Saab TransponderTech AB

Appendices

Project designation

R5 RIC

Document title

R5 RIC Software Interface Specification

Distribution

Saab TransponderTech AB

DOCUMENT VALIDATION

	Name	Dept. Code/ Function	Date/ Signature
Prepared	Marcus Malm	PT-MM	2016-09-06
Reviewed			
Approved	Christian Andersson		

DOCUMENT CHANGE RECORD

Issue	Change order	Date	Pages affected	Motive for and information on the change
A		2016-09-06	All	New document
B		2017-06-25	12, 13, 16, 20	

CONTENTS

1 Bibliography 4
2 Introduction 5
3 Device configuration 6
3.1 Serial protocol 6
3.1.1 Commands 6
3.1.2 Responses 7
3.2 Network protocol 7
3.2.1 Commands 7
3.2.2 Response 8
3.2.3 TCP session example 8
4 Parameters 10
4.1 Data types 10
4.2 Parameters by category 10
4.2.1 Azimuth front end 10
4.2.2 Video front end 11
4.2.3 Acquisition 11
4.2.4 Pulse compression 13
4.2.5 FTC filtering 13
4.2.6 Pulse integration 13
4.2.7 System configuration 14
4.2.8 System status 14
4.2.9 Narrow video status 14
4.2.10 Radar configuration 15
4.2.11 Package configuration 17
4.2.12 Package configuration, Asterix package 17
4.2.13 Package configuration, STTv2RAW package 18
4.2.14 Package configuration, STTv2 package 19
5 Digital video interfaces 20
5.1 Radar data in 20
5.1.1 Terma Scanter Network Video 20
5.2 Radar data out 20
5.2.1 RAW data 20
5.2.2 Narrow data 21

1 BIBLIOGRAPHY

1. **Terma A/S.** SCANTER Network Video Protocol. *304124-SI.* Lystrup, Denmark : s.n., 04 09 2015. Rev. H.

2 INTRODUCTION

This document describes the software interfaces between R5 RIC and the radar tracker.
There are two main categories of interfaces; configuration interfaces and radar data interfaces.

Chapter 3 describes the configuration interfaces. How to connect, commands and responses.

Chapter 4 lists all configuration parameters organized by function.

Chapter 5 gives an account of the radar video interfaces protocols.

3 DEVICE CONFIGURATION

R5 RIC can be configured through two different interfaces; RS-232 and Ethernet. The following chapters will describe the protocols for each of these interfaces.

3.1 Serial protocol

The following settings should be used to connect to the R5 RIC through the serial interface:

Parameter	Value
Baud rate	115200
Data bits	8
Stop bits	1
Parity	None
Hardware control	None

The serial protocol is an interactive prompt. The prompt takes ASCII commands that are terminated with CR-LF. The serial prompt also have some support for VT100.

- Arrow up, down - Command history.
- Array left, right - Command editing.
- Delete and backspace - Command editing.

The input prompt is marked by a greater then sign followed by a space. “> ”.

3.1.1 Commands

Following commands are available to enter over the serial link.

Command	Meaning
help <command>	Show general help or help for the provided command.
version	Returns version information.
start	Start raw radar stream
stop	Stop raw radar stream
list	Get the name and value of all configuration parameters.
get <name>	Get the value of the configuration parameter.
set <name> <value>	Set the named configuration parameter to the provided value.
store	Store configuration. Stores the current configuration to permanent configuration storage.
revert	Revert the current configuration to the stored one.
reset storage	Restore the configuration storage to the default configuration.
system reboot	Reboot the hardware.

The configuration parameters are described in chapter 4.

3.1.2 Responses

The serial service will respond with the result of the operation or a status indicator. Following status indicators exists.

Indicator	Meaning
OK	Command executed without problems.
ERROR	Service failed.
General error	Underlying configuration service failed.
Invalid operation	The operation sent is not a valid operation.
Invalid data	Failed to decode data.
Invalid value	The value sent is not valid.
Invalid type	The value sent is of invalid type.
Session expired	The configuration session expired.
Unknown error <id>	An unknown error has occurred.

