UNIVERSITY OF CALIFORNIA

LICK OBSERVATORY TECHNICAL REPORTS

No. 6

THE GAERTNER AUTOMATIC MEASURING SYSTEM
FOCAL Programs for Data Taking

Lloyd B. Robinson

Santa Cruz, California
May 1974

The Lick Observatory Gaertner Automatic Measuring System

FOCAL Programs for Data Taking

Other Relevant Lick Technical Reports

No. 1	The Lick Observatory PDP 8/I Computers
3	PDP 8 Machine Language Listings
5	Gaertner Automatic Measuring System ("AME") Data Reduction
7	AME - Machine Language Program
8	AME - Electronic Controls

Acknowledgement

Funds for the installation of the PDP 8 in the Automatic Measuring System were provided by NSF Grant GP 32459.

Cal Delaney and Terry Ricketts were responsible for the development and installation of the computer interface circuitry and new electronic controls system.

INDEX

Page

COMPUTER CONTROL OF AUTOMATIC MEASURING ENGINE

Introduction

The Lick Automatic Measuring System was built by the Gaertner Scientific Corporation, Chicago, in accordance with a design formulated in 1959. It was installed at the Santa Cruz laboratories of Lick Observatory in 1967.

It consists of a Survey Machine, which is used for initial inspection of plates and selection of objects for measurement, plus the Automatic Measuring Engine, which uses coordinates obtained from the Survey Machine to locate and precisely measure the position of images on the photographic plate. (Manual selection of objects on the measuring engine without prior survey is also possible.) When the system was first put into operation, approximate coordinates of selected objects were measured and recorded on punched cards. These cards then served as input for the automatic measuring process. A card was read, the Automatic Engine located the object and then photometric and precise coordinate measurements were made and punched on a card. Normally this process was automatically repeated until the last card of the survey deck was read.

The Automatic Measuring System was heavily used in the period before 1972 with a remarkable increase in precision and efficiency over traditional methods. There occurred, however, many interruptions in the automatic process, caused by occasionally insufficient accuracy of input coordinates, by errors and failures in the card reader and punch, in the relays which controlled the measuring sequence, by electrical noise, and by occasional mechanical malfunctions.

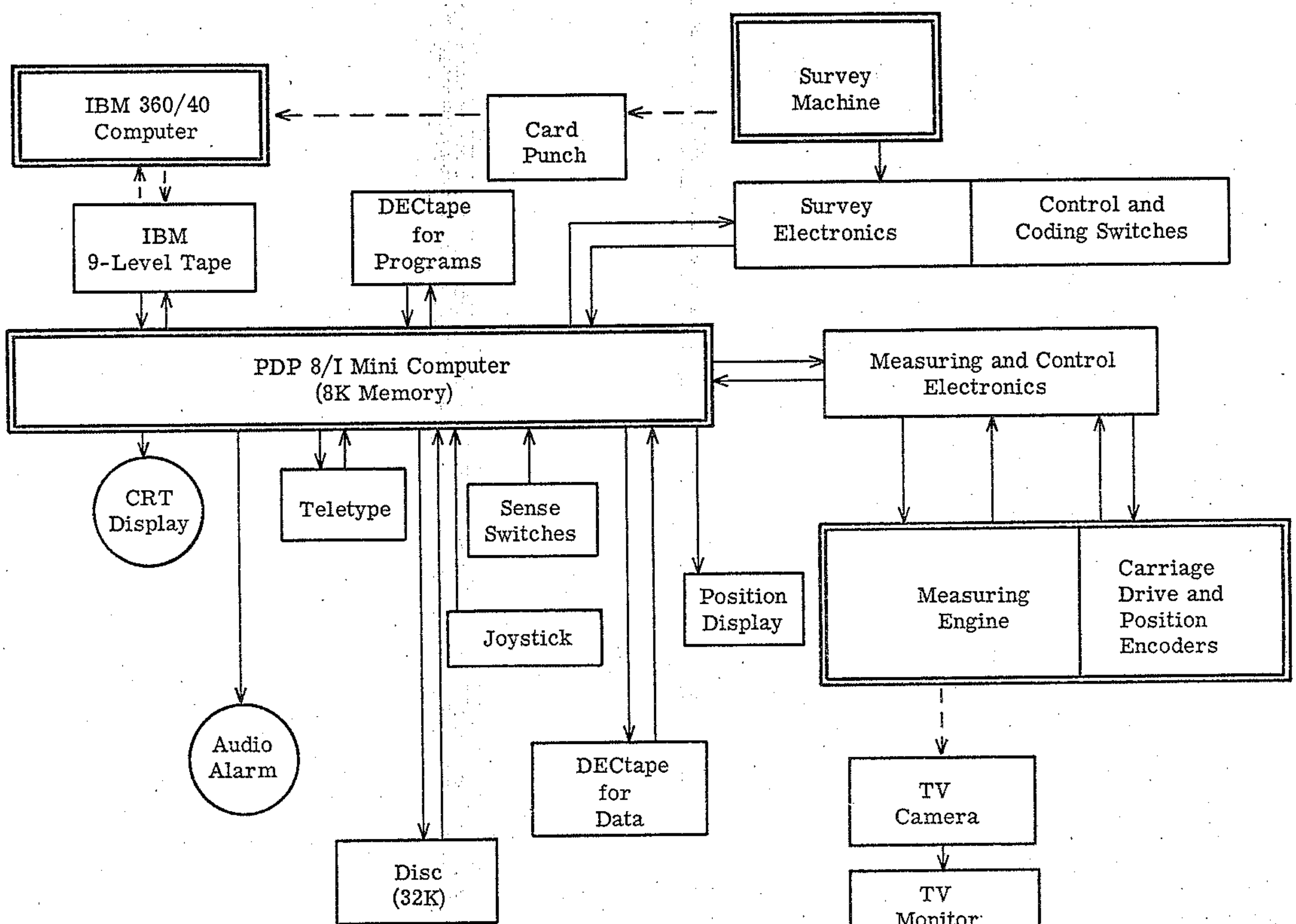
The number of interruptions was large enough to require an attendant when the machine was in operation. The attendant had such tasks as manual centering of poorly located stars, clearing the card punch or reader in case of malfunction, and restarting the operation in cases when no star was found,or when the servo system could not center and image for some reason. Difficult and critical adjustments of the mechanical position of several limit switches, and of the length of some relay controlled delays had to be made at fairly frequent intervals. The interruptions plus the large number of service calls, as well as the need to make the whole measuring operation more flexible, led in 1972 and 1973 to the replacement of most of the relay circuitry with solid state controls, the installation of a PDP 8/I mini-computer to perform many functions formerly done manually, and the replacement of the punched card equipment by magnetic tape units.

The PDP 8/I computer is identical to two others in use at Lick Observatory. It has an 8K 12 bit memory, small (32K) magnetic disc, dual miniature magnetic tapes (DECtape), a CRT display terminal, and a 9 level IBM compatible tape unit. A "joystick" and a specially labeled sense switch panel make it possible for any one to operate the system with little special training.

The FOCAL programming language (developed by Digital Equipment Corporation) has been chosen for this work. This is a high level language, very similar to BASIC and is easily mastered. (It is easier to use than FORTRAN, for example.) We have added a large number of additional commands, so that hardware devices such as the Automatic Measuring System are controlled by simple program statements which are easily learned and used. Program sequences using the commands are prepared by typing on a teletype and may be immediately tested, used and stored.

Programs are stored on a miniature magnetic tape (DECtape) which is fully addressable and on which individual segments of information can be written without disturbing other data. Programs are written, stored on tape, recalled, edited and stored again as often as desired. No compiling is necessary because the language is an interpretive one. Programs can link together, one program calling another as a subroutine from tape. All programs have all variables in common. A program is typically equivalent in computing power to about a printed page of FORTRAN coding. Over a hundred such programs can be linked together on one magnetic tape, so that even with our small computer we can prepare and use very large and complex program sequences.

To operate the system, a program tape is mounted on the computer, then core and disc memories are loaded from it. The program then controls the measuring process, with options and features being selected by switch position on the sense switch panel. Special descriptive information can be entered using the teletype in a question and answer mode. Certain switch selected options allow for manual selection and/or centering of the image before measurement. The programs divide naturally into those needed to locate, measure and record image positions, and those that are used for evaluation and processing of the measurements. This report describes data acquisition programs. The data reduction programs are described in Lick Observatory Technical Report No. 5.



COMPUTER CONTROLLED
AUTOMATIC MEASURING ENGINE

Fig. 1

Measuring Sequence

An overall schematic of the measuring system is shown in figure 1. In order to understand the functions of various items of hardware, it is worthwhile to follow the sequence of events involved in measurement of an image.

-- An image is located on a plate in the Survey Machine.

-- Coordinates are recorded on the disc, later copied to DECtape. (Coordinates can be card-punched but then must be transferred to magnetic tape using the IBM computer center equipment, since no card reader is available to the PDP 8/I.)

-- The plate is moved to the Measuring Engine at some later date.

-- Survey information for all the images on the plate is copied from DECtape to the magnetic disc. These coordinates may be sorted to minimize Measuring Engine travel distance. (Sorting time is only a few seconds.)

-- Plate alignment corrections are made, using the joystick in combination with the TV monitor and a computer program which calculates a plate alignment correction.

-- Survey coordinates for each image are read sequentially from the disc, corrected for plate alignment, and used to approximately center each image for the measuring engine.

-- The AME control electronics are triggered by the computer and operate automatically until the measurement of the exact position is complete, unless some abnormality exists in the image or the initial centering.

-- The micrometers search for the closest millimeter line, null on it, and count 0.2 micron intervals back to home position. The counts are stored in a scaler.

-- At the same time the scanner tries to center exactly on the image.

-- If the scanner centers and nulls successfully, the iris photometer switches on, nulls, and starts to count the distance between the "null" position and the fully open position.

-- When the photometer has nulled, the scanners recenter and count in 0.2 micron steps back to home position. The counts are stored in a scaler.

-- During the measurement, the computer monitors the action, checking that the scanner detected an image, that neither scanner nor micrometer have hit a limit and that the photometer cycle starts and finishes on schedule.

-- If the measurement is successfully completed, the computer reads the scalers, stores the numbers on the disc and moves to the next image.

-- The computer may abort a measurement, recenter the image, or reset any device that acts in an unexpected manner. Such action is needed quite often due to the wide range of image shapes and sizes, and large differences in plate quality, and the occasional occurence of survey errors.

-- When 125 images, or all those wanted on a plate have been measured, the computer copies the data from the disc to DECtape for a permanent record. Subsequent 125 image blocks are recorded on the tape until all images are recorded.

-- The computer FOCAL language program used to measure an image is far too large to fit into the core memory, so that during each image measurement, several subprograms are loaded from DECtape. (The tape transport will be seen to be quite active.)

