

Developer Advisory — Scan/Search performance To keep scan/search speeds high, configure ranges and store all scan/search data *inside the receiver* and let the radio run its own engines. Driving each step from a PC (e.g., repeatedly sending RF/reading status) is limited by serial throughput and will make scanning/searching **much slower**.

Recommended workflow

- **Configure & store on-device:** Use search-bank setup SE (with SL/SU/ST/SH/MD/AU/PT/TT), link banks with BK, and set groups via SG (or VFO search settings with VE).
- **Run on-device:** Start searches/scans with SSbb (Search Bank), MSbb (Scan), VS (VFO Search), or CYbb (Cyber Search).
- **Capture hits on-device:** Enable Auto Store AS1 (in SG/VE) so hits are saved to memory, or consolidate with MX after detection. Avoid per-hit writes over serial during an active run.
- **Transfer after completion:** When finished, read memories from the radio (MA/MR) or export via SD backups SD MMW (then inspect with SD DIR).

Avoid during active scan/search: high-rate polling (RX, LM) or step-by-step RF control, which forces slow serial round-trips. If you need status taps, use LC/RT/LT sparingly.

Notation: <CR> means Carriage Return, <LF> means Line Feed. Square brackets [...] in the examples denote the result code prefixed responses returned by the AR-DV3. Values, ranges, and sample payloads mirror the original specification exactly.

Result Codes

- **10: PC_RESULT_SEND** — Messages output unrelated to commands (e.g., shutdown notices)
 - **20: PC_RESULT_OK** — Command completed successfully
 - **+1: PC_RESULT_CONTINUE** — More response lines follow
 - **30: PC_RESULT_CAN_NOT_SET_ERR** — Cannot set due to current conditions
 - **40: PC_RESULT_FORMAT_ERR** — Command format error
 - **50: PC_RESULT_OUT_RANGE_ERR** — Argument out of range
 - **60: PC_RESULT_NONE** — Command does not exist
-

Version Information

Read

- VR<CR> — Pack-file version info → [20VR2402A <CR><LF>]
- VRP<CR> — Panel CPU version → [20VRP2402A <CR><LF>]
- VRC<CR> — Main CPU version → [20VRC2402B <CR><LF>]
- VRF<CR> — Main board FPGA version → [20VRF2401A <CR><LF>]
- VRD<CR> — Main board DSP version → [20VRD2401A <CR><LF>]
- VRA<CR> — All versions → [20VRA2402A C2402B F2401A D2401A P2402A <CR><LF>]

Version numbering: 1–2 chars = year (YY), 3–4 = month (MM), 5th = letter for build within month starting at “A”. A 6th variant character may appear.

Remote Lock Release

Write

- EX<CR> → [20 <CR><LF>]
-

Audio Volume

Write

- AGnn<CR> → [20 <CR><LF>] (nn = 00 mute to 99 max)

Read

- AG<CR> → [20AGnn <CR><LF>]
-

Frequency Setting

Write

- RF351.2<CR> → [20 <CR><LF>]
 - Unit MHz; decimal point required. Range 0.1–3000.0 MHz.

Read

- RF<CR> → [20RFnnnn.nnnnn <CR><LF>] (4 integer digits, fixed 5 decimals; MHz)
-

Demodulation Mode

Write

- MDdan<CR> or MDda<CR> → [20 <CR><LF>]

Where:

- d (digital demod **read** mode; ignored when setting): 0=None, 1=D-STAR, 2=YAESU, 3=ALINCO, 4=D-CR, 5=P25, 6=dPMR, 7=DMR, 8=TETRA(T-DM), 9=TETRA(T-TC)
- a (digital demod **set** mode): 0=Auto, 1=D-STAR, 2=YAESU, 3=ALINCO, 4=D-CR, 5=P25, 6=dPMR, 7=DMR, 8=TETRA(T-DM), 9=TETRA(T-TC), F=Digital OFF
- n (analog demod): 0=FM, 1=AM, 2=SAH, 3=SAL, 4=USB, 5=LSB, 6=CW. n can be omitted; if omitted, a is forced to F.

