EchoGuard™

Security and Surveillance Radar

Developer Manual for Model Numbers:

EchoGuard: 700-0005-203-100 (white radome), **700-0005-203-200** (tan radome),
700-0005-206-300 (green, convective heatsink)

EchoGuard-INTL: 700-0005-206-100 (white radome), **700-0005-206-200** (tan radome)

EchoGuard-CR: 700-0005-203-110 (white radome)

Release Date: 2025 – August

Doc # 700-0005-461 Rev26 Released with SW 18.1

ECHODYNE CORP
12112 115TH AVE NE, SUITE A
KIRKLAND, WA USA 98034
Corporate Phone: +1 (425) 454-3246
Corporate Email: info@echodyne.com
Corporate Web: www.echodyne.com
Tech Support: support@echodyne.com

ECHODYNE

TABLE OF CONTENTS

1. VERSION RELEASE NOTES

REV DATE	CHANGE NOTES & HIGHLIGHTS	AUTHOR
2025-July	**Release 18.1 – New NET user command.** • Added **net:port23enable** command to the list of NET User Commands. The **net:port23enable** command disables port 23. This new command results in additions to all SYSPARAM queries. This command/setting should be persistent such that it persists through power cycles. **New command to enable/disable CFAR:** • Added new command to enable/disable CFAR (Constant Fale Alarm Rate), measurements, and track processing. Default value for RSP:CFAR:ENABLE is now set to "true" such that CFAR processing is enabled. This is a particularly useful feature for customers who leverage RV Map data. When set to "false", CFAR is disabled – resulting in no Detect, Measurements, or Track packets. To reset the parameter back to an enabled state, the RESET:PARAMETER is required. You can also reset the parameter using CFAR: RSP:CFAR:ENABLE <pswd> true. **Fixes/Updates:** • Added additional warnings to the "Regulatory and Safety Notifications" section. • Changed Detect-and-Avoid (DAA) references to Airborne. • Moved TIME:CHANNEL:PWRTHRESHOLD from "SUPER-USER COMMANDS" section to its correct status (as an engineering command) in the "USER COMMANDS" section. • Performed minor copyedits to clean up minor grammatical errors and ensure clarity for technical content.	AN
Dec-2024	**Release 18.0 – New Dynamic Clutter Mask feature and new command port.** • Dynamic Clutter Mask: Dynamic Clutter Mask is a new feature to support over water collects by masking out clutter from waves. • New command port: Add new command port number 29978 while continuing support for the existing port 23. • Electrical specifications were corrected. • A new TCM state was added to support improved TCM performance. • Known performance issue: For the B1b waveform, when RV Maps are enabled, there can be dropped packets if the host PC isn't air gapped or if there is a lot of ethernet traffic.	DG
2024-July	**Release 17.1.4 – Added Enable Keep-Alive Parameter** The desire to add a User setable parameter (Super user permissions) to set the keep-alive timeouts on the radar. Today this compile time parameter is set to 15 seconds. **This release adds keep alive timeout to the radar and allows configuration of the keep alive count (number of probes). This corrects a bug found with release 17.1.2 which prevented reconnection to the radar following a network disruption.**	DG
2024-Mar	**NOTE: Release 17.1.2 is no longer recommended for use. See 17.1.4 change notes.** **Release 17.1.2 – Support for revised hardware configuration** This release adds support for upcoming changes to the hardware configuration on new radars that are transparent to the radar user. Release 17.1.2 also supports all hardware configurations supported by Release 17.0. Once new hardware makes its way to forward	CD, DG

	production, customers must use at minimum SW17.1.2 as the new HW is not recognized by older SW versions. • Section 4.6.1 was updated to show SW17.1.2 compatibility with the latest off radar SW versions. • SDK Section was updated to remove support for Ubuntu 18.04 and add support for Ubuntu 22.04. • Replaced 1Gb References to 1000Mb references throughout document. • Added shielded cabling comments to the "Physical" sub section of section SYSTEM SETUP • Replaced "Echoguard UI" with "RadarUI" throughout the document.	
2023-Sept	**Release 17.0 – Track-Level Classifier** • Track-level Classifier: The track classifier has been expanded to 100% of full tracking range, and now includes the following target types: ○ Drone/UAS ○ Plane ○ Bird ○ Vehicle ○ Person ○ Clutter ○ Other • Enabling extended track packets no longer requires a password. Extended track packets are still not enabled by default natively. However, extended track packets ARE automatically enabled by default within RadarUI when using RadarUI v4.0 and later. • SDK Updates: ○ Added install scripts for Windows and Linux to simplify installation ○ Added Linux-compatible executable of RadarIO • RadarIO Updates: ○ Report IO time in prompt ○ Write prompt history file to user home directory instead of current working directory ○ EchoK 17.0 based schema ○ Fix EchoK rvmap range and velocity bin ordering in parsed data-sets ○ Rename EchoK rvmap fields '0_range_bin_num' / '0_velocity_bin_num' -> 'zero_range_bin_num' / 'zero_velocity_bin_num' • Release 17.0 and beyond is not supported on radars with serial numbers ending in -000XXX due to an antenna hardware change (produced earlier than March 2020). Radars with -00XXX serial numbers ARE supported. If you have a radar with a serial number ending in -000XXX and would like to upgrade, please contact Echodyne. • Re-arranged manual sections for clarity **Known issues/Limitations:** • Using an ethernet interface slower than 1 Gb may result in data drops for RVmaps. This accumulation of errors will back up the network interface and cause partial messages to be sent out. • Internal RVMap buffer has an issue if MODE:SEARCH:START and MODE:SEARCH:STOP commands are issued too quickly in succession. Add a minimum 50 msec delay between START and STOP commands.	CD
2023-Mar	**Release Group 16.4.0 – 3D128 PRISM Channelization** • SW 16.4.0 is an interim / 'dot' release, so this software update does not include major new radar or tracking features nor breaking API changes.	SWS

	• **PRISM channel support added for 3D128 (aka 'B1b') alternate waveform**. 3D128 has twice the velocity resolution, twice the MUV and half the instrumented range, is optimized for slow walkers and offers 3dB additional processing gain. You can now use Frequency, TCM and PRISM channelization with this waveform. • **RANGE:MASK command is now dynamically adjustable** so one can change the maximum radar tracking range while operating in SWT mode. • **Added NOISEFLOOR suite of commands** to calculate summary RVmap noise floor values over range and angle. May be useful to determine if the noise floor is elevated in a particular installation. **Fixes:** • RCS field is now correctly reported in detection packet while running in Search mode with custom Search List. • MODE:SWT:TRACKER:POBJ_INIT in SYSPARAM and SYSPARAM:JSON now correctly print out as a float. • Clarified in Developer Manual that ZONE:MASK:CLEAR sets ZONE:MASK:ENABLE to False in addition to removing previously configured zone masks. **Known issues/Limitations:** • Using an ethernet interface slower than 1 Gb may result in data drops for RVmaps. This accumulation of errors will back up the network interface and cause partial messages to be sent out. • Internal RVMap buffer has an issue if MODE:SEARCH:START and MODE:SEARCH:STOP commands are issued too quickly in succession. Add a minimum 50 msec delay between START and STOP commands.	

1.1 Acuity Suite: Firmware & Software Versions for SW Suite

To assist users in decoding software versions and approved combinations of the Acuity software suite, we have included the following "cheat sheet." Please ensure that both the HPS and matching FPGA are updated to the software version you intend to use.

Supported Embedded Firmware Versions (released in last year)					
SW Suite	**Release Date**	**HPS #**	**HPS Release Number (Documentation)**	**FPGA Waveform #**	**FPGA Release Number (Documentation)**
16.1.1	Oct 2020	20.7.D.0.6	521-0013-921_A_Rev21	9514 (A5) 'B1a' 9D14 (A6) 'B1a'	521-0013-921_F_Rev4 521-0013-921_G_Rev3
		700-0005-921_E_Rev6 (combined HPS/FPGA .zip)			
16.2.0	Dec 2020	21.7.D.1.0 (33.1.0)	521-0013-921_A_Rev22	9516 (A5) 'B1a' 9D16 (A6) 'B1a'	521-0013-921_F_Rev5 521-0013-921_G_Rev4
		700-0005-921_E_Rev7 (combined HPS/FPGA .zip)			
16.3.0	Jul 2021	33.2.1	521-0013-921_A_Rev23	9516 (A5) 6D32 'B1a' 9D16 (A6) 6D32 'B1a'	521-0013-921_F_Rev5 521-0013-921_G_Rev4
		700-0005-921_E_Rev8 (combined HPS/FPGA .zip for 6D32 'B1a' waveform)			
		33.2.1	521-0013-921_A_Rev23	8519 (A5) 3D128 'B1b' 8D19 (A6) 3D128 'B1b'	521-0013-921_F_Rev2 200004-001_Rev1
		700-0005-921_F_Rev2 (combined HPS/FPGA .zip for 3D128 'B1b' waveform)			
16.4.0	May 2023	33.6.0	200006-003_Rev01	9524 (A5) 6D32 'B1a' 9D24 (A6) 6D32 'B1a'	200009-002_Rev01 200010-002_Rev01
		200013-003_Rev01_EchoGuard_B1a-6D32_SW16.4.0_Suite.zip (combined pkg)			

		33.6.0	200006-003_Rev01	8524 (A5) 3D128 'B1b' 8D24 (A6) 3D128 'B1b'	200011-002_Rev01 200004-003_Rev01
		200014-003_Rev01_EchoGuard_B1b-3D128_SW16.4.0_Suite.zip (combined pkg)			
17.0	Sept 2023	34.5.0	200006-004_Rev01	9D24 (A6) 6D32 'B1a'	200010-002_Rev01
		200013-004_Rev01_ECHOGUARD_B1a-6D32_SW17.0_SUITE			
		34.5.0	200006-004_Rev01	8D24 (A6) 3D128 'B1b'	200004-003_Rev01
		200014-004_Rev01_EchoGuard_B1b-3D128_SW17.0_Suite			
17.1.4	July 2024	34.5.8	200006-010_Rev01	9D26 (A6) 6D32 'B1b'	200010-003_Rev 02
		200013-010_rev01_EchoGuard_B1a-6D32_SW17.1.4_Suite			
		34.5.8	200006-010_Rev01	8D26 (A6) 3D128 'B1b'	200004-004_Rev 02
		200014-010_rev01_EchoGuard_B1b-3D128_SW17.1.4_Suite			
18.0.4	Dec 2024	35.8.5	200006-008_Rev05	9D26 (A6) 6D32 'B1a'	200010-003_Rev 02
		200013-008_rev05_EchoGuard_B1a-6D32_SW18.0.4_Suite			
		35.8.5	200006-008_Rev05	8D26 (A6) 3D128 'B1b'	200004-004_Rev 02
		200014-008_rev051_EchoGuard_B1b-3D128_SW18.0.4_Suite			
18.1.5	Aug 2025	35.9.5	200006-011_Rev06	9D26 (A6) 6D32 'B1a'	200010-003_Rev 02
		200013-011_Rev06 _EchoGuard_B1a-6D32_ SW18.1.5_Suite			
		35.9.5	200006-011_Rev06	8D26 (A6) 3D128 'B1b'	200004-004_Rev 02
		200014-011_Rev06 _EchoGuard B1b-3D128 SW18.1.5_Suite			

2. INTRODUCTION & BACKGROUND

2.1 Introduction to EchoGuard MESA Radar

Echodyne's MESA (Metamaterial Electronically Scanning Array) is an innovative new radar architecture that does not require the moving parts of dish antenna systems, nor the phase shifters of traditional phased array antennas. This simplicity makes EchoGuard accessible to a broad range of applications, even platforms which may be highly C-SWAP (Cost, Size, Weight, and Power) constrained.

EchoGuard™ Security and Surveillance Radar (SSR) is uniquely designed to enable multiple missions such as perimeter security for border or critical infrastructure applications, counter-UAS (cUAS) detection and tracking of intruders and Ground-Based Detect and Avoid (GB-DAA) applications. It can provide the same level of safe operation for small UAS vehicles too small to carry their own radar system, by providing localized situational pilot awareness of both own-ship position, along with cooperative and non-cooperative intruder air vehicles. Multiple EchoGuard units may be combined into flexible local networks for increased coverage.

2.2 Regulatory and Safety Notifications

The following warnings should be observed.

GENERAL WARNING: Read and understand all following icons and recommendations.

WARNING: Radar generates non-ionizing radiation when operating. See RF MPE exposure guidance below for keep out zone dimensions.

WARNING: Hot surface temperatures may be present during and after operation.

PROHIBITION: Do not touch radar surface during or recently after operation to avoid burns.

2.2.1 COUNTRY:MODE USA - FCC ID

Echodyne has received a Grant of Equipment Authorization from the FCC for the EchoGuard radar operating in the 24.45 to 24.65 GHz span. Units are clearly marked with the following label indicating safe operational distance and the official FCC ID number.

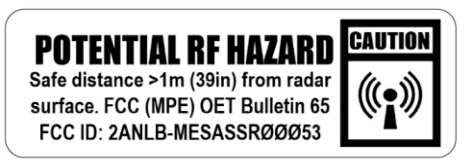


Figure 1: FCC ID Label

2.2.1.1 RF Radiated Emission Hazard

For safe operation, the radar should be positioned such that closest proximity to operators, or the general population is greater than 1 meter (39 inches) radial arc over the 120˚ azimuth and 80˚ elevation scanning Field of View (FoV).

This guideline meets FCC Requirements for Radio Frequency Exposure for EchoGuard, which contains a transmitter with a maximum power of 4 watts delivered to a scanning MESA array antenna with an absolute maximum gain of 22 dBi. Do not transmit when persons are within the MPE radius of the radar. The MPE radius is the minimum distance from the antenna axis that humans should maintain to avoid RF exposure higher than the allowable MPE level set by FCC.

The US radiolocation band radars have a primary operating frequency centered at 24.55 GHz with maximum transmit power under 4 watts. MPE (Maximum Permissible Exposure) – The FCC (Federal Communications Commission) specifies MPE differently by device and frequency band. While the transmit power is like other low power radios that humans encounter (cellular phones, handheld radios, CB radios etc.) there is slightly more risk due to the focused beam of the radar. A focused beam with a nominal gain of 21 dBi and an absolute maximum gain of 22dBi (Factor of 160) multiplies the effective power radiated. To address this, Echodyne has reviewed the FCC MPE as described in FCC OEP Bulletin 65 and used the principles and guidelines to develop a conservative safe operating distance to recommend. This is exactly the process used by marine radar manufacturers to recommend safe operational distances for humans.

WARNING: IT IS THE RESPONSIBILITY OF THE RADAR OPERATOR TO ENSURE THAT THE MAXIMUM PERMISSIBLE EXPOSURE LIMITS ARE OBSERVED AT ALL TIMES DURING RADAR TRANSMISSION.

2.2.1.2 Burn Hazard

IF the user fails to follow warning and instruction manual directions on handling the Radar, THEN there is the potential for skin burns when operating the Radar in high temperature environments (MIL-STD-810H A1 Climate Zones).

Regarding the heatsink, in a hot and dry environment, the heatsink delta temperature is not enough to burn skin. For more information, please refer to MIL-STD-810.

2.2.1.3 Cut Hazard

IF the user fails to follow warning and instruction manual directions concerning handling the Radar, THEN there is the potential for a finger or hand cut injury due to metal edges and corners.

2.2.1.4 Electric Shock Hazard

IF the user fails to follow warning and instruction manual directions, THEN there is the potential for electrocution.

2.2.1.5 Installation – Falling Object Hazard

IF the system installer fails to warning and instruction manual directions concerning system installation, THEN there is the potential for a Radar to fall injuring or killing a bystander on the ground.

2.2.1.6 Installation – Lift Hazard

IF the system installer fails to follow warning and instruction manual directions on proper lifting, THEN there is the potential for injury to the individual due to improper lifting of equipment.

2.2.1.7 Non-ionizing Electro-magnetic (EM) Radiation (in the form of microwaves and radio frequency) Emissions Hazard

IF the user fails to follow warning and instruction manual directions on human proximity, THEN there is the potential for high-energy radio waves and microwaves to cause destructive heating of biological tissue, blindness, and superficial skin burns.

Harmful effects of long-term exposure to moderate-to-intense radio-frequency (RF) fields and ELF fields are still being debated.

2.2.2 COUNTRY:MODE INTL – CE

For International operation (non-FCC), MESA radars have a primary operating frequency centered at 24.15 GHz with maximum transmit power under 4 watts and an operation frequency band of 24.05 to 24.25 GHz with three sub-channels. For information regarding availability and CE Declaration of Conformity, contact Echodyne. Datasheet specifications for EchoGuard International model can be found in section 3.3.

2.2.3 EchoGuard International Shipping Limitations

The US Commerce Department limits export of certain commodities to specific countries. Based on the product type, Echodyne has determined it is contained within ECCN #6A008.e for export licensing. For additional information, please contact Echodyne.

2.3 Radar Package Contents

Table 1: Radar Package Contents

Item	Description	Purpose
1	EchoGuard Radar	
2	Adapter Cable (purchased separately)	Cable which breaks out the precision circular snap lock radar interface multi-pin connector into separate DC power (Molex) and Ethernet (RJ45) connectors.
3	Molex plug pigtail	Plugs into Radar Adapter Cable to allow user integration into system power. USER Pig tail: MOLEX 43640-050 with crimp terminal 43031 plug contacts.
4	Digital Library	Available at Echodyne FTP site or customer portal. ***Contents:*** • Digital Version of Manual • Latest radar firmware • Radar Update Utility • Sample Radar Test Data • RadarUI • B-NET command executable & API • Example MATLAB Scripts • Mechanical ICD and CAD Model

2.3.1 MOLEX Pigtail Information

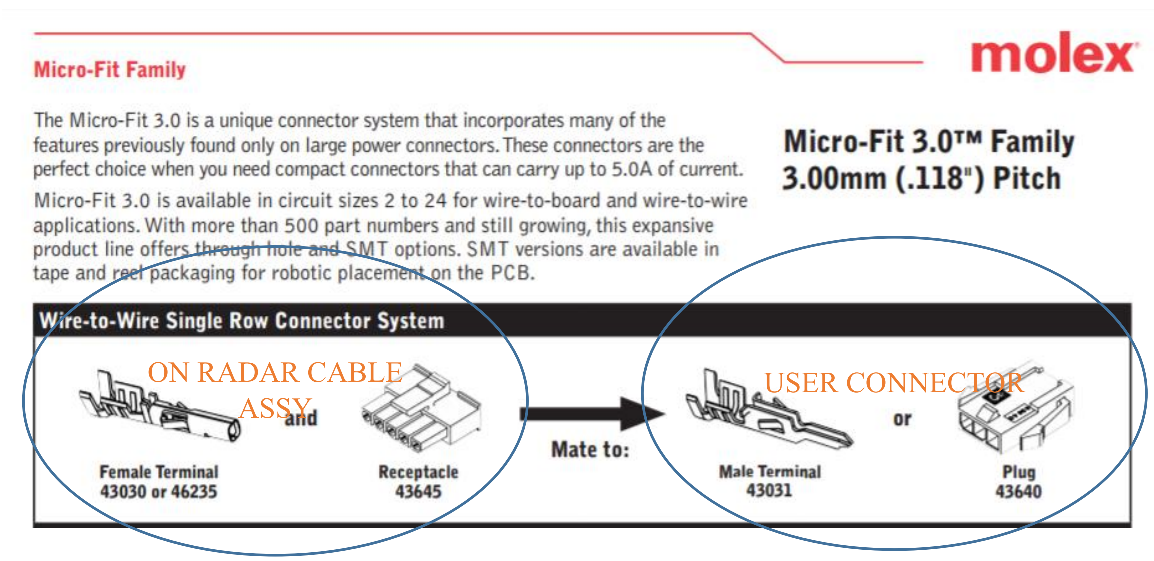


Figure 2: Cable connector for power & reset control

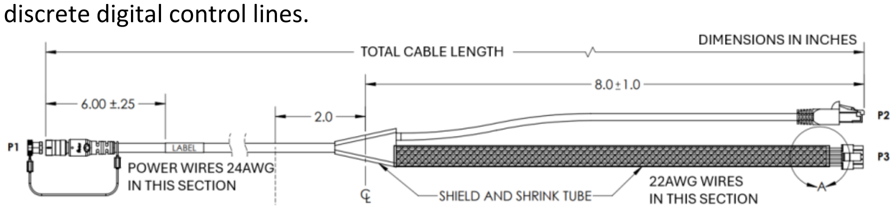
3. SYSTEM SPECIFICATIONS & PERFORMANCE

3.1 EchoGuard

Classic EchoGuard radar configuration SKUs:

- **700-0005-203-100** (white radome, black chassis, silver heatsink)
- **700-0005-203-200** (tan radome, black chassis, tan heatsink)

Table 2: EchoGuard Specifications

Note: Specifications subject to change without notice.	
INTERFACES	
DC/Control Cable Assembly	The radar is powered and controlled through a single Fischer MiniMax IP-68 watertight snap lock circular connector and custom cable harness. Only the snap lock radar end (P1) is watertight. This cable assembly is purchased separately from the Radar SKU. It is intended that the user would route the non-watertight P2/P3 inside a watertight enclosure for C2 integration. P1 – (Radar-End) Snap Lock watertight 12 pin connector. P2 – (User End) standard RJ45 shielded non-watertight Gbit Ethernet P3 – (User End) Molex 43645-0500 5 socket contact connector for DC power and discrete digital control lines. 
DC Power	DC Power is connected through the P3 Molex connector of the cable harness and is routed to the P1 radar interface connector. P3 Connections: PIN 1 - RED Wire +VDC 22 AWG PIN 2 – BLACK Wire -VDC RTN 22 AWG Input Voltage (V), full radar operation (Note 1): +15 to + 32 VDC, +24 VDC nominal Input Voltage (V), IDLE, limited operations (system communications and status, no RF output): +12 to + 15 VDC Transmit mode power: ≤ 50W IDLE mode power: ≤ 8W EchoGuard Current = 50 Watts/$V_{OPERATING}$ Notes: 1. Voltage requirements listed are as "delivered to the unit at the P1 radar interface." Please consider any tolerances in the power source and voltage drop in power cables (the Fischer cables use 24 AWG power wires).
Discrete Control Digital Lines	Discrete 3.3V TTL Control Wires PIN 3 – ORANGE – RESET_N (PIN2-BLK GND Hold 10 sec at bootup =Factory Reset) PIN 4 – GREEN – CTRL2 (Designed for future Use) PIN 5 – WHITE – Digital Shield Ground

	Note: These need not be connected at all in the base model radar. They should be taped off and insulated well to prevent damage. PIN 3 may be configured to reset the unit to factory configuration. Contact Echodyne Applications engineering for specific use case.
RF Transmit Power	Maximum RF power emission under all conditions < 4 watts. RF safe distance from aperture under operating conditions: >1m for FCC Maximum Human Exposure (MPE) per FCC Bulletin 65
PERFORMANCE	
Radar parameters	
Operation center frequency	24.45-24.65 GHz (Default – FCC Radiolocation Band)
Operating bandwidth	Baseline Configuration: Channel A1-A 24.4675 to 24.5125 GHz (Fc=24.49 GHz / 45MHz swept BW) Channel A1-B 24.5275 to 24.5725 GHz (Fc = 24.55 GHz / 45MHz swept BW) Channel A1-C 24.5875 to 24.6325 GHz (Fc = 24.61 GHz / 45MHz swept BW) Note: NTIA spectrum mask allows for about 15MHz overshoot on each edge.
Frequency Accuracy	± 10ppm (± 250kHz) including temperature and aging 8 years
Waveforms	6D32 (B1a) Waveform: Saw tooth consisting of 32 downward chirps across 45MHz swept BW with 200 µsec/10 µSec chirp timing (210 µsec PRI) 3D128 (B1b) Waveform: Saw tooth consisting of 128 downward chirps across 45MHz swept BW with 100 µsec/10 µSec chirp timing (110 µsec PRI)
Transmitted Power	EchoGuard 3.2 Watts CW (35 dBm) Blanking (~-50 dB) when not in use Transmitter only enabled during valid radar trigger event. Transmitter not enabled during BIT fault or bootup until issuing START
Aperture Characteristics	
Polarization	Linear, Horizontal
Cross-pol	Typically, 20 dB
AZ/EL (HPBW), typical	2° horizontal (E-plane, at broadside), 6° vertical (H-plane, at broadside) Note: Beam width increases 30-50% at FOV edges.
Field of regard Azimuth/Elevation	± 60° / ± 40°
Beam Step resolution/accuracy	2° Resolution, ≤1° pointing accuracy
Realized Gain at broadside	21 dBi typ. (safety absolute maximum gain of 22 dBi)
Gain average roll-off	2.0 dB typical over full field of regard
Side lobe level	Two-way radar Sidelobes Center (< ±40° Az or El) SLL -32 dB average Edges SLL -28 dB average
Next-beam buffer time	< 100 microsecs (serial write time)
Beam-to-beam transition time	< 1 microsecs (after enable)
Object Detection parameters	
Object RCS	-30 to +100 dBsm (Approximate User settable window – See RCS commands) 0dBSm @ >2500m @ 90% $P_D/10^{-6}$ P_{FA} Typical
Range	Minimum 20 meters, Maximum 5987 meters operational
Range Resolution	3.25 m typ.
Velocity Resolution	0.91 m/s typ.
Angular Resolution	±1° Az and ±3° El in Search Mode, smaller for tracked targets
Search & Track	
Search	User configurable scan volume
Track acquisition	New tracks acquired in < 1 sec
Track updates	5-10 updates /sec (for each track) depending upon Operation Mode
Max Tracks	20 simultaneous tracks dynamically allocated via i-SCAN

Modes	Search & Search-While-Track
General Physical	
External Finish	1. Anodized Aluminum Body 2. Radome: Plastic polycarbonate with color paint topcoat
Size (Packaged)	18.7cm x 16.2cm x 5.1cm
Weight (Packaged)	EchoGuard < 1250 grams fixed install natural convection (no-fans)
Mechanical Interface	EchoGuard has 75mm and 100mm square VESA mounting holes using standard M-4 screws. See – Mechanical ICD 700-0005-203
ENVIRONMENTAL	
Operational Temp	-40 ° C to 75 °C Continuous Operation
Storage Temp	-55 ° C to 95 °C (Non-Operational)
Humidity (operation)	< 95 % non-condensing
Moisture Resistance	Main Radar package is an IP67 moisture and dust protected package as delivered. It has a Gore vent gasket that supports air transport pressure changes
Fungus	None Specified
Salt Fog	None Specified
Sand & Dust	Tolerant with protective cap or connector attached.
MTBF	110,000 hours, per MIL-STD-217, Ground Fixed environment
Operational altitude	0-30,000 ft (0 to 9.1 km) AGL
Shock	None Specified
Vibration	Typical mounting on fixed structures requires no special mechanical isolation mounts. Care should be exercised with mount designs to avoid resonant modes in the 100-3000 Hz range with large deflections as this could potentially generate false Doppler signatures and impact angular pointing accuracy.
Prime Power	1. Maximum ripple provided to radar on prime power required to be less than 200 mV P-P 2. Ramp-Up Prime power voltage < 50 msec
ESD	Handle with normal sensitivity (Standard HBM)
Lightning	None Specified
REGULATORY	
FCC	FCC ID: 2ANLB-MESASSR00053 Emission Designator: 45M0FXN

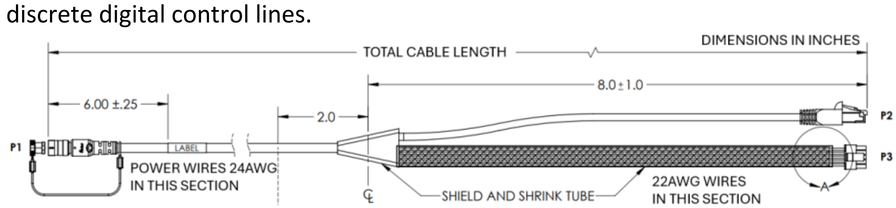
3.2 EchoGuard-CR

Reduced range EchoGuard-CR radar configuration SKUs:

- **700-0005-203-110** (white radome, black chassis, silver heatsink)

The EchoGuard-CR is a short range (~1/4th) variant of the EchoGuard 'classic'. By reducing the number of transmit power amplifiers utilized from 2 to 1, lowering the transmit power operating point and modifying the software processing, a radar with shorter range and reduced power consumption is achieved.

Table 3: EchoGuard-CR Specifications

Note: Specifications subject to change without notice.	
INTERFACES	
DC/Control Cable Assembly	The radar is powered and controlled through a single Fischer MiniMax IP-68 watertight snap lock circular connector and custom cable harness. Only the snap lock radar end (P1) is watertight. A cable assembly is provided with each radar and additional cables may be purchased as needed. It is intended that the user would route the non-watertight P2/P3 inside a watertight enclosure for C2 integration. P1 – (Radar-End) Snap Lock watertight 12 pin connector. P2 – (User End) standard RJ45 shielded non-watertight Gbit Ethernet P3 – (User End) Molex 43645-0500 5 socket contact connector for DC power and discrete digital control lines. 
DC Power	DC Power is connected through the P3 Molex connector of the cable harness and is routed to the P1 radar interface connector. P3 Connections: PIN 1 - RED Wire +VDC 22 AWG PIN 2 – BLACK Wire -VDC RTN 22 AWG Input Voltage (V), full radar operation (Note 1): +12 to + 32 VDC, +24 VDC nominal Transmit mode power: ≤ 24W IDLE mode power: ≤ 8W EchoGuard Current = 24 Watts/$V_{OPERATING}$ Notes: 1. Voltage requirements listed are as "delivered to the unit at the P1 radar interface." Please consider any tolerances in the power source and voltage drop in power cables (the Fischer cables use 24 AWG power wires).
Discrete Control Digital Lines	Discrete 3.3V TTL Control Wires PIN 3 – ORANGE – RESET_N (PIN2-BLK GND Hold 10 sec at bootup =Factory Reset) PIN 4 – GREEN – CTRL2 (Designed for future Use) PIN 5 – WHITE – Digital Shield Ground

	Note: These need not be connected at all in the base model radar. They should be taped off and insulated well to prevent damage. PIN 3 may be configured to reset the unit to factory configuration. Contact Echodyne Applications engineering for specific use case.
RF Transmit Power	Maximum RF power emission under all conditions < 4 watts. RF safe distance from aperture under operating conditions: >1m for FCC Maximum Human Exposure (MPE) per FCC Bulletin 65
PERFORMANCE	
Radar parameters	
Operation center frequency	24.45-24.65 GHz (Default – FCC Radiolocation Band).
Operating bandwidth	Baseline Configuration: Channel A1-A 24.4675 to 24.5125 GHz (Fc=24.49 GHz / 45MHz swept BW) Channel A1-B 24.5275 to 24.5725 GHz (Fc = 24.55 GHz / 45MHz swept BW) Channel A1-C 24.5875 to 24.6325 GHz (Fc = 24.61 GHz / 45MHz swept BW) Note: NTIA spectrum mask allows for about 15MHz overshoot on each edge.
Frequency Accuracy	± 10ppm (± 250kHz) including temperature and aging 8 years
Waveforms	6D32 (B1a) Waveform: Saw tooth consisting of 32 downward chirps across 45MHz swept BW with 200 µsec/10 µSec chirp timing (210 µsec PRI) 3D128 (B1b) Waveform: Saw tooth consisting of 128 downward chirps across 45MHz swept BW with 100 µsec/10 µSec chirp timing (110 µsec PRI)
Transmitted Power	EchoGuard-CR 0.25 Watts CW (24dBm) Blanking (~-50 dB) when not in use Transmitter only enabled during valid radar trigger event. Transmitter not enabled during BIT fault or bootup until issuing START
Aperture Characteristics	
Polarization	Linear, Horizontal
Cross-pol	Typically, 20 dB
AZ/EL (HPBW), typical	2° horizontal (E-plane, at broadside), 6° vertical (H-plane, at broadside) Note: Beam width increases 30-50% at FOV edges.
Field of regard Azimuth/Elevation	± 60° / ± 40°
Beam Step resolution/accuracy	2° Resolution, ≤1° pointing accuracy
Realized Gain at broadside	21 dBi typ. (safety absolute maximum gain of 22 dBi)
Gain average roll-off	2.0 dB typical over full field of regard
Side lobe level	Two-way radar Sidelobes Center (< ±40° Az or El) SLL -32 dB average Edges SLL -28 dB average
Next-beam buffer time	< 100 microsecs (serial write time)
Beam-to-beam transition time	< 1 microsecs (after enable)
Object Detection parameters	
Object RCS	-30 to +30 dBsm (Approximate User settable window – See RCS commands) 0dBSm @ >2500m @ 90% $P_D/10^{-6}$ P_{FA} Typical
Range	Minimum 10 meters, Maximum 1200 meters operational
Range Resolution	3.25 m typ.
Velocity Resolution	0.91 m/s typ.
Angular Resolution	±1° Az and ±3° El in Search Mode, smaller for tracked targets
Search & Track	
Search	User configurable scan volume
Track acquisition	New tracks acquired in < 1 sec
Track updates	5-10 updates /sec (for each track) depending upon Operation Mode

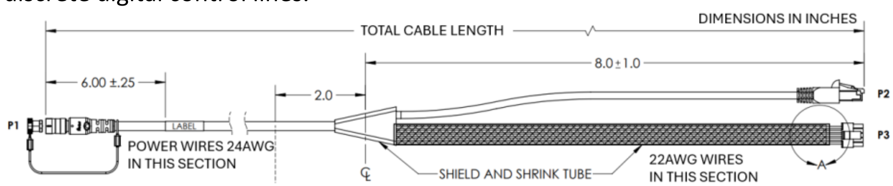
Max Tracks	20 simultaneous tracks dynamically allocated via i-SCAN
Modes	Search & Search-While-Track
General Physical	
External Finish	Anodized Aluminum Body Radome: Plastic polycarbonate with color paint topcoat
Size (Packaged)	18.7cm x 16.2cm x 5.1cm
Weight (Packaged)	EchoGuard < 1250 grams fixed install natural convection (no-fans)
Mechanical Interface	EchoGuard has 75mm and 100mm square VESA mounting holes using standard M-4 screws. See – Mechanical ICD 700-0005-203
ENVIRONMENTAL	
Operational Temp	-40 ° C to 75 °C Continuous Operation
Storage Temp	-55 ° C to 95 °C (Non-Operational)
Humidity (operation)	< 95 % non-condensing
Moisture Resistance	Main Radar package is an IP67 moisture and dust protected package as delivered. It has a Gore vent gasket that supports air transport pressure changes
Fungus	None Specified
Salt Fog	None Specified
Sand & Dust	Tolerant with protective cap or connector attached.
MTBF	110,000 hours, per MIL-STD-217, Ground Fixed environment
Operational altitude	0-30,000 ft (0 to 9.1 km) AGL
Shock	None Specified
Vibration	Typical mounting on fixed structures requires no special mechanical isolation mounts. Care should be exercised with mount designs to avoid resonant modes in the 100-3000 Hz range with large deflections as this could potentially generate false Doppler signatures and impact angular pointing accuracy.
Prime Power	Maximum ripple provided to radar on prime power required to be less than 200 mV P-P Ramp-Up Prime power voltage < 50 msec
ESD	Handle with normal sensitivity (Standard HBM)
Lightning	None Specified
REGULATORY	
FCC	FCC ID: 2ANLB-MESASSR00053 Emission Designator: 45M0FXN

3.3 EchoGuard-INTL

EchoGuard International radar configuration SKUs:

- **700-0005-206-100** (white radome, black chassis, silver heatsink)
- **700-0005-207-100** (white radome, black chassis, cold plate)

Table 4: EchoGuard-INTL Specifications

Note: Specifications subject to change without notice.	
INTERFACES	
DC/Control Cable Assembly	The radar is powered and controlled through a single Fischer MiniMax IP-68 watertight snap lock circular connector and custom cable harness. Only the snap lock radar end (P1) is watertight. A cable assembly is provided with each radar and additional cables may be purchased as needed. It is intended that the user would route the non-watertight P2/P3 inside a watertight enclosure for C2 integration. P1 – (Radar-End) Snap Lock watertight 12 pin connector. P2 – (User End) standard RJ45 shielded non-watertight Gbit Ethernet P3 – (User End) Molex 43645-0500 5 socket contact connector for DC power and discrete digital control lines. 
DC Power	DC Power is connected through the P3 Molex connector of the cable harness and is routed to the P1 radar interface connector. P3 Connections: PIN 1 - RED Wire +VDC 22 AWG PIN 2 – BLACK Wire -VDC RTN 22 AWG Input Voltage (V), full radar operation (Note 1): +15 to + 28.8 VDC, +24 VDC nominal Input Voltage (V), IDLE, limited operations (system communications and status, no RF output): +12 to + 15 VDC Transmit mode power: ≤ 50W IDLE mode power: ≤ 8W EchoGuard Current = 50 Watts/$V_{OPERATING}$ Notes: 1. Voltage requirements listed are as "delivered to the unit at the P1 radar interface." Please consider any tolerances in the power source and voltage drop in power cables (the Fischer cables use 24 AWG power wires).
Discrete Control Digital Lines	Discrete 3.3V TTL Control Wires PIN 3 – ORANGE – RESET_N (PIN2-BLK GND Hold 10 sec at bootup =Factory Reset) PIN 4 – GREEN – CTRL2 (Designed for future Use) PIN 5 – WHITE – Digital Shield Ground Note: These need not be connected at all in the base model radar. They should be taped off and insulated well to prevent damage. PIN 3 may be configured to reset the unit to factory configuration. Contact Echodyne Applications engineering for specific use case.

