M211320EN-E

User Guide

Radar Control Processor

RCP8

VAISALA

PUBLISHED BY

Vaisala Oyj
Street address: Vanha Nurmijärventie 21, FI-01670 Vantaa, Finland
Mailing address: P.O. Box 26, FI-00421 Helsinki, Finland
Phone: +358 9 8949 1
Fax: +358 9 8949 2227

Visit our Internet pages at www.vaisala.com.

Table of Contents

List of Figures

List of Tables

1. About This Document

1.1. Version Information

This document provides technical information on the Radar Control Processor (RCP8).

This document is used during troubleshooting or by those interested in understanding the use, the configuration, and data formats used by RCP8.

Table 1 Document Versions

Document Code	Date	Description
M211320EN-E	October 2016	This manual. Fifth version.
M211320EN-D	September 2014	Previous manual. Fourth version.
M211320EN-C	November 2013	Previous manual. Third version.

1.2. Related Documents

Table 2 Weather Radar Documentation

Document Code	Name
M211315EN	*IRIS and RDA Software Installation Guide*
M211318EN	*IRIS Programming Guide*
M211316EN	*IRIS and RDA Utilities Guide*
M211319EN	*IRIS Product and Display Guide*
M211317EN	*IRIS Radar User Guide*
M211452EN	*IRIS and RDA Dual Polarization User Guide*
M211322EN	*RVP900 Digital Receiver and Signal Processor User Guide*
M211320EN	*Radar Control Processor RCP8 User Guide*

For information on changes made since your current release was installed, download the latest document versions and check the IRIS and RDA Release Notes from www.vaisala.com.

Vaisala encourages you to send your comments or corrections to helpdesk@vaisala.com

1.3. Documentation Conventions

WARNING! Warning alerts you to a serious hazard. If you do not read and follow instructions carefully at this point, there is a risk of injury or even death.

CAUTION! Caution warns you of a potential hazard. If you do not read and follow instructions carefully at this point, the product could be damaged or important data could be lost.

Note highlights important information on using the product.

Tip gives information for using the product more efficiently.

Lists tools needed to perform the task.

Indicates that you need to take some notes during the task.

1.4. Trademarks

HydroClass$^{(TM)}$ is a trademark of Vaisala Oyj.

IRIS$^{(TM)}$ is a trademark of Vaisala Oyj.

All other product or company names that may be mentioned in this publication are trade names, trademarks, or registered trademarks of their respective owners.

2. Product Overview

2.1. Radar Control Processor

Radar control processor (RCP) software controls and monitors weather radar system sub-units, including the pedestal, power supply unit, transmitter, receiver, waveguide matrix, cabinet cooler, dehydrator, and safety interlock system.

The hardware interfaces for the subsystems are the CAN bus, ethernet, I/O connector panel, serial, and, USB interfaces.

The fail–safe features of RCP8 are designed to protect the radar and antenna system in the event of a failure.

The optional antenna stabilization is for moving platforms, such as ships or airplanes. For shipboard use, RCP8 accepts position, attitude and speed information from an inertial navigation unit (for example, the Honeywell MAPS Hybrid system.) The antenna scans in Earth coordinates regardless of the platform pitch, roll, or heading.

Radar Control Processor Software

RCP software steers the radar antenna in the defined measuring direction to read the azimuth and elevation angles from the angle encoders. The angle values are combined with the output of the RVP signal processing to display measured radar data as a function of azimuth, elevation, and time.

RCP software displays status information, such as the status and faults of radar system units as well as the controls for switching the transmitter radiation on and off.

2.1.1. Radar Server Computer

The radar server computer is a standard rack-mounted server running radar system software:

- RCP8 radar control processing software for controlling antenna and pedestal movement.
- RVP900 signal processing software.
- IRIS Radar software for overall radar system management and data product generation.

The radar server computer has two Ethernet interfaces, one connecting the server to the IFDR unit of the receiver and another connecting the server to the external network.

The radar server computer has a Vaisala IO-62 PCI card for radar system hardware interfaces. It is connected to a separate I/O connector panel with a 62-pin parallel cable.

For redundancy, the server is equipped with dual power supplies.

The server computer also includes standard interfaces, such as serial and USB ports, and provisions for connecting a display, keyboard, and mouse for local use.

The radar server computer fits in a standard 19″ cabinet, and is 2U in height.

2.1.2. I/O Connector Panel

The I/O connector panel is a rack-mounted unit that provides an interface between RCP8 and radar hardware, such as the transmitter and other sub units of the radar system for analog and digital controls and statuses.

The panel is connected to a RCP8 I/O PCI card in the radar server computer with a 62-pin cable.

Signals used by the weather radar systems include, for example:

- TTL inputs/outputs and external relay driver outputs for miscellaneous control and status.
- +/- 10 V analog inputs to monitor, for example, power supply voltages.
- serial interfaces
- triggers
- relays
- analog synchro inputs
- parallel angle inputs and outputs

Configuration

The I/O configuration defines which RCP8 software variable is assigned for each input or output. The variables are further configured for display in graphical user interfaces, such as the **Bitex** utility, the **Antenna** utility, or the IRIS radar **Status Menu**.

Configuration is mostly done in a text file in the radar server computer configuration directory (*/usr/sigmet/config/softplane.conf*). Some configurations are also made in the RCP8 setups through the **Antx** utility.

2.2. Safety

WARNING! Turn off power to RCP before installing or removing PCI boards. For safety, disconnect the line cord before opening the RCP.

CAUTION! The circuit boards contain static-sensitive components. Wear a properly grounded wrist strap to handle the PCI boards.

2.2.1. ESD Protection

Since the I/O lines are connected to the radar system, there is a potential for lightning or other ESD-type damage.

The following I/O-62 features make RCP resistant to transient surges:

- Wires are protected by a Tranzorb diode, which transitions from an open to a full clamp between ±27 to ±35 VDC. The connector panel also uses Tranzorb diodes on every I/O line for double protection.
- High-voltage tolerant front-end receivers/drivers. All components connected to the external pins can tolerate up to ±40 V. For example, the TTL and wide range inputs use protectors that normally look like 100 Ω resistors, but open at high voltage.

2.3. Product Nomenclature

Table 3 RCP Nomenclature

Code	Common Name
RVP902-IO	Signal processing computer with RCP8
RVP8-IO	IO62 card

3. Functional Description

3.1. Architecture

RCP8 is based on a standard PCI architecture running a Linux operating system. The hardware chassis is constructed with redundant power supplies, captive quick release fasteners, PCI card guides, and security hold-down bar.

A standard system contains the following components:

- Vaisala I/O-62 general purpose I/O board.
- Vaisala RCP8 connector panel (connects to I/O-62).

Depending on the application, other standard commercial PCI cards may be added for additional I/O capability such as the following:

- Dual port 1Gb Ethernet adapter for additional network I/O (for example, a backup network).
- RS-232/RS-422 serial cards for serial angles, remote TTY control, and similar.

The front of the unit also contains a DVD drive for backup, software installation, and maintenance.

RCP8 is configured locally using a keyboard, mouse and monitor, or remotely over the network.

The configuration menus are TTY text-based menus for configuring the antenna servos, host computer interface, and programmable control logics. The TTY menus also provide status and monitoring for diagnostic purposes and during the antenna stabilization process.

Pin assignments to the connector panel are made in the `softplane.conf` file.

3.1.1. I/O-62 PCI Card and I/O Connector Panel

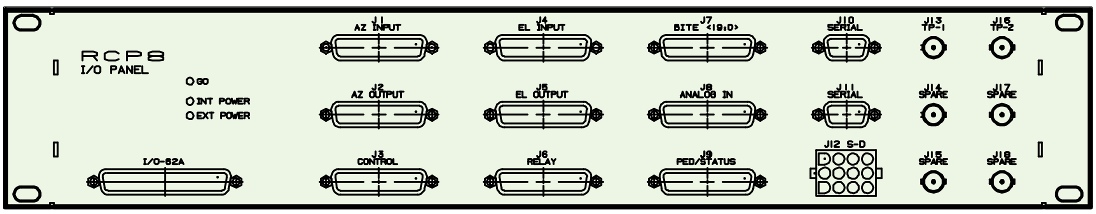

Figure 1 RCP8 Connector Panel

The Vaisala I/O-62 is a short format PCI card that provides I/O capabilities for RCP8. A typical installation has an I/O-62 PCI card and the RCP8 I/O connector panel.

The I/O-62 has a single 62-position, high-density D connector. This is attached to the rackmount RCP8 connector panel using a standard 1.6 m (6 ft) 1:1 cable. The connector panel is usually installed in the same rack as RCP8.

The I/O-62 is software-configurable. There is no need to open the chassis to configure jumpers or switches. When a spare board is added, there is no need to perform hardware configuration or custom wiring.

If more I/O is required, a second I/O-62 can be installed.

More Information

- Antenna Control I/O and Features (page 107)

3.1.1.1. Run Time FPGA Configuration

The Vaisala I/O-62 card is built around a 100-K Gate FPGA that drives the I/O signals on the 62-position connector and coordinates the PCI traffic.

The chips are SRAM-based, meaning that they are configured at run time. This allows the FPGA code to be automatically upgraded during each code release without needing to reprogram any parts.

The board's basic I/O services use up only 40 % of the complete FPGA. The leftover space makes it possible to add smart processing on the I/O-62 board to handle custom needs such as generating custom serial formats, data debouncing, and signal transition detection.

More Information

- Network Architecture Options (page 16)

3.1.2. Network Architecture Options

RCP8 provides flexibility to network operations. This allows remote control and monitoring of the system from anywhere on the network, subject to the user's security restrictions.

There are 3 basic types of workstations/computers to consider:

- RCP - this can be equipped with a local keyboard, mouse and monitor.
- Host host radar server computer - this is running the user's application software (for example, the IRIS/Radar software and utilities).
- Remote workstation - a networked workstation used for remote control and monitoring. This may be running only X-Windows or additionally the user's application software or IRIS application software.

RCP8 provides the following physical interfaces:

- RS232C serial line interface - typically running at 9600 baud.
- Ethernet socket interface - at 10/100/1000 BaseT. The `AntExport` software provides a socket interface to other workstations on the network.
- Native connection - RCP8 runs application software locally. The local connection is over an FIFO interface.

The following table shows the network connection options using the possible workstations and physical interfaces.

Table 4 RCP8 Network Connection Options

Option	Description	More Information
Standard Serial Line	Connects RCP8 to the host radar server computer.	3.1.3. Case 1: Standard Serial Line Interface (page 17)
Combined RCP8/RCW	RCP8 and radar server computer run on the same computer to eliminate the need for a separate host computer. Vaisala only guarantees the performance of this configuration if the radar server computer runs Vaisala IRIS software.	3.1.4. Case 2: Combined RCP8 and Radar Server Computer (page 18)
TCPIP Socket	Ethernet connection between RCP8 and the host radar server computer.	3.1.5. Case 3: Socket Interface Using AntExport (page 19)

3.1.3. Case 1: Standard Serial Line Interface

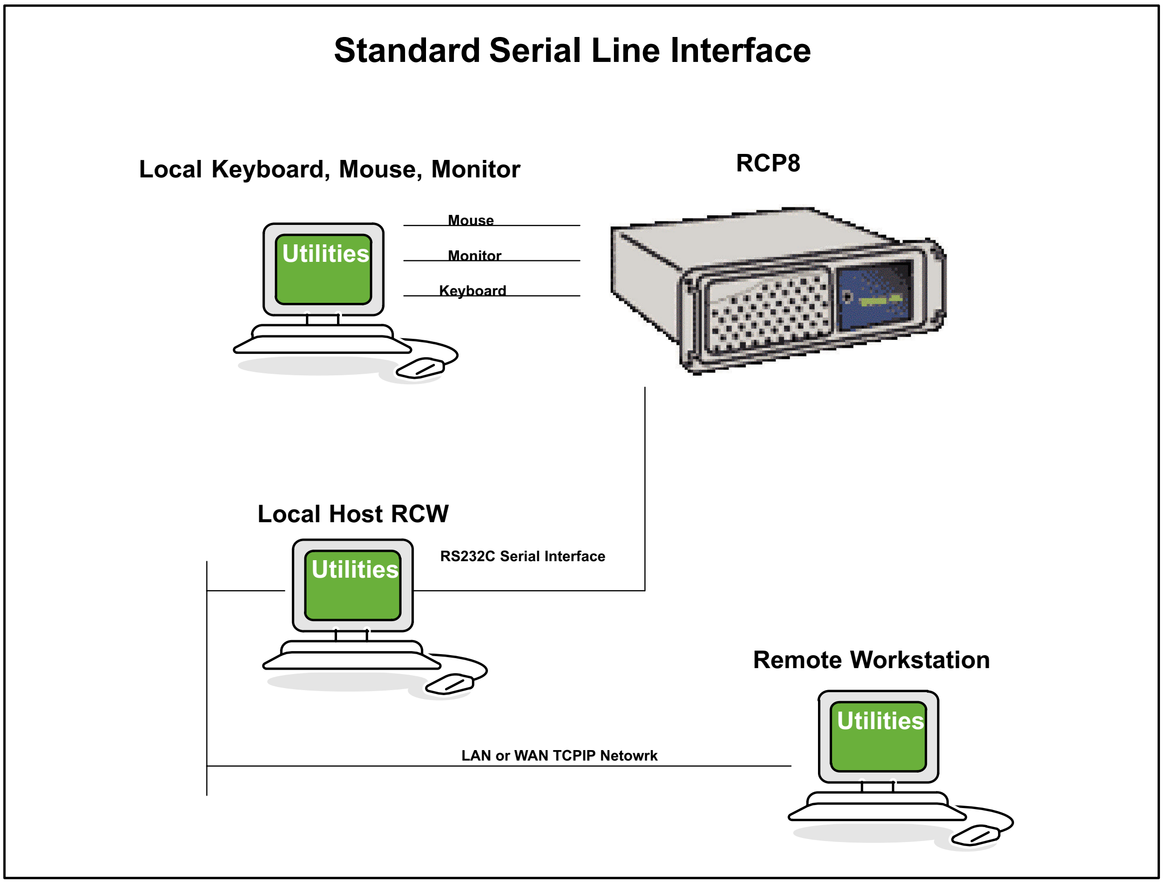

Figure 2 Network Architecture - Standard Serial Line Interface

Remote computer access assumes that there is sufficient band-with to export an X-Window from the radar server computer. This typically requires at least 128 kbps.

Note that satellite links may have this band width, but their latency cause slow X-Window export. In this case, it is better to use a hybrid serial/socket approach.

3.1.4. Case 2: Combined RCP8 and Radar Server Computer

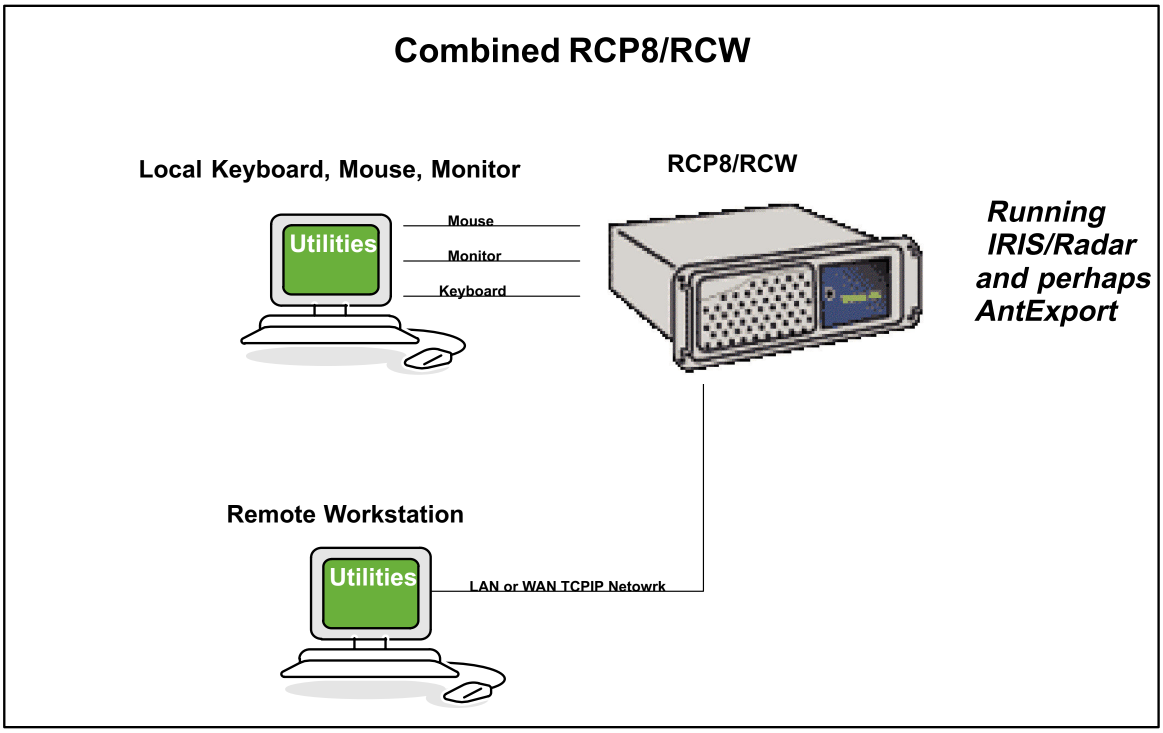

Figure 3 Network Architecture - Combined RCP8 and Radar Server Computer

Most applications use this configuration.

Modern computers can run RCP8 and IRIS/Radar software on the same machine. In this case, access from a remote workstation is done by X-Window export.

In the case of a slow link to the remote workstation, `AntExport` can be run on RCP8 and the radar server computer to service a low-speed connection to a remote workstation.

3.1.5. Case 3: Socket Interface Using AntExport

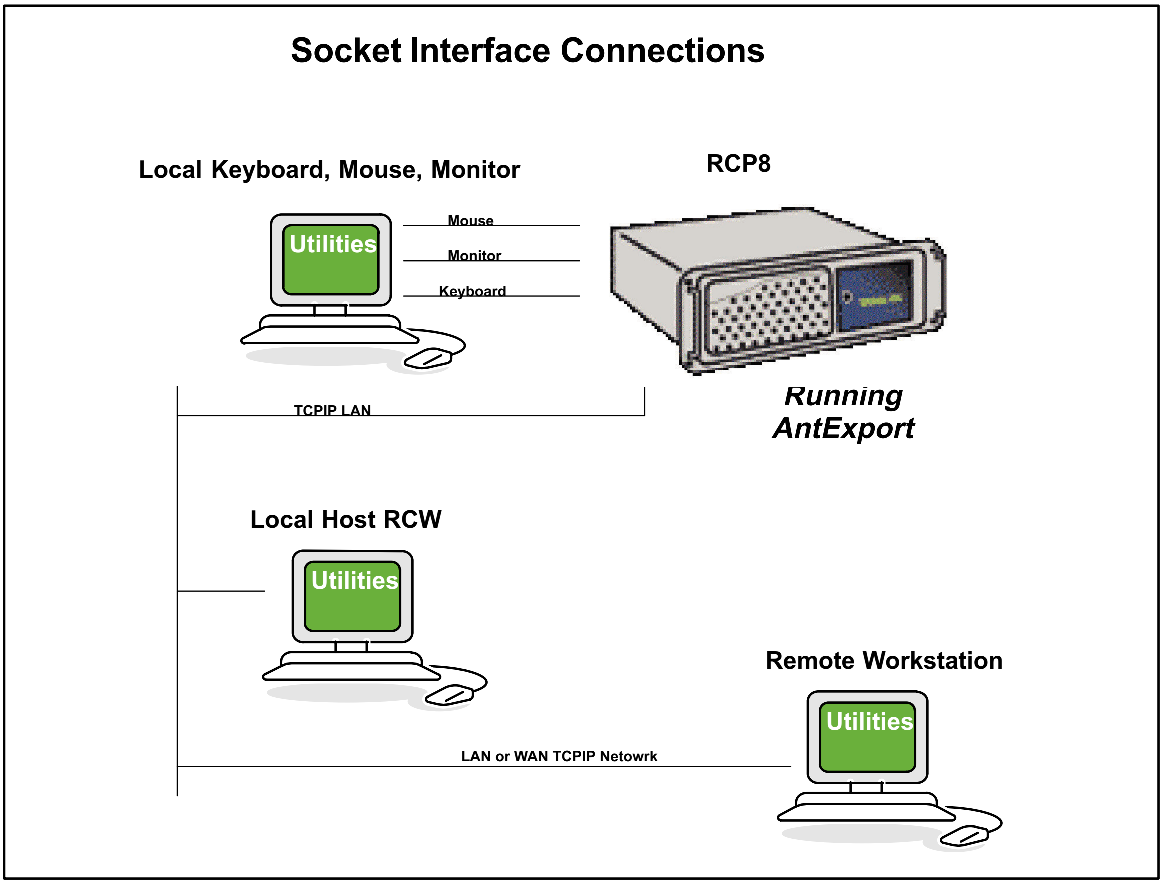

Figure 4 Network Architecture - Socket with AntExport

In this architecture, RCP8 runs `AntExport` to provide a socket connection between RCP8 and the host radar server computer. Note that any computer on the network can, in theory, function as a direct controller of RCP8.

More Information

- Configuring Network Connections (page 39)

3.2. Application Software for Testing and Monitoring

The radar server computer contains software utilities for testing and monitoring:

- **Antenna Utility**: For control and monitoring of the antenna and standard status and control parameters. Includes sun tracking feature for antenna alignment check.

- **Bitex Utility**: For status monitoring and control of BITE (built-in test equipment). Graphical backgrounds and status widgets can be customized by the user, including quantitative BITE from analog inputs.

For more information, see *IRIS and RDA Utilities Guide*.

4. Servo Operation Theory

4.1. Servo Concepts

RCP8 provides two independent, and nearly identical, motion servos for the radar antenna azimuth and elevation axes. These servos are implemented digitally in the RCP8 microprocessor.

The servo software takes as input the digital antenna position and analog tachometer velocity and provides, as output, an analog drive signal for the motor power amplifiers. The interface between the processor and the tachometer and drive signals is made using 12-bit A/D and D/A converters.

The servo software is periodically scheduled at 10 millisecond intervals and, in principal, can control antennas that have a significant response, such as 20 Hz. In practice, most weather radar antennas are much slower. Except for the limit switches in the elevation axis, the servos are identical in configuration and operation.

The following table shows how the RCP8 servos can operate.

Table 5 RCP Servo Operating Options

Option	Description
Open loop	The open loop is not really a servo. It applies fixed drive levels to the motor to measure the antenna performance. The open loop only runs during installation to measure the antenna's characteristics to set up the actual control parameters. This a manual procedure requires the local TTY. See 7.3. TTY MONITOR Command (page 46).
Velocity servo	The velocity and the position servos are related - each mechanism uses parts of the other during normal operation. The velocity servo always runs when either servo is activated. To achieve a particular velocity, the servo is used directly. To achieve a particular position, a non-linear position error is fed into the velocity servo from the position servo.
Position servo	The position servo is implemented in the following stages: 1. Convert the position error into a requested velocity. 2. Convert the requested velocity into a drive signal. Theoretically, this 2-stage position servo can always be made stable - the position can always be reached without overshoot or oscillation. The non-linear feedback function can also be tailored to achieve stability and high performance. This means that a requested position is reached in the shortest possible time.

More Information

- Using TTY Setup Menus (page 59)

4.2. Velocity Servo Theory

The following figure shows the block diagram of the velocity servo.

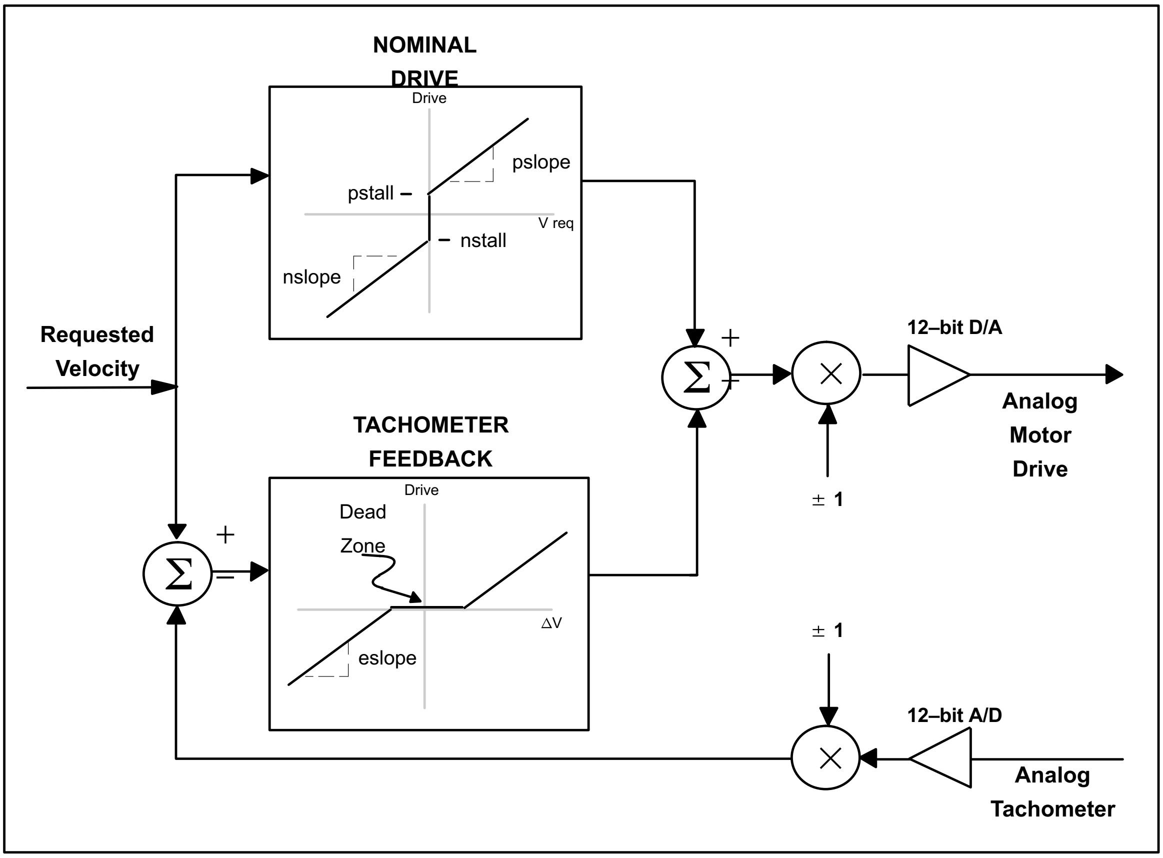

Figure 5 Digital Velocity Servo Block Diagram

4.2.1. Tachometer Input

The tachometer signal, from the motor gear box, is applied to a differential receiver and a 30 Hz analog, low-pass filter. The signal is then digitized and added to the processor.

The differential receiver ensures that any common-mode signal, on both tachometer leads (for example power-line noise) is not falsely interpreted as antenna motion. If there is no analog tachometer, a virtual tachometer based on the differentiated position can be selected. See 8.5. AXIS Command (page 75).

The drive levels that are computed by the processor are applied to a 14-bit D/A converter, scaled by the external analog amplifiers, and then applied to the motor power amplifiers.

The D/A and A/D convertors are signed. They generate and accept both positive and negative voltage levels.

The drive signal is the sum of the following components:

- A nominal level based solely on the requested velocity.
- Feedback term based on the difference between the requested and actual velocities.

The 2 graphed transfer functions indicate how the drive levels are derived for each component.

4.2.2. Nominal Drive Slope

The nominal component is an initial guess of the drive level that would sustain a given velocity in the steady state.

For a requested velocity of 0, the upper-transfer graph indicates that no drive was applied. Without a drive, the motor eventually comes to rest. For non-zero velocities, most motors exhibit a dead zone in which the armature magnetization is insufficient to overcome the starting friction. Therefore, the nominal drive graph takes a discontinuous jump from 0. Due to the antenna's imbalances, this dead zone can also be asymmetric for both directions of motion.

These positive and negative starting drives are designated as `pstall` and `nstall` on the graph. Once the motor starts, a nominal slope is designated as `pslope` for the positive velocity and `nslope` for the negative velocity. Both are used to predict the required drive for large requested velocities.

4.2.3. Velocity Feedback Slope and Dead Zone

The feedback component of the motor drive is based on the difference between the requested and the actual (tachometer) velocities.

The lower transfer graph demonstrates that the output is essentially linear, with a velocity error, except for the possible inclusion of a deadzone around 0.

- The slope is designated as `eslope` on the diagram.
  Typically, the `eslope` is fairly large in order to achieve a tight velocity servo however, this large value also magnifies the A/D errors.
- The deadzone, between -V and V , is used to minimize motor "chatter" that can result from uncertainty in the LSB of the tachometer voltage samples.
  A small inactive region (dead zone) in the feedback loop, typically two 1 or 2 T-units, eliminates the problem with A/D errors.

The sum of the nominal and feedback terms is clamped within the -100 ... +100 drive unit range and is applied to the D/A converter to produce the motor drive voltage.

Note that the nominal term does not need to be calculated with great accuracy. In traditional, hard-wired, analog velocity servos, this term is unused. In the digital servo, the term:

- Provides and easy way to take motor stall currents into account.
- Helps reduce the mean error that appears in the feedback term necessary to maintain a given velocity.

Every feedback system requires a non-zero error component to maintain control of a non-equilibrium position. By predicting the equilibrium drive requirements, the nominal term helps to ensure that the mean steady-state value of the velocity error is 0.

4.2.4. Drive and Tach Sign Correction

Two optional sign inversions can be introduced in the velocity servo loop: one for the tachometer input and one for the drive output. The inversions must be set so that:

- Overall servo is stable.
- Positive requested velocities results in the positive tachometer velocities.

If the first condition is not met, then flipping either sign makes it stable. If this leads to a violation of the second condition, then you must flip both signs. Therefore, both conditions can be met by a suitable choice of multipliers.

The need for the stability condition is obvious but the need for the correct tachometer sign results from the requirements that the position servo imposes when it is running.

4.3. Position Servo Theory

The position servo is implemented as a simple extension, as shown in the following figure.

To reach a given position:

1. The position error is used to calculate the velocity that is necessary to correct it.
2. The velocity is fed into the velocity servo, which continues to operate as described in the previous sections.

The elevation position servo works over the complete 360° interval from -90 ... +270°.

Servo motion is always directed over the top when the antenna moves from one position to another. For example if the antenna is at +200°, and a request is made to move to -30°, then the antenna traverses the 230° sector passing through 90°. This is different from the azimuth axis, where the shorter 130° path would be taken.

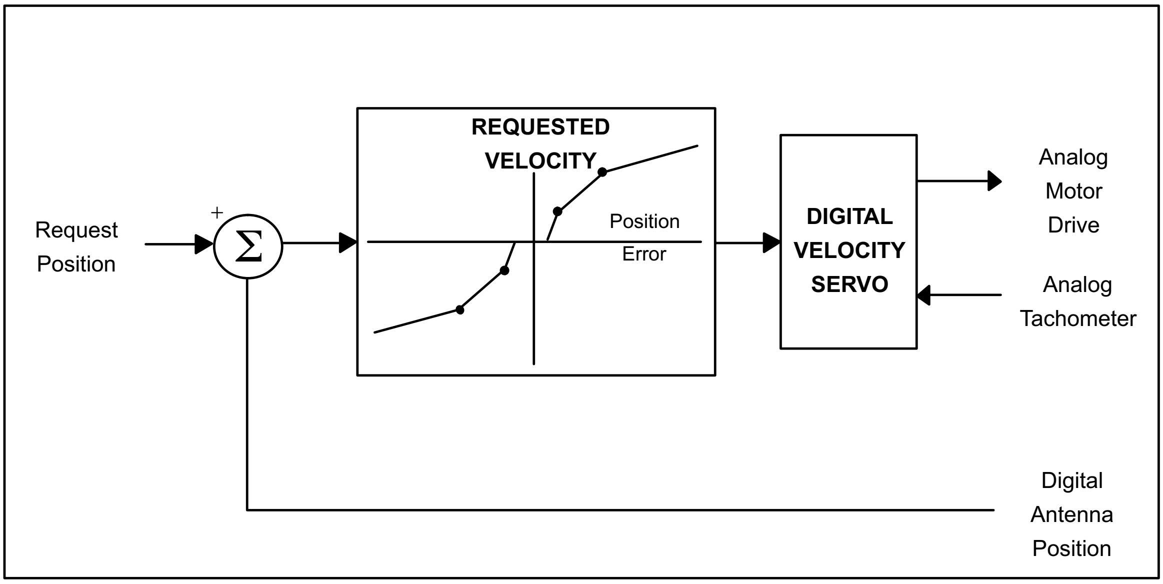

Figure 6 Digital Position Servo

4.3.1. Position Servo Response Curve

The mapping between the position error and the requested velocity, known as the position servo response curve, is non-linear and takes account of:

- Stored kinetic energy in the antenna mass
- Non-ideal, power driver characteristics
- Inductive and regenerative motor effects
- Friction

Overall, the concave downward shape of the curve can be understood as follows: The angular velocity of the antenna cannot be changed instantaneously, but is limited to a rate that may depend on the velocity itself. The time integral of such velocities produces roughly quadratically shaped positions. When approaching from a far distance at high speed, the distance covered—in the time required to reduce the velocity by half— is approximately three-quarters of the initial position error.

More Information

- PSERVO Command (page 89)

4.3.2. Feed Forward Position Servo

If configured, the feed forward position servo supports the position servo by providing:

- Pre-calculated drive output for position drive. No feedback needed during the drive.
- Acceleration and deceleration calculation using antenna mechanical resonance period to prevent excitation of resonant oscillation.
- Acceleration time and peak velocity calculations provide the shortest possible travel time to the demanded angle, without exceeding set limits.
- Self-learning algorithm to make next position drive even more accurate.

4.4. Fail-safe Antenna

Radar antenna systems consist of the reflector, pedestal, gears, motors, and drive amplifiers. These components must be protected in the event of failures.

If a critical failure is detected, RCP8 goes to a shutdown state as follows:

1. The drive output voltage is zeroed.
2. The drive control relay signal is set to low.
   If a drive control relay is used, RCP8 drive outputs are physically disconnected from the servo amplifiers. Depending on the installation, this may switch to an alternative drive system such as handwheels.
3. An error bit is set in the output to the host computer.

> **CAUTION!** In cases where the alternative drive system may attempt to move the antenna, it may be undesirable to automatically switch to the alternative drive when a shutdown occurs. In such cases, do not implement the RCP8 drive relay signal. Instead, implement a manual switching approach.

When a shutdown occurs, you must investigate the reason for the shutdown in the **Control/ Monitoring** menu. After the fault has been corrected, issue a `reset` command, either from RCP8 menus or over the host computer serial line.

One of the most potentially damaging situations is when the antenna operates outside the specified elevation range. There are several limits that are typically imposed to protect against this.

More Information

- Maximum Velocity Watchdog Algorithm (page 27)

4.4.1. Elevation Limit Switch Shutdown Algorithm

Elevation limit switches can be set to force an antenna shutdown. The algorithm checks 40 times per second for limit-switch contact.

More Information

- Miscellaneous Site Setups (page 75)

4.4.2. Elevation Shutdown Limit Shutdown Algorithm

The Elevation Axis setup allows you to specify upper and lower elevation limits that, if exceeded, cause the antenna to shutdown.

The limits are checked 100 times per second. There is no tolerance for this test.

More Information

- AXIS Command (page 75)

4.4.3. Elevation Soft Limit Watchdog Algorithm

When activated, the soft limit algorithm makes sure the antenna is brought to a safe stop at the soft limits, regardless of the servo mode (open loop, velocity, or position.)

CAUTION! Before activating this algorithm, make sure the elevation position servo has been configured and tested . See 7.3. TTY MONITOR Command (page 46) .

In order to enforce the soft position limits on the elevation axis, the velocity servo calls a few position servo subroutines on each iteration. This determines whether the currently requested velocity, which may not have come from the position servo, allows the antenna to be stopped before encountering the limit.

If the requested velocity is too high, it is replaced by the velocity that the position servo would have used in order to just reach the limit. This safeguard makes sure the antenna speed is reduced early enough to reach a controlled stop before encountering the specified soft limit.

Analogous soft limits can also be set for azimuth, but these are rarely used since most antennas can rotate freely on the azimuth.

4.4.4. Maximum Velocity Watchdog Algorithm

RCP8 performs the following checks on the velocity to ensure that the antenna operates within the safe limits:

- A velocity request limiter clamps any out-of-bounds velocity requests from the host computer or indirectly from the position servo at the maximum value.
- A continuous check on the antenna velocity determines that it operates within safe limits.

The checks are controlled with the following setup parameters:

- `Maximum Absolute Velocity` - 80 Tach units
- `Velocity Shutdown Safe Margin` - 5 Tach units
- `Velocity Shutdown Time Check` - 1 second

The watchdog forces an antenna shutdown if the velocity exceeds the `Maximum Absolute Velocity`, by more than the `Velocity Shutdown Safe Margin` margin, during a time period longer than that of the `Velocity Shutdown Time Check` check.

The following example shows the lower elevation limits for a typical system. Upper elevation limits are analogous.

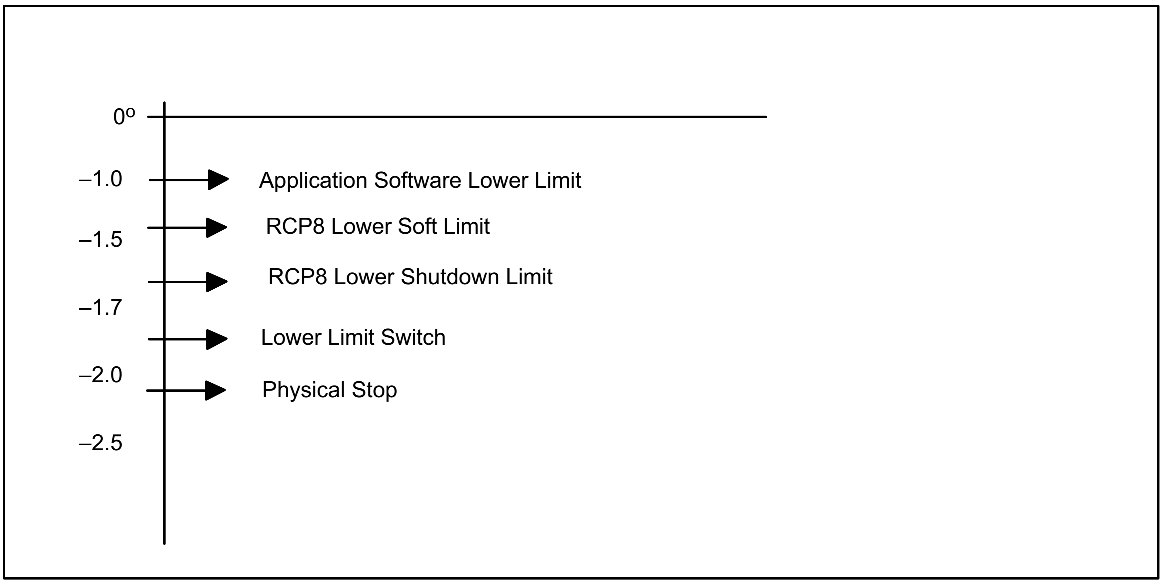

Figure 7 Example of the Lower EL LIMITS

More Information

- VSERVO Command (page 84)

4.4.5. Tach/Position Consistency Shutdown Algorithm

This algorithm requires that the Tach be calibrated in degrees/second. If the tach gain potentiometer is adjusted, you must redo the tachometer calibration.

If virtual tach is used rather than an actual tachometer, this algorithm is disabled.

When the Tach is properly calibrated, the observed change in the antenna position should match the integrated velocity. If these are inconsistent, this could indicate failure of either the Tach or the position sensing and continued operation could lead to antenna damage.

The following algorithm parameters, are defined as:

- Permissible fixed error - 1.5°
- Permissible relative error - 10.00 %

The watchdog algorithm computes the expected difference in position by integrating the velocity and comparing this to the observed position difference over the previous 1 second. The algorithm forces a shutdown of the antenna if the difference between the observed and the computed antenna displacements exceeds the larger of the fixed and the relative error.

For the antenna displacements greater than 15° in 1 second, the relative error of 10 %, for example, is used as the standard for the test, while for the displacements of less than 15°, the fixed error of 1.5° is used.

> The algorithm integrates over the prior one second interval and is updated 16 times per second.

More Information

- AXIS Command (page 75)

4.4.6. Unresponsive Antenna Watchdog Algorithms

When drive is applied to the antenna, the antenna generally accelerates. Failure of the antenna to accelerate could be the result of one or more of the following reasons.

- Servo amplifier failure or servo amplifier turned off
- RCP8 drive output failure
- Drive cable failure
- Catastrophic gear failure of the antenna drive
- Obstacle impeding the antenna motion, such as a person, a ladder, or a stow pin inadvertently left in the antenna

With the exception of the servo amplifiers being turned off, any of of these events warrants an antenna shutdown. However, if the antenna scans at its equilibrium velocity, the output drive does not cause the antenna to accelerate since it is just balancing against frictional losses. This must be taken into account to avoid false alarms.

The unresponsive antenna algorithm is based on a linear model of the antenna velocity, with a constant moment of inertia, and frictional losses that are proportional to velocity. Under this model, the expected change in velocity can be calculated by numerical integration. The expected change is then compared to the actual change in velocity.

The following **`AXIS`** setup parameters for this algorithm are defined as:

- Permissible Tach Prediction Error - 15 Tach units
- Maximum duration of such error - 2 seconds
- Moment of inertia - 4.00 Drive/Tach units

The moment of inertia is computed when the antenna accelerates and is shown in an **`alt`** displays in the **Control and Monitoring** menu. You can then enter a representative value entered in the setup.

> This algorithm does not require the Tach to be calibrated in degrees/second. However, if you adjust the Tach or drive potentiometers, you must reconfigure this algorithm.

The algorithm performs a numerical integration, over the previous 2.5 seconds, to obtain the expected change in velocity (in Tach units.) If the difference between the expected velocity and the current velocity exceeds the `Permissible Tach error` for a period greater than the `Maximum duration`, the watchdog forces an antenna shutdown.

> The algorithm integrates over the previous 2.5 second interval and is updated 8 times per second.

