

AD-A152 687

MANUFACTURING METHODS AND TECHNOLOGY FOR DIGITAL FAULT
ISOLATION OF HYBRID (U) HUGHES AIRCRAFT CO FULLERTON CA
GROUND SYSTEMS GROUP 01 MAR 82 HAC-FR-82-12-193

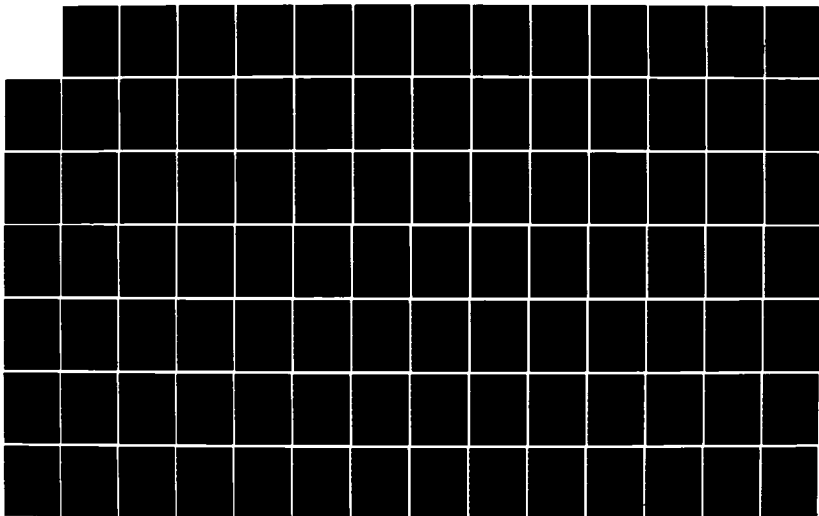
1/3

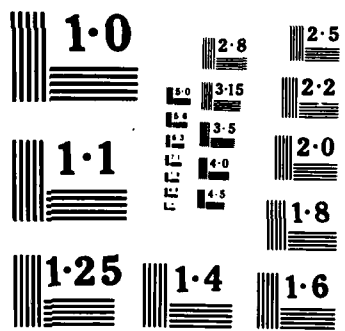
UNCLASSIFIED

DAAH01-81-D-002

F/G 9/5

NL





(1) ee

AD-A152 687

Final Report

Attachment I - Distributed Software Elements

**Manufacturing Methods And
Technology For**

**Digital Fault Isolation Of
Hybrid Microelectronic Assemblies**

Project No. 1023

1 MARCH 1982
CONTRACT NO. DAAH-01-81-D-A002

DTIC
ELECTE
APR 1 1 1985
S A

This document has been approved
for public release and sale; its
distribution is unlimited

DTIC FILE COPY

85 03 13 008

FINAL REPORT ATTACHMENT I -
DISTRIBUTED SOFTWARE ELEMENTS
Manufacturing Methods and Technology for
Digital Fault Isolation of Hybrid Microelectronic Assemblies

Project No. 1023

Prepared for
U.S. Army Missile Command
Redstone Arsenal, Alabama 35809

Project Officer: G. D. Little
DR SMI-RST
(205)-876-3604

Contract No. DAAH 01-81-D-A002
Task 0007
CDRL 002

Prepared by
Hughes Aircraft Company
Ground Systems Group
Fullerton, California 92634

Report Date: 1 March 1982
FR 82-12-193

FINAL REPORT ATTACHMENT I -

DISTRIBUTED SOFTWARE ELEMENTS

Manufacturing Methods and Technology for
Digital Fault Isolation of Hybrid Microelectronic Assemblies

Project No. 1023

Prepared for

U.S. Army Missile Command
Redstone Arsenal, Alabama 35809

Project Officer: G. D. Little
DR SMI-RST
(205)-876-3604

Contract No. DAAH 01-81-D-A002
Task 0007
CDRL 002

Prepared by

Hughes Aircraft Company
Ground Systems Group
Fullerton, California 92634

Report Date: 1 March 1982
FR 82-12-193

↓

FOREWORD

This final report presents the results, supporting data, and recommendations relating to an automatic test probe system (Autoprobe or AP), performed by Task 0007, under contract DAAH 01-81-D-A002. The program was identified as Project 1023 by the U.S. Army Missile Command at Redstone Arsenal, Alabama.

The manufacturing technology effort described by this report covers automatic test and fault isolation of digital hybrid microelectronic devices (D/HMAs) used in current and future production missile systems. A need for the single automatic probe, or Autoprobe, system was established by high production rate test and fault isolation of D/HMAs having complex LSI, microprocessor devices, and large number of I/O interconnections. Hardware and software resulting from project 1023 was further to be applied to the ATE system previously installed at Redstone Arsenal under project R783242.

An overview and objectives of project 1023 (or HFI Program) are presented in Section 1 of this report. Section 2 describes the work accomplished and includes the successful results of the Industry Demonstration for test and fault isolation of a D/HAM. Section 3 contains recommendations to achieve a full production type Autoprobe system and suggested future effort or study towards necessary AP improvements.

This document only presents two Fortran language programs and two machine language programs that were compiled. ←

Attch - see file



A-1

CONTENTS

SECTION I - PROBE CONTROL DRIVER

SECTION II - AUTOPROBE UTILITY PROGRAM

SECTION III - IEEE-488 DRIVER

SECTION IV - EXECUTIVE/COMMAND PROCESSOR

SECTION I
PROBE CONTROL DRIVER

REFERENCE: FINAL REPORT SECTION 2, SUBSECTION E.1,PAGE 2-20

1. INTPR - AUTOPROBE INITIALIZER
2. MONPR - AUTOPROBE MONITOR/DRIVER
3. SERCH - TEST POINT FILENAME LOOKUP
4. READY - WAIT FOR AUTOPROBE READY
5. ERROR - PROBE DRIVER ERROR HANDLER
6. BEROR - BUS ERROR HANDLER


```

0001 FTN4,L
0002 PROGRAM INTPR
0003 C =====
0004 C
0005 C SON PROGRAM INTPR IS CALLED BY FATHER PROGRAM PCDRV
0006 C ONCE ONLY. PCDRV PASSES HYBRID FILE NAME VIA RMPAR.
0007 C INTPR PASSES BACK NORMAL (1) OR ABNORMAL TERMINATION
0008 C (-1) VIA PRIN. IF ABNORMAL, BOTH PROGRAMS ABORT.
0009 C
0010 C
0011 INTEGER FNAME(3),DCB(144),BFR(13),PARM(5)
0012 INTEGER ASD(3),ASE(3),AASD(3),AASE(3),BASD(3),BASE(3)
0013 C
0014 EQUIVALENCE (ASD(1),BFR(8)), (ASE(1),BFR(11))
0015 C
0016 C ----- RETRIEVE HYBRID TABLE FILE NAME -----
0017 C
0018 CALL RMPAR (PARM)
0019 DO 10 I=1,3
0020 10 FNAME(I) = PARM(I)
0021 C
0022 C ----- OPEN HYBRID TABLE FILE -----
0023 C ----- SET BIT 0 FOR NON-EXCLUSIVE OPEN -----
0024 C
0025 IERR = 0
0026 CALL OPEN (DCB,IERR,FNAME,1B)
0027 IF (IERR.LT.0) GO TO 90
0028 C
0029 C ----- READ AND SAVE POINTS A AND B -----
0030 C
0031 CALL READF (DCB,IERR,BFR)
0032 IF (IERR.LT.0) GO TO 90
0033 DO 12 I=1,3
0034 AASD(I) = ASD(I)
0035 12 AASE(I) = ASE(I)
0036 C
0037 CALL READF (DCB,IERR,BFR)
0038 IF (IERR.LT.0) GO TO 90
0039 DO 14 I=1,3
0040 BASD(I) = ASD(I)
0041 14 BASE(I) = ASE(I)
0042 C
0043 C ----- PREPARE FOR MONITORING -----
0044 C ----- CHECK IF ALL READY, SET AUTO MODE -----
0045 C
0046 LU = 36
0047 CALL CLEAR (LU,1)
0048 C
0049 ITMP = 0
0050 IEFLG = 0
0051 CALL READY (LU,IERR)
0052 IF (IERR.NE.0) CALL BEROR (IERR,ITMP,IEFLG)
0053 IF (IEFLG.LT.0) GO TO 95
0054 C
0055 ITMP = 0

```

```

0056          IEFLG = 0
0057 20       ICMD = 2HMO
0058          WRITE (LU,120) ICMD
0059 120      FORMAT (A2)
0060 C
0061 C ----- RETRIEVE POINT A AND SEND COORDINATES -----
0062 C ----- FORMAT: AX <COORD> AY <COORD> <CR> <LF> -----
0063 C
0064          CALL READY (LU,IERR)
0065          IF (IERR.NE.0) CALL BEROR (IERR,ITMP,IEFLG)
0066          IF (IEFLG) 95,30,20
0067 C
0068 30       ITMP = 0
0069          IEFLG = 0
0070 35       ICM1 = 2HAX
0071          ICM2 = 2HAY
0072          WRITE (LU,135) ICM1,AASD,ICM2,AASE
0073 135      FORMAT (2(A2,3A2))
0074 C
0075 C ----- RETRIEVE POINT B AND SEND COORDINATES -----
0076 C ----- FORMAT: BX <COORD> BY <COORD> <CR> <LF> -----
0077 C
0078          CALL READY (LU,IERR)
0079          IF (IERR.NE.0) CALL BEROR (IERR,ITMP,IEFLG)
0080          IF (IEFLG) 95,40,35
0081 C
0082 40       ITMP = 0
0083          IEFLG = 0
0084 45       ICM1 = 2HBX
0085          ICM2 = 2HBY
0086          WRITE (LU,135) ICM1,BASD,ICM2,BASE
0087 C
0088 C ----- SEND LOAD HYBRID COMMAND -----
0089 C
0090          CALL READY (LU,IERR)
0091          IF (IERR.NE.0) CALL BEROR (IERR,ITMP,IEFLG)
0092          IF (IEFLG) 95,47,45
0093 C
0094 47       ITMP = 0
0095          IEFLG = 0
0096 50       ICMD = 2HLD
0097          WRITE (LU,120) ICMD
0098 C
0099          CALL READY (LU,IERR)
0100          IF (IERR.NE.0) CALL BEROR (IERR,ITMP,IEFLG)
0101          IF (IEFLG) 95,80,50
0102 C
0103 C ----- CLOSE FILE THEN SEND PROGRAM STATUS TO PAPA -----
0104 C
0105 80       PARM(1) = 1
0106 85       CALL CLOSE (DCB)
0107          CALL PRIN (PARM)
0108          CALL EXEC (6)
0109 C
0110 C ----- ERROR CONDITION, ABORT PROGRAM -----

```

0000 0000 INTER 8:54 PM MON., 14 DEC., 1981

```
0112 99 WRITE (1,10) IERN
0113 199 FORMAT ('(X) * FMT EXPORT ", IS, " ***')
0114 CALL CLEAR (LB,1)
0115 95 PARM(1) = -1
0116 GO TO 85
0117 END
```

FTN4 COMPILER: HP92060-16092 REV 2026 (010403)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00000 COMPILATION SUCCESSFUL

```
0118          SUBROUTINE BEROR (IERRC,IERRP,IEFLG)
0119 C          =====
0120 C
0121 C
0122 C ----- IS ERROR FATAL OR NON-FATAL? -----
0123 C
0124 C          IF (IAND (IERRC,100000B).NE.0) GO TO 30
0125 C
0126 C ----- NON-FATAL, IS IT THE SAME AS PREVIOUS CODE? -----
0127 C
0128 C          IERRC = IERRC/256
0129 C          IF (IERRC.NE.IERRP) GO TO 10
0130 C          IEFLG = IEFLG + 1
0131 C          GO TO 20
0132 C
0133 10          IEFLG = 1
0134 C          IERRP = IERRC
0135 C
0136 C ----- PRINT NON-FATAL ERROR MESSAGE -----
0137 C
0138 20          WRITE (1,120) IERRC
0139 120          FORMAT (/"*** NON-FATAL BUS ERROR ",I5," ***)
0140 C
0141 C ----- SAME NON-FATAL ERROR 3 TIMES? IF YES, ABORT -----
0142 C
0143 C          IF (IEFLG.EQ.3) GO TO 40
0144 C          PAUSE
0145 C          RETURN
0146 C
0147 C ----- FATAL, PRINT FATAL ERROR MESSAGE AND ABORT -----
0148 C
0149 30          IERRC = IERRC/256
0150 C          WRITE (1,130) IERRC
0151 130          FORMAT (/"*** FATAL BUS ERROR ",I5," ***)
0152 C
0153 40          IEFLG = -99
0154 C          CALL CLEAR (LU,1)
0155 C          PAUSE
0156 C          RETURN
0157 C          END
```

FTN4 COMPILER: HPS2060-16092 REV. 2026 (800423)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00125 COMMON = 00000

```
0158      SUBROUTINE READY (LU,IERR)
0159 C      =====
0160 C
0161 C
0162      INTEGER MSG(2)
0163 C
0164 C ----- DELAY FOR 500 MILLISECONDS -----
0165 C
0166      IRESL = 1
0167      IMULT = 0
0168      IOFST = -50
0169 10    CALL EXEC (12,0,IRESL,IMULT,IOFST)
0170 C
0171 C ----- READY FOR DATA? CHECK "BUSY" AND "ERROR" BITS -----
0172 C
0173      ICMD = 2H??
0174      WRITE (LU,100) ICMD
0175 100   FORMAT (A2)
0176      READ (LU,110) MSG
0177 110   FORMAT (2A2)
0178      IF (IAND (MSG(1),10000B).NE.0) GO TO 10
0179      IF (IAND (MSG(1),2000B).NE.0) GO TO 20
0180 C
0181      IERR = 0
0182      RETURN
0183 C
0184 20    ICMD = 2HSE
0185      WRITE (LU,100) ICMD
0186      READ (LU,110) MSG
0187      IERR = MSG(1)
0188      RETURN
0189      END
```

FTN4 COMPILER: HP92060-16092 REV. 2026 (800423)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00104 COMMON = 000

```

0001 FTN4,L
0002 PROGRAM HYBRD
0003 C =====
0004 C
0005 C
0006 INTEGER FNAME(3)
0007 C
0008 C ----- DISPLAY SELECTION OF TASKS -----
0009 C
0010 10 WRITE (1,100)
0011 100 FORMAT (// "TASKS FOR HYBRID TABLE FILES",
0012 &/4X, "1 - ENTER FILE", /4X, "2 - INITIALIZE A,B",
0013 &/4X, "3 - DISPLAY FILE", /4X, "4 - MODIFY FILE",
0014 &/4X, "5 - TEST PROBER", /4X, "6 - EXIT TASKS"/)
0015 C
0016 C ----- MAKE A SELECTION -----
0017 C
0018 20 WRITE (1,120)
0019 120 FORMAT ("ENTER NUMBER OF SELECTION: _")
0020 READ (1,*) N
0021 IF (N.LT.1.OR.N.GT.6) GO TO 20
0022 IF (N.EQ.6) GO TO 40
0023 C
0024 WRITE (1,130)
0025 130 FORMAT (// "NAME OF HYBRID TABLE FILE? _")
0026 READ (1,135) FNAME
0027 135 FORMAT (3A2)
0028 GO TO (1,2,3,4,5) N
0029 C
0030 C ----- CALL TASK PROGRAM AND EXECUTE -----
0031 C
0032 1 CALL ENTER (FNAME)
0033 GO TO 10
0034 C
0035 2 CALL INTAB (FNAME)
0036 GO TO 10
0037 C
0038 3 CALL DSPLY (FNAME)
0039 GO TO 10
0040 C
0041 4 CALL MODFY (FNAME)
0042 GO TO 10
0043 C
0044 5 CALL TEST (FNAME)
0045 GO TO 10
0046 C
0047 C ----- EXIT, NO MORE TASKS TO DO -----
0048 C
0049 40 WRITE (1,140)
0050 140 FORMAT (// "EXIT TASKS FOR HYBRID TABLE FILES")
0051 CALL EXEC (6)
0052 END

```

```

348      SUBROUTINE ALIGN (IXA,IYA,IXB,IYB)
349      C      =====
350      C
351      C
352      COMMON /VALS/ XA,YA,XB,YB,XM,YM,ABSLP,CDSLPL,VFLG,HFLG
353      C
354      INTEGER VFLG,HFLG
355      C
356      XA = FLOAT (IXA)
357      YA = FLOAT (IYA)
358      XB = FLOAT (IXB)
359      YB = FLOAT (IYB)
360      C
361      C ----- CHANGE PTS TO NEW COORD SYSTEM WITH MIDPT M -----
362      C
363      XM = (XA + XB) / 2.
364      YM = (YA + YB) / 2.
365      C
366      XA = XA - XM
367      YA = YA - YM
368      XB = XB - XM
369      YB = YB - YM
370      C
371      C ----- CHECK FOR VERTICAL/HORIZONTAL REFERENCE -----
372      C
373      VFLG = 0
374      HFLG = 0
375      IF (ABS (XA-XB).LT.1.0) GO TO 10
376      IF (ABS (YA-YB).LT.1.0) GO TO 20
377      C
378      C ----- COMPUTE SLOPES OF LINE AB AND ITS PERPENDICULAR CD -----
379      C
380      ABSLP = (YA - YB) / (XA - XB)
381      CDSLPL = (XA - XB) / (YB - YA)
382      RETURN
383      C
384      C ----- LINE AB IS VERTICAL, DETERMINE PERSPECTIVE -----
385      C
386      10      VFLG = 1
387      IF (YA.LT.YB) VFLG = -1
388      RETURN
389      C
390      C ----- LINE AB IS HORIZONTAL, DETERMINE PERSPECTIVE -----
391      C
392      20      HFLG = 1
393      IF (XA.LT.XB) HFLG = -1
394      RETURN
395      END

```

FTN4 COMPILER: HP92060-16092 REV. 2026 (800423)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00159 COMMON = 00000

```
306      SUBROUTINE SERCH (PINWT,BFR,DCB,IERR,NUM)
307 C      =====
308 C
309 C      ENTER: PINWT - PIN SEARCHING
310 C             DCB  - DISK CONTROL INFORMATION OF FILE SEARCHING
311 C      EXIT: BFR   - RECORD OF PIN SEARCHED
312 C             NUM  - RECORD NUMBER
313 C             IERR - ERROR STATUS OF SEARCH
314 C                 IERR =  0 => A MATCH FOUND
315 C                 IERR = -12 => EOF, NO MATCH
316 C                 IERR <  0 => FMP ERROR
317 C                 IERR = 12 => STRING ERROR
318 C
319 C
320 C      INTEGER DCB(144),BFR(13),PINWT(5)
321 C
322 C      NUM = 3
323 C      IERR = 0
324 C
325 C      ----- READ RECORD.  AT EOF? -----
326 C
327 10    CALL READF (DCB,IERR,BFR,13,LEN,NUM)
328      IF (IERR.LT.0) RETURN
329 C
330      IF (BFR(1).NE.2H .AND.BFR(2).NE.2H ) GO TO 20
331      IERR = -12
332      RETURN
333 C
334 C      ----- COMPARE PIN NAMES.  M = 0 => A MATCH -----
335 C
336 20    M = JSDBM (BFR,1,10,PINWT,1,IERR)
337      IF (IERR.LT.0) GO TO 90
338      IF (M.EQ.0) RETURN
339 C
340      NUM = NUM + 1
341      GO TO 10
342 C
343 C      ----- REPORT STRING ERRORS -----
344 C
345 90    IERR = 99
346      RETURN
347      END
```

FTN4 COMPILER: HP92060-16092 REV. 2026 (800423)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00092 COMMON = 00000

PAGE 0008 MODIFY 8:46 PM MON., 14 DEC., 1981

```
0296 B9 WRITE (1,189)
0297 189 FORMAT (/"TABLE FILE MODIFICATIONS COMPLETED")
0298 RETURN
0299 C
0300 C ----- REPORT ERRORS -----
0301 C
0302 90 CALL CLOSE (DCB)
0303 95 CALL ERROR (IERR)
0304 RETURN
0305 END
```

FTN4 COMPILER: HP92060-16092 REV. 2026 (800423)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00891 COMMON = 00000

```

0241      WRITE (1,157)
0242 157   FORMAT ("X,Y COORDINATES?")
0243      READ (1,*) IXC
0244      READ (1,*) IYC
0245      C
0246  C ----- CALCULATE RELATIVE D,E COORDINATES -----
0247      C
0248 60    BFR(6) = IXC
0249      BFR(7) = IYC
0250      CALL ALIGN (IXA,IYA,IXB,IYB)
0251      CALL GETDE (IXC,IYC,ASD,ASE)
0252      C
0253 65    WRITE (1,140) PINAM,BFR(6),BFR(7),ASD,ASE
0254      GO TO 80
0255      C
0256      C
0257  C ===== DELETE - FLAG THE RECORD =====
0258  C ----- PIN NAME IS CHANGED TO ASTERISKS -----
0259  C ----- NUMERIC DATA IS RESET TO NULLS -----
0260      C
0261 70    DO 75 I=1,5
0262      PINAM(I) = 2H**
0263      C
0264      DO 77 I=6,13
0265 77    BFR(I) = 0
0266      C
0267      C
0268  C ----- WRITE MODIFIED RECORD TO FILE -----
0269      C
0270 80    CALL WRITF (DCB,IERR,BFR,13,NUM)
0271      IF (IERR.LT.0) GO TO 90
0272      C
0273  C ----- MORE TO MODIFY IN THIS FILE? -----
0274      C
0275 82    WRITE (1,182)
0276 182   FORMAT (/"MORE TO MODIFY IN THIS FILE? _")
0277      READ (1,122) N
0278      IF (N.EQ.2HY ) GO TO 20
0279      IF (N.NE.2HN ) GO TO 82
0280      CALL CLOSE (DCB)
0281      C
0282  C ----- MODIFY ANOTHER FILE? -----
0283      C
0284 85    WRITE (1,185)
0285 185   FORMAT ("MODIFICATIONS IN ANOTHER FILE? _")
0286      READ (1,122) N
0287      IF (N.EQ.2HN ) GO TO 89
0288      IF (N.NE.2HY ) GO TO 85
0289      C
0290      WRITE (1,186)
0291 186   FORMAT (/"NAME OF NEXT TABLE FILE? _")
0292      READ (1,187) FNAME
0293 187   FORMAT (3A2)
0294      GO TO 10
0295      C

