NOAA Technical Report OAR ETL-69

RADS: A RADAR ACQUISITION AND DISPLAY SYSTEM FOR RESEARCH RADARS

J.S. Gibson
W.C. Campbell
B.E. Martner

Environmental Technology Laboratory
Boulder, Colorado
March 2004

noaa NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION / Oceanic and Atmospheric Research Laboratories

NOAA Technical Report OAR ETL-69

RADS: A RADAR ACQUISITION AND DISPLAY SYSTEM FOR RESEARCH RADARS

Janet S. Gibson
W. Carroll Campbell
Brooks E. Martner

Environmental Technology Laboratory
Boulder, Colorado
March 2004

UNITED STATES DEPARTMENT OF COMMERCE

Donald L. Evans
Secretary

NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION

VADM Conrad C. Lautenbacher, Jr.
Under Secretary for Oceans and Atmosphere/Administrator

Oceanic and Atmospheric Research Laboratories

Richard D. Rosen
Director

NOTICE:

Contents

FIGURE

TABLES

RADS: A RADAR ACQUISITION AND DISPLAY SYSTEM FOR RESEARCH RADARS

Janet S. Gibson, W. Carroll Campbell, Brooks E. Martner

ABSTRACT. The Radar Acquisition and Display System (RADS) is a transportable system designed to acquire, record and display data from a research Doppler radar with polarization diversity in real-time. It has been successfully demonstrated on several different ground-based radars as well as airborne radar systems in the past few years. The RADS has enabled NOAA's Environmental Technology Laboratory (ETL) and United States Coast Guard (USCG) radars to take more data faster than ever before, while allowing the operator to gain meaningful insights by viewing real-time displays of the data. This enables the operator to detect hardware problems immediately and to focus on important phenomena in the data. This document contains a complete description of RADS, including hardware and software, as well as a user's guide.

1. INTRODUCTION

NOAA's Environmental Technology Laboratory (ETL) has been developing and testing a dual-use atmosphere and ocean sensing radar for the past several years. Because this dual-polarized, variable pulse width, X-band radar can scan negative, as well as positive elevation angles, it has been used in several field projects for observing ocean surface features (Kropfli and Clifford, 1996), and numerous projects for over 15 years, where it performed only as an atmospheric radar.

The expanded capabilities of this radar required a new data-acquisition system that allows the radar to take data faster than previous systems and to calculate several additional parameters. With these new parameters come increased demands for real-time recording and display options. In the past the NOAA/ETL atmospheric Doppler radars have had more limited real-time acquisition and display capabilities (Moninger, 1983; Gibson and Martner, 1995).

The Radar Acquisition and Display System (RADS) is a moveable system designed in 1994 to record and display data from a research Doppler radar with polarization diversity. It has been successfully demonstrated on two ground-based radar systems (Ka-band and X-band), and a version was written and demonstrated to accommodate a Side Looking Airborne Radar (SLAR) at X-band. The new Ground-based Remote Sensing Icing Detection System (GRIDS), which integrates a Ka-band radar and a dual-channel microwave radiometer, will also use a derivative of RADS. Most of the discussion below relates to both the ground-based RADS and the SLAR RADS. The SLAR configuration is detailed in section 6.1. As stated earlier, the RADS has been successfully demonstrated for data collection and display in numerous atmospheric and oceanic studies over the past eight years. For a complete list of projects that successfully relied on RADS, see Appendix G.

ETL's scanning X-band and Ka-band radars have been designed for ultimate flexibility in a wide variety of research applications. As such, they demand a very sophisticated data system that can accommodate more versatile scanning and measurement capabilities than operational weather radars, which endlessly repeat a simple sequence of PPI scans with unchanging scan coverage and measurables. In contrast, the ETL radars can conduct complicated sequences of PPI, PPI-sector, RHI, fixed beam, vertical, and coplane scans, each with different azimuth, elevation, minimum and maximum range limits, and corresponding real-time display options. The radars can also shift between various measurement recording modes, including pulse pair and time-series processing, each with specific measurable fields. In some modes more than 30 measurement fields are computed from the raw data and are available for real-time display. This unusually high degree of flexibility and versatility accounts for the complexity of RADS and the need to document its features in this report.

1.1 Overview of the RADS

Physically, the unit consists of a 6U VME chassis, an external color monitor, keyboard and mouse. Connection to the radar is made at four levels: radar timing signals (trigger, polarization control and range gates), receiver (in-phase and quadrature video, log output from both horizontally and vertically polarized receivers), synchro for monitoring azimuth and elevation, and Ethernet for directing antenna scanning through a resident computer. The VME bus is used in this application because of its reliability, bandwidth, and the great number of functions that are available for it.

RADS contains the following VME cards: Radar Timing Generator (RTG), SPARC-based CPU, DSP (Digital Signal Processing) card containing four TMS320C40 DSP chips, a four-channel 20 MHz A/D card with 1 MB buffer memory, a synchro-digital card and a GPS card. Some configurations of RADS also include a motion control card. The chassis also contains a hard disk drive, a CD-ROM drive, and two 8 mm cartridge tape drives. Solaris 2.6.1, SUN Microsystems' version of UNIX, is run on the CPU; no operating system is used on the DSP card.

RADS is initialized by reading in previously defined configuration files at startup. The operator controls RADS through a Graphical User Interface (GUI) window on the color monitor. All radar parameters (see Appendix H) may be viewed, and certain subsets of parameters may be selected to be shown simultaneously on the screen. Thus, while running pulse pair mode, only those parameters that are pertinent to that mode, and that change frequently in that mode, will be displayed. In addition, because of the large number of radar parameters (123 in Version 1.2), they are grouped by eleven primary functions and may be displayed by these primary functions as well. The operator may select numerous functions through "buttons" on the GUI, and thus control processing, recording and real-time displays.

Plots, such as Doppler spectrum, delta-k spectrum, vertical profiles and A-scopes, as well as areal displays, such as PPI (plan-position indicator), RHI (range-height indicator) or B-scan (marching time range displays) are shown on the color monitor. Numerous fields may be selected for display in the areal formats, depending on the processing mode that has been selected. Since all processing modes are programmed, it is relatively easy to implement additional modes.

The RTG generates all timing parameters under program control and can generate pairs of pulses for use with the pulse pair algorithm. Therefore, all timing parameters, such as intra- and inter-pair trigger spacing and range gate delay, range spacing and number are completely programmable limited by the radar being driven and the processing and recording speed of the RADS. The speed limitations of the RADS are dependent upon the mode. In general, modes such as time-series and power spectrum that have little or no data compression are restricted by recording speed of the output devices. Modes such as pulse-pair that use data compression are limited by processing speed. See section 4 for a detailed description of the available modes.

1.2 Hardware/Software Specifications for RADS

Summarized in Table 1 is the set of components needed to create a full-featured RADS capable of antenna control.

Table 1. RADS components

Name	Description	
Chassis	VME, 12 slots (more slots allow for additional options)	Elma 12V-0920-RV12J12A-P500
Processor	Ultra SPARC 2 with a minimum of 128 MB memory, SCSI interface, Ethernet	Themis USPIIi-3V-512-128-300 with Creator Graphics
Display	1280 x 1024 color monitor (minimum resolution)	Various
Analog/Digital	4-channel, 12-bit, 20 MHz with 1 MB swinging buffer memory	Integrated Circuits & Systems ICS-150-AM
Digital Signal Processing	4 TMS320C40 DSP chips, 1 MB local memory, 2 MB global memory	Blue Wave Systems 72-4R1-221-B000
GPS time and position	Time & Frequency Processor with GPS	bc637VME with SV6
Synchro/Digital	2-channel (azimuth & elevation), 14-bit	Computer Conversions VBR-B1B1B1B1-X12-I
Motion Control	2-axis (azimuth & elevation) motion controller	Galil DMC-1300
Radar Timing Generator	Generates pulse-pair triggers and pre-triggers, range gates, transmits and receives polarization control, other timing signals under program control	Custom
Disk Drive	18.2 GB, Ultra wide SCSI	Seagate ST318203LW
Tape Drive	2 Exabyte tape drives, 7 GB per tape, 1 MB/sec record rate, 8 mm tape	Exabyte Eliant 820
Operating Systems	Solaris 2.6	Sun Microsystems
Language	C and C++	GNU
Display Libraries	Xlib, Xview	Sun Microsystems
Window Environment	Openwin	Sun Microsystems

2. RADS HARDWARE

RADS was originally designed to accommodate the faster data rates and dual polarization on the ETL X-Band radar. Because of its adaptable nature, it has also been utilized on the ETL Ka-band radar and the USCG SLAR systems.

NOAA/K is ETL's ground-based 35-GHz (Ka-band 8.6 millimeter wavelength) "cloud"-scanning Ka-band radar. It has been used in various cloud observation studies, including aircraft icing, weather modification studies and climate change. It has been used to observe non-precipitating clouds, drizzle, snowstorms and light rain.

NOAA/D is ETL's ground-based 9.3-GHz (X-band, 3.2 centimeter wavelength) scanning radar. It has been used for measurements of precipitation, boundary-layer airflow, air parcel tracking with chaff, and ocean surface features.

The USCG SLAR is an X-band radar mounted on a C130 aircraft. It is used for observing ocean surface features including mapping of icebergs and oil spills (see section 6).

For further information about these radars, please refer to the following papers: *NOAA/ETL's Polarization-Upgraded X-Band "Hydro" Radar*, Martner, et al (2001), and *An Overview of NOAA/ETL's Scanning Ka-Band Cloud Radar*, Martner, et al (2002).

2.1 System Architecture

All components of the system are purchased as commercial items, with the exception of one VME card, the Radar Timing Generator (RTG), which generates radar triggers, range gates and polarization control signals. Other VME cards are a SPARC-based CPU, a DSP (Digital Signal Processing) card containing four TMS320C40 DSP chips, a four-channel 12 bit 20 MHz A/D card with buffer memory, a GPS receiver, a synchro-digital card, a navigational card for airborne work and a two-axis motion controller card for antenna motion control. All boards are mounted in a VME 12 slot chassis. The chassis also contains a hard disk drive, a CD-ROM drive, and two 8 mm cartridge tape drives. Solaris 2.6.1, SUN Microsystems version of UNIX, is run on the CPU; however no operating system is needed on the DSP card. See Appendix H for a complete description of the Radar Parameters.

In operation, the SPARC CPU runs a radar control program called RCP_SPARC, which communicates with its counterpart, RCP_DSP, running on the DSP via an area in the CPU memory called the slave window which is accessible to any VME bus master. Radar parameters are kept in the slave window, including a mode word that serves to direct the action of the DSP board.

One of the advantages of the VME bus is that it is a true multi-processing bus, thus providing a flexible architecture. This allows for capabilities to be increased by adding processing elements. For example, another CPU or DSP card could be added to increase processing and display capabilities, or a SCSI controller and another tape or disk could be added to increase recording capabilities.

Other possible additions include an X-windows server and a digital receiver card. The addition of an X-windows server would require another monitor, but would provide the system with a monitor completely devoted to displays. The only real limitation is the bandwidth of the VME bus which is theoretically 80 Mbytes/sec. Presently, the only high bandwidth path is between the A/D and the DSP which peaks at 8.2 MB/s, therefore the bus should support a second DSP and/or A/D card. Other proposed cards would make almost no demands on bandwidth because the current architecture dramatically reduces data volume by the processing in the DSP board.

2.2 Data Flow

To start operation, the SPARC CPU sets trigger, gate and polarization information in the RTG. It then sets the mode word in the slave window via the RCP_SPARC process. The DSP, running the program RCP_DSP, reads the mode word and branches to a mode-dependent method. The DSP initializes the A/D board, turns the radar triggers off briefly for synchronization, and then turns the triggers back on. The A/D board begins acquiring data with the first range gate, and runs until an entire “beam” of data is stored in its first buffer memory. At this point, the A/D generates an interrupt that is received by the DSP, which directs the A/D to do a DMA (direct memory access) transfer of its data into the DSP’s global memory. In the meantime, the A/D is filling its second buffer memory which then “swings” back to the first buffer when complete.

When the DMA transfer is complete, another interrupt is generated by the A/D which is a signal to the DSP that the data in its memory are complete and available for processing. The four DSPs divide up the data by range gate, with DSP 0 (the “master”) taking the quarter of the gates at the closest range. Each DSP copies and unpacks its part of the data into its own local memory where all of the processing is done. At this time, the master takes a reading of time, azimuth and elevation via the VMEbus. It then stamps the data with this additional header information. When each DSP is done processing, it copies the data into an output area in global memory and sets a word in global memory indicating that it has finished. When the master completes processing, it copies its data into the output buffer, affixes the packet header containing the packet counter, beam counter, time and antenna position, checks if the other DSPs are finished, waits if necessary, and then generates a VME interrupt to the SPARC indicating that a packet of processed data is available in global memory. Then the DSP repeats the loop by waiting for more data to come in from the A/D.

The processed data in the DSP’s global memory uses a circular buffering scheme with five buffers. The reason for this is UNIX-based systems are sometimes slow to respond to interrupts, and since interrupts routinely occur at rates of about 16 per second (beams/second), experience has shown that when the CPU is heavily loaded, packets will be dropped. With circular buffering, when a packet is read, the program can check to see if another one is available while it still has control of the CPU and, if so, read it immediately. In other words, the program has five times longer to respond to an interrupt before data are lost. In practice, this has virtually eliminated the problem of lost data. However, if data are lost, it will be readily apparent because each packet of data has a serial number.

When the SPARC CPU receives the interrupt from the DSP, it starts a DMA transfer of data into its memory. When complete, the data are split into two parts: data to be recorded and data to be displayed, with each type of data going to a separate process. The SPARC records the data with a UF (Universal Format, see Appendix C) style format, except the data are represented in floating-point format rather than 16-bit integers. The Radar Parameters are recorded in the Local Use header. The recorded data have a UF header affixed to them which includes in the local use header the radar parameters that are used currently. The data are first cached to disk and then written to tape.

The display data is separated into FIFO buffers for each field, which are maintained for 600 packets or beams of data. When the operator selects one field for display, that field is rasterized and output to the screen. This process runs in a continuous loop; thus, if packets of data arrive faster than they can be displayed, all packets available will be drawn to the screen simultaneously. When the operator changes fields, 600 packets of the selected field are immediately displayed. In PPI mode, these are normally enough data to see an entire rotation.

2.3 Control

RADS control as used here is defined to be the control mechanisms used between the SPARC CPU and the DSP board, and can be summarized as follows. All radar parameters are stored in a special section of memory on the CPU card that is mapped for slave accesses from the VME bus, and is referred to as the slave window. This memory can be accessed by any device on the VME bus, in particular the DSP board. One of the radar parameters, DMOD, indicates the data mode and is also used to direct the DSP board. When the DSP is idle (not acquiring data), it reads the DMOD parameter at a 10 Hz rate, waiting for it to change.

If DMOD changes to a valid operating mode, the DSP copies the radar parameters from the slave memory into its own memory. It then accesses these parameters in its memory as needed for initialization of software and registers in the A/D board. When initialization is complete, it turns the RTG off, starts the A/D board, and then turns the RTG on. The SPARC will have previously initialized the RTG to the current parameters. This procedure assures that the first data received comes from the first range gate.

While data is being acquired, it is handled in units called “beams”. A beam of data will consist of a given number of radar triggers, and each trigger will contain a given number of range gates. Since most modes work with pairs of pulses, the number of triggers must be even. The product of the radar triggers and the number of range gates determines the size of the beam in samples. In most modes of operation, a sample consists of data from all four 12-bit A/D converters. Since the 12-bit A/D samples are packed into a two byte word, a sample is eight bytes. Because of hardware limitations, there are several restrictions placed on the number of triggers in the beam and the number of range gates. Software in RADS assures that only valid combinations of parameters are used, and in some cases, alters operator entered parameters slightly to give valid results.

While data is being acquired, the DSP checks parameter DMOD in the slave window after each beam. If DMOD has been set to zero, indicating an idle condition, the DSP terminates data collection and processing. Also if the DSP encounters an error while running, it will write a DMOD value of zero and terminate processing.

Hardware limitations in the A/D board impose certain limitations in data gathering. The first limitation is the size of the two swing buffers, which is 524,288 (2^{19}) bytes apiece. Since a sample is eight bytes, the product of the number of range gates and the number of triggers in the beam must be less than 65,536 or 2^{16}. The second limitation is that the A/D board will only generate a number of range gates that is a power of two, with 128 being the minimum. This restriction is circumvented by software in the DSP which discards extra gates from the A/D whenever a non power-of-two number of gates is requested.

With regard to the first limitation, it should be noted that it is a design feature of the system that no data is discarded between beams. Therefore if longer data sets are desired than can be accommodated by the swing buffers, they can be synthesized later in post-processing by concatenating beams.

The second type of limitation is imposed by the processing speed in the DSP board. This varies depending on the mode and the pulse repetition period (PRP) of the radar. For reference, the ETL ground-based scanning radars typically run with an average PRP of 500 µs. While running the basic pulse pair mode 140, 500 range gates with 128 triggers would compose a beam.

The 128 triggers in the beam result in a beam rate of 15.63 beams/sec (1/(128 x 500 µs)). This is identical to the interrupt request rate to the SPARC CPU. Experience has shown that this is about the maximum rate that can be accommodated with the existing hardware. This is the third type of limitation, having to do with how fast the SPARC can handle interrupts and transfer data from the DSP.

2.4 Calibration

Calibration is the process of specifying parameters that relate the intensity of the received signal to the radar reflectivity. No calibration is required for velocity. Calibration can conveniently be divided into two parts: determination of the radar constant and determination of the receiver gain. The radar constant includes all the characteristics of the radar that affects the received power at the antenna port. This includes factors such as transmitted power, wavelength, antenna gain, beamwidth and pulse length. The receiver gain relates power at the antenna port to the power measured at the A/D converter.

To measure receiver gain, a program called RCP_cal is run and microwave signals of various levels are injected into the receiver at the antenna port input. The program produces a text output file that may be used to trace out a receiver calibration curve. It also produces a single number that represents the entire calibration curve. RADS uses this single receiver gain number to produce some of its output displays. This is perfectly adequate for real-time displays. For quantitative work, particularly at low signal levels, the recorded signals may be run through programs that apply the complete calibration curve during data analysis.

After a calibration, the standard procedure is to produce a radar parameter table that incorporates the latest calibration results. Other radar parameter tables are derived from this calibration set, but do not alter the calibration parameters. The radar parameters, and thus the calibration parameters are then read into the RADS at runtime.

3. RADS SOFTWARE

3.1 Software Architecture

Since RADS is built with two processing units, the software is comprised of a single program that executes on the DSP CPU and several processes that execute on the SPARC CPU. The program that executes on the DSP is called RCP_DSP, as described in section 2.1. Seven additional processes are simultaneously executed on the SPARC CPU. The seven processes executing on the SPARC are 1) RCP_SPARC, 2) x_display 3) readDma, 4) writeDisk, 5) exabyte, 6) gps_serv, and 7) scanAntn. These are defined in more detail in the subsections below. Six of these processes communicate through interprocess communication protocols, including shared memory and semaphores. The seventh process, scanAntn, executes on the SPARC as well, but communicates with the control GUI, RCP_SPARC, through TCP/IP sockets. ScanAntn controls antenna motion by loading commands into the motion controller card. Additional processes are needed for VAD processing, navigational-ingest, radiometer ingest and display and/or playback functionality. These processes are described in greater detail in section 6. All of the source code is in C or C++ and makes calls to Xview and Xlib. Lex and yacc are used in RCP_SPARC to parse queue files. A detailed functional discussion of the SPARC processes follows.

3.2 Control Window Graphical User Interface (GUI)

RCP_SPARC is the process that enables the operator to control the radar through a control Graphical User Interface (GUI) window on the color monitor. It provides the user with buttons and menus to position, start and stop the radar, as well as read, write, view and edit scan tables and queue files. The operator may select functions through buttons on the control GUI, and thus control scanning, processing and recording. All radar parameters may be viewed, and subsets of parameters may be selected to be shown simultaneously on the screen. For example, if the operator has selected one of the pulse pair modes, only those radar parameters that are pertinent to that mode and are operator-enabled will be displayed in a pop-up mode window. In addition, because of the large number of radar parameters (123), they are grouped by the eleven primary functions or classes. These classes include timing, calibration, scanning, transmit, receiver, data processing, housekeeping, plate, gps, and nav. The radar parameters may be displayed by these primary classes as well. The control GUI also allows the operator to stop and start radar scans and to position the radar. See the Window 1 of Appendix F for a view of the main scan and display control window or base window for the control GUI.

RCP_SPARC allows the operator to turn the record mode on or off and to view and select as many as 12 fields to be displayed. This process also allows the operator to flush the disk, change the tape number, eject tapes, and start and stop a queue. It uses socket interprocess communication (IPC) protocols to communicate with scanAntn and shared memory IPC protocols to send commands and data between the other SPARC processes. When building scan tables, it allows the user to select from several data modes (see section 4.1) and from various sweep modes. Several other scan parameters such as sweep time, min/max elevation and azimuth, angle increment, number of range gates and scan name may be modified by the operator depending upon the selected scan mode. All parameters are viewable by scan type, each having its own window, but the majority of the 123 radar parameters are not operator-enabled. This process reads and writes radar parameters version 1.2 from shared memory, as outlined in the *RADS Radar Parameters Version 1.2* document (see Appendix H).

RADS supports from four to 768 range gates limited by the pulse repetition period and the data mode. In all data modes except time-series modes, up to 1024 triggers are supported. A typical number of triggers during operations is between 128 and 512. A typical number of range gates is between 128 and 500. Beam rates between four and 16 beams per second are typical. These numbers are based on a pulse repetition period of 500 μs.

Other functions that are enabled via the control GUI are briefly discussed. See Appendix F for examples of many of the pop-up windows and menus referred to below.

3.3 Data Archival

The control GUI allows the user to turn the recording capabilities on or off. When the recording option has been selected, (this is the default), the radar data are written first to disk and then to 8 mm tapes, and the radiometer data, if any, to disk only. Limited tape control is available through the control GUI i.e., the user can change the current tape number, eject tapes and flush the data disk of archived files.

3.4 Instrument Control

The user is able to run and pause the radar in continuous mode or in a queuing mode. The RADS includes extensive instrument control, including positioning, starting and stopping radar scans and exiting the RADS (see section 5 RADS Users' Guide). The RADS also triggers the radar by writing to the RTG.

3.5 Radar Parameters

The user is able to edit, store, load, list and check different individual scan control tables. From the available modes (see section 4.1) the operator is able to select both the operation mode (covariance, time-series, spectra, etc.) and the scan mode (Vertical, Sector, PPI, RHI, fixed.) The names of the existing scan tables can be viewed. These tables allow users to indicate which mode to run and to select which parameters to change (e.g., housekeeping) and display. The startup script passes the initial scan control filename to the RCP_SPARC process. For an example, see Appendix F, Figures 5 and 6.

3.6 Scan Table Queuing

Designed to be run autonomously for long periods of time, a queuing feature has been included. The operator can load, edit, store, list, start and stop individual queue files. Queue files are made up of scan table names and scan types that trigger automatic GIF files. Complicated sequences of a variety of scans can be preprogrammed to run in repeating loops using the queuing feature. The control GUI also allows the operator to view the names of all the existing queues. It also permits the user to select 12 of the up to 32 fields available for individual modes to be sent to x_display for real-time display purposes. See Appendix F, Figure 4.

This system, as described, has demonstrated the capability to process 500 range gates in a pulse-pair mode with 1 ms between the pairs and using 64 trigger pairs. Because all processing is done in the DSP card, which is fully programmable, other algorithms can be readily programmed as well. To date, the following modes have been programmed: time-series, spectrum, pulse-pair, delta-k and differential phase (see section 4.1). It is currently possible to switch between various modes. Modes should be judiciously chosen to keep calculations to a minimum, while giving the necessary information for the experiment at hand.

3.7 VAD Profile Parameter Selection

When the user selects the "VAD Params…" button from the base window, a mode-dependent pop-up window will appear, see Appendix F, Figure 8. This window interfaces with the rt-vad process, which is described in section 6.3 below. It allows the user to select correlation threshold, DBZ threshold, range limits, fields to be displayed, elevation and elevation tolerance and number of sweeps and volumes to average for VAD profile displays. It also allows the user to return to the default setting or save the current settings. For detailed information about this window, please refer to section 5, the RADS Users' Guide below.

3.8 Selection of Display Fields

When the user selects the "Display Fields" button, the fields for the chosen data mode appear in a pop-up window; see Appendix F, Figure 7. The user can select any 12 of those fields to be sent to the x_display process for viewing.

3.9 Status Window

A small status window is also controlled by the SPARC_RCP and includes information about what scan and queue is being run, as well as tape number and antenna status; see Appendix F, Figure 2.

3.10 Antenna Scanning

Scanning information flows from RADS to a process called scanAntn via a socket connection. ScanAntn directs the Galil motion controller card, which sends commands to the antenna. Actual antenna position is read directly by RADS from the synchro lines using a synchro-digital card.

3.11 Data Ingest

There are several data streams that can be ingested by RADS. These always include radar data and GPS data, but may optionally include radiometer data and navigational data. Two different processes ingest radar and radiometer data streams and communicate with the display processes and archival processes via shared memory objects. Since these two data streams are separate, they can be ingested at non-synchronous times.

Radar data come into the SPARC from the DSP's global memory via a DMA process. This process, called readDma, waits for an interrupt from the DSP and then reads the data from memory. It ingests two different kinds of radar datam; raw data and display data. It reformats the raw data by adding an Extended Universal Format (EF) header and sends that data to the writeDisk process via shared memory. It also reads the data fields to be displayed, depending upon the mode selected. It demultiplexes the data, adds a Universal Format (UF)-like header, and sends the data to x_display via a different type of shared memory packet. It only sends 12 of the available fields (which the user has pre-selected) to x_display because of memory constraints. EF file formats are discussed in the following section. More details are found in Appendix H. For a more detailed discussion of UF file definitions refer to Barnes, 1980 or Appendix C.

Radiometer data are discussed in section 6.2 and navigational data are discussed in section 6.1 of this document.

Process GPS_serv ingests GPS data and serves these data to RCP_SPARC through shared memory. The DSP program accesses the GPS card via the VME bus to obtain the time, which is then affixed to the data packet, as described in section 2.2.

3.12 Disk Files

Once the writeDisk process receives the raw data via a shared memory packet, it creates filenames with formats of *mm..dd.yy.hh.mm.ss..vol..vol#.swp#* and writes the raw data to EF disk files (see Appendix C). EF disk files record the data with a UF style header format making use of the Local Use Header block by including the radar parameters. See Appendix H for radar parameter definitions. EF data are stored in 32 bit floating point rather than 16 bit integers, for all modes except time-series mode. The SWTM (sweep time) and MODE parameters determine the size and length of the raw radar file. When the scan has completed and writeDisk has received an EOF command from the SPARC_RCP, it sends an interprocess message via shared memory to the Exabyte process with the filename, which is ready to be written to tape.

Because the 8 mm tape drives run at 1 Mbyte/sec and peak data rates are 4.1 Mbyte/sec, RADS needs to store the raw data files until they can be written to tape. Disk space limits the time and amount of data that can be stored on disk, but tape speed makes storing multiple gigabytes impractical. Data will be stored on disk until written to tape, or disk space is depleted. Once the data are written to tape by the Exabyte process, it sends a shared memory message to the deleteFile process. This message is the name of the file that deleteFile needs to delete from the disk. The deleteFile process is optional but practical and highly recommended for unmanned operations. Disk files live in the subdirectory, which has been specified in the command line that starts the writeDisk process in the RCP script.

3.13 Data Displays

The RADS process x_display receives the data from readDma via shared memory and displays it according to sweep mode type. For display purposes, variables and fields must be defined. A variable is a function only of time, such as integrated liquid water, and is plotted as a line versus time, referred to as an A-scope display. A field is a function of range and time, such as radar reflectivity. Here the magnitude of the field is indicated by a color on a plot of range vs. time, or BSCAN display.

Vertical and fixed-beam (target) modes are displayed as BSCANs or marching Range-Time displays which are updated in near real-time. These displays are 600 pixels wide by 338 pixels high. In normal operations, this would correspond to 600 beams of 338 range gates. x-display offers an averaging option for BSCANs that allows longer time periods to be displayed on one screen. The maximum range is also user selectable. Temporal resolution is dependent upon beam averaging and pulse repetition period. The user may change the minimum and maximum values of the color scale, thus highlighting important features.

When running in time-series mode, the BSCAN display becomes a display of stacked FFTs or spectra. The x axis represents frequency, the y range gates, and the colors indicate the power spectral density. Frequency of these updates depends upon the speed of the CPU, although typical numbers are every four seconds. The user may change the color scale, as if it were a marching time-height display.

AScope displays or time-range displays are available simultaneously in a separate window, independent of sweep type, for all sweep types. Again, the user can change the maximum y value of the AScope display. AScope displays are linear and not normalized. While running in time-series mode, x_display calculates one FFT for a given range gate, eight times a second. The resulting power spectral density function is displayed in the AScope window. The user may select which range and field is to be displayed and may change the maximum y value displayed.

PPIs and sectors are displayed in a PPI window and RHIs are displayed in a separate RHI window. Pan and zoom features allow the user to move the center of the display viewport and enlarge the area of interest. Field selection and dynamic color scale selections are available in RHI and PPI modes as well. PPIs or VADS are also displayed as height profiles. VAD displays allow the user to choose only four fields to be displayed

x_display provides a separate graphics control GUI that allows the user to select display type (i.e., BSCAN, PPI, RHI, A_SCOPE) and which of the 12 fields are to be displayed for a selected display type; see Appendix F, Figure 9. Any of the 12 selected fields can be simultaneously displayed on both the A-scope display and one of the other three display types. The software offers both an erase and a GIF capture capability. The user may capture multiple GIF images as often as once a second.

Written in Xview, Xlib and C++, the x_display process enables the operator to select color scales that maximize their ability to understand the atmospheric or oceanic phenomena being observed. x_display also allows thresholding of reflectivity fields against correlation fields and beam averaging for BSCAN displays. Several of the user options, including initial color scales, fields to be displayed and correlation threshold values are configurable by the user in advance and read from the configuration file into x_display at startup.

3.14 Automatic Capture of Desired Images

While running a queue, the user can specify which images to routinely capture as GIF images. The code will then take a snapshot of the specified completed scan. As many as three fields can be captured. The number of fields and which fields are to be captured are specified in the configuration file; see section 5.2 of the RADS Users' Guide. The user can also select a GIF button in the x_display window to take a snapshot of any of the four windows, at any time that they are open and completely on the screen. This does a screen capture, thus only the field shown on the screen will be sent to the GIF file. The windows do not need to be exposed, but can not be iconized. GIF files are stored as *yymmddhhmmss.scantype.radarname.gif* in a directory specified in the x_display command in the RCP script. If the user selects the image with the GIF button, an "S" is pre-pended to the filename. These files are each typically about 110 Kbytes, depending upon the data.

3.15 Convert Raw Time-series into Power Spectral Density

Although in spectral modes, the DSP performs the FFT, in mode 103 a process called convert103Packet calculates one FFT for each range gate of data. Because of CPU limitations, it only calculates these every eight beams and then sends these to the x_display process for displays. As noted in the x_display description above, that process calculates an FFT for every beam, but only for one range gate, and displays those as AScopes.

3.16 Thresholding

x_display thresholds all fields on the correlation field. Thresholds are defined in the configuration file but can also be defined interactively through the x_display GUI. V's (denoting vertical polarization fields) are thresheld on v correlation fields, and H's (horizontally polarized fields) on h correlation fields. If no correlation fields have been chosen to be displayed, thresholding cannot take place.

3.17 Playback Capabilities

The RADS architecture allows for a playback mode to be easily configured. By replacing readDma with readDisk and setting the preprocessor real-time switch to off, playback reads from local data files that were previously archived to disk. Currently, it displays only the radar data for SLAR or time-series modes, much as it does in real-time. (Another process to calculate the fields that the DSP calculates in real-time would be necessary to process the rest of the modes. This is straightforward, but has not been implemented yet.) Since playback RADS does not need to be a VME system and does not need a DSP, it can be run on the RADS system hardware or on any non-VME based ultraSPARC. Currently, playback mode runs at about ten times real-time and is limited by the disk access speed on the ultraSPARC system.

4. SIGNAL PROCESSING ALGORITHMS

4.1 Available Data Modes

The A/D board has four 12-bit channels that digitize simultaneously. RADS has been used with several different receiver channels connected to the four channels, which of course requires different processing modes. A "linear" channel requires in-phase (I) and quadrature (Q) receiver channels, and so consumes two A/D channels. A "logarithmic" channel consumes one A/D channel. Logarithmic channels are used to provide increased dynamic range over what can be achieved with a linear channel, but they are not capable of giving velocity or phase information.

One common configuration for a dual-polarization radar is to have one linear channel connected to the co-polarized receive channel, one log channel connected to the co-polarized receive channel, and one log channel connected to the cross-polarized receive channel. However this configuration is not capable of giving phase information between the co-polarized and cross-polarized channels, i.e., differential phase.

The configuration used most recently is to have two linear channels, one for the co-polarized channel and one for the cross-polarized channel. This allows for phase comparisons between the two channels.

