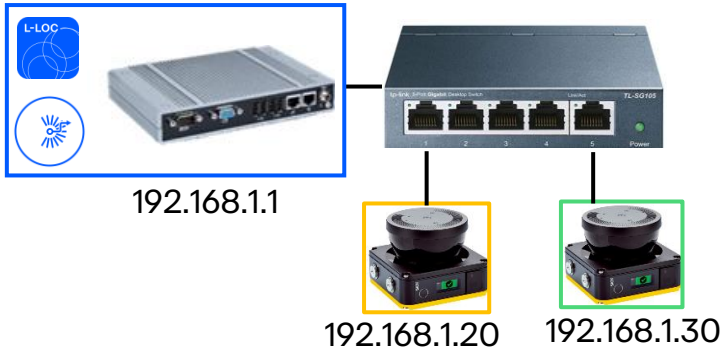
LLS Pose

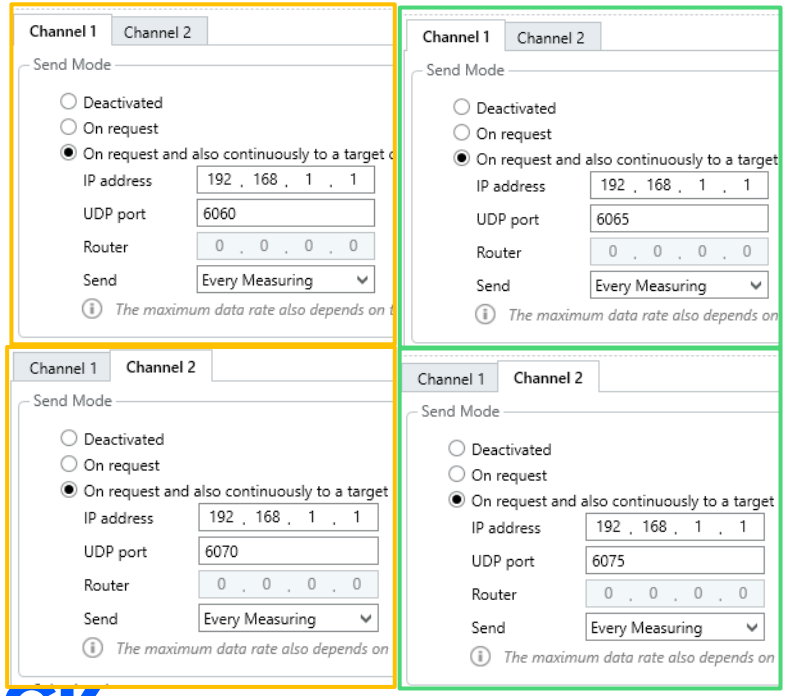
nS3_back

nS3_front

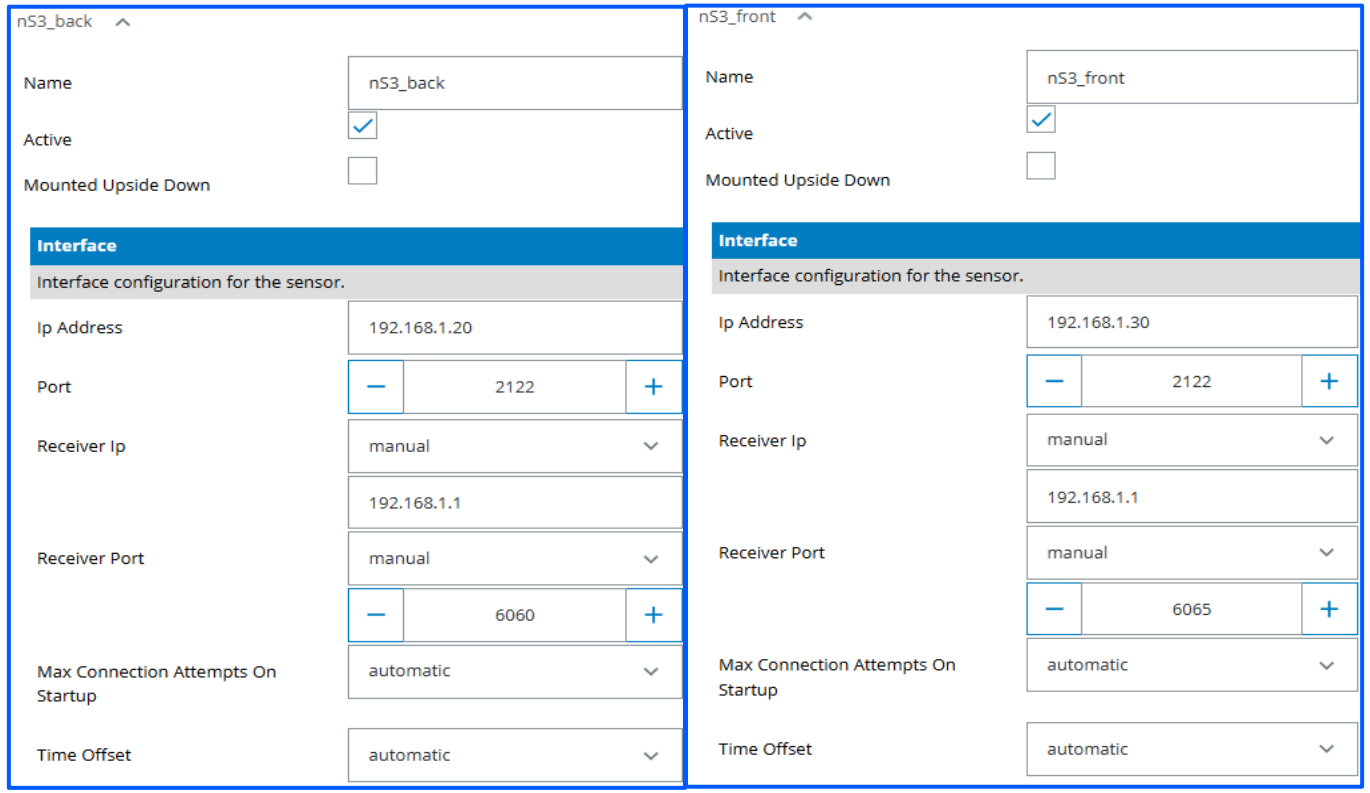
EXAMPLE 8

2 x Scanners (CH1&CH2), LiDAR-LOC & 2nd App in IPC



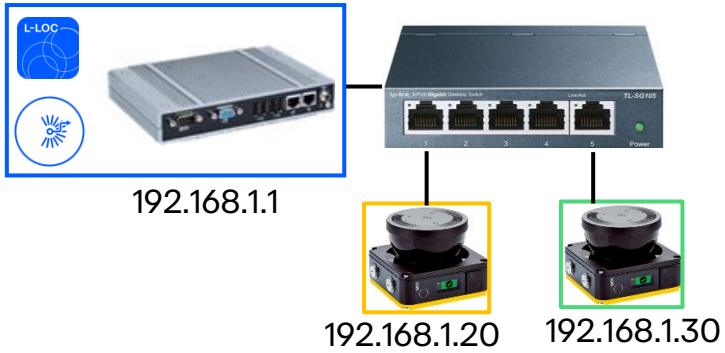


LiDAR settings in SOPASair



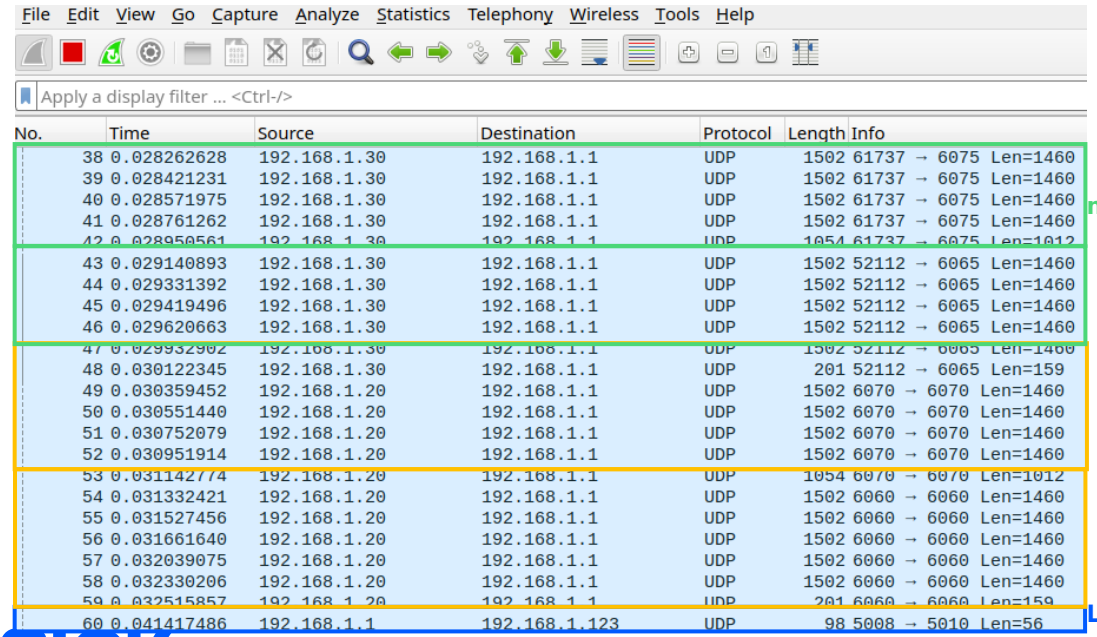
EXAMPLE 8

2 x Scanners (CH1&CH2), LiDAR-LOC & 2nd App in IPC



nS3 back 192.168.1.20					
LiDAR-LOC	ipAddress	192.168.1.20	Safety Designer	Channel 1	On request and continuously
	Port	2122		IP	192.168.1.1
	receiverIP	192.168.1.1		UDP port	6060
	receiverPort	6060		Channel 2	On request and continuously
				IP	192.168.1.1
				UDP port	6070