RF Transmit Power	Maximum RF power emission under all conditions < 4 watts. RF safe distance from aperture under operating conditions: >1m for FCC Maximum Human Exposure (MPE) per FCC Bulletin 65
PERFORMANCE	
Radar parameters	
Operation center frequency	24.05-24.25 GHz
Operating bandwidth	Baseline Configuration: Channel A1-A 24.0675 to 24.1125 GHz (Fc=24.09 GHz / 45MHz swept BW) Channel A1-B 24.1275 to 24.1725 GHz (Fc = 24.15 GHz / 45MHz swept BW) Channel A1-C 24.1875 to 24.2325 GHz (Fc = 24.21 GHz / 45MHz swept BW) Note: NTIA spectrum mask allows for about 15MHz overshoot on each edge.
Frequency Accuracy	± 10ppm (± 250kHz) including temperature and aging 8 years
Waveforms	6D32 (B1a) Waveform: Saw tooth consisting of 32 downward chirps across 45MHz swept BW with 200 µsec/10 µSec chirp timing (210 µsec PRI) 3D128 (B1b) Waveform: Saw tooth consisting of 128 downward chirps across 45MHz swept BW with 100 µsec/10 µSec chirp timing (110 µsec PRI)
Transmitted Power	3.2 Watts CW (35 dBm) Blanking (~-50 dB) when not in use Transmitter only enabled during valid radar trigger event. Transmitter not enabled during BIT fault or bootup until issuing START
Aperture Characteristics	
Polarization	Linear, Horizontal
Cross-pol	Typically, 20 dB
AZ/EL (HPBW), typical	2° horizontal (E-plane, at broadside), 6° vertical (H-plane, at broadside) Note: Beam width increases 30-50% at FOV edges.
Field of regard Azimuth/Elevation	± 60° / ± 40°
Beam Step resolution/accuracy	2° Resolution, ≤1° pointing accuracy
Realized Gain at broadside	21 dBi typ. (safety absolute maximum gain of 22 dBi)
Gain average roll-off	2.0 dB typical over full field of regard
Side lobe level	Two-way radar Sidelobes Center (< ±40° Az or El) SLL -32 dB average Edges SLL -28 dB average
Next-beam buffer time	< 100 microsecs (serial write time)
Beam-to-beam transition time	< 1 microsecs (after enable)
Object Detection parameters	
Object RCS	-30 to +100 dBsm (Approximate User settable window – See RCS commands) 0dBSm @ >2500m @ 90% $P_D/10^{-6}$ P_{FA} Typical
Range	Minimum 20 meters, Maximum 5987 meters operational
Range Resolution	3.25 m typ.
Velocity Resolution	0.91 m/s typ.
Angular Resolution	±1° Az and ±3° El in Search Mode, smaller for tracked targets
Search & Track	
Search	User configurable scan volume
Track acquisition	New tracks acquired in < 1 sec
Track updates	5-10 updates /sec (for each track) depending upon Operation Mode
Max Tracks	20 simultaneous tracks dynamically allocated via i-SCAN
Modes	Search & Search-While-Track
General Physical	

External Finish	Anodized Aluminum Body Radome: Plastic polycarbonate with color paint topcoat
Size (Packaged)	18.7cm x 16.2cm x 5.1cm
Weight (Packaged)	EchoGuard < 1250 grams fixed install natural convection (no-fans)
Mechanical Interface	EchoGuard has 75mm and 100mm square VESA mounting holes using standard M-4 screws. See – Mechanical ICD 700-0005-203
ENVIRONMENTAL	
Operational Temp	-40 ° C to 75 °C Continuous Operation
Storage Temp	-55 ° C to 95 °C (Non-Operational)
Humidity (operation)	< 95 % non-condensing
Moisture Resistance	Main Radar package is an IP67 moisture and dust protected package as delivered. It has a Gore vent gasket that supports air transport pressure changes
Fungus	None Specified
Salt Fog	None Specified
Sand & Dust	Tolerant with protective cap or connector attached.
MTBF	110,000 hours, per MIL-STD-217, Ground Fixed environment
Operational altitude	0-30,000 ft (0 to 9.1 km) AGL
Shock	None Specified
Vibration	Typical mounting on fixed structures requires no special mechanical isolation mounts. Care should be exercised with mount designs to avoid resonant modes in the 100-3000 Hz range with large deflections as this could potentially generate false Doppler signatures and impact angular pointing accuracy.
Prime Power	3. Maximum ripple provided to radar on prime power required to be less than 200 mV P-P 4. Ramp-Up Prime power voltage < 50 msec
ESD	Handle with normal sensitivity (Standard HBM)
Lightning	None Specified
REGULATORY	
	CE (Contact Echodyne for Declaration of Conformity)

4. SYSTEM SETUP

4.1 Quick Start Guide

- **COMPUTER INTERFACE NEEDS**
  - Find a modern high-speed computer > 2GHz clock speeds with quad processor.
  - Need one dedicated verified 1000Mb Ethernet port that will support >50Mbytes/sec (400Mbits/sec plus TCP IP overhead). Remember 1 Byte = 8 bits so a Gb Ethernet port is only rated at 1000 Mbits/sec and some ports might struggle at this rate. This speed will only be needed if you intend to grab the full range velocity matrix for all beams. Much lower data rate modes are supported in the radar using only detections and tracks.
  - Need an independent NIC card or Wi-Fi connection to company network/internet connection. The radar does not support DHCP and is a fixed IP address.
  - 16GByte RAM (might be able to go as low as 4GByte RAM if no RVmaps are processed).
  - Hard Drive - Also at full data rate you will need Fast SSD storage that will support >50Mbyte/Sec with at least 128GByte available space. The radar in full resolution mode will fill 3 GB of space every minute of operation saving RVmaps. If you want to try to use an SD card it must be the new fast type made to record 4k video like SANDISK – Extreme-Pro 95Mbyte/sec cards. We caution this as we have had issues with drivers supporting full speed windows file structure.

- **DC POWER NEEDS**
  - >50-watt DC power source capable of delivering between the appropriate input range. It can dip down to the limited operations voltage spec at start up in IDLE mode, but this is allowed while transmitting as it is over the current limit of the connectors. (**Caution** – 50W at 12V is ~4.2A!). We recommend using either a lab supply that you can observe the current on its display, or for field testing a handy solution is a 72 watt 24 Vdc power supply readily available on Amazon fitted with an AC 115V input and a 5.5mm x 2.1mm or 5.5mm x 2.5mm DC plug port for test.
  - We recommend 22 AWG (American Wire Gauge) stranded wire less than 96 inches for power delivery point. If you have longer wiring needs, please size wire appropriately to support the voltage and current load.
- **SOFTWARE**
  - Using BNET – This is a quick way to get familiar with radar communication. In the Digital Library is both an Executable complied W10 compatible program to issue manual commands. This is the path to software up-grades also. Included is both the compiled executable BNET and C++ un-compiled version to use as a foundation for developing your own communication solutions in your language of choice.
  - MATLAB Scripts – Another quick way to start is to utilize the Echodyne provided communication MATLAB script example code with associated *.mex files. These have been verified and are a good place to start even if you intend to re-create a script in an alternate language. You will need your own MATLAB license to read and execute the provided scripts. Included in the Digital Library is a text-based list of MATLAB code that is readable (no-execution) that will help you convert without a MATLAB license. Contact Echodyne for additional script generation support.
  - The RadarUI is also a great way to get the radar up and running quickly. This is a compiled Qt/C++ application that allows easy configuration and data plotting on a 2D map.

- **PHYSICAL**
  - The Y-cable providing the DC feed and Ethernet communication is a shielded cable. Shielded cables are always recommended to provide full electromagnetic immunity from outside sources effecting EchoGuard operation. It is recommended to connect the Ethernet cable shield (or DC cable shield) to Earth Ground at the end of the Echodyne supplied Radar Interface Cable when that cable length is less than 5m.
  - We recommend a quality SLR camera-grade tripod and a mechanical bracket that will support the radar in your field test configuration.
  - We offer an Echodyne test kit with tripod adapter, power supply, camera mount and various useful items to help with first level field tests. These can be shipped quickly if you decide you need this.

DO NOT POWER THE UNIT YET! – If you are in a hurry – READ THE SAFETY WARNINGS in section 2.2. Follow the test diagram and power up sequence at the end of this section.

4.2 Physical Mounting

The EchoGuard has a large back plate for improved passive cooling. Embedded in this back plate is a square hole pattern for VESA 75mm and VESA 100mm mounting brackets as well as an alternate three-hole AMP circular mounting pattern. Only one of these patterns need be used with standard industry brackets. Screws should be standard M-4 x 0.7mm hardware and torqued to 1.3 Nm (10 in-lbs.) for dry hardware in threaded steel inserts.

When mounting to a bracket or other physical installation, observe the coordinate system shown below (also laser engraved on top of radar) as it is tied to the internal coordinate reference system defined by Echodyne for beam scheduling and target tracking. The coordinate system defined for Echodyne radars is a standard Azimuth (AZ) / Elevation (EL) system with antenna coordinates defined in X/Y/Z coordinates. The official ICD provides a reference coordinate system anchoring the physical dimensions to the unit physical features. Also, for successful mounting, remember that the radar cable connector (J1, round, 12-pin Fischer connector) is towards the bottom of the radar and should be free and accessible for cable insertion and removal. The two LEDs should also be visible for confirming power and Ethernet link activity.

The radar should always be mounted with 0° roll (about the Z axis) and 0 to 20° pitch (about the X axis) depending on the target use case. Once mounted, the exact X, Y, Z radar geographic position (typically Latitude & Longitude in decimal degrees and Altitude in meters) should be recorded, typically measured via an external GPS instrument. Also, the final roll, pitch and yaw (compass heading) should also be measured and recording in degrees. For more information on mounting solutions and installation training, please contact support@echodyne.com

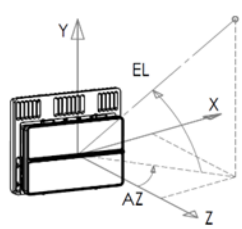


Figure 3: X, Y, Z Axis

4.3 Thermal Considerations

EchoGuard is designed to achieve maximum reliability with natural convection only in a fixed installed location without the need for any forced air consideration. It draws 45 watts of typical DC power and must dissipate about 42 watts of that through thermal conduction to the environment. All internal circuits are physically heat sunk through conduction to the T6061 aluminum housing and external fin sets. The thermal heat flow path is through the case and fin sets to the ambient air. For maximum reliability, it is important to provide enough air flow, such that the unit internal temperature sensors read no more than 15-25˚C above ambient temperature. In a typical outdoor configuration with near still air (no moving air other than convection), at our specified maximum operating temperature of 75˚C, this will keep the SSPA junction temperature under 140˚C, supporting an 8-10 year service life with margin.

4.4 Interface and Control – Test Configuration

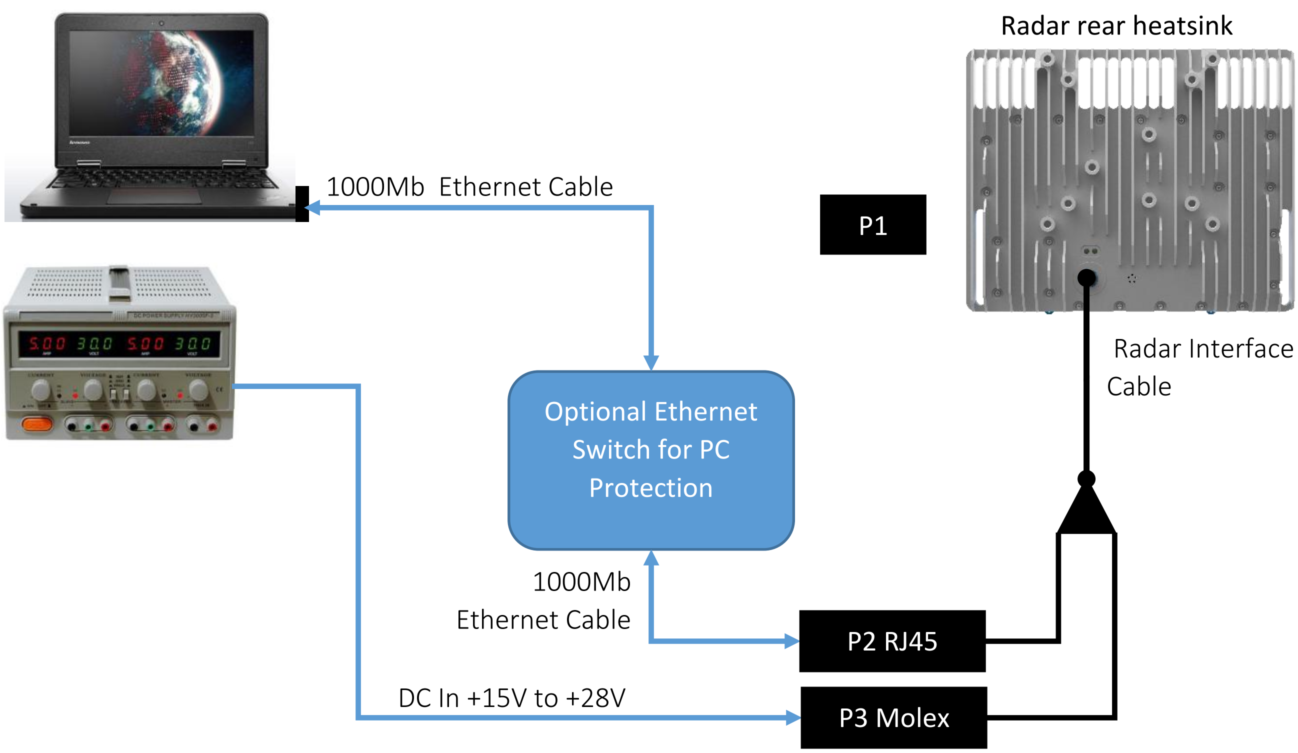


Figure 4: Bench setup for initial integration

4.5 Power-On Sequence and Setup

1. **OPTIONAL** – While not necessary, Echodyne suggests using a low-cost ethernet switch to prevent any possible damage to your PC or Laptop if something should unexpectedly occur while connecting to the radar.
2. Without the radar connected, set your DC power supply to a voltage between 15 and 28 Vdc and the current limit in accordance with; **Current Limit= 55Watts/$V_{OPERATING}$**). This will ensure the unit has enough peak current to start properly but prevent damage to the cable assembly in the event of a problem.
3. Next, make all connections shown in the diagram. Connect the Radar to cable harness end while ensuring it snap locks into position at the correct clocking angle (white dot- white dot). Plug the RJ45 connector into your switch or computer, plug the DC Power supply into the Molex connector RED/BLACK pins. Use a minimum of 22 AWG stranded wire for integration.
4. Turn “ON” the DC power source. As the unit boots up internally, the power will increase to about 8 Watts in IDLE state. It will hold here until given commands.
5. Within 60 seconds the radar should completely boot up and be ready to communicate through the Ethernet ports. Verify the LED indicator lights are flashing and communicating with the port connection.

The right-hand light should be steady. The right-hand light on confirms unit auto negotiation to 1000BT, left hand light indicates that the unit has negotiated to 100BT, and both LEDs indicate there has been a fault and is there is no communication to the radar. Please note that if no LEDs are lit, there has likely been an issue with the Ethernet connection.

6. Proceed with control commands to query the radar. Execute the command line BNET.exe program provided and start with [*IDN?] which should respond with a message stack that includes serial number, firmware revision and MCU code revision.
7. You are now ready to begin testing your new radar.

4.6 Software Update Instructions

As Echodyne regularly releases new features and capabilities while addressing any legacy bugs, it is routinely necessary to upgrade the radar firmware. Users are notified of available upgrades along with a description of the changes via our customer portal and Echodyne support. To check if your unit is up to date; power the radar and connect using the BNET executable provided. Enter the *IDN? command and a summary of the SW Suite (for example, SW17.0.0) as well as details of the MCU and FPGA versions are shown.

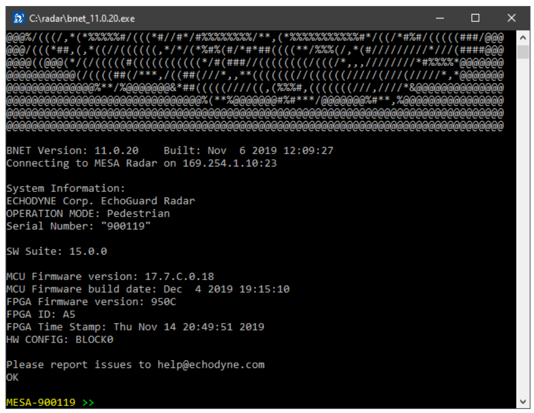


Figure 5: BNET application for checking S/W configuration

Updates can be performed via BNET (see BNET manual for details) or using the Radar Update Utility. Echodyne recommends using the Radar Update Utility as it is easier and less error prone. The HPS and FPGA images are provided as a combined .zip for automated loading via the Radar Update Utility for Window and Linux platforms. UBOOT-SPL:LOAD command will need to issued when upgrading from at minimum SW17.0 or greater to a later version of SW. For upgrading from SW16.4 or lower as the base SW, you'll need to first upgrade to SW17.0 or greater, issue the UBOOT-SPL:LOAD command, then reboot.

To upgrade using the Radar Update Utility, connect to the radar, start the update utility, select the .zip folder containing the HPS and FPGA images, then click **Launch Update**. Refer to the Radar Update Utility manual for details.

Note: **Command** Port, **Status** Port, and **TFTP** Port must be enabled to perform the upgrade. These are enabled by default.

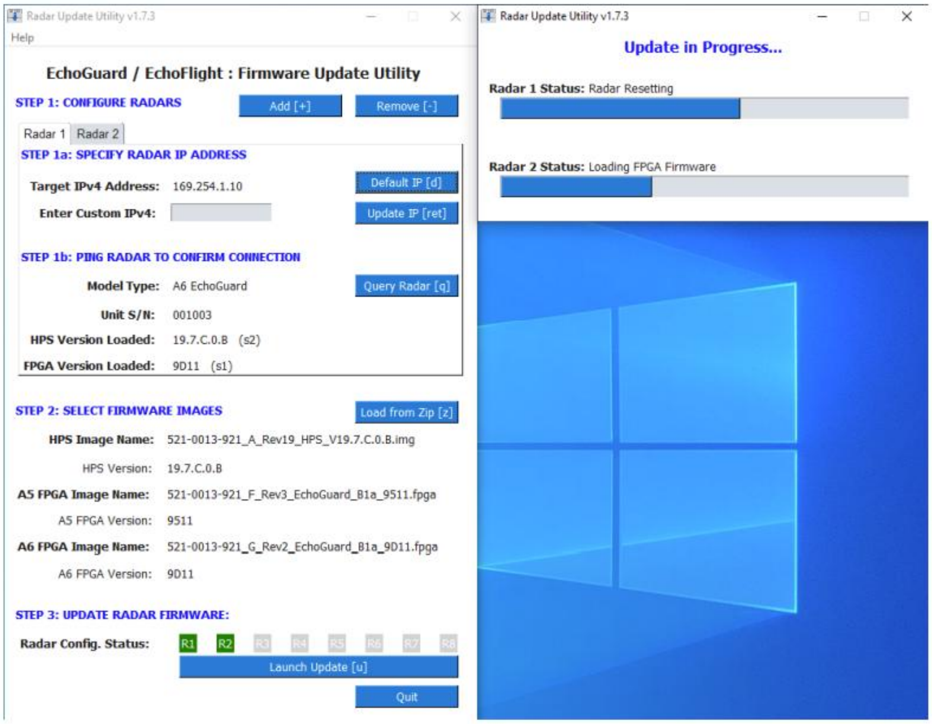


Figure 6: Radar Update Utility for Firmware Updates

4.6.1 Compatible Supporting Software Versions

The following table describes which versions of support software such as RadarUI are compatible with a given radar software version.

Table 5: Compatible Supporting Software Versions

SW Suite	Release Date	BNET Version	Radar Update Utility	RadarUI
16.3.0	Jul 2021	11.0.28	1.8.5	1.3.0
16.4.0	May 2023	11.1.5	2.0.8	2.1.0
17.0.0	Sept 2023	Not supported	2.0.8	3.2.2
17.1.2	July 2024	11.4.0	2.1.3	5.0.5
18.0.4	Dec 2024	11.5.0	2.2.0	5.1.1

4.7 Software Development Kit (SDK)

The EchoK SDK includes the RadarIO command line interface, example python scripts for recording and streaming data, reference libraries, and reference documentation including this manual. The SDK is compatible with Windows 10 as well as Ubuntu 20.04 and 22.04.

SThe SDK also includes install scripts for Windows and Linux to make installation easy. For further details, see the README.pdf included with the SDK.

5. RADAR SYSTEM OPERATION

The radar unit has seven distinct operation states: Initialization, Idle, Executing, Search, Error, and Search While Track (SWT). In each state, the radar outputs status and data on TCP ports to which the user is connected. The commands available to the user to interact with the MESA radar depend on the current state.

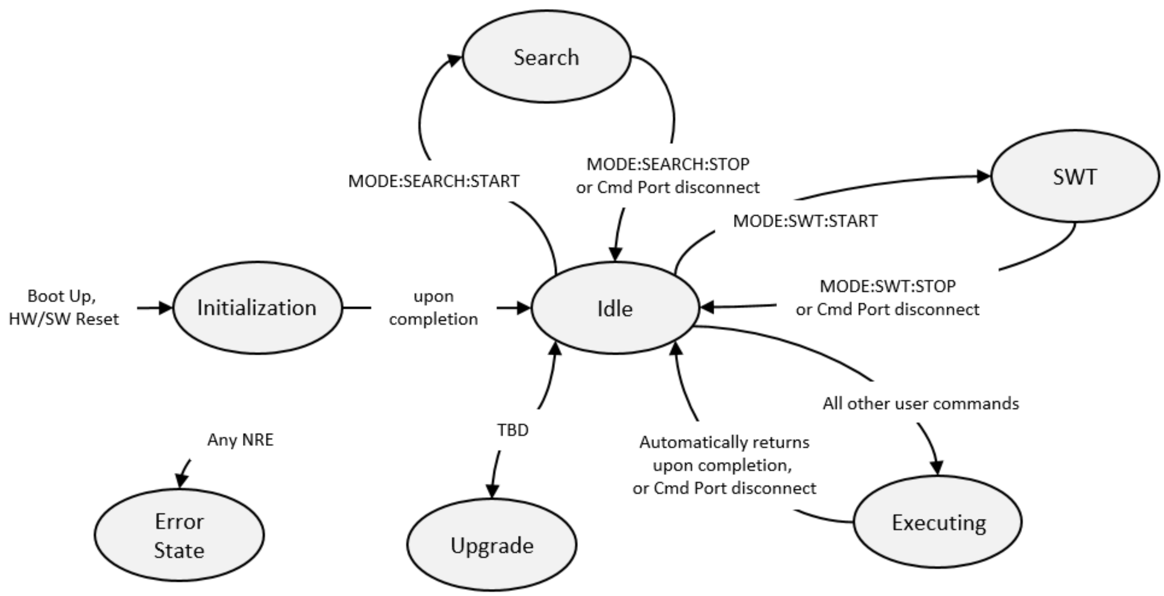


Figure 7: MESA radar system state diagram

5.1 Initialization

The radar enters Initialization at boot up or hardware/software system reset. While in the Initialization state, the radar will perform internal software/firmware loading needed, load user parameters from non-volatile memory, and configure hardware peripherals. Once these initializations are complete the system will enter the Idle state. While in the Initialization state, the radar will not interpret, respond to, or remember any user commands. If a non-recoverable error is detected the system will transition to the Error state. The user is advised to wait until the MESA is completely booted up prior to issuing commands as we have noted some fault conditions from early communication. Also, as previously noted, look for the Right LED light to come on signaling full speed 1000BT protocol has been negotiated. (approx. 30 – 60 seconds).

Active TCP Port(s) during **Initialization**: **none**.

5.2 Idle

After initialization is complete, the system enters the Idle state. The system state can only transition out of the Idle state in response to a user command. The user can use any commands to communicate with the system to configure/query mode parameters, configure/query system parameters, and command the system to execute actions. Upon completion of user command execution or if the user disconnects from the Command Port, the system returns to the Idle state. When reentering the Idle state after command execution, the system returns any reply to the user command over the Command port, followed by an “OK”.

Active TCP Port(s) during **Idle**: **Command** Port, **Status** Port.

5.3 Executing

Configuration and parameter commands return to an idle state after completion (all commands except MODE:SEARCH:START/STOP and MODE:SWT:START/STOP) and exist in an executing state for their duration. Once the user command is executed the system state returns to Idle and then returns any reply to the user command followed by an 'OK' indicating that the command execution is completed. In the Executing state, the system only performs actions that return automatically. While the state is Executing, the system will not process or remember new user commands. If a non-recoverable error is detected, the system will transition to the Error state.

The Search Mode and SWT (Search While Track) Modes, have distinct persistent system states, and do not enter executing and do not return to idle (until the user specifies STOP).

Active TCP Port(s) during **Executing**: **Status** Port.

5.4 Search

When in the Search state, the MESA radar scans the beam across a fixed azimuth/elevation sector of space, called the field-of-view (FOV), with fixed azimuth and elevation step sizes. Each pass over, the FOV is a single frame.

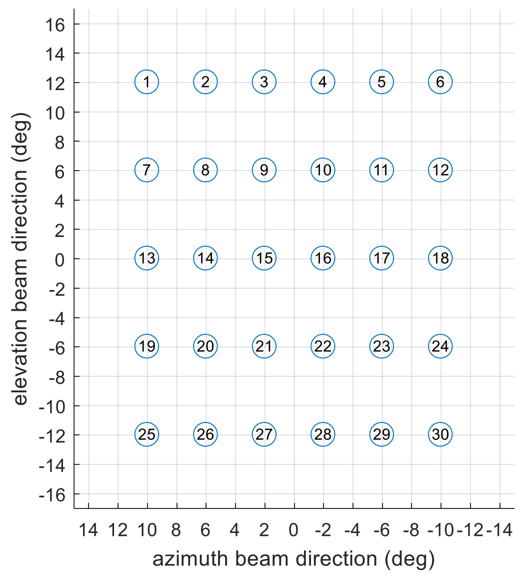


Figure 8: Search state

Figure 9: Example search grid for a search configuration with azimuth FOV of -10° to 10°, elevation FOV of -12° to 12°, azimuth step size of 4° and elevation step size of 6°. Each beam pointing direction is indicated with a circle and the sequence in which each beam direction is visited is labeled.

The user can define the extents of the searched FOV using the following commands:

- To set the azimuth minimum, maximum, and step size use:
  - MODE:SEARCH:AZFOVMIN, MODE:SEARCH:AZFOVMAX, MODE:SEARCH:AZSTEP.
  - Since the azimuth beam width (two-way) is 2˚, the user would typically set the AZSTEP to 2.
- To set the elevation minimum, maximum and step size use;
  - MODE:SEARCH:ELFOVMIN, MODE:SEARCH:ELFOVMAX, and MODE:SEARCH:ELSTEP.
  - For maximum search rate, avoid setting the elevation beams any lower or higher in altitude than needed. Also. consider the physical mounting of the radar on the vehicle or location. Since the two-way radar elevation beam width is 6˚, the user would typically set the ELSTEP to 6 or 8.

On entering the Search state, the system replies to the user over the Command Port with 'OK' indicating that the MODE:SEARCH:START command has been executed. In the Search state, the system will only respond to the MODE:SEARCH:STOP command; any other command will be ignored and not remembered. When the MODE:SEARCH:STOP command is received, the state will transition back to Idle. On returning to Idle, the system will reply with 'OK' indicating that the MODE:SEARCH:STOP command has been executed.

In the Search state, the system will begin searching to FOV using the user-specified search parameters or, if no user parameters were configured, the default search parameters. After each beam pointing direction is collected, the system will stream the collected data to the user over the connected ports. The system will repeat complete Field of Vew (FoV) searches indefinitely until the user issues the MODE:SEARCH:STOP command. If a non-recoverable error is detected, the system will transition to the Error state.

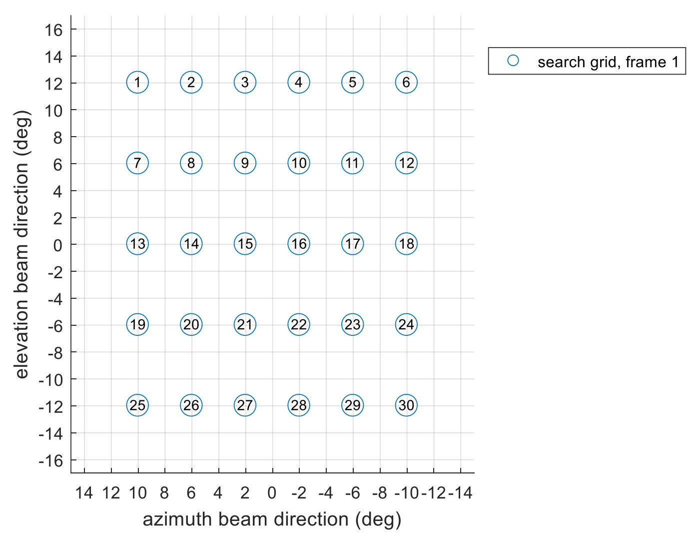
Active TCP Port(s) during **SEARCH**: **Command** Port (with limited command list), **Status** Port, **RVmap** Port, **Detection** Port.

5.5 Search While Track (SWT)

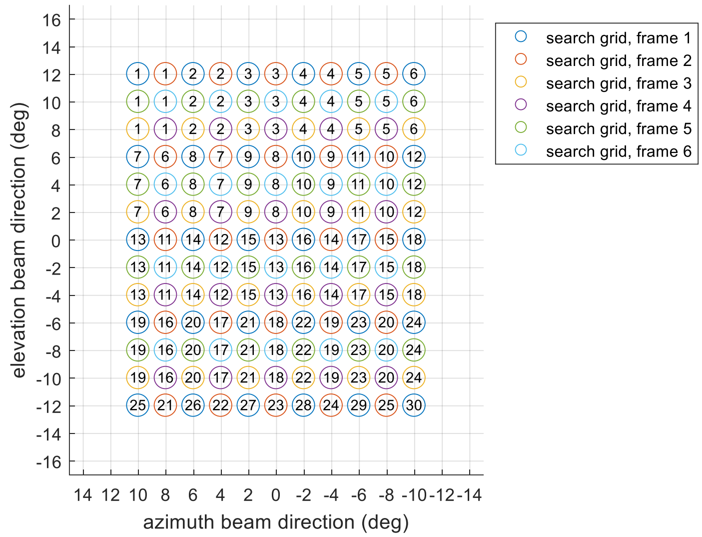
In the SWT state, the MESA radar searches for potential targets and attempts to track them. Track information is reported to the user at every track update time period. When intruders are identified, the MESA uses Echodyne's i-SCAN intelligent beam scheduler to point beams for both searching and tracking. This algorithm constantly optimizes the allocation of radar beams as needed to achieve the best track performance and search refresh rate.

