3. TTY Nonvolatile Setups

The RVP7 provides an interactive setup menu that can be accessed either from a serial TTY, or from the host computer interface. Most of the RVP7's operating parameters can be viewed and modified with this menu, and the settings can be saved in non-volatile RAM so that they take effect immediately on power-up. This permits custom trigger patterns, pulsewidth control, matched FIR filter specs, PRF, etc., to be configured by the user in the field.

The TTY menu also gives access to a collection of graphical setup and monitoring procedures that use an ordinary oscilloscope as a synthesized visual display. The burst pulse and receiver waveforms can be examined in detail (both in the time and frequency domain) and the digital FIR filter can be designed interactively to match the characteristics of the transmitted pulse.

3.1 Overview of Setup Procedures

This section describes basic operations within the setup menus such as making TTY connections, entering and exiting the menus, and saving and restoring the configurations.

The setup TTY should be plugged into the modular 6-pin phone jack located at the top edge of the RVP7 board. The electrical interface may be either RS232 or RS423. If the phone jack connection is inconvenient, the terminal may be wired directly to the TIOXMT and TIORCV signals on the P2 96-pin connector. The TTY should be configured for 7-bit or 8-bit data (the MSB is always zeroed), no parity, and either one or two stop bits.

With jumper JP4 in the "AB" position, the interface runs at 9600 baud; in the "BC" position the rate is 1200 baud (factory default), or some other rate set via the menu. Thus, the "AB" setting always makes a reliable 9600 baud connection, even if the the alternate rate is accidently set to a bad or forgotten value. Note: the reliable 9600 baud rate requires that the crystal located at X1 have a frequency of 4.9152MHz.

3.1.1 Initial Entry and Help List

The interactive setup menu is invoked by pressing the Escape key on the TTY. If that key can not be found on the keyboard, you can sometimes use Control "[" to generate the ESC code. The RVP7 then responds with the following banner and command prompt.

```
       SIGMET Incorporated, USA
RVP7 Digital IF Signal Processor Rev.A/01
-----------------------------------------
RVP7>
```

The banner identifies the RVP7 product, and gives the hardware version of the board (e.g., Rev.A) and software version (e.g., 01). This information is important whenever RVP7 support is required, and it is also repeated in the printout of the "**V**" command (See below).

The "**Q**" command is used to exit from the menus and to restart the RVP7 with the (possibly changed) set of current values. It is important to quit from the menus before attempting to resume normal RVP7 operation. Portions of the RVP7 command interpreter remain running while the menus are active (so that the TTYOP command works properly), but the processor as a whole will not function until the menus are exited.

From the command prompt, typing "**help**" or "**?**" gives the following list of available commands.

```
Command List:
  F: Use Factory Defaults
  S: Save Current Settings
  R: Restore Saved Settings
  M: Modify/View Current Settings
      Mb – Burst Pulse and AFC
      Mc – Board Configuration
      Mf – Clutter Filters
      Mp – Processing Options
      Mt<n> – Trigger/Timing <for PW n>
      Mz – Transmitter Phase Control
      M* – Stand–alone Settings
      M+ – Debug Options
  P: Plot with Oscilloscope
      Pb – Burst Pulse Timing
      Ps – Burst Spectra and AFC
      Pr – Receiver Waveforms
      P+ – Visual Test Pattern
  V: View Jumpers and Status
  ?: Cmd list (?? Settings list)
  *: Reboot <Max Slaves> <+>
  ~: Swap Burst/IF Inputs on IFD
  Q: Quit
```

3.1.2 Factory, Saved, and Current Settings

The *current settings* are the collection of setup values with which the RVP7 is presently operating; the *saved settings* are the collection of values stored in non-volatile RAM. The saved settings are restored (made current) each time the RVP7 is powered up. The "**S**" command saves the current settings into the non-volatile RAM, and the "**R**" command restores those non-volatile values so that they become the current settings. The "**F**" command initializes the current settings with factory default values. Thus, "**F**" followed by "**S**" saves factory defaults in non-volatile RAM, so that the RVP7 powers up in its original configuration as shipped.

The RVP7 retains all of its saved settings when new ROM upgrades are installed; the new version of code will automatically use all of the previous saved values. However, if the RVP7 detects that the new release requires a setup parameter that did not exist in the previous release, then a factory default value will automatically be filled in for that parameter. A warning is printed whenever this occurs (See also, Section 3.1.4).

There is also support for intermediate minor releases of RVP7 code. Each ROM has a major version number (the one that it always had), plus a minor version number for intermediate "unofficial" releases. The minor number starts from zero at the time of each "official" release, and then increments until the next "official" release. The RVP7 includes the minor release number (if it is not zero) in the printout of the "V" command. Likewise, the minor release number of the code that last saved the nonvolatile RAM is also shown. This is an improvement over having to check the date of the code to determine which minor release was running.

Note that the RVP7 does not actually begin using the current settings until after the "**Q**" command is entered, so that the processor exits the TTY setup mode and returns to normal operation.

3.1.3 Processor Reset Command

The "*****" command may be used to reset the signal processor from the TTY. This can be handy when the other methods of reset (power-up, parallel interface reset signal, or SCSI bus reset) can not easily be done. The command is robust in that pressing the Escape key followed by "*****", followed by two Returns, always resets the RVP7. There are certain wait conditions from which a TTY ESC does not immediately enter the setup monitor. However, the above four-key sequence always forces a full reset.

The RVP7 diagnostics can run in a continuous loop that is useful during production burn–in testing. In this mode the complete set of powerup tests is repeated approximately once per second. The green LEDs on the RVP7/Main and RVP7/AUX boards will blink on each run as a progress indicator. All red LEDs will initially be on, but each will begin to blink if any diagnostic ever fails on that board. A line of text is also printed to the setup TTY to show the progress of the tests and a summary of any errors.

The RVP7's Perpetual Diagnostic Loop maintains a histogram of receiver IF-Input noise levels in 1dB steps from –85dBm to –72dBm. You can view the accumulated noise distribution by typing "N" while the diagnostic loop is running. This feature is intended for use during factory burn-in and testing of RVP7/IFD units.

This special test mode can be started in two ways. One is to powerup the processor with the RVP7/Main I/O jumpers JP17–JP22 in the (somewhat illegal) pattern: JP17:BC, JP18:BC, JP19:AB, JP20:AB, JP21:AB, JP22:AB. This method has the advantage of not requiring a TTY connection. The second method is to reset the processor from the local TTY monitor using the "***+**" command. This is the normal reset command, but with a plus sign (debugging) suffix.

The reset command also takes an optional numeric argument to specify the maximum number of slave DSPs that are initially scanned for inclusion in the processing chain. The default value is 23, i.e., two RVP7/AUX boards having ten DSPs apiece, and three DSPs on the RVP7/Main board. This optional argument is not so much a field operational feature, but rather, is intended to help with production debugging.

Fine control is available of the probe for DSP chips that is performed during the RVP7's boot sequence. The goal is to be able to make use of a partially functioning RVP7/AUX board that might have failed during crucial field operations, and for which a spare is not available. If your system works properly with the RVP7/AUX board pulled out, you may be able to restore partial functionality as follows:

- With the RVP7/AUX board removed, set the "Number of Slave DSPs to use" to 3 using the setup question described in the M+ section (See section 3.3.8). Save this, power down the chassis, plug in the RVP7/AUX board, and power back up.

- Get into the RVP7's TTY monitor, and type "*4" followed by carriage return. This reboots the RVP7 using only four DSP chips, i.e., the three that are on the main board, plus the first one on the RVP7/AUX board. Hopefully this boot will succeed, indicating that some RVP7/AUX functionality can be salvaged.
- Continue trying "*5", "*6", etc. to add one more DSP chip each time, up to a maximum of 13. Eventually the boot will fail when the bad RVP7/AUX DSP chip is encountered.
- Finally, type in the count of working DSPs using the new setup question, save this value, and reboot the processor. The RVP7 will share the workload among the available DSPs. Be sure to remove this DSP constraint when you later install a repaired RVP7/AUX board. To make this easier to remember, IRIS will warn you for as long as you continue running with a diminished subset of DSP chips.

3.1.4 V — View Internal Status

The "**V**" command allows you to view some internal status within the RVP7. This information is available for inspection only, and can not be changed from the TTY. The view listing begins with the banner:

```
Jumpers and Internal Status
---------------------------
```

and then prints the following lines:

`Rev.B board, ROM V14.12 from Mon Jul 12 19:29:07 1999`

This line shows the revision level of the RVP7/Main board, the ROM code version, and the date and time that this release was compiled. This lets you know the age of the release, even if the release notes have been misplaced. The date can also be helpful in keeping track of "unofficial" interim releases.

`Values were last saved using ROM version V14`

This line tells which version of RVP7 code was the last to write into the non-volatile RAM. It is printed only if that last version was different from the ROM version that is currently running. The information is included so that a "smart upgrade" can often be done, i.e., values that did not exist in the prior release can be filled in with a guess that is better than merely taking the factory default.

`Warning: 3 automatic defaults were inserted.`

This warning will appear (accompanied by a beep) if one or more automatic factory defaults were required when the non-volatile RAM was last restored. It is likely that these automatic defaults will be acceptable operating values; but it would be wise to check the release notes to see what new parameters were added, and to decide on their proper settings. The warning will disappear once the **S** command is issued. This is because the missing saved slots are then filled in with valid values.

`Diagnostics: PASS   Slave DSP Count: 3`

If errors were detected by the powerup diagnostics then an error bitmask will be shown on the first line. The word "PASS" indicates that no errors were detected. The slave DSP count is also shown, which is the number of processors that were detected during the powerup sequence (and which will be used during subsequent processing). The RVP7 main board has three slave DSPs, and the each RVP7/AUX board supplies ten more. Up to two RVP7/AUX boards may be attached at the same time (23 slave DSPs total) for *extremely* intensive processing applications.

An itemized list (consisting of bit pattern and text) is printed whenever any of the powerup diagnostics fail. The possible messages that might appear are:

0x00000001 : No fiber downlink signal detected
0x00000002 : 16–Bit AFC level read/write
0x00000004 : IF Receiver reset request not sent
0x00000008 : I/O FIFO full before 4096 writes
0x00000010 : I/O FIFO not full after 4096 writes
0x00000020 : Transmit phase latch bits
0x00000040 : Downlink local counter test
0x00000080 : Receiver status bits & switches
0x00000100 : Test byte pattern from receiver
0x00000200 : Test word pattern from receiver
0x00000400 : Non–Volatile RAM 0x00 and 0xFF flags
0x00000800 : UART read/write check
0x00001000 : External RAM check
0x00002000 : SCSI controller chip error
0x00004000 : Range mask RAM and addressing
0x00008000 : I&Q FIFO interrupt & trigger flags
0x00010000 : I&Q FIFO data bits
0x00020000 : FIR processing of ramp pattern
0x00040000 : Boot words not accepted by first slave
0x00080000 : No reply slave DSP count
0x00100000 : Invalid count of slave DSPs
0x00200000 : Global communication port tests
0x00400000 : Internal tests failed on some slave
0x00800000 : Trigger Generator RAM and addressing
0x01000000 : Excessive coax/fiber round trip jitter
0x02000000 : No sync found in round trip test
0x04000000 : Internal error in compile/link

```
Host Computer Interface: SCSI  (ID = 4)
High-Byte-First (HP/IBM/SGI Style) I/O
```

The host computer interface summarizes the I/O parameters for the protocol that has been selected via jumpers on the main board. The type of interface (SCSI, DR11, GPIO, etc.), device particulars, and byte ordering convention are all shown. These parameters are chosen using jumpers, rather than TTY commands, so that the RVP7 can be installed on a host computer without the need for a temporary serial TTY. Once the I/O connections are established, the TTY setups can be further accessed via software on the host computer. A separate serial TTY is not mandatory.

HP-GPIO Parallel (WRITE:H IOREQ:H IOACK:H)
DEC-DR11 Parallel (WRITE:H IOREQ:H IOAUX:L IOACK:H)

If the jumpers have selected one of the parallel I/O protocols rather than SCSI, then the first line of the host computer interface information will resemble one of the these. The type of protocol is listed, along with the polarities that have been chosen for the interface lines that are actually used by that protocol.

SIO: Out=0, In=0, Err=0   AZ:359.99 EL:–0.01

The serial line input character count, input error count, and output character count are shown, as well as the current AZ and EL angles (from whatever source is supplying them). This status line is handy for debugging the serial interface in general — and especially for checking the reception of serial TAG angles. Using "**Vz**" will reset the various counts to zero and make subsequent changes easier to see.