Read

- MD<CR> → [20MDdan <CR><LF>]
-

IF Bandwidth

Write

- IFn<CR> → [20 <CR><LF>] (meaning of n depends on analog mode)
 - FM: 0=200kHz, 1=100kHz, 2=30kHz, 3=15kHz, 4=6kHz
 - AM: 0=15kHz, 1=8kHz, 2=5.5kHz, 3=3.8kHz
 - SAH/SAL: 0=5.5kHz, 1=3.8kHz
 - USB/LSB: 0=2.6kHz, 1=1.8kHz
 - CW: 0=500Hz, 1=200Hz

Read

- IF<CR> → [20IFn <CR><LF>]
-

ATT/AMP

Write ATn<CR> → [20]

- 0=AMP ON, 1=AMP OFF & ATT OFF, 2=10 dB ATT

Read AT<CR> → [20ATn]

Memory Channel (MemCh) Read

Read single channel

- MAbbcc<CR> → if registered: [20MXbbcc MPp RFffff.ffff STsss.ss SHaaa.aa MDdan AUn PTn TTtitle...]; else [20MXbbcc ---]
 - bb: 00–39 (Bank), cc: 00–49 (Ch). Sub-fields per their own commands.

Read bank (50 Ch)

- MAbb<CR> → lines [21MXbb00 ...] ... [20MXbb49 ...] (50 lines total)

Memory Channel Write

Write

- MXbbcc MPp RFffff.ffff STsss.ss SHaaa.aa MDdan AUn PTn TTtitle...<CR>
 - bb:00–39, cc:00–49. Put TT last. Notes:
 - If updating within ≤4 channels from current, the command is accepted immediately but may not reflect in MA read until a mode change (VFO/MemCh), an MM2/MM3, or when writing ≥4 channels away, which triggers the actual non-volatile write (takes time).
 - ST, SH, MD, AU interact; applied in order from the left.

MemCh Mode Select

Write MRbbcc<CR> → [20]

MemCh Pass

Write MPp<CR> (0=OFF, 1=ON) → [20]

Read MP<CR> → [20MPp]

Step Frequency

Write STsss.ss<CR> (kHz) → [20]

Read ST<CR> → [20STsss.ss]

Step Adjust

Write SHaaa.aa<CR> (kHz) → [20]

Read SH<CR> → [20SHaaa.aa]

MemCh Protect

Write PTp<CR> (0=OFF, 1=ON) → [20]

Read PT<CR> → [20PTp]

AGC

Write ACn<CR> → [20] where 0=Fast, 1=Mid, 2=Slow, 3=RF-G. Only valid in **AM**; other modes return code 30.

Read AC<CR> → [20ACn]

Digital Auxiliary Info (receive)

Read DK<CR> → [20DKssss....] (up to 48 bytes)

Digital Auxiliary Info Display

Write DJn<CR> → [20] where 0=Do not display, 1=Display on screen (per VFO/MemCh/SearchBank). **Read** DJ<CR> → [20DJn]

D-CR Scrambler Code

Write DCddddd<CR> (0=none, 1–32767=code) → [20]

Read DC<CR> → [20DCddddd]

DMR Slot

Write OTn<CR> → [20] where 0=1+2 (Slot1 priority), 1=2+1 (Slot2 priority), 2=Slot1 only, 3=Slot2 only

Read OT<CR> → [20OTn]

DMR Color Code

Write CCnn<CR> (0–16) → [20]

Read CC<CR> →

- If receiving DMR: [20CCnn,dd] (dd=in-use color code)
- If not receiving: [20CCnn]

DMR CC Mute

Write CMn<CR> → [20] where 0=No check (always hear), 1=Hear only when CC matches

Read CM<CR> → [20CMn]

