IBM PC MST MONITOR SYSTEM

J.C. Sprott
E.W. Goetz

PLP 1039

November 1988

Plasma Studies

University of Wisconsin

IBM PC MST Monitor System

J. C. Sprott and E. W. Goetz

I. Introduction

Routine tune-up and monitoring of MST is done with an IBM PC, connected to a variety of dedicated diagnostic sensors hardwired to the PC through a 16-channel, 12-bit, A-to-D converter and special analog circuitry. The monitor system is an upgrade of the original Radio Shack Model I system developed for Tokapole II many years ago and described in PLP 889. A similar IBM PC monitor system is currently in use on Tokapole II. This PLP contains a description of the MST monitor system. Section II describes the diagnostic sensors, section III describes the electrical interface between the sensors and the computer, section IV describes the computer program used to analyze and display the data, and section V describes the calibration method. A parallel system that uses the same diagnostic sensors and analog circuitry with CAMAC-standard A-to-D converters and software that runs under IDL on Athena is also available but will not be described here.

II. Diagnostic Sensors

The following diagnostics form the basis of the monitor system:

Internal sensors:

1. Rogowskii loop at the wall encircling the plasma poloidally

2. Flux loop at the wall encircling the plasma poloidally

3. Double Langmuir probe located near the wall

External sensors:

4. Rogowskii loops encircling each toroidal current feed

5. Flux loop encircling the main poloidal iron core

6. Current shunt in the poloidal-field primary

The first two sensors could actually be located outside the vacuum vessel. However, very little space is available outside for the Rogowskii loop, and having it inside allows one to detect arcs at the poloidal gap. The flux loop is inside primarily to avoid having to make a correction for the toroidal magnetic flux soaking into the wall as a function of time.

The double Langmuir probe is located sufficiently near the wall to protect it from plasma damage but sufficiently close to the plasma to provide a density measurement that is reasonably proportional to the average density throughout the machine. The probe is calibrated

against a microwave interferometer that reads a chord-averaged density. The probe also provides a floating-potential measurement which crudely correlates with plasma temperature.

The Rogowskii loops encircling the toroidal current feeds are wired in series and used to determine the toroidal field at the wall. The flux loop encircling the iron core is used to determine the poloidal gap voltage and core flux. The current-shunt signal in the poloidal-field primary is not digitized and used by the computer, but is available in analog form to detect poloidal-gap arcs and to measure core magnetizing current.

All sensors are electrostatically shielded by a shield connected to the electrical interface circuit and insulated from machine ground. The electrical interface and computer ground are connected to the main ground bus at a single point.

III. Electrical Interface

The general strategy is to perform all the linear operations on the signals from the diagnostic sensors with analog electronics (operational amplifiers) before digitizing the signals and to let the computer perform all the nonlinear operations. This technique has the advantage of providing useful analog outputs that can be monitored with oscilloscopes or fed into CAMAC channels for general use and archiving.

A schematic representation of the electronics is shown in figures 1-4. The operational amplifiers perform integration and subtraction of the various signals. The inputs are labeled to correspond to the sensors described in section II. The circuit gains are adjusted to provide output signals in the range of -10 to +10 volts to avoid saturating the digitizers while maintaining good resolution (0.5 mV per bit).

The equations (in MKS units) that relate the outputs to the inputs (where not obvious) are

$$V_{tg} = \pi a^2 d\langle B_t \rangle / dt$$

$$B_{tw} = \mu_o I_t \; / \; 2\pi R_o$$

$$V_{pg} = d\Phi/dt$$

$$I_{mag} = NI_{pri} - I_p$$

where a is the minor radius of the torus (0.52 m), R_o is the major radius (1.5 m), and N is the number of turns on the poloidal iron core (typically N = 20). Actually, since the flux loop is about 1 cm inside the wall, a value of a = 0.51 m is used in the calculation of $\langle B_t \rangle$ from V_{tg}. Also, because of the toroidal curvature, a value of R_o = 1.47 m is used in the calculation of B_{tw} from I_t so that B_{tw} is equal to $\langle B_t \rangle$ in the absence of plasma.

Eight of the signals provided by the analog circuit are fed into a Data Translation, Inc. (100 Locke Drive, Marlborough, MA 01752), DT2801-A, 12-bit A-to-D converter board that occupies an expansion slot in the IBM PC. The signals are filtered with 27-μs RC circuits and diode-clipped at ±10 volts to protect the A-to-D converter. The A-to-D converter is triggered from the poloidal-gap-voltage signal, and the eight channels are sequentially read at 50-μs intervals for a total time of about 100 ms. Thus, each channel is read every 400 μs, and the first reading of channel 8 is 400 μs after the trigger. The A-to-D converter has a 50 ns sample-and-hold aperture delay and a 10 ns uncertainty. The data are transferred in real time by direct memory access (DMA) to an array in memory that can be read by the computer software. The eight analog signals are also sent to the CAMAC digitizers through unity-gain buffer amplifiers and 2:1 resistive voltage dividers, providing signals clipped at ±5 volts.

IV. Computer Program

The source code for the monitor program (PCMSTMON.BAS) is written in IBM PC Basic A3.30 and contains about 600 lines and 2000 statements. The code can be run in the interpreter mode for debugging, but more typically is compiled with the Microsoft IBM PC Basic Compiler 1.00 using Microway 87BASIC 3.04 to access the 8087 math co-processor. The object code (PCMSTMON.EXE) is about 72 kbytes in length and runs under

MS-DOS or PC-DOS 2.0 or later on an IBM PC or compatible with at least 256K of RAM. The entire program is memory-resident, and thus, once loaded, does not require disk access except to archive data. To archive data, a formatted disk should be in drive B. The computer is assumed to have a CGA, EGA or VGA and appropriate graphics monitor. A second, monochrome monitor is desirable but not required. The computer is assumed to have an asynchronous communications adapter (RS-232) connected to the primary communications port [COMM(1)]. The program is self-booting if the PCMSTMON disk is in drive A when the computer is turned on. The source code listing is included in Appendix A.

The program first initializes itself by trying to read a file CONFIG.DAT from the PCMSTMON disk that contains all the data stored in memory arrays when the program was last exited, assuming a normal (Control-C) exit (i.e., not due to a power failure or other abnormality). It then reads the A-to-D signal levels to establish a baseline for each channel and stores the baseline signals in the two-byte integer array B(2016) where 2016 $\approx$ 100 ms of data / 50 µs per point. The program then prints a heading on both screens and waits for the A-to-D converter to trigger or a command from the keyboard. A block cursor in the upper-left-hand corner of the active screen indicates the program is ready to receive data. When the data come, they are stored in the array D(2016), and the default command sequence (display a table of raw data, a graph of all signals vs time, and transmit a summary of the shot to the serial port) is automatically executed. The values of the quantities at the peak of the plasma current as well as the amp-seconds are stored (for the last 201 shots)

in the array L(16,200), and the digitizer is rearmed for the next shot. The value of the ion saturation current density is used to detect a baseline shot (signal < 100 mA /cm^2), but no special action is taken when a baseline occurs other than replace the date and time with <<<<< Baseline >>>>>.

The commands are mostly accessed by single keystrokes. Pressing the space bar or any other key not corresponding to a legal command brings up a menu of the command options. The commands are mostly self-explanatory or can be learned by experimentation. If for some reason, one suspects a faulty baseline, pressing "B" will retake the baseline. Don't do this while the core is being biased or immediately after a pulse. Pressing "C" allows one to change the time scale for the tables and graphs from the default value. "F" causes the shot to be saved on a floppy disk in drive B to be later read by the "Q" command or by another program (PCREVIEW). "H" halts computer operation. An "H" in the upper-left-hand corner of the screen reminds one that any data that come in will be ignored. "N" calls up a submenu that permits control of gas puff tracking. "P" allows one to program the automatic command sequence that is executed after each shot. Placing a "D" in the command sequence will cause a 5-second delay. The "D" can be repeated for a longer delay; "DDD" thus causes a 15-second delay. "T" transmits summary data for the current shot to the serial port. The "T" can be moved within but not deleted from the command sequence. The command "Z" allows one to reset the real-time clock that records the date and time of each shot. Pressing "*" prints the active screen (the one most recently written to) on the line printer. Most of the graph commands

require 2 or 3 keystrokes; "VFH", for example, graphs the variation of F with theta during the shot. One can exit the command sequence at any time by pressing "X", in which case the currently executing command will continue to completion, and the D-to-A converter will then rearm and await a trigger or additional keystroke.

The data available for display fall into two categories--raw data ("R") and derived data ("A"). The raw data are the values of the eight channels that are digitized with the baseline subtracted off and multiplied by the appropriate scale factor. For example, the plasma current 450 μs after the trigger is calculated from SC(1)*(D(9)-B(9)) where SC(1) = 100 kA / 204.8. The derived data are calculated from the raw data using formulas from Phys. Fluids 31, 2266 (1988) and listed below in MKS units (except for T_e which is in eV):

$$F = B_{tw} / \langle B_t \rangle$$

$$\theta = \mu_o I_p / 2\pi a \langle B_t \rangle$$

$$V_\ell = V_{pg} - 4R_o\theta(1-\theta^2)V_{tg}/a(8+3\theta^3) - 9\mu_o R_o(2+3\theta^2)dI_p/dt/8(6+\theta^2)$$

$$P_{oh} = V_\ell I_p$$

$$T_e = 0.023[R_o(5+6\theta^2)I_p/a^2V_\ell(10+\theta^2)]^{2/3}$$

$$\langle n \rangle = 1.6\times10^{15}PJ_{sat} / T_e^{1/2}[1-\exp(-V_{bias}/T_e)]$$

$$\tau = 6\pi^2 \langle n \rangle T_e e R_o a^2 \; / \; P_{oh}$$

Terms appearing above and not previously defined are the major radius, R_o = 1.5 m, the minor radius, a = 0.52 m, the probe bias voltage, V_{bias} = 83 volts, the charge of the electron, e = 1.6×10^{-19} C, and a profile factor P of the order of 3 that is determined by comparison of the density deduced from the probe with that deduced from the interferometer. The profile factor P is not precisely constant but may depend upon other quantities such as F or θ in a way yet to be determined.

The A-to-D circuit board also has two D-to-A analog output channels (DAC 0 and DAC 1). These have 12-bit resolution and provide an output of -10 to +10 volts at up to 5 mA. They can drive 0.5-μF loads and are short-circuit protected. They are used for feedback control of the gas puffing. No attempt is made to control the gas puff waveform during a shot (the board is incapable of simultaneous A-to-D and D-to-A operation), but rather the circuits provide a dc output that is changed just after the end of each shot. DAC 0 provides negative feedback to the gas fill (early puff) according to whether the last shot had more or less peak plasma current than the proceeding shot. DAC 1 provides negative feedback to the late gas puff according to whether the last shot had more or less amp-seconds than the proceeding shot. The signals are incremented by a given voltage step after each shot. The steps are in units of 0.5 mV, and can be changed by the user from the "N" submenu. The user can also change the signal on which DAC 0 is controlled. The puff valve tracking can be toggled on and off or

locked at the present level and unlocked from within the "N" submenu.

V. Calibration Method

Since the monitor system represents the primary standard by which fundamental electrical quantities such as magnetic field and plasma current are measured in MST, it is important that the calibrations be traceable to the National Bureau of Standards. To accomplish this, resistive shunts have been installed in the poloidal-field and toroidal-field primaries to allow the absolute measurement of these currents. The shunts have values of 0.0500 milliohms (B_p) and 0.0497 milliohms (B_t) (at 72°F) as measured to a precision of 0.04% on 3/25/88 by the University of Wisconsin Instrumentation Systems Center Standards Laboratory (702 ERB).

