

REMRWA FUNCTION SET FOR RWA301

In the descriptions below parameter keywords are enclosed in brackets to show that they are optional as long as parameters are listed in the order specified. Enumerated alternatives are shown in braces like "{A|B}" (meaning "A" or "B"); F, I, and L represent float, 16-bit and 32-bit integer numericals, respectively, while "S" stands for a string. Defaults are shown on a separate line; where not shown, the defaults are not defined. If a command has any return value, an example of it is given on a separate line preceded by a ">".

Many state setting commands have associated queries, returning the setting currently in effect. This service is not supported by the RWA301; a table of current settings is maintained in memory by the software. Return values of queries, however, take into account the range and resolution of respective parameters inside RWA, and reflect values actually in effect. (THIS LAST STATEMENT IS NOT TRUE EVERYWHERE YET DUE TO IMPLEMENTATION PROBLEMS.) The name of the query function is usually the same as the name of the command, with a question mark appended. Where such query functions are supported, they are sometimes mentioned together with the corresponding command without further explanation.

I. DRIVER FUNCTIONS.

The functions in this section are accepted by Remote RWA software, passed on to the driver (S211, ESDI, etc., whichever is currently loaded), and cause the driver to send the appropriate execution commands to the drive (via RWA301, which relays them without modification). (Here and below the term "drive" is understood to include all types of drives and spinstands.) Some drives may not support some of the commands below; the commands are then ignored.

DRVNMB [val:]I

DRVNMB?

> 1

Select the drive to which the rest of the driver commands would go. (A driver may independently control several drives.)

MAXDRV?

> 1

Returns the maximum drive number known to the driver. (Drives are numbered from 1 up.)

START

STOP

Select/deselect a drive. On a spinstand, disk rotation starts, and the heads are loaded.

MOTOROFF

MOTORON

Stop / restart the disk in the drive. The heads are not unloaded / loaded.

DRESET

Reset an error condition (if any) on the drive.

READY?

> 1

Returns 0 if the drive is not ready.

ROTATION [dir:]{CW|CCW}

Default: dir: CW

ROTATION?

> CW

Disk rotation direction (clockwise/counterclockwise).

RPM [val:]F

Default: val: 3600.

RPM?

> 3600.

Set the disk rotational speed (1/min). Relative position of the read / write gates and the syncpoint are saved / restored across RPM changes. The query function gives out the target speed last set by the RPM command; unlike a direct measurement (see MEASRPM command), it would not reflect instability (if any) in the rotational speed.

REZERO

Seek track zero. Where available, use a special drive function to make the drive check heads position against its reference radius.

TRACK [val:]L**TRACK?**

> 100

Seek the specified track. If the current position is off the track center, the offset is retained, or reset to zero (the heads are positioned at the center of the new track), depending on the AUTOCENTER setting.

TRKIN**TRKOUT**

Move one track towards / away from the center of the disk.

TRKOFFS [ofs:]I

Default: ofs: 0

TRKOFFS?

> -2

Move to the specified offset position off the current track. I is the number of the offset position, which may be negative as well as positive; zero value corresponds to (and returns the heads to) the center of the track.

MAXOFFS?

> 5

A limit put by the drive on the range of offset values. (In this example, legal offset values range from -5 through 0 to +5.)

DIST AT OFFS? [ofs:]I

> 14.5

Returns the distance between the center of a track and the position of the heads at the specified offset off this track.

AUTOCENTER [mode:]{ON|OFF}

Default: mode:ON

AUTOCENTER?

> ON

Set AUTOCENTER mode. When the mode is on, each TRACK command positions the heads to the center of the specified track, and the offset value is reset to zero. When the mode is off, the offset is retained across TRACK commands, and the heads are, in general, positioned off the center of specified tracks; TRKOFFS 0 returns to track centers.

TRKSIZE [val:]FTRKSIZE?

> 800.0

Set track width (microinches). Even where variable track widths are supported, the resolution of the width is usually as good as the microstep size, however. The query may be used to determine the track size actually set.

MAXTRACK [val:]LMAXTRACK?

> 1200

Set the maximum track number (i.e., of the inner-most track).

RADID [val:]FRADID?

> 1.2

Set the ID radius (i.e., the inner-most track radius) for the drive (inches).