P25 NAC Code

Write PCxxx<CR> (0x000–0xFFF) → [20]

Read PC<CR> → while receiving: [20PCxxx,rrr] (rrr=current NAC, hex); otherwise [20PCxxx]

P25 NAC Mute

Write PMn<CR> → [20] where 0=No check, 1=Only when NAC matches

Read PM<CR> → [20PMn]

NXDN RAN Code

Write NCnn<CR> (00–64 decimal) → [20]

Read NC<CR> → while receiving: [20NCnn,rr] (rr current RAN); otherwise [20NCnn]

NXDN RAN Mute

Write NMn<CR> → [20] where 0=No check, 1=Only when RAN matches

Read NM<CR> → [20NMn]

TETRA Slot

Write TSrn<CR> → [20] (r ignored/optional). n: 0=AUTO, 1=Slot1, 2=Slot2, 3=Slot3, 4=Slot4

Read TS<CR> → [20TSrn] (r=current slot; n=configured slot)

CTCSS On/Off/Reverse

Write CIn<CR> → [20] where 0=OFF, 1=ON, 2=REV (reverse tone). Only when FM IF BW ≤ 15kHz.

Read CI<CR> → [20CIn]

CTCSS Tone Frequency

Write CNnn<CR> where nn maps to specific tone (01=60.0Hz ... 52=254.1Hz, 99=Search).

Code	Hz	Code	Hz	Code	Hz	Code	Hz
01	60.0	02	67.0	03	69.3	04	71.9
05	74.4	06	77.0	07	79.7	08	82.5
09	85.4	10	88.5	11	91.5	12	94.8
13	97.4	14	100.0	15	103.5	16	107.2
17	110.9	18	114.8	19	118.8	20	120.0
21	123.0	22	127.3	23	131.8	24	136.5
25	141.3	26	146.2	27	151.4	28	156.7
29	159.8	30	162.2	31	165.5	32	167.9
33	171.3	34	173.8	35	177.3	36	179.9
37	183.5	38	186.2	39	189.9	40	192.8
41	196.6	42	199.5	43	203.5	44	206.5
45	210.7	46	218.1	47	225.7	48	229.1
49	233.6	50	241.8	51	250.3	52	254.1

99: Search (hit if any tone is present)

Read CN<CR> → if receiving tone: [20CNnnrr] (setting & detected), else [20CNnn]

Voice Scrambler On/Off

Write SIn<CR> → [20] (0=OFF, 1=ON). Only when FM IF BW ≤ 15kHz.

Read SI<CR> → [20SIn]

Voice Scrambler Frequency

Write SCnnn<CR> → [20] where nnn=200–700 (i.e., 2000–7000 Hz). Only when FM IF BW ≤ 15kHz.

Read SC<CR> → [20SCnnnn]

DCS On/Off

Write DIIn<CR> → [20] (0=OFF, 1=ON). Only when FM IF BW ≤ 15kHz.

Read DI<CR> → [20DIIn]

DCS Code

Write DSnnn<CR> → [20] (nnn per table; 0=OFF, 999=Search)

Code Code Code Code Code Code Code Code Code

017	023	025	026	031	032	036	043
047	050	051	053	054	065	071	072
073	074	114	115	116	122	125	131
132	134	143	145	152	155	156	162
165	172	174	205	212	223	225	226
243	244	245	246	251	252	255	261
263	265	266	271	274	306	311	315
325	331	332	343	346	351	356	364
365	371	411	412	413	423	431	432
445	446	452	454	455	462	464	465
466	503	506	516	523	526	532	546
565	606	612	624	627	631	632	654
662	664	703	712	723	731	732	734
743	754						

Special: 000 = OFF, 999 = Search (hit if any code is present).