The following types of data modes have been implemented: delta-k, time-series, power spectrum, single-polarization pulse pair with logarithmic receiver channels, and dual-polarization pulse pair. There are four variations on this last mode, depending on the algorithm used to calculate HV correlation, and the fields desired. Table 2 outlines the data modes by number, name, display fields, recorded fields, receiver channels, and transmit and receive polarization.

Table 2. RADS/GRADS mode assignment chart*

Mode #	**Name**	**Receiver channels**	**Transmit polarization/ linear receiver polarization**	**# of recorded fields**	**# of displayed fields**	**Comment**
102	Time-series	1 linear, 2 log	Any			log channels averaged over beam; used only for diagnostics (not RCP)
103	Time-series	2 linear	Any	4	6	
117	Delta-k	1 linear, 2 log	Any			first mode implemented
118	Spectrum	1 linear, 2 log	Any			used only one DSP chip
140	Pulse pair	1 linear, 2 log	Any	5	13	DC always removed
150	Switched differential phase, method 1	1 linear, 2 log	HHVV (xmit) hhvv (recv)	14	32	Must use equally spaced pulses if $\rho_{HV}(0)$ is to be correct.
151	Switched high-speed differential phase only, method 1	1 linear, 2 log	HHVV (xmit) hhvv (recv)	8	16	no velocities, widths or linear powers
158	Switched differential phase, method 2	1 linear, 2 log	HHVV (xmit) hhvv (recv)	18	31	may use unequally spaced pulses
159	Switched differential phase, methods 1 & 2	1 linear, 2 log	HHVV (xmit) hhvv (recv)	22	34	
160	SLAR, left antenna	1 log	Any	2	2	SLAR modes use a different version of the RCP; all SLAR modes available for playback
161	SLAR, right	1 log	Any	2	2	
162	SLAR, both	1 log	Any	4	4	
163	SLAR, both + antn select fields	1 log	Any	6	6	
164	SLAR, both + antn select fields	1 log	Any	6	6	2048 range gates only. Never implemented.
170	Split-hv differential phase	2 linear	S (Split) h & v (simultaneous)	16	34	DC either removed or not depending on field.
171	Split-hv for ocean	2 linear	S (Split) h & v (simultaneous)	16	32	Never implemented

Mode #	Name	Receiver channels	Transmit polarization/ linear receiver polarization	# of recorded fields	# of displayed fields	Comment
180	GRIDS pulse pair	1 linear	C (Circular) ccxx (co- and cross-)	14	19	Under development for GRADS

*Shaded entries are not currently used, under development or not functional. Recorded fields are four bytes each and always floating-point except for time-series modes. Where two log channels are used, co- and cross-channel intensity information is available simultaneously from them. Changing receiver channels or polarization requires hardware reconfiguration. GRIDS modes use a different A/D board and DSP board than other modes.

4.2 General Notes on Covariance Algorithms

Most of the signal processing used in RADS is based on "pulse-pair" or covariance algorithms. The radar transmits pairs of pulses in the manner shown in Figure 1.

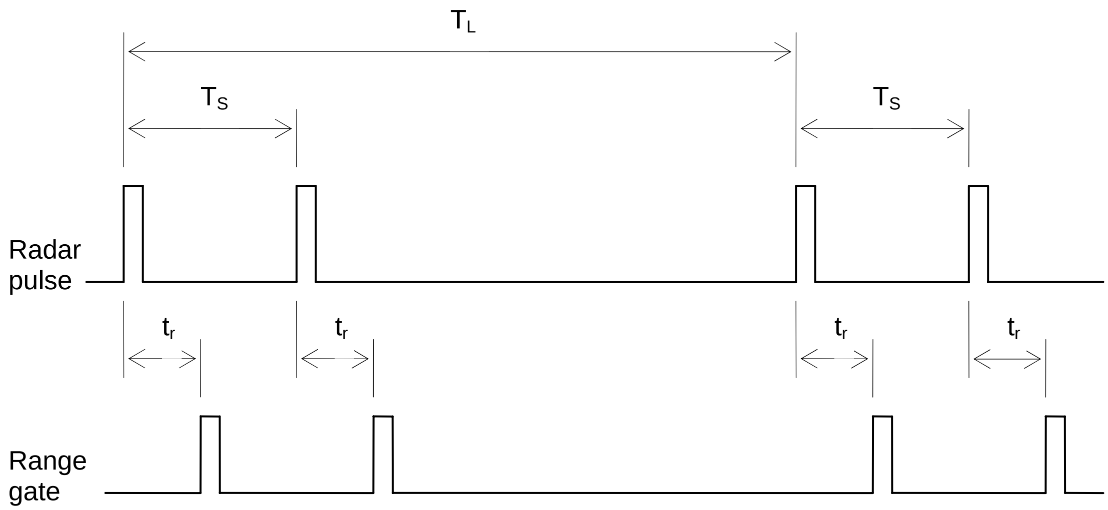

Figure 1. Transmitter pulses and a single range gate are shown for the double-pulse modulation used in RADS.

A single range gate is illustrated at a delay t_r from the transmitted pulses. Samples at the short pulse spacing, T_S, are used to give an estimate of the Doppler velocity. Results at the long pulse spacing, T_L, are summed together and used to reduce the uncertainty of the estimate. The advantage of this type of "double-pulse" modulation over equally spaced pulses is that by reducing T_S, the unambiguous velocity limit of the radar can be increased without increasing the transmitted power, which is limited by the characteristics of the transmitter tube. This is particularly useful for short-wavelength radars. The disadvantage is that the unambiguous range is limited by how close the pulses in the pair are. However, T_S can always be changed to half of T_L if desired.

In the following discussion, whenever a covariance product is calculated, it is done from pulses in the pair, while the sums of these products are performed over the pairs. The time T_L is not significant in the calculations, except indirectly in that increasing it reduces the number of samples taken in a given interval and reduces the beneficial effects of averaging.

The complex covariance function is used for many purposes in statistics, but here it is used only to calculate the phase change between two points in complex sine waves, and the power present in a complex sine wave. Complex quantities are in **bold** in the following equations and in Table 3.

Consider two complex sequences, $\mathbf{X}_i$ and $\mathbf{Y}_i$. Then the general definition of complex covariance is given by

$$\mathbf{R}_{XY}(n) = \left\langle \left(\mathbf{X}_i^* - \left\langle \mathbf{X}_i^* \right\rangle \right) \cdot \left(\mathbf{Y}_{i+n} - \left\langle \mathbf{Y}_{i+n} \right\rangle \right) \right\rangle, \qquad \textbf{(Equation 4-1)}$$

where $\mathbf{X}_i$ and $\mathbf{Y}_i$ represent complex sequences and n represents the number of lags to use in calculating the covariance. The "< >" notation represents the average and "*" denotes the complex conjugate.

This equation may be rewritten to give

$$\mathbf{R}_{XY}(n) = \langle \mathbf{X}_i^* \mathbf{Y}_{i+n} \rangle - \langle \mathbf{X}_i^* \rangle \langle \mathbf{Y}_{i+n} \rangle . \qquad \textbf{(Equation 4-2)}$$

In RADS, $\mathbf{X}_i$ and $\mathbf{Y}_i$ are usually the same channel, which is the digitized representation of a voltage. This signal will be denoted $\mathbf{E}_i$. Since we are now using an auto-covariance, we can simplify our nomenclature to

$$\mathbf{R}(n) = \langle \mathbf{E}_i^* \mathbf{E}_{i+n} \rangle - \langle \mathbf{E}_i^* \rangle \langle \mathbf{E}_{i+n} \rangle . \qquad \textbf{(Equation 4-3)}$$

For the zero-lag covariance, where i ranges over every pulse

$$\mathbf{R}(0) = \langle \mathbf{E}_i^* \mathbf{E}_i \rangle - \langle \mathbf{E}_i^* \rangle \langle \mathbf{E}_i \rangle = \langle |\mathbf{E}_i|^2 \rangle - \langle \mathbf{E}_i \rangle^* \langle \mathbf{E}_i \rangle ,$$

or

$$R(0) = \langle |\mathbf{E}_i|^2 \rangle - |\langle \mathbf{E}_i \rangle|^2 . \qquad \textbf{(Equation 4-4)}$$

Note that R(0) is real-valued. This represents the average of the "power" (voltage squared) minus the average DC value squared. In other words, R(0) is the DC-corrected "power."

R(1) may be written as follows, where i increments by two (for pairs):

$$\mathbf{R}(1) = \langle \mathbf{E}_i^* \mathbf{E}_{i+1} \rangle - \langle \mathbf{E}_i^* \rangle \langle \mathbf{E}_{i+1} \rangle , \qquad \textbf{(Equation 4-5)}$$

where $\mathbf{E}_{i+1}$ is the signal from the second pulse in the pair.

Now calculate $\mathbf{R}(1)$ for a complex sine wave as defined by

$$\mathbf{E}_i = Ae^{j\omega i T_S} ,$$

so

$$\mathbf{E}_{i+1} = Ae^{j\omega (i+1) T_S} .$$

Then

$$\mathbf{R}(1) = \langle Ae^{-j\omega i T_S} Ae^{j\omega(i+1)T_S} \rangle - \langle Ae^{-j\omega i T_S} \rangle \langle Ae^{j\omega(i+1)T_S} \rangle = \langle A^2 e^{j\omega T_S} \rangle - \langle Ae^{-j\omega i T_S} \rangle \langle Ae^{j\omega i T_S} e^{j\omega T_S} \rangle$$

$$\mathbf{R}(1) = A^2 e^{j\omega T_S} - A^2 e^{j\omega T_S} \langle e^{-j\omega i T_S} \rangle \langle e^{j\omega i T_S} \rangle = A^2 e^{j\omega T_S} \left(1 - \langle e^{-j\omega i T_S} \rangle \langle e^{j\omega i T_S} \rangle \right) \qquad \textbf{(Equation 4-6)}$$

$$\mathbf{R}(1) = A^2 e^{j\omega T_S} \left(1 - \langle \cos \omega i T_S - j \sin \omega i T_S \rangle \langle \cos \omega i T_S + j \sin \omega i T \rangle \right)$$

$$\mathbf{R}(1) = A^2 e^{j\omega T_S} \left(1 - \langle \cos \omega i T_S \rangle^2 - \langle \sin \omega i T_S \rangle^2 \right). \qquad \textbf{(Equation 4-7)}$$

Therefore,

$$\arg(\mathbf{R}(1)) = \omega T_S .$$

Since the angular frequency, ω, is related to the Doppler shift, this demonstrates how the Doppler velocity may be obtained, at least for the case of a sine wave. More elaborate derivations for signals plus noise may be found in the literature, but this illustrates the basic idea.

In RADS, the covariance sums are done on the DSP board and the results passed to the SPARC for further calculation. The details vary depending on the processing mode, but in the latest implementation we take the equation

$$\mathbf{R}(n) = \langle \mathbf{E}_i^* \mathbf{E}_{i+n} \rangle - \langle \mathbf{E}_i^* \rangle \langle \mathbf{E}_{i+n} \rangle$$

and partition it as follows. Let

$\mathbf{B}(n) = \langle \mathbf{E}_i^* \mathbf{E}_{i+n} \rangle$ and

$\mathbf{A}(n) = \langle \mathbf{E}_{i+n} \rangle$.

Then

$\mathbf{R}(n) = \mathbf{B}(n) - \mathbf{A}(0)^* \cdot \mathbf{A}(n)$,

so

$R(0) = B(0) - \mathbf{A}(0)^* \cdot \mathbf{A}(0) = B(0) - |\mathbf{A}(0)|^2$ and

$\mathbf{R}(1) = \mathbf{B}(1) - \mathbf{A}(0)^* \cdot \mathbf{A}(1)$.

Currently, n can be either 0 or 1, so the DSP calculates quantities **A0**, **A1**, B0 and **B1**, where **A0** is the complex sum from the first pulse in the pair, and **A1** is the complex sum from the second pulse in the pair. B0 is defined to be the sum over both the first and second pulse in the pair, and **B1** is just **B(1)**, which leads to the following equations:

$$R(0) = B0 - \frac{1}{2}\left(|\mathbf{A0}|^2 + |\mathbf{A1}|^2\right) \quad \textbf{(Equation 4-8)}$$

$$\mathbf{R}(1) = \mathbf{B1} - \mathbf{A0}^* \cdot \mathbf{A1} \ . \quad \textbf{(Equation 4-9)}$$

Calculating the A's and B's separately and passing them on to the host SPARC allows it to calculate both DC-corrected and non-DC-corrected fields. Non-DC-corrected fields are useful for producing ground clutter maps, since normally the ground targets would be suppressed.

A very similar case to the auto-covariance is a cross-covariance of two sine waves at zero lag. From above we have

$\mathbf{R}_{XY}(0) = \langle \mathbf{X}_i^* \mathbf{Y}_i \rangle - \langle \mathbf{X}_i^* \rangle \langle \mathbf{Y}_i \rangle$.

Now assume that $\mathbf{X}_i$ and $\mathbf{Y}_i$ are two sine waves of the same frequency, but different phase as given by

$\mathbf{X}_i = Ae^{j\omega iT}$

$\mathbf{Y}_i = Be^{j(\omega iT + \varphi)}$.

Then

$\mathbf{R}_{XY}(0) = \langle Ae^{-j\omega iT} Be^{j(\omega iT+\varphi)} \rangle - \langle Ae^{-j\omega iT} \rangle \langle Be^{j(\omega iT+\varphi)} \rangle = ABe^{j\varphi} - ABe^{j\varphi} \langle e^{-j\omega iT} \rangle \langle e^{j\omega iT} \rangle$

$\mathbf{R}_{XY}(0) = ABe^{j\varphi}\left(1 - \langle e^{-j\omega iT} \rangle \langle e^{j\omega iT} \rangle\right)$.

But the exponential product is the same as in Equation 4-6, so by similarity

$\mathbf{R}_{XY}(0) = ABe^{j\varphi}\left(1 - \langle \cos \omega iT \rangle^2 - \langle \sin \omega iT \rangle^2\right)$

and

$\arg(\mathbf{R}_{XY}(0)) = \varphi$.

The angle of $\mathbf{R}_{XY}(0)$ gives the phase difference between the sine waves. This property is used in differential polarization calculations to calculate the phase difference between two different polarizations that are received simultaneously.

In dual-polarization modes, the quantities **A0**, **A1**, B0 and **B1** are calculated for each polarization, as well as a covariance between the two polarizations. Table 3 shows the field names of the raw covariances calculated by the DSP for three different covariance modes, and their relation to the variable names presented here.

H and v in many of the names refer to horizontal and vertical polarizations, respectively. S refers to split polarization on transmission, where horizontal and vertical polarizations are transmitted simultaneously.

Mode 150 uses polarization switching on both transmit and receive channels, and is described in Appendix J. Because of the transmit switching, the algorithms used are much different than the algorithms that have been described here.

Table 3. Covariance products produced by various processing modes

	Mode 140 (single-polarization)	**Modes 150, 151, 157, 158** (dual-polarization)	**Mode 170** (dual-polarization)
Re(**R**(1))	A		
-Im(**R**(1))	B		
R(0)	R		
average of v log channel	Lv		
average of h log channel	Lh		
A0 for h			**ASh0**
A0 for v			**ASv0**
A1 for h			**ASh1**
A1 for v			**ASv1**
B(0) for h			BShSh0
B(0) for v			BSvSv0
B(1) for h			**BShSh1**
B(1) for v			**BSvSv1**
$B_{hv}(0)$			**BShSv0**
R(1) for h		**RsHH1**	
R(1) for v		**RsVV1**	
R(0) for h		RlHH0	
R(0) for v		RlVV0	
$\mathbf{R}_{hv}(1)$		**RsHV1**	
$\mathbf{R}_{vh}(1)$		**RsVH1**	
$\mathbf{R}_{hv}(2)$		**RlHV2**	
$\mathbf{R}_{vh}(2)$		**RlVH2**	
R(4)		**RlHH4**	
R(4)		**RlVV4**	
average of h log channel, transmit H		GIHh	
average of v log channel, transmit H		GIHv	
average of h log channel, transmit V		GIVh	
average of v log channel, transmit V		GIVv	

4.3 Calculation of Display Products from Covariance Data

All covariance modes divide the fields into two parts: data products and display products. The data products are the covariance results discussed above and are recorded on disk or tape for later use. The DSP also calculates other parameters, including the common radar meteorological parameters, and calls them display products because they are displayed by the host SPARC for the user. They are not recorded for reasons of data economy, but may be recalculated for later display from the data products.

All covariance modes calculate the display products shown in Table 4. Note that dual-polarization modes will have two sets of these parameters.

Table 4. Common display products

Name	Units	Range	Description
Velocity	m/s	[-nyquist, +nyquist]	Doppler radial velocity, + is away from radar
Width	M^2/s^2	[0, ∞]	Estimator of spectral width; because it is an estimator, it can occasionally be negative.
Correlation	none	[0.0, 1.0]	Correlation coefficient
Intensity	dBm	[-50, 16.2]	Power at receiver output; lower limit approximate
Power	dBm	[-110, -37]	Power at antenna; approximate range
Reflectivity	dBZ		Radar reflectivity

The equations used to calculate the common display products are shown in Table 5.

Table 5. Equations for common display products

Parameter/ Units	Equation	Comments
Velocity in m/s	$V = \frac{-c}{4\pi f_T T_S} arg(\mathbf{R}(1))$	c is speed of light in m/s f_T is the transmit frequency in Hz T_S is the time between pulses in a pair in seconds
Width in M^2/s^2	$W = \frac{c^2}{8\pi^2 f_T^2 T_S^2}\left(1 - \frac{\lvert\mathbf{R}(1)\rvert}{R(0) - R'(0)}\right)$	$R'(0)$ is R(0) calculated with receiver noise
Correlation Unitless	$C = \frac{\lvert\mathbf{R}(1)\rvert}{R(0)}$	
Intensity in dBm	$I = 10 \bullet log\left(\frac{R(0)}{z_0}\right) + 30$	z_0 is the system impedance of 50 ohms
Power in dBm	$P = I - r_g$	r_g is the receiver gain in dB
Reflectivity in dBZ	$Z = P + k_{RC} + 20 \bullet log(r) - 60$	k_{RC} is the radar constant in dB r is the range in meters

Dual-polarization modes allow for the calculation of many more parameters, as shown in Table 6 and Table 7.

Table 6. Dual-polarization parameters

Name	**Units**	**Range**	**Description**
differential phase, φ_{dp}	degrees	[-180, 180]	Phase shift between the two polarizations
magnitude of correlation between two polarizations, $\rho_{hv}(0)$	none	[0.0, 1.0]	Correlation between the two polarizations
differential reflectivity (ZDR)	dB		A measure of how much the reflectivity differs between co-polarized and cross-polarized signals. Ratio of reflectivity between two polarizations when transmitted and received polarizations are the same.
depolarization ratio (LDR)	dB	[-40, 0]	A measure of how much a scatterer depolarizes a reflected signal. Ratio of cross-polarized signal to co-polarized signal when a co-polarized signal is transmitted.
Corrected differential reflectivity	dB		Differential reflectivity corrected for attenuation using differential phase
Corrected reflectivity	dBZ		Reflectivity corrected for attenuation using differential phase
rainfall rate from reflectivity	mm/hour		Classic radar meteorology rainfall rate from Z-R relationships
rainfall rate corrected for attenuation	mm/hour		
median drop size	mm		In rainfall, an estimator of the median drop size
differential Doppler velocity	m/s		Expresses difference in velocities measured in the two polarizations

Table 7. Equations for dual-polarization parameters

Parameter	Equation	Comments
differential phase	$\varphi_{dp} = \frac{180}{\pi} arg(\mathbf{R}_{ab}(0))$	$\mathbf{R}_{ab}$ is the cross-covariance between the two polarizations of the radar, either horizontal and vertical, or left and right circular
magnitude of correlation between two polarizations	$\rho_{ab}(0) = \frac{\lvert\mathbf{R}_{ab}(0)\rvert}{\sqrt{R_a(0)R_b(0)}}$	$R_a(0)$ is auto-covariance from first polarization $R_b(0)$ is auto-covariance from second polarization
Differential reflectivity (ZDR) or depolarization ratio (LDR)	$Z_{DR} = Z_a - Z_b$	Z_a is reflectivity from the first polarization Z_b is reflectivity from the second polarization Whether this is ZDR or LDR depends on what the transmit polarization is.
corrected differential reflectivity	$Z_{DRC} = Z_{DR} + 0.044 \bullet \varphi_{dp}$	
corrected reflectivity	$Z_C = Z + p\varphi_{dp}$	p is 0.244 for horizontal polarization, 0.200 for vertical polarization
rainfall rate from reflectivity	$R = 10^{0.0667 \bullet Z - 1.720}$	
rainfall rate corrected for attenuation	$R_C = 10^{0.0667 \bullet Z_C - 1.720}$	
median drop size	$D_m = Z_{DRC}$	
differential Doppler velocity	$V_d = \left(V_a - V_b\right) / \sin\theta$	V_a is velocity from the first polarization V_b is velocity from the second polarization θ is the elevation angle of the radar

Many of the parameters shown in Table 6 and Table 7 can be calculated by replacing the R(0) or **R**(1) with B(0) or **B**(1) to produce a non-DC-corrected version of the parameter. In mode 170, this is done where it is meaningful.

Modes 140 and 15x both use linear and logarithmic receiver channels. Log channels cannot be used for phase measurements and are used for power estimates only. Fields from the log channels are designated by a "G" in the name. Table 8 shows the field names used by modes 140, 150, 151, 158, 159 and 170.

The general format of the field names, although not completely consistent, is that the first letter indicates the type of field, for example, V for velocity. However, in modes with log channels, the first letter may be "N" or "G" to indicate whether the field was derived from a liNear or loG channel, where necessary. The next letters indicate the transmitted and received polarizations, with transmitted polarizations in upper case and received polarizations in lower case. For example Hv means that horizontal polarization was transmitted and vertical polarization was received. S stands for split and means that H and V were transmitted simultaneously. In some field names, where there is no "h" or "v", it is understood that the received polarization is the same as the transmitted polarization. Finally, the last letter, particularly if capitalized, is often D, U or C, meaning DC-corrected or Uncorrected, or Corrected by φ_{dp}.

Table 8. Fields from all covariance modes

Description	Mode 140	Mode 150	Mode 151	Mode 158	Mode 159	Mode 170	Units
Velocity	VPab	VPHh, VPVv		VPHh, VPVv	VPHh, VPVv	VShD, VSvD VShU, VSvU	m/s
Width	WPab	WPHh, WPVv		WPHh, WPVv	WPHh, WPVv	WShD, WSvD WShU, WSvU	m^2/s^2
Correlation	CPab	CPHh, CPVv		CPHh, CPVv	CPHh, CPVv	CShD, CSvD CShU, CSvU	none
Intensity	NIab	NIHh, NIVv		NIHh, NIVv	NIHh, NIVv	IShD, ISvD (IShU, ISvU)*	watts
Power	NPab	NPHh, NPVv		NPHh, NPVv	NPHh, NPVv	PShD, PSvD PShU, PSvU	dBm
Reflectivity	NTab	NTHh, NTVv		NTHh, NTVv	NTHh, NTVv	ZShD, ZSvD ZShU, ZSvU	dBZ
Intensity from log	GIah, GIav	GIHh, GIHv GIVh, GIVv	GIHh, GIHv GIVh, GIVv	GIHh, GIHv GIVh, GIVv	GIHh, GIHv GIVh, GIVv		volts
Power from log	GPah, GPav	GPHh, GPHv GPVh, GPVv	GPHh, GPHv GPVh, GPVv	GPHh, GPHv GPVh, GPVv	GPHh, GPHv GPVh, GPVv		dBm at antenna
Reflectivity from log	GTah, GTav	GTHh, GTHv GTVh, GTVv	GTHh, GTHv GTVh, GTVv	GTHh, GTHv GTVh, GTVv	GTHh, GTHv GTVh, GTVv		dBZ or dB (ocean)
Differential reflectivity, linear		DNPHV		DNPHV	DNPHV	ZdrD	dB
Differential reflectivity, log		DGPHV	DGPHV	DGPHV	DGPHV		dB
Linear depolarization ratio, horizontal	DGPavah	DGPHvHh	DGPHvHh	DGPHvHh	DGPHvHh		dB
Linear depolarization ratio, vertical		DGPVhVv	DGPVhVv	DGPVhVv	DGPVhVv		dB
Differential Doppler velocity		DVVH		DVVH	DVVH		m/s

Description	Mode 140	Mode 150	Mode 151	Mode 158	Mode 159	Mode 170	Units
Differential phase, raw (= spacing)		PHIdps	PHIdps		PHIdps		deg
Differential phase, raw (≠ spacing)				PHIdpl	PHIdpl	PHdp	deg
Correlation coefficient, HV (= spacing)		**RHOHV0s**					none
Correlation coefficient, HV (≠ spacing)				**RHOHV0l**	**RHOHV0l**		none
Reflectivity corrected by φ_{dp}						ZShC, ZSvC	dBZ
Differential reflectivity corrected by φ_{dp}						ZdrC	dB
Rainfall rate from reflectivity						RnRt	mm/hr
Rainfall rate from reflectivity, corrected for attenuation						RnRC	mm/hr
Magnitude of cross-polar correlation coefficient, $\rho_{hv}(0)$						Rhv0	none
Median drop size, D_m						DSmC	mm
Differential Doppler velocity						DVvh	m/s

*Field is calculated by the DSP but is not displayed on the SPARC.

5. RADS USERS' GUIDE

5.1 Conventions

In this document, standard SUN Solaris notation has been adopted; i.e., when the user is to type to the command line, the prompt appears as **computerName**% or > for single-user mode. The command to be issued follows.

Example:

computerName% command

or

computer1% ls

There are two possible configurations. One machine can control the radar (computer 1), or two computers (computer 1 and computer 2) can share the tasks.

Since RADS runs under OpenWin, several of the windows conform to OpenWin conventions. By clicking the right mouse button while positioned over the top gray bar of a window, the OpenWin menu appears. This window includes the following options: Close, Full Size, Move, Resize, Properties, Back, Refresh and Quit. If the user selects Quit, the process associated with the window WILL be terminated and shared memory left in an undefined state. Therefore, it is best not to use this windowing feature. All pop-up windows have a thumbtack in the top left-hand corner of the gray bar and can be released by positioning the mouse over the thumb-tack and clicking the left mouse button.

Notification windows appear when a user error has been encountered. These are gray windows and require immediate interaction from the user. These do not occur when the program is running without user interaction. They WILL crash the program if they are not attended to immediately.

5.2 The Configuration Files

Before running RADS, the operator should set up the initial configuration file. This is an ASCII file which can be created with vi or Textedit. It is best if the user copies the *default.cfg* file from */export/home/rcp/config* and edits that file, thus reducing a lot of typing and the chance of error, since all keywords are in capital letters in the configuration file and must be spelled correctly. If no configuration file is available, default values are used. The edited configuration file should live in */export/home/rcp/config*. The filename should end with the suffix .cfg. The same configuration file is used by both x_display and rcp_1. The filename is given as an argument in the call to each of those processes and should be the same filename for both processes. The startup script file (typically called RCP) needs to be edited to ensure that the correct configuration file is included. See section 5.3for details. It is advised that one person maintain the configuration file for a project. However, if other configuration files are needed or desired, they should be edited and renamed.

5.2.1 Format

As previously mentioned, the keywords which appear on the left-hand side of the equal sign must be in capital letters and must be spelled correctly. Any part of a line after a "#" is a comment and is thus ignored. All white space is ignored. For an example of a complete configuration file, please refer to Appendix I.

5.2.1.1 Mode

The file should begin with the definition of which MODE is to be the default mode upon startup. For example:

MODE = 170

5.2.1.2 Number of fields

The number of fields (NFIELDS) to be displayed should be defined next. This number must be less than or equal to 12. In some cases (such as mode 103), the maximum is eight. For example:

NFIELDS = 11

5.2.1.3 Number of beams to average

Specify the number of beams to average in the BSCAN displays, using one for no averaging. For example:

BEAMS2AVERAGE = 2

5.2.1.4 Fields

For the number of fields, NFIELD, define the actual fields to be sent to x_display by default. Define the mode 103 fields first and then the covariance fields.

For example:

# first for mode 103
F1031 = Eh.R
F1032 = Eh.I
F1033 = Ev.R
F1034 = Ev.I
F1035 = PwSh
F1036 = PwSv
F1037 = VPab
F1038 = WPab

# then for other fields
F1 = VShD
F2 = WShD
F3 = CShD
F4 = IShD
F5 = PShD
F6 = ZShD
F7 = ZShC
F8 = ZdrD
F9 = ZdrC
F10 = PHdp
F11 = RnRC

Mode 170 fields

What follows next in the *default.cfg* file is a list of all the fields for mode 170. These occur as a comment to aid the user in setting the corresponding color scales in the following section. The list of all mode 170 fields and their corresponding number follows:

```
#  1 = VShD
#  2 = VShU
#  3 = VSvD
#  4 = VSvU
#  5 = WShD
#  6 = WShU
#  7 = WSvD
#  8 = WSvU
#  9 = CShD
# 10 = CShU
# 11 = CSvD
# 12 = CSvU
# 13 = IShD
# 14 = ISvD
# 15 = PShD
# 16 = PShU
# 17 = PSvD
# 18= PSvU
# 19 = ZShD
```

20 = ZShU
21 = ZSvD
22 = ZSvU
23 = ZShC
24 = ZSvC
25 = ZdrD
26 = ZdrC
27 = DVvh
28 = PHdp
29 = RnRt
30 = RnRC
31 = Rhv0
32 = DSmC

5.2.1.5 Scale

There are four scan types that define different kinds of displays available. They are BSCAN, PPI, RHI, and ASCOPE. This section of the configuration file defines the minimum and maximum values assigned to the color bar for these four scan types. For each scan type, each of the total available fields for the chosen MODE has a corresponding minimum and maximum value assigned. Mode 170 has 32 fields available and this is currently the maximum number of fields available. The fields for mode 170 and their corresponding numbers are defined in the commented section immediately above. The values assigned below define the default color scales for the four different scan types by field number. In general, the field number corresponds to the order seen in rcp_1 in the pop-up window that appears when the user selects the button "DISPLAY FIELDS" for that mode type.

ScaleMinBscan1 = -15. # for mode 170 VShD
ScaleMinBscan2 = -15. # VShU
ScaleMinBscan3 = -25. # VSvD

ScaleMinBscan32 = 0. # DSmC

ScaleMaxBscan1 = 15.
ScaleMaxBscan2 = 15.
ScaleMaxBscan3 = 45.

ScaleMaxBscan32 = 4.
ScaleMinPPI1 = -15.
ScaleMinPPI2 = -25.
ScaleMinPPI3 = -25.

ScaleMinPPI32 = 0.

ScaleMaxPPI1 = 15.
ScaleMaxPPI2 = 25.
ScaleMaxPPI3 = 25.

ScaleMaxPPI32 = 4.

ScaleMinRHI1 = -15.

ScaleMinRHI2 = -25.
ScaleMinRHI3 = -25.

ScaleMinRHI32 = 0.

ScaleMaxRHI1 = 15.
ScaleMaxRHI2 = 25.
ScaleMaxRHI3 = 25.

ScaleMaxRHI32 = 4.

ScaleMinAScope1 = -25.
ScaleMinAScope2 = -25.
ScaleMinAScope3 = -25.

ScaleMinAScope32 = 0.

ScaleMaxAScope1 = 25.
ScaleMaxAScope2 = 25.
ScaleMaxAScope3 = 25.

ScaleMaxAScope32 = 4.

5.2.1.6 Threshold

Define the threshold value for the same fields as above for the four scan types.

BScanThresh1 = 0.
BScanThresh2 = 0.
BScanThresh3 = 0.

BScanThresh32 = 0.

RHIThresh1 = 0.
RHIThresh2 = 0.
RHIThresh3 = 0.

RHIThresh32 = 0.

PPIThresh1 = 0.
PPIThresh2 = 0.
PPIThresh3 = 0.

PPIThresh32 = 0.

AScopeThresh1 = 0.
AScopeThresh2 = 0.
AScopeThresh3 = 0.

AScopeThresh32 = 0.

5.2.1.7 Display Field

Select the field to be displayed by default for each of the four scan types, mode 103 and the mode chosen in MODE above.

SelectedFieldBscan = ZShD
SelectedFieldBscan103 = PwSh

SelectedFieldPPI = ZShD
#SelectedFieldPPI103 = ZShD

SelectedFieldRHI = ZShD
#SelectedFieldRHI103 = ZShD

SelectedFieldA_SCOPE = ZShD
SelectedFieldA_SCOPE103 = PwSh

5.2.1.8 Colors

Select the desired color for the BSCAN and the RHI (colors for PPIs are the same as RHIs) displays to wrap to, i.e., minimum, maximum and below the threshold. White and black are the only colors currently recognized.