nS3 front 192.168.1.30					
LiDAR-LOC	ipAddress	192.168.1.30	Safety Designer	Channel 1	On request and continuously
	Port	2122		IP	192.168.1.1
	receiverIP	192.168.1.1		UDP port	6065
	receiverPort	6065		Channel 2	On request and continuously
				IP	192.168.1.1
				UDP port	6075



03

DATA TRANSMISSION EXAMPLES BROADCASTING

BROADCASTING

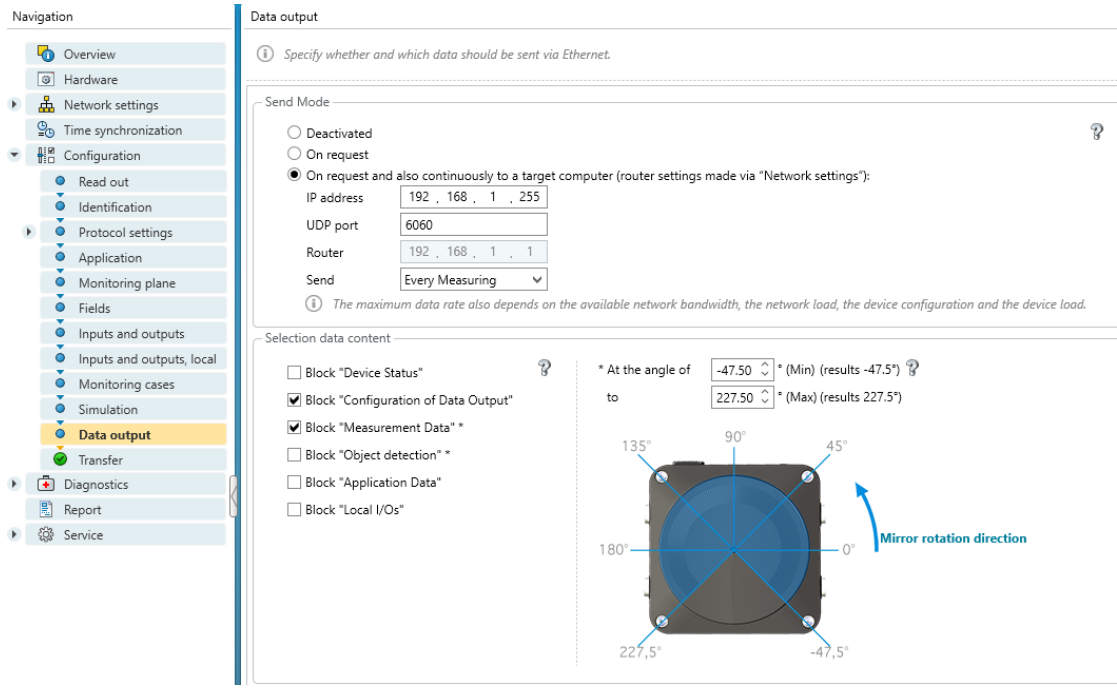
Safety Designer setting

For broadcast, LiDAR-LOC **does not** automatically configure or overwrite the scan data output of the LiDAR sensor.

All settings from the Data output configuration of the LiDAR sensor are kept and **must be configured manually**.

Approach

1. In the Safety Designer, open **Configuration > Data output**.
2. Choose **On request and additionally continuously**.
3. In section **Send Mode**, enter **the IP address** for broadcast.
   a. E.g., IP address 192.168.1.255, when the IP addresses of the receivers are in 192.168.1.x
4. Enter the **UDP port**, for example, 6060.
5. In **Send**, select **Every Measuring.**
6. In section **Selection data content**, select the information blocks **Measurement Data** and **Configuration of Data Output**.



EXAMPLE 9

1 x Scanner, LiDAR-LOC in SIM & 2nd App in IPC

Settings in SD

Send Mode

- ○ Deactivated
- ○ On request
- ◉ On request and also continuously to a target computer (router settings made via "Network settings"):

IP address	192 . 168 . 1 . 255
UDP port	6060
Router	0 . 0 . 0 . 0
Send	Every Measuring

The maximum data rate also depends on the available network bandwidth, the network load, the de

LiDAR settings in SOPASair

mS3_front

Name	mS3_front
Active	☑
Mounted Upside Down	☐

Interface

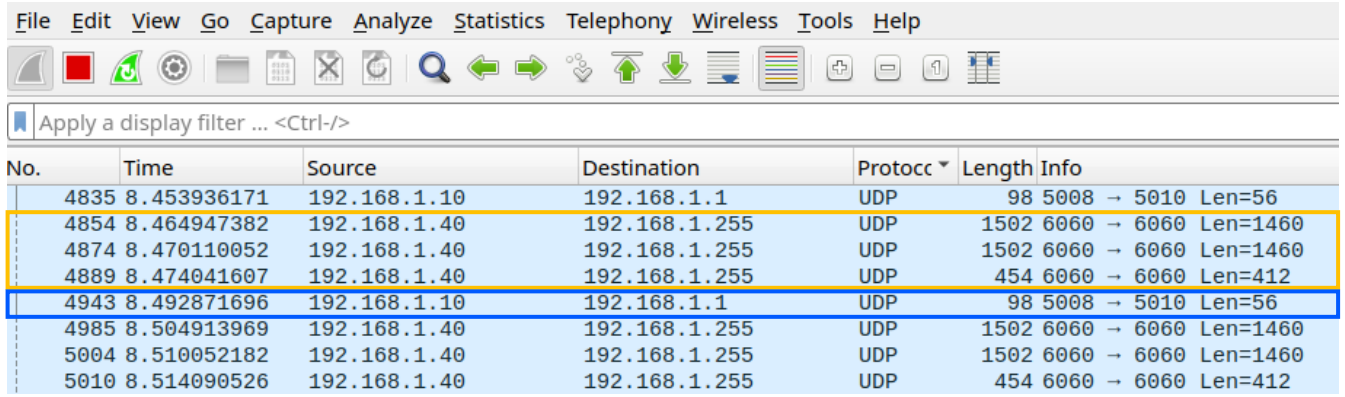
Interface configuration for the sensor.

Ip Address	192.168.1.40
Port	2122
Receiver Ip	manual
	192.168.1.10
Receiver Port	manual
	6060
Max Connection Attempts On Startup	automatic
Time Offset	automatic

EXAMPLE 9

1 x Scanner, LiDAR-LOC in SIM & 2nd App in IPC

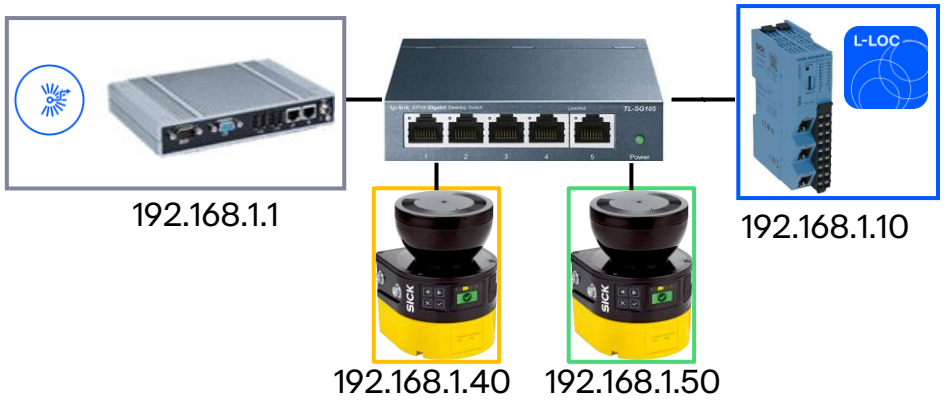
mS3_front					
LiDAR-LOC	ipAddress	192.168.1.40	Safety Designer	Channel 1	On request and continuously
	Port	2122		IP	192.168.1.255
	receiverIP	192.168.1.10		UDP port	6060
	receiverPort	6060			



mS3
LLS Pose

EXAMPLE 10

2 x Scanners, LiDAR-LOC in SIM & 2nd App in IPC



Settings in SD

Send Mode

- Deactivated
- On request
- On request and also continuously to a target computer (router settings made via "Network settings"):