Read DS<CR> → if receiving DCS: [20DSnnnnrr], else [20DSnnn]

Denoiser

Write NRn<CR> → [20] (0=OFF, 1=LOW, 2=MID, 3=HIGH)

Read NR<CR> → [20NRn]

Squelch Mode (Noise/Level)

Write SQn<CR> → [20] (0=AUTO, 1=Noise, 2=Level)

Read SQ<CR> → [20SQn]

Level Squelch

Write LQnn<CR> → [20] (00–99; sets both LevelSQ and NoiseSQ value)

Read LQ<CR> → [20LQnn]

Private read: LQ.<CR> → [20LQnnn.n] (dB-converted value)

Noise Squelch

Write NQnn<CR> → [20] (00–99; also sets LevelSQ)

Read NQ<CR> → [20NQnn]

Voice Squelch (VoiceSQ)

Write

- VQn<CR> → [20] (0=OFF, 1=ON; per current VFO/MemCh/SearchBank)
 - VQn VLm<CR> → set level m=0–7
 - VQn VTttt VLm<CR> → ttt ignored on DV3 (kept for DV1 compatibility)
Read VQ<CR> → [VQn VTttt VLm]
-

Beep Volume

Write BPn<CR> (0–7) → [20]

Read BP<CR> → [20BPn]

EQ Number

Write EQn<CR> (0–4) → [20]

Read EQ<CR> → [20EQn]

EQ Pattern

Write EPn xxxxxxxx<CR> → [20]

- n:0–4 (pattern slot); xxxxxxxx = 9 bands, each char maps: 9:-7, A:-6, B:-5, C:-4, D:-3, E:-2, F:-1, 0:±0, 1:+1, 2:+2, 3:+3, 4:+4, 5:+5, 6:+6, 7:+7
Read EPn<CR> → [20EPn xxxxxxxx]

LCD Color Set (UI Theme)

Write CSn<CR> (0–7) → [20] where pairs are Foreground/Background: 0=White/Black, 1=Black/White, 2=Gray/Black, 3=Black/Gray, 4=Orange/Black, 5=Black/Orange, 6=Green/Black, 7=Black/Green

Read CS<CR> → [20CSn]

Key Light Behavior

Write LBn<CR> → [20] where 0=OFF(always off), 1=CONT(always on), 2=AUTO(on 7s after operations)

Read LB<CR> → [20LBn]

Key Light Color

Write KLn<CR> → [20] where 0–7 → 0:OFF, 1:BLUE, 2:RED, 3:MAGENTA, 4:GREEN, 5:CYAN, 7:WHITE

Read KL<CR> → [20KLn]

Key Light Dimmer

Write LDn<CR> → [20] where 0=OFF(bright), 1=ON(dim)

Read LD<CR> → [20LDn]

LCD Backlight Brightness

Write BLn<CR> → [20] where 0=Bright, 1=Dark

Read BL<CR> → [20BLn]

SD Recording — SQ Skip

Write SD RSQn<CR> (0=OFF, 1=ON) → [20]

Read SD RSQ<CR> → [20SD RSQn]

Remote ID (Power-On ID)

Write ZInn<CR> → [20] (00–99; 00 means always accept ZP, other values require matching ID)

Read ZI<CR> → [20ZInn]

Result Code Prefixing

Write REn<CR> → [20] (0=OFF, 1=ON) — whether to prefix responses with result codes

Read RE<CR> → [20REn]

Power ON

Write ZP<CR> or ZP00<CR> → [20AOR AR-DV3] (powers on if off; if already on, re-emits boot message)

Write ZPnn<CR> (00–99) → Power ON only if matches ZI ID → [20AOR AR-DV3]

Power OFF

Write QP<CR> or QP00<CR> → [20AR-DV3 GOTO SHUTDOWN]

Write QPnn<CR> → power off only if matches ZI ID

SD Card Status

Read SD PST<CR> → [20SD STn] where 0=Ready to record, 1=Recording (00–99), 2=Playing, 3=SD BUSY, 4=No SD inserted

SD Recording Start/Stop

Write SD REC<CR> → start recording [20]