BscanMinColor = white
BscanMaxColor = white

#BscanThreshColor = black
BscanThreshColor = white

#RhiMinColor = white
RHIMinColor = black

RHIMaxColor = white
#RHIMaxColor = black

RHIThreshColor = black
#RHIThreshColor = white

5.2.1.9 Number of GIFS

For each scan type select the number of GIF images to be automatically generated (up to three) and the fields that will be captured in the GIF images. The fields selected must be available to the display software (i.e., you must have selected them in the configuration file earlier, or in rcp_1). You can also select the one field to be GIFed for mode 103. Note: Up to three fields can be defined to be GIFed even if you set the NUMBEROFGIFS variable to be less than 3.

NUMBEROFBSCANGIFS = 2

SelectedGifFieldBscan1 = ZShD
SelectedGifFieldBscan2 = VShD
SelectedGifFieldBscan3 = RnRt

NUMBEROFASCOPEGIFS = 0

SelectedGifFieldAscope1 = ZShD
SelectedGifFieldAscope2 = VShD
SelectedGifFieldAscope3 = RnRt

NUMBEROFRHIGIFS = 3

SelectedGifFieldRHI1 = ZShD
SelectedGifFieldRHI2 = VShD
SelectedGifFieldRHI3 = RnRt

NUMBEROFPPIGIFS = 2

SelectedGifFieldPPI1 = ZShD
SelectedGifFieldPPI2 = VShD
SelectedGifFieldPPI3 = RnRt

SelectedGifFieldBscan103 = PwSh
SelectedGifFieldAscope103 = PwSh

5.2.1.10 Mode 103 Scale

Select minimum and maximum values for color scale for mode 103 fields, BSCANS and A-Scope types.

ScaleMinBscan1031 = -1500.
ScaleMinBscan1032 = -1500.
ScaleMinBscan1033 = -2500.
ScaleMinBscan1034 = -2500.
ScaleMinBscan1035 = 0.
ScaleMinBscan1036 = 0.
ScaleMinBscan1037 = 0.
ScaleMinBscan1038 = 0.

ScaleMaxBscan1031 = 1500.
ScaleMaxBscan1032 = 1500.
ScaleMaxBscan1033 = 4500.
ScaleMaxBscan1034 = 4500.
ScaleMaxBscan1035 = 1000.
ScaleMaxBscan1036 = 10.
ScaleMaxBscan1037 = 10.
ScaleMaxBscan1038 = 10.

ScaleMinAScope1031 = -2500.
ScaleMinAScope1032 = -2500.
ScaleMinAScope1033 = -2500.
ScaleMinAScope1034 = -2500.
ScaleMinAScope1035 = 0.
ScaleMinAScope1036 = 0.
ScaleMinAScope1037 = 0.
ScaleMinAScope1038 = 0.

ScaleMaxAScope1031 = 2500.
ScaleMaxAScope1032 = 2500.
ScaleMaxAScope1033 = 2500.
ScaleMaxAScope1034 = 2500.

ScaleMaxAScope1035 = 1000.
ScaleMaxAScope1036 = 10.
ScaleMaxAScope1037 = 10.
ScaleMaxAScope1038 = 10.

5.3 The C Shell script file

Warning. Don't edit the startup C shell script file, unless you know what you are doing. Undefined results may occur.

There is a C Shell script file that starts all processes in the RADS, typically called RCP or a variation of that. Several of the processes that compose the RADS have line arguments that may be reconfigured in the script to allow for various implementations. See Appendix ??? for a sample RCP script file.

Begin by copying the RCP file to RCP.default in *directory /export/home/rcp/bin*. Edit the RCP file.

rcp_1

rcp_1 is the primary user interface process. It manages the base window, sends control packets to the radar, and has the following options:

1tst argument Name of the process. DON'T CHANGE

2nd argument Name of the scan you want to have in the local MODE window upon start-up. Do not include the '.rp' extension; however, you can include a sub-directory. By default it always looks in */export/home/rcp/rp*. For example, the second argument could be given as *OLDDIR/filename*, where OLDDIR is a subdirectory of rp and *filename.rp* exists in */export/home/rcp/rp/OLDDIR*.

3rd argument Name of machine running the Galil antenna scanning process

4th argument Title on the top margin of the main control window; must be in double quotes

5th argument Full filename (including full pathname) of the configuration file to be used; i.e., */export/home/rcp/config/xParams.new.cfg*. Needs to agree with argument 5 in x_display; otherwise, results are undefined.

6th argument "antenna" or "no_antenna" to indicate whether using the simulator or the actual antenna. For operations, this needs to be "antenna" and it should be in double quotes. Default is "antenna".

7th argument "&"; indicates that the process is to be run in the background; must be here for the system to operate properly. DON'T CHANGE

x_display

x_display controls and displays the four scan types and has the following options:

1st argument Shelltool and its arguments

2nd argument Name of the process. DON'T CHANGE

3rd argument “X” or “K” depending upon which radar you are using

4th argument Full pathname where the GIF files are to be sent for archival, e.g. */export/home/data/projectName/gif/*

5th argument Full filename (including full pathname) of the configuration file to be used, e.g. */export/home/rcp/config/xParams.new.cfg*. Needs to agree with argument 5 in rcp_1; otherwise, results are undefined.

6th argument “& “; indicates that the process is to be run in the background; must be here for the system to work properly. DON’T CHANGE

writeDisk

writeDisk writes files to disk and has the following options:

1st argument Shelltool and its arguments

2nd argument Name of the process. DON’T CHANGE

3rd argument Full pathanme of where the data files are to be written before they are sent to tape, e.g. */export/home/data/projectName*

4th argument “& “; indicates that the process is to be run in the background; must be here for the system to work properly. DON’T CHANGE

readDma

readDma reads data from the DMA and has the following options:

1st argument Shelltool and its arguments

2nd argument Name of the process. DON’T CHANGE

3rd argument “XBAND” , “KBAND”, “SLAR”, or “GRIDS”

4th argument password that zeroes out the lat/long in the header information; if anything else (or nothing at all) is in this field, the actual lat and long, as detected by the GPS, will be written into the header.

5th argument “& “; indicates that the process is to be run in the background; must be here for the system to work properly. DON’T CHANGE

exabyte

exabyte writes files to tape and has the following options:

1st argument Shelltool and its arguments

2nd argument Name of the process. DON’T CHANGE

3rd argument Switch to turn off automatic tape eject. The options are “eject” and “no_eject” and must be in double quotes. For operational purposes this should always be “eject”.

4th argument “& “; indicates that the process is to be run in the background; must be here for the system to work properly. DON’T CHANGE

rt_vad

rt_vad calculates and displays profile data for VADs and has the following options:

1st argument Shelltool and its arguments

2nd argument Name of the process. DON'T CHANGE

3rd argument Name of the machine for profile displays

4th argument Full pathname where the GIF profile files are to be sent for archival, e.g. */export/home/data/projectName/profiles/*.

The remaining processes in the RCP script have no configurable arguments.

5.4 Starting the RADS

5.4.1 From a Cold Start

If running the RADS on two SPARCs, do Step 1; otherwise, skip to Step 2 below. For several projects the X-Band radar was configured as a two-computer system. In the description below computer 1 would be numa and computer 2 would be lurch, or the computer where the motion control software resides.

Step 1. log into computer2 as rcp
If it doesn't come up in Openwin:
computer2% openwin
Bring up a window by pushing the right mouse button and selecting Programs, Shelltool or Programs, Cmdtool. Several other windows should then appear.
computer2% cd bin
computer2% xhost computer1
computer2% scan_antn3
computer 2% VRE

Step 2. log into computer1 as rcp :
If it doesn't come up in Openwin:
computer2% openwin
Enable scrolling on all windows
From a shelltool :
computer1% cd bin
computer1% slaveWin &
computer1% RCP_cal
3
exit
computer1% dsp_load0
The output looks like:
computer1% DSP 3 loaded
computer1% DSP 2 loaded
computer1% DSP 1 loaded
computer1% DSP 0 loaded

The LED on the DSP board should be blinking a slow green. If you do not see the four lines on the screen as above, physically push the reset button on the DSP card.

Proceed from the computer1 console in a shelltool window.
computer1% dsp_load0
It should work this time, if not, repeat. If repeated attempts do not work, reboot the system, otherwise continue as below.

-put two tapes in the drives on computer1
computer1% RCP

5.4.2 From a Normal Exit

When bringing up the RADS from a normal exit:
Be sure that you logged in as rcp and did a cd to bin.
Be sure you are in a shelltool
Enable scrolling on all windows
computer1% RCP
NOTE: Often the RCP script is modified for a particular project beforehand. For example, for PacJet the RCP script was RCPpj and for Trial Manhandle it was RCPtm.

5.5 Operating the RADS

RADS operates in continuous mode and queuing mode. Continuous mode runs one scan table over and over until stopped by the operator, while queuing mode allows the operator to build a queue of different scans and run the queue for a finite number of iterations.

The primary control window of RADS is associated with the rcp_1 process. It is a large window with a white background and is labeled "NOAA/ETL RADS; *specific* RADAR"; where *specific* is replaced with one of X-BAND, Ka-BAND, SLAR, or GRIDS. In this document it will be referred to as the base window. See Appendix F, Figure 1. The x_display process also has a control window. This window has a blue background and is labeled "X Display v2.0 Control Window". In this document it will be referred to as the display control window. See Appendix F, Figure 9.

5.5.1 Continuous Mode

5.5.1.1 To Start a Continuous Scan

Select the "RUN" button at the bottom left-hand side of the base window. This will run the last scan table that was loaded in continuous mode. If no scan table was explicitly loaded, it will run the scan table given as the second argument in the rcp_1 line of the RCP script. See section 5.3 for more details.

5.5.1.2 To Stop a Continuous Scan

Selecting the “STOP” button in the base window will stop the current continuous scan immediately. The “STOP EOV” (i.e., STOP End Of Volume) button is selected if the user wishes to stop at the end of the current continuous scan or volume.

5.5.1.3 To List ALL Existing Scan Table FileNames

With the mouse cursor located in the white background of the base window, depress the right-hand mouse button and select “List Scan Table Files”. A small blue pop-up window entitled “List Scan Tables” will appear with the complete names of all the files that exist in the */export/home/rcp /rp* subdirectory.

5.5.1.4 To View/Load/Edit Scan Tables

With the mouse cursor located in the white background of the base window, depress the right-hand mouse button and select "Load File..." In the corresponding pop-up window entitled “Load Existing Scan Table File” , type "W01_rhi" or the name of an existing scan file (WITHOUT the .rp suffix) to the right of the “Filename:” label. The mode-dependent pop-up MODE window will typically appear with a local copy of the scan table information displayed. This window displays most of the operator configurable parameters. At this point the operator can makes changes to the local copy of the scan table interactively. Changes must be followed with a carriage return. It is only at then that the first level of checking will occur. Illegal values will cause a notification window to appear and the illegal value will be changed to the default value for that parameter. See Appendix F, Figure 3.

If the MODE window does not appear or has been closed, it can be brought up by first right clicking on the arrow next to the “PARAM TYPES:” label in the base window. From the dropdown window that appears, select “MODE”.

You can “CHECK” the local copy of the scan table, but you cannot “SET” or “RUN” it if there is any other scan running (this includes continuous scans or scans running from a queue). In a case in which there are no scans running, the local version of the scan table may be run continuously by selecting the “RUN” button as described in Section 5.5.1.1.

5.5.1.5 To Select a Mode for a Scan Table

From the base window, select the arrow by clicking the right mouse button next to the “MODE:” label. From the drop-down menu, select a mode by clicking on the left mouse button. NOTE: Mode 103 ONLY RUNS in FIXED BEAM (or vertical) mode.

5.5.1.6 To Select a Sweep Mode for a Scan Table

From the base window, select the arrow by clicking the right mouse button next to the “SWEEP MODE:” label. From the drop-down menu, select a sweep mode by clicking on the left mouse button.

5.5.1.7 To Examine Parameters for a Scan Table

There are ten categories or types of radar parameters that define a scan table. They are: timing, transmit, receiver, housekeeping, scan, calibration, data processing, plate, gps, and nav. In order to view these parameters, the user must select the arrow from the base window, by clicking the right mouse button next to the "PARAM TYPES:" label. The drop-down menu includes the ten parameter types listed above plus one entitled "MODE". Select a parameter type by clicking on the left mouse button and the corresponding parameter type pop-up window will appear. See Appendix F, Figures 5 and 6. The MODE window is a mode-dependent pop-up window that displays parameters commonly changed by the operator. Although the user may look at any of the parameter-type pop-up windows, many of the parameters are not operator enabled and thus cannot be modified from within the RADS.

5.5.1.8 To Create or Rename a Scan Table

Follow the instructions in 5.5.1.4 to load and edit an existing scan table that has the desired calibration information. Once done editing this local copy of the scan table, locate the cursor in the white background of the base window and select "Save As…" Another small window will appear entitled "Save Current Scan Table." Next to the label "Filename" type in the name of the NEW scan table (don't type the .rp suffix). If you are overwriting a file, a message will appear to ask if you really want to do that. The file will not be written to disk until it has been successfully checked.

5.5.1.9 To Save a Scan Table

With the cursor located in the white background of the base window, click the right mouse button and select "Save". This overwrites the most recently loaded scan table file with the local version of the scan table. The operator is prompted to make sure the local version of the scan table has been checked and that the operator really does want to overwrite the original scan table file. No files are written to disk unless they have been successfully checked.

5.5.1.10 To Check a Scan Table

Once a copy of a scan table has been loaded, the local copy of it can be checked by selecting the "CHECK" button on the bottom right of the base window. If the parameters do not meet certain criteria, a notify window will appear to warn the user. The scan table cannot be saved or run until it has been successfully checked.

5.5.1.11 GIF Images

In the display control window, select the scan type that you wish to capture as a GIF image. Select the "gif" button located midway across the bottom of the same window. This will create a "special" GIF file. The filename will have the same format as all the other GIF filenames but will be pre-pended with an "S", e.g., *S021226054051.ZShD.ppi.noaad.gif*. These files will live in the directory defined by the 4th argument to the x_display process in the RCP script. See section 5.3.

5.5.2 Queuing Mode

5.5.2.1 To Load Queues

From the white background of the base window, depress the right-hand mouse button and select "Load Queue..." In the associated pop-up window entitled "Load Existing Queue" type "ocean" or "atmos" or the name of an existing queue file to the right side of the label "Queuename:". If you don't know the name of an existing queue, you may want to list the existing queue files as shown in section 5.5.2.3. See Appendix F, Table 4.

5.5.2.2 To Run a Queue

After loading a queue file, from the white background of the base window, depress the right-hand mouse button and select "Run Queue". This copies the queue file that was most recently loaded into the current queue and begins executing that queue.

5.5.2.3 To Stop a Queue

Once a queue file is running, from the white background of the base window, depress the right-hand mouse button and select "Stop Queue". Warning: Between scans, while it is positioning and before it comes back to the main shelltool window with "received a resume", try not to stop the queue or to scroll windows. If you do stop the queue while it is positioning, it will run the first scan table in the queue anyway and you may not be able to stop it. It will stop itself once that scan table is complete.

5.5.2.4 To List All Existing Queue FileNames

With the mouse cursor located in the white background of the bse window, depress the right-hand mouse button and select "List Queue Files". A small light-blue pop-up window will appear entitled "List Queue Files" with the complete names of all the files that exist in the */export/home/rcp/q* subdirectory.

5.5.2.5 To View or Edit an Existing Queue or Create a New Queue

From the white background of the base window, depress the right-hand mouse button and select "Create/Edit Queue File". A pop-up window entitled "Create Queue…" will appear. Next to the label "Queuename to Edit:" type in the name of the queue file you wish to edit. A Textedit window will appear displaying the file requested. Continue as in any Textedit session, saving the file or renaming the file using the "Save As…" edit option. In this case and only in this case, the operator must INCLUDE the .q suffix. Exit the Texteditor. It is best to start with an existing queue file since the syntax of the queue file is well defined. If the parser does not find the correct syntax, the file will not run.

5.5.2.5.1 Syntax of the queue file

As stated above the syntax of this file must be exactly as defined below.

The first line of the queue file must begin with :
for () {
The user must fill in a positive integer between the ()s. This will indicate how many times the queue is to be run.

The last line must be:
}

In between the {}s on separate lines the user must include at least one of the following:

1. a scan table file name enclosed in double quotes. The file name must not include the .rp filename suffix e.g., “scanfilename”
2. a scan type (choices are BSCAN, PPI, RHI, A_SCOPE) enclosed in angle brackets e.g., <BSCAN>

Any other syntax will cause the queue to be ignored.

5.5.2.5.2 Example queue file

For (24) {
“ScanTable1”
“Bscan2”
<BSCAN>
“Bscan2”
“Bscan2”
“RHI”
“rhi1”
“ppi1”
<PPI>
<RHI>
“Bscan1”
}

In this example there are eight scan tables queued and three GIF commands. The three GIF commands correspond to three types of scans. Each of these commands will capture from zero to three gif images depending upon the NUMBEROF*scantype*GIFS parameter defined in the configuration file, where *scantype* is one of the scan types defined in item 2. They will each be executed once for each cycle of the eight scan queue, capturing the fields requested in the configuration file for the corresponding scan type. The queue will cycle 24 times before stopping. See section 5.2.1.9above.

5.5.2.6 Showing the Loaded Queue

From the white background of the base window, depress the right-hand mouse button and select “Show Loaded Queue”. A pop-up window will appear with the title “Show Loaded Queue” and will display the contents of the queue that was last loaded. Note this is NOT necessarily the current queue. This feature not only allows you to view queue files without going into the text editor but allows you to be prepared with the next queue to be run as soon as the current queue terminates.

5.5.2.7 Showing the Current Queue

From the white background of the base window, depress the right-hand mouse button and select "Show Current Queue". A pop-up window will appear with the title "Show Current Queue" and will display the contents of the queue that is currently running. Note this is NOT necessarily the same as the loaded queue.

5.5.2.8 Save Queue As…

From the white background of the base window, depress the right-hand mouse button and select "Save Queue As…". A pop-up window will appear with the title "Save Current Queue". This is a misnomer; it should be "Save Loaded Queue" since that is the file that will be saved. Next to the label "Queuename:" type in the NEW name of the queue file that should be saved.

5.5.2.9 GIF Images

Assuming you have set up the queue file correctly (see section 5.5.2.5) and the configuration file (see section 5.2) correctly for capturing GIF images, and you have loaded and are running the queue (see sections 5.5.2.1 and 5.5.2.2), the graphics window to be displayed must be entirely on the screen and not iconized. It doesn't have to be completely exposed, i.e., there can be other windows on top of it, but it does need to be completely on the screen. GIF files reside in the directory, which is specified in the 4th argument to the x_display process in the RCP script. See section 5.3. The pathname is typically */export/home/data/nameOfProject/gif* or */data/nameOfProject/gif*. "Special" GIF files may be captured while in queuing mode as well and work exactly as in continuous mode. See section 5.5.1.8. GIF filenames follow the convention *yymmddhhmmss.noaad.ppi.gif.*

5.5.2.10 Building a Queue that Includes GIF File Generation

The GIF images that are to be captured are defined by the operator in the configuration file (see section 5.2), independent of what field is being drawn to the screen. In order to capture the GIF image, the window flashes to the requested field, takes a snapshot, switches to the next requested GIF field, takes a snapshot, etc., and finally returns to the field that was being displayed before the GIF process started.

5.5.3 To Position the Antenna

If there are no scan tables running, the radar can be positioned to the azimuth and elevation requested in the MODE window, by selecting the "SET" button in the bottom of the base window.

5.5.4 Selecting Display Fields

By clicking the left-hand mouse button while positioned over the "DISPLAY FIELDS" button in the base window, a mode-dependent pop-up window appears. See Appendix F, Figure 7. This window displays the fields available for graphical display purposes for the mode selected for the local scan table. These buttons are on/off switches. The depressed, darker gray buttons have been previously selected. In covariance or pulse pair mode a maximum of 12 fields may be selected by depressing the appropriate buttons. In time-series modes less than 12 fields are available and therefore, this maximum number is also less. Once the maximum number of fields has been selected, the operator must depress a previously selected field in order to turn it off and may then select another field. Selecting the “Apply” button sends this information to the x_display process.

5.5.5 Setting VAD Profile Parameters

If the user selects the “VAD Params…” button from the base window, a mode-dependent pop-up window will appear. See Appendix F, Figure 8. This window will be entitled “MODE nnn VAD_PARAMS”, where nnn depends upon what mode has been selected in the local scan table. Several options are available. Default values for mode 170 are currently read in from the file *vad170.defaults*. These will be integrated into the configuration file in the near future. It is advised that the user NOT change this file.

Correlation

There is a small arrow to the right of the “Correlation:” label. The user may select from the drop-down menu that appears. Values are from 0., to 1., with increments of .1. Only DBZ values with a corresponding correlation value above this threshold will be included in the calculations.

DBZ_threshold

DBZ values below this threshold will not be included in the calculations that produce the VAD profile displays.

Elevation_tolerance

VADS with elevation values from elevation +/- elevation tolerance will be profiled. VAD scans with elevation values outside this range will not be plotted.

Elevation

VADS with elevation values from elevation +/- elevation tolerance will be profiled. VAD scans with elevation values outside this range will not be plotted.

Select Fields Graphed

The “NOAA/ETL Shelby VAD” window consists of four x-y plots. The top two plots (wind speed and wind direction) are not selectable. The bottom two plots may be selected from the four mode-dependent fields. Listed here are the four fields available for mode 170: ZShD, ZdrD, CShD, ZSvD. Since these buttons are actually switches and only two fields may be selected at any one time, the user must switch off a field before switching another field on.

Range Limits

Currently disabled

Number of Volumes to Average

This needs to be a positive integer. It defines how many volumes will be used in one calculation for the profile plots.

Number of Sweeps to Average

Same as above, only used for multi-sweep volumes.

Send

Selecting this button sends the VAD parameters defined in this window to the VAD display process.

Default

By selecting this button, the VAD parameter values are read in from the file *vad170.defaults*.

5.5.6 Scan Status Window

A small yellow pop-up status window is associated with the base window. See Appendix F, Figure 2. The status window gives the user information about the radar, the scan and queue currently being run, recording status and current tape number. If this window has been released and needs to be recovered, position the mouse in the white background of the base window. Depress the right-hand mouse button and select "Scan Status" from the drop-down menu which appears. The pop-up status window should reappear.

5.5.7 Flush Disk

When in pause mode, this will send the last disk file to the tape routine.

5.5.8 Eject Tape

When in pause mode, this will eject the tape in the active tape drive.

5.6 Display Options

All of the options in Section 5.6 appear in the display control GUI window.

5.6.1 Scan Type

There are four kinds of displays or scan types. They are Bscan or time-height displays, RHIs, PPIs, and A-Scopes. A-Scopes are x-y plots of range vs. selected field.

5.6.2 Fields

The user must select the scan type and then the field that is to be displayed for that scan type or display.

5.6.3 Change Field Scale

The user must select the scan type and then the field whose scale is to be changed. A pop-up window will appear which will prompt the user to input a minimum and maximum value for the color bar associated with that field.

5.6.4 Change Max Range

This option only applies to Bscans, thus the Bscan scan type must be selected first. Values available are 4km, 12 km, 25km and max range. Max range refers to the maximum range for the particular scan.

5.6.5 X Display Center

This applies to PPI and RHI scan types only; thus, one of those two scan types must be selected before entering a value or sliding the scrollbar. By entering a value or sliding the scrollbar, the center of the display is moved in the X direction.

5.6.6 Y Display Center

This applies to PPI and RHI scan types only; thus, one of those two scan types must be selected before entering a value or sliding the scrollbar. By entering a value or sliding the scroll bar, the center of the display is moved in the Y direction.

5.6.7 Size of Viewport

This applies to RHIs and PPIs only. RHI or PPI scan type must be selected before setting this number. This number is a percentage that can be either typed in or adjusted using the slider and that has a range of 0 to 110. The feature allows the user to zoom. 100% fills the screen with the number of range gates. 110% makes the window 10% larger than the data. Anything less than 100% makes the viewport smaller than the data and thus zooms in on the data.

5.6.8 Select Correlation Threshold/100.

First, select the scan type and the field. Then type in or slide the scrollbar to the desired value. Correlation values are from 0 to 1. These numbers are values between 0 and 100; thus, they are divided by 100. Values for the desired field that have a correlation less than the value selected here will not be displayed.

5.6.9 Select Range

For mode 103 A-Scope displays only. A pop-up window will appear to query the user for the selected range in km. See section 5.7.2.1.

5.6.10 Select Number Beams to Average

For BScan displays only. A pop-up window will appear and the user may specify a positive integer number of beams to average for display purposes only. By setting a larger number, a longer period of time is displayed across the x-axis of the BScan display, which holds a total of 600 averaged beams. Typically, assuming a data rate of eight beams per second, if the operator uses a value of six, approximately eight minutes of data will be displayed before the oldest beams scroll off the left side of the image.

5.6.11 Erase

Select the scan type and then select the "Erase" button. Once the selected scan type becomes active, the erase command will be issued.

5.6.12 gif

Selecting this button allows the user to capture a "special" GIF image. This button conforms to the same method as previous buttons; i.e., the scan type must be selected before selecting the "gif" button in order to define which window will be GIFed. The selected window must be completely on the screen and not iconized. Filenames will have the same format as all GIF filenames but will have an S pre-pended, i.e., *Syymmddhhmmss.noaad.ppi.gif*.

5.6.13 OL on/off

This button toggles the geographic overlays for PPI scan types only. The PPI scan type should be selected before toggling the "OL on/off" button. The change of overlay status will take effect once the next PPI commences. Select the button once and then wait, otherwise you will not know what state the overlays are in.

5.6.14 Quit

It is best not to use this feature because it will stop the x_display process and will clean up the shared memory. The rest of the processes will continue to operate smoothly. At this point, recovering the displays is currently not an option.

5.7 Other Display Options

5.7.1 Xwindow Dumps

To capture an Xwindow dump of a window on the screen, type

computer1% xprint.x

Select the desired window by double clicking on the left-hand mouse button. This will save a copy of the post-script file in */data/projectName/xwd* and will print it on the color printer (Epson 660), which you might not have. (If none of this works, you might need to edit the xprint script.)

5.7.2 Time-series Displays (mode 103)

5.7.2.1 A-Scope Displays

Eh.R, Eh.I, Ev.R and Ev.I displays are for all range gates and the first trigger. The PwSh and PwSv displays show the power spectral density in log DB of a selected gate vs. +- Nyquist. These gates default to gate 1. The user can select other gates by selecting the "Select Range" button and inputting a legal value in km. Illegal values are ignored, but the error message only comes to the x_display shelltool screen.

5.7.2.2 BSCAN Displays

These display PwSh and PwSv. All 150 range gates are displayed, gate 1 at the bottom of the screen and gate 150 at the top. Again, the x axis is +- Nyquist. Colors are the amplitude of the log power spectral density function.

5.7.3 VAD Profiles

The window labeled "NOAA/ETL Shelby VAD" must be open in order to produce plots. If the window is iconized, the plots will not be drawn. The process draws to the screen when the radar does VADS at the angles defined by the user in the rcp_1 mode-dependent window entitled "MODE nnn VAD_PARAMS". See section 5.5.5 for more information on selecting VAD parameters. This version of the code automatically captures GIF images of the profiles and writes them into the directory, which was defined as the 4th argument in the rt_vad process line in the RCP script. A typical directory name would be */data/projectName/profiles*. See section 5.3 for more details.

5.8 Tape Handling

On the right-hand side of the base window, there is a "RECORD_MODE": switch which allows the user to select "Record" or "No Record". If you select "Record", a notification window asks "You have a current TNUM of nnn. Is this correct? Yes No". If you select "No" a pop-up window entitled "INITIALIZE RECORD MODE" appears. Next to the "Tape Number:" label, the correct tape number should be typed followed by a carriage return. Legal values for tape numbers are integers between 0 and 10000. The pop-up window will disappear automatically.

5.9 Normal Shutdown

Select the "STOP" button in the base window if you are running a continuous scan. If you are running a queue, position the cursor to the white background of the base window and depress the right mouse button. Select "Stop Queue" from the drop-down menu. Select the "EXIT" button in the top left-hand corner of the base window and select "YES" in the notification window.

5.10 Abnormal Exiting

5.10.1 Various Scenarios

SCENARIO 1

You are running and the RCP hangs, i.e., you lose mouse control or the windows disappear:

computer1% send_rw_command quit

If this doesn't clear all the windows or it hangs try:

computer1% ^C

If this still doesn't work, you should go to computer2:

computer2% rlogin -l rcp computer1

or **computer2%** ssh –l rcp computer1

computer1% zap_now real_time

SCENARIO 2

You start up the RCP and the color maps are very strange:

exit the RCP by selecting the exit button

try it again by typing:

computer1% RCP

If this doesn't work:

computer1% ipcs.sema.rmv

computer1% RCP

If this still doesn't work, type the following:

computer1% ipcs.sema.rmv

computer1% ipcs

The output will look like :

IPC status from <running system> as of Fri Feb 23 21:17:17 2001

Message Queue facility not in system.

T	ID	KEY	MODE	OWNER	GROUP
Shared Memory:					
m	0	0x50000c95	--rw-r--r--	root	root
m	1	0x00000666	--rw-rw-rw-	rcp	et6
m	2	0x00000667	--rw-rw-rw-	rcp	et6
m	103	0x00000064	--rw-rw-rw-	rcp	et6

Semaphores:

type:

computer1% ipcrm -m n

where *n* is 103 in this example. You DO NOT want to ipcrm -m the process owned by root or the id's corresponding to the 666 or 667 KEYs. (If you do, you'll need to start over from Section 5.4.1 Step 2 "From a Cold Start".) Proceed with:

computer1% RCP

SCENARIO 3

You are starting RADS and the color maps are very strange and you don't get a main RADS RCP control base window. Try it again:

computer1% RCP

SCENARIO 4

GDB errors:

zap_now real_time

follow the ipcs/ipcrm directions given in SCENARIO 2 above
proceed with:
computer1% RCP

SCENARIO 5

system panic :
There is only one thing to do :
On computer2 in VRE window type
> HV OFF
to turn down transmitter

If at the > prompt on computer1,
> boot

Start over at the very top of Section 5.4.1 "From a Cold Start". If currently not at the prompt boot on computer1, power cycle computer1.

SCENARIO 6

RADS hangs :

from a window on computer2 console type:
computer2% rlogin -l ops computer1
computer1% send_rw_command quit

If the windows do not disappear:
computer1% ^C
computer1% zap_now real_time
computer1% RCP_cal
3
exit
at this point it is necessary to be on the computer1 console in a shelltool window:
computer1% RCP

SCENARIO 7

Unserviced interrupt error messages start scrolling in the console window or take over the windowing system completely and write over the entire screen. Note: interrupt messages scroll in the console window for a few seconds between continuous scans; this is normal and NOT a problem; ignore it. If interrupt messages continue for a minute or more (or take over the screen), there is a problem. Proceed as follows:
computer2% rlogin -l rcp computer1

or
computer2% ssh –l rcp computer1
(should still be on computer2's console here)
computer1% RCP_cal
3
exit

If this does stop the interrupts from scrolling:
check the led on the front panel of the DSP card

If the light is flashing a slow green:

proceed from a computer1 shelltool window :
computer1% RCP

If you can't get to RCP_cal or this does not stop the interrupts from scrolling or the DSP light is not blinking a slow green:

physically push the reset button on the DSP card

proceed from the computer1 console in a shelltool window
computer1% dsp_load0
computer1% RCP

Continued failures indicate that a reboot of computer1 is necessary.

In the dual SPARC system:
In the VRE window on the auxiliary computer type
> HV OFF

On either system configuration :
Try a soft boot first on computer1. From superuser or root account, type
init 0

If this doesn't eliminate the problem, it will be necessary to do a hard boot (power off the system). Regardless, it will be necessary to start over.

Follow the directions in section 5.4.1, starting with Step 2 “log into computer1 as rcp”.

5.11 Warnings

It is best to start the RCP with two tapes in the drives, even if you are not currently in the record mode. Always try to enable the scrolling on a shelltool window and run the RCPscript from that window. Never depress the mouse button for very long, because it will cause the RADS system to crash and possibly cause the entire system to panic. (This occurs because the mouse has highest priority and RADS starts dropping interrupts).

In mode 103 (time-series mode) be very cautious about creating a lot of mouse activity. As the performance meter shows, the CPU is running well over 60% and peaking at 100% while calculating 150 plus FFTs. Any mouse activity at that time could be fatal. Also, scan tables of greater than 60 second duration (SWTM) in this mode will probably crash since the load on the system becomes too great and the SPARC starts dropping interrupts. Whenever this happens for more than 30 seconds, the system cannot recover.

If you bring up a scan table and change the parameters and the mode and it reverts back to all the old parameters, this is because the radar parameters are not written until “check” is selected. Selecting “check” before changing the mode will correct this problem.

It is necessary to load the queue in order to re-read the scan tables. If a change is made to a scan table while that scan table is in the current queue, the change will not appear until the queue has been stopped, loaded and then restarted.