3.1.3 Serial session example

Command	Response
version	"R5 RIC x.x.x (xxx)"
store	"Store succeded."
set raiko.net.src.port 1200	"Set raiko.net.src.port : OK"
get raiko.net.src.port	"raiko.net.src.port : 120"
set raiko.net.src.port 0	"Set raiko.net.src.port : Invalid value"

3.2 Network protocol

The device is configured via a text interface over TCP/IP, over the 100Mbit configuration connection. Each command consists of a single line of data, terminated by a CR-LF character pair, fields are delimited with space. The response is also a single line of text. The response always begins with a status indication followed by the command result. The response is also terminated with CR-LF.

The configuration service responds on TCP/IP Port 45000.

3.2.1 Commands

The following commands are available:

Command	Meaning
init	Initializes the device, returns the version information.
start	Start raw radar stream.
stop	Stop raw radar stream.
ping	Check the connection.
list	Receive a list of parameters to operate on.
set <parameter> <value>	Set value of <parameter> to <value> where <value> must conform to the type of parameter.

get <parameter>	Get value of parameter <parameter>.
store	Store parameter values to storage.
revert	Revert parameter values from current to stored.
reset	Reset stored configuration to default.
reboot	Reboot the hardware.

The configuration parameters are described in chapter 4.

3.2.2 Response

All response messages begin with a status indicator. The following status indicators exist.

Indicator	Meaning
OK	Command executed without problems. Result follows.
ERROR	Service failed.
General error	Underlying configuration service failed.
Invalid operation	The operation sent is not a valid operation.
Invalid data	Failed to decode data.
Invalid value	The value sent is not valid.
Invalid type	The value sent is of invalid type.
Session expired	The configuration session expired.
Unknown error <id>	An unknown error has occurred.

After the status indicator the command result is sent. The following table describes the response excluding the status indicator for above commands.

Command	Response
init	Version string
start	Status
stop	Status
ping	Status
list	Array of value names separated by space
set <parameter> <value>	Status
get <parameter>	Parameter value conforming to the parameter data type.
store	Status
revert	Status
reset	Status

3.2.3 TCP session example

Command	Response
init	OK SAAB R5RIC x.x (xxx)

Saab TransponderTech AB

ping	OK
set raiko.net.src.port 1200	OK
get raiko.net.src.port	OK 1200
set raiko.net.src.port 0	Invalid value
start	OK

4 PARAMETERS

This chapter describes the configuration parameters that can be set using the protocols described earlier.

4.1 Data types

These are the data types that parameters can have.

Data type	Description
Byte	8 bit value
Uint16	16-bit unsigned integer in ASCII representation
Int16	16-bit signed integer in ASCII representation
Bool	Boolean variable represented by "true" or "false"
Double	64-bit floating point number in c-style text representation (ex: 2.3, 0.12, .1, 1.4e-5)
Uint32	32-bit unsigned integer in ASCII representation
Int32	32-bit signed integer in ASCII representation
IP	String formatted as an IP address in dotted decimal representation (ex: “10.11.162.1”)
String	ASCII text.

4.2 Parameters by category

These tables describe the parameters available. All parameter names are in lowercase/underscore/dot format.

4.2.1 Azimuth front end

Name	Type	RW	Description
raiko.azimuth.acp.differential	Bool	RW	Enables/disables differential ACP input.
raiko.azimuth.acp.ok	Bool	R	ACP signal received during the last second.
raiko.azimuth.acp.termination	Uint16	RW	The ACP input termination. One of 0, 120, 75 or 65535. 0 is no termination and 65535 is high voltage.
raiko.azimuth.arp.differential	Bool	RW	Enables/disables differential ACP input.
raiko.azimuth.arp.ok	Bool	R	ARP signal received during the last ten seconds.
raiko.azimuth.arp.termination	Uint16	RW	The ACP input termination. One of 0, 120, 75 or 65535. 0 is no termination and 65535 is high voltage.
raiko.azimuth.max	Uint16	RW	Maximum azimuth per revolution. Used for raiko.azimuth.underflow and raiko.azimuth.overflow indicators.
raiko.azimuth.measured	Uint16	R	Measured azimuth per revolution.
raiko.azimuth.offset	Int16	RW	Azimuth correction in degrees
raiko.azimuth.overflow	Bool	R	Azimuth was greater than raiko.azimuth.max during the last revolution.
raiko.azimuth.source	Uint16	RW	Azimuth source. 0 = ARP/ACP, 1 = Syncro.
raiko.azimuth.syncro.bits	Uint16	RW	
raiko.azimuth.underflow	Bool	R	Azimuth was less than raiko.azimuth.max during the last revolution.