Write SD REC/<CR> → stop recording [20]

Last-One Memory Write

Write MM<CR> → [20]

Reset

Write RS n<CR> → [20] where 0=System Reset (keeps MemCh/SearchBank), 1=Full Reset (initializes all)

SD Playback Start/Stop

Write SD PLY dddddddd/ffffff<CR> → play /AR-DV3/dddddddd/ffffff.WAV (ext fixed to .WAV) [20]

Write SD PLY /<CR> → stop playback [20]

SD Capacity

Read SD INF<CR> → [20SD INF FREE: ffffMB(hhh.hH) TOTAL: ttttMB]

SD Directory Listing

Read SD DIR<CR> → list /AR-DV3 files; files: name, size(bytes), timestamp; subdirs have no extension and size shows <DIR>. Ends with filename ---.

Read SD DIR dddddddd<CR> → list /AR-DV3/dddddddd. For .WAV, “size” shows play time.

SD File Delete

Write

- SD DEL ffffffff.eee<CR> → delete /AR-DV3/fffffff.eee
- SD DEL dddddddd/fffffff.eee<CR> → delete under subdir
- SD DEL dddddddd<CR> → delete subdir **only if empty**

Notes: Even after [20], deletion may still be in progress; use SD PST until it returns 0. Issuing other SD commands while SD PST is 3 (busy) may interrupt deletion. Failure (e.g., during recording/playback or non-empty dir) returns [30SD DEL?].

D-CR Whitening Code

Write DWnnn<CR> → [20] (000–511; 000=auto analyze; others require match to demod)

Read DW<CR> → not receiving: [20DW nnn]; while receiving DCR/NXDN: [20DW nnn,rrr] where rrr is successfully analyzed code 001–511.

Search Lower/Upper Limits

Write SLffff.ffff<CR> / SUffff.ffff<CR> → [20] (MHz)

Read SL<CR> → [20SLffff.ffff]; SU<CR> → [20SUffff.ffff]

Search Bank Read/Write

Read SRnn<CR> → if registered: [20SEbb SL... SU... ST... SH... MD... AU... PT... TTtitle] (bb:00–39), else [30?]

Write SEbb SL... SU... ST... SH... MD... AU... PT... TTtitle<CR> → [20]

Notes:

- Setting within ≤4 banks is accepted immediately but may not reflect in SR until mode change; writing ≥4 banks away triggers actual write (takes time).
- ST, SH, MD, AU interact; applied left-to-right.

Search Start / Delete

Write SSbb<CR> → start searching bank bb:00–39 (unregistered returns [30?])

Write SXbb<CR> → delete search bank bb

Bank Link

Write BKaabbccddeeff...<CR> — concatenate 2-digit bank numbers to link → [20]

Read BK<CR> → [20BKaabb...] or 99 if no link

Search Group

Write SGnn DLddd FRnn ASn BKaabb...<CR> → [20]

- nn:00–09 (also selects that group), DL/FR/AS/BK per their commands
Read SGn<CR> → group nn settings; SG<CR> (no args) → current group number & settings

Delay Time

Write DLddd<CR> (000–100, unit 0.1s; 100 = HOLD) → [20]

Read DL<CR> → [20DLddd]

Free Time

Write FRnn<CR> (00–60 s; 00=OFF) → [20]

Read FR<CR> → [20FRnn]

Auto Store

Write ASn<CR> (0=OFF, 1=ON) → [20] ; **Read** AS<CR> → [20ASn]

Cyber Search Start

Write CYbb<CR> → [20] (bb:00–39; unregistered → [30?])