-- Timing loops in the program prevent the system from hanging up. The program will always move on to the next image after several tries. Abnormal conditions associated with any measurement are flagged by including a diagnostic code with the data for that image, so that the operator can recheck the measurement before the plate is removed from the measuring engine.

DATA: Disc and Tape Access By FOCAL Commands

1. Data is stored on the disc as 24 bit (double precision) integers. These 24 bit integers are copied as 32 bit words to the 9-track IBM tape. Each 24 bit integer uses two adjacent PDP 8 words. (Most significant 12 bits are in the first or lowest address of the two words).

2. FOCAL can access a single 24 bit integer on the disc by use of SET D=FTAKL(B, W) and X PUTL(B, W,D). One must define the disc block No."B"and first 12 bit word No."W."

- Single 12 bit integers are accessed by SET D = FTAK(B, W) and X PUT(B, W, D)

- Floating point variables can be stored and retrieved using X STOR(B,W; D) and SET D = FASK(B, W). Each floating point variable uses 4 12 bit disc words.

3. Images use 16 12 bit words each. The 8 individual segments of an image descriptor can be stored by the commands X SAVX(B, I, D1, D2, D3, D4)
X SAVY(B, I, D5, D6, D7, D8)

   The segments can be retrieved individually by: SET D = FIN(B, I, N); SET D = FOUT(B, I, N). N is given values 1 - - 8 to get segments D1 - - D8. I is the image number 1 - - 125.

4. Data is transferred between DISC and DECTAPE by the commands
   X MPUT(BD, BT, 16, U) - write tape
   X MTAK(BD, BT, 16, U) - read tape

   16 blocks are normally transferred at once, corresponding to one table of 125 image descriptors, either survey or output.

5. Data records on the DISC are transferred to and from IBM tape by the commands X IBMW(B); X IBMW(B+8) - write tape
   X IBMR(B); X IBMR(B+8) - read tape

   A "table" consists of 16 disc blocks, but only 8 can be transferred at once, hence the double write and read instructions.

   A more detailed description of these and other commands is given in Appendix Z. (Page 32.)

AME Data Format

- All data is stored in binary form on PDP8 disc and Dectape, and on IBM compatible Mag. tape. (All under FOCAL control)

- The data is broken up into "tables" holding up to 125 image measurements. A table uses 16 PDP 8 Disc or Dectape "blocks" (2064 12bit words). A PDP 8 "block" holds 129 12 bit words. Data will often cross block boundaries, but will never cross table boundaries.

- Each 125 image data table is written as two 516 word (2064 byte) "IBM-records" on the IBM mag. tape.

- The arrangement of data within a table is shown on the following page. Although data is stored in a non-compact form, to simplify transfer of information between computers, a single DECtape will hold up to 72 tables, and the disk can hold 12 tables simultaneously.

- An identification segment is included with each table, to store plate number, date, etc. This I.D. block will use the first 40 PDP 8 words (the first 20 IBM words) of each table.

SURVEY DATA FORMAT

Star No.	X IN	Y IN	System S	Order O	Class * 10 + Quality	Modify Survey Before Use	Grating Image 0,1,2

Survey: 1 Image

16 PDP 8 words
8 IBM words

DISC AND DECTAPE FORMAT

40 12 bit words I.D.	89 words Data	- - -	129 words Data Block = BI + 14	105 words of data Block = BI + 15	24 blank words
Block = BI					

Descriptors for 125 Images "TABLE"

AME OUTPUT DATA FORMAT

In Buffer BO

Star No.	X Stage	X Micr.	X Scan	Photometer	Y Stage	Y Micr.	Y Scan

Measured Output - 1 Image - Intermediate Format

In Buffer BM

Star No.	X Out	Y Out	B	-	Photo-meter	System Order Class Quality	Diagnostic code

Measured Output: 1 Image in IBM "Final" Format

16 PDP 8 words
8 IBM words

2064 PDP 8 words fill 1032 IBM words

Data Taking Program Sequence

The data recording sequence is best understood by following the flow diagrams on pages 15 to 21.

A flow diagram is provided for each of the "programs" on the program tape. The number above each box indicates the particular statement number that does the operation. The main operation sequence is connected by heavy lines. The listings of the FOCAL program follows the set of flow diagrams. (Page 22 ff.)

Linkage between programs is by "call" statements; for example X CALL(7,2), which would load program 7 from the program DECtape and start the program at the first statement in subroutine 2.

For a simple measurement sequence, the main flow is as follows:

-- Start at program 6, which initializes variables, and reads the program selection switches to determine which option is wanted. (For some continuation options, certain variables are retrieved from the disc, so that the main program can be restarted without losing partially acquired data.) Special information may be obtained via the keyboard.

-- Program 7 allows full count check on scanner and micrometer; loads a survey from DECtape, calculates plate alignment coefficients and stores critical variables at special locations on the disc so that the program can be restarted after any interruption.

-- Program 8 organizes data from the previous image and waits till the photometer motion is done. Then it gets survey input for the next image and calls program 9.

-- Program 9 moves to the next image and starts the automatic measuring sequence. While the photometer is measuring (an interval of about 6 sec.), the data from the previous image is printed out, and stored in a 125 image buffer on the disc. Program 11 is called next.

-- Program 10 carries out a simple search pattern if no image is detected.

-- Program 11 waits till the scanner and micrometer return to home position, reads them and checks for limit conditions or minimum or maximum readings. (Such readings occur if a poor survey coordinate is used or at certain settings where the micrometer travel has to be almost exactly zero or one millimeter.) If such a limiting condition is detected, the stage is moved to recenter the image and program 9 is restarted. Otherwise program 8 is called.

This rather convoluted program sequence is used to speed up the measuring process as much as possible. Calling programs from tape and printing and storing of data is done while waiting for the mechanical motions of the measuring engine. Our objective has been to ensure that the measuring speed limit is due to the mechanical limitations of the measuring machine and never due to delays in the computer operation.

Variable List for the Measuring Program

The names and functions of variables used in the FOCAL program are given below. Future program revisions should take account of these, since some variables are used throughout a whole plate measuring sequence, and a change in value would cause malfunctions. The variables are initialized during the set-up procedure for measuring a plate. Some critical variables are stored on the disc in block 1, words 1--50, so that the program can be restarted in case it is stopped for any reason before a plate is finished.

AS	-	AME in use
AX, AY, BX, BY		Plate alignment offset coefficients

BB	-	Temp store for survey code B	
BI		Survey input	Disc buffers 1st block number
BM		IBM data	Disc buffers 1st block number
BN			Disc buffers 1st block number
BO		Measured data	Disc buffers 1st block number
BR		Sorted survey	Disc buffers 1st block number
BT		Temporary buffer	Disc buffers 1st block number

C	-	Class descriptor
CX, CY		Plate alignment

DA	-	Data ready
DN	-	Continue-measure flag - inhibits first printout

ER	-	Error diagnostic
ER(1)	-	Error diagnostic saved for printing

H	-	(=100)
HM	-	Home test (=2560)

I	-	Input-output data index
IC	-	Image counter 1--125

MX - Maximum limit test (=1280)

NI - Input image count

O - Order descriptor

PH - Photometer reading

Q - Quality descriptor

R - (=40)
RC - Repeat counter
RF - Repeat flag

S - System descriptor
SC - Star code
SI - Survey output image
SN - DECtape storage survey input
SO - Survey output number
SQ - Descriptor S,O,C,Q
SR - Search counter
SS - Survey initialized flag
ST - Survey star code
SV - Star code saved for printing
SX } Survey reading
SY

TO - DECtape output number for data
TM - Time counter

UX }
UY } Constants for position adjust
UZ } (Proper Motion correction for parallax star)

VR - Variable counters

XM · YM - Micrometer reading
XS · YS - Scanner reading
XT · YT - Data for output (total)
XI · YI Input carriage coordinate
XV · YV - Input carriage coordinate for printout

A,B,D,E,J,K,L,M,N,V,X,Y - temporary variables. All variables common to all programs.

Measurement Error Diagnostics

Various abnormal conditions can arise during the measurement, due to poor images, poorly surveyed images, unusual plate background levels, and electronic malfunctions. The computer will in general recenter the image and measure it again, then go on to the next image.

As an aid in diagnosing the cause of measurement problems, a diagnostic code, labelled "ER" is generated and stored with each image measurement. The diagnostic is a 12 bit binary code, each bit representing a particular abnormal occurrence. Each diagnostic code "D" is saved using the "OR" function: SET ER = FOR(ER,D). (An ordinary sum won't work since one diagnostic code could occur several times for a single image.)

The diagnostic code can readily be separated into its component bits by the following code, which uses the "AND" function to test for each binary bit:

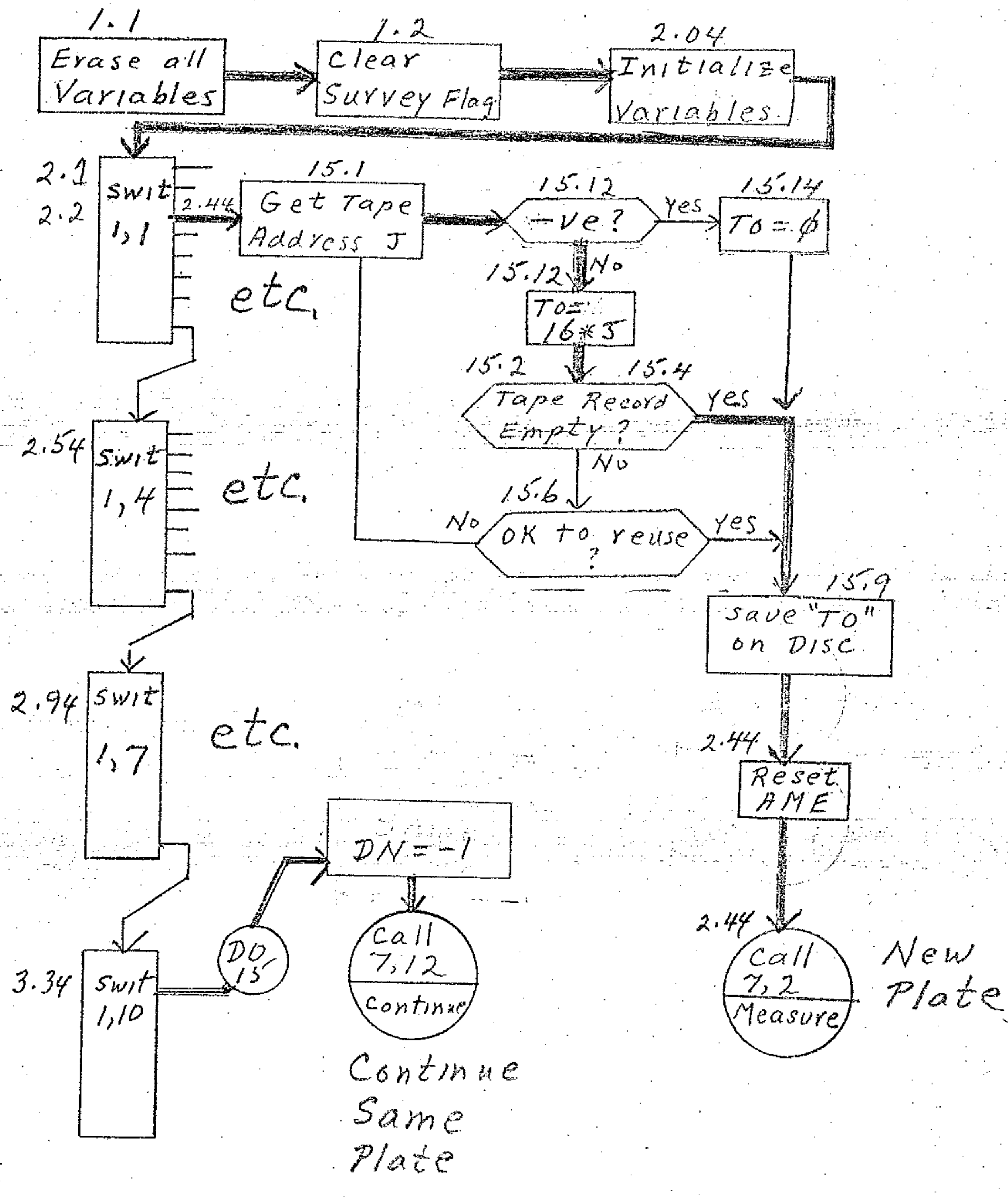
```
1.1  SET K=1

1.2  FOR J=1,12; TYPE  %4  FAND(ER,K); SET K=K*2
```