```

```

0186 135  FORMAT (SA2)
0187  C
0188  C ----- LOOK FOR PIN IN TABLE -----
0189  C ----- IF FOUND, CONTINUE -----
0190  C
0191          IERR = 0
0192          CALL SERCH (PINWT,BFR,DCB,IERR,NUM)
0193          IF (IERR.EQ.-12) GO TO 25
0194          IF (IERR.NE.0) GO TO 90
0195  C
0196          IF (M.EQ.2HD ) GO TO 70
0197  C
0198  C
0199  C ===== CHANGE - DISPLAY RECORD AND MAKE CHANGES =====
0200  C ----- PIN NAME? <CR> OR "N" IF NO CHANGE -----
0201  C ----- X,Y COORDS? ANY 2 NEG NUMS IF NO CHANGE -----
0202  C ----- D,E ARE CALCULATED IF X,Y ARE CHANGED -----
0203  C
0204          WRITE (1,140) PINAM,BFR(6),BFR(7),ASD,ASE
0205 140  FORMAT (/ " PIN NAME",8X,"X",6X,"Y",9X,"D",7X,"E",
0206          &/" ",5A2,I9,I7,5X,3A2,2X,3A2)
0207  C
0208          WRITE (1,142)
0209 142  FORMAT (/ "CHANGE PIN NAME? _")
0210          READ (1,135) PINWT
0211          IF (PINWT(1).EQ.2HN .OR. PINWT(1).EQ.2H ) GO TO 46
0212  C
0213          DO 44 I=1,5
0214 44  PINAM(I) = PINWT(I)
0215  C
0216 46  WRITE (1,146)
0217 146  FORMAT ("CHANGE X,Y COORDINATES?")
0218          READ (1,*) IXC
0219          READ (1,*) IYC
0220          IF (IXC.GE.0) GO TO 60
0221          GO TO 65
0222  C
0223  C
0224  C ===== ADD - LOOK FOR FIRST AVAILABLE RECORD =====
0225  C ----- I.E. AT DELETED RECORD OR AT EOF -----
0226  C ----- INPUT DATA FOR PIN NAME AND ITS COORDS -----
0227  C ----- (D,E) IS CALCULATED -----
0228  C
0229 50  DO 52 I=1,5
0230 52  PINWT(I) = 2H**
0231          IERR = 0
0232          CALL SERCH (PINWT,BFR,DCB,IERR,NUM)
0233          IF (IERR.EQ.-12) GO TO 55
0234          IF (IERR) 90,55,90
0235  C
0236  C ----- ADD PIN NAME AND X,Y COORDINATES -----
0237  C
0238 55  WRITE (1,132)
0239          READ (1,135) PINAM
0240  C

```

```

0131      SUBROUTINE MODIFY (FNAME)
0132 C      =====
0133 C
0134 C
0135      COMMON /VALS/ XA,YA,XB,YB,XM,YM,ABSLP,CDSLPL,VFLG,HFLG
0136 C
0137      INTEGER FNAME(3),DCB(144),BFR(13),VFLG,HFLG
0138      INTEGER PINAM(5),PINWT(5),ASD(3),ASE(3)
0139 C
0140      EQUIVALENCE (PINAM(1),BFR(1))
0141      EQUIVALENCE (ASD(1),BFR(8)), (ASE(1),BFR(11))
0142 C
0143 C ----- OPEN HYBRID TABLE FILE -----
0144 C ----- SET BIT 0 FOR NON-EXCLUSIVE OPEN -----
0145 C ----- SET BIT 1 FOR UPDATING -----
0146 C
0147 10     CALL OPEN (DCB,IERR,FNAME,3B)
0148      IF (IERR.LT.0) GO TO 95
0149 C
0150 C ----- READ FIRST TWO RECORDS FOR REF POINTS A AND B -----
0151 C
0152      CALL READF (DCB,IERR,BFR,13,LEN,1)
0153      IF (IERR.LT.0) GO TO 90
0154      IXA = BFR(6)
0155      IYA = BFR(7)
0156 C
0157      CALL READF (DCB,IERR,BFR,13,LEN,2)
0158      IF (IERR.LT.0) GO TO 90
0159      IXB = BFR(6)
0160      IYB = BFR(7)
0161 C
0162 C ----- REQUEST MODIFICATION - ADD/DELETE/CHANGE -----
0163 C
0164 20     WRITE (1,120)
0165 120    FORMAT (/"MODIFY - (A)DD, (D)ELETE, OR (C)HANGE? _")
0166      KNT = 0
0167      READ (1,122) M
0168 122    FORMAT (A2)
0169      IF (M.EQ.2HA ) GO TO 50
0170      IF (M.EQ.2HD .OR.M.EQ.2HC ) GO TO 30
0171      GO TO 20
0172 C
0173 C ----- ASK FOR PIN NAME IF CHANGING/DELETING -----
0174 C ----- ALLOW 3 TIMES FOR VALID PIN NAME REQUEST -----
0175 C ----- IF NOT FOUND AFTER 3RD TIME, STOP SEARCH -----
0176 C
0177 25     CALL ERROR (IERR)
0178 30     DO 32 I = 1,5
0179 32     PINWT(I) = 2H
0180      KNT = KNT + 1
0181      IF (KNT.GT.3) GO TO 82
0182 C
0183      WRITE (1,132)
0184 132    FORMAT (/"PIN NAME? _")
0185      READ (1,135) PINWT

```

```

0106      SUBROUTINE SERCH (FNAME,SLEN,STRG,IERR,BLEN,BFR)
0107 C      =====
0108 C
0109 C      ENTER:  FNAME - NAME OF FILE TO BE SEARCHED
0110 C              STRG - STRING SEARCHING IN FILE
0111 C              SLEN - LENGTH OF STRING
0112 C              BLEN - LENGTH OF RECORD IN FILE
0113 C      EXIT:   IERR  - ERROR STATUS OF SEARCH
0114 C              IERR = 0 => A MATCH FOUND
0115 C              IERR = -12 => EOF, NO MATCH IN HYBRID FILE
0116 C              LEN = -1 => EOF, NO MATCH IN SPOOL FILE
0117 C              IERR < 0 => FMP ERROR
0118 C              IERR = 12 => STRING ERROR
0119 C              BFR   - FILE RECORD OF STRING
0120 C
0121 C
0122      INTEGER FNAME(3),DCB(144),BFR(40),BLEN,STRG(5),SLEN
0123 C
0124 C ----- OPEN FILE TO BE SEARCHED -----
0125 C ----- SET BIT 0 FOR NON-EXCLUSIVE OPEN -----
0126 C
0127      CALL OPEN (DCB,IERR,FNAME,1B)
0128      IF (IERR.LT.0) RETURN
0129 C
0130 C ----- INITIALIZE FOR SEARCH -----
0131 C
0132      NUM = 3
0133      IERR = 0
0134 10      DO 15 I=1,40
0135 15      BFR(I) = 2H
0136 C
0137 C ----- READ RECORD. AT EOF? -----
0138 C
0139      CALL READF (DCB,IERR,BFR,BLEN,LEN,NUM)
0140      IF (IERR.LT.0) GO TO 30
0141      IF (LEN.EQ.-1.AND.BLEN.EQ.40) GO TO 20
0142      IF (BFR(1).EQ.2H .AND.BFR(2).EQ.2H .AND.BLEN.EQ.13) GO TO 20
0143 C
0144 C ----- COMPARE PIN NAMES. M = 0 => A MATCH -----
0145 C
0146      M = JSDCM (STRG,1,SLEN,BFR,1,IERR)
0147      IF (IERR.LT.0) GO TO 90
0148      IF (M.EQ.0) GO TO 30
0149 C
0150      M = JSDCM (STRG,2,SLEN-1,BFR,1,IERR)
0151      IF (IERR.LT.0) GO TO 90
0152      IF (M.EQ.0) GO TO 30
0153 C
0154      NUM = NUM + 1
0155      GO TO 10
0156 C
0157 C ----- NO MATCH AT EOF/A MATCH, CLOSE FILE AND RETURN -----
0158 20      IERR = -12
0159 30      CALL CLOSE (DCB)
0160      RETURN

```

```
0053      SUBROUTINE ERROR (IERR)
0054      C      =====
0055      C
0056      C
0057      IF (IERR.EQ.-12) GO TO 30
0058      IF (IERR.EQ.99) GO TO 20
0059      C
0060      WRITE (1,110) IERR
0061      110  FORMAT (/ "** FMP ERROR ",I5," **")
0062      C
0063      C
0064      IF (IERR.EQ.130)
0065      120  FORMAT (" INVALID CHARACTER(S) IN PIN NAME **")
0066      C
0067      C
0068      IF (IERR.EQ.131)
0069      130  FORMAT (" NO NAME FOR FOLLOWING TABLE FILE **")
0070      C
0071      END
```

FTN4 COMPILER: HP92060-18092 REV. 1.128 10/78

** NO WARNINGS ** NO ERRORS ** PROGRAM = ...

```

0056 C
0057 C ----- POSITION REFERENCE POINT B -----
0058 C ----- STORE DITTO IN RECORD #2, WORDS 8-13 -----
0059 C
0060 CALL READY (LU,IERR)
0061 IF (IERR.NE.0) CALL BEROR (IERR,ITMP,IEFLG)
0062 IF (IEFLG.LT.0) GO TO 95
0063 WRITE (LU,115)
0064 30 WRITE (1,130)
0065 130 FORMAT (/"POSITION REFERENCE POINT B - ")
0066 C
0067 ITMP = 0
0068 IEFLG = 0
0069 33 CALL GETAB (LU,ASX,ASY,IERR)
0070 IF (IERR.NE.0) CALL BEROR (IERR,ITMP,IEFLG)
0071 IF (IEFLG) 95,34,33
0072 C
0073 34 CALL READF (DCB,IERR,BFR,13,LEN,2)
0074 IF (IERR.LT.0) GO TO 90
0075 C
0076 DO 35 I=1,3
0077 BFR(I+7) = ASX(I)
0078 35 BFR(I+10) = ASY(I)
0079 CALL WRITF (DCB,IERR,BFR,13,2)
0080 IF (IERR.LT.0) GO TO 90
0081 C
0082 C ----- CLOSE FILE AND RETURN -----
0083 C
0084 70 CALL CLOSE (DCB)
0085 WRITE (1,170)
0086 170 FORMAT (/"INITIALIZATION OF TABLE FILE COMPLETED")
0087 RETURN
0088 C
0089 C ----- REPORT ERRORS -----
0090 C
0091 90 CALL ERROR (IERR)
0092 CALL CLEAR (LU,1)
0093 95 CALL CLOSE (DCB)
0094 RETURN
0095 END

```

FTN4 COMPILER: HP92060-16092 REV. 2026 (800423)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00543 COMMON = 00000

```

0001 FTN4,L
0002 SUBROUTINE INTAB (FNAME)
0003 C =====
0004 C
0005 C
0006 INTEGER FNAME(3),DCB(144),BFR(13),VFLG,HFLG,ASX(3),ASY(3)
0007 C
0008 C
0009 C ----- OPEN HYBRID TABLE FILE -----
0010 C ----- SET BIT 0 FOR NON-EXCLUSIVE OPEN -----
0011 C ----- SET BIT 1 FOR UPDATING -----
0012 C
0013 CALL OPEN (DCB,IERR,FNAME,3B)
0014 IF (IERR.LT.0) GO TO 90
0015 C
0016 C ----- SET LU FOR IEEE 488 BUS -----
0017 C ----- SEND DEVICE CLEAR, SET MODE TO MANUAL -----
0018 C
0019 LU = 36
0020 CALL CLEAR (LU,1)
0021 C
0022 CALL READY (LU,IERR)
0023 IF (IERR.NE.0) CALL BEROR (IERR,ITMP,IEFLG)
0024 IF (IEFLG.LT.0) GO TO 95
0025 WRITE (LU,117)
0026 117 FORMAT ("M2")
0027 C
0028 C ----- POSITION REFERENCE POINT A -----
0029 C ----- STORE COORDINATES IN RECORD #1, WORDS 8-13 -----
0030 C ----- SEND MOVE COMMAND TO (0,0) TO ALLOW JOGGING -----
0031 C ----- JOG PROBE TO THE POINT AND REQUEST POSITION -----
0032 C
0033 CALL READY (LU,IERR)
0034 IF (IERR.NE.0) CALL BEROR (IERR,ITMP,IEFLG)
0035 IF (IEFLG.LT.0) GO TO 95
0036 WRITE (LU,115)
0037 115 FORMAT ("X00000Y00000")
0038 C
0039 20 WRITE (1,120)
0040 120 FORMAT (/"POSITION REFERENCE POINT A - ")
0041 C
0042 ITMP = 0
0043 IEFLG = 0
0044 23 CALL GETAB (LU,ASX,ASY,IERR)
0045 IF (IERR.NE.0) CALL BEROR (IERR,ITMP,IEFLG)
0046 IF (IEFLG) 95,24,23
0047 C
0048 24 CALL READF (DCB,IERR,BFR,13,LEN,1)
0049 IF (IERR.LT.0) GO TO 90
0050 C
0051 DO 25 I=1,3
0052 BFR(I+7) = ASX(I)
0053 25 BFR(I+10) = ASY(I)
0054 CALL WRITE (DCB,IERR,BFR,13,1)
0055 IF (IERR.LT.0) GO TO 90

```


PAGE 0004 ENTER 8:46 PM MON., 14 DEC., 1981

```
0119 C
0120 CALL CLOSE (DCB)
0121 WRITE (1,185)
0122 185 FORMAT (/"TABLE FILE DATA ENTRY COMPLETED")
0123 RETURN
0124 C
0125 C ----- REPORT ERRORS -----
0126 C
0127 90 CALL CLOSE (DCB)
0128 95 CALL ERROR (IERR)
0129 RETURN
0130 END
```

FTN4 COMPILER: HP92060-16092 REV. 2026 (800423)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00628 COMMON = 00000

```

0064 C
0065 WRITE (1,150)
0066 150 FORMAT (/"ENTER POINT B COORDINATES")
0067 READ (1,*) IXB
0068 READ (1,*) IYB
0069 BFR(6) = IXB
0070 BFR(7) = IYB
0071 C
0072 CALL WRITF (DCB,IERR,BFR)
0073 IF (IERR.LT.0) GO TO 90
0074 C
0075 C ----- DETERMINE NEW COORDINATE SYSTEM -----
0076 C
0077 CALL ALIGN (IXA,IYA,IXB,IYB)
0078 C
0079 C ----- NOW INPUT DATA FOR HYBRID NODE POINT C -----
0080 C ----- RECORD FORMAT: WORDS 1-5 PIN NAME IN ASCII -----
0081 C ----- WORDS 6-7 ABSOLUTE COORDS IN INTEGER -----
0082 C ----- WORDS 8-13 RELATIVE COORDS IN ASCII -----
0083 C
0084 DO 80 KNT=1,N
0085 C
0086 WRITE (1,160)
0087 160 FORMAT (/"ENTER PIN NAME AND ITS X,Y COORDINATES")
0088 READ (1,165) PINAM
0089 165 FORMAT (5A2)
0090 READ (1,*) IXC
0091 READ (1,*) IYC
0092 BFR(6) = IXC
0093 BFR(7) = IYC
0094 C
0095 C ----- CALCULATE RELATIVE POINT (D,E) -----
0096 C
0097 CALL GETDE (IXC,IYC,ASD,ASE)
0098 C
0099 C ----- WRITE DATA TO FILE -----
0100 C
0101 CALL WRITF (DCB,IERR,BFR)
0102 IF (IERR.LT.0) GO TO 90
0103 C
0104 DO 75 I=1,5
0105 75 BFR(I) = 2H
0106 80 CONTINUE
0107 C
0108 C ----- INITIALIZE THE REST OF FILE WITH BLANKS AND ZEROES -----
0109 C
0110 DO 82 I=6,13
0111 82 BFR(I) = 0
0112 C
0113 DO 85 I=N+1,N+100
0114 CALL WRITF (DCB,IERR,BFR)
0115 IF (IERR.LT.0.AND.IERR.NE.-12) GO TO 90
0116 85 CONTINUE
0117 C
0118 C ----- DONE, CLOSE FILE AND RETURN -----

```

```

0009      SUBROUTINE ENTER (FNAME)
0010      C      =====
0011      C
0012      C
0013      COMMON /VALS/ XA,YA,XB,YB,XM,YM,ABSLP,CDSLPL,VFLG,HFLG
0014      C
0015      INTEGER FNAME(3),DCB(144),SIZE(2),BFR(13)
0016      INTEGER PINAM(5),ASD(3),ASE(3),VFLG,HFLG
0017      C
0018      EQUIVALENCE (PINAM(1),BFR(1))
0019      EQUIVALENCE (ASD(1),BFR(8)), (ASE(1),BFR(11))
0020      C
0021      C -----      CREATE HYBRID TABLE FILE, TYPE 2      -----
0022      C ----- (NUMBER OF NODES + 100) RECORDS/FILE, 13 WORDS/RECORD -----
0023      C -----      STORE ON CARTRIDGE 18 FOR NOW      -----
0024      C
0025      WRITE (1,110)
0026      110  FORMAT ("NUMBER OF HYBRID NODES IN THE FILE? _")
0027      READ (1,*) N
0028      C
0029      SIZE(2) = 13
0030      SIZE(1) = (SIZE(2) * (N+100)) / 128
0031      ICR = -18
0032      C
0033      CALL CREAT (DCB,IERR,FNAME,SIZE,2,0,ICR)
0034      IF (IERR.LT.0) GO TO 95
0035      C
0036      C -----      INPUT DATA FOR REFERENCE POINT A      -----
0037      C ----- RECORD #1 FORMAT:  WORDS 1-5  "A"      -----
0038      C -----      WORDS 6-7  COORDINATES IN INTEGER -----
0039      C -----      WORDS 8-13  NULLS (SET IN INTAB) -----
0040      C
0041      BFR(1) = 24A
0042      DO 20 I=2,5
0043      20  BFR(I) = 2H
0044      C
0045      WRITE (1,130)
0046      130  FORMAT (/"ENTER POINT A COORDINATES"/)
0047      READ (1,*) IXA
0048      READ (1,*) IYA
0049      BFR(6) = IXA
0050      BFR(7) = IYA
0051      C
0052      BFR(8) = 2H 0
0053      BFR(11) = 2H 0
0054      DO 40 I=9,10
0055      BFR(I) = 2H00
0056      40  BFR(I+3) = 2H00
0057      C
0058      CALL WRITF (DCB,IERR,BFR)
0059      IF (IERR.LT.0) GO TO 90
0060      C
0061      C -----      DITTO FOR REFERENCE POINT B IN RECORD #2 -----
0062      C
0063      BFR(1) = 2HB

```

```
0001 FTN4,L
0002     BLOCK DATA VALS
0003 C     =====
0004 C
0005 C
0006     COMMON /VALS/ XA,YA,XB,YB,XM,YM,ABSLP,CDSL P,VFLG,HFLG
0007     INTEGER VFLG,HFLG
0008     END
```

FTN4 COMPILER: HP92060-16092 REV. 2026 (800423)

** NO WARNINGS ** NO ERRORS **

BLOCK COMMON VALS SIZE = 00018

```
0396          SUBROUTINE GETDE (IXC,IYC,ASD,ASE)
0397          C          =====
0398          C
0399          C
0400          COMMON /VALS/ XA,YA,XE,YB,XM,YM,ABSLP,CDSL P,VFLG,HFLG
0401          C
0402          INTEGER ASD(3),ASE(3),DTMP(4),ETMP(4),VFLG,HFLG
0403          C
0404          C ----- CALCULATE HYBRID POINT (D,E) -----
0405          C
0406          XC = FLOAT (IXC)
0407          YC = FLOAT (IYC)
0408          CALL CALDE (XC,YC)
0409          C
0410          C ----- CONVERT TO ASCII SNNNNN.0 FORMAT -----
0411          C ----- (-----) RETURN THIS TO STORE IN FILE -----
0412          C
0413          CALL CODE
0414          WRITE (DTMP,100) XC
0415          CALL CODE
0416          WRITE (ETMP,100) YC
0417          100  FORMAT (F8.1)
0418          C
0419          DO 10 I = 1,3
0420             ASD(I) = DTMP(I)
0421          10  ASE(I) = ETMP(I)
0422          C
0423          C ----- NOW CHANGE LEADING BLANKS TO ZEROES KEEPING SIGN -----
0424          C
0425          CALL ZERDE (ASD)
0426          CALL ZERDE (ASE)
0427          RETURN
0428          END
```

FTN4 COMPILER: HP92060-16092 REV. 2026 (800423)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00095 COMMON = 00000

