

NATIONAL RADIO ASTRONOMY OBSERVATORY  
CHARLOTTESVILLE, VIRGINIA

ELECTRONICS DIVISION INTERNAL REPORT No. 232

FREQUENCY STANDARD TEST PROGRAM

S. WEINREB

JUNE 1983

NUMBER OF COPIES: 150

FREQUENCY STANDARD TEST PROGRAM

S. Weinreb

I.	Introduction . . . . .	3
II.	Test Configuration . . . . .	4
III.	Start-Up . . . . .	5
IV.	Program Function . . . . .	8
V.	Data Interruption . . . . .	11
VI.	Program Test . . . . .	12

Figures

Figure 1	Test Configuration . . . . .	4
Figure 2	Schematic of Temperature Sensor . . . . .	6
Figure 3	Example of Hard Copy at End of Data Batch . . . . .	10

Appendix

Appendix 1	Program Listing . . . . .	15
Reference . . . . .		14

## FREQUENCY STANDARD TEST PROGRAM

S. Weinreb

### I. Introduction

This report describes a program which computes the root Allan variance [1], SIGMA, as a function of time interval, TAU, which varies from 1 to 10,000 seconds in 1-2-5 steps. While taking data the program presents a video display of current samples of phase, frequency, and temperature and also plots vs. time SIGMA for TAU = 1, frequency, and temperature. BASIC language for the Apple II Plus computer is utilized. The test configuration will first be described and then be followed by descriptions of program start-up, function, and interrupt options. A listing of the program is in Appendix I.

### II. Test Configuration

The test configuration is shown in Figure 1. The two frequency standard outputs, usually at 5 MHz, are connected to an NRAO Precision Phase Comparator. This unit consists of a phase detector followed by a DC amplifier and also has a finely adjustable phase shifter in the input line. At the start of a test the phase shifter is adjusted to put the phase detector inputs in quadrature and output at null. The DC gain following the phase detector is selectable in decade steps so that full scale output of +10 volts is produced by time shifts of +10 ps, +100 ps, +1,000 ps, or +10,000 ps. For comparison of hydrogen masers the 1,000 ps scale is appropriate; thus, to stay on scale for  $10^4$  seconds the frequencies must be identical within 1 pp  $10^{13}$ .

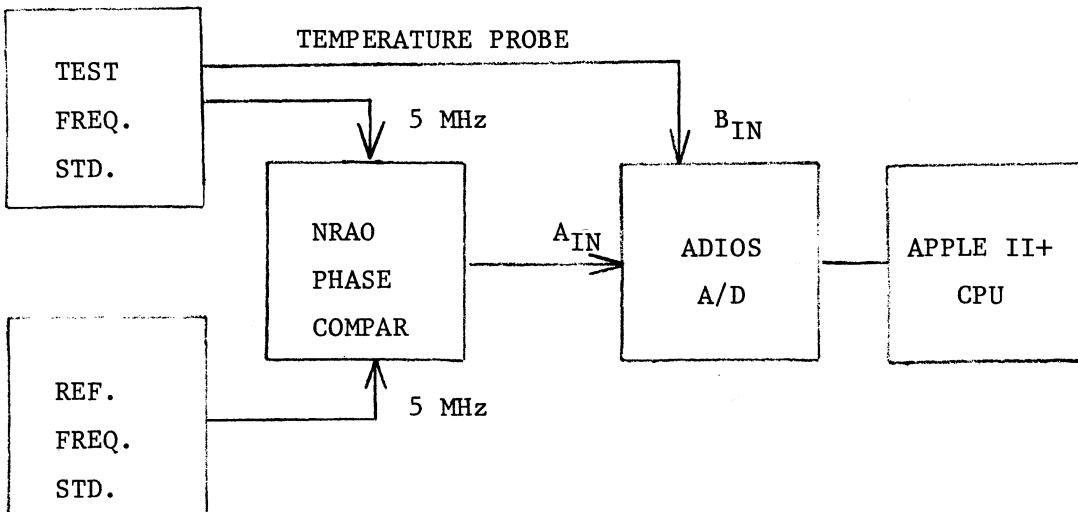


Fig. 1. Test configuration

The phase comparator output drives the A input of an ADIOS integrating A/D converter which is described in NRAO EDIR #212. The A/D conversion is accomplished with a precision 1 MHz voltage-to-frequency converter followed by a 32-bit counter.\* For this application the integration time or COUNT in the program is set at 160 ms followed by a dead time or BLANK of 840 ms. This process can be modeled as a 1 Hz low-pass filter followed by 1 Hz sampling. A second analog input channel, B, is driven by a temperature sensor with

---

\*The V/F output is 1 MHz with +10 volts input and 0 Hz at -10 volts input; inputs < -9 volts should be avoided to prevent counter resolution problems.

schematic shown in Figure 2. The toggle switches on the ADIOS front panel must be set at +/- and 10 volts for A and, for B, at + and 10 volts.

The required computer hardware is an Apple II Plus computer with 48k byte memory, one 5 $\frac{1}{4}$ " floppy disk (slot 6), CRT display, a Trendcom 200 thermal printer (slot 1) and California Computer Systems 7724 clock (slot 4). The program is stored on disk with the name FST along with required binary utility programs LIB 3.1 (for ADIOS service, see EDIR's #224 and #225), SHAPES (for plotting symbols), and HGR CHR GEN SHORT (for labeling plot). Also included on the disk are a text file, FST LOADER, which loads all of the above programs, and a program, SET TIME, to set the clock, if necessary.

### III. Start-Up

The program disk is inserted in the disk drive and AC power is turned on or IN#6 (CR)\* is typed. The disk catalog is displayed and then EXEC FST LOADER (CR) is typed to load FST and binary utilities and then run FST.

The following "edit parameters" screen will appear on the CRT:

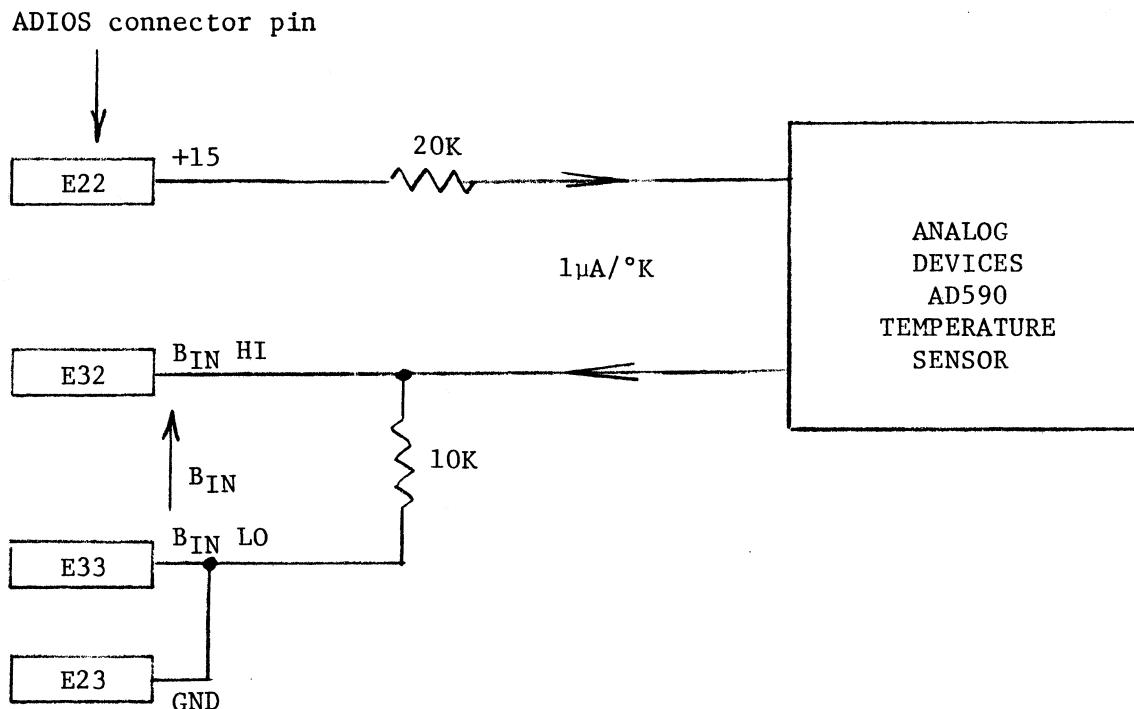
PRESENT PARAMETERS ARE:

- (1) PHASE DETECTOR OUTPUT, FULL SCALE, IN PS=1000
- (2) PERFECT (P) OR EQUIALLY (E) UNSTABLE REFERENCE STANDARD; NOW IS P
- (3) SECONDS PER DATA BATCH=100
- (4) VIEW BEFORE HARD COPY (V) OR CONTINUOUS DATA TAKING (C); NOW V
- (5) LOG SIGMA PLOT, MAX IS EXP -11
- (6) LIN DRIFT PLOT, MAX \*EXP-12 IS 100
- (7) TEMPERATURE PLOT SPAN IS 5C

CHANGE (1-7) OR (RETURN)?

---

\*Carriage return key.



$$B_{IN}(\text{mV}) = 10 * TE(\text{°K})$$

$$B_{IN}(\text{mV}) = 10 * (TE(\text{°C}) + 273.2)$$

$$IN(1) (\text{COUNTS}) = .100 * \text{MODE\%}(2) * B_{IN}(\text{mV})$$

$$\text{Thus, } TE(\text{°C}) = CK * IN(1) - CN$$

$$\text{where } CK = 1 / \text{MODE\%}(2) \quad CN = 273.2$$

Fig. 2. Schematic of temperature sensor and scaling equations to relate program variable  $TE(\text{°C})$  to counts,  $IN(1)$ , produced by temperature sensor.

To change a parameter, type the number (1 thru 7) of the parameter and a new value of the parameter will then be requested on the screen. The new value and the return key are then typed; the process can be repeated as often as desired. The editing is terminated by typing the return key instead of a number; data taking will then begin.