More Information

- AXIS Command (page 75)

4.5. Modifying Servos For a Moving Platform

Using base motion inputs, RCP8 can carry out electronic stabilization of an antenna that is mounted on a moving platform.

For moving platforms, you must modify the position and velocity servos so the antenna motion is referenced to the inertial (Earth) frame of reference. The positions and the velocities are requested by the user and reported back to the user, relative to the local horizon and local north, just as they would for a stationary pedestal. RCP8 managesthe coordinate transformations needed to convert between the Earth system of units and the pedestal system of units.

To stabilize an antenna on a moving pedestal, RCP requires the instantaneous roll, pitch, and heading of the pedestal base as well as the time derivatives of those three quantities. Pedestal orientation is used to convert between the two coordinate systems. RCP also requires the rate of change of pedestal orientation.

The following list shows the components that contribute to the net Earth velocity of a scanning antenna:

- The component results from rotation of the pedestal azimuth and elevation axes.
- The component is the result of projected motion of the entire pedestal assembly.
- The component is computable from the rate of change in the base orientation angles.

You can modify the basic velocity servo (see 4.4. Fail-safe Antenna (page 26)), to work in Earth coordinates by adding a coordinate conversion module. The modified servo is shown in the following figure.

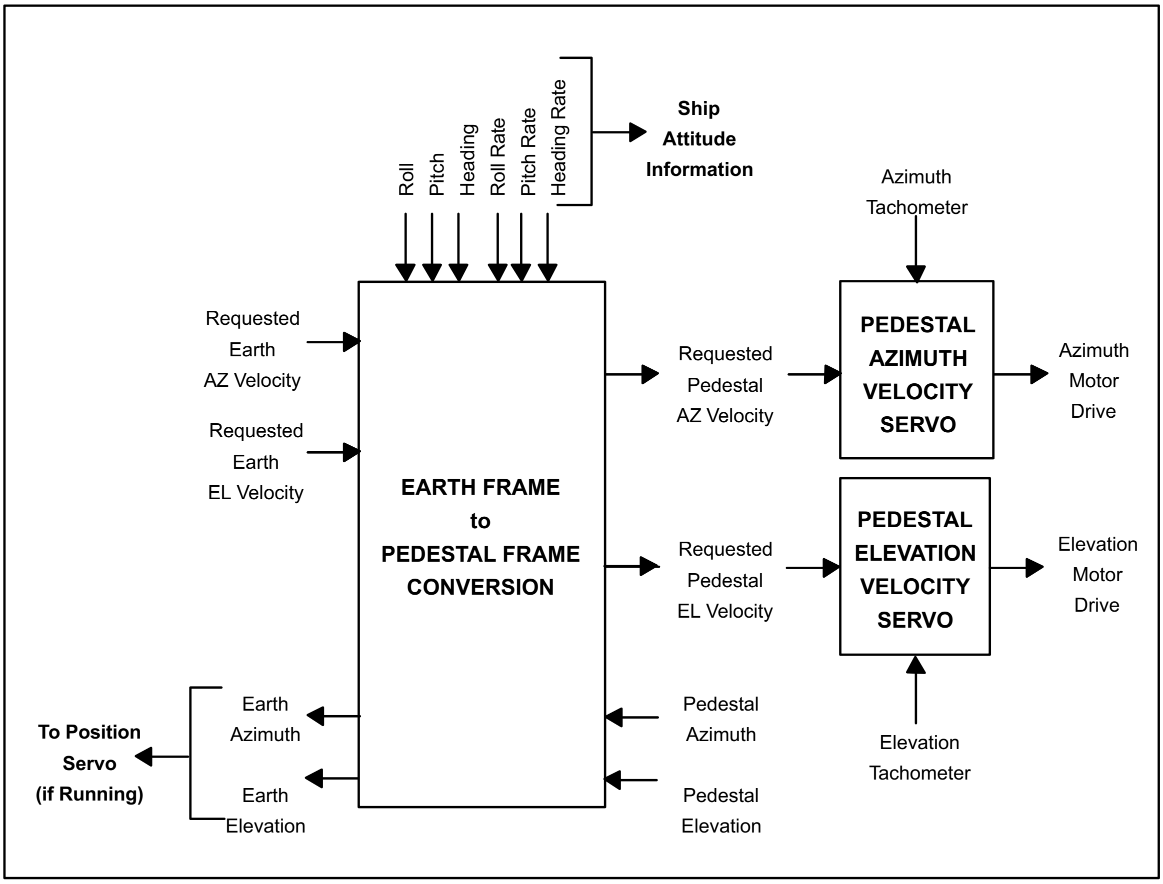

Figure 8 Modifying Velocity Servos for a Moving Platform

Rather than accepting requests for particular pedestal speeds, the new servo responds to commands to move at real Earth-relative velocities. The conversion module receives pedestal attitude information from an inertial navigation unit (INU) mounted on or near the pedestal. Based on these data, and on the current pedestal azimuth and elevation, Earth velocity requests are converted to equivalent pedestal velocity requests. These are fed to the 2 old-style velocity servos that use tachometer feedback to compute appropriate motor drives.

The velocity conversion module uses the pedestal azimuth and the elevation angles as input to project the Earth velocity into the pedestal frame. The Earth azimuth and the elevation angles are also computed.

When a position servo runs on one or both axes, these computed Earth angles are used by the position servo in the same way that pedestal angles are used in the land-based case. The position servo (see 4.3. Position Servo Theory (page 24)) is unchanged with the exception of the Earth angles, which are substituted where a pedestal angle previously appeared.

The angles can also be wired to a nearby signal processor and simultaneously sampled with data from the radar.

A complimentary addition to RCP8 moving platform servos is its ability to scan co-planes. By introducing an artificial bias to the pedestal attitude information, a 0° elevation scan can be transformed into a planar scan in any orientation, not simply along the horizontal plane. This technique works on land and sea.

The INU data stream may include status bits that convey the validity of the attitude angles. RCP8 coasts for up to 1 second when it receives an invalid INU Roll/Pitch/Heading bit, or until the invalid bit is cleared, whichever occurs first.

The last valid report of INU parameters are used for stabilization during this time (including computation of the earth-relative output angles). Since it is unlikely that the antenna azimuth and/or the ship attitude move more than 30° in 1 second, IRIS message `DSP AZ angles exceed 30 degrees` is not triggered by very short bursts of invalid INU data.

The option of continuing to use the new INU parameters for the one second interval (rather than coasting with the last valid ones) was rejected for safety reasons. There is a possibility that the new angles are bad.

5. Hardware Installation

5.1. Installation Overview

Vaisala weather radar systems are delivered with RCP pre-installed in the radar cabinent.

For RCP-only deliveries, the hardware installation includes mechanical installation and siting, electrical specifications of the interface signals, system-level considerations, and the standard connector panel.

RCP and IRIS Radar software can be installed on the same PC, the radar server computer, with no hardware changes.

RCP can be connected to weather radar systems from different manufacturers.

To support flexibility, much of RCP8 I/O is configured using software. Because there is nearly no custom wiring, it is easy to insert spare modules and circuit cards.

For information on RCP units and hardware components, see 3. Functional Description (page 15).

For software configuration instructions, see *IRIS and RDA Software Installation Guide*.

Save the original packaging.

5.2. Powering-up Before Connecting to the Radar

To avoid damaging the antenna system, you must perform an initial RCP power-up with no connections to the radar before configuring the safety parameters. This is because RCP must go through an antenna stabilization procedure before you can activate the fail-safe features (see in 6.1. Stabilizing the Antenna (page 41)).

WARNING! Turn off power to RCP before installing or removing PCI boards. For safety, disconnect the line cord before opening the RCP.

CAUTION! When performing the initial RCP power-up, make sure no damage is done to the antenna system.

CAUTION! The circuit boards contain static-sensitive components. Wear a properly grounded wrist strap to handle the PCI boards.

1. Install RCP in its rack on the slides provided by Vaisala.
   For additional structural support, install the rack-mount slide brackets supplied with the unit.
   You can convert a table top unit for rack mount by installing rack mount ears. Install the rack ears with #8-32 flat head screws.
2. Install the connector panel in the rack and cable it to the I/O-62 card in RCP8 using the 1.8 m (6 ft) cable provided.
   See 5.3.2. RCP8 Connector Panel (page 35).
3. Connect the mouse, monitor, and keyboard.
   You use these for local diagnostic and configuration work. They can be disconnected after the installation is completed.
4. Turn on the monitor.
5. Disconnect ALL I/O from the connector panel for the initial power-up.

The connectors are installed later, one-at-a-time, and then configured and tested according to 6.1. Stabilizing the Antenna (page 41).

6. Push the power button on the front panel.
   When RCP8 is powered-up, the Linux operating system boots-up and RCP8 software process starts, first running a set of diagnostic self-tests.

5.3. Installing RCP

1. Prepare the cables described in the following sections:
   - 5.3.1. PCI Card Connections (page 35)
   - 5.3.2. RCP8 Connector Panel (page 35)
   - 5.3.3. Host Computer Serial Interface (page 38)
   - 5.3.4. Configuring Network Connections (page 39)
2. Review the TTY control and monitoring commands in 7.2. TTY Main Menu (page 45) and 8.1. Using TTY Setup Menus (page 59)
   You use these commands when stabilizing the antenna.
3. Connect the cables.
4. Stabilize the antenna. See 6.1. Stabilizing the Antenna (page 41).

5.3.1. PCI Card Connections

Make the direct connections to the PCI card.

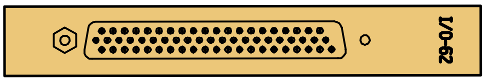

Figure 9 Back Panel PCI Card and Remote Connector Pannel

Table 6 Direct Connections to RCP8 Main Chassis

I/O-62 Connections		
<no label>	DB-62F	Vaisala-supplied cable to IO62/CP remote panel

5.3.2. RCP8 Connector Panel

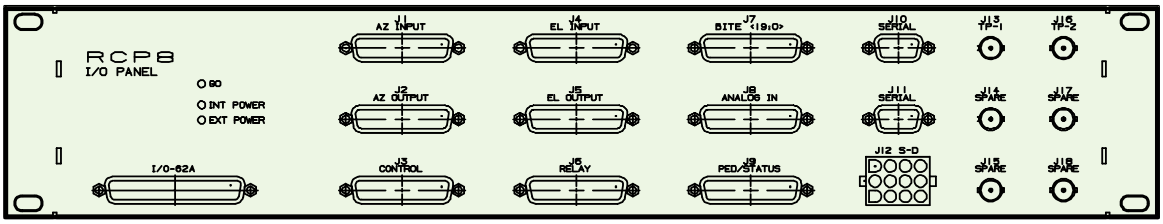

Figure 10 RCP8 Connector Panel

Most connections between the radar and RCP8 are made using the RCP8 connector panel that connects to the I/O-62 with a cable. The panel is usually mounted on the front or the back of the same 19 EIA rack that contains RCP8 chassis. The I/O-62 cable may be plugged into either the front or the back of the connector panel to optimize the cable run.

The connector panel uses a DC-DC converter to convert 12 V unregulated input from the PCI card into regulated +5 V, +3.3 V, and ±12 V to run the main electronics on the panel.

Table 7 RCP8 Connector Panel LEDs

LED	Description
EXT LED	Indicates that 12 V input power is present
INT LED	Indicates that +3.3 V is present

LED	Description
GO LED	Indicates that the panel properly communicates with the PCI card. It blinks slowly when communication is absent and very rapidly during the brief times that the backpanel firmware are being updated with an **`rdaflash`** command. It does not blink when the panel is used by RCP8 software.

The pin assignments to the panel are software-configured in the *`softplane.conf`* file. The labels reflect the default settings for the *`softplane.conf`* file. See *IRIS and RDA Software Installation Guide*.

For information on the I/O for each connector, see 9.6. I/O Connector Panel (page 111).

For information on the default pin-out assignments, see B. I/O-62 Connector Panel Pin Properties (page 169).

Table 8 RCP8 Connector Panel Connectors

Connectors	Description
J1 & J4 -AZ/EL Input: TTL parallel angles	Includes 20 digital inputs or outputs. In the default *`softplane.conf`* file, it is configured for input of 16-bit binary or 16-bit BCD angles. When antenna angle data are required, RCP8 reads the azimuth lines up to 10 times in a row (spaced by 0.5 µs) until 2 successive values compare as equal. This de-bouncing is done so that unsynchronized input data is latched in a valid state. If after, 10 retries, the lines are never observed in a consistent state, the last observed state is used. Sampling for elevation is identical. If fewer than 16 bits are used for the binary angles, then the high order bits should be connected (LSB on pin 1). If a wiring error is made, you can correct it in the *`softplane.conf`* file (for example, LSB and MSB reversed). The BCD format is as follows. • Hundreds digit: Bits 12-15 • Tens digit: Bits 8-11 • Ones digit: Bits 4-7 • Tenths digit: Bits 0-3 For example, if the tenths digits are unused, bits 3-0 are left unconnected but the wiring of the other BCD digits remains unchanged. The higher-order digits must be wired for the elevation axis, even if the elevation position is constrained to a limited angle. If this is not done, the negative angles are be read incorrectly. Input an elevation of -0.1° as 359.9°.
J2 & J5 - AZ/EL Output: TTL parallel angles Angle: TTL output	20 digital inputs or outputs. In the default *`softplane.conf`* file it is configured for output of a 16-bit TTL binary angle. For information on detailed pin assignments, see B. I/O-62 Connector Panel Pin Properties (page 169). This feature can output the parallel angles to a signal processor such as RVP.

Connectors	Description
J3 - Control:	16 lines can be used as differential RS422 or as single-ended TTL input or output. In the default `softplane.conf` file, it is configured for standard status and control I/O. Configuration can be made in groups of 4 with regard to RS422 or Single-ended, I/O sense and input termination of single-ended lines. For more information on pin assignments, see 6.1. Stabilizing the Antenna (page 41).
J6 - RELAY: Control for external equipment	The default `softplane.conf` file makes no assignments to this connector. Often, external equipment in the radar require relay control (for example, power on, radiate on, environmental systems, reset lines, slow polarization switch). This connector has connections for 3 internal relays that are on the connector panel itself. The maximum current through the relay contacts is 0.5 A continuous. The switching load is 0.25 A and 100 V, with the additional constraint that the total power not exceed 4 VA. If larger current and voltage loads are required, the connector panel relays can be used to switch external relays provided by the customer. Another alternative to power external relays is to use the additional 4, 12 V relay signals (up to 200 mA) that are also supported on this connector. **WARNING!** External relays must be equipped with proper diode protection against back-EMF or damage to the I/O-62 and or the connector panel might result.
J7 - BITE: Configurable 20 lines of TTL I/O	Supports 20 lines of TTL each of which can be configured as either input or output. The default `softplane.conf` file configures these as inputs. The inputs are multiplexed into the BITE message to the host computer and can be used internally by RCP8 in control logic equations.
J8 - SPARE: Analog Inputs	The default `softplane.conf` file makes no assignments to this connector. Ten differential analog inputs, up to ±20 V max multiplexed into a single A/D convertor sampling each at >1000 Hz. This can be used to monitor environmental systems at the radar site. Results are put into the Q-BITE (quantitative BITE) message to the host computer. RCP8 can also threshold the Q-BITE numerical values and use the logical results in control logic equations.
J9 - PED/STATUS: RS422 I/O, D/A and A/D	14/7 additional I/O-62 digital lines, 2 each dedicated (non-multiplexed) A/D inputs (±70 V with pot adjust) and D/A outputs (±10 V). For the digital lines, configuration can be made in groups of 4 with regard to RS422 and single-ended, I/O sense and input termination of single-ended lines. In the default `softplane.conf` file, this connector is configured for the differential AZ/EL tachometer inputs, AZ/EL drive outputs and several status variables.

Connectors	Description
J10, J11: RS232C I/O	These are not used in the default `softplane.conf` file. The 2 connectors can be used for serial angle input. The most common format is the RCV01 format (see A.1. Serial Data Format (page 115)), although custom formats from antenna/pedestal manufacturers such as Orbit, Andrew and Scientific Atlanta can also be supported. Note that J11 also has +12 V, -12 V and +5 V regulated power supply outputs for external equipment.
J12 - S-D: AZ and EL synchro input	For systems that have AZ/EL synchro position sensors, RCP8 can accept direct synchro inputs. The nominal voltage and frequency are 90 V @ 60 Hz. S/D conversion is performed in the I/O-62.
J13, J16 - TP1 & TP2: Programmable test point scope outputs	The default `softplane.conf` file makes no assignments to the test points and other BNC connectors. RCP8 offers programmable test points. These are usually used to connect to an oscilloscope. You can then specify the output to the test points in the form of an analog voltage for display on the scope. This can be useful for example to observe the results of logic equations. Technicians can leave the test points permanently connected to a rackmount oscilloscope and then select what is displayed. This saves time and reduces cabling errors when switching test cables.
J14, J15, J17, J18 - SPARE: Spare BNC connections	The default `softplane.conf` file makes no assignments to the test points and other BNC connectors.

5.3.3. Host Computer Serial Interface

RCP8 can connect to a host computer over the COM1 RS232C serial line. The default baud rate is 9600. The connector on RCP8 is on the RCP chassis. On some systems, an RS-232 serial cable may be required to connect to the host computer. On most systems, this is referred to as `/dev/ttyS0`.

Table 9 Host Computer Serial Interface Protocol

Packet Type	Examples	Host Computer Display and Testing Interface
Standard status packets from RCP8 to the host computer Several formats are supported.	Antenna angles and angular speed for AZ and EL, Interlock, Local Mode Switch, and similar.	**Antenna** utility
Standard control packets from the host to RCP8. Several formats are supported.	Position and velocity servo requests, **Radiate On**, and similar.	

Packet Type	Examples	Host Computer Display and Testing Interface
`BITE` packets from RCP8 to the host based on the auxiliary status input bits.	--	**Bitex** utility
Control packets from the host to RCP8 to set the auxiliary control output bits.	--	
`Q-BITE` packets from RCP8 to the host computer.	--	

More Information

- Serial Data Format (page 115)
- Antenna Status Formats (page 119)
- BITE Formats (page 128)

5.3.4. Configuring Network Connections

RCP8 can be configured to listen on a network port over the socket interface using the `AntExport` program. `AntExport` can also run some commands on RCP8.

RCP8 comes with some built-in Vaisala supplied utilities such as **Setup**, **Antenna**, and **Bitex**. See *IRIS and RDA Utilities Guide*.

AntExport

`AntExport` is a daemon program that can be configured to run all the time.

When it receives a socket connection request, it establishes a bi-direction connection to RCP8.

The remote client is normally another computer running the Vaisala antenna library. This remote library contains internal state storing current information about the antenna. This state is slaved to RCP8 state.

- To check if it is running on your RCP8, type: **`$ ps -aef | grep AntExport`**
- During development, start it manually from the shell prompt by typing **`AntExport`**
- For detailed logging, start it with the **`-v`** option .
- `AntExport` defaults to port 30745. If you wish to use another port, start it with an option such as **`-port:12345`**.
- For a list of options, type: **`-help`**

AntExport and Antenna Library Source Examples

The source code for `AntExport` and the antenna library is on the RCP8 release DVD. You can install the code using the upgrade procedure described in *IRIS and RDA Software Installation Guide*.

Table 10 AntExport and Antenna Library Source Examples

File	Description
${IRIS_ROOT}utils/antenna	`AntExport` location.
${IRIS_ROOT}libs/antenna	Antenna library, including example code which talks to `AntExport` in file *ant_iosubs.c*, *ant_rcv.c* and *iant_pwrp.c*. Search for the string `SOCKET`.

Socket protocol

The socket interface transmits to the remote system the commands that change state on the local system. These are all in the form of a sync character, followed by a single byte count, followed by an ASCII command.

More Information

- Case 3: Socket Interface Using AntExport (page 19)

6. Antenna Alignment

6.1. Stabilizing the Antenna

WARNING! In no event shall Vaisala be liable for any damage to the antenna/pedestal system that may occur during stabilization configuration performed by the customer.

After the initial power-up and cabling is complete, align the antenna. A suggested sequence is given below, based on the assumption that none of the parameters are yet correct.

For information on modifying parameter values, see 8.1. Using TTY Setup Menus (page 59).

Most of this configuration is made through the RCP8 chat interface.
You can access this by running rcp8 with **`rcp8 -int`** shell command.
When the host computer interface is correctly configured, you can also access this through the `antx` program on the controlling host.

1. Disable host computer control until the positioning is stabilized by answering `No` in the `site host` section of the **Process incoming servo control packets** question.
   While you are in there, setup the rest of the host computer interface questions.
2. If this is a shipboard system, disable the platform stabilization until the basic antenna control is tuned by answering `No` to the first question in the `INU` section.
3. Use the the **`axis`** command to define most of the fixed information for each axis, particularly the angle source.
4. Temporarily set the elevation shutdown limits 15° short of mechanical stops, leaving enough distance for the antenna to coast to a stop in case of errors.
5. Disable soft limits. These only work when the position servo is configured.
6. Set the limit switch options and polarity.
7. Use the RCP setup utility to configure the `Interface to RCP` section to match the other corresponding configurations.
   Once this is done, the **Antx** and **Antenna** utilities should work.
   Verify that the displayed angles match the RCP8 front panel.

8. On the azimuth axis, do the following:
    a. Set the `maximum output drive voltage` to +/-10 V and do the following:
        - Verify that a drive of 0 cannot move the antenna.
        - Set the drive sign correctly.
        - If you have a high-gain servo, it may go at full speed at a lower voltage, and it may move with a 0 drive (because of a small A/D offset voltage). In this case, you must add an external resister divider to lower the drive voltage. If the voltage needs to be lowered a small amount use the drive voltage question. Vaisala recommends the voltage divider below about 5 V.
    b. Adjust the tachometer voltage range using the gain potentiometers on the back of the CP.
    The potentiometers closer to the end of the CP controls elevation. The `T` units are displayed in RCP8 in a range of +/-100.
    The goal is that the maximum velocity of the antenna should be within the measured limits. Be aware that this speed may exceed the maximum velocity requested by RCP8. Use the **`monitor`** command to watch this, then give drive commands like **`ad 10`** to give small drives.
    Bring the speed up to 25 % of the maximum you expect, then adjust until the `T` display is below 25.
    c. Calibrate the tachometer by getting the sign and offset correct. Then enter level and speeds in the **`axis`** command.
    Set the **Tach zero- delay-smoother window** to a short (0.05 second) value.
    d. Determine the motor starting drives, the nominal drive slope, and the maximum angular velocity.
    Determine the drive sign, then set the drive slew rate fairly short (0.10 second). Set the velocity feedback dead zone to 0.3, and the feedback slope to 25.
    The velocity servo should now be stable.
    e. Set up the position servo parameters to achieve a stable motion between 2 positions that are separated by an angular distance.
    Make sure that both steps of 1° and of 10° perform quickly with no overshoot.
9. Repeat step 8 for the elevation axis.

For most of the elevation axis stabilization, keep the antenna in the middle of the range, away from the stops.

10. Set the final elevation shutdown limits approximately 0.2° short of the mechanical stops.
11. Enable the elevation soft limits, and set them approximately 0.5° short of the shutdown limits.
    In setup RCP section set the elevation limits to the same soft limits.
12. If this is a shipboard system, enable the data from the `INU` section.
    See 6.2. Aligning the Pedestal for Shipboard Use (page 43).

6.2. Aligning the Pedestal for Shipboard Use

In the best cases, the motion reference unit (MRU) sensor is mounted with an orientation exactly matching the antenna pedestal. Unfortunately, this is not always possible.

Perform the following steps while the ship is docked

1. Use a digital level to match the MRU's tilt to the pedestals tilt in both axis.
2. Make the pedestal's azimuth 0 when the MRU's heading is 0.
3. Check that the GPS antenna orientation match.
4. If the inertial navigation unit (INU) data may be shared by many different experiments, use a pitch and roll offset for the INU in RCP8.
   See 6.2.1. Aligning INU and Pedestal for Shipboard Use (page 43).
   You do not need an antenna heading offset, because you can make the zero headings match with a pedestal offset.

The roll, pitch, and heading for antenna stabilization are now correct. It does not really matter if this matches other parts of the ship.

6.2.1. Aligning INU and Pedestal for Shipboard Use

To align the Inertial Navigation Unit (INU) and pedestal for shipboard use, you must enter the following parameters in RCP8:

- Azimuth Axis Input offset from true orientation.
- Elevation Axis Input offset from true orientation.
- INU Roll offset from true orientation.
- INU Pitch offset from true orientation.
- INU Heading offset from true orientation.

1. Set the INU Heading offset to 0.
   Define this as the pedestal 0 azimuth.
2. Get a rough azimuth axis offset by manually pushing the antenna to point in the direction of the ship's heading.
3. Adjust the offset until the pedestal azimuth reads 0 on the **Antenna** utility.

4. Set the elevation angle:
    a. Make sure the INU stabilization is turned on in the RCP.
    b. In the **Antenna** utility, set the elevation angle to 0°.
    c. Place a digital level on the waveguide feed in front of the dish in a place parallel to the transmitted beam.
    d. Slowly sweep the antenna 360°, recording the tilt approximately every 30°.
       Monitor the sweep to make sure RCP maintains the earth elevation angle near zero. The recorded table should also show the pedestal azimuth. Check this in the **Antenna** utility by selecting **Options > Stable Platform Params**.
    e. Plot the resulting data and fit to a sine wave with offset.
       The fit offset gives you a rough elevation axis offset. It is not the exact offset because the beam pattern may not be exactly aligned with the feed waveguide.
        - The sine wave amplitude at 0° gives you the INU Pitch offset.
        - The sine wave amplitude at 90° gives you the INU Roll offset.
5. When you have finished adjusting the offsets, repeat the stabilization measurement scan to check that the corrections are complete.
   You may need to repeat this many times because of operator mistakes, sign confusions, and cross term contributions.
6. Run a sun calibration.
7. Use the results of the sun calibration to fine-tune the pedestal elevation offset, and pedestal azimuth offset.
   Check that the signs are correct.

7. TTY Menu Control and Monitoring

7.1. Starting TTY Menus

1. Depending on your hardware configuration, start the **TTY** menus by entering one of the following commands:
   - `$ rcp8 -int`
   - `$ antx`
     This mode is supported by Vaisala IRIS software and uses spare bandwidth, on the existing host computer serial or network interface, to allow you to communicate with RCP8 from an **XTERM** window.
     After the initial stabilization and setup, this is the preferred method of using the **TTY** menus. You can start the chat mode remotely without additional hardware or cabling.
2. Press ENTER to get to the RCP **TTY** menu prompt:
   The RCP **TTY** menu prompt appears: `RCP>`

7.2. TTY Main Menu

The **Main** menu is top level of communication between you and RCP8 in the TTY Setups. You can access all setup, monitor, and control functions through this menu.

> Type **?** to show a list of available commands.

The following example shows the **Main** menu as it appears on the TTY screen:

```
Available Commands:
  Axis <AZ><EL>                              General axis setup
  Control <Lines><Logic><VarMisc>
  Control parameters and <VarADC><VarAnt>    Boolean variables.
  Help <Support><Listall><View><Debug>       Help text (also '?')
  INU                                        Inertial Navigation Unit
  Monitor <Ang><INU><SIO><Sta><Con><ADC>     Live TTY monitor
  Pservo <AZ><EL>                            Position servo variables
  Reset  <seconds> / Reboot                  Soft/Hard Resets
  Restore <Factory><Saved><Undo>             Restore settings
  Save                                       Save settings
  Site <Display><Host><Custom><Misc>         Local site setups
  Status                                     Status input lines
  Vservo <AZ><EL>                            Velocity servo variables
```

To make a selection, type the command followed by any additional keywords or numerical values. Many commands require additional information.

Use the DELETE and BACKSPACE keys to correct typographical errors. Invalid selections result in a diagnostic message followed by a beep.

For information on commands for setting-up and calibrating RCP8, see 8.1. Using TTY Setup Menus (page 59).

For information on control and monitoring, see 7.3. TTY MONITOR Command (page 46).

7.3. TTY MONITOR Command

The **MONITOR** command provides a live display of changing parameters within RCP8.

You can select several displays, all consisting of a line of information that is continually retyped on the same position of the TTY screen. For terminals operating at 2400 baud or faster, the effect is similar to that of a stationary format display where each value is kept up-to-date.

1. Type user commands while the monitor display is running.
   The effect is as if the TTY cursor were located to the right of the text and the characters appeared in the usual manner. Since the entire status line is continually retyped, the implementation of these echoed characters is more complicated.
   When a valid command is input, the screen scrolls up a line and the status display continues to be printed on the following line, preserving a history of the commands that have been typed.

   > A blank line is a useful command. This no-operation command allows the display to scroll in an upward motion. It also creates a sequential record of observations on the TTY screen, allowing the information to be written down at a later time. This important when calculating the initial measurements of the antenna dynamics as required for the position and velocity servos.

   Invalid commands erase all command characters. The TTY beeps, and no scrolling occurs.

   - Press DELETE or BACKSPACE to correct errors.
   - Press ENTER to terminate the input.

2. When you have typed many commands, the initial heading eventually scrolls off the top of the screen. Enter the **.** command to automatically retype the heading line. The status display continues under it as before.
   Use the **.** command to prevent the misinterpretation of an unlabeled line of numerical information.

3. To view alternate data displays within each monitor command, use the **`ALT`** command to toggle through the different displays, and the **`MAIN`** command to return to the default presentation.
   If you exit from a monitor while an alternate display was in use, you are automatically returned to that display upon reentering.

4. To exit from the monitor command, type **`QUIT`** .
   You return to the **Main** menu.

7.3.1. TTY Antenna Monitor and Control

RCP8 can display most important real-time antenna parameters on the local TTY screen and can request antenna motion through a simple command interface. You can use the local control and monitoring capability during the initial installation and testing of RCP8 or for manual antenna control

To access the local antenna monitor, in the **Main** menu, type **`monitor angles`**.

The **MONITOR** command may be abbreviated to its unique first letter, **m**. The term `angles` is the default value of an optional keyword.

An initial heading is printed, followed by repeated lines of numerical text. For example:

```
RCP> monitor angles
AZ-Pos AzTach Az-Vel AzDrv EL-Pos ElTach El-Vel ElDrv Time
------ ------ ------ ----- ------ ------ ------ ----- ----
141.21  34.81   8.37  32.7  12.01   0.00   0.00   0.0 3.42
```

The displayed values are:

`AZ-Pos / EL-Pos`
: The Azimuth (AZ) position is unsigned and displayed in a 0 to 360° range. The Elevation (EL) position is signed and operates from -180 ... +180°.

`AZTach / ELTach`
: The `AZ` and `EL` tachometer levels represent 12-bit, A/D converter samples scaled to a range from -100 to ...100.

`AZ-Vel / EL-Vel`
: The `AZ` and `EL` velocity are computed as the end-product of the tachometer samples with a calibration slope for each axis. If there is no hardware tachometer then the position is differenced to obtain a virtual tachometer. Note that for the virtual tach, the internal dynamic antenna model is used for interpolation.

`AZDrv / ELDrv`
: The `AZ` and EL`EL` motor drive represents 12-bit, D/A converter values scaled to a range from -100 ... +100.

`Time`
: The seconds counter increments from 0 ... 10 with 0.01 second resolution. These values are included so the elapsed time, between displayed lines, can be easily measured. It is useful when manually calculating the antenna dynamic parameters.

7.3.1.1. Angle Monitor Commands

The following commands are available in the angle monitor:

```
Angle Monitor Commands:
   azd / eld <#>  Set AZ/EL drive (D-Units)
   azp / elp  #   Set AZ/EL position (degrees)
   azt / elt <#>  Set AZ/EL velocity (Tach-Units)
   azv / elv <#>  Set AZ/EL velocity (deg/sec)
         Alt      Switch among alternate presentations
        Main      Back to primary presentation
       Reset <#>  Reset from Shutdown (Unsafe sec)
           .      Reprint header labels
```

Use the following commands to set up drive levels, or to start an the internal servo, for both the azimuth and the elevation axes. The range of -100 ... +100 represent the digital value that is applied to the output D/A converters.

Table 11 Angle Monitor Drive Level Commands

Command	Value	Purpose
ad or **ed**	Number in the range of -100 ... +100	Output a given motor drive.
ap or **ep**	Angle in degrees (°)	Move the antenna to a fixed position.

The host computer serial interface continues to control RCP8 until you type a command that moves the antenna on the TTY screen. RCP8 remains under the terminal's control until you exit the local monitor mode.

The terminal may be used as a monitor however, do not input commands that can seize control from the host computer.

If commands are used to move the antenna, checks are usually performed that restrict the antenna's travel to ensure the soft limits (lower and upper) are not exceeded. The checks are done by executing the position servo silently in the background using the 2 soft limits as target points. If the present motor drive does not rest in between the calculated drives, the drive is automatically overridden by either one of those values. This safety measure prevents the antenna from running into its stops.

7.3.1.2. Alternate Display for Shipboard Platforms

The following alternate format is useful when moving platform stabilization is performed. This allows you to compare the pedestal and the Earth angles as the orientation of the platform changes.

```
Ped AZ/EL     Earth AZ/EL   Earth Vel     Roll   Pitch  Head
------------- ------------- ------------- ------ ------ ------
294.70  -0.98 359.72   9.43 -0.01    4.00 -7.99  -7.76  65.88
```

The displayed values are:

`Ped AZ/EL`
: Pedestal position angles in degrees (°).

`Earth AZ/EL`
: Earth position angles in degrees (°).

`Earth Vel`
: Earth AZ and EL angular velocities in degrees/seconds (°/s).

`Roll/Pitch/Head`
: Roll, pitch, and heading angles of the moving platform in degrees (°).

7.3.1.3. Alternate Display of Antenna Dynamics

The following alternate format prints several derived parameters related to the dynamic properties of each antenna axis. One axis is displayed at a time. The azimuth axis printout is shown below. The alternate display for the elevation axis is identical.

```
AZ-Pos AzTach AzDrv    T-Cal/Vel/Ratio    T-Dot T-Err I-Mom Time
------ ------ ----- ------------------- ------ ----- ----- ----
359.95 -11.67 -2.3  -13.72  -3.25  1.013   2.86  -1.0  2.81 5.29
```

The displayed values are:

`AZ-Pos`
: Pedestal position angle in degrees (°).

`AzTach`
: The represents the Pedestal tachometer levels, scaled to -100 ... +100 T-Units.

`AzDrv`
: Pedestal drive signals, scaled to -100 ... +100 D-Units.

`T-Cal / Vel / Ratio`
: The tachometer calibration values consist of a one-second averaged tachometer calibration level (T-units) and a computed actual velocity based on various positions (°/s). Both numbers define the map from the tachometer T-units to velocities (°/s). The slope ratio, implied by the current values to the stored slope from the axis menu, is displayed. This ratio should be close to 1.000 for all rates of rotation. The antenna must be in motion for these values to be valid.

`T-Dot / T-Err / I-Mom`

The time derivative of the tachometer (that is the acceleration) is displayed in T-units/sec followed by the extrapolated tachometer error in T-units, based on a 2.5-second integration of an internal antenna model. This tachometer error is the basis of an unresponsive antenna check that is continually executed in the background. The antenna's instantaneous moment of inertia is displayed in D-units and T-units/sec. The antenna must be accelerating for these values to be valid.

`Time`

The seconds counter increments from 0 ... 10 with 0.01-second resolution. These values are included for measuring the elapsed time between display lines.

7.3.2. TTY Serial I/O Monitor

Use this display when debugging the serial interface with the host computer. The TTY screen displays the I/O activity and the interpretation of the commands sent to RCP8.

To enter the serial I/O monitor, in the **Main** menu, type **`monitor sio`**

```
RCP> monitor sio
  Ch/Rec In  Time Err  Ch/Rec Out  AZ-Pos  AZ-Vel EL-Pos  EL-Vel
------------ ---- --- ------------ -------------- --------------
154867 11002  0.2   3 698660 11342   0.00 P  0.00   0.00 P  0.00
```

The displayed values are:

`Ch / Rec In`

The character input count represents the total number of characters received. The valid record count represents the number of properly formatted packets received.

`Time`

Time since the last valid record was received (s).

`Err`

The error count represents the total number of improperly formatted packets received.

`Ch / Rec Out`

The character output count represents the total number of characters and packets transmitted.

`AZ-Pos / AZ-Vel`

The requested azimuth position and azimuth velocity and are displayed regardless of the servo type. Other letters that may appear between the two values include:

- `P` - position servo
- `V` - velocity servo
- `D` - direct motor drive
- `X` - disabled

`EL-Pos / EL-Vel`

Requested elevation position and velocity, in the same format as for azimuth.

7.3.2.1. Serial I/O Monitor Commands

If you type command characters while the TTY screen displays the status text, the characters are echoed on the right side of the screen.

The commands are available within the Serial I/O Monitor are:

```
SIO Monitor Commands:
    Alt    Switch among alternate presentations
   Ri/Ro   Host computer record In/Out monitor
   Main    Back to primary presentation
   Zero    Clear SIO counters
      .    Reprint header labels
```

7.3.2.2. Alternate Displays of Raw SIO Records

You can use the **Ri** and **Ro** sub-commands to view the incoming and outgoing raw serial traffic with the host computer. This can be helpful when debugging interface problems at either end.

The data are shown in hexadecimal format, one (variable length) record per line. Note that the only data shown are character sequences that 1) begin with a byte with MSB set but not equal to 0xFF, 2) end with 0xFF, and 3) have MSBs clear in all intermediate bytes. For example:

```
Incoming Records from Host Computer
-----------------------------------
80 00 00 00 00 00 0A 00 0F 00 00 00 00 FF
80 00 00 00 00 00 0A 00 0F 00 00 00 00 FF
C0 01 00 00 02 00 00 00 00 00 00 00 FF
80 00 00 00 00 00 0A 00 0F 00 00 00 00 FF
C0 4D FF
80 00 00 00 00 00 0A 00 0F 00 00 00 00 FF

Outgoing Records to Host Computer
---------------------------------
80 00 00 00 00 00 00 00 00 10 00 00 00 24 30 FF
80 00 00 00 00 7F 7F 00 00 10 00 00 00 34 33 FF
C0 00 00 00 00 00 00 00 00 10 00 00 FF
C0 01 7F 7F 7F 7F 7F 7F 3F 00 00 00 FF
80 00 00 00 00 00 00 00 00 10 00 00 00 4B 36 FF
C0 4D FF
80 00 00 00 00 00 00 00 00 10 00 00 00 34 65 FF
```

7.3.3. TTY Inertial Navigation Unit Monitor

This display provides a view of the data stream arriving from an optional Inertial Navigation Unit (INU). To enter the INU monitor, in the **Main** menu, type **`monitor inu`**.

```
RCP> monitor inu
Roll    Pitch  Head  R.Dot P.Dot H.Dot   Time       Date
------ ------ ------ ----- ----- ----- -------- -----------
-1.04    4.74 345.96   0.6   3.1   8.3 00:27:58  1-Jan-1998
```

The displayed values are:

`Roll/Pitch/Head`
: Attitude angles in degrees (°).

`R.Dot / P.Dot / H.Dot`
: Rates of change of attitude angles in degrees (°) / second.

`Time/Date`
: Time and date, using the time zone set for the INU.

7.3.3.1. Alternate INU Monitor Presentations

To switch to the following alternate presentation, type **`alt`**:

```
 Latitude Longitude Height N.Vel E.Vel Z.Vel Char/Err Rec/Err
--------- --------- ------ ----- ----- ----- -------- -------
 42 31.0N  71  2.4W   40.9  10.0   3.0   0.5    0   0 161   0
```

The displayed values are:

`Latitude/Longitude/Height`
: Represent the physical location. Latitude and Longitude are in degrees and minutes, with N/S and E/W indicating the sign. Height is in meters relative to sea level.

`N.Vel / E.Vel / Z.Vel`
: Represent the linear velocities in meters/second the North, East, and Up directions.

`Char/Err and Rec/Err`
: The counts of the number of characters and records received, and the number of character and record errors that have been detected. A character error is a framing or parity error, whereas a record error results from an invalid CRC checksum in a record of data.
The record count should increase at a rate of approximately 100 records/second when INU data are being received correctly.
Use the **`zero`** sub-command to clear these counters so the changes are easier to spot.

7.3.4. TTY Status Line Monitor

This display provides a view of the status input lines sensed by RCP8. To enter the status line monitor from the **Main** menu, type **`monitor status`**.

```
RCP> monitor status

Hardware Electrical Inputs

Locl Pw1 Pw0 Rad Srv T/R Stby Intr Mag Air WGp Res ElLO ElHI IRIS DRCP

On
```

The characters "-" are printed under each unused status input. For the used inputs, the word `ON` is printed if the line is asserted, and blank space appears if the line is not asserted.

To switch to an alternate presentation that displays the internal status of each condition, type **`alt`**. This is different from the condition of the hardware input line because the status may be coming from another source, or may be spoofed from the requested control.

```
RCP Internal Status

Locl Pw1 Pw0 Rad Srv T/R Stby Intr Mag Air WGp Res ElLO ElHI IRIS

ON 000
```

If auxiliary status lines have been enabled, you can switch to the following bit presentation by typing **`alt`**. In the following example, 4 bytes of optional status have been selected in the **site custom** menu. High inputs are shown as a `1`, and low inputs are shown as a "." .

```
Qualified Auxiliary Status Bits
S[63:56]   S[55:48]   S[47:40]   S[39:32]   S[31:24]  S[23:16]   S[15:8 ]
S[ 7:0 ]
--------    --------    --------    --------    --------   --------
--------    --------
```

The **`Monitor Status`** command uses the **/** sub-command to toggle between the requested and qualified versions of the primary and auxiliary status bits, as well as the direct hardware inputs themselves.

The distinction between requested and qualified status bits exists because the status bits can appear on the left side of logic equations (See 8.8.2. Logic Equation Control Qualifiers (page 91)).

7.3.5. TTY Control Request Monitor

This display shows the control functions handled by RCP8.

To enter the control request monitor, in the **Main** menu, type: **`monitor control`**

The primary control functions that have been externally requested (usually from the host computer) are shown in the following display:

```
Requested Primary Control Bits
Pw1 Pw0 Rad Srv T/R Res IRIS
--- --- --- --- --- --- ----
             ON          000
```

To view the qualified state of each control function, give the **/** sub-command. This shows the actual control state, which may be different from the requested state if any internal logic equations override the request. See 8.8.2. Logic Equation Control Qualifiers (page 91).

```
Qualified Primary Control Bits
Pw1 Pw0 Rad Srv T/R Res IRIS
--- --- --- --- --- --- ----
             ON  ON      000
```

The **/** sub-command works as a toggle between the requested and qualified states of the shown control variables. This makes it easy to compare the bits, and to verify that custom logic equations are implemented correctly.