The internal Rogowskii loop is calibrated against the poloidal-field primary shunt (taking the turns ratio into account) for a case in which the iron core flux swing is sufficiently small that the magnetizing current is negligible. This is done either with a plasma discharge, or more directly, with a conductor placed inside the vacuum vessel in such a way that it links the Rogowskii loop. Similarly, the Rogowskii loop on the B_t secondary is calibrated against the toroidal-field primary shunt (taking the turns ratio into account) for a case in which the flux swing in the toroidal-field core is sufficiently small that the magnetizing current is negligible. The internal toroidal-flux

loop is calibrated against the B_t current-feed Rogowskii loops in such a way that F is made equal to one in the absence of a plasma. The flux in the iron core is measured by adjusting the time constant of its integrator to be exactly 0.2 second. The magnetizing current in the iron core is measured by subtracting the internal Rogowskii loop signal from the poloidal-field primary current (multiplied by the turns ratio) with a unity gain differential amplifier.

The Langmuir probe signal calibration is not critical, but is set for a given voltage output when the ion current density to the probe has a known value. The signal is used to deduce a density similar to what is read by the microwave interferometer. The calibration is, of course, profile dependent and thus should not be used for careful quantitative purposes.

Acknowledgments

Many people have helped with various aspects of the monitor system over the years. We would especially like to acknowledge W. Shen for constructing an early version of the circuit, E. Nilles, A Almagri, T Rempel and J. Laufenberg for providing diagnostic sensors, J. Beckstead fro interfacing the signals to the CAMAC and providing software transforms, and J. Sarff for integrator development and calibrations.

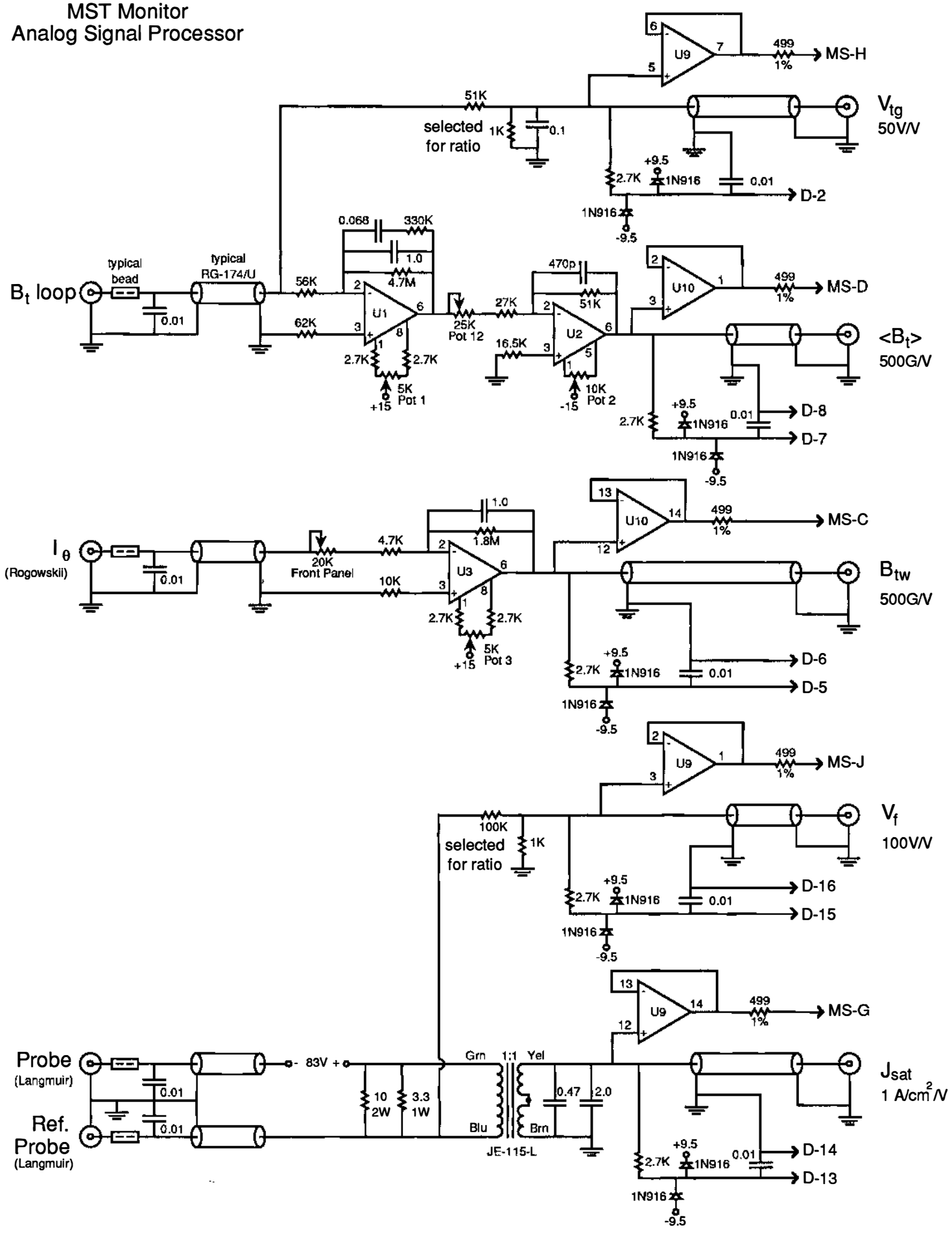

MST Monitor
Analog Signal Processor
U9
499
1%
MS-H
51K
selected for ratio
1K
0.1
V_{tg}
50V/V
+9.5
2.7K
1N916
0.01
D-2
1N916
-9.5
0.068
330K
1.0
4.7M
typical bead
typical RG-174/U
B_t loop
56K
62K
U1
0.01
25K Pot 12
470p
27K
51K
U10
MS-D
U2
16.5K
2.7K
5K Pot 1
+15
10K Pot 2
-15
$<B_t>$
500G/V
D-8
D-7
I_θ
(Rogowskii)
20K Front Panel
4.7K
10K
1.8M
U3
5K Pot 3
MS-C
B_{tw}
500G/V
D-6
D-5
MS-J
100K
V_f
100V/V
D-16
D-15
MS-G
Probe
(Langmuir)
Ref. Probe
(Langmuir)
- 83V +
10
2W
3.3
1W
Grn
1:1
Yel
Blu
Brn
JE-115-L
0.47
2.0
J_{sat}
1 A/cm^2/V
D-14
D-13

FIGURE 1

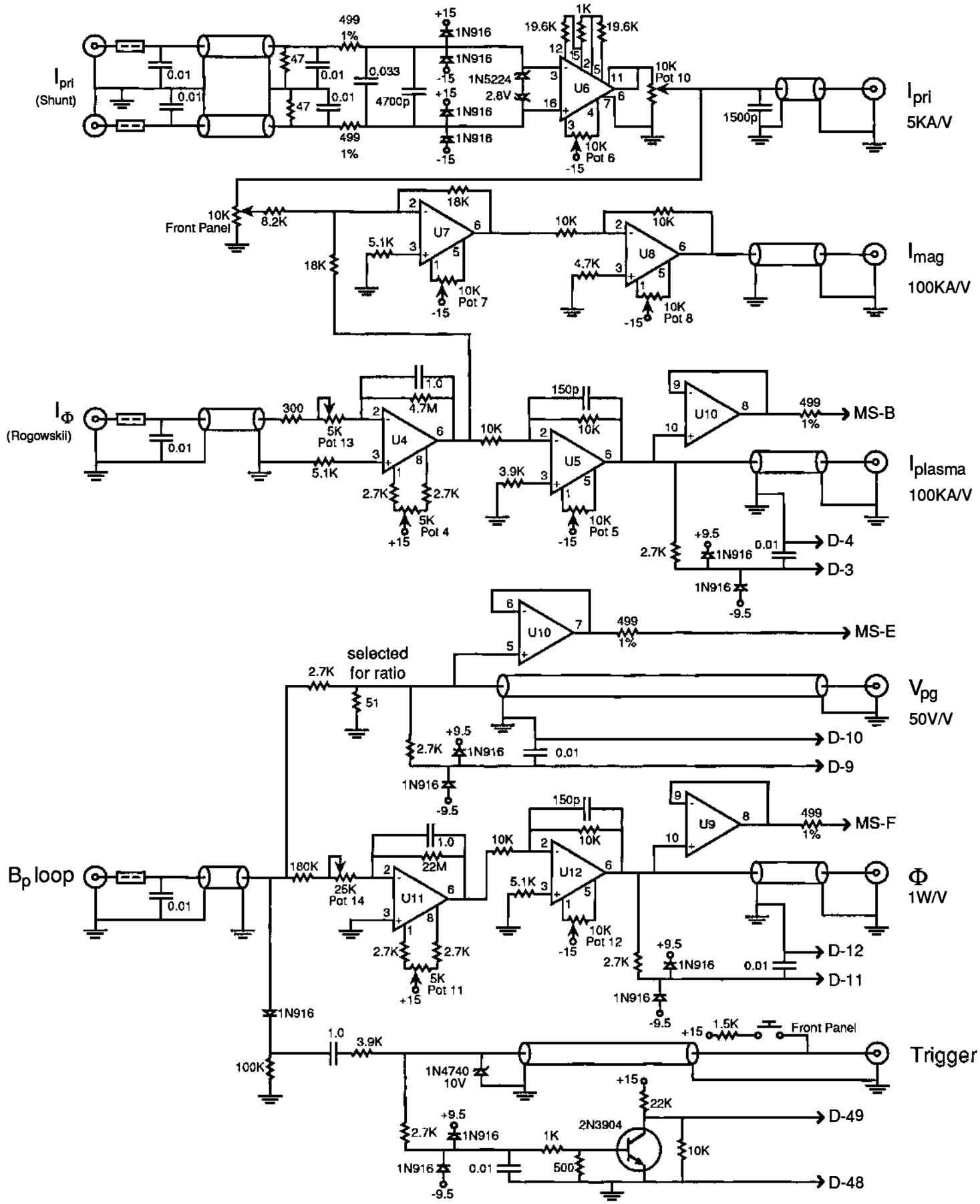

I_{pri} (Shunt)
I_{pri} 5KA/V
I_{mag} 100KA/V
I_{Φ} (Rogowskii)
I_{plasma} 100KA/V
MS-B
MS-E
MS-F
V_{pg} 50V/V
B_p loop
Φ 1W/V
Trigger
Front Panel
selected for ratio
D-4
D-3
D-10
D-9
D-12
D-11
D-49
D-48
U4
U5
U6
U7
U8
U9
U10
U11
U12
1N916
1N5224
2N3904
Pot 4
Pot 5
Pot 6
Pot 7
Pot 8
Pot 10
Pot 11
Pot 12
Pot 13
Pot 14

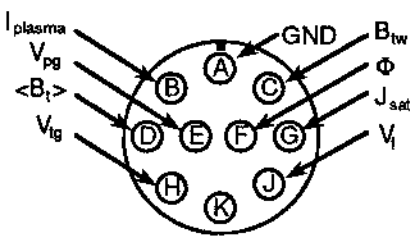
Figure 2

CONNECTORS TO ANALOG TO DIGITAL CONVERTORS

50 Pin D-Connector at IBM			
CH 0	1	2	CH 8 / CH 0 RET
CH 1	3	4	CH 9 / CH 1 RET
CH 2	5	6	CH10 / CH 2 RET
CH 3	7	8	CH11 / CH 3 RET
CH 4	9	10	CH12 / CH 4 RET
CH 5	11	12	CH13 / CH 5 RET
CH 6	13	14	CH14 / CH 6 RET
CH 7	15	16	CH15 / CH 7 RET
A GND	17	18	AMP LOW
+15V / +12V	19	20	-15V / -12V
P GND	21	22	DAC 0 OUT
DAC 0 GND	23	24	DAC 1 OUT
DAC 1 GND	25	26	D GND
D GND	27	28	DIO 0, BIT 1
DIO 0, BIT 1	29	30	DIO 0, BIT 1
DIO 0, BIT 1	31	32	D GND
DIO 0, BIT 1	33	34	DIO 0, BIT 1
DIO 0, BIT 1	35	36	DIO 0, BIT 1
D GND	37	38	DIO 0, BIT 1
DIO 0, BIT 1	39	40	DIO 0, BIT 1
DIO 0, BIT 1	41	42	D GND
DIO 0, BIT 1	43	44	DIO 0, BIT 1
DIO 0, BIT 1	45	46	DIO 0, BIT 1
D GND	47	48	D GND
EXT TRIGGER	49	50	EXT CLOCK

