4. TTY Setup Menus

This section describes how to configure the RCP8 software parameters to match the particular specifications of each radar site. These parameters are stored on the hard disk drive in a file called /usr/sigmet/config/rcp8.config. This ASCII file is convenient for backup and restore. For diagnostic work and trouble shooting, SIGMET may request that a copy of this file be emailed to SIGMET.

This section serves as a general reference for the configuration and stabilization procedure described in **Appendix XXX**. Note that SIGMET recommends that only trained personnel be permitted to perform antenna stabilization. Training should include basic knowledge of the theory of servo operation (**Section XXX**) as well as the procedures for using the RCP8 TTY menus (this section and **Section XXX**). SIGMET offers training to those customers who wish to perform antenna stabilization.

⚠ **Disclaimer: In no event shall SIGMET be liable for any damage to the antenna/pedestal system that may occur during stabilization configuration performed by the customer.**

In this chapter:

4.1 Using the SETUP Menus

The configuration parameters are setup in an interactive fashion with the use of the TTY. To change a parameter setting, refer to the summary dialogs on the following pages to determine the general category in which the parameter appears. Then, choose that category from the Main Menu. Additional information and prompts will be typed as appropriate. The parameter values will be displayed for each category and the RCP8 will pause for the input.

For each question, the parameter values are displayed then paused to allow the user to respond:

- If the current value is correct, press ENTER.
- If a new value is required, enter the new value. Be careful to enter the numeric values in the proper physical units. For example, do not input the time in seconds if the question is expecting the time in minutes. The RCP8 will then display the expected units with each question.

  Once ENTER is pressed, the RCP8 echoes the new value to verify that it has been correctly entered. If the new value is correct, press ENTER again to proceed to the next parameter.
- Type "up" or "u", and the SETUP menu will return to the previous question.
- Type "quit" or "q" to exit a submenu and return to the top level "RCP>" prompt.

When all of the questions have been answered, or if the "QUIT" command is entered, the RCP8 will run the new values and return to the command prompt. The user may select another SETUP menu command or the user may run the RCP8 with its new settings to verify that the changes are correct. When you are satisfied with the changes, use the "SAVE" command to make the settings permanent.

Note: The ESC key retains its general meaning and provides a single keystroke method of exiting the SETUP menus, as well as all other menu levels, within the RCP8.

4.2 Summary of Setup TTY Configuration Parameters

Below is a list of all configuration parameters and their factory default values.

"SITE" Command Summary Dialogs

```
RCP> site display
Front Panel Display Setups
--------------------------
Front panel display update time: 0.200 sec       New Value:
Front panel display brightness (0-7): 4          New Value:


Available Front Panel Display Templates
  TB )  |--AZ--        --EL--|    Title Bar
  EP1)  |123.45  Earth  23.46|    AZ/EL Earth Positions
  EP2)  |123.45   Pos   23.46|
  EP3)  |123.45  AZ/EL  23.46|
  PP )  |123.45   Ped   23.46|    AZ/EL Pedestal Positions
  PV )  | 12.00  Ped.V   0.00|    AZ/EL Pedestal Velocities
  EV1)  | 12.00  Geo.V   0.00|    AZ/EL Earth Velocities
  EV2)  | 12.00   Vel    0.00|
  CMD)  | 12.00 V Cmd P 10.50|    AZ/EL Servo Commands
  RTS)  |Rad:On T/R:-- Srv:On|    Radiate, T/R, and Servo
  HRP)  |123.4H -12.3R  23.1P|    Heading, Roll, Pitch
  LL )  |148 24.5W   42 35.2N|    Longitude / Latitude
  TM )  |17:22:35  3-Feb-1996|    Time and Date
  SS1)  |Status:OK           |    Status
  SS2)  |Status:OK    -<--->-|
  XX )  |                    |    Blank Line
Choices are: TB EP1 EP2 EP3 PP PV EV1 EV2 CMD RTS HRP LL TM SS1 SS2
Top display line template:    TB          New Value:
Choices are: TB EP1 EP2 EP3 PP PV EV1 EV2 CMD RTS HRP LL TM SS1 SS2
Second display line template: EP2         New Value:
Choices are: TB EP1 EP2 EP3 PP PV EV1 EV2 CMD RTS HRP LL TM SS1 SS2
Third display line template:  EV2         New Value:
Choices are: TB EP1 EP2 EP3 PP PV EV1 EV2 CMD RTS HRP LL TM SS1 SS2
Bottom display line template: SS2         New Value:
Show hundredths digit in AZ position: NO  New Value:
Show hundredths digit in EL position: NO  New Value:
Show hundredths digit in AZ velocity: NO  New Value:
Show hundredths digit in EL velocity: NO  New Value:


Customer-specific Setups
------------------------
Output serial TAG lines: YES                        New Value:
  Choose: IP-A IP-B IP-C IP-D IP-E IP-F
  Slot for IP-SERIAL interface module: IP-D         New Value:
  Baud rate of the serial TAG data:  9600           New Value:
Use Kavouras 'TAC' for Control/Status: YES          New Value:
  Choose: IP-A IP-B IP-C IP-D IP-E IP-F
  Slot for IP-SERIAL interface module: IP-D         New Value:
  Baud rate of the 'TAC' interface:  9600           New Value:
  BITE ID of resent 'TAC' packets:   3(decimal)     New Value:
Use Andrew-Kintec serial pedestal interface: YES    New Value:
  Choose: IP-A IP-B IP-C IP-D IP-E IP-F
  Slot for IP-SERIAL interface module: IP-D         New Value:
```

```
  Baud rate of Andrew interface: 19200             New Value:
  Choose: None Odd Even
  Serial I/O byte parity: Odd                      New Value:
  ID of Andrew BITE status packets:   2(decimal)   New Value:
  Map Andrew status into S[29:63] variables: NO    New Value:
  Apply boresite offsets to parallel angles: YES   New Value:
    Offsets are subtracted from angles: NO         New Value:
Num of auxiliary hardware status input bytes: 6    New Value:
Num of auxiliary hardware control output bytes: 2  New Value:
  Choose: IP-A IP-B IP-C IP-D IP-E IP-F
  Slot for first IP-DIGITAL-48 module: IP-A        New Value:
  Choose: IP-A IP-B IP-C IP-D IP-E IP-F
  Slot for second IP-DIGITAL-48 module: IP-C       New Value:
HPIB bus interface hardware is installed: YES      New Value:
  HPIB Address of RCP8 (system controller):  1    New Value:
  RF/IF Signal generator is on the bus : YES       New Value:
    HPIB Address of the signal generator: 19       New Value:
    Signal generator has pulse modulation : NO     New Value:
Generate trigger sector blanking output: YES       New Value:
  Choose: None TrPwr SvPwr RdOff Reset IRS0 IRS1 IRS2 PW0 PW1 Rly AZ0
  Hardware output line to use: TrPwr               New Value:
  Choose: None TrPwr MagCr ILock Air WGPrs IRS0 IRS1 IRS2 PW0 PW1
  Hardware  input line to use: None                New Value:
  Include sector #1 in overall test: YES           New Value:
    Sector #1 uses pedestal angles: YES            New Value:
    Sector #1 lower azimuth:     0 deg             New Value:
    Sector #1 upper azimuth:    30 deg             New Value:
    Sector #1 lower elevation:  10 deg             New Value:
    Sector #1 upper elevation:  20 deg             New Value:
  Include sector #2 in overall test: NO            New Value:
  Include sector #3 in overall test: NO            New Value:
  Include sector #4 in overall test: NO            New Value:
  Include sector #5 in overall test: NO            New Value:
  Include sector #6 in overall test: NO            New Value:
  Include sector #7 in overall test: NO            New Value:
  Include sector #8 in overall test: NO            New Value:


RCP>site host
Host Computer Setups
--------------------
Choices are: 1200 2400 9600 19200
Baud rate for host computer I/O: 9600            New Value:
Choices are: XMT01 XMT02 XMT04
Data format transmitted by host computer: XMT02  New Value:
Choices are: RCV01 RCV02 RCV03 RCV04
Data format received by host computer: RCV03     New Value:
Process incoming servo control packets: YES      New Value:
Maximum signal generator power level:    0 dBm   New Value:
RCP8 transmission rate:  2.50 records/sec       New Value:
RCP8 transmits Time-of-Day records: YES         New Value:
  Time between Time-of-Day records :  30 sec     New Value:
RCP8 transmits internal BITE packets: YES       New Value:
  ID of internal BITE packets:   0(decimal)      New Value:
RCP8 transmits AUX status BITE packets: YES     New Value:
  Xmt ID of status  BITE packets:   1(decimal)   New Value:
```

```
RCP8 receives AUX control BITE packets: YES     New Value:
  Rcv ID of control BITE packets:   1(decimal)   New Value:
Dead–Host–Computer detection time: 5.0 sec       New Value:
Choices are: LOCAL–TTY REMOTE–HOST–CHAT
Default user interface: LOCAL–TTY                New Value:


RCP> site misc
Miscellaneous Setups
--------------------
Lower EL limit switch causes shutdown: YES        New Value:
Upper EL limit switch causes shutdown: YES        New Value:
Present year: 1996                                New Value:
```