AME Diagnostic Codes

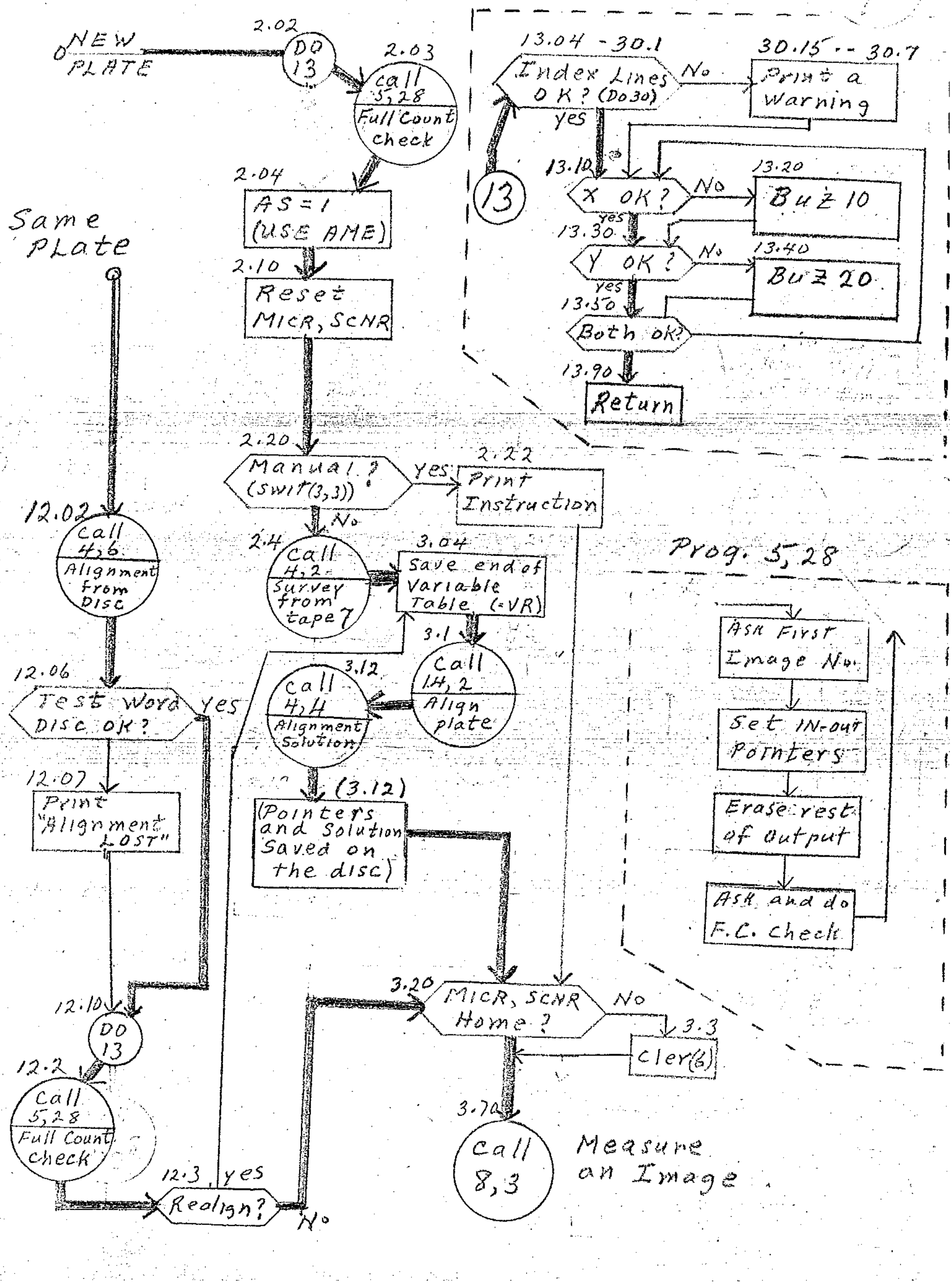
1	-	Impossible photometer reading
2	-	Repeated a measurement
4	-	Searched for the image
8	-	Time limit exceeded
16	-	Reset the micrometer
32	-	Couldn't measure the image
64	-	Reset the scanner
128	-	Close to a limit, recentered the image
256	-	Reset the photometer
512	-	Scanner stuck
1024	-	Micrometer stuck
2048	-	Unused at present (March 1974)

In general, the measurement will have been repeated after any of these diagnostics. Only the final measurement will be stored on the disc.

