Fermilab

TM-740
1600.000

E22-14 CRYOSTAT BOIL-OFF

M.Kuchnir

July 20, 1977

INTRODUCTION

The first cryostat of the present series of Energy Doubler/ Saver superconducting dipoles was dedicated for the thermal insulation evaluation reported below. The method used[1] was based on the evaporation rate of liquid helium stored in this cryostat when positioned as an inclined dewar. The plot of the evaporation rate as a function of liquid level presents a series of steps. These steps reflect the decreasing heat input into the liquid as individual sets of supports emerge from the receding level. As shown in Fig. 1 a typical junction between two cryostats was included as part of the inclined dewar. This set up was instrumented with a level indicator, several thermometers, several heaters and a set of external spring loaded frames for compressing the mechanical supports simulating the magnetic forces between the coil and the iron yoke outside the cryostat. This simulation was regarded as necessary since thermal contact with the supports is a major component of the insulation scheme.

Two kinds of boil-off runs were carried out: those in wnich the shield was cooled with boiling nitrogen and those in which the shield was cooled with cold helium gas. While the latter premitted us to find the heat load on the shield by determining the energy absorbed by the gas, the former provided us with the uniform temperature designed into Energy Doubler operation and permitted the study of the heat load into the 4K region of the cryostat.

An extensive set of data (18 runs) was collected by an automatic system based on a PDP-11 computer. The development of such a system was dictated by the need to monitor over 20 parameters for the ~32 hours duration of a run. After describing the instrumentation and the automatic system we present the results.

INSTRUMENTATION

The volume usually occupied by the superconducting coil was filled with helium except for 1-inch long sections of collars under the supports. An Advance wire, .004-inch diameter, coated with 60/40 SnPb solder was kept stretched on this volume by means of springs and insulators. It was used as a superconducting helium level sensor. Its proper operation required a bias current (dc) and an ac resistance measuring circuit based on a lock-in-amplifier. This level indicator was very useful in the first few runs. In the analysis of the data it was more convenient however to use the mass of gas evaporated instead of the level indicator. Two 50Ω 1% 5W resistors were installed also in the main helium volume: one of them, R_A, halfway between support set #4 and the anchor support and the second one, R_B, halfway between supports #8 and #9. Under a known constant current they acted as calibrated heat sources visible as an extra step in the evaporation rate data. This step however was not as sharp as one might have expected and in some runs these calibration heat sources were turned off.

Two calibrated[2] carbon resistor thermometers were installed in the cryostat itself during assembly. One (T6) measured the temperature at the base of a G-10 roller in the support set #8 and another (T5) the temperature of the shield in the point of the junction box where worst cooling was expected.

The lower end box, 4 feet of which is a typical E22-14 cryostat, provided means for returning the shield fluid through a 5/8-inch tube inside the beam tube to the upper end box and connected the single phase with the two phase spaces. Two thermometers: a carbon[2] resistor (T7) in the 4K region detected liquid helium presence and a platinum resistor (T2) in the end of the shield, together with vacuum gauges (a T.C. gauge and a Bayard-Alpert gauge), heaters, level indicator leads and a thermistor (T8) measuring the external wall temperature, constituted the instrumentation of the lower end box.

The upper end box provided the transfer tube connections needed for cooling, filling, monitoring the evaporation in the 4K region as well as flowing the shield cryogen. A carbon[2] resistor (T3) at the top of the single phase tube provided redundancy to the level gauge in the determination of the full condition. It was determined that the most effective way to fill the cryostat was by feeding liquid to the two phase space and venting through the single phase space. This fill-up operation took two hours. A manifold containing a pressure gauge (P14) connected both the single phase and the two phase parts to the pipe leading to a heat exchanger and gas meters during the boil-off operation.

The shield ports were equipped with calibrated glassy carbon[3] thermometers: (T1) immersed in the input stream and (T4) immersed in the output stream. Another thermometer in the input stream was used as a sensor for a feedback system[4] that controlled a heating element at the beginning of the input stream when this was cold helium gas generated inside a storage dewar. This system plus pressure regulators permitted stable conditions at the shield for long periods of time.

The line conducting the evaporated gas from the upper end box to the gas meter is a copper tube 7/16-inch diameter, ~43 feet long, instrumented with a thermostat and a heater, and partially coiled inside a water container to form a heat exchanger before reaching the gas meters. Its nonnegligible flow impedance provided some stability in the evaporation rate with regard to pressure changes further down the line. Pressure transducers before and after the gas meter as well as another thermometer completed the instrumentation of the line.

The gas meters[5] were adapted to automatic data collection by installation of an optical detector in the meter hand path. This detector sent an electric pulse everytime the pointer interrupted its light beam.

The recording of these pulses and their timing is the basis for the flow rate measurements. These pulses were also used to totalize the mass of gas flown out. The interval between pulses corresponded to a volume of 10 ft^3. Wet test meters were used to calibrate these meters.

DATA COLLECTING SYSTEM

In order to effectively monitor the over 20 variables of the boil-off which usually lasted 32 hours, an automatic system was developed. The hardware consisted of a CAMAC crate directed by a PDP-11/20 computer equipped with a Tektronix 4010-1 terminal and a RX01 floppy disk driving unit. The CAMAC crate was controlled by a 3911A Kinetic Systems Unibus Controller. It contained a 2232 LeCroy 32 channel ADC, a quad interrupt module and a 017 Fermilab Control 24 bit output unit.

Pulses from the gas meters set the LAM in the quad interrupt module. A real time KW11-P clock in the computer caused the quad interrupt module to be examined every .01 sec. Only three out of the possible four channels of the quad interrupt were used. Once a pulse was detected an assembly language subroutine stored a record of it in one of two alternating buffers. The record consisted of identification tag, counter reading and time of detection. This was the first priority operation of the system. All other operations were controlled by a Fortran program which communicated with the operator by sensing the switches of the PDP-11/20 and communicated with the CAMAC crate by calling subroutines.

A listing of ROIL.FOR, the latest version of the main program, is presented in the appendix. A short write-up about it follows: After an initializing sequence the program senses the setting of the PDP-11/20 switches. In response to their setting it proceeds or calls Fortran subroutines like 1) MANUAL, which permits the operator to command nonroutine reading and recording of parameters or recording of nonstandard information; 2) PARAM, which permits the operator to read or modify the parameters controlling the performance of the data collection. Other switches controlled the display of information in the scope of the 4010-1 or enabled sound alarms. After sensing the switches the program checks the status of the MB1 and MB2 buffers containing the records of gas meter pulses. If a buffer is full it calls UNLO, the Fortran subroutine that unloads and resets them storing their contents in the general STORE buffer. In its operation, UNLO keeps track of the STORE buffer counters and when needed calls TRANS, a Fortran subroutine that transfers blocks of records from the cyclic STORE buffer to permanent recording in a magnetic diskette. Returning from UNLO the program checks the clock for read-in time and returns to sense the switches. If it is time to read-in the status of the boil-off it proceeds to do it through the 32 channel ADC in the CAMAC, appropriately controlling the relays involved in the measurement of temperatures (four lead dc technique) through the 017 unit in the CAMAC and then returns to sense the switches.

A typical record contains five pieces of information: a code number MB31 identifying the variable; two intergers MB32 and MB33 with the raw value of the variable; a floating point number B34 with the processed value of the variable in proper units and the time of the recording B35 in seconds since the beginning of the run. Code number 0 was reserved for recording eight ASCII characters typed in by the operator instead of variable values. Such operators comments giving date, time of the day, name of the boil-off run, unusual conditions, etc., were invaluable in saving the recorded data in case of power failures or computer crash.

Several MACRO subroutines from a COIL.OBJ library are used: NCL clears the clock; BUF transfers the addresses of the counters NB1 and NB2 of the buffers MB1 and MB2 and their size NBO from the main program to the MACRO subroutines using them; TP transfers the addresses of the clock counters K00 and K01 as well as pulse counters L10 to L41 to the MACRO subroutines using them; CLOK starts the clock; CAMAC initializes the CAMAC crate; BITZ(I) senses the setting of the Ith switch of the PDP-11/20; CMCI (O, N, IA, MD1, MD2) reads the contents of the IAth address of the Nth module into MD1 (low data) and MD2 (high data); CMCO (16, N, IA, MD1, MD2) sets the IAth address of the Nth module to have MD1 in the high register and MD2 in the low register, since only one register is set at a time this subroutine has to be called twice in a row; BELL rings the bell in the terminal; STCL stops the clock; SEC updates the clock counters.

