R 81-2229

Auto: A Real Time Diffractometer Control System

Robert L. Snyder

Alfred University
Alfred, NY 14802

Camden R. Hubbard

Center for Materials Science
U.S. Department of Commerce
National Bureau of Standards
Washington, DC 20234

and

Nicolous C. Panagiotopoulos

JCPDS-International Centre for
Diffraction Data Research Associateship at
National Bureau of Standards
Washington, DC 20234

February 1981

U.S. DEPARTMENT OF COMMERCE
NATIONAL BUREAU OF STANDARDS

March 17, 1981

Enhancements to AUTO

Since the writing of this report, we have continued to enhance the program AUTO. We have also added a few more file handling subroutines and simplified the structure to permit overlaying. The changes are summarized below:

- Added PKFIND and SMTH13 subroutines which are called when peak height intensities are to be measured. They assure that the maximum is recorded in the peak height mode of selected region data collection.

- Divided QUANT into BGQUAN and RNQUAN in order to separate the initialization for quantitative data collection from the actual control.

- Rewrote STDFIL (new name RUNFIL) so that it may be initiated by a consol option instead of from QUANT.

- Added DELETE, OPEN and CLOSE subroutines for file handling.

- Removed all calls to POP from subroutines such as BGSLEW, QUAL, etc. (18 occurrences). Added call to POP to MAIN just before computed GOTO.

- Added BLOCKDATA subroutines in order to shorten subroutine INIT.

- On many of the inputs requested of the user, an 'X' input will terminate the current activity and return to subroutine INTRUP.

An up-to-date listing can be obtained from C. R. Hubbard, A221/Matls., National Bureau of Standards, Washington, D. C. 20234.

NBSIR 81-2229

AUTO: A REAL TIME DIFFRACTOMETER CONTROL SYSTEM

Robert L. Snyder

Alfred University
Alfred, NY 14802

Camden R. Hubbard

Center for Materials Science
U.S. Department of Commerce
National Bureau of Standards
Washington, DC 20234

and

Nicolous C. Panagiotopoulos

JCPDS-International Centre for
Diffraction Data Research Associateship at
National Bureau of Standards
Washington, DC 20234

February 1981

U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, *Secretary*
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Director*

TABLE OF CONTENTS

INTRODUCTION

The purpose of this system is to control simultaneously two powder diffractometers mounted on the same x-ray generator. The controlling computer provides the user with the ability to perform all conventional diffraction analyses from the scanning of an entire spectrum to the careful measurement of the intensities of individual lines for quantitative analysis.

HARDWARE

The hardware consists of an Interdata 7/16 minicomputer[†] with 64K bytes of memory. Other hardware features include power fail interrupt generation, integer multiply and divide, precision interval clock, and selector channel. The operating system is FLOXOS, an NBS derivative of the Interdata DOS operating system. The minicomputer peripherals consist of two floppy disk drives (200K bytes each disk), two hard disk drives (5 M bytes each disk), a high speed lineprinter/plotter, CRT console, and the NBS Bus instrument interface. The NBS Bus interface contains two timer/stepping motor driver cards. Each card is capable of a variety of functions including operation as a scaler and operation as a pulse generator for input to the stepping motor power supply. The scaler completes a cycle on either preset time or preset counts. The pulse generator features include a tracking counter which maintains a representation of the current angle and the ability to send out a preselected number of pulses to the stepping motor power supply. The minicomputer is connected to the NBS NET via an RS232 serial interface port. This connection permits file transfer to other computers attached to the net.

Each goniometer is capable of stepping in increments of 0.005° 2θ. The high and low angle limit switches, if tripped, terminate output in the same direction from the motor power supply and signal the NBS Bus interface card. The card then terminates operation and puts a flag in the card status register.

INTERRUPTS

The Interdata 7/16 allows for a variety of interrupts such as power fail, clock and console escape. Each interrupt, if enabled, stores a two character ASCII string on a stack.

[†]Identification of the minicomputer in this paper is given in order to adequately specify the equipment for which this program was written. In no case does such identification imply recommendation or endorsement by NBS, nor does it imply that the equipment identified is necessarily the best available for the purpose.

The contents of the top of this stack are obtained with the Interdata FORTRAN function IFLAG(Ø). Completion of an activity by the NBS Bus could have been handled by an interrupt and a string on the interrupt stack. Instead, we chose to poll each card in the NBS Bus in sequence using the FORTRAN function ISTAT. This informs the program which interface card, and thus which unit, has completed an operation.

Because, in general, one is not following changes in real time in x-ray powder diffraction, an interrupt signaling instrument completion of a task does not need to be handled immediately. For example, at the completion of a counting cycle neither the scaler nor goniometer will change state until it is directed by the computer. Thus, in program AUTO, the testing of the interrupt stack and the NBS Bus polling performed in subroutine INTRUP may be suspended without consequence other than slowing down data collection. However, other than when entering run parameters or during a file transfer, the program AUTO returns to the subroutine INTRUP within a fraction of a second.

SYSTEM DESIGN

Since two diffractometers need to be controlled simultaneously, a conventional, dedicated type of program design could not be employed. Instead, we adopted an event-driven approach where most of the computer time is spent in a loop waiting for an interrupt from the console or the diffractometer interface. For this reason, program AUTO is better thought of as a real-time executive computer-control system. The flow of logic which follows an interrupt, is controlled by a last-in-first-out (LIFO) stack of addresses as is done in most modern computer system software. However, as our system is written exclusively in FORTRAN, the LIFO stack contains integers which are used as the arguments of a FORTRAN computed GO TO statement. These arguments (or address pointers) are listed in Table 1. The contents of the LIFO stack are modified by subroutines PUSH and POP. Subroutine PUSH inserts an address at the top of the stack and moves all the previously stacked addresses down one. Subroutine POP removes the top address and moves all others up.

The main program controls the flow of logic by a master computed GO TO statement which allows access to the subroutines shown in Table 2. The subroutines listed in parentheses in Table 2 are called as secondary subroutines by the routine listed at the left. Operating system and NBS Bus servicing functions and subroutines are listed in Table 3. The principal options of the system allow for powder diffraction pattern collection (referred to as qualitative analysis), intensity data collection for specified peaks (referred to as quantitative

analysis), the moving of the detector to any specified 2θ angle at maximum speed (referred to as slewing) and the termination of any active operation.

In order to eliminate gear backlash in slewing, the goniometer is always driven to 0.05° below the target angle and then moved up to the target angle. Data collection then proceeds with stepping in the increasing 2θ direction.

FILE NAMING

In a multiuser environment, certain naming conventions must be adhered to in order to minimize confusion. The system uses two types of files: an output data file from either pattern collection or quantitative analysis; and a run file directing a quantitative analysis or relative intensity measurement run. The first rule is that an output data file must end in "D" and that a run file must end in "R". The naming and allocation of all data files is handled by three subroutines. Subroutine FILEUP first reads the date and time from the system clock by calling subroutine DATE and then requests the file name from the user. It also requests the user to enter the title of the run and the operator's name. Subroutine FILNAM is then called to check the legality of the file name. The full rules are:

characters 1 & 2 = The operators initials

characters 3, 4 & 5 = any three letters or numbers

character 6 = "R" or "D"

The name must have six characters and not contain any blanks or special symbols. The last subroutine involved in file control is ALOT which allocates the data or run file on disk number DØ and writes the first five records, as described in Table 5 or 6, depending on the type of run.

PROGRAM DESCRIPTION

The following section gives a sketch of the flow and logic of the automation program AUTO. Each of the major options is discussed. The flow of this sketch is representive of a plausible sequence; however, for purposes of clarity of presentation, all options and possible branches are not presented. The source language listings (Appendix 1) serve as the complete program description.

1. INITIALIZATION: Program AUTO first calls subroutine INIT to initialize the run parameters, the interface, and the two diffractometers. This subroutine sets all Input and Output (I/O) logical units (Table 4) and attaches the required physical devices (NBS Bus, precision interval clock, date and time clock, console terminal and printer) to the I/O

units via the FLOXOS subroutine SYSTEM. It then requests the user to enter the current 2θ position of both diffractometers. Both angles are required in order for software collision protection logic to be used. Subroutine INIT then calls SETANG, which clears each interface, checks the legality of the input angle (by calling CKANG) and setting the angle into the interface tracking counter.

Subroutine INIT also attaches the instrument configuration file called DACONF written by program CONFIG (Appendix 2). This file contains the values of all instrument parameters for the x-ray generator and goniometer and the coefficients of the external calibration polynomial for each goniometer. The configuration parameters are not used to control data collection. They are, however, available for use in data analysis routines. Subroutine INIT is called only once when the AUTO system is initially executed.

2. CONSOLE OPTIONS: On completing the initialization section, subroutine CONSOL is called. After the initial entry, CONSOL can also be called by hitting the "escape" key on the console CRT. When "ESCAPE" is pushed the operating system places an interrupt flag on the flag table. Subroutine INTRUP detects this flag and sets the first address of the LIFO stack (which is a variable called ITRANS) to a 1 and returns to MAIN. MAIN uses the 1 as an argument in the master computed GO TO to branch to the call to CONSOL. On entering CONSOL, the 1 on the LIFO stack is removed (popped off) by subroutine POP. Subroutine POP is called by most of the subroutines in the AUTO system and works in conjunction with a MODE flag. In general if the MODE flag, associated with a subroutine, is set to 0 it means that the subroutine was called by another subroutine (i.e., not by MAIN using the LIFO stack) and, therefore, the LIFO stack should not be popped. If the MODE flag is set to 1, the subroutine was called by MAIN via the LIFO stack and, therefore, the subroutine's GO TO address should be popped off of the stack. Not all subroutines make use of the MODE flag. Subroutine CONSOL offers the following options to the user (certain option codes are reserved for future expansion):

0: Print the current status of each unit (activity, file name, estimated completion time, current 2θ) and return to the interrupt detection subroutine INTRUP by pushing its GO TO address (12) onto the LIFO stack and returning to MAIN.

1: Perform a powder diffraction pattern data collection by calling subroutine QUA1 via the the LIFO stack.

2: Perform a quantitative analysis, relative intensity or instrument calibration run. This is done by calling subroutine QUANT via the LIFO stack.

4: Reset the tracking counter of either interface by calling subroutine SETANG. This option also resets the units' activity flag (variable IACT). This flag is Ø when the interface is inactive, 1 when counting, 2 when stepping, 3 when slewing, and -1 when any type of interface error has been detected. In the last case, option 4 is the only way of resetting an interface error and reinitializing the unit.

5: Slew to a user input 2θ value by calling SLEW, INTRUP, ENSLEW, and INTRUP via the LIFO stack.

6: Kill the job on one of the diffractometers in either of two modes. The first mode causes an immediate halt of the Unit by calling FINIS in reset mode. This option causes the tracking counter to drop one or two pulses (if a move was halted) and, therefore, option 4 above must be used to reinitialize the unit. The second mode kills the current job after the next interface completion occurs. This normal termination leaves the unit ready for a new run. It should be noted that both terminations will write any remaining data to the output file and close it with an endfile mark, so that the data collected so far during the run can be read without error. The reset sequence involves calling SLEW, INTRUP, ENSLEW, and INTRUP to reset the 2θ angle to 80°. This call sequence is made by MAIN via the LIFO stack.

7: Panic stop of both units by immediately clearing the NBS Bus and calling EXIT. Any buffered data will be lost.

C: List contents of a disk on the console.

D: Turn the debug flag for each diffractometer on or off. When the flag is on a tracing of the subroutine calls is produced on the printer.

E: Delete a file name from a disk.

F: Transfer files from hard disk to a floppy disk, to the printer, or to another computer via the NBS NET by calling subroutine TRANS via the LIFO stack.

3. POWDER PATTERN COLLECTION: Subroutine QUAL is called from MAIN via the LIFO stack, which was set by subroutine CONSOL at the user's request. It first requests the starting angle for the pattern to be determined. It then sets the LIFO stack to call SLEW in order to drive to this target angle (set into variable TANG). It further sets the QUAL mode flag

so that when subroutine QUAL is reentered, after the SLEW is initiated, it branches to the logic after the SLEW command. This is one of the best examples of the advantages of a LIFO stack in controlling logic flow.

While the SLEW motion is occurring, subroutine QUAL calls FILEUP which asks the user to input the six letter name of the disk output file, into which the data for the pattern will be written. In addition, FILEUP, which is also called by subroutine QUANT, requests the input of the run title and operator's name. It then reads the date and time from the system clock via subroutine DATE and returns to QUAL.

Subroutine QUAL next requests the parameters for the run: Starting angle, ending angle, step size in degrees, count time per point, counting method, codes for the internal standards being used, and the maximum number of counts to be collected at any point (default = 32700). The subroutine allows for reentry of these numbers, each time outputting the length of time the run would take, until the user is satisfied with a set of run parameters. At this point, the necessary size of the output data file is computed, the file is allocated on disk and five header records are written on it by subroutine ALOT. The format of the output file is given in Table 5.

Record 6, containing the run parameters, is written by subroutine QUAL just before it sets up the LIFO stack with the address of the step scan (STPSCN) subroutine which carries out the entire data collection procedure.

4. STEP SCAN: Subroutine STPSCN carries out a move and count procedure. When 16 data points have been collected it writes them to the output data file by calling subroutine WRFILE which also resets the array. The independent step-then-count process (signaled by variable MVCNT = 0) continues via LIFO stack calls to the begin move (BGMOVE), end move (ENMOVE), begin count (BGCNT) and end count (ENDCNT) subroutines until a count has been completed at the end angle as entered by the user in subroutine QUAL. A call is then issued to subroutine WRFILE which writes any remaining data points to the output file, and then to subroutine FINIS which clears the interface and resets run parameters, freeing the unit for a new run. If during the course of data collection the maximum number of counts (variable LIMCNT) were recorded before the requested count time elapsed, then the number recorded in the output data record will be ten times the number of seconds spent counting to reach LIMCNT number of counts. A negative sign is attached to this number as a flag to processing

programs. The alternate data collection method which simultaneously steps and counts (variable MVCNT = 1) proceeds via LIFO stack calls to BGMOVE and ENMOVE. This mode reduces the computer overhead. It integrates over the two theta range given by the input step size. The count time is equally divided between each 0.005° increment of the total step width.

4. QUANTITATIVE ANALYSIS: Subroutine QUANT is called from MAIN via the LIFO address stack as set up by CONSOL at the users request. It initially asks the user for the approximate starting 2θ for the run and begins the slew to this angle. While the slew is in progress it asks the user if the run file, which contains the run parameters on each standard, unknown or reference line to be run, exists. If this is the first time this particular quantitative analysis has been run then the run file does not exist and subroutine STDFIL is called to create it.

The file of run parameters created by subroutine STDFIL is compatible in format to the output data file produced by either QUAL or QUANT. The format of this file is given in Table 6. The information concerning the standard intensities (or slope of the calibration curve) is not entered by the user but is computed at a later time and written by the off line, host computer, processing program.

The intent here is that a quantitative analysis data file will contain all of the information necessary to completely process that file. The first step is to run the samples necessary to produce the calibration curve. Data collection for each mounting of a standard can be automatically repeated several times (Default = 3). The standard may also be remounted several times and data collection repeated. These numbers are input by the user for a particular run of the standard. Each phase to be analyzed requires a separate calibration run of QUANT and, as with each standard determination for a phase, produces a separate output data file. If we have four phases to be determined, we make four standards measurement runs with QUANT. Up to 20 different lines can be measured in each standard and unknown. The four output data files will be read by the off line computer processing program which in turn will output a new Run file which will be used to direct the quantitative analysis of samples containing the four phases in unknown amounts.

QUANT allows for three types of quantitative analysis: the internal standard method with or without the use of reference intensity ratios, the intensity ratio method, and the spiking method. For a detailed description of how to use these methods, see the write-up

(in preparation as an NBS internal report) for the offline Quantitative Analysis program which actually carries out the quantitative analysis. The program is very general, allowing for chemical composition constraints and the presence of overlapped lines for phases which are to be determined.

Subroutine STDFIL writes the complete run file of information for a single standard and returns to QUANT. The file is rewound and then read by QUANT to calculate the required size of the quantitative analysis output file as described in Table 6 and to make a maximum run time estimate. The data file is allocated by a call to subroutine ALOT. Next the user selects which lines in the Run File, unknown and reference, that are actually to be measured on the sample currently under study. ENSLEW is called to ensure that the slew to the approximate starting angle has completed. A loop is now initiated which performs the following sequence:

A. Request the user to enter the number of repetitions, IFLAG(IUNIT, 5) and number of mountings, IFLAG(IUNIT, 6) to be run for this sample.

B. Rewind the run file, skip the first five records and, if required, request that a new sample be placed on the diffractometer.

C. Request the user to enter the chemical analysis for each new sample if it is an unknown run and if chemical constraints have been requested. If entered it is written to the output file.

D. Read line parameter records (records 7 and 8 of Table 6) and check if the last line has been read. If so skip to step I below.

E. Check with the previously input list to see if this line is to be analyzed. If not skip to step D above.

F. Write the line parameter records to the output data file.

G. Call ESTIM in order to optimize the time to spend counting the peak and background. See the description of ESTIM below for details.

H. Call either STPSCN for an integrated intensity measurement (variable IPK=0) or PKSCN for a peak height determination (IPK=1). These routines are called via the LIFO stack. When they have completed data collection for a line and WRFILE has written the last data points to the output file, QUANT is reentered at point D above.

I. Test if the last repetition and last mounting of the sample has been run. If not, branch to step B. When all data is collected FINIS is called in reset mode to free the unit

for a new run.

4a. COUNTING MODES: There are four modes of data collection allowed by subroutine QUANT and all four optimize the peak and background count time by a call to subroutine ESTIM. This subroutine uses three input parameters in its optimization: TIMLIM - the total number of minutes the user wishes to spend counting a line, a count mode flag (IPK) which sets the program into peak count or integrate scan mode and lastly a count type flag (FIXERR) which requests a fixed time counting mode or a fixed error counting mode. The optimization of all four counting modes requires a knowledge of the peak and background counts in a line. For this reason subroutine ESTIM first issues a one second count at the low angle background then, for IPK = 0, integrates while driving at high speed to the high angle background 2θ. If IPK = 1, ESTIM drives to the peak maximum position (as input by the user on the run file) and then counts for one second. It then slew backs to the low angle background and optimizes the counting procedure in the following manner:

A. Fixed Time Count Mode: The number of background points to measure on each side of the peak in either peak count or integrate mode is input as NBG and therefore 2*NBG background points will be measured (if two backgrounds are requested). In integrate scan mode the number of points to be measured on the peak is NPK = (EANG-BANG)/DANG-2*NBG. In peak count mode NPK points, as input by the user, will be measured at the top of the peak. The total count time for the line (TIMLIM) is divided into TIME seconds to be spent at each peak point and TIMBKG seconds to be spent at each background point according to:

$$TIME = \frac{LIMTIM}{NPK} * \frac{1}{(1+1/R)}$$

and

$$BKGTIM = \frac{NPK*TIME}{2*NBG*R}$$

where: I = the average count rate at each point in the peak scan,

B = the average count rate at each point in background, and

$$R = \left\{ I/_B \right\}^{\frac{1}{2}}$$

This division of time will result in the lowest estimated standard deviation for the net

counts (i.e., it optimizes the choice of TIME and BKGTIM). In the integrated count (scan) mode, I is estimated from the rapid integration of the peak made by ESTIM. In the peak count mode, I is simply the observed peak count rate in cps.

B. Fixed Error Count Mode: This type of count mode has traditionally been accomplished by counting to a fixed number of counts. For example, to achieve a 1% counting error, 10000 counts above background must be collected. The difficulty with this type of counting is that extremely long count times will be encountered on weak peaks. To improve this method we require two input parameters. The first is FIXERR, the desired percent error in the measurement, and the second is TIMLIM which, in this mode of counting, is the upper limit on the total count time. The time to spend on peak in fixed error mode is computed as follows:

$$\text{TIME} = \frac{I+R*B}{\text{FIXERR}^2*\text{NPK}*[I-B]^2}$$

The background time is determined by the same equation used in fixed time mode:

$$\text{BKGTIM} = \frac{\text{NPK}*\text{TIME}}{2*\text{NBG}*R}$$

This division of time between peak and background regions minimizes the total count time. If the estimates of I and B made by ESTIM are such as to require a total time greater than TIMLIM, then the program switches to fixed time counting for this peak.

4b. STEP SCAN: Subroutine STPSCN is also called by QUANT. The only difference in the activity initiated by QUANT from that directed by QUAL (see page 6) is that the first and last NBG points are counted for BKGTIM seconds. All points in between are counted for TIME seconds. The offline summing of these peak and background counts will produce an integrated intensity. It should be noted that the input parameter NPT, (number of peak data points), is used in this mode only by the processing program. The NPT points about the observed maximum in the step scan data will be averaged to give an estimate of the line peak height. When STPSCN completes measurement of a peak as directed by QUANT it first calls WRFILE and then returns to QUANT.

4c. PEAK SCAN: Subroutine PKSCAN carries out a peak counting procedure. NBG points are measured for BKGTIM seconds each at the low angle background. The step width between points is DANG which defaults to the minimum step width of 0.005°. NPK points are measured at the peak by collecting (NPK)/2 points on each side of the peak. Again the points are separated by DANG degrees. Since we would like the division of points to be symmetric about the peak, NPK must be an odd number. If it is even, 1 will be subtracted from it to make it odd. odd. The 2θ of the peak maximum is input by the user and stored in variable PANG. If the high angle background (variable EANG) is input as zero, only one background will be measured.

5. SLEWING: Subroutine SLEW is called by MAIN via the LIFO stack. The subroutine initiates a maximum rate scan (30 degrees per minute) by calling the begin move subroutine (BGMOVE). Slew always checks the legality of the target angle by calling CKANG before attempting to move. Subroutine CKANG forces the target angle to be a multiple of .005° (a single motor step) and determines if a movement to this angle will cause a collision; it also computes the number of motor steps required to get to the target angle. The slew motion always moves 10 steps too many toward low angle.

On completion of the slew subroutine ENSLEW is called, usually via the LIFO stack. This subroutine in turn calls ENMOVE to ensure that the requested number of steps were actually executed and that the tracking counter agrees. If all went well, ENSLEW then calls BGMOVE to move 10 steps toward high angle and again calls ENMOVE to check that the request was successfully executed. This logic ensures that the diffractometer gear backlash is always tight in the high angle direction before any run is made. Any failure detected by ENSLEW or ENMOVE (or for that matter ENDCNT) results in an abort sequence which ends the run by calling FINIS.

6. RUN TERMINATION: Subroutine FINIS handles the completion of any run using the interface. It handles both normal and abnormal exits. For exits arising from an interface error the unit activity is set to -1 which forces a user to reinitialize the interface (CONSOL option 4) before any further use can be made of it. All exits will write any remaining data in the 16 point data array out to the data file and then put an endfile mark on this file.

After zeroing run parameters for the Unit, FINIS directs a slew operation to return the

2θ arm to 80° where it is unlikely to be bumped by users. The last action of FINIS is to print the time the job ended, the number of records in the file, and a message stating that the unit is available for another run.

7. FILE TRANSFER: This option is selected by the user in subroutine CONSOL and directs a LIFO stack call to subroutine TRANS. Subroutine TRANS allows file transfer from hard disk to a floppy, disk to the printer or to another computer via the NBS NET.

TABLE 1.

INTEGER ADDRESS CODES USED ON THE LIFO STACK

ADDRESS	OPERATION
1	CALL CONSOL
2	CALL QUANT
3	CALL QUAL
4	CALL STPSCN
5	CALL PKSCN
6	CALL SLEW
7	CALL ENSLEW
8	CALL BGMOVE
9	CALL BGCNT
10	CALL FINIS
11	CALL TRANS
12	CALL INTRUP
13	CALL ESTIM

TABLE 2.

FLOW CHART OF PROGRAM AUTO

MAIN
- INIT (SETANG-CKANG)
- CONSOL (SETANG-CKANG, POP, PUSH, FILNAM, DATE)
- QUAL (FILEUP, ALOT, ENSLEW-ENMOVE-FINIS, BGMOVE, ENTIM, POP, PUSH)
- QUANT (SLEW-CKANG-BGMOVE, FILEUP, STDFIL, ALOT, ENSLEW-ENMOVE, ENTIM, FINIS, BGMOVE, POP, PUSH)
- STPSCN (BGCNT, ENDCNT, ENMOVE-FINIS, CKANG, POP, PUSH)
- PKSCN (ENDCNT, ENMOVE-FINIS, CKANG, POP, PUSH)
- SLEW (CKANG, BGMOVE, POP, PUSH)
- ENSLEW (ENMOVE-FINIS, BGMOVE, POP, PUSH)
- BGMOVE (POP)
- ENMOVE (POP, FINIS-POP, PUSH)
- BGCNT (POP)
- ENDCNT (FINIS-POP, PUSH)
- FINIS (POP, PUSH)
- TRANS
- INTRUP (POP, PUSH)
- ESTIM (OPTIM)

Note: A dash indicates that the subroutine is called by the routine to the left. A comma indicates that the subroutine is called by the routine outside the parentheses. System and NBS Bus calls are omitted from this flow chart.

TABLE 3.

SYSTEM AND NBS BUS ROUTINES

SYSTEM ROUTINES

IFLAG (0) A function that returns the value on top of the Interdata 7/16's flag table. It is used to detect a consol escape key input by the user.

SYSTEM (STRING, NUMBER) A subroutine that sends a FLOXOS (operating system) command to the computer. STRING is an array or a quoted literal that forms the command and NUMBER is the number of characters and blanks in the STRING.

NBS BUS ROUTINES

ISTAT (LU, NBSADD) An integer function that returns the status of the interface card with the NBS Bus subaddress NBSADD.

INBUS (LU) An integer function that returns the contents of a subaddress that has been previously selected with the routines ISTAT or CLEAR.

OUTBUS (LU, IDATA) A subroutine that sends data (the integer IDATA) to a previously selected subaddress.

CLEAR (LU, NBSADD) A subroutine that selects the subaddress NBSADD and clears it.

COMAND (LU, STRING) A subroutine that sends commands to an NBS Bus card at a previously selected subaddress. STRING is an array of quoted literals formed of one or more of the following letters; A (address mode), I (increment mode), D (decrement mode), S (start), C (clear).

Where NBSADD is the subaddress of the selected interface and LU is the logical unit attached to the NBS Bus.

Note: All integer variables in the program are stored in two bytes except the variables used in referring to the NBS Bus (e.g. LU, NBSADD, IDATA, etc) which are stored as four byte integers.

TABLE 4.

INPUT/OUTPUT UNITS

Logical Unit Number	Variable Name	Physical Device	Use
1	NBS	NBS Bus	All I/O to interfaces
2	IDT	Date-Time Clock	To read date and time from system clock
3	ICLOCK	Real Time Clock	Not currently used
4	ISPEC	Disk File-DACONF	To read instrument specifications
5	IRD	Terminal input	For user option input
6	IPRT	Line Printer	For listing the run file and debugging
7	ITYP	Terminal output	For instruction listing
8	IDISK(1,1)	Run file for Unit #1	
9	IDISK(2,1)	Run file for Unit #2	
10(A)*	IDISK(1,2)	Output data file for Unit #1	
11(B)	IDISK(2,2)	Output data file for Unit #2	
12(C)	IFIN	Input file for TRANS	
13(D)	IFOUT	Output file for TRANS	

*Values in parentheses are the hexidecimal values of the logical unit numbers used by the operating system.

TABLE 5.

OUTPUT DATA FILE FORMAT FOR PATTERN COLLECTION

Record 1: Problem Specification
- Col 1-80 Title of run
- Col 81-100 Operators name
- Col 101-106 Date of run
- Col 107-112 Time of beginning of run

Record 2: Instrument Specification Parameters
- Col 1-4 The name of the diffractometer used for the run (e.g., AM-1)
- Col 5-10 The date the instrument specification (IS) file was created (e.g., 073080)
- Col 11-12 Initials of person who created the IS file (e.g., CH)
- Col 13-14 Element symbol of X-ray Tube Target (e.g., CU)
- Col 15-16 Characteristic X-ray line used (e.g., KA = Kα)
- Col 17-22 X-ray wavelength (e.g., 1.5406)
- Col 23-24 Focal spot type (e.g., FF = fine focus, SF = standard focus)
- Col 25-28 Take off angle in degrees (e.g., 5.50)
- Col 29-30 Divergent slit type (e.g., TC = Theta Compensating, FX = Fixed)
- Col 31-36 Divergent slit angle if Fixed slit or irradiated length in mm if Theta compensating (e.g., 12.500)
- Col 37-42 Goniometer circle radius in mm (e.g., 172.00)
- Col 43-46 Incident Soller Slit divergence angle in degrees (e.g., 4.00)
- Col 47-50 Receiving Soller slit divergence in degrees (e.g., 0.00)
- Col 51-54 Receiving slit divergence in degrees (e.g., 0.10)
- Col 55-56 Monochromator code (e.g., GR = graphite, LF = lathium floride, blank = none)
- Col 57-58 Detector Code (e.g., SC = scintillation)
- Col 59-62 Dead time in microseconds (e.g., 2.00)
- Col 63-66 Instrument Intensity stability factor in % (e.g., 0.10)
- Col 67-128 Not Used

Record 3: Instrument Calibration information
- Col 1 Flat specimen correction (0 - not applied, 1 - applied)
- Col 2 Axial divergence correction (0 - not applied, 1 - applied)
- Col 3 Lorentz correction (0 - not applied, 1 - applied)
- Col 7-16 2θ offset to be subtracted from the 2θ values before applying the polynomial correction (F10.3)
- Col 17-32 a_0 Y intercept of calibration polynomial (E16.8)
- Col 33-48 a_1 first order polynomial coefficient
- Col 49-64 a_2 second order polynomial coefficient
- Col 65-80 a_3 third order polynomial coefficient
- Col 81-96 a_4 fourth order polynomial coefficient

Record 4 and 5: Inverse Matrix from Instrument Calibration Run (8E16.6). The order is:
- Record 4: (1,1), (1,2), (1,3), (1,4), (1,5), (2,2), (2,3), (2,4)
- Record 5: (2,5), (3,3), (3,4), (3,5), (4,4), (4,5), (5,5)

Record 6: Run Parameters
- Col 1-7 Beginning angle of scan (BANG-F7.3)
- Col 8-14 (Reserved)
- Col 15-21 Ending angle of scan (EANG-F7.3)
- Col 22-28 Step width in degrees (DANG-F7.3)
- Col 29-32 Rate of movement between points in degrees per minute (RATE-F4.1)
- Col 33 Method: 0 = independent/1 = simultaneous move and count
- Col 46-50 Maximum count allowed for scaler-defaults to 32700 (LIMCNT-I5)
- Col 51-52 Code for internal standard number 1 (if one is present)
- Col 53-54 Code for internal standard number 2 if needed
- Col 55-56 Code for internal standard number 3 if needed
- Col 105-111 Count time for each point in seconds (TIME-F7.2)

Table 5. (cont.)

```
Record 7:  Intensity Data in (F8.3, 2X, 3I2, 16I7)
     Col   1-8   The angle of the last point in the record (CANG-F8.3)
     Col   9-16  The time the record was written (ITIM-3I2)
     Col  17-23  Counts at first point (CNT(1)-I7)
     Col  24-30  Counts at second point (CNT(2)-I7)
     .    .      .
     .    .      .
     .    .      .
     Col 122-128 Counts at sixteenth point (CNT(16)-I7)
```

As many of these records as needed are included to record the number of counts observed at each point. A negative data point indicates that LIMCNT counts were detected and the data point is the negative of ten times the number of seconds spent counting.

Table 6.

RUN FILE AND OUTPUT DATA FILE FORMAT FOR QUANTITATIVE ANALYSIS

Record 1: Problem specification - compatible with record 1 in Table 5.

Col	1-80	Title of Run (ITITLE (1-40) - 40A2)
Col	81-100	Name of operator (ITITLE (41-50)-10A2)
Col	101-106	Date of File Creation YYMMDD (IDAT-3I2)
Col	107-112	Time of File Creation HHMMSS (ITIM-3I2)
Col	113	IFLAG (1) = 0 This is an output data file from pattern collection
		= 1 This is an output data file from quantitative analysis
		= 2 This is a run file directing the determination of a standard for quantitative analysis or calibration
		= 3 This is a run file output from the data reduction program on the off line computer, directing the quantitative analysis of an unknown
		= 4 This is an output data file for a calibration run
Col	114	IFLAG(2) = 0 Qualitative analysis run
		= 1 Run to analyze an unknown
		= 2 Run to measure standards
		= 3 Run to measure relative intensities
Col	115	IFLAG(3) = 0 Qualitative analysis
		= 1 Internal Standard Method of Analysis
		= 2 Intensity Ratio Method of Analysis
		= 3 Spiking method of analysis
		= 4 Relative Intensity Measurement
		= 5 Profile measurement
Col	116	IFLAG(4) = 0 Elemental percent composition is not on the file
		= 1 Elemental percent composition is included in Record 6
Col	117	IFLAG(5) = 0 Qualitative analysis
		= N the number of repetitive runs for this sample for quantitative analysis (all to be output in the same data file) only needed when IFLAG(1)=2. (Default = 3).
Col	118	IFLAG(6) = N the number of mountings of each sample to be run

*Record 2: Instrument specification - identical with record 2 of Table 5.

*Record 3-5: Calibration parameters - identical with records 3-5 of Table 5.