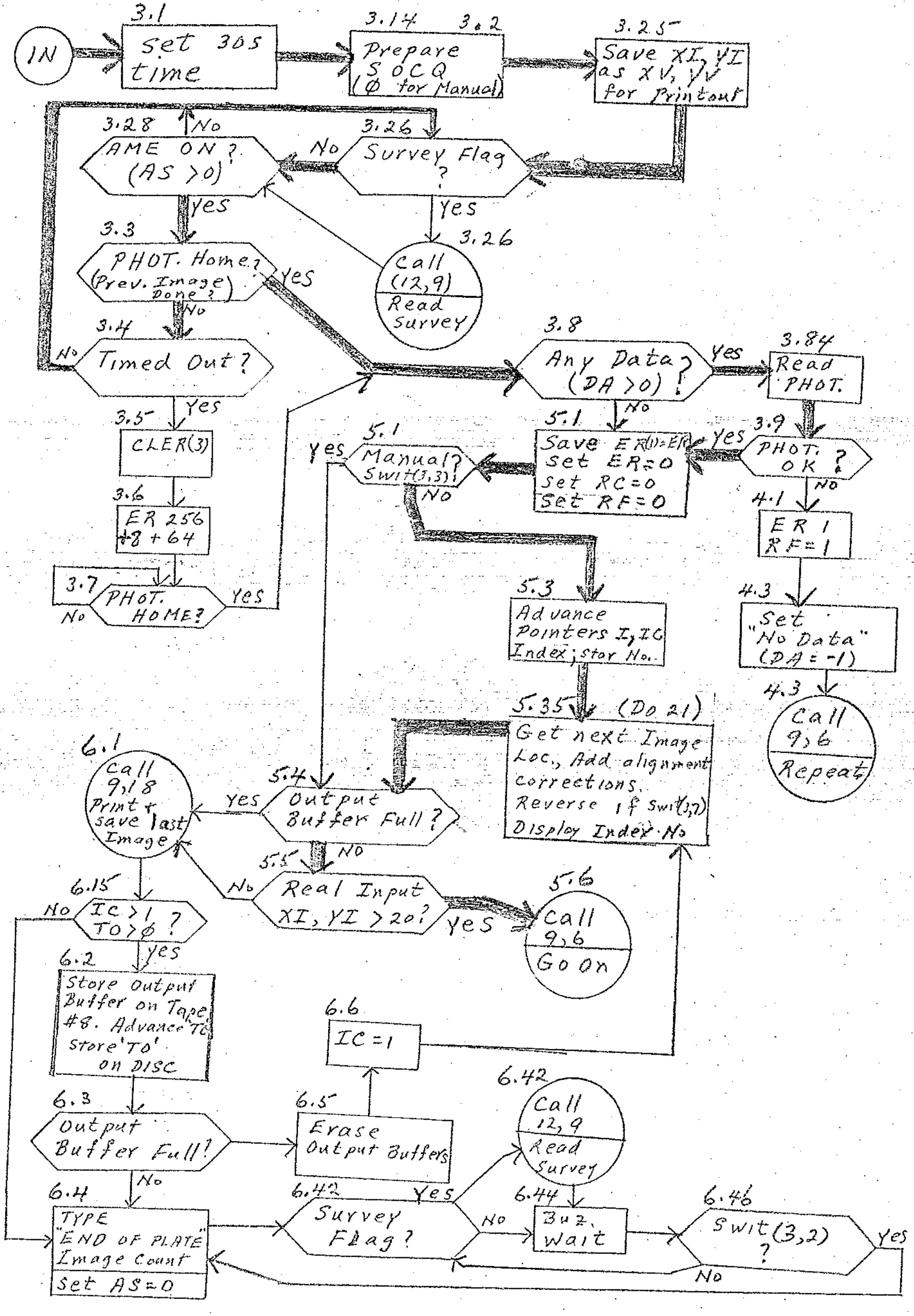
PROG. 6 Dispatcher



PROG. 7 - Initialize

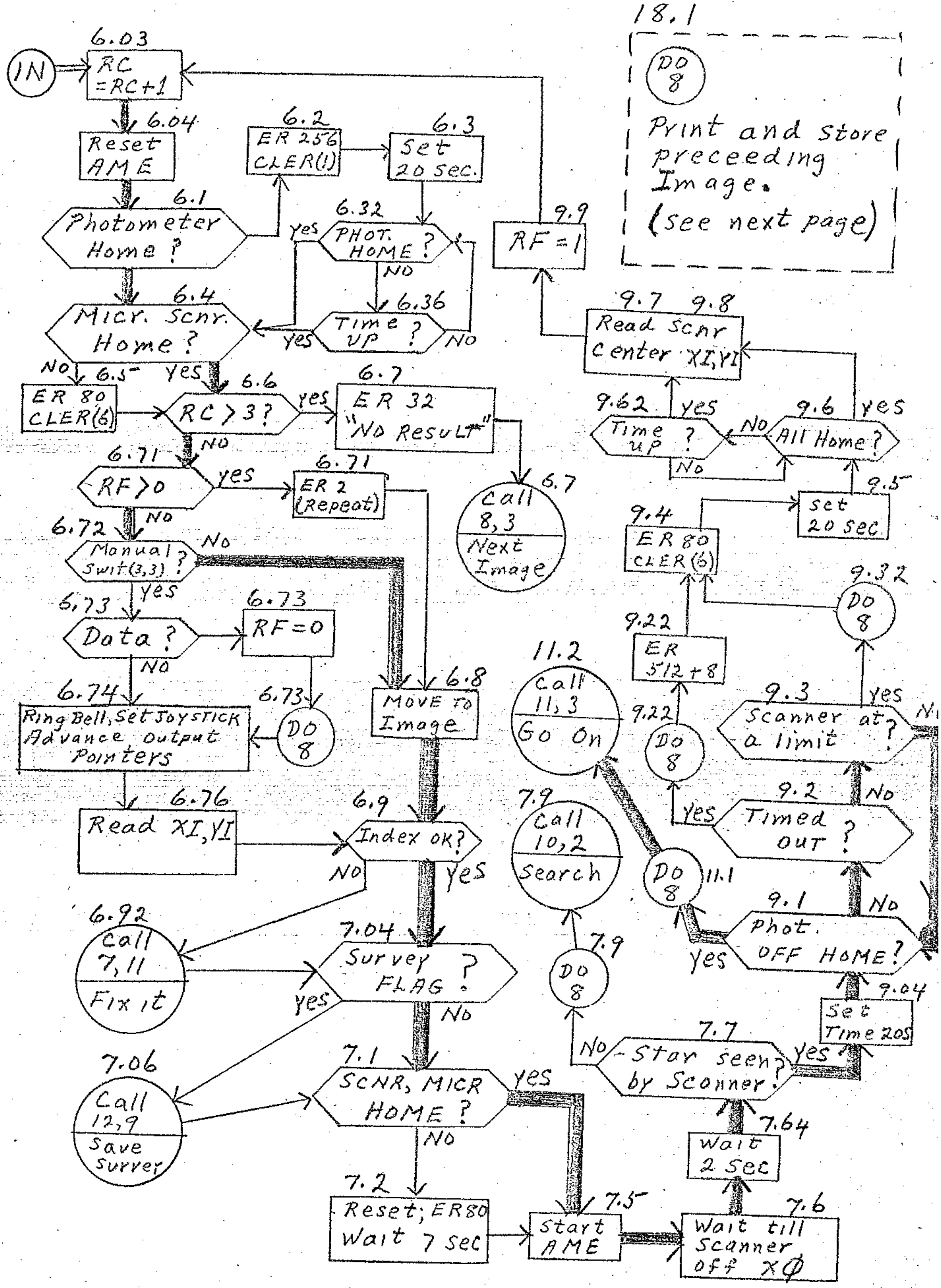


PROG. 8 "Next Image"



PROG. 9 Start Measuring

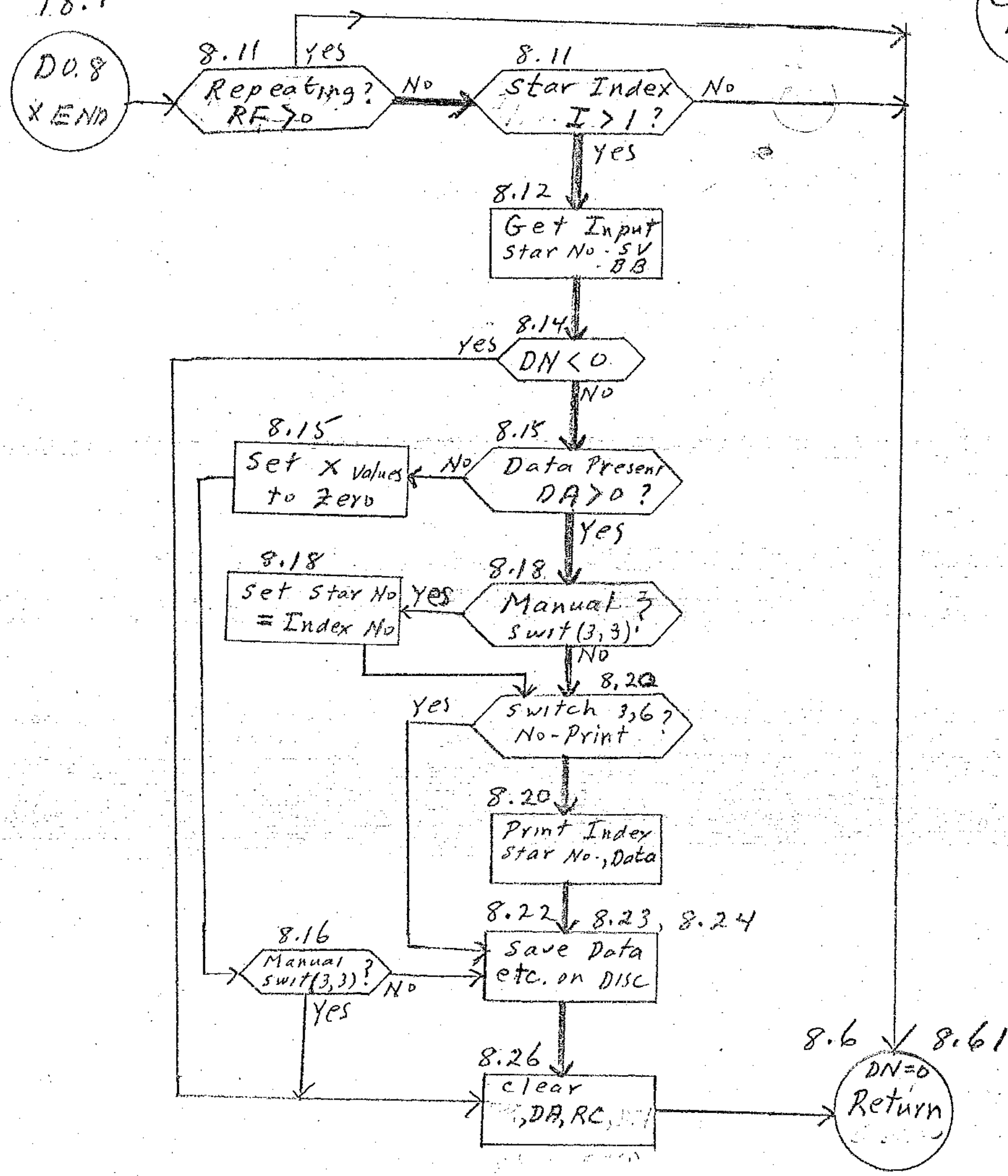
Apr. 19/74



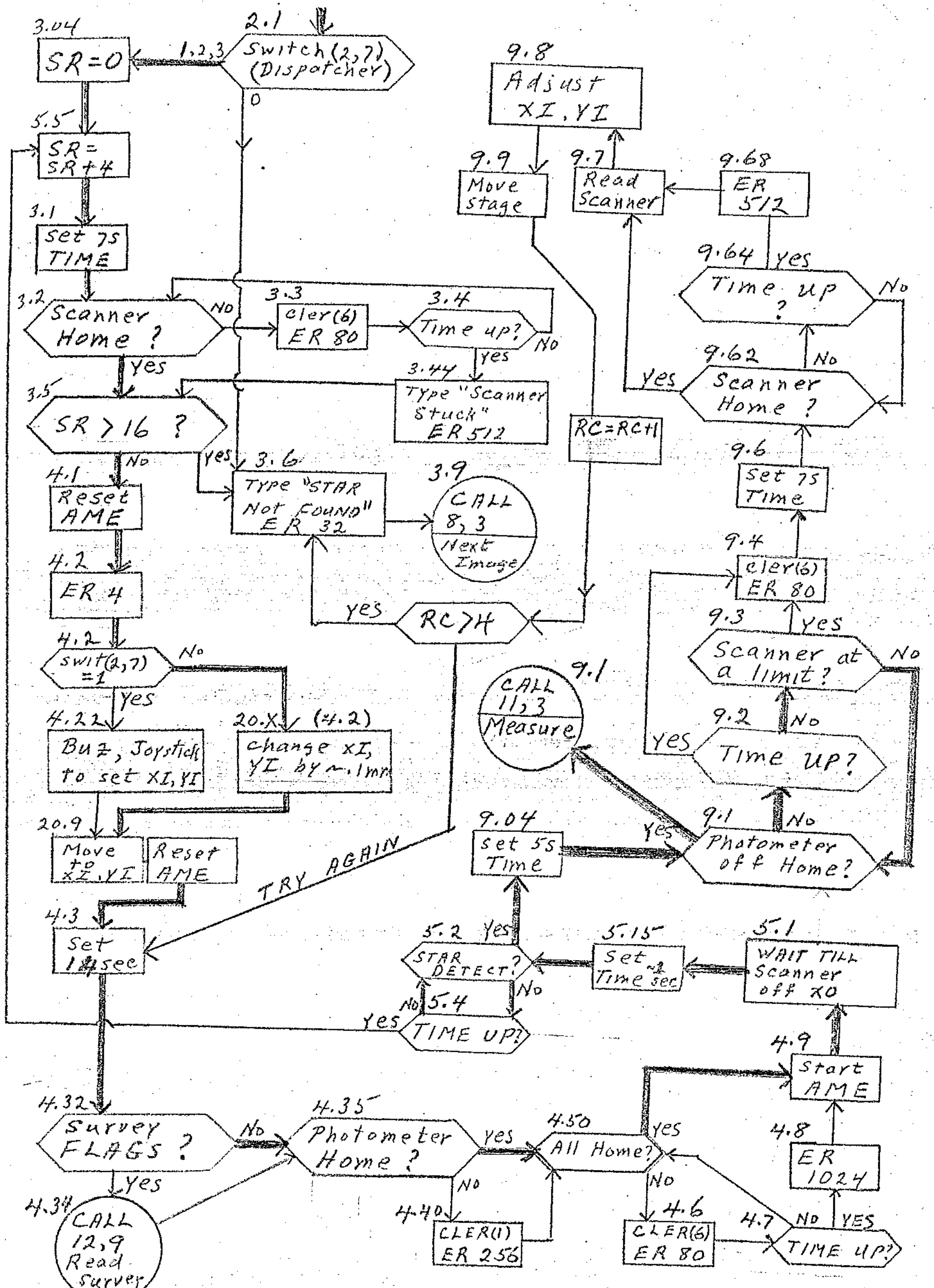
Prog. 9: Part 2 - Print and Save.