"AXIS" Command Summary Dialog

```
RCP> axis azimuth
AZIMUTH Axis Parameters
-----------------------
Use internal antenna simulator: NO                    New Value:
Use tachometer voltage to estimate velocity: YES      New Value:
  Tachometer calibration -- Level:  50.00 T–Units     New Value:  \ If
  Tachometer calibration -- Speed:  12.00 deg/sec     New Value:  / Yes
  Virtual Tach -- Full scale speed:  24.00 deg/sec    New Value:  \
  Virtual Tach -- Differentiation window: 0.50 sec    New Value:   \ If
  Virtual Tach -- Minimum travel: 0.05 deg/window     New Value:   / No
  Virtual Tach -- Use antenna model predictor: YES    New Value:  /
Enforce soft limits of position travel: NO            New Value:
Enforce shutdown limits of position travel: NO        New Value:
Force shutdown if tach/pos are inconsistent: YES      New Value:
  Permissible fixed error:     1.50 deg/sec           New Value:
  Permissible relative error: 10.00 %                 New Value:
Force shutdown for unresponsive antenna: YES          New Value:
  Permissible tach prediction error: 15.0 T–Units     New Value:
  Maximum duration of such error:  3.0 sec            New Value:
Moment of Inertia:  6.00 (D–Units / T–Units/sec)      New Value:
Enforce model–based acceleration limits: YES          New Value:
  Maximum acceleration:   6.0 deg/sec/sec             New Value:
Choices are: PARALLEL SYNCHRO
Angle input signal source: PARALLEL                   New Value:
Number of bits for angle input: 16                    New Value:
Input offset from true orientation:   0.00 deg        New Value:
Angle  inputs are active high: YES                    New Value:
Angle outputs are active high: YES                    New Value:
Angle  input format is BCD: NO                        New Value:
Angle output format is BCD: NO                        New Value:
Drive voltage is positive for positive motion: YES    New Value:
Tach voltage is positive for positive motion: YES     New Value:
Drive output offset:       0.00 D–Units               New Value:
Tachometer input offset:   0.00 T–Units               New Value:
```

"VSERVO" Command Summary Dialog

```
RCP> vservo azimuth
AZIMUTH Velocity Servo parameters
---------------------------------
Motor positive sustaining drive:  15.00 D-Units     New Value:
Motor negative sustaining drive: -15.00 D-Units     New Value:
Nominal positive drive slope:   0.800 D/T-Units     New Value:
Nominal negative drive slope:   0.800 D/T-Units     New Value:
Velocity feedback slope:    25.000 D/dT-Units       New Value:
Velocity feedback deadzone:      0.10 T-Units       New Value:
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:
Maximum absolute velocity:      95.00 T-Units       New Value:
Velocity shutdown safe margin:   4.00 T-Units       New Value:
Velocity shutdown check time:  1.00 sec             New Value:
Tach filter time constant:  0.025 sec               New Value:
Drive filter time constant: 0.025 sec               New Value:
Drive slew rate limit for Zero-->Max: 0.100 sec     New Value:
```

"PSERVO" Command Summary Dialog

```
RCP> pservo azimuth
AZIMUTH Position Servo parameters
---------------------------------
Hysteresis inner zone:      0.020 deg        New Value:
Hysteresis outer zone:      0.050 deg        New Value:
First position break point:  1.00 deg        New Value:
Second position break point: 5.00 deg        New Value:
First interval slope: 12.00 (T-Units)/deg    New Value:
Second interval slope: 3.00 (T-Units)/deg    New Value:
Third interval slope:  1.00 (T-Units)/deg    New Value:
```

"CONTROL" Command Summary Dialog

```
RCP> control lines
Antenna/Transmitter/Receiver Control Output Lines
-------------------------------------------------
PULSE-WIDTH-0 output active high:  NO         New Value:
PULSE-WIDTH-1 output active high:  NO         New Value:
RADIATE-ON output active high:  NO            New Value:
RADIATE-OFF output active high:  NO           New Value:
Choices are: LEVEL PULSE
RADIATE control protocol: PULSE               New Value:
RADIATE On/Off pulse duration: 0.75 sec       New Value:
SERVO-POWER output active high:  NO           New Value:
T/R-POWER output active high:  NO             New Value:
SIGGEN-ON output active high:  NO             New Value:
SIGGEN-CW output active high:  NO             New Value:
NOISE-GEN-ON output active high:  NO          New Value:
SERVO-CABINET-RELAY output active high:  YES  New Value:
IRIS-MODE output active high:  YES            New Value:
SYSTEM-RESET output active high:  NO          New Value:
```