*Record 6: Elemental Composition. This record will be blank if IFLAG(4)=0. (10(A2, 2F5.2))

Col	1- 2	Element symbol (A2)
Col	3- 7	Percent of element in unknown (F5.2)
Col	8-12	Standard deviation of percentage (F5.2)
Col	13-14	Element symbol number 2
Col	15-19	Percent of element 2
Col	20-24	Standard deviation of percentage
.	.	.
.	.	.
.	.	.
Col	109-110	Element Symbol number 10
Col	111-115	Percent of element 10
Col	116-120	Standard deviation of percentage

RECORD 7: Line parameters - This record is compatible with record 6 of Table 5 but contains additional information needed for quantitative analysis.

Col	1- 7	2θ of Low angle background of peak to be recorded (BANG-F7.3)
Col	8-14	2θ of peak maximum - used in peak count mode (PANG-F7.3)
Col	15-21	2θ of high angle background of peak to be recorded (EANG-F7.3). In peak count mode this may be zero if only one background is desired.
Col	22-28	Step width to be used in moving over the peak. This defaults to .005°. (DANG-F7.3)
Col	29-32	Rate of movement between points in degrees per minute (RATE-F4.1)

Table 6. (cont.)

Col 33 Method: 0 = independent/1 = simultaneous step and count
Col 34-40 The maximum time in minutes to be spent counting a peak. In fixed time mode this value will be optimally divided between background time and peak time. In fixed error count mode this will be treated as an upper limit on the count time. (TIMLIM-F7.3)
Col 41-45 The percent error to which a peak should be counted. If zero, fixed time count mode will be used. (FIXERR-F5.2)
Col 46-50 The maximum number of counts allowed in the scaler-usually 32700 (LIMCNT-I5)
Col 51-56 Name of the phase (3A2)
Col 57-62 hkℓ of the line (3I2)
Col 63 KODE (I1) = 1 this line belongs to the unknown phase being analyzed
= 2 this line belongs to the internal standard
= 3 this line is an intensity reference belonging to a special narrow slotted sample holder
= 4 this area is from an amorphous halo (NBG=0)
Col 64 IOVLP(I1) = 0 this is a fully resolved line
= N there are N more lines overlapped with this line whose phase name, hkℓ and I_{rel} values will supplied on Record 8.
Col 65 IPK (I1) = 0 Integrated step scan of the peak
= N peak counting mode
Col 66-67 The number of points to be measured at the top of a peak in peak count mode or the number of points to be used in averaging at the peak to estimate peak intensity in peak scan mode (NPK-I2).
Col 68-69 The number of points to count for BKGTIM seconds at the beginning and end of a peak. Equals zero if KODE = 4. (NBG-I2).
Col 70-75 Relative intensity of the line (FREL-F6.1)
Col 76-79 Standard deviation of the relative intensity (SIGREL-F4.1)
Col 80-88 The standard intensity to be used in analyzing an unknown line. This will be I_{pure} or I_{pure}/I_{ref} or I_{unk}/I_{std} depending on the analysis method. This value will be computed by the analysis program when run to measure the standards (STD-F9.4).
Col 89-96 ESD of standard intensity (SIGSTD-F8.4)
Col 97-104 The concentration of the added phase if IFLAG(3)=3. The weight fraction of the internal standard if IFLAG(3)=1. The mass absorption coefficient of this sample if IFLAG(3)=2. (CONC-F8.3)
Col 105-111 Count time at each peak point in seconds (TIME - F7.2)
Col 112-118 Count time at each background point in seconds (BKGTIM - F7.2)

RECORD 8: Line overlap information (present if IOVLP≠0)
Col 1- 6 Name of second phase overlapping this line (3A2)
Col 7-12 hkℓ of overlapping line (3I2)
Col 13-18 I_{rel} of overlapping line (F6.1)
Col 19-22 Standard deviation of I_{rel} of second overlapping line (F4.1)
Col 23-28 Name of third overlapping phase
Col 29-34 hkℓ of third overlapping line
Col 35-40 I_{rel} of overlapping line
Col 41-44 Standard deviation of I_{rel} of third overlapping line

The following chemical information on the first phase described by Record 7 is only required if IFLAG(4)=1 and KODE = 1.
Col 49-50 Symbol of element 1 of phase (A2)
Col 51-58 Percentage of element 1 in phase (F.8.4)
Col 59-60 Symbol of element 2 of phase
Col 61-68 Percentage of element 2 in phase
. . .
. . .
. . .
. . .

Table 6. (cont.)

Col 119-120 Symbol of element 8 of phase
Col 121-128 Percentage of element 8 in phase

Note: Record 8 will always be present but usually blank.

RECORD 9: Intensity data (F8.3, 2X, 3I2, 16I7)
Col 1- 8 The angle of the last data point in the record (CANG - F8.3)
Col 9-16 The time the record was written (ITIM - 3I2)
Col 17-23 Counts at first point
Col 24-30 Counts at second point
. . .
. . .
. . .
Col 122-128 Counts at sixteenth point

As many of these records as needed are included to record the number of counts observed at each point. A negative data point indicates that LIMCNT counts were detected and the data point is the negative of ten times the number of seconds spent counting.

*Record 2, 3, 4, and 6 are not contained in the Run File.

EXAMPLE 1†

This example shows the initiation of program AUTO. The current 2θ for each unit is entered by the user. AUTO then lists the option menu. Here, the user chooses to slew to 25.0° 2θ on unit #2.

```
*RUN AUTO
 ENTER THE CURRENT 2-THETA OF UNIT #1 (F10.3)
20.005
 ENTER THE CURRENT 2-THETA OF UNIT #2 (F10.3)
24.13

          ** AUTO CONTROL **

 ENTER ONE OF THE FOLLOWING OPTIONS (Z1):
     0 - STATUS
     1 - POWDER PATTERN COLLECTION
     2 - QUANTITATIVE ANALYSIS
     4 - SET 2-THETA
     5 - SLEW TO 2-THETA
     6 - TERMINATE RUN ON A UNIT
     7 - PANIC STOP ON BOTH UNITS
     C - LIST DISK CONTENTS
     D - DEBUG
     E - DELETE A FILE
     F - TRANSFER FILE
5
 ENTER UNIT NUMBER (I1)
2
 CURRENT 2-THETA= 24.130      ENTER TARGET 2-THETA (F7.3)
16.0
```

†User entries are underlined. Each entry is terminated by hitting RETURN.

EXAMPLE 2

When the ESC key is pressed the program AUTO responds by listing the option menu. A powder pattern collection was initiated on Unit #2 beginning at 3.0°. File name, run title and operator name were then input. Next the ending angle, step size, count time, and MVCNT method code were entered. Since the max count was not entered the default value was used. No internal standards were present so the user hit the return key after the inquiry. After the program echoes the input and estimates the run time the user can indicate if corrections and/or printing of these parameters is desired. At completion of the data collection, a termination message is printed which includes the file name and number or records written to the disk.

```
(ESC)
          ** AUTO CONTROL **

 ENTER ONE OF THE FOLLOWING OPTIONS (Z1):
     0 - STATUS
     1 - POWDER PATTERN COLLECTION
     2 - QUANTITATIVE ANALYSIS
     4 - SET 2-THETA
     5 - SLEW TO 2-THETA
     6 - TERMINATE RUN ON A UNIT
     7 - PANIC STOP ON BOTH UNITS
     C - LIST DISK CONTENTS
     D - DEBUG
     E - DELETE A FILE
     F - TRANSFER FILE
1
 ENTER UNIT NUMBER (I1)
2

 POWDER PATTERN DATA COLLECTION ON UNIT #2 ON 80/11/13 AT 11:31:47

 ENTER STARTING ANGLE FOR THE RUN-F7.3-(DEFAULT=5.0)
3.0

 ENTER FILE NAME- 6 CHARACTERS ENDING WITH D
CH053D
 ENTER RUN TITLE
SAMPLE POWDER PATTERN DATA COLLECTION FOR SILICON
 ENTER YOUR NAME:
CAMDEN HUBBARD

 ENTER:END ANGLE,STEPSIZE,COUNT TIME(SEC),METHOD,MAXCOUNTS UNDER DEFAULT VALUES
 90.000   0.020     1.0   0 32700
 60.0     0.025     0.2   1

 ENTER 2 LETTER CODES FOR UP TO 3 INTERNAL STANDARDS

 ___

 POWDER PATTERN DATA COLLECTION ON UNIT #2 ON 80/11/13 AT 11:31:47
 SAMPLE POWDER PATTERN DATA COLLECTION FOR SILICON
 CAMDEN HUBBARD
 OUTPUT FILE: CH053D
 STANDARDS:
 BEGIN ANGLE =     3.000
 END ANGLE(DEG)= 60.000
 STEP SIZE(DEG)=  0.025
 CNT TIME(SEC) =  0.200
 MVCNT METHOD  =      1
 MAX COUNTS    =  32700

 RUN TIME(HRS) =  0.266

 CORRECTIONS? Y/N
N
 PRINT INPUT? Y/N
N

 END OF RUN: UNIT # 2,FILE: CH053D,REC=   149 ON 80/11/13 AT 11:31:47
```

EXAMPLE 3

During the course of data collection the user can interrupt by pressing the ESC key. In this example, the status of both units was requested.

```
          ** AUTO CONTROL **

ENTER ONE OF THE FOLLOWING OPTIONS (Z1):
     0 - STATUS
     1 - POWDER PATTERN COLLECTION
     2 - QUANTITATIVE ANALYSIS
     4 - SET 2-THETA
     5 - SLEW TO 2-THETA
     6 - TERMINATE RUN ON A UNIT
     7 - PANIC STOP ON BOTH UNITS
     C - LIST DISK CONTENTS
     D - DEBUG
     E - DELETE A FILE
     F - TRANSFER FILE
0

          ** STATUS ** ON 80/11/21 AT  9:21:26

               UNIT:       1         2

FILE:                                 CH053D
RUN:                        INACTV 0  POWDER 1
TO END AT:          0/ 0    0: 0     11/21    9:22
2-THETA:                    79.995   16.140
```