Coax/Fiber/Pipeline Delay: 0.624 usec (Stdev: 0.014 usec)

During bootup the RVP7 measures the round trip delay along 1) the coax uplink to the receiver module, 2) the pipeline delays within the receiver module, 3) the optical fiber downlink to the main board, and 4) pipeline delays in the data decoding hardware. The time shown is accurate to within 14ns, and is used internally to insure that the absolute calibration of trigger and burst pulse timing remains unaffected by the distance between the main board and the receiver module. You may freely splice any lengths of coax and fiber without affecting the calibrations; the delay time will change, but the trigger and burst calibrations will remain constant.

The standard deviation of the measured delay is also shown. If the coax uplink and fiber downlink cables are run properly this variation should be less than the period of the acquisition clock, e.g., 0.028 μsec for the standard 35.975MHz rate. Larger errors may indicate a problem in the cabling. A diagnostic error bit is set if the error is greater than two acquisition clock periods.

IFD:Okay, Burst Pwr:–48.6 dBm, Freq:35.975 MHz

This line summarizes the receiver status and Burst input signal parameters. The status may show:

Okay RVP7/IFD and connecting cables are all working properly

NoFiber Problem in DownLink fiber cable from RVP7/IFD —> RVP7/Main

UpErr Problem in UpLink COAX cable from RVP7/Main —> RVP7/IFD

NoPLL RVP7/IFD PLL is not locked to external user-supplied clock reference

DiagSW RVP7/IFD test switches are not in their normal operating position

Reset by: Software   Up–time: 0–days 00:49:22

This line lists the origin of the last processor reset, as well as the total time that has elapsed since that reset occurred. The running time is given in days, followed by *hours* : *minutes* : *seconds.* The timer wraps around after approximately 180-days of continuous operation. The cause of the last reset will be one of the following:

1) Power-Up
2) External RESET line
3) SCSI Bus Reset
4) RESET OpCode with "Pwr" bit
5) RESET OpCode with "Rst" bit
6) RESET OpCode with "Dig" bit

7) BOOT OpCode
8) Internal Watchdog
9) TTY "*" command
10) IFD Power Sequencing
11) Burn-In Self Tests

PLDs– U39:1 U40:1 U47:2 U69:1 U73:1 U81:1 U86:1 IFD–U8:1

The revision levels of the Programmable Logic Devices (PLDs) are listed here. You may sometimes need to update one or more PLDs as major new features are added to the RVP7. The "U" numbers are the component designators on the RVP7/Main board, except for "IFD-U8" which is located in the RVP7/IFD module. The latter's revision level is transmitted to the main board via the optical downlink.

3.1.5 Burst-In / IF-In Swap Command

The "~" command swaps the Burst and IF inputs at the RVP7/IFD. Requests to toggle the state are made from the top level as follows:

```
RVP7> ~
IFD Burst/IF Inputs are: SWAPPED
RVP7> ~
IFD Burst/IF Inputs are: NORMAL
```

The selection remains in effect for the duration of the setup session, but then returns to NORMAL upon exiting the TTY monitor. The "~" command is very handy because it allows the **Pb**, **Pr**, and **Ps** plotting commands to easily run with one input or the other. Here are two examples of how this might be useful.

- When checking the range alignment on a Klystron system, the **Pb** plot can not be used in the usual way to center the Tx burst because a continuous-wave COHO (rather than a burst pulse) is typically used as the phase reference in these systems. However, if you swap the Burst and IF inputs, you can then use the **Pb** command to view and center the received leakage of the Tx pulse, and thus locate range zero.
- When setting up the AFC loop, you can use your RF signal generator to simulate the transmitter's frequency, and then run the loop with swapped RVP7/IFD inputs. The AFC servo will then hunt and follow the siggen frequency supplied via the receiver. You can then make step changes in that frequency to verify that the loop responds properly.

Note that the same input swapping function is also available via the RVP7/IFD toggle switches. However, those switches may be located far away from the operator's terminal; hence, the command interface is still a valuable addition. The "~" command can only be used with the new Rev.D RVP7/IFD; the command is unimplemented, and will not even show up in the "Help" list, when earlier receivers are connected.

3.2 Host Computer I/O Debugging

The RVP7 supports two very powerful monitoring functions that are helpful in debugging the I/O interface to the host computer. One examines the physical layer of the interface, i.e., the electrical handshake and data lines themselves. The other examines the application layer, i.e., the 16-bit opcodes and data that define the RVP7's application programming interface.

3.2.1 Physical-Level I/O Examiner

The RVP7 has TTY support for debugging the physical level of the host computer's SCSI or Parallel interface. The "X" (eXamine) command allows you to watch all incoming 16-bit words as they arrive from the host computer. In addition, you may also send 16-bit words back the other way. The "X" command is only available from the RS232 hardware TTY interface; it can not (obviously) be used via chat mode over the same I/O interface that it trying to examine. As such, the "X" command will not even be listed in the RVP7's top level help menu during a chat mode session.

While the "X" command is running, any words that arrive from the computer will immediately be printed in hex format, along with an "address" (word counter, starting from zero) at the start of each line. Meanwhile, the "W" subcommand can be used to write individual words back to the computer, and the "Q" subcommand will exit the I/O examiner entirely.

i **Note: When the "X" command is running, the RVP7 does not interpret the incoming 16-bit words as commands and arguments. Data sent to the RVP7 are discarded after being printed; and output from the RVP7 will occur only if the "W" subcommand is manually used. The "X" command is intended to debug the physical layer of the computer interface in a very controlled manner.**

The following dialog was captured in response to the host computer writing 100, 200, 300 (decimal) to the RVP7. The "W" subcommand was then used twice to output a 0x4000 and 0x8000 from the RVP7, and the computer then sent the values 1, 2, 3, 4, 5.

```
RVP7> X
Host Computer I/O Debug Monitor
-------------------------------
  Q: Exit the monitor
  W: Output a word to the computer

0x0000:  0x0064 0x00C8 0x012C
Output Word : 0x4000
Output Word : 0x8000
0x0003:  0x0001 0x0002 0x0003 0x0004  0x0005
```

3.2.2 Application-Level I/O Examiner

The RVP7 has TTY support for debugging the application level of the host computer's SCSI or Parallel interface. The Real Time TTY Monitor (RTM, see Section 3.3.8) can be configured to expose the computer's complete I/O stream while the RVP7 is running and processing commands in its normal manner. Because of the enormous amount of TTY output that can be generated by this option, all other RTM selections are disabled whenever host computer I/O is being monitored. Also, those other RTM selections would interfere with the multi-line formatting of the I/O text.

The TTY printout shows incoming opcodes called out by name, and subsequent input and output words formatted into a table. The data are printed in Hex, twelve words per line, and include a word offset (origin zero) at the start of each line. The offset is reset to zero at the start of each new input or output sequence.

Lines of data that are repeats of identical values will be skipped with a "..." indication. This shortens and simplifies the printout; but more importantly, it reduces TTY overhead so that the processor is less I/O bound. Also for this reason, the "0x" Hex prefix is omitted during the possibly lengthy printing of the data word tables.

i **Note: As with all other Real Time TTY Monitor (RTM) functions, the RVP7 remains completely functional while host computer I/O is being monitored. However, unlike all other RTM functions, the I/O monitor will stall the main processor whenever the TTY becomes I/O bound; and the performance of the RVP7 will be degraded, perhaps severely. It is recommended that you configure the TTY for 38.4-KBaud to minimize the serial I/O delays.**

The following sample transactions were captured in response to starting the IRIS/Open ZAUTO utility. An I/O RESET and diagnostic OTEST are first performed. The pulse width selection bits and maximum trigger rates are then set with PWINFO, and angle sync is disabled with LSYNC. The header words for processed data are decided using CFGHDR, operational parameters are loaded with SOPRM, and final RVP7 parameters are read back with GPARM. Finally, the trigger rate is set using SETPWF, and a dummy range mask consisting of a single bin is setup with LRMSK.

```
Opcode 0x008C (RESET)
Opcode 0x0004 (OTEST)
Output Words
    0: 0001 0002 0004 0008  0010 0020 0040 0080  0100 0200 0400 0800
   12: 1000 2000 4000 8000
Opcode 0x000F (PWINFO)
Input Words
    0: 8421 012C 0BB8 0FA0  1F40
Opcode 0x0011 (LSYNC)
Opcode 0x005F (CFGHDR)
Input Words
    0: 0001 0000
Opcode 0x0002 (SOPRM)
Input Words
    0: 0019 000F 07AE 0008  FE70 0080 00A0 0000  0003 000A AAAA 8888
   12: C0C0 C000 0000 0000  0000 AAAA 0000 2710
Opcode 0x0009 (GPARM)
Output Words
    0: 1200 0001 0960 FFFF  FFFF 0D5B 0000 0000  0000 4284 0000 0000
   12: 0019 743D 0007 0000  0000 230B 0032 5DC0  0BB8 1770 1D4C 2EE0
   24: 8421 0000 2EE0 2EE0  0960 0960 000F 07AE  0008 FE70 0080 00A0
   36: 0000 0000 0000 0000  0000 0000 0001 000E  0000 000E 0000 0D5B
   48: 8000 0000 0000 0000  0000 0000 0000 0000  0000 0000 0000 0000
   60: 0000 0000 0000 0000
```

```
Opcode 0x0010 (SETPWF)
Input Words
    0: 2EE0
Opcode 0x0001 (LRMSK)
Input Words
    0: 0001 0000 0000 0000  0000 0000 0000 0000  0000 0000 0000 0000
   12: ...
  504: 0000 0000 0000 0000  0000 0000 0000 0000
```

This RTM option to monitor computer I/O is automatically disabled at powerup, and therefore can not be saved permanently. This is to avoid confusing situations in which the monitor is accidently left running — the RVP7 would appear to be working, but at a puzzling level of degraded performance.

3.3 View/Modify Dialogs

The **M** command may be used to view, and optionally to modify, all of the current settings. The current value of each parameter is printed on the screen, and the TTY pauses for input at the end of the line. Pressing Return advances to the next parameter, leaving the present one unchanged. You may also type **U** to move back up in the list, and **Q** to exit from the list at any time.

Typing a numeric or YES/NO response (as appropriate to the parameter) changes the parameter's value, and displays the line again with the new value. All numbers are entered in base ten, and may include a decimal point and minus sign. In some cases, several parameters are displayed on one line, in which case, as many parameters are changed as there are new values entered. In all cases, the numbers are checked to be within reasonable bounds, and an error message (listing those bounds) is printed if the limits are exceeded. Note that changes to the settings (generally) do not take effect until after the **Q** command is typed, at which point the RVP7 exits the local TTY menu and resumes its normal processing operations.

Since the number of setup questions is large, follow the **M** command with a second letter to select the subcategory, i.e., **Mb** (Burst Pulse and AFC), **Mc** (Board Configuration), **Mf** (Clutter Filters), **Mp** (Processing Options), **Mt** (Triggers and Timing), **Mz** (Transmitter Phase Control), **M*** (Stand-alone Settings) or **M+** (Debug Options). The **M** command by itself prints the entire set of questions so that you can make a hard copy.

The **M** command always works from the current parameter values, not from the saved values in non-volatile RAM. If the host computer has modified some of the current values, then you will see these changes as you skip through the setup list. However, typing **S** at that point would save all of the current settings and would, perhaps, make many changes to the original non-volatile settings. In general, to make an incremental change to the saved settings, first type **R** to restore all of the saved values, then use **M** to make the changes starting from that point, and **S** to save the new values.

A listing of the parameters that can be viewed and modified with the **M** command is detailed in the following subsections. In each case, the line of text is shown exactly as it appears on the TTY with the factory default settings. A definition of each parameter is given and, if applicable, the lower and upper numeric bounds are shown.

3.3.1 Mc — Board Configuration

This set of commands configure general properties of the RVP7/IFD and RVP7/Main boards.

`Acquisition clock: 35.9751 MHz`

This is the frequency of the oscillator at U5 in the IF receiver module. Except for custom receivers, this will always be 35.9751 MHz; which gives a fundamental sample spacing of 1/240 km (approximately 4.17 meters).

Limits: 33.33 to 41.67 MHz

`DSP clock (U83) :  60.0000 MHz`

The master DSP clock frequency should be indicated here. The RVP7 bases many of its timing measurements on its own system clock, hence the need to know this frequency.

Limits: 40 to 80 MHz.