"CONTROL Variables" Command Summary Dialog

```
Boolean Variable Configuration
------------------------------
Choose: Retrig Single Filter Retard Extend Clock
Timer #0 trigger mode: Retrig                  New Value:
Timer #0 period/delay:   1.0 sec               New Value:

Choose: Retrig Single Filter Retard Extend Clock
Timer #1 trigger mode: Retrig                  New Value:
Timer #1 period/delay:   2.0 sec               New Value:

Choose: Retrig Single Filter Retard Extend Clock
Timer #2 trigger mode: Retrig                  New Value:
Timer #2 period/delay:   5.0 sec               New Value:

Choose: Retrig Single Filter Retard Extend Clock
Timer #3 trigger mode: Retrig                  New Value:
Timer #3 period/delay:  10.0 sec               New Value:

Choose: Retrig Single Filter Retard Extend Clock
Timer #4 trigger mode: Retrig                  New Value:
Timer #4 period/delay:  20.0 sec               New Value:

Choose: Retrig Single Filter Retard Extend Clock
Timer #5 trigger mode: Retrig                  New Value:
Timer #5 period/delay:  30.0 sec               New Value:

Choose: Retrig Single Filter Retard Extend Clock
Timer #6 trigger mode: Retrig                  New Value:
Timer #6 period/delay:  60.0 sec               New Value:

Choose: Retrig Single Filter Retard Extend Clock
Timer #7 trigger mode: Retrig                  New Value:
Timer #7 period/delay: 120.0 sec               New Value:

Minimum velocity for 'antstop':  0.50 deg/sec  New Value:
Minimum time for 'antstop':  2.00 sec          New Value:
```

"STATUS" Command Summary Dialog

```
RCP> status
Antenna/Transmitter/Receiver Status Input Lines
-----------------------------------------------
LOCAL input enabled:  NO                  New Value:
STANDBY input enabled:  NO                New Value:
INTERLOCK input enabled:  NO              New Value:
MAGNETRON-CURRENT input enabled:  NO      New Value:
AIRFLOW input enabled:  NO                New Value:
WAVE-GUIDE-PRESSURE input enabled:  NO    New Value:
PULSE-WIDTH-0 input enabled:  NO          New Value:
PULSE-WIDTH-1 input enabled:  NO          New Value:
RADIATE input enabled:  NO                New Value:
SERVO-POWER input enabled:  NO            New Value:
LOWER-LIMIT-SWITCH input enabled:  NO     New Value:
UPPER-LIMIT-SWITCH input enabled:  NO     New Value:
T/R-POWER input enabled:  NO              New Value:
SIGGEN-ON input enabled:  NO              New Value:
SIGGEN-CW input enabled:  NO              New Value:
SIGGEN-FAULT input enabled:  NO           New Value:
NOISE-GEN-ON input enabled:  NO           New Value:
IRIS-MODE input enabled:  NO              New Value:
SYSTEM-RESET input enabled:  NO           New Value:
```

"INU" Command Summary Dialog

```
RCP> inu
Inertial Navigation Unit (INU) Setups
-------------------------------------
Choose: Honeywell-INU Seapath-MRU
Reference unit type: Honeywell-INU                  New Value:
Choose: IP-A IP-B IP-C IP-D IP-E IP-F
Slot for IP-SERIAL interface module: IP-D           New Value:
Use platform stabilization algorithms: YES          New Value:
Negate sign of Roll angles:    NO                   New Value:
Negate sign of Pitch angles:   NO                   New Value:
Negate sign of Heading angles: NO                   New Value:
Roll offset from true orientation:      0.00 deg    New Value:
Pitch offset from true orientation:     0.00 deg    New Value:
Heading offset from true orientation:   0.00 deg    New Value:
Lead time for velocity extrapolation: 0.050 sec     New Value:
Dead INU detection time:  5.0 sec                   New Value:
CRC Generator/Checker preset to zero: NO            New Value:
Ignore CRC errors in INU data stream: NO            New Value:
Choose: Off Internal External
Built-in INU Simulation: Internal                   New Value:
Amplitude of motion for Roll axis:     12.00 deg    New Value:
Amplitude of motion for Pitch axis:     8.00 deg    New Value:
Amplitude of motion for Heading axis:  80.00 deg    New Value:
Center of motion for Roll axis:      0.00 deg       New Value:
Center of motion for Pitch axis:     0.00 deg       New Value:
Center of motion for Heading axis:   0.00 deg       New Value:
Period of motion for Roll axis:     15.0 sec        New Value:
Period of motion for Pitch axis:    13.0 sec        New Value:
Period of motion for Heading axis:  60.0 sec        New Value:
```

4.3 The "SAVE" and "RESTORE" Commands

Use the "SAVE" command to store the current RCP8 parameters in the non-volatile RAM. This will automatically preserve the settings the next time the RCP8 is powered up. The "SAVE" command prints an informational message that counts the actual number of bytes that were changed.

The "RESTORE" command is used to replace the current working parameters with a completely different set. With the "factory" argument, the command restores conservative default values for all parameters. This can be useful to return to a known baseline. With the "saved" argument, the command restores the parameters from the most recent "SAVE" command. The "undo" argument allows you to change your mind if the "factory" or "saved" option has inadvertently overwritten the actual desired settings.

4.4 The "SITE" Command

The SITE command is used to configure parameters for the local site. The following options represent the possible sub-menus:

- "Display" option — defines how the vacuum-fluorescent, front panel display is configured and represents the default menu if the SITE command is invoked with no argument.
- "Host" option — defines the communication choices and data protocols that are used with the host computer.
- "Custom" option — selects and configures customer specific features.
- "LOG" option — setup internal data and event logger
- "Miscellaneous" option — defines a few other things.

4.4.1 Front Panel Display Setups

These setup questions are accessed by typing "Site Display" at the "RCP>" prompt.

Front panel display update time: 0.200 sec

This shows the time in seconds, between updates, of the front panel display. Set the time to a rate that is comfortable for viewing.

Front panel display brightness (0–7): 4

The brightness of the display is controlled in this setup. For maximum tube life, choose the minimum brightness that still permits easy viewing.

Top display line template: TB
Second display line template: EP2
Third display line template: EV2
Bottom display line template: SS2

The RCP8 permits a flexible reconfiguration of the front panel display, so the parameters, which are most relevant for each site, can always be viewed. For example, if the RCP8 is on a moving ship, the user may wish to display both the pedestal and the Earth angles, as well as the Roll, Pitch, and Heading of the ship. For terrestrial radars, it may be desirable to use the available display lines for angular velocities, status bits, and monitoring of host computer commands.