RAD AT TRACK? [trk:]L

Default: trk:0

> 1.65

Returns the radius of the specified track (inches), calculated on the basis of current TRKSIZE, MAXTRACK, and RADID values.

HEAD [val:]IHEAD?

> 0

Select the specified head for read/write operations.

MAXHEAD [val:]I**MAXHEAD?****> 1**

Set the maximum head number available on the drive (heads are numbered from zero up).

WRCURRENT [val:]F**Default: val:30.****WRCURRENT?****> 30.**

Set write current. (The same current is used for erase operations.)

DRVCMND [com:]S**DRVCMND? [com:]S****> abcde**

Send a driver-specific command to the driver. One ASCII string is passed directly to the driver. Use the second command when a reply string is expected.

DRVID?**> ESDI 1.03 XT-8000**

On this command the driver identifies itself.

DRVFEAT?**> 0**

A 32-bit long integer reflecting features supported by the driver and the drive. Bits in this value are defined as follows (a bit is asserted when the feature or ability is present, or the parameter may be set and changed by appropriate commands described in this section):

Bit 00 - direction of rotation,

Bit 01 - rotational speed,

Bit 02 - motor on/off

Bit 12 - write current (the commands setting this are described under Miscellaneous),

Bit 13 - needs RWA for operation,

Bit 16 - inner-most track diameter,

Bit 17 - maximum track number,

Bit 18 - maximum head number,

Bit 19 - track width,

Bit 24 - drive number,

Bit 25 - move to offset positions.

The remaining bits are reserved.

II. CLOCK SYSTEM.

PLOSRC [source:]{INT|EXT} [freq:]F

Default: source:INT freq:10.0

PLOSRC?

> INT 10.

Select a base timer for the PLO clock: internal or external. If external, F gives the frequency of the timer (MHz) (otherwise F is ignored).

PERIOD [time:]F

Default: time:100.

PERIOD?

> 100.

Select PLO clock period (ns). THIS IS ONE OF THE COMMANDS WHERE THE QUERY MAY NOT REFLECT THE ACTUAL SETTING EXACTLY, THOUGH A PRECISION OF 1E-04 IS GUARANTEED.

PERIOD RANGE?

> 10. 1000.

Range of period values supported by the PLO clock.

III. PATTERN GENERATION.

In this release, each pattern encoding scheme being used must be described by a file of a special format. The set of available encoding schemes is determined by the set of such files present in the directory from which the RemRWA application is run, the 3-character names of the schemes coinciding with the first characters of the file names. (The appropriate file is loaded into memory when the PATCODE command is issued.) This mechanism may be changed in the future.

The patterns themselves are represented by binary (beginning with '0b') or hexadecimal (optionally beginning with '0x') numbers, but are passed to the Interpreter as strings (and should be quoted if a confusion with decimal numbers is otherwise possible). Note that a binary representation is the only way to represent a pattern with a length not divisible by 4 bits.

PATTERN [pat:]"S" [slot:]I

Default: pat:"0xF" slot:0

Set a pattern with encoding, using the encoding scheme currently selected by PATCODE. I - slot number (0-6). "S" -- pattern value (hexadecimal or binary). A newly loaded pattern is automatically selected (see SELPAT command).

Future extensions: additional parameter [code:]{DFL|MFM|NRZ|R27|...}, specifying the encoding scheme directly (DFL is the default for the parameter and stands for the scheme currently selected by PATCODE). Also, slot range extension to 0-7.

PATNRZ [pat:] "S" [slot:] I**Default:** pat:"0x1000" slot:7

A temporary substitute for PATTERN pat:"S" slot:I code:NRZ. Only slot 7 is allowed in this command.

PATTERN? [slot:] I**Default:** slot:0

> "0xAA" NRZ

Get a pattern value before encoding and the scheme it had been encoded with. The pattern is returned as it had been entered, in hexadecimal or binary form. Slot range: 0-7.

PATENC? [slot:] I**Default:** slot:7

> "0b1010"

Not implemented.

Future extension: Get a pattern value after encoding. The result is always in binary form.

PATCODE [code:] {MFM|NRZ|R27|...}**Default:** code:MFM**PATCODE?**

> "MFM"

Set the default encoding scheme. The corresponding scheme description file must be present in the current directory. Pattern environment (i.e. pattern values in slots 0-7) after loading a new encoding scheme is undefined.

Future extension: Encoded patterns in all 8 slots will be preserved across changes in the default encoding scheme.