More than one diskette can be used to record the data of a run. The first assignment is made during the initialization procedure, all others are called for by TRANS, the Fortran subroutine that transfers data to the diskette. If the PDP-11/20 switch 5 is on, TRANS signs in the screen the time it was run. This helps in determining the rate at which data is being recorded and estimating when a new diskette will be needed. The seven word records are unformatted and stored in a random access file for maximum density. IREC is the size of the file and NB8 its counter. During assignment of a file the operator is asked to name the file with a 14 character name DX1:XXXXXX.DAT where XXXXXX identifies the file. The diskette to receive the data must be in the DX1: unit of the RX01. The DX0: unit should contain the RT-11 system monitor and the ROIL.SAV program. The closing of a data file, asked through PARAM, is effected by TRANS which takes care of running UNLO to update the STORE buffer with the gas meter pulse data before transferring its last set of records.

The transfer and processing of the gas meter pulses (stored in the alternating MB1, MB2 buffers by the highest priority MACRO subroutines) to the STORE buffer is done periodically by the main program or by TRANS through the Fortran subroutine UNLO (NBU) where NBU = 1, 2 identify MB1, MB2 respectively. Like TRANS, UNLO signs in the screen the time it is called up if the PDP-11/20 switch 5 is on. This helps the selection of how big NBO (the size of the buffers MB1 and MB2) should be set. Note that NBO is transferred at initialization through BUF, so its resetting through PARAM is not complete. In forming the records for the gas meter pulses, UNLO calculates the mass of gas flown between pulses properly corrected for gas meter pressure and temperature and stores it as B34 (g/sec).

Modification of the main program itself while collecting data is possible by turning on the PDP-11/20 switch 3 calling the Fortran subroutine PARAM. It prompts with "ENTER CODE AND NEW INTEGER". By entering 0 or nothing and pressing CR (carriage return) in the terminal the file in DX1: is terminated.

By entering 1 and CR the file in DX1: is terminated and the whole program stops. By entering 2, N CR the buffer size NBO is set to be N (see observation in previous paragraph). By entering 3, N CR the waiting period DELAY used in the reading of the digital nanovoltmeter is set to N seconds. The digital nanovoltmeter reads the voltages caused by 10μA on the resistance thermometers, its analog output (0-2V) is read by the ADC. The parameter DELAY is used in the Fortran subroutine WAITER. By entering 4, N CR the interval DT between general readings of all boil-off variables is set to N seconds and KT, the counter of DT intervals, is reset for a general reading to follow. By entering 5, N CR a listing of the present values of program parameters (NBO, IREC, DELAY, DT, KT, NB7), counters (NB1, NB2, NB3, NB4, NB5, NB6, NB8, NFD, K00, K01) and time is presented in the screen of the terminal. The counters NB4, NB5 and NB6 are used in the record transfer accounting, NFD refers to number of floppy disk files, K00 and K01 are the time counters. NB7 is the code number of the thermometer to be monitored continuously by the chart records. By entering 6, N CR NB7 is set to N but the monitoring is only effected after a reading of all variables, or use of the MANUAL subroutine with code 50.

Since the system for reading the resistance thermometers was based on using four leads (to eliminate lead resistance), dc current reversal (to eliminate thermal emf) and reference to a standard resistor (to eliminate current dependence) a Fortran subroutine THER (IA) was used to do the relay manipulations needed to read the resistance RES (ohms) of the thermometer with code IA; conversion into temperature (K) was done using the function TABLE (IA, RES) which contains the calibration curves of all the thermometers. The standard resistor is not read automatically with each resistor and since its last reading is used in determining the current, it should be read before the resistor in question, specially if much time has passed since the standard last reading. There are two independent resistance thermometer circuits: one with all the low temperature thermometers (T1 to T9) fed by a 10μA source and having T9 as standard and the other with room temperature thermistors (T10, T28, T45 and T33) fed by 100μA source and having T10 as standard. The function TABLE (IA, RES) uses simple logarithmic interpolation. For values outside the range of the calibration table it responds with the default or error temperature 2.0K.

The Fortran subroutine MANUAL is called by switching on the PDP-11/20 switch 1. It prompts with "ID:". By entering -N CR (negative integer followed by carriage return) in the terminal the last value of the variable with code number N entered in the STORE buffer is displayed in the screen and no record is created. By entering N CR or +N CR the variable with code number N is read and a record is entered into the STORE buffer as well as displayed on the screen. The time entered in the record is the time t when the program found switch 1 on. By entering 0 CR or just CR the subroutine prompts with "AT (time t) SEC ENTER 8 ASCII COMMENT:"; it will then accept up to eight typed characters and form a record with them, all the other characters following these up to the CR are lost. The record formed is shown on the screen. A special variable code is introduced here, N = 50, for setting the relays into monitoring thermometer NB7. This MANUAL subroutine is very useful even in nonboil-off operations: cooldown, fill up, coast, etc., since it reads on demand the sensors needed by the operator and presents them in proper units. Used throughout the program, the function TIME translates the clock counters into the run time since initialization. Table I lists the variables monitored and their codes.

DATA HANDLING SOFTWARE

For the convenient retrieval, presentation and analysis of the data the time sharing system of the Fermilab PDP-10 is used through a remote 4006-1 Tektronics terminal provided with a 4662 Tektronics plotter. The program written for this boil-off plotting operation, BOP.F4, is linked with the SYS:TEKLIB library and uses the data stored in two or three kinds of files, identified by the name of the run and the extension .DAT, .TAB and .EAB if the run had the shield cooled by helium gas. These files were stored in the magnetic tape DP0002 through BACKUP and the tape is kept in the vault. Before running BOP.F4, BACKUP should be used to retrieve them into the disk. The comments in the listing BOP.F4 is judged sufficient for explaining how to use it. Here it is sufficient to say that it allows for two types of abscissas: time or volume of evaporated liquid; any monitored variable can be used as ordinate (and with some further improvement even the comments can be written at their corresponding abscissa value); two modes of multiple plots are available: several variables of the same boil-off run or the same variable in different boil-off runs. In the next three paragraphs it is described how the data files in the diskettes were transferred into the final files mentioned above.

The conversion of diskette data into uniformly formated data involved treating some of the runs in a tailor-made fashion due to their use of earlier versions of ROIL.FOR or fragmentation of data set in different files due to temporary data collection failure. The successively improved running versions of the data collecting program were named KOIL.FOR, JOIL.FOR, GOIL.FOR and finally ROIL.FOR. The diskette data files collected with them were labeled DX1:KOILXX.DAT, DX1:JOILXX.DAT, etc., where XX is a number identifying the run and the file fragment. At the Bison Station 21 or 22 (C.L. 7th NE) the data in the diskettes were transferred to magnetic tape DP0001 or DP0003 using the RT11 system and Fortran program TAPER.FOR. Here all the fragmented parts of a run were consolidated in one file labeled MT:YOIXX1.DAT, where Y stands for K, J, G or R identifying the version of the data collecting program and XX identifies the run. These files were transferred to the PDP-10 disks using CHANGE (CHANGE and BACKUP are PDP-10 magnetic tape handling utilities) and printed out for inspection. A set of small Fortran programs were written to examine and fix this file into its final version YOIXX6.DAT, as well as generate the auxiliary files YOIXX6.BAK, YOIXX6.TAB and YOIXX6.EAB. A preliminary examination of the file YOIXX1.DAT is done with FXBAK.F4 which generates a time × gas mass out file

YOIXX1.BAK. A custom made version of STROI.F4 is then made to fix the time and pulse counter sequence of YOIXX1.DAT into YOIXX6.DAT if it was a fragmented file. Otherwise YOIXX1.DAT is renamed YOXX6.DAT.