There are four 20-character lines available on the front panel display. Each line can assigned to carry any of the following templates and may be arranged in any order.

```
 TB )  |--AZ--        --EL--|    Title Bar
 EP1)  |123.45  Earth  23.46|    AZ/EL Earth Positions
 EP2)  |123.45   Pos   23.46|
 EP3)  |123.45  AZ/EL  23.46|
 PP )  |123.45   Ped   23.46|    AZ/EL Pedestal Positions
 PV )  | 12.00  Ped.V   0.00|    AZ/EL Pedestal Velocities
 EV1)  | 12.00  Geo.V   0.00|    AZ/EL Earth Velocities
 EV2)  | 12.00   Vel    0.00|
 CMD)  | 12.00 V Cmd P 10.50|    AZ/EL Servo Commands
 RTS)  |Rad:On T/R:-- Srv:On|    Radiate, T/R, and Servo
 HRP)  |123.4H -12.3R  23.1P|    Heading, Roll, Pitch
 LL )  |148 24.5W   42 35.2N|    Longitude / Latitude
 TM )  |17:22:35  3-Feb-1996|    Time and Date
 SS1)  |Status:OK           |    Status
 SS2)  |Status:OK    -<--->-|
 VAR)  | 0000 0000 0000 0000|    Local Variables
DRCP)  |Di/-- Ok/Ok --/On Au|    Dual/Redundant State
 XX )  |                    |    Blank Line
```

- **TB** — selects a title bar to label the AZ and EL columns.
- **EPn** — selects one of several AZ/EL Earth Position displays. The difference is in the text that appears in the center of the line. Use the "Earth" choice (EP1) to avoid confusion if the pedestal angles are also displayed. Use "Pos" if the Title Bar appears elsewhere on the display or if the user has little trouble remembering which is AZ and EL. Otherwise, use "AZ/EL."
- **PP** and **PV** — select pedestal position and velocity.
- **EVn** — selects one of several AZ/EL Earth velocities that contain different text labels in the center.
- **CMD** — displays the present command from the host computer. The letters "V" and "P" indicate if a position or velocity servo is requested on each axis.
- **RTS** — shows the present state of "Radiate," "T/R Power," and "Servo Power."
- **HRP**, **LL**, and **TM** — these are used only if INU data is available.

- **SSn** — selects one of several status lines. "SS2" includes an entertaining "heartbeat" indicator.
- **VAR** — shows the continuous states of the sixteen local variables V0–V15. This can be handy as a temporary live monitor of states assigned to these variables via logic equations.
- **DRCP** — shows the "Disabled", "Okay", and "Active" summary status of a Dual/Redundant system. The "A" and "B" systems are separated with a slash "/". Lastly, the A/B/Auto mode is shown at the right side.

Once a status template is assigned to any of the display lines, the RCP8 will use that line to display urgent status, such as a shutdown condition. If a status template has not been assigned, then the bottom display line will be overwritten whenever any urgent messages is displayed.

Show hundredths digit in AZ position: NO
Show hundredths digit in EL position: NO
Show hundredths digit in AZ velocity: NO
Show hundredths digit in EL velocity: NO

You may display either one or two digits to the right of the decimal point in the position and velocity readouts. Since the angle information for most antennas is only accurate to 0.1 degree, the defaults are to not show the hundredths digit.

User shutdown #1 text: 'User–Shutdown #1 '
User shutdown #2 text: 'User–Shutdown #2 '

The text to display for the two user-induced shutdown conditions can be selected here. You may type any 20-character string to pop-up and blink when these shutdowns are triggered.

4.4.2 Host Computer I/O Setups

These setup questions are accessed by typing "Site Host" at the "RCP>" prompt.

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: XMT02
Data format received by host computer: RCV02

The RCP8 can send and receive a variety of 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 the 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 via chat mode from the host computer. Previously the servos would resume responding to external control packets whenever the antenna was not explicitly under control of the chat interface, e.g., between monitor and setup command sequences. As a reminder, the RCP8 front panel will display "Lockout" on its status line whenever control packet commands are being ignored.

Maximum signal generator power level: 0 dBm

This question standardizes the encoding of power levels in the various RCVnn and XMTnn serial protocols. The I/O formats only allocate 7–bits for the integer-valued power levels (a 127-dB span). The maximum level specified here fixes that range on a particular absolute power interval. The RCP8 maximum level should match the maximum level set in the IRIS utility SETUP–>RCP–>Control & Support.

RCP8 transmission rate: 2.50 records/sec

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

1. providing the host computer with up-to-date angles, and
2. relieving the host computer of unnecessary I/O "flooding."

RCP8 transmits Time-of-Day records: YES
Time between Time-of-Day records : 30 sec

The user has the option of transmitting the INU's time-of-day to the host computer if the 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: 0(decimal)

The transmission of internal BITE status packets (See Table A–11) is decided by this question. These should be enabled if the host computer needs to monitor any of the information contained in these packets.

RCP8 transmits AUX status BITE packets: YES
Xmt ID of status BITE packets: 1(decimal)
RCP8 receives AUX control BITE packets: YES
Rcv ID of control BITE packets: 1(decimal)

The transmission and reception of auxiliary status and control BITE packets is selected here. See information at Table A–12.

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 the RCP8 will 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 the RCP8, so all 64 bits are always included in the 13–byte fixed–format BITE I/O packets. See related questions under "Site Custom" (Section 4.4.3).

Dead-Host-Computer detection time: 5.0 sec

This determines the required I/O inactivity time before the RCP8 disables the antenna motion. Once the RCP8 is under the control of the host computer, there is a possibility that the computer may "crash" or that the program, which interacts with the 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 via the chat-mode packets during the initial RCP8 power up. This question sets the power-up default.

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

4.4.3 Customer-Specific Site Setups

These setup questions are accessed by typing "Site Custom" at the "RCP>" prompt. They are used to select and configure customized features of the RCP8 that are added as the product evolves.

Output serial TAG lines: YES
Slot for IP–SERIAL interface module: IP–D
Baud rate of the serial TAG data: 9600

Normally the 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. You must supply an additional IP–SERIAL card, half of which will be used.

Num of auxiliary hardware status input bytes: 6
Num of auxiliary hardware control output bytes: 2
Choose: IP–A IP–B IP–C IP–D IP–E IP–F
Slot for first IP–DIGITAL–48 module: IP–A
Choose: IP–A IP–B IP–C IP–D IP–E IP–F
Slot for second IP–DIGITAL–48 module: IP–C

The RCP8 can support up to 64 additional TTL status inputs and/or control outputs (the total number is limited to 64). There must be room on the Platform–332 mother board to install an additional IP–DIGITAL–48 module for each group of 32 lines that

are to be added. The auxiliary status lines are reported to the host computer via a special BITE packet (See Table A–12) whose ID may be chosen. Similarly, the reception of this BITE packet (also with selectable ID) by the RCP8 is the mechanism for setting the auxiliary control outputs.

The hardware electrical lines are assigned to the numbered status and control variables beginning with zero. In the example above, the 48 status lines (six bytes) will be assigned to S[0:47], and the 16 control bits (two bytes) will be assigned to C[0:15]. These questions only define an optional association between C[0:63] and S[0:63], and external hardware lines supplied by additional IP–DIGITAL–48 module(s). See related questions under "Site Host" (Section 4.4.2).

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

Simply choose the serial port, and ID for the BITE and Q–BITE packets that will be 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. The built-in simulator can be used for debugging the main code, and 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 seventy 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 eight 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 will send a reset command to the physical TCU. In addition, a rising edge on Control Variable C63 will also reset 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 twiddle those two control bits the appropriate commands will be sent to the TCU. Likewise, status from the TCU will set the TrPower and Radiate status bits appropriately.