PATPRE [status:] {ENB|DSB}**Default:** status:ENB**PATPRE?**

> ENB

Enable / disable a 40-microsecond preamble at the beginning of sectors. The preamble is used to synchronize the data separator clock VFO.

BITMASK [pat:] "S" [slot:] I**Default:** pat:"0x00" [slot:] 0**BITMASK? [slot:] I****Default:** slot:0

> "0x00"

Set a bitmask for phase margin error detection. The command is similar to pattern setting, except the bit masks are never encoded. Slot range 0-7. The length of the bitmask must always be equal to that of the encoded pattern in the corresponding slot; a shorter bitmask is padded with zeroes on the right side; of a longer one, only the leading bits are used.

SELPAT [slot:]I

Default: slot:0

SELPAT?

> 1

Select a slot as current. The pattern loaded in the specified slot will be used in all write operations, and the bitmask loaded in the slot of the same number - in all phase margin error detection operations. The selection is changed by a PATTERN command, a PATCODE command, or another SELPAT command.

IV. INDEX, SECTOR, GATES.**IDXSKEW [time:]F**

Default: time:0

IDXSKEW?

> 5.

Select index pulses coming from the drive (so-called "hard index") as the source for index pulses in the system. F is the delay (microseconds) to be applied to the incoming pulses.

SECTORS [type:]{SOFT|HARDPOS|HARDNEG|OFF} [cnt:]I

Default: type:OFF cnt:2

SECTORS?

> HARD 2

Select the sectoring mode. SOFT selects soft sectoring, I is the number of (equal-sized) sectors maintained, I>0. HARDPOS / HARDNEG selects hard sectoring by positive / negative external pulses, I is the number of equal-sized sectors expected (read/write gates and the syncpoint may be displaced if external sectors are not of the expected size). OFF has the same effect as "SOFT cnt:1" (effectively, no sectoring is maintained), I is ignored. On query, "OFF 1" is always returned instead of "SOFT 1".

SECTORACT [mask:]"S"

Default: activate all sectors

SECTORACT?

> "+----+++"

THE FORMAT OF THIS COMMAND IS STILL NOT DECIDED, AND FOR THE TIME BEING IT IS NOT IMPLEMENTED IN ANY WAY.

SECTOR ACTIVATION MASK. "+" MEANS ACTIVATE THE SECTOR, "-" -- DEACTIVATE IT. IF THE STRING LENGTH IS LESS THAN THE NUMBER OF SECTORS, THE REMAINING SECTORS ARE ACTIVATED. IF IT IS GREATER THAN THE NUMBER OF SECTORS, EXCESS CHARACTERS ARE IGNORED. WHEN THE SECTORING MODE OR THE NUMBER OF SOFT SECTORS IS CHANGED, ALL SECTORS BECOME ENABLED.

WRGATE [start:]F1 [stop:]F2**Default: start:0. stop:1.****WRGATE?****> 0.1 0.987**

Select write gate. F1, F2 are the limits of the gate, specified as fractions of a sector, including the preamble, $0 \leq F1, F2 \leq 1$. The same fraction of a sector is covered even if the number of sectors and/or the rotational speed is changed. With hard sectoring, the gate may be misaligned if the sectors are not of the same length or their number is different from what had been specified in SECTORS.

RDGATE [start:]F1 [stop:]F2**Default: start:0. stop:1.****RDGATE?****> 0.1 0.995**

Select read gate. F1, F2 are the limits of the gate, specified as fractions of a sector excluding the preamble, $0 \leq F1, F2 \leq 1$. The same fraction of a sector is covered even if the number of sectors, the bit cell, and/or the rotational speed is changed. With hard sectoring, the gate may be misaligned if the sectors are not of the same length or their number is different from what had been specified in SECTORS. Also, stable and precise positioning of the read gate depends on good data synchronization.

SYNCPOINT [pos:]F [sect:]I**Default: pos:0. sect:0****SYNCPOINT?****> 0.1 2**

Select the syncpoint position inside the specified sector (sectors are counted from 0 up as they pass the head after the index pulse). F is specified as a fraction of a sector excluding the preamble, $0 \leq F \leq 1$. The same relative position is maintained even if the size of sectors, the bit cell, and/or the rotational speed is changed. With hard sectoring, the syncpoint may be misaligned if the sectors are not of the same length or their number is different from what had been specified in SECTORS. Also, a stable and precise positioning of the syncpoint depends on good data synchronization.