A preliminary program, TPOT.REL, linked with SYS:GRALIB plots in the 4006-1 terminal or on Calc Comp a flow rate × gas mass out of YOIXX1.BAK (note: a temporary renaming of this file extension to .TAB is needed). From visual inspection of this plot two trimming parameters are chosen: 1) ZERO the liters of helium out before which the data should be disregarded as out of equilibrium and 2) REFERENCE the mass gas out corresponding to the last little peak (present in all runs) selected for coincidence in multirun presentations. The program TABLER.F4 uses YOIXX6.BAK, prompts for ZERO and REFERENCE and generates YOIXX6.TAB,

The program SHIELD.F10 linked with HELIUM.REL generates the YOIXX6.EAB file. The program PRESHIELD.F4 was run before to determine the array sizes in SHIELD.F10. File YOXX6.EAB contains the information from the shield gas meter pulses, shield input temperature and shield output temperature processed into energy absorbed by the shield through the increase of the helium gas enthalpy.

RESULTS

For evaluation of heat loads into the 4K region of the cryostat the relevant data is the evaporation rate. Under the condition of the test (steady state flow, constant pressure above the liquid) the evaporation rate multiplied by the latent heat yields the heat load in watts. Some of the data for a typical run with boiling nitrogen in the shield is presented in Fig. 2. The individual contribution of support sets #4 and #A as well as #8 and #9 were hard to separate in some runs so they are presented together in Table II where we present the heat load due to the individual support sets under different spring compressions. No compression dependence is observable in the range examined.

Figure 3 presents some of the data of a typical run with shield cooled by helium gas. An examination of Table III with average shield load in the different runs reveals a relative independence from the spring compression. This indicates that substantial improvement can be obtained by further reducing infrared radiation incidence on the shield.

Using .57W as the heat load from the anchor set (Fig. 2) and averages from Tables II and III, the static heat load per magnet plus junction is expected to be: 4.5±.6 watts on the liquid helium and 30.±5 watts on the shield. These numbers agree with the 5.3W and 30W estimated by R.B.Jacobs[6].

ACKNOWLEDGEMENTS

L.Taff's interest and advice in the debugging of the automatic system was sincerely appreciated. The COIL.OBJ library of MACRO subroutines was provided by the Magnet Measurement Group.

REFERENCES

1. M.Kuchnir, "Heat Leak Measurements on a D-10' Cryostat", Fermilab TM-645 (February 1976).

2. The calibration of these resistors is discussed in Fermilab TM-647 (March 1976).

3. Lake Shore Cryotronics, Inc., Eden, N.Y.

4. Temperature Controller Model 5301, Artronix Instrumentation, Inc., St.Louis, Missouri.

5. Gas Meter Model 1600, Rockwell Mfg. Co., Pittsburg, Pa.

6. R.B.Jacobs memo to W.B.Fowler, February 1977.

TABLE I

BOIL-OFF VARIABLES: NOTATION, CODE, ETC.

Record Code Number	Software, Hardware Notations	ADC Channel + 1	Boil-Off Variable	Sensor
0	—— Comment ——			
1	T1	1	Shield Inlet Temp (K)	Glassy Carbon C-721
2	T2	1	Shield Bottom Temp (K)	Platinum N
3	T3	1	Top Full Sensor (K)	100Ω Carbon 760805-2
4	T4	1	Shield Outlet Temp (K)	Glassy Carbon C-732
5	T5	1	Shield Hot Spot Temp (K)	Platinum T
6	T6	1	Roller Base Temp (K)	1000Ω 760408-7
7	T7	1	Bottom Full Sensor (K)	100Ω carbon 760805-1
8	T8	1	Outside Temp (K)	Thermistor YSI-44001
9	T9	1	Cold Circuit Current ~10μA	100.11Ω
10	T10	16	Thermistors Circuit Current ~230mA	99.95Ω
11	T11	16	Wet Test Meter Temp (K)	Thermistor YSI-44001
12	D3(1), P12	2	Wet Test Meter Pressure (Atm)	10-20 psia NS Transducer
13	D3(2), F13	3	Boil-off Flow Rate (FFS)	Datametrics Hot Wire
14	D3(3), P14	4	Boil-off Top Pressure (Atm)	10-20 psia NS Transducer
15	D3(4), W	5	Level Indicator (Arbitrary)	PAR Phase Lock In
16	D3(5), Wa	6	Heater R_A Power (W)	50Ω1%5W
17	D3(6), Wb	7	Heater R_B Powers (W)	50Ω1%5W
18	D3(7), HHE	8	Heater Bottom He Space (V)	50 Ω
19	D3(8), P19	9	Shield Gas Meter Pressure (Atm)	0-15 psig NS Transducer
20	D3(9), Ical	10	Current Thru R_A & R_B (A)	51Ω1%
21	D3(10), V	11	Vacuum (log scale) (torr)	Ion Gauge
22	D3(11),P32	12	Atmospheric Pressure (Atm)	0-15 psia NS Transducer
23	D3(12),HSH	13	Heater Bottom Shield (V)	50 Ω
24	D3(13),—	14	Reference Voltage Source(V)	1.6V Battery
25	D3(14),F25	15	Shield Flow Rate (8/sec)	Hastings Flow Meter
28	T28	16	Shield Gas Meter Temp (K)	Thermistor YSI-44001
33	T33	16	Shield Line Alarm Temp (K)	Thermistor YSI-44001
40	—— File Information ——			
41	-	1st Quad	Boil-off Timed Pulse (g He/pulse)	West Test Meter
42	-	2nd Quad	Shield Timed Pulse (g He/pulse)	Gas Meter
44	-	3rd Quad	Boil-off Timed Pulse (g He/pulse)	Gas Meter
45	T45	16	Boil-off Line Alarm Temp (K)	Thermistor YSI-44001

TABLE II

HEAT LOAD CONTRIBUTION TO THE 4K REGION (W)

Boil-Off Run	Spring Compression	Support Sets: 3	4 + A	6	7	8 + 9	Junction
04	230 lb	-	-	.48	.37	.66	1.35
05	290	.35	.77	.38	.32	.68	1.23
06	400	.32	.78	.38	.37	.58	1.32
07	460	-	-	.41	.38	.71	1.25
08	520	.34	.83	.38	.36	.65	1.15

TABLE III

HEAT LOAD INTO THE SHIELD

(Cryostat + Junction + End Boxes)

Boil-off Run	Spring Compression	Average Shield Temperature	Average Shield Load
16	520 lb	47.7 K	34. W
17	750	43.5	34.
18	520	36.5	33.