EXAMPLE 4

The QUANT option was requested for Unit #2 beginning data collection near 25.0°. Because the user indicated that a run file does not exist the subroutine STDFIL requested the needed information. In this example a single reflection profile measurement is requested. The net integrated intensity is to be measured with a relative error of approximately 0.5% unless the total data collection time would exceed 10 mintues. The angle range choosen was 27.2 to 29.4° in steps of 0.005°. Twenty points on each side were selected as background points.

```
(ESC)          ** AUTO CONTROL **

 ENTER ONE OF THE FOLLOWING OPTIONS (Z1):
     0 - STATUS
     1 - POWDER PATTERN COLLECTION
     2 - QUANTITATIVE ANALYSIS
     4 - SET 2-THETA
     5 - SLEW TO 2-THETA
     6 - TERMINATE RUN ON A UNIT
     7 - PANIC STOP ON BOTH UNITS
     C - LIST DISK CONTENTS
     D - DEBUG
     E - DELETE A FILE
     F - TRANSFER FILE
2
 ENTER UNIT NUMBER (I1)
2

 QUANT ANALYSIS DATA COLLECTION ON UNIT #2

 ENTER APPROXIMATE STARTING 2-THETA (F7.3)
25.0

 RUN FILE ALREADY EXIST? Y/N
N

 ENTER FILE NAME- 6 CHARACTERS ENDING WITH R
CH054R
 ENTER RUN TITLE
COLLECTION OF A SELECTED PEAK PROFILE BY QUANT OPTION
 ENTER YOUR NAME:
CAMDEN HUBBARD

 METHOD OF ANALYSIS ? (I1)
 1 - INTERNAL STANDARD
 2 - INTENSITY RATIO
 3 - SPIKING
 4 - REL. INT.(DEFAULT)
 5 - PROFILE
5

 FOR LINE #   1 ENTER UNDER DEFAULT VALUES
  1 -6 NAME OF LINE(BLANK TERMINATES INPUT)
  7- 8, 9-10, 11-12 H K AND L
 13-14 =0 INTEGRATE, =1 PEAK COUNT
 15-16 =1 UNKNOWN, =2 STD, =3 REF, =4 AMORPH LINE
 17-22 =X.X THE DESIRED % ERROR (0=FIXED TIME)
 23-30 TIME(MIN) TO COUNT PK + BKG
 31-38 REL. INT.
 39-46 SIGMA(REL. INT.)

 NNNNNN H K L 0 1 1.000  10.000  ***.**  *.**
 SILICN 1 1 1     0.5
 SILICN 1 1 1 0 1 0.500  10.000    0.00    0.00
 OK? TYPE Y/N
 Y
```

```
 ENTER THE FOLLOWING UNDER THE DEFAULT VALUES:
 COL 1 -10 BACKGROUND LOW 2-THETA
 COL 11-20 PEAK 2-THETA
 COL 21-30 BACKGROUND HIGH 2-THETA
 COL 31-40 STEP WIDTH
 COL 41-46 MAX. COUNTS
 COL 47-50 # OF PEAK POINTS
 COL 51-55 # OF BACKGROUND POINTS
 COL    60 MVCNT METHOD

   LLL.LLL    PPP.PPP    HHH.HHH      0.005 32700    5    10     0
    27.2       28.443     29.4                       20     1
    27.200     28.443     29.400       0.005 32700   5    20     1
 OK? TYPE Y/N
Y

 FOR LINE #   2 ENTER UNDER DEFAULT VALUES
  1 -6 NAME OF LINE(BLANK TERMINATES INPUT)
  7- 8, 9-10, 11-12 H K AND L
 13-14 =0 INTEGRATE, =1 PEAK COUNT
 15-16 =1 UNKNOWN, =2 STD, =3 REF, =4 AMORPH LINE
 17-22 =X.X THE DESIRED % ERROR (0=FIXED TIME)
 23-30 TIME(MIN) TO COUNT PK + BKG
 31-38 REL. INT.
 39-46 SIGMA(REL. INT.)

 NNNNNN H K L 0 1 1.000  10.000  ***.**  *.**

---

 ENTER: # OF REPETIONS, # OF MOUNTINGS (2I1)
11

 ENTER FILE NAME- 6 CHARACTERS ENDING WITH D
CH054D
 ENTER RUN TITLE
PEAK PROFILE DATA COLLECTION FOR THE 1 1 1 REFLECTION OF SILICON
 ENTER YOUR NAME:
CAMDEN HUBBARD

 PLACE SAMPLE # 1 ON UNIT 2 WHEN READY HIT RETURN. ENTER 1 TO STOP

---

 END OF RUN: UNIT # 2,FILE: CH054D,REC=    36 ON 80/11/13 AT 11:31:47
```

```
      C_____________________________________________________________________
      C  PROGRAM AUTO - THIS IS THE MAIN PROGRAM FOR THE XRAY POWDER
      C_____________________________________________________________________
      C  DIFFRACTION DATA COLLECTION SYSTEM.
      C**********************************************************************
      C      UPDATE     11/10/80/
      C**********************************************************************
      C
      C  PROGRAMMED BY R. L. SNYDER, C.R. HUBBARD AND N.C. PANAGIOTOPOULOS  JUNE 1980
      C
      C
      C  COMMON DECLARATIONS FOR ALL SUBROUTINES
      C
            IMPLICIT INTEGER*2 (I-N)
            INTEGER*4 NBS, INTR
      C
            COMMON /ANG/       BANG(2),        CANG(2),        DANG(2),
           1 EANG(2),          IRATE(2),       PANG(2),        RATE(2),
           2 TANG(2),          IZRST(2),       NPTPK(2)
      C
            COMMON /BUFF/      IALPH(66),      NAME(15)
      C
            COMMON /CNTL/      IACT(2),        IDBUG(2),       INTR,
           1 IRUN(2),          ITRANS(2,20),   MODE(2,20),     MVCNT(2)
      C
            COMMON /COUN/      ICNT(2,16),     IPT(2),         ITOT(2),
           1 LIMCNT(2),        ISAVE(2)
      C
            COMMON /IO/        NBS,            IDT,            ICLOCK,
           1 ISPEC,            IRD,            IPRT,           ITYP,
           2 IDISK(2,2),       IFIN,           IFOUT
      C
            COMMON /PROB/      IDAT(3),        IFLAG(2,10),    ITIM(3),
           1 ITITLE(50)
      C
            COMMON /QUAN/      CONC,           FREL,           IOVLP,
           1 KODE,             LOOP(2),        LPK(2),         NUM(2,20),
           2 SIGSTD,           STD,            STDREL,         IREPT(2)
      C
            COMMON /RNDT/      NREC(2),        NMFIL(2,6),     ITR(2),
           1       IENRN(2,4)
      C
            COMMON /TIM/       BKGTIM(2),      FIXERR(2),      IPK(2),
           1 NBG(2),           NPK(2),         TIME(2),        TIMLIM(2)
      C
      C
      C  -----COMMON BLOCK GLOSSORY-----
      C
      C  I/O DEFINITIONS - COMMON /IO/
      C     NBS       = 1 - THE LOGICAL UNIT NUMBER OF THE NBS BUS
      C     IDT       = 2 - THE UNIT FOR READING THE DATE AND TIME
      C     ICLOCK    = 3 - THE UNIT FOR THE REAL TIME CLOCK
      C     ISPEC     = 4 - THE UNIT FOR READING THE CONF(IGURATION) FILE
      C     IRD       = 5 - TERMINAL INPUT
      C     IPRT      = 6 - LINE PRINTER OUTPUT
      C     ITYP      = 7 - TERMINAL OUTPUT
      C     IDISK(1,1) = 8 - STANDARD INPUT FILE FOR DIFFRACTOMETER #1
      C     IDISK(2,1) = 9 - STANDARD INPUT FILE FOR DIFFRACTOMETER #2
      C     IDISK(1,2) = 10- OUTPUT DATA FILE FOR DIFFRACTOMETER #1
      C     IDISK(2,2) = 11- OUTPUT DATA FILE FOR DIFFRACTOMETER #2
      C     IFIN       = 12 INPUT FOR TRANS ROUTINE
      C     IFOUT      = 13 OUTPUT FOR TRANS ROUTINE
      C
      C GENERAL PARAMETER DEFINITIONS-COMMONS /ANG/,/BUFF/,/CNTL/,/COUN/,/PROB/
      C     BANG          = THE BEGINNING 2-THETA FOR RUN ON UNIT #1 OR #2
      C     BKGTIM        = BACKGROUND COUNT TIME IN SECONDS
      C     CANG          = THE CURRENT 2-THETA OF THE DIFFRACTOMETER
      C     DANG          = THE DELTA 2-THETA OR STEP WIDTH
      C     EANG          = THE ENDING 2-THETA FOR THE RUN
      C     FIXERR        = 0 - FIXED TIME COUNT MODE (WITH OPTIMIZATION)
      C                     = XX - FIXED ERROR COUNT MODE. COUNT TO XX PERCENT ERROR
      C     IACT          = A CURRENT ACTIVITY CODE FOR A UNIT
      C                       =-1 LIMIT SWITCH HIT OR INTERFACE ERROR 2THETA LOST
      C                       = 0 UNIT NOT ACTIVE
```

```
C                       = 1 UNIT IS CCUNTING
C                       = 2 UNIT IS STEPPING
C                       = 3 UNIT IS SLEWING
C     IALPH         = A SCRATCH ARRAY USED IN VARIOUS PLACES
C     ICNT          = THE ARRAY FOR RECENTLY COLLECTED INTENSITY VALUES
C     IDAT          = THE DATE OF THE START OF THE RUN
C     IDBUG         = 0 - DO NOT PRINT DEBUG INFORMATION
C                     = 1 - PRINT DEBUG MESSAGES AND LOGIC TRACE
C     IFLAG         = RUN CONTROL FLAGS READ FROM QUANT RUN FILE
C     IFLAG(1,1)    = FILE TYPE DISCRIPTOR
C                       = 0 DATA FILE FROM PATTERN COLLECTION
C                       = 1 OUTPUT DATA FILE FROM QUANT
C                       = 2 QUANT RUN FILE FOR STANDARD DETERMINATION
C                       = 3 QUANT RUN FILE FOR UNKNOWN DETERMINATION
C                       = 4 OUTPUT FILE FROM CALIBRATION RUN
C     IFLAG(1,2)    = RUN TYPE DESCRIPTOR
C                       = 0 PATTERN COLLECTION
C                       = 1 QUANT ANALYSIS FOR WT%
C                       = 2 DETERMINE CONSTANTS FROM STANDARDS
C                         (QUANT, I-REL, OR CALIBRATION)
C     IFLAG(1,3)    = QUANTITATIVE ANALYSIS TYPE
C                       = 0 QUAL
C                       = 1 INTERNAL STANDARD METHOD
C                       = 2 INTENSITY RATIO METHOD
C                       = 3 SPIKING METHOD
C                       = 3 RELATIVE INTENSITY RUN
C                       = 4 CALIBRATION RUN
C     IFLAG(1,4)    = ELEMENTAL ANALYSIS FLAG
C                       = 0 QUAL
C                       = 0 ELEMENTAL COMPOSITION WILL NOT BE ENTERED
C                       = 1 ELEMENTAL COMPOSITION WILL BE USED
C     IFLAG(1,5)    = NUMBER OF REPETIONS FOR EACH SAMPLE MOUNT
C     IFLAG(1,6)    = NUMBER OF MCUNTINGS FOR THIS COMPOUND OR MIXTURE
C     IFLAG(1,7-10) = AVAILABLE FOR FUTURE EXPANSION
C     INTR          = THE STATUS WORD FROM THE INTERFACE WHICH JUST COMPLETED
C     IPK               = DATA COLLECTION MODE FLAG
C                         = 0 STEP SCAN (IE TO BE LATER INTEGRATED)
C                         = 1 PEAK COUNT MODE
C     IPT           = THE NUMBER OF THE CURRENT GROUP OF 16 DATA POINTS
C                     WHICH IS BEING COLLECTED OR WAS JUST COLLECTED
C     IRATE         = INTERFACE RATE CODE = 30/RATE
C     IRUN          = A STATUS FLAG FOR EACH DIFFRACTOMETER
C                     = 0 NOT IN USE
C                     = 1 QUANTITATIVE ANALYSIS RUN
C                     = 2 QUALITATIVE ANALYSIS RUN
C                     = 3 INTENSITY MEASURING RUN
C                     = 4 CALIBRATION RUN
C                     = 5 ALIGNMENT RUN
C     ISAVE         = 1 ENMOVE SAVES COUNTS ACCUMULATED DURING 2-THETA
C                       DRIVE
C                     = 0 DO NOT SAVE COUNTS
C     ITOT          = THE TOTAL NUMBER OF DATA POINTS COLLECTED SO FAR
C                     QUAL ALL POINTS,QUANT POINTS IN THE PEAK
C                     CURRENTLY MEASURED
C     ITIM          = TIME  WHEN SUBRTN DATE WAS CALLED
C     ITITLE(50)    = THE RUN TITLES
C                       - CHAR 1-80 = TITLE
C                       - CHAR 81-100 = THE OPERATORS NAME
C     ITRANS        = A LIFO (LAST IN FIRST OUT) ARRAY OF TRANSFER FLAGS
C                       THESE FLAGS, WHICH ARE SET BY VARIOUS SUBROUTINES,
C                       CONTROL THE FLOW OF THE PROGRAM BY DETERMINING WHICH
C                       SUBROUTINES ARE TO BE CALLED NEXT
C                       = 1 CALL SUBROUTINE CONSOL
C                       = 2 CALL QUANT
C                       = 3 CALL QUAL
C                       = 4 CALL STSCAN (IE STEP SCAN)
C                       = 5 CALL PKSCAN (IE PEAK SCAN)
C                       = 6 CALL BGSLEW
C                       = 7 CALL ENSLEW
C                       = 8 CALL BGMOVE (IE BEGIN MOVE)
C                       = 9 CALL BGCNT  (IE BEGIN COUNTING)
C                       = 10 CALL FINIS - END THE RUN
C                       = 11 CALL TRANS - TO TRANSFER A FILE TO THE 1108
C                       = 12 CALL INTRUP - THE EVENT DRIVEN SUBROUTINE
```

```
C                       = 13 CALL ESTIM - TO OPTIMIZE PEAK COUNTING
C                       = 14-20 AVAILABLE FOR FUTURE EXPANSION
C     IZRST           = A FLAG TO INDICATE A ZERO STEP REQUEST TO SLEW
C                       = 0 (FALSE) - NON ZERO STEP REQUEST
C                       = 1 (TRUE)  - A ZERO STEP REQUEST. WE ARE AT
C                         THE DESIRED ANGLE. INTERFACE WILL NOT SHOW
C                         A COMPLETION.
C     LIMCNT          = THE MAXIMUM COUNTS ALLOWED AT A POINT (DEFAULTS TO 32700)
C     MODE              = AN INTERNAL TRANSFER FLAG FOR THOSE SUBROUTINES WHICH
C                             NEED TO INITIATE AN ACTIVITY AND AWAIT A COMPLETION
C     MODE(IUNIT,1)   = CONSOL TRANSFER FLAG
C     MODE(IUNIT,2)   = QUANT FLAG
C     MODE(IUNIT,3)   = QUAL FLAG
C     MODE(IUNIT,4)   = STSCAN FLAG
C     MODE(IUNIT,5)   = PKSCAN FLAG
C     MODE(IUNIT,6)   = END COUNT FLAG (ENDCNT)
C     MODE(IUNIT,7)   = END SLEW FLAG (ENSLEW)
C     MODE(IUNIT,8)   = END MOVE FLAG (ENMOVE)
C     MODE(IUNIT,9)   = BEGIN COUNT FLAG (BGCNT)
C     MODE(IUNIT,10)  = FINIS FLAG
C    MODE(IUNIT,11)    (NOT USED BY TRANS)
C    MODE(IUNIT,12)    (NOT USED BY INTRUP)
C    MODE(IUNIT,13)  = ESTIM FLAG
C    MODE(IUNIT,14-20) AVAILABLE FOR FUTURE EXPANSION
C     MVCNT           = A FLAG TO ESTABLISH STEP AND COUNT METHOD
C                       = 1 FOR SIMULTANEOUS STEP AND COUNT
C                       = 0 FOR INDEPENDENT STEP THEN COUNT
C     NAME            = A SCRATCH ARRAY AVAILABLE FOR USE
C     NBG             = THE NUMBER OF BACKGROUND POINTS TO MEASURE ON
C                       ON EACH SIDE OF THE PEAK
C     NPK             = THE NUMBER OF POINTS TO MEASURE AT THE PEAK IN PEAK
C                       COUNT MODE OR THE NUMBER OF POINTS TO AVERAGE OVER
C                       AT THE MAXIMUM IN INTEGRATE MODE TO ESTIMATE THE
C                       THE PEAK INTENSITY
C   NPTPK           = NUMBER OF POINTS IN PEAK REGION (QUANT RUN)
C                       = NPK IF IPK=1
C                       = (EANG-BANG)/DANG+1-NPTBG WHERE NPTBG=NBG OR 2*NBG
C     PANG            = THE 2 THETA OF THE PEAK MAXIMUM
C     RATE            = THE MOVE RATE IN DEGREES PER MINUTE
C     TANG            = THE TARGET ANGLE OF THE CURRENT OR NEXT MOVE
C     TIME            = THE COUNT TIME PER POINT (IN SECONDS)
C     TIMLIM          = THE MAXIMUM TIME TO COUNT A LINE IN MINUTES
C                       (IE THE SUM OF THE TOTAL BACKGROUND AND PEAK TIMES)
C
C  QUANTITATIVE ANALYSIS PARAMETERS - COMMON /QUAN/
C
C     CONC            = CONCENTRATION OF ADDED STANDARD FOR SPIKING METHOD
C                           OR CONCENTRATION OF INTERNAL STANDARD.
C     FREL              = THE RELATIVE INTENSITY OF THE LINE
C     IOVLP           = THE NUMBER OF ADDITIONAL LINES OVERLAPPED WITH THIS ONE
C     IREPT           = THE CURRENT REPITION NUMBER FOR THE PRESENT MOUNT
C                       [<=IFLAG( ,5)]
C     KODE                = THE TYPE OF LINE BEING ANALYSED
C                           = 0 UNKNOWN LINE (IE OF PHASE TO BE ANALYSED)
C                           = 1 INTERNAL STANDARD LINE
C                           = 2 REFERENCE LINE (IE FROM SAMPLE HOLDER)
C     LOOP            = THE CURRENT NUMBER OF TIMES A SAMPLE HAS BEEN
C                       MOUNTED IN THE PRESENT RUN. [<=IFLAG( ,6)]
C     LPK             = THE SEQUENTIAL NUMBER OF THE PEAK FROM THE RUN
C                       FILE CURRENTLY BEING DETERMINED
C     NUM             = THE SEQUENCE NUMBERS OF THE LINES FROM THE
C                       RUN FILE TO BE DETERMINED FOR THIS UNKNOWN SAMPLE
C     SIGSTD              = THE STANDARD DEVIATION OF THE STANDARD I MEASURMENT
C     STD                 = THE STANDARD INTENSITY MEASURMENT. IF IFLAG(1,3):
C                           = 0 THEN STD IS I(UNK)/I(INT.STD.)
C                           = 1 THEN STD IS I(UNK)/I(PURE) OR I(PURE)/I(REF)
C                           = 2 THEN STD IS I(PURE)
C     STDREL          = THE STANDARD DEVIATION OF THE RELATIVE LINE INTENSITY
C
C  RUN.DATE-TIME PARAMETERS - COMMON /RNDT/
C
C     NMFIL(IUNIT,6)     NAME OF THE OUTPUT FILE
C     NREC(IUNIT)       NUMBER OF RECORDS WRITTEN ON THE OUTPUT FILE
C     ITR(IUNIT)         ACTIVITY OF THE UNIT
```

```
223       C                          1  INACTIVE
224       C                          2  POWDER DATA COLLECTION
225       C                          3  QUANT ANALYSIS
226       C                          4  SLEW
227       C     ITRUN(3,3)      DESCRIPTOR OF RUN ARRAY
228       C     IENRN(IUNIT,4)    ESTIMATED TIME OF END OF RUN
229       C
230       C
231       C-------------------------------------------------------------
232       C
233       C  INITIALIZE VARIABLES COMMON TO ALL PROGRAM FUNCTIONS
234            CALL INIT
235            IUNIT = 1
236       C
237       C  CONSOL HAS INTERRUPTED - SERVICE IT
238       C
239       10     CALL CONSOL(IUNIT)
240            GO TO 990
241       C
242       C  QUANTITATIVE ANALYSIS REQUESTED
243       C
244       20     CALL QUANT(IUNIT)
245            GO TO 990
246       C
247       C  QUALITATIVE ANALYSIS REQUESTED
248       C
249       30     CALL QUAL(IUNIT)
250            GO TO 990
251       C
252       C  INTERFACE HAS INTERRUPTED - FROM STEP SCAN
253       C
254       40     CALL STPSCN(IUNIT,NSTEP)
255            GO TO 990
256       C
257       C  THE INTERFACE HAS COMPLETED A REQUEST FROM PEAK COUNT
258       C
259       50     CALL PKSCAN(IUNIT,NSTEP)
260            GO TO 990
261       C
262       C  BEFORE CALLING SLEW CHECK IF WE ARE 10 STEPS BELOW TARGET ANGLE
263       C
264       60     CALL CKANG(IUNIT,NSTEP,IERROR)
265            IZRST(IUNIT) = 0
266            IF (NSTEP .EQ. 10) GOTO 66
267       C
268       C  INITIATE A MAXIMUM RATE MOVE TO ANGLE BANG
269       C
270            CALL BGSLEW(IUNIT)
271            GO TO 990
272       C
273       C NO INTERRUP COMPLETION EXPECTED. COMPLETE SLEW BY DRIVING UP 10 STEPS
274       C TO REMOVE THE BACKLASH IN THE GONIOMETER.REMOVE SLEW FLAG AND IF PRESENT
275       C REMOVE INTERUP FLAG FROM ITRANS. THEN BRANCH TO NEXT ROUTINE ON
276       C THE LIFO STACK
277       C
278       66     MODE(IUNIT,7) = 3
279            IZRST(IUNIT) = 1
280            IERROR = 1
281            CALL POP(IUNIT,IERROR)
282       C
283       C  IF INTRUP IS ON THE STACK REMOVE IT
284       C
285            IF(ITRANS(IUNIT,1).EQ.12) CALL POP(IUNIT,IERROR)
286            GOTO 990
287       C
288       C  CHECK FOR NORMAL SLEW COMPLETION
289       C
290       70     CALL ENSLEW(IUNIT,IERROR)
291            GO TO 990
292       C
293       C  BEGIN MOVING 2-THETA AT A RATE OF RATE(IUNIT) DEGREES PER MINUTE
294       C
295       80     CALL BGMOVE(IUNIT,NSTEP)
296            GO TO 990
```

```
297       C
298       C  BEGIN COUNTING AT THE CURRENT ANGLE
299       C
300       90     CALL BGCNT(IUNIT)
301              GO TO 990
302       C
303       C  THE LAST DATA POINT HAS BEEN COLLECTED - CLEAN UP
304       C
305       100    CALL FINIS(IUNIT)
306              GO TO 990
307       C
308       C  TRANSFER FILE
309       C
310       110    CALL TRANS(IUNIT)
311              GO TO 990
312       C
313       C  OPTIMIZE COUNTING TIME FOR A SELECTED PEAK REGION
314       C
315       130    CALL ESTIM(IUNIT,IERROR)
316              GOTO 990
317       C
318       C  ENTER THE EVENT DETECTION SUBROUTINE AND WAIT FOR SOMETHING TO HAPPEN
319       C
320       900    CALL INTRUP(IUNIT)
321       C
322       C  GET THE ADDRESS (IE THE GO TO CODE) FROM THE LIFO TRANSFER TABLE
323       C  CALLED ITRANS AND BRANCH WHEREVER IT SAYS
324       C
325       990    IGOTO = ITRANS(IUNIT,1)
326              IF(IDBUG(IUNIT).NE.1)GO TO 995
327              JPT = IPT(IUNIT)
328              WRITE(IPRT,1234)IUNIT,IACT(IUNIT),(ITRANS(IUNIT,J),J=1,10),
329             1 (MODE(IUNIT,J),J=1,20),JPT,NREC(IUNIT),MVCNT(IUNIT),
330             2 CANG(IUNIT),TANG(IUNIT),DANG(IUNIT),NSTEP,TIME(IUNIT),
331             3 BKGTIM(IUNIT),ICNT(IUNIT,JPT)
332       1234   FORMAT(1H ,'UNIT#',I1,' IACT=',I2,' ITRANS=',10I2,' MODE=',20I2,
333            1 ' IPT=',I2,' NREC=',I5/7X,' MVCNT=',I1,' CANG=',F7.3,
334            2 ' TANG=',F7.3,' DANG=',F7.3,' NSTEP=',I5,' TIME=',F6.2,
335            3 ' BKGTIM=',F6.2,' CNT=',I5)
336       995    GO TO(10,20,30,40,50,60,70,80,90,100,110,900,130),IGOTO
337       999    CALL EXIT
338              END
```

```
C-------------------------------------------------------------------
      SUBROUTINE ALOT(IUNIT,ID,ISIZE)
C-------------------------------------------------------------------
C
C  THIS SUBROUTINE ALLOCATES THE OUTPUT DATA FILE AND WRITES
C  THE FIRST FIVE RECORDS TO IT
C
C
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*4 NBS
      COMMON /BUFF/     IALPH(66),     NAME(15)
C
      COMMON /IO/       NBS,           IDT,          ICLOCK,
     1 ISPEC,           IRD,           IPRT,         ITYP,
     2 IDISK(2,2),      IFIN,          IFOUT
C
      COMMON /PROB/     IDAT(3),       IFLAG(2,10),  ITIM(3),
     1 ITITLE(50)
C
      DIMENSION XLEN(2)
      DATA XLEN/',128', ',80 '/
      DATA ID0/'D0'/,IF0/'F0'/
C*********************************************************************
C
C STANDARD DEVICE ALLOCATION
C
      IDEV=ID0
      RECL=XLEN(1)
      IF(ID.NE.IFOUT) GOTO 10
C  OUTPUT DEVICE IS TO FLOPPY F0 WITH RECORD LENGTH OF 80
      IDEV=IF0
      RECL=XLEN(2)
C
C  DETERMINE THE NUMBER OF DIGITS IN ISIZE
C
10    IF(ISIZE.GT.9)GO TO 100
C
C  ALLOCATE A FILE OF LESS THAN 10 CYLINDERS
C
      ENCODE(IALPH,1000)(NAME(I),I=1,6),ID,IDEV,ISIZE,RECL
1000  FORMAT('AL ',6A1,',',Z1,A2,',',I1,A4)
      ISIZE = 19
      GO TO 200
C
C  ALLOCATE A FILE OF GREATER THAN 9 CYLINDERS (BUT LESS THAN 100!)
C
100   ENCODE(IALPH,1010)(NAME(I),I=1,6),ID,IDEV,ISIZE,RECL
1010  FORMAT('AL ',6A1,',',Z1,A2,',',I2,A4)
      ISIZE = 20
C
C  ALLOCATE IT
C
200   CALL SYSTEM(IALPH,ISIZE)
C
C  IF THIS IS A FILE TRANSFER RUN SKIP TO END
C
      IF( ID .EQ. IFOUT) GOTO  900
C
C  WRITE THE FIRST 5 RECORDS
C
      WRITE(ID,1020)(ITITLE(I),I=1,50),(IDAT(I),I=1,3),
     1 (ITIM(I),I=1,3),(IFLAG(IUNIT,I),I=1,10)
1020  FORMAT(50A2,6I2,10I1)
C
C
C  WHICH UNIT REQUESTED ?
C
      IF(IUNIT .EQ. 1) GOTO 300
C
C SKIP THE FIRST 4 INSTRUMENT RECORDS WHICH BELONG TO UNIT # 1
C
      DO 220 J=1,4
220   READ(ISPEC,1030) (IALPH(I),I=1,64)
C
```

```
      C READ THE INSTUMENT SPECIFICATION AND CALIBRATION FILE
      C COPY TO OUTPUT FILE
      C
        300 DO 400 J=1,4
            READ(ISPEC,1030) (IALPH(I),I=1,64)
            WRITE(ID,1030) (IALPH(I),I=1,64)
       1030 FORMAT(64A2)
        400 CONTINUE
      C
      C  REWIND SPECIFICATION FILE
      C
            REWIND ISPEC
      C
      900   RETURN
            END
```

```
C-------------------------------------------------------------------
      SUBROUTINE BGCNT(IUNIT)
C-------------------------------------------------------------------
C
C  THIS SUBROUTINE STARTS THE COUNTER IN FIXED TIME/FIXED COUNT MODE
C  THE CALLING ROUTINE MUST SET UP TIME( ) AND LIMCNT( )
C  THE INTERFACE WILL STOP COUNTING WHENEVER EITHER IS REACHED.
C
C  A MAX TIME OF 3270 SEC IS ASSUMED DUE TO 16 BIT WORD SIZE AND USE
C  OF A 0.1 SEC TIME BASE (6.5K IF TIME =0).
C  MAXIMUM OF 32700 COUNT LIMIT OCCURS DUE TO 16 BIT WORDSIZE
C   ( 65K IF LIMCNT = 0)
C
C
C   MODES OF ACTION BASED ON THE BGCNT STATUS FLAG:
C
C  MODE(IUNIT,9) = 0 DO NOT POP THE TRANSFER ADDRESS STACK
C                    THIS USED WHEN ANOTHER SUBROUTINE CALLS BGCNT
C  MODE(IUNIT,9) = 1 POP THE BGCNT ADDRESS OFF OF THE STACK -
C                    THIS IS USED WHEN MAIN CALLS BGCNT.
C
C                    IN ADDITION MODES 0 AND 1 WILL COUNT BASE ON
C                    TIME IN TIME(IUNIT) OR THE VALUE IN KOUNT(IUNIT)
C  MODE(IUNIT,9) = 2 POP THE STACK AND DO A FIXED TIME COUNT ON BACKGROUND
C                    THE NUMBER OF SECONDS FOR THIS COUNT ARE FOUND
C                    IN BKGTIM NOT IN TIME.
C
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*4 NBS,INTR,NBSADD,NBSSUB,ICNTL,ICP,KONT
C
      COMMON /CNTL/     IACT(2),       IDBUG(2),      INTR,
     1 IRUN(2),         ITRANS(2,20),  MODE(2,20),    MVCNT(2)
C
      COMMON /COUN/     ICNT(2,16),    IPT(2),        ITOT(2),
     1 LIMCNT(2),       ISAVE(2)
C
      COMMON /IO/       NBS,           IDT,           ICLOCK,
     1 ISPEC,           IRD,           IPRT,          ITYP,
     2 IDISK(2,2),      IFIN,          IFOUT
C
      COMMON /TIM/      BKGTIM(2),     FIXERR(2),     IPK(2),
     1 NBG(2),          NPK(2),        TIME(2),       TIMLIM(2)
C
C
C*********************************************************************
C
C  SET THE FIRST 3 BITS OF THE CONTROL WORD TO 101 IN ORDER
C  TO ESTABLISH A TIME BASE OF 0.1 SECONDS.
C  (THE FIRST 3 BITS = THE POWER OF TEN WHICH MUST BE DIVIDED
C  INTO 10**6 TO GET THE DESIRED TIME BASE)
C
      DATA ICNTL/X'A9F2'/
C
C  SET INTERFACE  ADDRESS
C
      NBSADD = IUNIT * 4
C
C  SET THE ACTIVITY FLAG AND TEST MODE
C
      IACT(IUNIT) = 1
      IF(MODE(IUNIT,9).EQ.0)GO TO 200
C
C  POP FINIS ADDRESS OFF TRANSFER TABLE
C
      IERROR = 1
      CALL POP(IUNIT,IERROR)
C
C  THIS SECTION STARTS THE COUNTER IN FIXED TIME MODE TO
C  COUNT FOR THE NUMBER OF SECONDS STORED IN TIME(IUNIT)
C  OR IF MODE(IUNIT,9) = 2 THEN USE NUMBER OF SECONDS IN BKGTIM(IUNIT)
C
200   KONT = LIMCNT(IUNIT)
      TM = TIME(IUNIT)
      IF(MODE(IUNIT,9).NE.2)GO TO 300
```

```
      C
      C  THIS SECTION SETS UP FOR A BACKGROUND COUNT
      C
            TM = BKGTIM(IUNIT)
      C
      C  WE ARE NOW READY TO SET THE INTERFACE
      C  FIRST SELECT A NCN ZERO ADDRESS
      C
      300   NBSSUB = NBSADD + 1
            INTR = ISTAT(NBS,NBSSUB)
      C
      C  NOW SELECT THE FIRST ZERO REGISTER
      C
            INTR = ISTAT(NBS,NBSADD)
      C
      C  SEND THE CONTROL WORD
      C
            CALL OUTBUS(NBS,ICNTL)
      C
      C  SET INTERFACE FOR THE NUMBER OF CLOCK PULSES DESIRED
      C  (BY PLACING THE NUMBER INTO THE SECOND ZERO REGISTER)
      C
            ICP = IFIX2(TM * 10. + .5)
      C  MINIMUM TIME IS 0.1 SEC COUNT TIME
            IF(ICP.LT.1) ICP = 1
            CALL OUTBUS(NBS,ICP)
      C
      C  SET THE MAXIMUM NUMBER OF COUNTS INTO THE THIRD REGISTER
      C  ON THE INTERFACE
      C
            NBSSUB = NBSADD + 3
            INTR = ISTAT(NBS,NBSSUB)
            CALL OUTBUS(NBS,KONT)
      C
      C  RESET THE INTERFACE TO ZERO REGISTER
      C
            INTR = ISTAT(NBS,NBSADD)
      C
      C  START THE COUNTER
      C
            CALL COMAND(NBS,'S')
            CALL COMAND(NBS,'A')
      999   RETURN
            END
```

```
C_____________________________________________________________________
      SUBROUTINE BGMOVE(IUNIT,NSTEP)
C_____________________________________________________________________
C
C  SUBROUTINE TO START THE MOTOR MOVING NSTEPS
C  IF NSTEPS IS NEGATIVE MOTION IS TOWARD LOW ANGLE
C  IF NSTEPS IS POSITIVE MOTION IS TOWARD HIGH ANGLE
C
C  IF IACT = 3 (IE SLEW) THEN MOTION WILL BE SET AT MAXIMUM RATE
C  (IE 30 DEGREES PER MINUTE)
C
C  IF IACT < 3 THE RATE CODE STORED IN IRATE(IUNIT) WILL BE USED
C  THE VARIABLE IRATE(IUNIT) IN COMMON/ANG/ IS USED TO STORE AN
C  INTEGER RATE CODE = 30./RATE(IUNIT)
C
C  NOTE: BGMOVE DOES NOT HAVE A MODE FLAG ASSOCIATED WITH IT
C        IN THAT IT ALWAYS POP'S THE ADDRESS STACK
C
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*4 NBS,INTR,NBSADD,NBSSUB,ICNTL,KRATE
C
      COMMON /ANG/      BANG(2),       CANG(2),       DANG(2),
     1 EANG(2),         IRATE(2),      PANG(2),       RATE(2),
     2 TANG(2),         IZRST(2)
C
      COMMON /CNTL/     IACT(2),       IDBUG(2),      INTR,
     1 IRUN(2),         ITRANS(2,20),  MODE(2,20),    MVCNT(2)
C
      COMMON /IO/       NBS,           IDT,           ICLOCK,
     1 ISPEC,           IRD,           IPRT,          ITYP,
     2 IDISK(2,2),      IFIN,          IFOUT
C
C
C*********************************************************************
      DATA ICNTLP/X'82FC'/, ICNTLN/X'82DC'/
C
C  SET THE ACTIVITY FLAG
C
      IF(IACT(IUNIT).NE.3)IACT(IUNIT) = 2
C
C  POP BGMOVE ADDRESS OFF TRANSFER TABLE
C
50    IERROR = 0
      CALL PCP(IUNIT,IERROR)
C
C  SET TIME BASE BASED ON RATE OF 30 DEG/MIN IF IN SLEW MODE
C
      IF(IACT(IUNIT) .LT. 3) GOTO 80
      KRATE = 1
      GOTO 90
C
C  SET TIME BASE FOR A RATE = RATE(IUNIT) SET UP BY CALLING ROUTINE
C  INTO RATE CODE IRATE(IUNIT)
C
80    KRATE = IRATE(IUNIT)
C
C  ESTABLISH THE BUS ADDRESS OF THE INTERFACE DESIRED
C
90    NBSADD = IUNIT * 4
C
C  SET THE CONTROL WORD WITH A TIME BASE OF 0.01 SEC FOR POSITIVE
C  (IE TOWARD HIGH ANGLE) MOTION
C
      IF(NSTEP.LT.0)GO TO 100
      ICNTL = ICNTLP
      GO TO 200
C
C  SET THE SAME CONTROL WORD EXCEPT FOR NEGATIVE DIRECTION
C
100   ICNTL = ICNTLN
      NSTEP = -NSTEP
C
C  SET THE INTERFACE ADDRESS TO NON ZERO
C
200   NBSSUB = NBSADD + 2
```

```
      INTR = ISTAT(NBS,NBSSUB)
C
C  NOW SET BACK TO THE FIRST ZERO ADDRESS
C
      INTR = ISTAT(NBS,NBSADD)
C
C  LOAD THE CONTROL WORD INTO THE INTERFACE REGISTER '1ST ZERO'