After exiting the program, there will quite possibly be a long delay before the exabyte code and the delete_file code complete because they are writing the remaining files to tape. If the RCP is restarted, these two programs will automatically be aborted and the output data files will remain on the disk. In order to remove those files from the disk, there is code that is invoked with copyFiles. (copyFiles is found in */export/home/rcp* or */export/home/rcp/bin*). This needs no arguments, but will need at least one tape in a drive and will start up the exabyte code, the delete_file code and the feedExa code. This mini-system will write all the files in /data/projectName to tape and delete them. Even though there is typically a lot of disk space in /data, this is a good idea. This is also useful if the system aborts for some reason. It cannot be run simultaneously with the RCP, since there would be serious contention for shared memory as well as tapes. It is recommended that this be run before too many of these huge files accumulate on the disk. The more files on the disk, the longer this process takes. To kill this process (which doesn't lose any data) either start the RCP or type (preferred)

computer1% send_disk_command "disk_write quit"

If this doesn't work type :

computer1% send_rw_command quit

If this doesn't work type:

computer1% zap_now real_time

Only about 30 files can be written to disk and not written to tape and deleted while running the RCP. This occurs when you are running mode 103 continuously, if you forget to put the tapes into the drives, or if the disk space is exhausted.

6. ENHANCED VERSIONS OF RADS

6.1 SLAR

In cooperation with the U.S. Coast Guard, ETL installed a slightly modified RADS on an HC-130H aircraft as a demonstration for International Ice Patrol (http://www.uscg.mil/lantarea/iip/home.html) personnel. RADS was slaved to, and operated in parallel with, the existing data system which used dry film technology to record its images. The radar used was the Coast Guard's AN/APS-135 X-band Side-Looking Airborne Radar (SLAR) (http://www.uscg.mil/lantarea/iip/FAQ/faq25.html).

The AN/APS-135 radar operates at 9250 MHz and has a pulse width of 200 ns, which gives a 30-meter resolution. This is a simple non-Doppler radar. RADS is normally used on Doppler radars with polarization diversity. The images were produced from 768 range gates spaced at 200 ns.

The RADS was modified by adding a separate process to ingest navigational (Nav) data. This data stream includes roll, pitch, yaw, acceleration, altitude, longitude, and latitude. The x_display process was also modified to better display the SLAR images and include geo-positioning information.

This system was flown on many USCG missions. In February 2000 the system was flown off the coast of Newfoundland and produced several real-time images of icebergs. One of these images is presented as Figure 7 in Appendix F. These images are screen dumps done by the operators on the plane and have not been altered or enhanced in any way.

6.1.1 Image Format

The images are snapshots of what the operator sees on the aircraft. New data enters the image at the top, scrolls down and eventually falls off the bottom of the screen.

The color of the images represents the strength of the radar echo per the color bar at the bottom, with weaker echoes being represented by blues and greens and stronger echoes by oranges and reds. In addition, very strong echoes, those that are beyond the top of the red scale, are painted as black. The fact that these particular images tend to show open water as blue or green is only a coincidence, and cannot be relied upon in general, since the radar return from water is highly dependent on the sea state.

The scale at the bottom is the range in kilometers, looking out one side of the aircraft. Whether it is the left or right side can be discerned by whether the zero range point is on the left or right side of the image. There is a blind spot from about 0 to 3 km, and there may be artifacts in the image here. Along one side of the image is the time in GMT, which indicates that these images took about 3-4 minutes to acquire. Other information included is the latitude and longitude of the aircraft at the current time, which is the zero range point at the top of the image.

6.1.2 Iceberg Image

In this image, the sea is mostly covered in sea ice (yellow, orange and red) but icebergs, driven more by undersea currents than by the wind, have opened up leads (channels) in the ice field. Here the open water is mostly blue or green. If you look between the 15:51:27 and 15:52:02 time marks, you will see leads with black dots in the middle, or at the upper right, which represent the icebergs (some icebergs show as deep red). There is also an iceberg in the middle of the image right on the 15:52:37 time mark.

6.2 Radiometer

Time-stamped radiometer data can be sent to RADS from the Radiometer PC every minute or every 15 seconds via a serial port. The variables sent are typically integrated liquid, integrated vapor, and brightness temperature from each channel as well as surface meteorological data. Four additional shared memory processes are required for this configuration. readRadiom ingests radiometer data and then another process, radReformat reformats the data. writeRadiom writes radiometer data into hourly disk files. These files are typically between 30 and 50 Kbytes per hour.

radReformat also sends the data to the radiomDisplay process for display. The radiometer displays are updated as soon as RADS receives the data. Radiometer data are field types, displayed in an A_SCOPE window. Plots are x-y displays of integrated liquid water, integrated water vapor or brightness temperatures vs. time (on the x-axis). The variables are plotted in different colors on the same display.

In this configuration if the user chooses “record” mode in the RCP_SPARC control GUI, the radiometer data are written to disk only.

6.3 VADS

When VADS are being run, a process called rt_vad is often run in RADS. This process displays profiles of wind speed, wind direction, and two of the following three fields: Linear Reflectivity, Log Reflectivity, or Correlation. The user chooses two of those three fields to be displayed, as well as for which elevations. There is a separate pop-up GUI window which allows the user to set these and other VAD profile parameters. This window appears when the user selects the "VAD Params…" button from the base window. See sections 3.2 and section 5 for further details.

6.4 Dual SPARC System Configuration

In this configuration the actual scanning is done by a VME-based SPARC 5 that is resident in the radar. This computer communicates with RADS through Ethernet. Scanning information flows from RADS to the resident computer. The actual antenna position is read directly by RADS from the synchro lines.

6.5 GRADS

Recently, ETL has designed and is in the process of implementing a system to detect conditions likely to cause icing problems in aircraft. Called the Ground-based Remote Icing Detection System/FAA Icing Remote Sensor Testbed (GRIDS/FIRST), it will be the most sensitive cloud radar ever built. In brief, GRIDS will integrate a Ka-radar, a microwave radiometer and surface meteorological data as well as model input from a standard meteorological model, to calculate and display aircraft icing hazard information. The software for this new GRIDS RADS (GRADS) is being built upon the existing RADS architecture and implementation. While large modules of the code are being converted to Java, leveraging off of the RADS provides a flexible development path that allows for frequent demonstrations of GRADS while it is still being created. Much of the new GRADS will look and feel like the old RADS system.

Hardware upgrades are being driven by technological advances and the need for cost reduction. GRADS will be based on the PCI bus, rather than VMEbus, and will use a PC CPU running Linux. It will also incorporate digital receiver technology, simplifying the receiver in the radar and increasing dynamic range, among other things. This new architecture still allows for a flexible upgrade path.

7. SUMMARY

ETL has developed and deployed the RADS for its numerous ground-based and airborne radar systems. The RADS offers a versatile and reliable solution to remote sensing data acquisition and display. It has been proven to take dependable data on various radar systems for numerous projects (see Appendix G) over nearly a decade. Its strengths lie in its modular hardware and software design, which allows for much versatility. Because of the modular software architecture, it is straightforward to add new processes or replace old ones, as was demonstrated with the SLAR system.

RADS supports a configurable initial parameter definition file which the operator can build before the project commences. This configuration file is read in during RADS initialization. Since RADS is GUI driven, the operator has additional control of various parameter settings during operations. This makes it easy to change scanning strategies quickly without disrupting data collection.

Among the many assets of RADS is its ability to support fast scanning ground-based radar systems, as well as fixed-beam and airborne radar systems. By caching the data to disk, tape drive throughput does not limit the data acquisition.

RADS delivers high-resolution displays of radar data in several scanning mode types. It displays A-Scopes while displaying images of the data simultaneously and independently of scan type.

ACKNOWLEDGEMENTS

The authors wish to acknowledge and thank the following people: Robert Kropfli and M.J.Post for their support and suggestions over several years of development, testing and modifications; Kurt Clark and Bruce Bartram for hardware support and ideas for improvements; Irina Djalkvola for software support; Wendi Madsen for system administration and support; and Al Gasiewski, Tim Schneider, and Rich Beeler for continued support and direction.

8. REFERENCES

Barnes, S.L., 1980: Report on a meeting to establish a common Doppler radar data exchange format. *Bull. Amer. Meteorol. Soc.,* **61,** 1401-1404.

Gibson, J.S., and B.E. Martner, 1995: Interactive auxiliary real-time data system for NOAA/ETL Doppler radars. *Preprints, 11th International Conference on Interactive Information and Processing Systems (IIPS) for Meteorology,* Dallas, Texas.

Kropfli, R.A., and S.F. Clifford, 1996: The coastal ocean probing experiment: further studies on air-sea interactions with remote and in-situ sensors. *IGARSS '96,* Omaha, NE, IEEE, NY, NY, 1739-1.

Martner, B.E, Kurt A. Clark, Sergey Y. Matrosov, W. Carroll Campbell and Janet S. Gibson, 2001: NOAA/ETL's Polarization-Upgraded X-Band "Hydro" Radar*, 30th International Conference on Radar Meteorology,* Munich, Germany.

Martner, B.E, Bruce W. Bartram, Janet S. Gibson, W. Carroll Campbell, Roger F. Reinking, and Sergey Y. Matrosov, 2002: An Overview of NOAA/ETL's Scanning Ka-Band Cloud Radar, AMS, *16th Conference on Hydrology,* Orlando, Florida.

Moninger, W.R., 1983: Design and development of a radar control program for the NOAA/WPL pulse Doppler radars. *J. Appl. Meteor.,* **22,** 859-862.

Corresponding authors' addresses: Janet Gibson, NOAA/ETL, 325 Broadway, Boulder, CO 80305, USA; e-mail: Janet.S.Gibson@noaa.gov and Carroll Campbell, NOAA/ETL, 325 Broadway, Boulder, CO 80305, USA; e-mail: Carroll.Campbell@noaa.gov.

Appendix A.

P2.16 **A PROGRAMMABLE REAL-TIME DATA PROCESSING AND DISPLAY SYSTEM FOR THE NOAA/ETL DOPPLER RADARS**

W. Carroll Campbell
CIRES, University of Colorado, Boulder, CO

Janet S. Gibson
NOAA Environmental Technology Laboratory, Boulder, CO

1. INTRODUCTION

NOAA's Environmental Technology Laboratory (ETL) has been developing and testing a dual-use atmosphere and ocean sensing radar for three years. Because this dual-polarized, variable pulse width, X-band radar can scan negative, as well as positive, elevation angles, it has been used in two field projects for observing ocean surface features (Kropfli and Clifford, 1996), and numerous projects for over 15 years where it performed only as an atmospheric radar.

The expanded capabilities of this radar require a new data acquisition system that allows the radar to take data faster than previous systems and to calculate several additional parameters. With these new parameters come increased demands for real-time recording and display options. In the past the NOAA/ETL atmospheric Doppler radars have had more limited real-time acquisition and display capabilities (Moninger, 1983; Gibson and Martner, 1995).

2. The Radar Acquisition and Display System

The Radar Acquisition and Display System (RADS) is a transportable system designed to record and display data from a research Doppler radar with polarization diversity. Physically, the unit consists of a 6U VMEbus chassis, an external color monitor, keyboard and mouse.

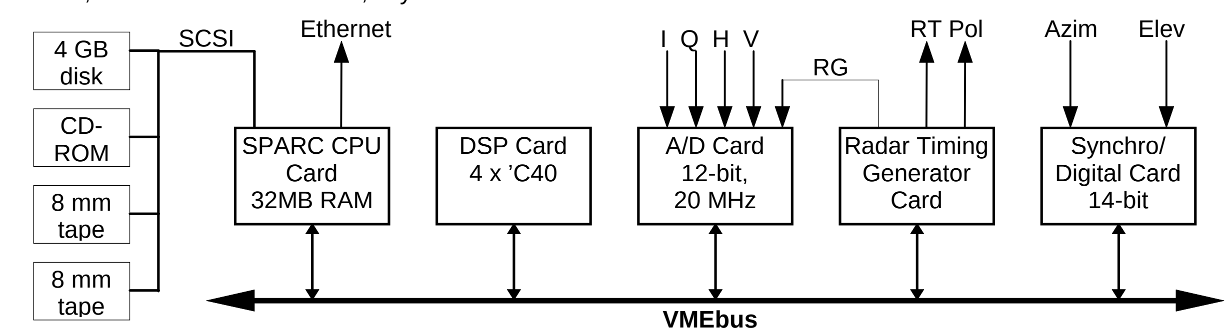

Figure 1. Radar Acquisition and Display System

Connection to the radar is made at four levels: radar timing signals (radar trigger and polarization control), receiver (in-phase and quadrature video, log output from both horizontally and vertically polarized receivers), synchro lines for monitoring azimuth and elevation, and Ethernet for directing antenna scanning (see Figure 1).

2.1 Hardware Design

All components of the system are purchased off-the-shelf except for one VMEbus card: the Radar Timing Generator (RTG). The RTG generates radar triggers, range gates and polarization control signals. Other VMEbus cards are a SPARC-based CPU, a DSP (Digital Signal Processing) card containing four TMS320C40 processors, a four-channel 20 MHz A/D card with buffer memory, and a synchro/digital card. The chassis also contains a hard disk drive, a CD-ROM drive, and two 8 mm cartridge tape drives.

* *Corresponding author address:* W. Carroll Campbell, NOAA/ERL, M/S R/E/ET6, 325 Broadway, Boulder, CO 80303; e-mail: wcc@etl.noaa.gov.

2.2 Software Design

Software is divided into two parts: that which runs on the SPARC, and that which runs on the DSP card. The SPARC runs Solaris 2.5, which is an SVID-compliant (System V Interface Definition) UNIX. UNIX is not often thought of as a real-time operating system because of its long interrupt response time (among other reasons), however it has proven to be adequate for our purposes because of the capabilities of the DSP card to moderate and buffer the data flow. No operating system is used on the DSP card, which is programmed in C and assembly.

3. DSP Processing

In operation, a beam (or ray) of data is buffered in the A/D card, and is then transferred to the DSP card over VMEbus. The four identical processors on the DSP card each take a fourth of those data, divided by range gate, and perform a processing algorithm. If an auto-covariance algorithm (*e.g.,* pulse-pair) is performed, there is typically a factor of 15 or more reduction in the amount of data. Other algorithms, such as power spectrum, typically give only a factor-of-two reduction in the amount of data.

Data from this processing are buffered in the memory of the DSP for up to five beams. This amount of buffering has been found to work quite well, as long as the SPARC is not performing compute-bound activities, such as compiling, while it is also trying to acquire, display and record data. Each beam of data is tagged with a header that contains the time, azimuth, elevation and a serial number. These data are then transferred to the SPARC.

The DSP calculates data products for recording and for display. The display data products will be discarded by the SPARC after being displayed, since they can be recalculated from the recorded data products, and recording them to tape would restrict the data rates possible.

4. SPARC Processing

When the SPARC receives the data from the DSP, it first divides the data into record and display packets and affixes a Universal Format (UF) header (Barnes, 1980) to each. These packets are then sent to different processes.

The record process writes the data to 8 mm tape. The tape format is very similar to the UF tape format, except that the data are floating-point. A post-processing program converts these data to 16-bit integer to produce a true UF-format tape that can be read by existing software.

The display process separates data into FIFO (First In, First Out) buffers for each field that are maintained for 600 beams of data. The operator may select one field for display and that field is output to the screen. This process runs in a continuous loop, such that if beams of data come in faster than they can be displayed, all beams available will be drawn to the screen simultaneously. The operator can change fields and immediately see up to 600 beams. In PPI mode, this is normally enough data to display an entire rotation.

The RADS display subsystem has been modeled after the Auxiliary Radar Control System (ARCS) described by Gibson and Martner (1995). It has the capability of real-time areal displays of at least eight derived fields for PPI, sectors, RHI, and fixed-beam scans as they occur. It can also display real-time two- and three-dimensional plots of Doppler spectra, derived VAD fields (wind speed, directions, *etc*.) and delta-k and differential phase derived fields. No special hardware is required to generate the images.

The operator controls the radar through a GUI (Graphical User Interface) window on the color monitor. All radar parameters may be viewed, and certain subsets of parameters may be selected to be shown simultaneously on the screen. Thus if the RADS is in Pulse-Pair mode, only those parameters that are pertinent to that mode, and that are changed frequently in that mode, will be displayed. In addition, because of the large number of radar parameters (> 80), they are grouped by primary function (timing, calibration, scanning, transmit, receiver, data processing and housekeeping), and may be displayed by these groupings.

The operator may select functions through soft buttons on the GUI, and in this way controls scanning, processing and recording. The actual scan control is done by another VME-based computer that is resident in the radar. This computer communicates with RADS through Ethernet, however antenna position is read directly by RADS from the synchro signals.

5. Initial Uses and Advantages of RADS

RADS accompanied the Doppler radar system to the Coastal Ocean Probing Experiment in Oregon in August /September, 1995. At that time, the new system demonstrated the ability to take real-time delta-k data and display them. The system also recorded time series data. In Snowrad '96, the system provided real-time power spectrum displays of Doppler velocity in three different formats, and recorded pulse-pair data.

In the laboratory, the system has demonstrated the capability to process 512 range gates in a pulse-pair processing mode with 1.0 ms between 64 pairs. To date, the following modes have been programmed: time-series, spectrum, pulse-pair, delta-k, and a differential phase mode which includes horizontal-vertical correlation, $\rho_{HV}(0)$.

RADS provides several advantages over the previous system, particularly in terms of data rates and operating mode flexibility. The delta-k, Doppler spectra and differential phase capabilities, for example, are entirely new features. RADS is made almost entirely from off-the-shelf purchased components, which reduces development, replication and replacement costs. Since it is completely programmable, new algorithms can be readily implemented. Range gates can be spaced as closely as 7.5 m. Use of a modern computer and operating system allows a much more capable radar control program to be implemented.

One advantage of using the VMEbus is that it is a true multi-processing bus, which allows capabilities to be increased by adding more processing elements. The main limitation is a bus bandwidth of 80 MB/s. The only high bandwidth path in RADS is between the A/D and the DSP which, with the parameters previously given, is 8.2 MB/s. Thus, available bandwidth could support a second DSP and/or A/D card. Other possible cards make almost no demand on bandwidth because of the reduction in data volume provided by the DSP card.

6. Acknowledgments

This work was supported by the Advanced Sensor Applications Program, U.S. Department of Defense.

7. References

Barnes, S.L., 1980: Report on a meeting to establish a common Doppler radar data exchange format. *Bull. Am. Meteorol. Soc.,* **61,** 1401-1404.

Gibson, J.S., and B.E. Martner, 1995: Interactive auxiliary real-time data system for NOAA/ETL Doppler radars. *Preprints, 11th International Conference on Interactive Information and Processing Systems (IIPS) for Meteorology,* Dallas, Texas.

Moninger, W.R., 1983: Design and development of a radar control program for the NOAA/WPL pulse Doppler radars. *J. Appl. Meteor.,* **22,** 859-862.

Kropfli, R.A., and S.F. Clifford, 1996: The coastal ocean probing experiment: further studies on air-sea interactions with remote and in-situ sensors. *IGARSS '96,* Omaha, NE, IEEE, NY, NY, 1739-1741.

Appendix B.

Radar Timing Generator Description (RTG)

1. INTRODUCTION

The Radar Timing Generator (RTG) resides on a single VME card, and generates all the repetitive signals necessary to trigger the radar, control data acquisition, control transmit and receive polarization, and drive A-scope displays. All timing parameters are programmable from the VME bus.

2. OVERVIEW

2.1 Programmers' Model

All programmable parameters are available on the VME bus in 16-bit words, although mostly the lower order bits are used. VME addressing is big-endian, and so the most significant byte (even address) is addressed. Before changing parameters, the Run Enable bit in the overall control word should be cleared. This causes all counting to stop, and loads all counters with the values in their associated hold registers. When the Run bit sets, counters are allowed to run, and radar triggers and range gates start.

The RTG is divided into five functions: Overall Controller, Radar Trigger Generator, Range Gate Generator, Transmitter Controller and Polarization Controller. Each of these functions has a group of 16 byte addresses, or eight word addresses, reserved for it.

The Overall Controller function controls the VME function of interrupt control, as well as internal/external clock selection for timing.

The Radar Trigger Generator controls the spacing between Radar Triggers (RTs), the delay between the first trigger (RT1) and the second trigger (RT2) in double pulse mode, the number of triggers in a beam, and the number of triggers between beams.

The Range Gate Generator controls the delay from RT to the first range gate, the range gate spacing, and the number of range gates in the block.

The Transmitter Controller controls the spacing of the two pre-triggers, and the length of the radar trigger.

The Polarization Controller controls transmitter and receiver polarization.

2.2 Signal Model

The RTG can either be a master, in which case it generates all timing from an on-board 20 MHz clock, or a slave, in which case it receives an external 16 MHz clock, external radar triggers, and an external sync signal.

Timing begins with an external 16 MHz clock or an on-board 20 MHz clock. A program-controlled bit selects which clock to use. The selected clock is divided down to 1 MHz and used to generate radar triggers. This 1 MHz clock is used to generate all timing parameters except for the range gate spacing. After Run is set, there is a delay of PRPR before the first RT1 occurs. The first trigger generated is actually the first occurring pre-trigger: PRT1B. The next pre-trigger, PRT1A, is generated as a delay from PRT1B, as is RT1. PRT1B also starts the RT2 counter, which generates PRT2B. PRT2A and RT2 are generated as delays from PRT2B. The correspondingly delayed triggers are ORed together to produce composite triggers RTB, RTA, and RTint. RTint is the same trigger that is eventually sent to the pulse modulator, RT, except that RT goes through a circuit that allows its length to be programmed. All triggers except for RT are one clock period long.

Triggers are counted in two ways: in a beam counter, which counts the number of RT1s in a beam (NTRG); and in a wait counter, which counts the number of RT1s to wait between beams (NPWT). These counters have no effect on the actual triggering of the radar, but rather serve to slow down data acquisition by inhibiting range gates during the wait period by use of signal RGinh. A Beam signal is also produced at prt1b time that marks the beginning of data acquisition for the beam. A MidBeam signal is also generated that generates a VME interrupt.

RT1 and RT2 presently drive a single range gate generator that produces blocks of range gates following each trigger. Future enhancements may generate two blocks of range gates, the first of which may overlap RT2. Software must still insure that the two blocks of range gates do not overlap each other.

3. FUNCTIONS WITHIN THE RTG

3.1 Overall Control

Interrupt control is here, including bits to enable interrupts and a programmable interrupt vector. Also the internal/external clock select and the Run control are programmed here.

The Run Enable bit should be turned off whenever other registers are being updated. When Run Enable is set, Run will either set immediately or will wait for an external sync signal before setting, depending on the setting of the External Sync Enable bit. Internally generated radar triggers will begin one PRPR period after Run sets. Externally generated radar triggers may begin at any time, depending on the timing of the master trigger generator.

3.2 Radar Trigger Generator

The RT Generator function is contained on one EPLD (trig_gen) which runs from a 1 MHz clock.   The RTG generates radar triggers in a double pulse mode. The two triggers (RT1 & RT2) each have a variable range gate block following them. Single-trigger mode works the same as double-trigger mode except that RT2 is turned off.

3.3 Range Gate Generator

The RG Generator function is contained on one EPLD (gate_gen) which runs from a 20 MHz local clock, or a 16 MHz external clock.

Data acquisition on the ICS-150 board is controlled by two external signals: RG0 and RGclk. Other parameters are programmable via the VME bus in the ICS-150 itself. These parameters include the number of range gates and a decimation value. The board must receive a continuous clock at the range gate rate, RGclk. To start data acquisition, it then must receive a signal, RG0, before the first range gate. When it receives RG0, the first range gate will occur at the next RGclk edge following. It will then acquire the number of range gates programmed into it, which may vary from 128 to 8192 in powers of two. The RGclk may only vary from 3.75 MHz to 20 MHz, and so a decimation register is provided to allow slower sampling. This register may be programmed from 1 to 256.

In operation, the RTG supplies a continuous 20 MHz clock for RGclk, and the decimation register is used to set the range gate spacing. This means that the spacing can be set in multiples of 50 ns. Therefore, the only RTG parameter that actually controls the data-acquisition process is the range gate delay. Everything else is controlled by the programming of the ICS-150.

It is desirable, however, to be able to view range gates on an A-scope or PPI and so range gates are generated that mirror the ICS-150's internal sampling pulses. For this reason, there are additional registers for range gate spacing and number of range gates. Note that although any number of range gates may be set for the A-scope display, the ICS150 only generates gates in powers of two. This means that the ICS150's gates may overlap RT2, although the gates shown on the A-scope do not overlap. If this occurs, data acquisition slows dramatically because no gates are generated following RT2.

The range gate delay is controlled in steps of 1 µs. The range gate spacing is controlled in steps of one clock period (either 50 ns or 62.5 ns), up to 256 clock periods (either 12.8 µs or 16 µs). The number of range gates can be programmed from 1 to 16384.

3.4 Transmitter

The Transmitter function controls the pre-trigger delays, and the length of the radar trigger, which may be programmed in increments of one clock period.

3.5 Polarization Control

There are two eight-bit registers: one for transmit polarization and one for receive polarization. A 0 signifies horizontal polarization and a 1 signifies vertical polarization. The MSB is the first polarization at the beginning of the beam. In single-trigger mode, each RT1 shifts out a bit; in-double trigger mode, RT1 and RT2 both shift out a bit. The shift register is arranged as a circular shift register and shifting continues until the beam is finished. At the beginning of a beam, the register is initialized to the starting pattern so that all beams have identical polarization.

4. EPLD PARTITIONING

The logic for the RTG is portioned into three EPLDs, as described below.

4.1 vme_xmit

This EPLD contains VME functions and the Transmitter Controller.

4.2 trig_gen

This EPLD contains the Radar Trigger Generator and the Polarization Controller.

4.3 gate_gen

This EPLD contains the Range Gate Generator and the Overall Controller.

5. RTG ADDRESS SPACE

All addressable registers on the RTG1 board are 16 bits and are addressed using a 10-bit address, with the LSB of the address always zero. VME addressing is big-endian, so the even address indicates the most significant byte in a 16-bit word.

Table 1. Overall Address Assignment

Hex Address	Function
80-8F	Overall control
90-9F	Radar Trigger Generator
A0-AF	Unassigned
B0-BF	Range Gate Generator
C0-CF	Transmitter Control
D0-DF	Unassigned
E0-EF	Polarization Control
F0-FF	Unassigned
100-1FF	Unassigned

Table 2.Overall Controller

	Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0x80 INTC	test bit				interrupt control	interrupt control*	green LED indicator	red LED indicator
Type	R/W				R/W	RO	WO	WO
=0	no function				interrupts disabled	no interrupt pending	green LED off	red LED on
=1	no function				interrupts enabled	interrupt pending	green LED on	red LED off
0x82 INTV	vector	vector	vector	vector	vector	vector	vector	vector
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x84 RTGC				16/20 MHz ext clk select	Run	Ext Sync Enable	Run Control	Clock Select
Type				WO	RO	WO	WO	WO
=0				20 MHz ext clk	Run false	don’t wait for ext sync	don’t Run	use local 20 MHz clock
=1				16 MHz ext clk	Run true	wait for ext sync	Run enabled	use external clock

*not implemented

Table 3. Radar Trigger Generator Addressing

Hex Address	**Function**	**Abv**	**Bits**	**Format**	**Range**
90|91	Control Word	RTCN	2		
92|93	Pulse Repetition Period in 1μsec increments	PRPR	14	2’s comp	3 to 16384 μsec
94|95	Spacing between RT1 & RT2 in 1 μsec increments	TRGS	10	2’s comp	2 to 1024 μsec
96|97	Number of RT1s in beam	NTRG	12	2’s comp	2 to 4096
98|99					
9A|9B	Number of RT1s to wait between beams	NPWT	10	2’s comp	0 to 1023
9C|9D					
9E|9F					

Table 4. Radar Trigger Control Word

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x90 RTCN					Skip RT*		Single/Double	Internal/ External
Type					WO		WO	WO
=0					don't skip		Single trigger	Internal triggers
=1					Skip RGs every other RT		Double trigger	External triggers

*not implemented

Table 5. Range Gate Generator Addressing

Hex Address	Function	Abv	Bits	Format	Range
B0|B1	Range Gate Control Word	RGCN	2		
B2|B3	Delay from RT to RGs in 1 μsec increments	DLAY	10	2's comp	2 to 1025 μsec
B4|B5	Spacing between range gates in 50 ns increments	SPAC	8	2's comp	1 to 256
B6|B7	Total number of range gates	NRGT	14	2's comp	1 to 16,384
B8|B9					
BA|BB					
BC|BD					
BE|BF					

Table 6. Range Gate Control Word

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0xB0								RG enable
Type								WO
=0								No gates/end gates at beam
=1								Start gates at beam

Table 7. Transmitter Controller

Hex Address	Function	Abv	Bits	Format	Range
C0|C1	Transmitter Control Word*	TXCN			
C2|C3	Trigger B to Trigger A spacing	BASP	6	2's comp	1 to 63 μsec
C4|C5	Trigger B to RT spacing	B0SP	6	2's comp	2 to 63 μsec
C6|C7	Radar trigger length	RTLN	6	2's comp	1 to 64 ticks
C8|C9					
CA|CB					
CC|CD					
CE|CF					

*not implemented

Table 8. Transmitter Control Word

Hex Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0xC0								
Type								
=0								
=1								

Table 9. Polarization Controller

Hex Address	Function	Abv	Bits	Range
E0|E1	Polarization Control Word*	PLCON		
E2|E3	Transmit Polarization Mode	XPOL	8	
E4|E5	Receive Polarization Mode	RPOL	8	
E6|E7				
E8|E9				
EA|EB				
EC|ED				
EE|EF				

*not implemented

H or h polarization is 0.
V or v polarization is 1.
MSB is the first polarization; LSB is the last.

6. RADAR TIMING GENERATOR CONNECTORS

The following general comments apply to all connectors, except J7 and J8.

1) All connectors are standard ribbon cable style on a 0.1″ grid.
2) Pin 1 and all even-numbered pins are grounded. Even-numbered pins are not shown.
3) All connectors are a different size.
4) All levels are standard TTL levels, unless noted.

Pin	Signal	I/O	Comment
1	GND	P	
3	RG_CLK	O	Clock at range gate rate
5	RG0	O	Range gate zero--precedes first range gate
7	BEAM	O	Start of Beam
9			
11	RUN	O	Start of Run
13	Spare	X	
15	Spare	X	
17	Spare	X	
19	Spare	X	

Jack 1. VME Chassis Connector (20 pin)

Pin	Signal	I/O	Comment
1	GND	P	
3	RT	O	Radar Trigger
5	PRTA	O	Immediately precedes RT
7	PRTB	O	Precedes PRTA
9	RT1	O	Radar Trigger 1
11	PRT1A	O	Immediately precedes RT1
13	PRT1B	O	Precedes PRT1A
15	RT2	O	Radar Trigger 2
17	PRT2A	O	Immediately precedes RT2
19	PRT2B	O	Precedes PRT2A
21	XMIT_POL	O	Transmitted polarization
23	RCV_POL	O	Received polarization
25	Spare	X	Reserved for other options
27	Spare	X	Reserved for other options
29	Spare	X	Reserved for other options
31	Spare	X	Reserved for other options
33	Spare	X	Reserved for other options

Jack 2. Receiver/Transmitter Connector (34 pin)

Pin	Signal	I/O	Comment
1	GND	P	
3	RT1	O	A-scope trigger
5	RG	O	Range gates for A-scope
7	RT1	O	A-scope trigger
9	RG	O	Range gates for A-scope
11	RT1	O	A-scope trigger
13	RG	O	Range gates
15	RT	O	Radar trigger
17	RG_CLK	O	Clock at fastest range gate rate (20 MHz or ext clk)
19	RG0	O	Range gate zero--precedes first range gate
21	BEAM	O	Start of Beam
23			
25	RUN	O	Start of run
27	XMIT_POL	O	Transmitted polarization
29	RCV_POL	O	Received polarization
31	Spare	X	
33	Spare	X	
35	Spare	X	
37	Spare	X	
39	arb_clk	O	Triggers ARB. Gives 2 pulses/RT that ARB needs.

Jack 4. Display Connector (40 pin)

Pin	Signal	I/O	Comment
1	GND	P	
3	P^3_CLK	I	P^3 Clock (16 MHz)
5	P^3_RT	I	P^3 radar trigger
7	P^3_PR	I	P^3 processor reset
9	Spare	X	

Jack 5. Slave Connector (10 pin)

Pin	Signal	I/O	Comment
24	GND	P	
26	RGCLK	O	20 MHz or external clock frequency

Jack 7. ICS150 A/D Connector (34 pin)

Pin	Signal	I/O	Comment
26	RG0	O	Precedes first range gate
28	GND	P	

Jack 8. ICS150 A/D Connector (34 pin)

Appendix C.

UF Header Use in RADS

Additional parameters are required in RADS (Radar Acquisition and Display System) beyond what is presently provided for in the UF header. The purpose of this document is to precisely document the UF header *as used for Raw Data only*, and to indicate the use of the additional parameters required.

The UF format was originally defined for fields of data produced from a radar, such as reflectivity, power, velocity, etc. For raw data, the definition is stretched a little by calling all the data associated with a range gate a "field". Some parameters, such as the noise power or the radar constant, used in calculating certain fields is subject to change and it is sometimes desirable to be able to recalculate the fields with different values of these parameters. Also, calculating and recording many different fields may reduce the maximum number of range gates possible.