BYTE_TRK?**> 1000000**

The number of bytes per track (a long integer) is returned, calculated on the basis of the current rotational speed and bit cell values.

V. ANALOG READ CHANNEL.**FILTER [type:]{F1|F2|PARAM}****Default: type:PARAM****FILTER?****> "F1"**

Select the analog filter. The currently selected filter is used during all measurements and error detections, unless otherwise noted.

DIFFRNT [mode:]{HIGH|LOW}

Default: mode:HIGH

DIFFRNT?

> "HIGH"

Makes the selection between the two modes of the internal differentiator.

THRESH [val:]F

Default: val:50.

THRESH?

> 50.

Set the digitizer threshold in percents relative to of peak-detected envelope values. Threshold 100%-level calibration (CLBRTHR command) increases precision of the percentage setting.

TIMECONST [val:]F

Default: val:12.

TIMECONST?

> 12.

Set peak detector time constant, microseconds.

ENVSMP?

> 4 1000

The number of samples in the envelope is being counted. The second value returned is the total number of samples in the current read gate; the first value is the number of the first sample in this gate.

ENVVAL? [adr:]I [src:]{MEM1|MEM2}

Default: adr:10 src:MEM1

> 12.345

Get an envelope sample value of the specified number, in millivolts. Previous calibration is necessary (see CLBRATA).

DATASRC [mode:]{INT|EXT}

Default: mode:INT

DATASRC?

> INT

The digitized data used for error detection may come from outside RWA (or from the internal analog channel of filters, differentiator and digitizer).

VI. ERROR REGISTRATION SYSTEM.

The Error registration occurs in active sectors, inside read gates.

MRGTYPE [mode:]{EARLY|LATE|BOTH}

Default: mode:BOTH

MRGTYPE?

> EARLY

Select margin window format: early margin only, late margin only, or both. The corresponding calibration table is loaded from the disk; if no calibration (CLBRBSA command) for this window format had been done, such calibration is forced, with the resulting table saved in a disk file for future use. The margin window size is reset by this command.

MRGWIND [val:]{F}

Default: val:35.

MRGWIND?

> 12.

Set bit shift margin (nanoseconds). If both early and late margins are enabled, the total size of the window is 2*F. Actual DAC setting in RWA for each window size is determined during BSA calibration (CLBRBSA command). The window format (EARLY, LATE, or BOTH) must be selected before the window size to ensure use of the correct calibration table when setting the window. The resolution and range of the table limit the resolution and range of window setting, and this must be kept in mind especially if the bitcell size has been changed from the time of calibration; see CLBRBSA command description. The window setting will be based on the nearest calibrated window size in the table if the specified size does not coincide with any of the calibrated sizes, or is outside the range of the table altogether; in this latter case the error may be very significant. AT THE MOMENT, NO ERROR OR WARNING MESSAGE IS GENERATED WHEN THIS ERROR IS INTRODUCED, NOR IS IT POSSIBLE TO VERIFY THE ACTUAL SETTING BY A QUERY COMMAND (THE QUERY COMMAND ABOVE IS ANOTHER ONE OF THOSE REFLECTING THE LAST ATTEMPTED SETTING RATHER THAN THE ACTUAL ONE).

ERRORLOG [src:]{CMP|PHM|CMP&PHM|EXTRA} [oper:]{READ|WRITE|NOP}

Default: src:CMP&PHM oper:READ

> 56

Run a specified operation for one revolution and register errors of the specified type (Comparator, Phase Margin, Comparator and Phase Margin, Extra Pulse). The return value is the number of errors registered. Detailed information on the first 127 errors is available to ERRORINF? / ERRORNEXT? commands immediately following ERRORREG.

ERRORINF? [errnum:]{I}

Default: errnum:0

> 5 123 0x0100 0x0200 0x0000

Get Error File information on a registered error with the specified number. The information includes error location (sector number and byte number within the sector), error data (2 bytes), Comparator error flags, and Phase Margin / Extra Pulse error flags.

ERRORNEXT?

> 5 123 0x0100 0x0200 0x0000

A series of consecutive ERRORNEXT?'s after an ERRORINF? is a more efficient way of fetching information on consecutive errors. The information returned is the same as in ERRORINF?.