Use the **`alt`** sub-command to switch to a display of requested auxiliary control bits:

```
Requested Auxiliary Control Bits

C[63:56] C[55:48] C[47:40] C[39:32] C[31:24] C[23:16] C[15:08] C[07:00]

........ ........ ........ ........ ........ 11...... 1....... ........
```

from which the **/** sub-command can switch to the qualified states:

```
Qualified Auxiliary Control Bits

C[63:56] C[55:48] C[47:40] C[39:32] C[31:24] C[23:16] C[15:08] C[07:00]

........ ........ ........ ........ ........ ........ 1....... ........
```

Use the following **alt** display to view timers and local logic variables:

```
Local Variables

V[15:08] V[07:00]

........ ........
```

```
Timers and Local Variables
T[15:8 ]   T[ 7:0 ]    V[31:24]   V[23:16]   V[15:8 ]   V[ 7:0 ]
........   ........    ........   ........   ........   ........
```

7.3.6. TTY Analog Voltage Input Monitor

Use the **Monitor ADC** command to view the sampled voltage on each analog input line. An internal loop-back measurement of the AZ and EL drive output voltages is also included. For example:

```
RCP> monitor adc
Analog Input Lines (Differential Input Voltage)
  0     1     2     3     4     5     6     7     8     9
----- ----- ----- ----- ----- ----- ----- ----- ----- ----- -----
-0.00 -0.00  0.00  4.39  9.25  9.18 11.51  9.13  3.53  0.00
```

7.4. TTY RESET Command

RCP performs continuous antenna consistency checks to guard against faults that could damage the mechanical system. When RCP detects such faults, it immediately enters a shutdown state.

WARNING! You must determine and correct the cause of the shutdown before restoring system operation.

The **RESET** command provides a restore capability that is more graceful than cycling the power by doing the following:

- Always places the controller in its momentary unsafe condition regardless of whether RCP is shutdown when the command is received. This means you can use the command to exit from stuck conditions, even if RCP has not actually shutdown.
- Causes a soft internal reset. The shutdown state is cleared and RCP8 continues running smoothly.

Exit a Shutdown State

To exit the shutdown state, do one of the following:

- In the **Antenna** of the host computer, issue the **`reset`** command.
- In the local computer, type the local TTY command, **`reset`**.
  The local **RESET** command may be followed by an optional numerical value between 0 and 10. This value represents the number of seconds that a shutdown is inhibited following the reset, with a default value of 1 second. This brief lockout period assists with the antenna's reposition so the shutdown condition can be remedied immediately following the reset. For example, if the antenna has contacted a limit switch, you can issue brief drive commands and attempt to move the antenna away from its limit.
  RCP8 only shuts down when it has control of the antenna. When the external `LOCAL` status input forces RCP8 into local mode, it does not shutdown even if the velocity limits are exceeded or if the tachometer signals are inconsistent with angular positions. When control returns to RCP, you must must ensure that no shutdown criteria is pending prior to the switch-over.

The `LOCAL` status places no restrictions on exiting from the shutdown state - only on entering it. Therefore, the **`reset`** command is always effective.

- If RCP requires a more drastic restart, turn the power on and off to reboot the system.

7.5. TTY Help View Command

Use this version of **`help view`** to view internal status and configuration. For example:

```
RCP> help view
Board Configuration and Status

------------------------------
RCP8 Radar Control Processor V13.6 IRIS-8.13.6
  Settings were last saved using V13.6
  RCP8 started at: 07:46:03 21 JUN 2016
  Current time is: 10:14:44 19 AUG 2016

Physical hardware inventory:
  Found PCI Card I/O-62  - Rev.B:1  Serial:4083
      Code:30 (/dev/rda/io62-0)
      \--> IO62CP Backpanel - Rev.B:3  Serial:2761  Code:4
          ( Supply Currents - Panel: 351 mA,  Relays: -15 mA )

Parallel execution threads:
  CS-Tick - PID:1571   Priority:12  Policy:RealTimeRR
    Servos - PID:1571   Priority:13
      Policy:RealTimeFIFO
    Watchdog - PID:1571     Priority:11  Policy:RealTimeRR
    Host-RCV - PID:1571     Priority:11  Policy:RealTimeRR
    Host-XMT - PID:1571     Priority:11  Policy:RealTimeRR
    Host-NET - PID:1571     Priority:11  Policy:RealTimeRR
    Canbus/Main - PID:1571  Priority:12  Policy:RealTimeRR
    Dehydrator - PID:1571   Priority:11  Policy:RealTimeRR

Shared library build dates:
  RCP8/Core: Thu May 19 11:09:02 UTC 2016
  RCP8/Open: Thu May 19 11:10:03 UTC 2016
  RCP8/Site: Thu May 19 11:09:03 UTC 2016

AZ Axis - Pos:   0.00   Off:  -0.00   Vel:  0.0
EL Axis - Pos:   0.00   Off:  -0.00   Vel:  0.0
```

The list includes:

- Board and code revision levels, and the date and time that the code was compiled.
- Inventory of used PCI cards.
- List of currently running threads.
- Current angle offsets that are being added to the parallel or synchro angle inputs. This value generally comes from the **`Axis`** setup command; but in some cases it may be supplied by external equipment.

8. TTY Setup Menus

8.1. Using TTY Setup Menus

You setup the configuration parameters interactively using the TTY.

> The software configuration parameters are in the */usr/sigmet/config/rcp8.conf* file.
> For information on the parameters, type: **`RCP > help view`**

1. To change a parameter setting, see to the summary dialogs in **`RCP > help view`** to determine the general category in which the parameter appears.
2. In the **Main** menu choose the general category. Type additional information and prompts as appropriate.
   The parameter values are displayed for each category and RCP pauses for the input.
3. For each question, do one of the following:
   a. If the current value is correct, press ENTER.
      If a new value is required, enter the new value. Make sure you enter the numeric values in the proper physical units. For example, do not input the time in seconds if the question expects the time in minutes. RCP8 displays the expected units with each question.
      When you press ENTER, RCP8 echoes the new value to verify that it is correct.
      If the new value is correct, press ENTER again to proceed to the next parameter.
   b. If needed, type **up** or **u** to return a previous question.
   c. Type **`quit`** or **`q`** to exit a submenu and return to the `RCP>` prompt.
   d. Press ESC to exit the **SETUP** menus.

   When you have answered all the questions, RCP runs the new values and returns to the command prompt.
4. Select another **SETUP** menu command or run RCP with its new settings to verify that the changes are correct.
5. When you are satisfied with the changes, type **SAVE**.

8.2. SAVE Command

Use the **SAVE** command to store the current RCP8 parameters in the non-volatile RAM.

This automatically preserves the settings the next time RCP8 is powered up.

The **SAVE** command prints a message that counts the number of changed bytes.

8.3. RESTORE Command

Use the **RESTORE** command to replace the current working parameters with a different set.

Use the **factory** argument to restore conservative default values for all parameters. This can be useful to return to a known baseline.

Use the **saved** argument to restore parameters from the most recent **SAVE** command.

Use the **undo** argument to change your mind if the **factory** or **saved** arguements inadvertently overwrite the desired settings.

8.4. SITE Command

Use the **SITE** command to configure parameters for the local site. The sub-menus include:

- `Display` — defines how the vacuum-fluorescent, front panel display is configured and represents the default menu if the **SITE** command is invoked with no argument.

> Unavailable for new installations.

- `Host` — defines the communication choices and data protocols that are used with the host computer.
- `Custom` — selects and configures customer specific features.
- `Miscellaneous` — defines miscellaneous items.

8.4.1. Host Computer Setups

> The set-up questions listed in your configuration vary depending upon your set-up and how your have responded to previous questions.

To access these questions, in the `RCP>` prompt, type: **`Site Host`**

Connection type for host computer I/O: `Network`

Multicast address: `224.0.0.3`

Port number: `30785`

Network interface: `lo`

Baud rate for host computer I/O: `9600`

This sets the baud rate for serial communication with the host computer. Available choices include 1200, 2400, 9600, and 19200.

Data format transmitted by host computer: `XMT05`

Data format received by host computer: `RCV05`

RCP8 can send and receive many serial protocols. Use these questions to match the transmit and receive protocols that coincide with the host computer.

Process incoming servo control packets: `YES`

Answering **`NO`** to this questions disables the interpretation of control packets that are received by RCP8, while still allowing BITE interrogation packets and chat packets to be treated normally. This makes it easier to setup and analyze the antenna servos using the chat mode on the host computer.

RCP8 transmission rate: `2.50 records/sec`

Antenna data packets are transmitted by RCP8 at this fixed rate. Choose a rate that makes the best tradeoff between:

- Providing the host computer with up-to-date angles
- Relieving the host computer of unnecessary I/O flooding

RCP8 transmits Time-of-Day records: `YES`

Time between Time-of-Day records: `30 sec`

You can optionally transmit the INU's time-of-day to the host computer if RCP8 is connected to an INU. Typically, this process is executed every few minutes.

RCP8 transmits internal BITE packets: `YES`

ID of internal BITE packets: `0x01`

This question decides on the transmission of internal BITE status packets. These should be enabled if the host computer must monitor any of the information contained in these packets.

RCP8 transmits AUX status BITE packets: `YES`

Xmt ID of status BITE packets: `0x02`

RCP8 receives AUX control BITE packets: `YES`

RCP8 transmits analog voltage Q-BITE packets: `YES`

ID of analog voltage Q-BITE packets: `0x06`

Simulate the incoming channel voltages: `NO`

Rcv ID of control BITE packets: `0x03`

Set-up the transmission and reception of auxiliary status and control BITE packets.

These setup questions for the auxiliary control and status bits are organized so that the associated host computer I/O is configured independently of the (optional) assignment of hardware electrical lines to those bits. These questions ask whether RCP8 send/receive auxiliary status/command BITE packets to/from the host computer. The `C[0:63]` and `S[0:63]` variables have many different uses within RCP8, so all 64 bits are always included in the 13-byte fixed-format BITE I/O packets. See related questions under **Site Custom**. See 8.4.2. Customer-Specific Site Setups (page 62).

Dead-Host-Computer detection time: `5.0 sec`

This determines the required I/O inactivity time before RCP8 disables the antenna motion. Once RCP8 is under the control of the host computer, the computer may crash or the program, which interacts with RCP8, may cease to function for other reasons. In such cases, it may be important to not allow the antenna servos to operate in accordance with the last computer command. For example, if the host computer requested a large antenna velocity prior to crashing, it makes little sense to continue honoring that request.

Default user interface: `REMOTE-HOST-CHAT`

The local TTY I/O can either be with the plug-in TTY or with the host computer with the chat-mode packets during the initial RCP8 power-up. This question sets the power-up default.

> RCP8 freely toggles between the TTY and the chat-mode channels. The interface, that most recently received incoming characters, is automatically assigned for the subsequent I/O.

8.4.2. Customer-Specific Site Setups

Use the following questions to select and configure customized RCP features.

To access these questions, in the `RCP>` prompt, type: **`Site Custom`**

Output serial TAG lines: `YES`

Serial port: `/dev/ttyS0`

Baud rate of the serial TAGs: `9600`

Choose: `None RCV01 RCV02 RCV03 RCV05`

Serial TAG data format: `RCV01`

Normally RCP8 outputs azimuth and elevation TAG angles as 16-bit parallel TTL outputs on the back panel. Use these questions to configure an optional serial output stream.

Use WSR-88D DCU Interface (Antenna/Pedestal): `YES`

Serial port: `/dev/ttyS0`

Baud rate for DCU: `19200`

Choose None Odd Even

DCU data parity: `Odd`

Bits in position binary angles: `13`

ID of DCU BITE status packets: `0x09`

ID of DCU Self-Test-1 BITE packets: `0x05`

ID of DCU Self-Test-2 BITE packets: `0x06`

Spare adjustment for Azimuth : `1.00000`

Spare adjustment for Elevation: `1.00000`

Additional time lag for Azimuth: `0.000 sec`

Additional time lag for Elevation: `0.000 sec`

Simulator port:

For NEXRAD systems, use this section to configure the DCU interface. You must set the angle source questions in the axis sections to `Custom` to read these angles.

Use WSR-88D DAU Interface (BITE/Status): `YES`

Serial port: `/dev/ttyS0`

Baud rate for DAU: `19200`

Choose None Odd Even

DAU data parity: `Odd`

ID of DAU standard BITE packets: `0x07`

ID of Quantitative BITE packets: `0x08`

Simulator port:

For NEXRAD systems, use this section to configure the DAU interface.

Use RVP9/IFD digital/analog I/O interfaces: `NO`

Use Kavouras TCU Interface (Radiate/BITE): `YES`

Serial port: `/dev/ttyS0`

ID of TCU standard BITE packets: `0x09`

ID of Quantitative BITE packets: `0x0A`

Simulator port: `/dev/ttyS1`

Choose which serial port, and ID for the `BITE` and `Q-BITE` packets are associated with the TCU. Note that the serial baud rate, parity, and stop bits are fixed at 9600/Odd/2 because they are fixed by the TCU.

Use the built-in simulator for debugging the main code. You can watch the live I/O from a real TCU using the **`Monitor SIO`** command followed by **`Raw rTcu`**.

The standard BITE packet for the TCU is 13 bytes long, and maps the 64 TCU status bits into the first 64 packet bits. The timeout bit (no TCU communication) appears in the MSB of Byte #12. These 70 bits of BITE status (10 words of 7 bits apiece) are mapped into status bits S64-S133. If you want to use any of the TCU status bits in a logic equation, those are the variables to grab.

The Q-BITE packet is 27 bytes long and holds 12 14-bit values. The first two are the `Max strike` and `Current strike` counts from the TCU status packets, and the next 8 are from the temperature report. The last two slots (11 and 12) are unused for now.

The TCU is reset using the standard BITE resetting mechanism. A BITEX reset that is directed at either the BITE or Q-BITE unit sends a reset command to the physical TCU. In addition, a rising edge on Control Variable C63 also resets the TCU.

The `TrPower` and `Radiate` control/status bits are the only ones needed for the TCU, giving you the states `OFF`, `STANDBY`, and `RADIATE`. When you toggle those two control bits, the appropriate commands are sent to the TCU. Likewise, status from the TCU sets the `TrPower` and `Radiate` status bits.

Use Radtec XCM Interface (Radiate/BITE):`YES`

Serial port: `New Value`

ID of XCM standard BITE packets: `0x09`

ID of Quantitative BITE packets: `0x0A`

offset of 70 mapped status bits: `64`

Use IPA15HC Servo Amp interface: `YES`

AZ IP Address: `New Value`

EL IP Address: `New Value`

ID of regular BITE packets: `0x0C`

offset of 133 mapped status bits: `128`

Use Andrew-Canada serial pedestal interface: `YES`

Serial port: `/dev/ttyS0`

ID of Andrew BITE status packets: `11(decimal)`

Map Andrew status into S[29:63] variables: `NO`

Apply boresite offsets to position angles: `NO`

Simulator port:

When the Andrew interface is enabled you may hookup the serial lines either through a standard Linux TTY port such as `/dev/ttyS0`, or through the special `io62-tty0` serializer that is built into the IO62 card firmware . RCP8 also contains a (minimal) serial simulation of a real Andrew ACU, which you can configure onto a TTY port for loopback testing.

Normally RCP8 receives high-speed parallel AZ/EL angles from the Andrew ACU. However, if you set the axis angle source questions to custom, it sets RCP8 to use the low-speed (5 Hz) serial angle status information from the ACU instead. This option can be useful during testing.

Use Applied Systems TWT Transmitter: `YES`

Choose: `177 337 377`

Model number of the transmitter: `337`

Serial port: `/dev/ttyS0`

Offset of 29 mapped status bits: `64`

Offset of 5 mapped control bits: `64`

ID of Standard STS BITE packets: `0×0C`

ID of Quantitative BITE packets: `0×0D`

Simulator port:

When the Applied Systems interface is enabled you may hookup the serial lines either through a standard Linux TTY port such as `/dev/ttyS0`, or through the special `io62-tty0` serializer that is built into the IO62 card firmware. RCP8 also contains a serial simulation of a real Applied Systems TWT, which you can configure onto a TTY port for loopback testing. A blank device disables the simulation.

The different Applied Systems model vary in the status bytes returned.

Model 177 is: STX Digital Byte 1, Digital Byte 2, Digital Byte 3, Digital Byte 4, Analog 1 (4 characters), Analog 2 (4 characters), Analog 3 (4 characters), EXT Checksum. Total = 19 bytes

Model 377 has a fourth Analog value, for a total of 23 bytes, while model 337 also has a fifth Analog value, for a total of 27 bytes.

The analog fields translate into a `Q`-byte packet, as shown below. Depending on the model, the last values are missing, and the packet is shorter.

Table 12 Q-byte Packet

Char	Function
1	SYNC Byte (AF Hex)
2	Identification byte (User Choice)
3-4	Analog 1 (14-bits)
5-6	Analog 2
7-8	Analog 3
9-10	Analog 4 (for Model 377 and 337 only)
11-12	Analog 5 (for Model 337 only)
13	END OF MESSAGE (FF Hex)

Use Orbit serial pedestal controller: `YES`

Serial port: `/dev/ttyS0`

Offset of 33 mapped status bits: `64`

ID of Standard STS BITE packets: `0×0E`

Fixed time lag of Orbit angles: `1.0 ms`

Simulator port:

Used for the Orbit pedestal controller. A blank device disables the simulation. Be sure to set the angle source questions in the axis sections to read these angles.

Use Dual-GigaCom serial Tx modulators: `YES`

Baud rate of the GigaCom interface: `19200`

Serial port for Tx-A: `New Value`

Serial port for Tx-B: `New Value`

ID of BITE status packets for Tx-A: `0x05`

ID of Q-BITE status packets for Tx-A: `0x06`

ID of BITE status packets for Tx-B: `0x07`

ID of Q-BITE status packets for Tx-B: `0x08`

Fault mask for status bits <15: 0>: `0x77F6`

Fault mask for status bits <31:16>: `0x0004`

System switchover transition time: `2.0 sec`

Debounce time for error status bits: `0.0 sec`

GTX_RESET also resets active system: `NO`

Control bit for gtx_auto: `16`

Control bit for gtx_reqb: `17`

Control bit for gtx_hold: `18`

Control bit for gtx_reset: `9`

Status bit for gtx_trans: `20`

Status bit for gtx_chanb: `21`

Status bit for gtx_fault: `22`

Use CAN-Bus serial control/status: `YES`

Use CAN-Bus to radar control: `YES`

Force shutdown for unresponsive antenna: `YES`

ID of Standard STS BITE packets: `0×0F`

ID of Quantitative BITE packets: `0×10`

Used for the CAN-Bus interface to control and monitor the Vaisala pedestal. Be sure to set the angle source questions in the axis sections to Canbus to read these angles.

Use dehydrator connected in serial port: `YES`

Serial port: `/dev/ttyS0`

ID of Standard STS BITE packets: `0×11`

ID of Quantitative BITE packets: `0×12`

Monitor the status of the ADH-2A COM Automatic Air Dehydrator through an RS-422 interface. Dehydrator status includes the waveguide pressure, device temperature, duty cycle and alarms and warnings for fault states.

Use klystron connected in serial port: `YES`

Serial port: `/dev/ttyS0`

ID of Standard STS BITE packets: `0×14`

ID of Quantitative BITE packets: `0×15`

Control bit for fault log print: `30`

Monitor the status of the Klystron transmitter in Vaisala WRK100 or WRK200 weather radars. The klystron transmitters voltages, currents and alarms are monitored. The status can be displayed on **Bitex**. This software thread retrieved the status and measurement information through a serial interface (RS-422) and creates `ITE` and `QBITE` packets from it. For information on the packet formats, see A.5.7. Klystron BITE Packets (page 138).

Use Power Monitoring: `YES`

Frequency of the radar (MHz): `5625`

Number of sensors: `4`

Ser.No of horizontal forward sensor: `0`

Ser.No of horizontal reverse sensor: `0`

Ser.No of vertical forward sensor: `0`

Ser.No of vertical reverse sensor: `0`

Offset of 2 mapped status bits: `162`

Control bit of rsenssor zeroing: `31`

ID of Standard STS BITE packets: `0×16`

ID of Quantitative BITE packets: `0×17`

The purpose of the `Power Monitor` thread is to open a connection through USB to the Rohde&Schwarz NRP-Z51 power sensors. The weather radar may have 2 power sensors for each polarizations on the wave guides measuring transmitted and received power level. When measurement is completed, `BITE` and `QBITE` packets are created. For information on the packet formats, see A.5.9. Power Monitor BITE Packets (page 141).

Use ARA ACU-3 Antenna: `YES`

Serial port: `/usr/sigmet/config/rcp8_ara_acu-y`

Baud rate: `19200`

Choose: `None Odd Even`

Data parity: `Odd`

ID of BITE status packets: `0×08`

Fixed time lag of angles: `1.0 ms`

Poll for position every 20 ms

Offset of 15 mapped status bits: `40`

Control bit for reset: `40`

Simulator Serial port: `/usr/sigmet/config/rcp8_ara_acu-x`

If the ARA ACU-3 interface is enabled and you supply a serial port, you get the `ARA_ACU3` thread visible on the help view screen. If you put a string into the Simulator port, you get the `ARA_ACU3-Sim` thread. The example above show how to configure the simulator to talk to the main thread using FIFOs. You need to create these 2 files using the **`mkfifo`** command.

There are 15 TSC TWT status bits output in the BITE packet. See A.5.1. ARA ACU-3 BITE Packet (page 129). For a description of the bit meanings and the command set see the ICD.

These same 15-bits are mapped to the specified status bits. The period at which RCP8 polls the ARA for position is set by the `Poll` for position question. All other activity, like polling for status happens once a second. Command output happens once a second unless there is a change.

Use TSC TWT Interface: `YES`

T/R Serial port: `/usr/sigmet/config/tsc_tr-y`

Modulator Serial port: `/usr/sigmet/config/tsc_mod-y`

ID of BITE status packets: `0×06`

ID of QBITE status packets: `0×07`

Offset of 23 mapped status bits: `20`

Offset of 10 mapped control bits: `20`

Simulator T/R Serial port: `/usr/sigmet/config/tsc_tr-x`

Simulator Mod Serial port: `/usr/sigmet/config/tsc_mod-x`

This is the TSC TWT transmitter used in the NOAA G4 aircraft. If you answer **`Yes`** to the initial question and supply either serial port, you get the TSC-TWT thread visible on the help view screen. If you put a string into either simulator ports, you get the `TSC-TWT-Sim` thread. The example above show how to configure the simulator to talk to the main thread using FIFOs. You must create these 4 files using the **`mkfifo`** command.

You can monitor the traffic transmitted and received from these 2 serial lines using the monitor **`sio`** command. Once you are in monitor mode, then type something like **`raw xtsc_tr rtsc_tr`**. Other available data is **`xtsc_mod`** and **`rtsc_mod`**.

For information on the `TSC TWT BITE` and `QBITE` packets, see A.5.11. TSC TWT Packets (page 142).

The TSC TWT simulator is fairly simple:

- For the T/R port, it sends a 9-byte response containing all zeros, except for the first and last byte, and bytes 2 and 3 are copied from bytes 2 and 3 of the command (which have the same meanings).
  The qualitative values are set to: Frequency code=50, Receiver protect leakage=100, Transmitter power=150, and Reflected power=200.
- If no command, or a bad command arrives, then the whole payload is zero. For the Modulator port, it sends the string `<1R0011?0000>\n`, where the `?` is set to 0 or 1 based on the command supplied.
- If no command, or a bad command arrives, then the payload is all 0.

For best performance on an ARA controller, get the option for parallel outputs. If not, or if using a simulator, be sure to set the angle source questions in the axis sections to `Custom` to read these serial angles.

Use TDRS pedestal angle input: `YES`

Serial port: `/dev/ttyS0`

The TDRS pedestal has a serial interface to get angle information.

Use TDRS pedestal control output: `YES`

IP Address: `191.165.99.99`

Port number: `32767`

The TDRS pedestal is controlled via a socket interface. You can set the IP and port number here.

Use remote signal generator control: `YES`

Choose: `Ethernet GPIB`

Signal generator interface: `GPIB`

HPIB device name: `siggen`

RF/IF Signal generator is on the bus : `YES`

Signal generator has pulse modulation : `YES`

Use MELCO TKY01 Serial Q-Bite: `YES`

Serial port: `/dev/ttyS0`

Baud rate: `19200`

ID of Quantitative BITE packets: `0×13`

Simulator port:

You can read serial `QBITE` packets in `MELCO Turkey-01` format. This message includes information from a generator and produces a `QBITE` packet. See A.5.8. MELCO Packets (page 140)

Use remote signal generator control: `YES`

Ethernet GPIB Signal generator interface: `Ethernet`

Signal generator IP Address: `10.0.2.10`

Signal generator Port Number: `5025`

Answer the first question `YES` if a USB-to-HPIB interface module is plugged into a USB slot of the RCP8 computer, in which case its Linux device name is supplied on the next line. If a signal generator is attached to the bus, enter its HPIB address.

The first class of instruments supported are RF/IF signal generators. RCP8 can both control and sense the signal generators output power level, output On/Off switch, and pulse modulation selection. These parameters are then directly accessible from IRIS/Antenna utility.

RCP8 always keeps the signal generator in its normal local mode, and polls its settings every 0.5 seconds. This means that the signal generator's front panel is always fully functional. However, if RCP8 detects a change in the host computers requested settings, the changes are sent immediately to the signal generator. The correct settings are put in place, but you can still make further changes using the manual controls. The signal generator should appear to operate normally, unless changes are requested by the host computer.

When an HPIB signal generator is not installed on RCP8, the signal generator status sent back to the host computer is spoofed from whatever `siggen` settings the host computer is currently requesting. The `RF-Level`, `On/Off`, and `Cont/Pulse` status are all echoed back, and the fault status is `FALSE` (no fault).

For HPIB/GPIB support, you must install a new library for RCP8 to run. If you are installing a new system, this is covered in the `sigconfig` script, or in the steps described in *IRIS and RDA Software Installation Guide*. If you are upgrading, you must install a new rpm. This is supplied on our FTP site, and on the CDROM. The installation command is:

`# rpm -Uhv linux-gpib-lib-3.2.09-1.EL.i686.rpm`

If you use the GPIB feature, you must install the kernel module. There is a common kernel module rpm, and a version specific to the installed kernel. Vaisala provides the driver RPMs for CentOS7:

```
# rpm -Uhv linux-gpib-kmod-common-3.2.09-1.EL.i686.rpm
# rpm -Uhv kmod-linux-gpib-3.2.09-1.EL.2.6.9_5.EL.i686.rpm
# rpm -Uhv linux-gpib-kmod-common-3.2.09-1.EL.i686.rpm
# rpm -Uhv kmod-linux-gpib-smp-3.2.09-1.EL.2.6.9_5.EL.i686.rpm
# rpm -Uhv linux-gpib-kmod-common-3.2.09-1.el5.i686.rpm
# rpm -Uhv kmod-linux-gpib-3.2.09-1.el5.2.6.18_8.el5.i686.rpm
```

Use Pulse Systems TR-1163 Ethernet Interface:`YES`

IP Address: `New Value`

Port Number: `23`

ID of BITE status packets: `0x20`

ID of QBITE status packets: `0x21`

Offset of 21 mapped status bits: `166`

Control bit for transmitter reset: `0`

Use EEC DDC pedestal control interface: `NO`

Use RPM pedestal control interface:`NO`

Use Dual/Redundant system configuration:`NO`

Generate trigger sector blanking output: `YES`

Hardware output line to use: `None`

Hardware input line to use: `None`

Include sector #1 in overall test: `YES`

Sector #1 uses pedestal angles: `YES`

Sector #1 lower azimuth: `0 deg`

Sector #1 upper azimuth: `30 deg`

Sector #1 lower elevation: `1 deg`

Sector #1 upper elevation: `3 deg`

Include sector #2 in overall test: `NO`

Include sector #3 in overall test: `NO`

Include sector #4 in overall test: `NO`

Include sector #5 in overall test: `NO`

Include sector #6 in overall test: `NO`

Include sector #7 in overall test: `NO`

Include sector #8 in overall test: `NO`

RCP8 can generate a trigger blanking output when the antenna falls within one the user-defined solid sectors in azimuth and elevation. Choose the remapped output line that should hold the blanking signal from: `TrPwr SvPwr RdOff Reset IRS0 IRS1 IRS2 PW0 PW1 Rly AZ0`. Choose an optional remapped input line to OR into the result from: `TrPwr MagCr ILock Air WGPrs IRS0 IRS1 IRS2 PW0 PW1`. For each sector that is enabled, choose whether Earth or Pedestal angles are to be used in the test, and the AZ and EL lower and upper limits.

The sector blanking latency is 3.5 ms. This latency is defined as the maximum time that can elapse between the antenna moving into or out of a blanked sector, and RCP8s mapped hardware output line that responds with that indication. The 3.5 ms latency is only realized when the mapped output line is AZ0 (LSB of the parallel azimuth output). All other output lines run with a 29 ms delay; as do any optional re-mapped input line that is fed into the blanking criteria. For example, at a 36 °/sec rotation rate, the 29 ms delay might have produced a 1.04° shift in the location of the blanked sector. The 3.5 ms delay would position the edge more precisely by introducing only a 0.13° shift.

Enable Shaft Encoder Simulator: `YES`

RCP8 can simulate the shaft encoder signals at 500 Hz. This only works at relatively slow antenna speeds. It produces outputs using the auxiliary control lines. The configuration is taken from the `ax az` and `ax el` setups.

The following table shows the output signals, including recommended cabling:

Table 13 Shaft Encoder Output Signals

Signal	Control	Backpanel J9	Backpanel J3
`EL Index`	`C78`	`14`	`1`

Signal	Control	Backpanel J9	Backpanel J3
`EL A`	`C76`	`15`	`2`
`EL B`	`C77`	`16`	`3`
`EL Prox`	`C79`	`17`	`8`
`AZ Index`	`C74`	`1`	`4`
`AZ A`	`C72`	`2`	`5`
`AZ B`	`C73`	`3`	`6`
`AZ Prox`	`C75`	`4`	`7`
`EL Limit Lo`	`C71`	-	-
`El Limit Hi`	`C70`	-	-

Inputting the hardware signals requires the following lines in the *`softplane.conf`* file:

```
splConfig.Io62[0].Opt.Cp.J3_01_14.lRS422 = 0
splConfig.Io62[0].Opt.Cp.J3_01_14.iTerm = 0
splConfig.Io62[0].Opt.Cp.J3_01_14.pinPos =
splConfig.Io62[0].Opt.Cp.J3_01_14.pinNeg = sAux[100]
splConfig.Io62[0].Opt.Cp.J3_02_15.lRS422 = 0
splConfig.Io62[0].Opt.Cp.J3_02_15.iTerm = 0
splConfig.Io62[0].Opt.Cp.J3_02_15.pinPos =
splConfig.Io62[0].Opt.Cp.J3_02_15.pinNeg = sAux[101]
splConfig.Io62[0].Opt.Cp.J3_03_16.lRS422 = 0
splConfig.Io62[0].Opt.Cp.J3_03_16.iTerm = 0
splConfig.Io62[0].Opt.Cp.J3_03_16.pinPos =
splConfig.Io62[0].Opt.Cp.J3_03_16.pinNeg = sAux[102]
splConfig.Io62[0].Opt.Cp.J3_04_17.lRS422 = 0
splConfig.Io62[0].Opt.Cp.J3_04_17.iTerm = 0
splConfig.Io62[0].Opt.Cp.J3_04_17.pinPos =
splConfig.Io62[0].Opt.Cp.J3_04_17.pinNeg = sAux[103]
splConfig.Io62[0].Opt.Cp.J3_05_18.lRS422 = 0
splConfig.Io62[0].Opt.Cp.J3_05_18.iTerm = 0
splConfig.Io62[0].Opt.Cp.J3_05_18.pinPos =
splConfig.Io62[0].Opt.Cp.J3_05_18.pinNeg = sAux[104]
splConfig.Io62[0].Opt.Cp.J3_06_19.lRS422 = 0
splConfig.Io62[0].Opt.Cp.J3_06_19.iTerm = 0
splConfig.Io62[0].Opt.Cp.J3_06_19.pinPos =
splConfig.Io62[0].Opt.Cp.J3_06_19.pinNeg = sAux[105]
splConfig.Io62[0].Opt.Cp.J3_07_20.lRS422 = 0
splConfig.Io62[0].Opt.Cp.J3_07_20.iTerm = 0
splConfig.Io62[0].Opt.Cp.J3_07_20.pinPos = sProxSwAZ
splConfig.Io62[0].Opt.Cp.J3_07_20.pinNeg =
splConfig.Io62[0].Opt.Cp.J3_08_21.lRS422 = 0
splConfig.Io62[0].Opt.Cp.J3_08_21.iTerm = 0
splConfig.Io62[0].Opt.Cp.J3_08_21.pinPos = sProxSwEL
splConfig.Io62[0].Opt.Cp.J3_08_21.pinNeg =
```

Outputting the hardware signals requires the following lines in the *softplane.conf* file:

```
splConfig.Io62[0].Opt.Cp.J9_01_14.lRS422 = 0
splConfig.Io62[0].Opt.Cp.J9_01_14.iTerm = 0
splConfig.Io62[0].Opt.Cp.J9_01_14.pinPos = cAux[74]
splConfig.Io62[0].Opt.Cp.J9_01_14.pinNeg = cAux[78]
splConfig.Io62[0].Opt.Cp.J9_02_15.lRS422 = 0
splConfig.Io62[0].Opt.Cp.J9_02_15.iTerm = 0
splConfig.Io62[0].Opt.Cp.J9_02_15.pinPos = cAux[72]
splConfig.Io62[0].Opt.Cp.J9_02_15.pinNeg = cAux[76]
splConfig.Io62[0].Opt.Cp.J9_03_16.lRS422 = 0
splConfig.Io62[0].Opt.Cp.J9_03_16.iTerm = 0
splConfig.Io62[0].Opt.Cp.J9_03_16.pinPos = cAux[73]
splConfig.Io62[0].Opt.Cp.J9_03_16.pinNeg = cAux[77]
splConfig.Io62[0].Opt.Cp.J9_04_17.lRS422 = 0
splConfig.Io62[0].Opt.Cp.J9_04_17.iTerm = 0
splConfig.Io62[0].Opt.Cp.J9_04_17.pinPos = cAux[75]
splConfig.Io62[0].Opt.Cp.J9_04_17.pinNeg = cAux[79]
```

To simulate the limit switches, use the following logic equations:

```
EQ00: # Set the lower limit switch
\--: sLowerEL = c71
EQ01: # Set the upper limit switch
\--: sUpperEL = c70
```

Automatically calibrate Shaft Encoder: `YES`

Use this only on systems with shaft encoders.

This causes RCP8 to initiate an automatic calibration of the shaft encoders each time they become uncalibrated. If a calibration attempt fails, a failed flag is set and the calibration attempt does not repeat.

Resetting from shutdown clears the last failed state. Setting the `lShaftForceCal` logic control variable forces a new calibration by clearing the calibrate bit and failed bit for each axis.

While running, the auto calibration blocks normal control of the antenna, similar to the TTY monitor mode. The front panel SS1 and SS2 display show `LockCal` in this case.

The algorithm scans at 2 rpm in azimuth until calibrated. In elevation, it scans at 1 deg/second down until the lower limit switch (it should disable shutdown while calibrating), then it goes up at 2 °/second until it is calibrated.

The elevation limit switch shutdown is disabled during auto calibration. The elevation shutdown limits are disabled unties the elevation is calibrated. The calibration fails if the antenna travels more than 1.5 times the expected distance, or more than 2 minutes elapses before calibration.

The source code is in *`/usr/sigmet/rda/rcp8/open/ShaftAutoCal.c`*.

8.4.3. Miscellaneous Site Setups

> The set-up questions listed in your configuration vary depending upon your set-up and how your have responded to previous questions.

To access these questions, in the `RCP>` prompt, type: **`Site Miscellaneous`**

External reset 'unsafe' duration: `3.0 sec`

Lower EL limit switch causes shutdown: `YES`

Upper EL limit switch causes shutdown: `YES`

If an elevation limit-switch closure is detected, RCP8 drive circuitry automatically inhibits further motor current in that direction. In this case, RCP8 can shut down at the users request to prevent further antenna motion until the cause of the switch contact can be determined.

Primary I/O-62 PCI card (-1:None) : `0`

Secondary I/O-62 PCI card (-1:None) : `-1`

Run I/O-62 external line powerup tests: `NO`

Reset all PCI cards on RCP8 shutdown: `YES`

Use network based panel for I/O: `NO`

Provide IRIS RPC network status server: `NO`

Pedestal has an auxiliary second antenna: `NO`

Echo error signals to the chat interface: `YES`

8.5. AXIS Command

You must complete the **`AXIS`** questions for both the azimuth (**`Axis AZ`**) and elevation (**`Axis EL`**).

> The set-up questions listed in your configuration vary depending upon your set-up and how your have responded to previous questions.

This command defines parameters related to the azimuth or elevation axes that are not related to the velocity or position servos.

Use internal antenna simulator: `NO`

This variable determines if the internal antenna simulator is normally `OFF` or normally `ON` for this axis. The simulator is controlled in this area, rather than from the host computer, to allow identical host computer code to be used regardless if the simulator is operating or not.

> The simulator may be used independently on each axis. This is useful when testing only one of the real antenna axes.

Angle input signal source: Parallel, Synchro, A/B/Index, Canbus, or Custom

There are several choices for how to read the angles into RCP8 (see below). You must set both axes to the same value.

Angle input signal source: `Parallel`

Angle input format is BCD: `NO`

Number of bits for angle input: `16`

Maximum angle update period: `0.0 ms`

The parallel antenna position inputs are TTL levels. The number of bits used to represent an angle vary from site to site, depending on the style of encoder and associated circuitry used by the antenna. RCP8 supports up to 16-bits of binary angle and 4-digit BCD angles. For the binary angles, if all 16 lines are not used the signals should be applied starting from the most significant line. Unused lines are then masked internally, and external connections are not necessary. For the BCD angles, good to 1/10º, the lowest 14 bits are used.

The maximum period configuration is used to handle cases in which the parallel angles are updated at a slower rate than the 600 Hz RCP8 polling rate. If you have a fast source, then type in 0. If the angles change slower than this, RCP8 thinks that the antenna has stopped, then suddenly moved. In that case, type in the longest expected period between updates. RCP8 only inserts angles when they change, or after this period has elapsed.

Angle input signal source: `Synchro`

Synchro reference frequency: `60 Hz`

Shutdown for invalid synchro voltages: `YES`

Calibration Gain #1: `1.00000`

Calibration Gain #2: `1.00000`

RCP8 implements 3-wire synchros as an optional method for measuring both position and velocity. The synchro voltages for both AZ and EL are applied to the 12-pin Molex connector on the IO62/CP backpanel. This connector uses the same wiring that was used for the synchro inputs to the old RCP02; so, for upgrades, you can simply move your existing cable from one to the other.

The Synchro-to-Digital (S/D) conversion is implemented entirely in FPGA code on the I/O-62 card, and in software running in RCP8. No additional hardware is required to begin using synchro inputs on your system. New setup questions in the **`AXIS`** command might be set as follows:

The first 2 questions establish that a 60 Hz synchro is used for angle (position) input on this axis. The voltages on the 2 Ref and 3 S1/S2/S3 lines are checked for validity, and RCP8 shuts down if these voltages drop below 10 % or rise above 95 % of full-scale A/D values. Note that the present voltage levels can be checked in the **help view** menu. The angle offset from true orientation is set to 0 in this case, but you can use it to null out any fixed position error.

When synchros are used for position input you can still use tachometers for velocity input in the usual way. However, if tachometers are unavailable, an alternative is to use the velocities that are generated by the S/D conversion process itself. The S/D converter is implemented as a Type-II tracking servo that provides 0 position error at any velocity when the acceleration is 0. The internal velocity that is maintained during this process can be used by RCP8 in place of a physical tachometer. When doing this, choose the velocity that corresponds to 100 T-Units of virtual tachometer level. Choose an upper bound that is equal to the fastest spin rate you ever intend to use.

The following table lists the maximum RMS voltage that can be applied to the backpanels Molex SYNCHRO connector for each value of plug-in SIP resistor. The AZ channel voltages are set by `SIP S1` and `S2` sets the EL voltage levels. These resistors are socketed, and can be changed by removing the back cover of the IO62-CP panel

Table 14 RMS Voltage Values

S1 or S2	Max Ref(RMS)	Max S-S(RMS)
47K	56V	31V
68K	81V	45V
100K	118V	66V
150K	178V	99V
220K	261V	145V

Note that the `Ref` inputs have somewhat lower gain than the 3 `S` inputs. This is because the precision of the S/D angle conversion is affected primarily by the precision at which the 3 `S` voltages can be measured. The backpanel therefore biases the gains so that the `S` voltages can be made as large as possible, that is, without the Ref voltages first filling the A/D conversion range.

The appropriate resistor is the smallest value such that the maximum `S-to-S` voltage of the synchro (which is angle dependent) still fits within the table range. The reference voltage should then fit easily into its corresponding maximum range, but do not worry if it does not fit, the important thing is to match the `S` line voltages.

For example, a traditional 90 Vrms 1:1 synchro would best use the 150 K resistor, whereas a 105 Vrms unit would require the 220 K value. Note that you can check for proper A/D conversion levels of the synchro inputs using the RCP**help view** menu.

Due to 1 % resistors being used in RCP8 Synchro inputs, it is possible to have small position errors, up to 1 % over a 120° span. The synchro gain corrections handle this. Because of the redundancy in the 3-wire synchro signals it is possible to examine a collection of (`S12,S23,S31`) measurements and deduce whether gain errors exist among the 3 terms. The 2 setup questions to set calibration gains for synchro inputs on RCP8. They default to 1.00000 (no correction).