raiko.rpm	Double	R	Revolutions per minute. Calculated using the measured time between ARP pulses.
raiko.revision		R	R5 RIC software revision number
raiko.version		R	R5 RIC software version number

4.2.2 Video front end

Name	Type	RW	Description
raiko.video.analog.differential	Bool	RW	Enables/disables differential analog video input.
raiko.video.analog.1.gain	Uint16	RW	Analog video I input analog gain.
raiko.video.analog.1.offset	Uint16	RW	Analog video I input analog offset.
raiko.video.analog.termination	Uint16	RW	Analog video input termination [Ω]. One of 50, 75 or 0 where 0 is no termination.
raiko.video.analog.2.gain	Uint16	RW	Analog video Q input analog gain.
raiko.video.analog.2.offset	Uint16	RW	Analog video Q input analog offset.
raiko.video.sync.differential	Bool	RW	Enables/disables differential analog video sync input.
raiko.video.sync.edge	Uint16	RW	Triggers acquisition on either 1=rising or 0=falling edge of sync pulse.
raiko.video.sync.ok	Bool	R	SYNC signal received during the last second.
raiko.video.sync.termination	Uint16	RW	Analog video sync input termination [Ω]. One of 0, 120, 75 or 65535. 0 is no termination and 65535 is high voltage.
raiko.prf	Double	R	Pulse repetition frequency. Calculated using the measured time between SYNC pulses.

The analog video offset (raiko.video.analog.x.offset) is applied before the analog video gain (raiko.video.analog.x.gain) which means that when adjusting the offset the gain may change. It is recommended to adjust the offset first.

4.2.3 Acquisition

Sampling radar echoes into packages. The acquisition feeds the radar echoes into two data flows, Raw and Narrow, which in turn sends packages on Radar Video Ethernet and Configuration Ethernet respectively.
Raw data contains the full video resolution as configured. Narrow (also called “DSP”) consists of a low resolution version of the Raw data.

Name	Type	RW	Description
raiko.datatype	Uint16	RW	Radar sample output bit width, either one of 4, 8 or 16.
raiko.frequency	Int32	RW	Radar echo sample frequency. 10000 – 100000000 Hz.
raiko.downsample.operation	Uint16	RW	The operation used when down sampling to the target frequency. • 0 = Last, use the last sample • 1 = Max, use the maximum sample.
raiko.offset	Uint16	RW	Radar echo sample offset. Should be 32767.
raiko.invert	Bool	RW	Radar echo sample invert.

raiko.net.mac	String	RW	Radar network interface MAC address.
raiko.net.dst.port	Uint16	RW	Radar network interface destination UDP port. From 1 to 65535.
raiko.net.ipv4address	IP	RW	Radar network interface IPV4 address.
raiko.net.src.port	Uint16	RW	Radar network interface source UDP port. From 1 to 65535.
raiko.samples	Uint16	RW	Numbers of samples for each radar echo. 64 – 8192.
raiko.sector.downsample.count	Uint16	RW	The resulting number of sectors per revolution. Requires sector down sampling enable to take effect.
raiko.sector.downsample.enable	Bool	RW	Enable sector down sampling.
raiko.sector.downsample.operation	Uint16	RW	The operation used when down sampling into sectors. • 0 = Last, use the last sample • 1 = Max, use the maximum sample.
raiko.acquisition.raw.functions.echo_downsample	Bool	RW	Use sector down sampling in raw video flow.
acquisition.pulse_compression.enable	Bool	RW	Enable pulse compression.
raiko.acquisition.dsp.functions.echo_downsample	Bool	RW	-
raiko.acquisition.echoes.dsp	Uint32	R	-
raiko.acquisition.echoes.echo_downsample	Uint32	R	-
raiko.acquisition.echoes.raw	Uint32	R	-
raiko.acquisition.echoes.sample_downsample	Uint32	R	-
raiko.acquisition.range_offset	Double	RW	Range offset.
raiko.board	Uint16	R	R5 RIC hardware revision number
raiko.build	Uint16	R	R5 RIC software build number
raiko.dsp.echoes.bus	Uint32	R	
raiko.net.echoes.in	Uint32	R	Number of radar echoes collected per second.
raiko.net.echoes.stt	Uint32	R	Number of generated STT packages sent per second.
raiko.net.echoes.sttv2raw	Uint32	R	Number of generated STTv2 packages sent per second.
raiko.net.echoes.used	Uint32	R	Number of echoes per second used for output.