IP address	192 . 168 . 1 . 255
UDP port	6060
Router	0 . 0 . 0 . 0
Send	Every Measuring

The maximum data rate also depends on the available network bandwidth, the network load, the de

Send Mode

- Deactivated
- On request
- On request and also continuously to a target computer (router settings made via "Network settings"):

IP address	192 . 168 . 1 . 255
UDP port	6061
Router	192 . 168 . 1 . 1
Send	Every Measuring

The maximum data rate also depends on the available network bandwidth, the network load, the d

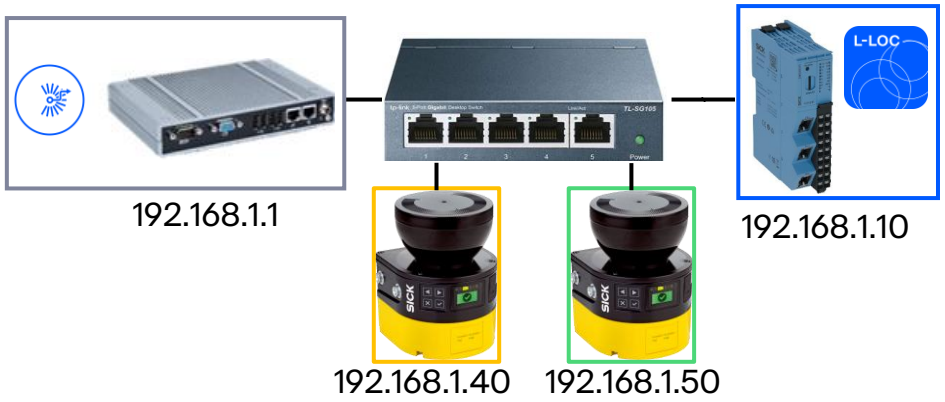
LiDAR settings in SOPASair

mS3_front	
Name	mS3_front
Active	✓
Mounted Upside Down	
Interface	
Interface configuration for the sensor.	
Ip Address	192.168.1.40
Port	2122
Receiver Ip	manual
	192.168.1.10
Receiver Port	manual
	6060
Max Connection Attempts On Startup	automatic
Time Offset	automatic

mS3_back	
Name	mS3_back
Active	✓
Mounted Upside Down	
Interface	
Interface configuration for the sensor.	
Ip Address	192.168.1.50
Port	2122
Receiver Ip	manual
	192.168.1.10
Receiver Port	manual
	6061
Max Connection Attempts On Startup	automatic
Time Offset	automatic

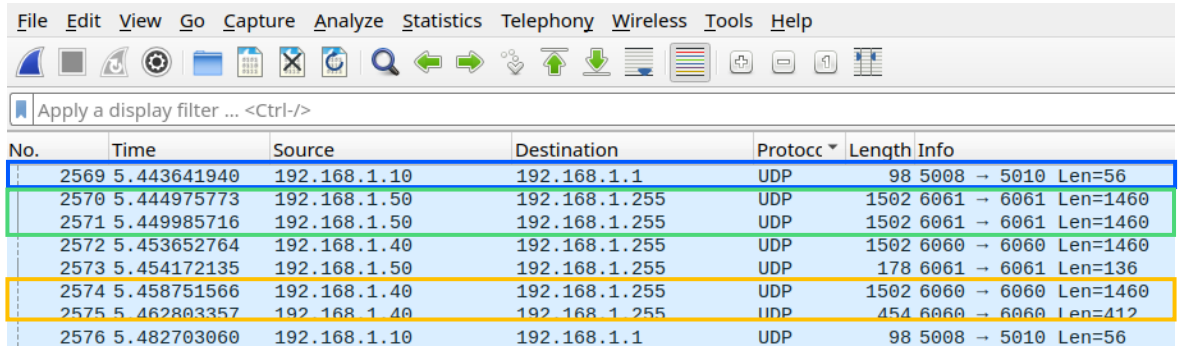
EXAMPLE 10

2 x Scanners, LiDAR-LOC in SIM & 2nd App in IPC



mS3_front				
LiDAR-LOC	ipAddress	192.168.1.40	Safety Designer	Channel 1	On request and continuously
	Port	2122		IP	192.168.1.255
	receiverIP	192.168.1.10		UDP port	6060
	receiverPort	6060			

mS3_back				
LiDAR-LOC	ipAddress	192.168.1.50	Safety Designer	Channel 1	On request and continuously
	Port	2122		IP	192.168.1.255
	receiverIP	192.168.1.10		UDP port	6070
	receiverPort	6070			



03

DATA TRANSMISSION EXAMPLES MULTICASTING

MULTICASTING

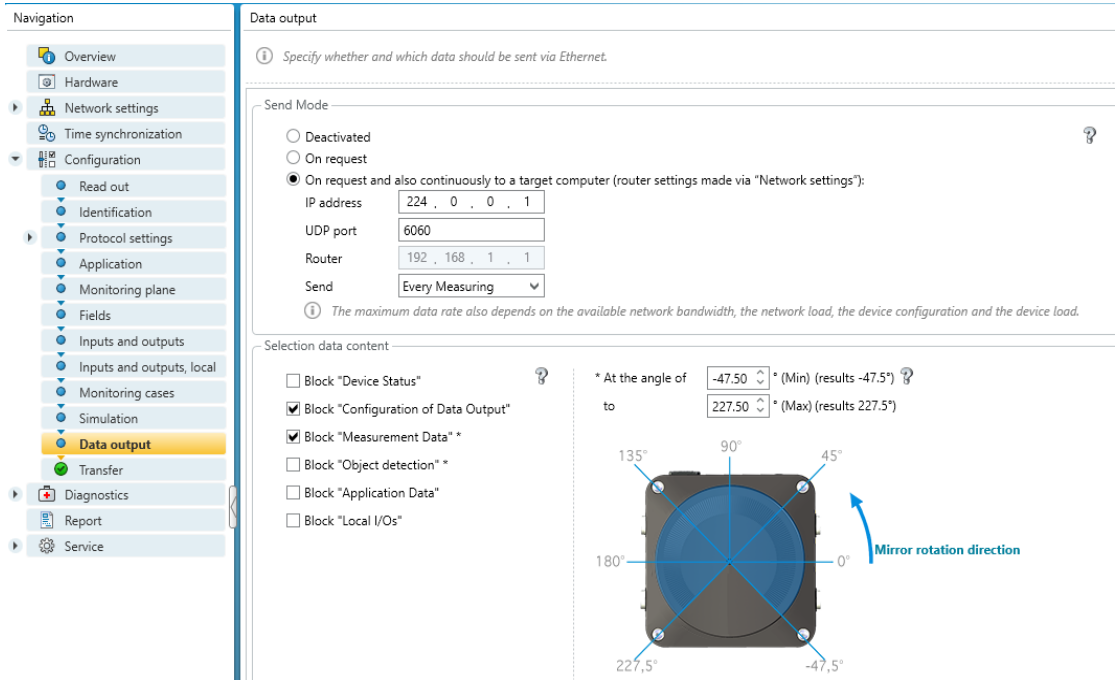
Safety Designer setting

For broadcast, LiDAR-LOC **does not** automatically configure or overwrite the scan data output of the LiDAR sensor.

All settings from the Data output configuration of the LiDAR sensor are kept and **must be configured manually**.