When in the SWT state, the MESA radar scans the beam across a fixed azimuth/elevation sector of space, called the Field of View (FoV), with fixed azimuth and elevation step sizes. Each pass over the FoV is a single frame. In consecutive frames, the search grid is shifted by 2^{o} first in azimuth, then in elevation so that over some number of frames every 2^{o} beam pointing direction is covered.

Active TCP Port(s) during **SWT**: **Command** Port (with limited command list), **Status** Port, **RVmap** Port, **Detection** Port, **Measurements** Port, **Track** Port.



a)



b)

Figure 10: Example search grids for a search configuration with azimuth FOV of -10° to 10°, elevation FOV of -12° to 12°, azimuth step size of 4° and elevation step size of 6°. Each beam pointing direction is indicated with a circle. a) shows the first search grid scanned through with the order of the beam pointing sequence labeled inside each pointing direction. b) shows all the search grids, for each distinct frame, used to cover the FOV. For this search configuration, there are six search grids. After the sixth frame is searched over, the radar will return to the first grid and repeat the same search grid sequence.

During the search sequence, intruders are automatically acquired and tracked by the radar. When a new intruder is first observed by the radar, its state is ”unconfirmed”. After the intruder is observed for some time, its state is promoted to ”confirmed”. At a fixed track-update period, the radar interrupts the search sequence and points the beam at unconfirmed and confirmed intruders to acquire more information about their range and azimuth/elevation bearing. For unconfirmed intruders, a single “unconfirmed-track update beam” is pointed at the bearing that the intruder is believed to be at. For confirmed intruders, four “confirmed-track update beams” are pointed around the intruder’s expected bearing and the collected data is combined to produce a more accurate estimate of the intruder’s bearing.

As an intruder is tracked, the radar continuously updates its confidence that the track is following a real intruder. This confidence level is reported for each active track and represents the radar’s confidence that it has not lost track and that the data feeding the track comes from a real object and not from clutter or noise. The confidence level rages from 0 for no confidence, to 100 for maximum confidence.

The MESA radar ranks tracks using this confidence level, in conjunction with other track-metric based ranking methods, to prioritize which tracks will be updated at each track-update period. This allows the MESA radar to actively track many more intruders than it points beams at during each update period while still maximizing the accuracy of each track.

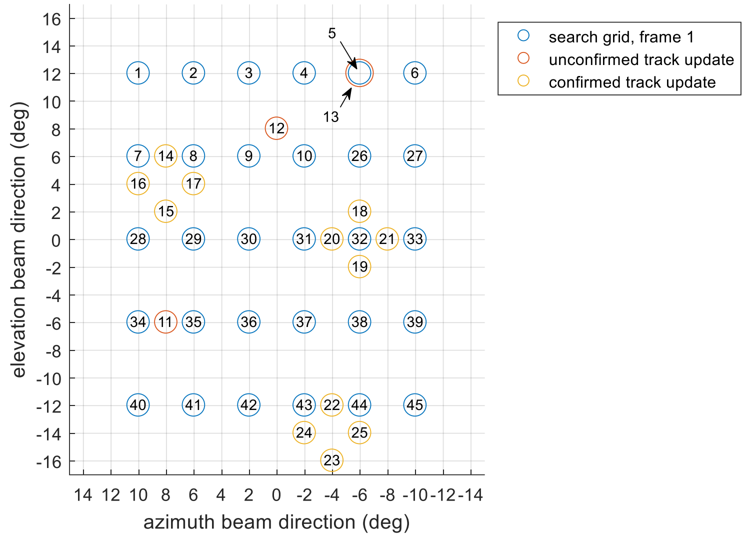


Figure 11: Example SWT beam sequence for a SWT FOV with azimuth FOV of -10° to 10°, elevation FOV of -15° to 15°, azimuth step size of 4° and elevation step size of 6°. After pointing the 10th search beam the radar switches to track beams. First the radar points the beam at three unconfirmed tracks, then at three confirmed tracks (with bearings of approximately (8°,4°), (-6°,0°), and (-4°,-14°)). Note that while the SWT FOV is limited to -12° in elevation, tracks can move outside this FOV by up to 2° and the corresponding confirmed-track update beams may then exceed the specified FOV by up to 4°. A track that exceeds the 2° boundary around the specified SWT FOV is deleted by the radar.

The user can define the MODE:SWT:SEARCH Field of View (FoV) and MODE:SWT:TRACK FoV separately. This is useful when increased frame rate is desired and the search FOV can be limited, but it is necessary to track beyond the search FOV extents of the searched FOV with;

- To set the MODE:SWT:SEARCH Azimuth/Elevation minimum and maximum use:
    - MODE:SWT:SEARCH:AZFOVMIN and MODE:SWT:SEARCH:AZFOVMAX
    - MODE:SWT:SEARCH:ELFOVMIN and MODE:SWT:SEARCH:ELFOVMAX
- To set the MODE:SWT:TRACK Azimuth/Elevation minimum and maximum use:
    - MODE:SWT:TRACK:AZFOVMIN and MODE:SWT:TRACK:AZFOVMAX
    - MODE:SWT:TRACK:ELFOVMIN and MODE:SWT:TRACK:ELFOVMAX

On entering the SWT state, the system replies to the user over the Command Port with "OK" indicating that the MODE:SWT:START command has been executed. When the MODE:SWT:STOP command is received, the state will transition back to Idle. On returning to Idle, the system will reply with "OK" indicating that the MODE:SWT:STOP command has been executed.

When the user starts SWT, the system will immediately begin using i-SCAN to search the FoV and track any intruders using the user specified SWT parameters or, if no user parameters have been configured, the default SWT parameters. The system will immediately begin streaming the collected radar data to the user on the connected data ports. The system will stay in SWT indefinitely until the user issues the MODE:SWT:STOP command. If a non-recoverable error is detected, the system will transition to the Error state.

5.5.1 Scene Tuning Order of Adjustments

Fundamental Goal – Reduce the likelihood of nuisance tracks (minimize tracker loading) and reserve time and likelihood of generate tracks on targets of interest

1) FoV selection (optimize Az & El for coverage; smaller is better/faster) have radar look only where intended
2) Scene Global Masking:

   a) RANGE masking (limit min/mask range to interest)
   b) RCS masking (optimize for targets sizes of interest with min/max RCS)
   c) Clutter Masking (increase if scene is full of low velocity targets; for example, windy day)
   d) AGL Masking if desired
3) Scene Zone Masking:
   a) Use to address specific areas of nuisance tracks (HVAC units, road traffic, and so on)
4) Track Filtering & Optimization (only after doing the above to reduce base tracker load)
   a) MIN reporting confidence (only report well-established tracks)
   b) Max target velocity
   c) Coast settings

Key user facing track parameters:

1. Use MODE:SWT:TRACKER:MINREPORTINGCONFIDENCE to adjust the confidence level tracks must have to be reported. This value ranges from 0 to 100 and is set to 20 by default. It is recommended to increase this value if there are far too many spurious tracks.
2. Use MODE:SWT:TRACKER:MAXTARGETVELOCITY to adjust the max velocity (m/s) of targets of interest. Ensure that all targets of interest move at velocities lower than this value for best results. For reference, walkers generally move at 1-3 (m/s); drones at 10-30 (m/s); and small aircraft at 40-100 (m/s).
3. Use MODE:SWT:TRACKER:MAXCONFIDENCETIME to adjust the rate at which tracks increase in confidence. This parameter adjusts the minimum amount of time a track can reach 100% confidence. It is generally not recommended to lower this value unless target acquisition times must be in the 0-3 second time frame. On the other hand, it is recommended to increase this value if there are a lot of spurious tracks and fast target acquisition times are not paramount (NOTE: It is recommended to always keep this value less than MODE:SWT:TRACKER:MAXCOASTTIME).
4. Use MODE:SWT:TRACKER:MAXCOASTTIME to adjust the rate at which a track's confidence decays when no data is associated to it. It is important to remember that when a track is coasting with no new associated data, the certainty of the true target's location and velocity degrades. Thus, it is recommended to keep the max coast time close to the default value unless extensive testing is conducted.

5.5.2 Track Metadata Fields

5.5.2.1 TOCA and DOCA

TOCA (time of closest approach) and DOCA (distance of closest approach) are calculated by taking the current (x, y, z) position of a track and projecting a line from that point based on the track's current (x, y, z) velocity. We then find the point on that projected line that is closest to the origin (radar location). The distance between the origin and this point is DOCA and it is represented in meters. TOCA is then calculated by determining the amount of time it will take to go from the current position to the DOCA point based off the current velocity of the track. This is then added to the current system time and is reported in (days, seconds). This time is then reported relative to the current time in (days, milliseconds).

5.5.2.2 Detection time vs track acquired time

Detections are acquired over a period of (1/track_update_rate) seconds. From the tracker's perspective, all of these measurements occurred at the same time (the system time when they are actually used by the tracker). Thus, to know which detections are correlated to the correct track update, you need to round the detection time up to the nearest track update time.

5.5.2.3 Track State

Currently represents what state the track is in: 0 means the track is inactive, 1 means the track is unconfirmed, and 2 means the track is confirmed. Users should only see tracks in state 2 (confirmed tracks).

5.5.2.4 Track Lifetime

Track lifetime represents the number of track update periods the track has been active.

5.6 System Errors

Whenever possible, the system should attempt to recover from an error and continue executing the current state. An NRE (Non-Recoverable Error) is an error detected by any process that cannot be recovered from through established error handling procedures, or errors or error corrections which severely interrupt an ongoing process. Use the *TST? command to obtain a report of any errors.

Examples of NREs are:

- Any error that requires a software system reset to recover from,
- Errors in the user interface that break the connection requiring actions on the user's part to reconnect,
- Critical hardware failures, or
- Software over/underflows resulting in detection or tracking instabilities.

If a non-recoverable error is detected in any of the other system states, the system state goes to the Error state, if possible. This state attempts to shut down as much of the hardware as possible while maintaining the user interface. The user is informed of the nature of the error and the radar transitions into the Error state. If a user interface cannot be established, the system may automatically attempt a software reset. No error handling should involve a software system reset without first entering the Error state.

Active TCP Port(s): **Command** Port (with limited command list), **Status** Port.

6. RADAR SYSTEM CONFIGURATION

6.1 RCS Masking

Detections can be filtered based on their measured RCS by setting the RCS thresholds with the RSP:RCSMASK:MINRCS, and RSP:RCSMASK:MAXRCS commands. The following table provides estimated RCS values for different target classes. The Track packet reports an average RCS value for each tracked target. Radar RCS fluctuates as a function of aspect angle, material composition, and target external and internal micro-reflectors. Use of the RCS Masking features should be used with caution and developed using experimental testing on specific targets in motion to prevent poor tracking performance. See application note for suggested parameter settings.

6.1.1 Radar Cross Section of Typical Targets

Table 6: Target Class and Nominal Radar Cross Section

Target Class	Nominal RCS
Human	-5 to 0 dBsm
Small drone/sUAS (for example, quadcopter)	-25 to -15 dBsm
General aviation aircraft	+0 to +10 dBsm
jetliner	>+15dBsm

6.2 Zone Masks

A feature called Zone Masks for SWT Mode enables the user to define an enclosed region of three-dimensional (3-D) polar space (R, Az, El) in which measurements (clustered and interpolated detections) will be masked and prevented from generating new tracks. Since the EchoGuard radar scans a large volumetric region, often within the Field of View (FoV) there are regions of spurious movement which can spawn numerous tracks that are not of interest. Examples include busy roads, irrigation heads, HVAC fans on buildings, bird locations, such as duck ponds, as well as windblown trees and bushes. To avoid saturating the tracker with uninteresting targets, the user can now place up to 32 unique zone masks throughout the FoV.

Zone Masks may be specified simply in terms of [min, max] dimension spans in range [meters], azimuth angle [°] and elevation [°]. Optionally, Zone Masks may be defined with additional parameters specifying the velocity range (radial Doppler velocity component [m/s]) and RCS (radar cross section [dBsm]) parameters to prevent generating tracks within the 3-D volume for certain subsets of targets.

Zone Masks extend the functionality of the more limited RANGE:MASK command which is still useful for specifying a minimum and/or maximum range of interest. It is important to note that Zone Mask filtering operates as the last step after RCS, Clutter and Range Mask filtering. Whereas RCS, Clutter and Range Mask filters block detections from propagating to the tracker, Zone Masks operates on measurements (an advanced form of detections which are clustered and interpolated), filtering right before the tracking module as shown below.

New commands added to the radar API include ZONE:MASK:ENABLE, ZONE:MASK[?] and ZONE:MASK:CLEAR as defined in the Super-User Commands section.

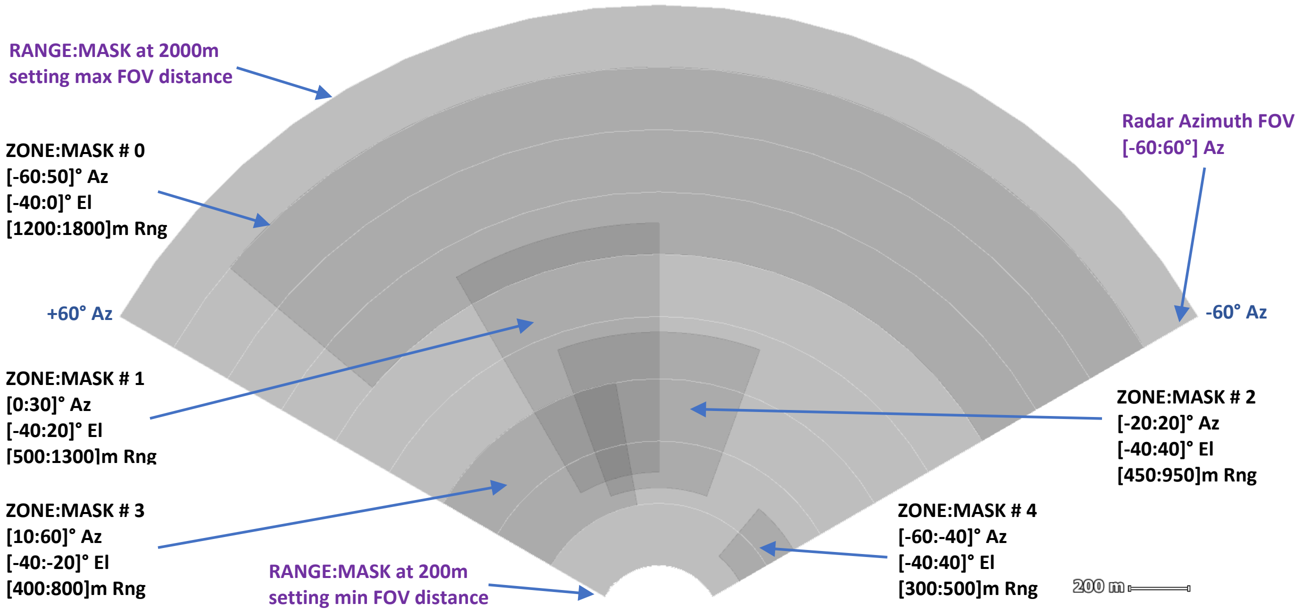


Figure 12: Example overlapping Zone Masks within Radar FOV, showing Range versus Azimuth Plan View

Details on Zone Mask Behavior:

- Specify up to 32 unique zone masks (0-31 slots)
- Zone masks may be changed while the radar is operating.
- Zone masks can be ‘soft’ or ‘hard’.
    - A “soft” zone mask prevents tracks from being generated in the region specified, but existing tracks are allowed to propagate thru the region and won't be killed.
    - A ‘hard’ zone mask kills all tracks that enter its region.
- Zone masks are bulk-cleared by the dedicated ZONE:MASK:CLEAR command.
    - An individual Zone Mask may be overwritten to update its value by rewriting to its slot location
    - Similarly, an individual zone mask maybe be cleared by writing all zeroes “0” to its specific slot parameters without having to clear all other masks via ZONE:MASK:CLEAR
- Zone masks <u>may overlap</u> in 3-D space without issue as the comparison engine checks measurements for inclusion within each zone sequentially starting at zone mask 0.
    - For optimal processing efficiency, overlapping zone masks should be specified starting at slot 0 in order from largest to smallest so each measurement is rejected with the fewest comparison operations.
- All zone mask parameters are currently integers.
    - One can specify zone masks in steps smaller than the beam step as they operate on measurements which are interpolated detections across angles and ranges.
- Zone Mask measurement velocity is the radial Doppler velocity component as detected by the radar within the maximum unambiguous velocity (MUV). This can be a complicated concept to initially grasp for new users. Remember, only objects moving toward or away from the radar have a detectable Doppler velocity component.
    - Consequently, the Zone Mask velocity range spans the -MUV to the +MUV of the radar with negative velocities approaching and positive values departing from the radar location.
- The default RCS range spans -30 dBsm (very small bird) to +100 dBsm (brighter than any conceivable target).

6.3 AGL Mask

In addition to global RCS and clutter masking capabilities, a new height-based ”AGL Mask” has been introduced. While Zone Masks define an angular sector of space to reject, they can be challenging to optimally configure as their effective height changes with range. The AGL Mask defines a global elevation range window in which measurements are permitted to propagate to the tracker. Masked elevations behave as a “hard mask” in which tracks are starved of measurements and prevented from propagating. Example use cases where this may be beneficial are for cUAS air security or Ground Based Detect and Avoid (GB-DAA) where one doesn’t care about tracking ground targets and can ignore anything below a certain height. Alternately, for pure ground security (where one doesn’t need air target awareness), this can reduce the impact of birds in the area. Note the radar does not possess an internal 3-D digital elevation map (DEM) so the elevation relative to ground assumes a constant mask height relative to the local radar defined elevation.

To use this mask, the radar (for example, PLATFORM) needs to know its own height relative to local ground (in meters) as well as angular orientation (roll, pitch, yaw) so that the internal measurements can be transformed to ENU (East, North, Up) coordinates for masking on the ‘Up’ component. Positive pitch is defined as the radar tilting backward from perpendicular to the ground. Roll should be close to zero degrees, but if there is a roll component, it’s defined as positive with radar rotated right side up, left side down. Yaw is defined as CW compass angle from North (0 degrees). While not critical to set yaw for AGL masking, it has been included for potential long-term ENU masking in a future software release.

Specific usage steps:

1. Set radar height in meters relative to local mean ground via the PLATFORM:STATE:ELEVATIONAGL command
2. Set the radar angular orientation in degrees using the PLATFORM:STATE:ORIENTATION command.
3. Set the mask height thresholds in meters using the AGLMASK:MAXAGL and AGLMASK:MINAGL commands relative to the same ground height you used to define the radar. Note, they can be used independently. For example, if you do not need a MAXAGL, feel free to leave it at its default 10,000 meters, well above the instrumented range of
4. Enable the mask to take effect by setting AGLMASK:ENABLE to TRUE.

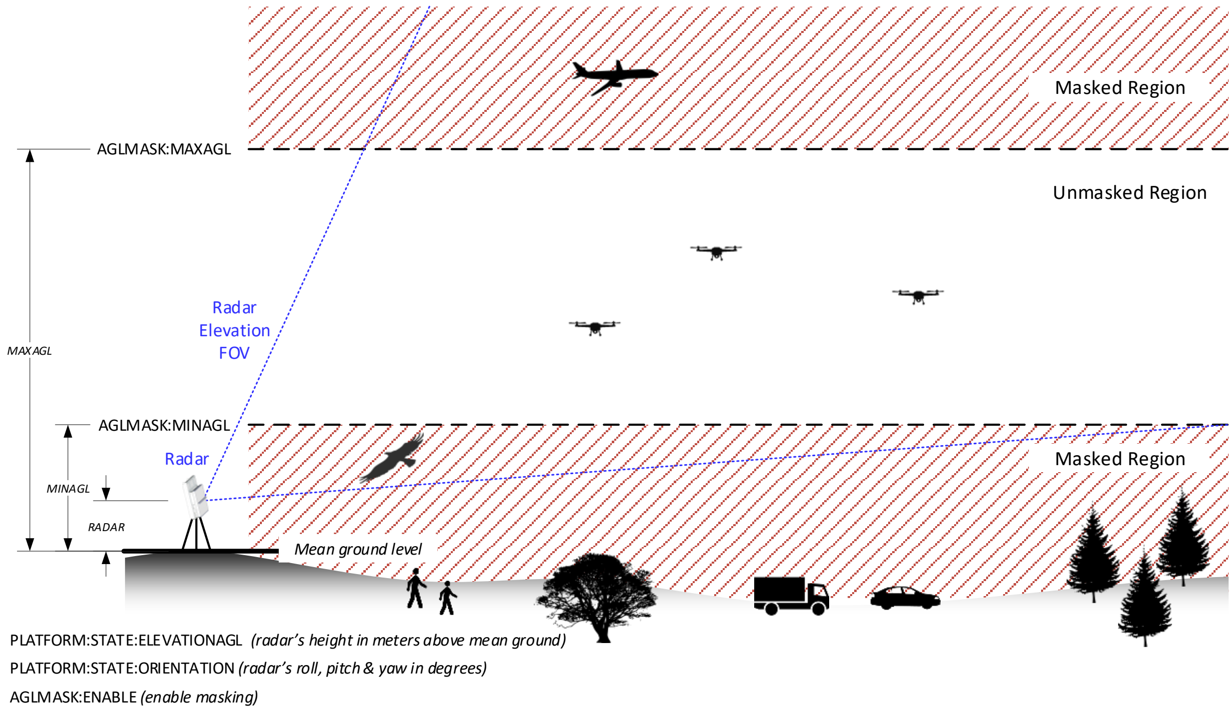


Figure 13: AGL Mask Example

6.4 Target Classification

Introduced with software release 17.0, the new track-level classifier uses machine learning to classify tracks one of 7 target classes:

1) UAV
2) Bird
3) Aircraft
4) Vehicle (ground)
5) Walker (human)
6) Clutter
7) Other

The classifier works at full range with the B1a waveform; soon after a track is formed on a target, the classifier assigns a probability that the target is one of the 7 target types. These probabilities are stored as fields in the extended track packet, and are updated at each track update period along with other track packet fields.

Filtering tracks by target class can be a powerful way to focus on targets of interest. This filtering is supported natively by RadarUI version 4.0 and later. See the RadarUI manual for more details.

Note: The classifier currently has not been validated to work with B1b or A1a waveforms and is set to "Disable" by default.

6.5 Operation Modes

Operation Modes will configure a variety of parameter presets to optimize the radar for certain target types (for example, drones, crewed aircraft, or walkers). Echodyne understands that end-users can apply the radar in non-traditional ways, so most parameters are user-configurable, either through Operation Mode (adjusts several parameters at once) or through individual parameter adjustments.

For Echoguard, the available Operation Modes to select are 0 (walkers / pedestrians), 1 (small drones/sUAS) and 2 (crewed aircraft). We assume that small drones will be traveling at or below 30m/s and are below -5dBsm. We also assume crewed aircraft are traveling at speeds greater than 30m/s and are greater than +5dBsm. **The default Operation Mode for EchoGuard is set to 0 for walkers.**

In EchoGuard, changing Operation Mode affects the following values:

Table 7: Operation Mode and Parameters

Parameter	0 (walkers)	1 (small drones)	2 (crewed aircraft)
MODE:SWT:SEARCH:ELFOVMAX	12	40	40
MODE:SWT:TRACKUPDATERATE	5	10	10
MODE:SWT:TRACKER:MAXCOASTTIME	5	4	4
MODE:SWT:TRACKER:MAXTARGETVELOCITY	10	30	100

NOTE: When changing between Operation Modes, always select the Operation Mode first before customizing any additional parameters as Operation Mode may overwrite custom settings. If you are using the radar for non-conventional use cases, please contact Echodyne for advice on tuning parameters.

6.6 Dynamic Clutter Mask

Another type of mask to help optimize the radar search resources is Dynamic Clutter Mask. Dynamic Clutter Mask mode will reduce the number of clutter tracks caused by small waves (<4ft) when radaring areas of interest over bodies of water. This helps free up radar resources to better track targets of interest. I

It is recommended to use Dynamic Clutter Mask when the majority of the intended FOV of the radar is over water. However, the feature does not guarantee masking of all clutter tracks caused by waves. If the sea state is high, then the radar is still likely to pick up clutter tracks from the larger waves. High cresting waves near the shoreline are the most challenging to eliminate using Ridgclutter.

Dynamic Clutter Mask is disabled by default for all waveforms and can only be enabled for B1a and A1a. It is not supported for B1b.

Note: Due to the large amount of data and processing needed for the Dynamic Clutter Mask algorithm, users are unable to collect RVMaps when Dynamic Clutter Mask is enabled. If there is a need to collect RVMap data, then Dynamic Clutter Mask should be disabled. Dynamic Clutter MaskDynamic Clutter Mask.

There are three levels of Dynamic Clutter Mask masks, Level 1-3. When to use a certain level is dependent on the application of the radar, not the specific environment of the body of water. Level 3 is the highest mask setting. This will result in the highest amount of clutter masking and the lowest number of false positive detections. Level 1 is the lowest setting and should be used if there is a concern of accidentally masking targets of interest.

Table 8: Dynamic Clutter Mask Commands

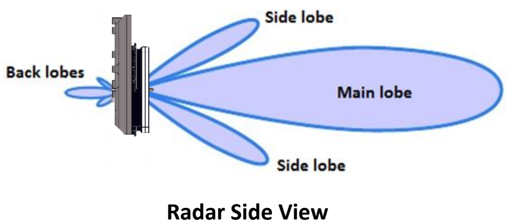
Dynamic Clutter Mask command	Function
RSP:DYNAMIC:CLUTTERMASK:ENABLE 0	`PSWD_SUPER_USER` [TRUE/FALSE] to enable Dynamic Clutter Mask option
RSP:DYNAMIC:CLUTTERMASK:LEVEL 1	`PSWD_SUPER_USER` [1-3] sets the mask level (1 is lowest setting, 3 is highest) Note: 3 will mask out the most waves but may mask out more targets of interest.
RSP:DYNAMIC:CLUTTERMASK:STARTRANGE 6	`PSWD_SUPER_USER` Starting range in meters for applying Clutter Mask. Note that static range masks take precedence here. If Start range is less than the minimum range mask then the range mask takes precedence.
RSP:DYNAMIC:CLUTTERMASK:RANGELENGTH 0	`PSWD_SUPER_USER` Range in meters for applying Clutter Mask. 0 for this parameter implies applying the mask across the full range. As an example, if

	STARTRANGE was 6 and RANGELENGTH was 994 then Clutter Mask would be applied from 6 meters to 1000 meters.
RSP:DYNAMIC:CLUTTERMASK:MOUNTINGAZANGLEOFFSET 0	PSWD_SUPER_USER Mounting Angle of radar in degrees. Default value: 0 Min value -180, max value 180 This parameter is only used if the productMode is DAA.
RSP:DYNAMIC:CLUTTERMASK:THR 3.0665	PSWD_ENGINEERING Do not change - Threshold value of ridgeness metric below which a ridge is present
RSP:DYNAMIC:CLUTTERMASK:THRWINDOW 11	PSWD_ENGINEERING Do not change - Size of the smoothing kernel used to smooth the ridgeness metric
RSP:DYNAMIC:CLUTTERMASK:RIDGEWIDTH 6	PSWD_ENGINEERING Do not change - Size of smoothing kernel used to smooth the ridge position
RSP:DYNAMIC:CLUTTERMASK:RANGEWIDTH 2	PSWD_ENGINEERING Do not change - Range extent of mask kernel in bins
RSP:DYNAMIC:CLUTTERMASK:VELOCITYWIDTH 1	PSWD_ENGINEERING Do not change - Velocity extent of mask kernel in bin

7. RADAR CHANNELIZATION

There are **three** channelization methods (**Frequency**, **PRISM** and **TCM**) for mitigating the effects of radar self-interference when multiple EchoGuard radars are operated within line of sight (LOS) of each other. The signal from an in-band interfering radar will look just like a real target if the chirp start-times align to put the interference within the instrumented range. While normal radar operation requires transmitted energy to propagate to the target and back to the radar receiver, incurring greatly attenuated R^4 (where R=range) signal path losses along the way, nearby interfering radar energy need only propagate one-way (R^2 signal attenuation), resulting in potential self-generated interference false targets from distant radars without proper steps to deconflict through correct radar channel selection.

By utilizing the 3 frequency channels, 4 PRISM channels, and 4 TCM channels, operators have up to 48 channel combinations (with supporting hardware) to address any realistic radar deployment density. Each of these channelization approaches offer different levels of radar-to-radar isolation which can be strongly impacted by pointing direction and the resultant main beam to sidelobe/backlobe beam link levels as the radars scan through their fields of regard. Radar spacing charts in meters is provided in the deployment Application Notes showing the latest recommendations by orientation.



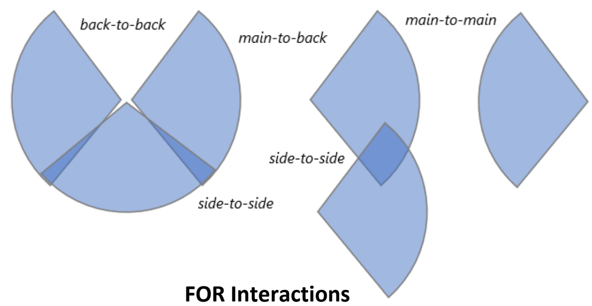


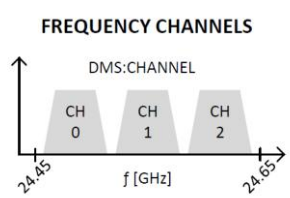
Figure 14: Radar beam lobes and potential radar-to-radar interactions

Also be aware, for long radar baselines, the curvature of the earth can be leveraged to limit radar LOS interactions at 24 GHz. For a 2.0-meter radar mounting height (for example, tripod stand), the Horizon Distance = 3.57 * sqrt(height) [km] = 5.0 km. For relatively flat terrain, radars > 5.0 km separation will have little to no interaction.

For complicated installations with many radars on hilly terrain, please contact Echodyne support, as we can provide expert guidance on optimal deployment and channel selections using our internal site-planning software resources that utilizes digital elevation models (DEM) and empirical antenna beam gain patterns.

7.1 Frequency Channels

The 200 MHz radiolocation/navigation band (24.45-24.65 GHz US, 24.05-24.25 GHz INTL) utilized by Echodyne is relatively free of other terrestrial transmissions. Within this band, EchoGuard radars have three frequency channels, each with 45MHz bandwidth and 15MHz of guard band separation.



In the software command layer, these channels are selected by the radar command DMS:CHANNEL 0, 1 and 2 respectively.

FREQ Ch Command	Function
DMS:CHANNEL	[0/1/2] Selects the specific frequency band for radar operation; persistent

When operating co-located radars, frequency channels should always be your first option as they offer maximum isolation. The opposite channels (0 & 2) offer the most isolation and are ideal for two very closely spaced radars. If additional co-located radars (for example, ~1-meter separation, 4-up for hemispherical coverage) are desired, a combination of Frequency and TCM channels is recommended. The center channel (DMS:CHANNEL 1) requires slightly greater radar separation for complete isolation. For pure frequency channelization, the general spacing guidance to prevent spurious interference detections is as follows.

Table 9: Distant and Adjacent Channel Spacing

Distant Channel (A↔C) Spacing

interaction	mainlobe	sidelobe	backlobe
mainlobe	20 m	2 m	1 m
sidelobe	-	1 m	1 m
backlobe	-	-	1 m

Adjacent Channel (A↔B, B↔C) Spacing

interaction	mainlobe	sidelobe	backlobe
mainlobe	230 m	24 m	8 m
sidelobe	-	2.5 m	1 m
backlobe	-	-	1 m

Interference from another radar in the <u>same</u> frequency channel will appear in the RVmap as returns that look like real targets, generating detections and related false tracks. The power level from interference can be very high (relative to a real target) as it is proportional to R^2 and can have an extent in both range and velocity depending on the very small but finite drift of the precision crystal oscillators in the two radars. The interfering signal is generally detectable regardless of beam direction (sidelobe link) and will therefore appear across most beams in the Field of View (FoV). Because the interference returns are unrelated to beam angle, they will form a “shell” or “sheet” of detections at a certain range that can slowly move inward or outward as the time references between the two radars drifts independently. Cluster processing will combine this ring of detections into generally one very bright measurement which will create a high RCS track that slowly drifts in range. The signature is fairly unique and identifiable for experienced operators.

Note: It is important to be aware that the EchoGuard radar instrumented range (~6 km) only utilizes 9 MHz or ~1/5th of the total chirp bandwidth. Consequently, <u>4/5ths of the time, interference may not be visible inside the instrumented range, and one can falsely infer there is line of sight (LoV) isolation</u> between distant radars on the same frequency channel. If not utilizing the additional channel options described below, it is recommended to operate for 20-30 minutes looking for the tell-tale signs of interference.

7.2 PRISM Channels

Introduced with the SW16 Suite release, **P**ulse **R**epetition **I**nterval **S**kew **M**itigation or PRISM channelization is a signal processing algorithm that creates virtual channels within the same frequency channel. This approach is very effective for non-co-located radars when the interference signal levels are reduced to a point (observing spacing guidelines below) where the PRISM algorithm can detect and remove the radar interference patterns from the RVmaps. Since this technique operates at the fundamental radar raw data level, it supports customers who elect to receive and process RVmaps with their own detection and tracking routines.