ERRORCNT [src:]{CMP|PHM|EXTRA} [rev:]I [oper:]{READ|WRITE|NOP}

Default: src:PHM rev:1 oper:READ

> 12.3

Count errors of the specified type while doing the specified operation for I revolutions. The return value is the average number of errors per revolution. Detailed information on errors would not be available through ERRORINF?.

VII. MISCELLANEOUS.

SCOPEPT1 [src:]{PLO2F|VFO2F|WRDATA|IRDATA|CMPERR|MRGERR|XTRERR}

SCOPEPT2 [src:]{PLO2F|VFO2F|WRDATA|IRDATA|CMPERR|MRGERR|XTRERR}

SCOPEPT3 [src:]{SCTR|WRGATE|RDGATE|ADRM}

Defaults: WRDATA for SCOPEPT1, VFO2F for SCOPEPT2, RDGATE for SCOPEPT3.

SCOPEPT1?

SCOPEPT2?

SCOPEPT3?

> WRDATA

Select signal sources for scope output points 1, 2, and 3. PLO and VFO clocks, write and digitized read data, error detection pulses, sector pulses, write and read gates, and the address mark signal are among the choices. See RWA301 Functional Overview for description of these signals.

INIT

Initialize RWA.

ONLINE?

> 1

Returns 0 if RWA is not online.

SERNMB?

> 1234567

Returns RWA serial number.

RWATYPE?

> RWA301

RWA type.

RWAFEAT?

> 0x0000

Returns an 16-bit RWA feature word, reflecting features supported in this particular model. Bits are not defined yet.

VIII. OPERATIONS.**READ** [revol:]I**WRITE** [revol:]I [cur:]F**ERASEP** [revol:]I [cur:]F**ERASEN** [revol:]I [cur:]F**WRTVFO** [revol:]I [cur:]F**Defaults:** revol:1 cur:0.

Read, Write, Erase Positive, Erase Negative, Write from VFO, for a specified number of revolutions.

The "cur" parameter for write operations is the write current. If specified other than 0, it overrides the value set by "WRCURRENT" command (but does not change this value). If "cur"=0, the value from WRCURRENT is used.

WRTFLUX [freq:]F [revol:]I**Default:** freq:1. revol:1

Writes the specified flux frequency (MHz) for the given number of revolutions. (Writing a flux frequency F means changing the magnetization direction on the media F times per second.)

IX. MEASUREMENTS.

Whenever in the command descriptions below it is noted that the results of one command depend on the previous execution of another, it must be assumed (unless otherwise stated) that no other measurement or test commands should interfere between invocations of these two. This is because measurement commands generally use and change both gain levels and envelope memories.

TAA [filter:]{PARAM|F1|F2|OVERWR|DFL} [mem:]{MEM1|MEM2}**Default:** filter:DFL mem:MEM1**> 2.345**

Measure Track Average Amplitude. The filter being used is specified as a parameter. DFL stands for the filter currently selected by the FILTER command; OVERWR selects the overwrite filter. The value is in millivolts only if the filter option is PARAM and amplitude calibration for current drive and head has been done (CLBRTAA command); otherwise the value has meaning only in relation to other measurement on the same drive, head, and filter. Envelope samples from the whole track are stored in the specified envelope memory, and the minimum / maximum samples are stored to be accessible to a subsequent MODUL command.

MODUL**> 23.34 12.43**

Calculate positive and negative modulation (<(max-average)/average> and <(min-average)/average>), based on the measurements left by the latest TAA command. Return values are in percents.

PW [thr:]F1 [tc:]F2

Default: thr:50. tc:0

> 23.3

Measure pulse width (nanoseconds) at the specified threshold (percent). Always uses the parametric filter. The threshold is always relative to the current signal envelope. Parameters set by the latest THRESH command are overridden but not changed. Threshold calibration is necessary (CLBRTHR). The second parameter, if specified as different from zero, overrides for the duration of the measurement the latest selected time constant value as set by TIMECONST (but does not change this value).

ASYM

> 1.7

Measure average asymmetry (difference between positive-negative and negative-positive interpulse intervals) in the read signal, nanoseconds. Always uses the parametric filter. Only pulses above the current threshold are taken into account.

RMSOR

> 12.23

Measure the RMS (Root Mean Square) of the read signal. The return value is in millivolts if the parametric filter is used and calibration has been done (CLBRTAA).