The monitor command contains an ALT display format in which synchro information is shown in detail for each axis that uses those inputs. The fields are:

`SyMag?`
: Magnitude of the synchro input, 0-to-1 range

`SyUse?`
: Fraction of synchro usage history table in use, 0-to-1 range

`Gains`
: The 2 estimated (suggested) gain terms

```
RCP> mo
AZ-Pos SyMag SyUse Synchro Gains EL-Pos SyMag SyUse Synchro Gains Time
------ ----- ----- --------------- ------ ----- ----- --------------- ----
120.90 0.93 0.00 -------,------- 0.00 0.00 0.00 -------,------- 9.77 res
120.89 0.93 0.00 -------,------- 0.00 0.00 0.00 -------,------- 1.39
62.38 0.94 0.17 1.00194,1.00111  0.00 0.00 0.00 -------,------- 7.00
```

The idea is to estimate the gain terms from synchro information that has been collected over the widest possible span of angles on each axis. In the above example, the **`reset`** command is first used to clear the history tables, then the antenna was moved slowly over a 60° interval. The `SyUse?` of 17 % corresponds to the 60/360 span of collected samples. A pair of gain terms are suggested when `SyUse?` exceeds 5 %. Take these gain numbers, type them in calibration gain setups, and save.

Angle input signal source: `A/B/Index`

Number of A/B ticks per Index pulse: `2048`

Number of Index pulses per revolution: `6`

Proximity sensor approximate angle: `110.0 deg`

Reverse direction of A/B quadrature lines: `NO`

Sample lines from secondary I/O-62: `NO`

RCP8 can accept angle input from A/B/Index (quadrature) shaft encoders using a variety of styles of gearing and indexing. The first 2 questions choose the number of quadrature transitions (ticks) per index pulse and the number of index pulses per revolution. In this example, the encoder unit produces 2048 ticks between each index pulse, and the gearing is such that the encoder spins around 6 times for each full revolution of the antenna. The index pulses reset the measured angle to 0, or to the closest multiple according to the gear ratio. If the index pulse(s) correspond to nonzero angle(s), then use the standard **Angle offset from true orientation** question to set the offset of the one closest to 0.

An auxiliary proximity sensor must be used to resolve the ambiguity of the index pulses when the number of index pulses per revolution is greater than one. The sensor can be positioned anywhere along the axis and we only need to know its approximate angle. In the above example with 60° sectors per index pulse, contacting the proximity sensor at 110° adds a multiple of 60° to the present angle such that the result lies between 80° and 140°. To define your sensors, assign **`sProxSwAZ`** and/or **`sProxSwEL`** status inputs in *`softplane.conf`*.

The following table shows the fixed I/O-62 pin assignments for the angle encoder inputs.

Table 15 Fixed I/O-62 Pin Assignments for Angle Encoder Inputs

Signal	I/O-62 Pin(s)	Backpanel J3
EL Index	5,26	1,14
EL A	6,27	2,15
EL B	7,28	3,16
AZ Index	8,29	4,17
AZ A	9,30	5,18
AZ B	10,31	6,19

You can choose either TTL or RS-422 electrical levels by assigning these pins as status inputs in *`softplane.conf`*. Assign them to some unused **`sAux`**`[]` lines. You can also monitor the inputs in logic equations for debugging.

Note that for test purposes, a simple pair of quadrature signals toggling at 2 Hz can be created using RCP8 itself:

```
EQ00: t1_single_1 = t0_clock_1 & !t2_single_1
EQ01: t2_single_1 = (!t0_clock_1) & t1_single_1
```

Angle input signal source: `Canbus`

RCP8 gets angles through the canbus.

Angle input signal source: `Custom`

RCP8 gets angles through one of the site custom interfaces. Many of these are serial in nature.

Multiplicative angle scale factor: `1.0000`

Input angles are multiplied by this factor before being inserted into the system.

Input offset from true orientation: `0.00 deg`

Use this offset value if the pedestal angle positions reported to RCP8 are biased. You may correct errors as large as ±180º.

> When synchro inputs are used, you may toggle the input offset and input sense to correct the errors in offset and direction of rotation that result if the synchro lines have accidently been swapped.

Angle offset from true orientation: `0.00 deg`

Use tachometer voltage to estimate velocity:

Use tachometer voltage to estimate velocity: `YES`

Tachometer calibration - Level: `50.00 T-Units`

Tachometer calibration - Speed: `12.00 deg/sec`

Answering **`YES`** to the first question informs RCP8 that a real tachometer is available on this axis, and that its voltage should be used in all instances where velocity feedback is required.

RCP8s internal velocity representation is, in many cases, in terms of the A/D converter units from the antenna tachometers (T-units). These units are arbitrary and need not correspond to any particular absolute angular velocity. However, it is often necessary to convert the tachometer units into absolute units to verify that the tach and position information are mutually consistent. This conversion is necessary for moving platform stabilization with an INU and for the benefit of host computer communication.

You can use the local TTY angle monitor to calibrate absolute velocity. Use the **`alt`** subcommand to choose the alternate display that displays the tachometer calibration values. Start the antenna moving on one axis using a velocity servo or a simple motor drive. For accuracy, ensure that the tachometer level is at least 50-half of the A/D converter range filled. The displayed calibration ratio-with-saved should remain approximately constant and be very close to 1.000. Enter a new pair of calibration values if the ratio is not close to 1.000.

Use tachometer voltage to estimate velocity: `NO`

Answering **`NO`** to the **Use tachometer voltage to estimate velocity:** question informs RCP8 that a real tachometer is not available on this axis, and that a Virtual Tachometer (based on position inputs) should be implemented in its place.

The Virtual Tachometer runs every 10 ms and operates in 2 steps:

- A quadratic fit is computed for the most recent W seconds of position data (time window is adjustable up to 1.2 seconds). The result is a nice instantaneous velocity estimate that is W/2 seconds old (from the center of the window). The niceness comes from having a large number (100 W) of position points to work with, and from the insensitivity to acceleration afforded by the quadratic fit. However, the W/2 second delay makes this estimate not directly usable in feedback loops.
- The delayed tachometer estimate is extrapolated forward using the known history of drive voltages that have been sent to the motor during the past W/2 seconds. RCP8 uses a first order differential equation model to predict the behavior of each axis. (The same model that is used for background consistency checks). The equation with initial conditions is numerically integrated over the W/2 second interval to produce the tachometer estimate for the present moment.

Virtual Tach - Full scale speed: `24.00 deg/sec`

Sets the actual velocity in degrees/second that 100 T-Units represent. Set this number set 20 % greater than the fastest anticipated rate of rotation. Do not make it unnecessarily large, as this introduces quantization errors in the Virtual Tach units.

Virtual Tach - Differentiation window: `0.50 sec`

Defines the width of the position history window. The 0.5 second default is appropriate in almost all cases. Making it larger produces smoother drive voltages at low scan speeds, but at the expense of greater errors in extrapolating the phase delay.

Virtual Tach - Minimum travel: `0.05 deg/window`

Allows you to minimize the effects of noise in the least significant bits of the antenna positions. This question sets a minimum travel that must be observed within the history window in order to produce a nonzero Virtual Tachometer estimate. The minimum travel is similar to a constraint on the standard deviation of the positions within the window. When quantization errors are dominant, the minimum travel should be set to approximately one half the weight of an LSB. The proper setting ensures that a tachometer value of 0 is produced when the antenna is genuinely at rest.

Virtual Tach - Use antenna model predictor: `YES`

Permits the extrapolation model to be switched `OFF` for testing purposes, but it should always remain `ON` during normal operation.

Canbus Tach -- Full scale speed: `43.60 deg/sec`

Note the folllowing about the the Virtual Tachometer setup questions.

Enforce soft limits of position travel: `YES`

Minimum soft limit of travel: `-1.00 deg`

Maximum soft limit of travel: `91.00 deg`

Most weather radar antennas can operate over a limited range of elevation angles - typically from slightly below the horizon to slightly beyond the zenith. Since mechanical stops are encountered, it is important not to run the antenna to its limits at any appreciable speed. To enforce this, RCP8 can be programmed with 2 soft angle limits, beyond which the antenna should not travel. These internal bounds are typically set slightly short of the actual mechanical stops and of any limit switches that might be activated. Never contact the stops during normal operation. Enter the 2 limits (lower/upper) in degrees (º).

> The angle span defined by these limits may be any clockwise sector as large as 359º.

Enforce shutdown limits of position travel: `YES`

Minimum shutdown limit of travel: `-3.00 deg`

Maximum shutdown limit of travel: `93.00 deg`

The travel limits represent the hard bounds between where the antenna must lie. If the angle is observed to be outside of these limits, RCP8 shuts down immediately. The shutdown limits are intended to catch preposterous angles that might result from broken cables or faulty position encoders. The limits should be set to the furthest downward (lower) and upward (upper) positions that are realizable, preferably just before any limit switches are contacted.

> The angle span, defined by these limits, may be any clockwise sector as large as 359º.

Force shutdown if tach/pos are inconsistent: `YES`

Permissible fixed error: `1.50 deg/sec`

Permissible relative error: `10.00 %`

RCP8 continually checks to ensure that the velocity measured by the antenna tachometers matches those obtained by differentiating the antenna position. If these quantities are dissimilar, then a failure may have occurred that could lead to the damage of the mechanical system. For example, if a velocity servo is running, and if the tachometer input signal were removed, then the processor would assume that the antenna was not up to speed and would continue to output a large drive. If the antenna were indeed spinning, this large drive could lead to difficulty.

Force shutdown for unresponsive antenna: `YES`

Permissible tach prediction error: `15.0 T-Units`

Maximum duration of such error: `3.0 sec`

One of the more damaging types of antenna failures can occur when the motor, the gearbox, or the antenna itself becomes jammed. In such cases, it is important that the servo system remove the motor drive immediately to minimize consequential damages. To accomplish this type of safety action, RCP8 makes a decision to shutdown based on a comparison of the actual antenna acceleration with the expected acceleration.

Moment of Inertia: `6.00 (D-Units / T-Units/sec)`

This parameter defines the moment of inertia in a linear dynamic model of the antenna motion. The moment of inertia can only be measured when the antenna is accelerating. Representative values can be read from the Angle Monitors **alt** display while applying **ad** or **ed** commands. This parameter is used as part of the background calculation that checks for an unresponsive or jammed antenna.

> The motor sustaining drives and the nominal drive slopes must be properly set, before the moment of inertia, so each axis can be measured.

When virtual tachometers are in use, the associated antenna model predictor must be disabled while the moment of inertia is being measured on each axis.

Enforce model-based acceleration limits: `YES`

Maximum acceleration: `6.0 deg/sec/sec`

Extension of bound toward zero drive: `50%`

RCP8 servos can operate under the constraint of bounding the maximum acceleration (on each axis) that the antenna experiences. This acceleration limiter is based on RCP8s existing first-order linear differential equation antenna model. When the limiter is enabled, output drive levels are clamped within the range of voltages that would keep the antenna acceleration within the configured bounds. This results in much gentler and smoother large-scale motion of the antenna, without compromising small-scale performance in any way.

Since the acceleration limiter is based on RCP8s internal antenna model, all steps up to and including the proper measurement of the moment of inertia must be complete before enabling this feature.

Acceleration limiting works by keeping the motor drive bounded within an interval that is centered on the voltage that would maintain the present velocity. If the simulated model of the antenna is tuned properly, this algorithm limits both the maximum acceleration (increasing drive) and maximum deceleration (decreasing drive) of the antenna.

The third question (**Extension of bound...**) gives finer control of the extent to which the model-based acceleration limiter is can extend the allowable drive interval down to include 0 V. To make sure that the antenna could always be stopped, even if the numerical model were badly mistuned, the original implementation of the acceleration limiter always extended the valid drive interval to include 0 V. This meant that 0 drive could always be applied to bring the antenna to a stop, but as a result, the maximum deceleration limit would sometimes be exceeded. In some cases this leads to gear strain as the antenna coasted to a stop from high speed under 0 drive.

The third question allows the deceleration region to be controlled. If the antenna can safely coast to a stop from any velocity, then the safest setting is the old default value of 100 %, that is, the allowable drive interval is extended all the way to 0. A value of 0 % enforces the deceleration limit just as strongly as the acceleration limit, but should only be used if the model is properly tuned and if the antenna could be strained by coasting to a stop. The default value of 50 % is usually a reasonable compromise.

Use drive compensation for unbalanced antenna: `YES`

Neutral droop position of antenna: `35.0 deg`

Drive required 90-deg off neutral: `51.4 D-Units`

RCP8 can apply drive compensation to an antenna that is not mechanically balanced. The result is that even a badly unbalanced axis can be properly stabilized without requiring any readjustment of the antenna counterweights. The new setup questions in the axis menu are:

The model used here is that the center of mass of the unbalanced antenna is offset some distance from the axis of rotation. Thus, when no other forces are applied, the antenna tends to droop to a neutral angle that puts that center of mass directly below that axis. In the previous example, the neutral droop angle is 35°, that is, no motor drive is required to hold the antenna at that position.

Once we know the neutral droop angle `N`, the drive that is required to compensate for the imbalance when positioned at some angle `P` is `D sin(P-N)`, where `D` is the drive required to hold the axis 90° away from the neutral point. The second setup question asks for that value `D`.

As an example, suppose we have the unbalanced antenna mentioned above that returns to 35° when no drive is applied. We wish to compensate for this, and have manually determined that a drive level of -28 D-Units is required to hold the antenna down when it is positioned at 2°. The drive that would be required at 90° offset is therefore `-28 / sin(2-35)` = 51.4 D-Units, which we then enter into the second setup question.

Maximum output drive voltage: `+/- 10.00 Volts`

Drive voltage is positive for positive motion: `NO`

Drive voltage is positive for positive motion: `YES`

Set this Boolean variable to either **`YES`** or **`NO`** if the numerically positive drive levels result in upward or downward velocities. The correct setting can easily be determined by setting up a small positive drive, using the local TTY angle monitor, and noting whether the positions are increasing or decreasing with time.

Tach voltage is positive for positive motion: `YES`

Set the Boolean variable to either **`YES`** or **`NO`** if the positive tachometer A/D values, while the antenna is moving and while using the local TTY angle monitor, correspond to upward (CW) or downward (CCW) velocities. The displayed tachometer values should be positive when the position increases with time. If the sign is incorrect, then toggle this variable.

Drive is normal(0), or always Neg(-1)/Pos(1): `0`

Drive output offset: `0.00 D-Units`

Use this offset value if the servo power amplifiers do not produce 0 drive when 0 V is applied. Do not use this value to attempt to compensate for asymmetric motor-drive requirements in the 2 directions. Instead, use the separate positive and negative sustaining motor drives and the separate nominal drive slopes.

Tachometer input offset: `0.00 T-Units`

Use this value if there are DC offsets in the tachometer signals. RCP8 automatically DC-balances its differential tachometer inputs, so residual offsets may be the result of contact potentials in the wiring of the tachometer. If necessary, adjust this input offset so that a stationary axis produces a tachometer reading of 0.

8.6. VSERVO Command

You must complete the **`VSERVO`** questions for both the azimuth (**`Vservo AZ`**) and elevation (**`Pservero EL`**).

> The set-up questions listed in your configuration vary depending upon your set-up and how your have responded to previous questions.

Once the velocity servo parameters are set to an initial-guess value, you must to exercise the servo on each axis to check for proper behavior. When properly setup, the servo should rapidly, and without overshoot, bring the antenna velocity to any requested rate as entered using the local TTY **`at`** and **`et`** commands. The tachometer readings should be reasonably stable, plus or minus 0.1 T-Units, and the output drive should exhibit minimal oscillation around the mean level necessary to obtain the requested velocity.

Use the following suggestions to help resolve problems in the servo setup.

Table 16 Servo Possible Problems and Solutions

Problem	Solution
Servo is completely unstable	Recheck the tach sign to make sure the positive velocities correspond to the positive position increments.
Tach sign is correct	Check the drive sign: • If the velocity overshoots, the tachometer and drive filtering time constants may be too long. • If the antenna is sluggish and does not quickly reach the desired velocity, the feedback gain is too low.
Antenna chatters Output drive oscillates around its mean value	Feedback gain is probably too high.
equilibrium velocities differ slightly from the requested rates For exampke, request 50 but get 51.	Nominal drive slope and/or sustaining drives are incorrect.

Motor positive sustaining drive: `15.00 D-Units`

Motor negative sustaining drive: `-15.00 D-Units`

These numbers indicate the drives that are required to just overcome the friction of the motor during positive (CW or upward) and negative (CCW or downward) motion. These are given in D-units, ranging from -100 to +100. To determine the proper values, use the local TTY control ad and ed commands. Start from initial rest and gradually increase the drive until the motor suddenly starts to move. Then decrease the drive until the motion stops due to friction. Enter the smallest drive values for which continuous motion could be sustained.

Nominal positive drive slope: `0.800 D/T-Units`

Nominal negative drive slope: `0.800 D/T-Units`

These parameters are used along with the sustaining drive levels to make an initial guess of the drive required to maintain a given velocity in the steady state.

Use the local TTY control to determine the following values:

1. Output a drive level that results in a velocity that is approximately 75 % of full speed.
2. When a steady tachometer reading has been achieved, record the `drive` and `tach` readings from the TTY .
   The required slope is `(Drive - Sustain) / Tach`, where `Sustain` refers to the sustaining drives that were measured in the previously.

If the motor amplifier has a different gain in each direction, 2 different slopes are permitted - the first value for positive (CW or upward) motion and the second one for negative (CCW or downward) motion.

> The slopes are used only as a first-order estimate. Extreme accuracy is not necessary to operate the velocity servos.

Velocity feedback slope: `25.000 D/dT-Units`

The tachometer error feedback slope controls the tightness of the velocity servo. The velocity servo is stable for most values of this parameter. If the value is too small, the motion is sluggish with relatively large errors in the final achieved velocity. If the value is too large, the currents thrash wildly as the servo attempts to maintain the exact requested tachometer level. The appropriate value must be determined empirically.

Connect an oscilloscope to the drive and tachometer signals and use the local TTY control to select different servo rates - the **`at`** and **`et`** commands. Choose the largest value of the parameter that brings the antenna rapidly to the requested velocities without excessive drive oscillation around its equilibrium value. If a scope is unavailable, you can make a fair judgment by observing the drive values that are displayed on the TTY. The feedback slope has units of `Drive/TachError`, typical values range from 10 ... 200.

Velocity feedback deadzone: `0.10 T-Units`

A deadzone is built into the tachometer feedback path to ensure that the uncertainty of the low bits, of the A/D converters, does not result in motor chatter. Typical values are 0.1...0.5 T-units. If values above these limits are necessary to control the chatter, then this is indicative of the excessive noise on the tachometer inputs.

Apply velocity error integral correction: `YES` **New Value:**

Characteristic time of the integral: `2.00 sec` **New Value:**

Maximum resulting drive bias: `+/-25.00 D-Units` **New Value:**

RCP8 velocity servos include a velocity error integral feedback term, in addition to the proportional error feedback term and bias terms that have always been available. The error integral effectively removes any remaining steady-state velocity bias from the servo, and guarantees that scans runs at precisely their requested speed. These questions to configure the velocity error integral feedback term. The feature is switched On/Off using the first question.

The second question establishes the characteristic time T_0 of the integral, which is defined as follows. Suppose that a fixed velocity error E was sustained for a period of time. The proportional feedback term would produce a drive D=SE, where S is the velocity feedback slope. Then, if that same error E were applied to the integrator for T_0 seconds, the same drive term D would also result.

The gain of the integrator effectively is established by T_0 ; larger times produce smaller gains. One rule of thumb (Ziegler-Nichols) for a first guess of S and T_0 is to disable the integral feedback, and increase S until reaching a value S_u , at which the antenna goes into unstable oscillation with an observed period P_u. Reasonable first settings are then obtained with $S = S_u / 2.2$, and $T0 = 2.2\ P_u$.

The integral can be clamped (the so called anti-windup feature) to prevent it from drifting into large values when the antenna is not in equilibrium. This clamp value is expressed as the maximum drive correction that can be contributed from the integral term alone. If your antenna is well characterized by its sustaining drives and nominal drive slopes, then this clamp value can be reduced (since the nominal guesses do not need to be adjusted very much). This helps reduce brief overshoots that can be caused by the integral feedback.

Generate stepper motor drive control signals: NO `NO`

Maximum absolute velocity: `95.00 T-Units`

This value represents the tachometer level which corresponds to the maximum antenna rotation rate that is considered safe. The lower-rate limit is the negative of the upper-rate limit. If the A/D converter hardware components have been set properly, the maximum value should be at least halfway through the converters full range - at least 50. If the safe value is less than 50, then the A/D range should be altered to make better use of the available 12 bits. The local TTY control, or an external manual control, can be used to cause the antenna to spin at the maximum safe rate while the tach levels are noted from the local TTY angle display.

Velocity shutdown safe margin: `4.00 T-Units`

Velocity shutdown check time: `1.00 sec`

RCP8 has provisions to shutdown if the observed velocity on either antenna axis exceeds the internal maximum velocity limits that are enforced by the velocity servo. If the velocity overshoots in the vicinity of these limits, the shutdown criterion can sometimes lead to false shutdowns when no actual problem exists. Ideally, the velocity servo is setup to ensure that overshoots does not occur however, given the influences of motor damping and wind gusts, this strict condition is difficult to enforce.

To minimize false shutdowns due to temporary velocity overshoots, the shutdown criterion is expressed in terms of a velocity tolerance and a time limit. If this condition persist longer than the specified time, shutdown occurs if the absolute value of the measured velocity exceeds the sum of the maximum limit and the specified tolerance. As an initial guess, the tolerance should be set slightly higher than the maximum sustained velocity error that is ever observed, under normal operating conditions, while taking into account wind loading and other operational effects. The time should then be set slightly longer than the time by which the longest transient overshoot exceeds the specified tolerance.

Tach zero-delay-smoother window:`0.010 sec`

Model order within the window:3

Tach filter time constant: `0.025 sec`

The tachometer inputs can be filtered with a simple, exponentially-weighted smoothing filter prior to applying to the velocity servo. This filtering is intended to remove spurious components from the digital tachometer samples. The filter time constant is typically set at approximately one-third the reciprocal of the antennas upper-frequency response limit.

The filter time constant is entered directly in seconds but the exact value must be determined by trial and error from an initial approximation. If the time constant is too large, the velocity servo becomes unstable and oscillates around the desired velocities before settling. If the time constant is too small, then no significant smoothing or spurious rejection is attained. The value should be increased until the velocity overshoots become noticeable on an oscilloscope display of the tachometer signals. The final time-constant value should be slightly less than this level. Velocity overshoots can also be detected by the human eye by requesting zero velocity and observing how the antenna comes to rest.

> The following drive filter time must be fine-tuned concurrently with the tachometer filtering. Also, both should be tuned when the Virtual Tachometer is in use.

Drive filter time constant: `0.025 sec`

This drive filter behaves much like the tachometer filter, as described in the previous paragraph, except this is applied to the output drive levels prior to D/A conversion. The purpose of the filter is to smooth the motor drive signal and to remove the high frequency feedback components that can be generated by the velocity servo. Although these components most likely would be filtered by the motor and mechanical system, the users power-drive electronics might be adversely affected by sudden changes in the motor current. The filter time constant should be set as large as possible, consistent with preventing velocity overshoots as described in the previous paragraph. The drive and tachometer filters has similar time constants however, from this common value, improved performance is usually obtained if the tachometer constant is decreased and the drive constant is increased.

Drive slew rate limit for Zero --> Max: `0.200 sec`

A slew rate limit can also be imposed on the output drive signals. The limit is expressed as the number of seconds required for the drive to slew from 0 ... 100. For example, a value of 0.2 seconds restricts the rate of change of the output drive to 500 D-Units/second. The slew rate limit is useful in preventing abrupt changes in motor drive. In some cases, such fluctuations can bring about unwanted oscillations in the antenna/pedestal mechanical system. The slew rate limiter is applied by the RCP8 software after the output filter.

The `Zero-to-Max` drive slew rate time can be set as large as 15 seconds. This allows RCP8 servos to work more gracefully with external motor controllers that incorporate a velocity feedback loop of their own. In such cases, the RCP8 velocity feedback slope should be set to 0, and internal (model based) acceleration limiting should be disabled. Acceleration limiting can be accomplished instead using RCP8s drive slew rate limiter, which can now work over a longer time span.

Note that the drive filtering and slew rate limiting are both overruled by the detection of shutdown conditions, and the enforcement of soft limits of travel. If the antenna is heading rapidly toward a soft limit, the drive is immediately adjusted to stop before the limit is reached.

8.7. PSERVO Command

PSERVO provides acess to the legacy position servo parameters. You must complete the questions for both the azimuth (**`Pservo AZ`**) and elevation (**`Pservero EL`**).

> The set-up questions listed in your configuration vary depending upon your set-up and how your have responded to previous questions.

Hysteresis inner zone: `0.051 deg`

Hysteresis outer zone: `0.020 deg`

These represent the position errors, in degrees, within which the position servo does not attempt to correct the antennas location. Two values are specified — one for the lower limit and one for the upper limit. When the actual position error is less than the lower limit, the position servo does not drive the antenna (that is, it requests 0 velocity from the velocity servo.) Likewise, when the actual position error is greater than the upper limit, the servo always drives the antenna to correct it. There is hysteresis between these limits which helps to prevent antenna chatter once it has reached the desired position (the servos state between the limits is whatever it was at the time the limits were entered.)

It is important that the inner limit is greater than half the weight of the least significant bit that encodes the antenna position. For example, for 12-bit binary encoding the inner limit must be at least 0.045 º. If it were smaller, the position servo might not realize that the final position had been reached and would continually move the antenna around a single LSB interval. As an initial guess, use an inner zone that is 10 % larger than half the LSB weight. The outer zone should then be set somewhat larger, perhaps by 50 %.

Servo type: `Legacy`

There are 2 types of position servo loops, `Legacy` and `Feedfwd`. `Legacy` has a desired speed which ramps to zero at the desired position, using multiple slopes shown here:

First position break point: `1.00 deg`

Second position break point: `5.00 deg`

These represent the values of the 2 position-error break points, in a piecewise linear definition, of the desired velocity-versus-position error.

First interval slope: `12.00 (T-units)/deg`

Second interval slope: `3.00 (T-units)/deg`

Third interval slope: `1.00 (T-units)/deg`

These represent the 3 piecewise linear definition slopes of a desired velocity-versus-position error.

The following 3 intervals are defined as:

1. Zero to first-break point,
2. first-break point to second-break point, and
3. second-break point to infinity.

Servo type: `Feedfwd`

The **`Feedfwd`** servo uses a feed forward servo to correct for the effects of a stretching rubber belt used on Vaisala pedestals, for example. Its tuning parameters are:

Period of mechanical resonance: `0.50 sec`

This is a mechanical property of the antenna–pedestal combination. Different for azimuth and elevation.

Drive constant: `0.5000 deg/sec/D-Unit`

This is a property of the motor driver and gear ratio, defining how many deg/sec drive is produced by a D-unit. This is different for azimuth and elevation in the Vaisala pedestal.

Drive end wait factor (n*T/2+T/4): `1`

At the end of the drive, wait for a period of time before reading the current position to check it. T is the period of mechanical resonance, and here you enter the n.

Maximum acceleration: `6.0 deg/sec/sec`

This is the maximum acceleration the antenna-pedestal is capable of doing.

1. The position servo cannot work properly with incorrect settings. Make sure the velocity servo has been thoroughly checked, in accordance with 8.6. VSERVO Command (page 84).
2. Set the inner and outer zones of the hysteresis as described below.
3. For **`Legacy`** servos,
   a. Set the first position break point to a small value `P`, such as 1.0 º , and attempt several values of first interval slope.
   Find the largest slope that results in no overshoot when steps of `P`-degrees or less are performed. Use the **ap** and **ep** angle monitor commands to test the slope values.
   b. Choose a larger second position break point and find the largest second interval slope that accomplishes larger steps without overshoot.
   c. Find the largest third interval slope that permits large steps of any size to be travelled without overshoot.

4. Exercise the servo on each axis to check for proper behavior.
   Try to move the antenna using the local TTY **ap** and **ep** commands and verify that:
   - Any position step can be requested without overshooting the final mark.
   - Very small steps cause antenna motion to occur. If the position feedback curve is incorrect, it is possible for the servo to work properly for some step sizes but not for others. Test a range of steps.

   This step attempts to tune the antenna for maximum performance (that is, the antenna arrives at a requested position as rapidly as possible.) On some systems, delays—usually in the response to a drive voltage—can lead to small position overshoots that can usually be eliminated by detuning the antenna performance. Detuning is accomplished by lowering the slopes, for the first and second endpoints, so the approximation to the braking curve lies below the observed curve. This usually eliminates any position overshoot, with a slight performance penalty.
5. Iterate the procedure as needed, making sure to complete the questions for both the azimuth (**`Pservo AZ`**) and elevation (**`Pservero EL`**).

More Information

‣ Position Servo Response Curve (page 25)

8.8. CONTROL Command

8.8.1. Output Line Configuration

To access these questions, in the `RCP>` prompt, type: **`Control Lines`**.

The command prompts you with a series of questions that define the polarity of the control lines that are driven by RCP8. Each control line can be either active-high or active-low.

The following choices are available for the `Radiate ON` and `Radiate OFF` control lines:

- Complementary levels that are either `ON` or `OFF`, according to the current radiate request.
- Pulse levels, in which `Radiate ON` pulses briefly to enable the radiate and `Radiate OFF` pulses briefly to disable it. The time duration of the pulses is adjustable.

8.8.2. Logic Equation Control Qualifiers

To access these questions, in the `RCP>` prompt, type: **`Control Logic`**.

The questions are a series of logic equations for qualifying RCP8 control functions. You can modify control bits according to a logical combination of control bits, status bits, and internal local variables. Using these questions, you can program RCP8 to perform custom safeguards and implement feedback specific to each site.

Each equation consists of a control variable on the left that is assigned to a logical combination of variables on the right. The syntax is that of an ordinary C language statement using the operators **&** for `AND`, **|** for `OR`, and **!** for `NOT`. You may include parentheses as needed. Type new equations and sub-commands following the **-->** prompt printed below the current equation. For example:

```
EQ00: v0 = airflow | wavegp | magcur
EQ01: cservo = cservo & !v0
EQ02: relay = relay | c10
EQ03: ctrpower = true
-->
```

In the above example, `EQ00` assigns the internal local variable `V0` to the logical OR of the "airflow", "waveguide pressure" and "magnetron current" status bits.

- `V0` is `TRUE` when any of those status lines are true.
- `EQ01` uses a `TRUE` sense of `V0` to force the servo power control line `FALSE`. Servo power is only on if it is requested to be on, and if none of the 3 control lines are asserted.
- `EQ02` allows the external local/remote relay to be forced on by the auxiliary control line `c10`.
- `EQ03` forces `T/R power` to be on all the time, regardless of any other conditions.

The rules for constructing equations are as follows:

- The right side of the equation must consist of a logical combination of any of the above control variables, plus any of the status variables such as: `spw0`, `spw1`, `sradiate`, `sservo`, `strpower`, `sreset`, `siris0`, `siris1`, `siris2`, `local`, `standby`, `ilock`, `magcur`, `airflow`, `wavegp`, `elimlo`, `elimhi`, `ngen_on`, `sgen_on`, `sgen_cw`, `sgen_flt`, `local_tr`, `shutdown`, `s[0:319]`. The numbered status variables refer to the optional auxiliary status input lines.
- The left side of the equation must consist of one of the control variables: `cpw0`, `cpw1`, `cradiate`, `cservo`, `ctrpower`, `creset`, `ciris0`, `ciris1`, `ciris2`, `ngen_on`, `sgen_on`, `sgen_cw`, `relay`, `shut1`, `shut2`, `v[0:15]`, `c[0:79]`. Control variables that have status counterparts are prefixed with the letter `c`. Thus, `cradiate` is the request to radiate, whereas `sradiate` is the detected radiate status. The numbered local variables `V[0:15]` can be used as temporary storage for sub-expressions, and the numbered control lines `C[0:63]` refer to the optional auxiliary control outputs.
- You can also write logic equations in which status variables appear on the left-hand side. The meaning given to such assignments is that the working value of the status variable is modified from its default "requested" value, that is, the value assigned from whatever hardware line or external condition is normally attached to the status bit. In this sense, the modification of a status variable is identical to the modification of a control variable. In both cases, when the variable appears on the right-hand side, it refers to its default requested value.
- The symbols `TRUE` and `FALSE` may be used on the right side of an equation to indicate always and never.

- The maximum number of variables on the right side of an equation is 4. If you must combine more than 4 four variables in an expression, you must use multiple equations using local variables to combine the bits together. For example, to force servo power off when any of the first 8 auxiliary status lines is high:

```
EQ00 : V0 = S0 | S1 | S2 | S3
EQ01 : V1 = S4 | S5 | S6 | S7
EQ02 : Cservo = Cservo & !(V0 | V1)
```

- Control variables that appear on the right side of an equation refer to the requested control state and those appearing on the left-hand-side reflect the final qualified state. For example, the equation **`cservo = cservo & !v0`** causes the requested servo state to be AND'ed with the negation of local variable `V0`, and the result is the actual servo state. This useful convention allows you to use control variables on the right without ambiguity. For example, the pair of equations: **`EQ00: cpw0 = cpw1`** and **`EQ01: cpw1 = cpw0`**, swaps the 2 pulse width control bits.
- Order of evaluation of each equation is from right-to-left. There is no evaluation precedence among the **&**, **|**, and **!** operators. Thus, **`!V0 & V1`** means **`!(V0 & V1)`**, rather than **`(!V0) & V1`**. You must use parentheses as needed to express your equations.
- Order of evaluation of the set of equations is from `EQ00` to `EQ79`. The order is important only when local variables are assigned in earlier equations so that they can be used in later equations.
- The error message that is printed when an ambiguous variable name is typed into a logic equation includes a list of all of the possible matches. This can help you identify how to type the variable uniquely.
- The logic equation editor prints a warning upon exit if multiple assignments are being made to the same variable. For normal status and control variables this almost certainly indicates an error, since only the final assignment has any lasting effect. For local variables, however, multiple sequential assignments are meaningful since an assignment on one line may be referenced on a subsequent line. For now, the warning is printed even for local variables since there is still a chance that is not what you had intended to do. To eliminate spurious local variable warnings, choose a unique set of numbered variables to use (that is, by not reusing them within the overall set of equations).

Comments and disabled equations are available:

- It is possible to add a line of comment text to each logic equation. Use the **#** command, followed by the text (which may be up to 74 characters in length). If no non–blank text is found, then the comment is removed. Whenever an equation includes a comment, the comment text is displayed in the line preceding the equation during the editing process.

- It is possible to enable and disable control logic equations while keeping the content of the equation intact. Use the **/** command within the equation editor to toggle whether the current equation is effectively "commented out". The text for disabled equations is still be shown in the editor, but is prefixed with the **#** character. This feature is useful when you want to temporarily disable some trusted and debugged equations without the risk of retyping them incorrectly later.

When using variables, note the following:

- A side effect of making assignments to local variables is that these bits are also reported to the host computer in the RCP8 Internal BITE packet. This opens many possibilities for designing customized status bits that could be monitored and reported by IRIS BITEX utility.
- The variables `shut1` and `shut2` cause a user-induced shutdown if they are set to `TRUE`. You may now write logic equations that force a shutdown of RCP8 when certain conditions are present. The 2 types of shutdowns are provided in case there is a need to distinguish among different causes. Along with this, the status variable `shutdown` is set `TRUE` if RCP8 is shutdown. For example, adding the equation **`CSERVO = CSERVO & !SHUTDOWN`** forces the servo power off when RCP8 is shutdown.
- The 4 status variables `usr0`, `usr1`, `usr2`, and `usr3` correspond to the 4 input lines at pins from 1 to 4 of header H9 on the RCP8 main board. These are general purpose TTL inputs that you may assign to any purpose you wish. For example, to include an additional status bit in RCP8's Internal BITE packet, include an equation such as **`v13 = usr0`**. Since the local variables states appear in the BITE packet, the `usr0` line shows in bit #6 of byte #12 as a result of this assignment.
- The status variable `unsafe` is `TRUE` if RCP8 is in the temporary unsafe mode following a **`reset`** command. The status variables `ovel_az` and `ovel_el` are `TRUE` when the velocity on the corresponding axis exceeds the maximum value setup from the **`velocity`** command.
- The 8 control variables `csgen0` through `csgen7` represent bits 0 through 7 of the signal generator level that is being requested by the host computer. Having access to these bits makes it possible to remap the level bits in cases where the signal generator is not controlled via the default HPIB interface.
- The 2 control variables `trig_normal` and `trig_blank` can be used to override the protected sector trigger blanking that is defined in the **Site Custom** menu. These new variables operate as follows (See also 8.8.4. Logic Equation Examples (page 99)):

  - When `trig_blank` is `FALSE` and `trig_normal` is `FALSE`, the trigger is blanked when the antenna is within one of the designated sectors. This is the normal operating mode. You can disable all 8 of the sectors if you don't want to use the trigger blanking feature at all.
  - When `trig_blank` is `FALSE` and `trig_normal` is `TRUE`, the trigger is always generated, no matter where the antenna is.
  - When `trig_blank` is `TRUE`, the trigger is always blanked, no matter where the antenna is. The assignment to `trig_normal` is ignored in this case.

When a new equation is entered, RCP8 immediately parses the input text and converts the right-hand-side into an internal representation that can be evaluated efficiently at run time. The original line of text is discarded. When the equation is re–displayed, RCP8 must recreate a printed line of text from this internal compiled representation. As a result, the equation that is echoed back does not always look the same as the equation that was originally typed. It is logically equivalent, but the exact syntax may be different. For example:

```
--> v0 = v1 & v2 & !v3
EQ00: v0 = v1 & v2 & !v3
```

Here, the output representation happens to be identical to the original input.

```
--> v0 = v1 & (v2 | v3)
EQ00: v0 = (v1 & v2) | (v1 & v3)
```

In this case, the equation is printed as an expanded version of the original. Note that the Boolean operators & and "|" both distribute symmetrically over each other, so that logical expressions can be factored and expanded over either operator.

```
--> v0 = (v1 & v2) | (v1 & !v2)
EQ00: v0 = v1
```

Here, what appears to be a function of two variables is really only dependent on one.

```
--> v0 = (!v1) & (!v2)
EQ00: v0 = !(v1 | v2)
```

In this case, the DeMorgan equivalent of the AND of 2 negations is printed as the negation of 2 OR's. This type of transformation is common, since RCP8 attempts to minimize the number of ! operators in its synthesized expressions.

The following list of sub-commands is printed at the beginning of the equation list.

```
Subcommands
Del - Delete text of current equation
Ins - Insert free slot before current equation
Pack - Pack equations to consolidate free slots
! - Negate logic sense of equation
?<v> - Additional help
```

`Del`
: Deletes the text of the current equation so that the line is blank

`Ins`
: Inserts a blank equation at the current location by shifting the current equation plus all subsequent equations ahead by one

`Pack`
: Removes blank equations and shifts all equations into the lower numbered slots

`!`
: Replaces the current equation with its logical negation. For example:

```
EQ00: v0 = v1 & v2
--> !
EQ00: v0 = !(v1 & v2)
EQ00: v0 = v1 & !v2
--> !
EQ00: v0 = (!v1) | v2
```

Note that DeMorgan's theorem was used to reprint the second of these 2 examples, because doing so removes an extra **`!`** from the equation. Perhaps the simplest equation to negate is:

```
EQ00: cservo = true
--> !
EQ00: cservo = false
```

Here, the output variable is forced `TRUE`/`FALSE` each time the **`!`** sub-command is used. This can be helpful when testing individual control lines for `ON`/`OFF` responses.

Before the equation list appears (in response to the **`Control Logic`** command), the following initial question is asked:

Enable logic override of control lines: `NO` **New Value:**

This question allows the entire set of equations to easily be switched `ON`/`OFF`, without having to change any of the equations themselves. Answering `NO` leaves the control functions unmodified (direct control from the host computer). `YES` applies all of the logical constraints.

The logic equation editor is a live menu. This means that each equation becomes active as soon as it is typed in. You can test individual control lines, and edit the set of equations until you obtain the desired effects.

The `!` sub-command is shortcut for quick `ON`/`OFF` toggles to test any control line.

8.8.3. Logic Equation Timer Variables

A collection of software timer variables are supported for use with logic equations.

Control variables are available with names having the generic form `tn_mode_time`, depending on how each timer has been configured. For example, if timer #3 is configured to be a retriggerable pulse generator with a period of 2.5 seconds, then the variable `t3_retrig_2.5` would appear in the control variable list. You could abbreviate the typed-in name to just `t3`, but the full mode and time is echoed in each equation so that the exact behavior of the timer variables is clear at a glance.

Timer variables can appear on both the left and right sides of logic equations. On the right they act as normal Boolean variables having `TRUE/FALSE` values that can be used in any logic equation. However, when they appear on the left, the value being assigned from the right-hand side acts as an input trigger to the timer. The timer's response to this input can take several forms, depending on the selected mode.

The following table shows the available modes.

Table 17 Supported Logic Equation Timer Modes

Mode	Description
Retriggerable Pulse Generator (`retrig`)	Generates a `TRUE` pulse when a `FALSE->TRUE` (rising edge) transition is applied to its input. Each rising edge continues to retrigger the output pulse, that is, a fresh pulse period is begun each time. For example, if a rising edge were presented once per second to the timer `t0_retrig_1.5`, then the timer output would be a steady `TRUE` value. Since the 1.5-second timeout begins again once per second, the output pulse never actually ends. `retrig` timers are useful for keeping track of whether any `FALSE->TRUE` transitions have occurred (perhaps irregularly) over a given period of time.
Change-Detecting Pulse Generator (`change`)	This timer is like the retriggerable timer, except that either input edge causes the period to reset. Use it if you require an output pulse in response to any change in measured conditions, for example, you could force `radiate` OFF briefly when the pulse width changes in either direction.