Since PRISM operates on the real-time received return signal, it is embedded in the FPGA waveform digital signal processor and necessitates a coordinated HPS and FPGA firmware update as provided in the SW16 release package. The extra PRISM processing logic also requires BLOCK1/A6 FPGA hardware which adds more fabric and signal routing resources to support the real-time processing. This upgraded hardware is currently shipping with new units and an option exists for upgrading legacy hardware at the factory if required. One can easily determine the nature of their hardware by typing the FPGA? command (A5 or A6 will be returned) or using the *IDN? command (A5/BLOCK0 or A6/BLOCK1) will be visible in the summary.

Again, the signal from an interfering radar with the same PRI and frequency band as the self-radar will look just like a real target if the chirp start-times put the interferer’s IF in-band. By making the PRIs between two units slightly different, interference can be made to step in range from one chirp to the next where it can be detected and removed. At a very high level, the PRISM routine involves the following steps.

1. The radar skews the Tx waveform pulse repetition interval (PRI) between units on a given PRISM schedule to make interference easily identifiable. This is similar to the PRI skewing introduced in SW15 for better isolation between adjacent frequency channels but is now parameterized based on the selected number of desired PRISM channels in a given frequency band.
2. The radar interference is detected during processing of the received signal.
3. The algorithm replaces the detected interference samples with an estimate of the underlying target signal in a way that avoids RVmap discontinuities and false detections.

To enable and configure the PRISM channelization system, there is a simple, intuitive command set, similar to TCM configuration, but without the complications of a control range, and so on.

Table 10: PRISM Ch Command and Function

PRISM Ch Command	Function
PRISM:CHANNEL:ENABLE	[TRUE/FALSE] to enable PRISM channelization option
PRISM:CHANNEL:MAX	[2-4] sets maximum # of PRISM channels desired in configured frequency band
PRISM:CHANNEL:NUM	[0-3] set specific PRISM channel # for that radar

As mentioned previously, the spacing requirements for PRISM are greater than co-located radars typically require but are an important tool in distributed deployments. The spacing guidelines below are considered reasonably conservative and should yield good performance in suppressing interference detections. Note, if one only needs 2 PRISM channels, one should set MAX=2 as this sets the PRI skew to a slightly larger value than MAX=4, resulting in more isolation for a given distance.

Table 11: PRISM Unit Spacing Guidelines

PRISM Unit Spacing Guideline		
# of Radars	Minimum Unit Separation sidelobe-sidelobe/backlobe link	Minimum Unit Separation mainlobe- sidelobe/backlobe link
2	50 m	150 m
3	75 m	225 m
4	100 m	300 m

7.3 TCM Channels

Time Channel Mitigation (TCM) allows radars on the same frequency channel ('co-channel' radars) to operate in near proximity without performance-degrading interference. Combinations of frequency channels and time channels allow for greater installation densities such as 4-up co-located radars on a pole. For TCM to operate properly, the radars must first be configured to operate on the same Frequency and PRISM channels to facilitate the tracking.

The recommended 4-up tower configuration is 2 frequencies with same-frequency radars facing opposite directions:

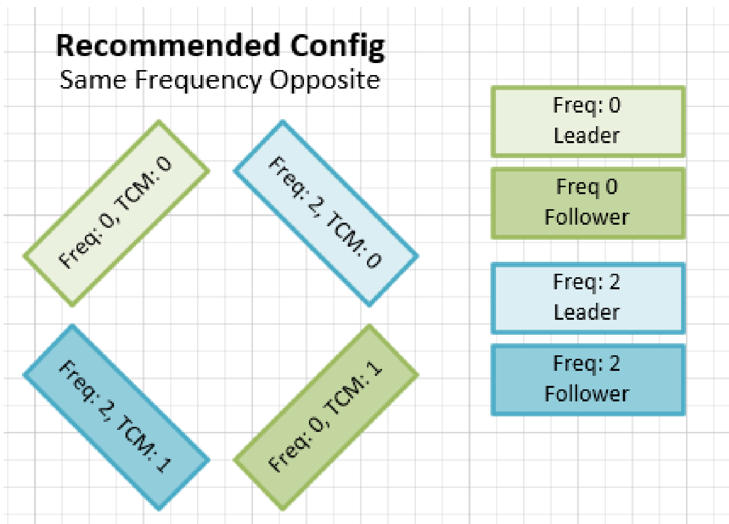


Figure 15: TCM Channels

Since the EchoGuard radar instrumented range (~6km) only utilizes 9MHz or ~1/5th of the total chirp bandwidth, there is available space during the chirp ramp period to sequence co-frequency radars chirps such that their resulting interference appears at a given time delay. This can be actively controlled as the respective radar oscillators drift so that the interference detections at a specific range can be safely ignored (distances > MROI).

There are three stages of successful TCM: discovery of interference, mitigation, and monitoring. In the discovery stage, the radar "listens" for other Echodyne radars. If another radar is detected, the mitigation stage begins. In this stage, the power of the interference signal is measured, and interference is moved to a controlled range—or a "buffer region"—that is greater than the user's maximum range of interest (MROI). In the monitoring stage, the radar will periodically check for interference drift and will snap the interference back to the controlled range, past the user's MROI. Interference between two radars manifests as returns that span the detection range over time. In TCM, we are adjusting the relationship between the two radars to hold this signal at a constant distance over time.

To ensure proper sequencing, after issuing a MODE:SWT:START or MODE:SEARCH:START command, there is a built-in time delay (default 5 seconds) for TCM channels > 0. This is to allow for the TCM Ch0 'leader' to establish itself before the 'follower' radars actively search and attempt to lock. For TCM channels > 2, there is a primary leader, 1st follower (locks to leader after 5 second delay) and 2nd follower (locks to 1st follower after 10 second delay), etc.

Time Channel Mitigation (TCM) uses predictive algorithm to track and optimally coast through periods when radar time channel tracking is lost. This may occur when radars are distributed in locations and linked via variable multipath. The TIME:CHANNEL:COAST command can be used to adjust the coast time which defaults to 30 seconds and should be optimal in most situations. As of SW 14, TCM has been refactored to use a multi-variate Kalman filter with a greatly enhanced ability to initiate channel tracking at start-up in multipath environments and maintain lock over multiple days, even in the presence of variable multipath. Also, in this release, the TCM state reported in the Status packet has more explicit granularity for monitoring lock status.

A summary of relevant TCM commands is provided here. More details in the API command section.

Table 12: TCM Commands

TCM Ch Command	Function
TIME:CHANNEL:ENABLE	[TRUE/FALSE] to enable TCM channelization option
TIME:CHANNEL:NUM	[0-3] sets TCM channel # (0=leader, 1-3 = follower)
TIME:CHANNEL:DELAY	[5 seconds] default delay before followers sequentially enable
TIME:CHANNEL:COAST	[30 seconds] follower coast time w/o measurements before lock is lost
TIME:CHANNEL:RANGEMAXINTEREST	[5000 meters] maximum range of interest, > MROI detections are ignored
TIME:CHANNEL:CONTROLRANGE	[5500 meters] location in range where TCM tracking interference is placed
TIME:CHANNEL:PWRTHRESHOLD	[90dB – Default] Minimum power threshold for TCM acquisition.

NOTE: For a detailed description and explanation for optimization of this TCM feature, please refer to the RADAR APPLICATION NOTE on INTERFERENCE TIME CHANNEL MITIGATION. For distributed TCM installations we recommend contacting Echodyne support for channelization planning.

8. COORDINATE SYSTEMS & IMU

8.1 Available Coordinate Systems

The radar outputs tracks with X/Y/Z and R/Az/El coordinates in its own reference frame and conversion to NUE or Lat/Long/Elev requires transformations performed by customer integration software. See RadarUI for an example.

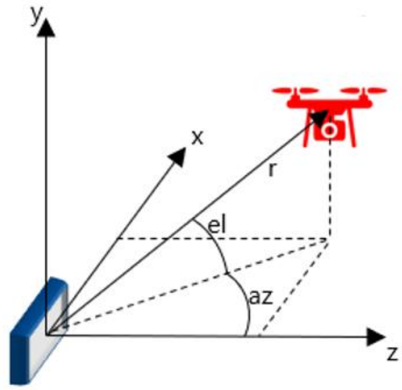


Figure 16: Radar X/Y/X and R/Az/El coordinate conventions

8.1.1 Antenna Coordinates

The radar uses a Cartesian coordinate system that is fixed to the face of the antenna to determine where beams are pointed relative to the unit. This antenna-relative coordinate system is called the 'antenna coordinate system' and is marked on the housing of the radar. The z-axis is the broadside direction normal to the face of the antenna. The y-axis is the direction pointing from the bottom of the unit to the top, in the plane of the antenna face. The x-axis then points from right to left - when looking with the antenna - in the plane of the antenna face. The datum of this coordinate system is at the front face of the MESA radar, aligned with the centroid of the mounting features identified in the ICD (Interface Control Document).

8.1.2 Radar Coordinates

Beam pointing directions and the range of radar returns use a spherical azimuth-over-elevation coordinate system called the 'radar coordinate system'. This coordinate system is specified relative to the antenna coordinate system and so moves with it when the radar orientation changes.

A beam pointing direction is specified by an azimuth (Az) elevation (El) angle pair. Azimuth is the angle, measured from the z-axis toward the x-axis, of the projection of the pointing vector in the xz-plane and elevation is the angle between the pointing vector and the xz-plane. Pointing a beam in a particular direction can be conceptualized as starting with a beam pointed at broadside, the z-axis, steering in the xz-plane away from the z-axis by azimuth degrees, then steering up from the xz-plane by elevation degrees.

The range of a radar return is then the radial distance along the beam pointing direction, r. The datum of the radar coordinate system is collocated with the datum of the antenna coordinate system, and so moves with it translationally.

8.2 World Coordinate Transforms

Echodyne's convention for defining yaw, pitch, and roll is given below. Echodyne defines the yaw, pitch, and roll (YPR) rotations as left-handed intrinsic rotations applied in about the radar's axes in the sequence y-x-z. The rotations are relative to a 'standard orientation' where zero yaw, pitch, and roll orients the radar with its z-axis pointed north and its y-axis pointed opposite gravity.

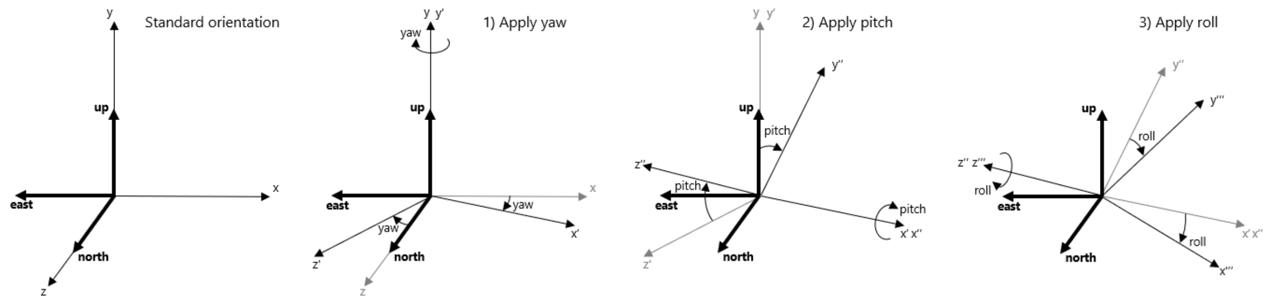


Figure 17: Coordinate Transforms

Figure 18 - Echodyne's yaw, pitch, and roll rotation sequence definition. Echodyne defines the yaw, pitch, roll (YPR) rotations as left-handed intrinsic rotations measured from the 'standard' orientation. The standard orientation is with the radar's z-axis pointed north, the radar's y-axis pointed up, and the radar's x axis pointed opposite to east. The intrinsic rotations are applied in the sequence 1) yaw about the y-axis (y), 2) then pitch about the new x-axis (x'), and then 3) roll about the new z-axis (z'').

A common need is to transform the coordinates of data points coming out of the radar from Az/El/R coordinates to world coordinates (ENU or east/north/up) and to the coordinate systems of other sensors such as pan-tilt-zoom (PTZ) cameras. An available white paper describes a standard procedure and provides example code to perform these transformations as well as test vectors to verify 3rd party integrations.

8.3 IMU Technical Description

The MESA Radar contains an internal six-axis Inertial Measurement Unit that is used to compensate for platform rotations. The IMU contains two independent sensors (accelerometer and gyroscope) whose measurements are fused in software to generate a quaternion that represents the unit's orientation and relative rotation. By default, as it is assume EchoGuard radars are mounted in stationary positions, active IMU compensation of the tracker is disabled, although the IMU quaternion is still available in the status packet for roll and pitch determination. If your use case requires motion compensation, contact Echodyne support for application assistance.

9. SYSTEM I/O INTERFACE

9.1 Physical Interface Description

The physical port on the radar is an IP-68 circular snap lock connector. All DC power, command and data control is provided through this single connector. Echodyne provides two choices for “Y” cables to separate the data and control. With each radar, Echodyne recommends a 24 inch “Y” cable also be purchased. Contact Echodyne if you would like a longer 96 inch “Y” cable assembly or would like us to develop a custom cable for your application. The “Y” cable splits the 12-pin circular connector radar interface into a standard RJ-45 Gb-Ethernet on one connector and a MOLEX DC Power input connector on the other. The user may choose to either use the provided MOLEX connector and mate for DC Power prime input power or cut this off and direct wire to an alternate connector of choice. Upon request, an unterminated 20’ cable (24 AWG) is also available. Any alternate connector chosen must support 15-28 VDC and a minimum of 5 Amperes current rating (22 AWG wire or larger is recommended). The following table is intended as a reference for color codes and pinouts of the supplied “Y” cable assembly.

	Circular Snap-Lock		RJ-45 Gbit Eth	Molex PN43645-0500
Signal Name	P1	WIRE	P2	P3
BI_DD+	6	ORANGE TwPr4	7	
CTL1	2	YELLOW		3 - ORGE
BI_DB-	3	GREEN TwPr2	6	
CTL2	4	VIOLET		4 - GREEN
DC INPUT	5	RED 24 AWG		1 - RED
BI_DD-	1	GRAY TwPr4	8	
BI_DC-	7	CLEAR TwPr3	5	
BI_DC+	8	BLUE TwPr3	4	
DC RTN	9	BLACK 24 AWG		2 - BLACK
BI_DB+	10	WHITE TwPr2	3	
BI_DA-	11	PINK TwPr1	2	
BI_DA+	12	BROWN TwP1	1	
SHIELD	BODY PIN	SHIELD	SHIELD	SHIELD-5 - WHT

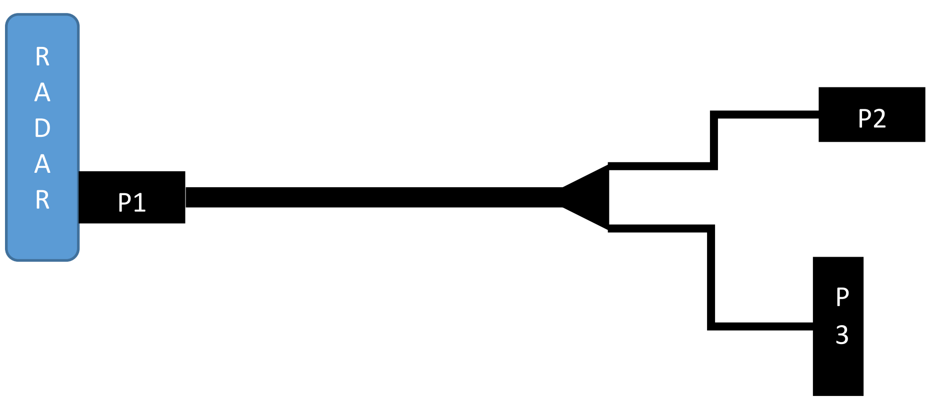


Figure 19: Cabling pin-out and diagram

9.2 Data Types and TCP/IP Data Ports

The radar transmits and receives data on 5 TCP ports: Command, Status, RVmap, Detection, Measurement and Track Port with each one dedicated to a particular type of data generated by the radar. Data is only transmitted to ports with user sockets connected. User may connect to all ports simultaneously or select only the ports desired for the data type in use. Connecting to the RVmap port will require the requisite data rate and speed so only connect to the ports planned to be read.

- The default netmask is fixed at **255.255.0.0**. with Static IP address factory set to **169.254.1.10**. If you are using more than a single radar connected to the network, you will want to create unique IP addresses. See the ETH:IP command to change the units IP address and netmask.
- ***WARNING: Do not lose the IP address as this will require a hardware reset to revert to default IP.***
- Multiple radars can be placed on a switch if they are set to separate IP addresses on the same sub-mask. Also, it should be reiterated that the unit has a static IP address and no DHCP server function. This generally prevents easily sharing a NIC card through a switch between a radar and corporate/general internet connection. Echodyne recommends either connecting to your corporate/internet connections through a Wi-Fi connection, or through a separate NIC card from the radar. In our corporate PC's we typically install dual NIC cards to allow simultaneous radar operation and corporate network access within a single computer. Typical stand-alone computer ports DHCP configured will default to suitable sub-mask to connect to the radar static IP.
- Within 60 seconds the radar should completely boot up and be ready to communicate through the Ethernet ports. Verify LED indicator lights are flashing and communicating with the port connection and <u>Right-hand light is on steady</u> (right-hand light confirms unit negotiation to 1000BT, left-hand light confirms unit negotiation 100BT, both LEDs on indicate a fault, and neither LED on indicates a potential problem with Ethernet connection).
- The data in the radar user packets, defined below, come out of the TCP Socket as little endian. Thus, on a little-endian processor, calling ntohl on the received data is not necessary.
  - Ref: https://en.wikipedia.org/wiki/Endianness#Endianness_in_networking
- '**Command Port', Port 23,** receives user commands and sends replies to user commands. Data is only sent in response to user commands; no data is streamed on this port. Port 23 will be deprecated in a future release. For SW18.0, both Port 23 and Port 29978 will be viable.
- '**Command Port', Port 29978,** receives user commands and sends replies to user commands. Data is only sent in response to user commands; no data is streamed on this port. In a future release, this will take over for port 23 as port 23 will be deprecated.
- **'Status Port', Port 29979,** transmits system status information every 50 milliseconds unless throttled.
- **'RVmap Port', Port 29980,** transmits range-velocity data with each packet containing data from a single beam direction.
- **'Detections Port', Port 29981** transmits detection data every beam step, if any new detections have been made.
- **'Measurements Port', Port 29984** transmits measurement information at the user specified track update rate.
- **'Track Port', Port 29982** transmits track information at the user specified track update rate.

The size of the detections, measurements and tracks packets vary with the number of items reported. The maximum data rate across all ports is the sum of the individual data rates on the following pages.

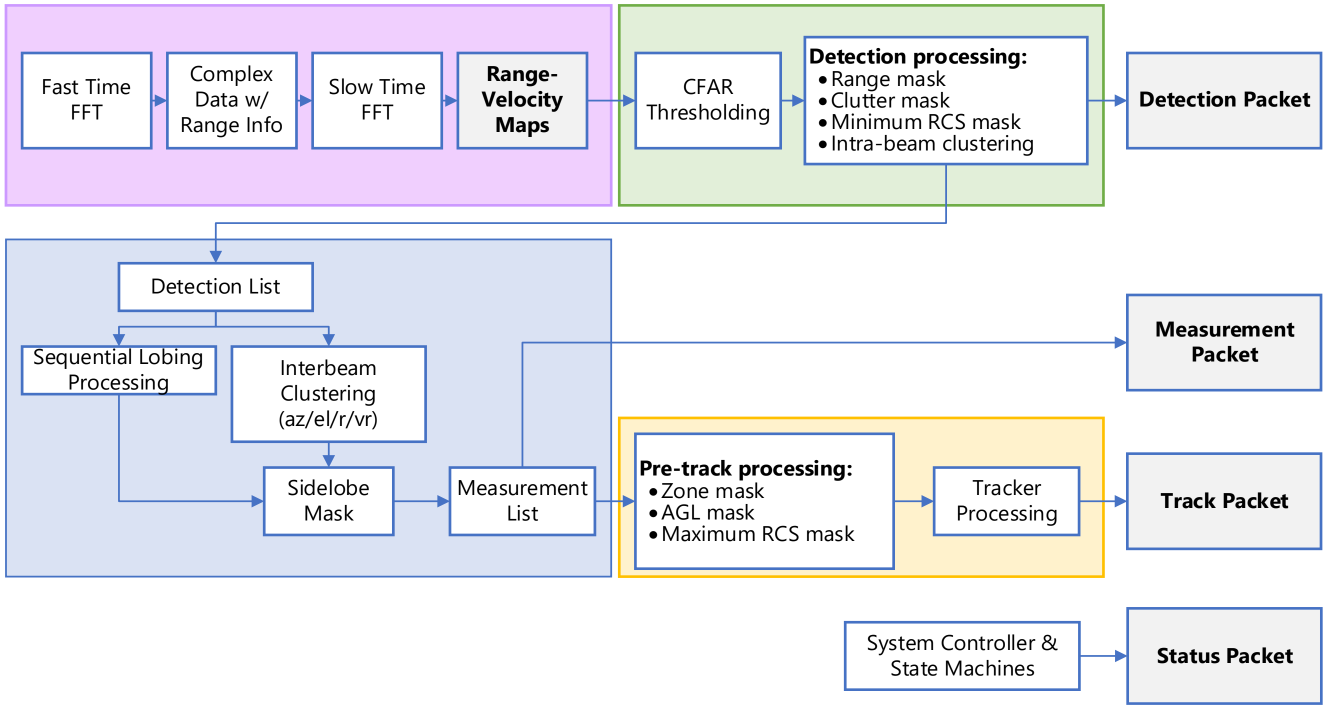


Figure 20: Processing Pipeline with Masking Order and Data Packet Generation

9.3 System Status Data Packet (Port 29979)

Contains data about current system state. Transmitted every 50 milliseconds (20 Hz) once the system is booted up and not in the Initialization state unless modified by the OUTPUT:STATUSRATE command. The mean payload data rate is 6.83594 Kbytes/sec. <u>It is strongly recommended that you monitor this port for awareness of critical radar operational states (for example, SWT, TCM, Ethernet speed, orientation).</u>

Table 13: System Status Data Packet – Status Port (29979)

System Status TCP Packet – Status Port (29979)		
Total Size *(352 bytes)*	Packet Start Tag *(12 bytes)*	12 bytes (char) <syststatus>
	Header *(24 bytes)*	(uint32) total # of bytes in this data packet, incl. Packet Start Tag
		(8 bytes RESERVED_HEADER0)
		4 bytes (char) packet schema version (X.X.X.X)
		8 bytes (char) radar serial number
	Data Block *(316 bytes)*	[(uint32) System state] 0 = Reset 1 = Init 2 = Idle 3 = Command Executing 4 = Search *<- Desired end state if Search Mode* 5 = SWT *<- Desired end state if SWT Mode* 6 = Error 7 = Upgrade 8 = Restart 9 = Interference Detection
		[(float32) search frame rate, FOV/sec]
		[(float32) platform height AGL, m]
		(4 bytes RESERVED0)
		[(float32) orientation quaternion x]
		[(float32) orientation quaternion y]
		[(float32) orientation quaternion z]
		[(float32) orientation quaternion w]
		[(uint32) system time, days]
		[(uint32) system time, milliseconds]
		[(float32) platform x-component of velocity over ground, m/s]
		[(float32) platform y-component of velocity over ground, m/s]
		[(float32) platform z-component of velocity over ground, m/s]
		[(uint32) Time Channel State] 0 = IDLE 1 = WAITING 2 = SEARCHING 3 = NO_CLEAR_TIME_CHANNEL 4 = CLEAR_LEADER ← this is the state you want for TCM Ch0 5 = LOCKED_FOLLOWER ← this is the state you want for Ch1-3 6 = LOST_TRACK_FOLLOWER 7 = TCM_ERROR
		(4 bytes RESERVED1) (TCM internal state machine)
		[(uint32) negotiated Ethernet speed] (0=1 Gbit/s, 1=100 Mbit/s, 2=10Mbit/s)
		(252 bytes RESERVED2)

Expressions in '<>' are ASCII character strings including the '<>' characters.
Expressions in '[]' are single values of the indicated data type.

9.4 Range-Velocity Map Data Packet (Port 29980)

Contains 2D range-velocity array data for a single beam direction. Transmitted every beam step period, ~7-8 msec. Only transmitted if the radar is in either BEAM, SEARCH, or SWT mode, and the port socket is connected. The RVmap 2D array data-block is arranged in range-major order. The mean payload data rate is 38108.9 kB/sec (38.1089 MB/sec). The bulk data of the 2D range-velocity data array is in linear amplitude.

Table 14: Range Velocity TCP Packet – RvMap Port (29980)

<table>
<tr><th colspan="4">Range Velocity TCP Packet – RVmap Port (29980)</th></tr>
<tr><td rowspan="25">Total Size
262252 bytes
(subject to change for different waveforms)</td><td>Packet Start Tag
(16 bytes)</td><td colspan="2"><rangevelocitym></td></tr>
<tr><td rowspan="23">Header
(92 bytes)</td><td colspan="2">[(uint32) total # of bytes in this data packet, incl. Packet Start Tag]</td></tr>
<tr><td colspan="2">[(float) beam azimuth, degrees]</td></tr>
<tr><td colspan="2">[(float) beam elevation, degrees]</td></tr>
<tr><td colspan="2">[(uint32) trigger time, days]</td></tr>
<tr><td colspan="2">[(uint32) trigger time, milliseconds]</td></tr>
<tr><td colspan="2">[(float) dR, meters]</td></tr>
<tr><td colspan="2">[(float) nRanges]</td></tr>
<tr><td colspan="2">[(float) dV, m/s]</td></tr>
<tr><td colspan="2">[(float) nVelocities]</td></tr>
<tr><td colspan="2">[(float) platform orientation quaternion x]</td></tr>
<tr><td colspan="2">[(float) platform orientation quaternion y]</td></tr>
<tr><td colspan="2">[(float) platform orientation quaternion z]</td></tr>
<tr><td colspan="2">[(float) platform orientation quaternion w]</td></tr>
<tr><td colspan="2">[(float) search frame rate, FOV/sec]</td></tr>
<tr><td colspan="2">[(uint32) zero range bin number, (n_0)]</td></tr>
<tr><td colspan="2">[(uint32) zero Doppler bin number, (m_0)]</td></tr>
<tr><td colspan="2">[(float) platform height AGL, meters]</td></tr>
<tr><td colspan="2">[(float) platform x-component of velocity over ground, m/s]</td></tr>
<tr><td colspan="2">[(float) platform y-component of velocity over ground, m/s]</td></tr>
<tr><td colspan="2">[(float) platform z-component of velocity over ground, m/s]</td></tr>
<tr><td colspan="2">(11 bytes RESERVED0)</td></tr>
<tr><td rowspan="2">(uint8) Status Byte</td><td>bit [7:1] unused</td></tr>
<tr><td>bit 0, LSB, 1 if the ADC is saturated during collection of this RVmap</td></tr>
<tr><td>Data Block
(262144 bytes
(subject to change for different waveforms)</td><td colspan="2">[(uint32) s[0][0]]
[(uint32) s[1][0]]
[(uint32) s[2][0]]
…
[(uint32) s[nranges-1][0]]
[(uint32) s[0][1]]
[(uint32) s[1][1]]
…
[(uint32) s[n][m]]
…
[(uint32) s[nRanges-1][nVelocities-1]]</td></tr>
</table>

Expressions in ‘<>’ are ASCII character strings including the ‘<>’ characters.
Expressions in ‘[]’ are single values of the indicated data type.

9.5 Detections Data Packet (Port 29981)

Contains data related to CFAR detections for a given beam. Transmitted every beam step period, 7-8 msec, if the radar is in either BEAM (not yet implemented), SEARCH, or SWT mode, and the detection port socket is connected. Each detection collected by the radar is transmitted only one time. The number of detections included in the packet varies and consequently so does the packet length. If no detections are processed for a particular beam, only the NULL Detections Packet is transmitted. This NULL detection helps keep data in-sync with RVMap data if so desired by at least receiving a NULL packet for each beam direction. The maximum mean payload data rate is 936.454 kB/sec (0.936 MB/sec).

Table 15: Detections TCP Packet – Detections Port (29981)

Detections TCP Packet – Detections Port (29981)		
Variable total # of bytes Minimum of 108 bytes for one detection Maximum of 6444 bytes for 100 detections	Packet Start Tag *(12 bytes)*	<detections>
	Header *(32 bytes)*	[(uint32) total # of bytes in this data packet, incl. Packet Start Tag]
		[(uint32) nDetections, number of detections in this packet]
		[(uint32) beam purpose] *0=Search, 1=Unconf Track Upd, 2=Conf Track Upd, 3=TCM Link*
		[(float) Az of generating beam, deg]
		[(float) El of generating beam, deg]
		[(uint32) detection time, days]
		[(uint32) detection time, milliseconds]
		4 bytes RESERVED_HEADER
	Detection 1 Data Block *(64 bytes)*	[(uint32) detection time, days]
		[(uint32) detection time, milliseconds]
		[(float) power, dB]
		[(float) snr, dB]
		[(float) range, meters]
		[(float) Az, deg]
		[(float) El, deg]
		[(float) vRadial, m/s]
		[(float) range_interp, m]
		[(uint32) detection_ID]
		(4 bytes RESERVED0)
		[(float) RCS, dBsm]
		(16 bytes RESERVED1)
	Detection Data Blocks 2 to nDetections nDetections ≤ 100 *(64 bytes per block)*	…

NULL Detections TCP Packet – Detections Port		
Total Size *(44 bytes)*	Packet Start Tag *(12 bytes)*	<detections>
	Header *(32 bytes)*	[(uint32) total # of bytes in this data packet, incl. Packet Start Tag]
		[(uint32) nDetections, 0x0000]
		[(float) search frame rate, FOV/sec]
		[(float) Az of generating beam, deg]
		[(float) El of generating beam, deg]
		[(uint32) detection time, days]
		[(uint32) detection time, milliseconds]
		(4 bytes RESERVED)

Expressions in '<>' are ASCII character strings including the '<>' characters.
Expressions in '[]' are single values of the indicated data type.

9.6 Measurements Data Packet (Port 29984)

Contains data about measurements created during the current track update period. Measurements are refined detections from inter and intrabeam clusters, sidelobe clusters as well as sequential lobing beams. Measurements are transmitted at the same rate as tracks (at the user specified rate between [1.0, 10.0] times per second). Measurements are only transmitted if the radar is in SWT mode and the measurement port socket is connected. The number of measurements included in the packet varies and so does the packet length. If there are no measurements, only the NULL Measurements Packet is transmitted. The maximum mean payload data rate is 950.6 kB/sec (0.928 MB/sec).

Table 16: Measurements TCP Packet – Measurements Port (29984)

Measurements TCP Packet – Measurements Port (29984)		
Variable total # of bytes Minimum of 444 bytes for one measurement Maximum of 97,344 bytes for 256 measurements	Packet Start Tag *(16 bytes)*	<measurements23>
	Header *(48 bytes)*	[(uint32) total # of bytes in this data packet, incl. Packet Start Tag]
		[(uint32) nMeasurements, # of measurements in this packet]
		[(uint32) measurement time, days]
		[(uint32) measurement time, milliseconds]
		32 bytes RESERVED
	Measurement 1 Data Block *(380 bytes)*	[(uint32) measurement_ID]
		[(uint32) measurement type]
		[(uint32) measurement reject mask]
		[(float) azimuth, degrees]
		[(float) elevation, degrees]
		[(float) range, meters]
		[(float) rcsEst, dBsm]
		[(float) vradial, m/s]
		[(uint32) nDetections, # of detections used to create measurement]
		[(uint32)[64] detIDs, array of detection ID numbers used to create measurement]
		24 bytes RESERVED
		[(float) north, meters]
		[(float) up, meters]
		[(float) east, meters]
		52 bytes RESERVED
	Measurement Data Blocks 2 to nMeasurements nMeasurements ≤ 256 *(380 bytes per block)*	

Table 17: NULL Measurements TCP Packet – Measurements Port

NULL Measurements TCP Packet – Measurements Port		
Total Size *(64 bytes)*	Packet Start Tag *(16 bytes)*	<measurements23>
	Header *(48 bytes)*	[(uint32) total # of bytes in this data packet, incl. Packet Start Tag]
		[(uint32) nMeasurements, # of measurements in this packet]
		[(uint32) detection time, days]
		[(uint32) detection time, milliseconds]
		32 bytes RESERVED

Expressions in '<>' are ASCII character strings including the '<>' characters.
Expressions in '[]' are single values of the indicated data type.

9.7 Tracks Data Packet (Port 29982)

Contains data about active tracks. Transmitted at the user specified track update rate in the range [0.0, 10.0] per second unless adjusted by the OUTPUT:MAXZTRACKRATE command. Only transmitted if the radar is in SWT mode and the port socket is connected. The number of tracks included in the packet varies and so does the packet length. If there are no active tracks, only the NULL Tracks Packet is transmitted. The maximum mean payload data rate is 24.4 kB/sec.