The meaning of the seven program parameters is as follows:

(1) Phase detector full scale - This is the number of ps time shift of one standard relative to the other which will cause a 10 volt change in the phase comparator output.

(2) Perfect or equal reference stability - If P is typed, it is assumed that the reference standard has perfect stability and all instability is assigned to the standard under test. If E is typed, all output rms deviations are divided by  $\sqrt{2}$  so that they apply to one of two equally unstable standards.

(3) Duration of data batches, KM - The data taken in batches of length KM seconds. After each batch statistics are viewed on the CRT or printed on the printer. Data taking stops during this viewing and printing time.

(4) View before hard copy or continuous data taking - If V is chosen, after each batch is complete statistics will be presented on the CRT and a choice may be made as to whether to hard copy or not - an operator must be present in order for data taking to continue. If C is chosen, statistics are hard copied and a new batch starts without operator intervention.

(5) Log SIGMA plot maximum - During each batch the 1 second SIGMA is plotted (  $\square$  symbol) vs. time; after the batch SIGMA vs. TAU is plotted. The abscissa of both plots cover 5 decades with maximum value entered here (i.e.,  $10^{-11}$  to  $10^{-16}$  if the parameter is -11).

(6) Linear drift plot scale, DM - During each batch, frequency averaged over  $\sim$ KM/100 seconds is also plotted (+ symbol) with linear scale of  $+ DM * 10^{-12}$  to  $- DM * 10^{-12}$ .

(7) Temperature plot span, TS - During each batch, temperature vs. time is plotted (solid line) with a total span of TS centered upon the temperature at batch-start time.

#### IV. Program Function

The program samples, once per second, the difference in phase of the two frequency standards; the current sample is denoted as  $P\emptyset$  and a sample taken  $TAU(L)$  seconds back in time is labeled  $P1(L)$ . Since the computed frequencies must be normalized to the phase comparison frequency,  $f_0$ , we divide each sample phase (in radians) by  $2\pi f_0$  in the process of scaling the ADIOS A/D output. This normalization puts the phase samples in units of time; picoseconds, ps, are used as units. Thus, the fractional frequency,  $F\emptyset$ , (i.e.,  $\Delta f/f$ ) over the time interval  $TAU(L)$  is given by

$$F\emptyset(L) = \frac{P\emptyset - P1(L)}{TAU(L)} \quad (1)$$

in units of  $10^{-12}$ . A statistical estimate of the Allan variance,  $\sigma^2(L)$  as given in NBS Monograph 140, Eq. 8.13A, is then one-half of the mean square of differences of successive values of  $F\emptyset$

$$\sigma^2(L) = \frac{1}{2N} \sum_{k=1}^N [F\emptyset(L,k) - F\emptyset(L,k-1)]^2 \quad (2)$$

where the index k is the sample number.

The program computes  $\sigma^2(L)$  for  $TAU(L)$  running from 1 to 10,000 seconds in a 1-2-5 sequence. The samples used for a particular  $TAU(L)$  are selected so that the computation load is spread out among all the samples as shown in the table below:

L	TAU(L)	SAMPLES UTILIZED
1	1	1, 2, 3, 4, ...
2	2	2, 4, 6, 8, ...
3	5	5, 10, 15,
4	10	1, 11, 21,
5	20	6, 26, 46,
6	50	5, 55, 105,
7	100	7, 107, 207,
8	200	1, 201, 401,
9	500	2, 502, 1002,
10	1,000	8, 1008, 2008,
11	2,000	4, 2004, 4004,
12	5,000	7, 5007, 10007,
13	10,000	9, 10009, 20009,

While taking data, the CRT displays the values of every other phase sample, two-second frequency,  $F\emptyset(2)$ , and probe temperature. A plot of three (3) variables versus time is also presented on the CRT; the time axis extends to the batch duration, KM. Points are plotted every DP seconds where DP is the largest value of  $TAU(L)$  less than  $KM/100$ . The three plotted variables are the 1 second root Allan variance estimate based upon DP samples, the fractional frequency,  $F\emptyset$ , over the time interval DP, and the probe temperature, TE. An example of a completed time plot is shown as the upper graph of Figure 3.

# TESTS OF NRAO 5 MHZ PHASE COMPARATOR

FST PROGRAM OF 01/14/83  
 BATCH START 04:57.9 02/03/83  
 BATCH END 08:57.9 02/03/83  
 RUN START 08:40.9 02/02/83

PERFECT REFERENCE ASSUMED  
 TEMP=29.07C PHASE=13.4PS  
 TEMP=29.01C PHASE=13.4PS  
 TLOW=26.69C, THIGH=29.05C

TAU SEC	LAST SAMPLES	SIGMA *EXP-15	CUMULATIVE		TIME ERROR PS
			SAMPLES	SIGMA *EXP-15	
1	14398	31.2	86388	33.6	0
2	7198	19.3	43188	17.0	0
5	2878	10.3	17268	8.7	0
10	1438	6.3	8628	5.1	.1
20	718	4.0	4308	3.1	.1
50	286	2.0	1716	1.5	.1
100	142	1.2	852	0.8	.1
200	70	0.5	420	0.4	.1
500	29	0.2	172	0.2	.1
1000	15	0.1	88	0.2	.2
2000	8	0.1	46	0.2	.4
5000	3	0.0	16	0.1	.6
10000	2	0.0	10	0.0	.4

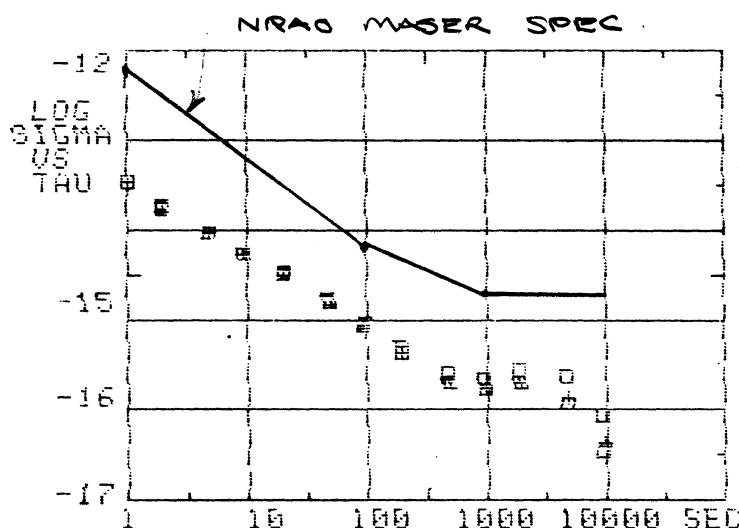
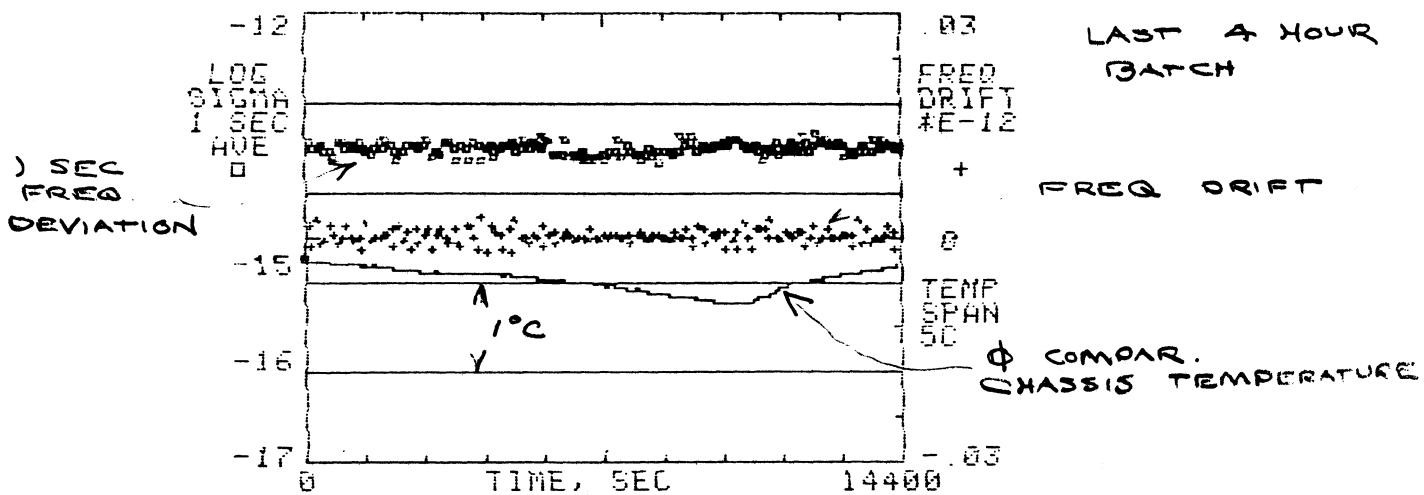


Fig. 3. Example of hard copy at end of data batch.

At the end of a data batch, all of the root Allan variance values are tabulated and plotted as shown in Figure 3. The batch values are also averaged with previously taken data (until a new run is started by typing RUN (CR)). On the plot of root-Allan variance, SIGMA vs. TAU, the cumulative average is plotted with a + symbol and the rms limits of  $SIGMA \times (1 \pm 1/\sqrt{N})$  are plotted with  $\square$  symbols to give a confidence range of the SIGMA value.

#### V. Data Interruption

The display and optional hard copy at the end of a batch interrupt the data taking sequence. Other interruptions can be produced by typing keys B, E, P, F, or H during data taking. The consequences and uses of the key interrupts will be discussed here.

An interruption of the program could cause the time duration between phase samples to be  $> TAU(L)$  and also cause the mean square frequency differences in the Allan variance to have some "dead time" between time intervals. If the interruption is  $< TAU(L)$  both effects will be negligible. The batch-end interruption lasts for  $\sim 200$  seconds and thus the last phase and frequency samples for  $TAU(L) < 500$  are discarded. The phase and frequency samples for  $TAU = 500, 1000, 2000, 5000$ , and  $10,000$  are kept as it would be wasteful of time and unnecessary to discard them.