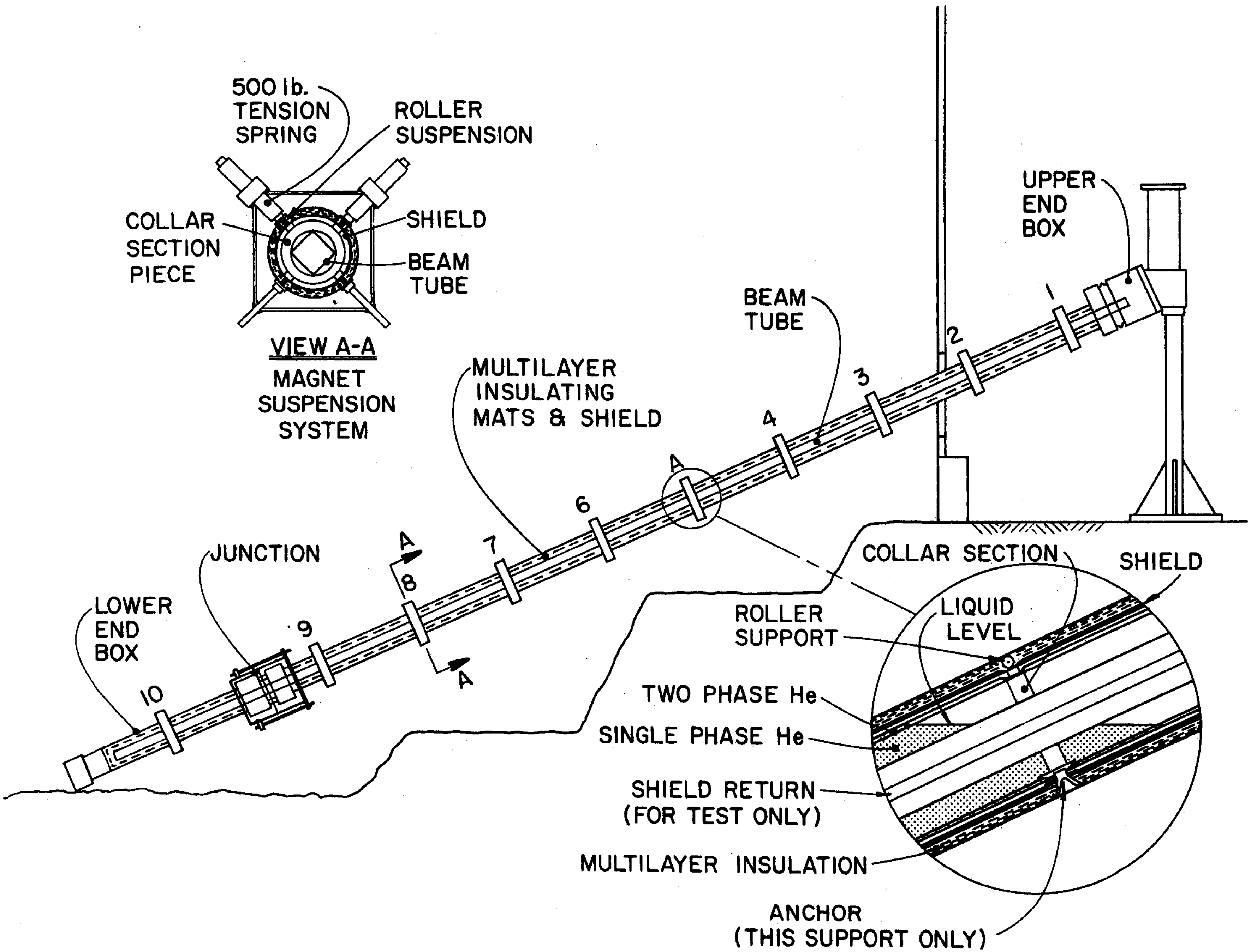

Figure 1 Experimental set up

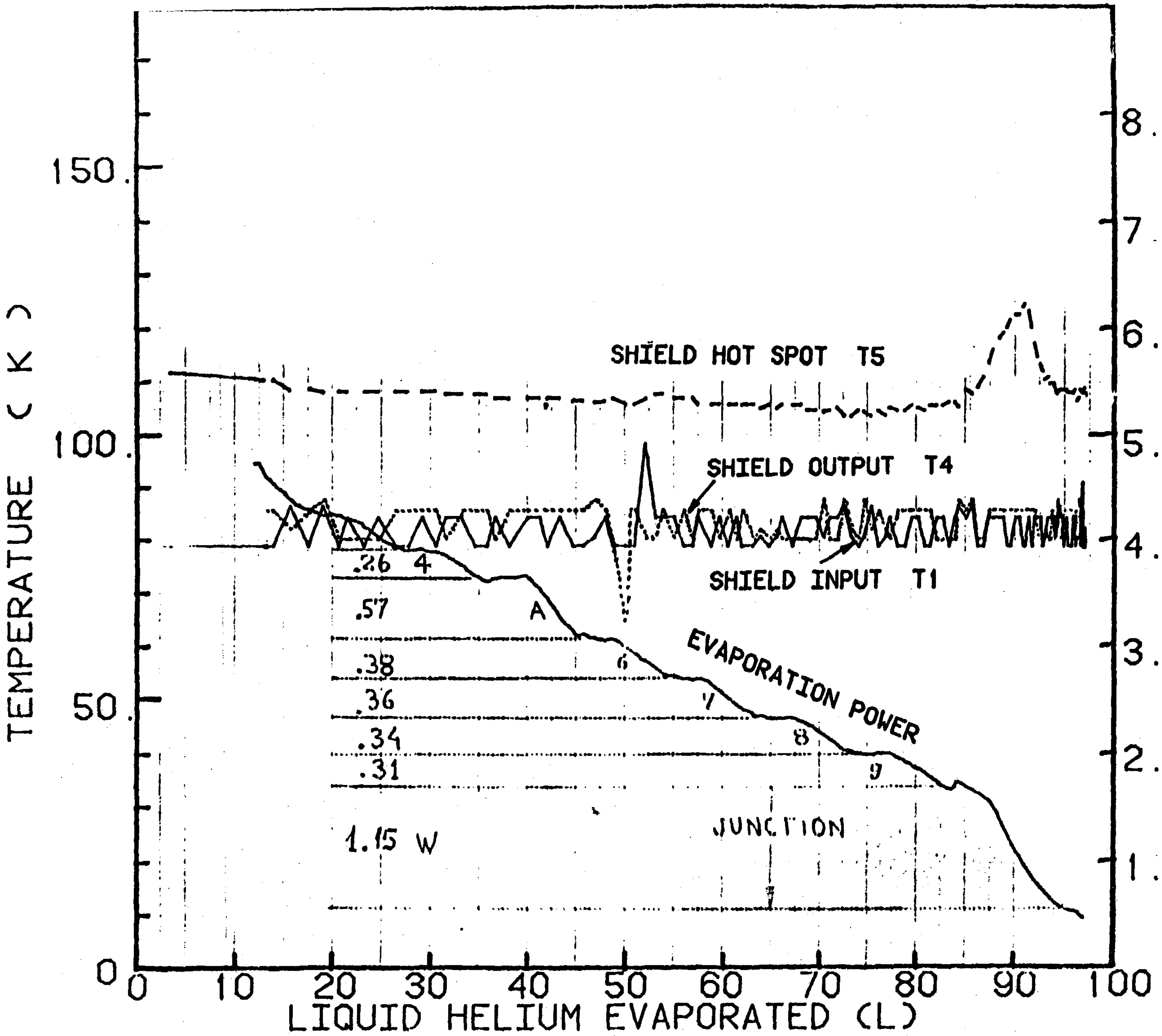

FIGURE 2 Data of Run 08. Spring compression force 516 lb. Shield cooled by boiling nitrogen.