Table 18: Tracks TCP Packet – Tracks Port (29982)

Tracks TCP Packet – Tracks Port (29982)		
Variable total # of bytes Minimum of 168 bytes for 1 active track Max of 2600 bytes for 20 active tracks	Packet Start Tag *(12 bytes)*	<tracktrack>
	Header *(28 bytes)*	[(uint32) total # of bytes in this data packet, incl. Packet Start Tag]
		[(uint32) nTracks, number of tracks in this packet]
		[(uint32) system time, days]
		[(uint32) system time, milliseconds]
		(8 bytes RESERVED_HEADER)
		[(uint32) packet_type] 0 = legacy, 1 = extended
	Active Track 1 Data Block *(128 bytes)*	[(uint32) ID]
		[(uint32) state]
		[(float) azest, degrees]
		[(float) elest, degrees]
		[(float) rest, meters]
		[(float) xest, meters]
		[(float) yest, meters]
		[(float) zest, meters]
		[(float) velxest, m/s]
		[(float) velyest, m/s]
		[(float) velzest, m/s]
		[(uint32) [3], associated measurement IDs]
		[(float) [3], associated measurement Chi^2 statistic]
		[(int32) TOCA, days]
		[(int32) TOCA, milliseconds]
		[(float) DOCA, meters]
		[(float) lifetime, track update periods]
		[(uint32) lastUpdateTime, days]
		[(uint32) lastUpdateTime, milliseconds]
		[(uint32) lastAssociatedDataTime, days]
		[(uint32) lastAssociatedDataTime, milliseconds]
		[(uint32) acquiredTime, days]
		[(uint32) acquiredTime, milliseconds]
		[(float) confidenceLevel, 0-100]
		[(uint32) # of measurements associated to this track in the update period]
		[(float) estRCS, dBsm]
		[(float) prob_unknown (0.0-1.0) or NaN], probability that the track results from an unknown class, ; NaN if classifier is disabled
		[(float) prob_uav (0.0-1.0) or NaN], probability that the track results from a UAV; NaN if classifier is disabled
	Active Track Data Blocks 2 to nTracks *(128 bytes per block)*	

NULL Tracks TCP Packet – Tracks Port		
Total Size *(40 bytes)*	Packet Start Tag *(12 bytes)*	<tracktrack>
	Header *(28 bytes)*	[(uint32) total # of bytes in NULL packet, incl Packet Start Tag -> 0x0028]
		[(uint32) nTracks, number of tracks in this packet -> 0x0000]
		[(uint32) system time, days]
		[(uint32) system time, milliseconds]
		(12 bytes RESERVED_HEADER)

Expressions in ‘<>’ are ASCII character strings including the ‘<>’ characters.
Expressions in ‘[]’ are single values of the indicated data type

10. USER COMMANDS

10.1 IDN?

*IDN?	
(Read) Returns a string that uniquely identifies the serial number and firmware configuration of the MESA unit.	
Write Parameters	Not Applicable.
Write Syntax	Not Applicable.
Read Syntax	***IDN?**
Read Returns	Information about the MESA unit. The string is of the form: *ECHODYNE Corp. EchoGuard Radar* *COUNTRY MODE: USA* *OPERATION MODE: Pedestrian* *Serial Number: "001044"* *SW Suite: 16.1.0* *MCU Firmware version: 20.7.D.0.5* *MCU Firmware build date: Sep 14 2020 22:41:13* *FPGA Firmware version: 9D14* *FPGA ID: A6* *FPGA Time Stamp: Thu Jul 16 17:27:51 2020* *HW CONFIG: BLOCK1* **Where:** • Model type • Country Mode • Operation Mode is selected parameter group for desired mode • Serial # should correspond to sticker label on outside of unit • SW Suite is release • MCU Firmware version is 'HPS' image for ARM9 processor • FPGA Firmware version is image for waveform • HW CONFIG is the capability level of the underlying hardware (for example, BLOCK0, BLOCK1)
Examples	>> *IDN? ECHODYNE Corp. EchoGuard Radar COUNTRY MODE: USA OPERATION MODE: Pedestrian Serial Number: "001044" SW Suite: 16.1.0 MCU Firmware version: 20.7.D.0.5 MCU Firmware build date: Sep 14 2020 22:41:13 FPGA Firmware version: 9D14 FPGA ID: A6 FPGA Time Stamp: Thu Jul 16 17:27:51 2020 HW CONFIG: BLOCK1 Please report issues to help@echodyne.com OK

10.2 SYSPARAM?

SYSPARAM?	
(Read) - Returns a list of all settable system parameters accessible by 'user' privilege level commands, and their current state. Derivative parameters such as range resolution are also reported. Dynamically changing parameters such as PCB temperature are not reported. All values are printed in ASCII. Configurable parameters are printed in the syntax of the command that sets that value, followed by the units of the parameter as COMMAND <value1>,…,<valuen> (units) NOTE: Derivative parameters such as range and velocity resolution queried after a boot-up but before a START command is issued may exhibit slightly different values than the actual parameters that take effect during actual scanning (for example, after a START command is issued). This is because these parameters at boot-up are only estimates and the actual derived parameters are fine-tuned after a handshake with the hardware takes place following a START command for a given mode.	
Write Parameters	n/a
Write Syntax	n/a
Read Syntax	SYSPARAM?
Read Returns	>> SYSPARAM? GETSERIAL 000699 Configurable parameters: ETH:IP 169.254.1.10 MAC 34.4C.C8.00.02.BB NETMASK 255.255.0.0 GATEWAY 0.0.0.0 DMS:CHANNEL 1 MODE:SWT:OPERATIONMODE 0 (0 = Pedestrian, 1 = UAS, 2 = Plane) MODE:SEARCH:AZFOVMIN -60.000000 (deg) MODE:SEARCH:AZFOVMAX 60.000000 (deg) MODE:SEARCH:AZSTEP 2.000000 (deg) MODE:SEARCH:ELFOVMIN -40.000000 (deg) MODE:SEARCH:ELFOVMAX 40.000000 (deg) MODE:SEARCH:ELSTEP 2.000000 (deg) MODE:SWT:SEARCH:AZFOVMIN -60.000000 (deg) MODE:SWT:SEARCH:AZFOVMAX 60.000000 (deg) MODE:SWT:SEARCH:ELFOVMIN 0.000000 (deg) MODE:SWT:SEARCH:ELFOVMAX 12.000000 (deg) MODE:SWT:TRACK:AZFOVMIN -60.000000 (deg) MODE:SWT:TRACK:AZFOVMAX 60.000000 (deg) MODE:SWT:TRACK:ELFOVMIN -40.000000 (deg) MODE:SWT:TRACK:ELFOVMAX 40.000000 (deg) RSP:RCSMASK:MINRCS -30.000000 (dB) RSP:RCSMASK:MAXRCS 100.000000 (dB) Derivative parameters: dR 3.253038 (meter) nRanges 2048 (units) dV 0.908293 (units) nVelocities 32 (units) zero range bin number, n0 128 (units) zero velocity bin number, m0 16 (units) OK

10.3 SYSPARAM:JSON?

SYSPARAM:JSON?	
(Read) - Returns a list of all settable system parameters accessible by user commands, and their current state in a JSON structured format with whitespaces for easier automatic software ingest and parsing. Derivative parameters such as range resolution are also reported. Dynamically changing parameters such as PCB temperature are not reported. (**NOTE:** Derivative parameters such as range and velocity resolution queried after a boot-up but before a START command is issued may exhibit slightly different values than the actual parameters that take effect during actual scanning (for example, after a START command is issued). This is because these parameters at boot-up are only estimates and the actual derived parameters are fine-tuned after a handshake with the hardware takes place following a START command for a given mode).	
Write Parameters	n/a
Write Syntax	n/a
Read Syntax	SYSPARAM:JSON?

Read Returns	>> SYSPARAM:JSON?
	{
	"schema_version":"0.1.0",
	"GETSERIAL":{
	"value":"000699",
	"description":"Serial Number",
	"units":"N/A"
	},
	"ETH:IP":{
	"value":"169.254.1.10",
	"description":"Ethernet IP Address",
	"units":"N/A"
	},
	"ETH:MAC":{
	"value":"34.4C.C8.00.02.BB",
	"description":"MAC Address",
	"units":"N/A"
	},
	
	… many more fields individually delimited by { },
	
	"dV":{
	"value":0.908293,
	"description":"Velocity Bin Size",
	"units":"meters per second"
	},
	"nVelocities":{
	"value":32,
	"description":"Number of Velocity Bins",
	"units":"none"
	},
	"zeroRangeBinNumber":{
	"value":128,
	"description":"Range Bin at Zero Range",
	"units":"none"
	},
	"zeroVelocityBinNumber":{
	"value":16,
	"description":"Velocity Bin at Zero Velocity",
	"units":"none"
	}
	}
	OK

10.4 OUTPUT:EXTENDEDPACKET:ENABLE

OUTPUT:EXTENDEDPACKET:ENABLE [?]	
(Read/Write) Enable/disable extended track packets with additional information regarding state of target. The extended track packets have the same body as the legacy packet but append additional statistical EKF state information for each active track. The 'packet_type' field in the track packet header is controlled by this flag and represents if legacy or extended mode is used for a particular packet.	
Write Parameters	<flag> - (string), [TRUE,FALSE]. Flag to enable/disable extended track packets
Write Syntax	OUTPUT:EXTENDEDPACKET:ENABLE [TRUE,FALSE]
Read Syntax	OUTPUT:EXTENDEDPACKET:ENABLE?
Read Returns	The flag state 0 (FALSE) or 1 (TRUE)
Factory Default Value	1
Examples	>> OUTPUT:EXTENDEDPACKET:ENABLE? 1 OK

10.4.1 Extended Track Packet Format

Tracks TCP Packet – Tracks Port (29982)		
Variable total # of bytes Minimum of 464 bytes for 1 active track Max of 8,520 bytes for 20 active tracks	Packet Start Tag *(12 bytes)*	<tracktrack>
	Header *(28 bytes)*	*Standard track header content*
	Active Track 1 Data Block **(Legacy Content)** *(128 bytes)*	*Legacy track data block content (see section 9.7 for definition)*
	Active Track 1 Data Block **(Extended Content)** *(296 bytes)*	[(float)[7] aggregate Trk Class Probs] CL_OTHER CL_UAV CL_AIRCRAFT CL_BIRD CL_CLUTTER CL_VEHICLE CL_WALKER
		[(float)[2] classification placeholder]
		[(uint32) last beam time, days]
		[(uint32) last beam time, msec]
		[(float)[36] 6x6 matrix, a posteriori estimate covariance matrix, row major] Pest[NORTH_POS][NORTH_POS] Pest[NORTH_POS][NORTH_VEL] Pest[NORTH_POS][UP_POS] Pest[NORTH_POS][UP_VEL] Pest[NORTH_POS][EAST_POS] Pest[NORTH_POS][EAST_VEL] Pest[NORTH_VEL][NORTH_POS] Pest[NORTH_VEL][NORTH_VEL] Pest[NORTH_VEL][UP_POS] Pest[NORTH_VEL][UP_VEL] Pest[NORTH_VEL][EAST_POS] Pest[NORTH_VEL][EAST_VEL] Pest[UP_POS][NORTH_POS] Pest[UP_POS][NORTH_VEL] Pest[UP_POS][UP_POS] Pest[UP_POS][UP_VEL] Pest[UP_POS][EAST_POS] Pest[UP_POS][EAST_VEL] Pest[UP_VEL][NORTH_POS] Pest[UP_VEL][NORTH_VEL] Pest[UP_VEL][UP_POS]

		Pest[UP_VEL][UP_VEL] Pest[UP_VEL][EAST_POS] Pest[UP_VEL][EAST_VEL] Pest[EAST_POS][NORTH_POS] Pest[EAST_POS][NORTH_VEL] Pest[EAST_POS][UP_POS] Pest[EAST_POS][UP_VEL] Pest[EAST_POS][EAST_POS] Pest[EAST_POS][EAST_VEL] Pest[EAST_VEL][NORTH_POS] Pest[EAST_VEL][NORTH_VEL] Pest[EAST_VEL][UP_POS] Pest[EAST_VEL][UP_VEL] Pest[EAST_VEL][EAST_POS] Pest[EAST_VEL][EAST_VEL]
		[(float) range measurement variance, meters^2]
		[(float) vradial measurement variance, (m/s)^2]
		[(float) vradial association variance, (m/s)^2]
		[(float) innovation estimate, Running Chi2 statistic of data match to model]
		[(float) state variance scaling factor]
		[(uint32) [3], additional associated measurement IDs]
		[(float) [3], additional associated measurement Chi^2 statistic]
		[(float) SNR Estimate (dB)]
		60 bytes RESERVED1 (future growth)
	Active Track Data Blocks 2 to nTracks w/extended content *(424 total bytes per block)*	Legacy + Extended Track data

10.5 GETSERIAL

GETSERIAL	
(Read) This will report the serial number of the radar unit.	
Read Syntax	GETSERIAL
Read Returns	Serial number of radar unit. Serial numbers are six-digit numbers, greater than 0.
Examples Reports that the serial number of the unit is "000123"	>> GETSERIAL Serial Number: "000123" OK

10.6 *TST?

*TST?	
(Read) Returns a string that shows information about the unit and an error log. This response is useful to provide with SYSPRARAM? report for support calls to resolve unexpected behavior.	
Write Parameters	n/a
Write Syntax	n/a
Read Syntax	*TST?
Read Returns	List of system status values
Examples	>> *TST? --------------- SYSTEM INIT STATUS--------------- SDRAM: 0 GLOBAL INTERRUPT INIT: 0 CPU INTERRUPT INIT: 0 GLOBAL TIMER: 0 FPGA: 0 SPI: 0 GPIO INIT: 0 GPIO CFG: 0 CPU INTERRUPT ENABLE: 0

	GLOBAL INTERRUPT ENABLE: 0 EMMC INIT: 0 ETHERNET INIT: 0 I2C INIT: 0 IMU INIT: 0 CACHE INIT: 0 ----------------- SYSTEM STATE ------------------ CURRENT STATE: SYSTEM_STATE_STANDBY --------------- ETHERNET STATUS ----------------- SPEED: 1000 Mb/s Filter log level: ERROR(4) Filter mask: FFFFFFFF ------------------- Saved Log Begins ------------------- ------------------- Saved Log Ends ------------------- OK

10.7 FPGA?

FPGA?	
(Read) This command returns the version of FPGA installed.	
Read Syntax	FPGA?
Read Returns	Returns A5 (legacy) or A6 (upgraded)
Example	>> FPGA? FPGA ID: A5 OK

10.8 ETH:IP

<table>
<tr><th colspan="2">ETH:IP[?] <ip-address> [<netmask> <default-gateway>]</th></tr>
<tr><td colspan="2">(Read/Write) This command <u>persistently</u> sets the Ethernet IP address and, optionally, the netmask and default gateway.

Caution: This setting is persistent and can only be reset to the default value through a Hardware Reset Operation. If the IP is lost/forgotten, it will not be possible to communicate with the radar until reset to default.</td></tr>
<tr><td>Write Parameters</td><td><address> - Ethernet IP Address
<netmask> - Optional netmask. If not supplied the previous netmask is retained
<default-gateway> - Optional default gateway. Required if netmask was provided. If not supplied, the previous default gateway it retained. Use 0.0.0.0 to clear.
The default address is 169.254.1.10. The default netmask is 255.255.0.0. And the default gateway is 0.0.0.0.</td></tr>
<tr><td>Write Syntax</td><td>ETH:IP <address>
-or-
ETH:IP <address> <netmask> <default-gateway></td></tr>
<tr><td>Read Syntax</td><td>ETH:IP?</td></tr>
<tr><td>Read Returns</td><td>The currently active Ethernet IP Address, netmask, and default gateway.
Note that the address set with ETH:IP <address> will not take effect until the next reboot.</td></tr>
<tr><td>Factory Default Value</td><td>IP address: 169.254.1.10; Netmask: 255.255.0.0; Default gateway: 0.0.0.0</td></tr>
<tr><td>Examples
Setting Ethernet IP Address:
Query:</td><td>>> ETH:IP 10.0.32.38 255.255.128.0 10.0.32.1
OK
>> ETH:IP?
Current Ethernet configuration:
 IP Address   : 169.254.1.10
 Netmask   : 255.255.0.0
 Default gateway: 0.0.0.0
Ethernet configuration on next boot:
 IP Address   : 10.0.32.38
 Netmask   : 255.255.128.0
 Default gateway: 10.0.32.1
OK

(after a reboot)
>> ETH:IP?
Current Ethernet configuration:
 IP Address   : 10.0.32.38
 Netmask   : 255.255.128.0
 Default gateway: 10.0.32.1
Ethernet configuration on next boot:
 IP Address   : 10.0.32.38
 Netmask   : 255.255.128.0
 Default gateway: 10.0.32.1
OK</td></tr>
</table>

10.9 ETH:MAC?

<table>
<tr><th colspan="2">ETH:MAC?</th></tr>
<tr><td colspan="2">(Read) This will report the MAC address of the radar unit.</td></tr>
<tr><td>Read Syntax</td><td>ETH:MAC?</td></tr>
<tr><td>Read Returns</td><td>Returns 6 octet MAC address.</td></tr>
<tr><td>Example</td><td>MESA-000699 >> ETH:MAC?
MAC Address: 34.4C.C8.00.02.BB
OK</td></tr>
</table>

10.10 SYS:TIME

SYS:TIME[?] <days>,<milliseconds>

(Read/Write) This command can be used to query the current radar system time (boot time + offset) or actively set the system time offset. At system boot, the radar time offset register is (0,0) and the system clock also initializes to at 0 days, 0 milliseconds and begins accumulating immediately. **This {day, millisecond} value is recorded in the headers of all data packets (tracks, status, etc.) and it is <u>strongly recommended</u> to be correctly set to the time base of your choosing for easy time alignment of radar outputs. The RadarUI follows this method to set radar time to UTC POSIX.**

Anytime you query SYS:TIME?, it will return the running boot clock time + time offset. If the offset has not been actively set, then it is still {0,0} and SYS:TIME? will return the system time since boot-up.

To set the time, best practice is to rapidly execute the following steps:
1) Issue a SYS:TIME 0,0 to ensure the offset register is cleared.
2) Query the local system time (in UTC POSIX or time zone/epoch of your choosing)
3) Query the radar time since boot-up by issuing a SYS:TIME?
4) Subtract the radar boot time from the local system time to calculate the correct {days, millisecond} offset to program.
5) Set the radar time by issuing a SYS:TIME {day offset, millisecond offset}
6) Query the radar via SYS:TIME? to verify the correct {days, msec} value is reported.

NOTE – While execution of the preceding steps may result in minor offsets (single digit msec range) from true local time, the effect is considered insignificant compared with track error terms. Contact Echodyne for sample Python code.

Write Parameters	<days> - (unsigned integer), [0, 2^32-1], the offset in days to apply to the boot time of the system. <milliseconds> - (unsigned integer), [0, 86399999], the offset in number of milliseconds since the start of the day to apply to the boot time of the system.
Write Syntax	SYS:TIME <days>,<milliseconds>
Read Syntax	SYS:TIME?
Read Returns	The current system time (boot clock + offset)
Factory Default Value	Offset = {0, 0}
Example Clear any prior offset by setting to 0,0 Query the current system time (returns 1 day and 30 minutes after bootup): Set the current system time to 6:00:01 July 13th, 2017 UTC Query the current time (returns 6:05:30 July 14th, 2017 UTC):	 >> SYS:TIME 0,0 >> SYS:TIME? 1, 1800000 OK >> SYS:TIME 17360,21601000 OK >> SYS:TIME? 17361, 21930000 OK

10.11 LIST

LIST	
(Read) Lists commands available in the current context (IDLE / SEARCH / SWT) and password privilege level. For example, when operating in MODE:SWT, one can use LIST to determine which commands are still available.	
Read/Write Parameters	None
Read Syntax	LIST
Examples	>> LIST The following commands are available: 0 RESET:PARAMETERS 1 SYSPARAM? … *(many more commands removed for brevity)* 47 SYS:TIME? 48 SYS:TIME OK

10.12 MODE:SWT:OPERATIONMODE

MODE:SWT:OPERATIONMODE[?] <value>			
(Read/Write) Sets the Operation Mode of the MESA radar. Operation modes are designed to provide optimal tracking for different classes of targets. Each Operation Mode sets a variety of parameters to the best-known values for tracking certain target types. These Operation Mode parameters can be overwritten (further customized) by the user for specific scenarios. **NOTE**: Operation Mode must be set initially before any other parameters are changed to avoid Operation Mode defaults overriding user customized settings. Precede this command with a RESET:PARAMETERS to clear all settings to a default.			
		PRODUCT MODE (set at factory)	
		EchoGuard (SSR)	**EchoFlight (DAA)**
Write Parameters	**0 (default)**	ground walkers	small drones
	1	small drones (good general-purpose mode)	small crewed aircraft
	2	small crewed aircraft	n/a
Write Syntax	MODE:SWT:OPERATIONMODE <OperationMode value>		
Read Syntax	MODE:SWT:OPERATIONMODE?		
Read Returns	Current operation mode of the tracker.		
Factory Default Value	0		
Example	>> MODE:SWT: OPERATIONMODE 0 OK >> MODE:SWT: OPERATIONMODE? 0 OK *Now modify a parameter that is part of an operation mode.* >> MODE:SWT: OPERATIONMODE? 0 (but modified by user) OK		

10.13 RESET:PARAMETERS

RESET:PARAMETERS	
(Read/Write) This command resets all system parameters, *except the selected boot image, serial number, and network parameters*, back to their factory default values including operation mode and any range and zone masks currently specified. It is recommended to issue this command at the beginning of a radar set-up SCRIPT to place the radar in a known configuration before specifying OPERATION MODE and further customizations. See RadarUI generated script for example usage.	
Write Parameters	None.
Write Syntax	RESET:PARAMETERS
Read Syntax	Not Applicable
Read Returns	Not Applicable
Examples Setting factory defaults:	>> RESET:PARAMETERS NOTE: Reset persistent configuration data to factory defaults (Configuration version: 49). OK

10.14 RESET:SYSTEM

RESET:SYSTEM	
(Write) Perform a reboot reset for the HPS. The connection to the Ethernet port will be lost and will need to be reconnected. System reset takes approximately 30 seconds. When using the BNET executable, issue a 'reconnect' command after the system completes the reboot process.	
Write Parameters	None
Write Syntax	RESET:SYSTEM
Examples	>> RESET:SYSTEM OK

10.15 DMS:CHANNEL

DMS:CHANNEL[?] <channel>	
(Read/Write) This command sets the frequency band {0, 1, 2} of the radar which is controlled by the DMS channel. This command is <u>persistent,</u> and the radar will use the selected channel when the next MODE:*:START command is issued and will continue to use the selected channel until it is explicitly changed. The RESET:PARAMETERS command will reset the selected channel to the default.	
Write Parameters	<channel> - indicates the DMS Frequency Channel. - Unsigned integer - valid settings are 0, 1, and 2. - 0 is the first 45MHz band - 1 is the second 45MHz band - 2 is the third 45MHz band
Write Syntax	DMS:CHANNEL <channel>
Read Syntax	DMS:CHANNEL?
Read Returns	The current Channel Selected.
Factory Default Value	1
Examples Setting to Channel 1: Query:	>> DMS:CHANNEL 1 OK >> DMS:CHANNEL? Channel: 1. OK

10.16 MODE:SEARCH:START

MODE:SEARCH:START	
(Action Only) Begins executing SEARCH mode using the current configuration. Executing will continue, without allowing configuration variables to be changed, until MODE:SEARCH:STOP is received. During SEARCH execution, the radar will stream subscribed search data out to the user.	
Write Parameters	Not Applicable.
Write Syntax	MODE:SEARCH:START
Read Syntax	Not Applicable.
Read Returns	Not Applicable.
Example	>> MODE:SEARCH:START OK

10.17 MODE:SEARCH:STOP

MODE:SEARCH:STOP	
(Action Only) Stop executing SEARCH functionality and return radar to idle state where the MODE:SEARCH:START command can be issued again without having to issue any other commands. No effect if radar is not executing SEARCH mode.	
Write Parameters	Not Applicable.
Write Syntax	MODE:SEARCH:STOP
Read Syntax	Not Applicable.
Read Returns	Not Applicable.
Example	>> MODE:SEARCH:STOP OK

10.18 MODE:SWT:START

MODE:SWT:START	
(Action Only) Begin executing search while track (SWT) mode using the current configuration. Executing will continue, without allowing configuration variables to be changed, until MODE:SWT:STOP is received. During SWT execution, the radar will stream subscribed search and track data out to the user.	
Write Parameters	Not Applicable.
Write Syntax	MODE:SWT:START
Read Syntax	Not Applicable.
Read Returns	Not Applicable.
Example	>> MODE:SWT:START OK

10.19 MODE:SWT:STOP

MODE:SWT:STOP	
(Action Only) Stop executing SWT functionality and return the radar to Idle state where the MODE:SWT:START command can be issued again without having to issue any other commands. No effect if the radar is not executing SWT.	
Write Parameters	Not Applicable.
Write Syntax	MODE:SWT:STOP
Read Syntax	Not Applicable.
Read Returns	Not Applicable.
Example	>> MODE:SWT:STOP OK

10.20 MODE:SEARCH:AZFOVMIN

MODE:SEARCH:AZFOVMIN[?] <value>

(Read/Write) Update the SEARCH azFovMin value. Scheduled beams will be multiples of two even if this is set to an odd numbered value.	
Write Parameters	<value> - (integer), [-60 to 60] degrees. Must be less than azFovMax.
Write Syntax	MODE:SEARCH:AZFOVMIN <AzFOVMin value>
Read Syntax	MODE:SEARCH:AZFOVMIN?
Read Returns	<AzFOVMin value>
Factory Default Value	-60
Example	>> MODE:SEARCH:AZFOVMIN -30 OK >> MODE:SEARCH:AZFOVMIN? -30 OK

10.21 MODE:SEARCH:AZSTEP

MODE:SEARCH:AZSTEP[?] <value>	
(Read/Write) Update the SEARCH azStep value. Multiples of 2 degrees.	
Write Parameters	<value> - (integer), [2 to 120, multiples of 2 only] degrees.
Write Syntax	MODE:SEARCH:AZSTEP <stepSize value>
Read Syntax	MODE:SEARCH:AZSTEP?
Read Returns	The current azimuth beam step size used for search mode.
Factory Default Value	2
Example	>> MODE:SEARCH:AZSTEP 10 OK >> MODE:SEARCH:AZSTEP? 10 OK

10.22 MODE:SEARCH:ELSTEP

MODE:SEARCH: ELSTEP[?] <value>	
(Read/Write) Update the SEARCH elStep value. Multiples of 2 degrees.	
Write Parameters	<value> - (integer), [2 to 80, multiples of 2 only] degrees.
Write Syntax	MODE:SEARCH: ELSTEP stepSize
Read Syntax	MODE:SEARCH: ELSTEP?
Read Returns	The current elevation beam step size used for search mode.
Factory Default Value	2
Example	>> MODE:SEARCH:ELSTEP 10 OK >> MODE:SEARCH:ELSTEP? 10 OK

10.23 MODE:SEARCH:ELFOVMIN

MODE:SEARCH:ELFOVMIN[?] <value>	
(Read/Write) Update the SEARCH elFovMin value. Scheduled beams will be multiples of two even if this is set to an odd numbered value.	
Write Parameters	<value> - (integer), [-40 to 40] degrees. Must be less or equal to elFovMax.
Write Syntax	MODE:SEARCH:ELFOVMIN <ElFOVMax value>
Read Syntax	MODE:SEARCH:ELFOVMIN?
Read Returns	<ElFOVMax value>
Factory Default Value	-40

Example	>> MODE:SEARCH:ELFOVMIN -30 OK >> MODE:SEARCH:ELFOVMIN? -30 OK

10.24 MODE:SEARCH:AZFOVMAX

MODE:SEARCH:AZFOVMAX[?] <value>	
(Read/Write) Update the SEARCH azFovMax value. Scheduled beams will be multiples of two even if this is set to an odd numbered value.	
Write Parameters	<value> - (integer), [-60 to 60] degrees. Must be greater than azFovMin.
Write Syntax	MODE:SEARCH:AZFOVMAX <AzFOVMax value>
Read Syntax	MODE:SEARCH:AZFOVMAX?
Read Returns	<AzFOVMax value>
Factory Default Value	60
Example	>> MODE:SEARCH:AZFOVMAX 30 OK >> MODE:SEARCH:AZFOVMAX? 30 OK

10.25 MODE:SEARCH:ELFOVMAX

MODE:SEARCH:ELFOVMAX[?] <value>	
(Read/Write) Update the SEARCH elFovMax value. Scheduled beams will be multiples of two even if this is set to an odd numbered value.	
Write Parameters	<value> - (integer), [-40 to 40] degrees. Must be greater than or equal to elFovMin.
Write Syntax	MODE:SEARCH:ELFOVMAX <ElFOVMax value>
Read Syntax	MODE:SWT:ELFOVMAX?
Read Returns	<ElFOVMax value>
Factory Default Value	40
Example	>> MODE:SEARCH:ELFOVMAX 30 OK >> MODE:SWT:ELFOVMAX? 30 OK

10.26 MODE:SWT:SEARCH:AZFOVMIN

MODE:SWT:SEARCH:AZFOVMIN[?] <value>	
(Read/Write) Update the SWT **search** field-of-view minimum azimuth boundary. Scheduled beams will be multiples of two even if this is set to an odd numbered value.	
Write Parameters	<value> - (integer), [-60 to 60] degrees. Must be less than or equal to swtSearchAzFovMax.
Write Syntax	MODE:SWT:SEARCH:AZFOVMIN <value>
Read Syntax	MODE:SWT:SEARCH:AZFOVMIN?
Read Returns	<swtSearchAzFovMin value>
Factory Default Value	-60
Example	>> MODE:SWT:SEARCH:AZFOVMIN -30 OK >> MODE:SWT:SEARCH:AZFOVMIN? -30 OK

10.27 MODE:SWT:SEARCH:AZFOVMAX

MODE:SWT:SEARCH:AZFOVMAX[?] <value>	
(Read/Write) Update the SWT **search** field-of-view maximum azimuth boundary. Scheduled beams will be multiples of two even if this is set to an odd numbered value.	
Write Parameters	<value> - (integer), [-60 to 60] degrees. Must be greater than or equal to swtSearchAzFovMin.
Write Syntax	MODE:SWT:SEARCH:AZFOVMAX <value>
Read Syntax	MODE:SWT:SEARCH:AZFOVMAX?
Read Returns	<swtSearchAzFovMax value>
Factory Default Value	60
Example	>> MODE:SWT:SEARCH:AZFOVMAX 30 OK >> MODE:SWT: SEARCH:AZFOVMAX? 30 OK

10.28 MODE:SWT:SEARCH:ELFOVMIN

MODE:SWT:SEARCH:ELFOVMIN[?] <value>	
(Read/Write) Update the SWT **search** field-of-view minimum elevation boundary. Scheduled beams will be multiples of two even if this is set to an odd numbered value.	
Write Parameters	<value> - (integer), [-40 to 40] degrees. Must be less than or equal to swtSearchElFovMax.
Write Syntax	MODE:SWT:SEARCH:ELFOVMIN <value>
Read Syntax	MODE:SWT:SEARCH:ELFOVMIN?
Read Returns	<swtSearchElFovMin value>
Factory Default Value	0 (Operation Mode = 0)
Example	>> MODE:SWT: SEARCH:ELFOVMIN -30 OK >> MODE:SWT: SEARCH:ELFOVMIN? -30 OK

10.29 MODE:SWT:SEARCH:ELFOVMAX

MODE:SWT:SEARCH:ELFOVMAX[?] <value>	
(Read/Write) Update the SWT search field-of-view maximum elevation boundary. Scheduled beams will be multiples of two even if this is set to an odd numbered value.	
Write Parameters	<value> - (integer), [-40 to 40] degrees. Must be greater than or equal to swtSearchElFovMin.
Write Syntax	MODE:SWT:SEARCH:ELFOVMAX <value>
Read Syntax	MODE:SWT:SEARCH:ELFOVMAX?
Read Returns	<swtSearchElFovMax value>
Factory Default Value	12 (Operation Mode = 0)
Example	>> MODE:SWT: SEARCH:ELFOVMAX 30 OK >> MODE:SWT: SEARCH:ELFOVMAX? 30 OK

10.30 MODE:SWT:TRACK:AZFOVMIN

MODE:SWT:TRACK:AZFOVMIN[?] <value>
(Read/Write) Update the SWT tracking field-of-view minimum azimuth boundary. Scheduled beams will be multiples of two even if this is set to an odd numbered value. Tracks outside this range will not be updated by the radar and will coast for less than or equal to the Max Out of Bounds Coast Time before being deleted by the tracker.

Write Parameters	<value> - (integer), [-60 to 60] degrees. Must be less than or equal to swtTrackAzFovMax.
Write Syntax	MODE:SWT:TRACK:AZFOVMIN <value>
Read Syntax	MODE:SWT:TRACK:AZFOVMIN?
Read Returns	<swtTrackAzFovMin value>
Factory Default Value	-60
Example	>> MODE:SWT:TRACK:AZFOVMIN -30 OK >> MODE:SWT: TRACK:AZFOVMIN? -30 OK

10.31 MODE:SWT:TRACK:AZFOVMAX

MODE:SWT:TRACK:AZFOVMAX[?] <value>	
(Read/Write) update the SWT tracking field-of-view maximum azimuth boundary. Scheduled beams will be multiples of two even if this is set to an odd numbered value. Tracks outside this range will not be updated by the radar and will coast for less than or equal to the Max Out Of Bounds Coast Time before being deleted by the tracker.	
Write Parameters	<value> - (integer), [-60 to 60] degrees. Must be greater than or equal to swtTrackAzFovMin.
Write Syntax	MODE:SWT:TRACK:AZFOVMAX <value>
Read Syntax	MODE:SWT:TRACK:AZFOVMAX?
Read Returns	<swtTrackAzFovMax value>
Factory Default Value	60
Example	>> MODE:SWT: TRACK:AZFOVMAX 30 OK >> MODE:SWT: TRACK:AZFOVMAX? 30 OK

10.32 MODE:SWT:TRACK:ELFOVMIN

MODE:SWT:TRACK:ELFOVMIN[?] <value>	
(Read/Write) Update the SWT tracking field-of-view minimum elevation boundary. Scheduled beams will be multiples of two even if this is set to an odd numbered value. Tracks outside this range will not be updated by the radar and will coast for less than or equal to the Max Out Of Bounds Coast Time before being deleted by the tracker.	
Write Parameters	<value> - (integer), [-40 to 40] degrees. Must be less than or equal to swtTrackElFovMax.
Write Syntax	MODE:SWT:TRACK:ELFOVMIN <value>
Read Syntax	MODE:SWT:TRACK:ELFOVMIN?
Read Returns	<swtTrackElFovMin value>
Factory Default Value	-40
Example	>> MODE:SWT: TRACK:ELFOVMIN -30 OK >> MODE:SWT: TRACK:ELFOVMIN? -30 OK

10.33 MODE:SWT:TRACK:ELFOVMAX

MODE:SWT:TRACK:ELFOVMAX[?] <value>	
(Read/Write) Update the SWT tracking field-of-view maximum elevation boundary. Scheduled beams will be multiples of two even if this is set to an odd numbered value. Tracks outside this range will not be updated by the radar and will coast for less than or equal to the Max Out Of Bounds Coast Time before being deleted by the tracker.	
Write Parameters	<value> - (integer), [-40 to 40] degrees. Must be greater than or equal to swtTrackElFovMin.
Write Syntax	MODE:SWT:TRACK:ELFOVMAX <value>
Read Syntax	MODE:SWT:TRACK:ELFOVMAX?