OVERWR

> 23.4

Used to conclude an overwrite test. This command:

1. Fetches from the RWA memory the result of the latest TAA measurement and calls it A.
2. Measures "TAA filter:OVERWR" and calls the result "B".
3. Divides A by B and returns the ratio in dB : $20 \cdot \lg(A/B)$.

For the result to be meaningful, the last TAA measurement must have used the overwrite filter ("TAA filter:OVERWR"). Also, no operations involving read cycles (including measurements) should interfere between this TAA measurement and the OVERWR command (gain level may be changed in such cycles, introducing an error in the result of OVERWR).

RESOL

> 2.5

Used to conclude a resolution test. Works exactly the same as the OVERWR command, except that the currently selected filter is used while measuring B and the same filter is expected to have been used while measuring A. Also, the result is in percent, not in dB.

SNR

> 23.4

Concludes a signal-to-noise test. The track must be all "noise" by the time this command is issued. The "signal" must have been measured before SNR using "TAA mem:MEM2". The RMS of the current read signal's ratio to the envelope of the signal measured by this preceding TAA command is counted by SNR. No measurement or test commands may interfere between TAA and related SNR. Also, SNR is measured with the currently selected filter, and the same filter is assumed to have been used during the related TAA measurement. The return value is in dB.

BSARATE [smplev:]I**Default: smplev:6****> -3.45**

Measure the margin error rate on the read signal (no new writing takes place). The bitmask associated with the currently selected pattern is used to mask the errors. The window format (EARLY, LATE, or BOTH) and the window size are used as selected by last MRGTYPE and MRGWINDOW commands. I determines the approximate number of bits to be tested: testing continues for a whole number of revolutions until at least $10^{**}I$ bits are tested. (Note that the actual number of bits tested may be only a fraction of this if some of the bits are masked by the bitmask in effect; also, the software may automatically reduce the number of revolutions if the error rate is high).

The return value is the logarithm of the error rate (the error rate is a ratio of the total number of errors to the number of unmasked bits tested). -100 is returned if the error rate is 0.

MEASRPM?**> 3599.**

Measure RPM. The measurement is performed by counting PLO clock ticks between two index pulses. The measurement is meaningless if soft indexing is in effect. The precision of the measurement is 0.5 rotations per minute. (A low-level COUNT command may be used to give greater precision.)

X. INTEGRATED MEASUREMENTS.

Certain standard test sequences are accessible through a single Remote RWA command. Using these commands provides convenience at the expense of flexibility of test sequences.

IPW50**> 23.3**

Integrated pulse width test. Erase, write a "0x10" (NRZ) pattern, and measure "PW" with a threshold at 50% and a time constant 8 times greater than the one currently selected by TIMECONST.

IASYM**> 1.7 2.5**

Integrated asymmetry test. Erase positive, write "0xAA" (NRZ), measure ASYM and call it A; erase negative, write "0xAA" (NRZ), measure ASYM and call it B; return A (positive asymmetry), and B (negative asymmetry), in this order.

IRESOL**> 33.**

Integrated resolution test. Erase, write "0xFF" (NRZ), measure "TAA filter:PARAM", erase, write "0xAA" (NRZ), measure "RESOL".

IOVERWR [freq:]F**Default: freq:1.****> 23.4**

Standard overwrite test. Erase, write the specified flux frequency F (MHz), measure "TAA OVERWR", write $2 * F$, and do "OVERWR".

ISNR [erdir:]{POS|NEG} [ermode:]{HI|HI&LO} [hicur:]F1 [locur:]F2 [freq:]F3

Default: erdir:POS ermode:HI hicur:0 locur:0 freq:1.

> 23.4

Standard signal-to-noise test:

- Write the specified flux frequency F (MHz). Measure "TAA filter:DFL mem:MEM2".
- Erase in the direction given by 'erdir' with the erase current F1 (mA), or with the currently selected current if F1=0.
- If ermode='HI&LO', do a second erase in the opposite direction with the erase current F2 (mA), or currently selected if F2=0.
- Measure signal-to-noise ("SNR" command).