Approach

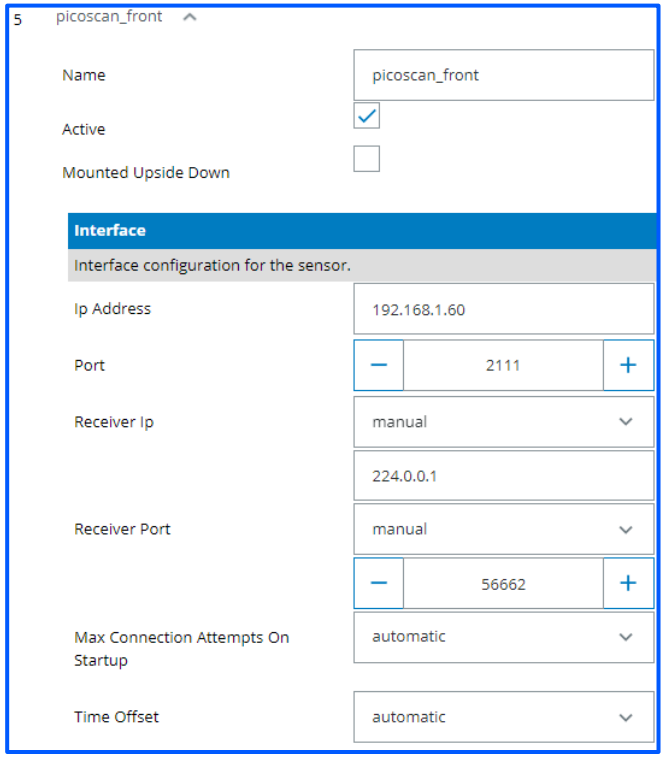
1. In the Safety Designer, open **Configuration > Data output**.
2. Choose **On request and additionally continuously**.
3. In section **Send Mode**, enter **the IP address** for broadcast.
   a. E.g., IP address 224.0.0.1, when the IP addresses of the receivers are in 192.168.1.x
4. Enter the **UDP port**, for example, 6060.
5. In **Send**, select **Every Measuring**.
6. In section **Selection data content**, select the information blocks **Measurement Data** and **Configuration of Data Output**.

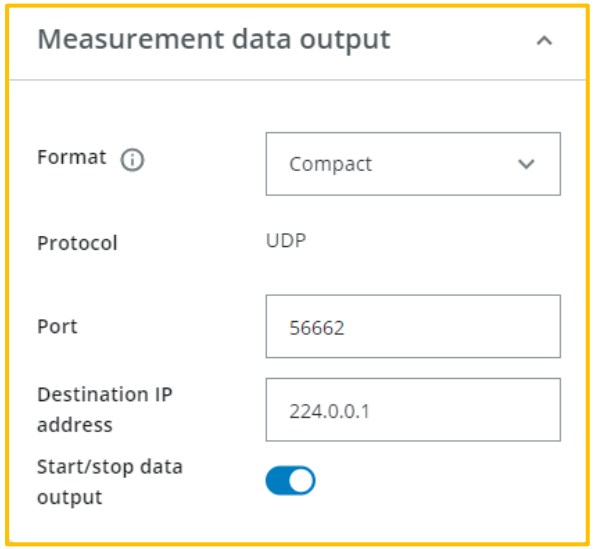


EXAMPLE 11

1 x Scanner, LiDAR-LOC & 2nd App in IPC

LiDAR settings in SOPASair





Settings in SOPASair (picoScan GUI)

EXAMPLE 11

1 x Scanner, LiDAR-LOC & 2nd App in IPC

PicoScan 192.168.1.60					
LiDAR-LOC	ipAddress	192.168.1.60	SOPASAir	Destination IP	224.0.0.1
	Port	2111		UDP port	56662
	receiverIP	224.0.0.1			
	receiverPort	56662			

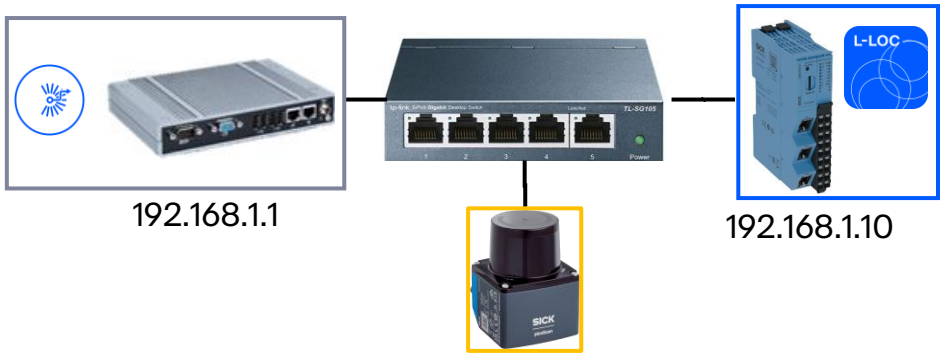
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info	
4681	6.800866931	192.168.1.1	192.168.1.123	HTTP	226	HTTP/1.1 200 OK (appli	
4682	6.803600662	192.168.1.60	224.0.0.1	UDP	990	58819 → 56662 Len=948	picoscan
4683	6.806894889	192.168.1.60	224.0.0.1	UDP	990	58819 → 56662 Len=948	
4684	6.810569889	192.168.1.60	224.0.0.1	UDP	990	58819 → 56662 Len=948	
4685	6.814670876	192.168.1.60	224.0.0.1	UDP	990	58819 → 56662 Len=948	
4686	6.818883499	192.168.1.60	224.0.0.1	UDP	990	58819 → 56662 Len=948	
4687	6.823263942	192.168.1.60	224.0.0.1	UDP	990	58819 → 56662 Len=948	
4688	6.827210789	192.168.1.60	224.0.0.1	UDP	990	58819 → 56662 Len=948	
4689	6.831466588	192.168.1.60	224.0.0.1	UDP	990	58819 → 56662 Len=948	
4690	6.833164263	192.168.1.60	224.0.0.1	UDP	990	58819 → 56662 Len=948	
4691	6.835656891	192.168.1.123	192.168.1.1	TCP	60	50818 → 80 [ACK] Seq=1	
4692	6.839756666	192.168.1.1	192.168.1.123	UDP	98	5008 → 5010 Len=56	LLS Pose
4693	6.839999270	192.168.1.1	192.168.1.123	UDP	82	5008 → 5010 Len=40	
4694	6.840190538	192.168.1.1	192.168.1.123	UDP	82	5008 → 5010 Len=40	
4695	6.840385837	192.168.1.1	192.168.1.123	UDP	82	5008 → 5010 Len=40	

EXAMPLE 12

1 x Scanner, LiDAR-LOC in SIM & 2nd App in IPC

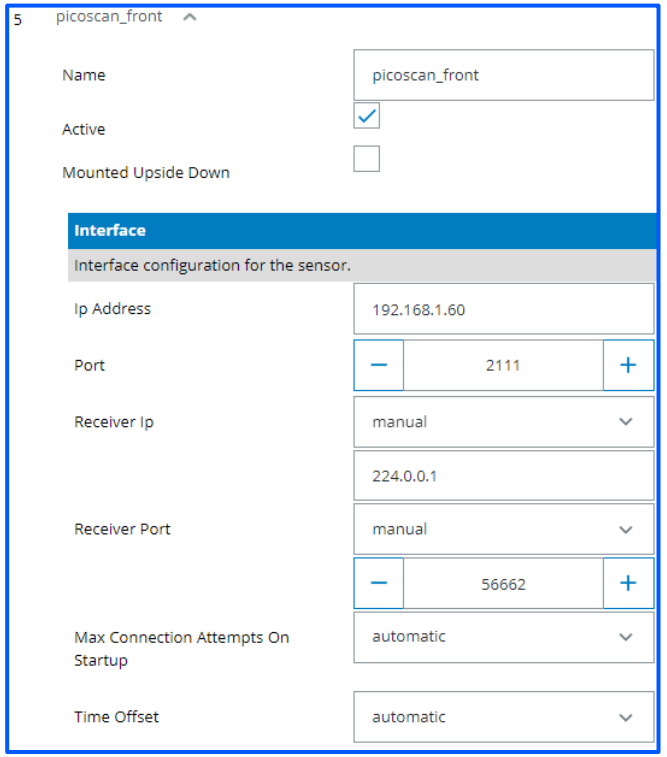


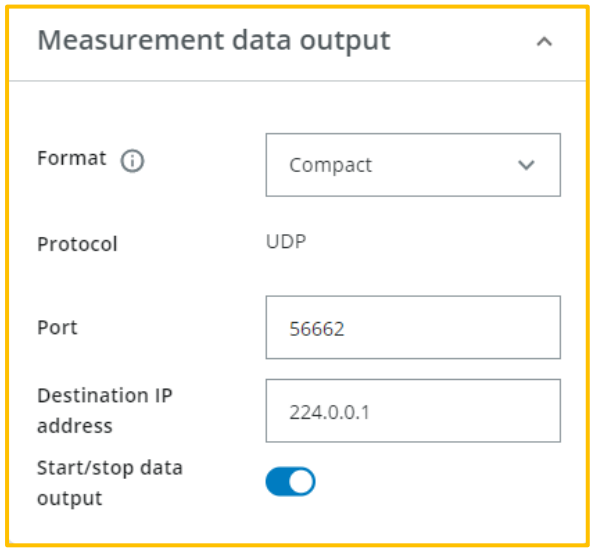
192.168.1.1

192.168.1.10

192.168.1.60

LiDAR settings in SOPASair





Settings in SOPASair (picoScan GUI)