Mode	Description

Single Pulse Generator (`single`)	Generates a pulse similar to `retrig`, except that an active pulse is not retriggered by additional input transitions.
	For example, if a rising edge were presented once per second to the timer `t0_single_1.5`, then the timer output would be a rectangular wave that is `TRUE` for 1.5 seconds and FALSE for 0.5 seconds. The 1.5-second `TRUE` pulse is first triggered by an input edge. One second later the timer is still active, so the next input edge is ignored. The pulse finally ends 1.5-seconds later, remains `FALSE` for 0.5 seconds, and then is triggered again by the next rising input edge.
	The active-low application of `single` is also useful, as in the following two equations which prevent `radiate` from being switched back on within 60 seconds of it being switched off. Note that if a `retrig` timer were used here, then repeated attempts at radiating would keep resetting the 60-second interval even though the transmitter had never actually turned back on.
	`EQ00: t0_single_60 = !cradiate`
	`EQ01: cradiate = cradiate & !t0_single_60`

Delay Line Filter/ Follower (`filter`)	The output of this timer attempts to follow its input, but with filtering and delay effects added. When a `TRUE` input is presented, an internal counter begins counting up until the timer period is reached. At that point the timer's output is set `TRUE`. Likewise, a `FALSE` input causes the counter to decrement until reaching 0, at which point the timer's output is set FALSE. The net result is that the output follows the mean value of the input, and thus, a `filter` timer can be used to clean up a noisy logic signal, or combination of logic conditions.
Decisive-Grant, Indecisive-Wait (`fickle`)	The `fickle` timer copies its input immediately to its output, unless the output has just changed recently (within the setup period of the timer), in which case the previous output level is held. Use this timer to cleanup requests for state changes so that "original" and "thoughtful" requests get honored (passed through) right away, but once honored, a given request can not be changed for some minimum amount of time. The `fickle` timer can be used to protect against needless or damaging cycling of equipment that should not be turned On/Off rapidly. Air conditioner thermostats typically have such a timer to prevent the compressor from frequently stopping and restarting if the temperature dial is twirled up and down in an indecisive manner. After remaining in a stable state for a while, new requests are honored immediately (unlike the `fickle` timer which always introduces a delay). Once honored, that new setting once again persists for a little while.
Leading Edge Retard (`retard`)	The output of this timer attempts to follow its input, except that rising input edges are delayed by the timer period, whereas falling input edges are passed through immediately. The result is that the leading edges of the input signal are delayed, but the falling edges are not. A `retard` timer is useful when one wants to delay only the onset portion of a signal, for example, to holdoff transmitting for a few seconds after a radiate request has been made. It is also useful when filtering signals to remove short spurious `TRUE` inputs in which, contrasted with the `filter` timer, an instant-off effect is also required.

Mode	Description
Trailing Edge Extend (extend)	This timer is the counterpart to retard, in that the falling input edge is extended by the timer period and the rising input edge is passed immediately. An extend timer forces a minimum time during which the timer output is TRUE in response to any (possibly momentary) TRUE input. It is useful for stretching a short input condition out to at least some minimum time, or for adding additional "hold time" to the end of a signal. Note that the output of an extend timer is logically equivalent to the negation of the output of a retard timer whose input is also negated. Although these two timer classes are inverted-logic duals of each other, it is still conceptually useful to have both the "retard" and extend concepts. An analogy is that AND and OR are both useful logic concepts, even though an OR-gate is merely an AND-gate with inverted inputs and outputs.
Periodic Clock Oscillator (clock)	Produces a free-running clock with specified period. The length of the timer's TRUE interval (and the duty cycle) is adjustable. The clock timers usually appear on the right side of equations, where they can supply any periodic input that the logic might require, for example, to make a light blink, or to perform a periodic reset. Their phase can also be resynchronized to the start of their TRUE output interval using a rising input edge.

8.8.4. Logic Equation Examples

The following examples show how to implement custom logic requirements using the logic equations and timer variables.

Example: Allow Transmitting While Antenna Is Stopped

Suppose that an operational site requires that the radar transmitter be switched off when the antenna is not rotating. This is fine for normal operation, but during maintenance periods there must also be a procedure to allow transmitting while stopped.

When such an override is requested, an audible warning and flashing light must first occur for 20 seconds; only then does the override actually take effect. At that point the horn becomes silent, but the warning light must continue to flash for the duration of the override.

An RCP8 external status input line S0 is used to request the override. Assume that control line C0 activates the horn, and that C1 activates the warning light. The necessary equations are:

```
EQ00: v0 = cradiate & antstop & s0
EQ01: t0_retard_20 = v0
EQ02: c0 = v0 & !t0_retard_20
EQ03: c1 = v0 & t1_clock_1.5
EQ04: cradiate = (cradiate & !antstop) | t0_retard_20
```

1. EQ00 assigns local variable V0 as a qualified override request.
   V0 is TRUE when there is a request to radiate while stopped, and when the external override request line is also TRUE.

2. `EQ03` combines this condition with a 1.5-second periodic clock to produce the flashing light.
   Meanwhile, `EQ01` passes `V0` through a 20-second "retard" timer.
3. When the timer output eventually becomes TRUE, `EQ04` allows the transmitter to radiate even though the antenna is stopped.
   Meanwhile, `EQ02` sounds the horn only during the timer's initial 20-second delay period.
4. As soon as the antenna starts moving, `V0` and the timer output immediately become FALSE.
   The horn and light are extinguished right away, and the override input is ignored. The first **&** expression in `EQ04` then allows the transmitter to be controlled in the normal On/Off manner.

Example: Control Signals

Logic equations can help supply the necessary control signals for getting out of stuck situations. In this example, an antenna servo power unit requires override signals to move away from the low and high elevation physical limit switches. Assume that `C0` and `C1` enable motion in the up and down directions.

The following equations allow the **`reset`** command to activate these lines briefly:

```
EQ00: c0 = antstop & unsafe & elimlo
EQ01: c1 = antstop & unsafe & elimhi
```

The `unsafe` status variable is `TRUE` for a short interval of time following an RCP8 reset.

Resets from the host computer serial port always give a 1.0-second unsafe interval.

Resets from the RCP8 command line take the number of seconds as an argument, for example, **`reset 2.5`**. The `antstop` test is added as an additional safeguard to insure that the antenna is motionless when the override is attempted.

Example: BITE Information

Logic equations can be used to supply the host computer with BITE information that would not ordinarily be available. To do so, make an assignment to one of the first 14 local variables, as those are then be transmitted in RCP8's Internal BITE Packet. For example, adding the equation:

```
EQ00: v13 = ovel_az | ovel_el
```

sends send a "velocity overspeed" bit to the host computer in Bit #6 of Byte #12 (See A.5.5. Internal BITE Packet (page 134) for the mapping of the local variable bits).

Example: Variable Assignments

When writing sets of logic equations for RCP8, note that assignments to most types of variables cannot be referenced as such on subsequent lines. When control and status variables appear on the right side of an equation, they always refer to their original requested value. Assignments made on the left modify the variable's effective working value and the original requested value still remains unchanged. This is why it is never correct to make more than one assignment to the same control or status variable, and why the pair of equations:

```
EQ00: cpw0 = cpw1
EQ01: cpw1 = cpw0
```

would swap the two pulse width control lines without using the temporary intermediate variable that would normally be required for sequential assignments. The only variables that can be referenced immediately after being assigned are the local variables `V[0:15]`. Thus, the pair of equations:

```
EQ00: v0 = v1
EQ01: v1 = v0
```

would not swap the 2 local variables, but instead, would leave both set to the original value of `V1` (probably not useful).

Example: Trigger Blanking Variables

As an example of how the trigger blanking variables might be used, consider a hypothetical farmhouse that is close enough to the radar that if the antenna is pointing at it, and the antenna is stationary, we would exceed the allowable microwave radiation limit. However, we are also allowed to average the power exposure over longer periods, so that if the antenna is moving we can radiate at the farmhouse as we sweep past it. We don't want to inhibit the trigger when the antenna stops; only when it stops within one of the protected sectors.

In sum, we want to stop transmitting while the antenna is stationary and is within one of the designated sectors, but we also want the radar to transmit when the antenna is moving. This is accomplished using the equation:

```
EQ00: TRIG_NORMAL = !ANTSTOP
```

For a related application in which we want to stop transmitting when the antenna becomes stationary, use:

```
EQ00: TRIG_BLANK = ANTSTOP
```

Note that the built-in timers could also be used to permit brief antenna stoppings without producing the trigger side effects immediately.

8.8.5. Logic Equation Configuration of Variables

Use these questions to configure the internal variables used within logic equations.

To access these questions, in the `RCP>` prompt, type: **`Control Variables`**

Choose: Retrig Single Filter Retard Extend Clock

Timer #0 trigger mode: `Retrig`

Timer #0 period/delay: `1.0`

There is a question in this form for each of the available timers. Choose the mode of each timer (See 8.8.3. Logic Equation Timer Variables (page 97)), and its associated period or delay (in seconds).

Minimum velocity for 'antstop': `0.50 deg/sec`

Minimum time for 'antstop': `2.00 sec`

The status variable `antstop` is set to:

- `TRUE` when the antenna seems to be stopped, that is, has been moving slower than a prescribed speed for more than a prescribed time.
  These setup questions configure the speed and time thresholds.
  Both the AZ and EL axes must appear to be stopped in order for `antstop` to be `TRUE`.
- `FALSE` when the antenna seems to be moving, that is, the speed on either axis has exceeded the threshold speed for more than the specified time.

8.8.6. Analog Voltage Input Control Logic Variables

You may configure Boolean variables whose values are based on comparison tests of the 8 analog voltage input lines. In this way, the analog inputs can be thresholded and used as additional inputs to logic equations within RCP8. Up to 16 such variables may be defined, that is, you may have, on the average, 2 threshold tests for each input line.

To setup the analog input variables, use the **`Control ADC`** command to define the following information for each voltage comparison test that you need:

```
Analog Input Test Variable Definitions

 --------------------------------------

A/D Logic Variable #0 is defined: YES

Description of A00 variable: 'HiTemp '

Input summation term #1: A0

Input summation term #2: -A4

Input summation term #3: Zero

Input summation term #4: Zero

Test for ( A0-A4 > 3.55 Volts )
```

This example defines a new Boolean status variable named `a00_HiTemp`. This variable name appears in the `?v` list of available status variables in the equation editor, and the variable may be used on the right side of any logic equation. The descriptive suffix makes the variable meaningful and readable within the text of the logic equations. You may choose any 8-character name that does not contain spaces or punctuation other than '.', '–', and '–'. The descriptive suffix can be omitted (not recommended) by entering a space at the prompt, but your logic equations become less readable.

The comparison test operates by summing the voltages on one or more input channels, and then testing whether that sum is greater than a specified voltage. If the test passes, the variable is `TRUE`, otherwise it is `FALSE`. The input channels can be either added or subtracted when computing the sum.

In the above example, `a0_HiTemp` is `TRUE` when the difference of the voltages on channels 0 and 4 is greater than 3.55 volts. If you wish to create variables with a negated sense, you may reverse the signs of the comparison tests.

For example, we could create `a1_LowTemp` by defining the variable as:

```
A/D Logic Variable #1 is defined: YES

Description of A01 variable: 'LowTemp '

Input summation Term #1: A4

Input summation Term #2: -A0

Input summation Term #3: Zero

Input summation Term #4: Zero

Test for ( A4-A0 > -2.55 Volts )
```

These can then be combined in a logic equation as follows:

```
EQ00: # V0 will be TRUE when the temperature is normal

\-: v0 = !(a0_HiTemp | a1_LowTemp)

-->
```

8.8.7. STATUS Command

This command prompts you with a series of questions to choose whether each of the status input lines are received by RCP8 and what the polarity of those inputs are.

8.8.8. INU Command

> The set-up questions listed in your configuration vary depending upon your set-up and how your have responded to previous questions.

This command configures the optional Inertial Navigation Unit (INU). The INU provides RCP8 with navigation and attitude information that is necessary for stabilizing a moving platform.

Use platform stabilization algorithms: `YES`

A **NO** response disables all INU features and inhibits all coordinate transformations between pedestal and Earth frames (the 2 frames are assumed identical.) None of the questions below are displayed.

A `YES` response enables the separate pedestal and Earth coordinate frames and prompts the user with the following additional questions:

Negate sign of Roll angles: `NO`

Negate sign of Pitch angles: `NO`

Negate sign of Heading angles: `NO`

Use these questions if you must change the sign of the attitude angles.

Roll offset from true orientation: `0.00 deg`

Pitch offset from true orientation: `0.00 deg`

Heading offset from true orientation: `0.00 deg`

Use these questions if there are fixed offsets in the attitude angles. Typically, this occurs if the INU is not bolted directly in line with the ship's principal axes.

Lead time for velocity extrapolation: `0.050 sec`

The stability of the Earth-frame velocity servo can sometimes be improved by leading the time derivatives of the attitude angles by a small amount. Typical values are between 0 and 80 ms.

Dead INU detection time: `5.0 sec`

RCP8 stops performing coordinate transformations if the INU data stream is absent for more than this length of time. During the dead time, all INU angles and velocities are artificially set to 0.

Built-in INU Simulation: `External`

RCP8 contains an internal INU simulator that is useful during program development as well as for testing simulated moving environments.

- The response **`OFF`** disables the simulator and suppresses the remaining questions in this section.
- The response **`EXTERNAL`** results in simulated INU SDLC data signals generated at the INU backpanel connector. This simulated stream can be looped back into the normal INU inputs for testing.
- The response **`INTERNAL`** runs the same INU data simulation, but internally loops it back into RCP8 and does not generate any SDLC output signals.

The simulated motion is sinuosity on each axis and includes an adjustable amplitude, a center value, and a period. The default values that are displayed are simulating rather rough conditions in presumably bad weather.

Amplitude of motion for Roll axis: `12.00 deg`

Amplitude of motion for Pitch axis: `8.00 deg`

Amplitude of motion for Heading axis: `80.00 deg`

Sets the peak amplitude of simulated motion on each axis.

Center of motion for Roll axis: `0.00 deg`

Center of motion for Pitch axis: `0.00 deg`

Center of motion for Heading axis: `0.00 deg`

Sets the center of simulated motion on each axis.

Period of motion for Roll axis: `15.0 sec`

Period of motion for Pitch axis: `13.0 sec`

Period of motion for Heading axis: `60.0 sec`

Sets the period of simulated motion of each axis.

Serial INU Simulation: `NO`

9. Technical Data

9.1. Antenna Control I/O and Features

Table 18 Antenna and Antenna Control I/O Features

Feature	Description
Approach	Digital position and velocity servos with interactive software parameter tuning.
AZ/EL Position Input	TTL 16-bit binary angle, 14-bit BCD or 90 V 60 Hz synchro (nominal).
AZ/EL Position Output	TTL 16-bit binary angle.
AZ/EL Tachometer Input (if available)	±70 V signed analog input voltage.
Servo drive error	±10 V analog output to AZ/EL servo amplifiers
Servo Control/Status	On/Off control using TTL or switch closure output. On/Off status via wide range input.
Antenna	Local mode switch input, switch closure or TTL.
Alternate control relay signal:	12 V output to external relays to switch to alternate control such as handwheels when the antenna is in local mode or shutdown state.

9.2. Fail-safe Antenna Protection

Table 19 Antenna Protection Features

Feature	Description
Elevation soft limits	Automatic software override brings antenna to a gentle stop at the specified limits.
Elevation shutdown limits:	Antenna is placed in shutdown state if the upper or lower limit is exceeded.
Elevation limit switch inputs	EL Hi/Low TTL or switch closure. Antenna is placed in a shutdown state if a limit switch is encountered.
Tachometer check watchdog	The calibrated tachometer is compared to the differenced position for consistency. Shutdown is invoked if the check fails. Protects against loss of tach or position sensors.

Feature	Description
Antenna response watchdog	Based on an internal model of the antenna inertia and damping moments, the response of the antenna to drive output requests is checked for consistency. Shutdown is invoked if the check fails. This protects the antenna from a jammed or broken gear, or if equipment such as a scaffold is inadvertently hit by the antenna.
Antenna max. speed watchdog	Shutdown is invoked if the specified maximum speed of the antenna is exceeded.
Antenna max. acceleration limit	Based on the internal antenna model, this check limits the drive output to stay within a specified acceleration limit.

9.3. Optional Shipboard Stabilization

Table 20 Shipboard Stabilization Features

Feature	Description
Approach	Stabilization algorithms scan the antenna in earth coordinates using AZ and EL velocity and position servos that are adjusted for the pitch, roll and heading (and rates of change) of the platform.
Platform Motion Sensor Input	Pitch, roll heading and rates and absolute platform position and velocity from inertial navigation system such as the Honeywell MAPS system on SDLC serial line or Seatex, Inc. Seapath 200 system on RS232C serial line. GPS update and at-sea alignment are highly recommended for any INU system.
Range of operation	Typical 0 ... 65°elevation (earth relative) for up to 15° of attitude change. Antenna pedestal should be capable of ~ -20° of elevation.
Typical performance	0.1° of accuracy for elevation angles in the range 0 to 65° for inclinations, up 15° over 10-second periods. Exact performance depends on servo drive performance.
Built-in display	Selectable earth or pedestal relative AZ/EL position and/or velocity.

9.4. Radar Status and Control I/O

Table 21 Radar Status and Control I/O Features

Feature	Description
I/O configuration.	The ***softplane.conf*** file is used to configure the I/O pin assignments to the connector panel. This removes the need for jumpers and significantly reduces custom wiring typical of legacy systems.
Standard status inputs	Wide range inputs. Standard parameters are Servo Power, Antenna Local Mode, Lower and Upper EL Limit switches, T/R power, T/R Local Mode, Radiate Standby, Radiate on, Magnetron Current, Wave Guide Pressure, Airflow, Interlock, external Reset input signal, Trigger Blanking input signal and pulse width (up to 4 coded in 2 bits).
Standard control outputs	Wide range outputs or switch closures to ground. Standard parameters are Servo power, T/R power On/Off, Radiate On/Off (TTL or switch closure), Cabinet Relay, Trigger Blanking, equipment Reset signal, and pulse width (up to 4 bits).
BITE I/O	Up to 100 TTL lines configurable in groups of 10 to be either input or output lines. These are used for BITE/IO. 100 additional lines can be added by adding a second I/O-62 card.
Programmable control logic	User programmable status/control logic actions in a flexible C-like programming interface. For example, if the antenna radome is opened, the system can automatically sound an alarm for a programmable time and immediately de-radiate and stop the antenna.

9.5. Vaisala I/O-62 PCI Card

Table 22 Vaisala I/O-62 Features

Feature	Description
PCI card	Short format PCI card with 62-position D connector. Multiple cards may be installed.
D/A, A/D, discrete inputs and outputs (TTL, wide range, RS422, and similar)	See the following summary table.
Expandable I/O	Allows the addition of a second I/O-62 and connector panel.
I/O pin assignment	Mapping by ***softplane.conf*** file allows easy reconfiguration of the pin assignments without custom wiring.

Feature	Description
ESD protection	Uses Tranzorb™ silicon avalanche diode surge suppression and high-voltage tolerant components.

Table 23 Vaisala I/O-62 Summary of Electrical Interfaces

Quantity	Description
40	Lines configurable in groups of 8 to be either inputs or outputs. The electrical specifications are software defined within each group as follows: • Single-ended TTL input or output with software-configured pull-up or pull-down resistors (2.2 kΩ) for inputs. Logic threshold +2.5 VDC. Output impedance 120 Ω. • Wide range inputs (±27 VDC, threshold +2.5 VDC), often used for lamp voltage status inputs. • RS-422/485 @ 10 MBit/sec (requires 2 lines each). Low output impedance. RS-422 receivers can be configured in software to have 100 ohm termination between each pair.
8	A/D convertors configurable as 0, 4, or 8 convertors, ±2 V, 12 bits @ 10 MHz, These lines are shared with some of the 40 I/O lines listed above.
2	D/A convertors, ±10 V 1 MHz update rate, output can drive a 75 ohm load.
2	SPDT relays on the board. These are often used for switching high power relays. Contacts are diode protected.
2	RS-232C full duplex lines (Tx and Rx)
4	12 V 75 Ω trigger drivers.
2	Power/Ground pairs of 12 V power (filtered, fused) for external equipment or remote backpanel use (up to 24 W total). Polyfuse technology acts like a circuit breaker with auto reset in the event of an overload.
8	Ground wires for signal grounds from the remote back panel.

Table 24 RVP and RCP8 I/O-62 Card Jumper Settings

Jumper	ID	Description	AB	BC	Not Installed
JP1	BOOT	Controls the card boot-up.	X	X	Normal boot

Jumper	ID	Description	AB	BC	Not Installed
JP2	`JTAG`	Enables on-board flash re-programming for code version upgrades. Other settings are reserved for maintenance functions.	Enable flash	Maintenance	Maintenance
JP3	`TTYX0/ RSV`	These jumpers assign dedicated hardware I/O lines to pins on DBF62 connector on the back of the I/O-62. The selections are made among: • The two RS232 lines noted as `TTY0` or `TTYt` with transmit and recieve for each. • Four trigger output lines. • The three contact positions of the onboard DIP relays (K1 and K2). The **AB** column shows the specific pins.	Pin 47 `TTYX0`	X	X
JP4	`TRIG0/ TTYX0`		Pin 49 `TRIG0`	`TTYX0`	X
JP5	`TRIG1/ K1NC`		Pin 51 `TRIG1`	K1 Normally closed contact	X
JP6	`TRIG2/ K1NO`		Pin 53 `TRIG2`	K1 Normally open contact	X
JP7	`TRIG3/ K1CT`		Pin 55 `TRIG3`	K1 center contact	X
JP8	`TTYR0/ K2NC`		Pin 57 `TTYR0`	K2 Normally closed contact	X
JP9	`TTYX1/ K2NO`		Pin 59 `TTYX1`	K2 Normally open contact	X
JP10	`TTYR1/ K2CT`		Pin 61 `TTYR1`	K2 Center contact	X

9.6. I/O Connector Panel

The RCP connector panel mounts on front or rear of standard 19 EIA rack and connects to I/O-62 via 1:1 62-pin 1.8 m (5 ft 11 in) cable (provided). It includes

- Three internal relays and 4 12 V relay control signals for switching external devices.
- Diagnostic power supply and self test LEDs for troubleshooting.

The connector panel uses programmable pin assignments made in `softplane.conf` and multiplexes the inputs and outputs of the I/O-62 PCI card.

Table 25 I/O Connector Panel Summary

J-ID	Label	Type	Description
J1	`AZ INPUT`	DBF25	20 TTL lines configurable to be either input or output lines. Maximum input level allowed is TTL.
J2	`AZ OUTPUT`	DBF25	20 TTL lines configurable to be either input or output lines. Maximum input level allowed is TTL.
J3	`CONTROL`	DBF25	16 assignable digital control/status lines: • Maximum input voltage is ±27 VDC • Configurable to RS-422 differential or TTL input/output • In TTL: Logical threshold of +2.5 VDC, output impedance of 120 Ω and configurable pull up (+5 VDC) or pull down (GND) resistor of 2.2 kΩ available. • In RS-422: 100 Ω termination across differential line pairs available. Low output impedance. Two fixed RS422 differential lines.
J4	`EL INPUT`	DBF25	20 TTL lines configurable to be either input or output lines. Maximum input level allowed is TTL.
J5	`EL OUTPUT`	DBF25	20 TTL lines configurable to be either input or output lines. Maximum input level allowed is TTL.
J6	`RELAY`	DBF25	Three internal relays, contact rating 0.5 A continuous. The switching load is 0.25 A and 100 V, with the additional constraint that the total power not exceed 4 VA. Four 12 V relay control signals, up to 200 mA. Note that external relays should be equipped with proper diode protection to shunt the back EMF.
J7	`BITE 19:0`	DBF25	20 TTL lines configurable to be either input or output lines. Maximum input level allowed is TTL.
J8	`ANALOG IN`	DBF25	10 differential analog inputs, up to ±20 V max multiplexed into A/D convertor sampling each at >1000 Hz.

J-ID	Label	Type	Description
J9	`PED/STATUS`	DBF25	AZ/EL tachometer differential inputs (±2 to ±70 VDC) and AZ/EL drive outputs (±10 VDC). 14 assignable digital control/status lines: • Maximum input voltage ±27 VDC - Configurable to RS-422 differential or TTL input/output • In TTL: Logical threshold of +2.5 VDC, output impedance of 120 Ω and configurable pull up (+5 VDC) or pull down (GND) resistor of 2.2 kΩ available. • In RS-422: 100 Ω termination across differential line pairs available. Low output impedance.
J10	`SERIAL`	DBF9	RS232C
J11	`SERIAL`	DBF9	RS232C
J12	`S-D`	Modular	3 × 4 matrix connector for AZ and EL synchro and reference inputs (nominally 90 V and 60 Hz).
J13	`TP1`	BNC	Programmable scope test point. 75 Ω
J14/15	`SPARE`	BNC	
J16	`TP2`	BNC	Programmable scope test point. 75 Ω
J17/18	`SPARE`	BNC	

9.7. Physical and Environmental Characteristics

Table 26 RVP902-IO Physical and Environmental Characteristics

Item	Description
Server	• Motherboard configuration 2U rackmount with 2 PCI slots • 700W redundant power supply • 3 * 3.5" hot-swap drive basy • IO 62 card
Dimensions Standard 2U chassis	height*width*depth: 89 mm (3.5") * 437 mm (17.2") * 450 mm (17.7")
Weight	15.9 kg (35 lbs)
Input power	100...240 V, 60...50 Hz, 10...4 A

Item	Description
Power consumption	Max. 700 W, typical 200 W
Environment	• Temperature: 5 °C (41 °F) … 35 °C (95 °F) • Humidity: 8 … 90 % non-condensing
Reliability	MTBF>22 500 hours

Appendix A. Communication Formats

A.1. Serial Data Format

RCP8 is controlled by a two–way, asynchronous RS-232 data line that is typically run at speed of 19.2 K baud.

A host computer controls the servo and the antenna while receiving feedback status.

Information is transferred in packets of 2 or more bytes. Each packet begins with a `SYNC` byte and ends with an `END` byte of `FF(Hex)`. All `SYNC` bytes have the MSB set and the value indicates the type of packet to follow. The available packets are `80(hex)` for antenna, `C0(hex)` for BITE, and `B0(hex)` for time. Each packet–type has a specific direction of travel, such as to or from RCP8. Packets can arrive in any order in the serial stream.

The time stamp is a 14-bit counter incremented by RCP8 once per millisecond. RCP8 should latch all the data for a packet at the same time. This counter allows the host computer to accurately judge the time between samples without the serial line latencies and fluctuations due to the time sharing operating system.

Table 27 Supported Serial Data Formats

Format	Purpose
RCV01 XMT01	For older systems
RCV02 XMT02	For newer systems.
RCV03	For systems on moving platforms, such as ships or airplanes. These correct the radar's measured radial velocity for the motion of the platform. To make this correction, the three–dimensional velocity and orientation of the platform must be recorded. Typically, the information comes from an inertial navigation system (INU). For shipboard systems, an update rate of approximately 20 reports per second can satisfy the velocity correction requirements at 19 200 baud.

Table 28 Angle Values in Serial Data Formats

Angle Value	Description	Transmit Value
Angular speed	In the XMT01 format, the angular speed is a signed number in units of 0.55º/s. In other formats, the angular rates are in signed 14-bit binary angles per second. The largest possible value is 180º/s (30 rpm) and the step is 0.022º/s. All velocities are in signed cm/sec with the altitude in signed meters. If some of the information is unavailable at the full resolution of the data format, the low bits are filled with zeros.	N/A
Azimuth and elevation	The azimuth and the elevation angles are corrected angles relative to the north and are the angles that the antenna is pointed relative to the deck of the platform. These calculations are derivable from the other angles but are also reported to assist in data analysis, especially if one of the sensors or the stabilization fails.	14-bit binary angles
Heading	The direction the platform is pointed. This is not the same as direction of motion. The platform could be pointed one way and drifting backwards.	14-bit binary angles
Latitude and longitude	The position of the platform is reported by the latitude, the longitude, and the altitude. On ships, since the altitude may not be implemented for systems on ships, the setting is 0.	21-bit binary angles
Pitch	The pitch is the angle between the fore-and-aft axis of the platform and the horizontal is measured in the vertical plane. The pitch is positive when the bow is down. The pitch is measured in the plane perpendicular to the fore-and-aft axis, which is generally not the vertical plane, and The pitch can be directly measured by a level on the fore-and-aft axis but the roll cannot be directly measured by a one-axis tilt meter.	14-bit binary angles
Roll	The roll is the rotation angle about the fore-and-aft axis in its pitched position. The roll is positive when the deck is down on the port side.	14-bit binary angles

A.2. Socket Data Format

RCP can interface with other machines using a socket interface instead of a serial line.

This format uses the serial data format with an additional 16-byte header added, as shown in the following table. Note that for the ASCII packet types, there are C-style `#defines` in the *`antenna_lib.h`* file with names of the form `ANT_PKT_TYPE*`.

This format uses multicast UDP packets. Typically the IP address is 224.0.0.3, port 30785 but it can be configured.

Note that with socket data, multiple packets can arrive as one, so the packet size is used to separate.

On a multicast address, the reader reads everything written. The first letter of the packet type is used to determine direction, and it ignores everything starting with an `R`.

If you write your own code to interface to this, note that on a computer with multiple network cards, you must explicitly specify which one to write out on. The destination address is not used at this time, you can fill it with **`htonl(INADDR_ANY)`**.

Table 29 Socket Header Format

Byte	Function
1–4	4-char ASCII size of the packet in %04d format.

Byte	Function	
5–12	8–char ASCII packet type, 0 padded, choices are:	
	XMT01	Standard antenna XMTnn packets
	XMT02	
	XMT05	
	RCV01	Standard antenna RCVnn packets
	RCV02	
	RCV03	
	RCV05	
	XMTSA	Scientific Atlanta controller
	RCVSA	
	XCHAT	Chat mode packet
	RCHAT	
	RTIME	Time packet
	XBITEC	BITE control/status packet
	RBITES	
	XBINTROG	BITE 'Interrogate' packet
	RBINTROG	
	XBSAMPLE	BITE 'Take Sample' packet
	RBSAMPLE	
	XBSETVAL	BITE 'Take Sample' packet
	RBSETVAL	
	XBRESET	BITE 'Reset' packet
	RBRESET	
12–16	4–byte destination address in network byte order (`INADDR_ANY` fine)	

A.3. Antenna Status Formats

Table 30 Status Packet RCV01 Format (RCP8 to Host)

Char	Function	
1	SYNC Byte (80 Hex)	
2	Azimuth Low 7 bits	
3	Azimuth High 7 bits	
4	Elevation Low 7 bits	
5	Elevation High 7 bits	
6	Status #1	D6 = Low air flow
		D5 = Low Waveguide pressure
		D4 = Servo power
		D3 = Antenna Local mode
		D2 = Interlock
		D1 = Standby
		D0 = Radiate On
7	Status #2	D6 = RCP8 is shutdown
		D5 = LSB pulse width
		D4 = T/R power On
		D3 = T/R Local mode
		D2 = Encoders calibrated
		D1 = MSB pulse width
		D0 = Magnetron current normal
8	End Of Message (FF Hex)	

Table 31 Status Packet RCV02 Format (RCP8 to Host)

Char	Function	
1	SYNC Byte (80 Hex)	

Char	Function	
2	Azimuth Low 7 bits	
3	Azimuth High 7 bits	
4	Elevation Low 7 bits	
5	Elevation High 7 bits	
6	Azimuth Rate Low 7 bits	
7	Azimuth Rate High 7 bits	
8	Elevation Rate Low 7 bits	
9	Elevation Rate High 7 bits	
10	Status #1	D6 = Low air flow
		D5 = Low Waveguide pressure
		D4 = Servo Power
		D3 = Antenna Local mode
		D2 = Interlock Open
		D1 = Standby
		D0 = Radiate On
11	Status #2	D6 = RCP8 is shutdown
		D5 = LSB pulse width
		D4 = T/R power On
		D3 = T/R Local mode
		D2 = Azimuth encoder calibrated
		D1 = MSB pulse width
		D0 = Mag. current normal

Char	Function	
12	Status #3	D6 = IRIS Mode 2
		D5 = IRIS Mode 1
		D4 = IRIS Mode 0
		D3 = Elevation encoder calibrated
		D2 = Signal Generator fault
		D1 = Signal Generator On
		D0 = Signal Generator CW
13	Signal generator level (0 = max power)	
14	Time Stamp Low 7 bits	
15	Time Stamp High 7 bits	
16	END OF MESSAGE (FF Hex)	

Table 32 Status Packet RCV03 Format (RCP8 to Host)

Char	Function	
1	SYNC Byte (80 Hex)	
2	Identification byte	
3	Azimuth Low 7 bits (Earth relative)	
4	Azimuth High 7 bits	
5	Elevation Low 7 bits (Earth relative)	
6	Elevation High 7 bits	
7	Train Order Low 7 bits (azimuth of pedestal relative to the ship)	
8	Train Order High 7 bits	
9	Elevation Order Low 7 bits (elevation of pedestal relative to the ship)	
10	Elevation Order High 7 bits	
11	Pitch Low 7 bits	
12	Pitch High 7 bits	

Char	Function	
13	Roll Low 7 bits	
14	Roll High 7 bits	
15	Heading Low 7 bits	
16	Heading High 7 bits	
17	Azimuth Rate Low 7 bits	
18	Azimuth Rate High 7 bits	
19	Elevation Rate Low 7 bits	
20	Elevation Rate High 7 bits	
21	Pitch Rate Low 7 bits (LSB = Zero)	
22	Pitch Rate High 7 bits	
23	Roll Rate Low 7 bits (LSB = Invalid Roll)	
24	Roll Rate High 7 bits	
25	Heading Rate Low 7 bits (LSB = Invalid Heading)	
26	Heading Rate High 7 bits	
27	Status #1	D6 = Low air flow
		D5 = Low Waveguide pressure
		D4 = Servo power
		D3 = Antenna Local mode
		D2 = Interlock open
		D1 = Standby
		D0 = Radiate ON

Char	Function	
28	Status #2	D6 = RCP8 is shutdown
		D5 = LSB pulse width
		D4 = T/R Power on
		D3 = T/R Local mode
		D2 = Azimuth encoder calibrated
		D1 = MSB pulse width
		D0 = Mag. current normal
29	Status #3	D6 = Reserved
		D5 = Reserved
		D4 = Reserved
		D3 = Elevation encoder calibrated
		D2 = Signal Generator fault
		D1 = Signal Generator On
		D0 = Signal Generator CW
30	Signal generator value (0 = full signal)	
31	Time Stamp Low 7 bits	
32	Time Stamp High 7 bits	
33	Latitude Low 7 bits	
34	Latitude Middle 7 bits	
35	Latitude High 7 bits	
36	Longitude Low 7 bits	
37	Longitude Middle 7 bits	
38	Longitude High 7 bits	
39	Altitude Low 7 bits	
40	Altitude High 7 bits	
41	Velocity East Low 7 bits (LSB = Invalid Lat/ Lon)	
42	Velocity East High 7 bits	

Char	Function	
43	Velocity North Low 7 bits (LSB = Zero)	
44	Velocity North High 7 bits	
45	Velocity Up Low 7 bits (LSB = Invalid Altitude)	
46	Velocity Up High 7 bits	
47	END OF MESSAGE (FF Hex)	

Table 33 Status Packet RCV05 Format (RCP8 to Host)

Char	Function	
1–15	These bytes exactly match RCV02 / RCV04 format	
1–16	Dual-System Status	D6 = RCP8 is configured as a Dual-System
		D5 = Dual-System Mode MSB
		D4 = Dual-System Mode LSB
		D3 = This packet was sent from Unit "A"
		D2 = Information is known about the "Other" unit
		D1 = Unit "A" is the preferred system
		D0 = Unit "B" is disabled
Note: The 2-bit Dual-System Mode codes are:		
00 : Unknown 01 : System "A" 10 : System "B" 11 : Auto Switch		
17	Dual-System Status	D6 = Unit "B" is okay
		D5 = Unit "B" Activity Code MSB
		D4 = Unit "B" Activity Code LSB
		D3 = Unit "A" is disabled
		D2 = Unit "A" is okay
		D1 = Unit "A" Activity Code MSB
		D0 = Unit "A" Activity Code LSB
Note: The 2-bit Dual-System Activity codes are:		

Char	Function	
00 : Inactive 01 : Warmup 10: Active Now 11 : Reserved		
18	Dual-System Status	D6 = RCP8 is configured for voluntary flipping
		D5 = Unit "B" is offering to give up control
		D4 = Unit "A" is offering to give up control
		D3 = Unit "B" would be used if it were available
		D2 = Unit "A" would be used if it were available
19	Polarization Status	D2:0 = Current Polarization XMT control
		0 = Horizontal; 1 = Vertical; 2 = Alternating; 3 = Simultaneous
		D3 = Polarization switch is OK to XMT
20	Spare	
21	Spare	
22	Spare	
23	Spare	
24	END OF MESSAGE (FF Hex)	

A.4. Antenna Control Formats

Table 34 Control Packet XMT01 Format (Host to RCP8)

Char	Function	
1	SYNC Byte (80 Hex)	
2	Azimuth Low 7 bits	
3	Azimuth High 7 bits	
4	Elevation Low 7 bits	
5	Elevation High 7 bits	

Char	Function	
6	Control Word #1	D6 = MSB of Pulse Width
		D5 = Leave Pulse Width unchanged
		D4 = Spare
		D3 = Signal Generator On
		D2 = Signal Generator CW
		D1 = EL (1 = Scan, 0 = Position)
		D0 = AZ (1 = Scan, 0 = Position)
7	Control Word #2	D6 = Reset RCP8 on edge
		D5 = Noise Source On
		D4 = LSB of Pulse width
		D3 = Radiate On complemented
		D2 = Radiate On
		D1 = Servo Power On
		D0 = T/R Power On
8	Control Word #3 (all spare)	
9	Signal generator level (unsigned 0–127 dB attenuation)	
10	AZ/EL Antenna speed (signed 7 bit, 0.55 degree resolution)	
11	END OF MESSAGE (FF Hex)	

Table 35 Control Packet XMT02 Format (Host to RCP8)

Char	Function	
1	SYNC Byte (80 Hex)	
2	Azimuth Low 7 bits	
3	Azimuth High 7 bits	
4	Elevation Low 7 bits	
5	Elevation High 7 bits	

Char	Function	
6	Control Word #1	D6 = MSB of Pulse Width
		D5 = Leave Pulse Width unchanged
		D4 = Spare
		D3 = Signal Generator On
		D2 = Signal Generator CW
		D1 = EL (1 = Scan, 0 = Position)
		D0 = AZ (1 = Scan, 0 = Position)
7	Control Word #2	D6 = Reset RCP8 on edge
		D5 = Noise Source On
		D4 = LSB of Pulse width
		D3 = Radiate On complemented
		D2 = Radiate On
		D1 = Servo Power On
		D0 = T/R Power On
8	Control Word #3	D6 = IRIS Mode 2
		D5 = IRIS Mode 1
		D4 = IRIS Mode 0
		D3 = Radar Workstation A okay
		D2 = Radar Workstation B okay
		D1 = Data Processor A okay
		D0 = Data Processor B okay
9	Signal Generator level (0–127 dB attenuation)	
10	AZ Antenna Speed Low 7 bits	
11	AZ Antenna Speed High 7 bits	
12	EL Antenna Speed Low 7 bits	
13	EL Antenna Speed High 7 bits	
14	END OF MESSAGE (FF Hex)	

Table 36 Control Packet XMT05 Format (Host to RCP8)

Char	Function	
1#endash13	These bytes exactly match the XMT02 / XMT04 format	
14	Control Word #4	D6 = Dual-System: Mode MSB
		D5 = Dual-System: Mode LSB
		D4 = Dual-System: Offer to relinquish control
		D3 = Dual-System: Unit would be used if available
		D2 = Spare
		D1 = Spare
		D0 = Spare
Note: The 2-bit Dual-System Mode codes are:		
	00 : No change	01 : System "A"
	10 : System "B"	11 : Auto Switch
15	Control Word #5	D2:0 = Requested Polarization XMT control
		0 = Horizontal 1 = Vertical 2 = Alternating 3 = Simultaneous
		7 = Unchanged
		D6:3 = Spare
16	Spare	
17	Spare	
18	END OF MESSAGE (FF Hex)	

A.5. BITE Formats

The **BITE** status packet consists of a packet from 3 ... 20 bytes in length. The first two bytes and the last byte are used for identification. The bytes in the middle must have their MSB zero, but can contain arbitrary status in the lower 7 bits.

This is typically used to report test results in the individual bits, such as cabinet interlocks, airflow sensors, and power supply checks.

A.5.1. ARA ACU-3 BITE Packet

Table 37 ARA ACU-3 BITE Packet (RCP8 to Host)

Char	Function
1	SYNC Byte (C0 Hex)
2	Identification byte (User Choice)
3	Status Bits EL1 AZ6 AZ5 AZ4 AZ3 AZ2 AZ1
4	Status Bits ELF AZF EL6 EL5 EL4 EL3 EL2
5	Status Bits - - - - - - Timeout
6	Spare byte
7	END OF MESSAGE (FF Hex)

A.5.2. Auxiliary Status/Control Packet

RCP8 contains 64 auxiliary status and control variables, labeled `S[0:63]` and `C[0:63]`.

These bits may be sent to and from the host computer in the form of 13-byte BITE packets holding the full set of 64 bits.

The format of these packets is the same in both directions, and the identification byte is selectable so that conflicts with other `BITE` packets can be avoided. A subset of the auxiliary bits may optionally be assigned to electrical input and output lines on an I/O-62 card using the *`softplane.conf`* file. The auxiliary can also be set and accessed with logic equations.