IEXTRA [erdir:]{POS|NEG} [ermode:]{HI|HI&LO} [hicur:]F1 [locur:]F2 [thr:]F3 [slot:]I

Default: erdir:POS ermode:HI hicur:30 locur:30 thr:25. slot:7

> 56

Integrated extra pulse test:

- Erase positive.
- Write the pattern from the specified slot.
- Erase according to 'erdir', 'ermode', and currents, like in the signal-to-noise test.
- Set threshold F3 (percent), ENV as the threshold source.
- Count extra pulses ("ERRORLOG src:EXTRA oper:NOP").
- Restore original threshold and threshold source.

IBSAWIND [smplev:]I [winc:]F1 [fitttype:]{PARAB|LINEAR} [intbeg:]F2 [intend:]F3 [winc2:]F4 [bsalev:]F5 [crlim:]F6

Default: smplev:6 winc:2. fitttype:PARAB intbeg:1. intend:6. winc2:2. bsalev:9. crlim:0.99

> 23.3 20

BSA test. The test consists of measuring margin error rates (BSARATE command) at window sizes from 0ns, with a certain increment, to 0.5*bitcell or until the error rate is below detection at the given sampling level. The data table (error rate logarithms versus window sizes) is then approximated by a parabolic curve or a linear spline, and interpolation (in the case of a parabolic curve - also extrapolation) is used to project the window size corresponding to an error rate of $10^{**(-<bsalev>)}$. The test is done on the current read signal (no new writing occurs); the bitmask corresponding to the currently selected pattern masks the errors, and other current settings are also in effect (filter, threshold, data source, write current, bit cell, etc.). The meaning of parameters in the command is:

- smplev: Sampling level, determines the number of bit samples to be processed for each window setting (as in BSARATE command).
- winc, winc2: Window size increment (between measurement points) to be used outside and inside the interpolation range, respectively.
- fitttype: Type of interpolation fit applied to the data when it has been collected: Parabolic least squares fit or Linear spline.
- intbeg, intend: Interpolation range (on the log-rate axis). A parabolic fit is only applied to data inside the range, but the resulting curve may be extrapolated outside it.
- crlim: For parabolic least squares fit, the least acceptable correlation coefficient for the curve fitting (if the curve cannot be fitted as good as this, it is reflected in the completion code). Ignored for linear splines.
- bsalev: Once the fitting is completed, window size at an error rate of $10^{**(-<bsalev>)}$ is projected by inter- or extrapolation of the resulting curve

(interpolation only for linear splines). This projected window size is the result of the test.

The first return value is the projected window size at a log-rate level of ($-\langle bsalev \rangle$); the second value is an integer completion code. The codes for linear splines are:

- 0 - No error, linear spline completed.
- 10 - The first bit error log-rate observed to be below $\langle bsalev \rangle$ is less than $10^{**}(-9)$, and the slope is increasing. The slope of the previous pair of points is extrapolated to calculate the estimated bit shift.
- 11 - All points are above $\langle bsalev \rangle$, no interpolation projection to this level is possible.

The codes for parabolic least squares fits are:

- 0 - No error, fit completed.
- 20 - Less than three points in the interpolation range.
- 21 - Parabola concave instead of convex.
- 22 - Parabola does not intersect $\langle bsalev \rangle$ level.
- 23 - Parabola intersects $\langle bsalev \rangle$ level at a window greater than $0.5 * \text{bitcell}$.
- 24 - Correlation coefficient is less than required amount.

XI. CALIBRATION COMMANDS.

CLBRTAA [voltage:]F

Default: voltage:1.

Calibrate the threshold/envelope system. The calibration is done by applying a known voltage to the head connectors of the active head while issuing this command. If several heads are used, their calibration is independent. F is the voltage (in volts) being applied. The calibration is always done with the parametric filter, and this is therefore the only filter that may give calibrated absolute measurements. Relative measurements (where only a ratio of two values is important) do not require calibration, though.

CLBRTHR [freq:]F

Default: freq:1.

Perform internal calibration to increase precision of threshold settings. The specified flux frequency (MHz) is written and then read with different threshold settings to check the setting corresponding to a 100% threshold. The result should not be frequency-dependent inside the working range of frequencies.

CLBRBSA [mode:] {EARLY|LATE|BOTH}

Default: mode:BOTH

Perform internal calibration of the margin windows. Each of the three options above creates a separate calibration table, which is used for measurements when the corresponding margin window mode is selected by MRGTYPE. This command creates a table for the specified window format and writes it out into a file, from which it is loaded back when necessary as a part of the MRGTYPE command execution. The table contains 256 entries for calibrated window sizes; the range and resolution of the sizes are, respectively, $1/512$ through $511/512$, and $1/256$, of the bitcell size at the time of calibration. If the bitcell is widened, a recalibration would extend window size range; if the bitcell is narrowed, a recalibration would improve resolution. A recalibration is always recommended when switching to a different bitcell size.