SECTION II

AUTOPROBE UTILITY PROGRAM

REFERENCE: FINAL REPORT SECTION 2, SUBSECTION E.4,.....PAGE 2-26

1. HYBRD - UTILITY FUNCTION SELECTION
2. ENTER - CREATE NEW HYBRID NODE FILE
3. INTAB - INITIALIZE A, B REFERENCE POINTS
4. DSPLY - DISPLAY/PRINT HYBRID NODE FILE
5. MODFY - HYBRID NODE FILE EDITOR
6. TEST - AUTOPROBE/NODE FILE INTERACTIVE TEST

```

0001 FTN4,L
0002 PROGRAM MONPR
0003 C =====
0004 C
0005 C SON PROGRAM MONPR IS CALLED BY FATHER ROUTINE MATCH
0006 C AFTER #1 SON INTPR IS EXECUTED. MONPR IS CALLED
0007 C EVERY LOOP OF THE HYBRID TEST UNTIL IT IS COMPLETED.
0008 C MATCH PASSES HYBRID FILE NAME VIA RMPAR. MONPR
0009 C PASSES BACK PROGRAM TERMINATION STATUS VIA PRTN.
0010 C STATUS > 0 => CONTINUE TESTING HYBRID
0011 C STATUS < 0 => ABNORMAL TERMINATION, ABORT
0012 C
0013 C
0014 C INTEGER HNAME(3),MNAME(3),BFR(40),PARM(5),SPARM(16)
0015 C INTEGER ASD(3),ASE(3),STRGP(3),STRGR(4),STRG(5)
0016 C
0017 C EQUIVALENCE (ASD(1),BFR(8)), (ASE(1),BFR(11))
0018 C
0019 C DATA MNAME /2HSP,2HLH,2HBD/
0020 C DATA STRGP /2H P,2HRD,2HBE/, STRGR /2H R,2HEP,2HRO,2HBE/
0021 C DATA SPARM /0,0,2HSP,2HLH,2HBD,0,0,00B,423B,10,0,0,0,0,0,0/
0022 C
0023 C ----- RETRIEVE HYBRID TABLE FILE NAME -----
0024 C
0025 C CALL RMPAR (PARM)
0026 C DO 10 I=1,3
0027 10 HNAME(I) = PARM(I)
0028 C
0029 C ----- SEARCH FOR "PROBE" MESSAGE IN SPOOL FILE -----
0030 C
0031 C CALL SERCH (MNAME,6,STRGP,IERR,40,BFR)
0032 C IF (IERR.EQ.-12) GO TO 20
0033 C IF (IERR.NE.0) GO TO 90
0034 C GO TO 30
0035 C
0036 C ----- SEARCH FOR "REPROBE" MESSAGE IN SPOOL FILE -----
0037 C
0038 20 CALL SERCH (MNAME,8,STRGR,IERR,40,BFR)
0039 C IF (IERR.NE.0) GO TO 90
0040 C
0041 C ----- "PROBE"/"REPROBE" FOUND, LOOK FOR PIN NAME -----
0042 C ----- I.E. LOOK FOR A "U" IN MSG -----
0043 C
0044 30 DO 35 I=1,40
0045 C IU1 = IAND (BFR(I),177400B) + 40B
0046 C IF (IU1.EQ.2HU ) GO TO 40
0047 C IU2 = IAND (BFR(I),377B) + 20000B
0048 C IF (IU2.EQ.2H U) GO TO 40
0049 C IF (BFR(I).EQ.2HGN.OR.BFR(I).EQ.2H G) GO TO 40
0050 C IF (BFR(I).EQ.2HVC.OR.BFR(I).EQ.2H V) GO TO 40
0051 35 CONTINUE
0052 C IERR = 98
0053 C GO TO 90
0054 C
0055 C ----- PIN NAME FOUND, SEARCH IN HYBRID TABLE FILE -----

```

```

0056 C
0057 40 DO 45 J=1,5
0058 45 STRG(J) = BFR(I+J-1)
0059 C
0060 CALL SERCH (HNAME,10,STRG,IERR,13,BFR)
0061 IF (IERR.NE.0) GO TO 90
0062 C
0063 C ----- FIRST RECREATE LINK BETWEEN LU #63 AND FILE SPLHBD -----
0064 C
0065 CALL SPOPN (SPARM,ISLU)
0066 IERR = 97
0067 IF (ISLU.LT.0) GO TO 90
0068 IF (ISLU.NE.63) GO TO 90
0069 C
0070 C ----- NAME FOUND IN TABLE, SEND PIN COORDINATES OVER 408 BUS -----
0071 C ----- FORMAT: D <+/-> <COORD> E <+/-> <COORD> <CR> <LF> -----
0072 C
0073 LU = 36
0074 C
0075 ITMP = 0
0076 IEFLG = 0
0077 CALL READY (LU,IERR)
0078 IF (IERR.NE.0) CALL BEROR (IERR,ITMP,IEFLG)
0079 IF (IEFLG.LT.0) GO TO 95
0080 C
0081 ITMP = 0
0082 IEFLG = 0
0083 60 ICM1 = 2HD
0084 ICM2 = 2HE
0085 WRITE (LU,160) ICM1,ASD,ICM2,ASE
0086 160 FORMAT (2(A2,3A2))
0087 CALL READY (LU,IERR)
0088 IF (IERR.NE.0) CALL BEROR (IEFLG,ITMP,IEFLG)
0089 IF (IEFLG) 95,70,60
0090 C
0091 70 PARM(1) = 1
0092 C
0093 C ----- MONITORING DONE, SEND PROGRAM STATUS TO PAPA -----
0094 C
0095 80 CALL PRTN (PARM)
0096 CALL EXEC (6)
0097 C
0098 C ----- REPORT ERRORS -----
0099 C
0100 90 CALL ERROR (IERR)
0101 CALL CLEAR (LU,1)
0102 C
0103 95 PARM(1) = -1
0104 GO TO 80
0105 END

```



```
0096      SUBROUTINE GETAB (LU,ASX,ASY,IERR)
0097      C      =====
0098      C
0099      C
0100      INTEGER MSG(7),ASX(3),ASY(3)
0101      C
0102      C ----- REQUEST AND RECEIVE COORDINATES FROM BUS -----
0103      C ----- FORMAT SENDING:  SP <CR> <LF> -----
0104      C ----- FORMAT RECEIVING: X <COORD> Y <COORD> <CR> <LF> -----
0105      C
0106      CALL READY (LU,IERR)
0107      IF (IERR.NE.0) RETURN
0108      WRITE (LU,100)
0109      100  FORMAT ("SP")
0110      C
0111      READ (LU,110) MSG
0112      110  FORMAT (7A2)
0113      C
0114      C ----- CONVERT TO CHOSEN ASCII FORMAT AND RETURN -----
0115      C
0116      DO 20 I=1,3
0117      ASX(I) = MSG(I)
0118      20  ASY(I) = MSG(I+3)
0119      C
0120      ASX(1) = IAND (ASX(1),377B) + 20000B
0121      ASY(1) = IAND (ASY(1),377B) + 20000B
0122      C
0123      IERR = 0
0124      RETURN
0125      END
```

FTN4 COMPILER: HP92060-16092 REV. 2026 (800423)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00107 COMMON = 00000

```

0126      SUBROUTINE TEST (FNAME)
0127      C      =====
0128      C
0129      C
0130      INTEGER FNAME(3),DCB(144),BFR(13),MSG(18)
0131      INTEGER PINAM(5),ASD(3),ASE(3),AASD(3),AASE(3),BASD(3),BASE(3)
0132      C
0133      EQUIVALENCE (PINAM(1),BFR(1))
0134      EQUIVALENCE (ASD(1),BFR(8)), (ASE(1),BFR(11))
0135      C
0136      C ----- PREP FOR TEST - DEVICE CLEAR -----
0137      C
0138      LU = 36
0139      CALL CLEAR (LU,1)
0140      110  FORMAT (A2)
0141      115  FORMAT (18A2)
0142      C
0143      C ----- OPEN HYBRID TABLE FILE -----
0144      C ----- SET BIT 0 FOR NON-EXCLUSIVE OPEN -----
0145      C
0146      CALL GPLN (DCB,IERR,FNAME,18)
0147      IF (IERR.LT.0) GO TO 90
0148      C
0149      C ----- RETRIEVE POINTS A AND B THEN SEND COORDINATES -----
0150      C ----- FORMAT: AX <COORD> AY <COORD> <CR> <LF> -----
0151      C ----- FORMAT: BX <COORD> BY <COORD> <CR> <LF> -----
0152      C
0153      CALL READF (DCB,IERR,BFR)
0154      IF (IERR.LT.0) GO TO 90
0155      DO 16 I=1,3
0156      AASD(I) = ASD(I)
0157      16  AASE(I) = ASE(I)
0158      C
0159      CALL READF (DCB,IERR,BFR)
0160      IF (IERR.LT.0) GO TO 90
0161      DO 17 I=1,3
0162      BASD(I) = ASD(I)
0163      17  BASE(I) = ASE(I)
0164      C
0165      ITMP = 0
0166      IEFLG = 0
0167      CALL READY (LU,IERR)
0168      IF (IERR.NE.0) CALL BEROR (IERR,ITMP,IEFLG)
0169      IF (IEFLG.LT.0) GO TO 95
0170      18  ICM1 = 2HAX
0171      ICM2 = 2HAY
0172      WRITE (LU,118) ICM1,AASD,ICM2,AASE
0173      118  FORMAT (2(A2,3A2))
0174      C
0175      CALL READY (LU,IERR)
0176      IF (IERR.NE.0) CALL BEROR (IERR,ITMP,IEFLG)
0177      IF (IEFLG) 95,19,18
0178      C
0179      19  ITMP = 0
0180      IEFLG = 0

```

```

0181 20   ICM1 = 2HBX
0182     ICM2 = 2HBY
0183     WRITE (LU,118) ICM1,BASD,ICM2,BASE
0184 C
0185     CALL READY (LU,IERR)
0186     IF (IERR.NE.0) CALL BEROR (IERR,ITMP,IEFLG)
0187     IF (IEFLG) 95,21,20
0188 C
0189 C ----- LOAD HYBRID AND SET TEST MODE -----
0190 C
0191 21     ITMP = 0
0192     IEFLG = 0
0193 22     ICMD = 2HLD
0194     WRITE (LU,110) ICMD
0195     CALL READY (LU,IERR)
0196     IF (IERR.NE.0) CALL BEROR (IERR,ITMP,IEFLG)
0197     IF (IEFLG) 95,40,22
0198 C
0199 C ===== PRESENT MENU OF COMMANDS =====
0200 C
0201 40     ITMP = 0
0202     IEFLG = 0
0203     CALL READY (LU,IERR)
0204     IF (IERR.NE.0) CALL BEROR (IERR,ITMP,IEFLG)
0205     IF (IEFLG) 95,400,480
0206 C
0207 400    WRITE (1,1400)
0208 1400   FORMAT (//8X,"KEYBOARD TEST PROBE COMMANDS",
0209         &/"IN",14X,"INITIALIZE PROBE",
0210         &/"AXNNNNNAYNNNNN SEND REFERENCE POINT A",
0211         &/"BXNNNNNBYNNNNN SEND REFERENCE POINT B",
0212         &/"LD",14X,"LOAD HYBRID",/"UL",14X,"UNLOAD HYBRID",
0213         &/"M0",14X,"SET MODE TO AUTOMATIC",
0214         &/"M1",14X,"SET MODE TO MANUAL",/"M2",14X,"SET MODE TO TEST",
0215         &/"XNNNNNYNNNNN GO TO POSITION (X,Y)",
0216         &/"DSNNNNNESNNNNN GO TO POSITION (D,E)",
0217         &/"HO",14X,"GO TO HOME POSITION",
0218         &/"UP",14X,"RAISE PROBE",/"DN",14X,"LOWER PROBE",
0219         &/"MS (MESSAGE) SEND A MESSAGE",
0220         &/"??",14X,"REQUEST STATUS",
0221         &/"SP",14X,"REQUEST (X,Y) POSITION INFORMATION",
0222         &/"SD",14X,"REQUEST (D,E) POSITION INFORMATION",
0223         &/"SE",14X,"REQUEST ERROR CODE",
0224         &/"FT",14X,"TEST HYBRID FILE SEQUENTIALLY",
0225         &/"EX",14X,"EXIT KEYBOARD TEST"/)
0226 C
0227 C ----- ENTER A COMMAND -----
0228 C
0229 410    WRITE (1,1410)
0230 1410   FORMAT (11X,"ENTER A COMMAND: _")
0231     READ (1,115) MSG
0232 C
0233 415    ICMD = IAND (MSG(1),177400B) + 40B
0234     IF (MSG(1).EQ.2HUP.OR.MSG(1).EQ.2HDN.OR.MSG(1).EQ.2HH0) GOTO 420
0235     IF (MSG(1).EQ.2HM0.OR.MSG(1).EQ.2HM1.OR.MSG(1).EQ.2HM2) GOTO 420

```

```

0236      IF (MSG(1).EQ.2HLD.OR.MSG(1).EQ.2HUL) GO TO 420
0237      IF (MSG(1).EQ.2HMS) GO TO 435
0238      IF (ICMD.EQ.2HA .OR.ICMD.EQ.2HB .OR.ICMD.EQ.2HD ) GO TO 432
0239      IF (ICMD.EQ.2HX ) GO TO 430
0240      C
0241      ICMD = MSG(1)
0242      IF (ICMD.EQ.2H?? .OR.ICMD.EQ.2HSE) GO TO 440
0243      IF (ICMD.EQ.2HSP .OR.ICMD.EQ.2HSD) GO TO 440
0244      C
0245      IF (MSG(1).EQ.2HFT) GO TO 50
0246      IF (MSG(1).EQ.2HIN) GO TO 437
0247      IF (MSG(1).EQ.2HEX) GO TO 80
0248      GO TO 410
0249      C
0250      C ----- EXECUTE PROBE MOVE/MESSAGE/INITIALIZE COMMAND -----
0251      C
0252      420  WRITE (LU,110) MSG(1)
0253      GO TO 480
0254      C
0255      430  WRITE (LU,1430) (MSG(I),I=1,6)
0256      1430  FORMAT (6A2)
0257      GO TO 480
0258      C
0259      432  WRITE (LU,1432) (MSG(I),I=1,8)
0260      1432  FORMAT (8A2)
0261      GO TO 480
0262      C
0263      435  WRITE (LU,1435) (MSG(I),I=1,17)
0264      1435  FORMAT (17A2)
0265      GO TO 480
0266      C
0267      437  CALL CLEAR (LU,1)
0268      GO TO 480
0269      C
0270      C ----- EXECUTE STATUS/POSITION INFORMATION REQUEST -----
0271      C ----- DELAY 250 MILLISECONDS FOR SD COMMAND -----
0272      C
0273      440  WRITE (LU,110) ICMD
0274      IF (ICMD.EQ.2HSD) CALL EXEC (12,0,1,0,-25)
0275      READ (LU,115) MSG
0276      C
0277      IF (ICMD.EQ.2H??) GO TO 470
0278      IF (ICMD.EQ.2HSP) GO TO 460
0279      IF (ICMD.EQ.2HSD) GO TO 465
0280      C
0281      MSG(1) = MSG(1)/256
0282      WRITE (1,1450) MSG(1)
0283      1450  FORMAT (/"THE ERROR CODE IS ",I5)
0284      GO TO 480
0285      C
0286      460  MSG(1) = IAND (MSG(1),377B) + 24000B
0287      MSG(4) = IAND (MSG(4),377B) + 26000B
0288      WRITE (1,1460) (MSG(I),I=1,6)
0289      1460  FORMAT (/"THE (X,Y) POSITION IS ",6A2,"")
0290      GO TO 480

```

```

0291 C
0292 465 MSG(1) = IAND (MSG(1),377B) + 24000B
0293 MSG(4) = IAND (MSG(4),177400B) + 54B
0294 WRITE (1,1465) (MSG(1),I=1,7)
0295 1465 FORMAT (/ "THE (D,E) POSITION IS ",7A2,"")
0296 GO TO 480
0297 C
0298 470 IST = 2HUP
0299 IF (IAND (MSG(1),20000B).NE.0) IST = 2HDN
0300 WRITE (1,1470) IST
0301 1470 FORMAT (/ "PROBER POSITION? ",A2)
0302 IST = 2HND
0303 IF (IAND (MSG(1),10000B).NE.0) IST = 2HY
0304 WRITE (1,1472) IST
0305 1472 FORMAT ("PROBER BUSY? ",A2)
0306 IST = 2HY
0307 IF (IAND (MSG(1),4000B).NE.0) IST = 2HND
0308 WRITE (1,1474) IST
0309 1474 FORMAT ("PROBER ONLINE? ",A2)
0310 IST = 2HND
0311 IF (IAND (MSG(1),2000B).NE.0) IST = 2HY
0312 WRITE (1,1476) IST
0313 1476 FORMAT ("ERROR CONDITION? ",A2)
0314 IST = 2HAU
0315 IF (IAND (MSG(1),1400B).EQ.0) GO TO 478
0316 IST = 2HMN
0317 IF (IAND (MSG(1),1000B).EQ.0) GO TO 478
0318 IST = 2HTS
0319 IF (IAND (MSG(1),400B).EQ.0) GO TO 478
0320 IST = 2HID
0321 478 WRITE (1,1478) IST
0322 1478 FORMAT ("PROBER MODE? ",A2)
0323 C
0324 C ----- REINITIALIZE, THEN CONTINUE WITH TESTING -----
0325 C
0326 480 WRITE (1,1480)
0327 1480 FORMAT (/ "PRESS RETURN TO CONTINUE _")
0328 READ (1,110) N
0329 C
0330 DO 490 I=1,18
0331 490 MSG(1) = 2H
0332 GO TO 40
0333 C
0334 C ----- TEST EACH RECORD OF FILE SEQUENTIALLY -----
0335 C ----- NOW READ A RECORD AND RETRIEVE POINT (D,E) -----
0336 C ----- SKIP DELETED RECORDS. CHECK FOR EOF. -----
0337 C
0338 50 ICMD = 2H??
0339 WRITE (LU,110) ICMD
0340 READ (LU,115) MSG
0341 IF (IAND (MSG(1),1400B).EQ.1000B) GO TO 510
0342 C
0343 WRITE (1,1500)
0344 1500 FORMAT (/ "FIRST, SET MODE TO TEST")
0345 GO TO 480

```

```

0346 C
0347 510 CALL READF (DCB,IERR,BFR)
0348 IF (IERR.LT.0) GO TO 90
0349 IF (BFR(1).EQ.2H** .AND .BFR(2).EQ.2H**) GO TO 50
0350 IF (BFR(1).EQ.2H .AND .BFR(2).EQ.2H ) GO TO 40
0351 C
0352 C ----- DISPLAY PIN NAME, THEN CONTINUE -----
0353 C
0354 WRITE (1,1520) PINAM
0355 1520 FORMAT (// "THE NEXT PIN IS: ",SA2,
0356 &/"PRESS 'ENTER' TO CONTINUE")
0357 C
0358 C ----- MOVE PROBER TO HYBRID POSITION (D,E) -----
0359 C ----- FORMAT: D <+/-> <COORD> E <+/-> <COORD> <CR> <LF> -----
0360 C
0361 IF (IAND (ASD(1),177400B).EQ.20000B)
0362 & ASD(1) = IAND (ASD(1),377B) + 25400B
0363 IF (IAND (ASE(1),177400B).EQ.20000B)
0364 & ASE(1) = IAND (ASE(1),377B) + 25400B
0365 C
0366 ITMP = 0
0367 IEFLG = 0
0368 CALL READY (LU,IERR)
0369 IF (IERR.NE.0) CALL BEROR (IERR,ITMP,IEFLG)
0370 IF (IEFLG.LT.0) GO TO 95
0371 530 ICM1 = 2HD
0372 ICM2 = 2HE
0373 WRITE (LU,118) ICM1,ASD,ICM2,ASE
0374 C
0375 CALL READY (LU,IERR)
0376 IF (IERR.NE.0) CALL BEROR (IERR,ITMP,IEFLG)
0377 IF (IEFLG) 95,535,530
0378 535 ICMD = 2HSD
0379 WRITE (LU,110) ICMD
0380 CALL EXEC (12,0,1,0,-25)
0381 READ (LU,115) MSG
0382 C
0383 C ----- NOTE THE FILE AND PROBER POSITIONS -----
0384 C
0385 MSG(1) = IAND (MSG(1),377B) + 24000B
0386 MSG(4) = IAND (MSG(4),177400B) + 54B
0387 C
0388 WRITE (1,1540) PINAM,ASD,ASE,(MSG(I),I=1,7)
0389 1540 FORMAT (// "THE PIN IS: ",SA2,
0390 &/"CALCULATED POSITION IS: (" ,3A2," ,",3A2," )",
0391 &/"THE ACTUAL POSITION IS: ",7A2,"")
0392 C
0393 CALL EXEC (12,0,2,0,-3)
0394 GO TO 510
0395 C
0396 C ----- DONE, CLOSE FILE AND/OR REINITIALIZE PROBER -----
0397 C
0398 80 CALL CLOSE (DCB)
0399 WRITE (1,180)
0400 180 FORMAT (// "PROBER TEST COMPLETED")

```

```
1401          RETURN
1402 C
1403 C ----- REPORT ERRORS -----
1404 C
1405 90      CALL ERROR (IERR)
1406        CALL CLEAR (LU,1)
1407 95      CALL CLOSE (DCB)
1408        RETURN
1409        END
```

FTN4 COMPILER: HP92060-16092 REV. 2026 (800423)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 02031 COMMON = 00000

```
0410      SUBROUTINE READY (LU,IERR)
0411      C      =====
0412      C
0413      C
0414      INTEGER MSG(2)
0415      C
0416      C ----- DELAY 500 MILLISECONDS -----
0417      C
0418      IRESL = 1
0419      IMULT = 0
0420      IOFST = -50
0421      10  CALL EXEC (12,0,IRESL,IMULT,IOFST)
0422      C
0423      C ----- READY FOR DATA? CHECK "BUSY" AND "ERROR" BITS -----
0424      C
0425      ICMD = 2H??
0426      WRITE (LU,100) ICMD
0427      100  FORMAT (A2)
0428      READ (LU,110) MSG
0429      110  FORMAT (2A2)
0430      IF (IAND (MSG(1),10000B).NE.0) GO TO 10
0431      IF (IAND (MSG(1),2000B).NE.0) GO TO 20
0432      C
0433      IERR = 0
0434      RETURN
0435      C
0436      20  ICMD = 2HSE
0437      WRITE (LU,100) ICMD
0438      READ (LU,110) MSG
0439      IERR = MSG(1)
0440      RETURN
0441      END
```

FTN4 COMPILER: HP92060-16092 REV. 2026 (800423)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00104 COMMON = 00000

GE 0011 FTN. 8:50 PM MON., 14 DEC., 1981

```
12      SUBROUTINE BEROR (IERRC,IERRP,IEFLG)
13      C      =====
14      C
15      C
16      C ----- IS ERROR FATAL OR NON-FATAL? -----
17      C
18      C      IF (IAND (IERRC,100000B).NE.0) GO TO 30
19      C
20      C ----- NON-FATAL, IS IT THE SAME AS PREVIOUS CODE? -----
21      C
22      C      IERRC = IERRC/256
23      C      IF (IERRC.NE.IERRP) GO TO 10
24      C      IEFLG = IEFLG + 1
25      C      GO TO 20
26      C
27      C      IEFLG = 1
28      C      IERRP = IERRC
29      C
30      C ----- PRINT NON-FATAL ERROR MESSAGE -----
31      C
32      C      WRITE (1,120) IERRC
33      C      120  FORMAT (/"*** NON-FATAL BUS ERROR ",I5," ***)
34      C
35      C ----- SAME NON-FATAL ERROR 3 TIMES? IF YES, ABORT -----
36      C
37      C      IF (IEFLG.EQ.3) GO TO 40
38      C      PAUSE
39      C      RETURN
40      C
41      C ----- FATAL, PRINT FATAL ERROR MESSAGE AND ABORT -----
42      C
43      C      IERRC = IERRC/256
44      C      WRITE (1,130) IERRC
45      C      130  FORMAT (/"*** FATAL BUS ERROR ",I5," ***)
46      C
47      C      IEFLG = -99
48      C      CALL CLEAR (LU,1)
49      C      PAUSE
50      C      RETURN
51      C      END
```