Offset Configuration

Write OF+nn<CR> → [20]

- + or - = offset direction; nn:1–39 index into offset list (frequencies set by OL). Applied to current VFO/MemCh/SearchBank.
Read OF<CR> → [20OF+nn]

Offset Frequency List Entry

Write OLnn RFffff.ffff<CR> → [20]

- nn:1–20 writable (21–39 are fixed); RF in MHz (< 1000 MHz)
Read OLnn<CR> (nn:1–39) → [20OLnn RFffff.ffff]

[MONI] Press/Hold Simulation

Write OXn<CR> → [20] where 0=MONI hold end, 1=MONI hold start

Sleep Timer

Write SPn<CR> → [20] where 0=OFF, 1=15 min, 2=30 min, 3=60 min, 4=90 min, 5=120 min

Read SP<CR> → [20SPn] (returns selected bucket as shown)

Alarm / Timer Recording

Write

TRn TYt RPr RMxxxx TSssssssss TEeeeeeeee WEwww AGvv<CR> → [20]

- n:1–3 alarm number
- t:0–2 type (0=OFF, 1=Alarm, 2=TimerRec)
- r:0–1 repeat (0=Single, 1=Weekly)
- xxxx source type:
 - VFO: VFv (v=A/B/Z)
 - VFO Search: vs
 - Search: SSbb (bb:00–39)
 - MemCh: MRbbcc (bb:00–39, cc:00–49)
 - Scan: MSbb (bb:00–39)
- ssssssss start time depends on repeat: mmddhhnn for Single; hhnn for Weekly
- eeeeeeee end time: mmddhhnn (Single) / hhnn (Weekly)
- www:000–127 weekday bitmask (bit0:Sun ... bit6:Sat)
- vv:00–99 volume at start

Read TRn<CR> → same parameter set [20TRn TYt RPr RMxxxx TS... TE... WE... AG...]

Auto Frequency (Band) Control

Write AUn<CR> → [20] (0=OFF, 1=ON)

Read AU<CR> → [20AUn]

Auto Frequency Region

Write AZn<CR> → [20] (0=USA, 1=JAPAN, 2=EURO.) ****Read**** AZ→[20Zn]

Dial Operations

Write ZJ<CR> → main dial one step counter-clockwise → [20]

Write ZK<CR> → main dial one step clockwise → [20]

S-Meter Readout

Read LM<CR> → [20LMvvvq] where vv = level (-vv dB), q squelch state: 0=Close, 1=Open, 2=Open by CTCSS/DCR/VoiceSQ, 3=Open by LevelSQ/NoiseSQ

MemBank Delete

Write MBbb<CR> (bb:00–39) → [20]

Scan Group

Write MGnn DLddd FRnn BKAabb...<CR> → [20] (also selects that group)

Read MGn<CR> → [20MGnn DLddd FRnn BKAabb...]

MemCh Delete

Write MQbbcc<CR> (bb:00–39, cc:00–49) → [20]

Scan Start

Write MSbb<CR> (bb:00–39) → [20]

MemBank — Title & Protect

Write MWbb PTp TTtitle...<CR> → [20] (bb:00–39, p:0/1)

Read MWbb<CR> → [20MWbb PTp TTtitle...] (if bank empty, returns [30?])

Pass Frequency — Delete

Write

- All: PD%%<CR> → [20]
- Whole bank: PDbb<CR> (bb:00–39) → [20]
- One entry: PDbbii<CR> (bb:00–39, ii:00–49) → [20]

Pass Frequency — List

Read

- VFO Search list: PR<CR> → many [21PRii,ffff.ffff] then [20PRii,---]
- Search bank list: PRbb<CR> → [21PRbbii,ffff.ffff] then [20PRbbii,---]

Pass Frequency — Add

Write

- Current frequency into current bank: PW<CR> → [20]
- Current frequency into specific bank: PWbb<CR> (bb:00–39) → [20]; PW%%<CR> applies to all search banks
- Specific frequency into specific bank: PWbb.ffff.ffff<CR> → [20]; PW%%.ffff.ffff<CR> applies to all banks

Receiver Status

Read RX<CR> → response begins [20RX then, if active:

- AL_n (active alarm number), SP_n (sleep timer rough remaining), TR_n (timer-rec alarm number), SD PST_n (SD state)
Then one of the following blocks by mode:
 - **VFO**: VFv RFffff.ffff STsss.ss MDdan LMvvvq (v=A/B/Z)
 - **MemRead**: MRbbcc MPp RFffff.ffff STsss.ss MDdan LMvvvq TTtitle
 - **Scan**: MSbbcc RFffff.ffff STsss.ss MDdan LMvvvq TTtitle
 - **Search**: SRbb RFffff.ffff STsss.ss MDdan LMvvvq TTtitle
 - **VFO Search**: VS RFffff.ffff STsss.ss MDdan LMvvvq
 - **Cyber Search**: CYbb RFffff.ffff STsss.ss MDdan LMvvvq TTtitle
- Response terminates with <CR><LF>].
-

VFO Search Group (for VS)

Write VE DLdd FLff ASa<CR> → [20] (DL delay, FL free-time, AS auto-save)

Read VE<CR> → [20VE DLdd FLff ASa]

VFO Select

Write VFv<CR> → [20] (v=A,B,Z)

VFO Info (all three VFOs)

Read VI<CR> →

[21VI VFA RF... ST... SH... MD...]

[21VI VFB RF... ST... SH... MD...]

[20VI VFZ RF... ST... SH... MD...]

VFO Search Start

Write VS<CR> → [20]

Clock

Write DTyymmddhhnn<CR> → [20] (sets to 20yy/mm/dd hh:mm:00)

Read DT<CR> → [20DTyymmddhhnn]

Output RX state when squelch opens

Write LCn<CR> → [20] (1=emit RX state on squelch-open; 0=do not)

Read LC<CR> → [20LCn]

Periodic S-Meter output

Write LTnn<CR> → [20] (nn in 5-step: 00,05,...,95; 00 disables; otherwise period=nn×0.1s)

Read LT<CR> → [20LTnn]

Periodic RX state output

Write RTnn<CR> → [20] (same scale as LT)

Read RT<CR> → [20RTnn]

Priority On/Off

Write POn<CR> → [20] (0=OFF, 1=ON)

Read PO<CR> → [20POn]

Priority MemBank/MemCh

Write PPbbcc<CR> → [20]

Read PP<CR> → [20PPbbcc]

Priority Interval

Write TInn<CR> (00–99, seconds) → [20]

Read TI<CR> → [20TInn]

Priority Delay

Write TDnn<CR> (00–99, 0.1 s units) → [20]

Read TD<CR> → [20TDnn]

Manual Gain

Write RGnn<CR> (00–99) → [20]

Read RG<CR> → [20RGnnn]

Read Serial Number

Read RN<CR> → [20RNnnnnnnnn]

SerialSpeed (dummy)

Write SBn<CR> → [SB] (always OK)

Read SB<CR> → [SB0] (always returns 0)

Model Identifier

Read WI<CR> → [20AOR AR-DV3]

IF 12k AGC

Write IAn<CR> → [20] (0=MAN., 1=AGC)

Read IA<CR> → [20IAn]

IF 12k Manual Gain

Write IGnn<CR> (00–25) → [20]

Read IG<CR> → [20IGnn]

BACKUP File — Write (dump to SD)

Write SD MMW ssssssss<CR> → emits [21] at start and [20] at completion.

- ssssssss one of: SRCHBK (Search Bank), SRCHGRP (Search Group), MEMCH (Mem Ch.), MEMBK (Mem Bank), SCANGRP (Scan Group), UAUTO (User Auto), SYSTEM (ALL)
- File is created at /AR-DV3/yyyyMMdd/ssssssss.CSV. Names are fixed. Do not issue other commands during write.

BACKUP File — Read (load from SD)

Write SD MMR dddddddd/ssssssss<CR> → emits [21] then [20] when done. Loads /AR-DV3/dddddddd/ssssssss.CSV (extension fixed .CSV).