`Serial clock (X1):  4.9152 MHz`

This is the frequency of the serial I/O crystal labeled "X1" on the RVP7 board. By changing this crystal you can attain unusual baud rates for the two serial ports.

Limits: 1 to 5 MHz.

`Dual simultaneous receivers are being used: NO`

Answer this question "Yes" if the RVP7 will be processing simultaneous signals from two separate receivers. Answering "No" will revert to normal operation with just a single receiver.

`Dual–LNA/Rcvr single–channel switched mode: NO`

For dual-polarization single-receiver systems, this question decides whether you have a single LNA and IF-Amplifier that switches between H&V (the typical case); or two separate receivers, each hard wired to H and V, with switching performed after the IF amplifiers. The question affects how noise levels are measured and applied to the data.

```
Dedicate one DSP to LOG video output: YES
  Upper 100.0 dB will occupy 85.0% of voltage span
  Force freerunning video mode: NO
  Plot data from secondary receiver: NO
```

The RVP7 supports the option of sourcing a LOG Video analog output signal from the backpanel of the main chassis. There are two ways that this signal can be configured:

- ***Self-Triggering, Free-Running Mode***
  This is the default mode that is available on all RVP7 boards. The output signal is periodic at approximately the PRF of the radar, but is free-running, i.e., not actually synchronized with the radar trigger. A synthetic 1.0 μsec wide, full scale, "trigger" pulse is embedded at the zero-range start of each LOG Video waveform. This marker can easily trigger an oscilloscope if the scope's trigger level is set just below the maximum LOG Video voltage level.
- ***Waveform Locked to Radar Trigger***
  This mode requires a (one-wire) hardware modification to the Rev.B RVP7/Main board. The LOG Video waveform then becomes locked to the radar trigger, so that the LOG signal can be displayed on any device that already receives the radar trigger.

In either case, the LOG Video output signal is unipolar, ranging from approximately 0.0V to 3.0V. It is active during all data processing modes that the host computer might request, as well as during the idle time between scans. The signal is absent (zero), however, during the short intervals of time that the RVP7 is being reconfigured by the host computer, or when the RVP7's local TTY setups are being used.

The time resolution of the synthesized LOG Video signal is fixed at 1.0 μsec per bin. This is independent of the actual range resolution of the FIR matched filter. Whatever (I,Q) data are actually being computed by the FIR front end are selected for a nearest fit to each 1.0 μsec synthetic output cell. The maximum number of incoming FIR range bins that can be selected among is 5460. Thus, for example, the maximum range of the LOG Video signal would be 682km when the FIR range resolution is 125–meters.

Answer the first question "Yes" if you would like the RVP7/Main board to synthesize and drive the LOG Video output signal. The cost of doing this is that one of the "slave" DSP chips will be removed from the normal Doppler processing chain, and dedicated to the task of LOG Video generation. On a single-board system, the three available slave DSPs would be reduced to two; whereas on a dual-board system, the 13 available DSPs would be reduced to 12. Obviously, the percentage penalty is less in a larger system.

The second question decides how the overall dynamic range of the receiver will fit into the 12-bit unipolar output voltage span of the DAC that produces the LOG Video waveform. The default setting calls for the upper 100dB of dynamic range to occupy 85% of the output voltage span. This means that the strongest IF input signal would produce 85% of the maximum DAC voltage (approximately 2.55 Volts); 50dB down would be 42.5%, and 100dB down would be 0%, i.e., zero volts.

If you are using a self-triggering LOG Video waveform, then the 15% of headroom provided by the default settings leaves room for the embedded trigger pulse. However, if your RVP7 has the hardware modification required to synchronize the LOG Video to the system trigger, then the full 100% of the DAC voltage span can freely be used. The third setup question can be used to force freerunning mode on an RVP7 that has the hardware modification. This question is included mostly for testing purposes.

The last question only appears in dual-receiver mode. Answer "Yes" if you would like the LOG video analog output signal to be based on the data from the secondary receiver rather than from the primary receiver.

`Scope plots– Holdoff ratio: 0.50, Stroke: 1000.0 usec`

The oscilloscope plotting commands are described in Chapter 4. This question allows you to vary the amount of holdoff time that is inserted between each drawing stroke, as well as the stroke length itself. Try increasing the holdoff if your scope is not triggering reliably. Longer holdoffs make it easier for the scope to find the initial trigger point, but may introduce visible flicker. To reduce flicker, try decreasing the stroke duration from its default value of 1000 microseconds.

Limits: Holdoff 0.05 to 5.00, Stroke 100 to 10000 μsec.

`Respond to SCSI Reset : NO`

This question determines whether the RVP7 will perform a complete reset (equivalent to its power-up sequence) when a SCSI bus "reset" signal is detected. A bus reset is a global condition that is received by all attached devices. If the RVP7 shares the SCSI bus with other devices, you may prefer to answer this question "NO".

`TTY secondary rate: 1200 Baud`

This is the baud rate at which the TTY runs when jumper JP4 is in the BC position. The AB position will always produce 9600 baud.

Limits: 300 to 76800 baud.

SIO usage– 0:Off, 1:RTDisp, 2:RCPrcv, 3:RCPxmt : 0

This question determines how the SIO serial line is to be used. There are several optional SIO questions that will follow, depending on the answer to this one.

The "RTDisp" option enables the serial Real Time Display (See Appendix B).

The "RCPrcv" option allows antenna angles to be received serially, rather than via the parallel TAG inputs.

The "RCPxmt" option causes the RVP7 to output serial antenna records which are based on the azimuth and elevation TAG bits (either parallel input, or simulated). These serial packets are in RCV02 format, and are transmitted at the rate of two per second. The AZ and EL angle fields are properly filled in, as are the AZ and EL velocity, millisecond time stamp, and the pulse width status bits. The velocities are estimated from the last half second of angle travel. This SIO option is intended for use on a system that has IRIS and an RVP7, but no antenna controller to generate the serial antenna information directly. The SIO receiver is unused in this mode, and it will ignore any characters that happen to be sent to the RVP7.

RCP angle record delay: 0.0083 sec

This question only appears if the SIO usage is set to "RCP". The specified time is used to compensate for delays in the encoding, transmission, and reception of the serial antenna angles. The default value corresponds to an 8-character delay at 9600-baud.

Limits: 0 to 0.25 seconds

Retransmit RCP stream: NO

This question only appears if the SIO usage is set to "RCP". Answering "YES" will cause the RVP7 to retransmit some or all of the RCP data that it receives. The "NO" response should be chosen, unless you plan on using the RVP7 to reduce the bandwidth of the incoming serial stream (see next question). Otherwise, the additional retransmission CPU load on the RVP7 is not serving any really useful purpose.

Retransmit one-in-1 RCP Angle Records :

This question only appears if the SIO usage is set to "RCP", and the RCP stream is being retransmitted. It allows you to convert the high-data-rate antenna packet stream entering the RVP7 into an equivalent stream at a lower rate. This reduced output stream can then be fed into the host computer, and be received there with less I/O and CPU overhead. Note that the data reduction only applies to the RCV01, RCV02, RCV03, and RCV04 status packets. All other packets that are received by the RVP7 are retransmitted without being changed or pruned.

Limits: x1 to x50 reduction factor.

RTDisp Mode- 1:Normal, 2:Auto : 1

This question only appears if the SIO usage is set to "RTDisp". Selecting "Normal" enables the serial line so that Parameter Data Packets can be output in sync with host computer requests for rays of data from the RVP7. "Auto" is similar, except that

Parameter Data Packets can be output immediately after powerup or board reset, i.e., without requiring the host computer at all. The "Auto" function ceases, however, as soon as any computer I/O is detected, and the display stream then reverts to the "Normal" status. See Appendix B for hints on setting up the RVP7 to use "Auto" mode.

`SIO Data rate: 9600 Baud`

This question only appears if the SIO usage is not "OFF". This is the baud rate for the RVP7 "SIO" serial interface signals which are located on the P1 96-pin connector.

Limits: 300 to 76800 baud.

`SIO StopBits- 0:1, 1:1.5, 2:2 : 0`

This question only appears if the SIO usage is not "OFF". The number of stop bits used by the "SIO" channel is selectable here. A single stop bit results in the minimum overhead, and will generally be the preferred setting.

`PHOUT[0:7] usage– 0:Off, 1:XMTPhase, 2:DigitalAFC : 0`

This question chooses how the RVP7's eight RS422 "phase control" outputs are to be assigned. Choices are "Off", "Transmit Phase Control", and "Digital AFC". Transmit Phase Control is configured with **Mz**, and Digital AFC is configured in the **Mb** setup section.

`PWINFO command enabled: No`

The "Pulsewidth Information" user interface command can be disabled, thus further protecting the radar against inappropriate combinations of pulsewidth and PRF. This is a more safe setting in general, and is even more important when DPRT triggers are being generated. It can also be useful when running user code that is not yet fully debugged.

`TRIGWF command enabled: NO`

The "Trigger Waveform" user interface command can be disabled if you want to prevent the host computer from overwriting the RVP7's stored trigger specifications. This is the default setting, based on the assumption that the built-in plotting commands would be used to configure the triggers. Answering "YES" will allow new waveforms to be loaded from the host computer.

`Watchdog autoreset: DISABLED`
`Watchdog autoreset in 3.0 seconds`

The purpose of the watchdog autoreset is to detect when the I/O interface between the RVP7 and the host computer seems to be hungup, so that a complete RVP7 reset can be performed. Then, when the reset sequence completes, the host computer would be able to resume communication with the RVP7, reconfigure all parameters, and continue with data acquisition.

The watchdog's timeout period can be set to any length of time up to 20 seconds; setting the time to zero will disable the auto-reset feature entirely. If the RVP7 ever fails to return to its top-level I/O command interface within the chosen time, then a reset will automatically occur. Make sure that you allot sufficient time to absorb normal I/O latencies and ray processing times. A typical timeout value is 3.0 to 5.0

seconds. Also make sure that the host computer does not try to reestablish communication with the RVP7 until the board reset sequence is complete. Allow at least three seconds more than the chosen watchdog time before resuming I/O.

Approximately 40 lines of diagnostic information are printed to the local TTY when the RVP7 executes an autoreset. This printout includes a stack dump, DMA uplink pointers, operational parameters, run-time parameters, and other interesting scalars. If you are able to capture these diagnostics, please e-mail them to "support@sigmet.com" so that we may help diagnose the condition.

Limits: Disabled, to 20 seconds.

`15V power is stable 2.5 seconds after +5V`

This question allows the RVP7 to startup properly when the $\pm$15V power supplies take longer to stabilize than the +5V supply. Most of the RVP7/Main board requires only +5V to operate; but the uplink to the RVP7/IFD module relies on +15V to drive the coaxial cable. The built-in diagnostics will fail if +15V is not present during the initial startup sequence.

The RVP7 can distinguish between various types of resets, i.e., power-up, external RESET line, watchdog autoreset, and software induced. The extra stabilization time for the 15V supply is added only during a genuine power-up reset.

Limits: 0 to 5 seconds.

`RVP6 Emulation: No`

The RVP7 implements a reasonably precise emulation of the RVP6 command set. This mode is useful because it allows an RVP7 to be plugged directly into a software system that used to run with an RVP6. All of the configuration steps that are new and unique to the RVP7 can be handled by the local TTY and Scope setups, thus making no demands on the user's system code for support. Answer this question "YES" for maximum compatibility with old driver software. However, if you are running IRIS version 6.11 or higher, then answer "NO" to enable using new RVP7 features as they are developed.

The RVP7 returns a version number of 35 when the processor is running in RVP6 compatibility mode. This fudged value will appear in the SCSI Inquiry Command reply, and in the GPARM parameter packet. Elsewhere, the correct RVP7 ROM version number will always appear. The reason for doing this is so that the RVP7 appears (to the host computer) to be a modern RVP6 with all of the latest opcodes and features.

3.3.2 Mp — Processing Options

`Major Mode- 0:User, 1:PPP, 2:FFT : 0`

The top level RVP7 operating modes are described in the documentation of SOPRM command word #9. This question allows you to use the mode that has been selected by that command, or to force the use of a particular mode.

Window- 0:User, 1:Rect, 2:Hamming, 3:Blackman : 0

Whenever power spectra are computed by the RVP7, the time series data are multiplied by a (real) window prior to computation of the Fourier Transform. You may use whichever window has been selected via SOPRM word #10, or force a particular window to be used.

R2 Processing- 0:Never, 1:User, 2:Always : 1

Controls R0/R1 versus R0/R1/R2 processing. Selecting ”0” unconditionally disables the R2 algorithms, regardless of what the host computer requests in the SOPRM command. Likewise, selecting ”2” unconditionally enables R2 processing. These choices allow the RVP7 to run one way or the other without having to rewrite the user code. This is useful for compatibility with existing applications.