FTN4 COMPILER: HP92060-16092 REV. 2026 (800423)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00125 COMMON = 00000

```

29      SUBROUTINE CALDE (XC,YC)
30      C      =====
31      C
32      C
33      COMMON /VALS/ XA,YA,XB,YB,XM,YM,ABSLP,CDSLPL,VFLG,HFLG
34      C
35      INTEGER VFLG,HFLG
36      C
37      C ----- CHANGE POINT TO NEW COORD SYSTEM USING MIDPOINT M -----
38      C
39      XC = XC - XM
40      YC = YC - YM
41      C
42      C ----- CHECK IF VERTICAL/HORIZONTAL REFERENCE -----
43      C
44      IF (VFLG) 30,10,40
45      IF (HFLG) 50,15,60
46      C
47      C ----- CALCULATE VALUES FOR COORDINATES D AND E -----
48      C
49      15 ZC = YC - (CDSLPL * XC)
50      D = ABS (ZC / (SQRT ((CDSLPL**2) + 1)))
51      C
52      E3 = SQRT ((ABSLP**2) + 1)
53      E = ABS ((YC - (ABSLP * XC)) / E3)
54      C
55      C ----- CALCULATE SIGNS FOR COORDINATES D AND E -----
56      C
57      XF = ZC / (ABSLP - CDSLPL)
58      YF = ABSLPL * XF
59      C
60      R1 = SQRT (((XF-XA) ** 2) + ((YA-YF) ** 2))
61      R2 = SQRT (((XB-XF) ** 2) + ((YF-YB) ** 2))
62      C
63      D = D + 0.5
64      IF (R2.LT.R1) D = -D
65      C
66      E = E + 0.5
67      K = 0
68      IF (YA-YB.GT.0.0) K = 4
69      IF (ABSLP.GT.0.0) K = K + 2
70      IF (YC.GT.YF) K = K + 1
71      IF (K.EQ.1.OR.K.EQ.2.OR.K.EQ.4.OR.K.EQ.7) E = -E
72      C
73      C ----- CALCULATIONS COMPLETED, RETURN D AND E -----
74      C
75      20 XC = D
76      YC = E
77      RETURN
78      C
79      C ----- VERTICAL REFERENCE, DETERMINE SIGNS AND VALUES -----
80      C
81      30 D = -YC
82      E = -XC
83      GO TO 20

```

9 HY10IEEE.SA:1 HY10IE IEEE INTERFACE ROUTINES

```

*
* DATA INPUT INTERRUPT - WHEN THE BI (BYTE I
* THE END (END OR IDENTIFY) BIT GETS SET, TH
* IS VECTORED TO. THIS ROUTINE WORKS OFF OF
* QUEUE AND WILL ACCEPT CHARACTERS UNTIL THE
*
0104 E6 E702 A DLSTN LDA HPREG2
0107 85 04 A BITA #LACBIT ;LISTEN ACTIVE STATE?
0109 26 07 0112 BNE DLST10 ;YES
*
* ERROR - NOT A LISTENER
*
010E C6 03 A LDB #NOLSTN
010D D7 AE A STB BUSERR
010F 7E 018E F JMP I3EXIT
*
0112 F6 E707 A DLST10 LDB HPREG7 ;GET CHARACTER
0115 96 E1 A LDA XFRST ;CURRENT TRANSMISSION
*
0117 81 FF A CMPA #ACTMOD ;IN ACTIVE STATE?
0119 26 17 0132 BNE DLST30 ;NO
*
011E C1 3F A CMPB #STCHAR ;IS IT STATUS CHARACTE
011D 26 0D 012C BNE DLST20 ;NO
*
011F 96 AA A LDA REG0SV
0121 85 02 A BITA #ENDBIT ;IS THISLAST BYTE OF M
0123 26 15 013A BNE DLST50 ;YES STORE CHAR ANYWAY
*
0125 86 00 A LDA #ALRMOD ;YES, SET TO ALARM MOD
0127 97 E1 A STA XFRST
0129 7E 018E F JMP I3EXIT
*
012C 86 02 A DLST20 LDA #DACMOD ;DEACTIVE STATUS ALARM
012E 97 E1 A STA XFRST
0130 20 08 013A BRA DLST50
*
0132 81 00 A DLST30 CMPA #ALRMOD ;IN ALARM MODE?
0134 27 3A 0170 BEQ DLST70 ;YES
*
0136 81 01 A CMPA #IGNMOD ;IGNORE CHAR?
0138 27 0F 0149 BEQ DLST60 ;YES
*
013A 8E 045E A DLST50 LDX #INECF ;POINTER TO INPUT BUF
013D ED 0000 A JSR PUTCHR ;GET CHAR FROM Q
0140 24 07 0149 BCC DLST60 ;NO ERRORS
*
* ERROR - INPUT QUEUE IS FULL
*
0142 C6 02 A DLST55 LDB #QUEFUL
0144 D7 AE A STB BUSERR
0146 7E 018E F JMP I3EXIT
*
* CHECK FOR END OR IDENTIFY TRANSMISSION
*
0149 96 AA A DLST60 LDA REG0SV
014E 85 02 A BITA #ENDBIT ;WAS THIS LAST BYTE?
014D 27 1E 016D BEQ DLST65 ;NO

```

008 HY10IEEE.SA:1 HY10IE IEEE INTERFACE ROUTINES

```

*
*  COMMAND INTERRUPT - LISTEN FOR COMMAND ONL
*  UPON INTERRUPT OF THE CMD BIT SET OF THE
*  INTERRUPT MASK REGISTER, THIS ROUTINE WI
*  VECTORED TO BY THE IRQ BUS HANDLER. THE
*  PURPOSE OF THE ROUTINE IS TO DETERMINE W
*  COMMAND IS REQUESTED AND SET THE APPROPR
*  VARIAELES AND FLAGS.
*
P 00C2 86   E701   A CLSTN  LDA    HFREG1
P 00C5 85   08     A      BITA  #RLCBIT  ;WAS IT REM/LOC CHANGE
P 00C7 27   0D    00D6  BEQ    CLST20  ;NO
*
*  CHANGE IN REMOTE/LOCAL STATE HAS OCCURED
*
P 00C9 0F   E0     A      CLR    OFMODE  ;SET TO LOCAL
P 00CB 85   40     A      BITA  #REMBIT  ;IS REMOTE ENAEBLED?
P 00CD 1027 00BD 018E  LBEQ   I3EXIT  ;NO
*
P 00D1 03   E0     A      COM    OFMODE  ;SET TO REMOTE
P 00D3 7E   018E   F      JMP    I3EXIT
*
*  CHECK FOR SERIAL POLL ACTIVE STATE
*
P 00D6 85   04     A CLST20 BITA  #SFABIT  ;IS SPAS BIT SET?
P 00D8 27   06    00E0  BEQ    CLST40  ;NO
*
P 00DA 7F   E705   A      CLR    HFREG5  ;NEVER US FOR SERIAL F
P 00DD 16   00AE 018E  LERA   I3EXIT
*
*  CHECK FOR DEVICE CLEAR ACTIVE STATE
*
P 00E0 85   02     A CLST40 BITA  #DCABIT  ;IS DCAS BIT SET?
P 00E2 27   08    00EC  BEQ    CLST60  ;NO
*
P 00E4 86   10     A      LDA    #10
P 00E6 B7   E703   A      STA    HFREG3  ;RELEASE DAC HOLDOFF
P 00E9 7E   0000   A      JMP    INTDEV  ;INIT DEVICE
*
P 00EC 85   01     A CLST60 BITA  #UUCBIT  ;IS IT UNDEF CMD?
P 00EE 26   08    00F8  BNE    CLST70  ;YES
P 00F0 85   80     A      BITA  #UACBIT  ;ADDRESS CMD UNDEF?
P 00F2 26   04    00F8  BNE    CLST70  ;YES
*
*  UNKNOWN CMD HAS BEEN RECEIVED
*
P 00F4 86   05     A      LDA    #UKINTR
P 00F6 20   07    00FF  ERA    CLST80
*
*  UNDEFINED UNIVERSAL COMMAND RECEIVED
*
P 00F8 86   10     A CLST70 LDA    #10
P 00FA B7   E703   A      STA    HFREG3  ;RELEASE DAC HOLDOFF
P 00FD 86   04     A      LDA    #EDUCMD
*
P 00FF 97   AE     A CLST80 STA    EUSERR
P 0101 7E   018E   F      JMP    I3EXIT
*

```

107 HY10IEEE.SA:1 HY10IE IEEE INTERFACE ROUTINES

```

0093 25 09 009E      BCS    TALK75  ;THIS IS A PROBLEM????
      *
0095 C1 0A      A      CMPB   #STRTRM ;END OF MESSAGE?
0097 27 07 00A0      BEQ    TALK80  ;YES
      *
0099 F7 E707      A      STB    HPREG7  ;NO,SEND OUT DATA BYTE
009C 20 21 00EF      BRA    TALKRT
      *
      *  TRANSFER IS NOW COMPLETE
      *
009E C6 0A      A TALK75 LDB    #STRTRM ;FORCE END OF STRING
00A0 B6 20      A TALK80 LDA    #20    ;SET EOI LINE
00A2 B7 E703      A      STA    HPREG3  ;TO HOLD OFF HANDSHAKE
      *
00A5 12                NOP                    ;MAY BE USED FOR TOGGL
00A6 12                NOP                    ;OF HPREG0
00A7 12                NOP
00A8 12                NOP
00A9 B6 EF      A      LDA    #EF    ;TURN OFF B0 BIT
00AB B7 E700      A      STA    HPREG0
00AE F7 E707      A      STB    HPREG7  ;SEND LAST CHARACTER
      *
00E1 0F AD      A      CLR    XINPRG  ;SET TO IDLE
00E3 E8 046C      A      LDX    OUTBCF+6 ;INIT BUFFER
00E6 EF 0468      A      STX    OUTBCF+2
00E9 EF 0466      A      STX    OUTBCF
00EC 0F AE      A      CLR    OUTCNT  ;Q EMPTY
00EE 3E                RTI
      *
00EF 7E 018E      P TALKRT JMP    I3EXIT

```

006 HY10IEEE.SA:1 HY10IE IEEE INTERFACE ROUTINES

```

17      *
18      *   TALKER INTERRUPT - THIS ROUTINE IS TRIGGE
19      *   THE BYTE OUT "B0" BIT OF THE INTERRUPT S
20      *   REGISTER (HPREG0) GETS SET. EAC TIME THI
21      *   BIT GETS SET, A SINGLE BYTE WILL BE SENT
22      *   CONTROLLER. CURRENT PHILOSOPHY IS IF A
23      *   IS IN PROGRESS AND THE STATUS TASK FLAG
24      *   (STTASK), ABORT THE CURRENT TRANSMISSION
25      *   STATUS.
26      *
27P 004E B6 E702 A TALKER LDA HPREG2 ;READ ADDRESS STAT REG
28P 004E B5 08 A BITA #TACBIT ;IN TALKER ACTIVE STAT
29P 0050 26 07 0059 BNE TALK10 ;YES
30
31      *
32      *   ERROR - NOT CURRENTLY ADDRESSED TO TALK
33      *
34P 0052 C6 06 A LDB #NOTALK
35P 0054 D7 AE A STB BUSERR
36P 0056 7E 018E F JMP I3EXIT
37
38      *
39P 0059 0D AC A TALK10 TST STTASK ;STATUS REQUIRED?
40P 005B 27 25 0082 BEQ TALK50 ;NO
41
42      *
43      *   STATUS REQUESTED OF PROBE
44      *
45P 005D 0F AC A CLR STTASK ;RESET STATUS ENABLE
46P 005F 0D AD A TST XINPRG ;TRANSFER IN PROGRESS
47P 0061 26 02 0065 BNE TALK20 ;YES
48
49      *
50P 0063 03 AD A COM XINPRG ;SET TRANSFER IN PROGR
51
52      *
53P 0065 8E 0466 A TALK20 LDX #OUTBCF ;PTR TO BCF
54P 0068 CC 04D4 A LDD #OUTBUF ;BEG OF BUF
55P 006B 108E 0539 A LDY #OUTEND ;END OF BUF
56P 006F ED 0000 A JSR INTBCF ;INIT BCF FIELD
57
58      *
59P 0072 ED 0000 A JSR FORMST ;FORM STATUS BYTE
60P 0075 8E 0466 A LDX #OUTBCF ;PTR TO OUTPUT BCF
61P 0078 1F 89 A TFR A,B
62P 007A ED 0000 A JSR PUTCHR ;SAVE STATUS BYTE
63P 007D ED 0000 A JSR DTCRLF ;OUTPUT CR/LF
64P 0080 20 0E 008D BRA TALK70
65
66      *
67      *   CHECK FOR DATA TO SEND
68      *
69P 0082 0D AD A TALK50 TST XINPRG ;IS TRANSFER IN PROGRE
70P 0084 26 07 008D BNE TALK70 ;YES SEND NEXT BYTE
71
72      *
73P 0086 0D AE A TST OUTCNT ;ANY MESSAGES WAITING?
74P 008B 26 01 008B BNE TALK60 ;YES
75P 008A 3E RTI
76
77      *
78P 008B 03 AD A TALK60 COM XINPRG ;SET TRANSFER TO IN PR
79
80      *
81      *   SEND OUT THE NEXT DATA BYTE
82      *
83P 008D 8E 0466 A TALK70 LDX #OUTBCF
84P 0090 ED 0000 A JSR GETCHR

```

005 HY10IEEE.SA:1 HY10IE IEEE INTERFACE ROUTINES

*
 * INITIALIZATION PROCESS TO SET UP THE IEEE
 * HFIB INTO THE INTERRUPT DRIVEN MODE.
 *

P 0031	B6	E704	A	INTI3E	LDA	HFREG4	;READ THE ADDRESS SWIT
P 0034	B4	1F	A		ANDA	##1F	;MASK THE BASE ADDRESS
P 0036	B7	E704	A		STA	HFREG4	;INTERP ON LISTEN TALK
P 0039	B6	80	A		LDA	##80	;SET RESET BIT
IP 003B	B7	E703	A		STA	HFREG3	
P 003E	4F				CLRA		
IP 003F	B7	E703	A		STA	HFREG3	;RESET 488 CIP
P 0042	C6	FF	A		LDB	##INTMOD	;CURRENT INTERRUPT
IP 0044	F7	E700	A		STB	HFREG0	;CONFIGURATION
IP 0047	B7	E702	A		STA	HFREG2	;NO SPECIAL FUNC
5P 004A	39				RTS		

GE 004 HY10IEEE.SA:1 HY10IE IEEE INTERFACE ROUTINES

```

119          *
120          * IEEE-68488 INTERRUPT PROCESSOR
121          *
122          * THIS SYSTEM IS CONFIGURED SUCH THAT WHEN
123          * 68488 BUS INTERRUPT IS GENERATED BY A CON
124          * THE HARDWARE IS "HARDWIRED" TO VECTOR TO
125          * WHERE IT WILL BE DETERMINED THE VALIDITY
126          * THE INTERRUPT AND DIRECT IT TO THE APPROP
127          * SUB-HANDLER.
128          *
0129F 0000 B6 E700 A IRQI3E LDA HPREG0 ;READ INTERRUPT REG
0130F 0003 2A 29 002E BFL IRQNOT ;TRY FOR KEYBOARD
0131          *
0132F 0005 97 AA A STA REG0SV ;SAVE COMMAND INFO
0133          *
0134F 0007 85 01 A BITA #BIBIT ;IS IT BI?
0135F 0009 1026 00F7 0104 LENE DLSTN ;YES
0136          *
0137F 000D 85 02 A BITA #ENDBIT ;IS IT END BIT?
0138F 000F 1026 00F1 0104 LENE DLSTN ;YES
0139          *
0140F 0013 85 04 A BITA #CMDBIT ;IS IT CMD BIT
0141F 0015 1026 00A9 00C2 LENE CLSTN ;YES
0142          *
0143F 0019 85 40 A BITA #BOBIT ;IS IT BO BIT?
0144F 001B 26 2E 004B BNE TALKER ;YES
0145          *
0146F 001D 85 20 A BITA #GETBIT ;GROUP EXEC TRIG?
0147F 001F 27 08 0029 BEQ IRQERR ;NO
0148          *
0149F 0021 86 10 A LDA #10
0150F 0023 B7 E703 A STA HPREG3 ;RELEASE DAC HOLDOFF
0151F 0026 7E 018E F JMP I3EXIT
0152          *
0153F 0029 C6 01 A IRQERR LDB #BADIRQ ;ILLEGAL INTERRUPT
0154F 002B D7 AB A STB BUSERR
0155F 002D 3B RTI
0156          *
0157F 002E 7E 0000 A IRQNOT JMP KINT ;KEY PROCESSOR

```


193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217

```

*
*XXXXXXXXXXXXX
*
*   NAME:   IEEE INTERFACE MODULE
*
*   FUNCTION: THIS MODULE WILL HANDLE ALL
*              COMMUNICATIONS BETWEEN THE DTS-
*              SYSTEM AND THE 2460 AUTOPROBER.
*              BASIC LISTENER/TALKER ROUTINES
*              IN THIS MODULE AND WORK OFF OF
*              QUEUE STRUCTURE.
*
*   CALLING SEQUENCE:
*              JSR INTI3E ;INIT IEEE BUS
*
*   NOTE: WHENEVER AN IRQ INTERRUPT OCCURS AN
*          IT IS NOT A TIMER INTERRUPT DO
*
*              JMP IRQI3E
*
*   REV      DATE      BY      REASON
*   <<<<    <<<<<    <<    <<<<<<<
*   00      10/22/82   SJF      INIT RELEA
*

```

```

00059      00FF      A INTMOD EQU      $FF      ;INTERRUPT ON ANYTHING
0  60
0  61      *
00062      * IEEE-68488 INTERFACE REGISTERS
00063      *
00064      E700      A HPREGS EQU      $E700
00065      E700      A HPREG0 EQU      HPREGS
00066      E701      A HPREG1 EQU      HPREGS+1
00067      E702      A HPREG2 EQU      HPREGS+2
00068      E703      A HPREG3 EQU      HPREGS+3
00069      E704      A HPREG4 EQU      HPREGS+4
00070      E705      A HPREG5 EQU      HPREGS+5
00071      E706      A HPREG6 EQU      HPREGS+6
00072      E707      A HPREG7 EQU      HPREGS+7
00073      *
00074
00075      *
00076      * LOW MEMORY EQUATES
00077      *
00078      00AA      A REG0SV EQU      170      ;SAVE COMMAND
00079      00AB      A BUSERR EQU      171      ;BUS ERROR CODE
00080      00AC      A STTASK EQU      172      ;SEND STATUS FLAG
00081      00AD      A XINPRG EQU      173      ;SET WHEN TRANSFER IN
00082      00AE      A OUTCNT EQU      174      ;SET WHEN MSG IN OUTBU
00083      00AF      A INCNT EQU      175      ;COUNT OF MSGS IN INBU
00084      00B0      A OPMODE EQU      176      ;OPERATING MODE LOC/RE
00085      00B1      A XFRST EQU      177      ;TRANSFER STATUS
0  36
00087      045E      A INBCF EQU      1118     ;INPUT BUF CONTROL FIE
00088      0466      A OUTBCF EQU      1126     ;OUTPUT BUF CONTROL FI
00089      04D4      A OUTBUF EQU      1236     ;100 BYTE OUTPUT BUFFE
00090      0539      A OUTEND EQU      1337     ;END OFOUTPUT BUFFER
00091

```

```

00001          NAM      HY10IEEE
0  )2         OPT      REL,CRE,NOG
C  )3
00004          TTL      IEEE INTERFACE ROUTINES
00005          *
00006          * INTERNAL REFERENCES
00007          *
00008          XDEF     IRQI3E,INTI3E
00009
00010          *
00011          * EXTERNAL REFERENCES
00012          *
00013          XREF     INTECF,FORMST
00014          XREF     PUTCHR,GETCHR,OTCRLF
00015          XREF     INTDEV,KINT
00016
00017          *
00018          * THE FOLLOWING ARE BYTE INTERRUPT EQUATES
00019          *
00020          0001      A BIBIT EQU      $01
00021          0002      A ENDBIT EQU   $02
00022          0004      A CMDBIT EQU   $04
00023          0020      A GETBIT EQU   $20
00024          0040      A BOBIT EQU   $40
00025          *
00026          * BUS ERROR EQUATES
00027          *
00028          0001      A BADI3EQ EQU   $1
00029          0002      A QUEFUL EQU   $2
00030          0003      A NOLSTN EQU   $3
00031          0004      A BDUCMD EQU   $4
00032          0005      A UKINTR EQU   $5
00033          0006      A NOTALK EQU   $6
00034          0007      A BDSTRG EQU   $7
00035          *
00036          * STATE BITS
00037          *
00038          0040      A REMBIT EQU   $40
00039          0008      A RLCBIT EQU   $08
00040          0008      A TACBIT EQU   $08
00041          0004      A LACBIT EQU   $04
00042          0080      A UACBIT EQU   $80
00043          0001      A UUCBIT EQU   $01
00044          0002      A DCABIT EQU   $02
00045          0004      A SPABIT EQU   $04
00046          *
00047          * MODES OF TRANSMISSION
00048          *
00049          00FF      A ACTMOD EQU   $FF      ;ACTIVE MODE
00050          0000      A ALRMOD EQU   $00      ;ALARM MODE
00051          0001      A IGNMOD EQU   $01      ;IGNORE MODE
00052          0002      A DACMOD EQU   $02      ;DEACTIVATE MODE
00053
00054          *
00055          * OTHER EQUATES
00056          *
00057          000A      A STRTRM EQU   $0A      ;STRING TERMINATOR
00058          003F      A STCHAR EQU   $3F      ;STATUS CHARACTER