The RADS RCP is driven by the Radar Parameters, which consists of 1000 bytes of information. Many parameters are in floating-point format, which is incompatible with the UF header that is built around 16-bit integers. The local use header, however, may contain any type of information and so the Radar Parameters, plus an additional 16-bit parameter placed in front of the Radar Parameters, are appended to the end of our present 60-word local use header. The purpose of the additional 16-bit parameter, which is the length of the Radar Parameters in bytes, is to align the Radar Parameters on a 32-bit boundary.

Floating-point data follow the parameter block, and its position is indicated by the appropriate word in the header. Because floating-point data are not supported by the UF format, we describe this format as an Extended UF format and place an "EF" at the beginning of the UF parameters, rather than a "UF." This means that in order to be compatible with Perusal, data must be run through a "ef2uf" conversion program designed for RADS.

The following shows the UF-like header and format as it will be used on "raw" RADS tapes.

Table 1. RADS "Raw" Tape Format (EF Format)

Name	**C structure**	**Length in 16-bit words**	**Position of first word**
Mandatory_header	m_hdr	45	1
Optional_header	o_hdr	14	46
Wp6_local_header	l_hdr	61 + 500	60
Data_header	d_hdr	5	397
Field_header	f_hdr	25	402
Data	varies	varies	427

This gives a header length of 650 16-bit words. Programs should not rely on these lengths, however, but should instead use C header files and sizeof( ) operators. This will allow header sizes to change (in particular the local header or Radar Parameters) with a minimum amount of impact on the code.

Table 2. Mandatory Header

Word index	Word index in UF	Contents	C Variable Name	Data type	Units	Derived from
1	1	EF	packet_type[2]	char[2]	2 ASCII	"EF"
2	2	Record length	record_length	u_short	16-bit words	format
3	3	Position of optional header	optional_hdr	u_short	16-bit words	format
4	4	Position of local header	local_h	u_short	16-bit words	format
5	5	Position of data header	data_hdr	u_short	16-bit words	format
6	6	Physical record number relative to beginning of file	physical_record_no	u_short	none	record
7	7	Volume scan number within tape	scan_volume_count	u_short	none	VLNM
8	8	Ray number within volume scan	ray_number	u_short	none	record
9	9	Physical record number within ray	physical_w_in_ray	u_short	none	1
10	10	Sweep number within this volume scan	sweep_count	u_short	none	scan
11-14	11-14	Radar Name	char_radar_[8]	char[8]	8 ASCII	RDNM
15-18	15-18	Site Name	char_site_[8]	char[8]	8 ASCII	STNM
19	19	Latitude, degrees	latitude_degrees	short	degrees	LAT
20	20	Latitude, minutes	latitude_minutes	short	minutes	LAT
21	21	Latitude, seconds	latitude_seconds	short	seconds * 64	LAT
22	22	Longitude, degrees	longitude_degrees	short	degrees	LONG
23	23	Longitude, minutes	longitude_minutes	short	minutes	LONG

Word index	Word index in UF	Contents	C Variable Name	Data type	Units	Derived from
24	24	Longitude, seconds	longitude_seconds	short	seconds * 64	LONG
25	25	Elevation above sea level	altitude_above_sealevel	short	meters	ELSL
26	26	Year	Year	u_short	years (00-99)	TIME, TMOF
27	27	Month	Month	u_short	month	TIME, TMOF
28	28	Day	Day	u_short	day of month	TIME, TMOF
29	29	Hour	Hour	u_short	hours	TIME, TMOF
30	30	Minute	Minute	u_short	minutes	TIME, TMOF
31	31	Second	Second	u_short	seconds	TIME, TMOF
32	32	Time zone	time_zone[2]	char[2]	2 ASCII	TMZN
33	33	Azimuth angle	Azimuth	short	degrees * 64	AZIM
34	34	Elevation angle	Elevation	short	degrees * 64	ELEV
35	35	Sweep mode	sweep_mode	u_short	none	SWPM
36	36	Fixed angle	fixed_angle	short	degrees * 64	ANGF
37	37	Sweep rate	sweep_rate	u_short	deg/sec * 64	scan parameters
38	38	Generation date	generation_date_year	u_short	years (00-99)	raw-to-UF
39	39	Generation date	generation_date_month	u_short	month	raw-to-UF
40	40	Generation date	generation_date_day	u_short	day of month	raw-to-UF

Word index	Word index in UF	Contents	C Variable Name	Data type	Units	Derived from
41-44	41-44	Tape generator facility name	word_41 - word_44	u_short	8 ASCII	raw-to-UF
45	45	Deleted or missing data flag	Missing	u_short	none	raw-to-UF

Table 3. Optional Header

Word index	Word index in UF	Contents	C Variable Name	Data type	Units	Derived from
1-4	46-49	Project name	project_name[8]	char	8 ASCII	PRNM
5	50	Baseline azimuth	baseline_az	short	deg * 64	BLAZ
6	51	Baseline elevation	baseline_el	short	deg * 64	BLEL
7	52	Start of current volume scan	start_hour	u_short	Hours	TIME, TMOF
8	53	Start of current volume scan	start_minute	u_short	Minutes	TIME, TMOF
9	54	Start of current volume scan	start_second	u_short	Seconds	TIME, TMOF
10-13	55-58	Field tape name	field_tape_name[8]	char	8 ASCII	FTNM
14	59	Range gate flag	Flag	u_short	None	raw-to-UF

Table 4. WP6 Local Use Header

Word index	Word index in UF	Contents	C Variable Name	Data type	Units	Derived from	Comment
1-4	60-63	Operator name	operator[8]	char[8]	8 ASCII	OPNM	
5	64	Radar x-coordinate w.r.t. origin	X	short	decameters	met params	future
6	65	Radar y-coordinate w.r.t. origin	Y	short	decameters	met params	future
7	66	Radar z-coordinate w.r.t. origin	Z	short	decameters	met params	future
8	67	Tape number	tape_no	u_short	None	TNUM	
9	68	Transmitter flag	trans_type	short	1 = magnetron -1 = klystron	TFLG	
10	69	Bandwidth of log receivers	log_bw	u_short	MHz	rgbw[0]	
11	70	Bandwidth of linear receivers	lin_bw	u_short	MHz	rnbw[0]	
12	71	RCP version #	rcp_ver	u_short	None	RCPN	
13-16	72-75	Scan name	scan_name[8]	char[8]	8 ASCII	met params	future
17	76	Scan center x-coordinate w.r.t. origin	scan_center_x	short	decameters	met params	future
18	77	Scan center y-coordinate w.r.t. origin	scan_center_y	short	decameters	met params	future
19	78	Scan radius w.r.t. origin	scan_radius	u_short	decameters	met params	future
20	79	Scan elevation, min, w.r.t. origin	Zmin	short	decameters	met params	future

Word index	Word index in UF	Contents	C Variable Name	Data type	Units	Derived from	Comment
21	80	Scan elevation, max, w.r.t. origin	Zmax	short	decameters	met params	future
22	81	Total volume scan time	total_vtime	u_short	seconds * 10	scannin g params	
23-25	82-84	Resolution: horizontal, vertical, time	resolution[3]	u_short	decameters decameters seconds*10	met params	future
26	85	Number of sweeps	num_sweeps	u_short	None	scannin g params	
27	86	Sweep time	sweep_time	u_short	Seconds * 10	SWTM	
28	87	Azimuth boundary, CCW	az1	u_short	degrees * 64	MNAZ	
29	88	Azimuth boundary, CW	az2	u_short	degrees * 64	MXAZ	
30	89	Elevation or coplane boundary, lower	el1	u_short	degrees * 64	MNEL	
31	90	Elevation or coplane boundary, upper	el2	u_short	degrees * 64	MXEL	
32	91	Minimum R foreground	min_r_fg	u_short	decameters	met params	future
33	92	Maximum R foreground	max_r_fg	u_short	decameters	met params	future
34	93	Mininum R background	min_r_bg	u_short	decameters	met params	future
35	94	Maximum R background	max_r_bg	u_short	decameters	met params	future

Word index	Word index in UF	Contents	C Variable Name	Data type	Units	Derived from	Comment
36-41	95-100	Elevation table for non-uniform elevation increments	el_table[6]	u_short	degrees * 64	met params	future
42	101	Number of radar triggers in sample	Ntrig	u_short	None	NTRG	
43	102	Long period, spacing between pairs	Lpd	u_short	µs * 4	PRPR	
44	103	Short period, spacing inside a pair	Spd	u_short	µs * 4	TRGS	
45	104	Number of triggers to wait between beams	Ntwt	u_short	None	NPWT	
46	105	Delay from trigger to first range gate	Dlay	u_short	µs * 4	DLAY	
47	106	Spacing between range gates in first group	Spac	u_short	µs * 4	SPAC	
48	107	Number of range gates in first group	nrg1	u_short	None	NRGT	
49	108	Spacing between range gates in second group	spc2	u_short	µs * 4	0	
50	109	Number of range gates in second group	nrg2	u_short	None	0	
51	110	Spacing between range gates in third group	spc3	u_short	µs * 4	0	
52	111	Number of range gates in all three groups	n_gates	u_short	None	NRGT	
53	112	Data acquisition mode or beam type	Mode	u_short	None	DMOD	
54	113	Polarization received	pol_received	u_short	None	RPL1, RPL2	
55	114	Nyquist velocity	Nyquist	u_short	m/s * 128	met params	future

Word index	Word index in UF	Contents	C Variable Name	Data type	Units	Derived from	Comment
56	115	Maximum unambigous range	max_unamb_range	u_short	km * 10	met params	future
57	116	Beam number of first beam of sweep	beam_no_of_1st_bm_of_sweep	u_short	None	record	
58	117	Number of transmit frequencies	Nfreq	u_short	None	NTFR	
59	118	Transmit frequency code	Xfreq	u_short	see following table	TFMD, TFST, NTFR	
60	119	Receiver pad code	rec_pad	u_short	see following table	RPDH, RPDV	
61	120	Length of Radar Parameters in bytes	rp_length_bytes	u_short	bytes	sizeof (Rp)	added for 32-bit alignment
62 - 338	121-396	Radar Parameters		structure			See Radar Parameters document

Table 5. Transmit Frequency Code -- Word 59 of Local Use Header

Bit	**Function**
0 (LSB)	=indicates uniform or Goloumb spacing mode
5-1	= transmitter frequency spacing in MHz for delta-k uniform spacing mode
10-6	= number of transmitter frequencies for delta-k mode (spacing as per bits 5-1 for uniform spacing) frequency spacing for Goloumb spacing mode: # freq spacing (units specified in bits 5-1) 1 0 2 1 3 2,1 4 2,3,1 5 2,5,3,1 6 2,5,6,3,1 7 2,5,8,6,3,1 8 7,3,6,2,12,1,4
15-11	Reserved

Table 6.Receiver Pad Code -- Word 60 of Local Use Header

Bit	**Function**
5-0 (LSB)	= horizontal linear receiver pad in dB
11-6	= vertical linear receiver pad in dB
15-12	= reserved

Table 7. Data Header -- Applies to all data fields

Word index	Word index in UF	Contents	C Variable Name	Data type	Units	Derived from	Comment
1	397	Total number of fields this ray	total_fields_this_ray	u_short	none, =0 if not field format	1	
2	398	Total number of records this ray	total_records_this_ray	u_short	none	1	
3	399	Total number of fields this record	total_fields_this_record	u_short	none	1	
4	400	Raw Data field name, “RD”	field_name[2]	char[2]	2 ASCII	DMOD	always “RD”
5	401	Position of 1st word of 1st field header	Fh	u_short	none	format	next word

Table 8. Field Header for Raw Data -- field headers precede each type of field data

Word index	Word index in UF	Contents	C Variable Name	Data type	Units	Derived from	Comment
1	402	Position of first data word	data_position	u_short	16-bit words	format	
2	403	Scale factor (met units = tape value / scale factor)	scale_factor	u_short	varies	SCLF	0 for floating point
3	404	Range to first gate	range_to_first_gate	u_short	kilometers	DLAY, pknl[0]	
4	405	Adjustment to center of first gate	adjust_to_center	u_short	meters	DLAY, pknl[0]	
5	406	Sample volume spacing	sample_volume_spacing	u_short	meters	SPAC	
6	407	Number of sample volumes	num_sample_volumes	u_short	none	NRGT	
7	408	Sample volume depth	sample_volume_depth	u_short	meters	SPAC* NRGT	
8	409	Horizontal beam width	hor_bm_width	u_short	deg * 64	azbw	
9	410	Vertical beam width	ver_bm_width	u_short	deg * 64	elbw	
10	411	Receiver bandwidth	receiver_bandwidth	u_short	MHz	rnbw[0]	
11	412	Transmit/receive polarization code	polarization_transmitted	u_short	see next table	TPL1, TPL2	
12	413	Wavelength	Wavelength	u_short	cm * 64	TFRQ	
13	414	Number of samples used in field estimate	num_samples_in_field_estimate	u_short	none	NTRG	
14	415	Threshold field	threshold_field[2]	2 char	2 ASCII	raw-to-UF	
15	416	Threshold value	threshold_value	u_short	varies	raw-to-UF	
16	417	Scale (used for scaling dB's, following)	Scale	u_short	none	64	
17	418	Edit code	edit_code	u_short	2 ASCII	raw-to-UF	

Word index	Word index in UF	Contents	C Variable Name	Data type	Units	Derived from	Comment
18	419	Pulse repetition period	Prpr	u_short	microseconds	TRGS	
19	420	Bits per sample volume	bits_per_sample_vol	u_short	none	BPDS	32-bit floating point
20	421	Radar constant dBZ = (RC_A + data)/Scale + 20 log(range in km) or dB = (RC_O + data)/Scale + 30 log(range in km)	radar_constant	short	dB	kRC[0]	
21	422	Noise power	noise_power	short	dBm * scale	R0h[0] or R0v[0], RPL1, RPL2, rnbw[0]	
22	423	Receiver gain	receiver_gain	short	dB * scale	rnh[ ] or rnv[0], RPL1, RPL2	
23	424	Peak power	peak_power	short	dBm * scale	TPWR	
24	425	Antenna gain	antenna_gain	short	dB * scale	antg	
25	426	Pulse duration	pulse_duration	u_short	microseconds * 64	RTLN	

Table 9. Transmit/Receive Polarization Code -- Word 11 of Field Header (not relevant for differential phase modes)

Bit	**Function**
0 (LSB)	Reserved
1	Reserved
3,2	00 = horizontal transmit polarization 01 = vertical transmit polarization 1x = pulse-to-pulse transmit polarization switching trigger 1 always vertical, trigger 2 always horizontal.
4	0 = horizontal linear receiver polarization 1 = vertical linear receiver polarization
6,5	00 = error 01 = .05us xmit pulse, 20 MHz lin BW, 20 MHz log BW 10 = .25us xmit pulse, 4 MHz lin BW, 4 MHz log BW 11 = 1.0us xmit pulse, 1 MHz lin BW, 4 MHz log BW
8,7	x0 = single transmit frequency 01 = delta-k mode, uniform frequency spacing 11 = delta-k mode, Goloumb frequency sequence
15-8	Reserved

Table 10. Data -- Raw Data in all modes are 32-bit floating-point. Complex data are in **bold**. Quantities shown are repeated for each range gate

Mode name	**Pulse Pair**	**Diff Phase, equally spaced**	**Diff Phase, high speed**	**Diff Phase, unequally spaced**	**Diff Phase, all fields**
Mode number	140	150	151	158	159
Quantities recorded for each range gate	A, B, R, Lv, Lh	**RsHH1, RsVV1,** RlHH0, RlVV0, **RsHV1, RsVH1,** GIHh, GIHv, GIVh, GIVv	**RsHV1, RsVH1,** GIHh, GIHv, GIVh, GIVv	**RsHH1, RsVV1,** RlHH0, RlVV0, **RlHV2, RlVH2, RlHH4, RlVV4,** GIHh, GIHv, GIVh, GIVv	**RsHH1, RsVV1,** RlHH0, RlVV0, **RsHV1, RsVH1, RlHV2, RlVH2, RlHH4, RlVV4,** GIHh, GIHv, GIVh, GIVv
bytes/gate	20	56	32	72	88

Appendix D.

VME Address Spaces for RADS

Table 1. A16 Space (short I/O)

Hex Address	Length	Function	Slave Data Type	Master Data Type
FF18-FFFF	.23kB	Unused		
FF00-FF17	.02kB	CCC S/D board	D16	None
F040-FEFF	3.69kB	Unused		
F000-F03F	.06kB	Datum bc637 GPS	D16	None
E000-EFFF	4kB	Unused		
D000-DFFF	4kB	Galil motion control	D16	None
AC00-CFFF	9kB	Unused		
A800-ABFF	1kB	Future radar control (IP carrier)	D16	None
A400-A7FF	1kB	Future radar control (IP carrier)	D16	None
A000-A3FF	1kB	Radar control (IP carrier)	D16	None
9800-9FFF	2kB	Unused		
9000-97FF	2kB	IP carrier for airborne K-band	D16	None
8800-8FFF	2kB	Unused		
8000-87FF	2kB	Mizar DSP	D32, D16, D8	None
6C00-7FFF	5kB	Unused		
6800-6BFF	1kB	IP carrier for phase-rotating plate	D16	None
6400-67FF	1kB	RTG	D08(O)	None
6000-63FF	1kB	IP carrier for navigation	D16	None
5800-5FFF	2kB	Unused		
5400-57FF	1kB	GRIDS RTG	D08(O)	None
0000-53FF	21kB	Unused		

Table 2. A32 Space

Hex Address	Length	Function	Slave Data Type	Master Data Type
B020.0000-FFFF.FFFF	1,308,672 KB	Unused		
B000.0000-B01F.FFFF	2048 KB	Mizar DSP global memory	D32, D16, D8 D32, D16 BLT D64 MBLT	D32, D16, D8 D32, D16 BLT D64 MBLT
8300.0000-AFFF.FFFF	737,280 KB	Unused		
8000.0000-82FF.FFFF	49,152 KB	Ixthos DSP		
1008.0000-7FFF.FFFF	1,834,496 KB	Unused		
1000.0000-1007.FFFF	512 KB	ICS-152 memory	D32	D32 BLT D64, D32 MBLT
0808.0000-0FFF.FFFF	130,560 KB	Unused		
0800.0000-0807.FFFF	512 KB	ICS-150 memory	D32	D32 BLT D64, D32 MBLT
0010.0000-07FF.FFFF	130,048 KB	Unused		
0000.0000-000F.FFFF	1024 KB	SPARC’s slave memory	D32, D16, D08(EO)	D64, D32, D16, D08(EO)

VME interrupt level 4 is reserved for communication between the ICS-150 A/D and the DSP and should be disabled on the SPARC in the OpenBoot PROM.

Appendix E.

Examples of graphics output

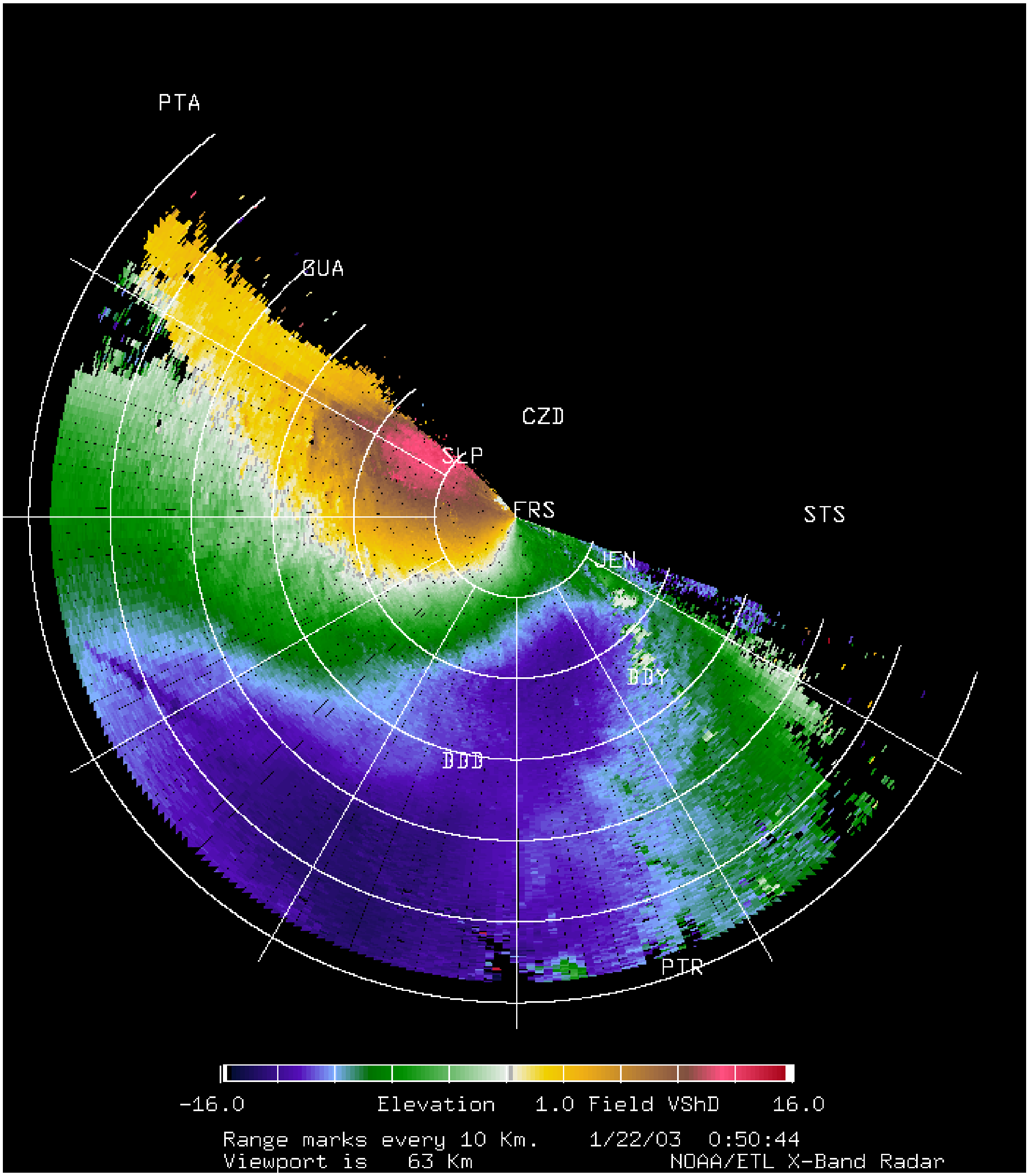

Figure 1. PPI scan display window. This shows a velocity field (VShD) where the radar is transmitting a “slant” polarization, receiving horizontal polarization, and the result has been DC corrected. The velocity is color coded for -16 to 16 m/sec per the color bar.

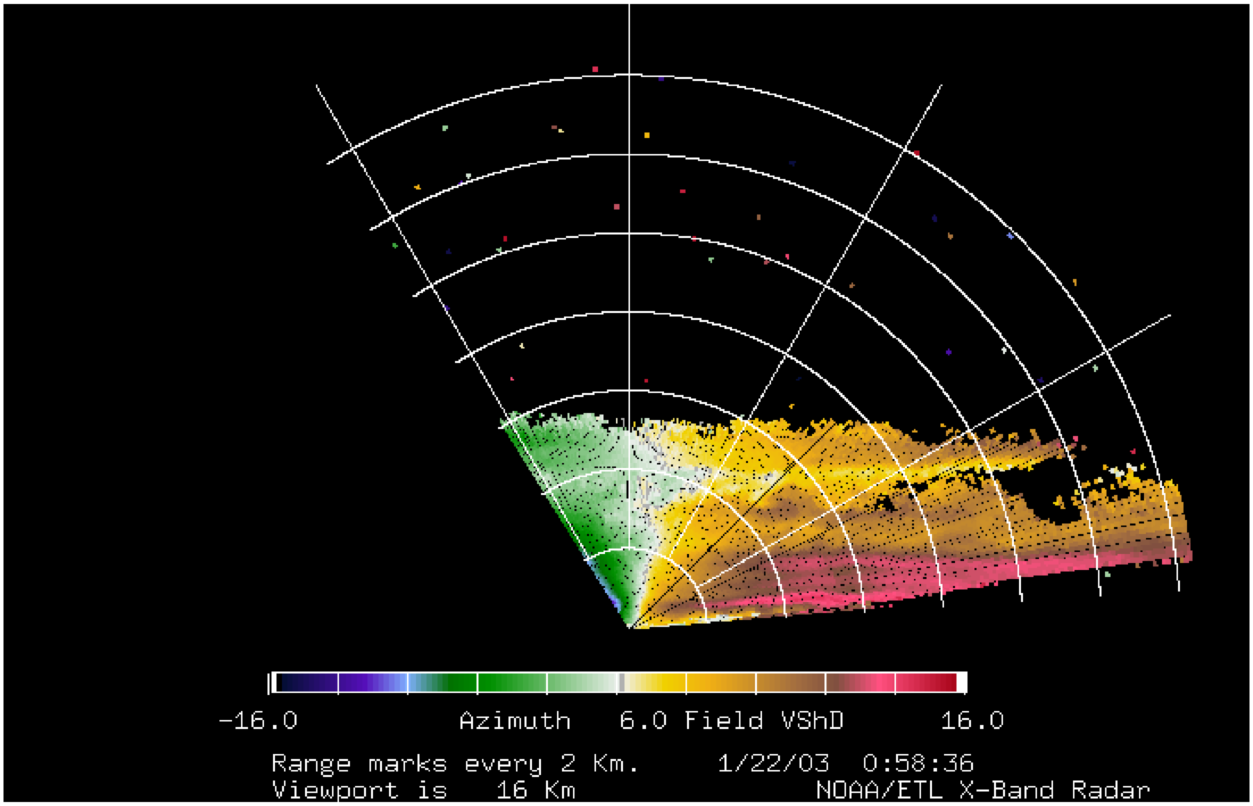

Figure 2. RHI scan display window. This image shows a velocity field (VShD) where the radar is transmitting a "slant" polarization, receiving horizontal polarization, and the result has been DC corrected. The velocity is color coded for -16 to 16 m/sec per the color bar. The radar is pointed at an azimuth of 6.0 degrees and is scanning "over the top" from approximately 5 degrees to 122 degrees.

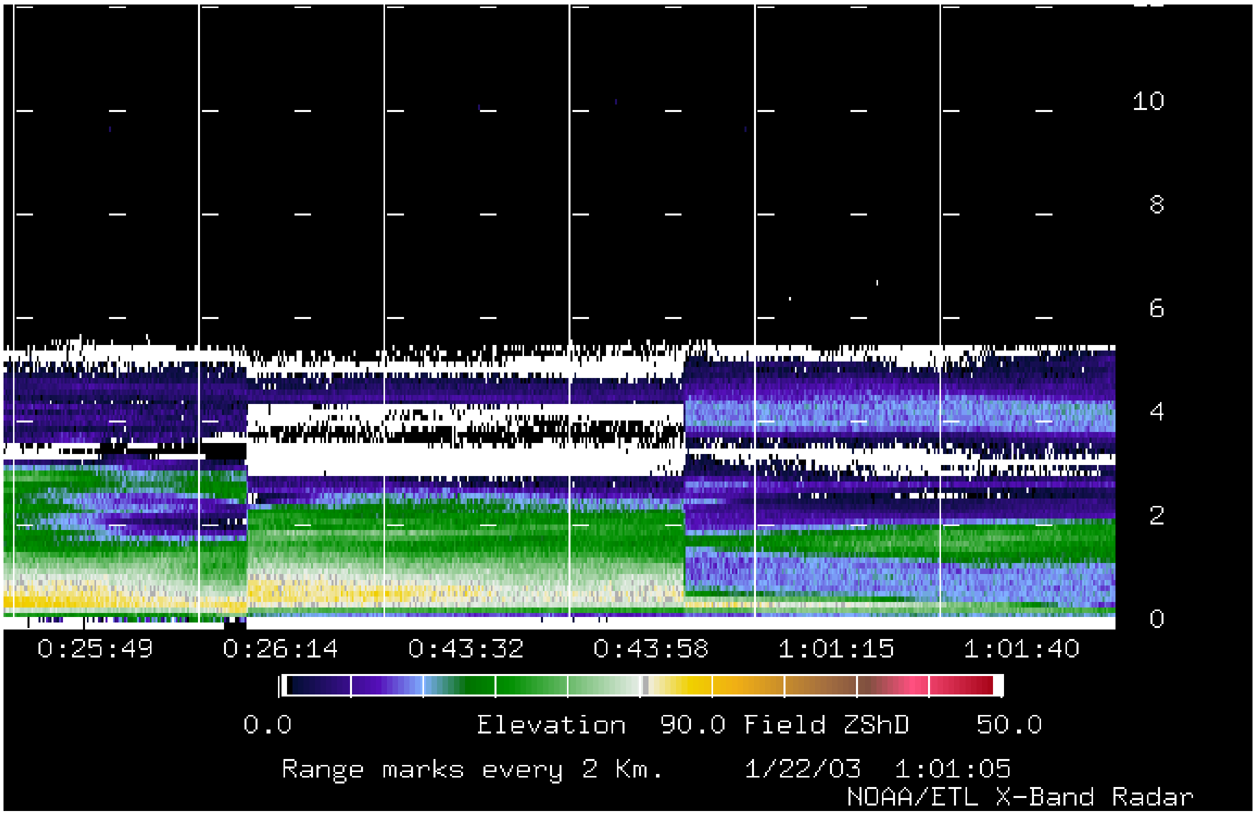

Figure 3. Time-range (B-scan) display window for fixed-beam data. This shows a reflectivity field (ZShD) where the radar is transmitting a "slant" polarization, receiving horizontal polarization, and the result has been DC corrected. The reflectivity is color coded for 0. to 50. DBZ per the color bar. The radar is looking directly overhead at 90 degrees out to a range of about 5.75 km.

VAD Scan Profiles Window

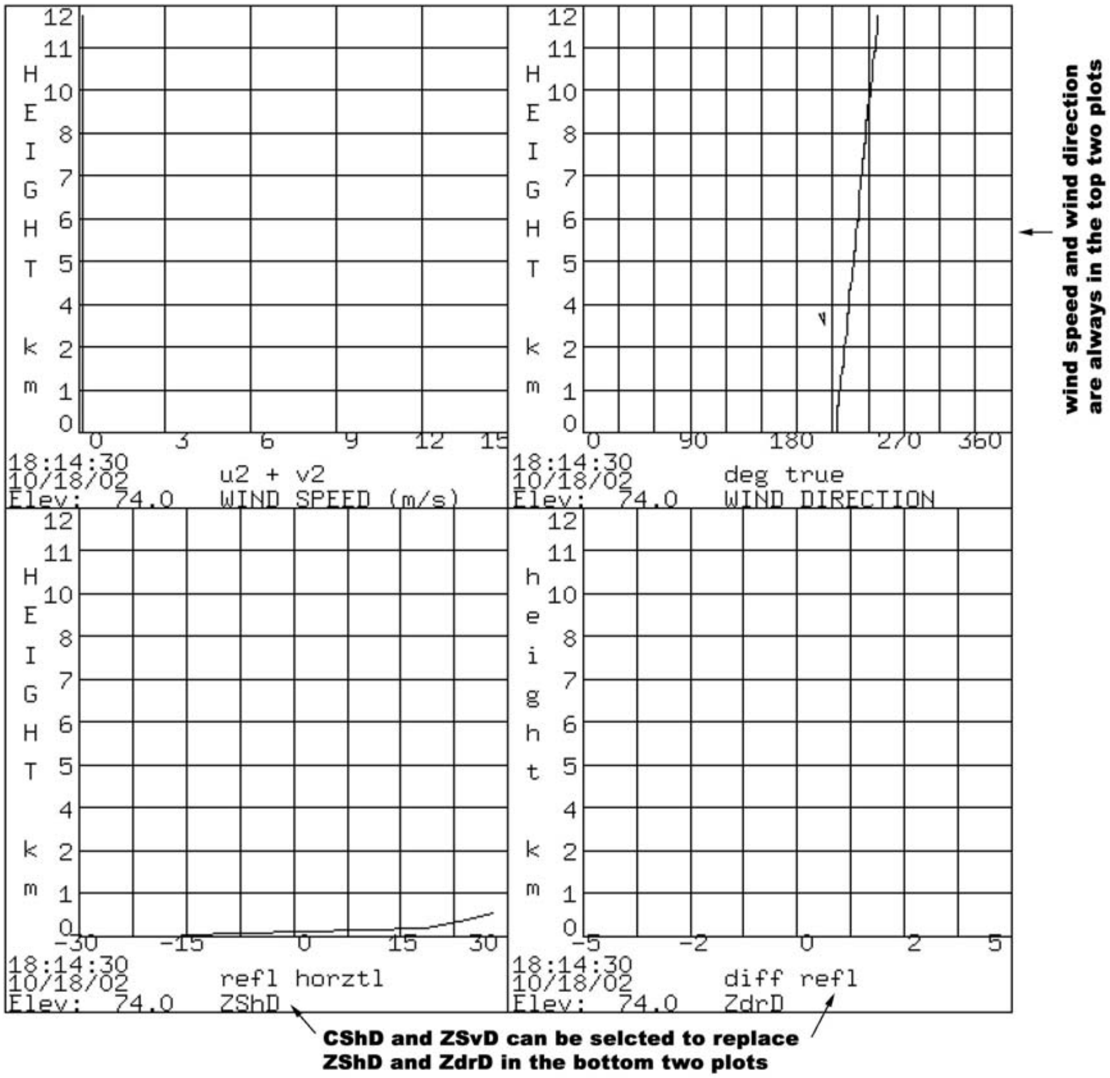

Figure 4. VAD scan profiles window. This shows the format for VAD profile plots. Wind speed, wind direction, horizontal reflectivity (ZShD) and differential reflectivity (ZdrD) are displayed in the four windows. All four plots have a y axis of 0. to almost 12 km. In this example, however, very little data was available for display purposes.