Use Andrew–Canada serial pedestal interface: YES
Serial port: io62–tty0
Get Pos/Vel from serial status packets: YES
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 . The RCP8 also contains a (minimal) serial simulation of a real Andrew ACU, which you can configure onto a TTY port for loopback testing.

Normally the RCP8 will receive high-speed parallel AZ/EL angles from the Andrew ACU. However, a setup question allows the RCP8 to use the low-speed (5Hz) serial angle status information from the ACU instead. This option can be handy for testing.

HPIB bus interface hardware is installed: YES
HPIB Address of RCP8 (system controller): 1
RF/IF Signal generator is on the bus : YES
HPIB Address of the signal generator: 19
Signal generator has pulse modulation : NO

Answer the first question "Yes" if an IP488 module is plugged into the IP–E slot of the RCP8's Platform 332 mother board. The RCP8 then acts as the system bus controller and is assigned a (rather arbitrary) default address of 1. If a signal generator is attached to the bus, then enter its HPIB address. Also tell whether it supports pulse modulation; answer "No" if the signal generator can only operate in CW mode.

Note that IEEE Std 488.2 specifies a command and query protocol that is independent of the hardware manufacturer. For this reason, it it not necessary to specify what brand of signal generator is being used (since the same minimal command set works with all 488.2 devices). The RCP8 initialization sequence reads the full identification string from the signal generator; and if you're curious, the "Help View" command will show its contents. This string contains the manufacturer's name, make and model numbers, serial number, software version, etc.

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

The RCP8 keeps the signal generator in its normal "local" mode at all times, and polls its settings every 0.5 seconds. This means that the signal generator's front panel is fully functional at all times. However, whenever the RCP8 detects a change in the host computer's requested settings, then those changes are sent immediately (but just

once) to the signal generator. The correct settings are thus put into place; though the user is still allowed to make further changes using the manual controls. The design philosophy is that the signal generator should simply appear to operate normally, except at those times when changes are requested by the host computer.

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

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

The RCP8 can generate a trigger blanking output whenever the antenna falls within one of eight user-defined solid sectors in azimuth and elevation. Choose the remapped output line that will 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 3.5ms. This latency is defined as the maximum time that can elapse between the antenna moving into or out of a blanked sector, and the RCP8's mapped hardware output line actually responding with that indication. The 3.5ms latency will only be realized when the mapped output line is AZ0 (LSB of the parallel azimuth output). All other output lines will run with a 29ms delay; as will any optional re-mapped input line that is fed into the blanking criteria. As an example, at a 36deg/sec rotation rate, the 29ms delay might have produced a 1.04deg shift in the location of the blanked sector. The 3.5ms delay would position the edge more precisely by introducing only a 0.13deg shift.

4.4.4 Data and Event Logging

An internal logging feature is provided in the RCP8 for recording unusual data and events during normal operation. You may choose the types of events to log, and then view the accumulated entries at any time. For now, the options allow for logging angle glitches and invalid INU parameters; but the RCP8 log is an expandable feature, and we expect to add many new types of entries in the future.

Data and Event Logging Setups
––––––––––––––––––––––––––
LOG glitches in AZ/EL output angles: YES
Maximum valid AZ/EL change: 1.00 deg / 23.3 ms
LOG invalid/reduced INU data records: YES

The angle glitch logger checks the AZ and EL output angles that are computed every 3.33ms, and makes a log entry if their change over an 8-sample interval is more than the maximum specified value. The log entry records the INU/Earth/Pedestal angle data for all eight samples, and then inhibits additional entries for the next seven samples (so that successive log entries will overlap nicely).

The following sample printout shows the AZ and EL Earth and Pedestal angles, followed by the Roll, Pitch, and Heading INU angles. If moving platform stabilization is not enabled, the printout is much simpler and only lists the AZ and EL pedestal angles.

```
AZ: 315.13  315.13   EL:  2.07   2.07   RPH:  4.72   –5.20  230.56
AZ: 315.25  315.25   EL:  2.07   2.07   RPH:  0.00    0.00    0.00
AZ: 315.25  315.25   EL:  2.02   2.04   RPH:  0.00    0.00    0.00
AZ: 185.84  315.25   EL:  2.37   2.04   RPH:  4.68   –5.17  230.64
AZ: 185.94  315.36   EL:  2.40   2.04   RPH:  4.68   –5.17  230.64
AZ: 185.94  315.36   EL:  2.37   2.01   RPH:  4.68   –5.17  230.64
AZ: 186.01  315.36   EL:  2.40   2.01   RPH:  4.63   –5.15  230.71
AZ: 186.12  315.48   EL:  2.42   2.01   RPH:  4.63   –5.15  230.71
```

When setting up the angle glitch logger, you should choose the maximum valid angle change according to the maximum scan speed that is expected; but it should never be less than the quantization of the incoming pedestal angles themselves (lest false alarms be constantly triggered). Because of this interaction, you are asked to express the maximum angle change directly as the angular change over a fixed period of time, rather than as a maximum speed. To compute this, simply multiply the maximum speed in deg/sec by 0.0233 sec, and round this angle up so that it at least exceeds the quantization of the incoming pedestal angles.

The INU data quality logger can be enabled to catch changes in the reported "Invalid" and "Reduced" flags for the attitude and motion parameters. Each log entry consists of the flag word, and the current Roll/Pitch/Heading. A new entry is made whenever any bits in the flag word change.

This sample printout shows the Reduced/Invalid flags for Horizontal, Vertical, Heading, and combined Roll and Pitch data. A zero indicates "okay". The actual Roll, Pitch, and Heading angles reported from the same INU record are also shown.

```
Hor:0/0  Vrt:0/0  Hed:0/0  Rol:0/0      RPH:–10.42    0.03  249.32
```

4.4.5 Miscellaneous Site Setups

These setup questions are accessed by typing "Site Miscellaneous" at the "RCP>" prompt.

Lower EL limit switch causes shutdown: YES
Upper EL limit switch causes shutdown: YES

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

Present year: 1996

The INU's time-of-day format does not include the year, hence it must be specified in this area. While the RCP8 is operational this value is automatically incremented if the year changes This will ensure that the correct time packets are sent to the host computer. It is necessary to save the incremented value, using the "SAVE" command, to establish the permanent change. All non-saved years will flash on the front panel's "TM" display.

4.5 The "AXIS" Command

This command defines many parameters pertaining to the azimuth or elevation axes. All such definitions, that are not specifically related to the velocity or position servos, are specified here.

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.