Clutter Microsuppression- 0:Never, 1:User, 2:Always : 1

Controls whether individual “cluttery” bins are rejected prior to being averaged in range. Same interpretation of cases as for ”R2 Processing” above.

2D Final Speckle/Unfold – 0:Never, 1:User, 2:Always : 1

The Doppler parameter modes (PPP, FFT, etc) include an optional 3x3 interpolation and speckle removal filter that is applied to the final output rays. This 2-dimensional filter examines three adjacent range bins from three successive rays in order to assign a value to the center point. Thus, for each output point, its eight neighboring bins in range and time are available to the filter. Only the *dBZ*, *dBT*, *Vel*, and *Width* data are candidates for this filtering step; all other parameters are processed using the normal 1-dimensional (three bins in range) speckle remover. See Section 5.3.3 for more details.

Unfold Velocity (Vh–Vl) – 0:Never, 1:User, 2:Always : 0

This question allows you to choose whether the RVP7 will unfold velocities using a simple $(V_{high} - V_{low})$ algorithm, rather than the standard algorithm described in Section 5.6. Bit-11 of SOPPRM word #10 is the host computer's interface to this function when the “1:User” case is selected (See Section 6.3).

i **Note: This setup question is included for research customers only. The standard unfolding algorithm should still be used in all operational systems because of its lower variance. For this reason, the factory default value of this parameter is “0:Never”.**

Process w/ custom trigs – 0:Never, 1:User, 2:Always : 0

This question allows you to choose whether the RVP7 will attempt to run its standard processing algorithms even when a custom trigger pattern has been selected via the SETPWF command. Generally it does not make sense to do this, so the default setting is “0:Never”. Bit-12 of OPPRM word #10 is the host computer's interface to this function when the “1:User” case is selected (See Section 6.3).

Minimum freerunning ray holdoff: 100% of dwell

This parameter controls the rate at which the RVP7 processes free-running rays in the FFT, DPRT, and Random Phase modes. This prevents rays from being produced at the full CPU limit or I/O limit of the processor (whichever was slower); which could result in highly overlapping data being output at an unusably fast rate. Note that this behavior will only occur when one of these non-PPP modes is chosen, and is then allowed to run without angle syncing. Such is likely the case for IRIS manual scans or during Passive IRIS mode.

To make these free-running modes more useful, you may establish a minimum holdoff between successive rays, expressed as a percentage of the number of pulses contributing to each ray. Choosing 100% (the default) will produce rays whose input data do not overlap at all, i.e., whose rate will be exactly the PRF divided by the sample size. Choosing 0% will give the unregulated behavior in which no minimum overlap is enforced and rays may be produced very quickly.

Limits: 0 to 100%

`Linearized saturation headroom: 4.0 dB`

The RVP7 uses a statistical saturation algorithm that estimates the real signal power correctly even when the IF receiver is overdriven (i.e., for input power levels above +4dBm). The algorithm works quite well in extending the headroom above the top end of the A/D converter, although the accuracy decreases as the overdrive becomes more severe. This parameter allows you to place an upper bound on the maximum extrapolation that will ever be applied. Choosing 0dB will disable the algorithm entirely.

Limits: 0 to 6dB

`Apply amplitude correction based on Burst/COHO: YES`
`  Time constant of mean amplitude estimator: 70 pulses`

The RVP7 can perform pulse-to-pulse amplitude correction of the digital (I,Q) data stream based on the amplitude of the Burst/COHO input. Please see Section 5.1.6 for a complete discussion of this feature.

Limits: 10 to 500 pulses

`IFD built–in noise dither source: –57.0dBm`

This question will only appear if the processor is attached to a Rev.D RVP7/IFD that includes an out-of-band noise generator to supply dither power for the A/D converters. The available power levels are { Off, –57dBm, –37dBm, –32dBm, –27dBm, –22dBm, –19dBm }. The closest available level to your typed-in value will be used. You can observe the band-limited noise easily in the **Pr** plot to confirm its amplitude and spectral properties.

For standard operation, we recommend running at –57dBm. The problem higher levels of dither level is that, for certain choices of (I,Q) FIR filter, the stopband of the filter may not give enough attenuation to preserve the RVP7/IFD's inherent noise level. For example, the factory default 1MHz bandwidth Hamming filter has a stopband attenuation near DC of approximately 43dB. You can see this graphically at the right edge of the **Ps** menu. The in-band contribution of dither power is therefore approximately (–37dBm) – 43dB = –80dBm, which exceeds the A/D converter's 1MHz bandwidth noise of –81.5dBm.

```
TAG bits to invert     AZ:0000    EL:0000
TAG scale factors      AZ:1.0000  EL:1.0000
TAG offsets (degrees)  AZ:0.00    EL:0.00
```

The incoming TAG input bits may be selectively inverted via each of the 16-bit words. The values are displayed in Hex. Setting a bit will cause the corresponding AZ (bits 0–15) or EL (bits 16–31) lines to be inverted. Note that the SOPRM command also specifies TAG bits to invert. Both specifications are XOR'ed together to yield the net inversion for each TAG line.

The overall operations are performed in the order listed. Incoming bits are first inverted according to the two 16-bit XOR masks. This yields an unsigned 16-bit integer value which is then multiplied by the signed scale factor. The result is interpreted as a 16-bit binary angle (in the low sixteen bits), to which the offset angle is finally added.

As an example, suppose that the elevation angle input to the RVP7 was in an awkward form such as unsigned integer tenths of degrees, i.e., 0x0000 for zero degrees, 0x000a for one degree, 0x0e06 for minus one degree, etc. If we apply a scale factor of 65536/3600 = 18.2044 to these units, we will get 16-bit binary angles in the standard format. If we further suppose that the input angle rotated "backwards", we could take care of this too using a multiplier of –18.2044.

```
Interference Filter– 0:None, Alg.1, Alg.2, Alg.3: 1
  Threshold parameter C1: 10.00 dB
  Threshold parameter C2: 12.00 dB
```

The RVP7 can optionally apply an interference filter to remove impulsive-type noise from the demodulated (I,Q) data stream. See Section 5.1.4 for a complete description of this family of algorithms.

```
Polarization Params – Filtered:YES  NoiseCorrected:YES
PhiDP – Negate: NO ,  Offset:0.0 deg
KDP   – Length: 5.00 km
T/Z/V/W computed from:   H–Xmt:YES   V–Xmt:YES
T/Z/V/W computed from:  Co–Rcv:YES  Cx–Rcv:NO
```

The first question decides whether all polarization parameters will be computed from filtered or unfiltered data, and whether noise correction will be applied to the power measurements.

The second and third questions define the sign and offset corrections for Φ_{DP} and the length scale for *KDP*.

The fourth and fifth questions control how the standard parameters (Total Reflectivity, Corrected Reflectivity, Velocity, and Width) are computed in a multiple polarization system. Answering *YES* to *H-Xmt* and/or *V-Xmt* means that data from those transmit polarizations should be used whenever there is more than one choice available. Thus, these selections only apply to the Alternating and Simultaneous transmit modes. Likewise, answering *YES* to *Co-Rcv* and/or *Cx-Rcv* means to use the received data from the co-channel or cross-channel. The receiver question will only appear when dual simultaneous receivers have been configured.

A typical installation might use *H-Xmt:YES*, *V-Xmt:YES*, *Co-Rcv:YES*, *Cx-Rcv:NO*. This will compute (T/Z/V/W) from the co-polarized receiver using both H&V transmissions. Including both transmissions will decrease the variance of (T/Z/V/W); although some researchers prefer excluding *V-Xmt* because that is more standard in the literature. Also, if your polarizations are such that the main power is returned on the cross channel, then you will probably want *Co-Rcv:NO* and *Cx-Rcv:YES*.

`DualRx – Sum H+V Time Series: NO`

In dual-receiver systems, you may choose whether the (H+V) time series data consist of the sum of the "H" and "V" samples or the concatenation of half the "H" samples followed by half the "V" samples. The later is more useful when custom software is being used to analyze the data from the two separate receive channels.

3.3.3 Mf — Clutter Filters

`Doppler Filter Set- 0:40dB, 1:50dB, 2:Saved : 0`

The RVP7 has two built-in IIR Doppler clutter filter sets; one set having 40dB of stopband attenuation, and the other having 50dB. This question chooses which set is loaded on powerup.

```
Spectral Clutter Filters
------------------------
Filter #1 – Type:0(Fixed)    Width:1   EdgePts:2
Filter #2 – Type:0(Fixed)    Width:2   EdgePts:2
Filter #3 – Type:0(Fixed)    Width:3   EdgePts:3
Filter #4 – Type:0(Fixed)    Width:4   EdgePts:3
Filter #5 – Type:1(Variable) Width:1   EdgePts:2   Hunt:2
Filter #6 – Type:1(Variable) Width:2   EdgePts:2   Hunt:2
Filter #7 – Type:1(Variable) Width:3   EdgePts:3   Hunt:3
```

These questions define the heuristic clutter filters that operate on power spectra during the FFT-type major modes. Filter #0 is reserved as "all pass", and is not redefinable here. For filters #1 through #7, enter a digit to choose the filter type, followed by however many parameters that type requires.

Fixed Width Filters (Type 0)
These are defined by two parameters. The "Width" sets the number of spectral points that are removed around the zero velocity term. A width of one will remove just the DC term; a width of two will remove the DC term plus one point on either side; three will remove DC plus two points on either side, etc. Spectral points are removed by replacing them with a linear interpolating line. The endpoints of this line are determined by taking the minimum of "EdgeMinPts" past the removed interval on each side.

Variable Width, Single Slope (Type 1)
The RVP7 supports variable-width frequency-domain clutter filters. These filters perform the same spectral interpolation as the fixed-width filters, except that their notch width automatically adapts to the clutter. The new filters are characterized by the same *Width* and *EdgePts* parameters in the **Mf** menu, except that the *Width* is now interpreted as a minimum width. An additional parameter *Hunt* allows you to choose

how far to extend the notch beyond *Width* in order to capture all of the clutter power. Setting *Hunt=0* effectively converts a variable-width filter back into a fixed-width filter.

The algorithm for extending the notch width is based on the slope of adjacent spectral points. Beginning *(Width–1)* points away from zero, the filter is extended in each direction as long as the power continues to decrease in that direction, up to adding a maximum of *Hunt* additional points. If you have been running with a fixed *Width=3* filter, you might try experimenting with a variable *Width=2* and *Hunt=1* filter. Perhaps the original fixed width was actually failing at times, but you were reluctant to increase it just to cover those rare cases. In that case, try selecting a variable *Width=2* and *Hunt=2* filter as an alternative. In general, make your variable filters "wider" by increasing *Hunt* rather than increasing *Width*. This will preserve more flexibility in how they can adapt to whatever clutter is present.

`Residual clutter LOG noise margin: 0.15 dB/dB`

Whenever a clutter correction is applied to the reflectivity data, the LOG noise threshold needs to be increased slightly in order to continue to provide reliable qualification of the corrected values. The reason for this is that the uncertainty in the corrected reflectivity becomes greater after the clutter is subtracted away.

For example, if we observe 20dB of total power above receiver noise, and then apply a clutter correction of 19dB, we are left with an apparent weather signal power of +1dB above noise. However, the uncertainty of this +1dB residual signal is much greater than that of a pure weather target at the same +1dB signal level.

The "Residual Clutter LOG Noise Margin" allows you to increase the LOG noise threshold in response to increasing clutter power. In the previous example, and with the default setting of 0.15dB/dB, the LOG threshold would be increased by 19x0.15 = 2.85dB. This helps eliminate noisy speckles from the corrected reflectivity data.

```
Whitening Parameters
--------------------
Noise threshold for replacing a point: 1.20
Replacement value multiplier: 0.5000
SNR in tails, for determining width: 0.25
```

These questions control the adaptive whitening filter that is used by the Random Phase processing algorithms. A spectral point will be whitened if the ratio of its power to the noise power exceeds the "Noise threshold for replacing a point." The whitened point will consist of a complex value whose ARG is identical to that of the original point, and whose MAG is the product of the noise level with the "Replacement value multiplier" term. The nominal spectral width of the whitened region is a function of the power and width of the coherent signal, and the noise level. Assuming a Gaussian model, the "SNR in tails..." value is the ratio of the coherent power in the tails of the distribution to the noise level.