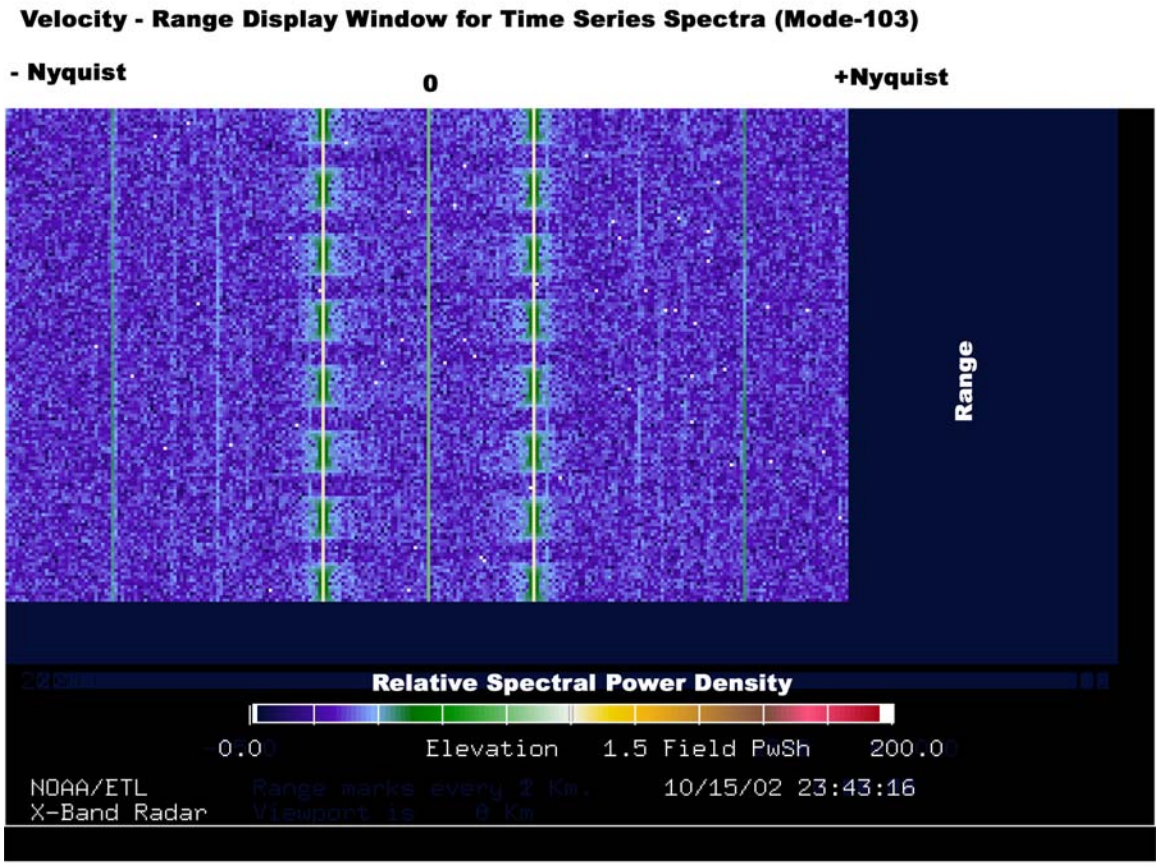

Figure 5. Velocity-range display window for time series spectra (mode 103). This image shows relative spectral polwer density (PwSh) where the radar is transmitting a "slant" polarization, receiving horizontal polarization, and the result has been DC corrected. The reflectivity is color coded from 0. to 200. per the color bar. The radar is pointing in a fixed location at an elevation of 1.5 degrees.

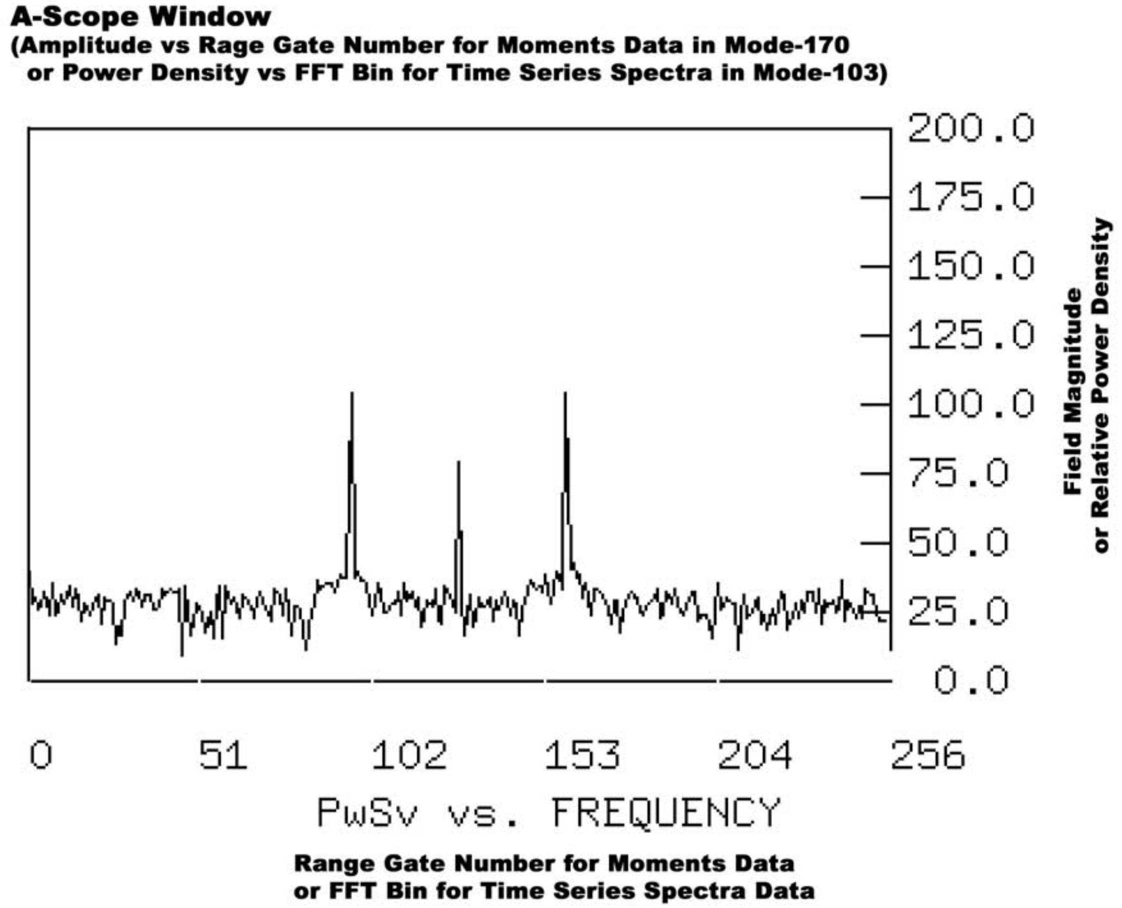

Figure 6. A-scope window. This shows power density in the vertical channel (PwSv) vs. FFT frequency bin for a chosen range for time series spectra data mode 103.

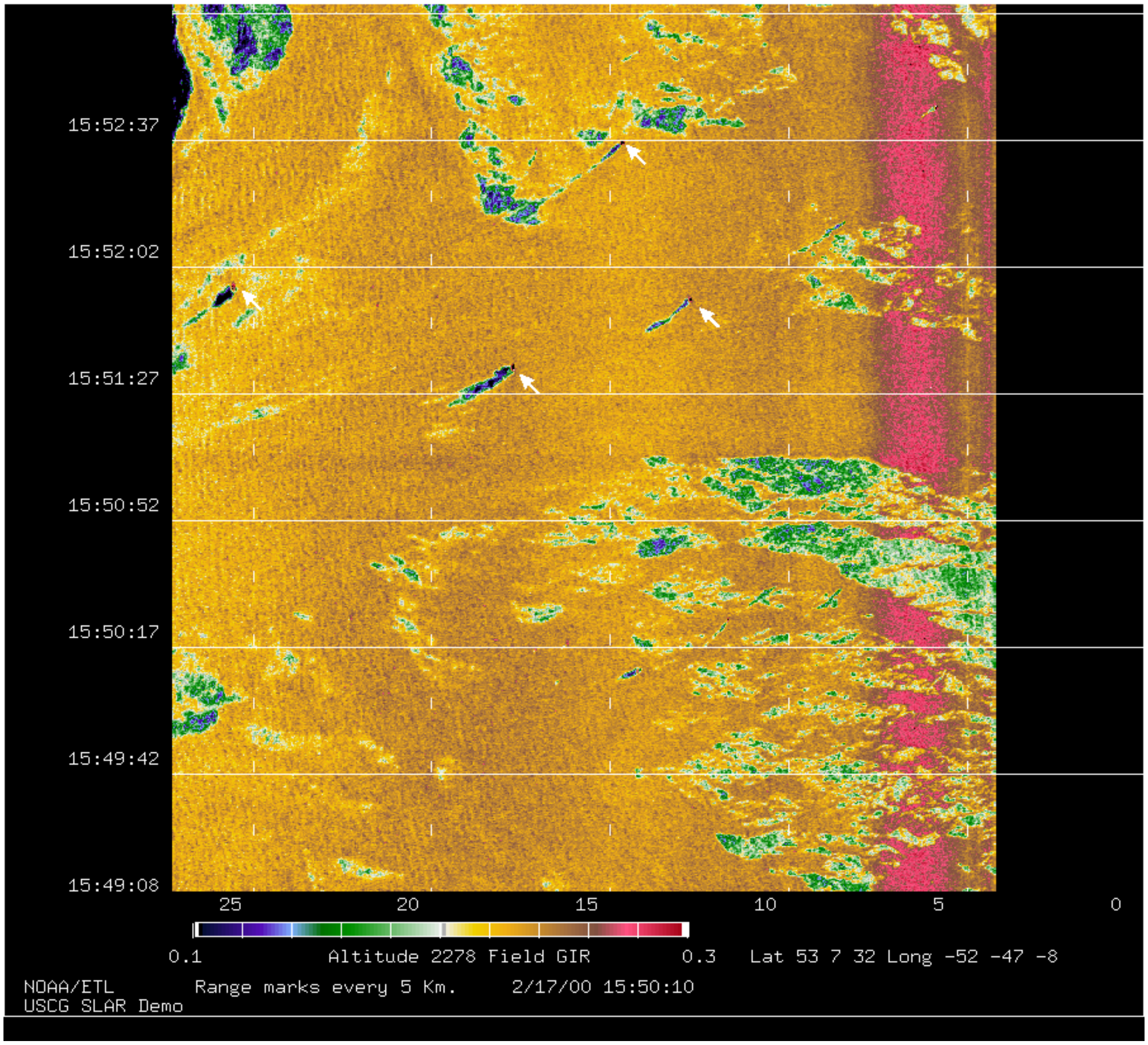

Figure 7. SLAR image of icebergs. This shows an intensity field (GIR) from the log channel of the left looking radar on the USCG RADS system. The intensity is color coded from .1 to .3 per the color bar. The C-130 is flying at an altitude of 2278 meters in the North Atlantic. The arrows are pointing to icebergs and are not part of the original RADS image as displayed at the radar.

Appendix F.

Examples of GUIs

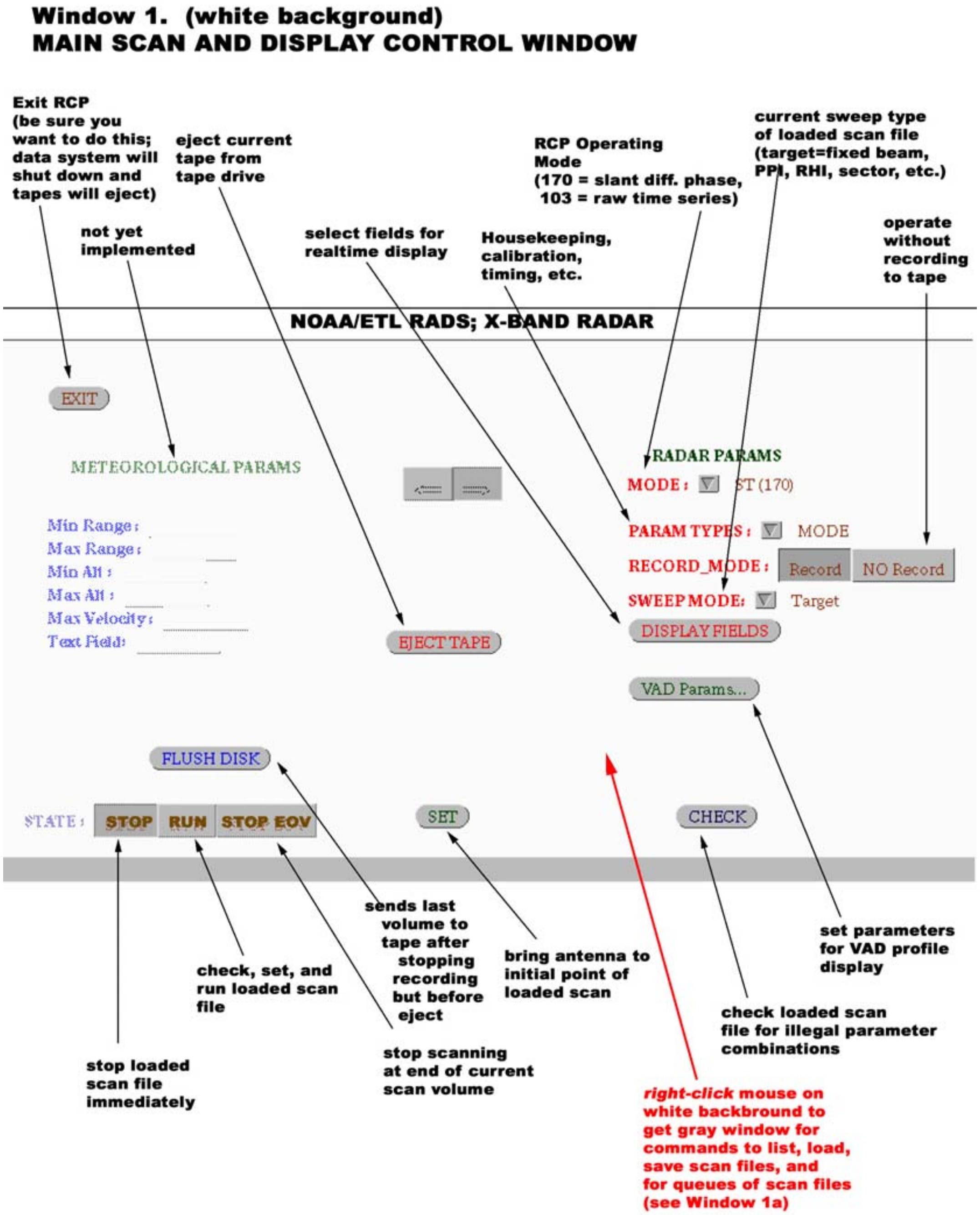

Figure 1. RADS Control GUI or base window; it appears when RADS is initialized. Among other things, it allows users to start and stop a scan, start a queue, change radar parameters and build and view scan tables and queues.

Figure 2. Typical RADS status window, owned by the base window, displays whether a continuous scan is running or a queue, which scan table it is running and when applicable, which queue. It also displays whether RADS is in record mode or not and the tape number.

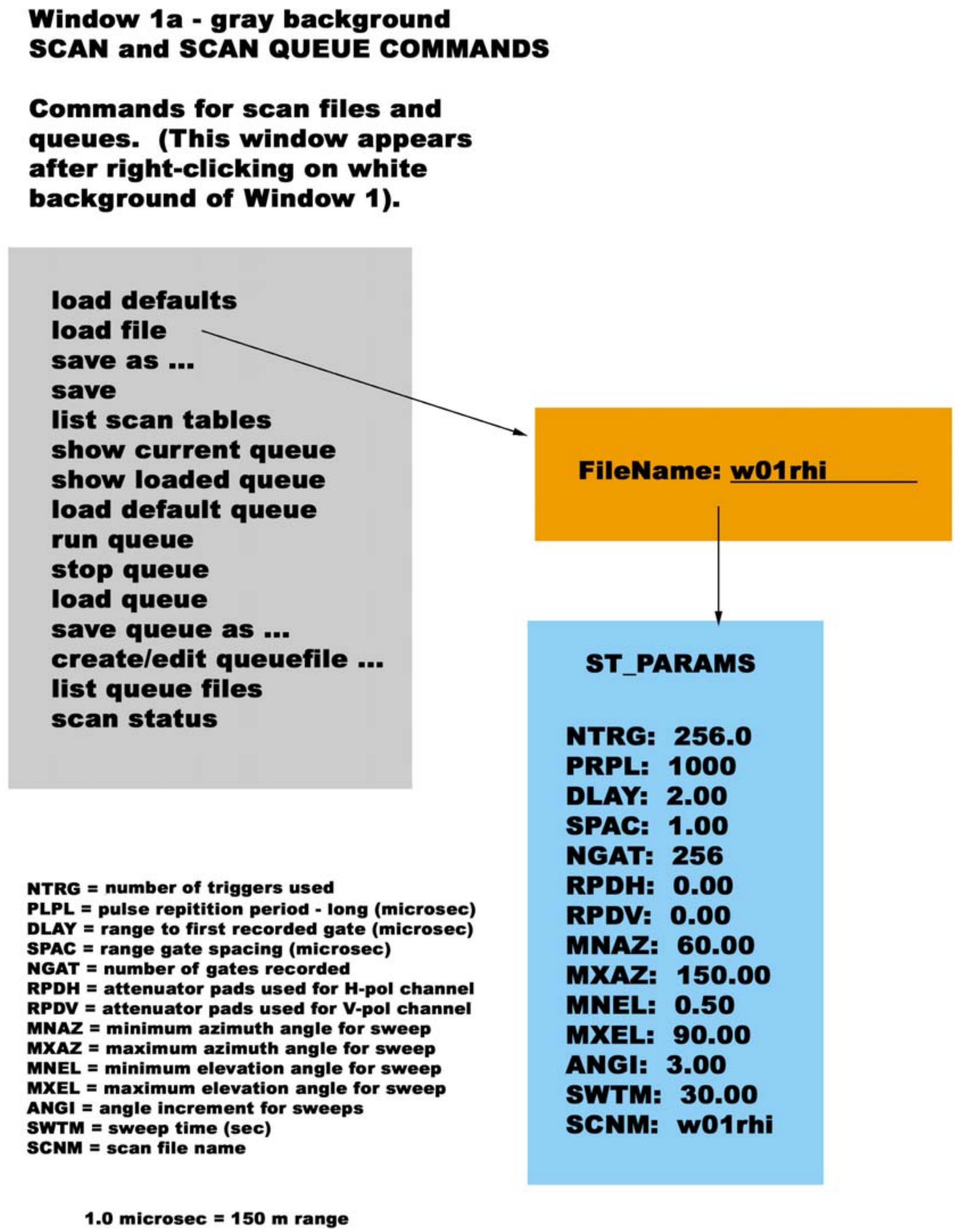

Figure 3. Pulldown menu for loading, editing and viewing scan tables and scan queues. The popup window on the top right appears when the user selects “load file” and queries the user for a scan table name. Once a name is input, the popup window on the bottom right appears displaying the most commonly changed operator enabled radar parameters. This window is mode dependent.

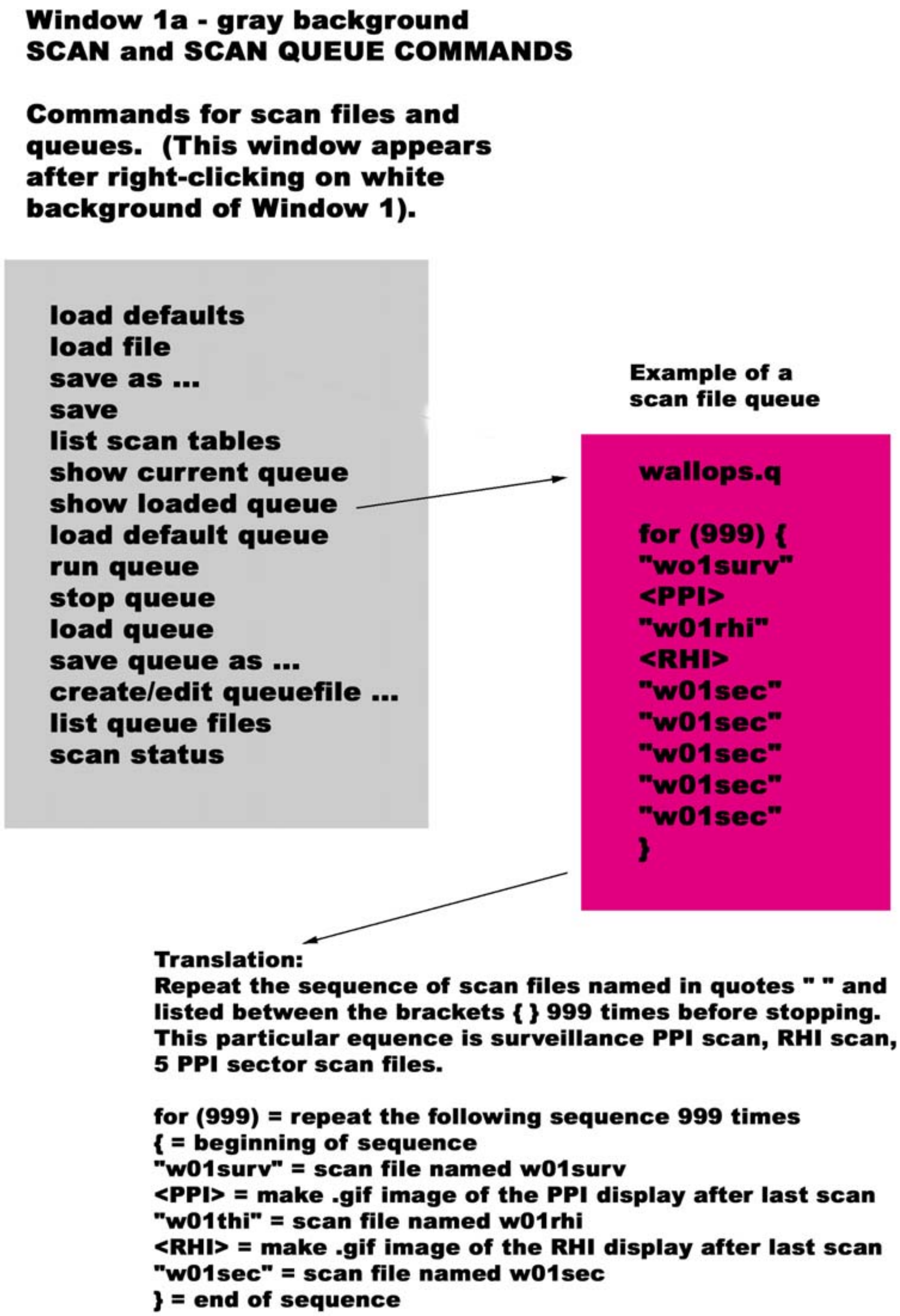

Figure 4. RADS Pulldown menu for loading, editing and viewing scan tables and scan queues. The popup window on the right shows the loaded queue.

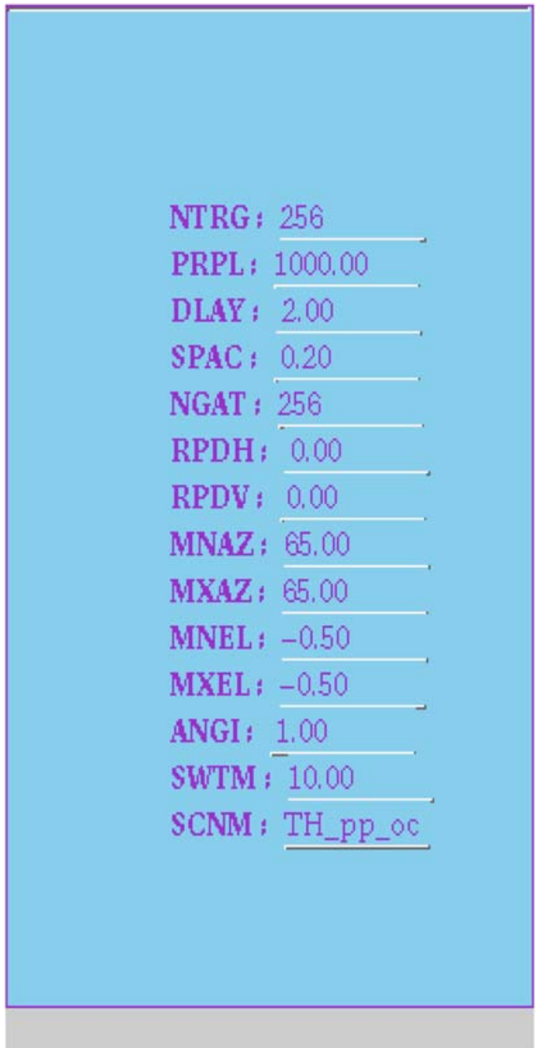

Figure 5. Typical RADS Mode Parameter Window, displaying the most common radar parameters which are operator enabled. Since these are mode dependent they will vary.

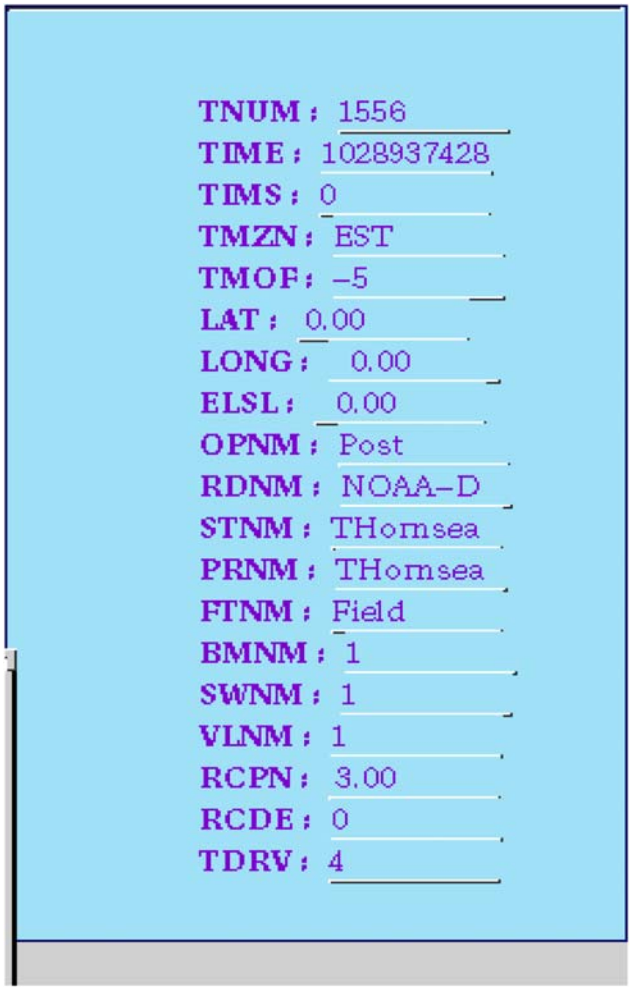

Figure 6. Typical RADS Housekeeping Parameter Window. This window displays the housekeeping parameters corresponding to the loaded scan table. This is one of the 12 popup parameter type windows which can be displayed by selecting the "Param Type" button on the base window and the desired type of parameter from the drop down menu.

Figure 7. Typical Display Fields Window allows the user to select which fields will be sent to the graphics program x-display for viewing. The selections available in this window are mode dependent and only twelve may be selected at any one time.

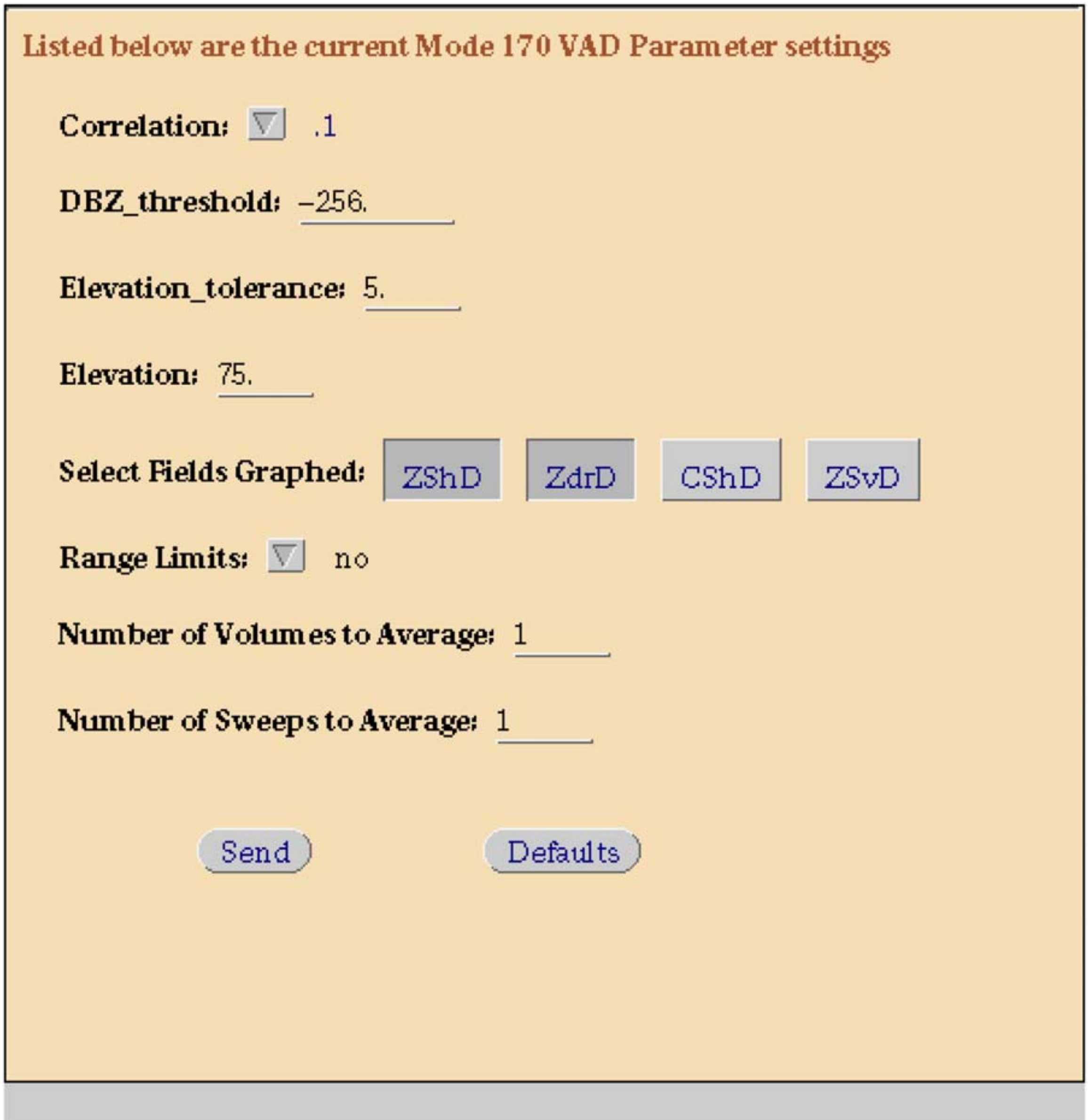

Figure 8. RADS VAD Parameter Settings Window allows the operator to change VAD profile display parameters such as which of the optional fields to display, the correlation threshold, desired elevation angle and tolerance angle.