The key interrupts have the following effects:

B - Batch end. Causes an immediate end to the batch with tables, plots, and resets as in a normal batch-end.

E - Edit parameters. Jumps to the edit screen as described in III. After parameters are edited, frequency and phase are reset as for batch-end and the batch continues.

P - Phase adjustment. Data taking stops and phase-comparator phase may be adjusted. All phase samples are discarded and the batch resumes when (CR) is typed.

F - Frequency adjustment. Data taking stops and phase and frequency may be adjusted. All phase and frequency samples are discarded and the batch resumes when (CR) is typed.

H - Halt. Program execution stops. Type RUN (CR) to make a fresh start - all variables erased and default parameters installed. GOTO 2600 will have same effect as B. GOTO 2100 will start a new batch but does not reset phase and frequency.

## VI. Program Test

The program can be checked by generating fictitious phase samples using the pseudo-random number generator command, RND, included in Apple BASIC. This is accomplished by removing the REM (remark) word in program lines 2176 and 2247. The program will then input from lines 4200 and 4210 phase and temperature samples with uniform probability from 100 to 110 ps and 25° to 26°C respectively.

The theoretical value for the root-Allan variance can be computed as follows. An estimate of SIGMA in terms of phase samples, P(k), is:

$$(\text{SIGMA})^2 = \frac{1}{2N} \sum_{k=1}^{N-1} \left[ \frac{P(k+L) - 2P(k) + P(k-L)}{L} \right]^2 \quad (3)$$

where L is the time difference between phase samples. The statistical average of  $(\text{SIGMA})^2$  can then be expressed in terms of the autocorrelation function, R(k), of the samples,

$$\overline{(\text{SIGMA})^2} = \frac{3R(0) - 4R(L) + R(2L)}{L^2} \quad (4)$$

In the case of independent samples (i.e., white phase noise),  $R(L) = 0 = R(2L)$  and  $R(0)$  is equal to the variance of a phase sample. For the uniform distribution of 100 to 110 ps  $R(0) = 8.333 \text{ (ps)}^2$  and finally,

$$\text{SIGMA} = 5/L \quad (5)$$

in units of  $10^{-12}$  or 5000/L in units of  $10^{-15}$  tabulated by the program. A long test of the program gave the following result which is within statistical limits of 5000/L.

TAU SEC	LAST BATCH		CUMULATIVE		TIME ERROR PS
	SAMPLES	SIGMA *EXP-15	SAMPLES	SIGMA *EXP-15	
1	49998	4985.8	349986	4993.6	5
2	24998	2506.9	174986	2512.9	5
5	9998	998.6	69986	1005.4	5
10	4998	495.9	34986	500.0	5
20	2498	247.8	17486	250.0	5
50	998	100.4	6986	102.9	5.1
100	498	46.8	3486	49.2	4.9
200	248	23.8	1736	24.8	5
500	100	9.5	698	9.5	4.7
1000	50	4.5	348	5.0	5
2000	25	2.4	173	2.5	5
5000	10	1.0	68	1.1	5.5
10000	5	0.7	33	0.5	4.9

REFERENCE

- [1] NBS Monograph 140, "Time and Frequency - Theory and Fundamentals,"  
National Bureau of Standards, Boulder, CO, pp. 156-157, 181-182.

```

01/17/83 15:52:4
PROGRAM LENGTH= 13472 BYTES      VARIABLES= 174 BYTES
FREE MEMORY= 6833 BYTES
START=16385 LOMEM=23857 FREE=23848 STR1NE=36781 HI MEM=36864

10 REM FST - FREQUENCY STANDARD TEST - PROGRAM. COMPUTES ROOT ALLAN
11 REM VARIANCE AS FUNCTION OF AVERAGING TIME. REFERENCE NES HANDBOOK #148;
12 REM IN EQUATION 8-9 NOTED.
20 REM WHILE PROGRAM IS TAKING DATA KEYBOARD RESPONDS TO (H) HALT,(B)
END BATCH,(E) EDIT PARAMETERS,(P) ADJUST PHASE. AND (F) ADJUST FREQUENCY.

30 REM FOR PROGRAM TESTS. EDIT LINES 2176,2247, AND 4816
90 HGR : HOME : SPEED= 255: HOLOR= 3
100 NMS = "FST PROGRAM OF 01/14/83"
105 PRINT NMS
110 PRINT : PRINT "REQUIRES LIB 3.1.SHAPES, HGR CHR GEN SHORT IN MEMORY"
115 REM L IS AN INDEX WHICH DETERMINES WHICH AVERAGING TIME,TAKL
120 REM KEY VARIABLES FOLLOW
125 REM L,1 IS THE MOST RECENT PHASE SAMPLE
130 REM P1(L) IS A PHASE SAMPLE TAKEN TAKL UNITS BACK IN TIME
135 REM F1(L) IS MOST RECENT FREQUENCY COMPUTED FROM PB AND P1(L)
140 REM PB IS MOST RECENT FREQUENCY COMPUTED FROM PB AND P1(L)
145 REM F1(L) IS A FREQUENCY COMPUTATION TAKL UNITS BACK IN TIME
150 REM F1(L) IS A FREQUENCY COMPUTATION TAKL UNITS BACK IN TIME
155 REM F1(L) IS A FREQUENCY COMPUTATION TAKL UNITS BACK IN TIME
160 REM K IS A RUNNING SAMPLE INDEX
165 REM K(L) IS THE NEXT DESIRED SAMPLE
170 REM KPL(L) IS A FLAG WHICH IS 0 UNTIL THE FIRST P1(L) IS VALID.
175 REM KDL(L) IS THE NUMBER OF SQUARED FREQUENCY DIFFERENCES,00, IN
180 REM THE SUM SD(L)
185 REM GDL(L) IS THE GRAND SUM OF SD(L) FREQUENCY DIFFERENCES
190 REM TE IS THE MEASURED TEMPERATURE
195 DIM TAKL(13),PK(13),KF(13),KK(13),SDX(13),KX(13),NO
200 C1=.70707: REM C34C2 MULTIPLIES RMS FREQUENCY DIFFERENCE
205 DATA 1.0,2.0,5.0,10.1,26.6,56.166,.7,268,1,500,2,1600,8,2000,4,3000
210 REM C1 = 1.00000*9
305 RESTORE
310 REM POINT PLOTTED AT K=2,340P,3+240P,ETC.
340 REM C1 IS THE NUMBER OF INPUT PER PS OF TIME ERROR
355 REM C2=9.76? IF RMS VALUES ARE TO APPLY TO ONE OF TWO IDENTICAL
STANDARDS.C2=1 IF REFERENCE IS PERFECT, SEE 5150.
360 C3 = .70707: REM C34C2 MULTIPLIES RMS FREQUENCY DIFFERENCE
400 REM KK IS NUMBER OF SAMPLES BEFORE PRINT OUT
435 CH = 10000:CY = 21714
440 CR = 10:CB = .5:EP = 1E-18
450 DEF FN R1(X) = INT ((CH * X + CB) / CR)
460 DEF FN R2(X) = INT ((CC * X + CB) / CC)
510 POKE 232,191: REM SHAPES IN #8127,L24
512 REM ENTRY POINTS FOR ADIOS INIT AND SERVICE
600 AI = 5126:BS = 840:MODEX(2) = 160: REM BLANK AND COUNT TIMES IN MS
620 CS = 26 : MODEX(2):DQ = 10000: REM SCALING CONSTANTS FOR AD105
IN +/10 VOLT MODE
630 CK = 1 / MODEX(2):CN = 273.2: REM SCALING CONSTANTS FOR AD598
TEMP PROBE IN BIN WITH +10 VOLT ADIOS SCALE
650 CALL AI,MODE(2),OUTZ(0):REM INITIALIZIZE ADIOS; SEE EDIR #224
889 REM INITIAL PARAMETERS NEXT