```

SECTION III

IEEE-488 DRIVER

REFERENCE: FINAL REPORT SECTION 2, SUBSECTION E.5,.....PAGE 2-27

1. IRQI3E - IEEE-488 BUS INTERRUPT PROCESSOR
2. INTI3E - IEEE-488 HARDWARE INITIALIZATION
3. TALKER - IEEE-488 BUS DATA OUTPUT
4. CLSTN - IEEE-488 BUS COMMAND PROCESSOR
5. DLSTN - IEEE-488 BUS DATA INPUT

PAGE 0016 DSPLY 8:46 PM MON., 14 DEC., 1981

```
0594 C
0595 C ----- REPORT ERRORS -----
0596 C
0597 90 CALL CLOSE (DCB)
0598 95 CALL ERROR (IERR)
0599 RETURN
0600 END
```

FTN4 COMPILER: HP92060-16092 REV. 2026 (800423)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00442 COMMON = 00000

```

0539      SUBROUTINE DSPLY (FNAME)
0540      C      =====
0541      C
0542      C
0543      INTEGER FNAME(3),DCB(144),BFR(13)
0544      INTEGER PINAM(5),ASD(3),ASE(3)
0545      C
0546      EQUIVALENCE (PINAM(1),BFR(1))
0547      EQUIVALENCE (ASD(1),BFR(8)),(ASE(1),BFR(11))
0548      C
0549      C -----      OPEN HYBRID TABLE FILE      -----
0550      C -----      SET BIT 0 FOR NON-EXCLUSIVE OPEN      -----
0551      C
0552      CALL OPEN (DCB,IERR,FNAME,1R)
0553      IF (IERR.LT.0) GO TO 95
0554      C
0555      C -----      REQUEST LISTING DEVICE - CRT/LP      -----
0556      C
0557      20      WRITE (1,120)
0558      120      FORMAT ("DISPLAY DEVICE - (1) CRT OR (6) LP? _")
0559      READ (1,*) LU
0560      IF (LU.NE.1.AND.LU.NE.6) GO TO 20
0561      C
0562      C -----      WRITE OUT DISPLAY HEADINGS      -----
0563      C
0564      WRITE (LU,140) (FNAME(I),I=1,3)
0565      140      FORMAT (//12X,"HYBRID TABLE FILE ",3A2,
0566      &///" PIN NAME",6X,"X",6X,"Y",9X,"D",7X,"E"//)
0567      C
0568      C -----      READ A RECORD      -----
0569      C -----      SKIP DELETED RECORDS AND CHECK FOR EOF      -----
0570      C
0571      50      CALL READ (DCB,IERR,BFR)
0572      IF (IERR.LT.0) GO TO 90
0573      IF (BFR(1).EQ.2H** .AND. BFR(2).EQ.2H**) GO TO 50
0574      IF (BFR(1).EQ.2H .AND. BFR(2).EQ.2H ) GO TO 70
0575      C
0576      C -----      DISPLAY THE RECORD - PIN NAME,(X,Y),(D,E)      -----
0577      C
0578      WRITE (LU,160) PINAM,BFR(6),BFR(7),ASD,ASE
0579      160      FORMAT (" ",5A2,19,I7,5X,3A2,2X,3A2)
0580      C
0581      GO TO 50
0582      C
0583      C -----      EOF, CLOSE FILE AND RETURN      -----
0584      C
0585      70      IF (LU.NE.6) GO TO 80
0586      DO 75 I=1,4
0587      75      WRITE (LU,175)
0588      175      FORMAT ("1")
0589      C
0590      80      CALL CLOSE (DCB)
0591      WRITE (1,180)
0592      180      FORMAT (// "TABLE FILE DISPLAY COMPLETED")
0593      RETURN

```

```

0497      SUBROUTINE ZERDE (ASC)
0498 C      =====
0499 C
0500 C
0501      INTEGER ASC(3)
0502 C
0503      DO 20 I=1,3
0504 C
0505 C ----- CHECK MSB. IF BLANK, CHANGE TO ZERO -----
0506 C
0507      IF (IAND (ASC(I),177400B).NE.20000B) GO TO 30
0508      ASC(I) = IAND (ASC(I),377B) + 30000B
0509 C
0510 C ----- CHECK LSB. IF BLANK, CHANGE TO ZERO -----
0511 C
0512      IF (IAND (ASC(I),377B).NE.40B) GO TO 40
0513      ASC(I) = IAND (ASC(I),177400B) + 60B
0514 C
0515 20    CONTINUE
0516      RETURN
0517 C
0518 C ----- MSB IS NOT BLANK. CHECK IF IT IS NEGATIVE SIGN -----
0519 C
0520 30    IF (IAND (ASC(I),177400B).NE.26400B) GO TO 50
0521      ASC(I) = IAND (ASC(I),377B) + 30000B
0522      GO TO 45
0523 C
0524 C ----- LSB IS NOT BLANK. CHECK IF IT IS NEGATIVE SIGN -----
0525 C
0526 40    IF (IAND (ASC(I),377B).NE.55B) GO TO 50
0527      ASC(I) = IAND (ASC(I),177400B) + 60B
0528 C
0529 C ----- COORDINATE IS NEGATIVE -----
0530 C
0531 45    ASC(1) = IAND (ASC(1),377B) + 26400B
0532      RETURN
0533 C
0534 C ----- COORDINATE IS POSITIVE -----
0535 C
0536 50    ASC(1) = IAND (ASC(1),377B) + 20000B
0537      RETURN
0538      END

```

FTN4 COMPILER: HP92060-16092 REV. 2026 (800423)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00142 COMMON = 00000

```
0484 C
0485 40 D = YC
0486 E = XC
0487 GO TO 20
0488 C
0489 C ----- HORIZONTAL REFERENCE, DETERMINE SIGNS AND VALUES -----
0490 C
0491 50 XC = -XC
0492 RETURN
0493 C
0494 60 YC = -YC
0495 RETURN
0496 END
```

FIN4 COMPILER: HP92060-16092 REV. 2026 (800423)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00329 COMMON = 00000


```

00382          *
01 13          * TRANSMISSION OF THIS MESSAGE IS COMPLETE
7 84          *
00385P 014F C1 0A A CMPEB #STRTRM ;STRING TERMINATOR?
00386P 0151 27 0E 0161 BEQ DLST62 ;YES
00387          *
00388P 0153 86 07 A LDA #EDSTRG ;SET ERROR
00389P 0155 97 AB A STA EUSERR
00390          *
00391P 0157 C6 0A A LDB #STRTRM ;PUT IN TERMINATOR
00392P 0159 8E 045E A LDX #INEICF
00393P 015C 8D 0000 A JSR PUTCHR
00394P 015F 25 E1 0142 BCS DLST55 ;BUFFER FULL
00395          *
00396P 0161 96 E1 A DLST62 LDA XFRST
00397P 0163 81 01 A CMPA #IGNMOD ;IN IGNORE MODE?
00398P 0165 27 02 0169 BEQ DLST63 ;YES
00399P 0167 0C AF A INC INCNT
00400P 0169 86 FF A DLST63 LDA #ACTMOD ;SET TO ACTIVE FOR NEX
00401P 016E 97 E1 A STA XFRST
00402P 016D 7E 018E P DLST65 JMP I3EXIT
00403          *
00404          * CHECK FOR STATUS REQUEST
00405          *
00406P 0170 C1 3F A DLST70 CMPEB #STCHAR ;IS IT STATUS CHAR?
00407P 0172 26 0A 017E BNE DLST80 ;NO
00408          *
00409P 0174 C6 FF A LDB #$FF
00410P 0176 D7 AC A STB STTASK
00411          *
00412P 0178 C6 01 A LDB #IGNMOD ;IGNORE REST OF BYTES
00413P 017A D7 E1 A STB XFRST
00414P 017C 20 CB 0149 BRA DLST60 ;CHECK END BIT BEFORE
00415          *
00416P 017E 34 04 A DLST80 PSHS B ;SAVE CHAR
00417P 0180 8E 045E A LDX #INEICF
00418P 0183 C6 3F A LDB #STCHAR ;PUT IN PREVIOUS CHAR
00419P 0185 8D 0000 A JSR PUTCHR
00420P 0188 35 04 A PULS B ;RESTORE CUR CHAR
00421P 018A 25 E6 0142 BCS DLST55 ;Q FULL - ERROR
00422P 018C 20 AC 013A BRA DLST50
00423

```

```

00425
01 16 *
C 27 * THE FOLLOWING ROUTINE WILL SET THE BUS IN
00428 * ANOTHER INTERRUPT IS PENDING. OTHERWISE,
00429 * EXITING FROM INTERRUPT WILL OCCURR.
00430 *
00431 018E P I3EXIT EQU *
00432P 018E 4F CLRA ;TOGGLE INTR MASK
00433P 018F B7 E700 A STA HFREG0
00434P 0192 43 COMA
00435P 0193 B7 E700 A STA HFREG0
00436P 0196 3E RTI
00437
00438 END
TOTAL ERRORS 00000--00000
TOTAL WARNINGS 00000--00000

```

```

00FF ACTMOD 00049*00343 00400
0000 ALRMOD 00050*00353 00361
0001 BDIRQ 00028*00153
0007 BDSTRG 00034*00388
0004 EDUCMD 00031*00319
0001 EIBIT 00020*00134
0040 EOBIT 00024*00143
00AE BUSERR 00079*00154 00194 00321 00337 00374 00389
J0D6 CLST20 00277 00290*
00E0 CLST40 00291 00298*
P 00EC CLST60 00299 00305*
P 00FB CLST70 00306 00308 00317*
P 00FF CLST80 00313 00321*
P 00C2 CLSTN 00141 00275*
0004 CMDEIT 00022*00140
0002 DACMOD 00052*00357
0002 DCABIT 00044*00298
P 0112 DLST10 00332 00340*
P 012C DLST20 00347 00357*
P 0132 DLST30 00344 00361*
P 013A DLST50 00351 00359 00367*00422
P 0142 DLST55 00373*00394 00421
P 0149 DLST60 00365 00369 00379*00414
P 0161 DLST62 00386 00396*
P 0169 DLST63 00398 00400*
P 016D DLST65 00381 00402*
P 0170 DLST70 00362 00406*
P 017E DLST80 00407 00416*
P 0104 DLSTN 00135 00138 00330*
0002 ENDBIT 00021*00137 00350 00380
R FORMST 00013*00213
0020 GETBIT 00023*00146
R GETCHR 00014*00234
E700 HFREG0 00065*00129 00172 00254 00433 00435
E701 HFREG1 00066*00275
E702 HFREG2 00067*00174 00187 00330
E703 HFREG3 00068*00150 00168 00170 00247 00302 00318
E704 HFREG4 00069*00163 00165
E705 HFREG5 00070*00293

```

E706 HPREG6 00071*
 707 HPREG7 00072*00240 00255 00340
 E700 HPREGS 00064*00065 00066 00067 00068 00069 00070 00071 00072
 018E I3EXIT 00151 00195 00264 00283 00286 00294 00322 00338 00355
 00375 00402 00431*
 0001 IGNMOD 00051*00364 00397 00412
 045E INBCF 00087*00367 00392 00417
 00AF INCNT 00083*00399
 R INTBCF 00013*00211
 R INTDEV 00015*00303
 DP 0031 INTI3E 00008 00163*
 00FF INTMOD 00059*00171
 P 0029 IRQERR 00147 00153*
 DP 0000 IRQI3E 00008 00129*
 P 002E IRQNOT 00130 00157*
 R KINT 00015*00157
 0004 LACBIT 00041*00331
 0003 NOLSTN 00030*00336
 0006 NOTALK 00033*00193
 00B0 OFMODE 00084*00281 00285
 R OTCRLF 00014*00217
 0466 OUTECF 00088*00208 00214 00233 00258 00259 00260
 04D4 OUTBUF 00089*00209
 00AE OUTCNT 00082*00225 00261
 0539 OUTEND 00090*00210
 R FUTCHR 00014*00216 00368 00393 00419
 0002 QUEFUL 00029*00373
 J0AA REG0SV 00078*00132 00349 00379
 0040 REMBIT 00038*00282
 0008 RLCBIT 00039*00276
 0004 SPABIT 00045*00290
 003F STCHAR 00058*00346 00406 00418
 000A STRTRM 00057*00237 00245 00385 00391
 00AC STASK 00080*00197 00202 00410
 0008 TACBIT 00040*00188
 P 0059 TALK10 00189 00197*
 P 0065 TALK20 00204 00208*
 P 0082 TALK50 00198 00222*
 P 008E TALK60 00226 00229*
 P 008D TALK70 00218 00223 00233*
 P 009E TALK75 00235 00245*
 P 00A0 TALK80 00238 00246*
 P 004E TALKER 00144 00187*
 P 00EF TALKRT 00241 00264*
 0080 UACBIT 00042*00307
 0005 UKINTR 00032*00312
 0001 UUCBIT 00043*00305
 00E1 XFRST 00085*00341 00354 00358 00396 00401 00413
 00AD XINFRG 00081*00203 00206 00222 00229 00257

SECTION IV

EXECUTIVE/COMMAND PROCESSOR

REFERENCE: FINAL REPORT SECTION 2, SUBSECTION E.3,....PAGE 2-24

1. EXEC - AUTOPROBE EXECUTIVE
2. CMINT - COMMAND INTERPRETOR/PROCESSOR

```

00001      NAM      EXEC
00002      OPT      REL,CRE,NOG
00003
00004      TTL      HY16EXEC 2460 EXECUTIVE PROCESS
00005
00006      *      DATE:      06/29/81
00007      *      10/26/81 MODS FOR DTS-70
00008
00009      *      INTERNAL DEFINITION
00010
00011
00012      XDEF      EXEC,EXCNTR,UNLDKY,LOADKY
00013      XDEF      COFFST
00014
00015
00016      *      EXTERNAL DEFINITION
00017
00018
00019      XREF      IKBPIA,DELAY2,ONSTOP,DSKOFF
00020      XREF      DSPL,START,CMINTF,TCLRDF
00021      XREF      ONLTE,OFFLTE,UPPROB,DNPROB
00022      XREF      HOPROB,AUTPRB,MANPRB,TSTPRB
00023      XREF      ENTER,CBHEX,CUROFF,FLSHYN
00024      XREF      ABSET,MATCH,RERROR,RSTART
00025      XREF      OFLNE,CAMOFF,MVPARK
00026
00027
00028      *      KEY BOARD LIGHT EQUATES
00029
00030      0041      A LRUN      EQU      $41      ;RUN LIGHT
00031      0043      A LLSTOP   EQU      $43      ;STOP LIGHT
00032      0035      A LUNLOD   EQU      $35      ;UNLOAD LIGHT
00033      0034      A LLOAD    EQU      $34      ;LOAD LIGHT
00034      0032      A LOFSET   EQU      $32      ;OFFSET LIGHT
00035
00036
00037      *      KEY BOARD KEY EQUATES
00038
00039
00040      002D      A KLSTOP   EQU      $2D      ;STOP/OFFLINE
00041      002C      A KRUN     EQU      $2C      ;RUN/ONLINE
00042      0030      A KABRT    EQU      $30      ;ABORT
00043      0025      A KUPDOWN  EQU      $25      ;OFFLINE PROBE UP
00044      0026      A KUNLOD  EQU      $26      ;HOME PROBE
00045      002A      A KAUTO    EQU      $2A      ;SET TO AUTO MODE
00046      0028      A KSTEP   EQU      $28      ;SET TO STEP MODE
00047      0029      A KTEST   EQU      $29      ;SET TO TEST MODE
00048      0031      A KINFAD  EQU      $31      ;INPUT DEBUG ADDR
00049      0041      A KDSPNX  EQU      $41      ;DISPLAY NEXT DEBUG AD
00050      0020      A KABLGN   EQU      $20      ;A/B ALIGNMENT
00051      0022      A KLOAD    EQU      $22      ;LOAD HYBRID
00052      0024      A KPTRIG  EQU      $24      ;PROBE TRIGGER
00053      0021      A KOFSET   EQU      $21      ;OFFSET FOR HYBRID
00054
00055      *      OTHER EQUATES
00056
00057      001F      A ABFLEN   EQU      $1F      ;DEBUG BUF LENGTH
00058      0003      A IDLMOD  EQU      $03      ;IDLE MODE

```

```

00059      00FF      A EOTCHR EQU      $FF      ;END OF TABLE CHAR
00060      0004      A BOFFST EQU      $4       ;OFFSET BIT IN ABFLAG
C   1
00062
00063          *      ERROR EQUATES
00064          *
00065      0080      A OPAERT EQU      $80      ;OPERATOR/FATAL ABORT
00066      0081      A IEEEEER EQU     $81      ;IEEE BUS ERROR
00067          *
00068          *      LOW MEMORY EQUATES
00069          *
00070          *
00071      0010      A KEY      EQU      16       LAST KEY BUTTON PUSHED
00072      00A4      A ZDIR      EQU      164      CURRENT Z DIRECTION
00073      00E2      A UOFFSET EQU      178      USE OFFSET FLAG
00074      00AF      A INCNT      EQU      175      INFUT MESSAGE COUNT
00075      009F      A ONOFFL EQU      159      ONLINE/OFFLINE
00076      01F3      A DSPADR      EQU      499      DISPLAY ADDR
00077      01F5      A ADREBUF EQU      501      DISPLAY BUFFER
00078      01F1      A HEXRES      EQU      497      ;HEX RESULT OF ADDR
00079      00AE      A BUSERR      EQU      171      BUS ERROR CODE
00080      00A2      A CURMOD      EQU      162      CURRENT OPERATING MODE
00081      00A7      A ABFLAG      EQU      167      AB ALIGNMENT FLAG
00082      00A1      A ONOFSV      EQU      161      ONLINE/OFFLINE MODE SA
00083      007E      A OFFSTF      EQU      126      OFFSET KNOWN FLAG
00084          *
00085          *
C   96          *      HARDWARE EQUATES
01  17          *
00088          *
00089      E070      A LIGHT      EQU      $E070     CPU LED LIGHTS
00090      E130      A FORCE       EQU      $E130     FORCE PIA
00091      E120      A BFFPIA     EQU      $E120     BOND FUNCTION PIA
00092      E710      A PRSWIT      EQU      $E710     ;PROBE SWITCH

```

AGE 003 HY16EXEC.SA:0 EXEC HY16EXEC 2460 EXECUTIVE PROCESSOR

```

0094          0000 10CE 03FF      P EXEC      EQU      *      EXECUTIVE ENTRY
0095P 0000 10CE 03FF      A      LDS      *03FF INITIALIZE STACK ADDR.
0096          *
0097P 0004 B6  40      A      LDA      *40      ON BIT 6 LED
0098P 0006 B7  E070     A      STA      LIGHT
0099          *
0100P 0009 0F  B2      A      CLR      UOFSSET CLEAR USE OF OFFSET FL
0101P 000B B6  20      A      LDA      *20      SET FORCE TO MIN.
0102P 000D 43          COMA
0103P 000E B7  E130     A      STA      FORCE
0104          *
0105P 0011 BD  0000     A      JSR      IKBPIA CLEAR ALL LIGHTS
0106          *
0107P 0014 BE  0190     A      LDX      *400     DELAY 400 M-SECOND
0108P 0017 ED  0000     A      JSR      DELAY2
0109          *
0110          *      TURN OFF DISK
0111          *
0112P 001A ED  0000     A      JSR      DSKOFF
0113          *
0114P 001D B6  E121     A      LDA      BFFPIA+1 USE TARGET CAMERA
0115P 0020 B4  F7      A      ANDA     *F7
0116P 0022 B7  E121     A      STA      BFFPIA+1
0117          *
0118P 0025 BD  0000     A      JSR      ONLTE     ;TURN ON RUN LIGHT
0119P 0028      41      A      FCB      LRUN
0120P 0029 B0  01DB     P      JSR      DSPMOD   ;DISPLAY CURRENT MODE
0121P 002C 0F  10      A      CLR      KEY      ;INIT KEY BUFFER
0122          *
0123          *      THE FOLLOWING THE THE MAIN CONTROL LOOP
0124          *
0125          002E      P EXCNTR EQU      *
0126P 002E 96  10      A      LDA      KEY
0127P 0030 26  23  0055  BNE      EXEC30 LOOP IF ZERO
0128          *
0129P 0032 0D  AE      A      TST      BUSERR   ;DID ANY BUS ERRORS OC
0130P 0034 27  10  0046  BEQ      EXEC10   ;NO
0131P 0036 BD  0000     A      JSR      DSPL     ;YES
0132P 0039      0225     P      FDB      BUSMSG
0133          *
0134P 003B ED  0000     A      JSR      TCLRDP
0135P 003E C6  81      A      LDB      *IEEER   ;REPORT IEEE BUS ERROR
0136P 0040 ED  0000     A      JSR      RERRR   ;FATAL ABORT
0137P 0043 7E  007B     P      JMP      FTLABT
0138          *
0139P 0046 0D  9F      A EXEC10 TST      ONOFFL   ;IN OFFLINE MODE?
0140P 004B 26  E4  002E  BNE      EXCNTR   ;YES
0141P 004A 0D  AF      A      TST      INCNT   ;ANY MESSAGES WAITING
0142P 004C 27  E0  002E  BEQ      EXCNTR   ;NO
0143P 004E BD  0000     A      JSR      CMINTF  ;YES, PROCESS
0144P 0051 0A  AF      A      DEC      INCNT   ;ONE LESS MESSAGE
0145P 0053 20  03  005B  BRA      EXEC50
0146          *
0147P 0055 BD  016E     P EXEC30 JSR      KEYPRC  ;PROCESS KEY
0148P 005B ED  01DB     P EXEC50 JSR      DSPMOD   ;DISPLAY CURRENT MODE
0149P 005E 20  D1  002E  BRA      EXCNTR
0150          *