The acquisition starts at the edge of the SYNC signal.

The acquisition frequency and number of samples dictates how long the acquisition time will be. Following example calculates the acquisition time.

$$f_s = 10\ MHz$$

$$N = 2048$$

$$t_{\text{acquisition}} = \frac{1}{f_s} \times N = 204.8\ \mu s$$

This is related to the pulse repetition frequency (PRF) of the radar. A new echo won't be captured even if new SYNC pulses appear until this acquisition time has elapsed. This means that the acquisition time must be less than the PRF or the acquisition will skip echoes.

$$PRF_{MAX} = \frac{1}{t_{\text{acquisition}}} = 4.882\ KHz$$

Moreover, when running at maximum acquisition frequency (100 MHz) and full sample resolution (16-bit) care must be taken not to saturate the acquisition process. At this acquisition configuration the amount of data generated is 1.6 Gbit/s which is more that the Ethernet link (1 Gbit/s) can process. At these speeds the acquisition time must be about half of the PRF.

Sector down sampling takes multiple echoes and merge them into one sector. The number of echoes per sector is a function of the number of echoes per revolution and the configured sector count i.e.

$$N_{sector} = 200$$

$$N_{echo} = 4096$$

$$Sector\ Divider = \frac{N_{Echo}}{N_{Sector}} = 20.48$$

4.2.4 Pulse compression

Pulse compression requires both video inputs to receive I and Q data.

Name	Type	RW	Description
acquisition.pulse_compression.enable	Bool	RW	Enable pulse compression.
acquisition.pulse_compression.code.step	Uint8	RW	Set the code step length. 1-16.
acquisition.pulse_compression.code.select	Uint8	RW	Select which code to use; 1-8 for 64-bit codes.
acquisition.video.channel	Int16	RW	Selection of receiver channel to use for radar video processing. Both channels are used when pulse compression is enabled.

4.2.5 FTC filtering

FTC filtering is an edge detect function based the subtraction of an averaged signal from the original signal.

Name	Type	RW	Description
acquisition.ftc.enable	Bool	RW	Enable FTC filtering.
acquisition.ftc.raw.level	Uint8	RW	Set number of positions in moving average filter (part of FTC filter). Range: 3-9 correlates to values 4, 8, 16 …512.
acquisition.ftc.raw.gain	Uint8	RW	Set gain in FTC filter. Range: 0-7 correlates to values 1, 2, 4 …128.

4.2.6 Pulse integration

Pulse integration is used to integrate a number of signals in order to achieve better SNR.

Name	Type	RW	Description
acquisition.pulse_integration.enable	Bool	RW	Enable Pulse integration function.
acquisition.pulse_integration.level	Uint8	RW	Set number of integrated pulses; 2-8 pulses.

4.2.7 System configuration

Name	Type	RW	Description
cfg.web.access	Bool	RW	Enables or disables configuration possibility of the HTTP page. I.e. enables or disables the Set and Get functions.
sys.net.mac	String	R	Ethernet control interface MAC address. Requires commit or reboot.
sys.net.ipv4.address	IP	RW	Ethernet control interface IPV4 address. Requires commit or reboot.
sys.net.ipv4.netmask	IP	RW	Ethernet control interface IPV4 net mask. Requires commit or reboot.
sys.net.ipv4.gateway	IP	RW	Ethernet control interface IPV4 gateway. Requires commit or reboot.
sys.net.ipv4.nameserver.[1-3]	IP	RW	The address to up to 3 name servers.
sys.net.ipv4.ntpserver.[1-3]	IP	RW	The address to up to 3 NTP servers that the built in NTP client will connect to. The time is used to time-stamp the radar video.

4.2.8 System status

Name	Type	RW	Description
sys.temp.adc	Double	R	Temperature at ADC in degrees Celsius.
sys.temp.power	Double	R	Temperature at Power in degrees Celsius.
sys.time.date	String	R	System date.
sys.time.time	String	R	System time.
sys.time.uptime	Double	R	System uptime.
sys.identity.board	String	R	Board product number.
sys.identity.board.serial	Uint32	R	Board serial number.
sys.identity.product	String	R	Product number.
sys.identity.product.serial	Uint32	R	Product serial number.