Read Returns	<swtTrackElFovMin value>
Factory Default Value	40
Example	>> MODE:SWT: TRACK:ELFOVMAX 30 OK >> MODE:SWT: TRACK:ELFOVMAX? 30 OK

10.34 RSP:RCSMASK:MINRCS

RSP:RCSMASK:MINRCS[?] <value>	
(Read/Write) Configures the minimum threshold used for RCS-based masking of detections. **Value must be ≤ MAXRCS**. When RCS masking is enabled, CFAR **detections** with an estimated RCS value less than this threshold are marked as rejected and not sent to the user nor used by the tracker. This parameter is dynamic and can be changed while radar is operating. NOTE: RCS values are not precisely calibrated except to the MESA radar system, therefore they should not be interpreted to correspond to the exact physical RCS values of target objects but are approximately correct over FoR.	
Write Parameters	<value> - (float) dBsm; minimum value of -50, max <= MACRCS value
Write Syntax	RSP:RCSMASK:MINRCS value
Read Syntax	RSP:RCSMASK:MINRCS?
Read Returns	The current minimum threshold value for RCS mask
Factory Default Value	-30 dBsm
Example	>> RSP:RCSMASK:MINRCS -14 OK >> RSP:RCSMASK:MINRCS? -14.000000 OK

10.35 RSP:RCSMASK:MAXRCS

RSP:RCSMASK:MAXRCS[?] <value>	
(Read/Write) Configures the minimum threshold used for RCS-based masking of detections. **Value must be ≥ MINRCS**. When RCS masking is enabled, CFAR **measurements** (post clustering and SQL interpolation) with an estimated RCS value greater than this threshold are marked as rejected and not sent to the user nor used by the tracker. This parameter is dynamic and can be changed while radar is operating. NOTE: RCS values are not precisely calibrated except to the MESA radar system, therefore they should not be interpreted to correspond to the exact physical RCS values of target objects but are approximately correct over FoR.	
Write Parameters	<value> - (float) dBsm; maximum value of 100, min >= MINRCS value
Write Syntax	RSP:RCSMASK:MAXRCS value
Read Syntax	RSP:RCSMASK:MAXRCS?
Read Returns	The current maximum threshold value for RCS mask
Factory Default Value	100 dBsm
Example	>> RSP:RCSMASK:MAXRCS? 100.000000 OK >> RSP:RCSMASK:MAXRCS 42 OK >> RSP:RCSMASK:MAXRCS? 42.000000 OK

10.36 DATAFORMAT:TRACK?

DATAFORMAT:TRACK?	
(Read) Returns a complete list of all current track packet fields parameters, hierarchy, datatype, size and brief description in a JSON structured format with whitespace for easier automatic software ingest and parsing.	
Write Parameters	n/a
Write Syntax	n/a
Read Syntax	DATAFORMAT:TRACK?
Read Returns	JSON formatted string as shown below (abbreviated for manual inclusion)
Example	>> DATAFORMAT:TRACK?

```
{
   "portNumber": 29982,
   "track_packet.packetStartTag": {
      "bytes": 12,
      "datatype":"char",
      "arraysize": 12,
      "description": "<tracktrack>"
   },
   "header": {
      "track_packet.packetSizeBytes": {
         "bytes": 4,
         "datatype": "uint32",
         "arraysize": 1,
         "description": "total number of bytes in this data packet"
      },

… additional fields omitted for brevity...

      "track_packet.reserved_header": {
         "bytes": 12,
         "datatype": "N/A",
         "arraysize": 1,
         "description": "reserved"
      }
   },
   "data": {
      "track_data.id": {
         "bytes": 4,
         "datatype": "uint32",
         "arraysize": 1,
         "description": "track ID"
      },
       "track_data.state": {
         "bytes": 4,
         "datatype": "uint32",
         "arraysize": 1,
         "description": "track state"
      },

… additional fields omitted for brevity...

      "track_data.prob_uav": {
         "bytes": 4,
         "datatype": "float",
         "arraysize": 1,
         "description": "0-1 or NaN"
      }
   }
}
OK
```

10.37 DATAFORMAT:DETECTION?

DATAFORMAT:DETECTION?	
(Read) Returns a complete list of all current detection packet fields parameters, hierarchy, datatype, size and brief description in a JSON structured format with whitespace for easier automatic software ingest and parsing.	
Write Parameters	n/a
Write Syntax	n/a
Read Syntax	DATAFORMAT:DETECTION?
Read Returns	JSON formatted string as shown below (abbreviated for manual inclusion)

Example

```
>> DATAFORMAT:DETECTION?
{
   "portNumber": 29981,
   "packetStartTag": {
      "bytes": 12,
      "datatype":"char",
      "arraysize": 12,
      "description": "<detections>"
   },
   "header": {
      "detection_packet.packetSizeBytes": {
         "bytes": 4,
         "datatype": "uint32",
         "arraysize": 1,
         "description": "total number of bytes in this data packet"
      },

… additional fields omitted for brevity...

      "detection_packet.reserved_header": {
         "bytes": 4,
         "datatype": "N/A",
         "arraysize": 1,
         "description": "reserved"
      }
   },
   "data": {
      "detection_data.detectionTimeDays": {
         "bytes": 4,
         "datatype": "uint32",
         "arraysize": 1,
         "description": "detection time in days"
      },
      "detection_data.detectionTimeMs": {
         "bytes": 4,
         "datatype": "uint32",
         "arraysize": 1,
         "description": "detection time in milliseconds"
      },

… additional fields omitted for brevity...

      "detection_data.rcs": {
         "bytes": 4,
         "datatype": "float",
         "arraysize": 1,
         "description": "dBsm"
      },
      "detection_data.reserved_data_1": {
         "bytes": 16,
         "datatype": "N/A",
         "arraysize": 1,
         "description": "reserved"
      }
   }
}
OK
```

10.38 DATAFORMAT:RVMAP?

DATAFORMAT: RVMAP?	
(Read) Returns a complete list of all current rvmap packet fields parameters, hierarchy, datatype, size and brief description in a JSON structured format with whitespace for easier automatic software ingest and parsing.	
Write Parameters	n/a
Write Syntax	n/a
Read Syntax	DATAFORMAT: RVMAP?
Read Returns	JSON formatted string as shown below (abbreviated for manual inclusion)

Example

```
>> DATAFORMAT:RVMAP?
{
   "portNumber": 29980,
   "packetStartTag": {
      "bytes": 16,
      "datatype":"char",
      "arraysize": 16,
      "description": "<rangevelocitym>"
   },
   "header": {
      "rvmap_packet.packetSizeBytes": {
         "bytes": 4,
         "datatype": "uint32",
         "arraysize": 1,
         "description": "total number of bytes in this data packet"
      },
      "rvmap_packet.az": {
         "bytes": 4,
         "datatype": "float",
         "arraysize": 1,
         "description": "degrees"
      },
      "rvmap_packet.el": {
         "bytes": 4,
         "datatype": "float",
         "arraysize": 1,
         "description": "degrees"
      },

… additional fields omitted for brevity...

      "rvmap_packet.reserved_header_0": {
         "bytes": 11,
         "datatype": "N/A",
         "arraysize": 1,
         "description": "reserved"
      },
      "rvmap_packet.statusByte": {
         "bytes": 1,
         "datatype": "N/A",
         "arraysize": 1,
         "description": "LSB 1 if ACD is saturated"
      }
   },
   "data": {
      "s": {
         "bytes": 262144,
         "datatype": "uint32",
         "arraysize": 65536,
         "description": "data value"
      }
   }
}
OK
```

10.39 DATAFORMAT:MEASUREMENT?

DATAFORMAT:MEASUREMENT?	
(Read) Returns a complete list of all current measurement packet fields parameters, hierarchy, datatype, size and brief description in a JSON structured format with whitespace for easier automatic software ingest and parsing.	
Write Parameters	n/a
Write Syntax	n/a
Read Syntax	DATAFORMAT:MEASUREMENT?
Read Returns	JSON formatted string as shown below (abbreviated for manual inclusion)

Example	>> DATAFORMAT:MEASUREMENT?
	{
	"portNumber": 29984,
	"packetStartTag": {
	"bytes": 16,
	"datatype":"char",
	"arraysize": 16,
	"description": "<measurements23>"
	},
	"header": {
	"measurement_packet.packetSizeBytes": {
	"bytes": 4,
	"datatype": "uint32",
	"arraysize": 1,
	"description": "total number of bytes in this data packet"
	},
	
	… additional fields omitted for brevity...
	
	"measurement_packet.reserved_header": {
	"bytes": 32,
	"datatype": "N/A",
	"arraysize": 1,
	"description": "reserved"
	}
	},
	"data": {
	"measurement_data.measurement_ID": {
	"bytes": 4,
	"datatype": "uint32",
	"arraysize": 1,
	"description": "measurement ID"
	},
	
	… additional fields omitted for brevity...
	
	"measurement_data.detIds": {
	"bytes": 256,
	"datatype": "uint32",
	"arraysize": 64,
	"description": "array of detections used to create measurement"
	},
	"measurement_data.reserved_data_1": {
	"bytes": 88,
	"datatype": "N/A",
	"arraysize": 1,
	"description": "reserved"
	}
	}
	}
	OK

10.40 DATAFORMAT:STATUS?

<table>
<tr><th colspan="2">DATAFORMAT:STATUS?</th></tr>
<tr><td colspan="2">(Read) Returns a complete list of all current status packet fields parameters, hierarchy, datatype, size and brief description in a JSON structured format with whitespace for easier automatic software ingest and parsing.</td></tr>
<tr><td>Write Parameters</td><td>n/a</td></tr>
<tr><td>Write Syntax</td><td>n/a</td></tr>
<tr><td>Read Syntax</td><td>DATAFORMAT:STATUS?</td></tr>
<tr><td>Read Returns</td><td>JSON formatted string as shown below (abbreviated for manual inclusion)</td></tr>
<tr><td>Example</td><td>>> DATAFORMAT:STATUS?
{
"portNumber": 29979,
"packetStartTag": {
"bytes": 12,
"datatype":"char",
"arraysize": 12,
"description": "<syststatus>"
},
"header": {
"system_status_packet.packetSizeBytes": {
"bytes": 4,
"datatype": "uint32",
"arraysize": 1,
"description": "total number of bytes in this data packet"
},

… additional fields omitted for brevity...

"system_status_packet.serialNumberStr": {
"bytes": 8,
"datatype": "char",
"arraysize": 8,
"description": "serial number string"
},
},
"data": {
"system_status_packet.systemState": {
"bytes": 4,
"datatype": "uint32",
"arraysize": 1,
"description": "0:Reset- 1:Init- 2:Idle- 3:Command Executing- 4:Search- 5:SWT- 6:Error- 7:Upgrade- 8:Restart- 9:Interference Detection"
},
"system_status_packet.searchFrameRate": {
"bytes": 4,
"datatype": "float",
"arraysize": 1,
"description": "FOV/sec"
},

… additional fields omitted for brevity...

"system_status_packet.reserved_data_2": {
"bytes": 252,
"datatype": "N/A",
"arraysize": 1,
"description": "reserved"
}
}
}
OK</td></tr>
</table>

10.41 RSP:CFAR:ENABLE/DISABLE

RSP:CFAR:ENABLE	
(Read/Write) Enables/disables CFAR (Constant Fale Alarm Rate), measurements, and track processing. The default value for RSP:CFAR:ENABLE is set to "true", which enables CFAR processing. When set to "false", CFAR is disabled – resulting in no Detect, Measurements, or Track packets. This is particularly useful when leveraging Range-Velocity (RV) Map data as only RVMaps and Status packets are output. To reset the parameter back to an enabled state, the RESET:PARAMETER (hard reset) is required. You can also reset the parameter using CFAR: RSP:CFAR:ENABLE <pswd> true. NOTE: Disabling CFAR reduces radar processing load, resulting in: • Improved beam-to-beam timing • Faster delivery of RV map packets • Significantly fewer dropped packets Once configured, the setting takes affect with the next Radaring Start command and persists across power cycles. This is reset by HARD RESET.	
Write Parameters	True or False
Write Syntax	RSP:CFAR:ENABLE <su_pswd> <value>
Read Syntax	RSP:CFAR:ENABLE <su_pswd> <value>
Read Returns	1 or 0
Factory Default Value	True
Example	RSP:CFAR:ENABLE? <su_pswd> 1 OK RSP:CFAR:ENABLE <su_pswd> true OK RSP:CFAR:ENABLE <su_pswd> false OK

10.42 RSP:DYNAMIC:CLUTTERMASK:ENABLE 0

RSP:DYNAMIC:CLUTTERMASK [?] <value>	
(Read/Write) Enables or Disables the Dynamic Clutter Mask feature.	
Write Parameters	<flag> - (string), [TRUE,FALSE]. Flag to enable/disable Dynamic Clutter Mask
Write Syntax	RSP:DYNAMIC:CLUTTERMASK:ENABLE [TRUE,FALSE]
Read Syntax	RSP:DYNAMIC:CLUTTERMASK:ENABLE?
Read Returns	The flag state 0 (FALSE) or 1 (TRUE)
Factory Default Value	0
Example	>> RSP:DYNAMIC:CLUTTERMASK:ENABLE? 1 OK

10.43 RSP:DYNAMIC:CLUTTERMASK:LEVEL 1

RSP:DYNAMIC:CLUTTERMASK [?] <value>	
(Read/Write) Sets the Level of Clutter rejection 0(least), 1 or 2(most)	
Write Parameters	<value> - (int) 0 (least), 1 or 2 (most)
Write Syntax	RSP:DYNAMIC:CLUTTERMASK:LEVEL <value>
Read Syntax	RSP:DYNAMIC:CLUTTERMASK:LEVEL?
Read Returns	The level 0,1,2
Factory Default Value	1
Example	RSP:DYNAMIC:CLUTTERMASK:LEVEL 1

10.44 RSP:DYNAMIC:CLUTTERMASK:STARTRANGE 6

RSP:DYNAMIC:CLUTTERMASK:STARTRANGE 6 [?] <value>	
(Read/Write) The start range in meters to apply the cluttermask. Note that Tone masks are used to set the minimum allowed.	
Write Parameters	<value> - (int) [6, 3000]
Write Syntax	RSP:DYNAMIC:CLUTTERMAKS:STARTRANGE <value>
Read Syntax	RSP:DYNAMIC:CLUTTERMASK:STARTRANGE?
Read Returns	The minimum range at which the dynamic clutter mask will be applied.
Factory Default Value	6
Example	RSP:DYNAMIC:CLUTTERMASK:STARTRANGE 6

10.45 RSP:DYNAMIC:CLUTTERMASK:RANGELENGTH 0

RSP:DYNAMIC:CLUTTERMASK:RANGELENGTH 0 [?] <value>	
(Read/Write) Set the leng	
Write Parameters	<value> - (int) 0 (default to maximum) [6, 3000(maximum)]
Write Syntax	RSP:DYNAMIC:CLUTTERMASK:RANGELENGTH <value>
Read Syntax	RSP:DYNAMIC:CLUTTERMASK:RANGELENGTH?
Read Returns	The length of the region where the dynamic clutter mask will be applied. 0 corresponds to the maximum length of 3000m.
Factory Default Value	0 (corresponds to the maximum value of 3000)
Example	RSP:DYNAMIC:CLUTTERMASK:RANGELENGTH 3000

10.46 RSP:DYNAMIC:CLUTTERMASK:MOUNTINGAZANGLEOFFSET

RSP:DYNAMIC:CLUTTERMASK:MOUNTINGAZANGLEOFFSET [?] <value>	
(Read/Write) Right-handed angle in degrees about the vertical axis of the platform between the direction of forward flight of the platform and the Z-axis of the radar, in the horizontal plane. (Only applies to AIRBORNE product mode)	
Write Parameters	<value> - (int) [-180, 180]
Write Syntax	RSP:DYNAMIC:CLUTTERMASK:MOUNTINGAZANGLEOFFSET <value>
Read Syntax	RSP:DYNAMIC:CLUTTERMASK:MOUNTINGAZANGLEOFFSET?
Read Returns	angle in degrees
Factory Default Value	0
Example	RSP:DYNAMIC:CLUTTERMASK:MOUNTINGAZANGLEOFFSET 60

10.47 RSP:DYNAMIC:CLUTTERMASK:THR 3.0665

RSP:DYNAMIC:CLUTTERMASK:THR 3.0665 [?] <value>	
(Read/Write) Threshold value of ridgeness metric below which a ridge is considered present (do not modify, set by LEVEL)	
Write Parameters	<value> (float) (0, 5)
Write Syntax	RSP:DYNAMIC:CLUTTERMASK:THR <value>
Read Syntax	RSP:DYNAMIC:CLUTTERMASK:THR?
Read Returns	threshold value
Factory Default Value	N/A
Example	N/A

10.48 RSP:DYNAMIC:CLUTTERMASK:THRWINDOW 11

RSP:DYNAMIC:CLUTTERMASK:THRWINDOW 11 [?] <value>	
(Read/Write) Size of kernel used to smooth ridgeness metric (do not modify, set by LEVEL)	
Write Parameters	<value> (int) [1, 30]
Write Syntax	RSP:DYNAMIC:CLUTTERMASK:THRWINDOW <value>
Read Syntax	RSP:DYNAMIC:CLUTTERMASK:THRWINDOW?
Read Returns	size of ridgeness metric smoothing kernel
Factory Default Value	N/A
Example	N/A

10.49 RSP:DYNAMIC:CLUTTERMASK:RIDGEWIDTH 6

RSP:DYNAMIC:CLUTTERMASK:RIDGEWIDTH 6 [?] <value>	
(Read/Write) size of kernel used to smooth ridge location (do not modify, set by LEVEL)	
Write Parameters	<value> (int) [1, 30]
Write Syntax	RSP:DYNAMIC:CLUTTERMASK:RIDGEWIDTH <value>
Read Syntax	RSP:DYNAMIC:CLUTTERMASK:RIDGEWIDTH?
Read Returns	size of ridge location smoothing kernel
Factory Default Value	N/A
Example	N/A

10.50 RSP:DYNAMIC:CLUTTERMASK:RANGEWIDTH 2

RSP:DYNAMIC:CLUTTERMASK:RIDGEWIDTH 6 [?] <value>	
(Read/Write) size of 2D mask kernel in the range dimension (do not modify, set by LEVEL)	
Write Parameters	<value> (int) [0, 5]
Write Syntax	RSP:DYNAMIC:CLUTTERMASK:RANGEWIDTH <value>
Read Syntax	RSP:DYNAMIC:CLUTTERMASK:RANGEWIDTH?
Read Returns	vertical size of ridge mask brush
Factory Default Value	N/A
Example	N/A

10.51 RSP:DYNAMIC:CLUTTERMASK:VELOCITYWIDTH 1

RSP:DYNAMIC:CLUTTERMASK:RIDGEWIDTH 6 [?] <value>	
(Read/Write) size of 2D mask kernel in the velocity dimension (do not modify, set by LEVEL)	
Write Parameters	<value> (int) [0, 5]
Write Syntax	RSP:DYNAMIC:CLUTTERMASK:VELOCITYWIDTH <value>
Read Syntax	RSP:DYNAMIC:CLUTTERMASK:VELOCITYWIDTH?
Read Returns	horizontal size of ridge mask brush
Factory Default Value	N/A
Example	N/A

10.52 TIME:CHANNEL:PWRTHRESHOLD

TIME:CHANNEL:PWRTHRESHOLD [?] <su_pswd> <value>

(Read/Write)) Sets a minimum target power below which detections will be ignored for the purposes of TCM acquisition. The default value of 90dB improves TCM acquisition reliability for co-located radars and should be sufficient for most cases. This thresholding can be disabled (by setting to 0dB) if there are reliability issues trying to achieve TCM lock between distantly located radars.	
Write Parameters	**<value> - float, [0, 200]**
Write Syntax	TIME:CHANNEL:PWRTHRESHOLD <su_pswd> <value>
Read Syntax	TIME:CHANNEL:PWRTHRESHOLD? <su_pswd>
Read Returns	TCM Minimum power threshold in dB
Factory Default Value	90 (dB)
Example	>> TIME:CHANNEL:PWRTHRESHOLD <su_pswd> 90 OK >> TIME:CHANNEL:PWRTHRESHOLD? <su_pswd> 90

11. SUPER-USER COMMANDS

These more advanced commands can be accessed using the super-user password “eldorado”.

11.1 SYSPARAM?

SYSPARAM? <su_pswd>	
(Read) - When the super-user password is provided with this command it returns a list of all settable/gettable system parameters accessible by User <u>and</u> Super-User commands, and their current state. Dynamically changing parameters such as PCB temperature are not reported. Derivative parameters such as range resolution are also reported.	
Write Parameters	Not Applicable.
Write Syntax	Not Applicable.
Read Syntax	SYSPARAM? <su_pswd>
Read Returns	>> SYSPARAM? <su_pswd> GETSERIAL 000699 Configurable parameters: ETH:IP 169.254.1.10 MAC 34.4C.C8.00.02.BB NETMASK 255.255.0.0 GATEWAY 0.0.0.0 DMS:CHANNEL 1 MODE:SWT:OPERATIONMODE 0 (0 = Pedestrian, 1 = UAS, 2 = Plane) MODE:SEARCH:AZFOVMIN -60.000000 (deg) MODE:SEARCH:AZFOVMAX 60.000000 (deg) MODE:SEARCH:AZSTEP 2.000000 (deg) MODE:SEARCH:ELFOVMIN -40.000000 (deg) MODE:SEARCH:ELFOVMAX 40.000000 (deg) MODE:SEARCH:ELSTEP 2.000000 (deg) MODE:SWT:SEARCH:AZFOVMIN -60.000000 (deg) MODE:SWT:SEARCH:AZFOVMAX 60.000000 (deg) MODE:SWT:SEARCH:ELFOVMIN 0.000000 (deg) MODE:SWT:SEARCH:ELFOVMAX 12.000000 (deg) MODE:SWT:TRACK:AZFOVMIN -60.000000 (deg) MODE:SWT:TRACK:AZFOVMAX 60.000000 (deg) MODE:SWT:TRACK:ELFOVMIN -40.000000 (deg) MODE:SWT:TRACK:ELFOVMAX 40.000000 (deg) RSP:RCSMASK:MINRCS -30.000000 (dB) RSP:RCSMASK:MAXRCS 100.000000 (dB) Configurable super-user parameters: RSP:CLUTTERMASKWIDTH 3 (units) MODE:SWT:TRACKER:TRACKNUM 20 (units) MODE:SWT:TRACKER:MINREPORTINGCONFIDENCE 20.000000 (units) MODE:SWT:TRACKER:MAXCONFIDENCETIME 2.000000 (time in seconds) MODE:SWT:TRACKER:MAXCOASTTIME 8.000000 (time in seconds) MODE:SWT:TRACKER:MAXOOBCOASTTIME 1.000000 (time in seconds) MODE:SWT:TRACKER:MAXTARGETVELOCITY 10.000000 (units) MODE:SWT:TRACKER:RCSESTSAMPLESIZE 25 AGLMASK:MINAGL -10000.000000 (m) AGLMASK:MAXAGL 10000.000000 (m) AGLMASK:ENABLE FALSE RANGE:MASK 0 -1, -1, -1, -1 RANGE:MASK 1 -1, -1, -1, -1 RANGE:MASK 2 -1, -1, -1, -1 RANGE:MASK 3 -1, -1, -1, -1 RANGE:MASK 4 -1, -1, -1, -1 RANGE:MASK 5 132, 134, 0, 31 TIME:CHANNEL:ENABLE FALSE TIME:CHANNEL:NUM 0 TIME:CHANNEL:RANGEMAXINTEREST 5000 (m) TIME:CHANNEL:CONTROLRANGE 5500 (m) CLF:ENABLE FALSE OUTPUT:THROTTLETRACKRATE:ENABLE FALSE OUTPUT:MAXTRACKRATE 10 (Hz) OUTPUT:STATUSRATE 20 (Hz) PLATFORM:STATE:ORIENTATION[HEADING] 0.000000 (degrees) PLATFORM:STATE:ORIENTATION[PITCH] 0.000000 (degrees) PLATFORM:STATE:ORIENTATION[ROLL] 0.000000 (degrees) PLATFORM:STATE:ELEVATIONAGL 0.000000 (m) Zone Mask: Enable: 0 MinAz MaxAz MinEl MaxEl MinRng MaxRng MinVel MaxVel MinRCS MaxRCS UsedBit HardMask Mask 0: 0 0 0 0 0 0 0.000000 0.000000 0 0 0 0 Mask 1: 0 0 0 0 0 0 0.000000 0.000000 0 0 0 0 ***… masks 3 – 30 skipped for brevity*** Mask 31: 0 0 0 0 0 0 0.000000 0.000000 0 0 0 0 Derivative parameters:

	dR 3.253038 (meter) nRanges 2048 (units) dV 0.908293 (units) nVelocities 32 (units) zero range bin number, n0 128 (units) zero velocity bin number, m0 16 (units) OK

11.2 RSP:CLUTTERMASKWIDTH

RSP:CLUTTERMASKWIDTH [?] <su_pswd> <value>	
(Read/Write) Updates the velocity mask width, expressed in number of Doppler bins, used to mask out low velocity clutter detections during detection processing. Masked detections will not be used for tracking. They also will not be sent to the user in detection packets. This parameter can be changed while radar is actively running.	
Write Parameters	<value>: integer, [0,32]
Write Syntax	RSP:CLUTTERMASKWIDTH <su_pswd> <value>
Read Syntax	RSP:CLUTTERMASKWIDTH? <su_pswd>
Read Returns	The current clutter velocity mask width used for detection processing.
Factory Default Value	3
Example	>> RSP:CLUTTERMASKWIDTH <su_pswd> 1 OK >> RSP:CLUTTERMASKWIDTH? <su_pswd> 1 OK

11.3 RANGE:MASK

RANGE:MASK[?] <su_pswd> <tone_num>,<r_min>,< r_max>,< v_min>,< v_max>	
(Read/Write) Defines a mask over all angles (Az & El) to mask out any detections within the specified range and velocity bounds. <u>This is most useful for setting a minimum and maximum range of interest.</u> Up to six (6) masks may be defined. A mask with all bounds equal to '-1' will not be applied. A mask may have no extent in range and/or velocity, indicating that only a single range and/or velocity should be masked. Masked detections will not be sent to the tracker. Refer to equations (1) and (2) in Section 4.2 to calculate r_min, r_max, v_min and v_max bin numbers. <u>As of SW16.4.0, this parameter can be changed while operating.</u> **NOTES:** • Mask #5 defaults to masking 10-20m near 'blind' range and is recommended to leave in place in all typical use cases. • EchoGuard-CR is limited to 1200m instrumented range in firmware. Use this command to reduce further if desired. CR minimum range is 10m so mask 5 is empty by default.	
Write Parameters	<mask_num> (integer), [0,4]. Mask number. <r_min> (uint), [0, wr_max], <= r_max. Minimum range bin number of tone mask. <r_max> (uint), [0, wr_max], >= r_min. Maximum range bin number of tone mask. <v_min> (uint), [0, wv_max], <= v_max. Minimum Doppler bin number of tone mask. <v_max> (uint), [0, wv_max], >= v_min. Maximum Doppler bin number of tone mask. For wv_max and wr_max, see section 6.2.
Write Syntax	RANGE:MASK <su_pswd> <tone_num>, <r_min>, < r_max>, < v_min>, < v_max>
Read Syntax	RANGE:MASK? <su_pswd>
Read Returns	The current masks being applied.
Factory Default Value	0: -1,-1,-1,-1 1: -1,-1,-1,-1 2: -1,-1,-1,-1 3: -1,-1,-1,-1 4: -1,-1,-1,-1 5: 131,134,0,31 (10-20m default near blind range mask for EchoGuard)
Example The reply assumes no other masks have been specified.	Example: You want to create a mask, to mask out any target detections at greater than 1200m, and with opening velocities greater than 6.5m/sec. Calculate Range Bin# at 1200m = 497 (bin#= Int[1200/3.25m] + 128), Max Bin=2047 Calculate Velocity Bin# >6.5m/s = 23 (bin# = Int(6.5/0.909m/s)+16, Max Vel Bin = 31

	>> RANGE:MASK <su_pswd> 0,497,2047,23,31 OK >> RANGE:MASK? <su_pswd> 0: 0,497,2047,23,31 1: -1,-1,-1,-1 2: -1,-1,-1,-1 3: -1,-1,-1,-1 4: -1,-1,-1,-1 5: 131,134,0,31 OK >> RANGE:MASK <su_pswd> 0,-1,-1,-1,-1 OK

11.4 MODE:SWT:TRACKER:TRACKNUM

MODE:SWT:TRACKER:TRACKNUM[?] <su_pswd> <value>	
(Read/Write) Updates the maximum number of tracked objects that can be active at the same time and reported via the track packets. Default value is optimal for operation mode defaults.	
Write Parameters	<value>: integer, [1,30]
Write Syntax	MODE:SWT:TRACKER:TRACKNUM <su_pswd> <value>
Read Syntax	MODE:SWT:TRACKER:TRACKNUM? <su_pswd>
Read Returns	<value>
Factory Default Value	20
Examples	>> MODE:SWT:TRACKER:TRACKNUM <su_pswd> 10 OK >> MODE:SWT:TRACKER:TRACKNUM? <su_pswd> 10 OK

11.5 MODE:SWT:TRACKER:MINREPORTINGCONFIDENCE

MODE:SWT:TRACKER:MINREPORTINGCONFIDENCE[?] <su_pswd> <value>	
(Read/Write) Updates the minimum confidence level a track must possess to be reported by the system. Increasing this value will help reduce the number of short lived or spurious tracks reported by the system but will also increase the time it takes for tracks on targets of interest to be reported to the user. Note, tracks not reported to user still consume tracker resources as the track is maintained internally as long as confidence is > 0. **Explanation:** All tracks start with a confidence of 0. As measurements are associated to a track the confidence increases by (track_update_dt / MAX_CONF_TIME) * 100 * (a scaling factor based on the measurement's Mahalanobis distance from the estimated track location). If during a track update cycle there were beams scheduled for the track and there were no associated measurements, the confidence decays by 100 * (track_update_dt / MAX_COAST_TIME). Finally, if no beams were schedule for a track and there were no associated measurements, the confidence decays by (100 * (track_update_dt / MAX_COAST_TIME)) / 10 .	
Write Parameters	<value>: float, [0.0, 100.0]
Write Syntax	MODE:SWT:TRACKER:MINREPORTINGCONFIDENCE <su_pswd> <value>
Read Syntax	MODE:SWT:TRACKER: MINREPORTINGCONFIDENCE? <su_pswd>
Read Returns	<value>
Factory Default Value	20.0
Examples	>> MODE:SWT:TRACKER:MINREPORTINGCONFIDENCE <su_pswd> 25 OK >> MODE:SWT:TRACKER: MINREPORTINGCONFIDENCE? <su_pswd> 25.000000 OK

11.6 MODE:SWT:TRACKER:MAXCONFIDENCETIME

MODE:SWT:TRACKER: MAXCONFIDENCETIME[?] <su_pswd> <value>	
(Read/Write) Updates the amount of time in seconds that a track must be observed, with high probability measurements associated to it, to attain full confidence level. This should typically be several times smaller than the MAXCOASTTIME.	
Write Parameters	<value>: float, [>=0.0]
Write Syntax	MODE:SWT:TRACKER: MAXCONFIDENCETIME <su_pswd> <value>
Read Syntax	MODE:SWT:TRACKER: MAXCONFIDENCETIME? <su_pswd>
Read Returns	<value>
Factory Default Value	1.0
Examples	>> MODE:SWT:TRACKER: MAXCONFIDENCETIME <su_pswd> 0.5 OK >> MODE:SWT:TRACKER: MAXCONFIDENCETIME? <su_pswd> 0.500000 OK

11.7 MODE:SWT:TRACKER:MAXCOASTTIME

MODE:SWT:TRACKER:MAXCOASTTIME[?] <su_pswd> <value>	
(Read/Write) This parameter controls the minimum amount of time in which a track will go from a confidence level of 100% to a confidence level of 0% when no measurements are associated to the track. It is recommended to keep this value between 2 and 10 seconds. In addition, it is recommended to always keep this value greater than MODE:SWT:TRACKER:MAXCONFIDENCETIME.	
Write Parameters	<value>: float, [>=0.0]
Write Syntax	MODE:SWT:TRACKER:MAXCOASTTIME <su_pswd> <value>
Read Syntax	MODE:SWT:TRACKER:MAXCOASTTIME? <su_pswd>
Read Returns	<value>
Factory Default Value	5 (Operation Mode = 0, walker)
Examples	>> MODE:SWT:TRACKER:MAXCOASTTIME <su_pswd> 5.2 OK >> MODE:SWT:TRACKER:MAXCOASTTIME? <su_pswd> 5.200000 OK

11.8 MODE:SWT:TRACKER:MAXOOBCOASTTIME

MODE:SWT:TRACKER:MAXOOBCOASTTIME[?] <su_pswd> <value>	
(Read/Write) This parameter controls the maximum amount of time a track can be outside the track Field of View (FoV) before the track confidence decays to 0. Note that the track and search fields of view can be different and are configured independently.	
Write Parameters	<value>: float, [>=0]
Write Syntax	MODE:SWT:TRACKER:MAXOOBCOASTTIME <su_pswd> <value>
Read Syntax	MODE:SWT:TRACKER:MAXOOBCOASTTIME? <su_pswd>
Read Returns	<value>
Factory Default Value	1
Examples	>> MODE:SWT:TRACKER:MAXOOBCOASTTIME <su_pswd> 5.2 OK >> MODE:SWT:TRACKER:MAXOOBCOASTTIME? <su_pswd> 5.200000 OK