```

GE 004 HY16EXEC.SA:0 EXEC HY16EXEC 2460 EXECUTIVE PROCESSOR

```

152
153      *
      * STOP KEY PROCESSOR
      *
0155P 005D C6 FF A STPKY LDB #FF
0156P 005F D7 A1 A STB ONOFSV ;SET TO OFFLINE
157      *
0158P 0061 BD 0000 A JSR ONLTE
0159P 0064 43 A FCB LLSTOP ;TURN ON DIAG LITES
0160P 0065 BD 0000 A JSR OFFLTE
0161P 0068 41 A FCB LRUN ;TURN OFF RUN LIGHT
0162      *
0163P 0069 39 RTS
0164      *
0165      * RUN KEY PROCESSOR
      *
0167P 006A 0F A1 A RUNKY CLR ONOFSV
0168P 006C BD 0000 A JSR OFFLTE
0169P 006F 43 A FCB LLSTOP ;TURN OFF STOP LIGHT
0170P 0070 BD 0000 A JSR ONLTE
0171P 0073 41 A FCB LRUN ;TURN ON RUN LIGHT
0172P 0074 39 RTS
0173      *
0174      * ABORT KEY PROCESSOR
      *
0175      *
0176P 0075 96 A2 A ABRTKY LDA CURMOD ;GET CURRENT MODE
0177P 0077 81 03 A CMPA #IDLMOD ;IN IDLE MODE?
0178P 0079 26 08 0083 BNE ABRT10 ;NO,PUT IN IDLE
0179      *
0180P 007E C6 80 A FTLABT LDB #OPABRT ;REPORT OPERATOR/FATAL
0181P 007D BD 0000 A JSR RERROR
0182P 0080 7E 0000 A JMP RSTART ;YES ABORT
0183      *
0184P 0083 BD 0000 A ABRT10 JSR DSPL DISP ABORT MSG
0185P 0086 0265 F FDB ABRTMS
0186      *
0187P 0088 BD 0000 A JSR FLSHYN ;FLASH YES/NO
0188P 008E 26 0F 009C BNE ABRT20 ;NO OR ABORT
0189      *
0190P 008D BD 0000 A JSR CUROFF TURN OF CUURENT MODE
0191P 0090 BD 0000 A JSR HOPROB
0192P 0093 86 03 A LDA #IDLMOD
0193P 0095 97 A2 A STA CURMOD ;SET TO IDLE MODE
0194P 0097 C6 80 A LDB #OPABRT ;SET ABORT ERR CODE
0195P 0099 BD 0000 A JSR RERROR
0196P 009C 39 ABRT20 RTS
0197      *
0198      * A/B ALIGNMENT KEY PROCESSOR
      *
0199      *
0200P 009D 96 A7 A ABLNKY LDA ABFLAG ;IS OFFSET KNOWN?
0201P 009F 85 04 A BITA #BOFFST
0202P 00A1 27 09 00AC BEQ ABLN40 ;NO
0203      *
0204P 00A3 BD 0000 A JSR ABSET
0205P 00A6 96 10 A LDA KEY ;DID USER ABORT
0206P 00AB 81 30 A CMPA #KABRT
0207P 00AA 27 C9 0075 BEQ ABRTKY ;YES
0208P 00AC 39 ABLN40 RTS
0209      *

```



```

)
* UP/DOWN PROBE KEY
*
2P 00AD 0D A4 A UPDNKY TST ZDIR ;CURRENTLY PROBE UP?
3P 00AF 27 03 00E4 BEQ UPDN10 ;YES, GO DOWN
4
*
5P 00B1 7E 0000 A JMP UPPROB ;NO, RAISE PROBE
6P 00B4 7E 0000 A UPDN10 JMP DNPROB ;YES
7
*
8
* UNLOAD KEY PROCESSOR
*
9P 00B7 8D 0000 A UNLDKY JSR DSFL
10P 00BA 0251 P FDB UNLMSG ;DISPLAY UNLOAD MSG
11P 00BC 8D 0000 A JSR ONLTE
12P 00BF 35 A FCB LUNLOD ;TURN ON UNLOAD LIGHT
13P 00C0 8D 0000 A JSR MUFARK ;MOVE TO PARK
14P 00C3 8D 0000 A JSR OFFLTE
15P 00C6 35 A FCB LUNLOD ;TURN OFF UNLOAD LIGHT
16P 00C7 39 RTS
17
*
18
* AUTO KEY PROCESSOR
*
19P 00C8 D6 A2 A AUTOKY LDB CURMOD
20P 00CA C1 03 A CMPB #IDLMOD ;INVALID IN IDLE
21P 00CC 27 08 00D6 BEQ AUTO20 ;SKIP NEXT TWO
22
*
23P 00CE 8D 0000 A JSR AUTPRE ;VALID?
24P 00D1 25 03 00D6 BCS AUTO20 ;NO
25
*
26P 00D3 8D 006A P AUTO10 JSR RUNKY ;TURN TO RUN MODE
27P 00D6 39 AUTO20 RTS
28
*
29
* STEP KEY PROCESSOR
*
30P 00D7 D6 A2 A STEPKY LDB CURMOD
31P 00D9 C1 03 A CMPB #IDLMOD ;VALID MODE
32P 00DE 27 F9 00D6 BEQ AUTO20 ;NO
33
*
34P 00DD 8D 0000 A JSR MANPRE
35P 00E0 24 F1 00D3 BCC AUTO10 ;VALID CHANGE
36P 00E2 39 RTS
37
*
38
* TEST KEY PROCESSOR
*
39P 00E3 8D 0000 A TESTKY JSR TSTPRE ;VALID?
40P 00E6 25 EE 00D6 BCS AUTO20 ;NO
41
*
42P 00E8 8D 0000 A TEST10 JSR OFLNE
43P 00EB 96 10 A LDA KEY
44P 00ED 81 30 A CMPA #KABRT ;WAS ABORT PRESSED?
45P 00EF 26 09 00FA BNE TEST20 ;NO
46P 00F1 8D 0075 P JSR ABRTKY ;DO ABORT PROCESSING
47
*
48P 00F4 96 A2 A LDA CURMOD ;DID WE ABORT TEST MOD
49P 00F6 81 03 A CMPA #IDLMOD
50P 00F8 26 EE 00E8 BNE TEST10 ;NO
51P 00FA 39 TEST20 RTS
52
*
53
* DEBUG INPUT ADDRESS KEY PROCESSOR

```

006 HY16EXEC.SA:0 EXEC HY16EXEC 2460 EXECUTIVE PROCESSOR

```

08
09P 00FE BD 0000 A INADKY JSR ENTER ;ENTER THE ADDRESS
10P 00FE 023A P FDB INPMSG
11P 0100 15 A FCB 21
12P 0101 05 A FCB 5
13
14P 0102 4D TSTA ;USER ABORT?
15P 0103 102B FF6E 0075 LBMI ABRTKY ;YES
16
17P 0107 10BF 01F3 A STY DSPADR
18
19 * ENTRY POINT FOR DISPLAY NEXT ADDRESS
30 * KEY PROCESSOR
31
32P 010B ED 0191 P DSNXKY JSR DISPLY
33P 010E BD 0000 A JSR TCLRDF
34P 0111 39 RTS
35
36 * PROBE TRIGGER KEY
37 * WHEN THE USER PRESSES THIS KEY, A ONE-S
38 * WILL BE TRIGGERED TO CLOSE THE PROBE SW
39 * FOR A FEW MILLISECONDS SO THAT THE DTS-
40 * CAN READ THE DESIRED INFORMATION
41
42P 0112 0D A4 A PRTRKY TST ZDIR ;IS THE PROBE DOWN?
43P 0114 27 09 011F BEQ PRTR20 ;NO
44
45P 0116 96 A2 A LDA CURMOD
46P 0118 81 02 A CMPA #2 ;IN M0 OR M1?
47P 011A 24 03 011F BCC PRTR20 ;NO,ILLEGAL KEY
48
49P 011C 7D E710 A PRTR10 TST PRSWIT ;CLOSE SWITCH
50P 011F 39 PRTR20 RTS
51
52 * OFFSET KEY PROCESSOR
53
54 * THIS PROCESSOR WILL CALCULATE THE NECES
55 * CAMERA OFFSET VALUES SO THAT THE PROBE
56 * APPEAR TO BE OVER A DESIRED POINT.
57
58P 0120 8D 09 012B OFSTKY BSR COFFST ;CALC CAMERA OFFSET
59
60P 0122 96 10 A OFST30 LDA KEY
61P 0124 81 30 A CMPA #KABRT ;DID USER ABORT?
62P 0126 1027 FF4B 0075 LBEQ ABRTKY ;YES
63P 012A 39 RTS ;NO
64
65 * MINI-PROCESSOR FOR CAMERA OFFSET
66
67P 012B BD 0000 A COFFST JSR ONLTE ;TURN ON OFFSET LIGHT
68P 012E 32 A FCB LOFFSET
69
70P 012F BD 0000 A JSR CAMOFF ;OFFSET CALC
71P 0132 96 10 A LDA KEY
72P 0134 81 30 A CMPA #KABRT ;DID USER ABORT?
73P 0136 27 06 013E BEQ COFF1 ;YES
74P 0138 96 A7 A LDA ABFLAG ;SET OFFSET KNOWN BIT
75P 013A 8A 04 A ORA #BOFFST

```

007 HY16EXEC.SA:0 EXEC HY16EXEC 2460 EXECUTIVE PROCESSOR

```

P 013C 97 A7 A STA ABFLAG
P 013E 8D 0000 A COFF1 JSR OFFLTE ;TURN OFF OFFSET LIGHT
P 0141 32 A FCB LOFSET
P 0142 39 RTS

```

```

*
* LOAD HYBRID KEY PROCESSOR
*
* THIS PROCEDURE MUST BE DONE FOR ALL
* HYBRIDS BEFORE ANY MOVES CAN BE
* PERFORMED. THIS KEY PERFORMS TWO FUNCTI
* 1. CALCULATES THE CAMERA OFFSET AND 2.
* SETS AND ALIGNS THE A/B ALIGNMENT POINT
*

```

```

P 0143 8D 0000 A LOADKY JSR ONLTE ;TURN ON LOAD LIGHT
P 0146 34 A FCB LLOAD
P 0147 8D 0000 A JSR MYPARK ;MOVE TO PARK
P 014A 8D 0000 A JSR DSFL
P 014D 0271 F FCB LOADMS

```

```

P 014F 8D 0000 A JSR TCLRDP
P 0152 26 14 0168 BNE LOAD15 ;USER ABORTED

```

```

P 0154 96 7E A LDA OFFSTF ;IS OFFSET KNOWN?
P 0156 81 01 A CMFA #1
P 0158 27 0E 0165 BEQ LOAD10 ;YES, SKIP OFFSET

```

```

P 015A 8D 0000 A JSR HOPROB ;HOME PROBE
P 015D 8D CC 012E BSR COFFST ;CALC OFFSET
P 015F 96 10 A LDA KEY
P 0161 81 30 A CMFA #KABRT ;DID USER ABORT?
P 0163 27 03 0168 BEQ LOAD15 ;YES

```

```

P 0165 8D 0000 A LOAD10 JSR ABSET ;ALIGN AB

```

```

P 0168 8D 0000 A LOAD15 JSR OFFLTE ;TURN OFF LOAD LIGHT
P 016E 34 A FCB LLOAD

```

```

P 016C 20 B4 0122 LOAD50 BRA OFST30

```

*
 * THE FOLLOWING ROUTINE WILL DETERMINE
 * WHICH KEY WAS LAST PRESSED AND VECTOR
 * THE EXECUTIVE PROCESSOR TO THE KEY
 * PROCESSOR FOR THAT PARTICULAR KEY.
 *

>	016E	8E	0289	P	KEYPRC	LDX	◆KEYTEL	;PTR TO KEY JUMP TABLE	
>	0171	108E	0010	A		LDY	◆KEY	;POINT TO KEY	
>	0175	C6	01	A		LDB	◆1	;MATCH COUNT	
>	0177	BD	0000	A		JSR	MATCH	;DID A MATCH OCCUR?	
>	017A	25	12	018E		BCS	KEYRTN	;NO	
*									
>	017C	96	9F	A		LDA	ONOFFL	;SAVE CUR ONLINE/OFFLI	
P	017E	97	A1	A		STA	ONOFFSV		
>	0180	86	FF	A		LDA	◆\$FF	;SET TO OFFLINE/BUSY	
P	0182	97	9F	A		STA	ONOFFL		
>	0184	EC	84	A		LDD	0,X	;GET RELATIVE BRANCH	
P	0186	30	8B	A		LEAX	D,X	;MAKE ABSOLUTE	
P	0188	AD	84	A		JSR	0,X	;GO TO ROUTINE	
P	018A	96	A1	A		LDA	ONOFFSV	;RESTORE ONLINE/OFFLIN	
P	018C	97	9F	A		STA	ONOFFL		
P	018E	0F	10	A	KEYRTN	CLR	KEY	;INIT KEY	
P	0190	39				RTS			

```

*
* NAME: DISPLY
*
* FUNCTION: TO DISPLAY THE NEXT 8 BYTES
*           OF MEMORY TO THE KEYBOARD DISPLA
*           THE START ADDRESS IS POINTED TO
*           DSPADR.
*
* CALLING SEQUENCE: JSR DISPLY
*
* UPON RETURN: DSPADR WILL AUTOMATICALLY
*              UPADTED TO POINT TO NEXT
*              BYTES OF DATA.

```

```

0191 34 40 A DISPLY PSHS U ;SAVE REG
0193 8E 01F5 A LDX #ADRBUF ;FORMAT BUFFER
0196 86 20 A LDA #20
0198 C6 1F A LDB #ABFLEN ;LENGTH OF BUFFER
*
019A A7 80 A DISP10 STA ,X+ ;BLANK BUFFER
019C 5A DECB
019D 26 FB 019A BNE DISP10
*
* OUTPUT THE START ADDRESS IN HEX
*
019F B6 01F3 A LDA DSPADR ;GET MSE OF CUR ADDR
01A2 ED 0000 A JSR CBHEX ;CONVERT TO HEX
01A5 01F5 A FDB ADRBUF
01A7 E6 01F4 A LDA DSPADR+1 ;GET MSE OF CUR ADDR
01AA BD 0000 A JSR CBHEX ;CONVERT TO HEX
01AD 01F7 A FDB ADRBUF+2
*
* DISPLAY 8 BYTES
*
01AF CE 01FE A LDU #ADRBUF+6
01B2 C6 08 A LDB #8
01B4 10BE 01F3 A LDY DSPADR
*
01B8 A6 A0 A DISP30 LDA ,Y+ ;GET BYTE
01BA BD 0000 A JSR CBHEX ;CONVERT TO HEX
01BD 01F1 A FDB HEXRES
*
01BF BE 01F1 A LDX HEXRES ;GET HEX RESULT
01C2 AF C4 A STX 0,U ;STORE IN FORMAT BUFFE
*
01C4 33 43 A LEAU 3,U ;UPDATE FORMAT POINTER
01C6 5A DECB
01C7 26 EF 01B8 BNE DISP30
*
01C9 86 0D A LDA #0D ;END OF MESSAGE
01CB A7 C4 A STA ,U
01CD 10BF 01F3 A STY DSPADR ;UPDATE CUR DISPLAY AD
01D1 ED 0000 A JSR DSPL ;DISPLAY TO KEYBOARD
01D4 01F5 A FDB ADRBUF
*
01D6 35 C0 A PULS U,PC

```

.0 HY16EXEC.SA:0 EXEC HY16EXEC 2460 EXECUTIVE PROCESSOR

*
 * THIS ROUTINE WILL DISPLAY THE
 * CURRENT OPERATING MODE TO THE
 * KEYBOARD DISPLAY.
 *

*
 01D8 96 A2 A DSPMOD LDA CURMOD ;IN AUTO MODE
 01DA 26 07 01E3 BNE DS2A ;NO
 01DC 8D 0000 A JSR DSPL
 01DF 01FD F FDB AUTMSG
 01E1 20 19 01FC BRA DSPRTN

*
 01E3 4A DS2A DECA ;IN STEP MODE?
 01E4 26 07 01ED BNE DS2B ;NO
 01E6 8D 0000 A JSR DSPL ;YES
 01E9 0207 F FDB STPMSG
 01EB 20 0F 01FC BRA DSPRTN

*
 01ED 4A DS2B DECA ;TEST MODE?
 01EE 26 07 01F7 BNE DS2C ;NO
 01F0 8D 0000 A JSR DSPL
 01F3 0211 F FDB TSTMSG
 01F5 20 05 01FC BRA DSPRTN

*
 01F7 8D 0000 A DS2C JSR DSPL ;ASSUME IDLE
 01FA 021E F FDB IDLMSG

*
 01FC 39 DSPRTN RTS
 *

HY11CMDP.SA:0 COMAND COMMAND PROCESSOR

*
 * ROUTINE TO MATCH STRINGS. LEADINGS BLANKS
 * SOURCE STRING (PASSED IN Y-REG) WILL BE SU
 * AND VALIDITY OF THE COMMAND DETERMINED. IF
 * MATCH OCCURS, THE CARRY BIT WILL BE CLEARED
 * THE X-REG WILL CONTAIN THE ADDRESS OF THAT
 * PARTICULAR COMMAND PROCESSOR. IF THERE IS
 * MATCH, THE CARRY WILL RETURN SET.

*
 * CALLING PARAMETERS: LDX #TABLE POINTE
 * LDY SOURCE POINTE
 * LDE MATCH COUNT
 * JSR MATCH

* NOTE: 1) % MAY BE USED AS A DONT CARE CH
 * 2) ALL STRINGS MUST BE OF EQUAL LE

```

009F F MATCH EQU *
09F D7 A0 A STB MTCNT ;SAVE MATCH COUNT
0A1 109F CF A STY SRCPTR ;SAVE SOURCE PTR
*
* MATCH LOOP
*
0A4 A6 B0 A MATCH2 LDA ,X+ ;GET TABLE CHAR
0A6 B1 00 A CMPA #DONTCR ;IS IT DONT CARE?
0A8 27 04 00AE BEQ MATCH3 ;YES, AUTOMATCH
*
0AA A1 A4 A CMPA 0,Y ;MATCH SOURCE CHAR?
0AC 26 08 00E6 BNE MATCH5 ;NO
*
0AE 31 21 A MATCH3 LEAY 1,Y
0E0 5A DECB ;MATCH COMPLETE?
0E1 26 F1 00A4 BNE MATCH2 ;NO
*
* MATCH COMPLETE - GET COMMAND PROCESSOR ADD
*
0E3 1C FE A ANDCC #FE
0E5 39 RTS
*
* THIS ENTRY DOES NOT MATCH. FLUSH THROUGH T
* NEXT ENTRY IF ANY
*
0E6 5A MATCH5 DECB
0E7 3A ABX ;POINT TO END OF STRIN
*
0E8 30 02 A LEAX 2,X ;SKIF JUMP VECTOR
0EA A6 B4 A LDA 0,X
0EC B1 FF A CMPA #EOTCHR ;DISPLAY END CHAR
0EE 27 07 00C7 BEQ MATCH7 ;YES
0C0 109E CF A LDY SRCPTR ;CONTINUE ON
0C3 D6 A0 A LDE MTCNT
0C5 20 DD 00A4 ERA MATCH2
*
0C7 1A 01 A MATCH7 ORCC #1
0C9 39 RTS
  
```

HY11CMDP.SA:0 COMAND COMMAND PROCESSOR

```

*
* GETCHR - IS DESIGNED TO FETCH A SINGLE BYT
* FROM THE CIRCULAR QUEUE BUFFER.
* ROUTINE RETURNS WITH THE CARRY SET
* IF THE BUFFER WAS EMPTY. OTHERWISE THE
* CARRY IS CLEAR AND THE CHARACTER IS RET
* IN THE B-REG.

```

```

* CALLING PARAMETERS: LDX #BCF POINTE
* JSR GETCHR
*

```

```

0086 P GETCHR EQU *
086 34 20 A PSHS Y
088 10AE 02 A LDY 2,X GET DEL POINTER
08E 10AC 84 A CMPY 0,X ;IS INS=DEL?
08E 27 F1 0081 BEQ PUTC30 ;YES, BUFFER EMPTY
*
090 E6 A0 A LDB ,Y+ ;GET NEXT CHAR
092 10AC 04 A CMPY 4,X ;AT EOB?
095 26 03 009A BNE GETC30 ;NO
*
* SET DEL POINTER TO BOB
*
097 10AE 06 A LDY 6,X ;GET BOB POINTER
09A 10AF 02 A GETC30 STY 2,X ;UPDATE BCF
09D 20 DD 007C BRA PUTC20

```



```

*
*   QUEUE INPUT ROUTINE
*   THIS ROUTINE IS DESIGNED TO PUT A CHAR
*   INTO A CIRCULAR QUEUE BUFFER. UPON RETU
*   IF THE CARRY IS CLEAR, THE CHARACTER WA
*   STORED IN THE BUFFER AND THE BUFFER
*   POINTERS WERE UPDATED. OTHERWISE, THE
*   CARRY IS SET INDICATING THAT THE BUFFER
*   FULL.

```

```

*           CALLING PARAMETERS:   LDB CHAR
*                                   LDX #BCF POIN
*                                   JSR PUTCHR

```

```

*   NOTE: THE B-REG IS SAVED.