9 Part 2

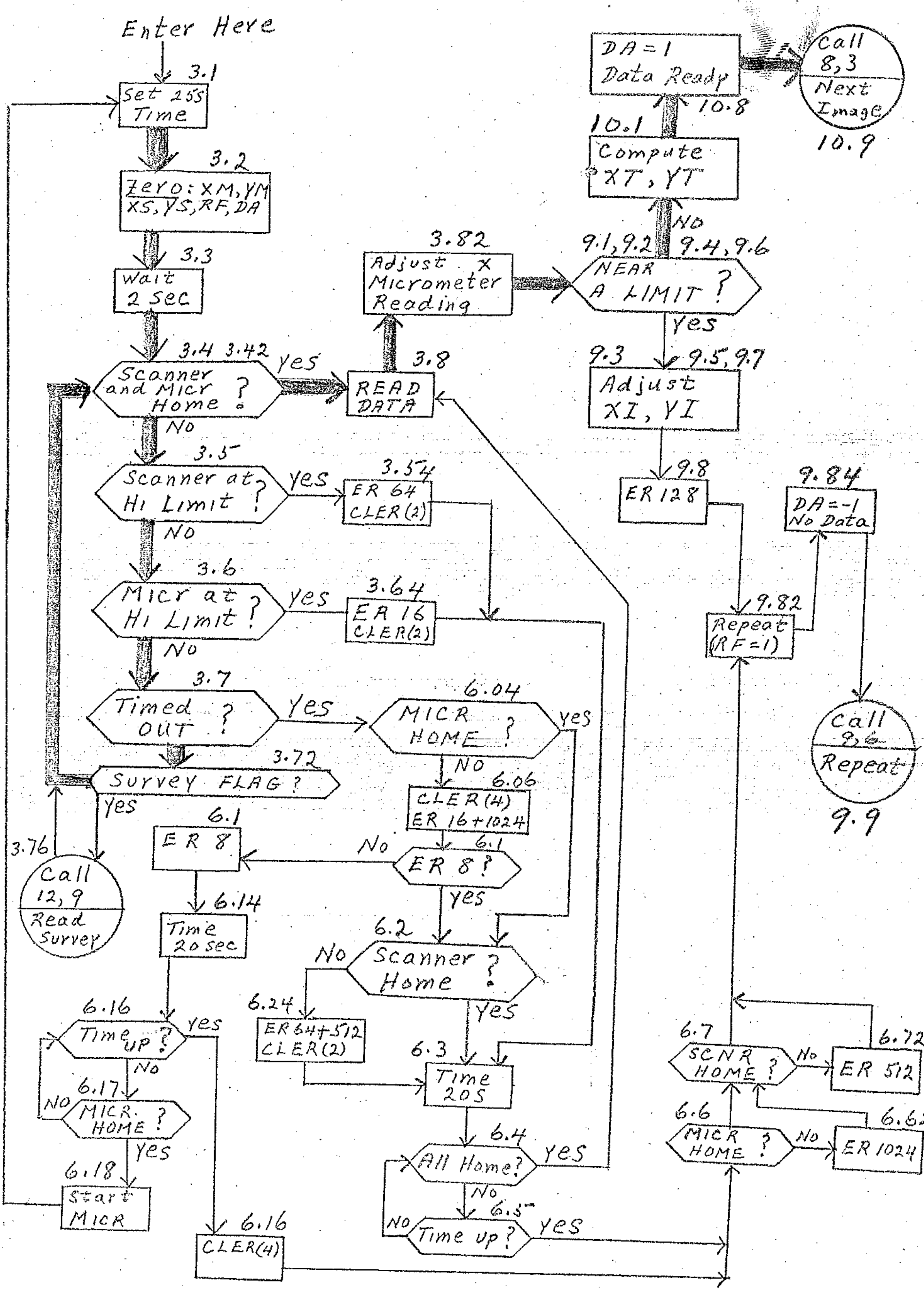
18.1



Search Program PROG:10



PROG. II Measure



Mar 17/74

(22)

```
C:LICK FOCAL AME74-D  UØEØ

Ø1.Ø1 C-PROG 6-DISPATCHER
Ø1.1Ø E
Ø1.2Ø X SWTS(Ø)

Ø2.Ø4 DO 13
Ø2.1Ø S D=FSWIT(1,1)
Ø2.2Ø X GO(2,4Ø+D*2)
Ø2.4Ø X JOY(Ø);C-JOYSTICK
Ø2.41 Q
Ø2.42 X CALL(3,12);C-VIEW BY X,Y
Ø2.44 D 15;X AME(Ø);X CLER(6);X CALL(7,2);C-MEASURE
Ø2.46 X CALL(17,2);C-SIMPLE DIFF CHECK
Ø2.48 X CALL(2Ø,3);C-DIR/REV CHECK
Ø2.5Ø X CALL(18,2);C-TV IMAGE CHECK
Ø2.52 X CALL(3,7);C-CORR AME DATA ON T8
Ø2.54 S D=FSWIT(1,4);X GO(2,8Ø+D*2);C-CONT
Ø2.8Ø X CALL(22,3);C-PRINT FROM T7 OR T8
Ø2.82 X CALL(16,2);C-REPLACE SURVEY BY AME MEAS ON T8
Ø2.84 X CALL(19,4);C-SURVEY-DELETE IMAGE SC
Ø2.86 X CALL(23,5);C-SURV:ADD IMAGE
Ø2.88 X CALL(18,3);C-SURV:TYPE ON T7
Ø2.9Ø X CALL(4,2);C-SURV:T7 TO DISK
Ø2.91 Q
Ø2.92 X CALL(22,2);C-DATA:T8 TO T7
Ø2.94 S D=FSWIT(1,7);X GO(3,2Ø+D*2);C-CONT

Ø3.2Ø X CALL(3,1Ø);C-VIEW BY IMAGE I
Ø3.22 X CALL(5,128*29+6Ø);C-FULL COUNT CHECK
Ø3.23 Q
Ø3.24 X CALL(16,3);C-SURVEY-FROM IBM TO T7
Ø3.26 X CALL(21,2);C-PRINT-OUTPUT DETAILS FROM DISK
Ø3.28 X CALL(21,3);C-PRINT-FINAL RESULTS FROM DISK
Ø3.3Ø S AS=Ø;A "SURVEY ALONE. SET SW(C) AND TYPE C",A,!
Ø3.31 X SWTS(Ø);X CALL(12,9);Q
Ø3.32 X CALL(5,2);C-PRINT PLATE DATA ON T7 OR T8
Ø3.34 S D=FSWIT(1,1Ø);X GO(3,4Ø+D*2);C-CONT
Ø3.4Ø X CALL(5,3);C-LIST PROGRAMS
Ø3.42 D 15;X CALL(7,12);C-MEAS USING SURVEY ON DISK. REPEAT MEAS
Ø3.44 S DN=-1;X CALL(7,12);C-RESTART MEAS AT IMAGE I
Ø3.46 X CALL(21,15);C-READ CONTENTS OF IBM RECORDS
Ø3.48 X CALL(19,2);C-DATA:T7/T8 TO IBM

13.Ø1 C-INITIALIZE
13.1Ø S I=Ø;S A=Ø;S J=Ø;S DA=-1;S IC=Ø
13.2Ø S XI=2ØØ;S YI=2ØØ
13.3Ø S BT=192;S BS=176;S BO=144;S BM=16Ø;S BI=16;S BR=12;S BN=128
13.4Ø S NB=16;S HM=2Ø48+512;S MX=1Ø24+256
13.5Ø S R=4Ø;S H=1ØØ
13.6Ø S AX=Ø;S BX=Ø;S CX=Ø;S AY=Ø;S BY=Ø;S CY=Ø
13.62 S UX=Ø;S UY=Ø;S UZ=Ø
13.99 R
```