C
      CALL OUTBUS(NBS,ICNTL)
C
C  LOAD THE RATE CODE INTO THE INTERFACE REGISTER '2ND ZERO'
C
      CALL OUTBUS(NBS,KRATE)
C
C LOAD THE RATE CODE INTO THE INTERFACE REGISTER 'ONE'
C
      NBSSUB = NBSADD + 1
      INTR = ISTAT(NBS,NBSSUB)
      KRATE=NSTEP
      CALL OUTBUS(NBS,KRATE)
C
C  RESET TO THE FIRST ZERO INTERFACE ADDRESS
C
      INTR = ISTAT(NBS,NBSADD)
C
C  START THE MOTOR DRIVING
C
      CALL COMAND(NBS,'S')
      CALL COMAND(NBS,'A')
999   RETURN
      END
```

```
C-------------------------------------------------------------------
      SUBROUTINE BGSLEW(IUNIT)
C-------------------------------------------------------------------
C  THIS SUBROUTINE MOVES AT MAXIMUM RATE TO BANG
C  CHECKING THAT COLLISSIONS WILL NOT OCCUR
C
C  INITIALLY SETS TO 0.05 DEG. BELOW TARGET ANGLE
C  AFTER COMPLETION, ENSLEW WILL DRIVE UP 0.05 TO REMOVE BACKLASH
C===================================================================
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*4 NBS, INTR
C
      COMMON /ANG/      BANG(2),       CANG(2),       DANG(2),
     1 EANG(2),         IRATE(2),      PANG(2),       RATE(2),
     2 TANG(2),         IZRST(2)
C
      COMMON /CNTL/     IACT(2),       IDBUG(2),      INTR,
     1 IRUN(2),         ITRANS(2,20),  MODE(2,10),    MVCNT(2)
C
      COMMON /IO/       NBS,           IDT,           ICLOCK,
     1 ISPEC,           IRD,           IPRT,          ITYP,
     2 IDISK(2,2),      IFIN,          IFOUT
C
C*******************************************************************
C
C  POP SLEW ADDRESS OFF THE TRANSFER TABLE
C
50    IERROR = 1
      CALL POP(IUNIT,IERROR)
C
C
C  DRIVE TO 10 STEPS (0.05 DEGREES) LOWER THAN THE TARGET ANGLE
C  ENSLEW WILL MOVE UP THE LAST 10 STEPS TO REMOVE BACKLASH
C
      TANG(IUNIT) = TANG(IUNIT)-0.05
C
C  CHECK THE LEGALITY OF THE TARGET ANGLE
C
      IERROR = 0
      CALL CKANG(IUNIT,NSTEP,IERROR)
      IF(IERROR.EQ.0)GO TO 100
C
C  CANCEL RUN - CALL FINIS
C
      MODE(IUNIT,10) = 1
      ITRANS(IUNIT,1) = 10
      GO TO 999
C
C  SET ACTIVITY CODE FOR SLEWING
C
100   IACT(IUNIT) = 3
C
C  PUSH A DUMMY ADDRESS ONTO THE TRANSFER STACK TO BE POPED BY BGMOVE
C
      CALL PUSH(IUNIT,8)
      CALL BGMOVE(IUNIT,NSTEP)
999   RETURN
      END
```

```
C-------------------------------------------------------------------
      SUBROUTINE CKANG(IUNIT,NSTEP,IERROR)
C-------------------------------------------------------------------
C
C  THIS SUBRCUTINE CHECKS FOR THE LEGALITY OF A TARGET ANGLE AND
C  COMPUTES THE NUMBER OF MOTER STEPS REQUIRED
C
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*4 NBS,INTR
C
      COMMON /ANG/      BANG(2),       CANG(2),       DANG(2),
     1 EANG(2),         IRATE(2),      PANG(2),       RATE(2),
     2 TANG(2),         IZRST(2)
C
      COMMON /CNTL/     IACT(2),       IDBUG(2),      INTR,
     1 IRUN(2),         ITRANS(2,20),  MODE(2,20),    MVCNT(2)
C
      COMMON /IO/       NBS,           IDT,           ICLOCK,
     1 ISPEC,           IRD,           IPRT,          ITYP,
     2 IDISK(2,2),      IFIN,          IFOUT
C
      DIMENSION ANGHI(2),ANGLO(2)
C
C*********************************************************************
C
C  SET THE UPPER AND LOWER ANGLE LIMITS FOR THE DIFFRACTOMETERS
C
      DATA ANGHI/155.0, 95.0/, ANGLO/2.94,2.94/
C
C  IF IERROR IS -1 SKIP THE COLLISION PROTECT SECTION -
C  THIS IS AN INITIALIZATION CALL FROM SETANG
C
      IF(IERROR.LT.0)GO TO 200
C
C  COLLISSION PROTECTION  LOGIC GOES HERE WHEN WE HAVE FIGURED IT OUT
C
C-------------------------------------------------------------------
C
C  TARGET ANGLE WILL NOT CAUSE A COLLISION - CHECK IF IT IS WITHIN
C  LIMITS
C
200   IF(TANG(IUNIT).LT.ANGLO(IUNIT).OR.TANG(IUNIT).GT.ANGHI(IUNIT))
     1 GOTO 900
C
C  ANGLE IS OK - CHECK IF IT IS A MULTIPLE OF .005 AND MAKE IT SO
C
      TANG(IUNIT) = FIXANG( TANG(IUNIT) )
      NSTEP = IFIX2( TANG(IUNIT) * 200. + .5 )
C
C  IF CALL WAS FROM INIT RETURN NSTEP AS 200 * TANG (WHICH IS CANG)
C
      IF(IERROR.LT.0)GO TO 999
C
C  IF ANY OTHER CALL RETURN NSTEP AS THE DIFFERENTIAL NUMBER
C  OF STEPS NEEDED TO MOVE TO TANG
C
      NSTEP = NSTEP - IFIX2(CANG(IUNIT)*200. + 0.5)
C
C
C  ALL CHECKS PASSED - RETURN
C
      IERROR = 0
      GO TO 999
C
C  ERROR DETECTED
C
900   IERROR = 1
      WRITE(ITYP,1100) TANG(IUNIT),IUNIT,ANGLO(IUNIT),ANGHI(IUNIT)
1100  FORMAT(1H0,'TARGET ANGLE',F8.3,' FOR UNIT#',I2,' IS OUTSIDE',
     1 ' LIMITS OF',F10.3,' AND',F10.3)
999   RETURN
      END
```

```
C-----------------------------------------------------------------------
      SUBROUTINE CONSOL(IUNIT)
C_______________________________________________________________________
C
C  THIS SUBROUTINE PROCESSES CONSOLE INTERRUPTS
C
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*4 NBS, INTB
C
      COMMON /ANG/      BANG(2),      CANG(2),      DANG(2),
     1 EANG(2),         IRATE(2),     PANG(2),      RATE(2),
     2 TANG(2),         IZRST(2)
C
      COMMON /BUFF/     IALPH(66),    NAME(15)
C
      COMMON /CNTL/     IACT(2),      IDBUG(2),     INTR,
     1 IRUN(2),         ITRANS(2,20), MODE(2,20),   MVCNT(2)
C
      COMMON /IO/       NBS,          IDT,          ICLOCK,
     1 ISPEC,           IRD,          IPRT,         ITYP,
     2 IDISK(2,2),      IFIN,         IFOUT
C
      COMMON /PROB/     IDAT(3),      IFLAG(2,10),  ITIM(3),
     1 ITITLE(50)
C
      COMMON /RNDT/     NREC(2),      NMFIL(2,6),   ITR(2),
     1       IENRN(2,4)
C
      DIMENSION ITRUN(4,3)
C
      DATA ITRUN/'IN','PO','QU','SL','AC','WD','AN','EW','TV','ER',
     1 'T ','  '/
C
C***********************************************************************
C
C
C  POP THE CONSOL ADDRESS OFF OF THE TRANSFER STACK
C
      IERROR = 0
      CALL PCP(IUNIT,IERROR)
C
C  WRITE TO THE CONSOL CRT THE OPTION MENU
C
10    WRITE(ITYP,1000)
1000  FORMAT(1H0,9X,'** AUTO CONTROL **'//
     1 1H ,'ENTER ONE OF THE FOLLOWING OPTIONS (Z1):'/
     2 5X,'0 - STATUS'/
     3 5X,'1 - POWDER PATTERN COLLECTION'/
     4 5X,'2 - QUANTITATIVE ANALYSIS'/
     5 5X,'4 - SET 2-THETA'/
     6 5X,'5 - SLEW TO 2-THETA'/
     7 5X,'6 - TERMINATE RUN ON A UNIT'/
     8 5X,'7 - PANIC STOP ON BOTH UNITS'/
     9 5X,'C - LIST DISK CONTENTS'/
     9 5X,'D - DEBUG'/
     9 5X,'E - DELETE A FILE'/
     8 5X,'F - TRANSFER FILE')
20    READ(IRD,1010,ERR=10)ICODE
1010  FORMAT(Z1)
C
C  WAS A LEGAL CODE ENTERED?
C
      IF(ICODE.GE.0.AND.ICODE.LT.16)GO TO 30
22    WRITE(ITYP,1020)ICODE
1020  FORMAT(1H ,'ILLEGAL OPTION',I2)
      GO TO 10
C
C  READ THE UNIT NUMBER IF NEEDED
C
30    IF(ICODE.EQ.13)GO TO 35
      IF(ICODE.LT.1.OR.ICODE.GT.6)GO TO 60
35    WRITE(ITYP,1030)
1030  FORMAT(1H ,'ENTER UNIT NUMBER (I1)')
```

```
74            READ(IRD,1032) IUNIT
75          1032  FORMAT(I1)
76          C
77          C  WAS A LEGAL UNIT NUMBER ENTERED?
78          C
79                IF(IUNIT.EQ.1.OR.IUNIT.EQ.2)GO TO 50
80                WRITE(ITYP,1040)IUNIT
81          1040  FORMAT(1H ,'UNIT#',I2,' DOESNT EXIST')
82                GO TO 10
83          C
84          C  IF KILL UNIT# BRANCH DIRECTLY TO 200
85          C
86          50    IF(ICODE.EQ.6) GOTO 200
87          C
88          C  DEBUG FLAG OPTION CAN BE SET WHILE A UNIT IS ACTIVE. SKIP ACTIVE TEST
89          C
90                IF(ICODE.EQ.13) GOTO 80
91          C
92          C  IS THE UNIT REQUESTED FREE?
93          C
94                IF(IRUN(IUNIT).EQ.0.AND.IACT(IUNIT).EQ.0)GO TO 60
95                IF(IACT(IUNIT).LT.0)GO TO 55
96                WRITE(ITYP,1050)IUNIT
97          1050  FORMAT(1H ,'UNIT#',I2,' IS ACTIVE')
98                GO TO 10
99          C
100         C  THE INTERFACE FOR THIS UNIT IS NOT OPERABLE
101         C
102         55    IF(ICODE.EQ.4)GO TO 80
103               WRITE(ITYP,1055)IUNIT
104         1055  FORMAT(1H0,'INTERFACE FOR UNIT',I2,' IS NOT WORKING',
105              1 '-RESET 2-THETA'/)
106               GO TO 10
107         C
108         C  GET DISK PACK CODE FOR OPTIONS C & E
109         C
110         60    IF(ICODE.NE.12 .AND. ICODE.NE.14) GOTO 80
111               WRITE(ITYP,62)
112         62    FORMAT(1H ,'ENTER DISK PACK (A2)')
113               READ(IRD,64) IPACK
114         64    FORMAT(A2)
115         C
116         C  NOW BRANCH TO SERVICE THE REQUEST MADE
117         C
118         80    ICODE = ICODE + 1
119         C  TRANSFERS TO LABEL 22 ARE RESERVED OPTIONS NOT CURRENTLY IMPLEMENTED
120               GOTO(970,400,300,22,950,500,200,600,22,22,22,22,750,900,800,
121              * 700), ICODE
122         C
123         C  KILL THE JOB ON IUNIT AS FOLLOWS:
124         C  FIRST CLEAR THE TRANSFER TABLE
125         C
126         200   IF(IACT(IUNIT).GT.0) GOTO 208
127               WRITE(ITYP,202) IUNIT
128         202   FORMAT(1H ,'UNIT #',I2,' IS INACTIVE')
129               GOTO 998
130         208   DO 210 I=1,10
131         210   ITRANS(IUNIT,I) = 0
132         C
133         C  IS THIS A PANIC STOP OR AN "I'VE HAD ENOUGH" STOP
134         C
135               WRITE(ITYP,1060)
136         1060  FORMAT(1H0,'ENTER 0 - PANIC STOP (2THETA WILL BE LOST)',
137              1 /7X,'1 - KILL JOB AFTER THE NEXT COMPLETION')
138               READ(IRD,1032)ICODE
139         C
140         C  PANIC STOP OF IUNIT - TRACKING COUNTER WILL BE OFF BY 1 OR 2 STEPS
141         C
142               MODE(IUNIT,10) = 5
143         C CALL FINIS
144               ITRANS(IUNIT,1)=10
145               IF(ICODE.EQ.0)GO TO 999
146         C
147         C  TO DO A SOFT ABORT WE WAIT FOR THE NEXT COMPLETION DETECTED BY INTUP
```

```
148       C  AND THEN CALL FINIS IN NORMAL TERMINATION MODE
149       C
150             MODE(IUNIT,10) = 2
151             GO TO 998
152       C
153       C  QUANTITATIVE ANALYSIS REQUESTED
154       C
155       C  IF THIS REQUEST WAS AFTER A NEW SAMPLE MOUNT SAVE THE MODE FLAG
156       C
157       300   IF(MODE(IUNIT,2).EQ.0)MODE(IUNIT,2) = 1
158             ITRANS(IUNIT,1) = 2
159             GO TO 999
160       C
161       C  POWDER PATTERN COLLECTION REQUESTED
162       C
163       400   MODE(IUNIT,3) = 1
164             ITRANS(IUNIT,1) = 3
165             GO TO 999
166       C
167       C  ABSOLUTE SLEW REQUESTED - GET TARGET ANGLE
168       C
169       500   WRITE(ITYP,1070) CANG(IUNIT)
170       1070  FORMAT(1H ,'CURRENT 2-THETA=',F7.3,5X,'ENTER TARGET 2-THETA
171            1 (F7.3)')
172             READ(IRD,1080)TANG(IUNIT)
173       1080  FORMAT(F7.3)
174       C  TRANSFER SEQUENCE: SLEW, INTRUP, ENSLEW, INTRUP
175             ITR(IUNIT) = 4
176       C
177             MODE(IUNIT,7) = 1
178             ITRANS(IUNIT,1) = 6
179             ITRANS(IUNIT,2) = 12
180             ITRANS(IUNIT,3) = 7
181             ITRANS(IUNIT,4) = 12
182             GO TO 999
183       C
184       C  REQUEST MADE TO KILL BOTH UNITS AND RETURN TO SYSTEM
185       C
186       600   CALL CLEAR(NBS,0)
187             CALL CLEAR(NBS,0)
188             CALL EXIT
189       C
190       C  FILE TRANSFER REQUESTED USE IUNIT =1 TEMPORARILY
191       C
192       700   IUNIT = 1
193             CALL PUSH(IUNIT,11)
194             GO TO 999
195       C
196       C  LIST TABLE OF CONTENTS ON LU5
197       C
198       750   ENCODE(IALPH,756) IPACK
199       756   FORMAT('LIS ',A2)
200             CALL SYSTEM(IALPH,6)
201             GOTO 998
202       C
203       C  DELETE A FILE CURRENTLY ON DISK =IPACK
204       C
205         800 CALL FILNAM(3)
206             ENCODE(IALPH,1098)(NAME(I),I=1,6),IPACK
207       1098  FORMAT('DE ',6A1,1H, ,A2)
208             CALL SYSTEM(IALPH,12)
209             WRITE(ITYP,1099) (NAME(I),I=1,6),IPACK
210        1099 FORMAT(1H ,'FILE ',6A1,' ON ',A2,' WAS DELETED')
211       C
212       C  RETURN TO INTRUP
213       C
214             GO TO 998
215       C
216       C  RESET THE DEBUG OUTPUT FLAGS
217       C
218       900   WRITE(ITYP,1100) IUNIT
219       1100  FORMAT(1H0,'ENTER DEBUG FLAGS FOR UNIT#', I2,
220            1 5X,'0 = OFF,1 = ON,2 = OPTIM')
```

```
221             READ(IRD,1032)IDBUG(IUNIT)
222       C
223       C  RETURN TO INTRUP
224       C
225             GO TO 998
226       C
227       C  RESET THE TRACKING COUNTER ON IUNIT
228       C
229       950   WRITE(ITYP,1120)IUNIT
230       1120  FORMAT(1H0,'ENTER CURRENT 2-THETA FOR UNIT#',I2)
231             READ(IRD,1080)CANG(IUNIT)
232             CALL SETANG(IUNIT)
233             GOTO 998
234       C
235       C  TYPE STATUS OF THE DIFFRACTOMETES
236       C
237         970 J= ITR(1)
238             K= ITR(2)
239             CALL DATE
240       C
241             WRITE(ITYP,1220)IDAT,ITIM,(NMFIL(1,I),I=1,6),(NMFIL(2,I),I=1,6),
242            1 (ITRUN(J,I),I=1,3),IACT(1),(ITRUN(K,I),I=1,3),IACT(2),
243            2 (IENRN(1,I),I=1,4),(IENRN(2,I),I=1,4)
244            3 ,CANG
245        1220 FORMAT(/10X,'** STATUS ** ON ',I2,'/',I2,'/',I2,' AT ',
246            1 I2,':',I2,':',I2//
247            1 15X,'UNIT:',6X,'1',8X,'2'//
248            2  1X,'FILE:',17X,6A1,3X,6A1/
249            3  1X,'RUN:' ,13X,2(3X,3A2,I2)/
250            4  1X,'TO END AT:',3X,2(3X,I2,'/',I2,2X,I2,':',I2)/
251            5 1X,'2-THETA:',11X,2F9.3)
252       C
253       C  SET UP ITRANS FOR CALL TO INTRUP
254       C
255       998   CALL PUSH(IUNIT,12)
256       C
257       C  ALL DONE - RETURN TO AUTO FOR BRANCH TO NEXT ROUTINE ON ITRANS STACK
258       C
259       999   RETURN
260             END
```

```
1       C-------------------------------------------------------------------
2             SUBROUTINE DATE
3       C-------------------------------------------------------------------
4       C
5       C     THIS SUBROUTINE READS THE DATE AND TIME FROM CLOCK
6       C     ( DEVICE 6D UNIT = IDT  ) AND STORES THEM IN ARRAYS IDATE AND ITIM
7       C       IF DATE IDAT(1) IS EQUAL TO 00 REQUESTS FROM THE USER TO
8       C     INPUT THE DATE AND TIME . TESTS THAT YEAR STARTS WITH 8
9       C
10            IMPLICIT INTEGER*2 (I-N)
11            INTEGER*4 NBS
12      C
13            COMMON /BUFF/      IALPH(66),          NAME(15)
14      C
15            COMMON /IO/        NBS,                IDT,              ICLOCK,
16           1 ISPEC,            IRD,                IPRT,             ITYP,
17           2 IDISK(2,2),       IFIN,               IFOUT
18      C
19            COMMON /PROB/      IDAT(3),            IFLAG(2,10),      ITIM(3),
20           1 ITITLE(50)
21      C
22      C*********************************************************************
23      C
24      C  NTRY IS ATTEMPT TO SOLVE OUR DATE-TIME CLOCK READING PROBLEM
25      C
26            NTRY=0
27      C
28      C  READ DATE AND TIME FROM CLOCK
29      C
30      200   READ(IDT,201,END=230,ERR=240) IDAT,ITIM
31      201   FORMAT(3I2,2X,3I2)
32            ICODE=0
33            IF( IDAT(1) .GE. 80 ) GOTO 999
34            GOTO 250
35      230   ICODE = 1
36            GOTO 250
37      240   ICODE = 2
38      C
39      C  CLOCK IS NOT SET
40      C
41      250   WRITE(ITYP,251) ICODE,IDT
42      251   FORMAT(1H0,'CLOCK NOT SET-ICODE=',I2,' LU#=',I2)
43            NTRY=NTRY+1
44            IF(NTRY.EQ.1) GOTO 200
45            GO TO 310
46      C
47      C  ILLEGAL DATE/TIME FORMAT, TRY AGAIN
48      C
49      300   WRITE(ITYP,301)
50      301   FORMAT(1H0,'ILLEGAL DATETIME')
51      C
52      C  READ DATE AND TIME
53      C
54      310   WRITE(ITYP,311)
55      311   FORMAT(1H0,'INPUT DATE AND TIME AS'
56           1,1X,'YYMMDDHHMMSS')
57            READ(IRD,321,ERR=300) IDAT,ITIM
58      321   FORMAT(6A2)
59      C
60      C  WAS IT A LEGAL ENTRY?
61      C  CHECK THAT YEAR STARTS WITH 8,*** CHANGE TO 9 IN 1990??
62      C
63            IF(IDAT(1) .LT. '80' .OR.
64           1   IDAT(2) .GT. '12' .OR.
65           2   IDAT(3) .GT. '31' .OR.
66           3   ITIM(1) .GT. '24' .OR.
67           4   ITIM(2) .GT. '60'  )  GOTO 300
68      C
69      C  LEGAL - SET CLOCK VIA THE SYSTEM COMMAND 'SC'
70      C
71            ENCODE(IALPH,331) IDAT,ITIM
72      331   FORMAT('SC ',3A2,3(',',A2))
73            CALL SYSTEM(IALPH,18)
74      C
```

```
75      C  BRANCH TO READ OF DATE AND CLOCK
76      C
77            GO TO 200
78        999 RETURN
79            END
```

```
C_____________________________________________________________________
      SUBROUTINE ENDCNT(IUNIT,IERROR)
C_____________________________________________________________________
C
C
C  THIS SUBROUTINE READS THE COUNTER FROM EITHER FIXED TIME
C  OR FIXED COUNT MODE
C  IN EITHER FIXED TIME OR FIXED COUNT MODE OR DUAL MODE THE DATA IS
C  STORED AS FOLLOWS:
C  TIME COMPLETION  STORE COUNTS (+) IN ICNT(IPT(IUNIT))
C  COUNT COMPLETICN STORE TIME AS A NEGATIVE # IN ICNT(IPT(IUNIT))
C
C
C  THERE ARE TWO MODES OF ACTION BASED ON THE ENDCNT STATUS FLAG:
C  MODE(IUNIT,6) = 0 WAIT FOR THE SUBROUTINE TO COMPLETE BEFORE CHECKING IT
C                    THIS IS FOR CASES WHEN ENDCNT IS CALLED BY ANOTHER
C                    SUBROUTINE.
C  MODE(IUNIT,6) = 1 PROCEED DIRECTLY TO CHECK IF NORMAL COUNT COMPLETION
C                    OCCURED. THIS IS FOR THE CASE WHEN THE COMPLETION
C                    WAS DETECTED BY SUBROUTINE INTRUP AND THE STATUS
C                    FLAG IS ALREADY IN COMMON AS INTR
C
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*4 NBS,INTR,NBSADD,NBSSUB,MASKA,MASKC,MASKD,
     1 ITM,KCNT
C
      COMMON /ANG/      BANG(2),      CANG(2),      DANG(2),
     1 EANG(2),         IRATE(2),     PANG(2),      RATE(2),
     2 TANG(2),         IZRST(2)
C
      COMMON /CNTL/     IACT(2),      IDBUG(2),     INTR,
     1 IRUN(2),         ITRANS(2,20), MODE(2,20),   MVCNT(2)
C
      COMMON /COUN/     ICNT(2,16),   IPT(2),       ITOT(2),
     1 LIMCNT(2),       ISAVE(2)
C
      COMMON /IO/       NBS,          IDT,          ICLOCK,
     1 ISPEC,           IRD,          IPRT,         ITYP,
     2 IDISK(2,2),      IFIN,         IFOUT
C
C**********************************************************************
      DATA MASKA/X'C8'/, MASKC/X'0001'/, MASKD/X'1000'/
C
C  CALCULATE THE ADDRESS OF THE SELECTED INTERFACE
C
      NBSADD = IUNIT * 4
C
C  SET THE ACTIVITY FLAG AND TEST MODE
C
      IERROR = 0
      IF(MODE(IUNIT,6).NE.0)GO TO 200
C
C  GO INTO A WAIT LOOP CHECKING INTERFACE FOR COMPLETION
C
100   INTR = ISTAT(NBS,NBSADD)
      IF(IAND(INTR,MASKA).NE.MASKA)GO TO 100
C
C  READ THE STATUS OF THE INTERFACE
C
200   INTR = INBUS(NBS)
C
C  TEST IF A PRESET TIME COMPLETION OCCURED
C
      IF(IAND(INTR,MASKC).EQ.0)GO TO 240
      ICMPL = 1
      GO TO 300
C
C  IT DIDN'T SO SEE IF A MAXIMUM COUNT COMPLETION OCCURED
C
240   IF(IAND(INTR,MASKD).EQ.0)GO TO 260
      ICMPL = 2
      GO TO 300
C
C  ERROR - NEITHER TIME NOR COUNTS COMPLETED
```

```
      C
260   WRITE(ITYP,1010)IUNIT,INTR,CANG(IUNIT)
1010  FORMAT(1H0,'ENDCNT ERROR UNIT#',I1,' STATUS=',Z4,' 2-THETA=',F7.3)
      C
      C  IF FIRST ERROR TRY TO RECOVER BY STARTING COUNT CYCLE AGAIN
      C
        IF(MODE(IUNIT,6).NE.1) GOTO 900
      C
      C  THE COUNTER MUST BE ZEROED
      C
        NBSSUB = NBSADD + 3
        CALL CLEAR(NBS,NBSSUB)
      C
          MODE(IUNIT,9) = 0
          CALL BGCNT(IUNIT)
          MODE(IUNIT,6) = 0
          GOTO 100
        GO TO 900
      C
      C  READ THE INTERFACE
      C
300   NBSSUB = NBSADD + 3
      C
      C  READ THE COUNT TIME FROM INTERFACE REGISTER '2ND ZERO'
      C
        INTR = ISTAT(NBS,NBSADD)
        ITM = INBUS(NBS)
      C
      C  READ THE COUNTS FROM INTERFACE REGISTER 'THREE'
      C
        INTR = ISTAT(NBS,NBSSUB)
        KONT = INBUS(NBS)
      C
      C  THIS SECTION READS AND CHECKS THE RESULTS
      C
500   JPT = IPT(IUNIT)
        GO TO(510,520),ICMPL
      C
      C  TIME COMPLETION OCCURED - SAVE THE COUNTS
      C
510   ICNT(IUNIT,JPT) = KONT
        GO TO 999
      C
      C  MAX COUNT COMPLETION OCCURED - STORE THE TIME
      C  AS -# OF 0.1SECONDS REQUIRED TO REACH LIMCNT(IUNIT)
      C
520   ICNT(IUNIT,JPT) = -ITM
        GO TO 999
      C
      C  AN INTERFACE ERROR HAS OCCURED - CALL FINIS IN MODE 5 TO
      C  CLEAR THE RUN PARAMETERS AND RESET THE INTERFACE
      C
900   MODE(IUNIT,10) = 5
        CALL FINIS(IUNIT)
        IERROR = 1
999   RETURN
        END
```

```
1       C_____________________________________________________________________
2             SUBROUTINE ENMOVE(IUNIT,IERROR)
3       C_____________________________________________________________________
4       C
5       C  SUBROUTINE END MOVE TO CHECK THAT A MOVE COMPLETED NORMALLY
6       C  ENSURES THAT THE TRACKING COUNTER = TARGET ANGLE (TANG)
7       C  RESETS CURRENT ANGLE (CANG) TO THE TARGET ANGLE
8       C  CLEARS THE ACTIVITY FLAG
9       C
10      C
11      C  MODES OF ACTION BASED ON THE ENMOVE STATUS FLAG:
12      C
13      C  MODE(IUNIT,8) = 0 WAIT FOR THE SUBROUTINE TO COMPLETE BEFORE CHECKING IT
14      C                    THIS IS FOR CASES WHEN ENMOVE IS CALLED BY ANOTHER
15      C                    SUBROUTINE.
16      C  MODE(IUNIT,8) = 1 PROCEED DIRECTLY TO CHECK IF NORMAL MOVE COMPLETION
17      C                    OCCURED. THIS IS FOR THE CASE WHEN THE COMPLETION
18      C                    WAS DETECTED BY SUBROUTINE INTRUP AND THE STATUS
19      C                    FLAG IS ALREADY IN COMMON AS INTR.
20      C
21      C
22            IMPLICIT INTEGER*2 (I-N)
23            INTEGER*4 NBS,INTR,NBSADD,NBSSUB,MASKA,MASKB,MASKC,MASKD,MASKE
24      C
25            COMMON /ANG/       BANG(2),        CANG(2),        DANG(2),
26           1 EANG(2),          IRATE(2),       PANG(2),        RATE(2),
27           2 TANG(2),          IZRST(2)
28      C
29            COMMON /CNTL/      IACT(2),        IDBUG(2),       INTR,
30           1 IRUN(2),          ITRANS(2,20),   MODE(2,20),     MVCNT(2)
31      C
32            COMMON /COUN/      ICNT(2,16),     IPT(2),         ITOT(2),
33           1 LIMCNT(2),        ISAVE(2)
34      C
35            COMMON /IO/        NBS,            IDT,            ICLOCK,
36           1 ISPEC,            IRD,            IPRT,           ITYP,
37           2 IDISK(2,2),       IFIN,           IFOUT
38      C
39      C**************************************************************************
40            DATA MASKA/X'C8'/, MASKB/X'20'/, MASKC/X'10'/, MASKD/X'08'/
41            DATA MASKE/X'80'/
42            IERROR = 0
43      C
44      C  SET UP THE INTERFACE ADDRESS
45      C
46            NBSADD = IUNIT * 4
47      C
48      C  CLEAR THE ACTIVITY FLAG AND TEST MODE
49      C
50      50    IF(MODE(IUNIT,8).EQ.1)GO TO 200
51      C
52      C  GO INTO A WAIT LOOP CHECKING INTERFACE FOR COMPLETION
53      C
54      100   INTR = ISTAT(NBS,NBSADD)
55            IF(IAND(INTR,MASKA).NE.MASKA)GO TO 100
56      C
57      C  INTERFACE HAS COMPLETED CHECK IF ALL WENT WELL
58      C
59      C  FIRST CHECK THAT THE BUS IS NO LONGER BUSY
60      C
61      200    IF(IAND(INTR,MASKB).NE.MASKB)GO TO 220
62      C
63      C  ERROR HAS BEEN DETECTED - BUS IS NOT BUSY AND INTERFACE
64      C  HAS NOT COMPLETED!
65      C
66            WRITE(ITYP,1000)IUNIT,INTR
67      1000  FORMAT(1H0,'ENMOVE ERROR UNIT #',I1,' INTERFACE IS BUSY'
68           1 ' STATUS='Z4' ABORT')
69            GO TO 900
70      C
71      C  READ THE 16 BIT INTERFACE STSTUS WORD
72      C
73      220   INTR = INBUS(NBS)
74      C
75      C  CHECK THAT THE CORRECT NUMBER OF STEPS COMPLETED
```

```
      C
            IF(IAND(INTR,MASKC).EQ.MASKC)GO TO 300
      C
      C  THE REQUESTED NUMBER OF STEPS WERE NOT MOVED
      C  CHECK IF THE LOW ANGLE LIMIT SWITCH WAS HIT (IE BIT 12=1)
      C
            IF(IAND(INTR,MASKD).NE.MASKD)GO TO 240
      C
      C  LIMIT SWITCH HIT - KILL RUN
      C
      230   WRITE(ITYP,1010)IUNIT,INTR
      1010  FORMAT(1H ,'LIMIT SWITCH,UNIT #',I1,' STATUS=',Z4,' ABORT')
            GO TO 900
      C
      C  CHECK IF HIGH ANGLE LIMIT SWITCH WAS HIT (IE BIT 8 = 1)
      C
      240   IF(IAND(INTR,MASKE).NE.MASKE)GO TO 260
      C
      C  HIGH ANGLE LIMIT WAS HIT - ABORT!
      C
            GO TO 230
      C
      C  PROBLEM IS NOT WITH LIMIT SWITCHS - IT MUST BE ELVES!
      C
      260   WRITE(ITYP,1020)IUNIT,INTR
       1020 FORMAT(1H0,'ENMOVE ERROR,UNIT #',I1,' STATUS ',Z4,' ABORT')
            GO TO 900
      C
      C  ALL WENT WELL WITH THE MOVE SO FAR. SO LASTLY CHECK THE
      C  TRACKING COUNTER TO SEE IF IT EQUALS TANG*200
      C
      300   NBSSUB = NBSADD + 2
            INTR = ISTAT(NBS,NBSSUB)
      C
      C  READ THE TRACKING COUNTER
      C
            INTR = INBUS(NBS)
            ITEMP = IFIX2(200. * TANG(IUNIT) + 0.5)
      C    STORE INTEGER*4 TO INTEGER*2 AND TEST
            IT = INTR
            IF(IT.NE.ITEMP)GO TO 320
      C
      C  NORMAL COMPLETION
      C
            CANG(IUNIT) = TANG(IUNIT)
      C
      C  READ COUNTS AND CLEAR THE COUNTER REGISTER
      C
            NBSSUB = NBSSUB + 1
            INTR = ISTAT(NBS,NBSSUB)
      C  DO WE READ COUNTS AND SAVE THEM IN ICNT(IPT(IUNIT))?
            IF(ISAVE(IUNIT).EQ.0) GOTO 310
            JPT = IPT(IUNIT)
            IF(JPT.LT.1 .OR. JPT.GT.16) JPT = 1
            INTR = INBUS(NBS)
            ICNT(IUNIT,JPT) = INTR
            ISAVE(IUNIT) = 0
      310   CALL CLEAR(NBS,NBSSUB)
      C
      C  RESET ACTIVITY CODE AND RETURN
      C
            IACT(IUNIT) = 0
            GOTO 999
      C
      C  TRACKING COUNTER AND TARGET ANGLE DO NOT AGREE - ABORT!
      C
      320   WRITE(ITYP,1030)IUNIT,INTR,TANG(IUNIT)
      1030  FORMAT(1H0,'ENMOVE TRACKING COUNTER .NE. TARGET ANGLE'/
           1 1H ,'UNIT #',I1,I10,F10.3,' ABORT')
      C
      C  ABNORMAL TERMINATION SECTION
      C  SET FINIS MODE FLAG TO 5 AND CALL FINIS
      C
      900   MODE(IUNIT,10) = 5
            CALL FINIS(IUNIT)
```

```
151             IERROR = 1
152       C
153       C  ENMOVE IS DONE - RETURN TO CALLING PROGRAM
154       C
155       999   RETURN
156             END
```

```
C_____________________________________________________________________
      SUBROUTINE ENSLEW(IUNIT,IERROR)
C_____________________________________________________________________
C
C  THIS SUBROUTINE CHECKS THAT A SLEW COMMAND PROPERLY TERMINATED
C
C  MODES OF OPERATION:
C
C  MODE(IUNIT,7) = 0 - ENSLEW WAS CALLED BY ANOTHER SUBROUTINE AND
C                      IS TO WAIT FOR SLEW COMPLETION
C  MODE(IUNIT,7) = 1 - ENSLEW WAS CALLED FROM MAIN VIA INTRUP SO
C                      THAT COMPLETION HAS ALREADY OCCURED.
C  MODE(IUNIT,7) = 2 - ENSLEW WAS CALLED FROM ANOTHER SUBROUTINE .
C                      WITHOUT PREVIOUS SLEW MOTION.
C                      DO NOT WAIT FOR COMPLETION.
C  MODE(IUNIT,7) = 3 -CALLED FROM AUTO(MAIN). POP THE ITRANS TABLE
C                     NO PREVIOUS SLEW WAS NECESSARY
C                     SKIP TEST FOR COMPLETION
C
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*4 NBS, INTR, NBSADD, MASK
C
      COMMON /ANG/      BANG(2),       CANG(2),       DANG(2),
     1 EANG(2),         IRATE(2),      PANG(2),       RATE(2),
     2 TANG(2),         IZRST(2)
C
      COMMON /CNTL/     IACT(2),       IDBUG(2),      INTR,
     1 IRUN(2),         ITRANS(2,20),  MODE(2,20),    MVCNT(2)
C
      COMMON /IO/       NBS,           IDT,           ICLOCK,
     1 ISPEC,           IRD,           IPRT,          ITYP,
     2 IDISK(2,2),      IFIN,          IFOUT
C
      COMMON /RNDT/     NREC(2),       NMFIL(2,6),    ITR(2),
     1       IENRN(2,4)
C
C*********************************************************************
      DATA MASK/X'C8'/
C
C  CHECK MODE OF SUBROUTINE CALL
C
      IF(MODE(IUNIT,7).EQ.0)GO TO 100
      IF(MODE(IUNIT,7).EQ. 2) GOTO 300
C
C  ENSLEW WAS CALLED FROM MAIN - POP ITS ADDRESS FROM TRANSFER TABLE
C
      IERROR = 0
      CALL POP(IUNIT,IERROR)
      IF(MODE(IUNIT,7).EQ.3) GOTO 300
      GO TO 200
C
C  AWAIT COMPLETION OF SLEW MOTION
C
100   NBSADD = IUNIT * 4
150   INTR = ISTAT(NBS,NBSADD)
      IF(IAND(INTR,MASK).NE.MASK)GO TO 150
C
C  UNIT HAS COMPLETED CHECK THAT TARGET ANGLE = TRACKING COUNTER