```

5 GOSUB 50000: REM EDIT PARAMETERS
1010 GOSUB 48000:TRS = "RUN START "+ TMS + DT\$  
2100 REM PREPARE FOR TAKING DATA BATCH
2105 POKE 34,21: REM TEXT BELOW GRAPHICS
2110 FOR L = 1 TO 13: READ TAKL,KDL)=0:NEXT L
2125 RESTORE
2130 XL = 0:YH = KH:YH = SH:YL = SH - 5
2135 XC\$ = "TIME, SEC"
2140 GOSUB 10000: GOSUB 10200:PO = 6:GP = 3: REM ERASE AND INIT TIME
PLOT
2150 GOSUB 46000:TMS = "BATCH START "+ TMS + DT\$  
2160 TL = 99:TH = 00: REM INITIAL LOH AND HIGH TEMPS
2170 CALL HS,INK 0),OUTZ(0)
2172 PB = CU \* INK 0) - CH
2175 TE = CK \* INK 1) - CN
2176 REM GOSUB 4200
2177 TE = TE + .55 \* TS:WD = TE -.45 \* TS: REM TEMP PLOT LIMITS
2178 YC = (Y9 - Y8) / (YC - Y0):DK = Y8 - DJ \* YD:YJ = DJ \* TE + DK:  
2180 REM MAIN MEASUREMENT LOOP FOLLOWS
2185 FOR K = 0 TO KH
2190 FOR K = 0 TO KH
2195 REM TEMP PLOT PARAMETERS
2200 FOR K = 0 TO KH
2205 FOR K = 0 TO KH
2210 IF K = KH THEN GOSUB 3400: GOTO 2100
2215 IR = PEEK (-16384):PUKE - 16388.0
2220 REM NEXT IF DATA POINT IS NOT USED AT THIS L
2225 CALL HS,INK 0),OUTZ(0)
2230 PB = CU \* INK 0) - CH
2240 REM MAIN MEASUREMENT LOOP FOLLOWS
2245 PB = CU \* INK 0) - CH
2247 REM GOSUB 4200: REM
2248 IF TE < TL THEN TL = TE
2250 IF TE > TH THEN TH = TE
2255 REM NEXT IF F0 IS FIRST FREQUENCY MEASUREMENT
2260 IF K = GP THEN GOSUB 2800: REM PLOT
2265 FOR L = 1 TO 13
2270 REM NEXT IF FIRST POINT AT THIS L
2275 IF KPL(L) = 0 THEN PK(L) = PB:KPL(L) = 1: NEXT L: NEXT K
2280 PK(L) = (PB - P1(L)) / THKL
2285 REM SAMPLES HAVE BEEN TAKEN
2290 IF L = 2 THEN UTAB 24: PRINT K;" PHASE="; FN R1(PB); TAB( 17);"  
FREQ="; FN R1(F0); TAB( 30); "TEM="; FN R2(TE),
2295 REM NEXT IF F0 IS FIRST FREQUENCY MEASUREMENT
2300 IF KF(L) = 0 THEN F1(L) = F0:KF(L) = 1: NEXT L: NEXT K
2305 IF PK(L) = 0 THEN PK(L) = PB:KDL(L) = KDL(L) + 1
2310 REM NEXT IF K < KDL THEN NEXT L: NEXT K
2315 K(L) = (K(L) + TAKL)
2320 REM NEXT IF FIRST POINT AT THIS L
2325 IF KPL(L) = 0 THEN PK(L) = PB:KPL(L) = 1: NEXT L: NEXT K
2330 PK(L) = (PB - P1(L)) / THKL
2335 REM SAMPLES HAVE BEEN TAKEN
2340 IF L = 2 THEN UTAB 24: PRINT K;" PHASE="; FN R1(PB); TAB( 17);"  
FREQ="; FN R1(F0); TAB( 30); "TEM="; FN R2(TE),
2345 REM NEXT IF F0 IS FIRST FREQUENCY MEASUREMENT
2350 IF KF(L) = 0 THEN F1(L) = F0:KF(L) = 1: NEXT L: NEXT K
2355 IF PK(L) = 0 THEN PK(L) = PB:KDL(L) = KDL(L) + 1
2360 PK(L) = (PB - P1(L)) / THKL
2365 REM SAMPLES HAVE BEEN TAKEN
2370 PK(L) = PB
2375 REM PLOT SIGNA US TIME
2380 REM PLOT SIGNA US TIME
2385 X = SOK(1) - PO: IF X < EP THEN VP = YL: GOTO 2920
2390 VP = C2 + CY \* L06 (X / DP),
2395 VP = K: GOSUB 10400
2400 IF KF(L) = 0 THEN F1(L) = F0:KF(L) = 1: NEXT L: NEXT K
2405 IF PK(L) = 0 THEN PK(L) = PB:KDL(L) = KDL(L) + 1
2410 PK(L) = (PB - P1(L)) / THKL
2415 REM SAMPLES HAVE BEEN TAKEN
2420 GOSUB 3400
2425 REM PLOT SIGNA US TIME
2430 REM PLOT SIGNA US TIME
2435 VP = C2 + CY \* L06 (X / DP),
2440 VP = F1(L):GOSUB 10500
2445 IF K < DP THEN RETURN
2450 VP = DU \* TE + DK
2455 IF YK < YS THEN YK = YS
2460 IF YK > YB THEN YK = YB
2465 REM INITIAL PARAMETERS NEXT

```

2957 XJ = 08 * (K - DP) + 09*XK = 08 * K + 09
2958 HPLOT XJ,YJ TO XK,YK
2959 YJ = YK: RETURN
3496 REM PROCESSING AFTER BATCH COMPLETE
3497 GOSUB 48680:TES = "BATCH END" + TMS + 0T$*
3498 CALL - 1659: CALL - 1659: REM BEEP
3499 T7 = TE:P7 = P0: REM BATCH END TEMP AND PHASE
3518 FOR L = 1 TO 13
3429 NDCL = NDCL + KDL
3430 KDL = 6DL + SDL
3435 NEXT L
3449 REM NEXT STARTS NEW P1 AND F1 SAMPLES FOR TAUL<500
3450 FOR L = 1 TO 8
3455 KFL = 0:KFL = 0
3458 NEXT L
3479 IF DS = "C" THEN GOSUB 5680: GOSUB 5880: GOSUB 3900: GOSUB 5880:
GOSUB 5400: RETURN : REM HARD COPY OF TABLE, TIME PLOT, AND SIGMA
PLOT
3480 GOSUB 3600: REM UTEM TABLE
3481 PRINT "HARD COPY (Y OR N)? "; GET Z$: PRINT Z$: HOME : POKE
3482 3548 IF Z$ = "Y" THEN GOSUB 5680
3549 TEXT : HOME
3550 TMS = 25: GOSUB 4326
3551 PRINT "HARD COPY (Y OR N)? "; GET Z$: PRINT Z$
3552 IF Z$ = "Y" THEN GOSUB 5880
3553 PRINT "SIGMA PLOT"
3554 GOSUB 3600: REM SIGMA PLOT
3555 PRINT "HARD COPY (Y OR N)? "; GET Z$: PRINT Z$
3556 IF Z$ = "Y" THEN GOSUB 5880
3565 IBS = 25: GOSUB 4326
3578 G010 2100: REM START NEW BATCH
3688 REM UTEM TABLE
3618 TEXT : HOME
3620 PRINT TSS: PRINT TRS
3621 PRINT "TEMPERATURE: LOH"; FN R2(XTL); "C, HIGH"; FN R2(XTH); "C"
3622 PRINT "START"; FN R2(XT6); "C"; FN R1(P6); "PS END"; FN R2(XT7); "C";
FN R1(P7); "PS"
3630 PRINT : PRINT TAB(10); "LAST BATCH"; TAB(2); "CUMULATIVE"
3631 PRINT : PRINT TAB(8); "SAMPLES"; TAB(17); "SIGMA"; TAB(24); "SAMPLES"
3632 PRINT : TAB(33); "SIGMA"
3650 PRINT "SEC"; TAB(16); "*EXP-15"; TAB(32); "EXP-15"
3668 FOR L = 1 TO 13
3678 PRINT TAUL; TAB(8); KDL; TAB(15);
3688 X = C4 * SQR (SDCL) / (NDCL + EP)
3685 WI = 7:01 = 1: GOSUB 11000
3690 PRINT TAB(24);NDCL; TAB(32);
3695 X = C4 * SQR (GOAL) / (NDCL + EP)
3698 DI = 1:HI = ?: GOSUB 11000
3700 PRINT : NEXT L: PRINT
3719 REM SIGMA US TAUL PLOT
3900 XL = 8:YH = 5:YL = SH - 5:YH = SH
3901 GOSUB 10000: GOSUB 10700: REM ERASE AND INIT PLOT
3938 FOR L = 1 TO 13
3935 IF NDCL < 1 THEN 4828
3940 XP = 2 * CY * LDG (TAUL)
3956 YP = C2 + CY * LOG (GOAL) / NDCL
3964 SCALE = 2: SH = 1: GOSUB 18400
3978 YP = YP + LD6 (1 + 1 / SQR (NDCL))
3980 SH = 2: GOSUB 18400
3985 IF NDCL = 1 THEN VP = SH - 5 + EP: GOSUB 4600
3988 VP = YP + LD6 (1 - 1 / SQR (NDCL)), SH = 2: GOSUB 18400
4020 NEXT L
4030 SCALE = 1: RETURN
4036 REM SHOW GRAPHICS
4110 POKE - 16394,0: POKE - 16390,0: POKE - 16297,0: POKE - 16391,0:
HJULOR= 3: UTAB 21: HTAB 1
4120 RETURN
4200 PB = 100 + 10 * RND (K)
4210 TE = 25 + RND (1)
4220 RETURN
4300 REM KEYBOARD INTERRUPT SERVICE
4305 IR = IR - 128
4310 IRS = CHR$ ("IR")
4315 IRS = "P" THEN GOSUB 4400
4320 IF IRS = "F" THEN GOSUB 4500
4325 IF IRS = "E" THEN GOSUB 4600
4330 IF IRS = "B" THEN 2680: REM BATCH END
4340 IF IRS = "H" THEN GOSUB 4700
4350 RETURN
4355 REM ADJUST PHASE
4400 PRINT "ADJUST PHASE AND PRESS (RETURN)": GET Z$: PRINT Z$
4410 FOR L = 1 TO 13:KFL) = 0: NEXT L
4420 RETURN
4430 REM ADJUST FREQUENCY
4435 PRINT "ADJUST FREQUENCY AND PRESS (RETURN)": GET Z$: PRINT Z$
4440 FOR L = 1 TO 13:KFL) = 0:KFL) = 0: NEXT L
4450 RETURN
4455 REM EDIT PARAMETERS
4460 GOSUB 5000
4465 FOR L = 1 TO 8
4470 KFL = 0:KFL) = 0: NEXT L
4475 REM ADOBE RESET PHASE AND FREE SAMPLES FOR TRU <500. APPROPRIATE
4480 FOR 68 SEC PHASE.
4485 XH = KI.YH = SH:YL = SH - 5: GOSUB 102000
4490 RETURN
4495 REM HALT DATA TAKING
4510 TEXT : HOME
4515 PRINT "HALT DATA TAKING": PRINT
4520 PRINT "(RUN) NEW RUN- FRESH START."
4525 PRINT "(G010 2100) NEW BATCH."
4530 PRINT "(G010 2800) OUTPUT DATA."
4535 END
4540 SETS TS TO TIME
4545 REM REMOVE RETURN IF YOU HAVE A CLOCK
4550 YRS = "83"
4555 PRINT
4560 DS = CHR$ (4)
4565 PRINT DS;"IN84"
4570 REM PRINT DS;"PRE4": REM THIS IS ONLY FOR MOUNTAIN HARDWARE
4575 CLOCK
4580 INPUT " ";TS
4585 GOSUB 1013
4590 DT$ = LEFTS (TS,5) + "/" + YRS + "
4595 SC$ = MID$ (TS,13,2)
4600 EH = INT ( VAL (SC$) / 6)
4610 EMS = STRS (EH)
4615 THS = MID$ (TS,7,2) + ":" + HIDS (TS,10,2) + ":" + EMS + "
4620 RETURN
4625 REM EDIT PARAMETERS
50000