Table 38 Auxiliary Status/Control BITE Packets

Char	Function
1	SYNC Byte (C0 Hex)
2	Identification byte (User Choice)
3	Control/Status Bits 6 5 4 3 2 1 0
4	Control/Status Bits 13 12 11 10 9 8 7
5	Control/Status Bits 20 19 18 17 16 15 14
6	Control/Status Bits 27 26 25 24 23 22 21
7	Control/Status Bits 34 33 32 31 30 29 28

Char	Function
8	Control/Status Bits 41 40 39 38 37 36 35
9	Control/Status Bits 48 47 46 45 44 43 42
10	Control/Status Bits 55 54 53 52 51 50 49
11	Control/Status Bits 62 61 60 59 58 57 56
12	Control/Status Bit 63
13	END OF MESSAGE (FF Hex)

A.5.3. CAN Bus BITE Packets

Table 39 CAN Bus BITE Packet (RCP8 to Host)

Char	Function	Status Bit	Default	Active
1	**SYNC (C0 Hex)**			
2	**Identification byte (User choice)**			
3	**CAN-bus Status and Fault Conditions**			
	D6 EL Belt Switch	S70	`LOW`	`HIGH`
	D5 EL Driver Current Limit	S69	`LOW`	`HIGH`
	D4 EL Driver Alarm	S68	`HIGH`	`LOW`
	D3 EL Driver Ready	S67	`LOW`	`HIGH`
	D2 AZ Driver Current Limit	S66	`LOW`	`HIGH`
	D1 AZ Driver Alarm	S65	`HIGH`	`LOW`
	D0 AZ Driver Ready	S64	`LOW`	`HIGH`

Char	Function	Status Bit	Default	Active
4	**CAN-bus Status and Fault Conditions**			
	D6 Spare	S77	LOW	--
	D5 Spare	S76	LOW	--
	D4 Spare	S75	LOW	--
	D3 Spare	S74	LOW	--
	D2 Lower EL Limit	S73	LOW	HIGH
	D1 Upper EL Limit	S72	LOW	HIGH
	D0 AZ Belt Switch	S71	LOW	LOW
5	**CAN-bus Status and Fault Conditions**			
	D6 Spare	S84	LOW	--
	D5 Gear Heaters	S83	HIGH	LOW
	D4 Motor Driver Heaters	S82	HIGH	LOW
	D3 Motor Driver Fans	S81	LOW	LOW
	D2 CAN-bus Broken	S80	LOW	HIGH
	D1 Spare	S79	LOW	--
	D0 Spare	S78	LOW	--
6	**CAN-bus Status and Fault Conditions**			
	D6 Spare	S91	LOW	--
	D5 Spare	S90	LOW	--
	D4 Spare	S89	LOW	--
	D3 Spare	S88	LOW	--
	D2 Spare	S87	LOW	--
	D1 EL Encoder OK	S86	LOW	HIGH
	D0 AZ Encoder OK	S85	LOW	HIGH

Char	Function	Status Bit	Default	Active
7	**CAN-bus Status and Fault Conditions**			
	D6 EL encoder cycle too short	S98	LOW	HIGH
	D5 EL encoder memory error	S97	LOW	HIGH
	D4 EL encoder hardware alarm	S96	LOW	HIGH
	D3 Spare	S95	LOW	--
	D2 Spare	S94	LOW	--
	D1 Spare	S93	LOW	--
	D0 Spare	S92	LOW	--
8	**CAN-bus Status and Fault Conditions**			
	D6 Spare	--	LOW	--
	D5 Spare	--	LOW	--
	D4 AZ encoder protocol error	--	LOW	HIGH
	D3 AZ encoder cycle too short	--	LOW	HIGH
	D2 AZ encoder memory error	--	LOW	HIGH
	D1 AZ encoder hardware alarm	--	LOW	HIGH
	D0 EL encoder protocol error	--	LOW	HIGH
9	END OF MESSAGE (FF Hex)			

Table 40 CAN Bus Q-BITE Packet (RCP8 to Host)

Char	Function	Scale
1	SYNC (AF Hex)	
2	Identification byte (User choice)	
3-4	Azimuth Motor Current [A]	scale 1000
5-6	Elevation Motor Current [A]	scale 1000
7-8.	Azimuth Motor Temperature [ºC]	scale 10
9-10	Elevation Motor Temperature [ºC]	scale 10

Char	Function	Scale
11-12	Radom Room Temperature [ºC]	scale 10
13-14	Equipment Bay Temperature [ºC]	scale 10
15-16	Spare	
17-18	Spare	
19	END OF MESSAGE (FF Hex)	

CAN-Bus can be controlled through auxiliary control bits.

Table 41 CAN Bus Auxiliary Control

Function	Control Bit
AZ Driver ON	c64
EL Driver ON	c65
AZ Alarm Reset	c66
EL Alarm Reset	c67

A.5.4. Generic BITE Packet

Table 42 Generic BITE Packet (RCP8 To/From Host)

Char	Function
1	SYNC Byte (C0 Hex)
2	Identification byte (00 Hex)
3	Status byte #1
4	Status byte #2
.	
.	
N-1	Status byte #N-3
N	END OF MESSAGE (FF Hex)

A.5.5. Internal BITE Packet

RCP8 can optionally generate an internal `BITE` packet. These bits convey additional status information that is not contained in the other transmission formats. The shutdown status of RCP8 (up to 32 different conditions) is contained in the first five bytes. The last five bytes hold other miscellaneous information. The identification byte is selectable to avoid conflicts with other `BITE` packets.

Table 43 Internal BITE Packet (RCP8 to Host)

Char	Function	
1	SYNC Byte (C0 Hex)	
2	Identification byte (User Choice)	
3	Shutdown Conditions 0-6	
		D6 = EL Velocity Exceeded
		D5 = AZ Velocity Exceeded
		D4 = EL Axis Unresponsive
		D3 = AZ Axis Unresponsive
		D2 = EL Tach Inconsistent
		D1 = AZ Tach Inconsistent
		D0 = Diagnostics Failed
4	Shutdown Conditions 7-13	
		D6 = IP-SERIAL Conflicts
		D5 = EL Upper Lim Switch
		D4 = EL Lower Lim Switch
		D3 = EL-UP Shutdown Limit
		D2 = EL-LO Shutdown Limit
		D1 = AZ-HI Shutdown Limit
		D0 = AZ-LO Shutdown Limit

Char	Function	
5	Shutdown Conditions 14-20	
		D6 = Reserved
		D5 = Reserved
		D4 = Reserved
		D3 = Reserved
		D2 = IP-DIGITAL-48 Conflicts
		D1 = Output Remap Conflict
		D0 = Missing IP-SYNCHRO
6	Shutdown Conditions 21-27	
		D6 = Reserved
		D5 = Reserved
		D4 = Reserved
		D3 = Reserved
		D2 = Reserved
		D1 = Reserved
		D0 = Reserved
7	Shutdown Conditions 28-31	
		D6 = Spare
		D5 = Power-up error(s) occured
		D4 = RCP8 is shutdown (OR of Bits 0-31)
		D3 = User Shutdown #2
		D2 = User Shutdown #1
		D1 = Reserved
		D0 = Reserved

Char	Function	
8	INU Status	
		D6 = Invalid horizontal position/velocity
		D5 = Reduced vertical position/velocity
		D4 = Invalid vertical position/velocity
		D3 = Reduced roll and pitch
		D2 = Invalid roll and pitch
		D1 = Reduced heading
		D0 = Invalid heading
9	Antenna/Radar/Servo and INU status	
		D6 = Reduced horizontal position/velocity
		D5 = No INU Data Stream
		D4 = T/R Power On
		D3 = T/R Local mode
		D2 = LSB pulse width
		D1 = MSB pulse width
		D0 = Mag. current normal
10	Antenna/Radar/Servo status	
		D6 = Low air flow
		D5 = Low Waveguide pressure
		D4 = Servo Power
		D3 = Antenna Local mode
		D2 = Interlock Open
		D1 = Standby
		D0 = Radiate On
11	Local Variables V6, V5, V4, V3, V2, V1, V0	
12	Local Variables V13, V12, V11, V10, V9, V8, V7	
13	END OF MESSAGE (FF Hex)	

A.5.6. BITE Interrogate and Request Packets

The `BITE` "interrogate" packet is a request to a remote device that it immediately reply with its current `BITE` packet(s). This is how the local device can insure that it has the most recent valid data.

RCP8 sends `BITE` "interrogate" packets to the host computer whenever RCP8 is expecting to receive `BITE` packets of any sort. These RCP8 "interrogate" requests are sent every 30 seconds beginning at startup. This insures that all control bits are valid in RCP8 immediately upon startup, and can resume their correct states after any serial line interruptions.

RCP8 responds to incoming `BITE` interrogate packets by sending the current version of all standard BITE status packets that it is configured to output. `Q-BITE` packets are not sent in response to this command.

Table 44 BITE Interrogate Packet (Host to RCP8)

Char	Function
1	SYNC Byte (`C0 Hex`)
2	Command (`0x4D` = Interrogate)
3	END OF MESSAGE (`FF Hex`)

This packet has the same function as the standard `BITE` interrogate packet, except that only the quantitative `BITE` units report back.

Table 45 Q-BITE Interrogate Packet (Host to RCP8)

Char	Function
1	SYNC Byte (`90 Hex`)
2	Command (`0x01` = Interrogate)
3	END OF MESSAGE (`FF Hex`)

The `BITE` individual request packet is used to request information about a single `BITE` unit, separate from the others.

RCP8 responds to an interrogate packet by sending the current version of the specified `BITE` status packet. RCP8 responds to a sample data packet by sending requests the the remote device to get information, then responding to the host computer with the new `BITE` status packet when the information arrives. RCP8 responds to the reset packet by sending a reset command to the remote device.

Table 46 BITE Individual Request Packet (Host to RCP8)

Char	Function
1	SYNC Byte (`C1 Hex`)
2	ID of the BITE unit for which the command is applied
3	Command: `0x4D`=Interrogate, `0x44`=Sample Data, `0x43`=Reset
4	END OF MESSAGE (`FF Hex`)

A.5.7. Klystron BITE Packets

Table 47 Klystron BITE Packet (RCP8 to Host)

Char	Function	Status bit
1	SYNC (C0 Hex)	
2	Identification byte (User choice)	
3	Klystron status and fault conditions (Machine state)	
	D0 = Power On	`S134`
	D1 = Standby	`S135`
	D2 = Radiate	`S136`
	D3 = Fault Sum	`S137`
	D4 = Fault Latch	`S138`
	D5 = Reserved	`S139`
	D6 = Reserved	`S140`
4	Klystron status and fault conditions (Faults)	
	D0 = HVPS Over Temp	`S141`
	D1 = Oil Level	`S142`
	D2 = Solenoid PS	`S143`
	D3 = HVPS Current	`S144`
	D4 = LVPS	`S145`
	D5 = HVPS (over cur/und volt)	`S146`
	D6 = Filament PS	`S147`

Char	Function	Status bit
5	Klystron status and fault conditions (Faults)	
	D0 = Ion Current	S148
	D1 = Klystron Current	S149
	D2 = Solenoid Current	S150
	D3 = Over Duty	S151
	D4 = Mod Over Temp	S152
	D5 = Disable radiate	S153
	D6 = Spare	S154
6	Klystron status and fault conditions (Spare)	
	D0 = Klystron Connected	S155
	D1 = IGBT Fault #1	S156
	D2 = IGBT Fault #2	S157
	D3 = IGBT Fault #3	S158
	D4 = IGBT Fault #4	S159
	D5 = IGBT Fault #5	S160
	D6 = IGBT Fault #6	S161
7	End of Message (FF Hex)	

Table 48 Klystron Q-BITE Packet (RCP8 to Host)

Char	Function	Klystron_status_c field
1	SYNC (AF Hex)	
2	Identification byte (User choice)	
3-4	Solenoid current	`fSolenoidCurrent` (scale 100)
5-6	Klystron current	`fKlystronCurrent` (scale 10)
7-8	Ion current	`fIonCurrent` (scale 100)
9-10	Filament voltage	`fFilamentVoltage` (scale 100)
11-12	Filament current	`fFilamentCurrent` (scale 100)

Char	Function	Klystron_status_c field
13-14	Modulator current	`fModulatorCurrent` (scale 100)
15-16	HVPS voltage	`iHVPSVoltage` (scale 1000)
17-18	Peak cathode voltage	`fPeakCathodeVoltage` (scale 100)
19-20	Peak cathode current	`fPeakCathodeCurrent` (scale 1)
21	End of Message (FF Hex)	

A.5.8. MELCO Packets

Table 49 MELCO TKY01 QBITE Packet (RCP8 to Host)

Char	Function
1	`SYNC` Byte (`AF Hex`)
2	Identification byte (User Choice)
3-4	Generator running hours
7-8	Generator fuel level
9-10	Generator voltage V1
11-12	Generator voltage V2
13-14	Generator voltage V3
11	END OF MESSAGE (`FF Hex`)

The serial format sends this information in 3 separate packets, each sending 4 bytes of payload, using a similar format. The Identification byte is set to 0, 1, or 2.

Table 50 MELCO Serial Packet (Generator to RCP8)

Char	Function
1	SYNC Byte (`AF Hex`)
2	Identification byte
3-6	Data
11	END OF MESSAGE (`FF Hex`)

A.5.9. Power Monitor BITE Packets

Table 51 Power Monitor BITE Packet (RCP8 to Host)

CharstPowerResults_c	Function	Status bit
1	SYNC (C0 Hex)	
2	Identification byte (User choice)	
3 `lAlarmH` (bit 0) `lAlarmV` (bit 1) `lAlarmSum` (bit 2) `lZeroing` (bit 3)	Power Monitor status/alarm bits	
	D0 = Horizontal Return Loss Alarm	offset+0
	D1 = Vertical Return Loss Alarm	offset+1
	D2 = Power Monitor Alarm sum	offset+2
	D3 = Sensor zeroing in progress	offset+3
	D4 = Reserved	
	D5 = Reserved	
	D6 = Reserved	
4	End of Message (FF Hex)	

Table 52 Power Monitor Q-BITE Packet (RCP8 to Host)

Char	Function	Scale
1	SYNC (AF Hex)	--
2	Identification byte (User choice)	--
3-4	Power measurement HOR / TX	Scale 10
5-6	Power measurement HOR / RX	Scale 10
7-8	Power measurement VER / TX	Scale 10
9-10	Power measurement VER / RX	Scale 10
11-12	Return Loss HOR	Scale 10
13-14	Return Loss VER	Scale 10
15-16	TX Peak Power HOR	Scale 0.01
17-18	TX Peak Power VER	Scale 0.01

Char	Function	Scale
19	End of Message (FF Hex)	

A.5.10. Q-BITE Status Packet

The `Q-BITE` data stream consists of a series of integer values. Each value is packed in a series of 7-bit characters, using 1 ... 5 depending on the desired resolution. The low bits come first, and IRIS supports up to 32 bits per value. IRIS can be configured to display any such values with appropriate units and scaling.

The `Q-BITE` (Quantitative `BITE`) status packets consist of 3 ... 128 bytes. The first two and last bytes are used for identification purposes. The middle bytes must have the MSB set to 0 and can contain an arbitrary value in the lower 7 bits. Typically this is used to report voltage/power levels. This report should not be sent by the `BITE` every time the status changes. This report is sent in response to the `Q-BITE` interrogate command. IRIS sends the interrogate command every 60 seconds.

Table 53 Q-BITE Status Packet (Both ways)

Char	Function
1	`SYNC` Byte (`AF Hex`)
2	BITE Unit ID byte (selectable in the range `00-7F Hex`)
3	Status byte #1
4	Status byte #2
.	
.	
N-1	Status byte #N-3
N	END OF MESSAGE (FF Hex)

A.5.11. TSC TWT Packets

There are 23 TSC TWT status bits output in the BITE packet, as follows. For a detailed description of the bit meanings contact Vaisala.

Table 54 TSC TWT BITE Packet (RCP8 to Host)

Char	Function
1	SYNC Byte (C0 Hex)

Char	Function
2	Identification byte (User Choice)
3	Status Bits 6 5 4 3 2 1 0
4	Status Bits 13 12 11 10 9 8 7
5	Status Bits 20 19 18 17 16 15 14
6	Status Bits 23 22 21
7	Spare byte
8	END OF MESSAGE (FF Hex)

There are 4 TSC TWT qualitative values output in the QBITE packet, as follows:

Table 55 TSC TWT QBITE Packet (RCP8 to Host)

Char	Function
1	SYNC Byte (AF Hex)
2	Identification byte (User Choice)
3-4	Frequency code (actually only 6 bits)
5-6	Receiver protector leakage measurement
7-8	Transmitter power measurement
9-10	Reflected power measurement
11	END OF MESSAGE (FF Hex)

A.5.12. WSR-88D DAU BITE Packets

RCP8 generates this `BITE` packet when the WSR-88D DAU pedestal interface has been enabled. The identification byte is selectable to avoid conflicts with other `BITE` packets. The `S` number after each table entry is the numbered status variable that is driven by the respective bit. Most bits in the `BITE` packet are copies of their DAU counterparts (with their original word and bit numbers shown in parenthesis). However, `S232` ... `S245` are supplied by RCP8.

Table 56 WSR-88D DAU BITE Packet (RCP8 to Host) When Connected via Serial Link

Char	Function
1	SYNC Byte (C0 Hex)
2	Identification byte (User Choice)
3	WSR-88D DAU Status and Fault Conditions
	D6 = (0/6) Maintenance Work Required (S126)
	D5 = (0/5) Maintenance Model/No Control (S125)
	D4 = (0/4) W/G PFN Transfer Interlock (S124)
	D3 = (0/3) W/G Switch Dummy Load (S123)
	D2 = (0/2) Transmitter Not Available (S122)
	D1 = (0/1) Klystron Preheat (S121)
	D0 = (0/0) Filament PS Off (S120)
4	WSR-88D DAU Status and Fault Conditions
	D6 = (1/5) Filament PS Voltage (S133)
	D5 = (1/4) +45 VDC PS Summary Fault (S132)
	D4 = (1/3) -15 VDC PS Summary Fault (S131)
	D3 = (1/2) +28 VDC PS Summary Fault (S130)
	D2 = (1/1) +15 VDC PS Summary Fault (S129)
	D1 = (1/0) +5 VDC PS Summary Fault (S128)
	D0 = (0/7) PFN Switch Long Pulse (S127)
5	WSR-88D DAU Status and Fault Conditions
	D6 = (2/4) Cabinet Air Temperature (S140)
	D5 = (2/3) Cabinet Interlock (S139)
	D4 = (2/2) W/G Arc/VSWR (summary) (S138)
	D3 = (2/1) Spectrum Filter Low Pressure (S137)
	D2 = (2/0) Circulator Over-Temperature (S136)
	D1 = (1/7) Focus Coil PS Voltage (S135)
	D0 = (1/6) Vacuum Pump PS Voltage (S134)

Char	Function
6	WSR-88D DAU Status and Fault Conditions
	D6 = (3/3) Main Power Overvoltage (S147)
	D5 = (3/2) Modulator Switch Failure (S146)
	D4 = (3/1) Modulator Inverter Current (S145)
	D3 = (3/0) Modulator Overload (S144)
	D2 = (2/7) Transmitter Spare (S143)
	D1 = (2/6) Transmitter Spare (S142)
	D0 = (2/5) Cabinet Airflow (S141)
7	WSR-88D DAU Status and Fault Conditions
	D6 = (4/2) Focus Coil Current (S154)
	D5 = (4/1) Transmitter Overcurrent (S153)
	D4 = (4/0) Transmitter Overvoltage (S152)
	D3 = (3/7) Transmitter Spare (S151)
	D2 = (3/6) Trigger Amplifier Failure (S150)
	D1 = (3/5) Inverse Diode Current/Undervoltage (S149)
	D0 = (3/4) Flyback Charge Failure (S148)
8	WSR-88D DAU Status and Fault Conditions
	D6 = (5/1) Klystron Filament Current (S161)
	D5 = (5/0) Klystron Overcurrent (S160)
	D4 = (4/7) Battery Charging (S159)
	D3 = (4/6) Oil Level (Transmitter) (S158)
	D2 = (4/5) PRF Limit (Summary) (S157)
	D1 = (4/4) Oil Temperature (Transmitter) (S156)
	D0 = (4/3) Focus Coil Airflow (S155)

Char	Function
9	WSR-88D DAU Status and Fault Conditions
	D6 = (6/0) 'One' Test Bit 0 (S168)
	D5 = (5/7) 'One' Test Bit 7 (S167)
	D4 = (5/6) 'One' Test Bit 6 (S166)
	D3 = (5/5) 'One' Test Bit 5 (S165)
	D2 = (5/4) Klystron Airflow (S164)
	D1 = (5/3) Klystron Air Temperature (S163)
	D0 = (5/2) Klystron Vacion Current (S162)
10	WSR-88D DAU Status and Fault Conditions
	D6 = (6/7) W/G, Pressure/Humidity (S175)
	D5 = (6/6) Post-Charge Regulator Maintenance (S174)
	D4 = (6/5) Modulator Switch Maintenance (S173)
	D3 = (6/4) 'One' Test Bit 4 (S172)
	D2 = (6/3) 'One' Test Bit 3 (S171)
	D1 = (6/2) 'One' Test Bit 2 (S170)
	D0 = (6/1) 'One' Test Bit 1 (S169)
11	WSR-88D DAU Status and Fault Conditions
	D6 = (7/6) 'Zero' Test Bit 6 (S182)
	D5 = (7/5) 'Zero' Test Bit 5 (S181)
	D4 = (7/4) 'Zero' Test Bit 4 (S180)
	D3 = (7/3) 'Zero' Test Bit 3 (S179)
	D2 = (7/2) 'Zero' Test Bit 2 (S178)
	D1 = (7/1) 'Zero' Test Bit 1 (S177)
	D0 = (7/0) 'Zero' Test Bit 0 (S176)

Char	Function
12	WSR-88D DAU Status and Fault Conditions
	D6 = (8/5) Spare (S189)
	D5 = (8/4) UART Error (S188)
	D4 = (8/3) COHO/Clock (S187)
	D3 = (8/2) Transmitter Inoperable (S186)
	D2 = (8/1) Transmitter Recycle (S185)
	D1 = (8/0) HV Off (S184)
	D0 = (7/7) 'Zero' Test Bit 7 (S183)
13	WSR-88D DAU Status and Fault Conditions
	D6 = (9/4) Batter Voltage Low (S196)
	D5 = (9/3) Auto-Transfer SW on Utility Power (S195)
	D4 = (9/2) Generator Maintenance Required (S194)
	D3 = (9/1) AC Unit 2 Compressor Shut Off (S193)
	D2 = (9/0) AC Unit 1 Compressor Shut Off (S192)
	D1 = (8/7) Spare (S191)
	D0 = (8/6) Spare (S190)
14	WSR-88D DAU Status and Fault Conditions
	D6 = (10/3) Equip Shelter Halon/Detect Sys Troub (S203)
	D5 = (10/2) Aircraft Hazard Light Failure (S202)
	D4 = (10/1) Generator Volt and Freq Available (S201)
	D3 = (10/0) Generator Selector SW Not Auto (S200)
	D2 = (9/7) TPS (Reserved) (S199)
	D1 = (9/6) TPS (S198)
	D0 = (9/5) Generator Engine Malfunction (S197)

Char	Function
15	WSR-88D DAU Status and Fault Conditions
	D6 = (11/2) Utility Voltage and Frequency Avail (S210)
	D5 = (11/1) Gen Shelter Halon/Detect Sys Trbl (S209)
	D4 = (11/0) Fire/Smoke in Generator Shelter (S208)
	D3 = (10/7) -9 V Receiver PS Summary Fault (S207)
	D2 = (10/6) +/-18 V Receiver PS Summary Fault (S206)
	D1 = (10/5) Fire/Smoke in Equipment Shelter (S205)
	D0 = (10/4) +5 V Receiver PS Summary Fault (S204)
16	WSR-88D DAU Status and Fault Conditions
	D6 = (12/1) Security System Equipment Trouble (S217)
	D5 = (12/0) Security Syst Unauthor Entry Alarm (S216)
	D4 = (11/7) -5.2 V A/D Converter PS Summ Fault (S215)
	D3 = (11/6) Spare (S214)
	D2 = (11/5) +5 V A/D Converter PS Sum. Fault (S213)
	D1 = (11/4) +/-15 V A/D Convert PS Sum. Fault (S212)
	D0 = (11/3) +9 V Receiver PS Summary Fault (S211)
17	WSR-88D DAU Status and Fault Conditions
	D6 = (13/0) AC Unit 1 Filter Dirty (S224)
	D5 = (12/7) Radome Access Hatch Open (S223)
	D4 = (12/6) Receiver Not Conn. to Antenna (S222)
	D3 = (12/5) Spare (S221)
	D2 = (12/4) Spare (S220)
	D1 = (12/3) +5 V Receivr Protect PS Sum. Fault (S219)
	D0 = (12/2) Security System Disabled (S218)

Char	Function
18	WSR-88D DAU Status and Fault Conditions
	D6 = (13/7) Spare (S231)
	D5 = (13/6) Spare (S230)
	D4 = (13/5) Spare (S229)
	D3 = (13/4) AC Unit 4 Filter Dirty (S228)
	D2 = (13/3) AC Unit 3 Filter Dirty (S227)
	D1 = (13/2) Transmitter Filter Dirty (S226)
	D0 = (13/1) AC Unit 2 Filter Dirty (S225)
19	WSR-88D DAU Status and Fault Conditions
	D6 = Spare (S238)
	D5 = Spare (S237)
	D4 = Spare (S236)
	D3 = Spare (S235)
	D2 = Spare (S234)
	D1 = Spare (S233)
	D0 = Spare (S232)
20	WSR-88D DAU Status and Fault Conditions
	D6 = No DAU reply to last command (S245)
	D5 = Spare (S244)
	D4 = Spare (S243)
	D3 = Spare (S242)
	D2 = Spare (S241)
	D1 = Spare (S240)
	D0 = Spare (S239)
21	END OF MESSAGE (FF Hex)

Table 57 WSR-88D DAU BITE Packet (RCP8 to Host) When Connected via RCP9 ORDA Network Interface Panel

Char	Function
1	SYNC Byte (C0 Hex)
2	Identification byte (User Choice)
3	WSR-88D DAU Status and Fault Conditions
	D6 = (0/6) Maintenance Work Required (S126)
	D5 = (0/5) Maintenance Model/No Control (S125)
	D4 = (0/4) W/G PFN Transfer Interlock (S124)
	D3 = (0/3) W/G Switch Dummy Load (S123)
	D2 = (0/2) Transmitter Not Available (S122)
	D1 = (0/1) Klystron Preheat (S121)
	D0 = (0/0) Filament PS Off (S120)
4	WSR-88D DAU Status and Fault Conditions
	D6 = (1/5) Filament PS Voltage (S133)
	D5 = (1/4) +45 VDC PS Summary Fault (S132)
	D4 = (1/3) -15 VDC PS Summary Fault (S131)
	D3 = (1/2) +28 VDC PS Summary Fault (S130)
	D2 = (1/1) +15 VDC PS Summary Fault (S129)
	D1 = (1/0) +5 VDC PS Summary Fault (S128)
	D0 = (0/7) PFN Switch Long Pulse (S127)
5	WSR-88D DAU Status and Fault Conditions
	D6 = (2/4) Cabinet Air Temperature (S140)
	D5 = (2/3) Cabinet Interlock (S139)
	D4 = (2/2) W/G Arc/VSWR (summary) (S138)
	D3 = (2/1) Spectrum Filter Low Pressure (S137)
	D2 = (2/0) Circulator Over-Temperature (S136)
	D1 = (1/7) Focus Coil PS Voltage (S135)
	D0 = (1/6) Vacuum Pump PS Voltage (S134)

Char	Function
6	WSR-88D DAU Status and Fault Conditions
	D6 = (3/3) Main Power Overvoltage (S147)
	D5 = (3/2) Modulator Switch Failure (S146)
	D4 = (3/1) Modulator Inverter Current (S145)
	D3 = (3/0) Modulator Overload (S144)
	D2 = (2/7) Transmitter Spare (S143)
	D1 = (2/6) Transmitter Spare (S142)
	D0 = (2/5) Cabinet Airflow (S141)
7	WSR-88D DAU Status and Fault Conditions
	D6 = (4/2) Focus Coil Current (S154)
	D5 = (4/1) Transmitter Overcurrent (S153)
	D4 = (4/0) Transmitter Overvoltage (S152)
	D3 = (3/7) Transmitter Spare (S151)
	D2 = (3/6) Trigger Amplifier Failure (S150)
	D1 = (3/5) Inverse Diode Current/Undervoltage (S149)
	D0 = (3/4) Flyback Charge Failure (S148)
8	WSR-88D DAU Status and Fault Conditions
	D6 = (5/1) Klystron Filament Current (S161)
	D5 = (5/0) Klystron Overcurrent (S160)
	D4 = (4/7) Battery Charging (S159)
	D3 = (4/6) Oil Level (Transmitter) (S158)
	D2 = (4/5) PRF Limit (Summary) (S157)
	D1 = (4/4) Oil Temperature (Transmitter) (S156)
	D0 = (4/3) Focus Coil Airflow (S155)

Char	Function
9	WSR-88D DAU Status and Fault Conditions
	D6 = (6/0) 'One' Test Bit 0 (S168)
	D5 = (5/7) 'One' Test Bit 7 (S167)
	D4 = (5/6) 'One' Test Bit 6 (S166)
	D3 = (5/5) 'One' Test Bit 5 (S165)
	D2 = (5/4) Klystron Airflow (S164)
	D1 = (5/3) Klystron Air Temperature (S163)
	D0 = (5/2) Klystron Vacion Current (S162)
10	WSR-88D DAU Status and Fault Conditions
	D6 = (6/7) W/G, Pressure (S175)
	D5 = (6/6) Post-Charge Regulator Maintenance (S174)
	D4 = (6/5) Modulator Switch Maintenance (S173)
	D3 = (6/4) 'One' Test Bit 4 (S172)
	D2 = (6/3) 'One' Test Bit 3 (S171)
	D1 = (6/2) 'One' Test Bit 2 (S170)
	D0 = (6/1) 'One' Test Bit 1 (S169)
11	WSR-88D DAU Status and Fault Conditions
	D6 = (7/6) 'Zero' Test Bit 6 (S182)
	D5 = (7/5) 'Zero' Test Bit 5 (S181)
	D4 = (7/4) 'Zero' Test Bit 4 (S180)
	D3 = (7/3) 'Zero' Test Bit 3 (S179)
	D2 = (7/2) 'Zero' Test Bit 2 (S178)
	D1 = (7/1) 'Zero' Test Bit 1 (S177)
	D0 = (7/0) 'Zero' Test Bit 0 (S176)

Char	Function
12	WSR-88D DAU Status and Fault Conditions
	D6 = (8/5) Waveguide Switch Position (S189)
	D5 = (8/4) Spare (S188)
	D4 = (8/3) COHO/Clock (S187)
	D3 = (8/2) Transmitter Inoperable (S186)
	D2 = (8/1) Transmitter Recycle (S185)
	D1 = (8/0) HV Off (S184)
	D0 = (7/7) 'Zero' Test Bit 7 (S183)
13	WSR-88D DAU Status and Fault Conditions
	D6 = (9/4) Batter Voltage Low (S196)
	D5 = (9/3) Auto-Transfer SW on Utility Power (S195)
	D4 = (9/2) Generator Maintenance Required (S194)
	D3 = (9/1) AC Unit 2 Compressor Shut Off (S193)
	D2 = (9/0) AC Unit 1 Compressor Shut Off (S192)
	D1 = (8/7) Spare (S191)
	D0 = (8/6) Circulator Temp (S190)
14	WSR-88D DAU Status and Fault Conditions
	D6 = (10/3) Equip Shelter Halon/Detect Sys Troub (S203)
	D5 = (10/2) Aircraft Hazard Light Failure (S202)
	D4 = (10/1) Generator Volt and Freq Available (S201)
	D3 = (10/0) Generator Selector SW Not Auto (S200)
	D2 = (9/7) TPS (Reserved) (S199)
	D1 = (9/6) TPS (S198)
	D0 = (9/5) Generator Engine Malfunction (S197)

Char	Function
15	WSR-88D DAU Status and Fault Conditions
	D6 = (11/2) Utility Voltage and Frequency Avail (S210)
	D5 = (11/1) Gen Shelter Halon/Detect Sys Trbl (S209)
	D4 = (11/0) Fire/Smoke in Generator Shelter (S208)
	D3 = (10/7) -9 V Receiver PS Summary Fault (S207)
	D2 = (10/6) +/-18 V Receiver PS Summary Fault (S206)
	D1 = (10/5) Fire/Smoke in Equipment Shelter (S205)
	D0 = (10/4) +5 V Receiver PS Summary Fault (S204)
16	WSR-88D DAU Status and Fault Conditions
	D6 = (12/1) Security System Equipment Trouble (S217)
	D5 = (12/0) Security Syst Unauthor Entry Alarm (S216)
	D4 = (11/7) Spare (S215)
	D3 = (11/6) Gen Shelter Door Open (S214)
	D2 = (11/5) Spare (S213)
	D1 = (11/4) Spare (S212)
	D0 = (11/3) +9 V Receiver PS Summary Fault (S211)
17	WSR-88D DAU Status and Fault Conditions
	D6 = (13/0) AC Unit 1 Filter Dirty (S224)
	D5 = (12/7) Radome Access Hatch1 Open (S223)
	D4 = (12/6) Receiver Not Conn. to Antenna (S222)
	D3 = (12/5) Waveguide Humidity (S221)
	D2 = (12/4) Waveguide Switch TX Interlock (S220)
	D1 = (12/3) +5 V Receivr Protect PS Sum. Fault (S219)
	D0 = (12/2) Security System Disabled (S218)

Char	Function
18	WSR-88D DAU Status and Fault Conditions
	D6 = (13/7) Radome Access Hatch2 Open (S231)
	D5 = (13/6) Waveguide Switch Ant Position Ind (S230)
	D4 = (13/5) Spectrum Filter Pressure (S229)
	D3 = (13/4) Spare (S228)
	D2 = (13/3) Spare (S227)
	D1 = (13/2) Transmitter Filter Dirty (S226)
	D0 = (13/1) AC Unit 2 Filter Dirty (S225)
19	WSR-88D DAU Status and Fault Conditions
	D6 = Pedestal Power (S238)
	D5 = Gen Sw to Utility (S237)
	D4 = Gen Sw To Gen (S236)
	D3 = Re Rx Channel12 (S235)
	D2 = Wg Sw Antenna (S234)
	D1 = Tx High Voltage On (S233)
	D0 = Rx Trans Power Head Adjust (S232)
20	WSR-88D DAU Status and Fault Conditions
	D6 = No DAU reply to last command (S245)
	D5 = Spare (S244)
	D4 = Spare (S243)
	D3 = Spare (S242)
	D2 = Spare (S241)
	D1 = Panel Power Sleep (S240)
	D0 = DAQ Power (S239)
21	END OF MESSAGE (FF Hex)

Many of the DAU functions can also be controlled from RCP8 using numbered control variables as follows. When DAU mode is enabled, a new DAU `Data Message` is sent every second, and immediately after a control `BITE` packet is received.

Table 58 DAU Functions

DAU Function	Code
Audible Alarm Control-1	(C63)
Audible Alarm Control-2	(C62)
High Voltage ON Command	(C61)
Antenna Command	(C60)
Channel 2 Command	(C59)
Pedestal Operate	(C58)
Spare Lamp Driver	(C57)
Switch to Diesel Generator	(C56)
Switch to Utility Power	(C55)
Audible Alarm Enable	(C54)

In addition to the standard `BITE` packets, RCP8 outputs the following `Q-BITE` packets that represent the quantitative values that are read from the DAU. The numbers shown in parenthesis are the original DAU status byte numbers that supplied each value. 30 14-bit values are sent.

Table 59 WSR-88D DAU Q-BITE Packet (RCP8 to Host)

Char	Function
1	SYNC Byte (AF Hex)
2	Identification byte (User Choice)
3-4	(14) Outside Ambient Temperature
5-6	(15) Equipment Shelter Temperature
7-8	(16) AC Unit 1 Discharge Air Temperature
9-10	(17) Transmitter Discharge Air Temperature
11-12	(18) Radome Area Temperature
13-14	(19) Generator Shelter Temperature
15-16	(20) Storage Tank Fuel Level
17-18	(27) AC Unit 3 Discharge Air Temperature

Char	Function
19-20	(28) AC Unit 2 Discharge Air Temperature
21-22	(31) Transmitter RF Power
23-24	(32) Antenna RF Power
25-26	(34) AC Unit 4 Discharge Air Temperature
27-28	(37) Pedestal +28 V Power
29-30	(38) Encoder +5 V Power
31-32	(39) Pedestal +15 V Power
33-34	(41) Pedestal +5 V Power
35-36	(44) Signal Processor +5 V Power
37-38	(46) Maintenance Console +28 V Power
39-40	(47) Maintenance Console +15 V Power
41-42	(48) Maintenance Console +5 V Power
43-44	(52) Pedestal -15 V Power
45-46	(55) Maintenance Console -15 V Power
47-48	(57) DAU Test 0
49-50	(58) DAU Test 1
51-52	(59) DAU Test 2
53-54	Spare
55-56	Spare
57-58	Spare
59-60	Spare
61-62	Spare
63	END OF MESSAGE (FF Hex)

A.5.13. WSR-88D DCU BITE Packets

RCP8 generates this `BITE` packet when the WSR-88D DCU pedestal interface has been enabled. The identification byte is selectable to avoid conflicts with other `BITE` packets. The `S` number after each table entry is the numbered status variable that is driven by the respective bit. Most bits in the `BITE` packet are copies of their DCU counterparts (with their original word and bit numbers shown in parenthesis). However, `S110` ... `S119` are supplied by RCP8.

Table 60 WSR-88D DCU BITE Packet (RCP8 to Host) When Connected via Serial Link

Char	Function
1	SYNC Byte (C0 Hex)
2	Identification byte (User Choice)
3	WSR-88D DCU Status and Fault Conditions
	D6 = (1/6) Elev Axis Enc Light Source Monitor (S70)
	D5 = (1/5) Spare (S69)
	D4 = (1/4) Elevation Axis minus Normal Limit (S68)
	D3 = (1/3) Elevation Axis plus Normal Limit (S67)
	D2 = (1/2) Spare (S66)
	D1 = (1/1) Elevation Axis Deal Limit (S65)
	D0 = (1/0) Elevation Axis PCU Data Parity (S64)
4	WSR-88D DCU Status and Fault Conditions
	D6 = (1/14) Elevation Axis Motor Over Temp. (S77)
	D5 = (1/13) +150 V Under Voltage (S76)
	D4 = (1/12) +150 V Over Voltage (S75)
	D3 = (1/11) EL Axis Servo Amp Over Temp (S74)
	D2 = (1/10) EL Axis Servo Amp Short Circuit (S73)
	D1 = (1/9) Elevation Axis Servo Amp Inhibit (S72)
	D0 = (1/7) Elevation Axis Gearbox Oil Level (S71)

Char	Function
5	WSR-88D DCU Status and Fault Conditions
	D6 = (2/5) Elevation Handwheel Engaged (S84)
	D5 = (2/4) Spare (S83)
	D4 = (2/3) Azimuth Axis Bull Gear Oil Level (S82)
	D3 = (2/2) Azimuth Axis Gearbox Oil Level (S81)
	D2 = (2/1) Azith Axis Encoder Light Source Mon (S80)
	D1 = (2/0) Azimuth Axis PCU Data Parity (S79)
	D0 = (1/15) Elevation Axis Stow Pin Engaged (S78)
6	WSR-88D DCU Status and Fault Conditions
	D6 = (2/13) Spare (S91)
	D5 = (2/12) Spare (S90)
	D4 = (2/11) AZ Axis Servo Amp Over Temp (S89)
	D3 = (2/10) AZ Axis Servo Amp Short Circuit (S88)
	D2 = (2/9) Azimuth Axis Servo Amp Inhibit (S87)
	D1 = (2/7) Spare (S86)
	D0 = (2/6) Azimuth Handwheel Engaged (S85)
7	WSR-88D DCU Status and Fault Conditions
	D6 = (3/4) Spare (S98)
	D5 = (3/3) Spare (S97)
	D4 = (3/2) Spare (S96)
	D3 = (3/1) Spare (S95)
	D2 = (3/0) Spare (S94)
	D1 = (2/15) Azimuth Axis Stow Pin Engaged (S93)
	D0 = (2/14) Azimuth Axis Motor Over Temp (S92)

Char	Function
8	WSR-88D DCU Status and Fault Conditions
	D6 = (3/12) Azimuth Axis Servo Amp PS (S105)
	D5 = (3/11) Spare (S104)
	D4 = (3/10) Spare (S103)
	D3 = (3/9) Spare (S102)
	D2 = (3/7) Spare (S101)
	D1 = (3/6) Spare (S100)
	D0 = (3/5) Spare (S99)
9	WSR-88D DCU Status and Fault Conditions
	D6 = Spare (S112)
	D5 = Spare (S111)
	D4 = Spare (S110)
	D3 = DCU Timeout (From DCU antenna record) (S109)
	D2 = (3/15) Ped Interlock (S108)
	D1 = (3/14) Servo Off (S107)
	D0 = (3/13) Elevation Axis Servo Amp PS (S106)
10	WSR-88D DCU Status and Fault Conditions
	D6 = No ANT record received for 0.5 seconds (S119)
	D5 = No BIT record received for 2.5 seconds (S118)
	D4 = Spare (S117)
	D3 = Spare (S116)
	D2 = Spare (S115)
	D1 = Spare (S114)
	D0 = Spare (S113)
11	END OF MESSAGE (FF Hex)

Table 61 WSR-88D DCU BITE Packet (RCP8 to Host) When Connected via RCP9 ORDA Network Interface Panel

Char	Function
1	SYNC Byte (C0 Hex)
2	Identification byte (User Choice)
3	WSR-88D DCU Status and Fault Conditions
	D6 = (1/6) Elev Axis Enc Light Source Monitor (S70)
	D5 = (1/5) Elev Axis Final plus Limit (S69)
	D4 = (1/4) Elevation Axis minus Normal Limit (S68)
	D3 = (1/3) Elevation Axis plus Normal Limit (S67)
	D2 = (1/2) Elevation Axis Final minus Limit (S66)
	D1 = (1/1) Spare (S65)
	D0 = (1/0) Spare (S64)
4	WSR-88D DCU Status and Fault Conditions
	D6 = (1/14) Elevation Axis Motor Over Temp. (S77)
	D5 = (1/13) +150 V Under Voltage (S76)
	D4 = (1/12) +150 V Over Voltage (S75)
	D3 = (1/11) EL Axis Servo Amp Over Temp (S74)
	D2 = (1/10) EL Axis Servo Amp Short Circuit (S73)
	D1 = (1/9) Elevation Axis Servo Amp Inhibit (S72)
	D0 = (1/7) Elevation Axis Gearbox Oil Level (S71)
5	WSR-88D DCU Status and Fault Conditions
	D6 = (2/5) Elevation Handwheel Engaged (S84)
	D5 = (2/4) Elevation Housing 5 V status (S83)
	D4 = (2/3) Azimuth Axis Bull Gear Oil Level (S82)
	D3 = (2/2) Azimuth Axis Gearbox Oil Level (S81)
	D2 = (2/1) Azimuth Axis Encoder Light Source Mon (S80)
	D1 = (2/0) Spare (S79)
	D0 = (1/15) Elevation Axis Stow Pin Engaged (S78)

Char	Function
6	WSR-88D DCU Status and Fault Conditions
	D6 = (2/13) Azimuth Housing 5 V status (S91)
	D5 = (2/12) Spare (S90)
	D4 = (2/11) AZ Axis Servo Amp Over Temp (S89)
	D3 = (2/10) AZ Axis Servo Amp Short Circuit (S88)
	D2 = (2/9) Azimuth Axis Servo Amp Inhibit (S87)
	D1 = (2/7) Spare (S86)
	D0 = (2/6) Azimuth Handwheel Engaged (S85)
7	WSR-88D DCU Status and Fault Conditions
	D6 = (3/4) Spare (S98)
	D5 = (3/3) Spare (S97)
	D4 = (3/2) Spare (S96)
	D3 = (3/1) Spare (S95)
	D2 = (3/0) Spare (S94)
	D1 = (2/15) Azimuth Axis Stow Pin Engaged (S93)
	D0 = (2/14) Azimuth Axis Motor Over Temp (S92)
8	WSR-88D DCU Status and Fault Conditions
	D6 = (3/12) Azimuth Axis Servo Amp PS (S105)
	D5 = (3/11) Spare (S104)
	D4 = (3/10) Spare (S103)
	D3 = (3/9) Spare (S102)
	D2 = (3/7) Spare (S101)
	D1 = (3/6) Spare (S100)
	D0 = (3/5) Spare (S99)

Char	Function
9	WSR-88D DCU Status and Fault Conditions
	D6 = Spare (S112)
	D5 = Spare (S111)
	D4 = Spare (S110)
	D3 = Spare (S109)
	D2 = (3/15) Ped Interlock (S108)
	D1 = (3/14) Servo Off (S107)
	D0 = (3/13) Elevation Axis Servo Amp PS (S106)
10	WSR-88D DCU Status and Fault Conditions
	D6 = Spare (S119)
	D5 = Spare (S118)
	D4 = Spare (S117)
	D3 = Spare (S116)
	D2 = Spare (S115)
	D1 = Spare (S114)
	D0 = Spare (S113)
11	END OF MESSAGE (FF Hex)

RCP8 generates this `BITE` packet when the WSR-88D DCU pedestal responds to a **`Self-Test1`** command. Most bits in the `BITE` packet are copies of their DCU counterparts (with their original word and bit numbers shown in parenthesis).