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

Angle input signal source: Synchro
Synchro reference frequency: 60 Hz
Shutdown for invalid synchro voltages: YES
Angle offset from true orientation: 0.00 deg
Use tachometer voltage to estimate velocity: NO
Synchro Tach –– Full scale speed: 80.00 deg/sec

The RCP8 now 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 the 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 two questions establish that a 60Hz synchro will be used for angle (position) input on this axis. Moreover, the voltages on the two "Ref" and three S1/S2/S3 lines will be checked for validity, and the RCP8 will shutdown 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 zero 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 not available, an excellent 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 zero position error at any velocity whenever the acceleration is zero. The internal velocity that is maintained during this process can be used by the RCP8 in place of a physical tachometer. When doing this, you choose the velocity that will correspond to 100 T-Units of virtual tachometer level. Simply 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 backpanel's Molex SYNCHRO connector for each value of plug-in SIP resistor. The AZ channel voltages are set by SIP S1, whereas S2 sets the EL voltage levels. These resistors are socketed, and can be changed by removing the back cover of the IO62–CP panel

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 three '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 backpanel therefore biases the gains so that the 'S' voltages can be made as large as possible, i.e., 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 90Vrms 1:1 synchro would best use the 150K resistor, whereas a 105Vrms unit would require the 220K value. Note that you can check for proper A/D conversion levels of the synchro inputs using the 'help view' menu of the RCP8.

Important: The synchro voltage input feature is only available on Rev.B and higher backpanels. If you are running an RCP8 with a Rev.A backpanel and would like to switch to synchro inputs, SIGMET will be happy to upgrade your panel at no cost.

Angle input signal source: PARALLEL

You may choose either TAG angle inputs (by typing "PARALLEL"), or 90V, 60Hz, 3–wire synchro angle inputs (by typing "SYNCHRO"). Other synchro voltages and frequencies are available by special order.

Number of bits for angle input: 16

The parallel antenna position inputs are TTL levels. The number of bits used to represent an angle will vary from site to site, depending on the style of encoder and associated circuitry used by the antenna. The RCP8 supports up to 16-bits of binary angle and 4-digit BCD angles. For the binary angles, if all sixteen 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 $1/10^{o}$, the lowest 14 bits are used.

When synchro angle inputs are used this question still has some importance. The S/D converter within the RCP8 can produce up to sixteen bits, but its ability to track a moving antenna is improved if fewer bits are demanded from it. The 14–bit setting will work well in most installations; but as few as 12–bits may be appropriate in rapid scanning applications.

Input offset from true orientation: 0.00 deg

Use this offset value if the pedestal angle positions reported to the RCP8 are biased. You may correct errors as large as $\pm 180^{o}$.

Angle inputs are active high: YES
Angle outputs are active high: YES

The antenna position input and output bits can be either active low or active high. This means that a high-level TTL signal can represent either a zero or a one.

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 input format is BCD: NO
Angle output format is BCD: NO

The position encoding can be either a binary angle or a Binary Coded Decimal (BCD). Binary angles are binary numbers in which the most significant bit has a weight of 180^{o}. The next bit has a weight of 90^{o}, and so forth. The representation precision of a sixteen-bit binary angle is $+0.005^{o}$ and the precision of a twelve-bit angle is $+0.09^{o}$. On the other hand, the BCD code represents angles, in tenths of a degree, in which the word is broken into four 4-bit fields. Within each field, there is a digit in the range of zero to nine. The concatenation of these digits, from MSB to LSB, forms the base-ten representation of the angle. For example, the BCD value 1024 represents 40.0^{o} and 1042 represents 41.2^{o}.

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 the RCP8 that a real tachometer is available on this axis, and that it's voltage should be used in all instances where velocity feedback is required.

The RCP8's 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.

Absolute velocity calibration is easily performed using the local TTY angle monitor. 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 best 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
Virtual Tach –– Full scale speed: 24.00 deg/sec
Virtual Tach –– Differentiation window: 0.50 sec
Virtual Tach –– Minimum travel: 0.05 deg/window
Virtual Tach –– Use antenna model predictor: YES

Answering "No" to the first question informs the 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 10ms and operates in two 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 nicely behaved instantaneous velocity estimate that is W/2 seconds old (from the center of the window). The "niceness" comes from having a large number (100W) 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. The 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.

There are four setup questions that configure the Virtual Tachometer. The first sets the actual velocity in degrees/second that 100 T–Units will represent. This number should be set 20% greater than the fastest anticipated rate of rotation. Do not make it unnecessarily large, as this will introduce quantization errors in the Virtual Tach units.

The second question defines the width of the position history window. The 0.5 second default will be appropriate in almost all cases. Making it larger will produce smoother drive voltages at low scan speeds, but at the expense of greater errors in extrapolating the phase delay. The fourth question permits the extrapolation model to be switched OFF for testing purposes, but it should always remain ON during normal operation.

The third question allows you to minimize the effects of noise in the least significant bits of the antenna positions. It 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 will ensure that a tachometer value of zero is produced whenever the antenna is genuinely at rest.

Enforce soft limits of position travel: NO
Minimum soft limit of travel: –1.00 deg
Maximum soft limit of travel: 91.00 deg

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

Note: The angle span defined by these limits may be any clockwise sector as large as 359^{O}.

Enforce shutdown limits of position travel: NO
Minimum shutdown limit of travel: –3.00 deg
Maximum shutdown limit of travel: 93.00 deg

These travel limits represent the hard bounds between where the antenna must lie. If the angle is observed to be outside of these limits, the RCP8 will shutdown 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.

Note: The angle span, defined by these limits, may be any clockwise sector as large as 359^{O}.

Force shutdown if tach/pos are inconsistent: YES
Permissible fixed error: 1.50 deg/sec
Permissible relative error: 10.00 %

The 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, the 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 Monitor's "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.

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

Important: 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%

The RCP8 servos can operate under the constraint of bounding the maximum acceleration (on each axis) that the antenna will experience. This acceleration limiter is based on the RCP8's 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.

Important: Since the acceleration limiter is based on the RCP8's 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 willing to extend the allowable drive interval down to include zero volts. To insure 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 zero volts. This meant that zero 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 would lead to gear strain as the antenna coasted to a stop from high speed under zero 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%, i.e., the allowable drive interval is extended all the way to zero. A value of 0% would enforce 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 new 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

The 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 being assumed 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 will tend to droop to some neutral angle that puts that center of mass directly below that axis. In the above example, the neutral droop angle is 35 degrees, i.e., 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 simply $D \sin(P–N)$, where *D* is the drive that would be required to hold the axis 90-degrees 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 naturally returns to 35 degrees whenever 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 two degrees. The drive that would be required at 90 degrees offset is therefore *–28 / sin(2–35)* = 51.4 D-Units, which we then enter into the second setup question.

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 whenever the position increases with time. If the sign is incorrect, then toggle this variable.

Drive output offset: 0.00 D–Units

Use this offset value if the servo power amplifiers do not produce zero drive when zero voltage is applied. Do not use this value to attempt to compensate for asymmetric motor-drive requirements in the two 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. The 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 zero.

4.6 The "VSERVO" Command

Once the velocity servo parameters is set to an initial-guess value, it is important 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 via 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.

The following suggestions will assist to resolve problematic areas in the servo setup.

- If the servo is completely unstable, recheck the tach sign to ensure the positive velocities correspond to the positive position increments.