11.9 MODE:SWT:TRACKER:MAXTARGETVELOCITY

MODE:SWT:TRACKER:MAXTARGETVELOCITY [?] <su_pswd> <value>	
(Read/Write) Sets the maximum velocity in meters/second (m/s) of objects to evaluate for interbeam detection clustering and track association for targets of interest.	
Write Parameters	<value>: float, [>=0.0] meters/second
Write Syntax	MODE:SWT:TRACKER:MAXTARGETVELOCITY <su_pswd> <value>
Read Syntax	MODE:SWT:TRACKER:MAXTARGETVELOCITY?<su_pswd>
Read Returns	<value>
Factory Default Value	10.0 (Operation Mode = 0)
Examples	>> MODE:SWT:TRACKER:MAXTARGETVELOCITY <su_pswd> 10 OK >> MODE:SWT:TRACKER:MAXTARGETVELOCITY? <su_pswd> 10.000000 OK

11.10 MODE:SWT:TRACKER:RCSESTSAMPLESIZE

MODE:SWT:TRACKER: RCSESTSAMPLESIZE [?] <su_pswd> <value>	
(Read/Write) Sets the maximum number of past sequential lobing measurements to use for estimating a particular track's Radar Cross Section (RCS) in dBsm. Default value is considered optimal for the majority of use cases. Increase to lowpass filter estimate.	
Write Parameters	<value>: integer, [1,1000]
Write Syntax	MODE:SWT:TRACKER:RCSESTSAMPLESIZE <su_pswd> <value>
Read Syntax	MODE:SWT:TRACKER:RCSESTSAMPLESIZE? <su_pswd>
Read Returns	<value>
Factory Default Value	25 (Operation Mode = 0)
Examples	>> MODE:SWT:TRACKER: RCSESTSAMPLESIZE <su_pswd> 100 OK >> MODE:SWT:TRACKER: RCSESTSAMPLESIZE? <su_pswd> 100 OK

11.11 MODE:SWT:AZSTEP

MODE:SWT:AZSTEP[?] <su_pswd> <value>	
(Read/Write) Update the SWT mode search azimuth step value. Default value is considered optimal for majority of use cases.	
Write Parameters	<value>: integer [2 to 120, multiples of 2 only] in degrees
Write Syntax	MODE:SEARCH:AZSTEP <su_pswd> <value>
Read Syntax	MODE:SEARCH:AZSTEP? <su_pswd>
Read Returns	The current azimuth beam step size used for searching in SWT mode.
Factory Default Value	6
Example	>> MODE:SWT:AZSTEP <su_pswd> 10 OK >> MODE:SWT:AZSTEP? <su_pswd> 10 OK

11.12 MODE:SWT:ELSTEP

MODE:SWT: ELSTEP[?] <su_pswd> <value>	
(Read/Write) Update the SWT mode search elevation step value. Default value is considered optimal for majority of use cases.	
Write Parameters	<value>: integer [2 to 80, multiples of 2 only] in degrees

Write Syntax	MODE:SWT: ELSTEP <su_pswd> stepSize
Read Syntax	MODE:SWT: ELSTEP? <su_pswd>
Read Returns	The current elevation beam step size used for searching in SWT mode.
Factory Default Value	12
Example	>> MODE:SWT:ELSTEP <su_pswd> 10 OK >> MODE:SWT:ELSTEP? <su_pswd> 10 OK

11.13 CLF:ENABLE

CLF:ENABLE [?] <su_pswd> <flag>	
(Read/Write) Enables/disables the onboard UAV classifier. Enabled by default in all Operation Modes except Pedestrian (where default parameters are not optimized for tracking UAS). This command enables/disables the internal classifier that returns the probability of acquired tracks belonging to a specific multi-copter target class. The classifier leverages machine learning techniques to extract multi-copter propeller detection signatures combined with target size (RCS) to determine a probability of UAV. When disabled, the classifier fields in the track packet will report as NaN.	
Write Parameters	<flag>: (string), [TRUE,FALSE] Flag to enable/disable the on-board classifier.
Write Syntax	CLF:ENABLE <su_pswd> <flag>
Read Syntax	CLF:ENABLE? <su_pswd>
Read Returns	1 if classifier is enabled, 0 if not enabled
Factory Default Value	TRUE in all Operation Modes except Pedestrian.
Examples Enable classifier	>> CLF:ENABLE <su_pswd> TRUE OK >> CLF:ENABLE? <su_pswd> 1 OK

11.14 PRISM:CHANNEL:ENABLE

PRISM:CHANNEL:ENABLE[?] <su_pswd> <flag>	
(Read/Write) Enable or disable the PRISM channelization feature which allows multiple radars to coexist on the same frequency channel to minimize interference and allow greater density installations. User should also set the desired number of PRISM virtual channels per frequency band and the specific PRISM channel # for each respective radar. Observe the required spacing guidelines as PRISM channelization cannot mitigate interference from very bright co-located radars.	
Write Parameters	<flag>: string, [TRUE, FALSE] Flag to enable/disable PRISM channel feature
Write Syntax	PRISM:CHANNEL:ENABLE <su_pswd> <value>
Read Syntax	PRISM:CHANNEL:ENABLE? <su_pswd>
Read Returns	Enable PRISM channelization feature
Factory Default Value	0 (FALSE)
Example	>> PRISM:CHANNEL:ENABLE? <su_pswd> 0 OK >> PRISM:CHANNEL:ENABLE <su_pswd> TRUE PRISM Enable:1 OK

11.15 PRISM:CHANNEL:MAX

PRISM:CHANNEL:MAX [?] <su_pswd> <value>
(Read/Write) Sets the maximum number of PRISM channels available for radars to use in a given frequency band. Lowering the max number allows for a greater Pulse Repetition Interval (PRI) value which improves the interference

mitigation between radars by allowing the algorithm to isolate the interfering radar signals more easily. Use the lowest value required for physical layout of radar network.	
Write Parameters	<value>: integer, [2:4]
Write Syntax	PRISM:CHANNEL: MAX <su_pswd> <value>
Read Syntax	PRISM:CHANNEL: MAX? <su_pswd>
Read Returns	PRISM channel assigned for the radar
Factory Default Value	4
Example Changing from default setting of 4 to 2	>> PRISM:CHANNEL: MAX? <su_pswd> 4 OK >> PRISM:CHANNEL: MAX <su_pswd> 2 OK

11.16 PRISM:CHANNEL:NUM

PRISM:CHANNEL:NUM [?] <su_pswd> <value>	
(Read/Write) Sets the PRISM channel of the radar for interference mitigation [0-(MAX-1)]. Indexing starts at 0 (default) and progresses up to 1 less than maximum.	
Write Parameters	<value>: integer, [0:(PRISM:CHANNEL:MAX-1)]
Write Syntax	PRISM:CHANNEL:NUM <su_pswd> <value>
Read Syntax	PRISM:CHANNEL:NUM? <su_pswd>
Read Returns	Time channel assigned for the radar
Factory Default Value	0
Example	>> PRISM:CHANNEL:NUM <su_pswd> 0 OK >> PRISM:CHANNEL:NUM? <su_pswd> 1 OK

11.17 TIME:CHANNEL:ENABLE

TIME:CHANNEL:ENABLE[?] <su_pswd> <flag>	
(Read/Write) Enable or disable the Time Channel Mitigation (TCM) feature which allows multiple radars to time align their FMCW chirps on the same frequency channel to minimize interference and allow greater density installations.	
Write Parameters	<flag>: string, [TRUE, FALSE] Flag to enable/disable Time Channel Mitigation feature
Write Syntax	TIME:CHANNEL:ENABLE <su_pswd> <value>
Read Syntax	TIME:CHANNEL:ENABLE? <su_pswd>
Read Returns	Enable status of interference mitigation mode
Factory Default Value	FALSE
Example	>> TIME:CHANNEL:ENABLE <su_pswd> TRUE OK >> TIME:CHANNEL:ENABLE? <su_pswd> TRUE OK

11.18 TIME:CHANNEL:NUM

TIME:CHANNEL:NUM [?] <su_pswd> <value>	
(Read/Write) Sets the time channel of the radar for TCM interference mitigation [0-3]. The value [0] sets the radar to the primary position (leader) which doesn't observe a delay and begins immediately searching for interference. A value [1-3] instructs the radar to wait the <value> x <delay period> to allow the primary radar to establish operation before enabling the following radar(s) to negotiate.	
Write Parameters	<value>: integer, [0:3]

Write Syntax	TIME:CHANNEL:NUM <su_pswd> <value>
Read Syntax	TIME:CHANNEL:NUM? <su_pswd>
Read Returns	Time channel assigned for the radar
Factory Default Value	0
Example	>> TIME:CHANNEL:NUM <su_pswd> 1 OK >> TIME:CHANNEL:NUM? <su_pswd> 1 OK

11.19 TIME:CHANNEL:DELAY

TIME:CHANNEL:DELAY [?] <su_pswd> <value>	
(Read/Write) Sets the delay (in milliseconds) for each radar time channel to sequentially discover and align the interfering radar (if detected) to the RANGEMAXINTEREST bin location. Total delay is <TIME:CHANNE:NUM> * <TIME:CHANNEL:DELAY>. The 'zeroth' or 'master' radar observes no delay.	
Write Parameters	<value>: integer, [1000, 10000], msec
Write Syntax	TIME:CHANNEL:DELAY <su_pswd> <value>
Read Syntax	TIME:CHANNEL:DELAY? <su_pswd>
Read Returns	The delay for each time channel to start searching in milliseconds
Factory Default Value	5000 (5 seconds)
Example	>> TIME:CHANNEL:DELAY <su_pswd> 3000 OK >> TIME:CHANNEL:DELAY? <su_pswd> 3000 OK

11.20 TIME:CHANNEL:COAST

TIME:CHANNEL:COAST [?] <su_pswd> <value>	
(Read/Write) Sets the amount of time in milliseconds that the time channel monitoring system is allowed to go without detecting the interferer it is following. After this amount of time, the time channel monitoring system searches for its interferer.	
Write Parameters	<value>: integer, [10000, 60000]
Write Syntax	TIME:CHANNEL:COAST <su_pswd> <value>
Read Syntax	TIME:CHANNEL:COAST? <su_pswd>
Read Returns	The amount of time in milliseconds TCM is allowed to run and update without seeing its interferer
Factory Default Value	30000 (30 seconds)
Example	>> TIME:CHANNEL:COAST <su_pswd> 60000 OK >> TIME:CHANNEL:COAST? <su_pswd> 60000 OK

11.21 TIME:CHANNEL:RANGEMAXINTEREST

TIME:CHANNEL:RANGEMAXINTEREST [?] <su_pswd> <value>	
(Read/Write) Sets the maximum range of interest (in meters) for interference mitigation function to locate and control the interfering radar tone. Less than the default range value may be desirable in scenarios with multipath reflections or multiple (more than 2) radars sharing the same frequency channel.	
Write Parameters	<value>: float, [100.0, 5985.0]
Write Syntax	TIME:CHANNEL:RANGEMAXINTEREST <su_pswd> <value>
Read Syntax	TIME:CHANNEL:RANGEMAXINTEREST? <su_pswd>

Read Returns	Range of max interest for interference mitigation
Factory Default Value	5000 (meters)
Example	>> TIME:CHANNEL:RANGEMAXINTEREST <su_pswd> 3000 OK >> TIME:CHANNEL:RANGEMAXINTEREST? <su_pswd> 3000 OK

11.22 TIME:CHANNEL: CONTROLRANGE

TIME:CHANNEL: CONTROLRANGE [?] <su_pswd> <value>	
(Read/Write) Sets the range at which the main tone of the interfering radar being tracked via TCM will be placed. For just two radars on the same frequency channel, the default value should suffice. For 3 or more radars sharing the same frequency channel requiring multi-radar TCM, the CONTROLRANGE = 5985 – N * 300 meters where N is the TCM channel number of the radar.	
Write Parameters	<value> - float, [RANGEMAXINTEREST + 300, 5700]
Write Syntax	TIME:CHANNEL: CONTROLRANGE <su_pswd> <value>
Read Syntax	TIME:CHANNEL: CONTROLRANGE? <su_pswd>
Read Returns	Range of max interest for interference mitigation
Factory Default Value	5500 (meters)
Example	>> TIME:CHANNEL: CONTROLRANGE <su_pswd> 3000 OK >> TIME:CHANNEL: CONTROLRANGE? <su_pswd> 3000 OK

11.23 ZONE:MASK:ENABLE

ZONE:MASK:ENABLE[?] <su_pswd> <flag>	
(Read/Write) Enables or disables the zone masking function. This doesn't clear the zone mask fields, but simply enables the defined zone masks to filter measurements from the tracker.	
Write Parameters	<flag>: string, [TRUE,FALSE] Flag to enable/disable the zone masking
Write Syntax	ZONE:MASK:ENABLE <su_pswd> <flag>
Read Syntax	ZONE:MASK:ENABLE? <su_pswd>
Read Returns	0 if disabled, 1 if enabled
Examples	>>ZONE:MASK:ENABLE <su_pswd> TRUE OK >> ZONE:MASK:ENABLE? <su_pswd> ZoneMask Enabled: 1 OK

11.24 ZONE:MASK:CLEAR

ZONE:MASK:CLEAR <su_pswd>	
(Write) Resets all zone masks back to defaults (all zone mask parameters set to 0) which effectively disables any zone masking. Also sets ZONE:MASK:ENABLE to False.	
Write Parameters	<su_pswd>
Write Syntax	ZONE:MASK:CLEAR <su_pswd>
Example	>> ZONE:MASK:CLEAR <su_pswd> OK

11.25 ZONE:MASK

ZONE:MASK[?] <su_pswd> <Mask #> <Min Az> <Max Az> <Min El> <Max El> <Min Range> <Max Range> <Min Vel> <Max Vel> <Min RCS> <Max RCS> <Hard-Soft Flag>	
(Read/Write) Sets the selected zone mask slot [0, 31] to the parameter ranges specified. If the selected zone mask is already defined, then it is overwritten. The zone mask defines an enclosed region of 3-D polar space (R, Az, El) in which measurements (clustered and interpolated detections) will be masked or blocked from generating new tracks. The zone mask slot may be specified simply in terms of minimum to maximum dimension spans in range [meters], azimuth angle [°] and elevation [°] in which case all RCS and velocity values will be blocked in the defined zone. Optionally, the Zone Mask may be specified with additional parameters specifying the min/max velocity range (radial Doppler velocity component [meters/second]), RCS (radar cross section [dBsm]) and hard/soft flag which must be specified if used. **Notes:** • Zone masks act last after RCS, Clutter and Range Mask filters • Zone masks may overlap in 3-D space w/o issue. The first one to match will reject the measurement. • 'Soft' zone masks do not affect existing tracks propagating thru a mask, only prevent starting new tracks. • 'Hard' zone masks kill all tracks entering a mask by starving it from associated measurements. • The 'UsedBit' is only used upon Read queries and indicates which Mask #s have been defined. Useful for UI to quickly parse actively defined zone masks. Not part of Write commands.	
Write Parameters	**<su_pswd>** Super user password **<Mask #>** integer, [0, 31] Index # that defines zone mask being configured. Up to 32 unique and overlapping zone masks may be specified. **<Min Az>** integer, [-60, 60] Minimum azimuth angle in degrees for the zone mask; must be less than or equal to <Max Az>. Negative azimuth angles are defined to the right from the radar point of view. **<Max Az>** integer, [-60, 60] Maximum azimuth angle in degrees for the zone mask; must be greater than or equal to <Min Az>. This defines the left size of the mask from the radar point of view. **<Min El>** integer, [-40, 40] Minimum elevation angle in degrees for the zone mask; must be less than or equal to <Max El>. Negative elevations are downward from radar point of view. **<Max El>** integer, [-40, 40] Maximum elevation angle in degrees for the zone mask; must be greater than or equal to <Min El>. Positive elevations are upward from radar point of view. **<Min Range>** integer, [20, 5987] Minimum range in meters for the zone mask; must be less than or equal to <Max Range>. **<Max Range>** integer, [20, 5987] Maximum range in meters for the zone mask; must be greater than or equal to <Min Range>. ***NOTE:*** *Below values are optional, but all must be specified together if any are used* **<Min Vel>** (optional) float, [-15.0,15.0, A1a waveform] Minimum velocity value in meters per second (m/s) for the zone mask; must be less than or equal to <Max Vel>. Defaults to -15.0 when not specified. Note: Velocity masked is the Doppler radial velocity component. Fast moving targets may have aliased velocities and may be masked inadvertently. **<Max Vel>** (optional) float, [-15.0,15.0, A1a waveform] Maximum velocity value in meters per second (m/s) for the zone mask; must be greater than or equal to <Min Vel>. Defaults to +15.0 when not specified. Same issue with aliasing as above. **<Min RCS>** (optional) integer, [-50,100] Minimum RCS value in dBsm for the zone mask; must be less than or equal to <Max RCS>. Defaults to -30 when not specified. **<Max RCS>** (optional) integer, [-50,100] Maximum RCS value in dBsm for the zone mask; must be greater than or equal to <Max RCS>. Defaults to 100 when not specified. **<UsedBit>** (Read Only) 0/1 reports if mask has been actively set during read back. Skip for write events.

	<Hard_Soft Flag> (optional) string [TRUE, FALSE] Flag to enable 'hard' mask (TRUE/1) or 'soft' mask (FALSE/0); defaults to 0 (soft) when not specified.
Write Syntax	ZONE:MASK <su_pswd> <Mask #> <Min Az> <Max Az> <Min El> <Max El> <Min Range> <Max Range> *<Min Vel> <Max Vel> <Min RCS> <Max RCS> <Hard-Soft Flag>*
Read Syntax	ZONE:MASK? <su_pswd>
Examples Set slot 0 with basic zone mask Set slots 1 & 2 with overloaded zone mask defining all parameters. 1 is a 'soft' mask and 2 is a 'hard' mask Read back Zone Mask settings. Note 'UsedBit' is shown in read back but not set explicitly on write.	>> ZONE:MASK:CLEAR <su_pswd> OK MESA-001071 >> ZONE:MASK? <su_pswd> Zone Mask: Enable: 0 MinAz MaxAz MinEl MaxEl MinRng MaxRng MinVel MaxVel MinRCS MaxRCS UsedBit HardMask Mask 0: 0 0 0 0 0 0 0.000000 0.000000 0 0 0 0 Mask 1: 0 0 0 0 0 0 0.000000 0.000000 0 0 0 0 Mask 2: 0 0 0 0 0 0 0.000000 0.000000 0 0 0 0 Mask 3: 0 0 0 0 0 0 0.000000 0.000000 0 0 0 0 Mask 4: 0 0 0 0 0 0 0.000000 0.000000 0 0 0 0 Mask 5: 0 0 0 0 0 0 0.000000 0.000000 0 0 0 0 Mask 6: 0 0 0 0 0 0 0.000000 0.000000 0 0 0 0 Mask 7: 0 0 0 0 0 0 0.000000 0.000000 0 0 0 0 *(… Mask8-31 not shown for brevity)* OK >> ZONE:MASK <su_pswd> 0 -20 20 -10 10 100 200 OK >> ZONE:MASK <su_pswd>1 -20 20 -10 10 100 200 -15 15 -30 100 OK >> ZONE:MASK <su_pswd>2 -45 45 -20 20 200 400 -15 15 -20 10 TRUE OK >> ZONE:MASK? <su_pswd> Zone Mask: Enable: 0 MinAz MaxAz MinEl MaxEl MinRng MaxRng MinVel MaxVel MinRCS MaxRCS UsedBit HardMask Mask 0: -20 20 -10 10 100 200 -15.000000 15.000000 -30 100 1 0 Mask 1: -20 20 -10 10 100 200 -15.000000 15.000000 -30 100 1 0 Mask 2: -45 45 -20 20 200 400 -15.000000 15.000000 -20 10 1 1 Mask 3: 0 0 0 0 0 0 0.000000 0.000000 0 0 0 0 Mask 4: 0 0 0 0 0 0 0.000000 0.000000 0 0 0 0 Mask 5: 0 0 0 0 0 0 0.000000 0.000000 0 0 0 0 Mask 6: 0 0 0 0 0 0 0.000000 0.000000 0 0 0 0 Mask 7: 0 0 0 0 0 0 0.000000 0.000000 0 0 0 0 *(… Mask8-31 not shown for brevity)* OK >>

11.26 IMU:GET:QUATERNION

IMU:GET:QUATERNION <su_pswd>	
(Read) This will return the quaternion 4-element vector from the radar's internal 9-axis IMU. The IMU's compass/magnetometer is not enabled due to internal interference drift issues, but the accelerometer and gyroscope (6-axis mode) is valid for a stable roll and pitch determination of the radar. NOTE - The yaw/heading component of the vector (due to gyroscope drift uncorrected by the disabled magnetometer) will change over time and is only useful for short timeframe tracker usage where the beam schedule timing makes any IMU angular drift negligible.	
Read Syntax	IMU:GET:QUATERNION <su_pswd>
Read Returns	4 element quaternion unit vector
Example *Returns the current orientation of the radar as a quaternion vector*	>> IMU:GET:QUATERNION <su_pswd> Quaternion: w: +0.02086488 x: +0.11075675 y: -0.99345207 z: -0.01872279 OK

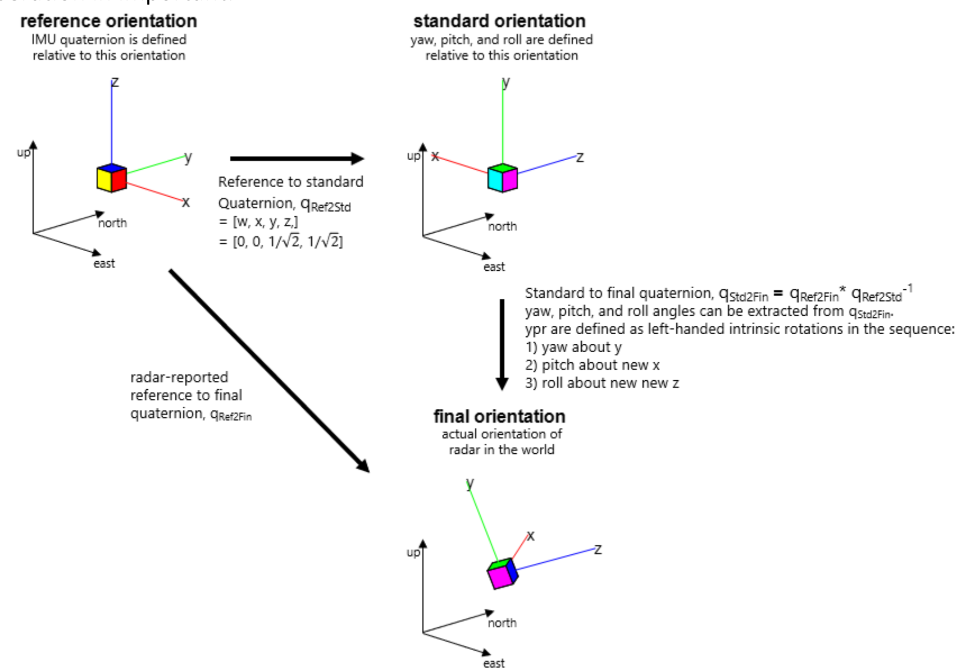
11.27 IMU:NO_RADAR_MOTION

IMU:NO_RADAR_MOTION[?] <su_pswd> <value>	
(Read/Write) Enables or disables the internal flag that causes the tracker/beam scheduler to bypass the internal IMU quaternion readings and instead assume a fixed angular orientation with a default quaternion [w, x, y, z] = [0 0 1/sqrt(2) 1/sqrt(2)] equivalent to [roll, pitch, yaw] = [0, 0, 0] degrees (for example, the radar Z-axis parallel with ground pointed North). This allows any IMU drift/noise to be ignored when it is known that the radar is mounted on a fixed and stable platform on the ground (hence this flag is enabled by default in EchoGuard and disabled by default in EchoFlight radars). The default orientation can be updated by the PLATFORM:STATE:ORIENTATION command to a custom roll, pitch and yaw. For EchoGuard, if mounted on a fixed mast w/no movement, the default setting of TRUE is recommended. If mounted on a swaying mast or mounting with dynamic roll and pitch of the platform, then set to FALSE to allow automatic tracker motion compensation. NOTE – This setting is not sticky, therefore, the defaults will be restored upon reboot or an Operation Mode change.	
Write Parameters	<flag>: string, [TRUE, FALSE] If TRUE (1), then the radar is assumed to be motionless and fixed quaternions are used for track generation. If FALSE (0), then the real-time quaternion is used for the tracker to compensate for radar motion.
Write Syntax	IMU:NO_RADAR_MOTION <su_pswd> <flag>
Read Syntax	IMU:NO_RADAR_MOTION? <su_pswd>
Read Returns	1 if Fixed, 0 if IMU input enabled
Factory Default Value	1 (Fixed, SSR), 0 (IMU, DAA)
Example	>> IMU:NO_RADAR_MOTION <su_pswd> TRUE OK >> IMU:NO_RADAR_MOTION? <su_pswd> 1

11.28 PLATFORM:STATE:ORIENTATION

PLATFORM:STATE:ORIENTATION[?] <su_pswd> <heading> <pitch> <roll>

(Read/Write) Sets the static orientation of the radar (orientation used if radar is not referencing the internal IMU as defined by the IMU:NO_RADAR_MOTION flag). This command provides the radar awareness of its external environment by defining the static angular orientation of the radar (roll, pitch, yaw) to enable height-based masking of measurements. Setting this value in conjunction with PLATFORM:STATE:ELEVATIONAGL and the AGLMASK commands provides the ability to mask measurements globally based on height relative to the radar. This is an effective method for minimizing ground contacts generating spurious tracks when primarily interested in UAV tracking or vice versa.
The coordinate convention for roll, pitch and yaw is given in the diagrams below using left-handed intrinsic rotations. From the perspective of the radar looking out as the scene, yaw/heading is about the Y axis (compass), pitch about the X axis (+ forward, - is backward), and roll is about the Z-axis. An Application Note is available to explain this in detail as the order of operation in important.



This command may be dynamically updated while the unit is in SWT mode which may be useful when the radar is in motion and changing orientation.

Write Parameters	<heading> <pitch> <roll>: (float), {-360 to +360} bounds for each angle
Write Syntax	PLATFORM:STATE:ORIENTATION[?] <su_pswd> <heading> <pitch> <roll>
Read Syntax	PLATFORM:STATE:ORIENTATION? <su_pswd>
Read Returns	The current angular orientation static setting of the radar.
Factory Default Value	Radar heading: 0.000000 (degrees) Radar pitch: 0.000000 (degrees) Radar roll: 0.000000 (degrees)
Examples	>> PLATFORM:STATE:ORIENTATION? <dev_pswd> Radar heading: 0.000000 (degrees) Radar pitch: 0.000000 (degrees) Radar roll: 0.000000 (degrees) OK >> PLATFORM:STATE:ORIENTATION <dev_pswd> 45, 20, -1 OK >> PLATFORM:STATE:ORIENTATION? <dev_pswd> Radar heading: 45.000000 (degrees) Radar pitch: 20.000000 (degrees) Radar roll: -1.000000 (degrees) OK

11.29 PLATFORM:STATE:ELEVATIONAGL

PLATFORM:STATE:ELEVATIONAGL[?] <su_pswd> <value>	
(Read/Write) This command provides the radar awareness of its external environment by defining the elevation or height above ground level (AGL) in meters. Setting this value in conjunction with PLATFORM:STATE:ORIENTATION and the AGLMASK commands provides the ability to mask measurements globally based on height. This is an effective method for minimizing tracks on ground contacts when primarily interested in UAV tracking. Note, the radar has no awareness of terrain variability (for example, no internal digital elevation model) so this command is assumed to be in reference to a mean ground level of interest. This command may be dynamically updated while the unit is in SWT mode which might be useful if the radar is in motion.	
Write Parameters	<value>: (float), [>=0.0] in meters.
Write Syntax	PLATFORM:STATE:ELEVATIONAGL <su_pswd> <value>
Read Syntax	PLATFORM:STATE:ELEVATIONAGL? <su_pswd>
Read Returns	Platform elevation in meters
Factory Default Value	0.0
Examples	>> PLATFORM:STATE:ELEVATIONAGL 45.7 OK >> PLATFORM:STATE:ELEVATIONAGL? 45.7 OK

11.30 AGLMASK:MINAGL

AGLMASK:MINAGL[?] <su_pswd> <value>	
(Read/Write) Sets or returns the minimum height for a measurement to not be masked (for example, all measurements below this floor will be prevented from associating to tracks). AGLMASK:MINAGL must be less than or equal to AGLMASK:MAXAGL. This command may be dynamically updated while the unit is in SWT mode.	
Write Parameters	<value>: (float), [-10,000.0, to min(AGLMASK:MAXAGL, 10,000.0] in meters
Write Syntax	AGLMASK:MINAGL <su_pswd> <value>
Read Syntax	AGLMASK:MINAGL? <su_pswd>
Read Returns	The minimum mask height in meters
Factory Default Value	-10,000.0
Examples	>> AGLMASK:MINAGL? <su_pswd> -10000.000000 OK >> AGLMASK:MINAGL <su_pswd> 45 OK >> AGLMASK:MINAGL? <su_pswd> 45.000000 OK

11.31 AGLMASK:MAXAGL

AGLMASK:MAXAGL[?] <su_pswd> <value>	
(Read/Write) Sets or returns the maximum height for a measurement to not be masked (for example, all measurements above this ceiling will be prevented from associating to tracks). AGLMASK:MAXAGL must be greater than or equal to AGLMASK:MINAGL. This command may be dynamically updated while the unit is in SWT mode.	
Write Parameters	<value>: (float), [max(-10,000.0,AGLMASK:MINAGL) to 10,000.0] in meters
Write Syntax	AGLMASK:MINAGL <su_pswd> <value>
Read Syntax	AGLMASK:MINAGL? <su_pswd>
Read Returns	The maximum mask height in meters
Factory Default Value	10,000.0
Examples	>> AGLMASK:MAXAGL? <su_pswd> 10000.000000

	OK >> AGLMASK:MINAGL <su_pswd> OK >> AGLMASK:MINAGL? <su_pswd> 45.000000 OK

11.32 AGLMASK:ENABLE

AGLMASK:ENABLE[?] <su_pswd> <flag>	
(Read/Write) Enables or disables the use of the height based AGLMASK feature. This feature is similar to a global hard zone mask based on min/max elevation relative to the radar. Enable this mask only after defining the radar height and min/max ALG mask levels. When enabled, measurements below the MINAGL or above the MAXAGL mask limits will be prevented from associated to tracks in the manner of a 'hard' mask. Can be changed while operating.	
Write Parameters	<flag> - (string), [TRUE,FALSE]. Flag to enable/disable the AGL mask
Write Syntax	AGLMASK:ENABLE <su_pswd> <flag>
Read Syntax	AGLMASK:ENABLE? <su_pswd>
Read Returns	The flag state (0 or 1) where 0 = disabled and 1 = enabled
Factory Default Value	0 (FALSE, disabled)
Examples	>> AGLMASK:ENABLE? <su_pswd> 0 OK >> AGLMASK:ENABLE <su_pswd> TRUE OK >> AGLMASK:ENABLE? <su_pswd> 1 OK

11.33 NOISEFLOOR?

NOISEFLOOR? <su_pswd>	
(Read) This command generates a simplified report of the RDmap (aka RVmap) mean amplitude in dB scale (for example, $20\log_{10}$(mean value) grouped in 250m range sections, over 30 degree Azimuth sections for a given specified Elevation cut. The elevation cut can be specified by NOISEFLOOR:EL and the number of averaged beams specified by NOISEFLOOR:NUMAVE. The command queries the result of the RDmap acquisition and prints out the summary. If the data is not ready (or even started) it will report the following message to user: "Data Not Ready" (or "Data Not Collected"). Note, operating on SEARCH mode may take a long time to complete depending on the defined step sizes.	
Read Syntax	NOISEFLOOR? <su_pswd>
Read Returns	Summary RDmap mean amplitude or "Date Not Ready" or "Data Not Collected"
Factory Default Value	n/a
Examples	>> MODE:SWT:START OK >> NOISEFLOOR:START <su_pswd> OK >> NOISEFLOOR? <su_pswd> Noise floor Estimate Summary (values in dB of mean amplitude) Elevation 0 (deg) Min Range(m) = 22 , Max Range(m) = 5984 Range(m) Az(-60:-30) Az(-30:0) Az(0:30) Az(30:60) 0 52.4 53.3 54.3 53.6 250 56.6 57.7 58.7 58.2 500 56.7 57.5 58.5 58.2 750 56.3 57.0 58.0 57.4 1000 55.8 56.5 57.1 56.7

	1250	55.3	55.9	56.6	56.4
	1500	54.8	55.4	55.9	55.7
	1750	54.4	55.0	55.4	55.3
	2000	54.0	54.5	55.1	54.8
	2250	53.9	54.3	54.7	54.4
	2500	53.6	53.9	54.3	54.3
	2750	53.3	53.8	54.4	53.9
	3000	53.2	53.7	53.9	53.7
	3250	53.1	53.4	53.7	53.6
	3500	53.1	53.3	53.5	53.5
	3750	52.8	53.2	53.2	53.2
	4000	52.7	52.9	53.1	53.2
	4250	52.6	52.9	53.2	53.0
	4500	52.5	52.9	53.1	52.8
	4750	52.5	52.8	52.8	52.9
	5000	52.6	52.7	52.9	52.8
	5250	52.4	52.6	52.8	52.7
	5500	52.2	52.6	52.7	52.5
	5750	52.2	52.3	52.5	52.5
	OK				
	>>				