Prog. 6 Continued.

```
15.10 A !"REC NO. ON T8 FOR AME OUTPUT ST=",J
15.12 IF (J)15.14;S TO=J*16;G 15.2
15.14 S TO=0;G 15.9
15.20 X MTAK(BT,TO,16)
15.30 S D=0
15.32 F J=0,25,500;S D=FTAK(BT,J)+D
15.34 X MTAK(BT,TO+16,16);DO 15.32
15.40 IF (D)15.5,15.9
15.50 T !"RECORD",%3 TO/16
15.52 A " NOT EMPTY. USE IT ANYWAY?",J
15.60 IF (J-0Y)15.1,15.9,15.1
15.90 X PUT(1,29,TO)
15.91 R

31.98 W
31.99 X END(0)
```

```
C:LICK FOCAL AME74-D  NVG$

01.01 C-PROG 7-INITIALIZE
01.02 X CALL(6,1)

02.01 C-GET SURVEY FROM T7 TO DISK
02.02 DO 13
02.03 X CALL(5,28);C-FCC
02.04 S AS=1
02.10 X AME(0);X CLER(6)
02.20 I (FSWIT(3,3))2.3,2.3;T !"MAN MEAS:AS BELL RINGS USE JOY",!
02.22 T "PRESS SW 3,11 TO MEAS",!; S I=I+1
02.24 GO 3.2
02.30 C
02.40 X CALL(4,2);C-GET SURVEY

03.01 C-ALIGN
03.04 S VR=FVAR(0)
03.10 X CALL(14,2)
03.12 X CALL(4,4)
03.20 IF (FSCNS(HM)+FMICS(HM)-2*HM)3.3,3.7
03.30 X CLER(6)
03.70 X CALL(8,3)

10.10 X CALL(5,28);C-FCC
10.20 Q

11.10 DO 13
11.90 X END(0)

12.01 C-MEAS USING SURVEY ON DISK. REPEAT PLATE.
12.02 X CALL(4,6)
12.06 I (D-1357)12.07,12.10
12.07 T !"ALIGNMENT LOST!"
12.10 DO 13
12.20 X CALL(5,28)
12.30 A !"REALIGN PLATE?",J
12.32 IF (J-0Y)12.4,3.01
12.40 G 3.2

13.01 C-CHECKING INDEX OK
13.04 DO 30
13.10 IF (-FPHOS<8>)13.3
13.20 X BUZ(10);F J=0,10;S A=A
13.30 IF (-FPHOS(4))13.5
13.40 X BUZ(20);F J=0,10;S A=A
13.50 IF (FPHOS(12)-12)13.6,13.9
13.60 G 13.1
13.90 R

30.10 IF (FPHOS(8)-8)30.15,30.6
30.15 T !!"CHECK 'AME DISABLED' SWITCH",!!
30.20 T !"X INDEX OFF"
30.60 IF (FPHOS(4)-4)30.7,30.9
30.70 T !"Y INDEX OFF"
30.90 R

31.98 W
31.99 X END(0)
```

```
C:LICK FOCAL AME74-D  SPR9

01.01 C-PROG.8;APR.18/74.
01.02 X CALL(6,1)

03.10 S TM=-130;C-30 SEC
03.14 S SQ=0;IF (-FSWIT<3,3>)3.25;C-MANUAL
03.20 S SQ=1000*FIN(BI,I,4)+100*FIN(BI,I,5)+FIN(BI,I,6);C-IMAGE 'I'
03.25 S XV=XI;S YV=YI
03.26 IF (FSURV(-1)+FSWTS(2,1))3.28,3.28;X CALL(12,9);C-SURVEYING
03.28 IF (AS)3.26,3.26
03.30 IF (FPHOS(704)-512)3.4,3.8
03.40 S TM=TM+1;IF (TM)3.26
03.50 X CLER(3)
03.60 S ER=FOR(ER,256+64+8)
03.70 IF (FPHOS(704)-512)3.7,3.8,3.7
03.80 IF (DA)5.1,5.1
03.84 S PH=FPHOR(0)+2532
                          03.90 IF (PH-2600)4.1;IF (5000-PH)4.1
03.94 G 5.1

04.10 S ER=FOR(ER,1);S RF=1
04.30 S DA=-1;X CALL(9,6);C-REPEAT FOR PHOTOMETER ERROR

05.10 S ER(I)=ER;S ER=0;S RF=0;S RC=0;IF (-FSWIT(3,3))5.4
05.30 S I=I+1;S IC=IC+1
05.35 D 21
05.40 IF (IC-126)5.5;G 6.1;C-END OF OUTPUT BLOCK
05.50 IF (XI-20)6.1,6.1;IF (YI-20)6.1,6.1,5.6
05.60 X CALL(9,6)

06.01 C-STORE DATA ON T8(AT TO) FROM DISK(FROM BM)
06.10 X CALL(9,18);C-PRINT LAST DATA
06.15 IF (TO)6.4,6.4;IF (IC-2)6.4
06.20 X MPUT(BM,TO,16);S TO=TO+16;X PUT(1,29,TO)
06.30 IF (IC-126)6.4,6.5,6.5
06.40 T !"END OF PLATE",%4 I-1," IMAGES RECORDED.";S AS=0
06.42 IF (FSURV(-1)+FSWTS<2,1>)6.44,6.44;X CALL(12,9)
06.44 X BUZ(30);F J=0,200;S A=A;S A=FSWIT(3,2)
06.46 IF (A)6.42,6.42,6.40
06.50 X PUTN(BO,40,0,2024);X PUTN(BM,40,0,2024)
06.60 S IC=1;G 5.35

21.01 C-GO TO NEXT SURVEY IMAGE D
21.04 S D=FTAK(BR,I)
21.06 IF (D)21.08,21.08,21.1
21.08 S XI=0;S YI=0;S SC=0
21.09 R
21.10 S SC=FIN(BI,D,1)
21.20 S X=FIN(BI,D,2)/100;S Y=FIN(BI,D,3)/100;I (X)21.30,21.30
21.21 I (UZ)21.22,21.22;I (SC-98)21.22;S X=X+UX;S Y=Y+UY
21.22 IF (FSWIT(3,7))21.3,21.3
21.24 S X=500-X;S Y=500-Y
21.30 S XI=X+AX*<X-250>+BX*<Y-250>+CX
21.40 S YI=Y+AY*<X-250>+BY*<Y-250>+CY
21.60 X PUT(1,30,I);X PUT(1,31,IC)
21.80 X SWIT(0,I)
21.90 R

31.98 W
31.99 X END(0)
*
```

```
        W
C:LICK FOCAL AME74-D  UCSP

01.01 C-PROG 9-APR. 18/74.
01.02 X CALL(6,1)

06.01 C-REPEATER
06.03 S RC=RC+1
06.04 X AME(0);X CLER(6)
06.10 IF (FPHOS(704)-512)6.2,6.4
06.20 X CLER(1);S ER=FOR(ER,256)
06.30 S TM=-600
06.32 IF (FPHOS(704)-512)6.36,6.4
06.36 S TM=TM+1;IF (TM)6.32
06.40 D 30;IF (D)6.5,6.6
06.50 X CLER(6);S ER=FOR(ER,80)
06.60 IF (RC-4)6.71;CHECK REPEAT COUNT
06.70 S ER=FOR(ER,32);T !"NO RESULT:STAR",%3 I;X CALL(8,3)
06.71 IF (RF)6.72,6.72;S ER=FOR(ER,2);G 6.8
06.72 I (-FSWIT(3,3))6.73,6.8
06.73 I (DA)6.74;S RF=0;D 8
06.74 T "";X JOY(0);S I=I+1;S IC=IC+1
06.76 S XI=FLOC(0)/40;S YI=FLOC(1)/40;G 6.9
06.80 DO 23
06.90 S D=12-FPHOS(12);IF (D)6.92,7.04
06.92 X CALL(7,11)

07.04 IF (FSURV(-1)+FSWTS(2,1))7.10,7.10;X CALL(12,9);C-SURVEY
07.10 DO 30.1;IF (D)7.2,7.5
07.20 X CLER(6);S ER=FOR(ER,80);F J=0,1000;S A=A;C-7 SEC
07.50 X AME(1)
07.60 IF (-FSCNS(2048))7.6
07.64 F J=0,200;S A=A
07.70 IF (FSCNS(16)-16)7.9,9.04
07.90 DO 8;X CALL(10,2);C-SEARCH

08.10 C-SAVE IMAGE DATA
08.11 IF (-RF)8.6;I (I-2)8.6
08.12 S D=FTAK(BR,I-1);S SV=FIN(BI,D,1);S BB=FIN(BI,D,8)
08.14 IF (DN)8.26;C-DON'T STORE
08.15 IF (-DA)8.18;S XM=0;S XV=0;S XS=0;S XT=0
08.16 IF (-FSWIT<3,3>)8.26;G 8.22
08.18 IF (FSWIT(3,3))8.2,8.2;S SV=I-1
08.20 I (-FSWIT(3,6))8.22;T !;T %4 I-1,SV,%9.04 XT;D 8.99
08.22 X SAVX(BO,IC-1,SV,XV*100,XM,XS)
08.23 X SAVY(BO,IC-1,PH,YV*100,YM,YS)
08.24 X SAVX(BM,IC-1,SV,XT*10000,YT*10000,BB);X SAVY(BM,IC-1,0,PH,SQ,ER)
08.26 S DA=-1;S RC=0
08.60 S DN=0
08.61 R
08.99 T YT,%4 PH,ER(1),SQ
```

(CONT'D next page)

Prog. 9 Cont'd.

```
09.04 S TM=-500;C-20 SEC
09.10 IF (FPHOS(512))11.1,11.1
09.20 S TM=TM+1;IF (TM)9.3
09.22 DO 8;S ER=FOR(ER,512+8);G 9.4
09.30 IF (FSCNS(3840))9.32,9.1
09.32 DO 8
09.40 X CLER(6);S ER=FOR(ER,80)
09.50 S TM=-100
09.60 DO 30;IF (D)9.62,9.7
09.62 S TM=TM+1;IF (TM)9.6
09.70 S X=FSCNR(0);S Y=FSCNR(1)
09.80 S XI=XI-.12+X/6000;S YI=YI-.12+Y/6000
09.90 S RF=1;G 6.03

11.10 DO 8
11.20 X CALL(11,3)

18.10 DO 8
18.20 X END(0)

23.10 F J=0,1;X MOV(XI*100,YI*100)
23.90 R

30.10 S D=FSCNS(HM)-HM+FMICS(HM)-HM

31.98 W
31.99 X END(0)
*
```

```
C:LICK FOCAL AME74-D  RBQS

01.01 C-PROG.10;FEB.27/74-SEARCH
01.02 X CALL(6,1)

02.01 C-CHOOSE IMAGE SEARCH MODE AT 2.10
02.10 S D=FSWIT(2,7)
02.12 X GO(2,20+D*2)
02.20 G 3.60
02.22 G 3.04
02.24 G 3.04
02.26 G 3.04

03.04 S SR=0;G 5.5
03.10 S TM=-20;C-7 SEC.
03.20 IF (FSCNS(HM)-HM)3.3,3.5
03.30 X CLER(6);S ER=FOR(ER,80)
03.40 S TM=TM+1;IF (TM)3.2
03.44 T !"SCANNER STUCK";S ER=FOR(ER,512)
03.50 IF (SR-17)4.1
03.60 T !"STAR",%3 I," NOT FOUND";S ER=FOR(ER,32)
03.90 X CALL(8,3)

04.10 X AME(0);X CLER(6)
04.20 S ER=FOR(ER,4);I (1-FSWIT(2,7))4.22;X GO(20,SR)
04.22 X BUZ(50);X JOY(0);S XI=FLOC(0)/40;S YI=FLOC(1)/40;G 20.90
04.30 S TM=-50;C-14 SEC
04.32 IF (FSURV(-1)+FSWTS(2,1))4.35,4.35;X CALL(12,9);C-SURVEY
04.35 IF (FPHOS(704)-512)4.4,4.5
04.40 X CLER(1);S ER=FOR(ER,256)
04.50 IF (FSCNS(HM)-HM+FMICS(HM)-HM+FPHOS(704)-512)4.6,4.9
04.60 X CLER(6);S ER=FOR(ER,80)
04.70 S TM=TM+1;IF (TM)4.5
04.80 S ER=FOR(ER,1024)
04.90 X AME(1)

05.10 IF (-FSCNS(2048))5.1
05.15 S TM=0
05.20 IF (FSCNS(16)-16)5.4,9.04,5.4
05.40 S TM=TM+1;IF (TM-45)5.2
05.50 S SR=SR+4;G 3.1

09.04 S TM=300;C-5 SEC
09.10 IF (-FPHOS(512))9.2;X CALL(11,3)
09.20 S TM=TM-1;IF (TM)9.4
09.30 IF (FSCNS(3840))9.4,9.1
09.40 X CLER(6);S ER=FOR(ER,80)
09.60 S TM=300;C-7 SEC
09.62 IF (FSCNS(HM)-HM)9.64,9.7
09.64 S TM=TM-1;IF (TM)9.68;G 9.62
09.68 S ER=FOR(ER,512)
09.70 S X=FSCNR(0);S Y=FSCNR(1)
09.80 S XI=XI-.12+X/5000;S YI=YI-.12+Y/5000
09.90 DO 23;[crossed out]      S RC=RC+1; IF (RC-5)4.3,3.6,3.6

20.04 S XI=XI+.06;S YI=YI+.06;G 20.9
20.08 S XI=XI-.12;G 20.9
20.12 S YI=YI-.12;G 20.9
20.16 S XI=XI+.12
20.90 X AME(0);X CLER(6);D 23
20.99 GO 4.3

23.10 F J=0,1;X MOV(XI*100,YI*100)
23.90 R

31.98 W
31.99 X END(0)
```

```
C:LICK FOCAL AME74-D  TGS:

Ø1.Ø1 C-PROG.11-APR. 2/74.
Ø1.Ø2 X CALL(6,1)

Ø3.1Ø S TM=6Ø;C-25 SEC.
Ø3.2Ø S RF=Ø;S XM=Ø;S YM=Ø;S XS=Ø;S YS=Ø;S PH=Ø;S DA=Ø
Ø3.3Ø F J=Ø,4ØØ;S A=A
Ø3.4Ø S D=FSCNS(HM)+FMICS(HM)-2*HM
Ø3.42 IF (D)3.5,3.8,3.5
Ø3.5Ø IF (FSCNS(MX))3.54,3.6
Ø3.54 X CLER(2);S ER=FOR(ER,64);G 6.3
Ø3.6Ø IF (FMICS(MX))3.64,3.7
Ø3.64 X CLER(4);S ER=FOR(ER,16);G 6.3
Ø3.7Ø S TM=TM-1;IF (-TM)3.72;G 6.Ø4
Ø3.72 IF (FSURV<-1>+FSWTS<2,1>)3.4,3.4
Ø3.74 X CALL(12,9)
Ø3.76 G 3.4
Ø3.8Ø S XM=FMICR(Ø);S YM=FMICR(1);S XS=FSCNR(Ø);S YS=FSCNR(1)
Ø3.82 S XM=XM*Ø.9943;C-LICK MICR SCALE CORR
Ø3.9Ø GO 9.1

Ø4.Ø1 C- FINAL RESULT
Ø4.2Ø S D=XI-FITR(XI);S M=XM;DO 14
Ø4.22 S XT=XI+N
Ø4.3Ø S D=YI-FITR(YI);S M=YM;D 14
Ø4.32 S YT=YI+N
Ø4.38 C-GAERTNER CORR
Ø4.4Ø S XT=FITR(XT)+(XM+XS)/5ØØØ+FCOR(XI)/1ØØØØ
Ø4.5Ø S YT=FITR(YT)+(YM+YS)/5ØØØ+FCOR(YI,1)/1ØØØØ
Ø4.9Ø R

Ø6.Ø4 IF (FMICS<HM>-HM)6.Ø6,6.2
Ø6.Ø6 X CLER(4);S ER=FOR(ER,16+1Ø24)
Ø6.1Ø IF (-FAND<ER,8>)6.2;S ER=FOR(ER,8)
Ø6.14 S TM=-5ØØ
Ø6.16 S TM=TM+1;IF (TM)6.17;X CLER(4);G 6.6
Ø6.17 IF (FMICS<HM>-HM)6.16,6.18,6.16
Ø6.18 X GMIC(Ø);G 3.1
Ø6.2Ø IF (FSCNS(HM)-HM)6.24,6.3
Ø6.24 X CLER(2);S ER=FOR(ER,64+512)
Ø6.3Ø S TM=1ØØ;C-2Ø SEC
Ø6.4Ø DO 3.4;IF (D)6.5,3.8
Ø6.5Ø S TM=TM-1;IF (-TM)6.4
Ø6.6Ø IF (FMICS(HM)-HM)6.62,6.7
Ø6.62 S ER=FOR(ER,1Ø24)
Ø6.7Ø IF (FSCNS(HM)-HM)6.72,9.82
Ø6.72 S ER=FOR(ER,512);G 9.82

Ø9.1Ø IF (XS-5Ø)9.3;IF (12ØØ-XS)9.3
Ø9.2Ø IF (YS-5Ø)9.3;IF (12ØØ-YS)9.3;G 9.4
Ø9.3Ø S XI=XI-.12+XS/5ØØØ;S YI=YI-.12+YS/5ØØØ;G 9.8
Ø9.4Ø IF (XM-5Ø)9.5;IF (XM-495Ø)9.6
Ø9.5Ø S XI=XI+(1+RC)*.Ø25*FSGN(XS-6ØØ);GO 9.8
Ø9.6Ø IF (YM-5Ø)9.7;IF (YM-495Ø)1Ø.1
Ø9.7Ø S YI=YI+(1+RC)*.Ø25*FSGN(YS-6ØØ)
Ø9.8Ø S ER=FOR(ER,128)
Ø9.82 S RF=1
Ø9.84 S DA=-1
Ø9.9Ø X CALL(9,6);C-REPEAT

1Ø.1Ø DO 4
1Ø.8Ø S DA=1
1Ø.9Ø X CALL(8,3);C-ALL OK

14.1Ø S N=Ø
14.2Ø IF (D-.9)14.3;IF (1ØØØ-M)14.9;S N=1;G 14.9
14.3Ø IF (.3-D)14.9;IF (M-3ØØØ)14.9;S N=-1
14.9Ø R

31.98 W
31.99 X END(Ø)
```