EXAMPLE 12

1 x Scanner, LiDAR-LOC in SIM & 2nd App in IPC

PicoScan 192.168.1.60					
LiDAR-LOC	ipAddress	192.168.1.60	SOPASAir	Destination IP	224.0.0.1
	Port	2111		UDP port	56662
	receiverIP	224.0.0.1			
	receiverPort	56662			

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

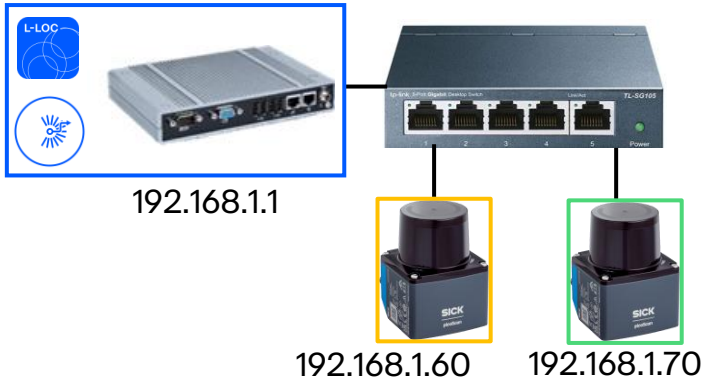
No.	Time	Source	Destination	Protocol	Length	Info
10	0.027853974	192.168.1.60	224.0.0.1	UDP	990	58819 → 56662 Len=948
11	0.031945877	192.168.1.60	224.0.0.1	UDP	990	58819 → 56662 Len=948
12	0.033413860	192.168.1.60	224.0.0.1	UDP	990	58819 → 56662 Len=948
13	0.050055684	192.168.1.60	224.0.0.1	UDP	990	58819 → 56662 Len=948
14	0.050345772	192.168.1.10	192.168.1.1	UDP	98	5008 → 5010 Len=56
15	0.053076684	192.168.1.60	224.0.0.1	UDP	990	58819 → 56662 Len=948
16	0.057135095	192.168.1.60	224.0.0.1	UDP	990	58819 → 56662 Len=948
17	0.061205410	192.168.1.60	224.0.0.1	UDP	990	58819 → 56662 Len=948
18	0.065375841	192.168.1.60	224.0.0.1	UDP	990	58819 → 56662 Len=948
19	0.069516836	192.168.1.60	224.0.0.1	UDP	990	58819 → 56662 Len=948
20	0.073750111	192.168.1.60	224.0.0.1	UDP	990	58819 → 56662 Len=948
21	0.077981010	192.168.1.60	224.0.0.1	UDP	990	58819 → 56662 Len=948
22	0.081992873	192.168.1.60	224.0.0.1	UDP	990	58819 → 56662 Len=948
23	0.083313532	192.168.1.60	224.0.0.1	UDP	990	58819 → 56662 Len=948

LLS Pose

picoscan

EXAMPLE 13

2 x Scanners, LiDAR-LOC & 2nd App in IPC



Settings in SOPASair (picoScan GUI)

Measurement data output

Format	Compact
Protocol	UDP
Port	56662
Destination IP address	224.0.0.1
Start/stop data output	

Measurement data output

Format	Compact
Protocol	UDP
Port	56663
Destination IP address	224.0.0.1
Start/stop data output	

LiDAR settings in SOPASair

5 picoscan_front

Name	picoscan_front
Active	✓
Mounted Upside Down	

Interface

Interface configuration for the sensor.

Ip Address	192.168.1.60
Port	2111
Receiver Ip	manual
	224.0.0.1
Receiver Port	manual
	56662
Max Connection Attempts On Startup	automatic
Time Offset	automatic

3 picoscan_back (inactive)

Name	picoscan_back
Active	
Mounted Upside Down	

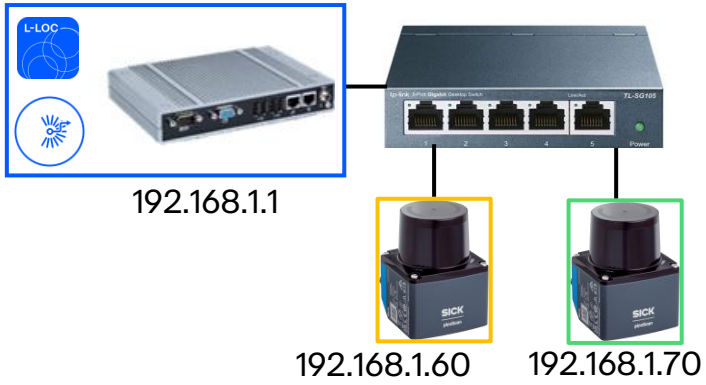
Interface

Interface configuration for the sensor.

Ip Address	192.168.1.70
Port	2111
Receiver Ip	manual
	224.0.0.1
Receiver Port	manual
	56663
Max Connection Attempts On Startup	automatic
Time Offset	automatic

EXAMPLE 13

2 x Scanners, LiDAR-LOC & 2nd App in IPC



PicoScan_front 192.168.1.60					
LiDAR-LOC	ipAddress	192.168.1.60	SOPASAir	Destination IP	224.0.0.1
	Port	2111		UDP port	56662
	receiverIP	224.0.0.1			
	receiverPort	56662			

PicoScan_back 192.168.1.70					
LiDAR-LOC	ipAddress	192.168.1.70	SOPASAir	Destination IP	224.0.0.1
	Port	2111		UDP port	56663
	receiverIP	224.0.0.1			
	receiverPort	56663			

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
28337	40.460812189	192.168.1.1	192.168.1.123	UDP	98	5008 → 5010 Len=56
28338	40.461092353	192.168.1.1	192.168.1.123	UDP	82	5008 → 5010 Len=40
28339	40.461310333	192.168.1.1	192.168.1.123	UDP	82	5008 → 5010 Len=40
28340	40.461530581	192.168.1.1	192.168.1.123	UDP	82	5008 → 5010 Len=40
28341	40.465975068	192.168.1.60	224.0.0.1	UDP	1500	53953 → 56662 Len=1458
28342	40.467990311	192.168.1.70	224.0.0.1	UDP	990	41485 → 56663 Len=948
28343	40.470468287	192.168.1.70	224.0.0.1	UDP	990	41485 → 56663 Len=948
28344	40.471326947	192.168.1.60	224.0.0.1	UDP	1500	53953 → 56662 Len=1458
28345	40.476478833	192.168.1.70	224.0.0.1	UDP	990	41485 → 56663 Len=948
28346	40.476689217	192.168.1.60	224.0.0.1	UDP	1500	53953 → 56662 Len=1458
28347	40.481415788	192.168.1.70	224.0.0.1	UDP	990	41485 → 56663 Len=948
28348	40.482315476	192.168.1.60	224.0.0.1	UDP	1500	53953 → 56662 Len=1458
28349	40.486558051	192.168.1.70	224.0.0.1	UDP	990	41485 → 56663 Len=948
28350	40.487579575	192.168.1.60	224.0.0.1	UDP	1500	53953 → 56662 Len=1458
28351	40.488120763	192.168.1.70	224.0.0.1	UDP	990	41485 → 56663 Len=948
28352	40.490602494	192.168.1.70	224.0.0.1	UDP	990	41485 → 56663 Len=948
28353	40.493223689	192.168.1.60	224.0.0.1	UDP	1500	53953 → 56662 Len=1458
28354	40.495145849	192.168.1.70	224.0.0.1	UDP	990	41485 → 56663 Len=948
28355	40.498579108	192.168.1.60	224.0.0.1	UDP	1500	53953 → 56662 Len=1458
28356	40.499055592	192.168.1.70	224.0.0.1	UDP	990	41485 → 56663 Len=948
28357	40.500563032	192.168.1.70	224.0.0.1	UDP	990	41485 → 56663 Len=948
28358	40.504178907	192.168.1.60	224.0.0.1	UDP	1500	53953 → 56662 Len=1458
28359	40.506604130	192.168.1.123	192.168.1.1	HTTP	678	POST /api/crown/ViewPro