```

```

5618 TEXT = HOME : PRINT PRESENT PARAMETERS. HOME:"; PRINT
5620 PRINT "(1) PHASE DETECTOR OUTPUT, FULL SCALE, IN PS=";
5622 PRINT FS: NORMAL : PRINT
5625 FLASH : PRINT "NORMAL : PRINT
5649 PRINT "(2) PERFECT (P) OR EQUALLY (E) UNSTABLE REFERENCE STANDARD; NO
      H IS "
5645 FLASH : PRINT RSS: NORMAL : PRINT
5649 PRINT "(3) SECONDS PER DATA BATCH=";
5650 FLASH : PRINT KH: NORMAL : PRINT
5650 PRINT "(4) UTM BEFORE HARD COPY (U) OR CONTINUOUS DATA TAKING
      (C), NOW ";
5655 FLASH : PRINT OS: NORMAL : PRINT
5679 PRINT "(5) LOG6 SIGMA PLOT MAX IS EXP ";
5675 FLASH : PRINT SH: NORMAL : PRINT
5688 PRINT "(6) LIN DRIFT PLOT, MAX *EXP-12 IS ";
5685 FLASH : PRINT DH: NORMAL : PRINT
5690 PRINT "(7) TEMPERATURE PLOT SPAN IS ";: FLASH : PRINT TS;: NORMAL
      : PRINT "C": PRINT
      : PRINT "(?) LOG6 SIGMA PLOT MAX IS EXP ";
      : PRINT
      : CHANGE (1-7) OR (RETURN)? ":" PRINT
5129 GET JS
5125 IF ASC (JS) > 55 THEN 5100
5130 DN UHL (JS) 6001 5210 5228 5230 5240 5250 5260 5270
5140 C1 = FS / 10000: REM INPUT PS/HU
5142 C2 = C1 * CS:CH = C1 * CQ
5145 YB = DM:VB = - DM: REM DRIFT PLOT SCALE
5150 IF RSS = "E" THEN C2 = .787:REM "EQUAL REFERENCE ASSUMED"
5152 IF RSS = "P" THEN C2 = 1:REM "PERFECT REFERENCE ASSUMED"
5155 C4 = C3 * C2 * CH: REM ".COLOR= *X.*? OR 1*1000
5157 C2 = .4343 * L06 ((C3 * C2) - 12*CY = .217: REM CONSTANTS FOR
SCALING LOG6 SIGMA
5160 REM NEXT FINDS INTERVAL,DP, BETWEEN POINTS IN TIME PLOT
5170 LP = 13
5170 LP = LP - 1: IF LP = 1 THEN 5198
5185 IF TAK(LP) > KH / 100 THEN 5188
5190 DP = TAK(LP): REM DP AND LP ARE NOW KNOWN
5205 RETURN
5210 INPUT "(1) INPUT PS= ";FS
5215 6010 5010
5220 INPUT "(2) REFERENCE, E OR P IS ";RSS
5225 6010 5010
5230 INPUT "(3) SECONDS PER BATCH=";KH
5235 6010 5010
5240 INPUT "(4) UTM (U) OR CONTINUOUS (C) ";OS
5245 6010 5010
5250 INPUT "SIGMA PLOT MAX, EXP ";SH
5260 INPUT "DRIFT PLOT SCALE, *EXP-12=";DH
5265 6010 5010
5270 INPUT "TEMP PLOT SPAN=";TS
5275 6010 5010
5300 REM TURN ON TRENCOM PRINTER
5308 PRINT CHR$(4); "PR#1"
5310 PRINT CHR$(8);
5328 PRINT CHR$(8);
5330 POKE 1913,6: POKE 1785,72: POKE 1637,80: REM MARGINS
5340 RETURN
5400 REM MOVE TO END OF PAGE
5410 600100 52000
5420 FOR LL = 1 TO 5: PRINT : NEXT
5430 CALL 1813: RETURN
5600 REM HARD COPY TABLE
5610 60300: REM TURN ON PRINTER
5615 HOME : PRINT CHR$(9); "N": REM DISABLE CRT
5618 POKE 33,80: REM MARGIN
5619 PRINT MM: TABX 40):REM
5620 PRINT TS$: TABX 40); "TEMP="; FN R2(X16); "C"; TABX 55); "PHASE=";
      FN R1(P6); "PS"
5623 PRINT TS$: TABX 40); "TEMP="; FN R2(X17); "C"; TABX 55); "PHASE=";
      FN R1(P7); "PS"
5626 PRINT TS$: TABX 40); "T0H="; FN R2(X11); "C"; THIGH="; FN R2(XH); "C";
      FN R2(XT); "C"; TABX 55); "PHASE=";
5628 PRINT "LAST BATCH"; TABX 37); "CONTINUOUS"; TABX 53); "TIME"
5629 PRINT "T0L"; TABX 14); "SAMPLES"; TABX 23); "SIGMA"; TABX 35); "SAMPLES
      "; TABX 44); "SIGMA"; TABX 53); "ERROR"
5630 PRINT "SEC"; TABX 22); "EXP-15"; TABX 43); "EXP-15"; TABX 54); "
      PS"
      FUR L = 1 TO 13
5630 PRINT TAK(L); TABX 14); KDL(L); TABX 21);
5675 D1 = 1:HI = 8
5680 X = CA * SQR (SDXL) / (KDL(L) + EP): 60SUB 11000
5680 PRINT TABX 35); NDL(L); TABX 43); "
5685 X1 = C4 * SQR (SDXL) / (NDL(L) + EP)
5688 X1 = C4 * SQR (SDXL) / (NDL(L) + EP)
5688 X = XI:DI = 1:HI = 7: 60SUB 11000
5690 TE = XI * TAK(L) / CH
5720 PRINT TABX 53); FN R(TE)
5730 NEXT L: PRINT CHR$(9); "I": CALL 1013: REM ENABLE CRT
5740 POKE 33,40: PRINT CHR$(9); "I": CALL 1013: REM ENABLE CRT
5740 RETURN
5800 REM PRINT GRAPHICS
5810 60300
5820 POKE 1145,105
5830 CALL - 16038
5840 CALL 1013: RETURN
5840 REM FORMATTED LIST
5850 60300: REM GET TIME
5850 6010 6010
5850 POKE 33,33
5855 60300: REM TURN ON PRINTER
5855 DEF FN CT(AD) = PEAK (AD) + 256 * PEAK (AD + 1)
5855 SR = FN CT(183)
5855 LH = FN CT(182):FR = FN CT(189)
5855 HH = FN CT(115):ST = FN CT(111)
5855 PRINT : PRINT DT$,TH$; PRINT
5855 PRINT "PROGRAM LENGTH= "; LH - SR;" BYTES"
      - ST + FR - LH;" BYTES"
5855 PRINT "FREE MEMORY= "; ST - FR;" BYTES"
5855 PRINT "START= ";SR;" LOWE= ";SLH;" FREE= ";FR;" STRING= ";ST;" HIHE= ";H
      H
5110 PRINT "TURN OFF PRINTER AND RECONNECT PLE
      H
5120 PRINT
5130 LIST
5140 CALL - 263838: REM TURN OFF PRINTER AND RECONNECT PLE
      END
100000 REM RECTANGULAR PLOT ROUTINES, ERASE @100000, INIT @100000, PLOT
100005 POINT @104000, COMMENT @100000
100005 HCR = MCOLOR= 3
100010 I = 0: POKE 8125,1: REM PLOT SHAPE PARAMETER
100015 SH = 2: RDT = 6: SCLE= 1
100020 X8 = 39:Y8 = 23:Y8 = 15:Y9 = 6
100025 X7 = (X9 - X8) / 10:Y7 = (Y8 - Y9) / 10
100030 REM PLT BORDER NEXT
100035 HPLT X8,Y8 TO X8,Y9 TO X9,Y9 TO X9,Y8 TO X8,Y8
100040 FOR YT = Y8 TO Y9 STEP 2 * V
100045 HPLT X8,YT TO X9,YT