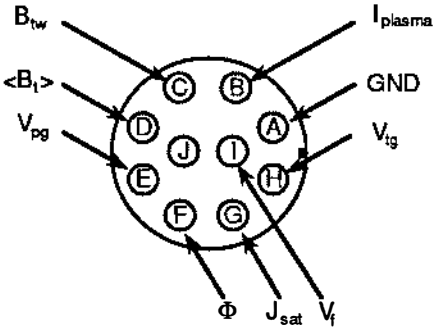
50 Pin D-Connector to IBM			
GND	1	2	V_{tg}
I_{plasma}	3	4	I_{plasma} GND
B_t	5	6	B_t
$<B_t$	7	8	$<B_t$
V_{pg}	9	10	V_{pg} GND
Φ	11	12	Φ GND
J_{sat}	13	14	J_{sat} GND
V_f	15	16	V_f GND
GND	17	18	
	19	20	
	21	22	DAC0
DAC0 GND	23	24	DAC1
DAC1 GND	25	26	
GND	27	28	
	29	30	
	31	32	
	33	34	
	35	36	
	37	38	
	39	40	
	41	42	
	43	44	
	45	46	
	47	48	Trig GND
TRIG	49	50	

MS Connector to CAMAC



solder side

MS Connector at CAMAC



solder side

MAIN BOARD COMPONENT LOCATION

Potentiometers	
1	$<B_t$ integrator offset
2	$<B_t$
3	B_t
4	I_{plasma} integrator offset
5	I_{plasma} inverter offset
6	I_{pri} offset
7	I_{mag} subtractor offset
8	I_{mag} inverter offset
9	$<B_t$
10	I_{pri} gain
11	Φ integrator offset
12	Φ inverter offset
13	I_{plasma} gain
14	Φ gain

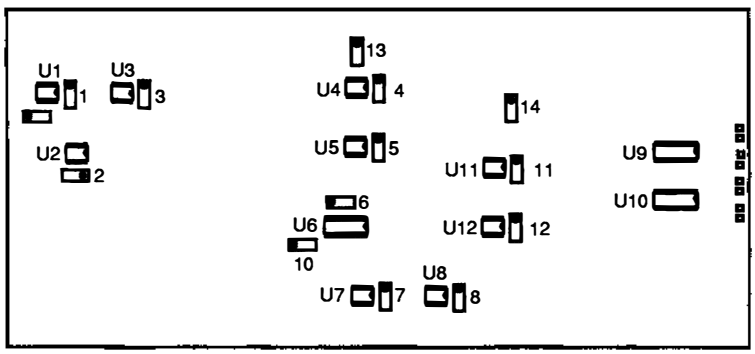
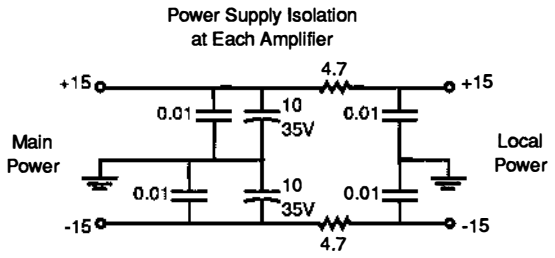


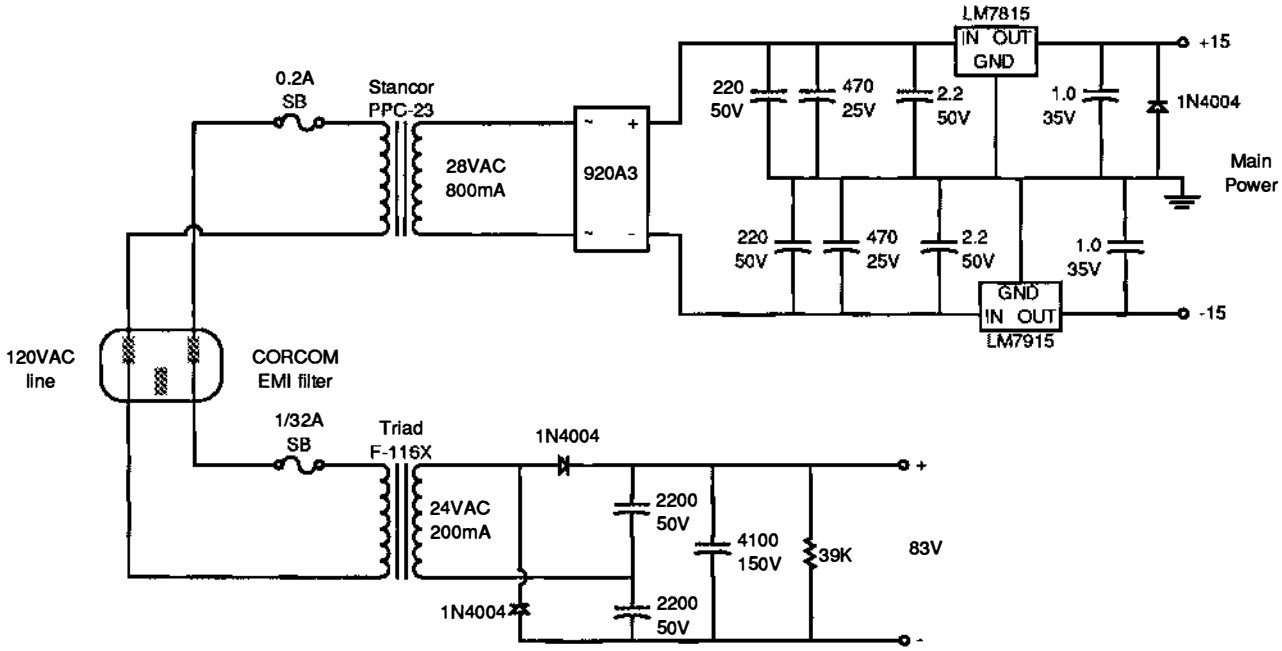
Figure 3

OPERATIONAL AMPLIFIERS

Part Number	Amplifier Type	Power Pins +	-
U1	OP-27-E	7	4
U2	LF13741	7	4
U3	OP-27-E	7	4
U4	OP-27-E	7	4
U5	LF13741	7	4
U6	AD625A	9	8
U7	LF13741	7	4
U8	LF13741	7	4
U9	LM324	4	11
U10	LM324	4	11
U11	OP-27-E	7	4
U12	OP-27-E	7	4