11.34 NOISEFLOOR:START

NOISEFLOOR:START <su_pswd>	
(Write) Starts the collection of RDmaps to support the NOISEFLOOR? analysis and reporting command. Can only be invoked while operating in SWT or SEARCH modes.	
Write Syntax	NOISEFLOOR:START <su_pswd>
Examples	>> NOISEFLOOR:START <su_pswd> Command not allowed in IDLE mode. Must be in SWT or SEARCH mode. Command Not Available NA MESA-000699 >> MODE:SWT:START OK MESA-000699 >> NOISEFLOOR:START <su_pswd> OK

11.35 NOISEFLOOR:EL

NOISEFLOOR:EL [?] <su_pswd> <value>	
(Read/Write) Specifies the desired Elevation angle to average RDmaps for the NOISEFLOOR command	
Write Parameters	<value>: even integer, [-40: 40]
Write Syntax	NOISEFLOOR:EL <su_pswd> <value>
Read Syntax	NOISEFLOOR:EL? <su_pswd>
Read Returns	Specified elevation angle
Factory Default Value	0 (degrees)
Example	>> NOISEFLOOR:START <su_pswd> OK >> NOISEFLOOR:EL? <su_pswd> 0 OK >> NOISEFLOOR:EL <su_pswd> 12 OK >> NOISEFLOOR:EL? <su_pswd>

	12 OK

11.36 NOISEFLOOR:NUMAVE

NOISEFLOOR:NUMAVE [?] <su_pswd> <value>	
(Read/Write) Specifies the desired number of RDmaps to average for the NOISEFLOOR command	
Write Parameters	<value>: integer, [1:4]
Write Syntax	NOISEFLOOR: NUMAVE <su_pswd> <value>
Read Syntax	NOISEFLOOR:NUMAVE? <su_pswd>
Read Returns	Number of averaged RDmaps
Factory Default Value	1
Example	>> NOISEFLOOR:NUMAVE? <su_pswd> 1 OK >> NOISEFLOOR:NUMAVE <su_pswd> 4 OK >> NOISEFLOOR:NUMAVE? <su_pswd> 4 OK

11.37 OUTPUT:THROTTLETRACKRATE:ENABLE

OUTPUT:THROTTLETRACKRATE:ENABLE[?] <su_pswd> <flag>	
(Read/Write) Enable/disable the throttling of the track reporting rate as specified by the OUTPUT:MAXTRACKRATE command.	
Write Parameters	<flag> - (string), [TRUE,FALSE]. Flag to enable/disable track reporting rate limiting
Write Syntax	OUTPUT:THROTTLETRACKRATE:ENABLE <su_pswd> <flag>
Read Syntax	OUTPUT:THROTTLETRACKRATE:ENABLE? <su_pswd>
Read Returns	The flag state (0 or 1) to disable or enable the flag
Factory Default Value	0 (FALSE, unthrottled)
Examples	>> OUTPUT:THROTTLETRACKRATE:ENABLE? <su_pswd> 0 OK >> OUTPUT:THROTTLETRACKRATE:ENABLE <su_pswd> TRUE OK >> OUTPUT:THROTTLETRACKRATE:ENABLE? <su_pswd> 1 OK

11.38 OUTPUT:MAXTRACKRATE

OUTPUT:MAXZTRACKRATE <su_pswd> <value>	
(Read/Write) Adjusts the maximum potential reporting rate of the track packet for systems and networks that have limited bandwidth and desire slower track reporting. The internal tracker module will still continue to function at the default track update rate to maintain accurate tracks and schedule the necessary SQL beams. This parameter sets the upper bound for track packet reporting frequency which will be limited by the available track packet updates aligned to the CPI boundaries. Also requires enabling the OUTPUT:THROTTLETRACKRATE:ENABLE flag.	
Write Parameters	<value>: float, [1.0, 15.0] in Hertz
Write Syntax	OUTPUT:MAXTRACKRATE <su_pswd> <value>
Read Syntax	OUTPUT:MAXTRACKRATE? <su_pswd>
Read Returns	The reporting rate in Hertz

Factory Default Value	10.0
Examples	>> OUTPUT:MAXTRACKRATE <su_pswd> 0.5 The entered track output rate is out of range (1.000000 - 15.000000) Hz Invalid Parameter IC >> OUTPUT:MAXTRACKRATE <su_pswd> 1 OK >> OUTPUT:MAXTRACKRATE? <su_pswd> 1.000000 OK

11.39 OUTPUT:STATUSRATE

OUTPUT:STATUSRATE <su_pswd> <value>	
(Read/Write) Adjusts the nominal reporting rate of the status packet for systems and networks that have limited bandwidth and desire slower reporting.	
Write Parameters	<value>: float, [0.1, 20.0] in Hertz
Write Syntax	OUTPUT:STATUSRATE <su_pswd> <value>
Read Syntax	OUTPUT:STATUSRATE? <su_pswd>
Read Returns	The reporting rate in Hertz
Factory Default Value	20.0
Examples	>> OUTPUT:STATUSRATE? <su_pwd> 20.000000 OK >> OUTPUT:STATUSRATE <su_pwd> 0.5 OK >> OUTPUT:STATUSRATE? <su_pwd> 0.500000 OK

11.40 NET:KEEPALIVECNT

NET:KEEPALIVECNT <su_pswd> <value>	
(Read/Write) Configures the number of unacknowledged keepalive probes that the radar will send before closing the TCP connection. Probes are spaced 5 sec apart. NOTE: Radar will refuse new connections until configured keepalive count expires.	
Write Parameters	<value>: integer, [1:20]
Write Syntax	NET:KEEPALIVECNT <su_pswd> <value>
Read Syntax	NET:KEEPALIVECNT? <su_pswd>
Read Returns	The current keepalive count
Factory Default Value	3
Examples	>> NET:KEEPALIVECNT? <su_pswd> 3 OK>> NET:KEEPALIVECNT <su_pswd> 2 OK >> NET:KEEPALIVECNT? <su_pswd> 2 OK

11.41 NET:PORT23ENABLE

<table>
<tr><th colspan="2">NET:PORT23ENABLE <su_pswd> <value></th></tr>
<tr><td colspan="2">(Read/Write) Adds a command to the radar that disables port 23. This command/setting should be persistent such that it persists through power cycles.

NET:PORT23ENABLE <super user pswd> false - turns off port 23
NET:PORT23ENABLE <super user pswd> true - turns on port 23
NET:PORT23ENABLE? <super user pswd> - returns the current value

This new command results in additions to all SYSPARAM queries.

Note: With any changes made to the Network Param Config block, downgrading from the new/latest version will result in a reset of the IP address to: 169.254.1.10

IMPORTANT: For this command to take effect, a power reset or RESET:SYSTEM is required.

Caution: This setting is persistent and can only be reset to the default value through a Hardware Reset Operation OR set NET:PORT23ENABLE <super user pswd> true. If the IP is lost/forgotten, it will not be possible to communicate with the radar until reset to default.

A RESET:PARAMETERS:HARD (same as shorting GPIO1 in the cable) should reset the NET:PORT23ENABLE to true (enabled). This is consistent with other params as the RESET:PARAMETERS:HARD is thought of as a reset to factory defaults. Executing the RESET:PARAMETERS:HARD command also resets the IP address, gateway, and so forth.</td></tr>
<tr><td>Write Parameters</td><td>True or False</td></tr>
<tr><td>Write Syntax</td><td>NET: PORT23ENABLE <su_pswd> <value></td></tr>
<tr><td>Read Syntax</td><td>NET: PORT23ENABLE <su_pswd></td></tr>
<tr><td>Read Returns</td><td>1 or 0</td></tr>
<tr><td>Factory Default Value</td><td>True</td></tr>
<tr><td>Examples</td><td>>> NET: PORT23ENABLE ? <su_pswd> 3
OK>> NET:PORT23ENABLE <su_pswd> 2
OK
>> NET: PORT23ENABLE ? <su_pswd>
2
OK</td></tr>
</table>

12. VOLATILITY OF USER PARAMETERS

Commands marked as (Read/Write) provide write access to system parameters. Most parameters are volatile and will return to their default value after power-cycling the MESA unit. However, some parameter changes will persist through power cycling and can be returned to their default values only by either issuing the RESET:PARAMETERS command, or performing a Hardware Reset Operation. The reset behavior of each command depends on what 'Persistence Category' it is in.

Persistent parameters – These parameters are not reset to their defaults by power cycling or issuing RESET:PARAMETERS. These parameters can only be reset to their default values by a Hardware Reset Operation.

Sticky parameters – These parameters are not reset to their default values by power cycling. They are reset to their defaults by either issuing RESET:PARAMETERS, or with a Hardware Reset Operation.

Volatile parameters – These parameters are reset to their default values by power cycling, issuing RESET:PARAMETERS, or with a Hardware Reset Operation.

Table 19 - The reset-to-default behavior of each of the parameter Persistence Categories.

	Parameter Type		
Reset Operation	**Persistent**	**Sticky**	**Volatile**
Power Cycle	No	No	Yes
Software Reset (RESET:PARAMETERS command)	No	Yes	Yes
Hardware Reset (pin to ground during boot)	Yes	Yes	Yes

Table 20 - Persistence Categories of all user and super-user system parameters, as defined by their write commands.

Parameter Categories		
Persistent	**Sticky**	**Volatile**
ETH:IP (IP address, netmask, and default gateway) IMU:CAL:SET:USER	MODE:SWT:OPERATIONMODE DMS:CHANNEL VOLTAGE:THRESHOLD	all other user parameters

13. TECHNICAL REFERENCES

13.1 Detailed Functional Description of MESA Radar

The MESA Radar block diagram below illustrates the basic radar architecture. Fundamentally this radar is a multi-chirp FMCW (Frequency Modulated Continuous Wave) radar that provides range and range rate (velocity) information. Range information from the delayed returned echo frequency spectrum is extracted from the first "Fast-Time" FFT (Fast Fourier Transform). Range rate is derived from the phase advance observed over multiple fast chirp cycles extracted from what is called the second or "Slow-Time" FFT. There are three main hardware sections of the radar: 1) Exciter Transmitter, 2) Receiver, 3) The Digital Controller and processor. The dual aperture architecture is classically referred to as Bi-Static since there are separate transmit and receive antenna. This separation provides greater than 60dB of TX to RX isolation allowing us to support simultaneous high-power CW transmission while maintaining a linear low noise figure receiver. The radar as delivered, has been fully tested and verified in our production synthetic target test chamber. This test system verifies the signal to noise ratio and detection of multiple objects at simulated velocity, ranges, and RCS levels over the entire MESA field of regard requiring no user calibration.

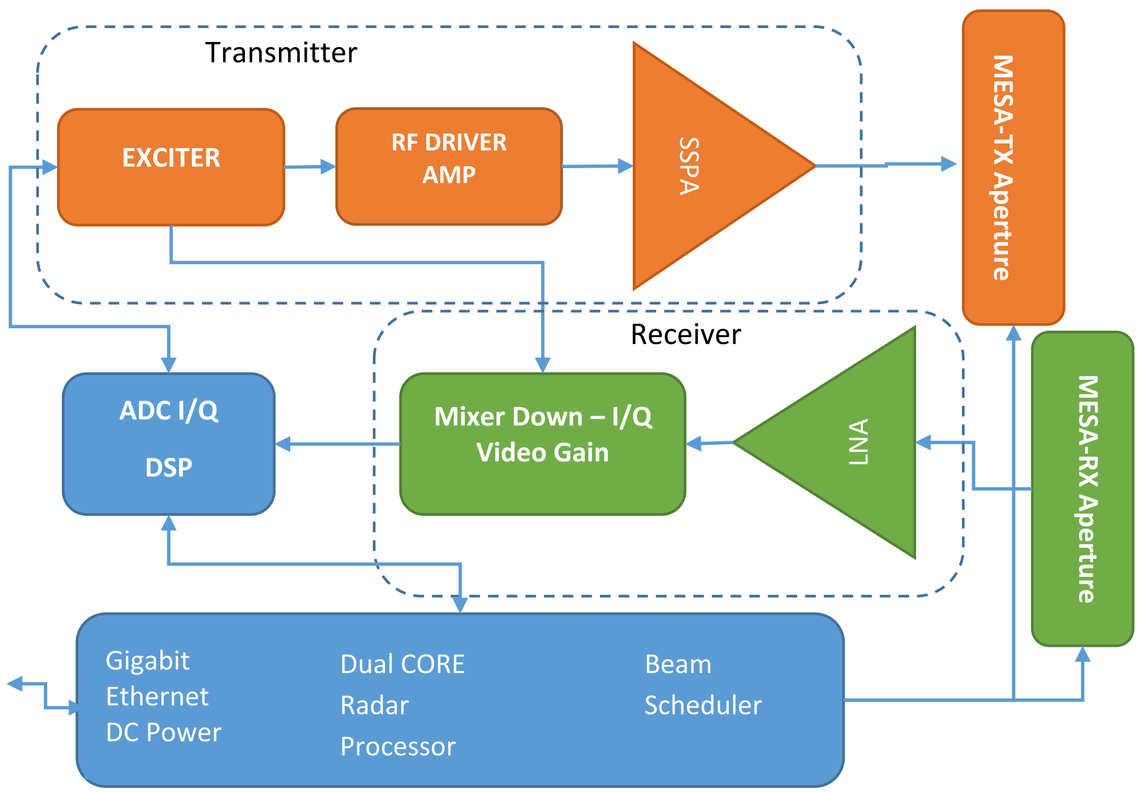


Figure 21 – Tx & Rx Paths

Radiating Apertures MESA-TX and RX – The Radar uses separate but identical MESA apertures for reception and transmission. Each aperture delivers a nominal gain of 21 dBi and has AZ/EL half power beam widths of 4°/12° respectively at bore sight. This results in a radar round trip half power effective beam width of nominally 2°/6° that is capable of scanning across 120° in azimuth and 80° of elevation. This phased array like performance is made possible by Echodyne's patented MESA. All possible antenna positions are pre-programed and verified requiring no user calibration.

Transmitter/Exciter – The transmitter is a solid-state balanced pair of PHEMT MMIC power amplifier devices set at a bias point to deliver 35dBm (3160mW) for EchoGuard. Transmit power is automatically blanked when radar is not being triggered to take data or when radar is stopped. Transmitter will also be disabled if the exciter is unlocked and reporting a BIT fault. To create the FMCW saw tooth radar waveform (chirp), an all-digital fractional N phase locked loop microchip is used to lock a high performance VCO to the system coherent reference crystal oscillator. This precisely controlled waveform generator drives the SSPA (Solid State Power Amplifier) and provides a second coherent sample to the receiver down-converter.

Receiver – The receiver consists of a single channel LNA (Low Noise Amplifier) followed by a high intercept balanced I/Q down-converter with video gain and filtering. The receiver is very robust against RF high power damage. First, the MESA antenna has a band-pass behavior with very high loss below 18GHz where most high-power emitters exist. Second, in-band reflected RF energy cannot damage the receiver until RX power exceeds +15dBm into the LNA. While our +33 to+35.5dBm transmitter is highly isolated from the receiver when radiating into free space, users are cautioned to maintain at least 1-meter safe distance from highly reflective surfaces to reduce risk of LNA damage. Following the down conversion, the I and Q signals are delivered to a precision balanced video gain chain that incorporates both high-pass and low-pass filter characteristics. The low-pass filter provides classic anti-aliasing while the high-pass filter blocks mixer DC-Offsets and prevents saturation from large close-range objects. The high-pass transfer function has a graceful R4 slope (-12dB Octave) creating a variable gain over the 0-200m range zone to reduce the effect of large-echo saturation at close ranges.

Digital Controller and Processer - The user communicates via the fast TCP/IP Gbit Ethernet packets by selecting ports (described in a later section). The Dual Core microprocessor parses commands, modes, and configures the user data stream selection. Once the radar is configured and triggered, all timing and control is managed using a synchronous FPGA (Field Programmable Gate Array) which aligns the sample timing with each radar waveform generation. Video I/Q Samples are delivered to the FPGA from the dual 14-bit ADC's. Dual FFT blocks process the I and Q data into a range velocity map and either package the data for delivery to the user or for further use in internal CFAR (Constant False Alarm Rate) detection and tracking algorithms. The Dual CORE radar processor also functions as the radar beam scheduler.

13.2 Search and Search-While-Track Data Processing

The MESA radar can be pointed in a user-specified direction using an automated scan using either the MODE:SEARCH:START (Sequential raster scan defined FOV) or MODE:SWT:START commands (smart Search-While-Track algorithm).

At every beam pointing direction the MESA points to, called a 'beam step', the MESA collects a single set of radar data and then performs two types of data processing on the collected data set. The first type of data processing produces a RVmap (Range-Velocity Map) that represents the magnitude of the radar return from all ranges and radial-velocities observed in the current beam direction. The second type of data processing uses a CFAR (Constant False Alarm Rate) detector to identify the range and radial velocities of objects in the current RVmap.

In Search-While-Track mode, a third layer of data processing then correlates in space and time the detections produced from all beam steps as the radar scans the Field of View (FoV) to track distinct, persistent, intruders. Each tracked intruder has an associated ID and various properties are estimated about each of them.

13.3 Radar Range-Velocity Map (RVmap) Description

The raw data produced by the MESA radar is a Range-Velocity Map (RVmap), also sometimes referred to as a Range Doppler Map. This standard type of radar data represents the amplitude of signal scattered from objects at all ranges and radial-velocities (velocities measured directly through the radial Doppler shift) in the direction that the radar is pointing the beam. Radial-velocity is defined as negative for closing objects. Each beam pointing direction has a unique range-velocity map.

range (m)	range bin number	Doppler bin number: 0 velocity (m/s): -14.55	1 -13.64	...	m 0.9·(m-16)	...	16 0.0	...	31 13.64
-416.7	0	$S_{0,0}$	$S_{0,1}$	...	$S_{0,m}$	...	$S_{0,16}$	...	$S_{0,31}$
-413.4	1	$S_{1,0}$	$S_{1,1}$	...	$S_{1,m}$	...	$S_{1,16}$	...	$S_{1,31}$
...	...	...	...	...	...	...	...	...	...
3.2552·(n-128)	n	$S_{n,0}$	$S_{n,1}$	...	$S_{n,m}$	...	$S_{n,16}$	...	$S_{n,31}$
...	...	...	...	...		...	...	...	...
0.0	128	$S_{128,0}$	$S_{128,1}$	...	$S_{128,m}$	...	$S_{128,16}$	...	$S_{128,31}$
...	...	...	...	...	...	...	...	...	...
6247.0	2047	$S_{2047,0}$	$S_{2047,1}$	...	$S_{2047,m}$	...	$S_{2047,16}$	...	$S_{2047,31}$

Figure 22 - Format of RVmap data array for range resolution dR=3.25m, radial-velocity resolution dV=0.91m/s, zero-range bin number n_0=128, and zero-Doppler bin number m_0=16.

The radar has a range resolution of dR and a radial-velocity resolution of dV. Thus, the RVmap is reported as a 2D array of range-velocity "bins" where each bin gives the total amount of signal coming from all objects within +-dR/2 and +-dV/2 of the bin's center range and velocity. Each bin's center range is given by its *zero-referenced* range bin number, n, minus the zero-range bin number, n_0, times the radar's range resolution,

$$R_{bin} = (n - n_0) \cdot dR \qquad (1)$$

The zero-range bin number as well as the total number of range bins are reported by the radar along with each range-velocity map dataset. Each bin's center velocity is given by its *zero-referenced* Doppler bin number, m, minus the zero-Doppler bin number, m_0, times the radar's velocity resolution,

$$v_{bin} = (m - m_0) \cdot dV \qquad (2)$$

The zero-Doppler bin number as well as the total number of Doppler bins are reported by the radar along with each range-velocity map dataset. The total number of ranges and velocities as well as the zero range and zero Doppler bin numbers depend on the waveform used by the radar.

The MESA radar has an MUV (Maximum Unambiguous (radial) Velocity) given by,

$$MUV = m_0 \cdot dV \qquad (3)$$

An object moving with a radial-velocity greater than MUV m/s or less than -MUV m/s will 'alias' around the RVmap and will be measured as having a radial-velocity that is modulo the MUV. An object moving with radial-velocity v relative to the radar will appear in the RVmap at,

$$v_{aliased} = (v + MUV)\%(2 \cdot MUV) - MUV = v - \mathrm{round}\left(\frac{v}{2 \cdot MUV}\right) \cdot 2 \cdot MUV \qquad (4)$$

$$= v - p \cdot 2 \cdot MUV$$

where,

$$p = \mathrm{round}\left(\frac{v}{2 \cdot MUV}\right) \in \mathbb{Z}$$

is the alias-zone number that the true velocity is in.

Despite the ambiguity of the aliased radial velocity in some situations it is useful for discriminating ground clutter from objects of interest. Ground clutter may be observed in certain environments when the MESA beam intersects the ground. The radial-velocity of ground clutter at radial range r, observed by a MESA radar at height h_{ground} above the ground and moving at velocity v_{ground} relative to the ground, when the beam is steered to an azimuth angle of az relative to the direction of motion is,

$$v_{ground\ clutter} = -\cos(az) \cdot \mathrm{sqrt}\left(1 - \left(\frac{h_{ground}}{r}\right)^2\right) \cdot v_{ground} \tag{5}$$

aliased by eqn. 4. Combining (5) and (4),

$$v_{ground\ clutter} = \beta \cdot v_{ground} + p \cdot 2 \cdot MUV = [\beta \quad -2 \cdot MUV] \begin{bmatrix} v_{ground} \\ p \end{bmatrix}$$

The observed clutter will have some velocity distribution around this predicted velocity due to Doppler spread of the clutter sources themselves, the finite width of the main-beam of the radar, and sidelobes with different az pointing directions than the main-beam.

The RVmap is streamed to the user after each beam step on the RVmap TCP/IP data port (See TCP/IP Data Ports section).

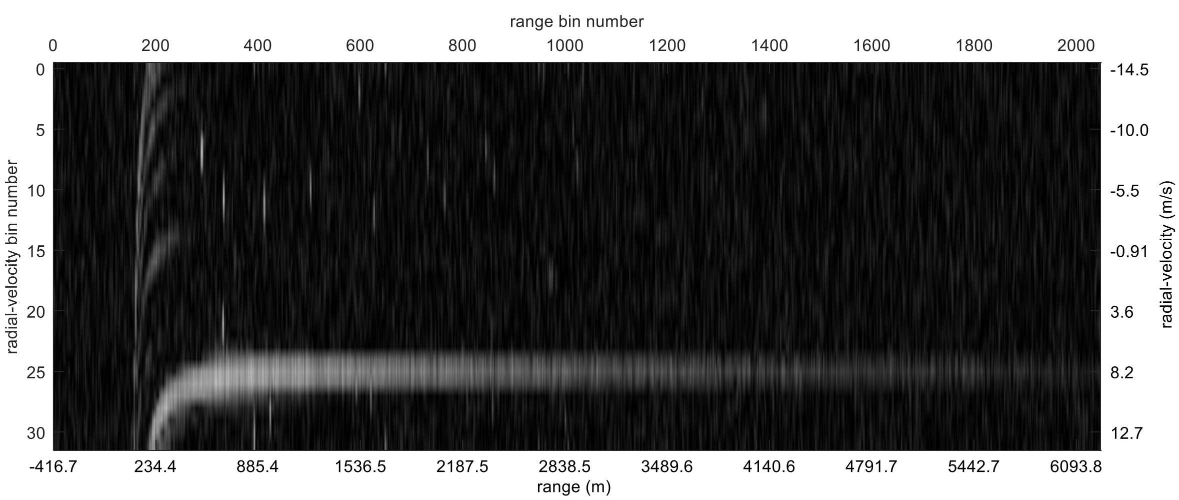


Figure 23 - A representative RVmap for a beam pointed at broadside when the radar is moving at 50m/s over the ground at an altitude of 100m. Several features typical of RVmap data can be identified in this figure. A dozen distinct objects at various ranges moving with various radial-velocities can be identified as bright white blips localized in range and radial-velocity. Ground clutter returns can be seen as a white band at all ranges beyond approximately 500m centered at 8.2m/s. This is where the beam intersects the ground. The radial velocity of the ground clutter is equal to the radar's aliased velocity over the ground of 8.2m/s. The swooping white bands at close ranges are where the antenna side-lobes intersect the ground.

13.3.1 Waveform Options – 6D32 vs 3D128

EchoGuard supports two waveform options. The legacy 32-downchirp waveform with RVmap dimensions of 32 x 2048 bins. This waveform has historically been referred to as 'B1a' and is now called 6D32 to define the maximum instrumented range (~6km), D for downchirp and 32 for the # of chirps which corresponds to velocity bin increments. This waveform has a MUV (maximum unambiguous velocity of +/-15 m/s and a velocity resolution of approximately 0.9 m/s.

A second waveform, historically referred to as 'B1b' and now called 3D128 to define the maximum instrumented range (~3km), D for downchirp and 128 for the # of chirps or velocity bin increments. This waveform has a MUV (maximum unambiguous velocity) of +/-30 m/s and a velocity resolution of approximately 0.45 m/s. The CPT (coherent processing time) of this waveform is twice the 6D32 waveform, resulting in ~3dB additional processing gain at the expense of half the effective beam rate. 3D128 is optimized for tracking slow moving targets such as walkers at greater ranges. While it can be used for cUAS missions, it is more prone to track saturation in busy scenes, requiring optimal site tuning and clutter filter settings. By default, the cUAS classifier is also not trained for this waveform, resulting in reduced accuracy, although it may be enabled at the user's discretion.

The waveforms are defined by the FPGA image loaded into the EchoGuard radar. To switch between waveforms, the user can load the respective .zip image file into the EchoGuard unit.

13.4 Radar CFAR Detection Description

Each RVmap collected by the MESA radar is processed for probable real objects, called 'detections', using a CFAR (Constant False Alarm Rate) algorithm. The false alarm rate is adjustable in the MESA radar and varies between 1E-7 and 1E-5. Each detection may be a return from a real object the radar is observing, or it may be a false alarm due to environmental clutter or noise. The time, relative location, radial-velocity, signal-to-noise-ratio, and power of each detection is logged and can be reported to the user. The relative location is recorded as (azimuth, elevation, range) in radar coordinates as well as (X, Y, Z) antenna coordinates. Detections are streamed to the user after each beam pointing direction on the Detections TCP/IP data port (See TCP/IP Data Ports).

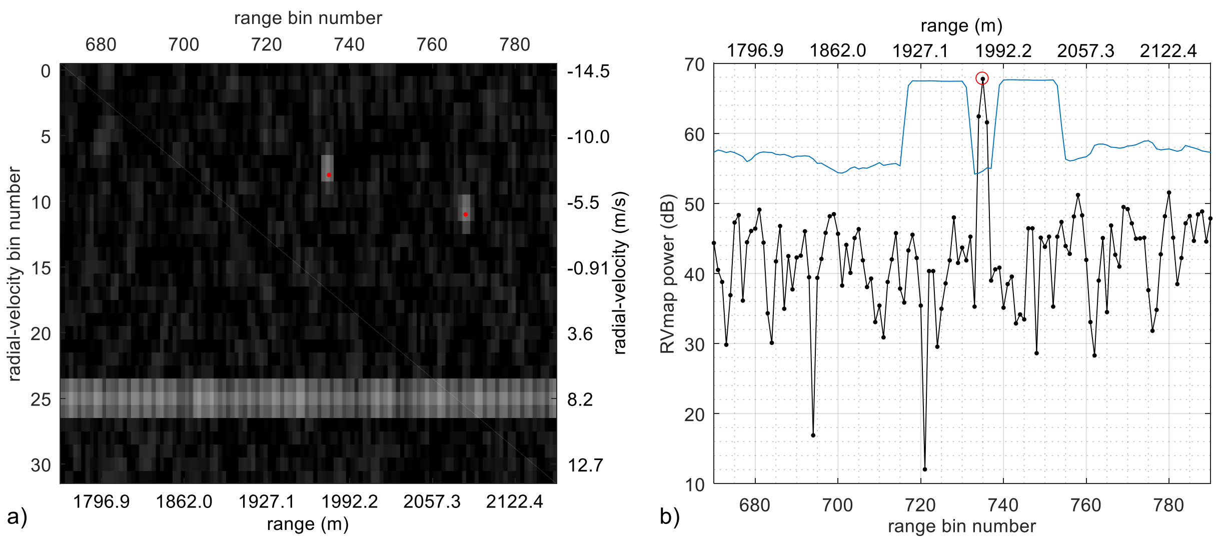


Figure 24 – a) Detections plotted on the 2D RVmap in red. For each detected object both the range and radial velocity can be identified from the radar data. b) A CFAR threshold applied to velocity row 8 of the RVmap from a). The RVmap data is shown in black and the CFAR threshold is shown in blue. While three points of the object return surpass the threshold, only the peak is reported as a detection, circled in red.

13.4.1 Sidelobe Clustering

One of the realities of a scanned radar with finite aperture size is that the main lobe response for the intended look direction is co-mingled with side lobe responses from adjacent angles. Consequently, bright objects (for example, trucks) moving in the aperture side lobes can generate detections equal in magnitude to dimmer objects in the main lobe (for example, drones, birds). This phenomenon manifests itself as rings of detections over angle at constant range that move with the bright object, causing the tracker to waste resources trying to follow targets that are not real. Since bright objects seen in sidelobes will present in most beams, they are easy to statistically separate from real targets and mask. This feature is enabled by default.

13.5 Radar Track Formation/Termination Process

Radar tracking is functionally a consolidation of a series of detections using a probabilistic predictive analysis to gain confidence in the position, bearing and velocity of objects of concern within the FOV. This method of reporting is the most concise and simplest to use for mapping. A track occurs while in the Search While Track (SWT) mode only, and represents a detection that has been repeatedly detected to achieve a confidence level sufficient to justify elevation to receive a TRACK ID. Once a track ID is created it is reported to the user and continuously verified in an automated sequence until terminated by the radar tracker. The process used for a general tracker is outlined in the flow chart. Tracking prediction is done through use of Extended Kalman filtering algorithms. More detail of this process is covered in the SWT description.

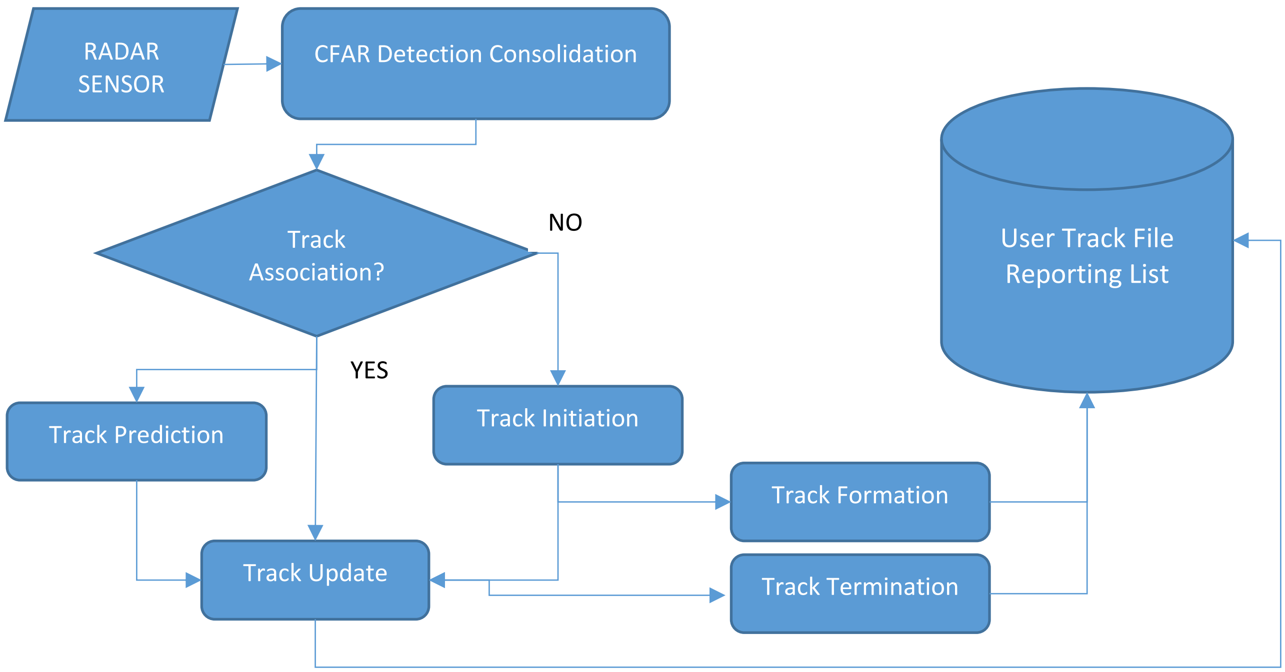


Figure 25 - Tracking Process Diagram

14. ACRONYM LIST

AGL	Above Ground Level
BW	Beam width
CFAR	Constant False Alarm Rate
C-SWAP	Cost, Size, Weight, and Power
CW	Continuous Wave
DAA	Detect-and-Avoid
dBi	Decibels-isotropic
dBsm	Relative magnitude; decibels relative to a square meter; Radar Cross Section of target
DOCA	Distance of Closest Approach
ERP	Effective Radiated Power
FCC	Federal Communications Commission
FFT	Fast Fourier Transform
FMCW	Frequency Modulated Continuous Wave
FOR	Field of Regard
FoV	Field of View
GBit	Gigabit
IMU	Inertial Measurement Unit
LNA	Low Noise Amplifier
MCU	Multipoint Control Unit
MESA	Metamaterials Electronically Scanning Array
MUV	Maximum Unambiguous Velocity
NUE	North-East-Up
OOB	Out of Bounds
PCB	Printed Circuit Board
Pd or Pdet	Probability of Detection
PFA	Probability of False Alarm
RCS	Radar Cross Section
RVMap	Range-Velocity Map
RX	Receive
SSPA	Solid State Power Amplifier
SSR	Security and Surveillance Radar
sUAS	Small Uncrewed Aerial System
SWT	Search-While-Track
TCM	Time Channel Mitigation
TCP/IP	Transmission Control Protocol/Internet Protocol
TOCA	Time of Closest Approach
TX	Transmit
Vdc	Voltage in a Direct Current
VRadial	Radial Velocity