```

```

16647 NEXT
16648 FOR VT = V9 TO V8 STEP V7
16649 REM
16650 HPLOT X9,YT TO X9 - 2,YT
16651 NEXT VT
16652 X6 = (X9 + X8) / 2; Y6 = ((V9 + V8) / 2
16653 X6 = (X9 + X8) / 2; Y6 = ((V9 + V8) / 2
16654 HPLOT X9,Y6 TO X8,Y6; HPLOT X9 - 4,Y6 TO X9,Y6
16655 HPLOT X8,Y8 - 4 TO X8,Y8; HPLOT X9,Y9 TO X6,Y9 + 4
16656 FOR XT = X8 TO X9 STEP X7
16657 HPLOT XT,Y8 - 2 TO XT,Y8
16658 HPLOT XT,Y9 TU XT,Y9 + 2
16659 NEXT XT
16660 REM
16661 INITIIZE PLOT, SET LIMITS (XL,XH,YL,YH) AND (YL,YB)
16662 X7 = (X9 - X8) / 10; Y7 = (V8 - V9) / 10
16663 X8 = (X9 - X8) / (XH - XL); D9 = X8 - D8 * XL
16664 D6 = (V9 - V8) / (VH - VL); D7 = V8 - D6 * VL
16665 D4 = (V9 - V8) / (VH - VL); D5 = V8 - D4 * V8
16666 POKE - 16384,0; POKE - 16384,0
16667 POKE - 16297,0; POKE - 16381,0; HCOLOR= 3
16668 LABEL WITH CHR GEN
16669 CALL 3872
16670 PRINT CHR$ (1); CHR$ (17)
16671 HTAB (1); HTAB (3); PRINT VH
16672 HTAB (3); HTAB (1); PRINT "LOG"
16673 HTAB (4); HTAB (1); PRINT "SIGMA"
16674 HTAB (5); HTAB (1); PRINT "1 SEC"
16675 HTAB (6); HTAB (1); PRINT "RAD"
16676 HTAB (19); HTAB (3); PRINT VL + 1
16677 HTAB (15); HTAB (3); PRINT VL + 2
16678 HTAB (20); HTAB (6); PRINT XL
16679 HTAB (1); HTAB (36); PRINT VH
16680 HTAB (3); HTAB (36); PRINT V8
16681 HTAB (4); HTAB (36); PRINT "FREQ"
16682 HTAB (20); HTAB (32); PRINT VH
16683 HTAB (5); HTAB (36); PRINT "HE-12"
16684 HTAB (18); HTAB (36); PRINT "0"
16685 HTAB (28); HTAB (15); PRINT XCS
16686 HTAB (12); HTAB (36); PRINT "TEMP"
16687 HTAB (13); HTAB (36); PRINT "SPON"
16688 HTAB (14); HTAB (36); PRINT TS,"C"
16689 CALL 1613; REM TURN OFF HLR CHR GEN
16690 POKE - 16381,0
16691 SCALE= 2; ROT= 0; SCALE= 1; RETURN
16692 DRAW 1 AT 289,52; DRAW 2 AT 17,52
16693 SCALE= 1
16694 REM CHANGES PLOT POINT SHAPE
16695 X7 = (X9 - X8) / 10; Y7 = (V8 - V9) / 10
16696 SH = 2; ROT= 0; SCALE= 1; RETURN
16697 UN 1 606308 163116,163115,163220,163325,163335
16698 I = 1 + 1; 6010 16340
16699 SH = 1; SCALE= 1; RETURN
16700 SH = 2; ROT= 0; SCALE= 2; RETURN
16701 SH = 3; ROT= 0; SCALE= 1; RETURN
16702 SH = 2; ROT= 0; SCALE= 2; RETURN
16703 SH = 1; ROT= 16; SCALE= 2; RETURN
16704 REM
16705 REM PLOT XP,YP IN LIMITS XL,XH,YL,YH
16706 XT = D8 * XP + D9; YT = D6 * VP + D7
16707 IF XT < X8 THEN XT = X8
16708 IF XT > X9 THEN XT = X9
16709 IF VT < V9 THEN VT = V9
16710 IF VT > V8 THEN VT = V8
16711 DRAW SH AT XT,YT
16712 RETURN
16713 REM ALTERNATE PLOT VP IN V8,VT
16714 XT = D8 * XP + D9; YT = D4 * VP + D5
16715 IF XT < X8 THEN XT = X8
16716 IF XT > X9 THEN XT = X9
16717 IF VT < V9 THEN VT = V9
16718 IF VT > V8 THEN VT = V8
16719 DRAW 1 AT XT,YT
16720 RETURN
16721 REM COMMENT ON PLOT
16722 INPUT "TYPE QUOTE THEN COMMENT ";CHR$ (17)
16723 DRAW SH RT 4.8 * (1.5 + 1)
16724 INPUT "TYPE QUOTE THEN BORDER ";CHR$ (17)
16725 HCOLOR= 0; REM ERASE BORDER
16726 HCOLOR= 8; REM
16727 HPLOT X8..8 * (1 + 1) TO X8..8 * (2 + 1)
16728 HCOLOR= 3
16729 CALL 3872; PRINT CHR$ (1); PRINT CHR$ (17)
16730 UTAB (2 + 1); HTAB (2); PRINT CHR$ (17)
16731 CALL 1613; POKE - 1630,0
16732 RETURN
16733 REM INITIALIZE PLOT, SET LIMITS (XL,XH,YL,YH)
16734 X9 = 239; Y9 = 150; YB = 0
16735 X8 = (X9 - X8) / 10; Y7 = (V8 - V9) / 10
16736 XH = (XH - XL); D9 = (VH - VL) / 10
16737 D8 = (V8 - V8) / (VH - VL); D7 = (YH - YL) / 10
16738 D6 = (V9 - V8) / (VH - VL); D5 = (YH - YL) / 10
16739 POKE - 16384,0; POKE - 16384,0
16740 POKE - 16297,0; POKE - 16381,0; HCOLOR= 3
16741 LABEL WITH CHR GEN
16742 CALL 3872
16743 PRINT CHR$ (1); CHR$ (17)
16744 HTAB (1); HTAB (3); PRINT VH
16745 HTAB (11); HTAB (3); PRINT YH
16746 HTAB (3); HTAB (3); PRINT YB
16747 HTAB (1); PRINT "LOG"
16748 HTAB (4); HTAB (1); PRINT "SIGMA"
16749 HTAB (5); HTAB (1); PRINT "US"
16750 HTAB (6); HTAB (1); PRINT "TRU"
16751 HTAB (19); HTAB (3); PRINT VL
16752 HTAB (15); HTAB (3); PRINT XCS
16753 HTAB (11); HTAB (3); PRINT YL + 1
16754 HTAB (28); HTAB (12); PRINT "10"
16755 HTAB (28); HTAB (17); PRINT "100"
16756 HTAB (28); HTAB (27); PRINT "1000"
16757 HTAB (28); HTAB (33); PRINT "SEC"
16758 CALL 1613; REM TURN OFF HLR CHR GEN
16759 FOR XT = X8 TO X9 STEP 2 * X7
16760 HPLOT XT,Y8 TO XT,Y9
16761 NEXT
16762 REM
16763 SCALE= 1
16764 REM CHANGES PLOT POINT SHAPE
16765 SH = 2; ROT= 0; SCALE= 1; RETURN
16766 UN 1 606308 163116,163115,163220,163325,163335
16767 I = 1 + 1; 6010 16340
16768 SH = 1; SCALE= 1; RETURN
16769 SH = 2; ROT= 0; SCALE= 2; RETURN
16770 SH = 3; ROT= 0; SCALE= 1; RETURN
16771 SH = 2; ROT= 0; SCALE= 2; RETURN
16772 SH = 1; SCALE= 1; RETURN
16773 SH = 2; ROT= 0; SCALE= 2; RETURN
16774 SH = 3; ROT= 0; SCALE= 1; RETURN
16775 SH = 4; ROT= 0; SCALE= 2; RETURN
16776 SH = 5; ROT= 0; SCALE= 1; RETURN
16777 SH = 6; ROT= 0; SCALE= 2; RETURN
16778 SH = 7; ROT= 0; SCALE= 1; RETURN
16779 SH = 8; ROT= 0; SCALE= 2; RETURN
16780 SH = 9; ROT= 0; SCALE= 1; RETURN
16781 SH = 10; ROT= 0; SCALE= 2; RETURN
16782 SH = 11; ROT= 0; SCALE= 1; RETURN
16783 SH = 12; ROT= 0; SCALE= 2; RETURN
16784 SH = 13; ROT= 0; SCALE= 1; RETURN
16785 SH = 14; ROT= 0; SCALE= 2; RETURN
16786 SH = 15; ROT= 0; SCALE= 1; RETURN
16787 SH = 16; ROT= 0; SCALE= 2; RETURN
16788 SH = 17; ROT= 0; SCALE= 1; RETURN
16789 SH = 18; ROT= 0; SCALE= 2; RETURN
16790 SH = 19; ROT= 0; SCALE= 1; RETURN
16791 SH = 20; ROT= 0; SCALE= 2; RETURN
16792 SH = 21; ROT= 0; SCALE= 1; RETURN
16793 SH = 22; ROT= 0; SCALE= 2; RETURN
16794 SH = 23; ROT= 0; SCALE= 1; RETURN
16795 SH = 24; ROT= 0; SCALE= 2; RETURN
16796 SH = 25; ROT= 0; SCALE= 1; RETURN
16797 SH = 26; ROT= 0; SCALE= 2; RETURN
16798 SH = 27; ROT= 0; SCALE= 1; RETURN
16799 SH = 28; ROT= 0; SCALE= 2; RETURN
16800 SH = 29; ROT= 0; SCALE= 1; RETURN
16801 SH = 30; ROT= 0; SCALE= 2; RETURN
16802 SH = 31; ROT= 0; SCALE= 1; RETURN
16803 SH = 32; ROT= 0; SCALE= 2; RETURN
16804 SH = 33; ROT= 0; SCALE= 1; RETURN
16805 CALL 1613; REM
16806 16845 POKE - 16301,0
16807 FOR XT = X8 TO X9 STEP 2 * X7
16808 HPLOT XT,Y8 TO XT,Y9
16809 NEXT
16810 REM
16811 SCALE= 1
16812 REM CHANGES PLOT POINT SHAPE
16813 SH = 33; ROT= 0; SCALE= 1; RETURN
16814 SH = 34; ROT= 0; SCALE= 2; RETURN
16815 SH = 35; ROT= 0; SCALE= 1; RETURN
16816 SH = 36; ROT= 0; SCALE= 2; RETURN
16817 SH = 37; ROT= 0; SCALE= 1; RETURN