4.2.9 Narrow video status

Name	Type	RW	Description
status.video.cpu.bus.index_errors	Int32	R	-
status.video.cpu.echoes	Int32	R	-
status.video.dsp.bus.index_errors	Int32	R	-
status.video.dsp.bus.index_outliers	Int32	R	-
status.video.dsp.bus.mark_errors	Int32	R	-
status.video.dsp.revolutions	Uint32	R	-
status.video.dsp.revolutions_since_error	Int32	R	-

4.2.10 Radar configuration

Name	Type	RW	Description
radar.kind	Uint16	RW	Radar manufacturer/model. 0 = not selected.

4.2.10.1 Radar configuration, Sperry Bridgemaster

radar.kind = 1

Name	Type	RW	Description
radar.bridgemaster.afc.enable	Bool	RW	Automatic Frequency Control
radar.bridgemaster.perfmon.enable	Bool	RW	Activation of Performance monitoring.
radar.bridgemaster.perfmon.tx	Bool	RW	Activation of Performance monitoring transmission.
radar.bridgemaster.swept.gain	Bool	RW	Swept gain
radar.bridgemaster.transmit	Bool	RW	Enables radar transmitter.
radar.bridgemaster.profile	Uint16	RW	Pulse length. 1 = Short, 2 = Medium, 3 = Long.
radar.bridgemaster.lo.tune	int32	RW	LO tune parameter.
radar.bridgemaster.perfmon.tune	int32	RW	Performance monitor LO tune
radar.bridgemaster.perfmon.xr	int32	RW	Performance monitor Xr value.
radar.bridgemaster.perfmon.xt	int32	RW	Performance monitor Xt value.
radar.bridgemaster.sector.blanking.[1-3].begin	Int16	RW	Sets the start angle 0-360 degrees for sector blanking (sector 1 of 3). The value “-1” disables sector blanking.
radar.bridgemaster.sector.blanking.[1-3].end	Int16	RW	Sets the end angle 0-360 degrees for sector blanking (sector 1 of 3). The value “-1” disables sector blanking.
radar.bridgemaster.bist.12v	Byte (7 bit value)	R	(BIST 1 msg)
radar.bridgemaster.bist.30v	Byte (7 bit value)	R	(BIST 2 msg)
radar.bridgemaster.bist.magnetron.current	Byte (7 bit value)	R	Magnetron current level. (BIST 3 msg)
radar.bridgemaster.bist.modulator.volts	Byte (7 bit value)	R	Magnetron voltage level. (BIST 4 msg)
radar.bridgemaster.bist.software	String	R	Version and revision. (BIST 5 msg)
radar.bridgemaster.config.status	Byte (5 bit value)	R	D4-D0 from Configuration msg. Containing: D4 (Bit 4) - P81 configuration (1) / non-P81 configuration (0) D3 (Bit 3) - S-Band (1) / X-Band (0) D2 (Bit 2) - 25kW (1) / 10 kW (0) D1 (Bit 1) -

			PM fitted D0 (Bit 0) - SART fitted [P81] BMstr (1) / Other (0) [non-P81]
radar.bridgemaster.error	Byte	R	D3-D0 from Error Tellback msg 1. Containing: D3 (Bit 7) - '0' D2 (Bit 6) - Spark Detect D1 (Bit 5) - Corrupt Data D0 (Bit 4) - Message Failure D3-D0 from Error Tellback msg 2. D3 (Bit 3) - '1' D2 (Bit 2) - Hmkr Fail D1 (Bit 1) - Charge Trig Fail D0 (Bit 0) - Mod Trig Fail
radar.bridgemaster.status	Byte	R	D3-D0 from Status 1 msg Containing: D3 (Bit 7) - Transmit on D2 (Bit 6) - Inhibit on D1 (Bit 5) - SP D0 (Bit 4) - MP D3-D0 from Status 2 msg Containing: D3 (Bit 3) - AFC on D2 (Bit 2) - Timer on D1 (Bit 1) - Wide-band D0 (Bit 0) - Swept Gain
radar.bridgemaster.tune.indicator	Byte (7 bit value)	R	Tune indicator value

4.2.10.2 Radar configuration, Simulator

This configuration is only applicable for STTv2 packages

radar.kind = 2

Name	Type	RW	Description
radar.simulator.prf	Uint16	RW	The simulated PRF in Hz.
radar.simulator.datatype	Uint16	RW	The sample data type. 4 = 4-bit, 8 = 8-bit, 16 = 16-bit.
radar.simulator.sample_count	Uint16	RW	Number of samples per echo.
radar.simulator.sector_count	Uint16	RW	Number of sectors per revolution.
radar.simulator.pattern	Uint16	RW	The pattern.
radar.simulator.pattern.arg.1	Uint16	RW	Pattern argument 1.
radar.simulator.pattern.arg.2	Uint16	RW	Pattern argument 2.