POWER SUPPLIES



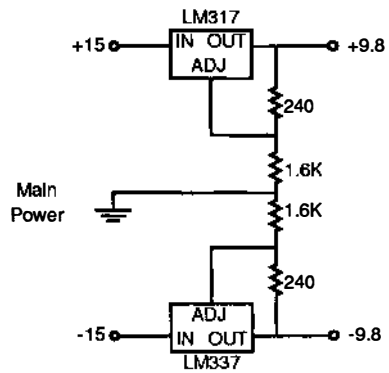


Figure 4

```
10      'I B M    P C    M S T    M O N I T O R    P R O G R A M
20 A1$="IBM PC MST  MONITOR PROGRAM"
30 A2$="25 Nov 1988 Version"
40 A3$="Copyright (c) 1985 by J.C. Sprott"
50      'INITIALIZE PARAMETERS
60 KEY OFF: CLS: PRINT"Initializing...": ON ERROR GOTO 2580
70 DEFINT A-Z: DIM B(2016),D(2016),PS(2): DAT=748: CMD=749
80 NC=8: REM Number of channels (1-15)
90 NA=7: REM Number of additional derived data
100 FS=20: REM Default number of msec full scale (1-100)
110 SA=20: REM Number of samples / msec
120 NL=201: REM Number of old shots to store
130 CS$="RG.T": REM Default command sequence
140 RO!=1.5: A!=.52: VBIAS!=83: P!=3: REM MST-specific quantities
150 DEF SEG=&HF000: YR=10*PEEK(-5)+PEEK(-4)-528
160 DIM
CHL$(16),CHS$(16),SC!(16),UNIT$(16),A!(8),X(259),Y(259),L(16,200)
: REM 16=NC+NA+1, 8=NA+1, 259=100*SA/NC+NC+1, 200=NL-1
170 DIM D!(9): D!(0)=1: D!(1)=2: D!(2)=5: D!(3)=5: D!(4)=5: FOR
I=5 TO 9: D!(I)=10: NEXT
180 DIM PUFF$(1): PUFF$(0)="off": PUFF$(1)="on": SS=64: IP0=1
190 CHL$(0)="Time": CHS$(0)="Time": SC!(0)=1/SA: UNIT$(0)="msec"
200 FOR I=1 TO NC: CHL$(I)="Channel"+STR$(I):
CHS$(I)="Ch"+STR$(I): SC!(I)=1000: UNIT$(I)="mV": NEXT:
CH$=LEFT$("123456789ABCDEF",NC)+"/."
210 GOSUB 2250
220 CHL$(NC+NA+1)="Integral of "+CHS$(1):
CHS$(NC+NA+1)=CHS$(1)+"*t": SC!(NC+NA+1)=40*SC!(0)*SC!(1):
UNIT$(NC+NA+1)=UNIT$(1)+"-"+UNIT$(0): GOSUB 5520: GOSUB 260:
GOSUB 3410: SSOLD=SS: SSGN=SGN(SS): IF SSGN=0 THEN SSGN=1
230 IF UNIT$(NC+NA+1)="kA-msec" THEN UNIT$(NC+NA+1)="Amp-sec"
240 GOSUB 290: GOTO 240
250     'TAKE BASELINE
260 TRIG=0: DEF SEG=0: SGM=PEEK(&H510)+256*PEEK(&H511):
OFFSET=VARPTR(B(1)): GOSUB 1120: GOSUB 4980
270 RETURN
280     'TAKE DATA
290 TRIG=1: DEF SEG=0: SGM=PEEK(&H510)+256*PEEK(&H511):
OFFSET=VARPTR(D(1)): GOSUB 1120: GOSUB 4980
300 IF Q$<>"" THEN GOSUB 500: GOTO 290
310 GOSUB 420: GOSUB 460: NS=NS+1: NS$=DATE$+"  "+TIME$: GOSUB
3260: GOSUB 3040: GOSUB 3350: GOSUB 5020
320 IF NS>9999 THEN NS=NS-10000: GOTO 320
330 GOSUB 460: SCREEN 0: WIDTH 40: LOCATE 9,15: COLOR 12:
PRINT"SHOT:";NS: LOCATE 12,16: COLOR 10: PRINT
INT(L(1,IL)*SC!(1)+.5);UNIT$(1): LOCATE 15,13,0: COLOR 9: PRINT
INT(L(NC+NA+1,IL)*SC!(NC+NA+1)+.5);UNIT$(NC+NA+1)
340 IF DATE$<>DAY$ THEN DAY$=DATE$: NS0=NS: GOSUB 5920
350 IF L(IP0,IL)*SSGN>=LBEST*SSGN THEN LBEST=L(IP0,IL): BESTIL=IL
360 FOR ICS=1 TO LEN(CS$): Q$=MID$(CS$,ICS,1)
370 IF (Q$="G" OR Q$="g" OR Q$="L" OR Q$="l" OR Q$="N" OR Q$="n")
AND ICS<LEN(CS$) THEN ICS=ICS+1: R$=MID$(CS$,ICS,1)
380 IF (Q$="U" OR Q$="u" OR Q$="V" OR Q$="v") AND ICS<LEN(CS$)-1
THEN ICS=ICS+2: R$=MID$(CS$,ICS-1,1): S$=MID$(CS$,ICS,1)
```

```
390 GOSUB 500: NEXT
400 RETURN
410     'SWITCH TO TEXT MODE
420 IF YR=82 THEN DEF SEG=0: POKE &H410,(PEEK(&H410) OR &H30)
430 SCREEN 0: WIDTH 80: COLOR 2,0: LOCATE,,0: CLS
440 RETURN
450     'SWITCH TO GRAPHICS MODE
460 IF YR=82 THEN DEF SEG=0: POKE &H410,(PEEK(&H410) AND &HCF) OR
&H10
470 SCREEN 1,0,0,0: CLS
480 RETURN
490     'EXECUTE COMMAND SEQUENCE
500 IF Q$="" THEN 760 ELSE IF ASC(Q$)>96 THEN Q$=CHR$(ASC(Q$)-32)
510 IF Q$="A" THEN GOSUB 2770: GOTO 750
520 IF Q$="B" THEN GOSUB  260: GOTO 750
530 IF Q$="C" THEN GOSUB 1820: GOTO 750
540 IF Q$="D" THEN GOSUB 1980: GOTO 750
550 IF Q$="E" THEN GOSUB 2440: GOTO 500
560 IF Q$="F" THEN GOSUB 2060: GOTO 750
570 IF Q$="G" THEN GOSUB  930: GOTO 750
580 IF Q$="H" THEN GOSUB 3950: GOTO 500
590 IF Q$="I" THEN GOSUB 5920: GOTO 750
600 IF Q$="L" THEN GOSUB 3080: GOTO 750
610 IF Q$="M" THEN GOSUB 2010: GOTO 750
620 IF Q$="N" THEN GOSUB 5580: GOTO 750
630 IF Q$="P" THEN GOSUB 1880: GOTO 760
640 IF Q$="Q" THEN GOSUB 2130: GOTO 500
650 IF Q$="R" THEN GOSUB  830: GOTO 750
660 IF Q$="S" THEN GOSUB 1750: GOTO 750
670 IF Q$="T" THEN GOSUB 4160: GOTO 750
680 IF Q$="U" THEN GOSUB 3510: GOTO 750
690 IF Q$="V" THEN GOSUB 3510: GOTO 750
700 IF Q$="X" THEN GOSUB 3990: GOTO 760
710 IF Q$="Z" THEN GOSUB  780: GOTO 750
720 IF Q$="*" THEN GOSUB 4020: GOTO 750
730 IF Q$=CHR$(3) THEN CLS: GOSUB 420: GOSUB 5480: CLS: END
740 GOSUB 1590
750 Q$=INKEY$: IF Q$<>"" THEN 500
760 RETURN
770     'RESET DATE AND TIME
780 GOSUB 420
790 PRINT"  Date Now: ";DATE$: INPUT"  New Date";Q$: IF Q$<>""
THEN DATE$=Q$: CLS
800 PRINT"  Time Now: ";TIME$: INPUT"  New Time";Q$: IF Q$<>""
THEN TIME$=Q$
810 RETURN
820     'DISPLAY RAW DATA
830 GOSUB 420: GOSUB 5340: NN=INT(72/NC): AA$=STRING$(NN,"#")
840 LOCATE 1,4,0: PRINT HEAD$;
850 PRINT: FOR I=0 TO NC: PRINT TAB(NN*I+8-LEN(CHS$(I)))
CHS$(I);: NEXT: PRINT
860 FOR J=1 TO 20: PRINT USING"#####.#";J*FS/20;:
IO=(J*FS/NC)*(SA/20): IO=IO*NC
870 FOR I=1 TO NC
```

```
880 PRINT USING AA$;(D(IO+I)-B(IO+I))*SC!(I);: NEXT: PRINT
890 NEXT: FOR I=0 TO NC: PRINT TAB(NN*I+8-LEN(UNIT$(I)))
UNIT$(I);: NEXT
900 IF NEEDBASE THEN LOCATE 25,4: PRINT"*** Take a baseline
shot.";
910 RETURN
920     'GRAPH CHANNEL IY VS TIME
930 IF R$="" THEN 980
940 IF ASC(R$)>96 THEN R$=CHR$(ASC(R$)-32)
950 IF R$="." THEN GOSUB 4390: GOTO 1100
960 IY=VAL(R$): IF IY=0 THEN IY=INSTR(CH$,R$)
970 IF IY>0 AND IY<=NC+NA THEN 1030
980 GOSUB 420: PRINT"H";TAB(32)"GRAPH OPTIONS": PRINT
990 FOR I=1 TO NC+NA: PRINT TAB(29)MID$(CH$,I,1);": ";CHL$(I):
NEXT
1000 PRINT TAB(29)".: Everything"
1010 LOCATE 25,32: PRINT"Choose graph: ";
1020 R$=INKEY$: IF R$="" THEN 1020 ELSE CLS: GOTO 940
1030 NP=SA*FS/NC: IX=0: GOSUB 5300: SCY!=SC!(IY): SCX!=NC/SA:
XMIN=0: XMAX=SA*FS/NC
1040 FOR I=0 TO NP: X(I)=I: NEXT
1050 IF IY<=NC THEN 1090
1060 FOR I=0 TO NP: IO=NC*I: GOSUB 2880
1070 IF (ABS(A!(IY-NC))+1)/SCY!>32766 THEN Y(I)=0 ELSE
Y(I)=A!(IY-NC)/SCY!
1080 NEXT: GOSUB 1280: GOTO 1100
1090 FOR I=0 TO NP: Y(I)=D(NC*I+IY)-B(NC*I+IY): NEXT: GOSUB 1280
1100 RETURN
1110     'READ A-TO-D CONVERTER
1120 PAGE=INT((SGM+OFFSET/16)/&H1000+.5): ADDRESS!=(SGM-
PAGE*&H1000)*16+OFFSET+.5: IF ADDRESS!>61000! THEN PRINT"  ***
Array spans two memory pages.": STOP
1130 BASEH=INT(ADDRESS!/256): BASEL=INT(ADDRESS!-256*BASEH)
1140 OUT CMD,15: TEMP=INP(DAT): GOSUB 4960: WAIT CMD,4: OUT
CMD,1: GOSUB 4960: WAIT CMD,4: OUT CMD,3
1150 IF (INP(CMD) AND 112)=112 THEN 1260
1160 GOSUB 4960: WAIT CMD,2,2: OUT DAT,800/SA: GOSUB 4960: WAIT
CMD,2,2: OUT DAT,0
1170 OUT 11,69: OUT 12,0: OUT 2,BASEL: OUT 2,BASEH: OUT 3,191:
OUT 3,15: OUT 131,PAGE: OUT 10,1
1180 GOSUB 4960: WAIT CMD,4: OUT CMD,13: GOSUB 4960: WAIT
CMD,2,2: OUT DAT,0
1190 GOSUB 4960: WAIT CMD,2,2: OUT DAT,1: GOSUB 4960: WAIT
CMD,2,2: OUT DAT,NC
1200 GOSUB 4960: WAIT CMD,2,2: OUT DAT,5: GOSUB 4960: WAIT
CMD,2,2: OUT DAT,5
1210 LOCATE 1,1,0: PRINT CHR$(219);: GOSUB 4960: WAIT CMD,4: OUT
CMD,30+128*TRIG
1220 Q$=INKEY$: IF Q$<>"" THEN 1250
1230 IF INP(CMD) AND 4 THEN 1240 ELSE 1220
1240 STATUS=INP(CMD): IF (STATUS AND 128) THEN CLS: PRINT"  ***
A-to-D error";: GOSUB 4220: GOTO 1120: ELSE BEEP
1250 OUT CMD,15: TEMP=INP(DAT): GOSUB 4960: WAIT CMD,4: OUT
CMD,1: LOCATE 1,1: PRINT" ";
```

```
1260 RETURN
1270      'PLOT GRAPH
1280 YMIN=0: YMAX=0
1290 FOR I=0 TO NP