C:LICK FOCAL AME74-D U+G

```
01.01 C-PROG 12-SURVEY HANDLER
01.02 X CALL(6,1)

09.01 C-SURV
09.04 S BB=FTAKL(BS,28);S ST=FTAKL(BS,30);S SI=FTAKL(BS,32)
09.05 S S0=FTAK(1,46)
09.10 I (FSWTS(3,240)/16-1)9.12,9.14,9.16
09.12 X SWTS(0);X DISP(ST);G 9.92
09.14 D 10.1;G 9.1
09.16 I (FSWTS(3,240)/16-3)9.18,9.20,9.22
09.18 X CALL(13,3,1);C-START:NEW
09.19 D 10.1;G 9.1
09.20 X CALL(13,4,1);C-START:INC
09.21 D 10.1;G 9.1
09.22 I (FSWTS(3,240)/16-5)9.23,9.24,9.25
09.23 S AS=1;G 9.27;C-SURV:AME ON
09.24 S AS=0;G 9.27;C-SURV:AME OFF
09.25 X CALL(13,2,1);C-CONT
09.26 D 10.1;G 9.1
09.27 G 10.3;C-STARTS SURVEY
09.28 S SX=FSURV(0)/20;S SY=FSURV(1)/20
09.29 S S=FSWTS(2,4)/4+1;S O=FSWTS(2,56)/8
09.30 S C=FSWTS(2,448)/64;S Q=FSWTS(2,3584)/512
09.31 S B=FSWTS(3,12)/4; S E=FSWTS(3,3);G 10.6
09.32 IF (SX)9.34,9.34;IF (SY)9.34,9.34,9.36
09.34 X DISP(0);DO 10.2;G 9.6;C-ERROR IN DATA
09.36 S SI=SI+1;I (FSWTS(2,2))9.40,9.40
09.37 S ST=ST+1
09.40 F J=0,100;S A=A
09.42 IF (FSWTS<2,2>-2)9.44,9.46
09.44 S ST=FSWTS(1);C-MANUAL
09.46 I (O)9.49,9.49;S SX=SX-GX*O;S SY=SY-GY*O;S B=1;G 9.5
09.47 S SX=SX+GX*O*2;S SY=SY+GY*O*2;S B=2;S SI=SI+1;G 9.5
09.49 S B=0
09.50 X SAVX(BS,SI,ST,SX*100,SY*100,S);X SAVY(BS,SI,O,10*C+Q,E,B)
09.51 X DISP(ST)
09.52 I (FSWIT(3,5))9.58,9.58
09.54 T !%3 SI,ST,%5.02 SX,SY,%1 S,O,C,Q,E,B
09.58 I (B-1)9.6,9.47,9.6
09.60 IF (FSWTS(2,1))9.82,9.82
09.62 IF (-SI)9.66
09.64 S SI=0;S ST=0;G 9.82
09.66 X SAVX(BS,SI);X SAVY(BS,SI)
09.70 D 9.4
09.74 S SI=SI-1
09.80 S ST=FIN(BS,SI,1)
09.82 X SWTS(0);X PUTL(BS,28,BB);X PUTL(BS,30,ST);X PUTL(BS,32,SI)
09.84 X DISP(ST)
09.86 I (-FSWIT(3,2))9.86,9.92,9.92
09.92 IF (-AS)9.96;G 9.1;C-AME OFF
09.96 X END(0)
```

Prog. 12 Cont'd.

```
10.10 A "SW(C)",A,!
10.20 F J=0,10;X BUZ(10);F K=0,20;S A=A
10.30 I (FSURV(-1))9.6,9.6
10.32 I (124-SI)10.34,9.28,9.28
10.34 S BB=BB+SI;X PUTL(BS,28,BB);X PUTL(BS,30,ST);X PUTL(BS,32,SI)
10.38 X MPUT(BS,16*SO,16,8)
10.40 X PUTN(BS,40,0,2024)
10.42 S SI=0;S SO=SO+1
10.44 X PUT(1,46,SO);G 9.86
10.60 I (B-1)10.62,10.63,10.64
10.62 S GX=0.0;S GY=0.0;G 9.32
10.63 S GX=0.0;S GY=0.250;G 9.32
10.64 S GX=0.575;S GY=0.0;G 9.32

31.98 W
31.99 X END(0)
```

Focal Language Control Commands

The special control commands used to operate the measuring system are in the form:

SET D = FABCD(ARG3,ARG4, . . . ARG10). These instructions return a floating point number to variable D. When only an operation but no resultant number is needed, the command can be in the form:

X ABCD(ARG3, . . .)

A list of the special commands follows.

Many of the control instructions to the Automatic Measuring Engine will seem a bit incomprehensible due to the use of numerical codes in the FOCAL arguments to define and mask specific bits of the machine language instruction code.

The numerical codes used in the FOCAL commands are for the most part directly converted into one or more bits in the machine language code. Thus the command SET D = FPHOS(512) will read the photometer status, word, masked (or "ANDED") by the 2^9 bit in the PDP 8 accumulator.

SET D = FPHOS(704) would give the status masked by bits $2^9 = 512$, $2^7 = 128$ and $2^6 = 64$. (512 + 128 + 64 = 704). If only the 2^7 bit was true in the photometer status, D would be set to the value 128.

The significance of the various status bits are set out in the Status Coding Table. (P42)

FUNCTION LIST: "LICK FOCAL" SUMMARY

Set D = FITR(N)	integer value of N. (D is set equal to integer value of N)
Set D = FLOG(N)	log N (D is set equal to log N to the base e)
Set D = FSIN(N)	sine N
Set D = FCOS(N)	cosine N
Set D = FEXP(N)	exponential e^N
Set D = FSGN(N)	sign of N
Set D = FABS(N)	absolute value of N
Set D = FSQT(N)	square root of N
Set D = FTAK(B,W)	get single precision value of word W in disk block B.**
Set D = FASK(B,W)	get 10 digit floating format variable starting at disk word W, block B. (4 words used) - See X STOR( )

A special command "X" (execute) can be used for functions which need not return a number to FOCAL:

X PUT(B,W,I)	Store integer I* in disk word W, block B.**
X STOR(B,W;V)	Store variable V starting at disk word W (Note semicolon).
X GO(S,L) X DO(S,L)	Like ordinary GO,DO but with computed arguments. (Subroutine S, line L.)
X FILE(N)	File program N on DECtape.
X CALL(N,S,Q)	Call program N, start at subroutine S (if S>0). If Q = 1, calls can be nested to 10 levels. Nesting list is cleared for Q = 0.
X CALL(N,S*128 + L)	Start at line L, subroutine S, Program N.
X CALL(N)	Call program N, don't start.
X END(Ø)	Return to calling program; next line.
X SHFT(B,N)	Move disk block B to an address N words higher. N < 2048
~~X PEN(X,Y)~~	~~Move chart recorder X steps, then move pen to Y.~~
X SHFT(B,-N)	Move disk block B to an address N words lower. N < 2048

X MPUT(D,T,N,U) — Copies from Disk to Tape
X MTAK(D,T,N,U) — Copies from Tape to Disk

- D — First disk block
- T — First tape block
- N — Number of blocks to transfer
- U — Tape Unit # (Ø and 8 are the same unit)

Disk blocks 213 to 225 are changed. Atempts to treat disk blocks above block 210 will produce a diagnostic "DISK END" with these 2 instructions.