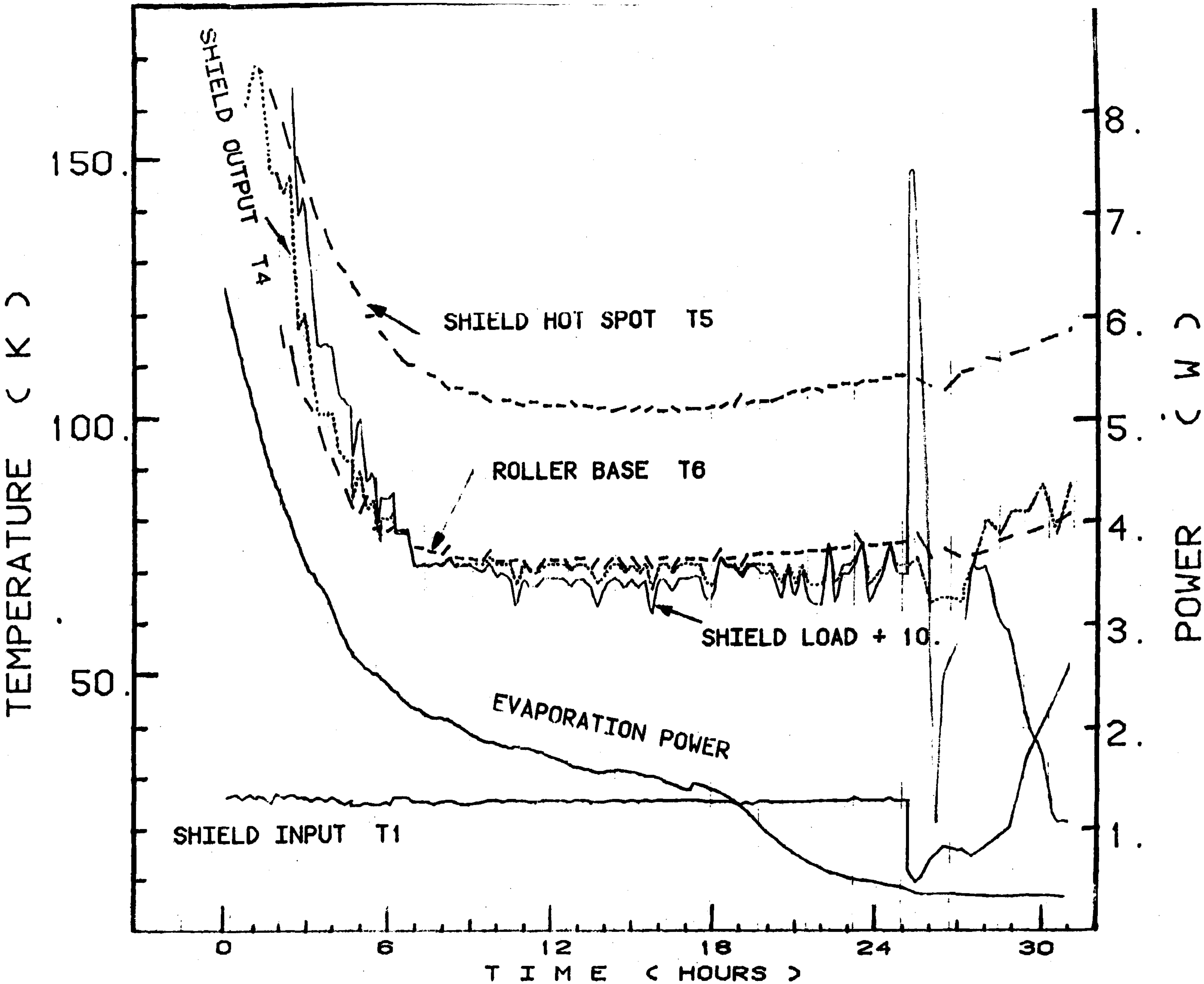

FIGURE 3 Data of Run 16. Spring compression force 516 lb. S ɜld cooled by 25K helium gas.

