Saab TransponderTech AB

Appendices

Project designation

R5 RIC

Document title

R5 RIC Software Protocol Specification

Distribution

Saab TransponderTech AB

DOCUMENT VALIDATION

	Name	Dept. Code/ Function	Date/ Signature
Prepared	Erik Svensson	PT-ES	2012-02-14
Reviewed			
Approved	Christian Andersson		2013-01-14

DOCUMENT CHANGE RECORD

Issue	Change order	Date	Pages affected	Motive for and information on the change
P1A		2012-02-14	All	New document
P5A		2012-09-17		EDS parameters
P6A		2012-10-18	End	Add chapter about R5 RIC Configurator.
A1		2013-01-14		Released revision
A2		2012-03-20		More information about serial configuration, added chapter 2.1.2. More information about configuration parameters. (2.3.2.2, 2.3.2.3)

CONTENTS

1 INTRODUCTION

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

2 DEVICE CONFIGURATION

R5 RIC can be configured through two different interfaces, RS-232 and Ethernet. Following chapters will describe the protocol for each of these interfaces.

2.1 Serial protocol

The serial port settings should be following.

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. “> ”.

2.1.1 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 2.3.

2.1.2 Response

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.

2.2 Network protocol

The device is configured via a text interface over TCP/IP, over the 1Gbit 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.

2.2.1 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 2.3.

2.2.2 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

2.2.3 TCP Session example

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

2.3 Parameters

2.3.1 Data Types

These are the data types that parameters can have.

Data type	Description
Uint16	16-bit unsigned 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)
Int32	32-bit signed integer in ASCII representation
IP	IP address in dotted decimal representation (ex: 10.11.162.1)
Text	ASCII text.

2.3.2 Parameters by category

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

2.3.2.1 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.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.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.

2.3.2.2 Video front end

Name	Type	RW	Description
raiko.video.analog.differential	Bool	RW	Enables/disables differential analog video input.

raiko.video.analog.gain	Uint16	RW	Analog video input analog gain.
raiko.video.analog.offset	Uint16	RW	Analog video input analog offset.
raiko.video.analog.termination	Uint16	RW	Analog video input termination. One of 50, 75 or 0 where 0 is no termination.
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. 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.offset) is applied before the analog video gain (raiko.video.analog.gain) which means that when adjusting the offset the gain may change. It is recommended to adjust the offset first.

2.3.2.3 Acquisition

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	Text	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.

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.

$$N_{sector} = 200$$
$$N_{echo} = 4096$$
$$Sector\ Divider = \frac{N_{Echo}}{N_{Sector}} = 20.48$$

2.3.2.4 System configuration

Name	Type	RW	Description
sys.net.mac	Text	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.

2.3.2.5 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.

2.3.2.6 Radar configuration

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

2.3.2.7 Radar configuration, Sperry Bridgemaster

radar.kind = 1

Name	Type	RW	Description
radar.bridgemaster.transmit	Bool	RW	Enables radar transmitter.
radar.bridgemaster.profile	Uint16	RW	Pulse length. 1 = Short, 2 = Medium, 3 = Long.

2.3.2.8 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 = EDS

2.3.2.9 Package configuration, EDS package

These parameters are described in 7000 114-953, P2A, R 5RIC GIRAFFE AMB digital video interface protocol specification.

Name	Type	RW	Description
raiko.pkg.eds.stream	UInt24	RW	
raiko.pkg.eds.sweep_time	UInt24	RW	
raiko.pkg.eds.transmission_mode	UInt24	RW	
raiko.pkg.eds.video_beam_select	UInt24	RW	
raiko.pkg.eds.video_type_select	UInt24	RW	
raiko.pkg.eds.tx_beam_shape	UInt24	RW	
raiko.pkg.eds.antenna_rotation	UInt24	RW	
raiko.pkg.eds.no_valid_video	UInt24	RW	
raiko.pkg.eds.bin_size	UInt24	RW	
raiko.pkg.eds.rx_beam_elevation	UInt24	RW	

3 RADAR VIDEO DATA

The radar video data is communicated via UDP for maximum throughput. Depending on the network layout, one complete pulse may or may not fit within one UDP packet. The packet size and data type used is determined by the device settings.

3.1 Data types

Following data types are used in this protocol. The byte order is little-endian.

Data type	Description
Int8	8-bit signed integer. (1 byte)
Uint8	8-bit unsigned integer. (1 byte)
Int16	16-bit signed integer. (2 bytes)
Uint16	16-bit unsigned integer. (2 bytes)
Uint32	32-bit unsigned integer. (4 bytes)

3.2 Pulse Header

At the beginning of each package a header is sent, with the following layout:

Pulse Header		
Bytes	**Data type**	**Contents**
0	Uint8	Message type = 0 (new pulse) = 1 (continued pulse)
1	Uint8	Data type = 2 (Int16) (see below)
2-3	Uint16	Azimuth (value of ACP counter)
4-7	Uint32	Size in bytes of raw radar data
8-11	Uint32	Offset in bytes of raw radar data.

The value of the data type field is interpreted as:

Data Type Constants	
Data Type	**Value**
Int8	0
Int16	2

The rest of the UDP packet contains raw radar data in the given format from the offset and forward.