```

```

*
*   0065   P PUTCHR EQU   *
065 34 20   A   PSHS   Y       ;SAVE REGISTERS
067 10AE 84  A   LDY    0,X     ;GET INS POINTER
06A E7 A0   A   STB    ,Y+    ;STORE CHAR
*
06C 10AC 04  A       CMPY   4,X     ;IS INS=EOB?
06F 26 03   0074  BNE   PUTC10  ;NO
*
*   SET INSERT POINTER = BOB
*
071 10AE 06  A       LDY    6,X     ;GET BOB
074 10AC 02  A PUTC10 CMPY   2,X     ;IS INS=DEL?
077 27 08   0081  BEQ   PUTC30  ;YES, BUFFER IS FULL
*
079 10AF 84  A       STY    0,X     ;UPDATE BCF
07C 35 20   A PUTC20 PULS   Y
07E 1C FE   A       ANDCC  #$FE   ;RETURN CC
080 39
*
081 35 30   A PUTC30 PULS   X,Y
083 1A 01   A       ORCC   #1     ;RETURN CS
085 39

```

07 HY11CMPD.SA:0 COMAND COMMAND PROCESSOR

0059	27	09	0064	BEG	CMIN70	;NO
005B	C6	32	A	LDB	⊕BADCTR	;YES, ERROR
005D	20	02	0061	BRA	CMIN60	

*
* ERROR - UNKNOWN COMMAND
*

005F	C6	30	A	CMIN50	LDB	⊕UNKCMD
0061	BD	015B	P	CMIN60	JSR	ERROR
0064	39			CMIN70	RTS	

5
6
7
8
9
0
1
2
3
4
5
6
7
8
9
0
1
2
3
4
5
6
7
8
9
0
1
2
3
4
5
6

```

*
*XXXXXXXXXXXXXXXXXXXXX
*
*   NAME:      COMMAND PROCESSOR
*
*   FUNCTION:   TO INTERPRET COMMANDS SENT BY
*               THE HOST AND EXECUTE ITS
*               INSTRUCTIONS.
*
*   CALLING SEQUENCE:
*               JMP CMINTP
*
*   NOTE:  WHENEVER THE STATE OF "INCNT" BEC
*           NON-ZERO, THIS MEANS THAT A COMMA
*           IS WAITING TO BE INTERPRETED IN T
*           INPUT QUEUE.
*
*   REV       DATE       BY       REASON FO
*   <<<<     <<<<<     <<     <<<<<<<<<<<<
*   00        10/22/81    SJF      INITIAL R
*
*
*

```

GE 004 HY11CMDP.SA:0 COMAND COMMAND PROCESSOR

139	00A2	A CURMOD EQU	162	;CURRENT OPERATION MOD
140		*		
141	00AE	A OUTCNT EQU	174	;OUTPUT MESSAGE COUNT
142	04D4	A OUTBUF EQU	1236	;OUTPUT BUFFER
143	0539	A OUTEND EQU	1337	
144		*		
145		* BUFFER CONTROL FIELDS (BCF)		
146		*		
147		* FORMAT: NEXT AVAIL RMB 2		
148		* LAST: RMB 2		
149		* END OF BUFFER: RMB 2		
150		* BEG OF BUFFER: RMB 2		
151		*		
152	045E	A INBCF EQU	1118	;INPUT BCF
153	0466	A OUTBCF EQU	1126	;OUTPUT BCF
154	00AF	A INCNT EQU	175	;INPUT MSG COUNT
155	01AA	A EXBUF EQU	426	;NEXT COMMAND BUFFER
156	00A0	A MTCNT EQU	160	;GET MATCH COUNT
157	00CF	A SRCPTR EQU	207	
158	00A6	A ESTATE EQU	166	;SET WHEN ERROR OCCURS
159	00A7	A AEFLAG EQU	167	;=\$43 WHEN GIVEN
160	00A8	A DEMODE EQU	168	;SET WHEN D,E COORD
161	009E	A VALSGN EQU	158	;SET WHEN SIGN OF COOR
162	00EF	A NUMBUF EQU	191	;STORAGE BUF FOR NUMER
163	00C5	A TEMPX EQU	197	;TEMP STORAGE FOR X BI
164	00C7	A TEMPY EQU	199	;SAME FOR Y
165	00C4	A XDSIGN EQU	196	;SIGN OF X OR D COORD
166	00C9	A YESIGN EQU	201	;SIGN OF Y OR E COORD
167	00A9	A OVRFLW EQU	169	;SET WHEN COMMAND OVER
168	009F	A ONOFFL EQU	159	;ONLINE/OFFLINE FLAG
169	01D3	A CXYPOS EQU	467	;X,Y FORMATTED BUFFER
170	01E1	A CDEPOS EQU	481	;D,E FORMATTED BUFFER
171	0094	A CXSAVE EQU	148	;SAVE CURRENT X POS
172	0096	A CYSAVE EQU	150	;SAVE CURRENT Y POS
173	0077	A ZSAVE EQU	119	;SAVE Z DIRECTION
174		*		
175	008C	A SREFAX EQU	140	;SAVE RAX POS
176	008E	A SREFAY EQU	142	;SAVE RAY POS
177	0090	A SREFEX EQU	144	;SAVE REX POS
178	0092	A SREFEY EQU	146	;SAVE RBY POS
179		*		
180		* HARDWARE EQUATES		
181		*		
182	E710	A PRSWIT EQU	*E710	;CLOSE TOUCHDOWN SWITC
183				

003 HY11CMDF.SA:0 COMAND COMMAND PROCESSOR

```

181          *
182          * OPERATIONS MODE EQUATES - "CURMOD" CAN TAKE
183          * ANY ONE OF THE FOLLOWING STA
184          *
185          0003  A IDLMOD EQU    $03      ;IDLE MODE
186          0002  A TSTMOD EQU    $02      ;TEST MODE
187          0001  A STPMOD EQU    $01      ;STEP/MANUAL MODE
188          0000  A AUTMOD EQU    $00      ;AUTOMATIC MODE
189
190          *
191          * THE FOLLOWING EQUATE TABLE IS THE BIT
192          * CONFIGUARATION OF EACH ERROR FLAG ABOVE.
193          *
194          0020  A ZBIT   EQU    $20      ;ZDIR = BIT 5
195          0010  A BEIT   EQU    $10      ;DEVBSY = BIT 4
196          0008  A OBIT   EQU    $08      ;ONOFFL = BIT 3
197          0004  A EBIT   EQU    $04      ;PROBER = BIT 2
198          0003  A MEIT   EQU    $03      ;CURMOD = BITS 1 AND 0
199
200          *
201          * MISC. EQUATES
202          *
203          000A  A STRTRM EQU    $0A      ;STRING TERMINATOR (LF
204          0024  A BUFLN  EQU    36      ;MAXLENGTH OF COMMAND
205          0000  A DONTCR EQU    $00      ;CONT CARE CHAR
206          00FF  A EOTCHR EQU    $FF      ;END OF TABLE
207          0004  A BOFFST EQU    $4      ;CAMERA OFFSET KNOWN B
208          4E20  A PARKX  EQU    20000   ;PARK POSITION X
209          4E20  A PARKY  EQU    20000   ;PARK POSITION Y
210
211          *
212          * LOW MEMORY EQUATES
213          *
214          0036  A CX      EQU    $36      ;CURRENT ABS LOC X
215          0039  A CY      EQU    $39      ;CURRENT ABS LOC Y
216          003C  A NX      EQU    $3C      ;NEXT ABS LOC X
217          003F  A NY      EQU    $3F      ;NEXT ABS LOC Y
218          0041  A DD      EQU    $41      ;D EQUATION VALUE
219          0045  A EE      EQU    $45      ;E EQUATION VALUE
220          005E  A NZ      EQU    $5E      ;NEXT ABS Z LOC
221          *
222          0100  A RAX     EQU    $100     ;X ABS LOC REF A
223          0102  A RAY     EQU    $102     ;Y ABS LOC REF A
224          0104  A RBX     EQU    $104     ;X ABS LOC REF B
225          0106  A REY     EQU    $106     ;Y ABS LOC REF B
226          *
227          0060  A MOVEZ   EQU    $60      ;Z MOVE COMPLETE FLAG
228          00E2  A UOFFST  EQU    178     ;CAMERA OFFSET KNOWN F
229          0078  A XOFFS   EQU    120     ;X CAMERA OFFSET
230          007A  A YOFFS   EQU    122     ;Y CAMERA OFFSET
231
232          *
233          * THE FOLLOWING FLAGS MAKE UP THE STATUS BYT
234          * THAT IS SENT TO THE CONTROLLER UPON REQUES
235          *
236          *
237          00A4  A ZDIR    EQU    164     ;CURRENT Z DIRECTION
238          00A3  A PROBER  EQU    163     ;PROBER STATUS CODE

```

```
00059 * 93 X,Y
00060 * 94 Z
  51 * 95 X,Z
00062 * 96 Y,Z
00063 * 97 X,Y,Z
00064 *
00065 *
00066 00A0 A AXISIN EQU $A0 ;BASE ADDR FOR AXIS IN
00067 *
00068 * TYPES OF AXIS INIT ERRORS ARE AS FOLLOWS:
00069 *
00070 * VALUE(HEX) AXISES IN ERROR
00071 * A1 X
00072 * A2 Y
00073 * A3 X,Y
00074 * A4 Z
00075 * A5 X,Z
00076 * A6 Y,Z
00077 * A7 X,Y,Z
00078 *
00079 *
```

```

00001          NAM      COMAND
00002          OPT      REL,CRE,NOG

00003
00004          TTL      COMMAND PROCESSOR
00005          *
00006          * INTERNAL DEFINITIONS
00007          *
00008          XDEF      CMINTP,PUTCHR,GETCHR
00009          XDEF      INTBCF,FORMST,OTCRLF
00010          XDEF      AUTPRB,MANPRB,TSTPRB
00011          XDEF      UPPRB,DNPRB,HOPRBE
00012          XDEF      ABSET,MATCH,RRORR,MVFARK
00013          *
00014          * EXTERNAL REFERENCES
00015          *
00016          XREF      CUROFF,ONLTE
00017          XREF      TOUCHD,CASCE,OFFLTE
00018          XREF      CEASC,SFLOAT,CXYFF,MVFOS
00019          XREF      JOG,ZTABLE,CDEV,FPSUE
00020          XREF      DSFL,TCLRDP,CREFD,ENTER
00021          XREF      FLSHYN,UNLDKY,LOADKY
00022          *
00023          * KEYBOARD LIGHT EQUATES
00024          *
00025          0045      A LAUTO EQU      $45      ;AUTO
00026          0042      A LTEST EQU      $42      ;TEST
00027          0040      A LSTEP EQU      $40      ;STEP
00028          0033      A LUPDWN EQU     $33      ;UP/DOWN
00029          0030      A LABLGN EQU     $30      ;A/B ALIGN
00030          *
00031          * PROBER NON-FATAL ERROR COMMANDS
00032          *
00033          0030      A UNKCMD EQU     $30      ;UNRECOGNIZED COMMAND
00034          0031      A INVCHR EQU     $31      ;INVALID CHAR IN COMMA
00035          0032      A BADCTR EQU     $32      ;IMPROPER TERMINATOR
00036          0033      A INVARG EQU     $33      ;INVALID COMMAND ARGUM
00037          0034      A WRGMOD EQU     $34      ;INVALID COMMAND FOR C
00038          0035      A UNDFAE EQU     $35      ;HYBRID A/B POINTS NOT
00039          0036      A NOFSET EQU     $36      ;CAMERA OFFSET NOT SET
00040          *
00041          * PROBER FATAL ERROR COMMANDS
00042          *
00043          0080      A OPAERT EQU     $80      ;OPERATOR ABORT
00044          0081      A IEEEER EQU     $81      ;IEEE BUS ERROR
00045          0082      A UNKINT EQU     $82      ;UNKNOWN INTERRUPT
00046          0083      A LIMERR EQU     $83      ;AXIS LIMIT ERROR
00047          0084      A TRSHER EQU     $84      ;THRESHOLD ERROR
00048          0085      A AFUERR EQU     $85      ;AFU ERROR
00049          0086      A AXISER EQU     $86      ;AXIS ERROR CONTROL
00050          0087      A DSKERR EQU     $87      ;DISK ERROR
00051          *
00052          0090      A AXISTO EQU     $90      ;TIME OUT BASE ADDR
00053          *
00054          * TYPES OF TIME OUT ERRORS ARE AS FOLLOWS:
00055          *
00056          *          VALUE (HEX)      AXISES IN ERROR
00057          *          91                X
00058          *          92                Y

```


0029 KTEST 00047*00515
0026 KUNLOD 00044*00509
0025 KUPDOWN 00043*00507
E070 LIGHT 00089*00098
0034 LLOAD 00033*00342 00363
0043 LLSTOP 00031*00159 00169
P 0165 LOAD10 00352 00360*
P 0168 LOAD15 00348 00358 00362*
P 016C LOAD50 00365*
DF 0143 LOADKY 00012 00341*00526
P 0271 LOADMS 00345 00494*
0032 LOFSET 00034*00318 00329
0041 LRUN 00030*00119 00161 00171
0035 LUNLOD 00032*00223 00226
R MANPRE 00022*00247
R MATCH 00024*00377
R MUPARK 00025*00224 00343
R OFFLTE 00021*00160 00168 00225 00328 00362
007E OFFSTF 00083*00350
R OFLNE 00025*00256
P 0122 OFST30 00310*00365
P 0120 OFSTKY 00308*00524
R ONLTE 00021*00118 00158 00170 00222 00317 00341
009F ONOFFL 00075*00139 00380 00383 00388
00A1 ONOFSV 00082*00156 00167 00381 00387
R ONSTOP 00019*
0080 OFABRT 00065*00180 00194
E710 FRSWIT 00092*00299
J11C PRTR10 00299*
P 011F PRTR20 00293 00297 00300*
P 0112 PRTRKY 00292*00528
R RERROR 00024*00136 00181 00195
R RSTART 00024*00182
P 006A RUNKY 00167*00238 00504
R START 00020*
P 00D7 STEPKY 00243*00514
P 005D STOPKY 00155*00502
P 0207 STPMSG 00463 00480*
R TCLRDP 00020*00134 00283 00347
P 00E8 TEST10 00256*00264
P 00FA TEST20 00259 00265*
P 00E3 TESTKY 00253*00516
P 0211 TSTMSG 00469 00482*
R TSTPRE 00022*00253
DF 00E7 UNLDKY 00012 00220*00510
P 0251 UNLMSG 00221 00490*
00E2 UOFSET 00073*00100
P 00E4 UPDN10 00213 00216*
P 00AD UPDNKY 00212*00508
R UPROB 00021*00215
00A4 ZDIR 00072*00212 00292

P 0225 BUSMSG 00132 00486*
 R CAMOFF 00025*00320
 K CBHEX 00023*00418 00421 00431
 R CMINTP 00020*00143
 P 013E COFF1 00323 00328*
 DP 012E COFFST 00013 00308 00317*00355
 00A2 CURMOD 00080*00176 00193 00231 00243 00262 00295 00454
 R CUROFF 00023*00190
 R DELAY2 00019*00108
 P 019A DISP10 00411*00413
 P 0188 DISP30 00430*00439
 P 0191 DISPLY 00282 00406*
 R DNPROB 00021*00216
 P 01E3 DS2A 00455 00460*
 P 01ED DS2B 00461 00466*
 P 01F7 DS2C 00467 00472*
 R DSKOFF 00019*00112
 F 010E DSNXKY 00282*00520
 01F3 DSPADR 00076*00277 00417 00420 00428 00443
 R DSPL 00020*00131 00184 00220 00344 00444 00456 00462 00468
 00472
 P 01D8 DSFMOD 00120 00148 00454*
 P 01FC DSPRTN 00458 00464 00470 00475*
 R ENTER 00023*00269
 00FF EOTCHR 00059*00529
 DP 002E EXCNTR 00012 00125*00140 00142 00149
 DP 0000 EXEC 00012 00094*
 0046 EXEC10 00130 00139*
 J055 EXEC30 00127 00147*
 P 0058 EXEC50 00145 00148*
 R FLSHYN 00023*00187
 E130 FORCE 00090*00103
 P 007B FTLAET 00137 00180*
 01F1 HEXRES 00078*00432 00434
 R HOPROB 00022*00191 00354
 0003 IDLMOD 00058*00177 00192 00232 00244 00263
 P 021E IDLMSG 00473 00484*
 0081 IEEEEER 00066*00135
 R IKBFIA 00019*00105
 P 00FB INADKY 00269*00518
 00AF INCNT 00074*00141 00144
 P 023A INFMSG 00270 00488*
 0020 KABLGN 00050*00521
 0030 KAERT 00042*00206 00258 00311 00322 00357 00505
 002A KAUTO 00045*00511
 0041 KDSPNX 00049*00519
 0010 KEY 00071*00121 00126 00205 00257 00310 00321 00356 00375
 00389
 P 016E KEYPRC 00147 00374*
 P 018E KEYRTN 00378 00389*
 P 0289 KEYTBL 00374 00501*
 0031 KINPAD 00048*00517
 0022 KLOAD 00051*00525
 002D KLSTOP 00040*00501
 J021 KOFSET 00053*00523
 0024 KPTRIG 00052*00527
 002C KRUN 00041*00503
 002B KSTEP 00046*00513

```

00497
00498 *
( 09 * THE FOLLOWING IS A RELATIVE JUMP TABLE
00500 * FOR ALL KEY PROCESSOR ROUTINES
*
00501P 0289 2D A KEYTBL FCB KLSTOP ;STOP KEY
00502P 028A FDD3 A FDB STOPKY-*
00503P 028C 2C A FCB KRUN ;RUN KEY
00504P 028D FDDD A FDB RUNKY-*
00505P 028F 30 A FCB KABRT ;ABORT KEY
00506P 0290 FDES A FDB ABRTKY-*
00507P 0292 25 A FCB KUPDWN ;UP/DOWN KEY
00508P 0293 FE1A A FDB UPDNKY-*
00509P 0295 26 A FCB KUNLOD ;UNLOAD KEY
00510P 0296 FE21 A FDB UNLDKY-*
00511P 0298 2A A FCB KAUTO ;AUTOKEY
00512P 0299 FE2F A FDB AUTOKY-*
00513P 029B 28 A FCB KSTEP ;STEP KEY
00514P 029C FE3B A FDB STEPKY-*
00515P 029E 29 A FCB KTEST ;TEST KEY
00516P 029F FE44 A FDB TESTKY-*
00517P 02A1 31 A FCB KINPAD ;INPUT ADDR KEY
00518P 02A2 FE59 A FDB INADKY-*
00519P 02A4 41 A FCB KDSFNX ;DISPLAY NEXT ADDR KEY
00520P 02A5 FE66 A FDB DSNXKY-*
00521P 02A7 20 A FCB KABLGN ;A/B ALIGN KEY
00522P 02A8 FDF5 A FDB ABLNKY-*
00523P 02AA 21 A FCB KOFFSET ;OFFSET KEY
00524P 02AE FE75 A FDB OFSTKY-*
00525P 02AD 22 A FCB KLOAD ;LOAD KEY
00526P 02AE FE95 A FDB LOADKY-*
00527P 02E0 24 A FCB KPTRIG ;PROBE TRIGGER KEY
00528P 02E1 FE61 A FDB PRTRKY-*
00529P 02E3 FF A FCB EOTCHR ;END OF TABLE
00530 *
00531 END

```

TOTAL ERRORS 00000--00000
TOTAL WARNINGS 00000--00000

```

00A7 ABFLAG 00081*00200 00324 00326
001F ABFLEN 00057*00409
P 00AC ABLN40 00202 00208*
P 009D ABLNKY 00200*00522
P 0083 ABRT10 00178 00184*
P 009C ABRT20 00188 00196*
P 0075 ABRTKY 00176*00207 00260 00275 00312 00506
P 0265 ABRTMS 00185 00492*
R ABSET 00024*00204 00360
01F5 ADRBUF 00077*00407 00419 00422 00426 00445
P 01FD AUTMSG 00457 00478*
P 00D3 AUTO10 00238*00248
00D6 AUTO20 00233 00236 00239*00245 00254
P 00C8 AUTOKY 00231*00512
R AUTPRE 00022*00235
E120 EFFPIA 00091*00114 00116
0004 BOFFST 00060*00201 00325
00AE EUSERR 00079*00129

```

00478P 01FD	41	A	AUTMSG FCC	/AUTO MODE/
00479P 0206	0D	A	FCB	\$0D
00480P 0207	53	A	STPMSG FCC	/STEP MODE/
00481P 0210	0D	A	FCB	\$0D
00482P 0211	54	A	TSTMSG FCC	/TEST MODE/
00483P 021A	0D	A	FCB	\$0D
00484P 021B	49	A	IDLMSG FCC	/IDLE MODE/
00485P 0224	0D	A	FCB	\$0D
00486P 0225	49	A	BUSMSG FCC	/IEEE-68488 BUS ERROR/
00487P 0239	0D	A	FCB	\$0D
00488P 023A	49	A	INPMSG FCC	/INPUT ADDRESS: /
00489P 0250	0D	A	FCB	\$0D
00490P 0251	55	A	UNLMSG FCC	/UNLOADING PROBE .../
00491P 0264	0D	A	FCB	\$0D
00492P 0265	41	A	AERTMS FCC	/ABORT TEST?/
00493P 0270	0D	A	FCB	\$0D
00494P 0271	53	A	LOADMS FCC	/SECURE HYBRID IN SOCKET/
00495P 0288	0D	A	FCB	\$0D

```

00402 *
00403 * VALIDATE STRING
C J4 *
00405 * VALIDATE THE STRING TO INSURE THAT IT
00406 * COMPLETELY NUMERIC. IF IN THE "DEMODE",
00407 * FIRST CHARACTER OF THE STRING WILL HAVE T
00408 * OPTION OF BEING A "+" OR "-" SIGN.
00409 *
00410 * CALLING SEQUENCE: LDY STRING PTR
00411 * LDB # OF CHAR (
00412 * JSR VALNUM
00413 *
00414 * UPON RETURN: CC - VALID NUMERIC S
00415 * B - SIGN (0=POS,FF=
00416 * Y - PTR TO NEXT CHA
00417 * PASSED STRING.
00418 *
00419 * OR
00420 * CS - INVALID NUMERIC
00421 * NOTE: THE PURE NUMERIC STRING IS STORED I
00422 * NUMBUF (5 BYTES).
00423 *
00424P 00CA 34 02 A P VALNUM EQU *
00425P 00CC 0F 9E A A PSHS A ;SAVE A REG
00426 * CLR VALSGN ;ASSUME POS
00427P 00CE A6 A0 A VAL010 LDA ,Y+ ;GET POSSIBLE SIGN CHA
00428P 00D0 81 20 A CMFA #20 ;LEADING BLANK?
00429P 00D2 27 FA 00CE BEQ VAL010
00430 *
00431P 00D4 0D AB A TST DEMODE ;IN DE MODE?
00432P 00D6 27 0C 00E4 BEQ VAL030 ;NO NEED TO CHECK FOR
00433 *
00434P 00D8 81 2E A CMFA #2E ;IS IT POS?
00435P 00DA 27 0A 00E6 BEQ VAL050 ;YES
00436 *
00437P 00DC 81 2D A CMFA #2D ;IS IT NEG?
00438P 00DE 26 04 00E4 BNE VAL030 ;NO, ASSUME POS
00439 *
00440P 00E0 03 9E A COM VALSGN ;REPORT MINUS
00441P 00E2 20 02 00E6 BRA VAL050
00442P 00E4 31 3F A VAL030 LEAY -1,Y ;BACKUP ONE CHAR
00443P 00E6 8E 00EF A VAL050 LDX #NUMBUF ;STORAGE BUFFER
00444P 00E9 A6 A0 A VAL060 LDA ,Y+ ;GET CHAR
00445P 00EB 81 30 A CMFA #30 ;VALID DIGIT?
00446P 00ED 25 10 00FF ECS VALERR ;NO
00447P 00EF 81 3A A CMFA #3A
00448P 00F1 24 0C 00FF ECC VALERR ;NO
00449 *
00450P 00F3 A7 80 A STA ,X+ ;STORE DIGIT
00451P 00F5 5A DECB
00452P 00F6 26 F1 00E9 BNE VAL060 ;CONTINUE
00453 *
00454P 00F8 35 02 A PULS A
00455P 00FA D6 9E A LDB VALSGN ;PASS SIGN
00456P 00FC 1C FE A ANDCC #FE
00457P 00FE 39 RTS
00458 *
00459P 00FF 35 02 A VALERR PULS A