1300 IF Y(I)>YMAX THEN YMAX=Y(I)
1310 IF Y(I)<YMIN THEN YMIN=Y(I)
1320 NEXT: GOSUB 460: IF Q$="L" THEN COLOR 8,1 ELSE IF Q$="U"
THEN COLOR 1,1: ELSE COLOR 8,0
1330 IF NS<=0 OR YMAX=YMIN OR XMAX=XMIN THEN PRINT"  *** No data
to graph for": PRINT"      ";TITLE$: PRINT"      vs ";XLABEL$:
GOTO 1560
1340 XOLD=0: IF Q$="G" THEN IL=(NS-1) MOD NL: YOLD=L(IY,IL): IF
YOLD THEN XOLD=L(0,IL)/NC: IF XOLD<=NP THEN YMAX=-
YMAX*(YMAX>=YOLD)-YOLD*(YMAX<YOLD): YMIN=-YMIN*(YMIN<=YOLD)-
YOLD*(YMIN>YOLD)
1350 I=0: D!=ABS(YMAX*SCY!): IF ABS(YMIN)>ABS(YMAX) THEN
D!=ABS(YMIN*SCY!)
1360 IF INT(D!)>9 THEN D!=D!/10: I=I+1: GOTO 1360
1370 D!=D!(INT(D!))*10^I: YMAX=(INT(D!)/SCY!)*SGN(YMAX):
YMIN=(INT(D!)/SCY!)*SGN(YMIN): IF YMAX=YMIN THEN 1330
1380 IF YMIN>YMAX THEN YMIN=-YMIN: YMAX=-YMAX
1390 LOCATE 1,21-LEN(TITLE$)/2: PRINT TITLE$: DY!=CSNG(YMAX)-
CSNG(YMIN): DX!=CSNG(XMAX)-CSNG(XMIN)
1400 IF Q$="U" THEN FOR I=0 TO NP: CIRCLE (20+280!*(X(I)-
XMIN)/DX!,190-180!*(Y(I)-YMIN)/DY!),2,3+(I=NP): NEXT: GOTO 1450
1410 IF Q$="U" THEN FOR I=0 TO NP: CIRCLE (20+280!*(X(I)-
XMIN)/DX!,190-180!*(Y(I)-YMIN)/DY!),2,3+(I=NP): NEXT: GOTO 1450
1420 PSET(20+280!*(X(0)-XMIN)/DX!,190-180!*(Y(0)-YMIN)/DY!),2
1430 FOR I=1 TO NP: LINE -(20+280!*(X(I)-XMIN)/DX!,190-
180!*(Y(I)-YMIN)/DY!),2
1440 NEXT: IF XOLD THEN CIRCLE (20+280!*(XOLD-XMIN)/DX!,190-
180!*(YOLD-YMIN)/DY!),2
1450 LINE (20,10)-(300,190),1,B
1460 IF Q$="L" AND NS0>XMIN AND NS0<XMAX THEN LINE (20+280!*(NS0-
XMIN)/DX!,10)-(20+280!*(NS0-XMIN)/DX!,190)
1470 IF SGN(YMIN)*SGN(YMAX)<0 THEN PSET(20,190+180!*YMIN/DY!),1:
LINE -(300,190+180!*YMIN/DY!),1: LOCATE (195+180!*YMIN/DY!)/8,2:
PRINT"0";
1480 IF SGN(XMIN)*SGN(XMAX)<0 THEN PSET(20-280!*XMIN/DX!,190),1:
LINE -(20-280!*XMIN/DX!,10),1
1490 FOR I=1 TO 9: FOR J=1 TO 4: PSET(20+J,10+18*I),1: PSET(300-
J,10+18*I),1: PSET(20+28*I,10+J),1: PSET(20+28*I,190-J),1: NEXT
J,I: PSET(20+280!*(X(NP)-XMIN)/DX!,190-180!*(Y(NP)-YMIN)/DY!),2
1500 LOCATE 2,1: PRINT STR$(INT(YMAX*SCY!+.5));
1510 LOCATE 24,1: PRINT STR$(INT(YMIN*SCY!+.5));
1520 LOCATE 25,3-.4*LEN(STR$(INT(XMIN*SCX!+.5))): PRINT
STR$(INT(XMIN*SCX!+.5));: LOCATE 25,21-LEN(XLABEL$)/2-
LEN(STR$(INT(XMAX*SCX!+.5)))/5: PRINT XLABEL$;
1530 LOCATE 25,38-.6*LEN(STR$(INT(XMAX*SCX!+.5))): PRINT
STR$(INT(XMAX*SCX!+.5));
1540 CR$=CHS$(IY)+"  "+UNIT$(IY): FOR I=1 TO LEN(CR$): LOCATE
I+12-INT(LEN(CR$)/2),1: PRINT MID$(CR$,I,1);: NEXT
1550 IF (R$="F" OR R$="f") AND (S$="H" OR S$="h") THEN GOSUB 5400
```

```
1560 R$="": S$="": IF Q$="L" OR Q$="U" THEN XMIN=NS-NP: XMAX=NS
ELSE XMIN=0
1570 RETURN
1580     'DISPLAY COMMAND OPTIONS
1590 GOSUB 420: LOCATE 1,32,0: PRINT"COMMAND OPTIONS": PRINT
1600 PRINT"A: Display derived data               N: Change puff
valve tracking"
1610 PRINT"B: Retake baseline                    P: Program
command sequence"
1620 PRINT"C: Change time scale                  Q: Recall old
data from disk"
1630 PRINT"D: Delay command by 5 sec             R: Display raw
data"
1640 PRINT"E: Examine analog inputs              S: Change shot
counter"
1650 PRINT"F: Save shot on floppy disk           T: Transmit data
to serial port"
1660 PRINT"G: Graph data vs time                 U: Graph history
of Y vs X"
1670 PRINT"H: Halt operations (standby)          V: Graph
variation of Y vs X"
1680 PRINT"I: Initialize best shot               X: Exit command
sequence"
1690 PRINT"L: Graph history of data              Z: Update date
and time"
1700 PRINT"M: Print message on screen            *: Print screen
on line printer"
1710 IF Q$="P" THEN 1730
1720 LOCATE 22,32,1: PRINT"Choose option:";: IF ICS>LEN(CS$) THEN
GOSUB 290
1730 RETURN
1740     'CHANGE SHOT COUNTER
1750 GOSUB 420: PRINT"Current shot number:";NS
1760 LINE INPUT"New shot number: ";Q$: CLS
1770 IF VAL(Q$)<0 OR VAL(Q$)>9999! THEN PRINT"  *** Must be in
the range 0-9999": GOTO 1760
1780 IF Q$<>"" THEN NS=VAL(Q$)
1790 LOCATE 1,4: PRINT"New shot number is";NS
1800 RETURN
1810     'CHANGE TIME SCALE
1820 GOSUB 420: PRINT"Full scale now:";FS;"msec"
1830 LINE INPUT"New full scale: ";Q$: CLS: IF VAL(Q$) THEN
FS=VAL(Q$)
1840 IF FS<1 OR FS>100 THEN PRINT"  *** Must be in the range 1-
100": GOTO 1830
1850 LOCATE 1,4: PRINT"New full scale is";FS;"msec"
1860 RETURN
1870     'PROGRAM COMMAND SEQUENCE
1880 CLS: GOSUB 460: WIDTH 80: PRINT TAB(32)"GRAPH OPTIONS":
PRINT
1890 PRINT TAB(29)"M: Time of maximum ";CHS$(1)
1900 FOR I=1 TO NC+NA+1: PRINT TAB(29)MID$(CH$,I,1);": ";CHL$(I):
NEXT
1910 PRINT TAB(29)".: Everything"
```

```
1920 GOSUB 1590: LOCATE 1,1: PRINT"H";: LOCATE 23,1: PRINT"Old
command sequence: ";CS$;: LOCATE 24,1,1: PRINT"New command
sequence: ";
1930 GOSUB 3990: LINE INPUT CN$: CLS: GOSUB 3410: Q$=""
1940 IF CN$<>"" THEN CS$=CN$
1950 IF INSTR(CS$,"T")=0 AND INSTR(CS$,"t")=0 THEN CS$=CS$+"T"
1960 RETURN
1970     'DELAY COMMAND BY 5 SEC
1980 LOCATE 1,1: PRINT"D";: FOR I=1 TO 92: SOUND 32767,1: NEXT:
LOCATE 1,1: PRINT" ";
1990 RETURN
2000     'PRINT MESSAGE ON SCREEN
2010 GOSUB 420: LOCATE,,1: LINE INPUT"Message? ";M$: CLS: IF
LEN(M$)<201 THEN MM$=STRING$(39," ")+M$ ELSE MM$=M$
2020 GOSUB 460: SCREEN 0: COLOR 13,0,0: PRINT"H": LOCATE 25,3,0:
PRINT"Press any key to resume taking data";
2030 FOR I=1 TO LEN(MM$): SOUND 32767,2: LOCATE 11,1,0: PRINT
MID$(MM$,I,39)+STRING$(40," "): NEXT I: Q$=INKEY$: IF Q$="" THEN
2030 ELSE GOSUB 3410
2040 RETURN
2050     'SAVE DATA ON FLOPPY DISK
2060 IF NS=SAVED THEN 2110 ELSE LOCATE 1,1: PRINT"F";
2070 OPEN"O",1,"B:HEADINGS.DAT": WRITE#1,NC,NA,CH$
2080 FOR I=1 TO NC+NA: WRITE#1,CHL$(I),CHS$(I),SC!(I),UNIT$(I):
NEXT: CLOSE
2090 OPEN"B:DATAFILE.DAT" FOR APPEND AS #1: PRINT
#1,NS;NS$;",";FS
2100 FOR I=0 TO SA*FS+NC: PRINT#1,D(I)-B(I);: NEXT: CLOSE: LOCATE
1,1: PRINT" ";: SAVED=NS
2110 RETURN
2120     'RECALL OLD DATA FROM DISK
2130 CLS: OPEN"I",1,"B:HEADINGS.DAT": INPUT#1,NC,NA,CH$
2140 FOR I=1 TO NC+NA: INPUT#1,CHL$(I),CHS$(I),SC!(I),UNIT$(I):
NEXT: CLOSE
2150 NSAVE=NS: NSAVE$=NS$: FSAVE=FS: OPEN"I",1,"B:DATAFILE.DAT"
2160 IF EOF(1)<>0 OR Q$="X" OR Q$="x" THEN 2220 ELSE
INPUT#1,NS,NS$,FS
2170 FOR I=0 TO SA*FS+NC: INPUT#1,D: D(I)=D+B(I): NEXT: GOSUB
420: GOSUB 460
2180 FOR ICS=1 TO LEN(CS$): Q$=MID$(CS$,ICS,1)
2190 IF (Q$="G" OR Q$="g" OR Q$="L" OR Q$="l") AND ICS<LEN(CS$)
THEN ICS=ICS+1: R$=MID$(CS$,ICS,1)
2200 IF(Q$="U" OR Q$="u" OR Q$="V" OR Q$="v") AND ICS<LEN(CS$)-1
THEN ICS=ICS+2: R$=MID$(CS$,ICS-1,1): S$=MID$(CS$,ICS,1)
2210 GOSUB 500: NEXT: GOTO 2160
2220 CLOSE: NS=NSAVE: NS$=NSAVE$: FS=FSAVE: NSAVE=0: RESTORE:
RESTORE 2260: GOSUB 2250
2230 RETURN
2240     'REDEFINE LABELS
2250 CH$="IWBVCJSGFHLPENT/": FOR I=1 TO NC+NA: READ
CHL$(I),CHS$(I),SC!(I),UNIT$(I): NEXT
2260 DATA Plasma Current, Ip, 100, kA
2270 DATA Toroidal Field @ Wall, Btw, 500, Gauss
2280 DATA Av Toroidal Field, <Bt>, 500, Gauss
```

```
2290 DATA Poloidal Gap Voltage, Vpg, 50, Volts
2300 DATA Core Flux, Flux, 200, mWb
2310 DATA Ion Sat Current, Jsat, 1000, "mA/cmy"
2320 DATA Floating Potential, Vf, 100, Volts