LLS Pose

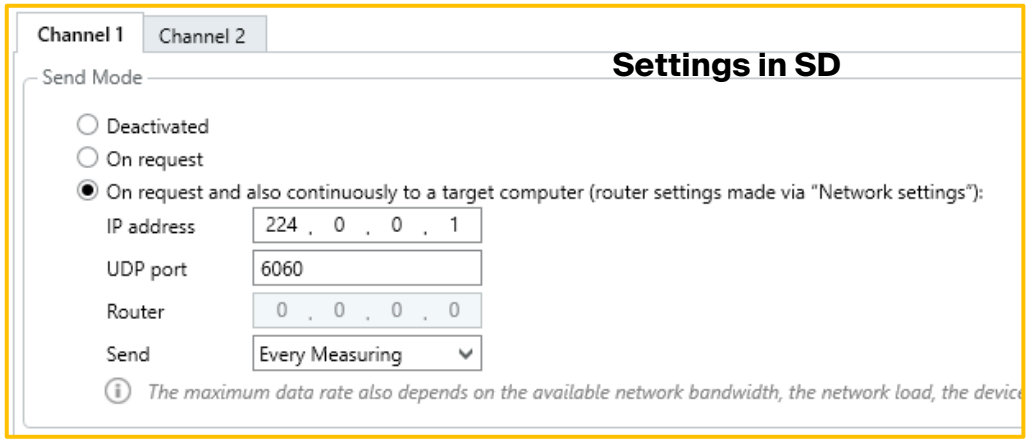
picoscan_front

picoscan_back

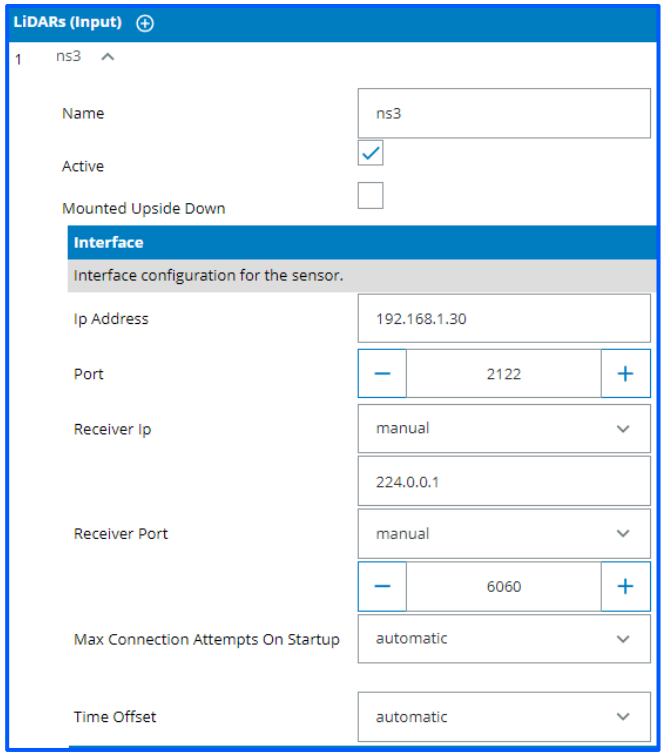
EXAMPLE 14

1 x Scanner (CH1 & CH2), LiDAR-LOC & 2nd App in IPC

Settings in SD

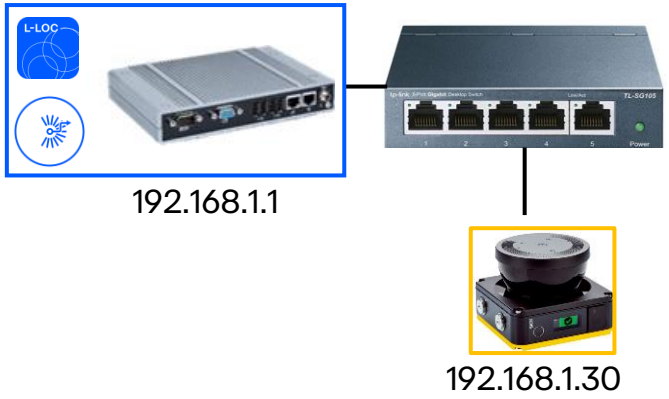


LiDAR settings in SOPASair

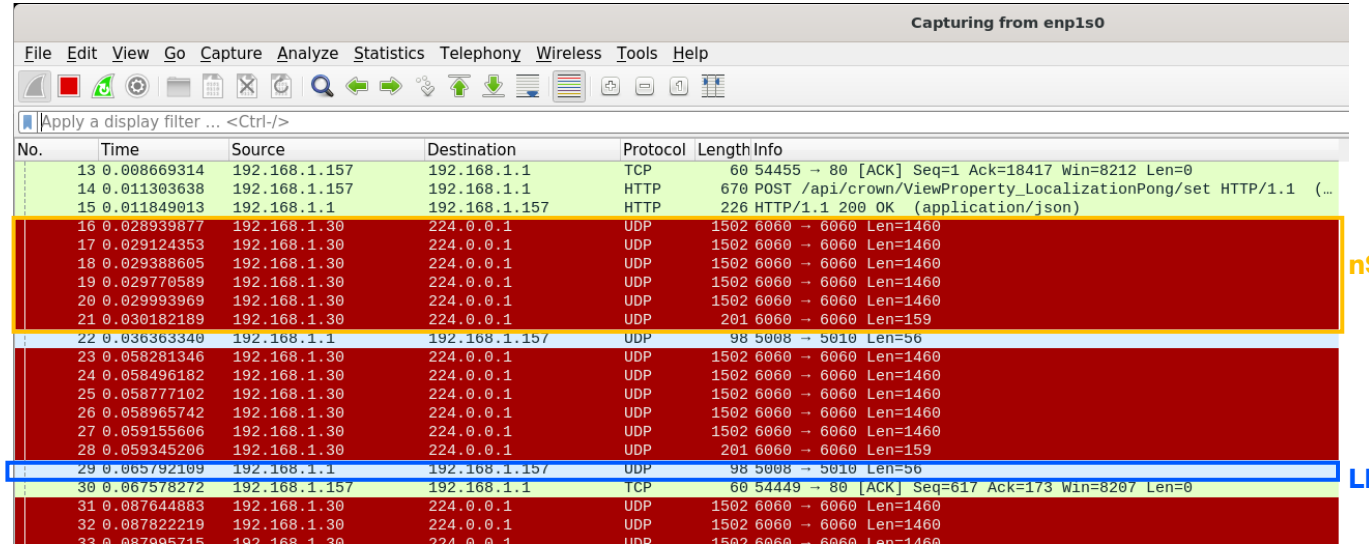


EXAMPLE 14

1 x Scanner (CH1 & CH2), LiDAR-LOC & 2nd App in IPC



nS3 front 192.168.1.40					
LiDAR-LOC	ipAddress	192.168.1.30	Safety Designer	Destination IP	224.0.0.1
	Port	2122		UDP port	6060
	receiverIP	224.0.0.1			
	receiverPort	6060			



nS3_front

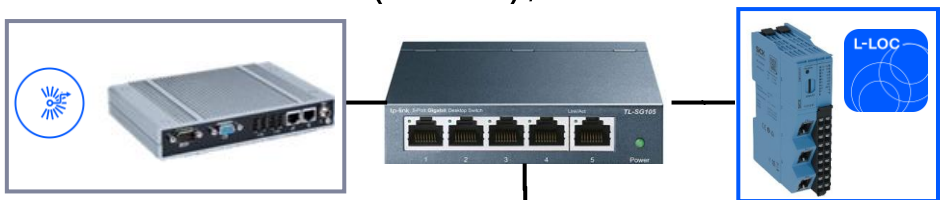
LLS Pose

03

DATA TRANSMISSION EXAMPLES TCP SCANNER

EXAMPLE 15

1 x Scanner (TCP), LiDAR-LOC in SIM & 2nd App in IPC



SOPAS ET settings

General

Addressing Mode: Static — Apply

IP-Address: 192 . 168 . 1 . 90

Subnet-Mask: 255 . 255 . 255 . 0

Default Gateway: 0 . 0 . 0 . 0

Speed: Auto — Negotiated: 100Mb Full Duplex

MAC-Address: 00 -06 -77 -2E -51 -32

Ethernet Host Port

CoLa Dialect: CoLa ASCII

Server / Client: Server — IP-Port: 2112

LiDAR settings in SOPASair

1 Tim571

Name: Tim571

Active: ☑

Mounted Upside Down: ☐

Interface

Interface configuration for the sensor.