Table 62 WSR-88D DCU Self-Test1 BITE Packet (RCP8 to Host)

Char	Function
1	SYNC Byte (C0 Hex)
2	Identification byte (User Choice)

Char	Function
3	D6 = (1/6) Az Command loopback
	D5 = (1/5) Az Command loopback
	D4 = (1/4) Az Command loopback
	D3 = (1/3) Az Command loopback
	D2 = (1/2) Az Command loopback
	D1 = (1/1) Az Command loopback
	D0 = (1/0) Az Command loopback
4	D6 = (1/13) Az Command loopback
	D5 = (1/12) Az Command loopback
	D4 = (1/11) Az Command loopback
	D3 = (1/10) Az Command loopback
	D2 = (1/9) Az Command loopback
	D1 = (1/8) Az Command loopback
	D0 = (1/7) Az Command loopback
5	D6 = (2/4) El Command loopback
	D5 = (2/3) El Command loopback
	D4 = (2/2) El Command loopback
	D3 = (2/1) El Command loopback
	D2 = (2/0) El Command loopback
	D1 = (1/15) Az Command loopback
	D0 = (1/14) Az Command loopback

Char	Function
6	D6 = (2/11) El Command loopback
	D5 = (2/10) El Command loopback
	D4 = (2/9) El Command loopback
	D3 = (2/8) El Command loopback
	D2 = (2/7) El Command loopback
	D1 = (2/6) El Command loopback
	D0 = (2/5) El Command loopback
7	D6 = Spare
	D5 = Spare
	D4 = Spare
	D3 = (2/15) El Command loopback
	D2 = (2/14) El Command loopback
	D1 = (2/13) El Command loopback
	D0 = (2/12) El Command loopback
8	END OF MESSAGE (FF Hex)

RCP8 generates this **BITE** packet when the WSR-88D DCU pedestal responds to a **Self-Test2** command. Most bits in the **BITE** packet are copies of their DCU counterparts (with their original word and bit numbers shown in parenthesis).

Table 63 WSR-88D DCU Self-Test2 BITE Packet (RCP8 to Host)

Char	Function
1	SYNC Byte (C0 Hex)
2	Identification byte (User Choice)

Char	Function
3	D6 = (1/6) AZ Power Amp
	D5 = (1/5) Spare
	D4 = (1/4) Spare
	D3 = (1/3) Spare
	D2 = (1/2) Spare
	D1 = (1/1) Spare
	D0 = (1/0) Digital PWA
4	D6 = (1/13) Spare
	D5 = (1/12) El Encoder
	D4 = (1/11) AZ Encoder
	D3 = (1/10) El Motor
	D2 = (1/9) AZ Motor
	D1 = (1/8) Analog PWA
	D0 = (1/7) El Power Amp
5	D6 = <unused>
	D5 = <unused>
	D4 = <unused>
	D3 = <unused>
	D2 = <unused>
	D1 = (1/15) Spare
	D0 = (1/14) Spare
6	END OF MESSAGE (FF Hex)

A.6. Miscellaneous Formats

Table 64 Time Packet (RCP8 to Host)

Char	Function
1	SYNC Byte (B0 Hex)
2	Year Low 7 bits
3	Year High 7 bits
4	Month
5	Day
6	Hour
7	Minute
8	Second
9	1/100 of second
10	Status (unused, zero)
11	END OF MESSAGE (FF Hex)

These packets are sent in both directions to convey serial TTY communication. Up to six 7–bit characters can be sent in each packet with two characters of overhead for `SYNC` and `END`. This allows up to 75 % of the available serial bandwidth to be used for chatting. If a chat–mode packet contains fewer than six characters, then a `NULL` (0 byte) is inserted after the last one.

Table 65 Chat–Mode Packet

Char	Function
1	SYNC Byte (F1 Hex)
2 to 7	7–Bit ASCII characters (possibly NULL terminated)
8	END OF MESSAGE (FF Hex)

Appendix B. I/O-62 Connector Panel Pin Properties

The following tables show the pin assignments to the connectors. The tables show the basic electrical properties of each pin and an example signal assignment (if any) can be made in the `softplane.conf` file.

Table 66 J1 AZ INPUT

Pin	Electrical Specification	Example Signal Name
1	TTL	sPedAZ[0]
2	TTL	sPedAZ[1]
3	TTL	sPedAZ[2]
4	TTL	sPedAZ[3]
5	TTL	sPedAZ[4]
6	TTL	sPedAZ[5]
7	TTL	sPedAZ[6]
8	TTL	sPedAZ[7]
9	TTL	sPedAZ[8]
10	TTL	sPedAZ[9]
11	TTL	sPedAZ[10]
12	TTL	sPedAZ[11]
13	TTL	sPedAZ[12]
14	TTL	sPedAZ[13]
15	TTL	sPedAZ[14]
16	TTL	sPedAZ[15]
17	TTL	
18	TTL	
19	TTL	
20	TTL	

Pin	Electrical Specification	Example Signal Name
21	GND	
22	GND	
23	GND	
24	GND	
25	GND	

Table 67 J2 AZ OUTPUT

Pin	Electrical Specification	Example Signal Name
1	TTL	cEarthAZ[0]
2	TTL	cEarthAZ[1]
3	TTL	cEarthAZ[2]
4	TTL	cEarthAZ[3]
5	TTL	cEarthAZ[4]
6	TTL	cEarthAZ[5]
7	TTL	cEarthAZ[6]
8	TTL	cEarthAZ[7]
9	TTL	cEarthAZ[8]
10	TTL	cEarthAZ[9]
11	TTL	cEarthAZ[10]
12	TTL	cEarthAZ[11]
13	TTL	cEarthAZ[12]
14	TTL	cEarthAZ[13]
15	TTL	cEarthAZ[14]
16	TTL	cEarthAZ[15]
17	TTL	
18	TTL	
19	TTL	

Pin	Electrical Specification	Example Signal Name
20	TTL	
21	GND	
22	GND	
23	GND	
24	GND	
25	GND	

Table 68 J3 CONTROL

Pin	Electrical Specification	Example Signal Name
1	Configurable I/O-62 Digital Lines:	`cPWidth[0]`
2		`cRadiateOn`
3		`cServoPwr`
4		`cReset`
5		`sPWidth[0]`
6		`sRadiate`
7		`sServoPwr`
8		`sReset`
9	RS422+	
10	RS422+	
11	GND	
12	GND	
13	GND	

Pin	Electrical Specification	Example Signal Name
14	Configurable I/O-62 Digital Lines:	`cPWidth[1]`
15		`cRadiateOff`
16		`cTransmitPwr`
17		
18		`sPWidth[1]`
19		`sTransmitPwr`
20		
21		
22	RS422-	
23	RS422-	
24	GND	
25	GND	

I/O-62 lines can be configured in *`softplane.conf`* for the following options:

- RS-422 differential (...`lRS422 = 1` in *`softplane.conf`*) or TTL/CMOS singe-ended (...`lRS422 = 0`).
- Input sense (variable starting with `s` in *`softplane.conf`*) or output sense (variable starting with `c`).
- Input termination for the single-ended lines can be pull-up (...`term=1` in *`softplane.conf`*), pull-down (...`term=-1`) or un-terminated (...`term=0`).

Always apply the same configuration for a group of pins. There are 2 groups in J3. The first group consists of pins 1-4 and 14-17. The second group consists of pins 5-8 and 18-21. In RS-422, pins 1-8 are the positive and pins 14-21 the negative wires of the differential pairs.

Table 69 J4 EL INPUT

Pin	Electrical Specification	Example Signal Name
1	TTL	`sPedEL[0]`
2	TTL	`sPedEL[1]`
3	TTL	`sPedEL[2]`
4	TTL	`sPedEL[3]`

Pin	Electrical Specification	Example Signal Name
5	TTL	sPedEL[4]
6	TTL	sPedEL[5]
7	TTL	sPedEL[6]
8	TTL	sPedEL[7]
9	TTL	sPedEL[8]
10	TTL	sPedEL[9]
11	TTL	sPedEL[10]
12	TTL	sPedEL[11]
13	TTL	sPedEL[12]
14	TTL	sPedEL[13]
15	TTL	sPedEL[14]
16	TTL	sPedEL[15]
17	TTL	
18	TTL	
19	TTL	
20	TTL	
21	GND	
22	GND	
23	GND	
24	GND	
25	GND	

Table 70 J5 EL OUTPUT

Pin	Electrical Specification	Example Signal Name
1	TTL	cEarthEL[0]
2	TTL	cEarthEL[1]
3	TTL	cEarthEL[2]

Pin	Electrical Specification	Example Signal Name
4	TTL	cEarthEL[3]
5	TTL	cEarthEL[4]
6	TTL	cEarthEL[5]
7	TTL	cEarthEL[6]
8	TTL	cEarthEL[7]
9	TTL	cEarthEL[8]
10	TTL	cEarthEL[9]
11	TTL	cEarthEL[10]
12	TTL	cEarthEL[11]
13	TTL	cEarthEL[12]
14	TTL	cEarthEL[13]
15	TTL	cEarthEL[14]
16	TTL	cEarthEL[15]
17	TTL	
18	TTL	
19	TTL	
20	TTL	
21	GND	
22	GND	
23	GND	
24	GND	
25	GND	

Table 71 J6 RELAY

Pin	Electrical Specification	Example Signal Name
1	Relay K1: CT	`cPWidth[0]`
2	Relay K1: NO	
3	Relay K1: NC	
4	Relay K2: CT	`cPWidth[1]`
5	Relay K2: NO	
6	Relay K2: NC	
7	Relay K3: CT	
8	Relay K3: NO	
9	Relay K3: NC	
10	-	
11	GND	
12	GND	
13	GND	
14	+12 VDC	`External Relay Control Power`
15	+12 VDC	
16	+12 VDC	
17	+12 VDC	
18	+12 V Unreg	
19	+12 V Return14	`External Control Returns`
20	+12 V Return15	
21	+12 V Return16	
22	+12 V Return17	
23	-	
24	GND	
25	GND	

WARNING! To avoid possible damage to the connector panel, all external relays must be equipped with diode protection against the back EMF generated when the external relay coil is opened. Relays can be purchased with a diode installed or a diode can be added to the relay across the coil supply and return.

Table 72 Dry Contacts Used as Internal relays K1, K2, K3 on the Connector Panel

Contact	Description
CT	Center contact
NO	Normally open contact
NC	Normally closed contact

Table 73 J7 19:0

Pin	Electrical Specification	Example Signal Name
1	TTL	`sAux[0]`
2	TTL	`sAux[1]`
3	TTL	`sAux[2]`
4	TTL	`sAux[3]`
5	TTL	`sAux[4]`
6	TTL	`sAux[5]`
7	TTL	`sAux[6]`
8	TTL	`sAux[7]`
9	TTL	`sAux[8]`
10	TTL	`sAux[9]`
11	TTL	`sAux[10]`
12	TTL	`sAux[11]`
13	TTL	`sAux[12]`
14	TTL	`sAux[13]`
15	TTL	`sAux[14]`
16	TTL	`sAux[15]`

Pin	Electrical Specification	Example Signal Name
17	TTL	`sAux[16]`
18	TTL	`sAux[17]`
19	TTL	`sAux[18]`
20	TTL	`sAux[19]`
21	GND	
22	GND	
23	GND	
24	GND	
25	GND	

Table 74 J8 ANALOG IN

Pin	Electrical Specification	Example Signal Name
1		`Amux0+`
2	±20 VDC	`Amux1+`
3	Differential	`Amux2+`
4	Analog	`Amux3+`
5	Inputs	`Amux4+`
6		`Amux5+`
7	Positive	`Amux6+`
8	Side	`Amux7+`
9		`Amux8+`
10		`Amux9+`
11	GND	
12	GND	
13	GND	
14		`Amux0−`
15	±20 VDC	`Amux1−`

Pin	Electrical Specification	Example Signal Name
16	Differential	Amux2-
17	Analog	Amux3-
18	Inputs	Amux4-
19		Amux5-
20	Negative	Amux6-
21	Side	Amux7-
22		Amux8-
23		Amux9-
24	GND	
25	GND	

Table 75 J9 RVP: MISC I/O ; RCP8: PED/STATUS

Pin	Electrical Specification	Example Signal Name
1	Configurable I/O-62 Digital Lines:	sWavegpFlt
2		sInterlockFlt
3		sLocal
4		sLowerEL
5		
6		
7		
8	±6 to ±70 VDC Input	AzTach+
9	±6 to ±70 VDC Input	ElTach+
10	±10 VDC Output	AzDrive
11	GND	
12	GND	
13	GND	

Pin	Electrical Specification	Example Signal Name
14	Configurable I/O-62 Digital Lines:	`sAirflowFlt`
15		`sMagCurrentFlt`
16		`sStandby`
17		`sUpperEL`
18		
19		
20		
21	±6 to ±70 VDC Input	`AzTach-`
22	±6 to ±70 VDC Input	`ElTach-`
23	±10 VDC Output	`ElDrive`
24	GND	
25	GND	

I/O-62 lines can be configured in *`softplane.conf`* for the following options:

- RS-422 differential (...`lRS422 = 1` in *`softplane.conf`*) or TTL/CMOS singe-ended (...`lRS422 = 0`).
- Input sense (variable starting with `s` in *`softplane.conf`*) or output sense (variable starting with `c`).
- Input termination for the single-ended lines can be pull-up (...`term=1` in *`softplane.conf`*), pull-down (...`term=-1`) or un-terminated (...`term=0`).

Always apply the same configuration for a group of pins. There are 2 groups in J3. The first group consists of pins 1-4 and 14-17. The second group consists of pins 5-8 and 18-21. In RS-422, pins 1-8 are the positive and pins 14-20 the negative wires of the differential pairs.

Table 76 J10 SERIAL

Pin	Electrical Specification	Comment
1	GND	
2	RS232C Rx	
3	RS232C Tx	
4	-	

Pin	Electrical Specification	Comment
5	GND	
6	-	
7	-	
8	-	
9	-	

Table 77 J11 SERIAL

Pin	Electrical Specification	Comment
1	GND	
2	RS232C Rx	Channel 0
3	RS232C Tx	Channel 0
4	RS232C Rx	Channel 1
5	GND	
6	RS232C Tx	Channel 1
7	-12 VDC @ 50mA max regulated	Regulated power supply
8	+12 VDC @ 50 mA max	Regulated power supply
9	+5 VDC @ 50 mA max	Regulated power supply

Table 78 J12 S-D

Pin	Electrical Specification	Example Signal Name
1	Nominal 90 V 60 Hz Synchro Signals	`RefEL+`
2RefEL-		`RefEL-`
3SyEL1		`SyEL1`
4SyEL2		`SyEL2`
5SyEL3		`SyEL3`
6	GND	

Pin	Electrical Specification	Example Signal Name
7	Nominal 90 V 60 Hz Synchro Signals	RefAZ+
8RefAZ-		RefAZ-
9SyAZ1		SyAZ1
10SyAZ2		SyAZ2
11SyAZ3		SyAZ3
12	GND	

The pin numbers are embossed on the J12 plastic connector. Facing the back panel connector the pin arrangement is:

```
1 RefEL+        4 SynEL2        7 RefAZ+        10 SynAZ2
2 RefEL-        5 SynEL3        8 RefAZ-        11 SynAZ3
3 SynEL1        6 Ground        9 SynAZ1        12 Ground
```

The mating plug is AMP 350735-1 using Amplatch pins 350547-1. The corresponding hood comes in 2 identical pieces: AMP 640717-1, along with #6 x 1/2" self-tapping screw. You must use 2 hoods and 2 screws per plug.

The following table lists the maximum RMS voltage that can be applied to the back panel's Molex SYNCHRO connector for each value of plug-in SIP resistor. The AZ channel voltages are set by `SIP S1` and `S2` sets the EL voltage levels. These resistors are socketed, you can change them by removing the back cover of the IO62-CP panel

Table 79 Maximum RMS Voltage

S1 or S2	Max Ref(RMS)	Max S-S(RMS)
47K	56V	31V
68K	81V	45V
100K	118V	66V
150K	178V	99V
220K	261V	145V

Note that the `Ref` inputs have a lower gain than the `S` inputs. This is because the precision of the S/D angle conversion is affected primarily by the precision at which the three `S` voltages can be measured. The back panel biases the gains so that the `S` voltages can be made as large as possible. That is, without the `Ref` voltages first filling the A/D conversion range.

The appropriate resistor is the smallest value such that the maximum `S-to-S` voltage of the synchro (which is angle dependent) still fits within the table range. The reference voltage should then fit easily into its corresponding maximum range. Don't worry if it doesn't; the important thing is to match the `S` line voltages.

For example, a traditional 90 Vrms 1:1 synchro would best use the 150 K resistor, whereas a 105 Vrms unit would require the 220 K value.

To check for proper A/D conversion levels of the synchro inputs, type: **`RCP > help view`**

Table 80 RCP8 BNC Connector Pin Assignments

Ref Designator	Label	Electrical Specification	Signal Name
J13	TP1	5 V 75 Ω	
J14	SPARE		
J15	SPARE		
J16	TP2	5 V 75 Ω	
J17	SPARE		
J18	SPARE		

Appendix C. Operating Dual Systems

C.1. Dual System Applications

A dual-system is the use of two separate transmitters and receivers through a single antenna. There are 2 primary applications for dual (A/B) systems: redundentat and dual frequency systems.

Both applications share common elements with regard to system control, monitoring and data acquisition.

Here, we assume that the signal processor is a Vaisala RVP and that the host computer runs Vaisala IRIS software. It is also possible to use dual-RCP control with other signal processor and software applications.

Redundant Systems

Redundant systems are set up so that if A fails, B provides backup or vice versa.

For redundant systems, the two systems function exclusively, that is, they never operate simultaneously. The 2 systems could be separate radar transmitter/receiver systems sharing the same antenna and/or separate RCP8 systems.

In the case of redundant operation, the systems must be capable of switching automatically when a system fails.

Dual Frequency Systems

Dual frequency systems include two independent transmitters and receivers that share the same antenna.

The systems can be operated either exclusively (that is, one-at-a-time), or simultaneously (in parallel) with one system acting as the master and the other acting as the slave (in passive mode).

In the case of dual frequency operation, the systems must be capable of switching sequentially, as in the redundant case, as well as switching between “active” and “passive” state.

C.2. Dual System Architecture

The dual-system architecture features include:

- Two RCPs that coordinate the system operation via the A/B Comm Link. The RCPs monitor status from the radar, the signal processor, and IRIS, and decide between them which system is active.
- A “Disable Switch” that removes either system A or B. The switch (or 2 separate switches) could also be wired in such a way to make it possible to disable both A and B simultaneously.

- Two IRIS/Radar Systems that are treated as “unreliable” systems. In other words, even in the event of a failure of IRIS, RCP must still make the correct decision about which system is active.
- Two RVPs (radar signal processors) communicating with the IRIS Radar. Note that the 3 functions (RCP, RVP, and IRIS) are normally running on the same computer, but they can be implemented on 2 or 3 computers.

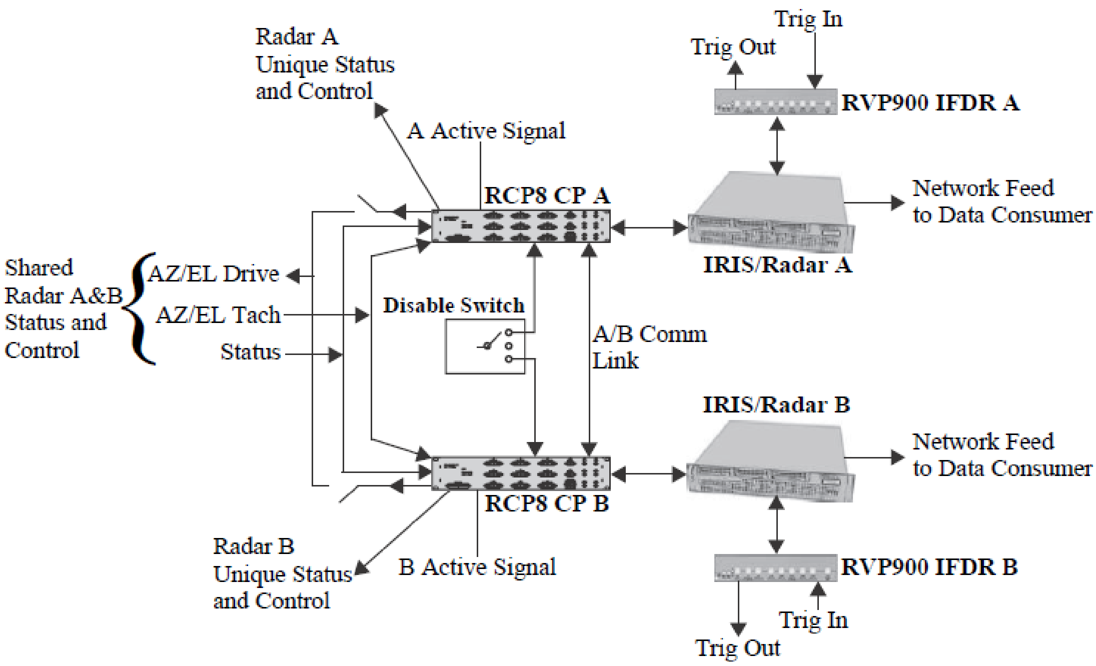

Figure 11 Dual System Architecture

The following table summarizes the signals shown in the figure.

Table 81 Dual System Architecture

Signal	Description
Unique Status and Control	Status and control signals that are unique to each system. Examples are radiate on/off status and control, and transmitter status. These use the normal status/control and extended BITE inputs/outputs.

Signal	Description
Common Status and Control	Status and control signals that are common to both systems. Examples of common status variables are waveguide pressure, safety interlock on the radome door, antenna servo status, and site/environment status. Common status inputs should be wired in parallel to both RCP8s to the normal status and BITE inputs. **CAUTION!** Do not wire common system control outputs in parallel (from both RCPs). This could result in damage to RCP8 output drivers. **CAUTION!** Do not wire critical common system control outputs to both RCPs without using an external relay to select which is used. A "critical" control output is one that could damage the system in the event that both RCPs commanded the control simultaneously. The A/B Active signal is available for controlling external relays so that only one system can control the output. Examples of common control outputs are servo power on/off and any environment control such as obstruction light on/off. Common control functions must be handled differently to prevent the possibility of simultaneous, and perhaps conflicting, control by both RCP. Critical control functions (that is, those that could potentially damage the radar if both RCPs were to command them), should be routed through an external relay that is controlled by the A/B Active Indicator signal (which can of course be used to control a master relay). Non-critical control functions (that is, those that would not damage the radar if both RCPs were to command them), can be routed through any of the spare internal relays in RCP8 (there are 8 total). The approach of using spare internal relays for these common control outputs relies on the RCP8 control logic equations. The internal variable cDrcpActive would be equated to one of the extended BITE control output variables. This would be physically wired to a spare TTL relay on RCP8. The control output would then be wired through the relay (for example, servo power on/off). This approach is not fail-safe, since the user could make an error in the control logics, or simply disable the control logics. Therefore it is not appropriate for critical control functions.
Azimuth and Elevation Drive Output Signals	Routed through an internal relay in RCP8 that connects the drive lines when RCP8 is in the active state. This allows only 1 RCP to control the antenna. The normal drive output back panel connector assignments are used.
Azimuth and Elevation Tachometer Input Signals	Wired in parallel to the tachometer inputs on both RCPs. Tach is sensed by both RCP8s simultaneously, but is not used on the inactive unit. The normal tachometer input back panel connector assignments are used.
Trigger Inhibit	Output line from RCP to the signal processor or trigger generator that can be used to inhibit triggers on the inactive system. In the case of an RVP, the line is the LSB of the normal azimuth output tag line (`AZ0`). No special cabling is required.

Signal	Description
A/B Active Indicator Output Line	This is an active low TTL signal that is output to indicate that an RCP8 is in active mode. This signal should be used directly to switch (via external relay) critical control functions, that is, those functions that, if operated simultaneously by both RCP8s could cause damage to the system. The internal logic variable name for this indicator is `cDrcpActive`. This signal can also be configured as a RS422 differential signal.
A/B Disable Switch Input	A switch closure to ground on this input disables a system so that it is not available for automatic switching. This is used, for example, to put a system in “Maintenance Mode”. The input can be implemented as a single, three position selector switch as shown in the figure, or as two separate switches. In the case of a single selector switch, the switch can be labeled as an enable rather than a disable (for example, `A-Auto-B`) to indicate the exclusive use of either `A` or `B`.
A/B Communications Link	This link is on a special cable between 2 RCP8s. The link implements a serial protocol that passes status information and requests for control between the 2 RCPs. For information on pin assignments, see C.3. Dual System Cabling and Modifications (page 186).

C.3. Dual System Cabling and Modifications

Dual System Connection Cable and Switch

All cabling requirements for the dual-system are handled on connector J3, a 25DBF connector on the back panel of RCP8 labelled **CONTROL**. This connector contains the input for the A/B Disable Switch, the output for the A/B Active Indicator signal and the A/B Communications link. The wiring is shown below. For signal information, see C.2. Dual System Architecture (page 183).

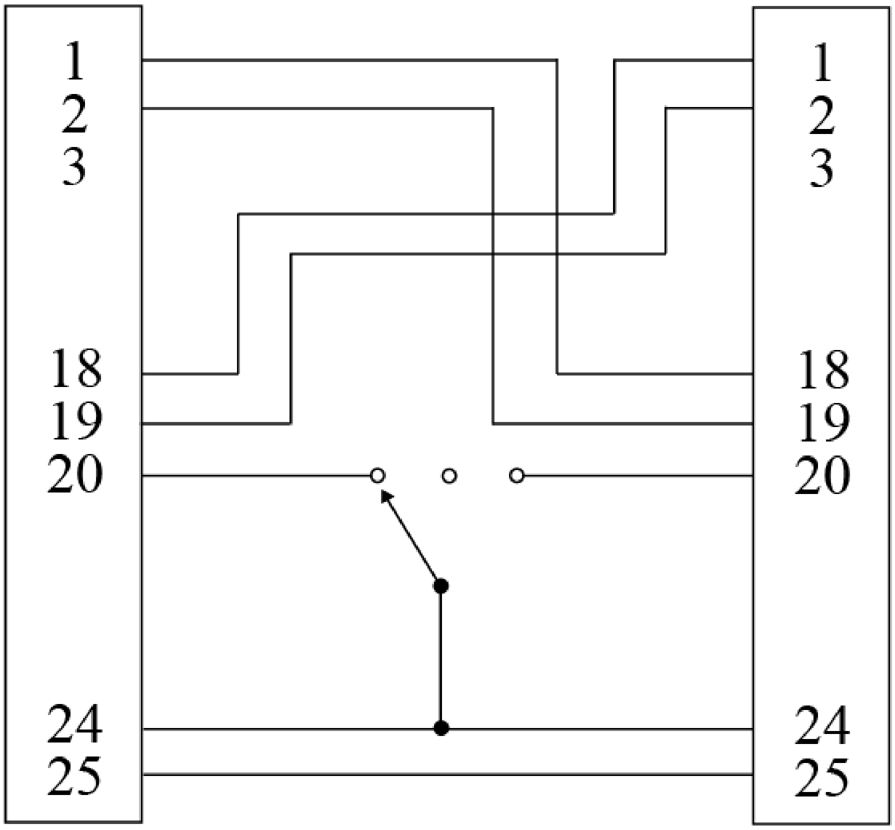

Figure 12 Connector J3 CONTROL Dual-System Link Cable

1 cDropComm[0]
2 cDropComm[1]
3 cDropActive (A or B is ActiveSignal)
18 sDropComm[0]
19 sDropComm[1]
20 sDropEnable
24 GND
25 GND

Antenna Drive/Internal Relay Wiring

The cabling to route the output drive signals through the internal relays is shown in the following figure.

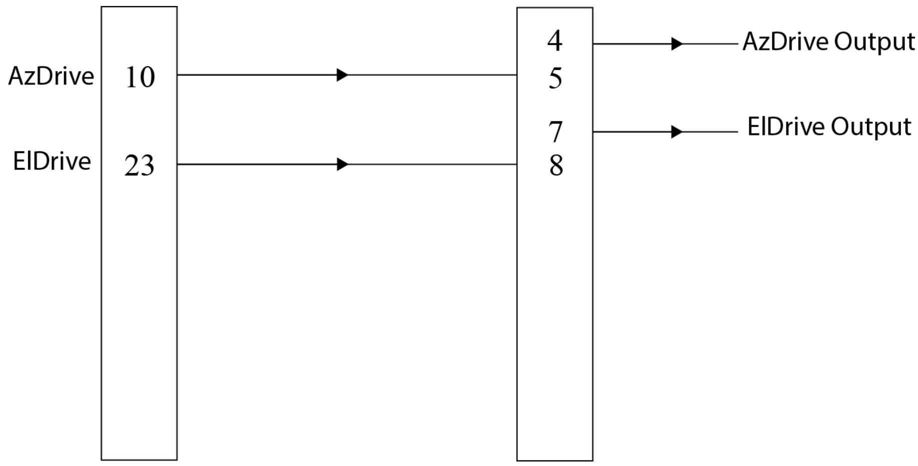

Figure 13 Cabling between Output Drive Signals and Internal Relays

You must modify the *softplane.conf* file to control the internal relays as follows:

- *splConfig.Io62[0].Opt.Cp.J6_IntRelay2 = "cDrcpActive"*
- *splConfig.Io62[0].Opt.Cp.J6_IntRelay3 = "cDrcpActive"*

C.4. RCP TTY Site Settings for Dual System

The dual-system setups are in the **site custom** section of the RCP8 non-volatile setups. You can access these from a setup terminal or from IRIS systems through the `awtx` program from an X terminal.

1. **Use Dual/Redundant system configuration:** `YES`
   Answer `YES` to enable dual-system support.
   Answer `NO` to disable dual-system support.

> If you answer **NO** to this question, the safety features that prevent simultaneous usage of the 2 RCP8s are not in effect. Vaisala does not recommend that you answer **NO** for a dual-system.

2. **This RCP8 is the 'A' unit:** `NO`
   The 2 RCP8s are named the `A` and `B` units.

> To avoid confusion, Vaisala recommends that you put physical **A** and **B** labels on the 2 units.

   If you do not want to use the **A** and **B** labels, use this question to declare the single-letter identifier that appears in the TB title bar for each unit.
   Note that if you use the `TB` (title bar) option for the first line of the RCP8 front panel display (configured in the site display section), then the characters `[A]` or `[B]` appear on the top line of the RCP8 front panel to reflect your choice.

3. **Identifying letter for this unit:** `'A'`
   Respond `YES` for your `A` unit and then for the `B` unit setups respond `NO`.

4. **Default powerup operating mode:** `Auto`
   You may choose the initial power up mode (`None/A/B/Auto`) of a dual RCP8 system. On power up, RCP8 first waits for guidance, either from IRIS or from the other RCP8, about which mode to enter. If the other RCP8 is dead, and if no mode requests have come in from IRIS, then the unit switches to its default powerup mode. Otherwise, the unit acquires the mode of the other RCP8, or follows the direction of IRIS.
   Selecting the `AUTO` initial powerup mode handles the case of starting an active scan with no user intervention when just one RCP8 is first switched on. Without this, the user would first have to explicitly choose `AUTO` from an **IRIS Dual Switching Menu**. But sometimes this is what you want; and by selecting the powerup mode of `None`, the system remains in maintenance mode until IRIS user makes a specific choice.

5. **This RCP8 is the 'Preferred' unit:** `NO`
   The `Preferred` unit value resolves negotiation "ties" by the switching algorithm. When confronted with a choice of using the `A` or `B` unit, and all else is equal, the `Preferred` unit is used. Vaisala recommends that you make system A the preferred unit:
   - In the system A setup, respond `YES`
   - In the system B setup, respond `NO`

6. **Include Data Processor NST faults:** `NO`
   This question allows you to include the fault status of the Data Processor reported by IRIS when determining whether a given channel is okay.
   - Answering `YES` means that both the data processor and the radar workstation must be working in order for the channel to be considered okay.
   - Answering `NO` causes only the radar server computer to be checked.

7. **Cooldown time after becoming inactive:** `3.0 sec`

8. **Additional warmup time when switching:** `3.0 sec`
   For redundant switching of antenna control, a minimum of 2 seconds is required by RCP8 itself. The value that is used depends on the specific characteristics of the system and should be measured for each system.

9. **Allow voluntary flipping between units:** `NO`
   The Dual/Redundant system code can switch between systems in response to requests from the host computer. RCV05 and XMT05 serial formats include 2 bits to control these transitions. One bit (WouldUse) announces that the host computer would like to use the antenna (whether or not it is available). The other bit (`Relinquish`) indicates that control can be voluntarily relinquished to the other system.
   When RCP8 receives a `Relinquish` offer, it checks the other unit to verify that it:

   - Communicates properly
   - Does not indicate any faults
   - Has `WouldUse TRUE` and `Relinquish FALSE`

   Under these conditions, if control were offered to the other unit, it would actually be in a position to accept it; and so, the switch-over is made at that instant. Since the algorithm only flips to a system that is actually ready to go, it automatically optimizes scheduling of the antenna as each radar is able to use it.
   The additional logic variables, `cDrcpWouldUse`, `cDrcpRelinquish`, `cDrcpDisabled`, `cDrcpWarmup`, `cDrcpOkay`, `cDrcpMaint`, `sOtherDrcpWouldUse`, `sOtherDrcpRelinquish`, `sOtherDrcpAlive`, `sOtherDrcpDisabled`, `sOtherDrcpWarmup`, `sOtherDrcpActive`, and `sOtherDrcpOkay`, appear in the control logic editor when voluntary flipping is enabled.
   The next group of questions concerns how IRIS Mode (as defined in the **Radar Status Menu**) is forced by RCP8 when an RCP8 switches from `INACTIVE` to `ACTIVE` state and vice versa.
   The first set of questions is for the switch from `INACTIVE` to `ACTIVE`. The example responses are for a redundant system with the modes configured as described in C.6. Configuring IRIS for Dual-System Support (page 193):

```
Choose: None Fixed Inherit Resume
Mode switch strategy when ACTIVE: Inherit
    IRIS mode #1 is valid to request: NO
    IRIS mode #2 is valid to request: NO
    IRIS mode #3 is valid to request: YES
    IRIS mode #4 is valid to request: YES
    IRIS mode #5 is valid to request: NO
    IRIS mode #6 is valid to request: NO
    IRIS mode #7 is valid to request: NO
Default mode to resolve illegal requests:3
```

   The 2 RCP8s negotiate which is the active system such that only one system can be `ACTIVE` at any given time. (Note that both systems could be `INACTIVE`). When a system is switched into active mode, it commands its IRIS to change operating modes (that is, the Radar Status Menu mode is loaded per RCP8 command). This question is used to determine what IRIS mode is commanded when a system is switched to active. RCP8 calls IRIS modes 1-7. The relation to IRIS mode names is made in IRIS Setups (RCP section). See C.6. Configuring IRIS for Dual-System Support (page 193). The strategy choices are:

`Fixed`
: RCP forces IRIS into a particular operating mode.

`None`
: In this case RCP does not request any IRIS modes.

`Inherit`
: RCP switches IRIS into the operating mode that was being used before the switch. If you respond `Inherit`, you are prompted to say what modes are valid to inherit.

`Resume`
: RCP switches IRIS to the mode that RCP was last run in. This is useful in the case of dual frequency systems that are sharing an antenna system since it allows a system to resume operation in passive mode even if it is not the active controller. It is not used for redundant systems.

Here the recommended response for a redundant system is `Inherit` so that when a system becomes active, it continues operation in whatever mode the other RCP had been using. This assures that if a system is running and faults, it continues in the same mode of operation after an automatic switch.

In the case of `Inherit` or `Resume`, RCP8 prompts you to specify which modes of operation are valid to inherit (or resume) when the system becomes active. In this case, based on the example IRIS mode configuration in C.6. Configuring IRIS for Dual-System Support (page 193), the `AIRPORT` and `AERIAL` modes would be allowed.

The final question in this sequence above specifies the mode to use when the inherited mode does not match any of the allowed modes. For example, if both systems are in `STANDBY` when a switch is made, the new `ACTIVE` system tries to inherit the `STANDBY` mode (mode 2 in our example). This is not a valid active mode so the `ACTIVE` system would start mode 3 (`AIRPORT`) instead.

10. **Allow mode changes within IRIS:** `YES`

Responding **NO** prevents both automatic and manual IRIS mode changes (from IRIS Radar Status Menu). RCP8 continually forces IRIS mode.

The second set of questions is for the switch from `ACTIVE` to `INACTIVE`.

```
Choose: None Fixed Inherit Resume
Mode switch strategy when INACTIVE:
FIXED Fixed IRIS mode to request: 2
    Allow mode changes within IRIS: NO
```

Fixed is recommended for redundant systems, since it can be used to force IRIS into a `STANDBY` mode when its RCP8 becomes inactive. In this example, mode `Z` is used, which corresponds to the mode called `STANDBY`.

```
Mode to request during Maintenance ACTIVE: 0
Mode to request during Maintenance INACTIVE: 1
    Allow mode changes from within IRIS: YES
```

In the example, the `STANDBY` mode is forced when the system goes `INACTIVE`. Note that this makes it impossible to do any modification of IRIS mode for development or maintenance. In this case, it is recommended that you temporarily respond `YES` to this question.
These questions specify what IRIS modes should be set when RCP8 is intentionally disabled (placed in "Maintenance Mode") by either the hardware `A/B` switch or the **Select** switch in IRIS Switching menu.
You may choose IRIS mode to request for `Maint Active` and for `Maint Inactive`. If the requested mode is nonzero, then an additional question appears to choose whether auto mode switching is allowed. A recommended strategy is to request a mode of zero in `Maint Active`, so that a running RCP continues doing whatever it was doing already. `Maint Inactive` should request IRIS maintenance RST mode, and allow auto switching.

C.5. RVP TTY Setups for Dual System Support

In RVP, you must define whether to inhibit the trigger when its associated RCP is inactive. Whether or not this is necessary is determined by the radar manufacturer. Respond to the following questions in the **mt** (general trigger setup) section:

```
Blank output triggers according to TAG#0: YES
Blank when tag input is high: YES
Blank triggers 1:YES 2:YES 3:YES 4:YES 5:YES 6:YES
```

These questions control trigger blanking based on the `TAG0` input line. You first select whether the trigger blanking feature is enabled; and then optionally choose the polarity of TAG0 that results in blanking (consistent with RCP8 setups), and which subset of the 6 user definable triggers are to be blanked.

Note that for maintenance, RVP900 on the inactive system can be made to generate a trigger by any one of the following techniques:

- One of the 6 triggers can be set to `Blank triggers NO`. This could be used to provide a permanent maintenance trigger.
- Temporarily respond `Blank output .. TAG#0: NO`
- Install a physical override switch on one of RCP8 BITE inputs and then, using the control logic, enable the triggers.

C.6. Configuring IRIS for Dual-System Support

You must configure IRIS to support dual systems.

1. Separately configure systems `A` and `B` so they can function independently:
   a. C.6.1. IRIS Radar Status Menu Mode Configuration (page 193)
   b. C.6.2. IRIS Setup Utility Mode Configuration (page 195)
   c. C.6.3. IRIS Status Product Configuration (page 196)
   d. C.6.4. BITEX Setup (page 198)
2. Test dual-system operation

C.6.1. IRIS Radar Status Menu Mode Configuration

IRIS `Mode` is determined by the **Radar Status Menu**, that is, the name at the top of the menus. When IRIS first starts, the `DEFAULT` mode loads. During operation, you can change the mode using one of the following mechanisms:

- Manually in the **Radar Status Menu** by selecting **File > Change RST...**.
- Automatically with a warning product. This is often used to switch between a surveillance mode and perhaps a volume scan or wind shear detection mode.
- Externally, forced by RCP. You c an configure RCP to either constantly force a mode (prohibiting manual or automatic mode changes), or trigger a mode change and then allow a manual or automatic mode change from the **Radar Status Menu**.

This example used is for redundant system operation of a wind shear detection system that has two operational modes called `AERIAL` (for general weather monitoring) and `AIRPORT` (for optimized for wind shear detection. The following table shows a summary of IRIS configuration.