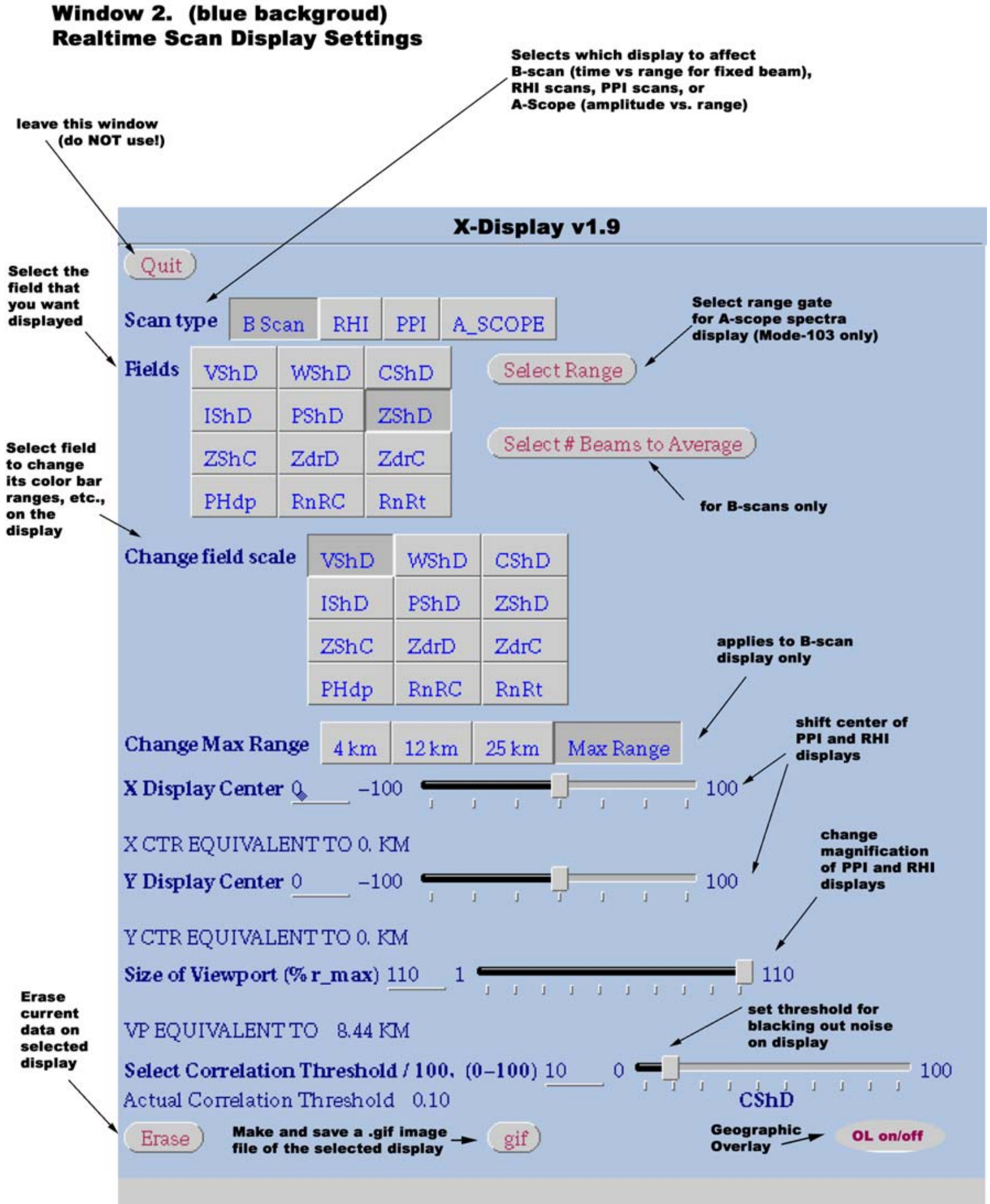

Figure 9. RADS x_display Control GUI allows the user to change the existing graphical displays. It allows the operator to select which field to display, pan, zoom, scale and threshold. It also allows the operator to gif images, input the number of average beams in the B-scan mode and erase selected images.

Appendix G.

ETL Projects Using RADS/GRADS with Ground-Based Scanning and SLAR Radars

Project	Radar	Year	Location	RADS Mode	Comments
COPE	NOAA/D	1995	Cannon Beach, OR	117	1st RADS deployment; Delta-k; ocean observations
RAIN-X	NOAA/D	1997	Erie, CO	159	Transmit H, V using transmit switch; rain observations
TRMM	NOAA/D	1998	Houston, TX	150	Transmit H, V using transmit switch; storm observations
MWISP	NOAA/D	1999	Bretton Woods, NH	140	No cross-polar data; transmit H, cloud observations
MWISP	NOAA/K	1999	Bretton Woods, NH	140	Phase-rotating plate in use; Usually transmit slant linear; cloud observations
AMSR-Rainstat	NOAA/D	1999	Erie, CO	140	Transmit slant linear; rain observations
HAWC	NOAA/K	2000	Hanscom AFB, MA	140	Transmit slant linear; cloud observations
ARM	NOAA/K	2000	Lamont, OK	140	Transmit slant linear; cloud observations
AMSR-Rainstat	NOAA/D	2000	Erie, CO	170	Transmit slant linear; rain observations
LLCC	NOAA/K	2001	Cape Canaveral, FL	140	Transmit slant linear; cloud observations
AMSR-Wallops	NOAA/D	2001	Wallops Island, VA	170	Transmit slant linear; rain observations
TH	NOAA/D	2001	Bahamas	103 and 170	Transmit slant linear; ocean & atmospheric observations
USCG/NOAA	SLAR	1997-2000	N. Carolina, Nova Scotia, Newfoundland, Caribbean, Florida, Irish Sea	160	Non-Doppler radar; ocean surface observations including iceberg detection
PACJET-03	NOAA/D	2003	Fort Ross, CA	170	Transmit slant linear; storm observations

Project	Radar	Year	Location	RADS Mode	Comments
TM	NOAA/D	2003	Bahamas	103 and 170	Transmit slant linear; ocean & atmospheric observations
AIRS 1.5	NOAA/K	2003	Erie, CO	170	GRIDS version; Transmit slant linear; Icing hazard detection
AIRS II	NOAA/K	2003	Mirabel, Canada	170	RADS & GRIDS version; Transmit slant linear; Icing hazard detection

Appendix H.

RADS Radar Parameters
Version 1.2

Radar Parameters are defined as all those parameters required to direct the radar transmitter, receiver, antenna pointing and real-time data processing, plus any additional housekeeping parameters required in the UF header, expressed in a form that can most directly control the radar. For example, the pulse repetition period is expressed, rather than the maximum unambiguous velocity or maximum unambiguous range. These parameters form a complete set, with no other parameters being required for radar operation or data recording.

Because there are so many parameters (about 123), they are classified into ten primary types: timing (TM), calibration (CL), transmit (TX), receive (RX), scanning (SC), data processing (DP), housekeeping (HK), GPS (GP), navigation (NV), phase-retarding plate (PL) and spare (SP). In general, timing and calibration take precedence over the other types in situations where a parameter might be considered to be in more than one type. For instance, PRPL is a TM parameter rather than a TX parameter, and receiver gain is a CL parameter rather than an RX parameter. This classification is used only for convenience in displaying groups of parameters, *i.e.*, the classification of a parameter has no functional significance beyond specifying how it is displayed to the operator.

Setting a bit in a 32-bit word associated with that parameter identifies each parameter type. This allows for the identification of additional overlapping types of parameters. For instance, all parameters normally varied when running in pulse-pair mode have a certain bit set so that they can be conveniently displayed when operating in that mode.

There are two bits that control access to the parameters: OE and PA, for Operator Enabled and Password Accessible. OE set indicates that any radar operator running the Radar Control Program (RCP) can change the value of that parameter. PA set indicates that the parameter can be changed, but only by knowledge of a password. Those parameters without either bit set are initialized by a different program, and then read into the RCP, or else they are parameters set by the RCP, such as TIME. Setting both OE and PA is not allowed.

Many of the calibration parameters depend on the receiver bandwidth, and so these parameters are in the form of a vector to accommodate three possible receiver bandwidths. With parameters of this type, which can take on several values, the actual value being used (determined by whatever receiver bandwidth has been selected) is stored in the first element (zeroth element in C nomenclature), and the possible values are stored in the following elements. Thus parameters that can take three possible values are stored in a four-element vector. The radar constant has two possible values depending on whether oceanic or atmospheric targets are being observed, and so is similarly stored in a three-element vector.

Also many calibration parameters have an “h” or “v” in their names, for horizontal and vertical polarization. In radars where this is not appropriate, they stand for co-polarized and cross-polarized, respectively.

*.rp’s are stored in /export/home/rcp/rp with an identifying name. When the RCP is started, it loads the file default.rp or some other programmer-specified file. Other sets of RPs can be loaded by operator command.

Most of the UF header can be derived from the Radar Parameters; the remaining UF header entries have a fixed value, or are dependent on the position of parameters within the UF header itself.

The Radar Parameters are all defined, with the exception of ASCII parameters, in terms of 32-bit numbers in order to be compatible with the DSP which has only 32-bit data types. These parameters reside in the SPARC's slave window where they are accessible to both the DSP, or any other VME master, and any process running on the SPARC. Note that because the DSP uses 32 bits to represent a char, that the structure used for the DSP must be slightly different than the structure used on the SPARC. A declaration like SCNM[8] on the SPARC must be changed to SCNM[2] for the DSP. The following table shows the SPARC structure.

The limits shown on individual parameters in the table are worst case limits that are large enough to encompass all modes of operation. Depending on the particular mode of operation, or the particular radar, other more restrictive limits are placed on certain parameters.

Changes from Version 1.1 are shaded.

Table 1. Revision history.

Version 1.0	1000-byte header released primarily to support USCG activities and Phase-Retarding Plate information.
Version 1.1	Added RLEN parameter to support mode 103.
Version 1.2	Added 12 parameters to support GRIDS.

Table 2. Radar parameters, version 1.2

	Parameter	Description	Data type	Units	Bytes	Type of parameter*	Lower limit	Upper limit	GRADS use?	Comment
1	RPSV	Radar Parameter structure version number (this structure)	float	none	4	HK	0.0	20.0	yes	Major rev when structure change alters parameter displacements (Version 1.2)
2	NPRM	Number of parameter names, excluding spares	uint	none	4	HK	50	200	yes	123 in this version
3	RTGC	Radar Timing Generator Control	uint	none	4	TM	0	65535	yes	Controls local/slave mode and Run bit
4	CLKF	Basic clock frequency of radar timing generator	float	Hz	4	TM | PA	16e6	20e6	no	16 MHz or 20 MHz
5	NTRG	Number of triggers in beam	uint	none	4	TM | OE	32	1048576	yes	Dependent parameter in FFT modes: NTRG = NCOH x NSMP x NSPC
6	PRPL	Inter- pair repetition period	float	seconds	4	TM | OE	70e-6	16e-3	yes	formerly PRPR
7	PRPS	Intra-pair trigger spacing	float	seconds	4	TM | OE	.05e-3	1.024e-3	no	formerly TRGS
8	NPWT	Number of pairs of triggers to wait between beams	uint	none	4	TM | OE	0	1023	no	
9	DLAY	Delay from trigger to first range gate	float	seconds	4	TM | OE	2e-6	1025e-6	yes	
10	SPAC	Spacing between range gates	float	seconds	4	TM | OE	50e-9	16e-6	yes	
11	NGAT	Number of range gates	uint	none	4	TM | OE	1	16384	yes	formerly NRGT
12	RTLN	Length of radar trigger	float	seconds	4	TM | OE	50e-9	12.8e-6	yes	OE only in GRADS 0 turns off transmitter in GRADS Same as TXWD in GRADS
13	BASP	Spacing from first (B) to second pre-trigger (A)	float	seconds	4	TM | PA	1e-6	63e-6	no	
14	B0SP	Spacing from first pre-trigger (B) to main trigger (0)	float	seconds	4	TM | PA	2e-6	64e-6	no	
15	TPL1[4]	Transmit polarization for first four polarizations in beam	char	none	4	TM | OE	0	0xffffffff	no	H – horizontal V – vertical S – split (both H and V) C – co-polar X – cross-polar
16	TPL2[4]	Transmit polarization for second four polarizations in beam	char	none	4	TM | OE	0	0xffffffff	no	same as above
17	RPL1[4]	Receive polarization for first four polarizations in beam	char	none	4	TM | OE	0	0xffffffff	yes	h – horizontal v – vertical s – split (both h & v using two rcvrs) c – co-polar x – cross-polar

	Parameter	Description	Data type	Units	Bytes	Type of parameter*	Lower limit	Upper limit	GRADS use?	Comment
18	RPL2[4]	Receive polarization for second four polarizations in beam	char	none	4	TM | OE	0	0xffffffff	yes	same as above
19	RPDL	Received polarization delay	float	seconds	4	TM | PA	100e-9	12.8e-6	yes	50 ns resolution
20	NSYC	Number of iSync's per SYNC	uint	none	4	TM	1	4	yes	Values 1, 2 or 4. DMOD dependent.
21	spare		uint		4	SP	0	0		
22	spare		uint		4	SP	0	0		
23	spare		uint		4	SP	0	0		
24	spare		uint		4	SP	0	0		
25	DMOD	Data mode (tmsr, spec, plpr, delk, etc.)	uint	none	4	DP | OE	0	255	yes	
26	ERRC	Error code from DSP	uint	none	4	DP	0	255	yes	returned from DSP on error
27	NDEC	Number of decimation steps (number of frequency cycles in average)	uint	none	4	DP | OE	1	8192	no	delta-k mode only
28	NSMP	Number of samples (meaning varies w/mode)	uint	none	4	DP | OE	1	16384	yes	In FFT modes, represents number of spectral points.
29	SCLF	Scale factor for integer data met units = tape value/SCLF	int	varies	4	DP	0	32767	no	0 signifies floating point
30	BPDS	Bits per data sample	uint	bits	4	DP	8	32	no	32 for floating point
31	OCEN	Non-zero indicates ocean mode.	uint	Boolean	4	DP	0	0xffffffff	no	
32	NCOH	Number of coherent averages	uint	none	4	DP | OE	1	64	yes	FFT modes
33	NSPC	Number of spectrum averaged	uint	none	4	DP | OE	1	65536	yes	FFT modes
34	spare		uint		4	SP	0	0		
35	spare		uint		4	SP	0	0		
36	spare		uint		4	SP	0	0		
37	spare		uint		4	SP	0	0		
38	spare		uint		4	SP	0	0		
39	spare		uint		4	SP	0	0		
40	TPWR	Transmitter peak power	float	watts	4	TX | PA	1.0	200e3	yes	
41	TFLG	Transmitter flag	int	none	4	TX	-1	1	yes	1 -- magnetron -1 -- klystron
42	TFMD	Transmit frequency mode	uint	none	4	TX	0	2	yes	0 -- fixed frequency 1 -- linear stepping 2 -- Goloumb stepping
43	TFRQ	Transmit frequency of radar. In stepping mode, the lowest or first frequency.	float	Hz	4	TX | OE	9e9	35e9	yes	
44	TFST	Transmit frequency step	float	Hz	4	TX | OE	1000	10e6	no	Stepping only

	Parameter	Description	Data type	Units	Bytes	Type of parameter*	Lower limit	Upper limit	GRADS use?	Comment
45	NTFR	Number of different transmit frequencies	uint	none	4	TX | OE	1	64	no	Stepping only
46	TMDL	TMEN (Transmit Envelope) delay	float	seconds	4	TM|TX|PA	100e-9	12.8e-6	yes	50 ns resolution.
47	TMWD	TMEN width	float	seconds	4	TM|TX|PA	50e-9	12.8e-6	yes	50 ns resolution.
48	TXDL	TXPL (Transmit Pulse) delay	float	seconds	4	TM|TX|PA	100e-9	12.8e-6	yes	50 ns resolution.
49	spare		uint		4	SP	0	0		
50	spare		uint		4	SP	0	0		
51	spare		uint		4	SP	0	0		
52	spare		uint		4	SP	0	0		
53	RPDH	Receiver pad, horizontal	float	dB	4	RX | OE	0.0	50.0	no	
54	RPDV	Receiver pad, vertical	float	dB	4	RX | OE	0.0	50.0	no	
55	TRDL	TREN (Transmit-Receive Envelope) delay	float	seconds	4	TM|RX|PA	100e-9	12.8e-6	yes	50 ns resolution.
56	TRWD	TREN width	float	seconds	4	TM|RX|PA	50e-9	12.8e-6	yes	50 ns resolution.
57	BLDL	BLNK (Blank) delay	float	seconds	4	TM|RX|PA	100e-9	12.8e-6	yes	50 ns resolution.
58	BLWD	BLNK width	float	seconds	4	TM|RX|PA	50e-9	12.8e-6	yes	50 ns resolution.
59	spare		uint		4	SP	0	0		
60	spare		uint		4	SP	0	0		
61	spare		uint		4	SP	0	0		
62	SWPM	Sweep mode	uint	none	4	SC | OE	0	6	yes	1 -- PPI (constant elevation) 2 -- Coplane (future) 3 -- RHI (constant azimuth) 4 -- Vertical 5 -- Target (stationary) 6 -- Manual 7 -- Idle (out of control) 8 – Sector (non-UF code, use 1 in UF)
63	MNAZ	Minimum (CCW) azimuth	float	degrees	4	SC | OE	-180.	360.	no	
64	MXAZ	Maximum (CW) azimuth	float	degrees	4	SC | OE	-180.	360.	no	
65	MNEL	Minimum elevation angle	float	degrees	4	SC | OE	-20.	135.	yes	
66	MXEL	Maximum elevation angle	float	degrees	4	SC | OE	-20.	135.	yes	
67	ANGF	Fixed angle	float	degrees	4	SC	-180.	360.	yes	generated by scan process
68	ANGI	Fixed angle increment	float	degrees	4	SC | OE	0.0	90.	yes	
69	BLAZ[3]	Baseline azimuth	float	degrees	12	SC | PA	0.0	360.	no	future: allows for COPLANE scanning with 2 other radars.
70	BLEL[3]	Baseline elevation	float	degrees	12	SC | PA	-20.	20.	no	Same as above.
71	SWTM	Sweep time	float	seconds	4	SC | OE	2.0	1e6	yes	In non-scanning modes, indicates how long files are.

	Parameter	Description	Data type	Units	Bytes	Type of parameter*	Lower limit	Upper limit	GRAD S use?	Comment
72	AZIM	Current azimuth angle	float	degrees	4	SC | PA	0.0	360.0	yes	fixed in GRADS. PA in GRADS only.
73	ELEV	Current elevation angle	float	degrees	4	SC | PA	-20.0	180.0	yes	fixed in GRADS. PA in GRADS only.
74	SCNM[8]	Scan name	char	none	8	SC | OE	0	0xffffffff	yes	specified by scan table name in RCP
75	spare		uint		4	SP	0	0		
76	spare		uint		4	SP	0	0		
77	spare		uint		4	SP	0	0		
78	spare		uint		4	SP	0	0		
79	spare		uint		4	SP	0	0		
80	spare		uint		4	SP	0	0		
81	spare		uint		4	SP	0	0		
82	TNUM	Tape number	uint	none	4	HK | OE	0	65535	yes	
83	TIME	UNIX time. Seconds since 01/01/70	uint	seconds	4	HK	820e6	1578e6	yes	Always in GMT
84	TIMS	Time past second in nanoseconds	uint	ns	4	HK	0	10000000 00	yes	Standard UNIX structure
85	TMZN[4]	Local time zone	char	none	4	HK | PA	0	0xffffffff	yes	Local time zone. GMT, MST, *etc.* First 2 characters go into UF header.
86	TMOF	Time zone offset from GMT	int	hours	4	HK | PA	-12	12	yes	For MST, TMOF = -7
87	LAT	Latitude of radar, degrees	float	degrees	4	HK	-90.0	90.0	yes	
88	LONG	Longitude of radar, degrees	float	degrees	4	HK	-180.0	180.0	yes	West long is negative
89	ELSL	Elevation above sea level of radar	float	meters	4	HK	-300.0	8840.0	yes	
90	OPNM[8]	Operator name	char	ASCII	8	HK | OE	0	0xffffffff	yes	
91	RDNM[8]	Radar name	char	ASCII	8	HK | PA	0	0xffffffff	yes	
92	STNM[8]	Site name	char	ASCII	8	HK | PA	0	0xffffffff	yes	
93	PRNM[8]	Project name	char	ASCII	8	HK | PA	0	0xffffffff	yes	
94	FTNM[8]	Field tape name	char	ASCII	8	HK | OE	0	0xffffffff	yes	
95	BMNM	Beam number within sweep	uint	none	4	HK	0	65535	yes	
96	SWNM	Sweep number within volume	uint	none	4	HK	0	65535	no	
97	VLNM	Volume number within tape	uint	none	4	HK	0	65535	no	
98	RCPN	RCP version number	float	none	4	HK | PA	0	20.0	yes	
99	RCDE	Indicates data is to be recorded	uint	Boolean	4	HK | OE	0	1	yes	
100	TDRV	Tape drive SCSI address	uint	none	4	HK	0	7	no	Useful in debugging drive problems
101	RLEN	Record size (RP header + data)	uint	bytes	4	HK	0	0xffffffff	yes	Added in 1.1 to support mode 103
102	RDID	Radar ID number	uint	none	4	HK|PA	0	0xffffffff	yes	
103	spare		uint		4	SP	0	0		
104	spare		uint		4	SP	0	0		
105	spare		uint		4	SP	0	0		

	Parameter	Description	Data type	Units	Bytes	Type of parameter*	Lower limit	Upper limit	GRADS use?	Comment
106	spare		uint		4	SP	0	0		
107	spare		uint		4	SP	0	0		
108	txph	Horizontal transmit power	float	watts	4	CL | PA	0	1e6	yes	h & v may also represent co-polar & cross-polar in all parameters they appear in
109	txpv	Vertical transmit power	float	watts	4	CL | PA	0	1e6	yes	
110	rnbw[4]	Possible receiver linear bandwidths	float	Hz	16	CL	100e3	40e6	yes	
111	rgbw[4]	Possible receiver log bandwidths	float	Hz	16	CL	100e3	40e6	no	
112	rnh[4]	gain of the linear horizontal receiver by BW	float	dB	16	CL | PA	0.0	120.0	yes	
113	rnv[4]	gain of the linear vertical receiver by BW	float	dB	16	CL | PA	20.0	120.0	yes	
114	rg1h[4]	RF gain of the log horizontal receiver by BW	float	dB	16	CL | PA	100.0	150.0	no	
115	rg2h[4]	IF gain of the log horizontal receiver by BW	float	dB	16	CL | PA	-30.0	10.0	no	
116	rg1v[4]	RF gain of the log vertical receiver by BW	float	dB	16	CL | PA	100.0	150.0	no	
117	rg2v[4]	IF gain of the log vertical receiver by BW	float	dB	16	CL | PA	-30.0	10.0	no	
118	R0h[4]	Noise value of R from the linear horizontal receiver by BW	float	none	16	CL | PA	0.0	4e-3	yes	
119	R0v[4]	Noise value of R from the linear vertical receiver by BW	float	none	16	CL | PA	0.0	4e-3	yes	
120	khRC[3]	horizontal radar constant by target	float	dB	12	CL | PA	-20.0	120.0	yes	khRC[1] for atmosphere khRC[2] for ocean
121	kvRC[3]	vertical radar constant by target	float	dB	12	CL | PA	-20.0	120.0	yes	kvRC[1] for atmosphere kvRC[2] for ocean
122	pknl[4]	pre-knock from linear in local mode actual dlay = DLAY + pknn by BW	float	seconds	16	CL | PA	-5e-6	0.0	yes	
123	pkgl[4]	pre-knock from log channels in local by BW	float	seconds	16	CL | PA	-5e-6	0.0	no	
124	pksa	pre-knock adjustment for slave mode actual pre-knock = pkxl + pksa	float	seconds	4	CL | PA	0.0	3e-6	no	
125	azbw	Azimuthal beamwidth	float	degrees	4	CL | PA	0.1	5.0	yes	
126	elbw	Elevation beamwidth	float	degrees	4	CL | PA	0.1	30.0	yes	
127	antg[2]	Antenna gain	float	dB	8	CL | PA	20.0	60.0	yes	antg[0] for horizontal pol antg[1] for vertical pol

	Parameter	Description	Data type	Units	Bytes	Type of parameter*	Lower limit	Upper limit	GRADS use?	Comment
128	aplr	Measure of antenna polarization ratio for elliptically polarized antennas.	float	none	4	CL | PA	0.1	10.0	yes	For elliptically polarized antennas only. H/V
129	ktar[2]	Magnitude of complex refractive index for target	float	none	8	CL | PA	0.25	1.0	yes	Used in radar equation. [0] = horizontal, [1] = vertical value
130	phi0	Differential phase offset	float	radians	4	CL | PA	-6.3	6.3	no	
131	sgnl	Signal generator level	float	dBm	4	CL | PA	-500.0	10.0	no	Used while calibrating
132	spare		uint		4	SP	0	0		
133	spare		uint		4	SP	0	0		
134	spare		uint		4	SP	0	0		
135	spare		uint		4	SP	0	0		
136	spare		uint		4	SP	0	0		
137	spare		uint		4	SP	0	0		
138	spare		uint		4	SP	0	0		
139	GPOS[3]	latitude, longitude, altitude	float	deg, m	12	GP	-1000	5000	yes	If GPS receiver present, this value copied into LAT, LONG, ELSL
140	TFIX	GPS time of fix	float	seconds	4	GP	0	604800	yes	Time within GPS week. Added 5/98
141	GPFX[2]	time of GPOS fix in UNIX seconds & nanoseconds	uint	secs, nsec	8	GP	0	$2^{31} - 1$	yes	
142	GVEL[3]	3-axis velocity (East-North-Up)	float	m/s	12	GP	-1000	+1000	no	
143	GVFX[2]	time of GVEL fix in UNIX seconds & nanoseconds	uint	secs, nsec	8	GP	0	$2^{31} - 1$	no	
144	spare		uint		4	SP	0	0		
145	spare		uint		4	SP	0	0		
146	spare		uint		4	SP	0	0		
147	spare		uint		4	SP	0	0		
148	spare		uint		4	SP	0	0		
149	spare		uint		4	SP	0	0		
150	HEAD	heading from true north	float	degrees	4	NV	-360.0	360.0	no	ARINC 582 or 571 (specify later)
151	TRCK[2]	speed, bearing over ground	float	m/s, deg	8	NV	-360.0	360.0	no	ARINC 582 or 571 (specify later)
152	NPOS[3]	latitude, longitude, altitude	float	deg, m	12	NV	-1000	5000	no	(582, 582, NA)
153	NATD[3]	pitch, roll, yaw	float	degrees	12	NV	-360.0	360.0	no	synchro?
154	NVEL[3]	N velocity, E velocity, Up velocity	float	m/s	12	NV	-1000	1000	no	(582, ??, 582). Up is integrated vertical acceleration
155	NWND[2]	wind speed, wind angle	float	m/s, deg	8	NV	-360	360	no	(582, 582)
156	NTFX[4]	time of acquisition of ARINC 582 & 571 data in Unix secs and ns	uint	secs, ns	16	NV	0	$2^{31} - 1$	no	(582, 571)
157	spare		uint		4	SP	0	0		

	Parameter	Description	Data type	Units	Bytes	Type of parameter*	Lower limit	Upper limit	GRADS use?	Comment
158	spare		uint		4	SP	0	0		
159	spare		uint		4	SP	0	0		
160	spare		uint		4	SP	0	0		
161	spare		uint		4	SP	0	0		
162	spare		uint		4	SP	0	0		
163	spare		uint		4	SP	0	0		
164	spare		uint		4	SP	0	0		
165	PLTR	maximum plate retarding phase	float	degrees	4	PL	OE	0	360.0	no	currently 79.5 or 178.7 degrees
166	PLTA	plate orientation angle, actual	float	degrees	4	PL	PA	0	360.0	no	
167	PLTN	minimum plate angle	float	degrees	4	PL	OE	0	360.0	no	for plate scanning
168	PLTX	maximum plate angle	float	degrees	4	PL	OE	0	360.0	no	for plate scanning
169	PLTS	plate step angle	float	degrees	4	PL	OE	0	360.0	no	synchronized to antenna sweeps
170	PLTV	plate angular velocity	float	deg/sec	4	PL	OE	0	1000	no	
171	PLTM	plate mode	uint	none	4	PL	OE	0	10	no	0 – plate fixed 1 – plate rotating continuously 2 – plate stepped 3 – plate oscillating
172	PLTI[4]	plate identification	char	none	4	PL	OE	0	0xffffffff	no	for example, “HWP3”
173	PLTB	plate binary	uint	none	4	PL	0	0xffffffff	no	bits from plate controller
174	spare		uint		4	SP	0	0		
175	spare		uint		4	SP	0	0		
176	spare		uint		4	SP	0	0		
177	spare		uint		4	SP	0	0		
178	spare		uint		4	SP	0	0		
179	spare		uint		4	SP	0	0		
180	spare		uint		4	SP	0	0		
181	spare		uint		4	SP	0	0		
					1000					

*Types of parameters: TM - timing; CL - calibration; DP - data processing; TX - transmitter; RX - receiver; SC - scanning; HK - housekeeping; GP - GPS (Global Positioning System); NV – navigation (INS) system; PL – plate (phase-retarding plate); SP - spare; OE - operator enabled; PA – password accessible.

Appendix I.

Example of Configuration File

```
# xParams.cfg 2001/10/15 JSG

# Any part of a line after a "#" is a comment
# This file sets the xDisplay defaults

# define the mode

#MODE = 103
MODE = 170

# define the number of fields to be displayed

NFIELDS = 11
#NFIELDS170 = 11
#NFIELDS = 12

BEAMS2AVERAGE = 2

# define the nfields sent to xDisplay

F1031 = Eh.R
F1032 = Eh.I
F1033 = Ev.R
F1034 = Ev.I
F1035 = PwSh
F1036 = PwSv
F1037 = VPab
F1038 = WPab

F1 = VShD
F2 = WShD
F3 = CShD
F4 = IShD
F5 = PShD
F6 = ZShD
F7 = ZShC
F8 = ZdrD
F9 = ZdrC
F10 = PHdp
F11 = RnRC
#F12 = Rhv0
```

#F1 = VShD
#F2 = WShD
#F3 = CShD
#F4 = IShD
#F5 = ISvD
#F6 = PShD
#F7 = ZShD
#F8 = ZShC
#F9 = ZdrD
#F10 = ZdrC
#F11 = PHdp
#F12 = RnRC

define max and max for the four types of scans for all 32 of the available
fields

NOTE: for mode 170 the following numbers refer to the corresponding
fields as shown below

1 = VShD
2 = VShU
3 = VSvD
4 = VSvU
5 = WShD
6 = WShU
7 = WSvD
8 = WSvU
9 = CShD
10 = CShU
11 = CSvD
12 = CSvU
13 = IShD
14 = ISvD
15 = PShD
16 = PShU
17 = PSvD
18 = PSvU
19 = ZShD
20 = ZShU
21 = ZSvD
22 = ZSvU
23 = ZShC
24 = ZSvC
25 = ZdrD
26 = ZdrC
27 = DVvh

28 = PHdp
29 = RnRt
30 = RnRC
31 = Rhv0
32 = DSmC

```
#ScaleMinBscan1 = -25.     # for mode 170 VShD
#ScaleMinBscan2 = -25.     # VShU
ScaleMinBscan1 = -15.      # VShD
ScaleMinBscan2 = -15.      # VShU
ScaleMinBscan3 = -25.      # VSvD
ScaleMinBscan4 = -25.      # VSvU
ScaleMinBscan5 = 0.        #WShD
ScaleMinBscan6 = 0.        #WShU
ScaleMinBscan7 = 0.        #WSvD
ScaleMinBscan8 = 0.        #WSvU
ScaleMinBscan9 = 0.        #CShD
ScaleMinBscan10 = 0.       #CShU
ScaleMinBscan11 = 0.       #CSvD
ScaleMinBscan12 = 0.       #CSvU
ScaleMinBscan13 = -45.     #IShD
ScaleMinBscan14 = -45.     #ISvD
ScaleMinBscan15 = -140.    #PShD
ScaleMinBscan16 = -140.    #PShU
ScaleMinBscan17 = -140.    #PSvD
ScaleMinBscan18 = -140.    #PSvU
ScaleMinBscan19 = 0.       #ZShD
ScaleMinBscan20 = 0.       #ZShU
ScaleMinBscan21 = 0.       #ZSvD
ScaleMinBscan22 = 0.       #ZSvU
ScaleMinBscan23 = 0.       #ZShC
ScaleMinBscan24 = 0.       #ZSvC
ScaleMinBscan25 = -2.      # ZdrD
ScaleMinBscan26 = -2.      #ZdrC
ScaleMinBscan27 = -1.      #DVvh
ScaleMinBscan28 = -10.     # Phdp
ScaleMinBscan29 = 0.       #RnRt
ScaleMinBscan30 = 0.       #RnRC
ScaleMinBscan31 = 0.       #Rhv0
ScaleMinBscan32 = 0.       #DSmC

#ScaleMaxBscan1 =  25.
#ScaleMaxBscan2 =  25.
ScaleMaxBscan1 =  15.
ScaleMaxBscan2 =  15.
ScaleMaxBscan3 =  45.
```

ScaleMaxBscan4 = 45.
ScaleMaxBscan5 = 10.
ScaleMaxBscan6 = 10.
ScaleMaxBscan7 = 10.
ScaleMaxBscan8 = 10.
ScaleMaxBscan9 = 1.
ScaleMaxBscan10 = 1.
ScaleMaxBscan11 = 1.
ScaleMaxBscan12 = 1.
ScaleMaxBscan13 = 15.
ScaleMaxBscan14 = 15.
ScaleMaxBscan15 = -40.
ScaleMaxBscan16 = -40.
ScaleMaxBscan17 = -40.
ScaleMaxBscan18 = -40.
ScaleMaxBscan19 = 60.
ScaleMaxBscan20 = 60.
ScaleMaxBscan21 = 60.
ScaleMaxBscan22 = 60.
ScaleMaxBscan23 = 60.
ScaleMaxBscan24 = 60.
ScaleMaxBscan25 = 4.
ScaleMaxBscan26 = 4.
ScaleMaxBscan27 = 1.
ScaleMaxBscan28 = 110.
ScaleMaxBscan29 = 50.
ScaleMaxBscan30 = 50.
ScaleMaxBscan31 = 1.
ScaleMaxBscan32 = 4.