Ip Address: 192.168.1.90

Port: 2111

Receiver Ip: automatic

Receiver Port: automatic

Max Connection Attempts On Startup: automatic

Time Offset: automatic

EXAMPLE 15

1 x Scanner (TCP), LiDAR-LOC in SIM & 2nd App in IPC

TiM571 192.168.1.90					
LiDAR-LOC	ipAddress	192.168.1.90	Sopas ET	Destination IP	unnecessary
	Port	2111		port	default
	receiverIP	default			
	receiverPort	default			

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
940	1.966487471	192.168.1.123	192.168.1.1	TCP	60	62306 → 80 [ACK] Seq=1 Ack=1972 Win=8209 Len=0
941	1.966528163	192.168.1.123	192.168.1.1	TCP	60	54631 → 3389 [ACK] Seq=1 Ack=2915906 Win=8246 Len=0
942	1.966557695	192.168.1.123	192.168.1.1	TCP	60	62164 → 80 [ACK] Seq=8041 Ack=2237 Win=8207 Len=0
943	1.979251892	192.168.1.90	192.168.1.1	TCP	1514	2111 → 37953 [ACK] Seq=95235 Ack=1 Win=362 Len=1448
944	1.979304380	192.168.1.90	192.168.1.1	TCP	1514	2111 → 37953 [ACK] Seq=96683 Ack=1 Win=362 Len=1448
945	1.979351348	192.168.1.90	192.168.1.1	TCP	446	2111 → 37953 [PSH, ACK] Seq=98131 Ack=1 Win=362 Len=
946	1.979462900	192.168.1.1	192.168.1.90	TCP	66	37953 → 2111 [ACK] Seq=1 Ack=98511 Win=714 Len=0 TSv
947	1.994668445	192.168.1.10	192.168.1.1	UDP	98	5008 → 5010 Len=56
948	2.045750728	192.168.1.90	192.168.1.1	TCP	1514	2111 → 37953 [ACK] Seq=98511 Ack=1 Win=362 Len=1448
949	2.045801128	192.168.1.90	192.168.1.1	TCP	1514	2111 → 37953 [ACK] Seq=99959 Ack=1 Win=362 Len=1448
950	2.045848276	192.168.1.90	192.168.1.1	TCP	444	2111 → 37953 [PSH, ACK] Seq=101407 Ack=1 Win=362 Le
951	2.045963584	192.168.1.1	192.168.1.90	TCP	66	37953 → 2111 [ACK] Seq=1 Ack=101785 Win=714 Len=0 TS
952	2.053993358	192.168.1.1	192.168.1.123	TCP	105	80 → 62306 [PSH, ACK] Seq=1972 Ack=1 Win=501 Len=51
953	2.054211170	192.168.1.1	192.168.1.123	TCP	16700	80 → 62307 [PSH, ACK] Seq=123265 Ack=1 Win=501 Len=1
954	2.054337806	192.168.1.1	192.168.1.123	TCP	248	80 → 62306 [PSH, ACK] Seq=2023 Ack=1 Win=501 Len=194
955	2.054820194	192.168.1.123	192.168.1.1	TCP	60	62307 → 80 [ACK] Seq=1 Ack=136405 Win=8212 Len=0
956	2.054893622	192.168.1.123	192.168.1.1	TCP	60	62307 → 80 [ACK] Seq=1 Ack=139911 Win=8212 Len=0
957	2.054982698	192.168.1.123	192.168.1.1	TCP	60	62306 → 80 [ACK] Seq=1 Ack=2217 Win=8208 Len=0
958	2.059499401	192.168.1.123	192.168.1.1	HTTP	672	POST /api/crown/ViewProperty_LocalizationPong/set HT
959	2.060195125	192.168.1.1	192.168.1.123	HTTP	226	HTTP/1.1 200 OK (application/json)

LLS Pose

TiM571

04

ROS DRIVER CONFIGURATIONS

ROS DRIVERS

sick_lidar_localization.launch example configuration

```
<launch>
  <!-- Launch sick_lidar_localization -->
  <arg name="hostname"    default="127.0.0.1"/>        <!-- IP address of the localization controller -->
  <arg name="serverpath" default="api"/>               <!-- Relative path to the rest api, i.e. url of rest requests is "http://<hostname>/<serverpath>/" -->
  <arg name="verbose"     default="0"/>                <!-- If verbose>0: print informational messages, otherwise silent except for error messages -->
  <arg name="udp_ip_lls_output" default=""/>           <!-- IP address for output UDP messages, or "" for broadcast (INADDR_ANY), default: "", use IP address of your local machine -->
  <arg name="udp_ip_lls_input" default="127.0.0.1"/>  <!-- IP address for input UDP messages, or "" for broadcast, default: "192.168.0.1", (IP adress of the localization controller) -->
  <arg name="udp_port_lls_input" default="5009"/>      <!-- UDP port of input messages -->
  <arg name="udp_lls_input_source_id" default="21"/>   <!-- Source id of UDP input messages (e.g. source ID of odom sender), has to match the ID in the localization controller configuration
  <arg name="udp_port_lls_output" default="5010"/>     <!-- UDP port of output messages -->
  <arg name="udp_lls_output_logfile" default=""/>      <!-- Optional logfile for human readable UDP output messages, default: "" (no outputlogfile) -->
  <arg name="software_pll_fifo_length" default="7"/>   <!-- Length of fifo in SoftwarePLL -->
  <arg name="odom_topic" default="/odom"/>             <!-- Topic of ros odom messages -->
  <arg name="ros_odom_to_udp_msg" default="3"/>        <!-- Convert ros odom message to udp: -->
```

NOTE: This launch file is configured for a device running LiDAR-LOC and ROS on the same device.
If the user an external device to run LiDAR-LOC, the “hostname”, “udp_ip_lls_output” and “udp_ip_lls_input” will need to be modified according to the corresponding IP addresses.

[GitHub - SICKAG/sick_lidar_localization · GitHub]

ROS DRIVERS

sick_safetyscanners.launch example configuration

UDP port needs to be specified. Port needs to fit the Safety Designer setting.

LiDAR-LOC can bin only to **Channel 1** (Safety Designer), **Channel 2** (Safety Designer) is for other application.

ROS is so-called "zero leading".

→ **Channel 1** of Safety Designer is **channel 0** in ROS

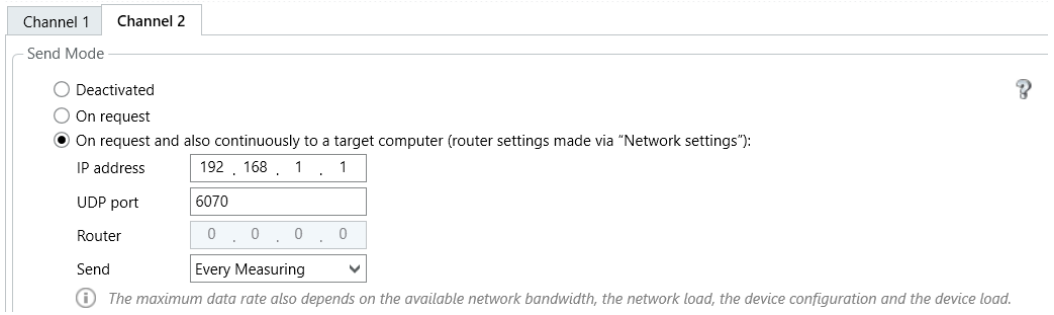
→ **Channel 2** of Safety Designer is **channel 1** in ROS

NOTE: For each safety sensor used, a separate sick_safetyscanner launch file and node needs be created

This launch file is related to examples 6 -8 on this guide.

[GitHub - SICKAG/sick_safetyscanners]

```
<launch>
  <arg name="sensor_ip"              default="192.168.1.20" />
  <arg name="host_ip"                default="192.168.1.1" />
  <arg name="interface_ip"           default="0.0.0.0" />
  <arg name="host_udp_port"          default="6070" />
  <arg name="frame_id"               default="map" />
  <arg name="skip"                   default="0" />
  <arg name="angle_start"            default="0" />
  <arg name="angle_end"              default="0" />
  <arg name="time_offset"            default="0.0" />
  <arg name="min_intensities"        default="0.0" doc="minimal intensity for a laserscan point" />
  <arg name="channel"                default="1" />
  <arg name="channel_enabled"        default="True" />
  <arg name="general_system_state"   default="True" />
  <arg name="derived_settings"       default="True" />
  <arg name="measurement_data"       default="True" />
  <arg name="intrusion_data"         default="True" />
  <arg name="application_io_data"    default="True" />
  <arg name="use_persistent_config"  default="False" />
```



ROS DRIVERS

sick_safetyscanners.launch example configuration for multicasting

Specifying multicast IP address.

UDP port needs to be specified. Port needs to fit the Safety Designer setting.

LiDAR-LOC can bin only to **Channel 1** (Safety Designer), **Channel 2** (Safety Designer) is for other application.

ROS is so-called "zero leading".