2330 DATA Toroidal Gap Voltage, Vtg, 50, Volts
2340 DATA Field Reversal Parameter, F, .001, ""
2350 DATA Pinch Parameter, Theta, .01, ""
2360 DATA Loop Voltage, Vloop, .1, Volts
2370 DATA Ohmic Input Power, Poh, .01, MW
2380 DATA Conductivity Temperature, Te, .1, eV
2390 DATA Average Density, <n>, .1, e12/cc
2400 DATA Confinement Time, Tau, 1, usec
2410 FOR I=1 TO NC: SC!(I)=SC!(I)/204.8: NEXT
2420 RETURN
2430    'EXAMINE ANALOG INPUTS
2440 GOSUB 420: PRINT"H";: LOCATE 1,25: PRINT"ANALOG INPUT FOR
EACH CHANNEL"
2450 FOR I=0 TO 15: LOCATE I+4,22,0: PRINT"Channel";STR$(I);:
LOCATE I+4,44: PRINT"VOLTS": NEXT
2460 FOR I=1 TO NC: LOCATE I+4,52: PRINT "[";CHS$(I);"]": NEXT
2470 LOCATE 25,22: PRINT"Press any key to resume taking data";
2480 FOR I=0 TO 15
2490 OUT CMD,15: TEMP=INP(DAT): GOSUB 4960: WAIT CMD,4: OUT
CMD,1: GOSUB 4960: WAIT CMD,4: OUT CMD,12
2500 IF (INP(CMD) AND 112)=112 THEN 2550
2510 GOSUB 4960: WAIT CMD,2,2: OUT DAT,0: GOSUB 4960: WAIT
CMD,2,2: OUT DAT,I
2520 GOSUB 4960: WAIT CMD,5: LOW=INP(DAT): GOSUB 4960: WAIT
CMD,5: HIGH=INP(DAT)
2530 GOSUB 4960: WAIT CMD,4: STATUS=INP(CMD): IF (STATUS AND 128)
THEN CLS: PRINT"  *** A-TO-D ERROR";: GOSUB 4220: GOTO 2560
2540 LOCATE I+4,34: PRINT USING"###.###";(256*HIGH+LOW)/204.8-
10;: NEXT
2550 Q$=INKEY$: IF Q$="" THEN 2480 ELSE CLS
2560 RETURN
2570    'TRAP ERRORS
2580 GOSUB 420: LOCATE 1,3: PRINT"*** ";
2590 IF ERR=71 AND ERL>2040 AND ERL<2230 THEN PRINT"Insert disk
in right-hand drive and close drive door.": Q$="": RESUME 2230
2600 IF ERR=72 AND ERL>2040 AND ERL<2230 THEN PRINT"Bad or
improperly installed disk in right-hand drive.": Q$="": RESUME
2230
2610 IF ERR=57 AND ERL>1600 AND ERL<2230 THEN PRINT"Disk in
right-hand drive is not formatted.": RESUME 2230
2620 IF ERR=25 AND ERL>4000 AND ERL<4140 THEN PRINT"Line printer
not operational.": RESUME 4140
2630 IF ERR=27 AND ERL>4000 AND ERL<4140 THEN PRINT"Line printer
out of paper.": RESUME 4140
2640 IF ERR=55 AND ERL>2040 AND ERL<2110 THEN PRINT"Can't write
to floppy.": RESUME 2110
2650 IF ERR=61 AND ERL>2040 AND ERL<2110 THEN PRINT"Disk in
right-hand drive full.  Replace with new data disk.": RESUME 2110
```

```
2660 IF ERR=53 AND ERL>2110 AND ERL<2230 THEN PRINT"Disk in
right-hand drive does not have readable data on it.": Q$="":
RESUME 2230
2670 IF ERR=55 AND ERL>2110 AND ERL<2230 THEN PRINT"Can't read
from floppy.": CLOSE: Q$="": RESUME 2230
2680 IF ERR=62 AND ERL>2110 AND ERL<2230 THEN PRINT"All data from
disk has been read.": CLOSE: RESUME 2230
2690 IF ERR=57 AND ERL>4140 AND ERL<4210 THEN PRINT"Error in data
transmission.": RESUME 4190
2700 IF ERR=53 AND ERL>5500 AND ERL<5560 THEN PRINT"CONFIG.DAT
file not found on disk.": RESUME 5560
2710 IF ERR=62 AND ERL>5500 AND ERL<5560 THEN PRINT"CONFIG.DAT
file is defective.": RESUME 5560
2720 IF ERR=5 AND ERL=790 THEN PRINT"Improper response": RESUME
790
2730 IF ERR=5 AND ERL=800 THEN PRINT"Improper response": RESUME
800
2740 PRINT"Untrapped error";ERR;"in line";ERL: IF ERL>240 THEN
GOSUB 1980: RESUME 2750 ELSE ON ERROR GOTO 0
2750 RETURN
2760     'DISPLAY DERIVED DATA
2770 GOSUB 420: IF NEEDBASE THEN 2850
2780 NN=INT(36/NA+4.9): AA$=STRING$(NN-2,"#")+".#"
2790 LOCATE 1,4,0: GOSUB 5340: PRINT HEAD$;: PRINT
2800 PRINT"   Time";
2810 FOR I=1 TO NA: PRINT TAB(NN*I+8-LEN(CHS$(NC+I)))
CHS$(NC+I);: NEXT: PRINT
2820 FOR J=1 TO 20: PRINT USING"#####.#";J*FS/20;:
IO=J*FS*SA/20/NC: IO=IO*NC: GOSUB 2880
2830 FOR I=1 TO NA: PRINT USING AA$;A!(I);: NEXT: PRINT: NEXT:
PRINT"   msec";
2840 FOR I=1 TO NA: PRINT TAB(NN*I+8-LEN(UNIT$(NC+I)))
UNIT$(NC+I);: NEXT
2850 IF NEEDBASE THEN LOCATE 25,4: PRINT"*** Take a baseline
shot.";
2860 RETURN
2870     'CALCULATE ADDITIONAL DERIVED DATA ARRAY A!(NA) AT STEP
IO
2880 IP!=(.125*D(IO+1)+.875*D(IO+1+NC)-.125*B(IO+1)-
.875*B(IO+1+NC))*SC!(1)
2890 IF NEEDBASE THEN FOR K=1 TO NA: A!(K)=0: NEXT: GOTO 3020
2900 BTW!=(.25*D(IO+2)+.75*D(IO+2+NC)-.25*B(IO+2)-
.75*B(IO+2+NC))*SC!(2)
2910 BT!=(.375*D(IO+3)+.625*D(IO+3+NC)-.375*B(IO+3)-
.625*B(IO+3+NC))*SC!(3)
2920 VPG!=.5*(D(IO+4)+D(IO+4+NC)-B(IO+4)-B(IO+4+NC))*SC!(4)
2930 JSAT!=(.75*D(IO+6)+.25*D(IO+6+NC)-.75*B(IO+6)-
.25*B(IO+6+NC))*SC!(6)
2940 VTG!=(D(IO+8)-B(IO+8))*SC!(8)
2950 IPDOT!=(3*(D(IO+17)-B(IO+17))+2*(D(IO+9)-B(IO+9))-
5*(D(IO+1)-B(IO+1)))*SC!(1)*SA/64: TIME!=.001*INT(1000*IO/SA+.5)
2960 IF BT!=0 THEN A!(1)=.99: A!(2)=0 ELSE A!(1)=BTW!/BT!:
A!(2)=2*ABS(IP!/BT!/A!)
```

```
2970 A!(3)=VPG!-4*RO!*A!(2)*(1-
A!(2)*A!(2))*VTG!*SGN(BT!)/A!/(8+3*A!(2)*A!(2)*A!(2))-
1.2566*RO!*(18+27*A!(2)*A!(2))*IPDOT!/(48+8*A!(2)*A!(2))
2980 A!(4)=.001*IP!*A!(3)
2990 IF A!(3)=0 THEN A!(5)=0 ELSE
A!(5)=2.3*(ABS(RO!*(5+6*A!(2)*A!(2))*IP!/A!/A!/A!(3)/(10+A!(2)*A!
(2))))^.667
3000 IF A!(5)=0 OR JSAT!<0 THEN A!(6)=0 ELSE
A!(6)=.016*P!*JSAT!/SQR(A!(5))/(1-EXP(-VBIAS!/A!(5)))
3010 IF A!(4)=0 THEN A!(7)=0 ELSE
A!(7)=9.475001*RO!*A!*A!*A!(6)*A!(5)/A!(4): IF A!(7)<0 THEN
A!(7)=0
3020 RETURN
3030     'TEST FOR BASELINE
3040 GOSUB 4980: IF (ABS(D(IO+6)-B(IO+6)))*SC!(6)>100 THEN 3060
3050 NS$="<<<<< Baseline >>>>>": GOSUB 420: LOCATE 12,31,0: PRINT
NS$: NEEDBASE=0
3060 RETURN
3070     'GRAPH CHANNEL IY FOR PREVIOUS SHOTS
3080 IF R$="" THEN 3140
3090 IF ASC(R$)>96 THEN R$=CHR$(ASC(R$)-32)
3100 IF R$="." THEN GOSUB 4670: GOTO 3240
3110 IF R$="M" THEN IY=0: GOTO 3200
3120 IY=VAL(R$): IF IY=0 THEN IY=INSTR(CH$,R$)
3130 IF IY>0 AND IY<=NC+NA+1 THEN 3200
3140 GOSUB 420: PRINT"H";TAB(32)"GRAPH OPTIONS": PRINT
3150 PRINT TAB(29)"M: Time of maximum ";CHS$(1)
3160 FOR I=1 TO NC+NA+1: PRINT TAB(29)MID$(CH$,I,1);": ";CHL$(I):
NEXT
3170 PRINT TAB(29)".: Everything"
3180 LOCATE 25,32: PRINT"Choose graph: ";
3190 R$=INKEY$: IF R$="" THEN 3190 ELSE CLS: GOTO 3090
3200 NP=NS-1: IF NS>NL THEN NP=NL-1
3210 GOSUB 5300: SCY!=SC!(IY): XLABEL$="Shot Number": SCX!=1:
XMIN=NS-NP: XMAX=NS
3220 IF IY=0 THEN TITLE$="Time of maximum "+CHS$(1)+" (msec)"
3230 FOR I=0 TO NP: IL=(NS-NP+I) MOD NL: Y(I)=L(IY,IL): X(I)=NS-
NP+I: NEXT: GOSUB 1280
3240 RETURN
3250     'DETERMINE TIME OF CH 1 MAX SIGNAL AND SAVE DATA AT THAT
TIME
3260 DMAX=0: IO=1: FOR I=1 TO SA*FS+NC STEP NC: IF D(I)-B(I)>DMAX
THEN DMAX=D(I)-B(I): IO=I
3270 NEXT: IL=NS MOD NL
3280 FOR I=0 TO NC-1: L(I+1,IL)=D(IO+I)-B(IO+I): NEXT
3290 L(0,IL)=IO: IF IO>NC THEN IF D(IO-NC)+D(IO+NC)<>2*D(IO) THEN
L(0,IL)=IO+NC*(D(IO-NC)-D(IO+NC))/(D(IO-NC)-2*D(IO)+D(IO+NC))/2
3300 IO=IO-1: GOSUB 2880: FOR I=1 TO NA
3310 IF ABS(A!(I))/SC!(I+NC)>19900 THEN L(I+NC,IL)=0 ELSE
L(I+NC,IL)=A!(I)/SC!(I+NC)
3320 NEXT
3330 RETURN
3340     'INTEGRATE CHANNEL 1 VS TIME
```

```
3350 IL=NS MOD NL: L(NC+NA+1,IL)=0: AMPSEC!=0: IF NEEDBASE THEN
3390
3360 FOR I=1 TO 40*SA STEP NC: IF D(I)>B(I) THEN
AMPSEC!=AMPSEC!+D(I)-B(I)
3370 NEXT: AMPSEC!=AMPSEC!*NC/40
3380 IF ABS(AMPSEC!)<19900 THEN L(NC+NA+1,IL)=AMPSEC!
3390 RETURN
3400     'DISPLAY HEADING ON BOTH SCREENS
3410 GOSUB 460: SCREEN 0: WIDTH 40: COLOR 11,1
3420 LOCATE 9,7,0: PRINT A1$: LOCATE 11,11: PRINT A2$: LOCATE
13,4: PRINT A3$