C  AND RESET CURRENT ANGLE AND TURN OFF SLEW FLAG
C
200   MODE(IUNIT,8) = 1
      CALL ENMOVE(IUNIT,IERROR)
      IF(IERROR.EQ.1)GO TO 999
C
C  NOW WE MUST MOVE UP THE LAST 10 STEPS TO GET TO BANG AND
C  REMOVE THE GEAR LASH
C  FIRST PUT A DUMMY ADDRESS ON THE TRANSFER STACK TO BE
C  POPED BY BGMOVE. THEN ISSUE AN ABSOLUTE CALL TO BGMOVE TO
C  RETURN HERE (IE NOT USING TRANSFER STACK).
C
  300 TANG(IUNIT) = CANG(IUNIT) + 0.05
      IACT(IUNIT) = 3
      CALL PUSH(IUNIT,10)
      CALL BGMOVE(IUNIT,10)
```

```
75      C
76      C  NOW ISSUE A CALL TO END MOVE IN MODE 0 WHICH MEANS WAIT UNTIL
77      C  MOVE IS COMPLETED BEFOR RETURNIG
78      C
79            MODE(IUNIT,8) = 0
80            CALL ENMOVE(IUNIT,IERROR)
81            IF(IERROR.EQ.1)GO TO 999
82      C  IF ITR  = 4 ,SET FLAG ITR = 1 ( UNIT INACTIVE )
83      C
84            IF(ITR(IUNIT) .EQ. 4) ITR(IUNIT) =1
85      999   RETURN
86            END
```

```
C-----------------------------------------------------------------------
      SUBROUTINE ENTIM(IUNIT,RUN)
C-----------------------------------------------------------------------
C
C  THIS ROUTINE ADDS THE ESTIMATED RUN TIME TO THE PRESENT TIME
C  AND CALCULATES THE DATE AND TIME THAT THE JOB WILL BE COMPLETED
C
C  THE RESULT IS STORED IN THE ARRAY IENRN AS MONTH DAY HOURS & MIN
C
C
      IMPLICIT INTEGER*2 (I-N)
C
      COMMON /PROB/     IDAT(3),       IFLAG(2,10),   ITIM(3),
     1 ITITLE(50)
C
      COMMON /RNDT/     NREC(2),       NMFIL(2,6),    ITR(2),
     1       IENRN(2,4)
C
C**********************************************************************
C
C  GET THE DATE AND TIME
C  TO GET DATE INFORMATION ON THE ARRAY NAME
C
      CALL DATE
      IDAY = IDAT(3)
      IMON = IDAT(2)
C
C  CONVERT RUN TIME TO HRS AND MIN
C
      IHR = IFIX2(RUN)
      IMN = 60*(RUN-IHR)
C
C  ADD THE HRS AND MIN TO THE CURRENT TIME  -  FIRST THE MINUTES
C
      IMN = ITIM(2) + IMN
      IF(IMN .LT. 60) GOTO 100
      IHR = IHR + 1
      IMN = IMN - 60
C
C  NEXT HOURS
C
100   IHR = ITIM(1) + IHR
      IF (IHR .LT. 24) GOTO 200
      IDAY = IDAY + 1
      IHR = IHR -24
C
C  STORE INTO END RUN ARRAY IENRN(IUNIT, )
C
200   IENRN(IUNIT,4) = IMN
      IENRN(IUNIT,3) = IHR
      IENRN(IUNIT,2) = IDAY
      IENRN(IUNIT,1) = IMON
      RETURN
      END
```

```
C-------------------------------------------------------------------------
      SUBROUTINE ESTIM(IUNIT,IERROR)
C-------------------------------------------------------------------------
C
C  THIS SUBROUTINE MAKES A RAPID ESTIMATE OF THE PEAK AND BACKGROUND
C  INTENSITIES OF A LINE SO THAT THE PEAK AND BACKGROUND COUNT TIMES
C  CAN BE OPTIMIZED BY SUBROUTINE OPTIM
C
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*4 NBS, INTR
C
      COMMON /ANG/      BANG(2),      CANG(2),      DANG(2),
     1 EANG(2),         IRATE(2),     PANG(2),      RATE(2),
     2 TANG(2),         IZRST(2),     NPTPK(2)
C
      COMMON /CNTL/     IACT(2),      IDBUG(2),     INTR,
     1 IRUN(2),         ITRANS(2,20), MODE(2,20),   MVCNT(2)
C
      COMMON /COUN/     ICNT(2,16),   IPT(2),       ITOT(2),
     1 LIMCNT(2),       ISAVE(2)
C
      COMMON /IO/       NBS,          IDT,          ICLOCK,
     1 ISPEC,           IRD,          IPRT,         ITYP,
     2 IDISK(2,2),      IFIN,         IFOUT
C
      COMMON /QUAN/     CONC,         FREL,         IOVLP,
     1 KODE,            LOOP(2),      LPK(2),       NUM(2,20),
     2 SIGSTD,          STD,          STDREL
C
      COMMON /TIM/      BKGTIM(2),    FIXERR(2),    IPK(2),
     1 NBG(2),          NPK(2),       TIME(2),      TIMLIM(2)
C
C**************************************************************************
C
C  MODE(IUNIT,13) = 0   FIRST ENTRY. 2THETA MUST BE AT BANG!
C
C                 = 1   LOW BKG COUNT COMPLETED
C
C                 = 2   SLEW TO PANG OR EANG COMPLETED(ISTOP=0)
C                       SLEW BACK TO BANG COMPLETED   (ISTOP=1)
C
C                 = 3   PEAK COUNT COMPLETED
C
C==========================================================================
C
C  POP ESTIM ADDRESS OFF THE TRANSFER TABLE
C
      IERROR=1
      CALL POP(IUNIT,IERROR)
C
C  BRANCH DEPENDING ON MODE(IUNIT,13)
C
      IGO = MODE(IUNIT,13) + 1
      GOTO (10,50,260,270),IGO
C
C  IF NO BACKGROUND POINTS ARE TO BE MEASURED SET ICNT( ,1)=0 AND
C  SKIP TO PEAK ESTIMATION
C
10    IF(NBG(IUNIT).NE.0) GOTO 20
      ICNT(IUNIT,1)=0
      IGO=2
      GOTO 200
C
C  WE ARE NOW AT THE LOW ANGLE BACKGROUND - COUNT FOR ONE SECOND
C
20    TIME(IUNIT) = 1.0
      MODE(IUNIT,9) = 0
      CALL BGCNT(IUNIT)
C
C  CALL INTRUP TO AWAIT COMPLETION THEN RETURN TO ESTIM VIA LIFO STACK
C
      GOTO 990
C
C  STORE THE BACKGROUND COUNT INTO ICNT(IUNIT,1)
```

```
      C
50    IPT(IUNIT) = 1
      C
      C  CALL END COUNT FOLLOWING THE COMPLETION
      C
          MODE(IUNIT,6) = 1
          CALL ENDCNT(IUNIT,IERROR)
          IF(IERROR.EQ.1)GO TO 999
      C
      C  CHECK THE BACKGROUND FOR OVERFLOW AND ZERO
      C
          IF(ICNT(IUNIT,1))100,150,200
      C
      C  MAX COUNTS OCCURED - ICNT(IUNIT,1) = -10.0*SECONDS
      C
100   ICNT(IUNIT,1) = IFIX2(LIMCNT(IUNIT)/(-ICNT(IUNIT,1)/10)+0.5)
          GO TO 200
      C
      C  DEFAULT BACKGROUND TO 1 CPS
      C
150   ICNT(IUNIT,1) = 1
      C SET ICNT ARRAY POINTER
200   IPT(IUNIT) = 2
      C  WHICH DATA COLLECTION MODE? PEAK HEIGHT OR INTEGRATED SCAN
          IF(IPK(IUNIT).EQ.0) GO TO 220
      C  PEAK HEIGHT - SET TARGET ANGLE TO THE PEAK ANGLE
          TANG(IUNIT) = PANG(IUNIT)
          GOTO 240
      C  SCAN MODE - RAPIDLY INTEGRATE ENTIRE PEAK AREA
220   TANG(IUNIT)=EANG(IUNIT)
      C  INDICATE THAT COUNTS FROM SCAN ARE TO BE SAVED
          ISAVE(IUNIT) = 1
      C
      C  NEXT MOVE TO THE TARGET ANGLE AT SLEW RATE
      C  FIRST PUSH A DUMMY ONTO THE LIFO STACK FOR POPING BY BGMOVE
      C  SECOND CHECK FOR VALID TARGET ANGLE, THEN INITIATE MOVE
      C
240   CALL PUSH(IUNIT,8)
          CALL CKANG(IUNIT,NSTEP,IERROR)
          IF(IERROR.EQ.1) GOTO 999
          IACT(IUNIT) = 3
          CALL BGMOVE(IUNIT,NSTEP)
          GOTO 990
      C
      C  DRIVE COMPLETED
      C
260   MODE(IUNIT,8) = 1
          CALL ENMOVE(IUNIT,IERROR)
          IACT(IUNIT)=0
          IF(IERROR.EQ.1 .OR. ISTOP.EQ.1)GO TO 999
      C  IF SCAN MODE THEN THE COUNTS ARE SAVED IN ICNT(IUNIT,2) BY ENMOVE
          IF(IPK(IUNIT).EQ.0) GOTO 280
      C
      C  BEGIN A 1 SECOND COUNT AT THE PEAK
      C
          IPT(IUNIT) = 2
          CALL BGCNT(IUNIT)
          GOTO 990
      C
      C  PEAK COUNT COMPLETION
      C
270   MODE(IUNIT,6) = 1
          CALL ENDCNT(IUNIT,IERROR)
          IF(IERROR.EQ.1) GO TO 999
      C
      C  CHECK THE PEAK INTENSITY
      C
280   IF(ICNT(IUNIT,2))300,350,400
      C
      C  MAX COUNT COMPLETION OCCURED
      C
300   ICNT(IUNIT,2) = IFIX2(LIMCNT(IUNIT)/(-ICNT(IUNIT,2)/10)+0.5)
          GO TO 400
      C
```

```
      C  DEFAULT INTENSITY TO 1
      C
  350   ICNT(IUNIT,2) = 1
        GOTO 500
      C
      C  IF WE ARE IN PEAK COUNT MODE GO STRAIGHT TO THE TIME
      C  OPTIMIZING ROUTINE
      C
  400   IF(IPK(IUNIT).EQ.1)GO TO 500
      C
      C  WE ARE IN INTEGRATE COUNT MODE.  COMPUTE THE AVERAGE CPS
      C
        DELTM=60.0*(EANG(IUNIT)-BANG(IUNIT))/(RATE(IUNIT))
        ICNT(IUNIT,2) = FLOAT2(ICNT(IUNIT,2))/DELTM + 0.5
      C
      C  NOW OPTIMIZE THE PEAK AND BACKGROUND COUNT TIMES
      C
  500   CALL OPTIM(IUNIT)
      C
      C  DEBUG PRINT OUT IF IDBUG( )=2
      C
        IF(IDBUG(IUNIT).NE.2) GOTO 520
        WRITE(ITYP,510) ICNT(IUNIT,1),ICNT(IUNIT,2),FIXERR(IUNIT),
       1 TIMLIM(IUNIT),NPTPK(IUNIT),NBG(IUNIT),TIME(IUNIT),BKGTIM(IUNIT)
  510   FORMAT(1H0,'OPTIM RESULTS:',2I7,2F6.2,2I5,2F6.2)
      C
      C  SET TIME( ) AND BKGTIM( ) AS REQUIRED BY INTERFACE AND DATA COLLECTION
      C  METHOD
      C
  520   TIME(IUNIT) = FIXTIM(TIME(IUNIT),MVCNT(IUNIT))
        BKGTIM(IUNIT) = FIXTIM(BKGTIM(IUNIT),MVCNT(IUNIT))
        IF(IDBUG(IUNIT).EQ.2) WRITE(ITYP,530)TIME(IUNIT),BKGTIM(IUNIT)
  530   FORMAT(1H ,'PK & BG TIMES=',2F6.2)
      C
      C  SLEW BACK TO BANG BEFORE LEAVING ESTIM
      C
        TANG(IUNIT) = BANG(IUNIT)
        CALL PUSH(IUNIT,7)
        CALL PUSH(IUNIT,12)
        CALL PUSH(IUNIT,6)
        GOTO 999
      C
      C  COMMON CODE FOR LIFO STACK SET UP FOR CALLS TO INTRUP AND RETURN TO ESTIM
      C
  990   MODE(IUNIT,13) = IGO
        CALL PUSH(IUNIT,13)
        CALL PUSH(IUNIT,12)
  999   RETURN
        END
```

```
C-------------------------------------------------------------------
      SUBROUTINE FILEUP(IUNIT,ID)
C-------------------------------------------------------------------
C
C  THIS SUBRCUTINE GETS THE TITLE, OPERATOR NAME, DATE AND TIME AND
C  THE OUTPUT DATA FILE NAME
C
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*4 NBS, INTR
C
      COMMON /BUFF/     IALPH(66),     NAME(15)
C
      COMMON /IO/       NBS,           IDT,          ICLOCK,
     1 ISPEC,           IRD,           IPRT,         ITYP,
     2 IDISK(2,2),      IFIN,          IFOUT
C
      COMMON /PROB/     IDAT(3),       IFLAG(2,10),  ITIM(3),
     1 ITITLE(50)
C
      COMMON /RNDT/     NREC(2),       NMFIL(2,6),   ITR(2),
     1       IENRN(2,4)
C
C**********************************************************************
C
C  GET THE DATE AND TIME
C
      CALL DATE
C
C  GET THE NAME OF THE OUTPUT FILE
C
      CALL FILNAM(ID)
C
C  STORE FILE NAME IN NMFIL
C
      DO 200 I=1,6
  200 NMFIL(IUNIT,I)= NAME(I)
C
C  READ THE TITLE
C
   40 WRITE(ITYP,1030)
 1030  FORMAT(1H ,'ENTER RUN TITLE')
       READ(IRD,1040)(ITITLE(I),I=1,40)
 1040  FORMAT(40A2)
C
C  READ THE OPERATORS NAME
C
       WRITE(ITYP,1050)
 1050 FORMAT(1H ,'ENTER YOUR NAME:')
       READ(IRD,1040)(ITITLE(I),I=41,50)
       RETURN
       END
```

```
C-------------------------------------------------------------------
      SUBROUTINE  FILNAM(ID)
C-------------------------------------------------------------------
C
C THIS SUBROUTINE CHECKS IF THE NAME OF A FILE TO BE CREATED
C STARTS WITH A LETTER ,THE REMAINING 5 SYMBOLS HAVE TO BE
C LETTERS OR NUMBERS
C
C  THE ARGUMENT ID TELL WHO CALLED FILENAME
C     ID = 1 CALL FOR A DATA FILE
C     ID = 2 CALL FOR A RUN FILE
C     ID = 3 CALL FOR A TRANS FILE OR DELETING A FILE
C
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*4 NBS
C
      COMMON /BUFF/      IALPH(66),      NAME(15)
C
      COMMON /IO/        NBS,            IDT,            ICLOCK,
     1 ISPEC,            IRD,            IPRT,           ITYP,
     2 IDISK(2,2),       IFIN,           IFOUT
C
      DIMENSION KPRT(3,3)
C
      DATA KPRT/'D ','R ','D ','  ','  ','OR','  ','  ',' R'/
C
      DATA IC0/'0'/,IC9/'9'/,ICA/'A'/,ICZ/'Z'/
C
C**********************************************************************
C
C  BECAUSE OF CORE LIMITATIONS WE DO NOT CHECK ALL ASPECTS OF
C  INPUT FILE NAME
C
C
C  FILE NAME STRUCTURE
C
C     1-2  INITIALS
C     3-5  ALPHANUMERIC
C      6   R FOR A RUN FILE
C          D FOR A DATA FILE
C
C
C  ASK FOR A FILE NAME
C
300   WRITE(ITYP,301) (KPRT(ID,I), I=1,3)
301   FORMAT(1H0,'ENTER FILE NAME- 6 CHARACTERS ENDING WITH ',3A2)
      READ(IRD,311)(NAME(I),I=1,6)
311   FORMAT(6A1)
C
C  CHECK IF NAME IS LEGAL
C  CHECK THAT THE FIRST TWO CHARACTORS ARE LETTERS
C
      DO 400 I=1,2
      IF(NAME(I).LT.ICA.OR.NAME(I).GT.ICZ)GO TO 600
400   CONTINUE
C
C  CHECK THAT THE LAST LETTER IS R OR D
C
      IDM=NAME(6)
      IF( ID .EQ. 1 .AND. IDM .EQ. 'D') GOTO 450
      IF( ID .EQ. 2 .AND. IDM .EQ. 'R')GOTO 450
      IF( ID .GT. 2 .AND. ( IDM .EQ. 'R' .OR. IDM .EQ. 'D'))GOTO 450
      GO TO 600
C
C  CHECK THAT LETTERS 3,4 AND 5 ARE ALPHAMERIC
C
450   DO 500  I=3,5
      IF(NAME(I) .GE. ICA .AND. NAME(I) .LE. ICZ) GOTO 500
      IF(NAME(I) .GE. IC0 .AND. NAME(I) .LE. IC9) GOTO 500
      GOTO 600
500   CONTINUE
C
C  NAME IS LEGAL
C
```

```
75              GOTO 999
76        C
77        C  ILLEGAL NAME TRY AGAIN
78        C
79        600   WRITE(ITYP,601)
80        601   FORMAT(1H ,'ILLEGAL FILE NAME')
81              GOTO 300
82        999   RETURN
83              END
```

```
C-------------------------------------------------------------------
      SUBROUTINE FINIS(IUNIT)
C-------------------------------------------------------------------
C
C
C  THIS SUBROUTINE RESETS A UNIT AND TERMINATES A RUN
C
C  THERE ARE FOUR MODES OF ACTION BASED ON THE FINIS STATUS FLAG:
C  MODE(IUNIT,10) = 1 - RESET MODE
C                          IMMEDIATLY HALT IUNIT
C                         WRITE ANY REMAINING DATA TO THE OUTPUT FILE
C                         RESET ALL RUN PARAMETERS
C                         START SLEW TO 80.0
C  MODE(IUNIT,10) = 2 - NORMAL TERMINATION OF DATA COLLECTION
C                         WRITE REMAINING DATA
C                         RESET RUN PARAMETERS
C                         BEGIN SLEW TO 80.0 DEGREES
C  MODE(IUNIT,10) = 3 - (NOT USED)
C  MODE(IUNIT,10) = 4 - (NOT USED)
C  MODE(IUNIT,10) = 5 - INTERFACE ERROR EXIT
C                          HALT IUNIT
C                          WRITE ANY REMAINING DATA TO OUTPUT FILE
C                          RESET ALL RUN PARAMETERS
C                        SET IACT TO -1 AND DO NOT SLEW TO 80.0
C
C
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*4 NBS, INTR, NBSADD, NBSSUB, ICNTL
C
      COMMON /ANG/      BANG(2),      CANG(2),      DANG(2),
     1 EANG(2),         IRATE(2),     PANG(2),      RATE(2),
     2 TANG(2),         IZRST(2)
C
      COMMON /CNTL/     IACT(2),      IDBUG(2),     INTR,
     1 IRUN(2),         ITRANS(2,20), MODE(2,20),   MVCNT(2)
C
      COMMON /IO/       NBS,          IDT,          ICLOCK,
     1 ISPEC,           IRD,          IPRT,         ITYP,
     2 IDISK(2,2),      IFIN,         IFOUT
C
      COMMON /PROB/     IDAT(3),      IFLAG(2,10),  ITIM(3),
     1 ITITLE(50)
C
      COMMON /RNDT/     NREC(2),      NMFIL(2,6),   ITR(2),
     1       IENRN(2,4)
C
      COMMON /TIM/      BKGTIM(2),    FIXERR(2),    IPK(2),
     1 NBG(2),          NPK(2),       TIME(2),      TIMLIM(2)
C
C
C*********************************************************************
      DIMENSION IASGN(3), IO1011(2)
      DATA ICNTL/X'000C'/
C
C  DATA ASSIGNMENTS FOR FREEING IO UNIT 10 OR 11
C
      DATA IASGN/'AS',' ','20'/, IO1011/' A',' B'/
      IASGN(2) = IO1011(IUNIT)
C
C  IF IN INTERFACE ERROR MODE DO NOT POP STACK
C
      IF(MODE(IUNIT,10).EQ.5)GO TO 50
C
C IF ERROR FROM SLEW (ITR=4) ,SET INACT AND STOP
C
      IF(MODE(IUNIT,10).NE.1.OR.ITR(IUNIT).NE.4)GOTO 10
      MODE(IUNIT,10) = 5
      GOTO 140
C
C POP FINIS ADDRESS OFF TRANSFER TABLE
C
10    IERROR = 0
      CALL POP(IUNIT,IERROR)
C
```

```
      C  IF NORMAL TERMINATION SKIP FORWARD TO WRITE REMAINING DATA
      C
           IF(MODE(IUNIT,10).EQ.2)GO TO 140
      C
      C  RESET UNIT MODE  SEND CONTROL WORD TO HALT
      C
      50   NBSADD = IUNIT * 4
           NBSSUB = NBSADD + 2
      C
      C  GET TO FIRST ZERO REGISTER ON INTERFACE AND HALT BY SENDING CONTROL WORD
      C
           INTR = ISTAT(NBS,NBSSUB)
           INTR = ISTAT(NBS,NBSADD)
           CALL OUTBUS(NBS,ICNTL)
      C
      C  IF WE CLEARED THE UNIT DURING A MOVE WE MUST NOW ESTABLISH
      C  THE CURRENT 2-THETA POSITION
      C
           INTR = ISTAT(NBS,NBSSUB)
           INTR = INBUS(NBS)
           CANG(IUNIT) = FLOAT(INTR) / 200.
      C
      C  WRITE ANY REMAINING DATA TO THE OUTPUT FILE AND ZERO ARRAY
      C
      140  IF(ITR(IUNIT).EQ.4) GOTO 300
           CALL WRFILE(IUNIT)
      C
      C  ENDFILE AND CLOSE THE OUTPUT FILE
      C
           IFILE = IDISK(IUNIT,2)
           ENDFILE IFILE
           CALL SYSTEM(IASGN,6)
      C
      C  PRINT COMPLETION MESSAGE
      C
      C   WRITE DATE ON CONSOL
      C
        200 CALL DATE
      C
           WRITE(ITYP,1010)IUNIT,(NMFIL(IUNIT,I),I=1,6),NREC(IUNIT),
          1 IDAT,ITIM
       1010 FORMAT(1H0,'END OF RUN: UNIT #',I2,',FILE: ',6A1,',REC=',I6,
          1 ' ON ',I2,'/',I2,'/',I2,' AT ',I2,':',I2,':',I2)
      C
      C  CLEAR ALL RUN PARAMETERS FOR IUNIT
      C
           IRUN(IUNIT) = 0
           IACT(IUNIT) = 0
           ITR(IUNIT) = 1
           NREC(IUNIT) = 0
           RATE(IUNIT) = 30.0
           DO 250 I= 1,4
        250 IENRN(IUNIT,I) = 0
           IF(MODE(IUNIT,10).EQ.5)GO TO 300
      C
      C  NOW SET THE TARGET ANGLE TO 80.0 AND SLEW,INTRUP,ENSLEW,INTRUP
      C
           ITR(IUNIT) = 4
           TANG(IUNIT) = 80.
           MODE(IUNIT,7) = 1
           ITRANS(IUNIT,1)= 6
           ITRANS(IUNIT,2)=12
           ITRANS(IUNIT,3)= 7
           ITRANS(IUNIT,4)=12
           GO TO 999
      C
      C  TERMINATION DUE TO INTERFACE ERROR
      C
      300  IACT(IUNIT) = -1
           DO 310 I=1,20
      310  ITRANS(IUNIT,I) = 0
           ITRANS(IUNIT,1) = 12
      999  RETURN
           END
```

```
C-------------------------------------------------------------------
      REAL FUNCTION FIXANG(ANGLE)
C-------------------------------------------------------------------
C   FIX ANGLE SO THAT IS A MULTIPLE OF .005 DEGREES
C
      IMPLICIT INTEGER*2 (I-N)
C
C
      NSTEP = IFIX2((ANGLE+0.0005)*200.0)
      FIXANG = FLOAT2(NSTEP)/200.
C
      RETURN
      END
```

```
      C-------------------------------------------------------------------
            REAL FUNCTION FIXTIM(TIME,MVCNT)
      C-------------------------------------------------------------------
      C  SET TIME TO THE NEAREST MULTIPLE OF
      C    0.1SEC SET IN BGCNT WITH LIMIT OF 3270.0 SEC MAXIMUM (16 BIT INTEGER)
      C    0.01SEC SET IN BGMOVE
      C
            IMPLICIT INTEGER*2 (I-N)
      C
            XMULT=10.0
            IF(MVCNT.EQ.1) XMULT=100.0
      C
            ITIME = IFIX2(TIME*XMULT + 0.5)
            IF(ITIME.EQ.0) ITIME = 1
            FIXTIM = FLOAT2(ITIME)/XMULT
      C
            RETURN
            END
```

```
C-------------------------------------------------------------------
      SUBROUTINE INIT
C-------------------------------------------------------------------
C
C  THIS SUBROUTINE SETS UP ALL PROGRAM PARAMETERS COMMON
C  TO ALL PROGRAM SEGMENTS.  IT REQUIRES THE READING OF
C  THE CURRENT ANGLE ON BOTH DIFFRACTOMETERS SO THAT
C  SUBROUTINE CKANG CAN CHECK THEIR RELATIVE POSITIONS
C  IN ORDER TO PREVENT COLLISIONS.
C
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*4 NBS, INTR
C
      COMMON /ANG/      BANG(2),      CANG(2),      DANG(2),
     1 EANG(2),         IRATE(2),     PANG(2),      RATE(2),
     2 TANG(2),         IZRST(2)
C
      COMMON /BUFF/     IALPH(66),    NAME(15)
C
      COMMON /CNTL/     IACT(2),      IDBUG(2),     INTR,
     1 IRUN(2),         ITRANS(2,20), MODE(2,20),   MVCNT(2)
C
      COMMON /COUN/     ICNT(2,16),   IPT(2),       ITOT(2),
     1 LIMCNT(2),        ISAVE(2)
C
      COMMON /IO/       NBS,          IDT,          ICLOCK,
     1 ISPEC,           IRD,          IPRT,         ITYP,
     2 IDISK(2,2),      IFIN,         IFOUT
C
      COMMON /PROB/     IDAT(3),      IFLAG(2,10),  ITIM(3),
     1 ITITLE(50)
C
      COMMON /RNDT/     NREC(2),      NMFIL(2,6),   ITR(2),
     1       IENRN(2,4)
C
      COMMON /TIM/      BKGTIM(2),    FIXERR(2),    IPK(2),
     1 NBG(2),          NPK(2),       TIME(2),      TIMLIM(2)
C
C*********************************************************************
C
C  SET THE I/O UNITS
C
      NBS = 1
      IDT = 2
      ICLOCK = 3
      ISPEC = 4
      IRD = 5
      IPRT = 6
      ITYP = 7
      IDISK(1,1) = 8
      IDISK(1,2) = 10
      IDISK(2,1) = 9
      IDISK(2,2) = 11
      IFIN = 12
      IFOUT = 13
C
C  HANDLE POWER FAIL
C
      CALL SYSTEM('PF C',4)
C
C  ATTACH THE LOGICAL UNITS TO PHYSICAL DEVICES
C
      CALL SYSTEM('ZU',2)
      CALL SYSTEM('ND D0',5)
      CALL SYSTEM('AS 18B',6)
      CALL SYSTEM('AS 26D',6)
      CALL SYSTEM('AS 36C',6)
      CALL SYSTEM('AS 520',6)
      CALL SYSTEM('AS 662',6)
      CALL SYSTEM('AS 720',6)
C
C  ATTACH THE INSTRUMENT CONFIGURATION FILE
C
      ENCODE(IALPH,1080)ISPEC
```

```
 1080  FORMAT('AC DACONF,',Z1)
       CALL SYSTEM(IALPH,11)
C
C  CHECK IF THE DATE AND TIME HAVE BEEN SET
C
       CALL DATE
C
C  INITIALIZE ALL RUN PARAMETERS AND SET ACTIVITY FLAGS
C  TO INACTIVE
C
       DO 100 I=1,2
       BANG(I) = 0.0
       TANG(I) = 0.0
       EANG(I) = 0.0
       IACT(I) = 0
       IDBUG(I) = 0
       ISAVE(I)=0
       IPT(I) = 0
       IRATE(I) = 0
       IRUN(I) = 0
       ITR(I) = 1
       NREC(I) = 0
       RATE(I) = 30.0
       TIME(I) = 0.0
       BKGTIM(I) = 0.0
       FIXERR(I) = 0.0
       TIMLIM(I) = 0.0
       LIMCNT(I) = 0
       MVCNT(I) = 0
       DO 50 J=1,20
       ITRANS(I,J) = 0
       MODE(I,J)  = 0
       IF(J.GT.16) GOTO 50
       ICNT(I,J) = 0
       IF(J.GT.10) GOTO 50
       IFLAG(I,J) = 0
       IF(J.GT.6)  GOTO 50
       NMFIL(I,J) ='  '
       IF(J.GT.4)  GOTO 50
       IENRN(I,J) = 0
 50    CONTINUE
 100   CONTINUE
C
C  ESTABLISH THE CURRENT POSITION OF BOTH DIFFRACTOMETERS
C
       DO 200 I=1,2
       WRITE(ITYP,1050)I
 1050  FORMAT(1H ,'ENTER THE CURRENT 2-THETA OF UNIT #',I1,1X,
      1 '(F10.3)')
       READ(IRD,1060)CANG(I)
 1060  FORMAT(F10.3)
C
C  INITIALIZE THE TRACKING COUNTER
C
       IUNIT = I
       CALL SETANG(IUNIT)
 200   CONTINUE
C
C  ENABLE THE REAL TIME CLOCK TO INTRUPT -
C    ( FUTURE USE )
C
C     CALL SYSTEM('EF 6C',5)
C
C  CLEAR THE FLAG TABLE
C
       CALL SYSTEM('CF',2)
       RETURN
       END
```

```
C-------------------------------------------------------------------
      SUBROUTINE INTRUP(IUNIT)
C-------------------------------------------------------------------
C
C  THIS SUBRCUTINE PROCESSES INTERRUPTS FROM THE CONSOL (IE WHEN
C  ESCAPE IS HIT) OR FROM THE PRECISSION INTERVAL CLOCK OR
C  FROM A POWER FAIL.
C  IT ALSO DETECTS COMPLETIONS ON EITHER INTERFACE
C
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*4 NBS, INTR, NBSADD, NBSSUB, MASK
C
      COMMON /CNTL/     IACT(2),       IDBUG(2),      INTR,
     1 IRUN(2),         ITRANS(2,20),  MODE(2,20),    MVCNT(2)
C
      COMMON /IO/       NBS,           IDT,           ICLOCK,
     1 ISPEC,           IRD,           IPRT,          ITYP,
     2 IDISK(2,2),      IFIN,          IFOUT
C
      DIMENSION JFL(3)
C
      DATA MASK/X'C8'/, JFL/'ES','PF',X'6C'/
C**********************************************************************
C
C  POP TRANSFER ADDRESS OFF TABLE
C
      IERROR = 0
      CALL PCP(IUNIT,IERROR)
C
C  CHECK IF AN ENTRY HAS BEEN MADE INTO THE FLAG TABLE
C
100   IFL = IFLAG(0)
C
C  WAS THERE A CODE ON THE FLAG TABLE
C
      IF(IFL.EQ.0)GO TO 200
C
C  WE HAVE RECEIVED AN INTERRUPT - SEE WHO DID IT
C  CONSOL ESCAPE KEY PUTS AN 'ES' ON FLAG TABLE
C  POWER FAIL PUTS AN 'MM' ON FLAG TABLE
C  THE CLOCK PUTS AN X'6C' ON FLAG TABLE (FUTURE USE)
C
      DO 120 I=1,3
      J = I
      IF(IFL.EQ.JFL(J))GO TO 140
120   CONTINUE
      WRITE(ITYP,1000)IFL
1000  FORMAT(1H0,'ILLEGAL DEVICE INTERRUPT ',
     1 A2,' IGNORED')
C
C  CLEAR THE FLAG TABLE OF ANY MULTIPLE ENTRIES
C
130   CALL SYSTEM('CF',2)
      GO TO 100
C
C  BRANCH ON TYPE OF FLAG ON THE FLAG TABLE
C
140   GO TO(150,160,170),J
C
C  A CONSOLE INTERRUPT OCCURRED - USE IUNIT=1 TEMPORARILY
C
150   IUNIT = 1
      MODE(IUNIT,1) = 0
      CALL PUSH(IUNIT,1)
      GO TO 999
C
C  A POWER FAIL INTERRUPT HAS OCCURRED - PRINT MESSAGE
C
160   WRITE(ITYP,1010)
1010  FORMAT(1H0,'PF INTERRUPT-IGNORED')
      GO TO 100
C
C  THE PRECISSION INTERVAL CLOCK HAS INTERRUPTED
C
```

```
75      C170  WRITE(ITYP,1020)
76      C1020 FORMAT(1H0,'CLOCK INTERRUPT-IGNORED')
77      170   GO TO 100
78      C
79      C  THIS SECTION CHECKS FOR AN INTERFACE COMPLETION
80      C
81      200   IUNIT = 0
82      210   IUNIT=IUNIT + 1
83            IF(IUNIT.GT.2) GOTO 100
84            IF(IACT(IUNIT).LE.0) GOTO 210
85            NBSADD = IUNIT * 4
86            INTR = ISTAT(NBS,NBSADD)
87            IF(IAND(INTR,MASK).NE.MASK)GO TO 210
88      C
89      C  A COMPLETION HAS OCCURRED ON IUNIT - RETURN AND SERVICE IT
90      C
91      999   RETURN
92            END
```

```
C-------------------------------------------------------------------
      SUBROUTINE CPTIM(IUNIT)
C-------------------------------------------------------------------
C
C  SUBROUTINE CPTIM TO OPTIMIZE THE TIME SPENT COUNTING THE
C  PEAK AND BACKGROUND POINTS IN EITHER FIXED TOTAL TIME MODE
C  OR FIXED ERROR MODE
C
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*4 NBS, INTR
C
      COMMON /ANG/      BANG(2),       CANG(2),       DANG(2),
     1 EANG(2),         IRATE(2),      PANG(2),       RATE(2),
     2 TANG(2),         IZRST(2),      NPTPK(2)
C
      COMMON /CNTL/     IACT(2),       IDBUG(2),      INTR,
     1 IRUN(2),         ITRANS(2,20),  MODE(2,20),    MVCNT(2)
C
      COMMON /COUN/     ICNT(2,16),    IPT(2),        ITOT(2),
     1 LIMCNT(2),       ISAVE(2)
C
      COMMON /IO/       NBS,           IDT,           ICLOCK,
     1 ISPEC,           IRD,           IPRT,          ITYP,
     2 IDISK(2,2),      IFIN,          IFOUT
C
      COMMON /QUAN/     CONC,          FREL,          IOVLP,
     1 KODE,            LOOP(2),       LPK(2),        NUM(2,20),
     2 SIGSTD,          STD,           STDREL
C
      COMMON /TIM/      BKGTIM(2),     FIXERR(2),     IPK(2),
     1 NBG(2),          NPK(2),        TIME(2),       TIMLIM(2)
C
C**********************************************************************
      DUM1=FLOAT2(ICNT(IUNIT,1))
      DUM2=FLOAT2(ICNT(IUNIT,2))
      DUM3=FIXERR(IUNIT)/100.0
      BKGTIM(IUNIT) = 0.0
C
C  DETERMINE THE NUMBER OF BACKGROUND POINTS TO BE COLLECTED
C
      NPTBG=NBG(IUNIT)
      IF(EANG(IUNIT).GT.0.1) NPTBG=NPTBG+NPTBG
      XPTBG=FLOAT2(NPTBG)
C
C  DETERMINE THE NUMBER OF PEAK OR SCAN POINTS TO BE COLLECTED
C
      IF(IPK(IUNIT).EQ.1) XPTPK=FLOAT2(NPK(IUNIT))
      IF(IPK(IUNIT).EQ.0) XPTPK=(EANG(IUNIT)-BANG(IUNIT))/DANG(IUNIT)
     1 +1.0-XPTBG
      NPTPK(IUNIT) = IFIX2(XPTPK+0.001)
C
C  TEST IF BACKGROUND POINT WAS MEASURED. IF NOT SET R AND RINV TO 0.0
C
      IF(NPTBG.NE.0) GOTO 10
      R    = 0.0
      RINV = 0.0
      GOTO 16
C
C  CALCULATE THE RATIO OF PEAK TO BACKGROUND INTENSITIES
C
10    R = SQRT(DUM2/DUM1)
      IF(R.LT.1.0) R = 1.0
      RINV = 1.0/R
C
C  BRANCH TO THE PROPER COUNT MODE ROUTINE
C
16    IF(FIXERR(IUNIT).LT.0.01)GO TO 100
C
C  WE ARE IN FIXED ERROR MODE.  CHECK THAT PEAK AND BKG
C  COUNTS ARE DIFFERENT - IF NOT CONVERT TO FIXED TIME MODE
C
      IF(R.EQ.1.0)GO TO 100
C
C  COMPUTE THE PEAK COUNT TIME (IN SECONDS) FOR FIXED ERROR MODE
```

```
      C
   20    TIME(IUNIT)= (DUM2+R*DUM1)/(DUM3*DUM3*XPTPK*
     1 (DUM2-DUM1)*(DUM2-DUM1))
         GO TO 200
      C
      C  WE ARE IN FIXED COUNT TIME MODE (TIMELIM IS IN MINUTES)
      C  DETERMINE COUNT TIME=TIME(IUNIT) IN SECONDS
      C
  100    TIME(IUNIT) = 60.0*TIMLIM(IUNIT) / (XPTPK*(1.0+RINV))
      C
      C  NOW COMPUTE THE BACKGROUND COUNT TIME FOR BOTH MODES
      C
      C  IF THE MAXIMUM NUMBER OF COUNTS ALLOWED IN THE SCALER USING A TIME
      C  BASE OF .1 SEC IS EXCEEDED DEFAULT TO MAX TIME OF 3270.
      C
  200    IF(TIME(IUNIT).GT.3270.)TIME(IUNIT) = 3270.
         IF(NPTBG.EQ.0) GOTO 300
         BKGTIM(IUNIT) = XPTPK * TIME(IUNIT) /
     1 (XPTBG * R)
      C
      C  IF WE ARE IN FIXED ERROR MODE MAKE SURE THAT THE SUM OF TIME
      C  AND BKGTIM DO NOT EXCEED THE TOTAL TIME (TIMLIM)
      C
  300    IF(FIXERR(IUNIT).LT.0.001)GO TO 999
         TEMP = BKGTIM(IUNIT) * XPTBG
         TEMP = (TEMP + TIME(IUNIT) * XPTPK) / 60.
         IF(TEMP.LE.TIMLIM(IUNIT))GO TO 999
      C
      C  TOO MUCH TIME REQUIRED FOR FIXED ERROR MODE CONVERT TO FIXED TIME
      C
         FIXERR(IUNIT) = 0.0
         GO TO 100
  999    RETURN
         END
```

```
1       C-------------------------------------------------------------------
2             SUBROUTINE PKSCAN(IUNIT,NSTEP)
3       C-------------------------------------------------------------------
4       C
5       C  SUBROUTINE PEAK COUNT PERFORMS THE FOLLOWING FUNCTIONS:
6       C  1. COUNTS NBG POINTS (SEPERATED BY DANG) AT THE FIRST BACKGROUND
7       C     FOR BKGTIM SECONDS EACH
8       C  2. MOVES TO THE PEAK ANGLE - NPK/2 AND COUNTS NPK POINTS (ALSO
9       C     SEPERATED BY DANG) FOR TIME SECONDS EACH.
10      C  3. MOVES TO THE HIGH ANGLE BACKGROUND, IF IT IS NON ZERO,
11      C     AN COUNTS NBK POINTS FOR BKGTIM SECONDS EACH
12      C  4. WRITE FINAL DATA BUFFER VIA CALL TO WRFILE
13      C
14      C  WHEN ALL COUNTING IS COMPLETED RETURN TO MAIN
15      C
16            IMPLICIT INTEGER*2 (I-N)
17            INTEGER*4 NBS, INTR
18      C
19            COMMON /ANG/       BANG(2),       CANG(2),       DANG(2),
20           I EANG(2),          IRATE(2),      PANG(2),       RATE(2),
21           2 TANG(2),          IZRST(2),      NPTPK(2)
22      C
23            COMMON /CNTL/      IACT(2),       IDBUG(2),      INTR,
24           1 IRUN(2),          ITRANS(2,20),  MODE(2,20),    MVCNT(2)
25      C
26            COMMON /IO/        NBS,           IDT,           ICLOCK,
27           1 ISPEC,            IRD,           IPRT,          ITYP,
28           2 IDISK(2,2),       IFIN,          IFOUT
29      C
30            COMMON /TIM/       BKGTIM(2),     FIXERR(2),     IPK(2),
31           1 NBG(2),           NPK(2),        TIME(2),       TIMLIM(2)
32      C
33            COMMON/COUN/       ICNT(2,16),    IPT(2),        ITOT(2),
34           1 LIMCNT(2),        ISAVE(2)
35      C
36            DIMENSION TEANG(2)
37      C
38      C*********************************************************************
39      C
40      C  POP THE PEAK SCAN ADDRESS OFF OF THE STACK
41      C
42            CALL POP(IUNIT,0)
43      C
44      C  BRANCH BASED ON THE VALUE OF THE MODE FLAG
45      C
46            IGOTO = MODE(IUNIT,5)
47            GO TO(100,200,300,400),IGOTO
48      C
49      C  FIRST ENTRY - WE ARE ALREADY AT BANG (THE FIRST BACKGROUND)
50      C  SAVE THE ANGLE OF THE HIGH ANGLE BACKGROUND
51      C
52      100   TEANG(IUNIT) = EANG(IUNIT)
53      C
54      C  THE NBG BACKGROUND POINTS WILL BE MEASURED BY SUBROUTINE STEP
55      C  SCAN.  SET EANG SO THAT STEP SCAN WILL TERMINATE AFTER NBG POINTS
56      C
57            EANG(IUNIT) = BANG(IUNIT) + DANG(IUNIT) * FLOAT2(NBG(IUNIT)-1)
58      C
59      C SET PEAK SCAN MODE TO RETURN TO THE PEAK COUNTING SECTION
60      C
61            MODE(IUNIT,5) = 2
62            GO TO 998
63      C
64      C  LOW ANGLE BACKGROUND IS COMPLETE.  WE NOW MUST COUNT THE PEAK
65      C  FOR NPK POINTS. SET UP THE TARGET ANGLE FOR THE MOVE
66      C
67      200   DUMMY = FLOAT2(NPK(IUNIT)/2) * DANG(IUNIT)
68            TANG(IUNIT) = PANG(IUNIT) - DUMMY
69      C
70      C  SET THE ENDING ANGLE FOR THE PEAK POINTS
71      C
72            EANG(IUNIT) = PANG(IUNIT) + DUMMY
73            MODE(IUNIT,5) = 3
74            GO TO 998
75      C
```

```
C  WE ARE NOW FINISHED WITH THE PEAK.  HAS A HIGH ANGLE BACKGROUND
C  BEEN REQUESTED?
C
300   IF(TEANG(IUNIT).EQ.0.0)GO TO 400
C
C  HIGH ANGLE BACKGROUND IS TO BE COLLECTED. SET MODE=4 IS TO SIGNAL
C  A RETURN TO CALLING PROGRAM(VIA MAIN) AT END OF STPSCN.
C
      MODE(IUNIT,5) = 4
C
C  SET THE NEW ENDING ANGLE
C
      TANG(IUNIT) = TEANG(IUNIT) - DANG(IUNIT) * FLOAT2(NBG(IUNIT)-1)
      EANG(IUNIT) = TEANG(IUNIT)
      GO TO 998
C
C  ALL DONE - WRITE THE REMAINING DATA TO THE OUTPUT FILE
C
400   CALL WRFILE(IUNIT)
      GOTO 999
C
C  CHKANG ERROR  -  ABORT RUN BY CALLING FINIS
C
900   MODE(IUNIT,10) = 1
      ITRANS(IUNIT,1) = 10
      GOTO 999
C
C  SET UP THE TRANSFER SEQUENCE:  INTRUP, STPSCN, PKSCN
C
998   CALL PUSH(IUNIT,5)
      CALL PUSH(IUNIT,4)
C
C  SET MODE FLAG FOR STPSCN TO INDICATE POSITIONED AT BANG
C
      MODE(IUNIT,4) = 3
C  IF FIRST CALL TO PKSCN WE ARE AT STARTING ANGLE - SKIP BGMOVE CALL
      IF(IGOTO.EQ.1) GOTO 999
C  CHECK TARGET ANGLE FOR LEGAL RANGE
      CALL CKANG(IUNIT,NSTEP,IERROR)
      IF(IERROR.EQ.1) GOTO 900
C  PUSH CALL TO INTERRUPT ONTO STACK AS BGMOVE WILL BE CALLED
      CALL PUSH(IUNIT,12)
C
C  TARGET ANGLE IS OK. MOVE TO TARGET ANGLE VIA BGMOVE
C  PUSH GOTO POINTER ONTO STACK TO BE POPPED BY BGMOVE
C
      CALL PUSH(IUNIT,8)
C
C  SET MODE FLAG FOR STPSCN TO INDICATE THAT BGMOVE WAS CALLED AND THAT
C  STPSCN MUST CALL ENMOVE
C
      MODE(IUNIT,4) = 1
999   RETURN
      END
```

```
C_____________________________________________________________________
      SUBROUTINE POP(IUNIT,IERROR)
C_____________________________________________________________________
C
C  THIS SUBROUTINE POPS THE "GO TO" FLAG OUT OF ITRANS(IUNIT,1)
C  AND PUSHES ALL OF THE REST OF THE TABLE DOWN ONE.
C
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*4 NBS, INTR
C
      COMMON /CNTL/     IACT(2),       IDBUG(2),      INTR,
     1 IRUN(2),         ITRANS(2,20),  MODE(2,20),    MVCNT(2)
C
      COMMON /IO/       NBS,           IDT,           ICLOCK,
     1 ISPEC,           IRD,           IPRT,          ITYP,
     2 IDISK(2,2),      IFIN,          IFOUT
C
C*********************************************************************
C
C  IF IERROR = 0 THEN SKIP ERROR TEST ON END OF TRANSFER TABLE
C
      DO 100 I=1,19
      ITRANS(IUNIT,I) = ITRANS(IUNIT,I+1)
100   CONTINUE
      ITRANS(IUNIT,20) = 0
      IF(IERROR.EQ.0)GO TO 999
      IERROR = 0
      IF(ITRANS(IUNIT,1).GT.0.AND.ITRANS(IUNIT,1).LT.14)GO TO 999
C
C  ILLEGAL TRANSFER ADDRESS IS NOW ON TABLE
C
      WRITE(ITYP,1000)IUNIT,(ITRANS(IUNIT,I),I=1,20)
1000  FORMAT(1H0,'POP: ILLEGAL ITRANS(1)- UNIT#',I1,' ITRANS=',20I2)
      IERROR = 1
999   RETURN
      END
```

```
      C_____________________________________________________________________
            SUBROUTINE PUSH(IUNIT,IPUSH)
      C_____________________________________________________________________
      C
      C  THIS SUBROUTINE PUSHES TH "GO TO" ADDRESS IN IPUSH ONTO THE
      C  TRANSFER TABLE AND BUMPS THE REST UP BY ONE
      C
            IMPLICIT INTEGER*2 (I-N)
            INTEGER*4 NBS, INTR
      C
            COMMON /CNTL/      IACT(2),        IDBUG(2),       INTR,
           1 IRUN(2),          ITRANS(2,20),   MODE(2,20),     MVCNT(2)
      C
            COMMON /IO/        NBS,            IDT,            ICLOCK,
           1 ISPEC,            IRD,            IPRT,           ITYP,
           2 IDISK(2,2),       IFIN,           IFOUT
      C**********************************************************************
      C
      C  CHECK IF ANYTHING IS GOING TO BE PUSHED OFF
      C
            IF(ITRANS(IUNIT,20).NE.0)GO TO 200
            DO 100 I=1,19
      C  NOTE: A DECREMENTING LOOP DOSN'T WORK
            J = 20 - I
            ITRANS(IUNIT,J+1) = ITRANS(IUNIT,J)
      100   CONTINUE
            ITRANS(IUNIT,1) = IPUSH
            GO TO 999
      C
      C  TABLE IS FULL
      C
      200   WRITE(ITYP,1000)IUNIT,(ITRANS(IUNIT,I),I=1,20)
      1000  FORMAT(1H0,'ITRANS FULL- UNIT#',I1,' ITRANS=',20I2)
            CALL EXIT
      999   RETURN
            END
```

```
C-------------------------------------------------------------------
      SUBROUTINE QUAL(IUNIT)
C-------------------------------------------------------------------
C
C  QUALITATIVE ANALYSIS SET UP SUBROUTINE
C  TO COLLECT A STEP SCANNED POWDER PATTERN
C
C
C   MODE(IUNIT,3) = 2  RETURN FROM SLEW TO BANG
C                      COMPLETE INPUT OF RUN PARAMETERS
C
C
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*4 NBS, INTR
C
      COMMON /ANG/      BANG(2),       CANG(2),       DANG(2),
     1 EANG(2),         IRATE(2),      PANG(2),       RATE(2),
     2 TANG(2),         IZRST(2)
C
      COMMON /BUFF/     IALPH(66),     NAME(15)
C
      COMMON /CNTL/     IACT(2),       IDBUG(2),      INTR,
     1 IRUN(2),         ITRANS(2,20),  MODE(2,20),    MVCNT(2)
C
      COMMON /COUN/     ICNT(2,16),    IPT(2),        ITOT(2),
     1 LIMCNT(2),       ISAVE(2)
C
      COMMON /IO/       NBS,           IDT,           ICLOCK,
     1 ISPEC,           IRD,           IPRT,          ITYP,
     2 IDISK(2,2),      IFIN,          IFOUT
C
      COMMON /PROB/     IDAT(3),       IFLAG(2,10),   ITIM(3),
     1 ITITLE(50)
C
      COMMON /RNDT/     NREC(2),       NMFIL(2,6),    ITR(2),
     1       IENRN(2,4)
C
      COMMON /TIM/      BKGTIM(2),     FIXERR(2),     IPK(2),
     1 NBG(2),          NPK(2),        TIME(2),       TIMLIM(2)
C
      DIMENSION ISTD(3)
C
C**********************************************************************
C
C  POP QUAL'S TRANSFER ADDRESS OFF OF THE LIFO STACK
C
      IERROR = 0
      CALL POP(IUNIT,IERROR)
C
C  IS THIS A RETURN CALL FROM SLEW?
C
      IF(MODE(IUNIT,3).EQ.2)GO TO 100
C
C  WE ARE STARTING A QUAL RUN - SAY HELLO
C
C SET ITR TO QUAL
      ITR(IUNIT) = 2
      WRITE(ITYP,1000)IUNIT,IDAT,ITIM
 1000 FORMAT(1H0,'PCWDER PATTERN DATA COLLECTION ON UNIT #',I1
     1 ' CN ',I2,'/',I2,'/',I2,' AT ',I2,':',I2,':',I2)
      WRITE(ITYP,1005)
 1005 FORMAT(1H0,'ENTER STARTING ANGLE FOR THE RUN-F7.3-(DEFAULT=5.0)')
      READ(IRD,1010)EANG(IUNIT)
 1010 FORMAT(F7.3)
C
C ROUND BANG TO .005 DEGREES
C
      BANG(IUNIT)=FIXANG( BANG(IUNIT) )
C
C SET DEFAULT VALUE FOR BANG
C
      IF(BANG(IUNIT).LT. 0.1)BANG(IUNIT) = 5.0
C
C
```

```
75            TANG(IUNIT) = BANG(IUNIT)
76      C
77      C  BEGIN SLEWING TO THE STARTING ANGLE.THEN RETURN TO 100 TO CONTINUE
78      C  INPUT WHILE DIFFRACTOMETER IS MOVING TO TANG-10 STEPS
79      C
80            ITRANS(IUNIT,1) = 6
81            ITRANS(IUNIT,2) = 3
82            MODE(IUNIT,3) = 2
83            GO TO 999
84      C
85      C  MOTOR IS NOW MOVING - GET THE RUN PARAMETERS
86      C
87      100   CALL FILEUP(IUNIT,1)
88      C
89      C  SET RUN PARAMETERS FOR FIXED TIME COUNTING
90      C  IN STEP SCAN MODE
91      C
92            IPK(IUNIT) = 0
93            FIXERR(IUNIT) = 0.0
94      C
95      C  READ ENDING ANGLE, STEP SIZE, COUNT TIME AND SPEED
96      C
97      120   WRITE(ITYP,1020)
98      1020  FORMAT(1H0,'ENTER:END ANGLE,STEPSIZE,COUNT TIME(SEC)'
99           1 ,',METHOD,MAXCCUNTS UNDER DEFAULT VALUES'/
100          3 1H ,'90.000  0.020    1.0  0 32700')
101           READ(IRD,1022)EANG(IUNIT),DANG(IUNIT),TIME(IUNIT),
102          1 MVCNT(IUNIT),LIMCNT(IUNIT)
103     1022  FORMAT(2F7.3,F7.1,I3,I6)
104     C
105     C  SET THE DEFAULT VALUES
106     C
107           IF(EANG(IUNIT).LT.0.1)EANG(IUNIT) = 90.0
108           IF(DANG(IUNIT).LT.0.005)DANG(IUNIT) = 0.02
109           IF(TIME(IUNIT).LT.0.01)TIME(IUNIT) = 1.0
110           IF(LIMCNT(IUNIT).EQ.0)LIMCNT(IUNIT) = 32700
111     C
112     C  SET ANGLES AND CCUNT TIME TO MULTIPLES OF INTERFACE UNITS
113     C
114           DANG(IUNIT) = FIXANG( DANG(IUNIT) )
115           EANG(IUNIT) = FIXANG( EANG(IUNIT) )
116           TIME(IUNIT) = FIXTIM(TIME(IUNIT),MVCNT(IUNIT))
117     C
118     C  COMPUTE THE RUN TIME AND OUTPUT
119     C
120           DELTA = (EANG(IUNIT) - BANG(IUNIT)) / DANG(IUNIT)
121           IF(DELTA.GT.0.0)GO TO 130
122           WRITE(ITYP,1025)
123     1025  FORMAT(1H0,'ILLEGAL EANG-MUST BE > BANG')
124           GO TO 120
125     130   RUN = DELTA * (TIME(IUNIT)+.22) / 3600.
126     C*****************************************************************
127     C  NOTE: A PRECISE COMMUNICATION TIME FACTOR MUST BE MEASURED AND
128     C  INSERTED HERE. A TEMPORARY VALUE OF .21 SEC  WILL BE USED FOR NOW
129     C*****************************************************************
130     C
131     C  READ THE CODES FOR THE INTERNAL STANDARDS USED
132     C
133           WRITE(ITYP,1045)
134     1045  FORMAT(1H0,'ENTER 2 LETTER CODES FOR UP TO 3 INTERNAL STANDARDS')
135           READ(IRD,1048)ISTD
136     1048  FORMAT(3A2)
137           ID = ITYP
138       125 WRITE(ID,1000) IUNIT,IDAT,ITIM
139           WRITE(ID,1031) (ITITLE(I),I=1,39)
140      1031 FORMAT(1H ,39A2)
141           WRITE(ID,1031) (ITITLE(I),I=41,50)
142           WRITE(ID,1032) (NAME(I),I=1,6),ISTD
143      1032 FORMAT(1H ,'OUTPUT FILE: ',6A1/1H ,'STANDARDS:'3(2X,A2))
144           WRITE(ID,1030)BANG(IUNIT),EANG(IUNIT),DANG(IUNIT),TIME(IUNIT),
145          1 MVCNT(IUNIT),LIMCNT(IUNIT),RUN
146     1030  FORMAT(1H ,'BEGIN ANGLE = ',F8.3/
147          1 1X,'END ANGLE(DEG)=',F7.3/
148          2 1X,'STEP SIZE(DEG)=',F7.3/
```

```
149           3 1X,'CNT TIME(SEC) =',F7.3/
150           4 1X,'MVCNT METHOD  =',I7/
151           4 1X,'MAX COUNTS    =',I7//
152           5 1X,'RUN TIME(HRS) =',F7.3)
153            IF(ID .EQ. IPRT) GOTO 140
154            WRITE(ITYP,1034)
155       1034 FORMAT(1H0,'CORRECTIONS? Y/N' )
156            READ(IRD,1040)IY
157       1040  FORMAT(A1)
158            IF(IY .NE. 'N')GOTO 100
159            WRITE(ITYP,1042)
160       1042  FORMAT(1H ,'PRINT INPUT? Y/N')
161            READ(IRD,1040) IY
162            IF(IY .NE. 'Y') GOTO 140
163            ID = IPRT
164            GOTO 125
165       C
166       C  SET THE NUMBER OF PEAK AND BACKGROUND POINTS
167       C
168         140 NBG(IUNIT) = 0
169            NPK(IUNIT) = IFIX2(DELTA + 1.5 )
170       C
171       C  COMPUTE THE NUMBER OF CYLINDERS REQUIRD TO STORE THE FILE
172       C  ON THE XEBEC DISK. 16 DATA POINTS PER 128 CHARACTER RECORD AND 96
173       C  RECORDS PER CYLINDER (16*96=1536).
174       C
175            ISIZE = IFIX2(DELTA/1536.0 + 1.0)
176       C
177       C  NOW ALLOCATE THE OUTPUT DATA FILE
178       C
179            ID = IDISK(IUNIT,2)
180            CALL ALCT(IUNIT,ID,ISIZE)
181       C
182       C  CALCULATE THE HOUR THE RUN IS GOING TO END
183       C
184            CALL ENTIM(IUNIT,RUN)
185       C
186       C  WRITE THE RUN PARAMETERS TO THE OUTPUT FILE
187       C
188            WRITE(ID,1050)EANG(IUNIT),EANG(IUNIT),DANG(IUNIT),
189           1 RATE(IUNIT), MVCNT(IUNIT),LIMCNT(IUNIT), ISTD,TIME(IUNIT)
190       1050  FORMAT(F7.3,7X,2F7.3,F4.1,I1,12X,I5,3A2,48X,F7.2)
191       C
192       C  NUMBER OF RECORDS WRITTEN ON OUTPUT FILE
193       C
194            NREC(IUNIT) = 6
195       C
196       C  SET STATUS TO QUAL
197       C
198            IRUN(IUNIT) = 2
199            ITOT(IUNIT) = 0
200       C
201       C  NOW CHECK IF SLEW HAS COMPLETED - IF NOT HAVE ENSLEW WAIT FOR IT
202       C
203            MODE(IUNIT,7) = 0
204       C
205       C TEST IF SLEW REQUEST RESULTED IN ZERO STEPS. IF SO CALL ENSLEW TO
206       C MOVE UP 10 STEPS TO REMOVE GONIOMETER BACKLASH
207       C
208            IF(IZRST(IUNIT) .EQ. 1) MODE(IUNIT,7) = 2
209       C
210            CALL ENSLEW(IUNIT,IERROR)
211            IF(IERROR.EQ.1)GO TO 999
212       C
213       C  NOW WE BEGIN DATA COLLECTION BY CALLING STEP SCAN ON COMPLETING
214       C  IT WILL PUT THE FINIS ADRESS ON THE STACK
215       C
216            ITRANS(IUNIT,1) = 4
217       C
218       C  PUT MODE FOR CALL TO STEPSCAN
219       C
220            MODE(IUNIT,4) = 1
221            IF(MVCNT(IUNIT).EQ.1) MODE(IUNIT,4) = 3
222       999   RETURN
223            END
```

```
C-------------------------------------------------------------------
      SUBROUTINE QUANT(IUNIT)
C-------------------------------------------------------------------
C
C  THIS GENERAL DATA COLLECTION SUBROUTINE ALLOWS FOR FIVE MODES
C  OF DATA COLLECTION.  THREE ARE FOR QUANTITATIVE ANALYSIS AND TWO ARE
C  FOR SELECTED AREA DATA COLLECTION. THEY ARE:
C
C  INTERNAL STANDARD METHOD (IFLAG(IUNIT,3) = 1)
C  INTENSITY RATIO METHOD   (IFLAG(IUNIT,3) = 2)
C  SPIKING METHOD           (IFLAG(IUNIT,3) = 3)
C  PEAK AND INTEGRATED INT. (IFLAG(IUNIT,3) = 4)
C  CALIBRATION              (IFLAG(IUNIT,3) = 5)
C
C  NOTE: INTENSITY AND CALIBRATION DATA COLLECTION IS IDENTICAL. THE
C  DIFFERENT IFLAG VALUES ARE ONLY FOR PURPOSES OF ANALYSIS OF THE
C  RAW DATA.  GENERALLY THE PEAK AND INTEGRATED INTENSITY MODE WILL BE
C  USED FOR DETERMINATION OF RELATIVE INTENSITIES
C
C  MODE(IUNIT,2)  = 1 BEGIN OF QUANT RUN,SET UP FILES
C                 = 2 RETURN FROM INITIAL SLEW ,SETUP FILES
C                 = 3 RETURN FROM STPSCN
C                 = 4 RETURN FROM SLEW-INTRUP-ENSLEW-ESTIM
C
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*4 NBS, INTR
C
      COMMON /ANG/      BANG(2),       CANG(2),       DANG(2),
     1 EANG(2),         IRATE(2),      PANG(2),       RATE(2),
     2 TANG(2),         IZRST(2),      NPTPK(2)
C
      COMMON /BUFF/     IALPH(66),     NAME(15)
C
      COMMON /CNTL/     IACT(2),       IDBUG(2),      INTR,
     1 IRUN(2),         ITRANS(2,20),  MODE(2,20),    MVCNT(2)
C
      COMMON /COUN/     ICNT(2,16),    IPT(2),        ITOT(2),
     1 LIMCNT(2),       ISAVE(2)
C
      COMMON /IO/       NBS,           IDT,           ICLOCK,
     1 ISPEC,           IRD,           IPRT,          ITYP,
     2 IDISK(2,2),      IFIN,          IFOUT
C
      COMMON /PROB/     IDAT(3),       IFLAG(2,10),   ITIM(3),
     1 ITITLE(50)
C
      COMMON /RNDT/     NREC(2),       NMFIL(2,6),    ITR(2),
     1       IENRN(2,4)
C
      COMMON /QUAN/     CONC,          FREL,          IOVLP,
     1 KODE,            LOOP(2),       LPK(2),        NUM(2,20),
     2 SIGSTD,          STD,           STDREL,        IREPT(2)
C
      COMMON /TIM/      BKGTIM(2),     FIXERR(2),     IPK(2),
     1 NBG(2),          NPK(2),        TIME(2),       TIMLIM(2)
C
      DIMENSION NAM(3), JKL(3)
C
      DATA IBL/'  '/
C
C*********************************************************************
C
C  POP QUANT ADDRESS OFF OF THE LIFO STACK
C
      IERROR = 0
      CALL POP(IUNIT,IERROR)
C
C  BRANCH TO APPROPRIATE SECTION BASED ON THE MODE FLAG
C
      IGOTO = MODE(IUNIT,2)
      GO TO(100,110,300,400),IGOTO
C
C  THIS IS THE BEGINNING OF A QUANT RUN - SET UP THE FILES
C  CLEAR TOTAL RUN TIME
```

```
      C
  100   IRUN(IUNIT) = 1
        RUN = 0.0
        WRITE(ITYP,1000)IUNIT
 1000   FORMAT(1H0,'QUANT ANALYSIS DATA COLLECTION ON UNIT #',I1)
      C
      C SET RUNNING FLAG
      C
        ITR(IUNIT) = 3
      C
      C  GET THE APPROXIMATE STARTING ANGLE OF THE RUN
      C
        WRITE(ITYP,1005)
 1005   FORMAT(1H0,'ENTER APPROXIMATE STARTING 2-THETA (F7.3)')
        READ(IRD,1080)EANG(IUNIT)
      C
      C  SET UP THE LIFO STACK TO BEGIN THE SLEW TO BANG
      C
        TANG(IUNIT) = EANG(IUNIT)
        MODE(IUNIT,2) = 2
        ITRANS(IUNIT,1) = 6
        ITRANS(IUNIT,2) = 2
        GO TO 999
      C
      C  GET THE NAME OF THE RUN FILE
      C
  110   ID = IDISK(IUNIT,1)
        WRITE(ITYP,1010)
 1010   FORMAT(1H0,'RUN FILE ALREADY EXIST? Y/N')
        READ(IRD,1020)IY
 1020   FORMAT(A1)
        IF(IY.EQ.'Y')GO TO 120
      C
      C  CREATE A NEW STANDARD FILE
      C
        CALL STDFIL(IUNIT)
        GO TO 130
      C
      C  ACTIVATE THE EXISTING DATA FILE
      C
  120   CALL FILNAM(2)
        ENCODE(IALPH,1050)(NAME(I),I=1,6),ID
 1050   FORMAT('AC ',6A1,',',Z1)
        CALL SYSTEM(IALPH,11)
      C
      C  READ IN THE RUN FILE
      C
  130   READ(ID,1060)(ITITLE(I),I=1,50),IDAT,ITIM,(IFLAG(IUNIT,I),I=1,10)
 1060   FORMAT(56A2,10I1)
        ISIZE = 1
        LINES = 0
      C
      C  READ THE INSTRUMENT SPECIFICATION PARAMETERS
      C
  135   DO 137 I=1,4
        READ(ID,1072)(IALPH(J),J=1,64)
 1072   FORMAT(64A2)
        ISIZE = ISIZE + 1
  137   CONTINUE
      C
      C  SKIP THE EMPTY CHEMICAL COMPOSITION OF THE UNKNOWN RECORD
      C
  138   READ(ID,1072)
      C
      C  INPUT THE NUMBER OF REPETIONS AND MOUNTINGS FOR THIS SAMPLE
      C
        WRITE(ITYP,1040)
 1040   FORMAT(1H0,'ENTER: # OF REPETIONS, # OF MOUNTINGS (2I1)')
        READ(IRD,1045)IFLAG(IUNIT,5),IFLAG(IUNIT,6)
 1045   FORMAT(2I1)
        IF(IFLAG(IUNIT,5).EQ.0) IFLAG(IUNIT,5)=1
        IF(IFLAG(IUNIT,6).EQ.0) IFLAG(IUNIT,6)=1
      C
      C  WHICH LINES ON THE RUN FILE ARE TO BE DETERMINED?
```

```
C  (ALL LINES TO BE MEASURED IF I-REL OR CALIBRATION RUN
C
      IF(IFLAG(IUNIT,3).GT.3) GOTO 140
      WRITE(ITYP,1100)
1100  FORMAT(1H0,'ENTER THE NUMBER OF EACH LINE TO BE MEASURED'/
     1 1H ,'DEFAULT=ALL LINES (20I2)')
      READ(IRD,1110)(NUM(IUNIT,I),I=1,20)
1110  FORMAT(20I2)
      IF(NUM(IUNIT,1).GT.0)GO TO 140
      DO 139 I=1,20
139   NUM(IUNIT,I) = I
C
C  NOW READ THE LINE INFORMATION
C
140   CONTINUE
      READ(IO,1080,END=180)BANG(IUNIT),PANG(IUNIT),EANG(IUNIT),
     1OANG(IUNIT),RATE(IUNIT),MVCNT(IUNIT),TIMLIM(IUNIT),FIXERR(IUNIT),
     2 LIMCNT(IUNIT),NAM,JKL,KODE,IOVLP,IPK(IUNIT),NPK(IUNIT),
     3 NBG(IUNIT),FREL,STDREL,STD,SIGSTD
1080  FORMAT(4F7.3,F4.1,I1,F7.3,F5.2,I5,6A2,3I1,2I2,F6.1,
     1 F4.1,F9.4,F8.4,F8.3,2F7.2)
      LINES = LINES + 1
C
C  TEST IF THIS LINE IS TO BE MEASURED.  IF SO THEN ADD TO # RECORDS
C  AND ESTIMATED RUN TIME
C
      DO 144 I=1,20
      IF(LINES.EQ.NUM(IUNIT,I)) GOTO 148
144   CONTINUE
C  NO MATCH - DO NOT MEASURE THIS ONE
      GOTO 170
C
C  COMPUTE THE NUMBER OF OUTPUT RECORDS NEEDED FOR THIS PEAK
C
148   IF(IPK(IUNIT).EQ.0)GO TO 150
C
C  THIS IS A PEAK COUNTED LINE
C
      IREC = NPK(IUNIT) + NBG(IUNIT)
C
C  IF A SECOND BACKGROUND WAS REQUESTED ADD ANOTHER NBG
C
      IF(EANG(IUNIT).GT.0.0)IREC = IREC + NBG(IUNIT)
      IREC = IREC/16+1
      GO TO 160
C
C  THIS PEAK WILL BE COUNTED IN INTEGRATE (STEP SCAN) MODE
C
150   IREC = (EANG(IUNIT) - BANG(IUNIT))*200. / 16. + .999
160   ISIZE = ISIZE + IREC + 2
C
C  ADD MAXIMUM RUN TIME FOR THIS LINE TO THE ESTIMATED TOTAL TIME
C
      RUN = RUN + TIMLIM(IUNIT)
C
C  NOW READ THIS LINES RECORD NUMBER 8
C
170   READ(ID,1072)(IALPH(J),J=1,64)
      GO TO 140
C
C  WE CAN NOW CALCULATE THE SIZE OF THE OUTPUT FILE AND ALLOCATE
C  IT ON THE BASIS OF 96 RECORDS PER CYLINDER
C  MULTIPLY CALCULATED NUMBER OF RECORDS BY THE NUMBER OF
C  REPEATS (IFLAG( ,5)) AND THE NUMBER OF MOUNTINGS (IFLAG( ,6))
C
180   ISIZE = IFLAG(IUNIT,5)*IFLAG(IUNIT,6)*ISIZE
      ISIZE = ISIZE/96+1
      CALL FILEUP(IUNIT,1)
C
C  ALLOCATE THE FILE AND WRITE THE FIRST 5 RECORDS
C
      CALL ALOT(IUNIT,IDISK(IUNIT,2),ISIZE)
C
C  MULTIPLY ESTIMATED RUN TIME BY THE NUMBER OF REPETIONS AND
```

```
223       C  MOUNTINGS. CONVERT FORM MINUTES TO HOURS. THEN SAVE THIS VALUE
224       C
225             RUN = IFLAG(IUNIT,5) * IFLAG(IUNIT,6) * RUN/60.0
226             CALL ENTIM(IUNIT,RUN)
227       C
228       C  IF ELEMENTAL COMPOSITION CONSTRAINTS ARE GOING TO BE USED
229       C  ENTER THE COMPOSITION OF THE UNKNOWN OR STANDARD
230       C
231             J = 0
232             IF(IFLAG(IUNIT,4).EQ.0)GO TO 190
233             I = 1
234             J = 30
235       185   WRITE(ITYP,1090)
236       1090  FORMAT(1HO,'ENTER SAMPLE COMPOSITION AS ELEMENT SYMBOL, '
237            1'WEIGHT %, SIGMA'//
238            3 'EL1.000 0.10EL2.000 0.20EL3.000 0.30EL4.000 0.40EL5.000 ',
239            4 '0.50')
240             READ(IRD,1072)(IALPH(K),K=I,J)
241             IF(J.EQ.60)GO TO 190
242             WRITE(ITYP,1095)
243       1095  FORMAT(1HO,'MORE ELEMENTS(UP TO 5) ?Y/N')
244             READ(IRD,1020)IY
245             IF(IY.NE.'Y')GO TO 190
246             I = 31
247             J = 60
248             GO TO 185
249       C
250       C  BLANK OUT THE REST OF THE IALPH ARRAY
251       C
252       190   I = J + 1
253             DO 200 J=I,64
254       200   IALPH(J) = IBL
255       C
256       C  WRITE RECORD 6 ON THE OUTPUT DATA FILE
257       C
258             ID = IDISK(IUNIT,2)
259             WRITE(ID,1072)(IALPH(J),J=1,64)
260       C
261       C  # OF RECORDS ON OUTPUT FILE
262       C
263             NREC(IUNIT) = 6
264       C
265       C  CHECK THAT SLEW IS FINISHED OR WAIT FOR IT
266       C  IF ZERO STEP SLEW WAS CALLED SET ENSLEW MODE
267       C
268       220   MODE(IUNIT,7) = 0
269             IF(IZRST(IUNIT).EQ.1) MODE(IUNIT,7) = 2
270             CALL ENSLEW(IUNIT,IERROR)
271             IF(IERROR.EQ.1)GO TO 999
272       C
273       C
274       C
275       C++++++++++++++++++++++++  BEGIN QUANT RUN ++++++++++++++++++++++
276       C
277       C
278       C  WE ARE NOW READY TO BEGIN THE QUANTITATIVE ANALYSIS
279       C  WE START BY INITIATING A LOOP OF IFLAG(IUNIT,6) MOUNTINGS
280       C  AND IFLAG(IUNIT,5) REPETIONS OF THE UNKNOWN OR STANDARD
281       C  REQUEST THE MOUNTING OF THE SAMPLE
282       C
283             LOOP(IUNIT) = 0
284       230   LOOP(IUNIT) = LOOP(IUNIT) + 1
285             IF(LOOP(IUNIT).GT.IFLAG(IUNIT,6))GO TO 500
286             WRITE(ITYP,1120)LOOP(IUNIT),IUNIT
287       1120  FORMAT(1HO,'PLACE SAMPLE # ',I1,' ON UNIT ',I1,
288            1 1X,'WHEN READY HIT RETURN. ENTER 1 TO STOP')
289             READ(IRD,1045)ISKIP
290             IF(ISKIP.EQ.1)GO TO 500
291       C
292       C  REQUEST WT% OF REFERENCE OR SPIKING PHASE
293       C
294             CONC = 100.0
295             I = IFLAG(IUNIT,3)
296             GO TO(232,238,232,238,238), I
```

```
      C  INTERNAL STANDARD OR SPIKING METHOD - ENTER CONCENTRATION OF PHASE
  232    WRITE(ITYP,234)
  234    FORMAT(1H0,'ENTER WT% OF REFERENCE OR SPIKING PHASE(F7.3)')
         READ(IRD,1080) CONC
      C
      C INITIALIZE THE REPETION COUNTER
      C
  238    IREPT(IUNIT) = 0
      C
      C  INITIALIZE THE POINTER OF THE NEXT LINE TO BE ANALYSED
      C
  240    LPK(IUNIT) = 0
         IREPT(IUNIT) = IREPT(IUNIT) + 1
         IF(IREPT(IUNIT).GT.IFLAG(IUNIT,5)) GOTO 230
      C
      C  IN RE-RUNNING A SAMPLE WE REWIND THE RUN FILE AND RE EXECUTE IT
      C
         ID = IDISK(IUNIT,1)
         REWIND ID
      C
      C  SKIP OFF THE FIRST 6 RECORDS
      C
         DO 250 I=1,6
         READ(ID,1072)
  250    CONTINUE
      C
      C  WE ARE NOW READY TO ANALYSE THE NEXT LINE ON THE RUN FILE
      C  THIS IS THE ENTRY POINT ON RETURNING FROM STPSCN OR PKSCN
      C
  300    LPK(IUNIT) = LPK(IUNIT) + 1
      C
      C  ZERO THE COUNTER OF THE POINT NUMBER BEING MEASURED
      C
         ITOT(IUNIT) = 0
      C
      C  READ A LINE FROM THE RUN FILE. ON END OF FILE BRANCH TO NEXT REPETION
      C
         ID = IDISK(IUNIT,1)
         READ(ID,1080,END=240)BANG(IUNIT),PANG(IUNIT),EANG(IUNIT),
        1DANG(IUNIT),RATE(IUNIT),MVCNT(IUNIT),TIMLIM(IUNIT),FIXERR(IUNIT),
        2 LIMCNT(IUNIT),NAM,JKL,KODE,IOVLP,IPK(IUNIT),NPK(IUNIT),
        3 NBG(IUNIT),FREL,STDREL,STD,SIGSTD
      C
      C  READ THE ASSOCIATED RECORD NUMBER 8
      C
         READ(ID,1072)(IALPH(J),J=1,64)
      C
      C  IS THIS LINE TO BE MEASURED?
      C  (IF I-REL OR CALIBRATION RUN THE MEASURE ALL LINES)
      C
         IF(IFLAG(IUNIT,3).GT.3) GOTO 330
         DO 320 I=1,20
         IF(LPK(IUNIT).EQ.NUM(IUNIT,I))GO TO 330
  320    CONTINUE
      C
      C  NO - SKIP TO NEXT LINE
      C
         GO TO 300
      C
      C  YES IT IS
      C  1. SLEW TO BANG
      C  2. WAIT FOR INTRUP
      C  3. CALL ENSLEW
      C  4. CALL ESTIM TO DETERMINE THE OPTIMUM AMOUNT OF TIME TO SPEND
      C     COUNTING THE PEAK AND BACKGROUND REGIONS
      C  5. RETURN TO QUANT
      C
  330    TANG(IUNIT) = BANG(IUNIT)
         ITRANS(IUNIT,1)=6
         ITRANS(IUNIT,2)=12
         ITRANS(IUNIT,3)=7
         MODE(IUNIT,7) = 1
         ITRANS(IUNIT,4)=13
         MODE(IUNIT,13)=0
```

```
371             ITRANS(IUNIT,5)=2
372             MODE(IUNIT,2) = 4
373             GOTO 999
374       C
375       C    WRITE LINE COLLECTION PARAMETERS TO OUTPUT FILE
376       C
377       400   ID = IDISK(IUNIT,2)
378             WRITE(ID,1080)BANG(IUNIT),PANG(IUNIT),EANG(IUNIT),
379            1DANG(IUNIT),RATE(IUNIT),MVCNT(IUNIT),TIMLIM(IUNIT),FIXERR(IUNIT),
380            2 LIMCNT(IUNIT),NAM,JKL,KODE,IOVLP,IPK(IUNIT),NPK(IUNIT),
381            3NBG(IUNIT),FREL,STDREL,STD,SIGSTD,CONC,TIME(IUNIT),BKGTIM(IUNIT)
382             WRITE(ID,1072)(IALPH(J),J=1,64)
383       C
384       C # OF OUTPUT RECORDS
385       C
386             NREC(IUNIT)=NREC(IUNIT)+2
387       C
388       C  RESET IPT(IUNIT) AS IT WAS USED BY ESTIM
389       C
390             IPT(IUNIT) = 0
391       C
392       C  TO BEGIN DATA COLLECTION WE NOW NEED ONLY TO CALL STPSCN OR PKSCN
393       C
394       C  TRANSFER SEQUENCE: , STPSCN OR PKSCN AND QUANT
395       C
396             ITRANS(IUNIT,2) = 2
397             MODE(IUNIT,2) = 3
398       C
399       C  BRANCH BASED ON COUNT MODE
400       C