TM-740

```
C       ROIL.FOR-AUTOMATIC DATA ACQUISITION FOR E22-14 BOIL OFF
        COMMON/STORE/MB30,MB31(980),MB32(980),MB33(980)
        180)
        COMMON/BUFFR/MB1(5,100),MB2(5,100),NB1,NB2,NB0
        COMMON/CLOCK/K00,K01,L10,L11,L20,L21,L40,L41
        COMMON/COUNT/NB3,NB4,NB5,NB6,NB7,NB8,NFD,IREC
        COMMON/WATCH/DELAY,DT,KT
        COMMON/VALUE/D3(16),T11,T28,T9,T10
        COMMON/CALIB/C1,C2,C3,C4,C5
        DIMENSION MD1(16),MD2(16),D1(16)
        CALL NLC
        MB30=979
        NB3=0
        NFD=0
        KT=0
        DT=1800.
        DELAY=2.
        NB7=2
        IREC=17554
        NB8=2
        NB0=200
        NB6=0
        NB4=0
        NB5=0
        NB1=0
        NB2=0
        C1=2.442E-3
        C2=68.046E-3
        C3=101.56E-3
        C4=1.
        C5=.6804
        K00=0
        K01=0
        L10=0
        L11=0
        L20=0
        L21=0
        L40=0
        L41=0
        CALL BUF(NB0,MB1(1,1),MB2(1,1),NB1,NB2)
        CALL TP (K00,K01,L10,L11,L20,L21,L40,L41)
C       GAS METERS P,T LAST VALUES(INITIAL GUESS)
        T9=.001/C1
        T10=.01/C1
        T11=300.
        T28=300.
        D3(1)=1.
        D3(3)=1.
        D3(O)=0.
        D3(11)=1.
        CALL CLOK (77,30000)
        CALL CAMAC
C       NEW FLOPPY DISK ASSIGNMENT
        CALL ASSIGN(6,'DX1:BOILNN.DAT',-14)
        DEFINE FILE 6(IREC,7,U,NB8)
C       SENSE CPU SETTING
4       I=1
        CALL BITZ (I)
        IF(I.EQ.0) GO TO 5
        CALL MANUAL
5       I=2
        CALL BITZ(I)
        IF(I.EQ.0) GO TO 6
        CALL PROCES
6       I=3
        CALL BITZ(I)
        IF(I.EQ.0) GO TO 7
        CALL PARAM
```

```
C       AUTOMATIC DATA COLLECTION
7       IF(NB1.LT.NB0) GO TO 8
        NBU=1
        CALL UNLO(NBU)
8       IF(NB2.LT.NB0) GO TO 9
        NBU=2
        CALL UNLO(NBU)
9       ETIME=DT*FLOAT(KT)
        TIM=TIME(K00,K01)
        IF(TIM.LT.ETIME) GO TO 4
        IFLAG=1
        KT=KT+1
C       READ IN RAW VALUES
        DO 10 IA=1,14
        CALL CMCI (0,1,IA,MD1(IA),MD2(IA))
        D1(IA)=MD1(IA)
        D1(IA)=C1*D1(IA)
10      CONTINUE
        D3(1)=D1(1)*C2+C5
        D3(9)=D1(9)/51.00
        D3(5)=D1(5)*D3(9)
        D3(6)=D1(6)*D3(9)
        D3(8)=D1(8)*C3/2.
        D3(10)=10.**((50.*D1(10))-9.)
        D3(11)=D1(11)*C3
        D3(3)=D1(3)*C2*C4+C5
        D3(14)=D1(14)*27.733
        D3(2)=D1(2)/10.
        D3(4)=D1(4)
        D3(7)=D1(7)
        D3(12)=D1(12)
        D3(13)=D1(13)
C       FORM BUFFER B3
        DO 12 IA=1,14
        IF(NB3.GT.MB30) NB3=0
        NB3=NB3+1
        MB31(NB3)=IA+11
        MB32(NB3)=MD1(IA)
        MB33(NB3)=MD2(IA)
        B34(NB3)=D3(IA)
        B35(NB3)=TIM
        NB5=NB5+1
12      CONTINUE
        IF(NB5.LT.64) GO TO 15
        CALL TRANS
C       READ TEMPERATURES
15      IF(IFLAG.EQ.0)GO TO 4
        IA=9
        CALL THER(IA)
        DO 13 IA=1,8
        CALL THER(IA)
13      CONTINUE
        DO 16 IA=10,14
        CALL THER(IA)
16      CONTINUE
        IF(NB7.GT.14)NB7=2
        I=2**NB7+1
C       LEAVE RELAYS READING +T2 OR TNB7
        CALL CMCO(16,11,0,I,16)
        CALL CMCO(16,11,0,I,16)
        IFLAG=0
        I=4
        CALL BITZ(I)
        IF(I.EQ.0)GOTO 4
        CALL BELL
        GO TO 4
        END
```

```
        SUBROUTINE TRANS
        COMMON/STORE/MB30,MB31(980),MB32(980),MB33(980),B34(980),B35(9
        180)
        COMMON/BUFFR/MB1(5,100),MB2(5,100),NB1,NB2,NB0
        COMMON/CLOCK/K00,K01
        COMMON/COUNT/NB3,NB4,NB5,NB6,NB7,NB8,NFD,IREC
        COMMON/WATCH/DELAY,DT,KT
        COMMON/VALUE/D3(16),T11,T28,T9,T10
        COMMON/CALIB/C1,C2,C3,C4,C5
        T=TIME(K00,K01)
        TTIM=T/60.
        ITRAP=0
        I43=40
        L=5
        CALL BITZ(L)
        IF(L.EQ.0)GOTO 20
        WRITE(7,8) TTIM
8       FORMAT (' TRANS ON ',F6.0,'TH MINUTE')
20      CONTINUE
7       DO 1  J=1,64
        NB6=NB6+1
        IF(NB6.GT.MB30)NB6=0
        NB5=NB5-1
        IF(MB31(NB6).EQ.40) GOTO 9
        IF(NB5.LE.0)GOTO 10
        WRITE (6'NB8) MB31(NB6),MB32(NB6),MB33(NB6),B34(NB6),B35(NB6)
1       CONTINUE
10      IF(ITRAP.EQ.1)GOTO 16
        IF(NB8.LT.IREC)GOTO 3
        IREC=0
16      WRITE(7,2)NFD,NB8
        NB8=1
        WRITE(6'NB8)I40,NFD,NB8,T,T
        CALL CLOSE (6)
        FORMAT(' FILE',I3,' HAS',I5,' RECORDS')
6       WRITE (7,4)
4       FORMAT (' RESET SWITCH 0 + PRESS RETURN FOR NEW FILE',)
13      I=0
        CALL BITZ(I)
        IF(I.NE.0)GOTO 14
        CALL BELL
        GOTO 13
14      READ (5,5)
5       FORMAT (' ')
        IF(IREC.LT.54)GOTO 17
        NFD=NFD+1
        CALL ASSIGN(6,'DX1:BOILNN.DAT',-14)
        DEFINE FILE 6(IREC,7,U,NB8)
        NB8=2
3       IF(NB5.GE.64) GO TO 7
        RETURN
9       IREC=((IREC-(NB8+NB1/10+NB2/10))/37)*36
C       RETRIEVE LAST BUFFER
        IF(NB1.LE.0)GOTO 11
        NBU=1
        CALL UNLO(NBU)
11      IF(NB2.LE.0)GOTO 15
        NBU=2
        CALL UNLO(NBU)
15      ITRAP=1
        GOTO 7
17      WRITE(7,18) IREC
        FORMAT(' DISC FULL,IREC=',I5/' INSERT NEW DISK')
        IREC=17554
        GOTO 6
        END
```

```
      SUBROUTINE UNLO(NBU)

      COMMON/STORE/MB30,MB31(980),MB32(980),MB33(980),B34(980),B35(9
     180)
      COMMON/BUFFR/MB1(5,100),MB2(5,100),NB1,NB2,NB0
      COMMON/CLOCK/K00,K01
      COMMON/COUNT/NB3,NB4,NB5,NB6,NB7,NB8,NFD,IREC
      COMMON/WATCH/DELAY,DT,KT
      COMMON/VALUE/D3(16),T11,T28,T9,T10
      COMMON/CALIB/C1,C2,C3,C4,C5
      TIM=TIME (K00,K01)/60.
      L=5
      CALL BITZ(L)
      IF(L.EQ.0)GOTO 20
      WRITE(7,1)NBU,TIM
1     FORMAT (' UNLO',I1,' ON ',F6.0,'TH MINUTE')
20    CONTINUE
      NL=NB1/10
      IF(NBU.EQ.2)NL=NB2/10
      DO 5 N=1, NL
      IF (NBU.EQ.1) GO TO 2
      IF (NBU.EQ.2) GO TO 3
4     RETURN
2     IF(NB3.GT.MB30) NB3=0
      NB3=NB3+1
      MB31(NB3)=MB1(1,N)+40
      TL0=MB1(2,N)
      TL1=MB1(3,N)
      MB32(NB3)=MB1(4,N)
      MB33(NB3)=MB1(5,N)
      GO TO 6
3     IF(NB3.GT.MB30) NB3=0
      NB3=NB3+1
      MB31(NB3)=MB2(1,N)+40
      TL0=MB2(2,N)
      TL1=MB2(3,N)
      MB32(NB3)=MB2(4,N)
      MB33(NB3)=MB2(5,N)
C     PROCESSING
6     B35(NB3)=500.*TL1+TL0/60.
      IF(MB31(NB3)-42) 7,8,9
7     P=(D3(1)+D3(11))/2.
      T=T11
      V=.25
      GO TO 10
8     P=D3(8)+D3(11)
      T=T23
      V=1.
      GO TO 10
9     P=(D3(1)+D3(11))/2.
      T=T11
      V=1.
10    IF(P.LE.1.) GO TO 11
      B34(NB3)=(1379.3*P+1.2)*V/T
      GO TO 12
11    B34(NB3)=(1377.3*P+3.2)*V/T