→ **Channel 1** of Safety Designer is **channel 0** in ROS

→ **Channel 2** of Safety Designer is **channel 1** in ROS

NOTE:

→ For each safety sensor used, a separate sick_safetyscanner launch file and node needs be created

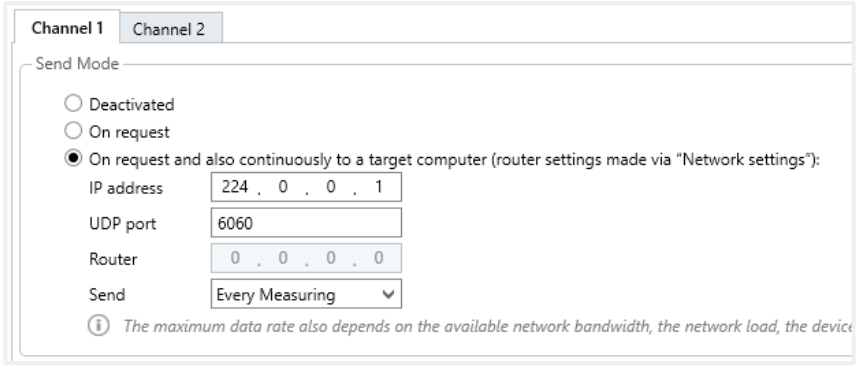
→ Depending on the IPC's specifications, skipping every 2nd second scan may be necessary to avoid crashes on the ROS driver.

This launch file is meant to use SICK Safety Scanners' **multicasting** function.

This launch file is related to **example 12** on this guide.

[GitHub - SICKAG/sick_safetyscanners]

```
<launch>
  <arg name="sensor_ip"          default="192.168.1.20" />
  <arg name="host_ip"            default="192.168.1.1" />
  <arg name="interface_ip"       default="0.0.0.0" />
  <arg name="host_udp_port"      default="6070" />
  <arg name="frame_id"           default="map" />
  <arg name="skip"               default="1" />
  <arg name="angle_start"        default="0" />
  <arg name="angle_end"          default="0" />
  <arg name="time_offset"        default="0.0" />
  <arg name="min_intensities"    default="0.0" doc="minimal intensity for a laserscan point" />
  <arg name="channel"            default="1" />
  <arg name="channel_enabled"    default="True" />
  <arg name="general_system_state" default="True" />
  <arg name="derived_settings"   default="True" />
  <arg name="measurement_data"   default="True" />
  <arg name="intrusion_data"     default="True" />
  <arg name="application_io_data" default="True" />
  <arg name="use_persistent_config" default="False" />
```



ROS DRIVERS

sick_safetyscanners – multicasting modification

```
#include <sick_safetyscanners/communication/AsyncUDPClient.h>

                // Keep io_service busy
m_io_work_ptr = std::make_shared<boost::asio::io_service::work>(m_io_service);
try
{
 auto endpoint = boost::asio::ip::udp::endpoint(boost::asio::ip::udp::v4(), local_port);
 m_socket_ptr  = std::make_shared<boost::asio::ip::udp::socket>(m_io_service);
 m_socket_ptr->open(boost::asio::ip::udp::v4());
 m_socket_ptr->set_option(boost::asio::ip::udp::socket::reuse_address(true));
 m_socket_ptr->bind(endpoint);
 m_socket_ptr->set_option(boost::asio::ip::multicast::join_group(host_ip, interface_ip));

}

catch (const std::exception& e)
```

In AsyncUDPClient.cpp make the following changes:

→ Replace the code from line 81 to 89, with the code shown on this slide.

→ This change on the ROS driver corresponds to example 12 on this guide (nS3 with multicasting).

Note: This change is for ROS1

GitHub - SICKAG/sick_safetyscanners

ROS DRIVERS

sick_safetyscanners2 – multicasting modification

```
diff --git a/src/communication/UDPClient.cpp b/src/communication/UDPClient.cpp
index b2037e1..08881ea 100644
--- a/src/communication/UDPClient.cpp
+++ b/src/communication/UDPClient.cpp
@@ -76,7 +76,7 @@ UDPClient::UDPClient(boost::asio::io_service& io_service,
                      boost::asio::ip::address_v4 host_ip,
                      boost::asio::ip::address_v4 interface_ip)
   : m_io_service(io_service)
-  , m_socket(io_service, boost::asio::ip::udp::endpoint{boost::asio::ip::udp::v4(), server_port})
+  , m_socket(io_service)
   , m_packet_handler()
   , m_recv_buffer()
   , m_deadline(io_service)
@@ -86,6 +86,10 @@ UDPClient::UDPClient(boost::asio::io_service& io_service,
     LOG_ERROR("Multicast IP specified, however the interface IP is undefined.");
     exit(-1);
   }
+  auto endpoint = boost::asio::ip::udp::endpoint{boost::asio::ip::udp::v4(), server_port};
+  m_socket.open(boost::asio::ip::udp::v4());+  m_socket.set_option(boost::asio::ip::udp::socket::reuse_address(true));
+  m_socket.bind(endpoint);
   m_socket.set_option(boost::asio::ip::multicast::join_group(host_ip, interface_ip));

   m_deadline.expires_at(boost::posix_time::pos_infin);
```

Install sick_safetyscanners2 as shown on the git repository.

After installation, the git patch shown on the slide needs to be installed on [GitHub - SICKAG/sick_safetyscanners_base]

Please contact your SICK partner to obtain this patch file since copying and pasting this text may cause formatting issues.

Note: This change is for ROS2

[GitHub - SICKAG/sick_safetyscanners2]

ROS DRIVERS

sick_scan_xd: sick_picoscan.launch - example configuration for multicasting

```
<launch>

    <!-- Launch sick_picoscan -->
    <!-- env name="ROSCONSOLE_CONFIG_FILE" value="/tmp/rosconsole_loglevel_warn.conf" / -->
    <arg name="hostname" default="192.168.1.70"/>  <!-- IP address of picoScan150, overwrites default ip address "192.168.0.1" in sick_scansegment_xd.yaml -->
    <arg name="udp_receiver_ip" default="224.0.0.1"/>        <!-- UDP destination IP address (ip address of udp receiver), overwrites default in sick_scansegment_xd.yaml -->
    <arg name="nodename" default="sick_picoscan1"/>
    <arg name="publish_frame_id" default="world" />                              <!-- frame id of ros Laserscan messages, default: "world_<layer-id>" = "world_1" -->
    <arg name="publish_laserscan_segment_topic" default="scan_segment" />        <!-- topic of ros Laserscan segment messages -->
    <arg name="publish_laserscan_fullframe_topic" default="scan_fullframe" />    <!-- topic of ros Laserscan fullframe messages -->
    <arg name="add_transform_xyz_rpy" default="0,0,0,0,0,0"/>
—
    <param name="udp_port" type="int" value="56662" />
—
   <param name="host_FREchoFilter" type="int" value="1" />
```

NOTE: This launch file is configured for a device running LiDAR-LOC and ROS on the same device.

To activate the multicast option for this ROS driver, use the assigned IP address “224.0.0.1” and UDP port “56661”.

If two picoScans are used, two launch files must be created and “*nodename*” must be re-labeled, as shown on this example.

An additional modification has to be done in the udp_sockets.h file, which will be shown on the following slide.

This ROS driver example is related to example 11 in this guide.

GitHub - SICKAG/sick_scan_xd

ROS DRIVERS

sick_scan_xd – multicasting modification

```
#include "sick_scan/sick_scan_base.h" /* Base definitions included in all header files, added by add_sick_scan_base_header.py. Do not edit this line. */

                // #endif
                // setsockopt(m_udp_socket, SOL_SOCKET, SO_BROADCAST, &broadcast_opt, sizeof(broadcast_opt));
                int reuse_addr_opt = 1;
                setsockopt(m_udp_socket, SOL_SOCKET, SO_REUSEADDR, &reuse_addr_opt, sizeof(reuse_addr_opt));
                struct sockaddr_in sim_servaddr = { 0 };

                    {
                        ROS_ERROR_STREAM("## ERROR UdpSenderSocketImpl::init(" << server_address << ":" << udp_port << "): setsockopt(SO_BROADCAST) failed, error: " << getErrorMessage());
                    }
                    int reuse_addr_opt = 1;
                    setsockopt(m_udp_socket, SOL_SOCKET, SO_REUSEADDR, &reuse_addr_opt, sizeof(reuse_addr_opt));
```

In udp_sockets.h make the following changes:

→ Uncomment “ setsockopt(m_udp_socket, SOL_SOCKET, SO_REUSEADDR, … ; ” (line 153) and add a reuse flag on the previous line: int reuse_addr_opt = 1;

→ Uncomment “ setsockopt(m_udp_socket, SOL_SOCKET, SO_REUSEADDR, &….; ” (line 276) and add a reuse flag on the previous line: int reuse_addr_opt = 1;

Note: This solution is valid for ROS1 and ROS2

GitHub - SICKAG/sick_scan_xd

THANK YOU

→ SICK.COM