401             IF(IPK(IUNIT).EQ.1)GO TO 450
402       C
403       C  SET TO CALL STPSCN FOR INTEGRATED COUNTS
404       C
405             MODE(IUNIT,4) = 3
406             ITRANS(IUNIT,1) = 4
407             GO TO 999
408       C
409       C SET TO CALL PKSCN FOR PEAK COUNTING
410       C
411       450   MODE(IUNIT,5) = 1
412             ITRANS(IUNIT,1) = 5
413             GO TO 999
414       C
415       C  END OF QUANT RUN - CALL FINIS
416       C
417       500   MODE(IUNIT,10) = 2
418             MODE(IUNIT,2) = 0
419             ITRANS(IUNIT,1) = 10
420       999   RETURN
421             END
```

```
C-------------------------------------------------------------------
      SUBROUTINE SETANG(IUNIT)
C-------------------------------------------------------------------
C
C  THIS SUBROUTINE SET THE CURRENT ANGLE (IN CANG) INTO
C  THE INTERFACE TRACKING COUNTER
C  AND ZEROS THE INTERFACE SCALER
C
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*4 NBS, INTR, NBSADD, NBSSUB, MASK, MASKLS, ICNTL
C
      COMMON /ANG/      BANG(2),       CANG(2),       DANG(2),
     1 EANG(2),         IRATE(2),      PANG(2),       RATE(2),
     2 TANG(2),         IZRST(2)
C
      COMMON /CNTL/     IACT(2),       IDBUG(2),      INTR,
     1 IRUN(2),         ITRANS(2,20),  MODE(2,20),    MVCNT(2)
C
      COMMON /IO/       NBS,           IDT,           ICLOCK,
     1 ISPEC,           IRD,           IPRT,          ITYP,
     2 IDISK(2,2),      IFIN,          IFOUT
C
      DATA MASK/X'40'/, ICNTL/X'0D'/
      DATA MASKLS/X'8888'/
C
C*********************************************************************
C
C  FIRST CHECK IF THE INTERFACE IS PRESENT AND ACKNOWLEDGING REQUESTS
C
C  NBSADD = ADDRESS OF INTERFACE ON NBS BUS - A MULTIPLE OF 4
C  NBSSUB = SUBADDRESS OF ONE OF THE 4 REGISTERS ON THE NBS BUS
      NBSADD = IUNIT * 4
      NBSSUB = NBSADD + 2
C
C  CLEAR ALL INTERFACE REGISTERS
C
      CALL CLEAR(NBS,NBSADD)
      CALL CLEAR(NBS,NBSADD)
C
C  CHECK THAT BOARD IS PRESENT AND NOT ACTIVE
C
      INTR = ISTAT(NBS,NBSADD)
      IF(IAND(INTR,MASK).EQ.MASK)GO TO 100
C
C  THERE IS SOME PROBLEM WITH THE INTERFACE - PRINT WARNING
C
      WRITE(ITYP,1000)IUNIT,INTR
1000  FORMAT(1H0,'SETANG INTERFACE #',I1,' NOT READY-ISTAT=',Z4)
      IACT(IUNIT)=-1
      GOTO 999
C
C  TEST IF A LIMIT SWITHCH IS ACTIVATED
C
100   IF(IAND(INTR,MASKLS).EQ.0) GOTO 120
      WRITE(ITYP,1006) IUNIT,INTR
1006  FORMAT(1H0,'UNIT',I2,' LIMIT SWITCH SET.  ISTATUS=',Z4)
      IACT(IUNIT) = -1
      GOTO 999
C
C  INITIALIZE THE INTERFACE TO A NON ZERO ADDRESS
C
120   INTR = ISTAT(NBS,NBSSUB)
C
C  INTERFACE RESPONDS - RESET TO THE FIRST ZERO REGISTER
C
      INTR = ISTAT(NBS,NBSADD)
C
C  NOW LOAD THE CONTROL WORD INTO THE FIRST OF THE TWO
C  ZERO REGISTERS OF THE INTERFACE
C
      CALL OUTBUS(NBS,ICNTL)
C
C  USE CKANG TO CHECK LEGALITY OF CANG AND TO COMPUTE THE INTEGER
C  TRACKING CODE (IE 200 * CANG).
```

```
75      C
76            IERROR = -1
77            TANG(IUNIT) = CANG(IUNIT)
78            CALL CKANG(IUNIT,NSTEP,IERROR)
79            IF(IERROR.LE.0)GO TO 200
80            WRITE(ITYP,1010)IUNIT
81      1010  FORMAT(1H0,'USE OPTION 4 TO RESET UNIT#',I2)
82            IACT(IUNIT) = -1
83            GO TO 999
84      C  THE CURRENT ANGLE AS ENTERED IS OK - SET IT INTO TRACKING COUNTER
85      C
86      200   INTR = ISTAT(NBS,NBSSUB)
87            INTR = NSTEP
88            CALL OUTBUS(NBS,INTR)
89      C
90      C  OPERATION COMPLETED. SET ACTIVITY FLAG TO INACTIVE
91      C
92            IACT(IUNIT)=0
93      C
94      C  CLEAR THE INTERFACE COUNTER REGISTER
95      C
96            NBSSUB = NBSSUB + 1
97            CALL CLEAR(NBS,NBSSUB)
98      999   RETURN
99            END
```

```
C-------------------------------------------------------------------
      SUBROUTINE STPSCN(IUNIT,NSTEP)
C-------------------------------------------------------------------
C
C  SUBROUTINE STEP SCAN COLLECTS FIXED TIME COUNT DATA STARTING
C  AT BANG AND ENDING AT EANG STEPPING BY DANG BETWEEN POINTS.
C
C  QUANTITATIVE ANALYSIS INTEGRATED SCANS ARE SIMPLY THE SUM OF
C  THE INTENSITIES COLLECTED AT EACH POINT IN THE PROFILE
C  DEFINED BY BANG AND EANG
C
C  AFTER EACH SET OF 16 DATA POINTS ARE COLLECTED THEY ARE WRITTEN
C  TO THE OUTPUT DISK FILE.
C
C  WHEN EANG IS REACHED FINIS IS CALLED TO SAY BYBY
C
C
C
C
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*4 NBS, INTR
C
      COMMON /ANG/     BANG(2),       CANG(2),       DANG(2),
     1 EANG(2),        IRATE(2),      PANG(2),       RATE(2),
     2 TANG(2),        IZRST(2),      NPTPK(2)
C
      COMMON /CNTL/    IACT(2),       IDBUG(2),      INTR,
     1 IRUN(2),        ITRANS(2,20),  MODE(2,20),    MVCNT(2)
C
      COMMON /COUN/    ICNT(2,16),    IPT(2),        ITOT(2),
     1 LIMCNT(2),      ISAVE(2)
C
      COMMON /IO/      NBS,           IDT,           ICLOCK,
     1 ISPEC,          IRD,           IPRT,          ITYP,
     2 IDISK(2,2),     IFIN,          IFOUT
C
      COMMON /RNDT/    NREC(2),       NMFIL(2,6),    ITR(2),
     1       IENRN(2,4)
C
      COMMON /TIM/     BKGTIM(2),     FIXERR(2),     IPK(2),
     1 NBG(2),         NPK(2),        TIME(2),       TIMLIM(2)
C
C*********************************************************************
C
C  **FOR MVCNT=0**
C
C  MODE(IUNIT,4) = 1 FOR BGMOVE COMPLETED. CALL ENMOVE THEN BGCNT
C
C                = 2 COUNTING COMPLETED SO CALL ENDCNT THEN BGMOVE
C
C                = 3  AT BANG, BEGIN WITH COUNTING. DO NOT CALL
C                     ENMOVE.
C
C  **FOR MVCNT=1**
C
C                = 3 FIRST CALL TO SIMULTANEOUS COUNTING AND STEPPING
C                    SET UP CONTROL WORD, NSTEP AND IRATE BY CALLING
C                    BGMOVE
C
C                = 4 SUBSEQUENT CALLS FOR SIMULTANEOUS COUNTING AND STEPPING
C                    CALL ENMOVE TO CHECK COMPLETION OF MOVE, TO READ COUNTER
C                    AND INITIATE THE NEXT MOVE
C
C*********************************************************************
C
C  POP THE STPSCN ADDRESS OFF THE TRANSFER TABLE
C
      IERROR = 0
      CALL POP(IUNIT,IERROR)
C
C  MAKE END ANGLE A MULTIPLE OF 0.005 DEG FOR CHECKING AGAINST TANG
C
      CALL FIXANG(EANG(IUNIT))
```

```
C
C  BRANCH BASED ON THE VALUE OF THE MODE FLAG
C
      IGOTO = MODE(IUNIT,4)
      GO TO(300,400,310,300),IGOTO
C
C  THIS IS THE NORMAL STEP AND COUNT SECTION WHERE MOST OF
C  THE ENTIRE DATA COLLECTION PROCEDURE OCCURS.
C
C  IF THIS IS THE FIRST ENTRY GO STRAIGHT TO BEGIN COUNTING
C
300   IF(ITOT(IUNIT).EQ.0)GO TO 310
C
C  CHECK THAT THE PREVIOUS MOVE OPERATION COMPLETED NORMALLY
C
      MODE(IUNIT,8) = 1
      CALL ENMOVE(IUNIT,IERROR)
      IF(IERROR.EQ.1)GO TO 999
C
C  IF PATTERN COLLECTION SET MODE( ,9)=1 TO FLAG USE OF TIME(IUNIT)
C  IN CALCULATION OF COUNTING TIME(MVCNT=0) OR DRIVE RATE(MVCNT=1)
C
310   MODE(IUNIT,9) = 1
      IF(IRUN(IUNIT).EQ.2) GOTO 316
C
C  QUANTITATIVE ANALYSIS RUN.
C  BEGIN COUNTING THE NEXT DATA POINT FOR TIME(IUNIT) SECONDS
C  OR BKGTIM(IUNIT) SECONDS IF IT IS A BACKGROUND POINT
C  IF NEXT POINT IS AT PEAK SET MODE(IUNIT,9) TO 1
C  IF NEXT POINT IS ON BACKGROUND SET MODE(IUNIT,9) = 2
C
      IF(ITOT(IUNIT).LT.NBG(IUNIT).OR.ITOT(IUNIT).GE.
     1 NPTPK(IUNIT) + NBG(IUNIT))MODE(IUNIT,9) = 2
C
C  IF SIMULTANEOUS STEP AND COUNT FLAG IS SET THEN SET FLAG TO SAVE
C  COUNTS AT END OF ENMOVE AND SET ENMOVE MODE TO INITIATE THE NEXT MOVE
C
316   IF(MVCNT(IUNIT).EQ.0) GOTO 320
      ISAVE(IUNIT) = 1
      MODE(IUNIT,4) = 4
      GOTO 450
C
C  CALL SEQUENCE: BGCNT, INTRUP, STPSCN
C
320   CALL PUSH(IUNIT,4)
      MODE(IUNIT,4) = 2
      CALL PUSH(IUNIT,12)
      CALL PUSH(IUNIT,9)
      GO TO 999
C
C  COUNTING IS COMPLETE - CHECK FOR NORMAL TERMINATION
C  AND READ OUT THE COUNTER
C
400   MODE(IUNIT,4) = 1
      MODE(IUNIT,6) = 1
      ITOT(IUNIT) = ITOT(IUNIT) + 1
      IPT(IUNIT)  = IPT(IUNIT)  + 1
      CALL ENDCNT(IUNIT,IERROR)
      IF(IERROR.EQ.1)GO TO 999
C
C  IF THAT WAS THE 16TH POINT WRITE THEM TO OUTPUT FILE
C
450   IF(IPT(IUNIT).LT.16)GO TO 458
      CALL WRFILE(IUNIT)
C
C  INCREMENT TARGET ANGLE AND TEST IF WE ARE FINISHED
C
458   TANG(IUNIT) = CANG(IUNIT) + DANG(IUNIT)
      CALL CKANG(IUNIT,NSTEP,IERROR)
      IF(IERROR.EQ.1)GO TO 459
      IF((TANG(IUNIT)-EANG(IUNIT)).GT.+0.0005) GOTO 500
      GOTO 460
C
C  CALL FINIS - ERROR DETECTED IN CKANG
```

```
149       C
150       459   MODE(IUNIT,10) = 1
151             ITRANS(IUNIT,1) = 10
152             GOTO 999
153       C
154       C  NOT DONE NOR FOUND ERROR IN TARGET ANGLE.  SET STACK POINTER
155       C  TO RETURN TO STEP SCAN FOR NEXT POINT
156       C
157       460   CALL PUSH(IUNIT,4)
158       C
159       C  SET UP IRATE(IUNIT)
160       C
161             IF(MVCNT(IUNIT).EQ.1) GOTO 470
162       C  INDEPENDENT STEP AND COUNT MOVE AT RATE(IUNIT)
163             IRATE(IUNIT) = IFIX2(30.0/RATE(IUNIT))
164             GOTO 480
165       C  SIMULTANEOUS STEP AND COUNT MOVE AT DANG/TIME
166       C
167       470   TEMPRY = TIME(IUNIT)
168             IF(MODE(IUNIT,9) .EQ. 2) TEMPRY = BKGTIM(IUNIT)
169       C  IRATE = 30.0 DIVIDED BY RATE IN DEGREES PER MINUTE (TEMPRY IS IN SEC.)
170             IRATE(IUNIT) = IFIX2(0.5*TEMPRY/DANG(IUNIT) + 0.5)
171             ITOT(IUNIT) = ITOT(IUNIT) + 1
172             IPT(IUNIT)  = IPT(IUNIT)  + 1
173       C
174       C  CALL SEQUENCE: BGMOVE, INTRUP
175       C
176       480   MODE(IUNIT,8) = 1
177             CALL PUSH(IUNIT,12)
178             CALL PUSH(IUNIT,8)
179             GOTO 999
180       C
181       C  STEPSCAN OF BANG TO TANG IS COMPLETED
182       C  IF CALLED BY PEAK SCAN RETURN TO IT VIA ITRANS FLAG
183       C
184       500   IF(IPK(IUNIT).EQ.1) GOTO 999
185       C
186       C  WRITE BUFFERED DATA TO COMPLETE DATA RECORD
187       C
188             CALL WRFILE(IUNIT)
189       C
190       C  IF PATTERN COLLECTION RUN THEN SET ITRANS CODE TO CALL FINIS NEXT
191       C
192             IF(IRUN(IUNIT).NE.2) GOTO 999
193       C
194       C  CALL FINIS - PATTERN COLLECTION RUN HAS COMPLETED
195       C
196             CALL PUSH(IUNIT,10)
197             MODE(IUNIT,10) = 2
198       C
199       C  STPSCN HAS COMPLETED
200       C
201       999   RETURN
202             END
```

```
      C-------------------------------------------------------------------
            SUBROUTINE STDFIL(IUNIT)
      C-------------------------------------------------------------------
      C
      C
      C  THIS SUBROUTINE CREATES A STANDARD FILE FOR THE COLLECTION OF
      C  QUANTITATIVE ANALYSIS DATA
      C
      C  THE FORMAT OF THE STANDARD DATA FILE IS AS FOLLOWS:
      C
      C  RECORD 1: COL 1 -80 TITLE OF DATA FILE
      C             COL 81-100 OPERATORS NAME
      C             COL 101-106 DATE OF FILE CREATION
      C             COL 113-122 IFLAG - RUN OPTION SELECTION
      C
      C  RECORD 2-5: COL 1-128 INSTRUMENT PARAMETERS
      C
      C  RECORD 6: ELEMENTAL ANALYSIS INFORMATION
      C
      C  RECORD 7: ONE FOR N LINES TO BE ANALYSED (ONE RECORD PER PROFILE)
      C  COL 1  - 7  STARTING ANGLE OF PROFILE OR LOW ANGLE BACKGROUND
      C  COL 8  -14  2-THETA OF PEAK MAXIMUM
      C  COL 15 -21  ENDING ANGLE OF PROFILE OR HIGH ANGLE BACKGROUND
      C  COL 22 -28  STEP WIDTH (NOT USED FOR PEAK COUNTING)
      C  COL 29 -32  RATE OF MOTION BETWEEN DATA POINTS (DEGREES/MIN)
      C  COL 33 -40  TIMLIM THE MAX TIME TO COUNT A LINE IN MINUTES
      C  COL 41 -45  FIXERR THE PERCENT ERROR TO COUNT TO.  IF ZERO
      C              A FIXED TIME COUNT WILL BE PERFORMED
      C  COL 46 -50  LIMCNT THE MAX NUMBER OF COUNTS TO COLLECT AT A POINT
      C  COL 51 -56  THE NAME OF THE PHASE
      C  COL 57 -62  THE H K AND L OF THE LINE (3I2)
      C  COL 63      KODE DESCRIBING THE TYPE OF LINE BEING ENTERED
      C                 = 1 A LINE BELONGING TO THE UNKNOWN PHASE BEING ANALYSED
      C                 = 2 A LINE OF THE INTERNAL STANDARD MATERIAL
      C                 = 3 A REFERENCE LINE FROM THE SAMPLE HOLDER
      C                 = 4 AN AMORPHOUS HALO LINE. NO BACKGROUND POINTS ARE
      C                     TO BE MEASURED.
      C  COL 64      IOVLP THE NUMBER OF OTHER LINES OVERLAPPING THIS ONE
      C  COL 65      IPK = 0 - STEP SCAN THE PEAK, = 1 - PEAK COUNT MODE
      C  COL 66 -67  NPK THE NUMBER OF PEAK POINTS TO MEASURE. IN INTEGRATE
      C              SCAN MODE THIS IS NUMBER TO AVERAGE AT MAX FOR PEAK I.
      C  COL 68 -69  NBG THE NUMBER OF BACKGROUND POINTS ON EACH SIDE OF PEAK
      C  COL 70 -75  RELATIVE INTENSITY OF THE LINE
      C  COL 76 -79  STANDARD DEVIATION OF THE RELATIVE INTENSITY
      C  COL 80 -88  STANDARD INTENSITY VALUE (I/ISTD OR IPURE OR I/IREF)
      C  COL 89 -96  SIGMA OF STANDARD INTENSITY
      C  COL 97 -104 CONCENTRATION OF ADDED PHASE FOR SPIKING METHOD
      C                    OR OF REFERENCE PHASE FOR INTERNAL STD METHOD
      C
      C  RECORD 8: LINE OVERLAP INFORMATION AND CHEMICAL COMPOSITION
      C             OF STANDARD LINES WHEN NEEDED
      C
            IMPLICIT INTEGER*2 (I-N)
            INTEGER*4 NBS, INTR
      C
            COMMON /ANG/      BANG(2),       CANG(2),       DANG(2),
           1 EANG(2),         IRATE(2),      PANG(2),       RATE(2),
           2 TANG(2),         IZRST(2)
      C
            COMMON /BUFF/     IALPH(66),     NAME(15)
      C
            COMMON /CNTL/     IACT(2),       IDBUG(2),      INTR,
           1 IRUN(2),         ITRANS(2,20),  MODE(2,20),    MVCNT(2)
      C
            COMMON /COUN/     ICNT(2,16),    IPT(2),        ITOT(2),
           1 LIMCNT(2),       ISAVE(2)
      C
            COMMON /IO/       NBS,           IDT,           ICLOCK,
           1 ISPEC,           IRD,           IPRT,          ITYP,
           2 IDISK(2,2),      IFIN,          IFOUT
      C
            COMMON /PROB/     IDAT(3),       IFLAG(2,10),   ITIM(3),
           1 ITITLE(50)
      C
            COMMON /QUAN/     CONC,          FREL,          IOVLP,
```

```
     1 KODE,            LOOP(2),        LPK(2),         NUM(2,20),
     2 SIGSTD,          STD,            STDREL
C
      COMMON /TIM/      BKGTIM(2),      FIXERR(2),      IPK(2),
     1 NBG(2),          NPK(2),         TIME(2),        TIMLIM(2)
C
      DATA IBL/'  '/
C
C*****************************************************************************
C
C  INPUT THE NAME OF THE RUN FILE
C
      CALL FILEUP(IUNIT,2)
C
C  ESTABLISH THE NEEDED FLAG VALUES
C
      IFLAG(IUNIT,1) = 2
      IFLAG(IUNIT,2) = 2
      WRITE(ITYP,1000)
1000  FORMAT(1H0,'METHOD OF ANALYSIS ? (I1)'/
     1 1H ,'1 - INTERNAL STANDARD'/
     2 1H ,'2 - INTENSITY RATIO'/
     3 1H ,'3 - SPIKING'/
     4 1H ,'4 - REL. INT.(DEFAULT)'/
     5 1H ,'5 - PROFILE')
      READ(IRD,1010)IFLAG(IUNIT,3)
1010  FORMAT(I1)
      IF(IFLAG(IUNIT,3).EQ.0)IFLAG(IUNIT,3) = 4
      IF(IFLAG(IUNIT,3).EQ.4)IFLAG(IUNIT,2) = 3
C
C  IF SPIKING I-REL OR PROFILE ANALYSIS, THEN CHEMISTRY IS NOT USED
C
      IFLAG(IUNIT,4) = 0
      IF(IFLAG(IUNIT,3).GE.3)GO TO 100
      WRITE(ITYP,1020)
1020  FORMAT(1H0,'WILL CHEMICAL COMPOSITION BE ENTERED? Y/N')
      READ(IRD,1030)IY
1030  FORMAT(A1)
      IF(IY.EQ.'Y')IFLAG(IUNIT,4) = 1
C
C  ALLOCATE THE RUN FILE WITH 1 CYLINDER
C  AND WRITE THE FIRST 5 RECORDS
C
100   ID = IDISK(IUNIT,1)
      CALL ALOT(IUNIT,ID,1)
C
C  WRITE A DUMMY RECORD 6 FOR NOW - QUANT WILL ASK FOR CHEM
C
      WRITE(ID,1030)
C
C  ENTER INFORMATION ON EACH LINE
C
      K = 0
101   K = K + 1
102   WRITE(ITYP,1050)K
1050  FORMAT(1H0,'FOR LINE #',I4,' ENTER UNDER DEFAULT VALUES'/
     1 1H ,' 1 -6 NAME OF LINE(BLANK TERMINATES INPUT)'/
     A 1H ,' 7- 8, 9-10, 11-12 H K AND L '/
     2 1H ,'13-14 =0 INTEGRATE, =1 PEAK COUNT'/
     3 1H ,'15-16 =1 UNKNOWN, =2 STD, =3 REF, =4 AMORPH LINE'/
     4 1H ,'17-22 =X.X THE DESIRED % ERROR (0=FIXED TIME)'/
     B 1H ,'23-30 TIME(MIN) TO COUNT PK + BKG'/
     C 1H ,'31-38 REL. INT.'/
     D 1H ,'39-44 SIGMA(REL. INT.)'//
     E 1H ,'NNNNNN H K L 0 1  1.00   10.00  ###.##  #.##')
      READ(IRD,1060)(IALPH(I),I=1,6),IPK(IUNIT),KODE,
     1 FIXERR(IUNIT),TIMLIM(IUNIT),FREL,STDREL
1060  FORMAT(1X,6A2,2I2,F6.2,2F8.2,F6.2)
C
C  ARE WE FINISHED?
C
      IF(IALPH(1).EQ.'  ')GO TO 300
C
C  SET DEFAULT VALUES
```

```
      C
            IF(KODE.EQ.0) KODE=1
            IF(FIXERR(IUNIT).LT.0.01) FIXERR(IUNIT)=1.0
            IF(TIMLIM(IUNIT).LT.0.01) TIMLIM(IUNIT)  = 10.0
      C
      C  ECHO INPUT
      C
            WRITE(ITYP,1060) (IALPH(I),I=1,6),IPK(IUNIT),KODE,
           * FIXERR(IUNIT),TIMLIM(IUNIT),FREL,STDREL
            WRITE(ITYP,1070)
      1070  FORMAT(1H ,'OK? TYPE Y/N')
            READ(IRD,1080) IOK
      1080  FORMAT(A1)
            IF(ICK.NE.'Y') GOTO 102
      C
      C  REQUEST ANGLE CONTROL INFORMATION FOR THE CURRENT LINE
      C
      104   WRITE(ITYP,1150)
      1150  FORMAT(1H0,'ENTER THE FOLLOWING UNDER THE DEFAULT VALUES:'/
           1 1H ,'COL 1 -10 BACKGROUND LOW 2-THETA'/
           2 1H ,'COL 11-20 PEAK 2-THETA'/
           3 1H ,'COL 21-30 BACKGROUND HIGH 2-THETA'/
           4 1H ,'COL 31-40 STEP WIDTH'/
           5 1H ,'COL 41-46 MAX. COUNTS'/
           6 1H ,'COL 47-50 # OF PEAK POINTS'/
           7 1H ,'COL 51-55 # OF BACKGROUND POINTS'/
           8 1H ,'COL    60 MVCNT METHOD'//
           9 3X,'LLL.LLL',3X,'PPP.PPP',3X,'HHH.HHH',5X,'0.005',
           A 1X,'32700    5   10    0')
            READ(IRD,1160)BANG(IUNIT),PANG(IUNIT),EANG(IUNIT),DANG(IUNIT),
           1 LIMCNT(IUNIT),NPK(IUNIT),NBG(IUNIT),MVCNT(IUNIT)
      1160  FORMAT(4F10.3,I6,I4,2I5)
      C
      C  DEFAULT PARAMETERS
      C
            IF(DANG(IUNIT).LT.0.005)DANG(IUNIT) = .005
            IF(LIMCNT(IUNIT).EQ.0)LIMCNT(IUNIT) = 32700
            IF(NPK(IUNIT).EQ.0)NPK(IUNIT) = 5
            IF(NBG(IUNIT).EQ.0)NBG(IUNIT) = 10
            IF(KODE.EQ.4) NBG(IUNIT) = 0
      C
      C  ONLY MOTION TOWARD HIGH ANGLE WILL BE ALLOWED IN INTEGRATE MODE
      C
            IF(IPK(IUNIT).EQ.1.OR.EANG(IUNIT).GT.BANG(IUNIT))GO TO 106
            WRITE(ITYP,1116)
      1116  FORMAT(1H0,'ILLEGAL ANGLES - EANG MUST BE > BANG')
            GO TO 104
      C
      C  IF NPK IS NOT ODD FORCE IT
      C
      106   IF((NPK(IUNIT)/2)*2.EQ.NPK(IUNIT))NPK(IUNIT)=NPK(IUNIT)-1
      C
      C  ECHO INPUT PARAMETERS
      C
            WRITE(ITYP,1160) BANG(IUNIT),PANG(IUNIT),EANG(IUNIT),DANG(IUNIT),
           * LIMCNT(IUNIT),NPK(IUNIT),NBG(IUNIT),MVCNT(IUNIT)
            WRITE(ITYP,1070)
            READ(IRD,1080) IOK
            IF(IOK.NE.'Y') GOTO 104
      C
      C  READ OVERLAPPING LINE INFORMATION - NONE ALLOWED IF SPIKING
      C  I-REL OR PROFILE MEASUREMENTS
            IOVLP = 0
            IF(IFLAG(IUNIT,3).GE.3)GO TO 160
            WRITE(ITYP,1190)
      1190  FORMAT(1H0,'ENTER THE # OF LINES WHICH OVERLAP THIS LINE(MAX=2)')
            READ(IRD,1010)IOVLP
      C
      C  WE NOW HAVE ALL THE PARAMETERS NEEDED TO WRITE RECORD 7 OF THE RUN FILE
      C
      160   WRITE(ID,1200)BANG(IUNIT),PANG(IUNIT),EANG(IUNIT),DANG(IUNIT),
```

```
221           1 RATE(IUNIT),MVCNT(IUNIT),TIMLIM(IUNIT),FIXERR(IUNIT),
222           2 LIMCNT(IUNIT),(IALPH(I),I=1,6),KODE,IOVLP,IPK(IUNIT),NPK(IUNIT),
223           3 NBG(IUNIT),FREL,STDREL
224         1200  FORMAT(4F7.3,F4.1,I1,F7.3,F5.2,I5,6A2,3I1,2I2,F6.1,F4.1)
225         C
226         C  READ THE OVERLAP LINE INFORMATION FOR RECORD 8
227         C
228         164   DO 165 I=1,64
229         165   IALPH(I) = IBL
230               IF(IOVLP.EQ.0)GO TO 170
231               WRITE(ITYP,1210)IOVLP
232         1210  FORMAT(1H0,' FOR THE ',I1,' LINE(S) '
233              1 1H ,'OVERLAPPING THE CURRENT LINE ENTER:'/
234              2 1H ,'COL 1-6 NAME OF PHASE 2'/
235              3 1H ,'COL 7-8, 9-10, 11-12, H, K AND L OF LINE 2'/
236              4 1H ,'COL 13-18 REL. INT.'/
237              5 1H ,'COL 19-22 SIGMA(REL. INT.)'//
238              7 1H ,'PHASE2 H K L  15.0 1.0 PHASE3 H K L  10.0 1.5')
239               READ(IRD,1220)(IALPH(I),I=1,24)
240         1220  FORMAT(64A2)
241         C
242         C  ECHO INPUT PARAMETERS
243         C
244               WRITE(ITYP,1220) (IALPH(I),I=1,24)
245               WRITE(ITYP,1070)
246               READ(IRD,1080) IOK
247               IF(IOK.NE.'Y') GOTO 164
248         C
249         C  THE CHEMICAL ANALYSIS SECTION OF RECORD 8 WILL BE SUPPLIED
250         C  BY THE PROCESSING PROGRAM ON THE 1108 FROM THE INFO
251         C  CONTAINED IN RECORD 6 AS READ BY QUANT
252         C
253         170   WRITE(ID,1220)(IALPH(I),I=1,64)
254         C  RETURN TO TOP FOR NEXT LINE
255               GOTO 101
256         300   ENDFILE ID
257               REWIND ID
258               RETURN
259               END
```

```
C-------------------------------------------------------------------
      SUBROUTINE TRANS(IUNIT)
C-------------------------------------------------------------------
C
C  THIS ROUTINE TRANSFERS DATA FILES FROM THE DISK DO TO FLOPPY F0
C  PART II WILL TRANSFER DATA VIA NBS NET . NOT YET READY
C
C                    GLOSSARY
C
C     IFIN  =12 INPUT FILE
C     IFOUT  =13 OUTPUT FILE
C     NRC       # OF RECORDS READ
C     IALPH        ARRAY FOR READING AND WRITTING
C
C     IDOX  IS THE USER INPUT OPTION CODE
C           = 1 FOR TRANSFER TO FLOPPY F0
C           = 2 FOR TRANSFER VIA THE NBS NET
C           = 3 FOR TRANSFER TO PRINTER
C
C-------------------------------------------------------------------
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*4 NBS, INTR
C
      COMMON /BUFF/      IALPH(66),      NAME(15)
C
      COMMON /IO/        NBS,            IDT,          ICLOCK,
     1 ISPEC,            IRD,            IPRT,         ITYP,
     2 IDISK(2,2),       IFIN,           IFOUT
C
C-------------------------------------------------------------------
C
C  POP ADDRESS OFF THE TRANSFER TABLE
C
       IERROR=0
       CALL POP(IUNIT,IERROR)
C
C  GET FUNCTION DESIRED
C
90     WRITE(ITYP,100)
100    FORMAT(1H0,'ENTER TRANSFER OPTION: 1=TO FLOPPY'/
     1 1H ,23X,'2=VIA NBS NET'/1H ,23X,'3=TO PRINTER')
       READ(IRD,110) IDOX
110    FORMAT(I1)
       IF(IDOX .LT.1 .OR. IDOX .GT. 3)GOTO 90
       IF(IDOX.EQ.2) GOTO 500
C
C ENTER THE NAME OF THE FILE TO BE COPPIED
C
       CALL FILNAM (3)
C
C   ACTIVATE INPUT FILE
C
       ENCODE(IALPH,120) (NAME(I),I=1,6),IFIN
120    FORMAT('AC ',6A1,',',Z1)
       CALL SYSTEM(IALPH,11)
C
C IF PRINT ( IDOX = 3 ) SKIP THE REST OF TO INPUT
C
       IF( IDOX .NE. 3 ) GOTO 130
       WRITE(IPRT,124) (NAME(I),I=1,6)
124    FORMAT(1H1,57X,'**** ',6A1,' ****')
       GOTO 300
C
C NUMBER OF RECORDS
C
130    WRITE(ITYP,135)
135    FORMAT(1H ,'ENTER # OF RECORDS(I3)')
       READ(IRD,140,ERR=130)INREC
140    FORMAT(I3)
       WRITE(ITYP,145)INREC
145    FORMAT(1H ,'IS ',I3,' CORRECT? Y/N')
       READ(IRD,148) IY
148    FORMAT(A1)
       IF ( IY .EQ. 'N') GOTO 130
```

```
75      C
76      C  REQUEST MCUNTING OF FLOPPY - WAIT UNTIL IN PLACE
77      C
78      150   WRITE(ITYP,151)
79      151   FORMAT(1H0,'MCUNT FLOPPY ON F0, WHEN READY ENTER R')
80            READ(IRD,148) IY
81            IF ( IY .NE. 'R') GO TO 150
82      C
83      C  CALCULATE THE NUMBER OF CYLINDERS FOR THE FLOPPY
84      C CURRENTLY THE NBS 1108 HANDELS ONLY 80 CHARACTER RECORDS EASILY
85      C SINCE FLOPPY TRANSFER IS MOST CONVENIENT WE WILL WRITE OUR 128
86      C CHARACTER RECORD IN TWO PARTS: 80 CHARACTERS THEN 48 CHARACTERS
87      C FOR 80 BYTES RECORD LENGTH THERE ARE 48 RECORDS PER CYLINDER
88      C
89            ISIZE=2*INREC/48+1
90            CALL ALCT(IUNIT,IFOUT,ISIZE)
91      C
92      C  COPY FROM DISK TO FLOPPY
93      C
94      300   NRC=0
95      310   READ(IFIN,311,END=380,ERR=340) (IALPH(I),I=1,64)
96      311   FORMAT(64A2)
97            NRC=NRC+1
98      C
99      C  OUTPUT THE RECORD TO EITHER THE FLOPPY OR THE PRINTER
100     C
101           IF(IDOX.EQ.1) WRITE(IFOUT,320,END=370,ERR=360) (IALPH(I),I=1,64)
102     320   FORMAT(40A2)
103           IF(IDOX.EQ.3) WRITE(IPRT,311,ERR=360) (IALPH(I),I=1,64)
104           GOTO 310
105     C
106     C  ERRORS ON INPUT
107     C
108     340   WRITE(ITYP,341)
109     341   FORMAT(1H ,'READ ERROR-EXIT')
110           GOTO 390
111     C
112     C  OUTPUT ERROR
113     C
114     360   WRITE(ITYP,361)
115     361   FORMAT(1H ,'WRITE ERROR-EXIT')
116           GOTO 390
117     C
118     C  ERROR - FLOPPY IS FULL. TRANSFER CAN NOT BE COMPLETED
119     C
120     370   WRITE(ITYP,371) NRC
121     371   FORMAT(1H ,'FLOPPY OUT OF SPACE AT RECORD',I4,'EXIT')
122           GOTO 390
123     C
124     C  END OF COPY
125     C
126     380   IF(IDOX.EQ.3) GOTO 400
127           ENDFILE IFOUT
128     C
129     C  CLOSE ALL FILES ATTATCHED TO FLOPPY F0
130     C
131     390   CALL SYSTEM('CH F0',5)
132     C
133     C  PRINT COMPLETION MESSAGE
134     C
135     400   WRITE(ITYP,410) NRC
136     410   FORMAT(1H ,'TRANSFER COMPLETED:',I4,'RECORDS')
137     C
138     C  IF TRANSFER TO PRINTER THEN SEND A FINAL FORM FEED
139     C  (REQUIRED ON VERSATEC PRINTER TO GET IT OUT OF SOLUTION)
140     C
141           IF(IDOX.EQ.3) WRITE(IPRT,420)
142     420   FORMAT(1H1)
143           GOTO 999
144     C
145     C
146     C  PART II TRANSFER VIA NBS NET NOT READY YET
147     C
148     500   WRITE(ITYP,501)
```

```
149       501   FORMAT(1H ,'NBS NET TRANSFER NOT READY-EXIT')
150       C
151       C  RETURN TO INTRUP
152       C
153       999   CALL PUSH(IUNIT,12)
154             RETURN
155             END
```

```
      C-------------------------------------------------------------------
            SUBROUTINE WRFILE(IUNIT)
      C-------------------------------------------------------------------
      C
      C  SUBROUTINE WRITES THE OUTPUT FILE OF COUNT TIMES AND COUNTS TO DISK
      C
            IMPLICIT INTEGER*2 (I-N)
             INTEGER*4 NBS
      C
            COMMON /ANG/       BANG(2),       CANG(2),       DANG(2),
           1 EANG(2),          IRATE(2),      PANG(2),       RATE(2),
           2 TANG(2),          IZRST(2),      NPTPK(2)
      C
            COMMON /COUN/      ICNT(2,16),    IPT(2),        ITOT(2),
           1 LIMCNT(2),        ISAVE(2)
      C
            COMMON /IO/        NBS,           IDT,           ICLOCK,
           1 ISPEC,            IRD,           IPRT,          ITYP,
           2 IDISK(2,2),       IFIN,          IFOUT
      C
            COMMON /TIM/        BKGTIM(2),    FIXERR(2),     IPK(2),
           1 NBG(2),            NPK(2),       TIME(2),       TIMLIM(2)
      C
            COMMON /PROB/      IDAT(3),       IFLAG(2,10),   ITIM(3),
           1 ITITLE(50)
      C
            COMMON /RNDT/      NREC(2),       NMFIL(2,6),    ITR(2),
           1 IENRN(2,4)
      C
      C*************************************************************************
      C
      C  IS THERE ANY DATA IN THE BUFFER TO BE WRITTEN? IF NOT THEN RETURN
      C
            J = IPT(IUNIT)
            IF(J.EQ.0) GOTO 999
      C
      C  GET THE TIME
      C
            CALL DATE
      C
      C  WRITE THE DATA TO THE OUTPUT FILE. CANG IS THE ANGLE OF THE LAST DATA POINT
      C
            IFILE = IDISK(IUNIT,2)
            WRITE(IFILE,1000)CANG(IUNIT),ITIM,(ICNT(IUNIT,I),I=1,J)
      1000  FORMAT(F8.3,2X,3I2,16I7)
      C
      C  KEEP TRACK OF THE NUMBER OF RECORDS WRITTEN
      C
            NREC(IUNIT) = NREC(IUNIT) + 1
      C
      C  ZERO BUFFER ARRAY AND RESET POINTER
      C  SUM THE COUNTS SAVED IN ICNT( ,I) FOR TEST IF SHUTTER IS OPEN
      C
            ISUM=0
            IPT(IUNIT) = 0
            DO 100 I=1,16
            ISUM=ISUM+ICNT(IUNIT,I)
      100   ICNT(IUNIT,I) = 0
      C
      C  TEST SUM OF COUNT ARRAY FOR SUM>0 AN INDICATION THAT SHUTTER IS OPEN
      C
            IF(ISUM.GT.1) GOTO 999
            WRITE(ITYP,120)
      120   FORMAT(1H0,'**WARNING**ICNT ARRAY IS FILLED WITH ZEROS')
      C
      C  ALL DONE (NOTE: FINIS WILL WRITE FILE MARK AT THE END OF THE FILE)
      C
      999   RETURN
            END
```