Table 82 Example of a Redundant IRIS Dual System Set-up

RCP02 Mode Number	Radar Status Menu (IRIS mode name)	TASK Schedule	Product Schedule	Output Schedule
1	`MAINTENANCE` Mode entered when the system is intentionally disabled by either the hardware or software selector switches. It is recommended to blank the task schedule, clear all output assignments and generate no products. For safety, the servo power and radiate can be set to **OFF** in the **Radar Status Menu**.	`MAINTENANCE`	`MAINTENANCE`	`MAINTENANCE`
2	`STANDBY` Task scheduler is set to inactive (no task scheduled). For safety, the servo power and radiate can be set to **OFF** in the **Radar Status Menu**.	`STANDBY` (inactive)	`PRODUCT`	`OUTPUT`
3	`AERIAL` Configured for a long range surveillance scan and a volume scan for routine weather monitoring.	`AERIAL`	`PRODUCT`	`OUTPUT`

RCP02 Mode Number	Radar Status Menu (IRIS mode name)	TASK Schedule	Product Schedule	Output Schedule
4	`AIRPORT` Optimized for wind shear detection.	`AIRPORT`	`PRODUCT`	`OUTPUT`

It is assumed that the same product schedule and output schedules are configured for all modes except the `MAINTENANCE` mode, that is, these schedules include all of the required products for all of the modes. It is possible to have different product and output schedules, but this increases the configuration maintenance.

> For the recommended redundant configuration, you need to have the system disabled by the hardware A/B switch or the A/B select in the **Switching** menu to configure IRIS modes. For normal redundant configuration of RCP, this forces IRIS into `MAINTENANCE` mode and then release IRIS to allow manual mode changes for configuration of the other IRIS modes.

C.6.2. IRIS Setup Utility Mode Configuration

You must configure the modes of operation in the **IRIS Setup Utility** in RCP section as shown below. Note that the first column in the table in C.6.1. IRIS Radar Status Menu Mode Configuration (page 193) gives the numbers for each mode used in the example.

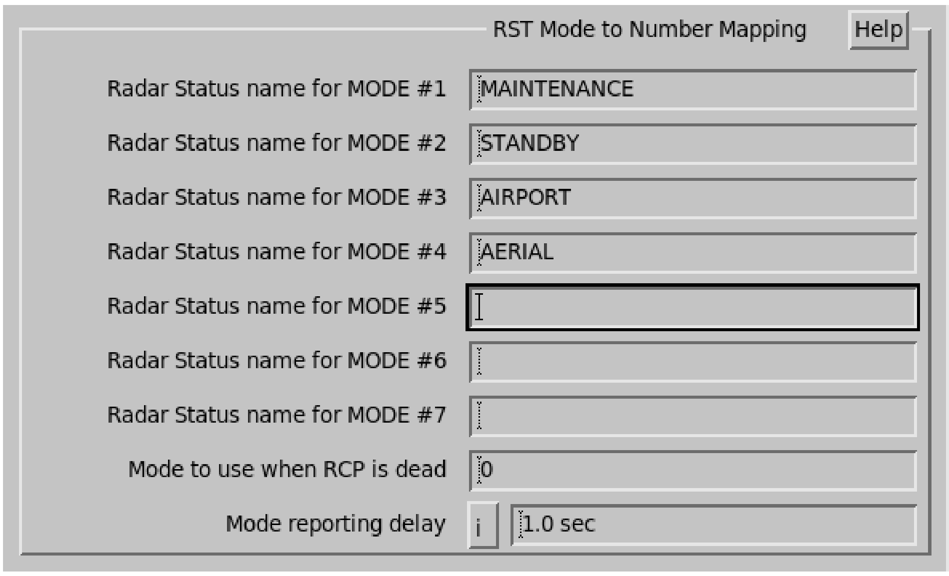

Figure 14 Radar Control Processor Setups - RST Mode to Number Mapping

The modes are coded 1-7 and must match the mode names configured in **IRIS Radar Status Menu**. To force IRIS to switch to the requested mode you must also enable **External RCP Mode Change"** in **IRIS Setup General** question as shown below.

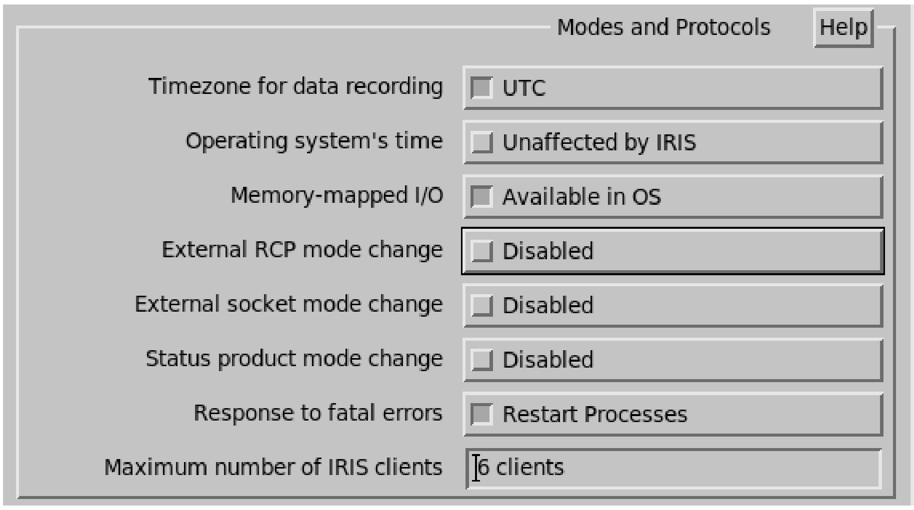

Figure 15 IRIS General Setups - Modes and Protocols

C.6.3. IRIS Status Product Configuration

Status products are produced at regular intervals at each IRIS radar workstation (A and B). These are used by the RCPs to assess whether a system is OK or in FAULT, that is, whether a system is available for use. The **Status** product collects information from various sources and faults if any of the following occurs:

- BITEX critical faults
- RCP communication failure (RCP DEAD)
- RVP signal processor error
- IRIS internal critical fault. These are internal errors that are flagged as critical in IRIS.

The **Status** product provides information on all system components. For a redundant system, IRIS A must be configured to send its **Status** product results to both RCP A and IRIS B and vice versa.

You must configure the **Status** product for dual-system support.

1. Enable status product generation in IRIS `setup/product` as as shown in the following figure.

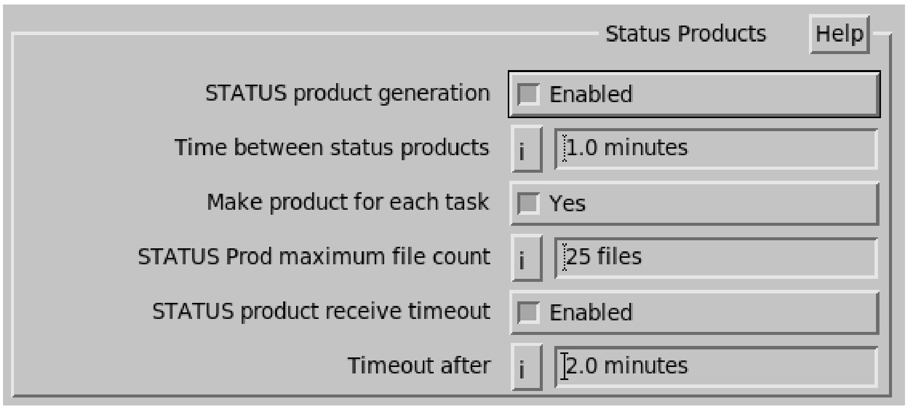

Figure 16 IRIS Product Setups - Status Products

If there is a change in status such as a critical fault, the **Status** product is generated immediately. The configuration is in setup/product is for the maximum time between **Status** products.

IRIS **Status** product result (`OK` or `FAULT`) is sent to RCP on the serial line.

2. To be able to to identify which sites are being reported, use IRIS *Setup/RCP* to configure the network identifiers. Make the configurations identical on both systems. Use the 3-letter site codes that are configured in **Setup General** as for a standard IRIS system.
   In the following example, two sites are in a redundant system called RDA and RDB.

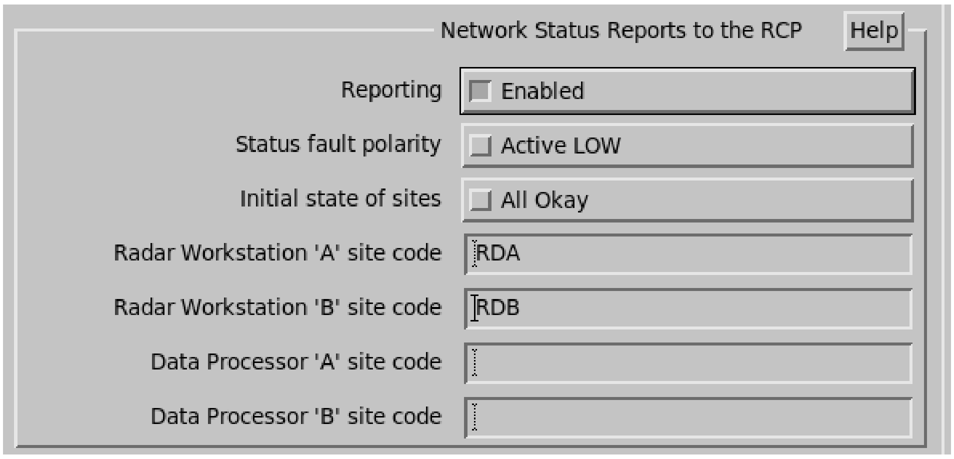

Figure 17 Radar Control Processor Setups - Network Status Reports to RCP

3. Use the **Product Output Menu** to send the **Status** product from IRIS A to IRIS B and vice versa.

C.6.4. BITEX Setup

By default, **BITEX** is configured to display all of the available status parameters.

You must identify those parameters that signal a `Critical Fault`. To do this, you must enter as **`bitex-setup`** with operator privilege. The sub menu for configuring each field has a critical flag (right-click on the field number).

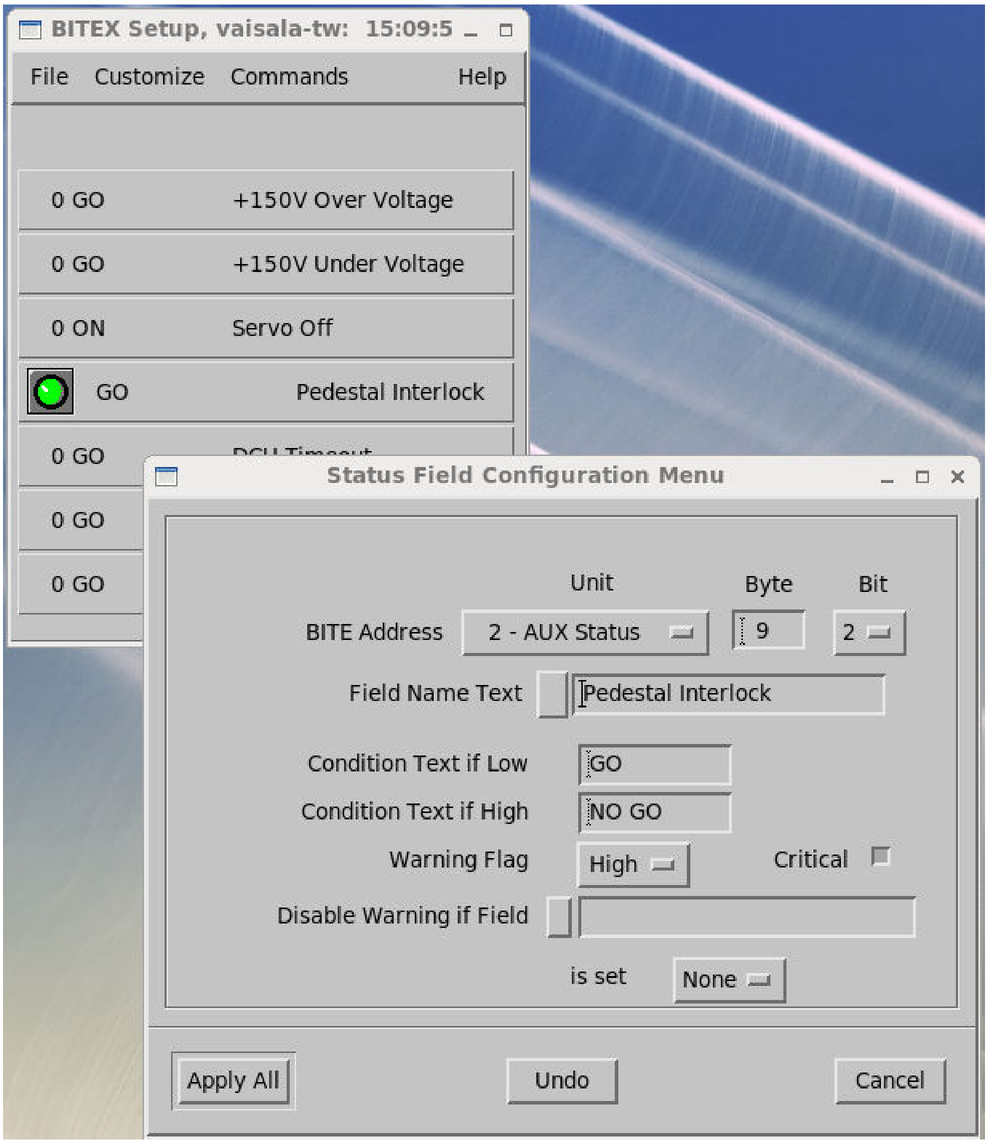

Figure 18 BITEX - Status Field Configuration Menu

In the example shown in the figure, the fields for **Pedestal Interlock** is set as critical fault. When IRIS receives a critical fault message from BITEX, IRIS site status is set to fault. This message is passed back to RCP and, by network transfer of the status product, to the other IRIS system.

C.7. Configuring Dual Frequency Systems

Dual-system RCP can operate two separate transmitter/receiver systems through the same antenna pedestal system, that is, a dual-frequency system.

In this case, a radar can be thought of as the transmitter/receiver of each system and its associated RCP, RVP, and IRIS system. These 2 radars share the same antenna and pedestal. The constraint is that only one RCP can control the antenna/pedestal at a time.

There are 2 modes of operation possible in this case: Flip mode and Simultaneous Active/Passive mode.

> In principle it is possible to operate in using the “flip” mode (A or B) and the active/passive (A and B) mode simultaneously. In this case the radars would flip between active and passive. However, this is more complex to configure. For active/passive operation, it is simpler to select one radar to always be the active system and never “flip” the active system to the other radar.

RVP setups do not require special consideration for dual-system operation.

A or B (“Flip”) mode: Selectable or Alternating Active Radar:

This case is identical to the operation in the case of a redundant system, that is, either radar A or radar B is used exclusively.

The switching menu or the hardware selector switch can be used to force the exclusive use of one “radar” or the other. In addition, the **TASK Scheduler** menu provides support to “flip” between the two radars so that IRIS can automatically run a task from on one radar and then relinquish control so that the other radar can run a TASK.

A and B mode: Simultaneous Active/Passive Radar

In this case, one “radar” system is used to actively control the antenna scanning while the other radar system acquires data passively. This allows both radars to operate simultaneously.

RVP of the passive systemstill generates pulse width change output and triggers and the associated passive RCP still has control to turn-on the radiate and T/R power.

The IRIS Status Product, sent from the active system, is used by the passive system to determine which task in the schedule should be run in passive mode.

Comaprison of Dual-redundant and Dual System Parallel Systems

A major difference between the dual-redundant case and the dual-system parallel operation is that RCP does not typically set the mode of operation, that is, it does not control IRIS mode selection in the **Radar Status Menu**.

In the redundant case, it is important that in the event of a failure, IRIS be told what to do after an automatic switch-over, that is, the RCP that takes control tells IRIS the operating mode.

For dual-system parallel operation, IRIS tells the RCPs how to operate, that is, which is active and which is passive. This means that the RCP setup (on RCP and on IRIS) does not involve defining all the operating modes, mode reporting and mode switching strategies.

In most cases, RCP8 and IRIS setups for dual-system parallel operation are the same as for the dual redundant system.

C.7.1. RCP Setups for Flip or Simultaneous Operation

For more information, see C.4. RCP TTY Site Settings for Dual System (page 188).

1. **Site Custom > Allow voluntary flipping > YES**
   Set this to `Yes` on both RCPs if you intend to use the Flip (A or B) feature.
   This allows the **IRIS TASK Scheduler** of one radar to assume control, and then relinquish control ("flip") to the other radar.
   IRIS TASK Scheduler menu provides support for this in the **Flags** column (that is, the "Flip" flag).
2. **Site Custom > Mode Switch Strategy When ACTIVE (INACTIVE) > NONE**
   Set the responses to **NONE** on both RCPs.
   In this case, the mode `0` is always requested by RCP, which means that the operator controls IRIS mode, not RCP.
3. **Site Custom > Mode to request during Maint ACTIVE (INACTIVE) > 0**
   Respond `0` on both RCP8s.
   "Maintenance Mode" is defined to be when either the hardware switch or the software switch (**Switching Menu**) is set to disable a radar.
   Setting the mode request from RCP to `0` allows the normal IRIS **Radar Status Menu** mode to be controlled by the operator without interference from RCPs, that is, RCP does not force IRIS mode changes.

C.7.2. IRIS Setups for Flip or Simultaneous Operation

1. **setup/ingest/Scanning Options > Task Scheduling Control > Active/Passive**
   To use the simultaneous active/passive feature, set this to **Active/Passive** on both RCPs. This is recommended since it allows either radar to assume the active role.
   You can also set one radar to **Active Only** and the other to **Passive Only** if you want to dedicate the systems to these roles. In this case, you cannot use the passive system on its own.
2. **setup/ingest/Scanning Options > Passive: use external trigger rate > No**
   Responding `NO` allows the RVP of the passive system to generate its own trigger.
   This is generally recommended so that the two radars can be triggered independently by their own RVP, for example, to use dual PRF. Set this on both systems.
3. **setup/ingest/Scanning Options > Passive: use status product task > Yes**
   Respond `YES` to slave the passive system task to the active one using the Status Product.
   Do this on both systems.
   You also must make sure that Status Products are enabled (C.6. Configuring IRIS for Dual-System Support (page 193)) and set the **Product Output Menu** to automatically send the Status product to the other system. This should be done for both systems.

4. **setup/rcp02/Status Reports to the RCP**

```
Reporting : Enabled
Status fault polarity : Active LOW
Initial state of sites : All Faulted
Radar Workstation ’A’ site code : MPK
Radar Workstation ’B’ site code : MPW
Data Processor ’A’ site code :
Data Processor ’B’ site code :
```

This question asks for the 3-letter site code of the other system. Do this for both systems, each referring to the other.
These settings enable the 2 RCP8s to know the status of each radar. This is necessary for switching between the 2 systems in “flip” mode or using one or the other system under manual control. If the status of a system is `FAULT`, then switching of active control is disabled to that system.

5. **setup/rcp02/RST Mode to Number Mapping**

```
Radar Status name for MODE #1 : DEFAULT
Radar Status name for MODE #2 : DEFAULT
Radar Status name for MODE #3 : DEFAULT
Radar Status name for MODE #4 : DEFAULT
Radar Status name for MODE #5 : DEFAULT
Radar Status name for MODE #6 : DEFAULT
Radar Status name for MODE #7 : DEFAULT
Mode to use when RCP is dead : 0
Mode reporting delay : 1.0 sec
```

RCP8 can send commands to IRIS to change operating modes. These commands are codes (1–7) that are associated here with different **Radar Status Menu** names. In general, RCP is always requesting mode 0 which is a special code that allows the operator to specify IRIS mode in the **Radar Status Menu**. As a safety feature, the DEFAULT IRIS mode is set for all of the other numerical codes (1–7) that could be commanded by RCP. Thus, in the event that RCP were to request a mode (other than 0), it would be the DEFAULT **Radar Status Menu** configuration.

6. **setup/general/Modes and Protocols**

```
External RCP mode change: Disabled
```

Setting this to disabled assures that IRIS modes changes cannot be forced by RCP.

C.7.3. IRIS Task Scheduler: Flip Operation

To use the Flip feature, you must set the **Switching Menu** to the **Auto** position so the RCPs can negotiate which radar to use.

1. In the **TSC Editor Menu**, right-click the **Flags** column to toggle the **Flip flag on** or **off**.

> Do NOT to set the **Late Skip flag** to **Yes**.
> This could interfere with the flipping since in general a flip is forced by a task that is late. That is, it wants to run but it cannot because the other radar is in control.

   For example, if you want a task to run on radar A and then another task on radar B:

   a. Set up the task schedules with the appropriate tasks on the 2 systems and set each to **Flip**.
   b. After radar A runs its task, it sees that the `flip` flag is set and tells the RCPs that it is willing to relinquish radar control.
   c. If radar B wants to run its task (because it is the scheduled time), it tells the RCPs that it wants to run. RCP then automatically releases control to radar B.
   d. If each task is set to run continuously, (**Repeat** set to 00:00), so the tasks on the A and B radars alternate.

2. Add an appropriate name to the task in the **Task Schedule**.

> Vaisala recommends that including the word `flip` or other code in the **Task Schedule** and perhaps the **Radar Status Menu** operating mode to indicate that the systems can flip. For example:
>
> - In the the **Task Schedule**: `PPIVOL_FLIP`
> - In the **IRIS Radar Status Menu**: `FLIP_MODE`

C.7.4. IRIS Task Scheduler: Simultaneous Active and Passive Operation

To use the simultaneous active/passive feature, you must set the **Switching Menu** to specify the active system.

The active system can be forced by selecting it exclusively. You can also leave the **Switching Menu** in **Auto** mode after the active system is forced.

1. At the topic of the **TSC Editor Menu**, toggle the **Active** or **Passive** field.
   You can toggle between these two choices if the setup/ingest/scanning is set to active/passive. Otherwise, the text is fixed to either active or passive depending on your selection in setup.
   For simultaneous active/passive operation, the **TSC Editor Menu** of one system should be set to **Active** and the other set to **Passive**.
   a. In the active system, configure the task as usual.
   b. On the passive system, configure a task with the same name.
      You can configure most task parameters can be configured independently except for the antenna scanning parameters (scan mode, azimuth, elevation, scan speed). The scan resolution does not have to be the same.
      For example, for a PPI scan mode, the elevation angles in the passive system task must match those in the active system task.
      Similarly, if you are doing RHIs, the elevation limits and selected azimuth angles must match on the 2 tasks.
      Other than the scan parameters, you have complete freedom to select PRF, processing mode, range, resolution, and similar.
2. Make sure the Status Products are enabled on the active system and that the Status Product is output (in the **Product Output Menu**) to the passive system.
   This is how the passive system knows which task to run.
   In operation, when the active system starts a task, it sends a Status product to the passive system. The passive system reads the status product and checks to see if there is a passive task in the **TASK Scheduler** with the same name. If there is, it runs the passive task.
3. Save the **TSC Editor** menu with a name to indicate active or passive.
4. Save the IRIS mode (**Radar Status Menu** name) with a name that reflects active or passive (for example, `ACTIV_1` or `PASSV_1`).

C.8. IRIS Dual System Switching Menu

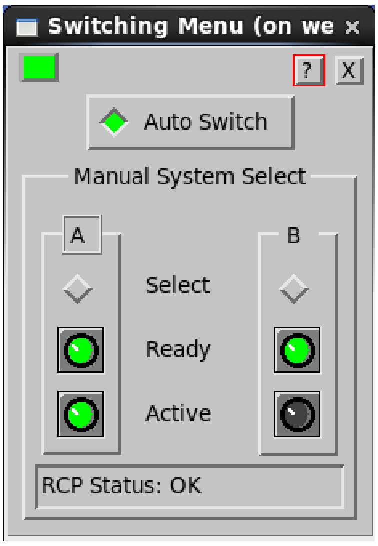

Figure 19 Dual-System Switching Menu

The dual system **Switching Menu** provides a interface for controlling and monitoring the dual-system by allowing the operator to:

- See which system (A and/or B) is ready for operation, and which one is currently active (only one is active at any time).
- Select the exclusive use of either A or B, or enable automatic switching, which occurs in the event of a fault.

By default, the menu is set for automatic switching. The menu can run locally at either Radar A or B, or remotely over the network to allow remote controlling and monitoring.

When running the menu on system A, the menu communicates directly to RCP8 A through RCP–to–IRIS serial line and indirectly to B through RCP–to–RCP communications link.

The **Switching Menu** does not control RCP8 switching, rather it submits switch requests to RCP8s, which then negotiate whether the request can be fulfilled.
For example, if the operator requests that System A be used, but system A is not available because it is in fault or has been disabled by the hardware A/B switch, then the request is not fulfilled. In this case neither system becomes active. If the operator requests auto switching (the recommended mode of operation), then RCP8s negotiates to activate the available system.

The following table shows the contents of the **Switching Menu**.

Table 83 Switching Menu Description

Button or Light	Possible Values	Description
Flashing Light (upper left)	• Green - Communication with both RCPs is OK • Red - Communication failure of either IRIS-to-RCP link, or RCP-to-RCP link.	The flashing light at the top-left indicates that the menu is live.
Auto Switch	Green -OK	For normal operation **Auto Switch** is enabled. This allows RCP8s to negotiate which system to use. However, operators can request the use of only system A or B. For normal maintenance functions however, you should select a single system should be done using the hardware A/B selector switch at the radar and keep the **Switching** menu in the Auto position. Sometimes it is useful to temporarily force the systems to switch and then immediately reset back into the Auto mode, just to exercise or test the other system.
Ready	• Green - The system status is OK. • Red - The system status is in fault, that is, it cannot be used until the fault is cleared. • Yellow Bezel - Indicates that the hardware A/B select switch at the radar has disabled the system, for example, placed it in a maintenance mode. The system cannot be used operationally until the switch is set to enable the system. • X - The status is unknown because of a communication problem.	Shows the system status and whether the hardware A/B switch at the radar is set to disable the radar. For a system to be used for operation, **Ready** must be green (status OK) and the Bezel must be gray (not disabled). This makes it easy to see whether the system is ready for operation, and if not, what course of action would make it operational.
Active	• Green - The system is active, that is, it is running or could run an IRIS task. • Off - The system is inactive. Either the system is in standby (ready to become active) or it has been disabled. • Yellow - The system is transitioning from inactive to active during its warm-up period. See C.4. RCP TTY Site Settings for Dual System (page 188).	An active indicator is also displayed on the RCP8 front panel. See C.4. RCP TTY Site Settings for Dual System (page 188). The RCP8 front panel either On or −−.

Button or Light	Possible Values	Description
Status field	• Gray - Communication with both RCPs is OK • Red - Communication failure of either IRIS-to-RCP link, or RCP-to-RCP link.	If there is a failure, shows red and describe the nature of the failure. If there is a failure, X symbols appear over the status indicators for the system for which communication has failed. See C.9. Troubleshooting Dual Systems (page 207).

C.8.1. Starting Switching Menu

1. Make sure you have Operator priviledges.
   Operator privilege is required to access this menu. The **Switching Menu**.
   The Operator password protection allows the System Manager to limit access so that only authorized personnel can use the menu.
2. Started the menu from either :
   - In the **Antenna** utility, select **Options > Dual-System Selection**.
     This is convenient if IRIS is not running, for example, for configuration and initial testing. You can start antenna by typing `antenna &` in an X terminal or through the **Utilities Menu** which is most easily accessed through IRISnet.
     For more information, see *IRIS and RDA Utilities Guide*.
   - **IRIS Radar Status Menu**
     On the **IRIS Radar Status Menu**, select **Mode > Dual-System Selection** .
     The IRIS Menus must be connected to either Radar Workstation A or B, although the menus themselves can be running on any networked workstation. This method is very convenient for overall control and monitoring since the **Radar Status menu** provides access to all status monitoring features of IRIS including the bitex utility.
     For more information, see *IRIS Radar User Guide*.
3. Choose whether you want to connect the menu to system A or B.
   A small box appears around the letter A/B over the status lights to indicate the system to which it is connected.
4. If you wish, you can run multiple instances of the **Switching Menu**.
   This is convenient for monitoring status from multiple locations on the network. The IRIS server processes requests in the order they are received.

C.9. Troubleshooting Dual Systems

The **Switching Menu** is the primary user interface for troubleshooting problems related to switching between radar systems.

C.9.1. Normal Switching Menu

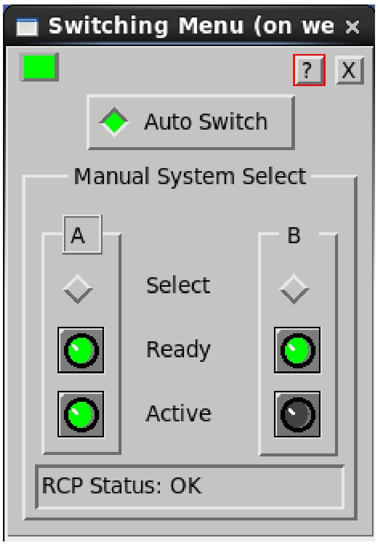

Figure 20 Dual-System Normal Switching Menu

Auto Switch Green		
A		B
Off	Select	Off
Green/Gray	Ready	Green/Gray
Green	Active	Off

Both systems have **OK** status. Neither system is disabled by the hardware A/B switch. System A is currently active, but system B is ready to run in the event of a fault.

C.9.2. System B Disabled (Maintenance Mode) by Hardware A/B Switch

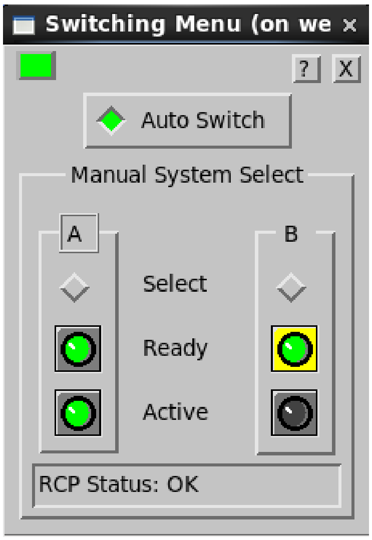

Figure 21 Dual-System B Disabled (Maintenance Mode)

Auto Switch Green		
A		B
Off	Select	Off
Green/Gray	Ready	Green/Yel
Green	Active	Off

The yellow bezel of the **Ready** light indicates that maintenance personnel have disabled system B.

Action

To make system B ready for operation set the hardware switch to enable B (that is, to allow automatic switching).

C.9.3. System A in Fault, System B Running

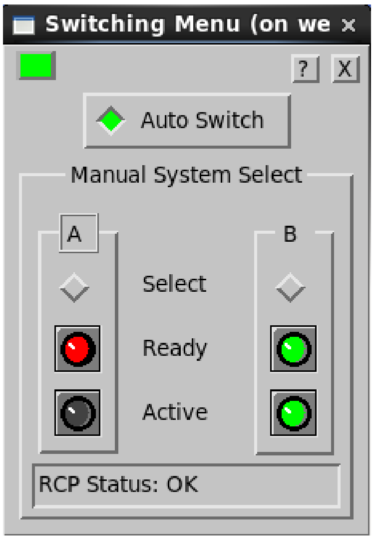

Figure 22 Dual-System A in Fault, B Running

Auto Switch Green		
A		B
Off	Select	Off
Red/Gray	Ready	Green/Gray
Off	Active	Green

System A has faulted as indicated by the red **Ready** light. System B is now running.

Action

Use the **Radar Status Menu** for system A to determine the nature of the fault. Check:

- Subsystem Status of RVP and RCP (lower right)
- BITE status (summary display and access lower left)
- Message menu (top middle)

C.9.4. System B in Fault and Placed in Maintenance Mode

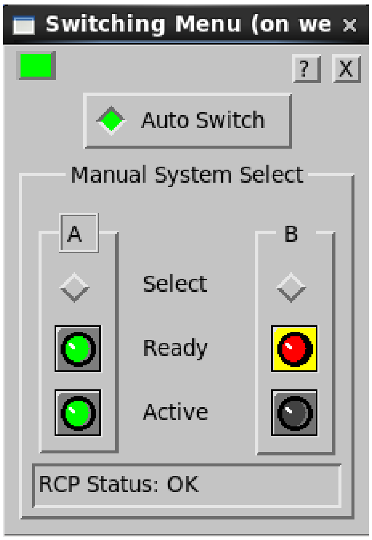

Figure 23 Dual-System B in Fault and Place in Maintenance Mode

Auto Switch Green		
A		B
Off	Select	Off
Green/Gray	Ready	Red/Yel
Green	Active	Off

System B has faulted as indicated by the red **Ready** light. The yellow bezel on the system B **Ready** light indicates that B has been disabled by the hardware switch. It has probably under repair by maintenance personnel.

Action

Use the **Radar Status Menu** to identify the fault.

When the fault is cleared, set the switch to enable system B.

C.9.5. Both Systems Faulted, No Operation Possible

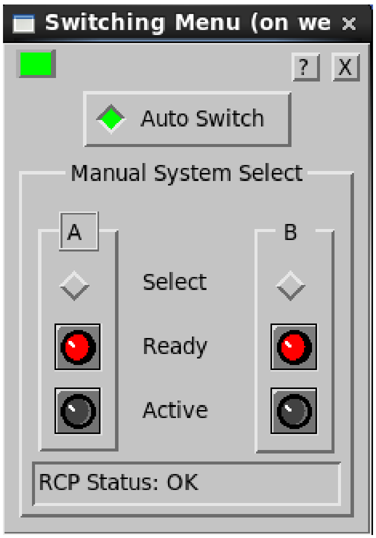

Figure 24 Dual-System Both Systems Faulted (No Operation)

Auto Switch Green		
A		B
Off	Select	Off
Red/Gray	Ready	Red/Gray
Green	Active	Off

Both **Ready** lights are red, indicating that neither system can be run.

The **Active** lights are both off, indicating that neither system is running

The **Ready** light bezels are both gray indicating, that maintenance personnel may not be working on the problem (the hardware switch has not been set to maintenance mode for either system).

Action

Use the **Radar Status Menu** to identify the faults. When the fault is cleared, the system automatically resumes operation on the good system.

C.9.6. RCPB is Unreachable (As Viewed from System A)

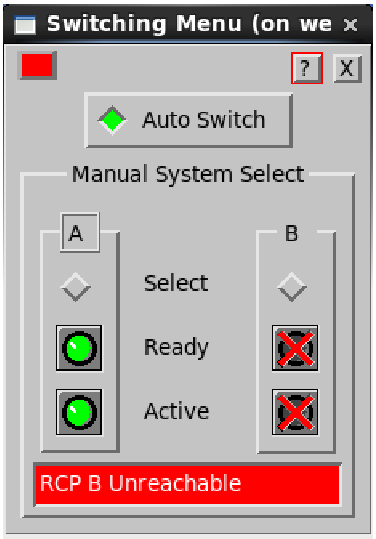

Figure 25 Dual-System RCPB is Unreachable

Auto Switch Green		
A		B
Off	Select	Off
Green/Gray	Ready	X
Green	Active	X

In this example, the menu is viewed from system A as indicated by the small box around the letter "A".

The X's for the **Ready** and **Active** lights on system B indicate that status information is unavailable because of a communication failure of the RCP–to–RCP link.

Problem

RCP–to–RCP link has failed, or RVPB has been turned off.

Action

Check if RCP8 B has been turned off.

Check RCP–to–RCP link cable:

1. Check the front panel.
2. Check the **Radar Status Menu** for system B ("RCP Dead" in Subsystem Status).
3. Start the **Switching Menu** on system B and check the 4 Xs.

C.9.7. RCP is Dead

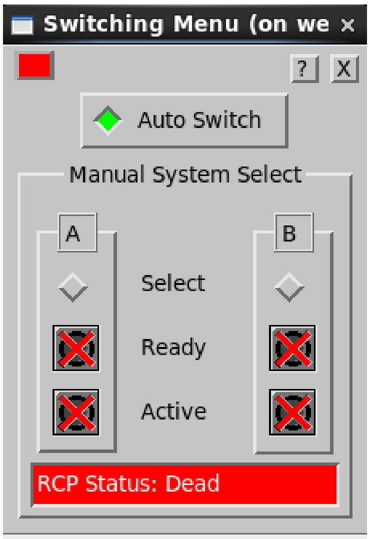

Figure 26 Dual-System RCP Status: Dead

Auto Switch Green		
A		B
Off	Select	Off
X	Ready	X
X	Active	X

The 4 X's indicate that the RCP (in this example, system B as indicated by the [B]) is not communicating to the IRIS workstation.

Problem

IRIS–to–RCPB link has failed, or RCP8B has been turned off.

Action

Check if RCP8 B has been turned off. Check IRIS–to–RCPB link cable. To see if RCP has been turned off:

1. Check the front panel.
2. Check the **Radar Status Menu** for system B ("RCP Dead" in Subsystem Status).

C.9.8. Operation Forced to A—Auto Switching and B Disabled

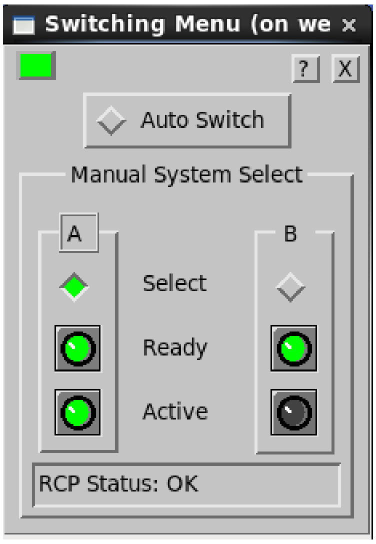

Figure 27 Dual-System Operation Forced to A (Auto Switching and B Disabled)

Auto Switch Off		
A		B
Off	Select	Off
Green	Ready	Green
Green	Active	Off

The A **Select** is green, indicating that it has been selected (by pushing the button in this menu) to be the exclusive system.

The A **Ready** and **Active** lights are both green. B is disabled by this action, similar to disabling B using the hardware switch. However, B's **Ready** light is green so that it could be used.

Action

Restore the system to **Auto Switch** mode as soon as possible.

If the selection of A was made because of intermittent behavior of B, then isolate the problem.

It is recommended to set the hardware A/B switch to disable B during maintenance and then re-enable **Auto Switch** . This way when maintenance on B is complete, the system is ready to switch automatically without operator intervention.

C.9.9. Avoiding Non-Operational Conditions

C.9.9.1. Non Operational, Both Systems Disabled (page 216) and C.9.9.2. Non-Operational, System A is Menu Selected, but Faulted (page 217) show that it is best to leave the **Switching Menu > Auto Switch** enabled at all times.

To place a system in maintenance mode, it is recommended that the hardware switch be used.

If absolutely necessary, you can use the **Switching Menu** to disable a system temporarily until someone can go to the radar site and set the hardware A/B switch.

The only other reason for selecting A or B in the **Switching Menu** is to force the systems to switch to "even-out" or test the usage of each system. After the switch is forced, you should immediately re-enable the **Auto Switch** .

C.9.9.1. Non Operational, Both Systems Disabled

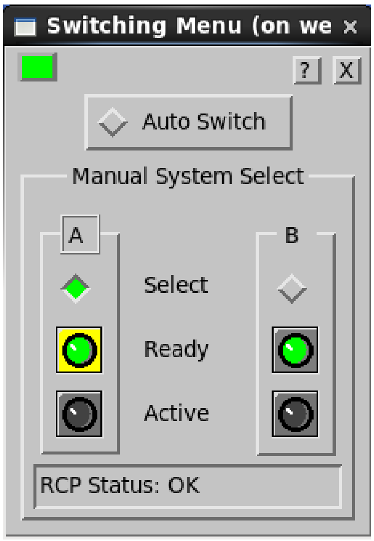

Figure 28 Dual-System Non-Operational, Both Systems Disabled

Auto Switch Off		
A		B
Green	Select	Off
Green/Yellow	Ready	Green/Gray
Off	Active	Off

Both **Active** lights are off indicating that no operation is possible. The reason is that the hardware A/B switch has disabled A and the software switch in this menu has disabled B.

Action

Set **Auto Switch** so that system B automatically switches to active.

C.9.9.2. Non-Operational, System A is Menu Selected, but Faulted

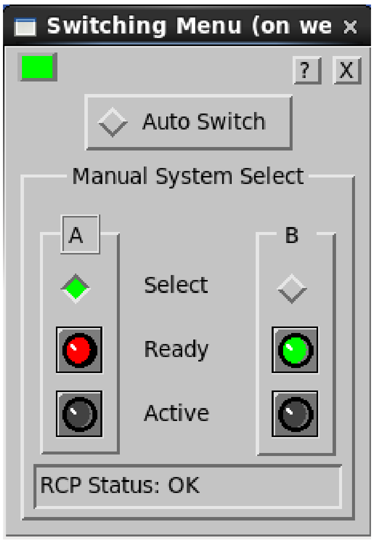

Figure 29 Dual-System Non-Operational, A is Menu Selected, but Faulted

Auto Switch Off		
A		B
Green	Select	Off
Red/Gray	Ready	Green/Gray
Off	Active	Off

Both **Active** lights are off indicating that no operation is possible. The reason is that system A has faulted and software switch in this menu has disabled B. However, system B is ready to use.

Action

Set **Auto Switch** on. System B automatically switches to active.

Use the **Radar Status Menu** on system A to determine the fault.

Index

Technical Support

Contact Vaisala technical support at helpdesk@vaisala.com. Provide at least the following supporting information:

- Product name, model, and serial number
- Name and location of the installation site
- Name and contact information of a technical person who can provide further information on the problem

For Vaisala Service Center contact information, see www.vaisala.com/servicecenters.

Warranty and Product Returns

For standard warranty terms and conditions, see www.vaisala.com/warranty.

Please observe that any such warranty may not be valid in case of damage due to normal wear and tear, exceptional operating conditions, negligent handling or installation, or unauthorized modifications. Please see the applicable supply contract or Conditions of Sale for details of the warranty for each product.

If the product is faulty, these steps help speed up the return process and avoid extra costs.

1. Read the warranty information.
2. Contact Vaisala technical support and request a Return Material Authorization (RMA) and shipping instructions.

> Always request the RMA before returning any faulty material.
> Provide the failure report as requested.

Recycling

Recycle all applicable material.

Follow the statutory regulations for disposing of the product and packaging.

VAISALA