#ScaleMinPPI1 = -25.
ScaleMinPPI1 = -15.
ScaleMinPPI2 = -25.
ScaleMinPPI3 = -25.
ScaleMinPPI4 = -25.
ScaleMinPPI5 = 0.
ScaleMinPPI6 = 0.
ScaleMinPPI7 = 0.
ScaleMinPPI8 = 0.
ScaleMinPPI9 = 0.
ScaleMinPPI10 = 0.
ScaleMinPPI11 = 0.
ScaleMinPPI12 = 0.
ScaleMinPPI13 = -45.
ScaleMinPPI14 = -45.
ScaleMinPPI15 = -140.

ScaleMinPPI16 = -140.
ScaleMinPPI17 = -140.
ScaleMinPPI18 = -140.
ScaleMinPPI19 = 0.
ScaleMinPPI20 = 0.
ScaleMinPPI21 = 0.
ScaleMinPPI22 = 0.
ScaleMinPPI23 = 0.
ScaleMinPPI24 = 0.
ScaleMinPPI25 = -2.
ScaleMinPPI26 = -2.
ScaleMinPPI27 = -1.
ScaleMinPPI28 = -10.
ScaleMinPPI29 = 0.
ScaleMinPPI30 = 0.
ScaleMinPPI31 = 0.
ScaleMinPPI32 = 0.

#ScaleMaxPPI1 = 25.
ScaleMaxPPI1 = 15.
ScaleMaxPPI2 = 25.
ScaleMaxPPI3 = 25.
ScaleMaxPPI4 = 25.
ScaleMaxPPI5 = 10.
ScaleMaxPPI6 = 10.
ScaleMaxPPI7 = 10.
ScaleMaxPPI8 = 10.
ScaleMaxPPI9 = 1.
ScaleMaxPPI10 = 1.
ScaleMaxPPI11 = 1.
ScaleMaxPPI12 = 1.
ScaleMaxPPI13 = 15.
ScaleMaxPPI14 = 15.
ScaleMaxPPI15 = -40.
ScaleMaxPPI16 = -40.
ScaleMaxPPI17 = -40.
ScaleMaxPPI18 = -40.
ScaleMaxPPI19 = 60.
ScaleMaxPPI20 = 60.
ScaleMaxPPI21 = 60.
ScaleMaxPPI22 = 60.
ScaleMaxPPI23 = 60.
ScaleMaxPPI24 = 60.
ScaleMaxPPI25 = 4.
ScaleMaxPPI26 = 4.
ScaleMaxPPI27 = 1.

ScaleMaxPPI28 = 110.
ScaleMaxPPI29 = 50.
ScaleMaxPPI30 = 50.
ScaleMaxPPI31 = 1.
ScaleMaxPPI32 = 4.

#ScaleMinRHI1 = -25.
ScaleMinRHI1 = -15.
ScaleMinRHI2 = -25.
ScaleMinRHI3 = -25.
ScaleMinRHI4 = -25.
ScaleMinRHI5 = 0.
ScaleMinRHI6 = 0.
ScaleMinRHI7 = 0.
ScaleMinRHI8 = 0.
ScaleMinRHI9 = 0.
ScaleMinRHI10 = 0.
ScaleMinRHI11 = 0.
ScaleMinRHI12 = 0.
ScaleMinRHI13 = -45.
ScaleMinRHI14 = -45.
ScaleMinRHI15 = -140.
ScaleMinRHI16 = -140.
ScaleMinRHI17 = -140.
ScaleMinRHI18 = -140.
ScaleMinRHI19 = 0.
ScaleMinRHI20 = 0.
ScaleMinRHI21 = 0.
ScaleMinRHI22 = 0.
ScaleMinRHI23 = 0.
ScaleMinRHI24 = 0.
ScaleMinRHI25 = -2.
ScaleMinRHI26 = -2.
ScaleMinRHI27 = -1.
ScaleMinRHI28 = -10.
ScaleMinRHI29 = 0.
ScaleMinRHI30 = 0.
ScaleMinRHI31 = 0.
ScaleMinRHI32 = 0.

#ScaleMaxRHI1 = 25.
ScaleMaxRHI1 = 15.
ScaleMaxRHI2 = 25.
ScaleMaxRHI3 = 25.
ScaleMaxRHI4 = 25.
ScaleMaxRHI5 = 10.

ScaleMaxRHI6 = 10.
ScaleMaxRHI7 = 10.
ScaleMaxRHI8 = 10.
ScaleMaxRHI9 = 1.
ScaleMaxRHI10 = 1.
ScaleMaxRHI11 = 1.
ScaleMaxRHI12 = 1.
ScaleMaxRHI13 = 15.
ScaleMaxRHI14 = 15.
ScaleMaxRHI15 = -40.
ScaleMaxRHI16 = -40.
ScaleMaxRHI17 = -40.
ScaleMaxRHI18 = -40.
ScaleMaxRHI19 = 60.
ScaleMaxRHI20 = 60.
ScaleMaxRHI21 = 60.
ScaleMaxRHI22 = 60.
ScaleMaxRHI23 = 60.
ScaleMaxRHI24 = 60.
ScaleMaxRHI25 = 4.
ScaleMaxRHI26 = 4.
ScaleMaxRHI27 = 1.
ScaleMaxRHI28 = 110.
ScaleMaxRHI29 = 50.
ScaleMaxRHI30 = 50.
ScaleMaxRHI31 = 1.
ScaleMaxRHI32 = 4.

ScaleMinAScope1 = -25.
#ScaleMinAScope1 = -10.
ScaleMinAScope2 = -25.
ScaleMinAScope3 = -25.
ScaleMinAScope4 = -25.
ScaleMinAScope5 = 0.
ScaleMinAScope6 = 0.
ScaleMinAScope7 = 0.
ScaleMinAScope8 = 0.
ScaleMinAScope9 = 0.
ScaleMinAScope10 = 0.
ScaleMinAScope11 = 0.
ScaleMinAScope12 = 0.
ScaleMinAScope13 = -45.
ScaleMinAScope14 = -45.
ScaleMinAScope15 = -140.
ScaleMinAScope16 = -140.
ScaleMinAScope17 = -140.

ScaleMinAScope18 = -140.
ScaleMinAScope19 = 0.
ScaleMinAScope20 = 0.
ScaleMinAScope21 = 0.
ScaleMinAScope22 = 0.
ScaleMinAScope23 = 0.
ScaleMinAScope24 = 0.
ScaleMinAScope25 = -2.
ScaleMinAScope26 = -2.
ScaleMinAScope27 = -1.
ScaleMinAScope28 = -10.
ScaleMinAScope29 = 0.
ScaleMinAScope30 = 0.
ScaleMinAScope31 = 0.
ScaleMinAScope32 = 0.

ScaleMaxAScope1 = 25.
#ScaleMaxAScope1 = 10.
ScaleMaxAScope2 = 25.
ScaleMaxAScope3 = 25.
ScaleMaxAScope4 = 25.
ScaleMaxAScope5 = 10.
ScaleMaxAScope6 = 10.
ScaleMaxAScope7 = 10.
ScaleMaxAScope8 = 10.
ScaleMaxAScope9 = 1.
ScaleMaxAScope10 = 1.
ScaleMaxAScope11 = 1.
ScaleMaxAScope12 = 1.
ScaleMaxAScope13 = 15.
ScaleMaxAScope14 = 15.
ScaleMaxAScope15 = -40.
ScaleMaxAScope16 = -40.
ScaleMaxAScope17 = -40.
ScaleMaxAScope18 = -40.
ScaleMaxAScope19 = 60.
ScaleMaxAScope20 = 60.
ScaleMaxAScope21 = 60.
ScaleMaxAScope22 = 60.
ScaleMaxAScope23 = 60.
ScaleMaxAScope24 = 60.
ScaleMaxAScope25 = 4.
ScaleMaxAScope26 = 4.
ScaleMaxAScope27 = 1.
ScaleMaxAScope28 = 110.
ScaleMaxAScope29 = 50.

ScaleMaxAScope30 = 50.
ScaleMaxAScope31 = 1.
ScaleMaxAScope32 = 4.

BScanThresh1 = 0.
BScanThresh2 = 0.
BScanThresh3 = 0.
BScanThresh4 = 0.
BScanThresh5 = 0.
BScanThresh6 = 0.
BScanThresh7 = 0.
BScanThresh8 = 0.
BScanThresh9 = 0.
BScanThresh10 = 0.
BScanThresh11 = 0.
BScanThresh12 = 0.
BScanThresh13 = 0.
BScanThresh14 = 0.
BScanThresh15 = 0.
BScanThresh16 = 0.
BScanThresh17 = 0.
BScanThresh18 = .1
BScanThresh19 = .1
BScanThresh20 = .1
BScanThresh21 = .1
BScanThresh22 = .1
BScanThresh23 = .1
BScanThresh24 = 0.
BScanThresh25 = 0.
BScanThresh26 = 0.
BScanThresh27 = 0.
BScanThresh28 = 0.
BScanThresh29 = 0.
BScanThresh30 = 0.
BScanThresh31 = 0.
BScanThresh32 = 0.

RHIThresh1 = 0.
RHIThresh2 = 0.
RHIThresh3 = 0.
RHIThresh4 = 0.
RHIThresh5 = 0.
RHIThresh6 = 0.
RHIThresh7 = 0.
RHIThresh8 = 0.
RHIThresh9 = 0.

RHIThresh10 = 0.
RHIThresh11 = 0.
RHIThresh12 = 0.
RHIThresh13 = 0.
RHIThresh14 = 0.
RHIThresh15 = 0.
RHIThresh16 = 0.
RHIThresh17 = 0.
RHIThresh18 = .1
RHIThresh19 = .1
RHIThresh20 = .1
RHIThresh21 = .1
RHIThresh22 = .1
RHIThresh23 = .1
RHIThresh24 = 0.
RHIThresh25 = 0.
RHIThresh26 = 0.
RHIThresh27 = 0.
RHIThresh28 = 0.
RHIThresh29 = 0.
RHIThresh30 = 0.
RHIThresh31 = 0.
RHIThresh32 = 0.

PPIThresh1 = 0.
PPIThresh2 = 0.
PPIThresh3 = 0.
PPIThresh4 = 0.
PPIThresh5 = 0.
PPIThresh6 = 0.
PPIThresh7 = 0.
PPIThresh8 = 0.
PPIThresh9 = 0.
PPIThresh10 = 0.
PPIThresh11 = 0.
PPIThresh12 = 0.
PPIThresh13 = 0.
PPIThresh14 = 0.
PPIThresh15 = 0.
PPIThresh16 = 0.
PPIThresh17 = 0.
PPIThresh18 = .1
PPIThresh19 = .1
PPIThresh20 = .1
PPIThresh21 = .1
PPIThresh22 = .1

PPIThresh23 = .1
PPIThresh24 = 0.
PPIThresh25 = 0.
PPIThresh26 = 0.
PPIThresh27 = 0.
PPIThresh28 = 0.
PPIThresh29 = 0.
PPIThresh30 = 0.
PPIThresh31 = 0.
PPIThresh32 = 0.

AScopeThresh1 = 0.
AScopeThresh2 = 0.
AScopeThresh3 = 0.
AScopeThresh4 = 0.
AScopeThresh5 = 0.
AScopeThresh6 = 0.
AScopeThresh7 = 0.
AScopeThresh8 = 0.
AScopeThresh9 = 0.
AScopeThresh10 = 0.
AScopeThresh11 = 0.
AScopeThresh12 = 0.
AScopeThresh13 = 0.
AScopeThresh14 = 0.
AScopeThresh15 = 0.
AScopeThresh16 = 0.
AScopeThresh17 = 0.
AScopeThresh18 = .1
AScopeThresh19 = .1
AScopeThresh20 = .1
AScopeThresh21 = .1
AScopeThresh22 = .1
AScopeThresh23 = .1
AScopeThresh24 = 0.
AScopeThresh25 = 0.
AScopeThresh26 = 0.
AScopeThresh27 = 0.
AScopeThresh28 = 0.
AScopeThresh29 = 0.
AScopeThresh30 = 0.
AScopeThresh31 = 0.
AScopeThresh32 = 0.

SelectedFieldBscan = ZShD
SelectedFieldBscan103 = PwSh

```
SelectedFieldPPI =  ZShD
#SelectedFieldPPI103 =  ZShD

SelectedFieldRHI =  ZShD
#SelectedFieldRHI103 =  ZShD

SelectedFieldA_SCOPE =  ZShD
SelectedFieldA_SCOPE103 =  PwSh

BscanMinColor = white
BscanMaxColor = white

#BscanThreshColor = black
BscanThreshColor = white

#RhiMinColor = white
RHIMinColor = black

RHIMaxColor = white
#RHIMaxColor = black

RHIThreshColor = black
#RHIThreshColor = white

# since Rhi and Ppi share the rasterizing code, they will need to be the same
# for now; Rhi will define the wrap min and wrap max and thresh color for both
# rhi and ppi

SelectedGifFieldBscan = ZShD
SelectedGifFieldAscope =  ZShD
SelectedGifFieldRHI =  ZShD
SelectedGifFieldPPI =  ZShD

SelectedGifFieldBscan103 = PwSh
SelectedGifFieldAscope103 =  PwSh
#SelectedGifFieldRHI103 =  PwSh
#SelectedGifFieldPPI103 =  PwSh

ScaleMinBscan1031 = -1500.
ScaleMinBscan1032 = -1500.
ScaleMinBscan1033 = -2500.
ScaleMinBscan1034 = -2500.
ScaleMinBscan1035 = 0.
ScaleMinBscan1036 = 0.
ScaleMinBscan1037 = 0.
```

ScaleMinBscan1038 = 0.
#
ScaleMaxBscan1031 = 1500.
ScaleMaxBscan1032 = 1500.
ScaleMaxBscan1033 = 4500.
ScaleMaxBscan1034 = 4500.
ScaleMaxBscan1035 = 1000.
ScaleMaxBscan1036 = 10.
ScaleMaxBscan1037 = 10.
ScaleMaxBscan1038 = 10.
#
ScaleMinAScope1031 = -2500.
ScaleMinAScope1032 = -2500.
ScaleMinAScope1033 = -2500.
ScaleMinAScope1034 = -2500.
ScaleMinAScope1035 = 0.
ScaleMinAScope1036 = 0.
ScaleMinAScope1037 = 0.
ScaleMinAScope1038 = 0.
#
ScaleMaxAScope1031 = 2500.
ScaleMaxAScope1032 = 2500.
ScaleMaxAScope1033 = 2500.
ScaleMaxAScope1034 = 2500.
ScaleMaxAScope1035 = 1000.
ScaleMaxAScope1036 = 10.
ScaleMaxAScope1037 = 10.
ScaleMaxAScope1038 = 10.

Appendix J.

Differential polarization calculations using switched transmit phase

1. INTRODUCTION

This document discusses algorithms that can be used for calculating differential polarization. The algorithms seen in the literature are for equally spaced pulses, and so must be modified slightly for use with pairs of pulses that are unequally spaced. This document also discusses some of the trade-offs involved between different polarization sequences and single and dual-receiver systems. Some of the functions that are used in the signal processing code are presented.

2. ALGORITHMS

Measurement of rainfall by the differential phase method is based on the fact that horizontally and vertically polarized waves propagate through rainfall at different rates, which causes a phase shift between them. By measuring the phase shift over range, and then looking at how fast the phase is shifting vs. range, the rainfall rate can be inferred. A dual receiver/transmitter system that can transmit and receive both polarizations simultaneously can make this measurement by comparing the phases between the horizontal and vertical channels. With only one transmitter, some time elapses between the horizontal and vertical sampling, and thus the phase is changed by the Doppler shift. This must be accounted for in calculating differential phase. The following algorithms account for this effect in their derivation.

2.1 Calculation of differential phase and derived products

2.1.1 Calculation of differential phase from a single receiver

The following is from Sachidananda and Zrnic (1989), and assumes a single receiver.

The differential phase algorithm for an equally spaced transmit sequence of HVHV... and a corresponding receive sequence of hvhv... requires forming sums of the form

$$R_a(T_s) = \frac{1}{M}\sum_{i=0}^{M-1} H_{2i}^{*} V_{2i+1}\ , \qquad R_b(T_s) = \frac{1}{M}\sum_{i=0}^{M-1} V_{2i+1}^{*} H_{2i+2}\ ;$$

where H and V, which are complex, represent the echo from an H- or V-transmitted wave, M is half the number of pulses in the beam, and i is the pulse index. Note that H and V in the equations are redundant, since whether i is odd or even specifies whether the signal is from an H or V pulse; however, their use clarifies what is being done.

Differential phase, as defined by Φ_{HH} - Φ_{VV}, is estimated from

$$\Phi_{DP} = \frac{1}{2} arg(R_a R_b^*) \ .$$

Φ_{DP} is a generally increasing function with range when waves propagate through most types of hydrometeors. The preceding equation only returns arguments from $\pm\pi$, so it will be necessary to monitor when the phase folds and add 2π to it to provide a continuous function.

2.1.2 Calculation of K_{DP}

K_{DP} is the range derivative of Φ_{DP} , but Φ_{DP} data is usually too noisy to take the derivative directly. Instead, filtered derivatives must be used.

2.1.3 Calculation of Rainfall Rate

From Jameson (1991), Table 2, there is the following relationship:

$$R = C\Phi^p \ ,$$

where R is rainfall rate in mm/hr, and Φ is phase shift in deg/km, which is equivalent to our K_{DP}. Using the numbers in his table for 9 GHz:

$$R = 13.03 K_{DP}^{0.9403} \ .$$

Using this (or similar relationships), plots of rainfall vs. range can be produced.

2.2 Calculation of ρ_{hv}

The following is from Zahrai and Zrnic (1993), pp. 653 - 655.

ρ_{hv} represents the correlation between signals in Hh and Vv. Because simultaneous Hh and Vv samples are not available, some assumptions were made, including that of a Gaussian spectrum shape, to derive the following equations.

First, compute

$$S_h = \frac{1}{M} \sum_{i=1}^{M} \left| H_{2i} \right|^2 , \qquad S_v = \frac{1}{M} \sum_{i=1}^{M} \left| V_{2i+1} \right|^2 \ .$$

Then compute the following running-sum

$$\rho(2T_s) = \frac{\left| \sum_{i=1}^{M} \left(H_{2i}^* H_{2i+2} + V_{2i+1}^* V_{2i+3} \right) \right|}{M(S_h + S_v)} \ .$$

Then compute

$$\left|\rho_{hv}\left(T_s\right)\right| = \frac{\left|R_a\right| + \left|R_b\right|}{2\sqrt{S_h S_v}} .$$

Finally,

$$\left|\rho_{hv}\left(0\right)\right| = \frac{\left|\rho_{hv}\left(T_s\right)\right|}{\left[\rho\left(2T_s\right)\right]^{0.25}} .$$

The paper states that this estimator is biased by white noise, and that the bias can be removed if the signal-to-noise ratio is known; however, the formula to do this is not given.

2.3 Calculation of Velocity and other Pulse Pair Products

Velocities can be calculated by running HV data through the standard pulse pair algorithm with overlapping pairs, but there are problems: the velocity is only unambiguous when $\Phi_{DP} < 90°$, and the estimator becomes increasingly noisy as it approach this value. There are ways to correct for these effects, but they add computational complexity. For this application, a better way is to process the data as pairs (*e.g.,* HH VV HH VV) with the same received polarization within the pair such that our standard estimators for velocity, width, power and correlation can be used. This is discussed in the next section.

2.4 Algorithm Modifications when using Pulse Pairs

2.4.1 Polarization Sequences

If we restrict ourselves to having the same transmit and receive polarization within a pair, some simplifications can be made in the programming. We will always number our pulses starting with zero, so the last pulse is the N - 1 pulse. Pulse sequences will repeat after m pulses, so N/m must be an integer. Because it is desired to obtain all possible co- and cross-polarization products, four pairs of pulses will be needed. In order to have the products in the equations for R_a and R_b come from equally spaced samples, the co-polar pairs must be equally spaced. If we choose to first transmit horizontally, then the sequence must look like this:

H	H			V	V			H	H			V	V		
h	h			v	v			h	h			v	v		
0	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1
										0	1	2	3	4	5

The missing cross-polar pairs can be filled in two different ways, and we have chosen the following:

H	H	H	H	V	V	V	V	H	H	H	H	V	V	V	V
h	h	v	v	v	v	h	h	h	h	v	v	v	v	h	h
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

This illustrates a sequence where N = 16 and m = 8, and is referred to as the primary polarization sequence.

2.4.2 Differential Phase Algorithms

The preceding algorithms, which are given for an equally spaced pulse train of HVHV... and hvhv..., must be modified for use with pulse pair. Use as an example the following sums:

$$R_a(T_s) = \frac{1}{M}\sum_{i=0}^{M-1} H_{2i}^{*} V_{2i+1}, \qquad R_b(T_s) = \frac{1}{M}\sum_{i=0}^{M-1} V_{2i+1}^{*} H_{2i+2} .$$

Using the primary polarization sequence, we want to form our R_a sum from the factors (0,4) and (1,5); and our R_b from the factors (4,8) and (5,9). Both pulses in the pair are used to reduce the uncertainty of the estimate. Since similar sums are used in several places, it is desirable to define a general function for these covariance functions that will operate on an arbitrary polarization sequence. If we let E_i represent the i^{th} complex sample in the sequence, then we can write

$$R_p(s,n,m,N) = \frac{m}{2(N-m)}\sum_{i=0}^{\frac{N}{m}-2} \left(E_{s+mi}^{*} E_{s+n+mi} + E_{s+1+mi}^{*} E_{s+1+n+mi}\right),$$

where s is the starting index, n is the lag, m is the length of the sequence cycle, and N is the total length of the sequence (number of triggers). The "p" in R_p indicates the data were processed as pairs of pulse pairs. Then

$R_a = R_p(0, 4, 8, 16)$

in our example of a 16-pulse sequence, and

$R_b = R_p(4, 4, 8, 16)$.

Also, the sum

$$\sum_{i=1}^{M}\left(H_{2i}^{*} H_{2i+2} + V_{2i+1}^{*} V_{2i+3}\right)$$

can be represented as

$R_p(0, 8, 8, 16) + R_p(4, 8, 8, 16)$.

2.4.3 The Pulse Pair Algorithm

We can also use a similar scheme for the standard pulse-pair algorithm. The running sums for the algorithm can be written as

$$R(T_s)=\frac{2}{N}\sum_{i=0}^{\frac{N}{2}-1}E_{2i}E_{2i+1}^{*} \qquad R(0)=\frac{1}{N}\sum_{i=0}^{\frac{N}{2}-1}\left(\left|E_{2i}\right|^2+\left|E_{2i+1}\right|^2\right)$$

for a polarization sequence such as HH HH HH HH ....
For a generalized polarization sequence, we can define functions

$$R_s(s,m,N)=\frac{m}{N}\sum_{i=0}^{\frac{N}{m}-1}E_{s+im}E_{s+1+im}^{*} \text{ and } R_0(s,m,N)=\frac{m}{2N}\sum_{i=0}^{\frac{N}{m}-1}\left(\left|E_{s+im}\right|^2+\left|E_{s+1+im}\right|^2\right).$$

These functions can be used to compute any of the running sums from a specified polarization pair in the sequence.

3. HARDWARE TRADEOFFS

The following discussion concerns what can be done within a single beam. Note that H and V refer to transmitted polarization, and h and v refer to received polarization.

3.1 Single Receiver

The non-pulse pair transmit sequence HVHV... gives the shortest dwell time, but it has some problems. In applying a modified pulse-pair estimator to this sequence, the unambiguous velocity is cut in half. There are methods to deal with this, but they add computational complexity. Since the receiver polarization is hvhv..., less than half the data products possible are produced, and the velocity is a mix of horizontally and vertically polarized co-polar data.

The next possibility is to use pulse pair and transmit a sequence of HH VV HH VV and a matching receive sequence of hh vv hh vv. Now the pulse pair algorithm on the HH and VV pairs can be used separately to obtain co-polar velocity for both polarizations with the expected unambiguous velocity; however, only half of the available data products are obtained as there are no cross-polarized products. The dwell time is twice as long as the previous case for pulse-pair products (although co- and cross-polarized data are being mixed together in the first case). The dwell time for differential phase products is the same as in the first case, but the time between the differential phase samples is twice as long.

Another possibility is to use pulse pair and transmit HH HH VV VV and receive hh vv vv hh. The differential phase products are formed by combining data from the first and third pairs. Now all four data products are obtained: Hh, Hv, Vh and Vv. The dwell time for pulse-pair products is four times as long as in the first case. The dwell time for differential phase products is twice as long as in the first and second cases. The time between samples for differential phase is four times as long as in the first case, and twice as long as in the second case, which may have some implications for the differential phase estimators.

3.2 Dual Receiver

A dual receiver has the inherent problem of matching phase and amplitude characteristics between the two receiver channels, but this has not proven difficult in practice. Looking vertically in rain can be used as a validation of calibration, since the circular cross-section of the rain drops provides a signal that should be matched between the two channels.

The advantage of the dual receiver approach is that the data are gathered twice as fast. The obvious choice for a transmit polarization sequence is HH VV HH VV while receiving hh and vv simultaneously all the time.

3.3 Dual Receiver -- Dual Transmitter

Being able to transmit and receive both polarizations simultaneously is the ultimate polarization capability, but requires two transmitter tubes or a high-power signal splitter. The CHILL S-band radar has this capability. This provides a further factor-of-two speed up in data acquisition. Unfortunately, this is not possible with our existing equipment configuration.

3.4 Summary

The following table summarizes the characteristics of different polarization sequences. For comparison purposes, everything is compared to an eight-pulse sequence, even though some sequences repeat after two or four pulses. B or b represents both polarizations simultaneously. The beam time is for an average PRP of 500 μs.

Table 1. Summary of different polarization sequences.

Polarization sequence	**Polarization products**	**Pulse pair samples/ sequence/product**	**Beam time for 128 samples**	**Comment**
HVHVHVHV hvhvhvhv	1	8	64 ms	Pulse-pair polarization products mixed together
HH VV HH VV Hh vv hh vv	2	4	128 ms	Only co-polar products
HH HH VV VV Hh vv vv hh	4	2	256 ms	co- and cross-polar products
HH VV HH VV Bb bb bb bb	4	4	128 ms	dual receiver, all products
BB BB BB BB Bb bb bb bb	4	8	64 ms	dual receiver and transmitter, all products

4. DATA PRODUCTS

4.1 Nomenclature

The covariance (running-sums) algorithms run through an input data stream and combine variables in various positions in certain ways without taking into account the polarization used to gather these variables. Thus, it is useful to define variables at this point that only depend on their position in the sequence. There are three types of covariance variables: Rs, Rp and Rz.

Rs represents variables where the conjugate product is formed between two samples in the same pair. These are used for the standard pulse-pair algorithm where the lag is always one sample time, and the sum is over the same polarizations. There are at most four of these complex variables: RsHh, RsHv, RsVv and RsVh. The generic (polarization-independent) form of these variables is Rsi, where $0 \leq i \leq m - 2$ and i represents the first location in the sequence of the variable. The angle of these variables times a constant gives the velocity.

Rp represents variables where the conjugate product is formed between samples in different pairs. In this case, it is always desirable to form a conjugate product between both sets of pulses in the two pairs in order to reduce the uncertainty of the estimate. The algorithms specified so far require the formation of four of these complex variables: RpVH, RpHV, RpHH and RpVV. The two capital letters following Rp imply that the transmitted and received polarizations are the same. If we wanted to form cross-polarized products, more letters would be required, e.g., RpHvVh. The generic form of these variables is Rpin, where i represent the first location in the sequence of the variable, and n represents the number of lags to the second variable.

Rz represents an autocovariance at zero lag, which is the square of the magnitude of the complex sample. When we are performing this operation, we always want to use both pulses in the pair. There are four possible values of this variable: RzHh, RzHv, RzVv and RzVh. The sequence-independent form of this variable is Rzi.

4.2 Summary of Naming Rules for Variables

R indicates the covariance function. If it is followed by an s, it indicates that the product was done within the pair and the following letters indicate the transmit/receive polarization (this should never change within a pair). If it is followed by a p, it indicates that the product was done from one pair to another, and both samples in the pair were used to reduce the uncertainty. If both of the following letters are capitals, it indicates the transmit polarization and implies that the received polarization was the same.

If the R is followed by a z, it indicates a zero-lag autocovariance (proportional to power). Both samples in the pair are used to form this quantity. The next two letters indicate the transmit/receive polarization.

4.3 Products for Recording

As is done for the normal pulse-pair mode, only covariance (running-sums) data is recorded, both to save tape and to allow recalculation of derived fields with different parameters. All fields are DC-corrected. The required fields are shown in the following table. Complex quantities are in **bold** type.

Table 2. Recorded Products

Generic variable name	Polarization variables	Mathematical symbol	Function used to calculate variable
Rs0	**RsHh**	$\mathbf{R(T_s)}$	**Rs**(0, 8, N)
Rs2	**RsHv**	$\mathbf{R(T_s)}$	**Rs**(2, 8, N)
Rs4	**RsVv**	$\mathbf{R(T_s)}$	**Rs**(4, 8, N)
Rs6	**RsVh**	$\mathbf{R(T_s)}$	**Rs**(6, 8, N)
Rz0	RzHh	R(0) or S_h	Rz(0, 8, N)
Rz2	RzHv	R(0)	Rz(2, 8, N)
Rz4	RzVv	R(0) or S_v	Rz(4, 8, N)
Rz6	RzVh	R(0)	Rz(6, 8, N)
Rp04	**RpHV**	$\mathbf{R_a(T_s)}$	**Rs**(0, 4, 8, N)
Rp44	**RpVH**	$\mathbf{R_b(T_s)}$	**Rs**(4, 4, 8, N)
Rp08	**RpHH**	$\sum \mathbf{H^*_{2i} H_{2i+2}}$	**Rs**(0, 8, 8, N)
Rp48	**RpVV**	$\sum \mathbf{V^*_{2i+1} V_{2i+3}}$	**Rs**(4, 8, 8, N)

4.4 Products for Display

Using the polarization sequence HH HH VV VV and hh vv vv hh, the following basic quantities from the linear channel can be calculated using the standard relations.

Table 3. Pulse-Pair Variables.

Field					**Units**
Velocity	VPHh	VPHv	VPVv	VPVh	m/s
Width	WPHh	WPHv	WPVv	WPVh	m^2/s^2
Correlation	CPHh	CPHv	CPVv	CPVh	none
Intensity	NIHh	NIHv	NIVv	NIVh	watts at receiver output
Power	NPHh	NPHv	NPVv	NPVh	dBm at antenna terminals
Reflectivity	NZHh	NZHv	NZVv	NZVh	dBZ

These quantities can be combined to compute the following:

Z_{DR} = DZHV = 10 log NIHH - 10 log NIVV

LDR_{hv} = DIVhVv = 10 log NIVh - 10 log NIVv

DDV = DVVH = (VPVV - VPHH)/sin(elev)

Of course, numerous other combinations could be computed as well, if so desired.

Using the previously described algorithms, Φ_{DP}, R and $\rho_{hv}(0)$ can be calculated. The existing in-house algorithm can be used to compute K_{DP}.

The list of 32 quantities available for display is as follows:

Table 4. Display Variables.

Description					**Units**
Velocity	VPHh	VPHv	VPVv	VPVh	m/s
Width	WPHh	WPHv	WPVv	WPVh	m^2/s^2
correlation	CPHh	CPHv	CPVv	CPVh	none
intensity (power at rcvr output)	NIHh	NIHv	NIVv	NIVh	watts
power (power at antenna terminals)	NPHh	NPHv	NPVv	NPVh	dBm
reflectivity	NZHh	NZHv	NZVv	NZVh	dBZ
differential phase, raw	PHIdp				deg
differential phase, smoothed	PHIdps				deg
differential propagation constant	Kdp				deg/km
rainfall rate	Rain				mm/hr
correlation coefficient, H and V	RHOHV				none
differential reflectivity	DZHV				dB
linear depolarization ratio	DIVhVv				dB
differential Doppler velocity	DVVH				m/s

5. SUMMARY

Based on the preceding information, equipment capabilities, and the nature of our work, the dual receiver mode was implemented. The Radar Timing Generator was enhanced to control transmit and receive polarization by adding two eight-bit shift registers to control the transmit and receive sequences independently. Arbitrary Waveform Generators were used to simulate receiver signals and check the implementation of the algorithms.

REFERENCES

Sachidananda, M., and D. S. Zrnic, 1989: Efficient processing of alternately polarized radar signals. *J. Atmos. Oceanic Technol.*, **6**, 1310-1318

Jameson, A. R., 1991: A comparison of microwave techniques for measuring rainfall. *J. Appl. Meteor.*, **30,** 32-54.

Zahrai, A., and D. S. Zrnic, 1993: The 10-cm-wavelength polarimetric weather radar at NOAA's National Severe Storms Laboratory. *J. Atmos. Oceanic Technol.*, **10**, 649-662.