16818 SH = 38; ROT= 0; SCALE= 2; RETURN
16819 SH = 39; ROT= 0; SCALE= 1; RETURN
16820 SH = 40; ROT= 0; SCALE= 2; RETURN
16821 SH = 41; ROT= 0; SCALE= 1; RETURN
16822 SH = 42; ROT= 0; SCALE= 2; RETURN
16823 SH = 43; ROT= 0; SCALE= 1; RETURN
16824 SH = 44; ROT= 0; SCALE= 2; RETURN
16825 SH = 45; ROT= 0; SCALE= 1; RETURN
16826 SH = 46; ROT= 0; SCALE= 2; RETURN
16827 SH = 47; ROT= 0; SCALE= 1; RETURN
16828 SH = 48; ROT= 0; SCALE= 2; RETURN
16829 SH = 49; ROT= 0; SCALE= 1; RETURN
16830 SH = 50; ROT= 0; SCALE= 2; RETURN
16831 SH = 51; ROT= 0; SCALE= 1; RETURN
16832 SH = 52; ROT= 0; SCALE= 2; RETURN
16833 SH = 53; ROT= 0; SCALE= 1; RETURN
16834 SH = 54; ROT= 0; SCALE= 2; RETURN
16835 SH = 55; ROT= 0; SCALE= 1; RETURN
16836 SH = 56; ROT= 0; SCALE= 2; RETURN
16837 SH = 57; ROT= 0; SCALE= 1; RETURN
16838 SH = 58; ROT= 0; SCALE= 2; RETURN
16839 SH = 59; ROT= 0; SCALE= 1; RETURN
16840 SH = 60; ROT= 0; SCALE= 2; RETURN
16841 SH = 61; ROT= 0; SCALE= 1; RETURN
16842 SH = 62; ROT= 0; SCALE= 2; RETURN
16843 SH = 63; ROT= 0; SCALE= 1; RETURN
16844 SH = 64; ROT= 0; SCALE= 2; RETURN
16845 SH = 65; ROT= 0; SCALE= 1; RETURN
16846 SH = 66; ROT= 0; SCALE= 2; RETURN
16847 SH = 67; ROT= 0; SCALE= 1; RETURN
16848 SH = 68; ROT= 0; SCALE= 2; RETURN
16849 SH = 69; ROT= 0; SCALE= 1; RETURN
16850 SH = 70; ROT= 0; SCALE= 2; RETURN
16851 SH = 71; ROT= 0; SCALE= 1; RETURN
16852 SH = 72; ROT= 0; SCALE= 2; RETURN
16853 SH = 73; ROT= 0; SCALE= 1; RETURN
16854 SH = 74; ROT= 0; SCALE= 2; RETURN
16855 SH = 75; ROT= 0; SCALE= 1; RETURN
16856 SH = 76; ROT= 0; SCALE= 2; RETURN
16857 SH = 77; ROT= 0; SCALE= 1; RETURN
16858 SH = 78; ROT= 0; SCALE= 2; RETURN
16859 SH = 79; ROT= 0; SCALE= 1; RETURN
16860 SH = 80; ROT= 0; SCALE= 2; RETURN
16861 SH = 81; ROT= 0; SCALE= 1; RETURN
16862 SH = 82; ROT= 0; SCALE= 2; RETURN
16863 SH = 83; ROT= 0; SCALE= 1; RETURN
16864 SH = 84; ROT= 0; SCALE= 2; RETURN
16865 SH = 85; ROT= 0; SCALE= 1; RETURN
16866 SH = 86; ROT= 0; SCALE= 2; RETURN
16867 SH = 87; ROT= 0; SCALE= 1; RETURN
16868 SH = 88; ROT= 0; SCALE= 2; RETURN
16869 SH = 89; ROT= 0; SCALE= 1; RETURN
16870 SH = 90; ROT= 0; SCALE= 2; RETURN
16871 SH = 91; ROT= 0; SCALE= 1; RETURN
16872 SH = 92; ROT= 0; SCALE= 2; RETURN
16873 SH = 93; ROT= 0; SCALE= 1; RETURN
16874 SH = 94; ROT= 0; SCALE= 2; RETURN
16875 SH = 95; ROT= 0; SCALE= 1; RETURN
16876 SH = 96; ROT= 0; SCALE= 2; RETURN
16877 SH = 97; ROT= 0; SCALE= 1; RETURN
16878 SH = 98; ROT= 0; SCALE= 2; RETURN
16879 SH = 99; ROT= 0; SCALE= 1; RETURN
16880 SH = 100; ROT= 0; SCALE= 2; RETURN
16881 SH = 101; ROT= 0; SCALE= 1; RETURN
16882 SH = 102; ROT= 0; SCALE= 2; RETURN
16883 SH = 103; ROT= 0; SCALE= 1; RETURN
16884 SH = 104; ROT= 0; SCALE= 2; RETURN
16885 SH = 105; ROT= 0; SCALE= 1; RETURN
16886 SH = 106; ROT= 0; SCALE= 2; RETURN
16887 SH = 107; ROT= 0; SCALE= 1; RETURN
16888 SH = 108; ROT= 0; SCALE= 2; RETURN
16889 SH = 109; ROT= 0; SCALE= 1; RETURN
16890 SH = 110; ROT= 0; SCALE= 2; RETURN
16891 SH = 111; ROT= 0; SCALE= 1; RETURN
16892 SH = 112; ROT= 0; SCALE= 2; RETURN
16893 SH = 113; ROT= 0; SCALE= 1; RETURN
16894 SH = 114; ROT= 0; SCALE= 2; RETURN
16895 SH = 115; ROT= 0; SCALE= 1; RETURN
16896 SH = 116; ROT= 0; SCALE= 2; RETURN
16897 SH = 117; ROT= 0; SCALE= 1; RETURN
16898 SH = 118; ROT= 0; SCALE= 2; RETURN
16899 SH = 119; ROT= 0; SCALE= 1; RETURN
16900 SH = 120; ROT= 0; SCALE= 2; RETURN
16901 SH = 121; ROT= 0; SCALE= 1; RETURN
16902 SH = 122; ROT= 0; SCALE= 2; RETURN
16903 SH = 123; ROT= 0; SCALE= 1; RETURN
16904 SH = 124; ROT= 0; SCALE= 2; RETURN
16905 SH = 125; ROT= 0; SCALE= 1; RETURN
16906 SH = 126; ROT= 0; SCALE= 2; RETURN
16907 SH = 127; ROT= 0; SCALE= 1; RETURN
16908 SH = 128; ROT= 0; SCALE= 2; RETURN
16909 SH = 129; ROT= 0; SCALE= 1; RETURN
16910 SH = 130; ROT= 0; SCALE= 2; RETURN
16911 SH = 131; ROT= 0; SCALE= 1; RETURN
16912 SH = 132; ROT= 0; SCALE= 2; RETURN
16913 SH = 133; ROT= 0; SCALE= 1; RETURN
16914 SH = 134; ROT= 0; SCALE= 2; RETURN
16915 SH = 135; ROT= 0; SCALE= 1; RETURN
16916 SH = 136; ROT= 0; SCALE= 2; RETURN
16917 SH = 137; ROT= 0; SCALE= 1; RETURN
16918 SH = 138; ROT= 0; SCALE= 2; RETURN
16919 SH = 139; ROT= 0; SCALE= 1; RETURN
16920 SH = 140; ROT= 0; SCALE= 2; RETURN
16921 SH = 141; ROT= 0; SCALE= 1; RETURN
16922 SH = 142; ROT= 0; SCALE= 2; RETURN
16923 SH = 143; ROT= 0; SCALE= 1; RETURN
16924 SH = 144; ROT= 0; SCALE= 2; RETURN
16925 SH = 145; ROT= 0; SCALE= 1; RETURN
16926 SH = 146; ROT= 0; SCALE= 2; RETURN
16927 SH = 147; ROT= 0; SCALE= 1; RETURN
16928 SH = 148; ROT= 0; SCALE= 2; RETURN
16929 SH = 149; ROT= 0; SCALE= 1; RETURN
16930 SH = 150; ROT= 0; SCALE= 2; RETURN
16931 SH = 151; ROT= 0; SCALE= 1; RETURN
16932 SH = 152; ROT= 0; SCALE= 2; RETURN
16933 SH = 153; ROT= 0; SCALE= 1; RETURN
16934 SH = 154; ROT= 0; SCALE= 2; RETURN
16935 SH = 155; ROT= 0; SCALE= 1; RETURN
16936 SH = 156; ROT= 0; SCALE= 2; RETURN
16937 SH = 157; ROT= 0; SCALE= 1; RETURN
16938 SH = 158; ROT= 0; SCALE= 2; RETURN
16939 SH = 159; ROT= 0; SCALE= 1; RETURN
16940 SH = 160; ROT= 0; SCALE= 2; RETURN
16941 SH = 161; ROT= 0; SCALE= 1; RETURN
16942 SH = 162; ROT= 0; SCALE= 2; RETURN
16943 SH = 163; ROT= 0; SCALE= 1; RETURN
16944 SH = 164; ROT= 0; SCALE= 2; RETURN
16945 SH = 165; ROT= 0; SCALE= 1; RETURN
16946 SH = 166; ROT= 0; SCALE= 2; RETURN
16947 SH = 167; ROT= 0; SCALE= 1; RETURN
16948 SH = 168; ROT= 0; SCALE= 2; RETURN
16949 SH = 169; ROT= 0; SCALE= 1; RETURN
16950 SH = 170; ROT= 0; SCALE= 2; RETURN
16951 SH = 171; ROT= 0; SCALE= 1; RETURN
16952 SH = 172; ROT= 0; SCALE= 2; RETURN
16953 SH = 173; ROT= 0; SCALE= 1; RETURN
16954 SH = 174; ROT= 0; SCALE= 2; RETURN
16955 SH = 175; ROT= 0; SCALE= 1; RETURN
16956 SH = 176; ROT= 0; SCALE= 2; RETURN
16957 SH = 177; ROT= 0; SCALE= 1; RETURN
16958 SH = 178; ROT= 0; SCALE= 2; RETURN
16959 SH = 179; ROT= 0; SCALE= 1; RETURN
16960 SH = 180; ROT= 0; SCALE= 2; RETURN
16961 SH = 181; ROT= 0; SCALE= 1; RETURN
16962 SH = 182; ROT= 0; SCALE= 2; RETURN
16963 SH = 183; ROT= 0; SCALE= 1; RETURN
16964 SH = 184; ROT= 0; SCALE= 2; RETURN
16965 SH = 185; ROT= 0; SCALE= 1; RETURN
16966 SH = 186; ROT= 0; SCALE= 2; RETURN