`RPhase SQI Threshold  Slope:0.50  Offset:–0.05`

The two values in this question define a secondary SQI threshold that is used to qualify the LOG data during Random Phase processing. The secondary SQI level is computed by multiplying the primary user-supplied SQI threshold by the SLOPE, and adding the OFFSET. See also Section 5.9.3.

Limits: SLOPE: 0.0 to 2.0, OFFSET –2.0 to 1.0

3.3.4 Mt — General Trigger Setups

These questions are accessed by typing "Mt" with no additional arguments. They configure general properties of the RVP7 trigger generator

```
Pulse Repetition Frequency: 500.00 Hz
```

This is the Pulse Repetition Frequency of the internal trigger generator.

Limits: 50 to 6000Hz.

```
Transmit pulse width: 0
```

Limits: 0 to 3

```
Use external pretrigger: NO
  PreTrigger active on rising edge: YES
  PreTrigger fires the transmitter directly: NO
```

When an external pretrigger is applied to the TRIGIN input of the RVP7, either the rising or falling edge of that signal initiates operation. This decision also affects which signal edge becomes the reference point for the pretrigger delay times given in the "**Mt<n>**" section.

Answer the second sub-question according to whether the radar transmitter is directly fired by the the external pretrigger, rather than by one of the RVP7's trigger outputs. In other words, answer "YES" if the transmitter would continue running fine even if the RVP7 TRIGIN signal were removed. This information is used by the "L" and "R" subcommands of the "Pb" plotting command, i.e., when slewing left and right to find the burst pulse, the pretrigger delay will be affected rather than the start times of the six output triggers.

```
2–way (Tx+Rx) total waveguide length: 0 meters
```

Use this question to compensate for the offset in range that is due to the length of waveguide connecting the transmitter, antenna, and receiver. You should specify the total 2-way length of waveguide, i.e., the span from transmitter to antenna, plus the span from antenna to receiver. The RVP7 range selection will compensate for the additional waveguide length to within plus-or-minus half a bin, and works properly at all range resolutions.

```
POLAR0 is high for vertical polarization : NO
POLAR1 is high for vertical polarization : NO
```

These questions define the logical sense of the two polarization control signals POLAR0 and POLAR1. In a dual-polarization radar POLAR0 should be used to select one of two possible states (nominally horizontal and vertical, but any other polarization pair may also be used). The control signal will either remain at a fixed level, or will alternate from pulse to pulse with a selectable transition point (See Section 3.3.5). POLAR1 is identical to POLAR0, but may be configured with a different polarity and switch point. This second signal could be used if the radar's polarization switch required more than one control line transition when changing states.

```
Quantize trigger PRT to ((1 x AQ) + 0) clocks
```

It is possible to control the exact quantization of the PRT of the internal trigger generator. Normally the trigger PRT is chosen as the closest multiple of AQ (the acquisition clock period) that approximates the requested period. This question allows the possible PRT's to be constrained to ((N x AQ) + M) clock cycles. This feature can be useful for synchronous receiver systems in which the trigger period must be some exact multiple of the COHO period.

```
Blank output triggers according to TAG#0 : NO
  Blank when TAG input is high : NO
  Blank triggers 1:YES 2:YES 3:YES 4:YES 5:YES 6:YES
```

These questions control trigger blanking based on the TAG0 input line. You first select whether the trigger blanking feature is enabled; and then optionally choose the polarity of TAG0 that will result in blanking, and which subset of the six user definable triggers are to be blanked.

```
Blank output triggers during noise measurement : NO
```

The RVP7 can inhibit the subset of blankable trigger lines whenever a noise measurement is taken. This will be forced whenever trigger blanking (based on TAG0) is enabled, but it can also be selected in general via this question. Since noise triggers must be blanked whenever trigger blanking is enabled, this question only appears if trigger blanking is disabled.

This question permits the state of the triggers during noise measurements to be consistent and known, regardless of whether the antenna happens to be within a blanked sector; and you have the additional flexibility of choosing blanked noise triggers all the time.

```
Rx–Fixed Triggers: #1:N #2:N #3:N #4:N #5:N #6:N P0:N P1:N Z:N
```

You have explicit control over which RVP7 trigger outputs are timed relative to the transmitter pre-fire sequence, versus those which are relative to the actual received target ranges. Triggers in the first category will be moved left/right by the "L/R" keys in the **Pb** plot, and will also be slewed in response to Burst Pulse Tracking. Triggers in the second category remain fixed relative to "receiver range zero", and are not affected by the "L/R" keys or by tracking.

This question specifies which triggers are Tx-relative and which are Rx-relative. Answer with a sequence of "Y" or "N" responses for each of the six trigger lines, for the two polarization control lines, and for the timing of the phase control lines. You should answer *No* for any trigger that is involved with the pre-fire timing of the transmitter. If you enable the Burst Pulse Tracker (Section 5.1.3) you will probably want to assign a *Yes* to some of your triggers so that they remain fixed relative to the burst itself.

It is very helpful to have these two categories of trigger start times. Triggers that fire the transmitter, either directly or indirectly, should all be moved as a group when hunting for the burst pulse and moving it to the center of the FIR window. However, triggers that function as range strobes should be fixed relative to range zero, i.e., the center of that window, and the center of the burst. This distinction becomes important when the transmitter's pre-fire delay drifts with time and temperature.

```
Replace triggers with alternate waveforms: YES
  Trigger #1 – 0:Normal, 1–2:Pol0–1, 3–6:PW0–3 : 0
  Trigger #2 – 0:Normal, 1–2:Pol0–1, 3–6:PW0–3 : 0
  Trigger #3 – 0:Normal, 1–2:Pol0–1, 3–6:PW0–3 : 0
  Trigger #4 – 0:Normal, 1–2:Pol0–1, 3–6:PW0–3 : 1
  Trigger #5 – 0:Normal, 1–2:Pol0–1, 3–6:PW0–3 : 0
  Trigger #6 – 0:Normal, 1–2:Pol0–1, 3–6:PW0–3 : 4
```

These questions make it possible to reassign the waveforms that are driven onto the six user trigger (TRIG1–6) BNC outputs on the backpanel of the RVP7. This makes it easier to adapt the external cabling of the RVP7 so as to make better use of the available BNC connectors and related 15V drivers. You may substitute either of the two polarization control lines or the four pulsewidth control lines in place of any of the six normal triggers.

In the example above, triggers #1, #2, #3, and #5 are all driven with their normal waveforms. However trigger #4 will have a copy of the POLAR0 polarization control line, and trigger #6 will have a copy of the PWBW1 pulsewidth control line. Neither POLAR0 nor PWBW1 themselves are changed by these assignments.

Whenever any of the six user trigger lines is reassigned from its normal setting, the plot of that trigger within the **Pb** command will show a hashed line across the screen. This is a graphical reminder that that trigger has been replaced by some other waveform.

```
Merge triggers to create composite waveforms: YES
  Merge Trigger #1 into : #1:  #2:  #3:  #4:  #5:  #6:
  Merge Trigger #2 into : #1:  #2:  #3:  #4:  #5:  #6:
  Merge Trigger #3 into : #1:Y #2:  #3:  #4:  #5:  #6:
  Merge Trigger #4 into : #1:  #2:Y #3:  #4:  #5:  #6:
  Merge Trigger #5 into : #1:  #2:Y #3:  #4:  #5:  #6:
  Merge Trigger #6 into : #1:  #2:  #3:  #4:  #5:  #6:
```

These questions allow you to merge the six user triggers together; resulting in trigger patterns that can be much more complex. In this example, Trigger #3 will be merged into Trigger #1; Trigger #3 will be unaltered, and Trigger #1 will be the "OR" of itself with Trigger #3. Likewise, Triggers #4 and #5 will be merged into Trigger #2 so that the later will contain three distinct pulses within each PRT. Answer each question with a sequence of up to six "Y" or "N" responses in order to set the merged destinations for each trigger line.

Note that the six triggers are still defined in the usual way in the **Mt<n>** menu, i.e., start time, width, etc. The only change is that you may now combine these individual pulse definitions into a more complex composite output waveform.

3.3.5 Mt<n> — Triggers for Pulsewidth #n

These questions are accessed by typing "Mt", with an additional argument giving the pulsewidth number. They configure specific trigger and FIR bandpass filter properties for the indicated pulsewidth only.

```
Trigger #1 – Start:    0.00 usec
        #1 – Width:    1.00 usec     High:YES
Trigger #2 – Start:    0.00 usec + ( 0.500000 * PRT )
        #2 – Width:   10.00 usec     High:YES
Trigger #3 – Start:   –3.00 usec
        #3 – Width:    1.00 usec     High:YES
Trigger #4 – Start:   –2.00 usec
        #4 – Width:    1.00 usec     High:YES
Trigger #5 – Start:   –1.00 usec
        #5 – Width:    1.00 usec     High:YES
Trigger #6 – Start:   –5.00 usec + (–0.001000 * PRT )
        #6 – Width:    2.00 usec     High:NO
```

These parameters list the starting times (in microseconds relative to range zero), the widths (in microseconds), and the active sense of each of the six triggers generated by the internal trigger generator. Setting a width to zero inhibits the trigger on that line.

The Start Time can include an additional term consisting of the pulse period times a fractional multiplier between –1.0 and +1.0. This allows you to produce trigger patterns that would not otherwise be possible, e.g., a trigger that occurs half way between every pair of transmitted pulses, and remains correctly positioned regardless of changes in the PRF Enter this multiplier as "0" if you do not wish to use this term, and it will be omitted entirely from the printout..

In the above example, Trigger #2 is a 10.0 μsec active-high pulse whose leading edge occurs precisely halfway between the zero-range of every pair of pulses. Likewise, Trigger #6 is a 2.0 μsec active-low pulse whose falling edge is nominally 5.0 μsec prior to range zero, but which is advanced by 1.0 μsec for every millisecond of trigger period. All other triggers behave normally, and have fixed starting times that do not vary with trigger period.

Some subtleties of these variable start times are:

- The PRT multipliers can only be used in conjunction with the RVP7's internal trigger generator. The PRT-relative start times are completely disabled whenever an external trigger source is chosen from the **Mt** menu.
- When PRT-relative triggers are plotted by the **Pb** command, the active portion of the trigger will be drawn cross-hatched and at a location computed according to the current PRF. The cross-hatching serves as a reminder that the actual location of that trigger may vary from it's presently plotted position.
- The PRT multiplier for a given pulse is applied to the interval of time between that pulse and the next one. This distinction is important whenever the RVP7 is generating multiple-PRT triggers, e.g., during DPRT mode, or during Dual-PRF processing. Multipliers from 0.0 to +1.0 are generally safe to use because they shift the trigger into the same pulse period that originally defined it. For example, a start time of (0.0 μsec + (0.98 * PRT)) would position a trigger 98% of the way up to the next range zero. But, if –0.98 were used, and if the period of

the previous pulse was shorter than the current one, then that shorter period would become incorrect (longer) as a result of having to fit in the very early trigger.

A small but important detail is built into the algorithm for producing the six user trigger waveforms. It applies whenever a) the trigger period is internally determined, i.e., the external pretrigger input is not being used, and b) the overall span of the six trigger definitions combined does not fit into that period. What happens in this case is that any waveforms that do not fit will be zeroed (not output) so that the desired period is preserved. This means that you can define triggers with large positive start times, and they will pop into existence only when the PRF is low enough to accommodate them.

For example, if Trigger #2 is defined as a 200.0μsec pulse starting at +400.0μsec, then that trigger would be suppressed if the PRF were 2000Hz, but it would be present at a PRF of 1000Hz. Whenever a trigger does not completely fit within the overall period it is suppressed entirely. Thus, even though the +400.0μsec start time is still valid at 2000Hz, the entire 200.0μsec pulse would not fit, and so the pulse is eliminated altogether.

Start limits: –5000 to 5000 μsec.   Width limits: 0 to 5000 μsec.

`Maximum number of Pulses/Sec:  2000.0`
`Maximum instantaneous 'PRF' :  2000.0 (/Sec)`

These are the PRF protection limits for this pulsewidth.

The wording of the "Maximum number of Pulses/Sec" question serves as a reminder that the number shown is not only an upper bound on the PRF, but also a duty cycle limit when DPRT mode is enabled.

The "Maximum instantaneous 'PRF'" question allows you to configure the maximum instantaneous rate at which triggers are allowed to occur, i.e., the reciprocal of the minimum time between any two adjacent triggers. This parameter is included so that you can limit the maximum DPRT trigger rate individually for each pulsewidth. Note that the maximum instantaneous PRF can not be set lower than the maximum number of pulses per second.