12    NB5=NB5+1
5     CONTINUE
      IF(NBU.EQ.1) GO TO 13
      NB2=0
      GO TO 14
13    NB1=0
14    IF(NB5.LT.64) GO TO 4
      CALL TRANS
      GO TO 4
      END


      FUNCTION TIME (K00,K01)
```

```
      CALL SEC(K00,K01)
      H=K01
      D=30000-K00
      TIME=500.*H+D/60.
      RETURN
      END


      SUBROUTINE PROCES

      RETURN
      END

      SUBROUTINE PARAM

      COMMON/STORE/MB30,MB31(980),MB32(980),MB33(980),B34(980),B35(9
      180)
      COMMON/BUFFR/MB1(5,100),MB2(5,100),NB1,NB2,NB0
      COMMON/CLOCK/K00,K01
      COMMON/COUNT/NB3,NB4,NB5,NB6,NB7,NB8,NFD,IREC
      COMMON/WATCH/DELAY,DT,KT
      COMMON/VALUE/D3(16),T11,T28,T9,T10
      COMMON/CALIB/C1,C2,C3,C4,C5
      WRITE(7,1)
1     FORMAT(' ENTER CODE AND NEW INTEGER:',$)
      READ(5,2)IC,IN
2     FORMAT(2I5)
      T=TIME(K00,K01)
      IF(IC.EQ.0.OR.IC.EQ.1) GOTO 3
      IF(IC.EQ.2)NB0=IN
      IF(IC.EQ.3)DELAY=IN
      IF(IC.EQ.4)GOTO 5
      IF(IC.EQ.5)GOTO 6
      IF(IC.EQ.6)NB7=IN
      RETURN
C     TERMINATE FILE DX1:
3     IF(NB3.GT.MB30)NB3=0
      NB3=NB3+1
      MB31(NB3)=40
      MB32(NB3)=NB1
      MN33(NB3)=NB2
      B34(NB3)=T
      B35(NB3)=T
      NB5=NB5+1
      CALL TRANS
      IF(IC.EQ.0)GOTO 4
      CALL STCL
      STOP
C     NEW INTERVAL BETWEEN READINGS
5     DT=IN
      IF(DT.EQ.0)GOTO 4
      KT=(T/DT)-1.
      GOTO 4
C     LIST OUT PROGRAM COUNTERS
6     WRITE(7,7)NB0,NB1,NB2,NB3,NB4,NB5,NB6,NB7,NB8,NFD,IREC
      1,K00,K01,T,DELAY,DT,KT
7     FORMAT(' NB0=',I3,' NB1=',I3,' NB2=',I3/
      1' NB3=',I5,' NB4=',I5,' NB5=',I5,' NB6=',I5,' NB7=',I5,' NB8=
      2',I5/' NFD=',I2,' IREC=',I5/
      3' K00=',I5,' K01=',I5,'    TIME=',F8.0,' SEC'/
      4' DELAY=',F4.0,' SEC  DT=',F6.0,' SEC  KT=',I5)
      GOTO 4
      END


      SUBROUTINE THER (IA)
C     READ ONE THERMOMETER

      COMMON/STORE/MB30,MB31(980),MB32(980),MB33(980),B34(980),B35(9
```

```
      180)
      COMMON/BUFFR/MB1(5,100),MB2(5,100),NB1,NB2,NB0
      COMMON/CLOCK/K00,K01
      COMMON/COUNT/NB3,NB4,NB5,NB6,NB7,NB8,NFD,IREC
      COMMON/WATCH/DELAY,DT,KT
      COMMON/VALUE/D3(16),T11,T28,T9,T10
      COMMON/CALIB/C1,C2,C3,C4,C5
      NR=0
      IF(IA.GE.10)NR=15
      NN=0
      IB=IA
      IF(IA.EQ.10)NN=8
      IF(IA.EQ.28)IB=12
      IF(IA.EQ.45)IB=13
      IF(IA.EQ.33)IB=14
      N=2**IB
      IF(IA.EQ.10)N=0
C     SET RELAY
      CALL CMCO (16,11,0,N,NN)
      CALL CMCO (16,11,0,N,NN)
      CALL WAITER
      CALL CMCI (0,1,NR,MP0,MP1)
      N=N+1
      IF(IA.GE.10)N=N+1023
      CALL CMCO (16,11,0,N,NN)
      CALL CMCO (16,11,0,N,NN)
      CALL WAITER
      CALL CMCI(0,1,NR,MN0,MN1)
9     VP=MP0
      VN=MN0
      V=ABS((VP-VN)/2.)
      IF(IA.EQ.9) GO TO 1
      IF(IA.EQ.10) GO TO 2
      IF(IA.GT.10) GO TO 3
      IF(T9.EQ.0.)GOTO 5
      R=V/T9
      GO TO 4
3     IF(T10.EQ.0.)GOTO 5
      R=V/T10
4     TEMP=TABLE(IA,R)
C     FORM BUFFER B3 RECORD
6     IF(NB3.GT.MB30) NB3=0
      NB3=NB3+1
      MB31(NB3)=IA
      IF(IA.EQ.11)T11=TEMP
      IF(IA.EQ.28.OR.IA.EQ.12)T28=TEMP
      MB32(NB3)=MP0
      MB33(NB3)=MN0
      B34(NB3)=TEMP
      B35(NB3)=TIME(K00,K01)
      NB5=NB5+1
      IF(NB5.LT.64) GO TO 5
      CALL TRANS
5     RETURN
C     CALIBRATION RESISTORS
1     T9=V/100.11
      TEMP=T9*C1
      GOTO 6
2     T10=V/99.95
      TEMP=T10*C1
      GO TO 6
      END


      FUNCTION TABLE (IA,RES)

      DIMENSION T(8,23),R(8,23),NE(8)
      DATA NE(1),NE(2),NE(3),NE(4),NE(5),NE(6),NE(7),NE(8)
      1/23,16,18,23,16,21,18,23/
C     TEMP=1 IA ERROR
```

```
C         TEMP=2. RESISTANCE OUTSIDE CALIBRATED RANGE
          IF(IA.GT.8) GO TO 12
          N=IA
          GO TO 13
12        IF(IA.EQ.28.OR.IA.EQ.45.OR.IA.EQ.33.OR.IA.EQ.11) N=8
          IF(N.EQ.8) GO TO 13
          IF(N.GT.8) TEMP =1.
          GO TO 9
13        MAX=NE(N)-1
          DO 3 J=1,MAX
          IF(RES.LE.R(N,J).AND.RES.GE.R(N,J+1)) GO TO 2
          IF(RES.GE.R(N,J).AND.RES.LE.R(N,J+1)) GO TO 2
3         CONTINUE
          TEMP=2.
          GO TO 9
2         IF(RES.EQ.R(N,J)) GO TO 7
          IF(RES.EQ.R(N,J+1)) GO TO 8
          TL=ALOG(T(N,J)/T(N,J+1))
          RL=ALOG(R(N,J)/R(N,J+1))
          IF(RL.EQ.0.)GOTO 3
          E=TL/RL
          TEMP=T(N,J)*(RES/R(N,J))**E
9         TABLE=TEMP
          RETURN
7         TEMP=T(N,J)
          GO TO 9
8         TEMP=T(N,J+1)
          GO TO 9
          DATA R(1,1),R(1,2),R(1,3),R(1,4),R(1,5),R(1,6)
          1,R(1,7),R(1,8),R(1,9),R(1,10),R(1,11),R(1,12),R(1,13)
          2,R(1,14),R(1,15),R(1,16),R(1,17),R(1,18),R(1,19),R(1,20)
          3,R(1,21),R(1,22),R(1,23)
          4/1580.90,1221.33,938.75,597.09
          1,446.95,294.19,192.38,136.14,110.67,76.548,58.963
          2,40.957,33.791,30.608,28.343,26.612,25.204,23.566
          3,22.569,21.623,20.455,19.957,19.600/
          DATA T(1,1),T(1,2),T(1,3),T(1,4),T(1,5),T(1,6)
          1,T(1,7),T(1,8),T(1,9),T(1,10),T(1,11),T(1,12),T(1,13)
          2,T(1,14),T(1,15),T(1,16),T(1,17),T(1,18),T(1,19),T(1,20)
          3,T(1,21),T(1,22),T(1,23)
          4/4.230,4.625,5.097,6.142,7.029
          1,8.757,11.373,14.656,17.434,24.88,33.939,57.878
          2,82.156,100.702,119.452,138.575,158.429,188.233
          3,211.386,236.713,276.316,295.763,310.961/
          DATA R(2,1),R(2,2),R(2,3),R(2,4),R(2,5),R(2,6)
          1,R(2,7),R(2,8),R(2,9),R(2,10),R(2,11),R(2,12),R(2,13)
          2,R(2,14),R(2,15),R(2,16)
          4/21.20,37.96,57.37,78.32,99.78
          1,121.23,143.18,185.58,227.49,269.40,310.30,351.21
          2,391.62,451.99,498.88,585./
          DATA T(2,1),T(2,2),T(2,3),T(2,4),T(2,5),T(2,6)
          1,T(2,7),T(2,8),T(2,9),T(2,10),T(2,11),T(2,12),T(2,13)
          2,T(2,14),T(2,15),T(2,16)
          4/40.,50.,60.,70.,80.,90.,100.
          1,120.,140.,160.,180.,200.,220.,250.,273.15,320.3/
          DATA R(3,1),R(3,2),R(3,3),R(3,4),R(3,5),R(3,6)
          1,R(3,7),R(3,8),R(3,9),R(3,10),R(3,11),R(3,12),R(3,13)
          2,R(3,14),R(3,15),R(3,16),R(3,17),R(3,18)
          4/907.48,659.22,532.63,427.66
          1,304.85,233.37,209.78,181.70,160.30,144.84,128.67
          2,118.69,111.16,106.45,104.13,102.41,101.71,100.81/
          DATA T(3,1),T(3,2),T(3,3),T(3,4),T(3,5),T(3,6)
          1,T(3,7),T(3,8),T(3,9),T(3,10),T(3,11),T(3,12),T(3,13)
          2,T(3,14),T(3,15),T(3,16),T(3,17),T(3,18)
          4/4.256,5.109,6.083,7.329,10.403
          1,13.548,17.574,22.854,29.957,38.920,54.227