3430 COLOR 1,2: LOCATE 6,1: PRINT STRING$(39,15): LOCATE 16,1:
PRINT STRING$(39,15)
3440 FOR I=7 TO 15: LOCATE I,1: PRINT CHR$(15): LOCATE I,39:
PRINT CHR$(15): NEXT
3450 GOSUB 420
3460 LOCATE 9,27: PRINT A1$: LOCATE 11,31: PRINT A2$: LOCATE
13,24: PRINT A3$
3470 LOCATE 6,22: PRINT STRING$(37,176): LOCATE 16,22: PRINT
STRING$(37,176)
3480 FOR I=6 TO 16: LOCATE I,20: PRINT CHR$(176);CHR$(176):
LOCATE I,59: PRINT CHR$(176);CHR$(176): NEXT
3490 RETURN
3500     'GRAPH VARIATION OR HISTORY OF Y VS X
3510 IF R$="" THEN 3560
3520 IF ASC(R$)>96 THEN R$=CHR$(ASC(R$)-32)
3530 IF Q$="U" AND R$="M" THEN IY=0: GOTO 3610
3540 IY=VAL(R$): IF IY=0 THEN IY=INSTR(CH$,R$)
3550 IF IY>0 AND IY<=NC+NA THEN 3610
3560 GOSUB 420: PRINT"H";TAB(32)"Y axis of graph": PRINT
3570 IF Q$="U" THEN PRINT TAB(29)"M: Time of maximum ";CHS$(1)
3580 FOR I=1 TO NC+NA: PRINT TAB(29)MID$(CH$,I,1);": ";CHL$(I):
NEXT
3590 LOCATE 25,32: PRINT"Choose Y axis: ";
3600 R$=INKEY$: IF R$="" THEN 3600 ELSE CLS: GOTO 3520
3610 IF S$="" THEN 3660
3620 IF ASC(S$)>96 THEN S$=CHR$(ASC(S$)-32)
3630 IF Q$="U" AND S$="M" THEN IX=0: GOTO 3710
3640 IX=VAL(S$): IF IX=0 THEN IX=INSTR(CH$,S$)
3650 IF IX>0 AND IX<=NC+NA THEN 3710
3660 GOSUB 420: PRINT"H";TAB(32)"X axis of graph": PRINT
3670 IF Q$="U" THEN PRINT TAB(29)"M: Time of maximum ";CHS$(1)
3680 FOR I=1 TO NC+NA: PRINT TAB(29)MID$(CH$,I,1);": ";CHL$(I):
NEXT
3690 LOCATE 25,32: PRINT"Choose X axis: ";
3700 S$=INKEY$: IF S$="" THEN 3700 ELSE CLS: GOTO 3620
3710 GOSUB 5300: SCY!=SC!(IY): SCX!=SC!(IX)
3720 IF Q$<>"U" THEN 3780
3730 NP=NS-1: IF NS>NL THEN NP=NL-1
3740 GOSUB 5300: SCY!=SC!(IY): SCX!=SC!(IX)
3750 IF IY=0 THEN TITLE$="Time of maximum "+CHS$(1)+" (msec)"
3760 IF IX=0 THEN XLABEL$="Time of maximum "+CHS$(1)+" (msec)"
3770 FOR I=0 TO NP: IL=(NS-NP+I) MOD NL: Y(I)=L(IY,IL):
X(I)=L(IX,IL): NEXT: GOTO 3870
```

```
3780 NP=20*FS/NC: IF IY>NC OR IX>NC THEN 3790 ELSE 3850
3790 FOR I=0 TO NP: IO=NC*I: GOSUB 2880
3800 IF IY<=NC THEN 3820
3810 IF (ABS(A!(IY-NC))+1)/SCY!>19900 THEN Y(I)=0 ELSE
Y(I)=A!(IY-NC)/SCY!
3820 IF IX<=NC THEN 3840
3830 IF (ABS(A!(IX-NC))+1)/SCX!>19900 THEN X(I)=0 ELSE
X(I)=A!(IX-NC)/SCX!
3840 NEXT
3850 IF IY<=NC THEN FOR I=0 TO NP: Y(I)=D(NC*I+IY)-B(NC*I+IY):
NEXT
3860 IF IX<=NC THEN FOR I=0 TO NP: X(I)=D(NC*I+IX)-B(NC*I+IX):
NEXT
3870 XMIN=0: XMAX=0: FOR I=0 TO NP: IF X(I)>XMAX THEN XMAX=X(I)
3880 IF X(I)<XMIN THEN XMIN=X(I)
3890 NEXT
3900 I=0: D!=ABS(XMAX)*SCX!: IF ABS(XMIN)>ABS(XMAX) THEN
D!=ABS(XMIN)*SCX!
3910 IF INT(D!)>9 THEN D!=D!/10: I=I+1: GOTO 3910
3920 D!=D!(INT(D!))*10^I: XMAX=(INT(D!)/SCX!)*SGN(XMAX):
XMIN=(INT(D!)/SCX!)*SGN(XMIN): GOSUB 1280
3930 RETURN
3940    'HALT OPERATIONS (STANDBY)
3950 LOCATE 1,1,0: PRINT"H";
3960 Q$=INKEY$: IF Q$="" THEN 3960 ELSE LOCATE 1,1: PRINT" ";
3970 RETURN
3980    'EXIT COMMAND SEQUENCE
3990 ICS=LEN(CS$)+1: Q$=INKEY$
4000 RETURN
4010    'PRINT SCREEN ON LINE PRINTER
4020 WIDTH"LPT1:",255: DEF SEG=0: IF PEEK(&H410)=95 THEN 4050
4030 FOR I=1 TO 25: FOR J=1 TO 80: LPRINT
CHR$(ABS(SCREEN(I,J)));: NEXT: LPRINT: NEXT
4040 FOR I=1 TO 8: LPRINT: NEXT: GOTO 4130
4050 IF XMIN THEN LPRINT STRING$(16,32);"Graph of shots";XMIN;"-
";XMAX;"  ";DATE$: LPRINT: LPRINT: GOTO 4070
4060 LPRINT STRING$(16,32);"Graph of shot";NS;"  ";NS$: LPRINT:
LPRINT
4070 LPRINT CHR$(27);CHR$(19);CHR$(18);: FOR Y=0 TO 199 STEP 7
4080 IMAX=6: IF Y+IMAX>199 THEN IMAX=199-Y
4090 LPRINT STRING$(77,128);: FOR X=0 TO 319
4100 DOTS=128: IO=1: FOR I=0 TO IMAX: DOTS=DOTS OR (-
SGN(POINT(X,Y+I)) AND IO): IO=2*IO: NEXT
4110 LPRINT CHR$(DOTS);: NEXT: LPRINT: NEXT
4120 LPRINT CHR$(30): FOR I=1 TO 12: LPRINT: NEXT
4130 WIDTH"LPT1:",80
4140 RETURN
4150    'TRANSMIT DATA TO SERIAL PORT
4160 LOCATE 1,1: PRINT"T";: OPEN"COM1:4800,E,7,1,CS,DS,CD,LF" AS
#2
4170 IL=NS MOD NL: IF L(9,IL)<0 THEN PRINT#2,"F<0"; ELSE
PRINT#2,"F>0";
4180 PRINT#2,CHR$(7);" ";HEAD$
4190 CLOSE#2: LOCATE 1,1: PRINT" ";
```

```
4200 RETURN
4210     'DECODE A-TO-D ERROR
4220 OUT CMD,15: TEMP=INP(DAT): GOSUB 4960: WAIT CMD,4: OUT
CMD,2: GOSUB 4960: WAIT CMD,5: ERR1=INP(DAT): GOSUB 4960: WAIT
CMD,5: ERR2=INP(DAT)
4230 PRINT ERR1+256*ERR2,
4240 IF ERR1 AND 2 THEN PRINT"Command overwrite"
4250 IF ERR1 AND 4 THEN PRINT"Clock set"
4260 IF ERR1 AND 8 THEN PRINT"Digital port select"
4270 IF ERR1 AND 16 THEN PRINT"Digital port set"
4280 IF ERR1 AND 32 THEN PRINT"DAC select"
4290 IF ERR1 AND 64 THEN PRINT"DAC clock"
4300 IF ERR1 AND 128 THEN PRINT"DAC number of conversions"
4310 IF ERR2 AND 1 THEN PRINT"A/D channel"
4320 IF ERR2 AND 2 THEN PRINT"A/D gain"
4330 IF ERR2 AND 4 THEN PRINT"A/D clock"
4340 IF ERR2 AND 8 THEN PRINT"A/D multiplexer"
4350 IF ERR2 AND 16 THEN PRINT"A/D number of conversions"
4360 IF ERR2 AND 32 THEN PRINT"Data where command expected"
4370 RETURN
4380     'GRAPH EVERYTHING VS TIME
4390 GOSUB 460: COLOR 8,0: NP=20*FS/NC: FOR J=1 TO NC: YMIN=0:
YMAX=0
4400 FOR I=0 TO NP: Y(I)=D(NC*I+J)-B(NC*I+J)
4410 IF Y(I)>YMAX THEN YMAX=Y(I)
4420 IF Y(I)<YMIN THEN YMIN=Y(I)
4430 NEXT: IF YMAX=YMIN THEN YMAX=YMAX+1
4440 DY!=1.1*(CSNG(YMAX)-CSNG(YMIN))
4450 LOCATE J*23/NC-1,6-LEN(CHS$(J)): PRINT CHS$(J);
4460 LINE (44,188*(J+YMIN/DY!)/NC)-(144,188*(J+YMIN/DY!)/NC),1
4470 PSET(44,188*(J-Y(0)/DY!+YMIN/DY!)/NC),2
4480 FOR I=1 TO NP: LINE -(44+100*CSNG(I)/NP,188*(J-
Y(I)/DY!+YMIN/DY!)/NC),2: NEXT
4490 NEXT: LINE (44,0)-(144,188),1,B: LOCATE 25,6: PRINT"0   t
(ms)";: LOCATE 25,20-LEN(STR$(FS)): PRINT STR$(FS);
4500 FOR I=1 TO 9: PSET(44+10*I,1),1: PSET(44+10*I,2),1:
PSET(44+10*I,186),1: PSET(44+10*I,187),1: NEXT
4510 IF NEEDBASE THEN LOCATE 2,26: PRINT"*** Take a": LOCATE
3,23: PRINT"baseline shot": GOTO 4640
4520 FOR J=1 TO NA: YMIN=0: YMAX=0
4530 FOR I=0 TO NP: IO=NC*I: GOSUB 2880: IF
(ABS(A!(J))+1)/SC!(J+NC)>32766 THEN Y(I)=0 ELSE
Y(I)=A!(J)/SC!(J+NC)
4540 IF Y(I)>YMAX THEN YMAX=Y(I)
4550 IF Y(I)<YMIN THEN YMIN=Y(I)
4560 NEXT: IF YMAX=YMIN THEN YMAX=YMAX+1
4570 DY!=1.1*(CSNG(YMAX)-CSNG(YMIN))
4580 LOCATE J*23/NA-1,26-LEN(CHS$(J+NC)): PRINT CHS$(J+NC);
4590 LINE (204,188*(J+YMIN/DY!)/NA)-(304,188*(J+YMIN/DY!)/NA),1
4600 PSET(204,188*(J-Y(0)/DY!+YMIN/DY!)/NA),2
4610 FOR I=1 TO NP: LINE -(204+100*CSNG(I)/NP,188*(J-
Y(I)/DY!+YMIN/DY!)/NA),2: NEXT
4620 NEXT: LINE (204,0)-(304,188),1,B: LOCATE 25,26: PRINT"0   t
(ms)";: LOCATE 25,40-LEN(STR$(FS)): PRINT STR$(FS);
```

```
4630 FOR I=1 TO 9: PSET(204+10*I,1),1: PSET(204+10*I,2),1:
PSET(204+10*I,186),1: PSET(204+10*I,187),1: NEXT
4640 R$="": XMIN=0
4650 RETURN
4660     'GRAPH EVERYTHING VS SHOT NUMBER
4670 GOSUB 460: COLOR 8,1: IF NS<=1 THEN PRINT"  *** No data to
graph.": GOTO 4930
4680 NP=NS-1: IF NS>NL THEN NP=NL-1
4690 FOR J=1 TO NC+1: YMIN=0: YMAX=0
4700 FOR I=0 TO NP: IL=(NS-NP+I) MOD NL: Y(I)=L(J-1,IL)
4710 IF Y(I)>YMAX THEN YMAX=Y(I)
4720 IF Y(I)<YMIN THEN YMIN=Y(I)
4730 NEXT: IF YMAX=YMIN THEN YMAX=YMAX+1
4740 DY!=1.1*(CSNG(YMAX)-CSNG(YMIN))
4750 LOCATE J*23/(NC+1)-1,6-LEN(CHS$(J-1)): PRINT CHS$(J-1);
4760 LINE (44,188*(J+YMIN/DY!)/(NC+1))-
(144,188*(J+YMIN/DY!)/(NC+1)),1