4.2.10.3 Radar configuration, ATM Air Seeker

radar.kind = 3

No exclusive parameters.

4.2.10.4 Radar configuration, GPS-100

radar.kind = 4

No exclusive parameters.

4.2.10.5 Radar configuration, SPS-95K

radar.kind = 5

No exclusive parameters.

4.2.10.6 Radar configuration, Terma Scanter 2202

radar.kind = 6

Name	Type	RW	Description
radar.terma.ctrl.cells.per.sweep	Uint16	R	Cells per strobe as reported by transceiver.
radar.terma.ctrl.connected	Bool	R	Connections status. (Transceiver <--> R5 RIC)
radar.terma.ctrl.ipv4.address	IP	RW	Ethernet control interface IPV4 address of the Scanter radar.
radar.terma.ctrl.range	Uint16	R	Range information as reported by transceiver.
radar.terma.ctrl.sweeps.per.scan	Uint16	R	Sweeps per scan as reported by transceiver.
radar.terma.ctrl.sweeps.per.sector	Uint16	R	Sweeps per sector as reported by transceiver.
radar.terma.ctrl.version	Double	R	Version number as reported by transceiver.

4.2.11 Package configuration

This section describes how to select and configure the resulting radar video package.

Name	Type	RW	Description
raiko.pkg.type	Uint16	RW	Radar video data package type, 0 = STT, 1 = STTv2RAW, 2 = Asterix
raiko.pkg.fragmentate	Bool	RW	Enable package fragmentation, applies to STT and STTv2RAW.

4.2.12 Package configuration, Asterix package

More information regarding the Asterix protocol supported by R5 RIC can be found in **7000 114-949, A1, R5 RIC Asterix Video Protocol Specification**.

Name	Type	RW	Description
raiko.pkg.asterix.sac	UInt8	RW	Asterix package SAC (System Area Code) value.
raiko.pkg.asterix.sic	UInt8	RW	Asterix package SIC (System Identification Code) value.

4.2.13 Package configuration, STTv2RAW package

These parameters are described in **7000 114-966, A1, R5 RIC STTv2 Video Protocol Specification**.

The field parameter is disabled by adding 16777216 (0x01000000) to the parameter value.

```
32324 => 0x7e44 => disable => 0x01007e44 => 16809540
```

When disabled, the field is removed from the package header.

If the value is marked with “Value substitution” the value will be substituted with a calculated or measured value in the software. Substituted fields can be disabled as described above.

Name	Type	RW	Description
raiko.pkg.sttv2raw.stream	UInt24	RW	Stream Id.
raiko.pkg.sttv2raw.start_bearing	UInt24	RW	Value substitution. First bearing angle recorded.
raiko.pkg.sttv2raw.end_bearing	UInt24	RW	Value substitution. Last bearing angle recorded.
raiko.pkg.sttv2raw.start_sector	UInt24	RW	Value substitution. Sector index for echo.
raiko.pkg.sttv2raw.range_samples	UInt24	RW	Value substitution. Number of samples in echo.
raiko.pkg.sttv2raw.sweep_time	UInt24	RW	Configurable value.
raiko.pkg.sttv2raw.transmission_mode	UInt24	RW	Configurable value.
raiko.pkg.sttv2raw.video_beam_select	UInt24	RW	Configurable value.
raiko.pkg.sttv2raw.video_type_select	UInt24	RW	Configurable value.
raiko.pkg.sttv2raw.tx_beam_shape	UInt24	RW	Configurable value.
raiko.pkg.sttv2raw.antenna_rotation	UInt24	RW	Configurable value.
raiko.pkg.sttv2raw.bin_size	UInt24	RW	Configurable value.
raiko.pkg.sttv2raw.rx_beam_elevation	UInt24	RW	Configurable value.
raiko.pkg.sttv2raw.date	UInt24	RW	Date
raiko.pkg.sttv2raw.time	UInt24	RW	Time
raiko.pkg.sttv2raw.milliseconds	UInt24	RW	Time [ms]
raiko.pkg.sttv2raw.blob_size	UInt24	RW	Value substitution.
raiko.pkg.sttv2raw.sample_size	UInt24	RW	Value substitution. Number of bits per sample.
raiko.pkg.sttv2raw.end_sector	UInt24	RW	Value substitution. Sector index for last echo in package.
raiko.pkg.sttv2raw.digital_io	Uint24	RW	Value substitution. Send digital IO value.
raiko.pkg.sttv2raw.analog_io_1	Uint24	RW	Value substitution. Send value for analog in channel 1.
raiko.pkg.sttv2raw.analog_io_2	Uint24	RW	Value substitution. Send value for analog in channel 2.
raiko.pkg.sttv2raw.analog_io_3	Uint24	RW	Value substitution. Send value for analog in channel 3.
raiko.pkg.sttv2raw.analog_io_4	Uint24	RW	Value substitution. Send value for analog in channel 4.
raiko.pkg.sttv2raw.package_sequence	Uint24	RW	Value substitution. Package sequence number.