PRF limits: 50 to 20000Hz.

`External pretrigger delay to range zero: 3.00 usec`

Range Zero is time at which the signal from a target at zero range would appear at the radar receiver outputs. This parameter adjusts the delay from the active edge of the external trigger to range zero. It is important that this delay be correct when the RVP7 is operating with an external trigger, since the zero range point is a fixed time offset from that trigger. When the transmitter is driven from the internal trigger signals, those signals themselves are adjusted (see Burst Pulse alignment procedures) to accomplish the alignment of range zero.

Limits: 0.1 to 500 μsec.

`Range resolution: 125.00 meters`

The range resolution of the RVP7 is determined by the decimation factor of the digital matched FIR filter that computes "I" and "Q". This decimation factor is the ratio of the filter's input and output data rates, and can be any integer from six to sixteen. The Acquisition Clock (See **Mc** Section) sets the input data rate. At its standard frequency of 35.9751MHz, the available range resolutions (in meters) are: 50.0, 58.3, 66.7, 75.0, 83.3, 91.7, 100.0, 108.3, 116.7, 125.0, and 133.3.

The ranges that are selected by the bit mask in the LRMSK command are spaced according to the range resolution that is chosen here. Also, the upper limit on the impulse response length of the matched FIR filter (see below) is constrained by the range resolution. If you choose a range resolution that can not be computed at the present filter length, then a message of the form: "Warning: Impulse response shortened from 72 to 42 taps" will appear.

Limits: 50.0 to 133.3 meters.

`FIR-Filter impulse response length: 1.33 usec`

The RVP7 computes "I" and "Q" using a digital FIR (Finite Impulse Response) matched filter. The length of that filter (in microseconds) is chosen here. At the standard Acquisition Clock rate of 35.9751MHz, a 1.00 microsecond impulse response corresponds to a filter that is 36 taps long.

The filter length should be based on several considerations:

- It should be at least as long as the transmitted pulsewidth. If it were shorter, then some of the returned energy would be thrown away when "I" and "Q" are computed at each bin. The SNR would be reduced as a result.
- It should be at least as long as the range bin spacing. The goal here is to choose the longest filter that retains statistical independence among successive bins. If the filter length is less than the bin spacing, then no IF samples would be shared among successive bins, and those bins would certainly not be correlated.
- It should be "slightly longer" than either of the above bounds would imply, so that the filter can do a better job of rejecting out-of-band noise and spurious signals. The SNR of weak signals will be improved by doing this.

In practice, a small degree of bin-to-bin correlation is acceptable in exchange for the filter improvements that become possible with a longer impulse response. The FIR coefficients taper off to zero on each end; hence, the power contributed by overlapping edge samples is minimal. SIGMET recommends beginning with an impulse response length of 1.2–1.5 times the pulsewidth or bin spacing, whichever is greater.

The maximum possible filter length is bounded according to the range resolution that has been chosen; a finer bin spacing leaves less time for computing a long filter. For the RVP7 Rev.A processor, the filter length must be less than 2.92 μsec at 125-meter resolution; for Rev.B and higher this limit increases to 6.67 μsec.

NOTE: Cascade filter software is being contemplated that will extend the maximum impulse response length to at least 50 μsec. This is of interest when very long (uncoded CW) transmitted pulses are used.

FIR-Filter prototype passband width: 0.503 MHz

This is the passband width of the ideal lowpass filter that is used to design the matched FIR bandpass filter. The actual bandwidth of the final FIR filter will depend on 1) the filter's impulse response length, and 2) the design window used in the process. The actual 3dB bandwidth will be:

- Larger than the ideal bandwidth if that bandwidth is narrow and the FIR length is too short to realize that degree of frequency discrimination. In these cases it may be reasonable to increase the filter length.

- Smaller than the ideal bandwidth if the FIR length easily resolves the frequency band. This is because of the interaction within the filter's transition band of the ideal filter and the particular design window being used. For example, for a Hamming window and sufficiently long filter length, the ideal bandwidth is an approximation of the 6dB (not 3dB) attenuation point. Hence, the 3dB width is narrower than the ideal prototype width.

This parameter should be tuned using the TTY output and interactive visual plot from the "**Ps**" command. The actual 3dB bandwidth is shown there, so that it can be compared with the ideal prototype bandwidth.

Limits: 0.05 to 10.0 MHz.

Output control 4–bit pattern: 0001

These are the hardware control bits for this pulsewidth. The bits are the 4-bit binary pattern that is output on PWBW0:3

Bit Limits: 0 to 15 (input must be typed in decimal)

Current noise level: –75.00 dBm
Powerup noise level: –75.00 dBm
–or–
Current noise levels – PriRx: –75.00 dBm, SecRx: –75.00 dBm
Powerup noise levels – PriRx: –75.00 dBm, SecRx: –75.00 dBm

These questions allow you to set the current value and the power-up value of the receiver noise level for either a single or dual receiver system. The noise level(s) are shown in dBm, and you may alter either one from the TTY. The power-up level(s) are assigned by default when the RVP7 first starts up, and whenever the RESET opcode is issued with Bit #8 set. Likewise, the current noise level is revised whenever the SNOISE opcode is issued. These setup questions are intended for applications in which the RVP7 must operate with a reasonable default value, up until the time that an SNOISE command is actually received. They may also be used to compare the receiver noise levels during normal operation, which serves as a check that each FIR filter is behaving as expected when presented with thermal noise.

`Transmitter phase switch point: –1.00 usec`

This is the transition time of the RVP7's phase control output lines during random phase processing modes. The switch point should be selected so that there is adequate settling time prior to the burst/COHO phase measurement on each pulse. This question only appears if the PHOUT[0:7] lines are actually configured for phase control (See Section 3.3.1).

Limits: –500 to 500 μsec.

`Polarization switch point for POLAR0:  –1.00 usec`
`Polarization switch point for POLAR1:   1.00 usec`

The RVP7's POLAR0 and POLAR1 digital output lines control the polarization switch in a dual-polarization radar. During data processing modes in which the polarization alternates from pulse to pulse, the transition points of these control signals are set by these two questions. The values are in microseconds relative to range zero; the same units used to define the start times of the six user triggers. The logical sense of POLAR0 and POLAR1 is set by questions described in Section 3.3.4.

Limits: –500 to 500 μsec.

3.3.6 Mb — Burst Pulse and AFC

These questions are accessed by typing "Mb". They set the parameters that influence the phase and frequency analysis of the burst pulse, and the operation of the AFC feedback loop.

`Receiver Intermediate Frequency: 30.0000 MHz`

This is the center frequency of the IF receiver and burst pulse waveform. The RVP7 can operate at an intermediate frequency from any of the three alias bands 22–32MHz, 40–50MHz, and 58–68MHz. These bands are delineated by 4MHz safety zones on either side of integer multiples of half the RVP7/IFD's 36MHz sampling frequency. The value entered here implicitly defines the band, and hence, the boundaries of the 18MHz window in which the IF is assumed to fall.

Limits: 22 to 68 MHz.

`Primary Receiver Intermediate Frequency:   30.0000 MHz`
`Secondary Receiver Intermediate Frequency: 24.0000 MHz`

These alternate questions will replace the previous question whenever the RVP7's dual-receiver mode is selected. You should enter the two intermediate frequencies for your primary and secondary (nominally horizontal and vertical polarized) receivers. Note that you can easily swap receiver channels merely by exchanging the two frequency values.

`IF increases for an approaching target: YES`

The intermediate frequency is derived at the receiver's front end by a microwave mixer and sideband filter. The filter passes either the lower sideband or the upper sideband, and rejects the other. Depending on which sideband is chosen, an increase in microwave frequency may either increase (STALO below transmitter) or decrease (STALO above transmitter) the receiver's intermediate frequency. This question influences the sign of the Doppler velocities that are computed by the RVP7.

PhaseLock to the burst pulse: YES

This question controls whether the RVP7 locks the phase of its synthesized "I" and "Q" data to the measured phase of the burst pulse. For an operational magnetron system this should always be "YES", since the transmitter's random phase must be known in order to recover Doppler data. The "NO" option is appropriate for non phase modulated Klystron systems in which the RVP7/IFD sampling clock is locked to the COHO. It is also useful for bench testing in general. In these "NO" cases the phase of "I" and "Q" is determined relative to the stable internal sampling clock in the RVP7/IFD module.

Minimum power for valid burst pulse: –15.0 dBm

This is the minimum mean power that must be present in the burst pulse for it to be considered valid, i.e., suitable for input into the algorithms for frequency estimation and AFC. The reporting of burst pulse power is described in Section 4.4; the value entered here should be, perhaps, 8 dB less. This insures that burst pulses will still be properly detected even if the transmitter power fades slightly.

The mean power level of the burst is computed within the narrowed set of samples that are used for AFC frequency estimation. The narrow subwindow will contain only the active portion of the burst, and thus a mean power measurement is meaningful. The full FIR window would include the leading and trailing pulse edges and would not produce a meaningful average power. Since radar peak power tends to be independent of pulse width, this single threshold value can be applied for all pulsewidths.

Limits: –60 to +10 dBm.

Design/Analysis Window– 0:Rect, 1:Hamming, 2:Blackman : 1

You may choose the window that is used in 1) the design of the FIR matched filter, and 2) the presentation of the power spectra for the various scope plots. Choices are rectangular, Hamming, and Blackman; the Hamming window being the best overall choice. The Blackman window is useful if you are trying to see plotted spectral components that are more than 40dB below the strongest signal present. It is especially useful in the "Pr" plot when a long span of data are available. FIR filters designed with the Blackman window will have greater stopband attenuation than those designed with the Hamming window, but the wider main lobe may be undesirable. The rectangular window is included mostly as a teaching tool, and should never be used in an operational setting.

Settling time (to 1%) of burst frequency estimator: 5.0 sec

The burst frequency estimator uses a 4^{th} order correlation model to estimate the center frequency of the transmitted pulses. Each burst pulse will typically occupy approximately one microsecond; yet the frequency estimate feeding the AFC loop needs to be accurate to, perhaps, 10KHz. Obviously this accuracy can not be achieved using just one pulse. However, several hundred of the (unbiased) individual estimates can be averaged to produce an accurate mean. This averaging is done with an exponential filter whose time constant is chosen here.

Limits: 0.1 to 120 seconds.

`Lock IFD sampling clock to external reference: NO`

This question determines the usage of the shared SMA connector that is labeled "AFC/(CLK)" on the RVP7/IFD. It is generally *not* necessary to phase lock the IFD sampling clock to the radar system clock, since very good stability is obtained from the burst phase measurements during normal operation. However, two cases that benefit from clock locking are 1) using the RVP7 in a klystron system where an external trigger is provided, and 2) dual-receiver systems in which computation of Φ_{DP} is important.

The following two questions will appear only if you have requested that the IFD sampling clock be locked to an external clock reference. See Section 2.1.6 for a description of the hardware setups that must accompany this selection.

`PLL ratio of (1/1) ==> Input reference at 17.9876 MHz`

The VCXO phase-locked-loop (PLL) in the RVP7/IFD can work with any input reference clock whose frequency is a rational multiple (P/Q) of half the desired sampling frequency, i.e., center frequency of the VCXO. This question allows this ratio to be established. In general, the best PLL performance will be attained when the ratio is reduced to lowest terms, e.g., use a ratio of 6/5 rather than 12/10.

Limits: 1 to 128 for both numerator and denominator.

`VCXO has positive frequency deviation: YES`

Most VCXOs have positive frequency deviation, i.e., their output frequency increases with increasing input control voltage. This question will generally be answered "yes", but is included to accommodate the other case as well. The PLL will not lock, and will be completely unstable, if the wrong choice is made.

`Enable AFC and MFC functions: YES`

AFC is required in a magnetron system to maintain the fixed intermediate frequency difference between the transmitter and the STALO. AFC is not required in a klystron system since the transmitted pulse is inherently at the correct frequency.

The following rather long list of questions will appear only if AFC and MFC functions have been enabled.

`AFC Servo– 0:DC Coupled, 1:Motor/Integrator : 0`

The AFC servo loop can be configured to operate with an external Motor/Integrator frequency controller, rather than the usual direct-coupled FM control. This type of servo loop is required for tuned magnetron systems in which the tuning actuator is moved back and forth by a motor, but remains fixed in place when motor drive is removed. These systems require that the AFC output voltage (motor drive) be zero when the loop is locked; and that the voltage be proportional to frequency error while tracking. Please see Section 3.3.6.1 for more details.