**If B=W=Ø, the previously used disk address will be incremented and taken as the current disk address.

*Integers can have values $0 \le I \le 4095$

STORAGE AND RETRIEVAL OF DISC DATA

X PUTL(B,W,V) — Store variable as a double precision integer on the disc
- B — Block number
- W — First Word number, (two words used)
- V — Variable

SET D = FTAKL(B,W) — retrieve double precision integer from the disk

Set D = FIN(B,I,J) — Get Survey input from Disc
- B — First Disc Block of Survey Table
- I — Star number in Table
- J:

1:Star Code	5:order
2:X Carriage	6:class *10 +Quality
3:Y Carriage	7:modify survey
4:System	8:grating image

Data Output Format for PDP 8 Analysis

SET D = FOUT(B,I,J) — get Image Descriptor from DISC
- B — First Disk Block of Table
- I — Star Position in Table
- J — Segment of Descriptor

Segments: J =

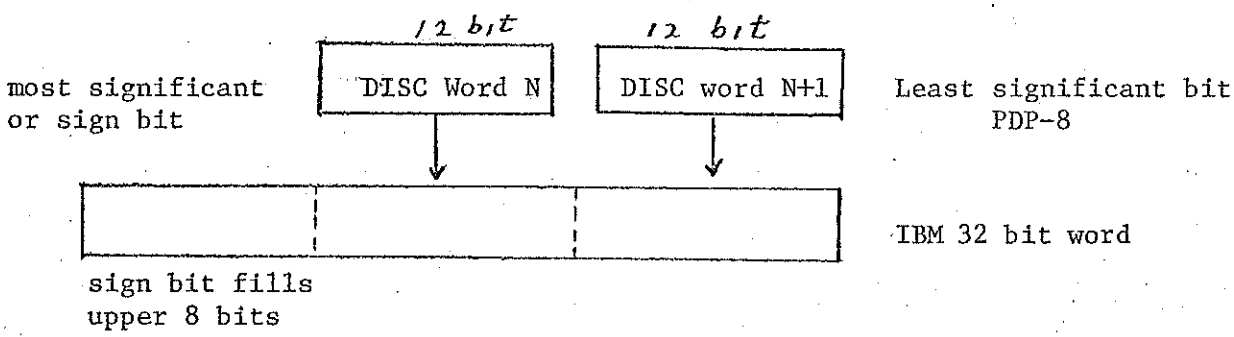
1-STAR CODE,	5-photometer
2-X Carriage	6-Y Carriage
3-X Micr.	7-Y Micrometer
4-X Scanner	8-Y Scanner

X SAVX(B,I,SC,XI,XM,XS) — Save X output on Disc
- B — First Disk Block of Table
- I — Star Position in Table
- SC — Star Code No.
- XI — Carriage Position
- XM — Micrometer
- XS — Scanner

X SAVY(B,I,PH,YI,YM,YS) — Save Y output on Disc
- PH — Photometer

IBM Tape for AME

Double precision words are transferred between the DISC and IBM tape. Two 12 bit words are used to form a 32 bit IBM word



SET D = FADV(N,J) or X ADV(N,J) space forward N records.

SET D = FBAK(N,J) or X BAK(N,J) space back N records.

If J = 0, stop at first file mark and space forward over it.
D is the count of bad records found

X IBME(0) Erase about 4 feet of tape

X EOF(0) Write an end of file mark

X RWND(0) Rewind to BOT (beginning of tape mark)

X HUNT(0) Move forward until a double file mark is passed.

```
                          First Disc Block
                          First Disc Word
                          Double word count (uses 516)
                          Use 3 bytes per word if non zero
                          No disk if non-zero
SET D = FIBMR(B,W,N,Ø, S,Q)    or X IBMR(    ) read

SET D = FIBMW(B,W,N,Ø, Q)      or X IBMW(    ) write
```

POSITIONING and DISPLAY

X ALOC(X,Y) - Displays integers X and Y on the position display lamps

X DISP(N) - Display N as Star No. in Survey Box

X JOY(0) Allow motion control using the joystick. Press switch 3, 11 when done; press switch 3, 12 for high speed.

X MOV (X*100, Y*100) Move carriage to position X, Y mm.

(not used)

X STEP(NX,NY,TX,TY,P,M) Move carriage with N-drive pulses.

- NX, NY — Pulse count (±for direction)
- TX, TY — Pulse width
- P — Pulse period (uses 4095 if Ø)
- M — Leave carriage moving

X STAT(X,Y,S) Set output print to CRT

- X, Y — Location of first character on CRT
- S — Letter size

Device Control Commands

X AME(N) - Start measurement --N=1
STOP All (General Reset) -- N=Ø

X CLER(M) Select and reset device(s)
- 1-photometer
- 2-Scanner
- 4-micrometers

Micrometer

SET D = FMICR(N) Read Data
*SET D = FMICS(M) Read Status
X GMIC(Q,C) Enable and Start

Photometer

SET D = FPHOR(Ø) Read photometer Data
*SET D = FPHOS(M) Read photometer Status
X GPHO(Q,C) Enable and Start Photom.

Scanner

SET D = FSCNR(N) Read Data
*SET D = FSCNS(M) Read Status

X GSCN(Q,C) Enable and Start

M-Mask to select certain bits
N-0 for X; 1 for Y
Q-Disable Null unless Ø. (PDP 8 can still detect null)
C-Set full count check unless Ø.

MISC. COMMANDS

SET D = FOR(M,N)-D becomes the inclusive 'OR' the binary codes for M + N

SET D = FAND(M,N)-D is the binary "AND" product M.N

SET D = FCOR(L,N) — Calculate correction to X, Y scale. D is correction in tenths of a micron if reading is L centimeters
N:=0 for X
=1 for Y

X SORT(B,W,N,BR)
- B — First block in Survey buffer
- W — First word
- N — Number images to be sorted
- BR — First output block

Sorts Survey data for shortest travel and puts the best measuring sequence in BR.

SET D = FTEST(M)
- D / M:
  - 1 Photometer Done
  - 2 Scanner Done
  - 4 Micrometer Done

Test AME Done

Position and Switch Reading, DISPLAY

SET D = FLOC(N) Read stage location
N: 0-for X
1-for Y
Divide reading by 40 to convert to millimeters

SET D = FSURV(N,K)
- N — 0:read X 1:read Y
- K — ignore error indications

Read Survey input, and test survey flag. If N = -1, D also becomes negative unless flag is set.

SET D = FSWTA(M). Read status of manual switches with mask M.

*See status coding

SET D = FSWTS(S,N)
- S — Switch group
- N — Switch number

Read Survey switches (use front panel labels to select S,N.)
N is a mask; all switches in the group are selected if N = 0.
clear flags if S = 0

SET D = FSWIT(G,S,X,Y,M)
- G — Switch Group
- S — Switch Number
- X — Initial position for joystick
- Y — Curser on CRT (no curser if zero)
- M — Mask can select several switches of a group

Read condition of sense switch panel

If X or Y are non-zero, returns 1024*X+Y to give current position of the CRT curser when switch G, S is pressed.

Control Codes

Bit Value	Control Register	Select A Data Register	*Select A Position Or Status Registers
1	Always zero	Always zero	Enable this group
2	Always zero	Enable this group	Always zero
4	Load this register	Always zero	Always zero
8	Enable Photom. skip		
16	Enable Scan. skip	Photometer high	Switch status
32	Enable Micr. skip	Photometer low	Photometer status
64	Stop everything	Y Scanner	Scanner status
128	Disable null	X Scanner	Micrometer status
256	Full count check	Y Micrometer high	Carriage Y high
512	Enable Photom.	Y Micrometer low	Carriage Y low
1024	Enable Scanner	X Micrometer high	Carriage X high
2048	Enable Micrometer	X Micrometer low	Carriage X low

Loaded with IOT command 6723 "SETAME"

*Select only one register per group

--After the register is selected, data are read to the accumulator by 6727 "READAM" or by 6737 "REDSUR"

*Select a survey input or carriage drivers

1	Y reverse drive
2	Y forward drive
4	X reverse drive
8	X forward drive
16	Buzzer turned on
32	Survey switches #3
64	Survey switches #2
128	Survey switches #1
256	Survey Y high
512	Survey Y low
1024	Survey X high
2048	Survey X low

Loaded with IOT command 6733 "SETSUR"

STATUS Coding

```
SET D = FPHOS(M) - Photometer Status, etc.
    M = 1 Micrometer Direction
        2 Scanner Direction
        4 Y Index OK
        8 X Index OK
       16    -
       32 Photometer Drive Direction
       64 Photometer Servo on
      128 Photometer Drive on
      256 Photometer Null on
      512 Photometer Park Position (Home)
     1024 Photometer Max Position
     2048 Photometer Zero position
```

```
SET D = FSWTA(M) Mechanical Switch Settings
    M = 1 System Reset
        2 Photometer enabled
        4 Photometer Adjust
        8 Photometer Manual Reset
       16 Photometer Manual Start
       32 Scanner Manual Reset
       64 Scanner Manual Start
      128 AME Normal
      256 Full Count Check
      512 Index Override
     1024 Micrometer Manual Reset
     2048 Micrometer Manual Start
```

```
SET D = FSCNS(M) Scanner Status
    M = 1-Y Servo on
        2-X Servo on
        4-Y Drive on
        8-X Drive on
       16-Star Detect
       32-Null Detect
       64-Y Sector Center Relay
      128-X Sector center relay
      256-y Max. limit
      512-YMin. limit (Home)
     1024-X Max limit
     2048-X Min. limit (Home)
```

```
SET D = FMICS(M) Micrometer Status
    M = 1-Y Servo
        2-X Servo
        4-Y Drive
        8-X Drive
       16-Y Null
       32-X Null
       64-Y Line Detect
      128-X line Detect
      256-Y Max limit
      512-Y Min limit (Home)
     1024-X Max limit
     2048-X Min. Limit (Home)
```

Several conditions can be tested at once by use of several bits in the mask. Eg. SET D = FSCNS(256+512+1024+2048) will give a non-zero answer of 512 if Y scanner is at Min. limit. or 2560 if X and Y scanner are both at Min. limit (home).