raiko.pkg.eds.stream	UInt24	RW	Alias for raiko.pkg.sttv2raw.stream.
raiko.pkg.eds.sweep_time	UInt24	RW	Alias for raiko.pkg.sttv2raw.sweep_time.
raiko.pkg.eds.transmission_mode	UInt24	RW	Alias for raiko.pkg.sttv2raw.transmission_mode.
raiko.pkg.eds.video_beam_select	UInt24	RW	Alias for raiko.pkg.sttv2raw.video_beam_select.
raiko.pkg.eds.video_type_select	UInt24	RW	Alias for raiko.pkg.sttv2raw.video_type_select.
raiko.pkg.eds.tx_beam_shape	UInt24	RW	Alias for raiko.pkg.sttv2raw.tx_beam_shape.
raiko.pkg.eds.antenna_rotation	UInt24	RW	Alias for raiko.pkg.sttv2raw.antenna_rotation.
raiko.pkg.eds.no_valid_video	UInt24	RW	Alias for raiko.pkg.sttv2raw.no_valid_video.
raiko.pkg.eds.bin_size	UInt24	RW	Alias for raiko.pkg.sttv2raw.bin_size.
raiko.pkg.eds.rx_beam_elevation	UInt24	RW	Alias for raiko.pkg.sttv2raw.rx_beam_elevation.

4.2.14 Package configuration, STTv2 package

This package type is described in **7000 114-966, A1, R5 RIC STTv2 Video Protocol Specification**. These packages are sent through the processor and therefore the Configuration Ethernet port.

Name	Type	RW	Description
raiko.pkg.sttv2.enable	Bool	RW	Enable STTv2 packages.
raiko.pkg.sttv2.ipv4.destination.address	IP	RW	Destination address for packages.
raiko.pkg.sttv2.ipv4.destination.port	Uint16	RW	Destination port for packages.
raiko.pkg.sttv2.processing.threshold.max	Uint8	RW	Threshold where the value is assigned max value.
raiko.pkg.sttv2.processing.threshold.min	Uint8	RW	Threshold where the value is assigned zero.
raiko.pkg.sttv2.compression.method	Uint16	RW	0 = No compression, 1 = RLE
raiko.pkg.sttv2.package.max_size	Uint32	RW	STTv2 package maximum size.

5 DIGITAL VIDEO INTERFACES

This chapter explains the different protocols of digital Ethernet radar video interfaces *Radar Data In* and *Radar Data Out*.

5.1 Radar data in

Currently only one protocol of digital network video input is supported; the Terma Scanter Network Video protocol.

5.1.1 Terma Scanter Network Video

(Radar kind = 6)
This parameter selections instructs the FPGA to use SCANTER Network Video radar data instead of ADC radar data.

Terma Scanter Network Video data in accordance to SCANTER Network Video Protocol **(1)**

- 4096 sweeps per scan
- Up to 8192 cells per sweep
- Each cell represented as 8 bit uncompressed data

The R5 RIC will automatically connect via TCP on the Config port to the SCANTER radar in order to initiate a digital video stream over UDP.
Once the R5 RIC has established a connection to the radar, the RIC will detect the number of cells per sweep and start processing the data.
Since the Terma Scanter network video is already digitized, the data is only conformed to the R5 RIC standardized output.