`Wait time before applying AFC: 10.0 sec`

After a magnetron transmitter is first turned on, it may be several seconds or even minutes until its output frequency becomes stable. It would not make sense for the AFC loop to be running during this time since there is nothing gained by chasing the startup transient. This question allows you to set a holdoff delay from the time that valid burst pulses are detected to the time that the AFC loop actually begins running.

Limits: 0 to 300 seconds.

AFC hysteresis -- Inner: 5.0 KHz, Outer: 15.0 KHz

These are the frequency error tolerances for the AFC loop. The loop will apply active feedback whenever the outer frequency limit is exceeded, but will hold a fixed level once the inner limit has been achieved. The hysteresis zone minimizes the amount of thrashing done by the feedback loop. The AFC control voltage will remain constant most of the time; making small and brief adjustments only occasionally as the need arises.

AFC outer tolerance during data processing: 50.0 KHz

In general, the AFC feedback loop is active only when the RVP7 is not processing data rays. This is because the Doppler phase measurements are seriously degraded whenever the AFC control voltage makes a change. To avoid this, the AFC loop is only allowed to run in between intervals of sustained data processing. This is fine as long as the host computer allows a few seconds of idle time every few minutes; but if the RVP7 were constantly busy, the AFC loop would never have a chance to run. This question allows you to place an upper bound on the frequency error that is tolerated during sustained data processing. AFC is guaranteed to be applied whenever this limit is exceeded.

Limits: 15 to 4000 KHz.

AFC feedback slope: 0.0100 D-Units/sec / KHz
AFC minimum slew rate: 0.0000 D–Units/sec
AFC maximum slew rate: 0.5000 D-Units/sec

These questions control the actual feedback computations of the AFC loop.

The overall span of the AFC output voltage is set by Gain and Offset potentiometers on the RVP7/IFD module (See Section 2.1.5). The control level that is applied to the AFC's 16-bit Digital-to-Analog converter is specified here in "D-Units", i.e., arbitrary units ranging from –100 to +100 corresponding to the complete span of the D/A converter. Since the D–Unit corresponds in a natural way to a percentage scale, the shorter "%" symbol is sometimes used.

AFC feedback will be applied in proportion to the frequency error that the algorithm is attempting to correct. The feedback slope determines the sensitivity and time constant of the loop by establishing the AFC's rate of change in (D-Units / sec) per thousand Hertz of frequency error. For example, a slope of 0.01 and a frequency error of 30KHz would result in a control voltage slew of 0.3 D-Units per second. At that rate it would take approximately 67 seconds for the output voltage to slew one tenth of its total span (20 D-Units / (0.3 D-Units / sec) = 67 sec). AFC is intended to track very slow drifts in the radar system, so response times of this magnitude are reasonable.

Keep in mind that the feedback slew is based on a frequency error which itself is derived from a time averaging process (see Burst Frequency Estimator Settling Time described above) . The AFC loop will become unstable if a large feedback slope is used together with a long settling time constant, due to the phase lag introduced by the averaging process. Keep the loop stable by choosing a small enough slope that the loop easily comes to a stop within the inner hysteresis zone.

See Section 3.3.6.1 for more information about these slope and slew rate parameters.

AFC span– [–100%,+100%] maps into [–32768 , 32767]
AFC format– 0:Bin, 1:BCD, 2:8B4D: 0, ActLow: NO
AFC uplink protocol– 0:Off, 1:Normal, 2:PinMap : 1

The RVP7's implementation of AFC has been generalized so that there is no difference between configuring an analog loop and a digital loop. The AFC feedback loop parameters are setup the same way in each case; the only difference being the model for how the AFC information is made available to the outside world. Many types of interfaces and protocols thus become possible according to how these three questions are answered. AFC output follows these three steps:

- The internal feedback loop uses a conceptual [–100%,+100%] range of values. However, this range may be mapped into an arbitrary numeric span for eventual output. For example, choosing the span from –32768 to +32767 would result in 16-bit AFC, and 0 to 999 might be appropriate for 3-digit BCD; but any other span could also be selected from the full 32-bit integer range.
- Next, an encoding format is chosen for the specified numeric span. The result of the encoding step is another 32-bit pattern which represents the above numeric value. SIGMET will make an effort to include in the list of supported formats all custom encodings that our customers encounter from their vendors.

  Available formats include straight binary, BCD, and mixed-radix formats that might be required by a specialized piece of equipment. The "8B4D" format encodes the low four decimal digits as four BCD digits, and the remaining upper bits in binary. For example, 659999 base-10 would encode into 0x00419999 Hex.
- Finally, an output protocol is selected for the bit pattern that was produced by encoding the numeric value. The bits may be written to the eight RVP7/Main backpanel RS232 outputs, or sent on the uplink as a value to be received by the RVP7/IFD and converted to an analog voltage. Yet another option is for the bits to be sent on the uplink and received by the RVP7/DAFC board, which supports arbitrary remapping of its output pins.

To summarize: the internal AFC feedback level is first mapped into an arbitrary numeric span, then encoded using a choice of formats, and finally mapped into an arbitrary set of pins for digital output. We are hopeful that this degree of flexibility will allow easy hookup to virtually any STALO synthesizer that one might encounter.

```
PinMap Table (Type '31' for GND, '30' for +5)
---------------------------------------------
Pin01:00   Pin02:01   Pin03:02   Pin04:03   Pin05:04
Pin06:05   Pin07:06   Pin08:07   Pin09:08   Pin10:09
Pin11:10   Pin12:11   Pin13:12   Pin14:13   Pin15:14
Pin16:15   Pin17:16   Pin18:17   Pin19:18   Pin20:19
Pin21:20   Pin22:21   Pin23:22   Pin24:23   Pin25:24
FAULT status pin (0:None): 0,  ActLow: NO
```

These questions only appear when the "PinMap" uplink protocol has been selected. The table assigns a bit from the encoded numeric word to each of the 25 pins of the RVP7/DAFC module. For example, the default table shown above simply assigns the low 25 bits of the encoded bit pattern to pins 1-25 in that order. You may also pull a pin high or low by assigning it to +5 or GND. Note that such assignments produce a logic-high or logic-low signal level, not an actual power or ground connection. The latter must be done with actual physical wires.

One of the RVP7/DAFC pins can optionally be selected as a Fault Status indicator. You may choose which pin to use for this purpose, as well as the polarity of the incoming signal level. Note that the standard RVP7/DAFC module only supports the selection of pins 1, 3, 4, 13, 14, and 25 as inputs. This setup question allows you to choose any pin, however, because it does not know what kind of hardware may be listening on the uplink and what its constraints might be.

`Burst frequency increases with increasing AFC voltage: NO`

If the frequency of the transmit burst increases when the AFC control voltage increases, then answer this question "Yes"; otherwise answer "No". When this question is answered correctly, a numerical increase in the AFC drive (D–Units) will result in an increase in the estimated burst frequency. If the AFC loop is completely unstable, try reversing this parameter.

`Mirror AFC voltage on– 0:None, 1:I, 2:Q : 0`

AFC/MFC can be mirrored on a backpanel output of the main chassis using this question. When either "I" or "Q" is selected, the AFC/MFC voltage will be present on the corresponding BNC output, and the other output will be used for scope plotting. This configuration would be useful, for example, in a dual-receiver magnetron system that needs a phase locked acquisition clock in the RVP7/IFD, but also needs an AFC tuning voltage to control the transmit frequency. When "None" is selected, scope plotting will revert to its normal "Q" output.

The voltage range of the "I" and "Q" outputs is approximately $\pm$1 Volt, and is not adjustable. When AFC/MFC is mirrored on these lines, you will probably need to add an external Op-Amp circuit to adjust the voltage span and offset to match your RF components. We also recommend that you add significant low-pass filtering (cutoff at 3Hz) to remove any power line noise or crosstalk that may be originating within the RVP7/Main chassis.

```
Enable Burst Pulse Tracking: YES
```

This question enables the Burst Pulse Tracking algorithm that is described in Section 5.1.3. Remarkably, for such an intricate new feature, there are no additional parameters to configure. The characteristic settling times for the burst are already defined elsewhere in this menu, and the tracking algorithm uses dynamic thresholds to control the feedback.

```
Enable Time/Freq hunt for missing burst: No
  Number of frequency intervals to search: 5
  Settling time for each frequency hop: 0.25 sec
  Automatically hunt immediately after being reset: YES
    Repeat the hunt every: 60.00 sec
```

These questions configure the process of hunting for a missing burst pulse. The trigger timing interval that is checked during Hunt Mode is always the maximum ±20μsec; hence no further setup questions are needed to define the hunting process in time. The hunt in frequency is a different matter. The overall frequency range will always be the full –100% to +100% AFC span; but the number of subintervals to check must be specified, along with the STALO settling time after making each AFC change. With the default values shown, AFC levels of –66%, –33%, 0%, +33%, and +66% will be tried, with a one-quarter second wait time before checking for a valid burst at each AFC setting.

You should choose the number of AFC intervals so that the hunt procedure can deduce an initial AFC level that is within a few megaHertz of the correct value. The normal AFC loop will then take over from there to keep the radar in tune. For example, if your radar drifts considerably in frequency so that the AFC range had to be as large as 35MHz, then choosing fifteen subintervals might be a good choice. The hunt procedure would then be able to get within 2.3MHz of the correct AFC level. The settling time can usually be fairly short, unless you have a STALO that wobbles for a while after making a frequency change. Note that hunting in frequency is not allowed for Motor/Integrator AFC loops, and the two AFC questions will be suppressed in that case.

The RVP7 can optionally begin hunting for a missing burst pulse immediately after being reset, but before any activity has been detected from the host computer. This might be useful in systems that both drift a lot and generally have their transmitter *On*. However, this option is really included just as a work around; the correct way for a burst pulse hunt to occur is via an explicit request from the host computer which "knows" when the pulse really should be present. Blindly hunting in the absence of that knowledge can not be done because there are many reasons why the burst pulse may legitimately be missing, e.g., during a radar calibration.

The automatic hunt for the burst pulse will always run at least once whenever the feature is enabled. The automatic hunting ceases, however, as soon as any activity is detected from the host computer. Only use this feature on radars with a serious drifting problem with their burst pulse timing.

```
Simulate burst pulse samples: NO
```

The RVP7 can simulate a one microsecond envelope of burst samples. This is useful only as a testing and teaching aid, and should never be used in an operational system.

A two-tone simulation will be produced when the RVP7 is setup in dual-receiver mode. The pulse will be the sum of two transmit pulses at the primary and secondary intermediate frequencies. To make the simulation more realistic, the two signal strengths are unequal; the primary pulse is 3dB stronger than the secondary pulse.

`Frequency span of simulated burst: 27.00 MHz to 32.00 MHz`

The simulated burst responds to AFC just as a real radar would. The frequency span from minimum AFC to maximum AFC is given here.

3.3.6.1 AFC Motor/Integrator Option

The question "*AFC Servo– 0:DC Coupled, 1:Motor/Integrator*" selects whether the AFC loop runs in the normal manner (direct control over frequency), or with an external Motor/Integrator type of actuator. The question "*AFC minimum slew request:...*" provides additional control when interfacing to mechanical actuators whose starting and sustaining friction needs to be overcome.

The DC-Coupled AFC loop questions (changes shown in bold) are:

```
AFC Servo– 0:DC Coupled, 1:Motor/Integrator : 0
Wait time before applying AFC: 10.0 sec
AFC hysteresis– Inner: 5.0 KHz,  Outer: 15.0 KHz
AFC outer tolerance during data processing: 50.0 KHz
AFC feedback slope:     0.0100 D–Units/sec / KHz
AFC minimum slew rate:  0.0000 D–Units/sec
AFC maximum slew rate:  0.5000 D–Units/sec
```

and the Motor/Integrator loop questions are:

```
AFC Servo– 0:DC Coupled, 1:Motor/Integrator : 1
Wait time before applying AFC: 10.0 sec
AFC hysteresis– Inner: 5.0 KHz,  Outer: 15.0 KHz
AFC outer tolerance during data processing: 50.0 KHz
AFC feedback slope:         1.0000 D–Units / KHz
AFC minimum slew request:  15.0000 D–Units
AFC maximum slew request:  90.0000 D–Units
```

Notice that the physical units for the feedback slope and slew rate limits are different in the two cases. In the DC-Coupled case the AFC output voltage controls the frequency directly, so the units for the feedback and slew parameters use *D-Units/Second*. In the Motor/Integrator case, the AFC output determines the rate of change of frequency; hence *D-Units* are used directly.