          2,71.180,93.037,115.49,133.40,167.87,188.54,295.37/
          DATA R(4,1),R(4,2),R(4,3),R(4,4),R(4,5),R(4,6)
          1,R(4,7),R(4,8),R(4,9),R(4,10),R(4,11),R(4,12),R(4,13)
          2,R(4,14),R(4,15),R(4,16),R(4,17),R(4,18),R(4,19),R(4,20)
```

```
3,R(4,21),R(4,22),R(4,23)
4/1633.41,1243.34,954.83,605.58
1,451.04,294.54,192.74,135.79,110.44,76.808,58.942
2,41.296,34.105,30.693,28.527,26.812,25.419,23.750
3,22.764,21.823,20.679,20.202,19.925/
DATA T(4,1),T(4,2),T(4,3),T(4,4),T(4,5),T(4,6)
1,T(4,7),T(4,8),T(4,9),T(4,10),T(4,11),T(4,12),T(4,13)
2,T(4,14),T(4,15),T(4,16),T(4,17),T(4,18),T(4,19),T(4,20)
3,T(4,21),T(4,22),T(4,23)
4/4.230,4.625,5.097,6.142,7.029
1,8.757,11.373,14.656,17.434,24.881,33.939,57.878
2,82.156,100.70,119.45,138.58,158.43,188.23,211.39
3,236.71,276.32,295.76,310.961/
DATA R(5,1),R(5,2),R(5,3),R(5,4),R(5,5),R(5,6)
1,R(5,7),R(5,8),R(5,9),R(5,10),R(5,11),R(5,12),R(5,13)
2,R(5,14),R(5,15),R(5,16)
4/21.03,37.65,56.90,77.69,98.96
1,120.24,142.01,184.07,225.63,267.20,307.77,348.35
2,388.42,448.30,494.81,585./
DATA T(5,1),T(5,2),T(5,3),T(5,4),T(5,5),T(5,6)
1,T(5,7),T(5,8),T(5,9),T(5,10),T(5,11),T(5,12),T(5,13)
2,T(5,14),T(5,15),T(5,16)
4/40.,50.,60.,70.,80.,90.,100.
1,120.,140.,160.,180.,200.,220.,250.,273.15,322.93/
DATA R(6,1),R(6,2),R(6,3),R(6,4),R(6,5),R(6,6)
1,R(6,7),R(6,8),R(6,9),R(6,10),R(6,11),R(6,12),R(6,13)
2,R(6,14),R(6,15),R(6,16),R(6,17),R(6,18),R(6,19),R(6,20)
3,R(6,21)
4/27468.,17387.,13056.,10211.
1,8326.,8318.,7241.,6359.,3908.,3072.,2506.,2157.
2,1872.,1600.,1456.,1340.,1284.,1238.,1141.,1084.8
3,1044.6/
DATA T(6,1),T(6,2),T(6,3),T(6,4),T(6,5),T(6,6)
1,T(6,7),T(6,8),T(6,9),T(6,10),T(6,11),T(6,12),T(6,13)
2,T(6,14),T(6,15),T(6,16),T(6,17),T(6,18),T(6,19),T(6,20)
3,T(6,21)
4/4.378,5.39,6.252,7.180,8.165
1,8.167,8.997,9.934,15.635,20.826,27.865,35.659
2,46.521,64.629,80.925,100.91,114.32,128.29,172.15
3,219.06,286.95/
DATA R(7,1),R(7,2),R(7,3),R(7,4),R(7,5),R(7,6)
1,R(7,7),R(7,8),R(7,9),R(7,10),R(7,11),R(7,12),R(7,13)
2,R(7,14),R(7,15),R(7,16),R(7,17),R(7,18)
4/1130.,665.69,537.51,431.29
1,307.14,250.42,211.12,182.70,161.13,145.61,129.31
2,119.20,111.67,107.05,104.57,102.57,102.21,101.35/
DATA T(7,1),T(7,2),T(7,3),T(7,4),T(7,5),T(7,6)
1,T(7,7),T(7,8),T(7,9),T(7,10),T(7,11),T(7,12),T(7,13)
2,T(7,14),T(7,15),T(7,16),T(7,17),T(7,18)
4/4.2,5.109,6.083,7.329,10.403
1,13.548,17.547,22.854,29.957,38.92,54.227,71.18
2,93.037,115.49,133.40,167.87,188.54,295.37/
DATA R(8,1),R(8,2),R(8,3),R(8,4),R(8,5),R(8,6)
1,R(8,7),R(8,8),R(8,9),R(8,10),R(8,11),R(8,12),R(8,13)
2,R(8,14),R(8,15),R(8,16),R(8,17),R(8,18),R(8,19),R(8,20)
3,R(8,21),R(8,22),R(8,23)
4/9656.,3499.,2045.,782.1,507.9
1,232.7,224.4,208.7,201.4,194.3,187.6,181.1,174.9
2,163.3,152.5,142.6,133.4,125.0,109.9,94.0,48.5
3,23.1,10.1/
DATA T(8,1),T(8,2),T(8,3),T(8,4),T(8,5),T(8,6)
1,T(8,7),T(8,8),T(8,9),T(8,10),T(8,11),T(8,12),T(8,13)
2,T(8,14),T(8,15),T(8,16),T(8,17),T(8,18),T(8,19),T(8,20)
3,T(8,21),T(8,22),T(8,23)
4/196.,213.,223.,243.,253.,273.
1,274.,275.,276.,277.,278.,279.,280.,282.,284.
2,286.,288.,290.,295.,300.,323.2,353.2,393.2/
END
```

```
      SUBROUTINE MANUAL
      COMMON/STORE/MB30,MB31(980),MB32(980),MB33(980),B34(980),B35(9
      180)
      COMMON/BUFFR/MB1(5,100),MB2(5,100),NB1,NB2,NB0
      COMMON/CLOCK/K00,K01
      COMMON/COUNT/NB3,NB4,NB5,NB6,NB7,NB8,NFD,IREC
      COMMON/WATCH/DELAY,DT,KT
      COMMON/VALUE/D3(16),T11,T28,T9,T10
      COMMON/CALIB/C1,C2,C3,C4,C5
      DIMENSION U(10)
      N0=0
      N1=0
      N2=0
      W=0
      T=0
      WRITE(7,1)
1     FORMAT(' ID:',$)
      T=TIME(K00,K01)
      READ(5,2)IC
2     FORMAT(I3)
      IF(IC)9,3,10
C     NULL CODE:COMMENT
3     WRITE(7,4)T
4     FORMAT(' AT',F8.0,'SEC ENTER 8 ASCII COMMENT:    ',$)
      READ(5,5)N1,N2,W
5     FORMAT(2A2,A4)
C     FORM BUFFER B3 RECORD
17    IF(NB3.GT.MB30)NB3=0
      NB3=NB3+1
      MB31(NB3)=N0
      MB32(NB3)=N1
      MB33(NB3)=N2
      B34(NB3)=W
      B35(NB3)=T
      NB5=NB5+1
      IF(NB5.LT.64) GO TO 6
      CALL TRANS
6     IF(N0.NE.0) GO TO 13
      WRITE(7,7) N0,N1,N2,W,T
7     FORMAT(' ',I2,2A2,A4,F9.1)
      GO TO 8
C     NEGATIVE CODE: WRITE OUT LAST ENTRY
9     N0=-IC
16    DO 11 J=0,MB30
      JJ=NB3-J
      IF(JJ.LE.0) JJ=JJ+MB30
      IF(MB31(JJ).EQ.N0) GO TO 12
11    CONTINUE
      GO TO 8
12    N1=MB32(JJ)
      N2=MB33(JJ)
      W=B34(JJ)
      T=B35(JJ)
      IF(N0.EQ.0) GO TO 6
C     SELECT UNIT
13    IF(N0.LT.9.OR.N0.EQ.11.OR.N0.EQ.28.OR.N0.EQ.33.OR.N0.EQ.45)
      1IU=1
      IF(N0.EQ.9.OR.N0.EQ.10)IU=2
      IF(N0.EQ.12.OR.N0.EQ.14.OR.N0.EQ.19.OR.N0.EQ.22)IU=3
      IF(N0.EQ.13)IU=4
      IF(N0.EQ.15.OR.N0.EQ.18.OR.N0.EQ.23.OR.N0.EQ.24)IU=5
      IF(N0.EQ.16.OR.N0.EQ.17)IU=6
      IF(N0.EQ.41.OR.N0.EQ.42.OR.N0.EQ.44)IU=7
      IF(N0.EQ.20)IU=8
      IF(N0.EQ.21)IU=9
      IF(N0.EQ.25)IU=10
      WRITE(7,14)N0,N1,N2,W,U(IU),T
14    FORMAT(' ',I2,2I6,E11.4,A4,F9.1,'SEC')
8     RETURN
```

```
C       POSITIVE CODE:READ IN AND WRITE OUT
10      N0=IC
        IF(N0.EQ.50)GOTO 18
        IF(N0.LE.11.OR.N0.EQ.28.OR.N0.EQ.33.OR.N0.EQ.45) GO TO 15
        IF(N0.EQ.41.OR.N0.EQ.42.OR.N0.EQ.44) GO TO 16
C       LECROY DATA
        IA=IC-11
        CALL CMCI(0,1,IA,N1,N2)
        W=N1
        W=W*C1
C       PROCESS DATA
        IF(IA.EQ.1) W=W*C2+C5
        IF(IA.EQ.9) W=W/51.
        IF(IA.EQ.5.OR.IA.EQ.6) W=W*D3(9)
        IF(IA.EQ.8) W=W*C3/2.
        IF(IA.EQ.10) W=10.**((50.*W)-9.)
        IF(IA.EQ.11) W=W*C3
        IF(IA.EQ.3) W=W*C2*C4+C5
        IF(IA.EQ.14) W=27.733*W
        IF(IA.EQ.2) W=W/10.
        GO TO 17
C       TEMPERATURE DATA
15      CALL THER(N0)
        GO TO 16
        DATA U(1),U(2),U(3),U(4),U(5),U(6),U(7),U(8),U(9)
        1,U(10)/' K  ','(MA)',' ATM',' FFS',' V  '
        2,' W  ','G HE',' A  ','TORR','MG/S'/
18      IF(NB7.GT.14)NB7=2
        I=2**NB7+1
C       LEAVE RELAYS READING +T2 OR TNB7
        CALL CMCO(16,11,0,I,16)
        CALL CMCO(16,11,0,I,16)
        GOTO 8
        END

        SUBROUTINE WAITER

        COMMON/STORE/MB30,MB31(980),MB32(980),MB33(980),B34(980),B35(9
        180)
        COMMON/BUFFR/MB1(5,100),MB2(5,100),NB1,NB2,NB0
        COMMON/CLOCK/K00,K01
        COMMON/COUNT/NB3,NB4,NB5,NB6,NB7,NB8,NFD,IREC
        COMMON/WATCH/DELAY,DT,KT
        COMMON/VALUE/D3(16),T11,T28,T9,T10
        COMMON/CALIB/C1,C2,C3,C4,C5
        BONG=TIME(K00,K01)+DELAY
1       T=TIME(K00,K01)
        IF(BONG.GT.T)GO TO 1
        RETURN
        END
```