16967 SH = 187; ROT= 0; SCALE= 1; RETURN
16968 SH = 188; ROT= 0; SCALE= 2; RETURN
16969 SH = 189; ROT= 0; SCALE= 1; RETURN
16970 SH = 190; ROT= 0; SCALE= 2; RETURN
16971 SH = 191; ROT= 0; SCALE= 1; RETURN
16972 SH = 192; ROT= 0; SCALE= 2; RETURN
16973 SH = 193; ROT= 0; SCALE= 1; RETURN
16974 SH = 194; ROT= 0; SCALE= 2; RETURN
16975 SH = 195; ROT= 0; SCALE= 1; RETURN
16976 SH = 196; ROT= 0; SCALE= 2; RETURN
16977 SH = 197; ROT= 0; SCALE= 1; RETURN
16978 SH = 198; ROT= 0; SCALE= 2; RETURN
16979 SH = 199; ROT= 0; SCALE= 1; RETURN
16980 SH = 200; ROT= 0; SCALE= 2; RETURN
16981 SH = 201; ROT= 0; SCALE= 1; RETURN
16982 SH = 202; ROT= 0; SCALE= 2; RETURN
16983 SH = 203; ROT= 0; SCALE= 1; RETURN
16984 SH = 204; ROT= 0; SCALE= 2; RETURN
16985 SH = 205; ROT= 0; SCALE= 1; RETURN
16986 SH = 206; ROT= 0; SCALE= 2; RETURN
16987 SH = 207; ROT= 0; SCALE= 1; RETURN
16988 SH = 208; ROT= 0; SCALE= 2; RETURN
16989 SH = 209; ROT= 0; SCALE= 1; RETURN
16990 SH = 210; ROT= 0; SCALE= 2; RETURN
16991 SH = 211; ROT= 0; SCALE= 1; RETURN
16992 SH = 212; ROT= 0; SCALE= 2; RETURN
16993 SH = 213; ROT= 0; SCALE= 1; RETURN
16994 SH = 214; ROT= 0; SCALE= 2; RETURN
16995 SH = 215; ROT= 0; SCALE= 1; RETURN
16996 SH = 216; ROT= 0; SCALE= 2; RETURN
16997 SH = 217; ROT= 0; SCALE= 1; RETURN
16998 SH = 218; ROT= 0; SCALE= 2; RETURN
16999 SH = 219; ROT= 0; SCALE= 1; RETURN
17000 SH = 220; ROT= 0; SCALE= 2; RETURN
17001 SH = 221; ROT= 0; SCALE= 1; RETURN
17002 SH = 222; ROT= 0; SCALE= 2; RETURN
17003 SH = 223; ROT= 0; SCALE= 1; RETURN
17004 SH = 224; ROT= 0; SCALE= 2; RETURN
17005 SH = 225; ROT= 0; SCALE= 1; RETURN
17006 SH = 226; ROT= 0; SCALE= 2; RETURN
17007 SH = 227; ROT= 0; SCALE= 1; RETURN
17008 SH = 228; ROT= 0; SCALE= 2; RETURN
17009 SH = 229; ROT= 0; SCALE= 1; RETURN
17010 SH = 230; ROT= 0; SCALE= 2; RETURN
17011 SH = 231; ROT= 0; SCALE= 1; RETURN
17012 SH = 232; ROT= 0; SCALE= 2; RETURN
17013 SH = 233; ROT= 0; SCALE= 1; RETURN
17014 SH = 234; ROT= 0; SCALE= 2; RETURN
17015 SH = 235; ROT= 0; SCALE= 1; RETURN
17016 SH = 236; ROT= 0; SCALE= 2; RETURN
17017 SH = 237; ROT= 0; SCALE= 1; RETURN
17018 SH = 238; ROT= 0; SCALE= 2; RETURN
17019 SH = 239; ROT= 0; SCALE= 1; RETURN
17020 SH = 240; ROT= 0; SCALE= 2; RETURN
17021 SH = 241; ROT= 0; SCALE= 1; RETURN
17022 SH = 242; ROT= 0; SCALE= 2; RETURN
17023 SH = 243; ROT= 0; SCALE= 1; RETURN
17024 SH = 244; ROT= 0; SCALE= 2; RETURN
17025 SH = 245; ROT= 0; SCALE= 1; RETURN
17026 SH = 246; ROT= 0; SCALE= 2; RETURN
17027 SH = 247; ROT= 0; SCALE= 1; RETURN
17028 SH = 248; ROT= 0; SCALE= 2; RETURN
17029 SH = 249; ROT= 0; SCALE= 1; RETURN
17030 SH = 250; ROT= 0; SCALE= 2; RETURN
17031 SH = 251; ROT= 0; SCALE= 1; RETURN
17032 SH = 252; ROT= 0; SCALE= 2; RETURN
17033 SH = 253; ROT= 0; SCALE= 1; RETURN
17034 SH = 254; ROT= 0; SCALE= 2; RETURN
17035 SH = 255; ROT= 0; SCALE= 1; RETURN
17036 SH = 256; ROT= 0; SCALE= 2; RETURN
17037 SH = 257; ROT= 0; SCALE= 1; RETURN
17038 SH = 258; ROT= 0; SCALE= 2; RETURN
17039 SH = 259; ROT= 0; SCALE= 1; RETURN
17040 SH = 260; ROT= 0; SCALE= 2; RETURN
17041 SH = 261; ROT= 0; SCALE= 1; RETURN
17042 SH = 262; ROT= 0; SCALE= 2; RETURN
17043 SH = 263; ROT= 0; SCALE= 1; RETURN
17044 SH = 264; ROT= 0; SCALE= 2; RETURN
17045 SH = 265; ROT= 0; SCALE= 1; RETURN
17046 SH = 266; ROT= 0; SCALE= 2; RETURN
17047 SH = 267; ROT= 0; SCALE= 1; RETURN
17048 SH = 268; ROT= 0; SCALE= 2; RETURN
17049 SH = 269; ROT= 0; SCALE= 1; RETURN
17050 SH = 270; ROT= 0; SCALE= 2; RETURN
17051 SH = 271; ROT= 0; SCALE= 1; RETURN
17052 SH = 272; ROT= 0; SCALE= 2; RETURN
17053 SH = 273; ROT= 0; SCALE= 1; RETURN
17054 SH = 274; ROT= 0; SCALE= 2; RETURN
17055 SH = 275; ROT= 0; SCALE= 1; RETURN
17056 SH = 276; ROT= 0; SCALE= 2; RETURN
17057 SH = 277; ROT= 0; SCALE= 1; RETURN
17058 SH = 278; ROT= 0; SCALE= 2; RETURN
17059 SH = 279; ROT= 0; SCALE= 1; RETURN
17060 SH = 280; ROT= 0; SCALE= 2; RETURN
17061 SH = 281; ROT= 0; SCALE= 1; RETURN
17062 SH = 282; ROT= 0; SCALE= 2; RETURN
17063 SH = 283; ROT= 0; SCALE= 1; RETURN
17064 SH = 284; ROT= 0; SCALE= 2; RETURN
17065 SH = 285; ROT= 0; SCALE= 1; RETURN
17066 SH = 286; ROT= 0; SCALE= 2; RETURN
17067 SH = 287; ROT= 0; SCALE= 1; RETURN
17068 SH = 288; ROT= 0; SCALE= 2; RETURN
17069 SH = 289; ROT= 0; SCALE= 1; RETURN
17070 SH = 290; ROT= 0; SCALE= 2; RETURN
17071 SH = 291; ROT= 0; SCALE= 1; RETURN
17072 SH = 292; ROT= 0; SCALE= 2; RETURN
17073 SH = 293; ROT= 0; SCALE= 1; RETURN
17074 SH = 294; ROT= 0; SCALE= 2; RETURN
17075 SH = 295; ROT= 0; SCALE= 1; RETURN
17076 SH = 296; ROT= 0; SCALE= 2; RETURN
17077 SH = 297; ROT= 0; SCALE= 1; RETURN
17078 SH = 298; ROT= 0; SCALE= 2; RETURN
17079 SH = 299; ROT= 0; SCALE= 1; RETURN
17080 SH = 300; ROT= 0; SCALE= 2; RETURN
17081 SH = 301; ROT= 0; SCALE= 1; RETURN
17082 SH = 302; ROT= 0; SCALE= 2; RETURN
17083 SH = 303; ROT= 0; SCALE= 1; RETURN
17084 SH = 304; ROT= 0; SCALE= 2; RETURN
17085 SH = 305; ROT= 0; SCALE= 1; RETURN
17086 SH = 306; ROT= 0; SCALE= 2; RETURN
17087 SH = 307; ROT= 0; SCALE= 1; RETURN
17088 SH = 308; ROT= 0; SCALE= 2; RETURN
17089 SH = 309; ROT= 0; SCALE= 1; RETURN
17090 SH = 310; ROT= 0; SCALE= 2; RETURN
17091 SH = 311; ROT= 0; SCALE= 1; RETURN
17092 SH = 312; ROT= 0; SCALE= 2; RETURN
17093 SH = 313; ROT= 0; SCALE= 1; RETURN
17094 SH = 314; ROT= 0; SCALE= 2; RETURN
17095 SH = 315; SH = 2; ROT= 0; SCALE= 2; RETURN
17096 SH = 316; SH = 3; ROT= 0; SCALE= 1; RETURN
17097 SH = 317; SH = 32; SCALE= 1; RETURN
17098 SH = 318; SH = 2; ROT= 0; SCALE= 2; RETURN
17099 SH = 319; SH = 1; ROT= 0; SCALE= 1; RETURN
17100 SH = 320; SH = 1; ROT= 0; SCALE= 2; RETURN
17101 SH = 321; SH = 1; ROT= 0; SCALE= 1; RETURN
17102 SH = 322; SH = 1; ROT= 0; SCALE= 2; RETURN
17103 SH = 323; SH = 1; ROT= 0; SCALE= 1; RETURN
17104 SH = 324; SH = 1; ROT= 0; SCALE= 2; RETURN
17105 SH = 325; SH = 1; ROT= 0; SCALE= 1; RETURN
17106 SH = 326; SH = 1; ROT= 0; SCALE= 2; RETURN
17107 SH = 327; SH = 1; ROT= 0; SCALE= 1; RETURN
17108 SH = 328; SH = 1; ROT= 0; SCALE= 2; RETURN
17109 SH = 329; SH = 1; ROT= 0; SCALE= 1; RETURN
17110 SH = 330; SH = 1; ROT= 0; SCALE= 2; RETURN
17111 SH = 331; SH = 1; ROT= 0; SCALE
```