XII. LOW LEVEL COMMANDS.**IDXSOFTE** [rpm:] F

Default: rpm:5000.

IDXSOFTE?

> 5000.

WRGTTIME [start:] F1 [stop:] F2

Default: start:0. stop:10000000.

WRGTTIME?

> 5. 10000.

Set write gate limits directly by time (nanoseconds) from a sector pulse. Only the relative position of the gate is maintained across changes in sectoring and/or rotational speed, however.

RDGTBYTE [start:] L1 [stop:] L2

Default: start:3 stop:100000000

RDGTBYTE?

> 3 100000

Set read gate limits directly by byte locations inside sectors. Only the relative position of the gate is maintained across changes in bitcell, sectoring, and/or rotational speed, however.

SYNCPTBYTE [pos:] L [sect:] I

Default: pos:0 sect:1

SYNCPTBYTE?

> 10 2

Set the syncpoint directly by byte location inside the sector. Only relative position of the pulse in the sector is maintained across changes in bitcell, sectoring, and/or rotational speed, however.

MGAINSTG [stage:] I

Default: stage:0

MGAINSTG?

> 0

Set main attenuator gain level. The range is 0 (most sensitive) to 5 (least sensitive), the step is 6dB.

OGAINSTG [stage:] I

Default: stage:3

Set gain level of the auxiliary overwrite attenuator. The range is 3 (most sensitive) to 0 (least sensitive), the step is 14dB.

THRSRC [src:]{ENV|MEM1|MEM2|ABS} [val:] F

Default: src:ENV val:50.

THRSRC?

> ENV 50.

Sets both the source of the digitizer threshold and the fraction (percents) of the signal at this source to be used as a threshold. Threshold source options include run-time peak-detected signal (ENV), samples

of a peak-detected signal which had been stored in either of the two envelope memories, and a separate software-set amplitude. With MEM1, MEM2, and ENV, F is the percentage of the source signal to be used as the digitizer threshold. Effective value of a memory-based threshold may change if current gain or filter is not the same as had been used when sampling into the memory. With ABS, the value F is not a percentage, but an absolute amplitude (millivolts); a set absolute value will only be valid, however, as long as the gain, filter, as well as drive and head selection, stay the same. The precision of absolute settings depends on TAA calibration for the particular head (CLBRTAA command); the precision of all settings depends on the 100%-level calibration (CLBRTHR command).

THRDIG [val:]I**THRDIG?**

> 228

Set threshold value directly in units of the DAC being used (the range is 0 through 255). The source of the threshold (Envelope, Memory1, Memory2, or Absolute) is not affected by this command and remains as last set by THRESH command.

THR100?

> 225

Return the value (in DAC units) used on the threshold generation DAC to achieve a 100% threshold. The value is originally determined by internal calibration (CLBRTHR command), and then stored in computer memory for use as a basis of all relative and absolute thresholds.

ENVDIG? [adr:]I [src:]{MEM1|MEM2}

Default: adr:10 src:MEM1

> 201

Fetch a value from the specified envelope memory. I is the sample number. The value is returned directly in DAC units (range 0-255).

MAXDIG?

> 200

MINDIG?

> 102

Fetch the minimum / maximum sample value from the latest envelope sampling operation (these values are normally used to calculate modulations). The values are returned directly in DAC units (range 0-255).

MRGDIG [val:]I**MRGDIG?**

> 1234

Direct access to the margin window setting DAC. The range of DAC values is 0 (the widest possible window) to 32767 (the narrowest possible window).

COUNT [src:]{WRD|PLO2F|CE|ME|XE|IRD|VFO2F|NCNT|OSC|ESCT|TAACNT|
[oper:]{READ|WRITE|ERSN|NOP} [rev:]I [gate:]{ON|OFF}

Default: src:PLO2F oper:NOP rev:1 gate:ON

> 35.4

Count the average number of pulses per revolution from the specified source, for I revolutions while doing the specified operation. See the description of the principal RWA301 source signals. Counting inside the read gate only or throughout the track may be selected.