5.2 Radar data out

The data output of R5 RIC consist of a digitized radar video stream over Ethernet as UDP.
To comply with different customer needs, the output data can be packaged differently.
There are three different protocols for outgoing data when it comes to the raw data feed intended for a plot extractor; STT, STTv2RAW and Asterix.
There is also a secondary output format for a low-resolution video. This is referred to as “Narrow”, and is intended for direct visualisation and not for a plot extractor.

5.2.1 RAW data

5.2.1.1 STT

STT is the main protocol developed by Saab AB TransponderTech.
It consists of a 12 byte large header which results in a low meta-data overhead.

For more information and details please see specification document
7000 114-965, A1, R5 RIC STT Video Protocol Specification.

5.2.1.2 STTv2RAW

To enable more meta-data and the possibility for reduced data rate, a second version of STT was developed. This is named STTv2 and STTv2RAW. More information regarding STTv2 is found in section 5.2.2.
Among these two STTv2RAW is the protocol for digitized radar data intended for plot extraction.
The STTv2RAW protocol implements a dynamic header where the information included is configurable.

For more information, see specification **7000 114-966, A1, R5 RIC STTv2 Video Protocol Specification**

Table 1 - Example STTv2RAW package.

Name	Id	Value/Resolution	Comment
SWEEP_TIME	5		Always zero
TRANSMISSION_MODE	6		Always zero
VIDEO_BEAM_SELECT	7		Always zero
VIDEO_TYPE_SELECT	8		Always zero
TX_BEAM_SHAPE	9		Always zero
RX_BEAM_ELEVATION	13		Always zero
SAMPLE_SIZE	129	4,8,16	Selector for 4, 8 or 16-bit video samples.
COMPRESSION	130	0 = None	Compression algorithm used in the binary block. Default is None.
DIGITAL_IO	144	0-65535	Bitmask for 16-bit digital in
ANALOG_IO_1	145	0-4095	Value of analogue input 1
ANALOG_IO_2	146	0-4095	Value of analogue input 2
ANALOG_IO_3	147	0-4095	Value of analogue input 3
ANALOG_IO_4	148	0-4095	Value of analogue input 4

5.2.1.3 Asterix

An initial implementation of Asterix is available in the R5 RIC.
Soo far a subset of Asterix Category 240 - Radar Video Transmission v.1.2 is available.

For more information, see specification **7000 114-949, A1, R5 RIC Asterix Video Protocol Specification.**

5.2.2 Narrow data

To lessen the data rate, lowering the overhead and enabling processing of low resolution video, there is an option to send multiple sectors per package. This means that multiple sectors are sent in order and fields in the header indicates which sectors are sent.

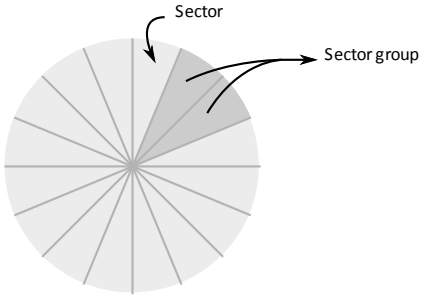

Figure 1: Sector groups

5.2.2.1 STTv2

As mentioned in section 5.2.1.2, STTv2 protocol enables variable header size (configurable meta-data) and also reduced data rates. The difference between STTv2 and STTv2RAW is that STTv2 is sent over the Config Ethernet port whereas STTv2RAW is sent over the Radar data out Ethernet port. Some of the header information is also exclusively applicable STTv2 and STTv2RAW respectively.

Please see specification document **7000 114-966, A1, R5 RIC STTv2 Video Protocol Specification** for more information.

Table 2 - Example STTv2 package.

Name	Id	Value	Comment
DATE	14		Always zero
TIME	15		Always zero
MILLISECONDS	16		Always zero
SAMPLE_SIZE	129	2,4,8	Selector for 2, 4 or 8-bit video samples. Default is 4.
COMPRESSION	130	0 = None, 1 = RLE	Compression algorithm used in the binary block. Default is None.
RANGE_OFFSET	134	1m	Offset of the first sample from the centre of the radar.