- If the tach sign is correct, then observe the drive sign. If velocity overshoots are observed, the tachometer and drive filtering time constants may be too long. If the antenna is sluggish and does not quickly reach the desired velocity, then the feedback gain is too low.
- If the antenna chatters, or if the output drive oscillates around its mean value, the feedback gain is probably too high.
- If the equilibrium velocities differ slightly from the requested rates, such as request 50 but get 51, the 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.

The following values can be determined by using the local TTY control.

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

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

Note: 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 virtually all values of this parameter however, if the value is too small, the motion will be sluggish with relatively large errors in the final

achieved velocity. If the value is too large, the currents will 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 will bring the antenna rapidly to the requested velocities without excessive drive oscillation around its equilibrium value. If a scope is not available, the user can also make a fair judgement by observing the drive values that are displayed on the TTY. The feedback slope has units of Drive/TachError; typical values range from 10 to 200.

Velocity feedback deadzone: 0.10 T–Units

A deadzone is built in to 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 to 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:

The 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 will run 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 will then be obtained with $S = S_u / 2.2$, and $T_0 = 2.2\ P_u$.

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 will help reduce brief overshoots that can be caused by the integral feedback.

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 will be 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 converter's 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

The RCP8 has provisions to shutdown whenever the observed velocity on either antenna axis exceeds the internal maximum velocity limits that are enforced by the velocity servo. In the event that 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 will be setup to ensure that overshoots will 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 will occur whenever 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 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 antenna's 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 will become unstable and will oscillate around the desired velocities before settling. If the time constant is too small, then no significant smoothing or spurious rejection will be 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.

Important: The following "drive filter time" must be fine-tuned concurrently with the tachometer filtering. Also, both should be tuned whenever 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 user's 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 will have 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 zero to 100. For example, a value of 0.2 seconds would restrict 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 since, 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 the 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 zero, and internal (model based) acceleration limiting should be disabled. Acceleration limiting can be accomplished instead using the RCP8's 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 will be immediately adjusted in order to stop before that limit is reached.

4.7 The "PSERVO" Command

The position servo parameters can be set up using a few iterations of the following procedures:

1. Make sure the velocity servo has been thoroughly checked, in accordance with the previous section, since the position servo cannot work properly with incorrect settings.
2. Set the inner and outer zones of the hysteresis as described below.
3. 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.
4. Choose a larger "second position break point" and find the largest "second interval slope" that accomplishes larger steps without overshoot.
5. Find the largest "third interval slope" that permits large steps of any size to be travelled without overshoot.

After setting the position servo variables, it is important to exercise the servo on each axis to check for proper behavior. Attempt to move the antenna using the local TTY "ap" and "ep" commands. Verify that any position step can be requested without overshooting the final mark. Also, verify that very small steps cause antenna motion to occur. If the position feedback curve is not correct, it is possible for the servo to work properly for some step sizes but not for others. Therefore, a range of steps should be tested.

The procedure, described in the previous paragraph, is an attempt to tune the antenna for maximum performance (i.e., the antenna will arrive 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 will usually eliminate any position overshoot, with a slight performance penalty.

Hysteresis inner zone: 0.020 deg
Hysteresis outer zone: 0.050 deg

These represent the position errors, in degrees, within which the position servo will not attempt to correct the antenna's location. Two values are specified—one for the lower limit and one for the upper limit. Whenever the actual position error is less than the lower limit, the position servo will not drive the antenna (i.e., it will request zero velocity from the velocity servo.) Likewise, whenever the actual position error is greater than the upper limit, the servo will always drive 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 servo's state between the limits is whatever it was at the time the limits were entered.)

It is important that the inner limit be 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%.

First position break point: 1.00 deg
Second position break point: 5.00 deg

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

The following three 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.

4.8 The "CONTROL" Command

4.8.1 Output Line Configuration

These setup questions are accessed by typing "Control Lines" at the "RCP>" prompt. The command prompts the user with a series of similar questions which choose the polarity of the control lines that are driven by the RCP8. Each control line can be either active-high or active-low.

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

1. Complementary levels that are either "ON" or "OFF," according to the current radiate request.
2. 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.

4.8.2 Logic Equation Control Qualifiers

These setup questions are accessed by typing "Control Logic" at the "RCP>" prompt. The questions are actually a series of up to 32 logic equations that allow the user to qualify the RCP8 control functions in a very general way. Control bits can be

modified according to any logical combination of control bits, status bits, and special internal local variables. The result is that the RCP8 can be programmed to perform custom safeguards and implement special feedbacks that are specific to each site.

Each equation consists of a control variable on the left that is assigned to some 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. Parentheses may be included anywhere, as needed. New equations and subcommands are typed following the special "—>" prompt that is printed below the current equation. An example of how a few logic equations might appear is shown below.

```
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 will be TRUE whenever any of those status lines are true. The second equation then uses a TRUE sense of V0 to force the "servo power" control line FALSE. Thus, servo power will only be on if it is requested to be on, and if none of the three control lines are asserted. Likewise, EQ02 allows the external local/remote relay to be forced on by the auxiliary control line C10. Finally, 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 some logical combination of any of the above control variables, plus any of the status variables: 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[0:63]. The numbered status variables S[0:63] 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[0:63]. 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.
- It is also possible to 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, i.e., the value being 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 special symbols TRUE and FALSE may be used on the right side of an equation to indicate "always" and "never".

- The maximum number of different variables on the right side of an equation is four. If you need to combine more than four variables into an expression, then make use of multiple equations using local variables to combine all of the bits together. For example, to force servo power off whenever any of the first eight 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, whereas those appearing on the left-hand-side reflect the final qualified state. Thus, 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 convention is very handy, and allows you to use control variables on the right without ambiguity. For example, the pair of equations: "EQ00: cpw0 = cpw1" and "EQ01: cpw1 = cpw0", would swap the two pulsewidth 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". Be sure to use parentheses as needed to express your equations properly.

- Order of evaluation of the set of equations is from EQ00 to EQ31. 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 will have 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 that is not what you had intended to do. Spurious local variable warnings can be eliminated by choosing a unique set of numbered variables to use (i.e., 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 will be 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 will still be shown in the editor, but will be prefixed with the "#" character. This feature is very handy when you want to temporarily disable some trusted and debugged equations without the risk of retyping them incorrectly later.

Some additional notes on the usage of variables should also be kept in mind:

- A side effect of making assignments to local variables is that these bits are also reported to the host computer via the RCP8's Internal BITE packet. This opens many possibilities for designing customized status bits that could be monitored and reported by the IRIS BITEX utility.
- The variables "shut1" and "shut2" cause a user-induced shutdown if they are set to TRUE. Thus, you may now write logic equations that force a shutdown of the RCP8 when certain conditions are present. The two 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 whenever the RCP8 is shutdown. For example, adding the equation "`CSERVO = CSERVO & !SHUTDOWN`" will force the servo power off whenever the RCP8 is shutdown.
- The four status variables "usr0", "usr1", "usr2", and "usr3" correspond to the four input lines at pins 1–4 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 the 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 will show up in bit #6 of byte #12 as a result of this assignment.
- The status variable "unsafe" is TRUE if the RCP8 is in the temporary unsafe mode following a "reset" command. The status variables "ovel_az" and "ovel_el" are TRUE whenever the velocity on the corresponding axis exceeds the maximum value setup from the "velocity" command.
- The eight control variables **csgen0** through **csgen7** represent bits zero through seven 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 two 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 Section 4.8.4):
  - When **trig_blank** is FALSE and **trig_normal** is FALSE, the trigger is blanked whenever the antenna is within one of the designated sectors. This is the normal operating mode. You can disable all eight 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, the 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. Thus, when the equation is redisplayed, the RCP8 must recreate a printed line of text from this internal compiled representation. As a result, the equation that is echoed back will not always look the same as the equation that was originally typed. It will be logically equivalent, of course, but the exact syntax may be different. Some examples follow:

```
  --> 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 two negations" is printed as the "negation of two OR's". This type of transformation will often be observed, since the RCP8 attempts to minimize the number of "!" operators in its synthesized expressions.