4770 PSET(44,188*(J-Y(0)/DY!+YMIN/DY!)/(NC+1)),2
4780 FOR I=0 TO NP: LINE -(44+100*I/NP,188*(J-
Y(I)/DY!+YMIN/DY!)/(NC+1)),2: NEXT
4790 NEXT: LINE (44,0)-(144,188),1,B: LOCATE 25,5: PRINT STR$(NS-
NP);: LOCATE 25,20-LEN(STR$(NS)): PRINT STR$(NS);: LOCATE 25,11:
PRINT"Shot";
4800 FOR I=1 TO 9: PSET(44+10*I,1),1: PSET(44+10*I,2),1:
PSET(44+10*I,186),1: PSET(44+10*I,187),1: NEXT
4810 FOR J=1 TO NA+1: YMIN=0: YMAX=0
4820 FOR I=0 TO NP: IL=(NS-NP+I) MOD NL: Y(I)=L(J+NC,IL)
4830 IF Y(I)>YMAX THEN YMAX=Y(I)
4840 IF Y(I)<YMIN THEN YMIN=Y(I)
4850 NEXT: IF YMAX=YMIN THEN YMAX=YMAX+1
4860 DY!=1.1*(CSNG(YMAX)-CSNG(YMIN))
4870 LOCATE J*23/(NA+1)-1,26-LEN(CHS$(J+NC)): PRINT CHS$(J+NC);
4880 LINE (204,188*(J+YMIN/DY!)/(NA+1))-
(304,188*(J+YMIN/DY!)/(NA+1)),1
4890 PSET(204,188*(J-Y(0)/DY!+YMIN/DY!)/(NA+1)),2
4900 FOR I=0 TO NP: LINE -(204+100*I/NP,188*(J-
Y(I)/DY!+YMIN/DY!)/(NA+1)),2: NEXT
4910 NEXT: LINE (204,0)-(304,188),1,B: LOCATE 25,25: PRINT
STR$(NS-NP);: LOCATE 25,40-LEN(STR$(NS)): PRINT STR$(NS);: LOCATE
25,31: PRINT"Shot";
4920 FOR I=1 TO 9: PSET(204+10*I,1),1: PSET(204+10*I,2),1:
PSET(204+10*I,186),1: PSET(204+10*I,187),1: NEXT
4930 R$="": XMIN=NS-NP: XMAX=NS
4940 RETURN
4950     'DELAY FOR A-TO-D CONVERSION
4960 RETURN
4970     'KLUGE TO CORRECT MYSTERIOUS A-TO-D DMA PROBLEM
4980 IF B(NC+3)>B(NC+2) THEN FOR I=2016 TO NC+1 STEP-1: B(I)=B(I-
1): NEXT: B(NC)=B(2*NC)
4990 IF D(NC+4)<D(100*NC+4) THEN FOR I=2016 TO NC+1 STEP-1:
D(I)=D(I-1): NEXT: D(NC)=2*D(2*NC)-D(3*NC)
5000 RETURN
5010     'SET D-TO-A CONVERTER OUTPUTS
5020 D0$="-": D1$="-": IF POFF THEN PUFFH=8: PUFFL=0: GOTO 5080
```

```
5030 IF NEEDBASE THEN PUFFH=15: PUFFL=255: GOTO 5080
5040 IF DP0=0 THEN DP0=1
5050 IL=NS MOD NL: DP0=SS*SGN(L(IP0,IL)-LOLD)*SGN(DP0): P0=P0+DP0
5060 IF P0>3967 THEN P0=3967 ELSE IF P0<2176 THEN P0=2176
5070 PUFFH=INT(P0/256): PUFFL=P0-256*PUFFH: IF DP0>0 THEN
D0$=CHR$(25) ELSE IF DP0<0 THEN D0$=CHR$(24)
5080 OUT CMD,15: TEMP=INP(DAT): WAIT CMD,4: OUT CMD,1
5090 IF (INP(CMD) AND 112)=112 THEN 5240
5100 WAIT CMD,4: OUT CMD,8: WAIT CMD,2,2: OUT DAT,0
5110 WAIT CMD,2,2: OUT DAT,PUFFL: WAIT CMD,2,2: OUT DAT,PUFFH
5120 WAIT CMD,4: STATUS=INP(CMD): IF (STATUS AND 128) THEN CLS:
PRINT"  *** D-to-A error";: GOSUB 4220: GOTO 5020
5130 IF POFF THEN 5190
5140 IF NEEDBASE THEN 5190
5150 IF DP1=0 THEN DP1=1
5160 DP1=ABS(SS)*SGN(AMPSEC!-ASOLD!)*SGN(DP1): P1=P1+DP1
5170 IF P1>3967 THEN P1=3967 ELSE IF P1<2176 THEN P1=2176
5180 PUFFH=INT(P1/256): PUFFL=P1-256*PUFFH: IF DP1>0 THEN
D1$=CHR$(25) ELSE IF DP1<0 THEN D1$=CHR$(24)
5190 OUT CMD,15: TEMP=INP(DAT): WAIT CMD,4: OUT CMD,1
5200 WAIT CMD,4: OUT CMD,8: WAIT CMD,2,2: OUT DAT,1
5210 WAIT CMD,2,2: OUT DAT,PUFFL: WAIT CMD,2,2: OUT DAT,PUFFH
5220 WAIT CMD,4: STATUS=INP(CMD): IF (STATUS AND 128) THEN CLS:
PRINT"  *** D-to-A error";: GOSUB 4220: GOTO 5130
5230 IF PCHANGE=0 THEN ASOLD!=AMPSEC!: LOLD=L(IP0,IL):
D0H$=RIGHT$(D0H$,69)+D0$: D1H$=RIGHT$(D1H$,69)+D1$
5240 RETURN
5250     'TOGGLE PUFF VALVE TRACKING
5260 IF POFF=1 THEN POFF=0: P0=2304: P1=P0: ELSE POFF=1
5270 PCHANGE=-1: GOSUB 5020: PCHANGE=0
5280 RETURN
5290     'CONSTRUCT GRAPH LABELS
5300 IF UNIT$(IY)="" THEN TITLE$=CHL$(IY) ELSE TITLE$=CHL$(IY)+"
("+UNIT$(IY)+")"
5310 IF UNIT$(IX)="" THEN XLABEL$=CHL$(IX) ELSE
XLABEL$=CHL$(IX)+" ("+UNIT$(IX)+")"
5320 RETURN
5330     'COMPOSE HEADER
5340 IL=NS MOD NL: HEAD$="Shot:"+STR$(NS)+"  "+NS$+"    "
5350 IF NSAVE THEN 5380
5360 HEAD$=HEAD$+CHS$(1)+" ="+STR$(INT(L(1,IL)*SC!(1)+.5))+"
"+UNIT$(1)+" at"+STR$(INT(10*L(0,IL)*SC!(0)+.4)/10)+" "+UNIT$(0)
5370 HEAD$=HEAD$+"  "+STR$(INT(L(NC+NA+1,IL)*SC!(NC+NA+1)+.5))+"
"+UNIT$(NC+NA+1)
5380 RETURN
5390     'GRAPH F-THETA THEORY CURVES (BFM AND MBFM)
5400 FOR I=21 TO 299: H!=(I-20)*SCX!*XMAX/280!
5410 F!=1-.5*H!*H!*(1+.166667*H!*H!*(1+.25*H!*H!))
5420 J=190-180!*(F!/SCY!-YMIN)/(YMAX-YMIN): IF J<190 THEN PSET
(I,J),3
5430 F!=1-.5*H!*H!
5440 J=190-180!*(F!/SCY!-YMIN)/(YMAX-YMIN): IF J<190 THEN PSET
(I,J),3
5450 NEXT
```

```
5460 RETURN
5470     'SAVE CONFIGURATION
5480 PRINT"Saving configuration...": CLOSE:
OPEN"O",1,"CONFIG.DAT":
WRITE#1,NS,NS$,AMPSEC!,FS,CS$,NEEDBASE,POFF,SS,IPO,DOH$,D1H$,PO,P
1
5490 FOR I=0 TO 2016: WRITE#1,D(I): NEXT: FOR I=0 TO NC+NA+1: FOR
J=0 TO NL-1: WRITE#1,L(I,J): NEXT J,I: CLOSE
5500 RETURN
5510     'RESTORE CONFIGURATION
5520 OPEN"I",1,"CONFIG.DAT":
INPUT#1,NS,NS$,AMPSEC!,FS,CS$,NEEDBASE,POFF,SS,IPO,DOH$,D1H$,PO,P
1
5530 IF LEN(DOH$)<70 THEN DOH$=STRING$(70,"-")
5540 IF LEN(D1H$)<70 THEN D1H$=STRING$(70,"-")
5550 FOR I=0 TO 2016: INPUT#1,D(I): NEXT: FOR I=0 TO NC+NA+1: FOR
J=0 TO NL-1: INPUT#1,L(I,J): NEXT J,I: CLOSE
5560 RETURN
5570     'CHANGE PUFF VALVE TRACKING
5580 GOSUB 420: GOSUB 5340: LOCATE 1,4,0: PRINT HEAD$;: PRINT
5590 PRINT: IF POFF THEN PRINT"DAC0 is off"; ELSE IF SS THEN
PRINT"DAC0 is tracking ";CHL$(IPO);" with a step size of";SS;
ELSE PRINT"DAC0 is locked";
5600 PRINT" (Level =";STR$(PO-2048);")": PRINT"History: ";DOH$:
PRINT
5610 IF POFF THEN PRINT"DAC1 is off"; ELSE IF SS THEN PRINT"DAC1
is tracking ";CHL$(NC+NA+1);" with a step size of";ABS(SS); ELSE
PRINT"DAC1 is locked";
5620 PRINT" (Level =";STR$(P1-2048);")": PRINT"History: ";D1H$:
PRINT: PRINT"Best recent shot:
#";NL*INT(NS/NL)+BESTIL;INT(L(1,BESTIL)*SC!(1)+.5);UNIT$(1);"
/";INT(L(NC+NA+1,BESTIL)*SC!(NC+NA+1)+.5);UNIT$(NC+NA+1);
5630 IF IPO<>1 AND IPO<=NC+NA THEN PRINT"  (";CHS$(IPO);"
=";INT(10*LBEST*SC!(IPO)+.5)/10;UNIT$(IPO);")";
5640 PRINT: PRINT: IF R$="X" OR R$="x" THEN R$="": GOTO 5900
5650 IF R$<>"" THEN 5780
5660 PRINT TAB(26)"PUFF VALVE TRACKING OPTIONS": PRINT
5670 PRINT TAB(29)"C: Change DAC0 control"
5680 PRINT TAB(29)"D: Decrease step size"
5690 PRINT TAB(29)"I: Increase step size"
5700 PRINT TAB(29)"L: Lock puff level"
5710 PRINT TAB(29)"O: Turn tracking ";PUFF$(POFF)
5720 PRINT TAB(29)"R: Restore defaults"
5730 PRINT TAB(29)"S: Set puff level"
5740 PRINT TAB(29)"U: Unlock puff level"
5750 PRINT TAB(29)"X: Exit puff submenu"
5760 LOCATE 1,1,0: PRINT"H";: LOCATE 24,32,1: PRINT"Choose
option:";: I=0
5770 R$=INKEY$: I=I+1: SOUND 32767,1: IF R$="" THEN IF I<92 THEN
5770 ELSE R$="X"
5780 IF ASC(R$)>96 THEN R$=CHR$(ASC(R$)-32)
5790 IF R$="C" THEN IPO=1+IPO MOD (NC+NA+1): GOSUB 5920
5800 IF R$="D" AND POFF=0 THEN IF ABS(SS)>1 THEN SS=SS/2 ELSE
R$="L"
```

```
5810 IF R$="I" AND POFF=0 AND ABS(SS)<2048 THEN SS=2*SS
5820 IF R$="L" THEN IF SS THEN SSOLD=SS: SS=0
5830 IF R$="O" THEN GOSUB 5260
5840 IF R$="R" THEN POFF=0: SS=64: IPO=1: GOSUB 5920
5850 IF R$="S" THEN GOSUB 420: INPUT"New puff level (0-2047)";PO:
CLS: PO=PO+2048: P1=PO: PCHANGE=-1: GOSUB 5020: PCHANGE=0
5860 IF R$="U" THEN SS=SSOLD
5870 IF R$="X" THEN CLS: GOTO 5580
5880 SS=ABS(SS): SSGN=1: IF IPO=2 OR IPO=9 THEN SS=-ABS(SS):
SSGN=-1
5890 R$="": Q$="": CLS: GOTO 5580
5900 RETURN
5910    'INITIALIZE BEST SHOT
5920 BESTIL=IL: LBEST=L(IPO,IL): GOSUB 420
5930 LOCATE 1,4: PRINT"Best shot has been reinitialized to
current shot"
5940 RETURN
```