Appendix 2

LISTING OF PROGRAM CONFIG

This program creates the instrument specification file DACONF read by AUTO and written as records 2, 3, 4, and 5 on any output file. The file DACONF is eight records long. The first four describe goniometer #1 and the second four describe goniometer #2. The format for either set of four is given in Table 5 as records 2 to 5. The function READX permits free field input from the consol terminal.

```
C-------------------------------------------------------------------
C-------------------------------------------------------------------
C     *****CONFIG*****CONFIG*****CONFIG*****CONFIG*****CONFIG*****
C-------------------------------------------------------------------
C-------------------------------------------------------------------
C
C     THIS RCUTINE READS AND/OR CREATES THE UNIT CONFIGURATION FILE.
C     IBRNCH = 1    ACTIVATE, READ AND LIST CONTENTS
C            = 2    CREATE FILE .
C
C     IFLAG  = 0    FILE CREATION MODE
C            = 1    FILE REVISION MODE
C            = 2    FILE READING MODE (REVISION POSSIBLE FOLLOWING READ)
C
C*********************************************************************
C
C COMMONS ARE IDENTICAL WITH THOSE USED IN AUTO
C  EXCEPTION IS ARRAY JFLAG(2,10) WHICH IS IFLAG(2,10) IN AUTO
C
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*4 NBS
C
      COMMON /BUFF/     IALPH(66),     NAME(15)
C
      COMMON /IO/       NBS,           IDT,          ICLOCK,
     1 IO4,             IRD,           IPRT,         ITYP,
     2 IDISK(2,2),      IFIN,          IFOUT
C
      COMMON /PROB/     IDAT(3),       JFLAG(2,10),  ITIM(3),
     1 ITITLE(50)
C
       DIMENSION XCALIB(2,6),ICALIB(2,6),XMAT(2,15)
      DIMENSION ISPEC(2,64)
      DATA IBLNK/'  '/
C
C ASSIGN LOGICAL UNIT NUMBERS
C
      IDT = 2
      IO4=4
      IRD=5
      IPRT=6
      ITYP=7
C
C  ATTATCH THE LOGICAL UNITS TO THE PHYSICAL DEVICES
C
      CALL SYSTEM('ZU',2)
      CALL SYSTEM('ND D0',5)
      CALL SYSTEM('AS 26D',6)
C  LOGICAL UNIT #4 WILL BE ALLOCATED OR ACTIVATED LATER
      CALL SYSTEM('AS 520',6)
      CALL SYSTEM('AS 662',6)
      CALL SYSTEM('AS 720',6)
C
C     SET IUNIT TO 1 FOR FIRST ENTRY. WILL BE RESET TO 2 FOR SECOND PASS
C
      IUNIT = 1
C
C  READ SYSTEM CLOCK TO GET CURRENT DATE AND TIME
C
      CALL DATE
C
C  READ FUNCTION DESIRED (=2 READ, =1 CREATE)
C
      WRITE(ITYP,10) IDAT,ITIM
10    FORMAT(1H0,'CONFIGURATION FILE CREATION AND UPDATING ROUTINE'/
     * 1H ,15X,'RUN ON ',I2,2(1H/,I2),2X,I2,2(1H:,I2)/
     * 1H0,'INPUT FUNCTION DESIRED: 1=CREATE  OR  2=READ/UPDATE')
      IBRNCH=READX(IW,N,2,IRD,ITYP,IEOF)
C-----BRANCH DEPENDING ON WHO CALLED (OR PURPOSE)
      IF(IBRNCH.EQ.2) GOTO 400
C*********************************************************************
C      FILE CREATION BLOCK
C*********************************************************************
```

```
C     ENCODE IALPH FOR ALLOCATION OF THE CONFIGURATION FILE ON LOGICAL
C     UNIT 104
C
C
      ENCODE(IALPH,50) I04
50    FORMAT('AL DACONF,', Z1, ',1,128')
      ISIZE=17
C
C     ALLOCATE FILE DACONF WITH 1 CYLINDER AND 128 BYTES
C     ATTATCH LOGICAL UNIT I04
C
      CALL SYSTEM(IALPH,ISIZE)
C
C     SET FLAG FOR INITIAL ENTRY (=0)
C
      IFLAG = 0
C
C     ENTER UNIT NAME INTO ISPEC ARRAY
C
100   WRITE(ITYP,102) IUNIT
102   FORMAT(/1X,'ENTER NAME FOR UNIT#',I2,' (A4)' )
      READ(IRD,110) (ISPEC(IUNIT,I),I=1,2)
110   FORMAT(2A2)
C
C     STORE CURRENT DATE (6 CHARACTERS)
C
      ENCODE(IALPH,118) IDAT
118   FORMAT(3I2)
      DO 120 I=1,3
      KK = I + 2
120   ISPEC(IUNIT,KK) = IALPH(I)
C
C     WHO IS ENTERING CONFIG INFORMATION
C
      WRITE(ITYP,130)
130   FORMAT(1X,'ENTER INITIALS (A2)' )
      READ(IRD,110) ISPEC(IUNIT,6)
C  FOR UPDATE BRANCH TO ECHO LINE  **NA**
      IF(IFLAG.EQ.1) GOTO 600
C
C     TARGET, LINE, WAVELENGTH, FOCAL SPOT, TAKE OFF ANGLE
C
140   WRITE(ITYP,141) (ISPEC(IUNIT,I),I=1,2)
141   FORMAT(/1X,'ENTER IN FREE FORMAT FOR UNIT ',2A2/
     * 1X,'TUBE TARGET, LINE, LAMBDA, TYPE, TAKE OFF'/
     * 1X,'    A2         A2    F6.4    A2     F4.1  ')
C     TARGET (A2)
      IDUM=READX(IW,N,2,IRD,ITYP,IEOF)
      ISPEC(IUNIT,7)=IW
C     CHARACTERISTIC LINE FROM TARGET (A2)
      IDUM=READX(IW,N,1,IRD,ITYP,IEOF)
      ISPEC(IUNIT,8)=IW
C     WAVELENGTH (F6.4 = 3A2)
      XLAM = READX(IW,N,1,IRD,ITYP,IEOF)
      ENCODE(IALPH,142) XLAM
142   FORMAT(F6.4)
C  STORE DATA INTO ARRAY ISPEC(IUNIT,9-11)
      DO 144 I =1,3
      KK = I + 8
144   ISPEC(IUNIT,KK)=IALPH(I)
C     FOCAL SPOT CODE (A2)
      IDUM = READX(IW,N,1,IRD,ITYP,IEOF)
      ISPEC(IUNIT,12)=IW
C     TAKE OFF ANGLE (F4.1=2A2)
      TOF = READX(IW,N,1,IRD,ITYP,IEOF)
      ENCODE(IALPH,146)TOF
146   FORMAT(F4.1)
      DO 148 I=1,2
      KK = I + 12
148   ISPEC(IUNIT,KK)=IALPH(I)
C  FOR UPDATE BRANCH TO ECHO LINE  **TG**
      IF (IFLAG .EQ. 1) GO TO 620
C
C     DIV SLIT CODE AND ANGLE (DEG) OR LENGTH (MM)
```

```
148       C      GONIOMETER RADIUS IN MM
149       C      INCIDENT AND DIFFRACTED BEAM SOLLER SLIT ANGLES
150       C      RECEIVING SLIT ANGLE (DEG)
151       C
152       160    WRITE(ITYP,161)(ISPEC(IUNIT,I),I=1,2)
153       161    FORMAT(1HO,'ENTER IN FREE FORMAT FOR UNIT ',2A2/,1X,
154            &'DIV SLIT TYPE, ANG/LEN, RADIUS, IN SOLLER, OUT SOLLER, REC SLIT'/
155            &'       A2          F6.1      F6.1        F4.1       F4.1         F4.1')
156       C       DIVERGENT SLIT TYPE: FX=FIXED/TC=THETA-COMP
157             IDUM = READX(IW,N,2,IRD,ITYP,IEOF)
158             ISPEC (IUNIT,15)=IW
159       C     DIVERGENT SLIT ANGLE(FX) OR
160       C     IRRADIATED LENGTH IN MM (TC)
161             ANG=READX(IW,N,1,IRD,ITYP,IEOF)
162             ENCODE(IALPH,164) ANG
163       164   FORMAT(F6.1)
164       C  STORE DATA IN ISPEC(IUNIT,16-18)
165             DO 166 I=1,3
166             KK = I + 15
167       166   ISPEC(IUNIT,KK)=IALPH(I)
168       C     GONIOMETER CIRCLE RADIUS IN MM
169             RAD=READX(IW,N,1,IRD,ITYP,IEOF)
170             ENCODE(IALPH,164)RAD
171       C  STORE IN ISPEC(IUNIT,19-21)
172             DO 170 I=1,3
173             KK = I + 18
174       170   ISPEC(IUNIT,KK)=IALPH(I)
175       C     GET INCIDENT SOLLER,DIFF SOLLER AND RECEIVING SLIT ANGLES
176             DO 190 J=1,3
177             ANG = READX(IW,N,1,IRD,ITYP,IEOF)
178             ENCODE(1ALPH,180)ANG
179       180   FORMAT(F4.1)
180       C  STORE DATA INTO ISPEC(IUNIT,22-23 24-25 26-27)
181             DO 184 I=1,2
182             KK = I + 19 + J * 2
183       184   ISPEC(IUNIT,KK)=IALPH(I)
184       190   CONTINUE
185       C  FOR UPDATE BRANCH TO ECHO LINE  **SL**
186             IF (IFLAG.EQ.1) GO TO 630
187       C
188       C     MONOCHROMATOR CODE: GR=GRAPHITE/NN=NONE
189       C     DETECTOR CODE     : SC=SCINTILLATOR
190       C     DEAD TIME PARAMETER IN MICROSECONDS
191       C
192       200   WRITE(ITYP,201)(ISPEC(IUNIT,1),I=1,2)
193       201   FORMAT(/1X,'ENTER IN FREE FORMAT FOR UNIT ',2A2/1X,
194            &'MONO CODE, DETECTOR CODE, DEAD TIME, INT. STABILITY '/
195            &'    A2          A2            F4.2       F4.2')
196       C     MONO CODE
197             IDUM=READX(IW,N,2,IRD,ITYP,IEOF)
198             ISPEC(IUNIT,28)=IW
199       C     DETECTOR CODE
200             IDUM = READX(IW,N,1,IRD,ITYP,IEOF)
201             ISPEC(IUNIT,29)=IW
202       C     DEAD TIME
203             TAU = READX(IW,N,1,IRD,ITYP,IEOF)
204             ENCODE(IALPH,180)TAU
205       C  STORE DATA IN ISPEC(IUNIT,30-31)
206             DO 210 I=1,2
207             KK = I + 29
208       210   ISPEC(IUNIT,KK)=IALPH(1)
209       C INSTRUMENT STABILITY IN PERCENT
210       C
211             STAB = READX(IW,N,1,IRD,ITYP,IEOF)
212             ENCODE(1ALPH,180)STAB
213       C
214       C  STORE DATA IN 32-33
215             DO 220 I=1,2
216             KK = I + 31
217       220   ISPEC(IUNIT,KK) = IALPH(I)
218       C
219       C-----BLANK FILL REMAINDER OF 128 BYTE RECORD
220       C
```

```
221           DO 290 I=34,64
222     290   ISPEC(IUNIT,I)=IBLNK
223     C  FOR UPDATE BRANCH TO ECHO LINE  **DT**
224           IF (IFLAG.EQ.1) GO TO 640
225     C
226     C   SET UP CALIBRATION RECORD FLAGS
227     C
228           WRITE(ITYP,292) (ISPEC(IUNIT,I),I=1,2)
229       292 FORMAT(/1X,'ENTER CALIBRATION FLAGS FOR UNIT ',2A2/1X,
230          1 '0/1 FOR (DO NOT)/(DO) APPLY FLAT SPEC, AXIAL DIV,'
231          2 ' LORENTZ CORRECTIONS')
232           DO 294 I=1,6
233           ICALIB(IUNIT,I)=0
234       294 CONTINUE
235           ICALIB(IUNIT,1) = READX(IW,N,2,IRD,ITYP,IEOF)
236           ICALIB(IUNIT,2) = READX(IW,N,1,IRD,ITYP,IEOF)
237           ICALIB(IUNIT,3) = READX(IW,N,1,IRD,ITYP,IEOF)
238     C
239     C-----SET UP CALIBRATION RECORD(128 BYTES)
240     C
241     300   WRITE(ITYP,310)(ISPEC(IUNIT,I),I=1,2)
242     310   FORMAT(/1X,'ENTER CALIBRATION CONSTANTS FOR UNIT ',2A2
243          1 /1X,'ZERO OFFSET, A0, A1, A2, A3, A4')
244     C     ZERO CALIB ARRAY
245           DO 320 I=1,6
246     320   XCALIB(IUNIT,I)=0.0
247           XCALIB(IUNIT,1) = READX(IW,N,2,IRD,ITYP,IEOF)
248           DO 330 I=2,6
249           XCALIB(IUNIT,I)=READX(IW,N,1,IRD,ITYP,IEOF)
250           IF (N.EQ.0) GO TO 330
251           XCALIB(IUNIT,I)=0.0
252           GO TO 340
253     330   CONTINUE
254     C  FOR UPDATE BRANCH TO ECHO LINE  **CL**
255     340   IF (IFLAG.EQ.1) GO TO 650
256     C
257     C  SET UP THE INVERSE MATRIX
258     C
259     360   WRITE(ITYP,362) (ISPEC(IUNIT,I),I=1,2)
260     362   FORMAT(/1X,'ENTER FOR UNIT ',2A2,' INVERSE MATRIX ELEMENTS: 00',
261          * ' 01 02 03 04 11 ... 34 44')
262           DO 370 I=1,15
263     370   XMAT(IUNIT,I) = 0.0
264           XMAT(IUNIT,1) = READX(IW,N,2,IRD,ITYP,IEOF)
265           DO 380 I=2,15
266           XMAT(IUNIT,I) = READX(IW,N,1,IRD,ITYP,IEOF)
267           IF(N.EQ.0)GO TO 380
268           XMAT(IUNIT,I) = 0.0
269           GO TO 390
270     380   CONTINUE
271     C  FOR UPDATE BRANCH TO ECHO LINE **MX**
272     390   IF(IFLAG.EQ.1) GOTO 650
273     C
274     C     IF IUNIT=1 RETURN TO ENTER DATA FOR IUNIT=2
275     C     ELSE ECHO INPUT
276     C
277           IF(IUNIT.EQ.2) GOTO 600
278           IUNIT=2
279           GO TO 100
280     C*********************************************************************
281     C     ACTIVATE CONFIG DATA FILE AND READ IT
282     C*********************************************************************
283     400   ENCODE(IALPH,410)IO4
284     410   FORMAT('AC DACONF,',Z1)
285           ISIZE = 11
286           CALL SYSTEM(IALPH,ISIZE)
287     C
288     C     READ CONFIG FILE FOR INSTRUMENT SPECIFICATIONS AND CALIBRATION
289     500   DO 530 IUNIT=1,2
290           READ(IC4,510,ERR=540)(ISPEC(IUNIT,I),I=1,64)
291     510   FORMAT(64A2)
292     C
293     C  STORE CURRENT DATE
294     C
295           DO 512 I=1,3
```

```
296           KK = I + 2
297       512 ISPEC(IUNIT,KK) = IDAT(I)
298           READ(IO4,520,ERR=540)(ICALIB(IUNIT,I),I=1,6),(XCALIB(IUNIT,I)
299          1 ,I=1,6)
300       520 FORMAT(6I1,F10.3,5E16.8))
301           READ(IC4,524,ERR=540)(XMAT(IUNIT,I),I=1,15)
302       524 FORMAT(8E16.7)
303       530 CONTINUE
304           IFLAG=2
305           GO TO 600
306       C-----ERROR DETECTED
307       540 WRITE(ITYP,550)IO4
308       550 FORMAT(1HO,'***ERROR ON LU=',I1,'  TERMINATING')
309           CALL EXIT
310       C*************************************************************************
311       C     ECHO INSTRUMENT SPECIFICATIONS
312       C*************************************************************************
313       600 WRITE(ITYP,610)(I,(ISPEC(I,J),J=1,6),I=1,2)
314       610 FORMAT(1HO,'**NA**',2X,'UNIT',3X,'NAME',4X,'DATE',3X,'INIT'/
315          *(9X,I2,4X,2A2,3X,3A2,3X,A2))
316           IF (IFLAG.EQ.1) GO TO 700
317       620 WRITE(ITYP,621)(I,(ISPEC(I,J),J=7,14),I=1,2)
318       621 FORMAT(1HO,'**TG**',2X,'UNIT',3X,'TARGET',3X,'LINE',3X,'LAMBDA',
319          *3X,'TYPE',3X,'TCA'/
320          *(9X,I2,6X,A2,6X,A2,4X,3A2,4X,A2,4X,2A2))
321           IF (IFLAG.EQ.1) GO TO 700
322       630 WRITE(ITYP,631)(I,(ISPEC(I,J),J=15,27),I=1,2)
323       631 FORMAT(1HO,'**SL**',2X,'UNIT',3X,'DIV SLIT',3X,'RADIUS',4X,
324          *'SOLLERS',3X,'REC SLIT'
325          */(9X,I2,5X,A2,3A2,3X,3A2,3X,2A2,1X,2A2,3X,2A2))
326           IF (IFLAG.EQ.1) GO TO 700
327       640 WRITE(ITYP,641)(I,(ISPEC(I,J),J=28,33),I=1,2)
328       641 FORMAT(1HO,'**DT**',2X,'UNIT',3X,'MONO',3X,'DETECTOR',3X,
329          *'DEAD TIME,  INT. STABILITY'/(9X,I2,4X,A2,7X,A2,10X,2A2,
330          * 10X,2A2,1H%))
331           IF (IFLAG.EQ.1) GO TO 700
332       650 WRITE(ITYP,652)(I,(ICALIB(I,J),J=1,3),(XCALIB(I,K),K=1,6),I=1,2)
333       652 FORMAT(1HO,'**CL**',2X,'UN FS AX LZ OFFSET    A0',5X,'A1',
334          * 10X, 'A2',10X,'A3',10X,'A4'/(9X,I2,3(1X,I2),F6.2,F6.3,4E12.5))
335           IF(IFLAG.EQ.1) GOTO 700
336       670 WRITE(ITYP,672) (I,(XMAT(I,J),J=1,15),I=1,2)
337       672 FORMAT(1HO,'**MX**  UNIT'/
338          * (10X,I1,2X,5E12.5/25X,4E12.5/37X,3E12.5/
339          * 49X,2E12.5 /61X,E12.5//))
340       C*************************************************************************
341       C     ALLOW FOR CORRECTION
342       C*************************************************************************
343       C
344       C-----WHAT LINE? SET UPDATE FLAG
345       C
346       700 WRITE(ITYP,731)
347       731 FORMAT(1HO,'TO EDIT ENTER LINE CODE AND UNIT# (A2,I1) , "OP",
348          1 "TY", OR "OK"')
349       738 READ(IRD,740,ERR=749) IALPH(1),IUNIT
350       740 FORMAT(A2,I1)
351           IF(IALPH(1).EQ.'OK') GOTO 800
352       C-----DO WE LIST OPTIONS?
353           IF(IALPH(1).EQ.'OP') GOTO 760
354       C-----ALLOW FOR COMPLETE LISTING
355           IFLAG=0
356           IF(IALPH(1).EQ.'TY')GO TO 600
357       C  AFTER REENTRY OF DATA LIST ONLY THE LINE INPUT (IFLAG=1)
358           IFLAG=1
359       C
360       C  CHECK FOR VALID UNIT NUMBER
361       C
362           IF(IUNIT.NE.1 .AND. IUNIT.NE.2) GO TO 749
363           IF (IALPH(1).EQ.'NA') GO TO 100
364           IF (IALPH(1).EQ.'TG') GO TO 140
365           IF (IALPH(1).EQ.'SL') GO TO 160
366           IF (IALPH(1).EQ.'DT') GO TO 200
367           IF (IALPH(1).EQ.'CL') GO TO 300
368           IF(IALPH(1).EQ.'MX') GOTO 360
369       749 WRITE(ITYP,750)
```

```
      750   FORMAT(1H ,'**ILLEGAL CODE OR UNIT # ENTERED**TRY AGAIN**')
      C
      C  LIST EDIT CODE OPTIONS
      C
      760   WRITE(ITYP,770)
      770   FORMAT(/1X,'LEGAL OPTIONS ARE:  NA#, TG#, SL#, DT#, CL#, MX#'/
           1 1X,'                PLUS: TY to list all'/
           2 1X,'                      OP to list options'/
           3 1X,'                      OK to terminate edit')
            GOTO 700
      C*****************************************************************
      C     WRITE CALIB FILE (IFLAG = 0,1) TO IO4
      C     PRINT CALIB FILE TO IPRT
      C*****************************************************************
      800   IF(IFLAG.EQ.2) GO TO 999
            WRITE(ITYP,810)
      810   FORMAT(1H0,'WRITE/PRINT SPECIFICATION FILE? (Y/N) DEFAULT=Y')
            READ (IRD,820)IALPH(1)
      820   FORMAT(A1)
            IF (IALPH(1).EQ.'N') GO TO 999
            REWIND IO4
            DO 840 I=1,2
            WRITE(IO4,830)(ISPEC(I,J),J=1,64),(ICALIB(I,L),L=1,6),
           1 (XCALIB(I,K),K=1,6),(XMAT(I,M),M=1,15)
            WRITE(IPRT,831) (ISPEC(I,J),J=1,64),(ICALIB(I,L),L=1,6),
           * (XCALIB(I,K),K=1,6), (XMAT(I,M),M=1,15)
      831   FORMAT(1H0,64A2/1X,6I1,F10.3,5E16.8/1X,8E16.8/1X,7E16.8)
      830   FORMAT(64A2/6I1,F10.3,5E16.8/8E16.8/7E16.8)
      840   CONTINUE
      C
      C  PROGRAM COMPLETION
      C
            WRITE(ITYP,9999) IO4
      9999  FORMAT(/1X,'FILE "DACONF" WRITTEN TO LOGICAL UNIT ',I2)
      999   STOP
            END
```

FEDERAL INFORMATION PROCESSING STANDARD SOFTWARE SUMMARY

01. Summary date — Yr. 80 Mo. 11 Day 25

02. Summary prepared by (Name and Phone)
Camden R. Hubbard 301/921-2921

03. Summary action
New [X] Replacement [ ] Deletion [ ]
Previous Internal Software ID

04. Software date — Yr. 80 Mo. 11 Day 25

05. Software title
AUTO - A Real Time Diffractometer Control System

06. Short title AUTO

07. Internal Software ID

08. Software type
- [X] Automated Data System
- [ ] Computer Program
- [ ] Subroutine/Module

09. Processing mode
- [X] Interactive
- [ ] Batch
- [ ] Combination

10. Application area

General
- [ ] Computer Systems Support/Utility
- [X] Scientific/Engineering
- [ ] Bibliographic/Textual
- [ ] Management/Business
- [ ] Process Control
- [ ] Other

Specific
X-ray powder diffraction

11. Submitting organization and address
Ceramics Glass & Solid State Science Division
A221, MATL
National Bureau of Standards
Washington, D. C. 20234

12. Technical contact(s) and phone
Camden R. Hubbard
301/921-2921

13. Narrative

Two x-ray powder diffractometers are controlled by this event driven control program. AUTO allows for file handling including transfers. The data collection options include complete pattern and quantitative analysis. The program optimizes time or counting precision for quantitative analysis. The data file structure permits complete analysis without further input by the user.

14. Keywords
X-ray powder diffraction; goniometer automation; quantitative analysis

15. Computer manuf'r and model	16. Computer operating system	17. Programing language(s)	18. Number of source program statements
Interdata 7/16	FLOXOS	FORTRAN	~4300
19. Computer memory requirements	**20. Tape drives**	**21. Disk/Drum units**	**22. Terminals**
54 K bytes (=8 bits)	none	1 - 5M bytes	1

23. Other operational requirements
NBS Bus instrument interface

24. Software availability
Available [X] Limited [ ] In-house only [ ]
from Camden R. Hubbard on magnetic tape

25. Documentation availability
Available [X] Inadequate [ ] In-house only [ ]
from Camden R. Hubbard NBSIR

26. FOR SUBMITTING ORGANIZATION USE

Standard Form 185
1974 July
U.S. Dept. of Commerce—NBS
(FIPS. Pub. 30)

INSTRUCTIONS

01. **Summery Date.** Enter date summery prepared. Use Yeer, Month, Day formet: YYMMDD.

02. **Summary Prepared By.** Enter name and phone number (Inciuding erea code) of Individuai who prepared this summary.

03. Summary Action. Mark the eppropriate box for new summery, **replacement** summery or **deletion** of summary. If this software summary is a replacement, enter under "Previous Internal Softwere ID" the internal software Identificetion as reported in item 07 of the original summary, and enter the new internal software identification in item 07 of this form; complete all other items as for a new summary. If a software summary is to be **deleted**, enter under "Previous Internal Software ID" the internal softwere identification as reported in Item 07 of the original summery; complete only items 01, 02, 03 and 11 on this form.

04. **Softwere Date.** Enter date software was completed or last updated. Use Year, Month, Day formet: YYMMDD.

05. **Software Title.** Make title as descriptive as possible.

06. **Short Title.** (Optional) Enter commonly used abbreviation or acronym which identifies the software.

07. **Internal Software ID.** Enter a unique identification number or code.

08. **Software Type.** Mark the appropriate box for an **Automated Data System** (set of computer programs), **Computer Program**, or **Subroutine/Module**, whichever best describes the software.

09. **Processing Mode.** Mark the appropriate box for an **Interactive, Batch**, or **Combination** mode, whichever best describes the software.

10. Application Area.
    General: Mark the appropriate box which best describes the general area of application from among:

Computer Systems Support/Utility	**Process Control**
Management/Business	**Bibliographic/Textual**
Scientific/Engineering	**Other**

    Specific: Specify the sub-area of application; e.g.: "COBOL optimizer" if the general area is "Computer Systems Support/Utility"; "Payroll" if the general area is "Management/Business"; etc. Elaborate here if the general area is "Other."

11. **Submitting Organization and Address.** Identify the organization responsible for the software as completely as possible, to the Branch or Division level, but including Agency, Department (Bureau/Administration), Service, Corporation, Commission, or Council. Fill in complete mailing address, including mail code, street address, city, state, and ZIP code.

12. **Technical Contact(s) and Phone:** Enter person(s) or office(s) to be contacted for technical information on subject matter end/or operational espects of software. Include telephone area code. Provide organization name and mailing address, if different from that in item 11.

13. **Narrative.** Describe concisely the problem addressed and methods of solution. Include significant factors such as special operating system modifications, security concerns, relationships to other software, input and output media, virtual memory requirements, and unique hardware features. Cite references, if appropriate.

14. **Keywords.** List significant words or phrases which reflect the functions, applications end features of the software. Separate entries with semicolons.

15. **Computer Manufacturer and Model.** Identify mainframe computer(s) on which software is operational.

16. **Computer Operating System.** Enter name, number, end releese under which software is operating. Identify enhancements in the Nerretive (item 13).

17. **Programing Language(s).** Identify the language(s) in which the software is written, inciuding version; e.g., ANSI COBOL, FORTRAN V, SIMSCRIPT II.5, SLEUTH II.

18. **Number of Source Program Statements.** Include statements in this software, separate macros, called subroutines, etc.

19. **Computer Memory Requirements.** Enter minimum internal memory necessary to execute software, exclusive of memory required for the operating system. Specify words, bytes, characters, etc., and number of bits per unit. Identify virtual memory requirements in the Narrative (item 13).

20. **Tape Drives.** Identify number needed to operate software. Specify, if critical, manufacturer, model, tracks, recording density, etc.

21. **Disk/Drum Units.** Identify number and size (in same units es "Memory"—item 19) needed to operate software. Specify, if critical, manufacturer, model, etc.

22. **Terminals.** Identify number of terminals required. Specify, if critical, type, speed, character set, screen/line size, etc.

23 **Other Operational Requirements.** Identify peripheral devices, support software, or related equipment not indicated above, e.g., optical character devices, facsimile, computer-output microfilm, graphic plotters.

24. **Software Availability.** Mark the appropriate box which best describes the software availability from among: **Available to the Public, Limited Availability** (e.g.: for government use only), and **For-In-house Use Only.** If the software is "Available", include a mail or phone contact point, as well as the price and form in which the software is available, if possible.

25. **Documentation Availability.** Mark the appropriate box which best describes the documentation availability from among: **Available to the Public, Inadequate for Distribution**, and **For In-house Use Only.** If documentation is "Available", include a mail or phone contact point, as well as the price and form in which the documentation is available, if possible. If documentation is presently "Inadequate", show the expected availability date.

26. **For Submitting Organization Use.** This area is provided for the use of the organization submitting this summary. It may contain eny information deemed useful for internal operation.

GPO : 1974 O - 554-655

NBS-114A (REV. 9-78)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET	1. PUBLICATION OR REPORT NO. NBSIR 81-2229	2. Gov't. Accession No.	3. Recipient's Accession No.

4. TITLE AND SUBTITLE

AUTO:
A REAL TIME DIFFRACTOMETER CONTROL SYSTEM

5. Publication Date: February 1981

6. Performing Organization Code

7. AUTHOR(S)

R. L. Snyder, C. R. Hubbard, and N. C. Panagiotopoulos

8. Performing Organ. Report No.

9. PERFORMING ORGANIZATION NAME AND ADDRESS

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, DC 20234

10. Project/Task/Work Unit No.

11. Contract/Grant No.

12. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS *(Street, City, State, ZIP)*

13. Type of Report & Period Covered

14. Sponsoring Agency Code

15. SUPPLEMENTARY NOTES

[X] Document describes a computer program; SF-185, FIPS Software Summary, is attached.

16. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)*

Two x-ray powder diffractometers are independently driven by the program. The program provides the capability to scan an entire spectrum and to carefully measure the intensities of individual lines for quantitative analysis. For intensity measurements, an optimization routine is used to either minimize data collection time, or to minimize the estimated standard deviation of the net intensity. The data file structures are designed to permit complete data analysis without further data input. The automation program also provides for all aspects of file handling including opening, writing, transferring and deleting.

17. KEY WORDS *(six to twelve entries; alphabetical order; capitalize only the first letter of the first key word unless a proper name; separated by semicolons)*

Automation; powder diffraction; quantitative analysis; x-ray.

18. AVAILABILITY [XX] Unlimited

[ ] For Official Distribution. Do Not Release to NTIS

[ ] Order From Sup. of Doc., U.S. Government Printing Office, Washington, DC 20402, SD Stock No. SN003-003-

[X] Order From National Technical Information Service (NTIS), Springfield, VA. 22161

19. SECURITY CLASS (THIS REPORT) UNCLASSIFIED

21. NO. OF PRINTED PAGES

20. SECURITY CLASS (THIS PAGE) UNCLASSIFIED

22. Price

USCOMM-DC