The following list of subcommands 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
```

The first three subcommands are used to move and edit the list of equations. "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.

The "!" subcommand replaces the current equation with its logical negation. Some examples are:

```
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 two 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 "!" subcommand is used. This can be very helpful when testing individual control lines for simple ON/OFF response.

Before the actual 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 first question allows the entire set of equations to easily be switched ON/OFF, without having to change any of the equations themselves. Answering "NO" will leave the control functions unmodified (direct control from the host computer); whereas "YES" will apply all of the logical constraints.

Unlike all other RCP8 setup menus, the logic equation editor is a live menu. This means that each equation becomes active as soon as it is typed in. It then becomes easy to test individual control lines, and to edit the set of equations until the desired effects are obtained. Also, the "!" subcommand becomes a simple shortcut to perform a quick ON/OFF toggle test of any control line.

4.8.3 Logic Equation Timer Variables

A collection of software timer variables are supported for use with logic equations. Eight control variables are available whose names have the generic form "t*n_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 will be 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 mode that has been selected. The available modes are:

- **Retriggerable Pulse Generator** ("retrig")
  This timer generates a TRUE pulse whenever a FALSE→TRUE (rising edge) transition is applied to its input. Each rising edge continues to retrigger the output pulse, i.e., 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 anew once per second, the output pulse will never actually end. "Retrig" timers are handy 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 will cause the period to reset. Use it whenever you require an output pulse in response to any change in measured conditions, e.g., you could force "radiate" OFF briefly whenever the pulsewidth changed in either direction.

- **Single Pulse Generator** ("single")
  This timer generates a pulse similar to "retrig", except that an active pulse will not be 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 second, 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")
  This output of this timer attempts to follow its input, but with filtering and delay effects added. Whenever 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 zero, 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 will be honored right away (unlike the "filter" timer which always introduces a delay); but once honored, that new setting will once again persist 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, e.g., 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.

- **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 will be 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 merely 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")
  This timer produces a free-running clock having a specified period. The length of the timer's TRUE interval (and hence, the duty cycle) is also adjustable. The "clock" timers usually appear on the right side of equations, where they can supply any periodic input that the logic might require, e.g., to make a light blink, or to perform a periodic reset. But their phase can also be resynchronized to the start of their TRUE output interval by the application of a rising input edge.

4.8.4 Logic Equation Examples

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

Example #1

Suppose that an operational site requires that the radar transmitter be switched off whenever 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" will be 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
```

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

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 #2

Logic equations can help supply the necessary control signals for backing 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 will 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, whereas those from the RCP8 command line take the number of seconds as an argument, e.g., "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 #3

Logic equations can be used to supply the host computer with BITE information that would not ordinarily be available. The trick is to make an assignment to one of the first fourteen local variables, as those will then be transmitted via the RCP8's Internal BITE Packet. For example, adding the equation:

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

will send a "velocity overspeed" bit to the host computer in Bit #6 of Byte #12 (See Table A–11 for the mapping of the local variable bits).

Example #4

When writing sets of logic equations for the RCP8, keep in mind that assignments to most types of variables can not 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 will modify the variable's effective working value; but 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 pulsewidth control lines without the use of 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 two local variables, but instead, would leave both set to the original value of V1 (probably not useful).

Example #5

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 whenever the antenna stops; only when it stops within one of the protected sectors.

In summary, 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 whenever the antenna is moving. This is accomplished using the single equation:

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

For a related application in which we want to stop transmitting whenever the antenna becomes stationary, simply 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 right away.

4.8.5 Logic Equation Configuration of Variables

These setup questions are accessed by typing "Control Variables" at the "RCP>" prompt. Their purpose is to configure the internal variables that can be used within logic equations.

```
Choose: Retrig Single Filter Retard Extend Clock
Timer #0 trigger mode: Retrig
Timer #0 period/delay:   1.0
```

There are eight questions of this form, one for each of the eight available timers. Choose the mode of each timer (See Section 4.8.3), 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 TRUE whenever the antenna seems to be stopped, i.e., has been moving slower than a prescribed speed for more than a prescribed time. These setup questions configure these speed and time thresholds. Both the AZ and EL axes must appear to be stopped in order for "antstop" to be TRUE. Likewise, "antstop" is set FALSE whenever the antenna seems to be moving, i.e., the speed on either axis has exceeded the threshold speed for more than the specified time.

4.8.6 Analog Voltage Input Control Logic Variables

You may configure Boolean variables whose values are based on comparison tests of the eight analog voltage input lines. In this way, the analog inputs can be thresholded and used as additional inputs to logic equations within the RCP8. Up to sixteen such variables may be defined, i.e., you may have, on the average, two 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 will appear in the "**?v**" list of available status variables within the equation editor, and the variable may be used on the right side of any logic equation. The descriptive suffix is intended to make 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 will then 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, then the variable is TRUE; otherwise it is FALSE. Moreover, the input channels can be either added or subtracted when computing the sum.

In the above example, "a0_HiTemp" will be TRUE whenever 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 have created "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 could then be combined in a logic equation as follows:

```
EQ00: # V0 will be TRUE when the temperature is normal
 \--: v0 = !(a0_HiTemp | a1_LowTemp)
  -->
```

4.9 The "STATUS" Command

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

4.10 The "INU" Command

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

Use platform stabilization algorithms: YES

A "NO" response will disable all INU features and will inhibit all coordinate transformations between pedestal and Earth frames (the two frames are assumed identical.) None of the questions below will be displayed.

A "YES" response will enable the separate pedestal and Earth coordinate frames and prompt 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 it is necessary to 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 will occur 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 zero and 80ms.

Dead INU detection time: 5.0 sec

The RCP8 will stop performing coordinate transformations if the INU data stream is absent for more than this length of time. During the deadtime, all INU angles and velocities are artificially set to zero.

Built-in INU Simulation: External

The RCP8 contains an internal INU simulator that is very useful during program development as well as for testing simulated moving environments. Typing the response "OFF" will disable the simulator and suppress the remaining questions in this section. Typing the response "EXTERNAL" will result 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. Typing the response "INTERNAL" will run the same INU data simulation, but will internally loop it back into the RCP8 and will 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.