The above example illustrates typical values that might be used with a Motor/Integrator servo loop. The feedback slope of 1.0 *D-Units/KHz* means that a frequency error of 100KHz would produce the full-scale (100 *D-Units*) AFC output. But this is modified by the minimum and maximum slew requests as follows:

- A zero *D-Unit* output will always be produced whenever AFC is locked.
- When AFC is tracking, the output drive will always be at least ± 15 *D-Units*. This minimum non-zero drive should be set to the sustaining drive level of the motor actuator, i.e., the minimum drive that actually keeps the motor turning.
- When AFC is tracking, the output drive will never exceed ± 90 *D-Units*. This parameter can be used to limit the maximum motor speed, even when the frequency error is very large.

The AFC Motor/Integrator feedback loop works properly even if the motor has become stuck in a "cold start", i.e., after the radar has been turned off for a period of time. The mechanical starting friction can sometimes be larger than normal, and additional motor drive is required to break out of the stuck condition. But once the motor begins to turn at all, then the normal AFC parameters (minimum slew, maximum slew, feedback slope) all resume working properly. The algorithm operates as follows:

- Whenever AFC correction is being applied, the RVP7 calculates how long it would take to reach the desired IF frequency at the present rate of change. For example, if we are 1MHz away from the desired IF frequency, and the measured rate of change of the IF burst frequency is 20KHz/sec, then it will be 50 seconds until the loop reaches equilibrium.
- Whenever the AFC loop is in Track-Mode, but the time to equilibrium is greater than two minutes, then the "Minimum Slew" parameter will be slowly increased. The idea is to gradually increase the starting motor drive whenever it appears that the IF frequency is not actually converging toward the correct value, i.e., the motor is stuck.
- As soon as the frequency is observed to begin changing, such that the desired IF would be reached in less than two minutes, then the "Minimum Slew" parameter is immediately put back to its correct setup value. The loop then continues to run properly using its normal setup values.

Manual Frequency Control (MFC) operates unchanged in both of the AFC servo modes. Whenever MFC is enabled in the **Ps** command, it always has the effect of directly controlling the output voltage of the AFC D/A converter. The MFC mode can be useful when testing the motor response under different drive levels, and when determining the correct value for the minimum slew request.

3.3.7 M* — Standalone Settings

These questions allow you to fine tune the RVP7's powerup state with sufficient detail that a real time display could be driven without any further setup requirements from a host computer.

`Noise sample PRF: 200.0 Hz`
`Noise sample range: 250.0 km`

These questions allow you to choose the PRF and range for RVP7 noise samples.

Limits: PRF 50 to 6000 Hz; Range 50 to 750 km.

`Max range: 255.0km   Bin spacing: 1000.0m`

These questions allow you to choose the maximum range and bin spacing for the power-up range mask. The starting range is always zero. The quotient of the maximum range divided by the bin spacing must yield a bin count that can be handled by the current processor configuration. If this is not the case, then a larger bin spacing will automatically be filled in.

`Range bins averaged: 1`

Specifies number of contiguous input bins that will be averaged together in range in order to produce each output bin. A value of one means that no range averaging is performed.

Limits: 1 to 16 bins.

`Parameters -- UdBZ:YES, CdBZ:NO , Vel:NO , Wid:NO , PRF:x1`

These questions choose which data parameters will be computed, as well as the type of Dual-PRF trigger that will be generated for velocity unfolding. "x1" means that a fixed PRF is used (no velocity unfolding). "x2" means that a 3:2 trigger ratio is used, resulting in a factor of two increase in unambiguous velocity.

PRF Limits: x1 to x4 Nyquist expansion

`Sample size: 25 pulses`

Specifies the (nominal) number of pulses to use for each output ray. This number is modified if the RVP7 is operating in Dynamic-Angle-Sync mode.

Limits: 1 to 256 pulses.

`Calibration Reflectivity (at 1km): –22.00 dBZ`

Sets the reflectivity calibration value of dBZ_o referenced at 1.0 kilometers.

`LOG output slope: 0.480,  Locked:NO`

This question permits the LOG conversion slope to be viewed, changed, and locked at its present value. The RVP7 uses the LOG slope only for the purpose of converting internal calibrated power levels into the older style [0–256] LOG units when the latter are needed for output. This question allows the RVP7 to work properly with old versions of ZAUTO that would try to change the slope midway through the calibration process. By manually locking the RVP7's LOG slope at some reasonable value (we recommend 0.480 dB/#), the reload attempted by an old ZAUTO will be ignored, and the calibration steps can be completed successfully.

`LOG Noise threshold: 0.50 dB`
`Clutter threshold: –25.00 dB`
`Signal PWR threshold: 10.00 dB`
`SQI threshold: 0.50`

These are data quality thresholds, as described in the SOPRM command in Section 6.3.

`Clutter filters -- Doppler:#0, Range:50.0km`

A clutter filter can optionally be applied within a nearby range interval. Specify the filter number (zero means "none"), and maximum range for which it will be applied.

Limits: Filter 0 to 7; Range 0.0 to 1000.0 km.

`Remove speckles: YES`

If set to "YES", then speckles (single bins surrounded by two rejected bins) will be removed from both the LOG and Doppler output data.

`Use angle sync:NO   BCD:NO   Spacing: 1.0000 degrees`

Use these questions to choose whether angle syncing will be used, the format of the angles that are input, and the azimuth spacing between each one.

3.3.8 M+ — Debug Options

A collection of debugging options has been added to the RVP7 to help users with the development and debugging of their applications code. For the most part, these options should remain disabled during normal radar operation. These questions are included so that the RVP7 can be placed into unusual, and perhaps occasionally useful, operating states.

`Simulate TAGs: NO`

Answering "YES" to this question makes it appear that the RVP7 TAG input lines are sampling AZ and EL positions from an antenna that is scanning in azimuth. This can be useful for debugging angle syncing code when TAG connections to a real antenna are unavailable.

`AZ Scan Rate: 0.00 deg/sec`

This question is only asked if we are simulating TAGS. It gives the rate of rotation of the simulated azimuth axis in degrees/second.

Limits: –180 deg/sec to 180 deg/sec.

`EL Position: 0.00 deg`

This question is only asked if we are simulating TAGS. It gives the fixed position of the simulated elevation axis in degrees.

Limits: –180 deg to 180 deg.

`LEDs- 0:Go/User, 1:Go/Proc : 1`

This question permits alternate uses of the front panel LEDs.

- 0 : The green LED is a "go" indicator. The red LED is controlled by software in the host computer. This was the default in the RVP6.
- 1 : The green LED is a "go" indicator. The red LED flashes ON whenever the RVP7 is acquiring and/or processing radar data. This permits you to see when the processor is working, and when it is idle.

`Noise level for simulated data: –50.0 dB`

This is the noise level that is assumed when simulated "I" and "Q" data are injected into the RVP7 via the LSIMUL command. The noise level is measured relative to the power of a full-scale complex (I,Q) sinusoid, and matches the levels shown on the slide pots of the ASCOPE digital signal simulator.

Limits: –100dB to 0dB

`Simulate output rays: NO`

Answering "YES" to this question causes the RVP7 to output bands of simulated data. The bands can occupy a selectable range interval, and span a selectable interval of data values.

`Start bin:0,  Width:10 bins,  Bands:16`

This question is only asked if we are simulating output rays. The Start Bin chooses the bin number (origin zero) where the simulated bands will begin. The width of each band (in bins), and the total number of bands are also selected. The upper limit for all parameters is the maximum bin count for the RVP7 (which depends on board configuration, and number of attached RVP7/AUX boards).

Limits: Start: 0-Max, Width: 1-Max, Bands: 1-Max

Start data value:0, Increment:16

This question is only asked if we are simulating output rays. The data value that will be assigned to the first simulated band, and the data increment from one band to the next, are selected. The permissible values are from 0 to 65535, i.e., the full unsigned 16-bit integer range. This full range is useful when simulating 16-bit output data; for the more typical 8-bit output formats, only the low byte of the start and increment are significant.

Limits: 0 to 65535

Number of slave DSPs to use (0=ALL): All

The default value of zero means to probe for additional RVP7/AUX boards, and then use as many DSPs as could be found. The RVP7 powerup diagnostics will flag an error (Bit #12 of GPARM Word #12) if the DSP count was not 3, 13, or 23. This helps detect a partially failed RVP7/AUX board that is reporting back fewer than its full complement of ten DSP chips.

A non-zero value means to probe for exactly that many chips, ignoring others that may be further down the chain. Bit #14 of GPARM Immediate Status #2 will be set in this case to indicate that the RVP7 may not be utilizing the full quota of available DSP chips. In addition, the powerup diagnostics will flag an error if the DSP count does not exactly match the requested value.

Real Time TTY Monitor: NO

The Real Time TTY Monitor is a stream of characters that are continuously sent to the serial output port of the setup TTY, and which monitor selectable internal variables. When this live monitor is enabled, status lines will be printed whenever the RVP7 is not chatting, i.e., whenever it is either idle or performing normal data processing operations. Normally the setup TTY would be inactive during that time. You may choose the update rate, and which parameters are to be printed, using the questions that optionally follow.

```
Update rate: 2.00 lines/sec
Show burst frequency: NO
Show burst pulse power: NO
Show AFC information: NO
Show pulse width: NO
Show PRF: NO
Show LOG noise: NO
Show Polarization: NO
Show IFD and link info: NO
Show host computer I/O: NO
Show burst timing slew: NO
```

Note that the live monitoring functions are only available on the setup TTY port, and not via "chat" mode on the host computer. This is because the chat interface would interfere with the normal collection of data by the host computer. Also, no input is recognized from the TTY, other than the usual <ESC> key to enter the menu chat.

Most of the data fields are printed in self-explanatory scientific units. The PRF is in Hertz, but is printed without an "Hz" suffix. For historical reasons, the LOG Noise is printed in old fashioned 8-bit A/D units, taken as the upper eight bits of the 12-bit long LOG format described in Section 6.7. An 8-bit *NSE* value may be converted to an absolute dBm level using:

$$dBm = P_{Max} + 16\,s\,(NSE - 224)$$

Where s is the LOG Slope (nominally equal to 0.03), and P_{Max} is +4.5dBm for the 12-bit IFD and +6.0dBm for the 14-bit IFD.

3.3.9 Mz — Transmitter Phase Control

These questions are used to configure the 8-Bit phase modulation codes that may be used to control the phase of a coherent transmitter. The RVP7 will output a pseudo-random sequence of phase codes that are chosen from a specified set of available codes, i.e., all 8-bit patterns that are valid for the phase modulation hardware. The random sequence is output only when the RVP7 is in one of its random phase processing modes (time series or parameter). At all other times, a fixed "idle" phase code pattern is output. See also Sections 3.3.1 and 3.3.5 where related phase control questions are found.

`8–Bit code to output when idle: 0x00`

This is the bit pattern to be output whenever the RVP7 is not in a random phase processing mode. Note that this "idle" code does not have to be one of the "active"codes that are enabled below.

```
Selection of Valid 8-Bit States
-------------------------------
00–0F: Y – – – – – – –  – – – – – – – –
10–1F: – – – – – – – –  – – – – – – – –
20–2F: – – – – – – – –  – – – – – – – –
30–3F: – – – – – – – –  – – – – – – – –
40–4F: – – – – – – – –  – – – – – – – –
50–5F: – – – – – – – –  – – – – – – – –
60–6F: – – – – – – – –  – – – – – – – –
70–7F: – – – – – – – –  – – – – – – – –
80–8F: – – – – – – – –  – – – – – – – –
90–9F: – – – – – – – –  – – – – – – – –
A0–AF: – – – – – – – –  – – – – – – – –
B0–BF: – – – – – – – –  – – – – – – – –
C0–CF: – – – – – – – –  – – – – – – – –
D0–DF: – – – – – – – –  – – – – – – – –
E0–EF: – – – – – – – –  – – – – – – – –
F0–FF: – – – – – – – –  – – – – – – – –
```

This set of questions defines the subset of active 8-bit codes that are valid states for the transmit phase modulator. Answer each line with a sequence of Y's or N's to indicate whether the corresponding 8-bit code is enabled. Only the codes that appear with a "Y" will be used by the RVP7; the "–" indicates an unused code. The "–' character was used instead of "N" so that the visual contrast of the printed table would be improved.

As an example, if your klystron transmitter has an octant phase modulator that is controlled by three digital lines, you might enable phase codes zero through seven, and then cable the modulator to the low three bits of the 8-bit code. The upper five bits would not need to be used in this case.