```

PAGE 012 HY11CMDP.SA:0 COMAND COMMAND PROCESSOR

00460F 0101 1A 01 A ORCC #1
00461F 0103 39 RTS
2

```

00464
00465      *
00465      * INITIALIZE BUFFER CONTROL FIELD
00466      *
00467      * CALLING PARAMETERS:  LDX  #BCFPTR
00468      *                               LDD  BEG OF BUFF
00469      *                               LDY  END OF BUFF
00470      *                               JSR  INTBCF
00471      *
00472      *
00472      * 0104 P INTBCF EQU *
00473P 0104 ED 84 A STD 0,X ;SET NEXT AVAIL
00474P 0106 ED 02 A STD 2,X ;SET LAST FULL
00475P 0108 ED 06 A STD 6,X ;SET B.O.B.
00476P 010A 10AF 04 A STY 4,X ;SET E.O.B.
00477F 010D 39 RTS
00478
00479      *
00480      * OUTPUT CR/LF TO BCF
00481      *
00482      * CALLING PARAMETERS  LDX #BCFP
00483      *                               JSR OTCRL
00484      *
00485      *
00485      * 010E P OTCRLF EQU *
00486P 010E 34 04 A PSHS B
00487P 0110 C6 0D A LDE #0D
00488P 0112 BD 0065 P JSR PUTCHR ;SAVE CR
00489P 0115 C6 0A A LDE #0A
00490P 0117 BD 0065 P JSR PUTCHR ;SAVE LF
00491P 011A 35 04 A PULS B
00492P 011C 39 RTS

```

```

00494 *
00495 * NAME: LOCATE
00496 *
00497 * FUNCTION: TO LOCATE A REFERENCE POINT. TH
00498 * ROUTINE WILL JOG TO THE REF POI
00499 * LOCATION AND VERIFY THE POSITIO
00500 * CORRECT.
00501 *
00502 * CALLING SEQUENCE: LDX X POSITION
00503 * LDY Y POSITION
00504 * JSR LOCATE
00505 *
00506 * UPON RETURN IF CC - X REG = CURRENT X
00507 * Y REG = CURRENT Y
00508 * ELSE CS - USER ABORTED
00509 *
00510 * NOTE: IF CC - CX AND CY ARE UPDATED
00511 *
00512P 011D 9F 3C A LOCATE STX NX
00513P 011F 109F 3F A STY NY
00514P 0122 8D 01C2 F JSR UPALWY
00515P 0125 8D 0000 A JSR MVPOS
00516 *
00517P 0128 8D 0000 A LOC010 JSR DSPL ;DISPLAY ON TARGET?
00518P 0128 0513 F FDB ONTRGM ;MESSAGE
00519 *
00520P 012D 8D 0000 A JSR FLSHYN
00521P 0130 27 10 0142 BEQ LOCRTN ;YES PRESSED
00522P 0132 28 0E 013F BMI LOCRT1 ;ABORT PRESSED
00523 *
00524P 0134 8D 0000 A JSR DSPL ;DISPLAY "MOVE TO
00525P 0137 051E F FDB MOVMSG ;TEST POINT"
00526 *
00527P 0139 4F CLRA
00528P 013A 8D 0000 A JSR JOG
00529P 013D 27 E9 0128 BEQ LOC010 ;ENTER WAS PRESSED
00530 *
00531P 013F 1A 01 LOCRT1 SEC
00532P 0141 39 RTS
00533P 0142 9E 36 A LOCRTN LDX CX ;PASS NEW POS
00534P 0144 109E 39 A LDY CY
00535P 0147 39 RTS
00536 *

```


AGE 015 HY11CMDP.SA:0 COMAND COMMAND PROCESSOR

```
0538      *
0539      *   THE FOLLOWING ROUTINE WILL MOVE THE PROBE
      10      *   TO THE PARK POSITION SO THAT THE NEXT
0541      *   HYBRID MAY BE LOADED.
0542      *
0543P 0148 8E 4E20 A MVPARK LDX  #PARKX ;GET PARKPOS
0544P 014B 108E 4E20 A      LDY  #PARKY
0545P 014F 9F 3C A      STX   NX
0546P 0151 109F 3F A      STY   NY
0547P 0154 B0 01C2 P      JSR   UPALWY
0548P 0157 ED 0000 A      JSR   MVPOS ;GO TO PARK
0549P 015A 39      A      RTS
0550      *
```

()

AGE 016 HY11CMDP.SA:0 COMAND COMMAND PROCESSOR

```
0552 *
0553 * THE FOLLOWING CODE IS THE COMMON ERROR
0554 * HANDLER. IT WILL SAVE THE ERROR CODE AND
0555 * SET THE ERROR STATE FLAG TO ACTIVE. THE
0556 * ERROR IS NOT REPORTED IN THE OFFLINE MODE.
0557 *
0558 * CALLING SEQUENCE: LDB ERROR CODE
0559 * JSR ERROR
0560 *
0561 015E P ERROR EQU *
0562P 015E 0D 9F A TST ONOFFL ;IN OFFLINE MODE?
0563P 015D 26 06 0165 BNE ERRTN ;YES
0564 *
0565P 015F D7 A3 A RERROR STB PROBER
0566P 0161 C6 FF A LDB #FF
0567P 0163 D7 A6 A STB ESTATE
0568 *
0569P 0165 1A 01 A ERRTN ORCC #1
0570P 0167 39 RTS
```

017 HY11CMDP.SA:0 COMAND COMMAND PROCESSOR

```

72 *
73 * THE FOLLOWING ARE THE COMMAND PROCESSORS
74 *
75 *
76 *
77 * GET MESSAGE PROCESSOR
78 *
79P 0168 BD 0000 A GETMSG JSR DSPL ;DISPLAY TO KEYBOARD
80P 0168 01AC A FDE EXBUFR+2
81 *
82P 016D BD 0000 A JSR TCLRDF ;WAIT FOR USER ACK
83P 0170 39 RTS
84 *
85 * PROBE MODESET TO AUTOMATIC
86 *
87 * AT ANY TIME, THE USER MAY CHANGE MODE OF
88 * OPERATION TO AUTOMATIC. THE ONLY TIME THA
89 * THIS IS PROHIBITED IS IF THE CURRENT MODE
90 * OF OPERATION IS THE TEST MODE. ENTRY INTO
91 * THIS ROUTINE OCCURS BY EITHERPRESSING THE
92 * AUTO KEY OR BY DECODING A 'M0' COMMAND BY
93 * THE CONTROLLER.
94 *
95 * CALLING SEQUENCE: JSR AUTFRB
96 *
97 * RETURN: CC - IF VALID CHANGE
98 *
99 *
00P 0171 96 A2 A AUTFRB LDA CURMOD ;GET CURRENT MODE
001P 0173 81 00 A CMPA #AUTMOD ;ALREADY AUTO?
002P 0175 27 0F 0186 BEQ AUTFRT ;YES
003 *
004P 0177 81 02 A CMPA #TSTMOD ;VALID CHANGE?
005P 0179 27 3C 01B7 BEQ PRBERR ;NO
006 *
007P 017E BD 0000 A JSR CUROFF ;TURN OFF CURRENT MODE
008P 017E 86 00 A LDA #AUTMOD
009P 0180 97 A2 A STA CURMOD ;UPDATE NEW MODE
010P 0182 8D 0000 A JSR ONLTE
011P 0185 45 A FCB LAUTO
012P 0186 1C FE A AUTFRT ANDCC #FE
013P 0188 39 RTS
014 *
015 *
016 * PROBE MODE SET TO MANUAL
017 *
018 * THIS ROUTINE WILL FOLLOW THE SAME RULES A
019 * IN THE AUTOMATIC TESTMODE. THIS MODE MAY
020 * ENTERED BY ONE O9F TWO METHODS: 1)BY PRES
021 * THE 'STEP' KEY OR BY AN 'M1' COMMAND
022 * GENERATED BY THE CONTROLLER
023 *
024 * CALLING SEQUENCE: JSR MANPRB
025 *
026 * RETURN CC- IF VALID MODE CHANGE
027 *
028P 0189 96 A2 A MANPRB LDA CURMOD
029P 018B 81 01 A CMPA #STPMOD ;SAME MODE

```

018 HY11CMDP.SA:0 COMAND COMMAND PROCESSOR

```

P 018D 27 0F 019E      BEQ  MANPRT
P 018F 81 02          A    CMPA  #TSTMOD ;IN TEST MODE?
P 0191 27 24 01B7      BEQ  PRBERR ;YES
P 0193 8D 0000        A    JSR  CUROFF ;TURN OFF CURRENT MODE
P 0196 86 01          A    LDA  #STPMOD
P 0198 97 A2          A    STA  CURMOD ;SET TO NEW MODE
P 019A 8D 0000        A    JSR  ONLTE  ;TURN ON NEW LIGHT
P 019D          40          A    FCB  LSTEP
P 019E 20 E6 0186     MANPRT BRA  AUTPRT

```

```

*
* PROBE MODE SET TO TEST
*
* ONLY IF THE CURRENT MODE OF THE BONDER IS
* IDLE CAN THE TEST MODE BE ENTERED. THIS
* ROUTINE IS TRIGGERED WHENEVER THE "TEST"
* ON THE KEYBOARD IS PRESSED OR IF THE
* CONTROLLER TRANSMITS A "M2" CODE.

```

CALLING SEQUENCE: JSR TSTPRB

RETURN: CC - IF VALID MODE CHANGE

```

P 01A0 96 A2          A    TSTPRB LDA  CURMOD ;GET CURRENT MODE
P 01A2 81 02          A    CMPA  #TSTMOD ;ALREADY IN TEST?
P 01A4 27 0F 01B5      BEQ  TSTRTN ;YES
P 01A6 81 03          A    CMPA  #IDLMOD ;IN IDLE?
P 01A8 26 0D 01B7      BNE  PRBERR ;NO
P 01AA 8D 0000        A    JSR  CUROFF ;TURN OFF CURRENT LIGH
P 01AD 86 02          A    LDA  #TSTMOD
P 01AF 97 A2          A    STA  CURMOD
P 01B1 8D 0000        A    JSR  ONLTE  ;TURN ON TEST LITE
P 01B4          42          A    FCB  LTEST
P 01B5 20 CF 0186     TSTRTN BRA  AUTPRT

```

ERROR - WRONG MODE FOR REQUEST

```

P 01B7 C6 34          A    PRBERR LDB  #WRGMOD
P 01B9 7E 015B       F    JMP  ERROR

```

```

*
* COMMAND TO RAISE THE PROBE
*
* THE COMMAND TO RAISE THE PROBE WILL BE
* HONORED IN ANY MODE. BEFORE ANY CHANGE OF
* AN (X,Y) POSITION, AN AUTOMATIC PROBE UP
* COMMAND IS PERFORMED. CURRENTLY, THE PROBE
* WILL BE RAISED TO ITS MAXIMUM Z POSI
* TION BUT FUTURE ENHANCEMENTS MAY ONLY RAIS
* THE PROBE THE MINIMUM WORK HEIGHT CLEARANC
* LEVEL.

```

CALLING PARAMTERS: JSR UPFROE

019 HY11CMDP.SA:0 COMAND COMMAND PROCESSOR

```

      *
01BC 96 01BC P UPPROB EQU *
01BE 81 A2 A LDA CURMOD ;GET CURRENT MODE
01C0 27 03 A CMPA #IDLMOD ;IN IDLE MODE?
      1E 01E0 BEQ DNRTN ;YES
      *
01C2 C6 04 A UPALWY LDB #4 ;RAISE PROBE
01C4 ED 0000 A JSR ZTABLE
      *
01C7 BD 0000 A JSR OFFLTE ;TURN OFF UP/DOWN LIGH
01CA 33 A FCB LUPDWN
01CB 0F A4 A CLR ZDIR ;SET Z DIR TO UP
01CD 39 RTS

```

020 HY11CMDP.SA:0 COMAND COMMAND PROCESSOR

```
*  
* COMMAND TO LOWER THE PROBE  
*  
* EXECUTING THIS ROUTINE WILL LOWER THE  
* PROBE UNTIL ULTRA-SONIC PROBE TOUCHDOWN  
* HAS BEEN MADE.  
*  
* CALLING PARAMETERS: JSR DNPROB
```

```
*  
01CE P DNPROB EQU *  
> 01CE 96 A2 A LDA CURMOD ;GET CURRENT MODE  
> 01D0 81 03 A CMPA #IDLMOD ;IN IDLE?  
> 01D2 27 0C 01E0 BEQ DNRTN ;YES  
*  
> 01D4 ED 0000 A JSR TOUCHD  
> 01D7 86 FF A LDA #FF  
> 01D9 97 A4 A STA ZDIR  
> 01DE ED 0000 A JSR ONLTE ;TURN ON UP/DOWN LIGHT  
> 01DE 33 A FCE LUPDWN  
> 01DF 39 RTS  
*  
> 01E0 C6 34 A DNRTN LDB #WRGMOD ;SET WRONG MODE ERROR  
> 01E2 7E 015E P JMP ERROR
```

```

*
* MOVE PROBE TO (X,Y) POSITION
*
* THIS ROUTINE HAS SEVERAL TASKS. THIS
* ROUTINE WILL ALSO HANDLE (D,E) COORDINATE
* FIRST, THE (X,Y) OR (D,E) COORDINATES ARE
* GATHERED AND VALIDATED. NEXT THE NUMBERS
* CONVERTED INTO A VALID BINARY FORMAT. IF
* THE NUMBERS ARE INVALID, AN ERROR WILL BE
* GENERATED UP REQUEST BY THE HOST COMPUTER
* OTHERWISE, THE PROBER WILL BE MOVED TO TH
* REQUESTED (X,Y)POS. HOW THIS IS DONE DEPE
* ON WHAT OPERATION MODE IS REQUESTED.
*
* MODE 0 = AUTOMATIC - A AUTOMATIC PROBE UP
* THEN AN AUTOMATIC MOVE TO (X,Y)
* AND THEN FINALLY AND AUTOMATIC
* DOWN.
*
* MODE 1 = MANUAL - EXACTLY AS IN MODE 0 EX
* THAT THE USER WILL VERIFY THAT
* CORRECT (X,Y) POSITION. ONCE VE
* APROBE DOWN COMMAND WILL BE AUT
* GIVEN.
*
* MODE 2 = TEST - THE ONLY AUTOMATIC MOVEME
* PROBE UP COMMAND. SPECIFIC MOV
* (X,Y) OR (D,E) AND PROBE UP/DOW
* BE GIVEN TO THE HOST COMPUTER.
*
* MODE 3 = IDLE - MOVES TO (X,Y) WILL BE IG

```

```

01E5 96 A2 A XYMOVE LDA CURMOD ;VALID OPER MODE?
01E7 81 03 A CMPA #IDLMOD
01E9 26 05 01F0 BNE XYM010 ;YES
*
* ERROR - CANT OPERATE IN IDLE MODE
*
01EB C6 34 A LDB #WRGMOD
01ED 16 00C4 02B4 LBRA DEMERR
*
01F0 96 A7 A XYM010 LDA ABFLAG ;IS CAMERA OFFSET SET?
01F2 85 04 A BITA #BOFFST
01F4 26 05 01FB BNE XYM020 ;YES
*
01F6 C6 36 A LDB #NOFSET
01F8 16 00B9-02B4 LBRA DEMERR ;REPORT ERROR
*
01FB 10BE 01AB A XYM020 LDY #EXBUFR+1 ;START OF STRING
01FF C6 05 A LDB #5
0201 8D 00CA P JSR VALNUM ;VALID NUMBER?
0204 25 26 022C BCS XYM050 ;NO
0206 D7 C4 A STB XDSIGN ;SAVE SIGN OF X OR D
*
0208 8D 0000 A JSR CASCB
0208 00BF A FDB NUMBUF
020D 05 A FCB 5
020E 00C5 A FDB TEMPX ;SAVE BINARY RESULT

```

22 HY11CMDP.SA:0 COMAND COMMAND PROCESSOR

*
* CHECK FOR PROPER FORMAT OF STRING
*

0210 A6 A0 A LDA ,Y+ ;GET NEXT CHAR
0212 0D AB A TST DEMODE ;IN DE MODE?
0214 27 06 021C BEQ XYM025 ;NO
*
0216 B1 45 A CMPA #45 ;IS IT AN 'E' ;YES
0218 27 08 0225 BEQ XYM040 ;YES, VALID FORMAT
021A 20 04 0220 BRA XYM030 ;NO

*
021C B1 59 A XYM025 CMPA #59 ;IS IT A 'Y'
021E 27 05 0225 BEQ XYM040 ;YES

*
* ERROR -INVALID FORMAT
*

0220 C6 31 A XYM030 LDB #INVCHR
0222 16 008F 02B4 LBRA DEMERR
*
0225 C6 05 A XYM040 LDB #5
0227 ED 00CA F JSR VALNUM ;VALID 'Y' OR 'E'?
022A 24 05 0231 BCC XYM060 ;YES

*
* ERROR - INVALID COMMAND ARGUMENT
*

022C C6 33 A XYM050 LDB #INVARG
022E 16 0083 02E4 LBRA DEMERR
*
0231 D7 C9 A XYM060 STB YESIGN ;SAVE Y/E SIGN
*
0233 ED 0000 A JSR CASCB ;CONVERT TO BINARY
0236 00EF A FDB NUMEBF
0238 05 A FCB 5
0239 00C7 A FDB TEMPY

*
023E 9E C5 A LDX TEMPX
023D 109E C7 A LDY TEMPY
0240 0D AB A TST DEMODE ;IN DE MODE?
0242 26 07 024E BNE XYM070 ;YES

*
0244 9F 3C A STX NX ;UPDATE NEXT X
0246 109F 3F A STY NY ;UPDATE NEXT Y
0249 20 2E 0276 BRA XYM100

*
024E 9F 41 A XYM070 STX DD ;UPDATE D EQUATION
024D 109F 45 A STY EE ;UPDATE E EQUATION

*
0250 BD 0000 A JSR SFLOAT ;FLOAT TO 32 BIT INTEG
0253 0041 A FDB DD
0255 0041 A FDB DD

*
0257 BD 0000 A JSR SFLOAT
025A 0045 A FDB EE
025C 0045 A FDB EE

*
025E D6 41 A LDB DD
0260 0D C4 A TST XDSIGN ;IS D VAL POS?
0262 27 04 0268 BEQ XYM080 ;YES

HY11CMDP.SA:0 COMAND COMMAND PROCESSOR

```

64 CA 80 A ORB *$B0 ;NO, SET MANTISSA
66 D7 41 A STB DD ;SIGN NEGATIVE
*
68 D6 45 A XYM080 LDB EE
6A 0D C9 A TST YESIGN ;IS E VAL POS?
6C 27 04 0272 BEQ XYM090 ;YES
6E CA 80 A ORB *$B0 ;NO, SET MANTISSA
70 D7 45 A STB EE ;SIGN NEGATIVE
*
72 4F XYM090 CLRA ;USE REF 1
73 BD 0000 A JSR CXYFF ;CONVERT TO (X,Y) FORM
*
* AT THIS POINT, BOTH PARAMETERS HAVE BEEN U
* CONVERTED AND SIGN ADJUSTED IF NECESSARY.
* ON WITH THE MOVEMENT.
*
76 BD 01BC F XYM100 JSR UPPROB ;RAISE PROBE
79 96 A2 A LDA CURMOD
7E 81 00 A CMPA #AUTMOD ;IN AUTO MODE?
7D 27 19 0298 BEQ XYM105 ;YES
*
* SEEK POINT RELATIVE TO CAMERA
*
7F BD 0000 A JSR MVPOS ;MOVE TO NEW (X,Y)
*
* USER IN MANUAL MODE. ALLOW TO JOG.
*
82 4F CLRA
83 ED 0000 A JSR JOG
86 26 1F 02A7 BNE XYM120 ;USER ABORTED
*
88 96 A2 A LDA CURMOD
8A 81 02 A CMPA #TSTMOD ;ARE WE FINISHED?
8C 27 19 02A7 BEQ XYM120 ;YES
*
* SEEK TRUE POINT
*
8E 9E 36 A LDX CX ;UPDATE NEXT
90 109E 39 A LDY CY
93 9F 3C A STX NX
95 109F 3F A STY NY
98 86 01 A XYM105 LDA #1
9A 97 B2 A STA UOFSET ;REQUEST OFFSET WHEN M
9C BD 0000 A JSR MVPOS
9F 0F B2 A CLR UOFSET
*
A1 BD 01CE F XYM110 JSR DNPROB ;LOWER THE PROBE
A4 7D E710 A TST PRSWIT ;CLOSE SWITCH
A7 0F AB A XYM120 CLR DEMODE
A9 39 RTS

```

HY11CMDP.SA:0 COMAND COMMAND PROCESSOR

```

*
* MOVE PROBE TO (D,E) POSITION
*
* THE PROBER SOFTWARE HAS BEEN DESIGNED TO
* ONLY MOVE TO (X,Y) COORDINATE POSITIONS.
* THEREFORE,THE (D,E) COORDINATES MUST BE
* TRANSFORMED INTO THEIR RESPECTIVE (X,Y)
* COORDINATES. ALL OF THE RESTRICTIONS AND
* MODE RULES OF THE XYMOVE ARE VALID AND,
* IN ADDITION, THE (D,E) COORDINATES WILL
* BE IGNORED UNTIL BOTH A ND B ALIGNMENTS
* HAVE BEEN DESIGNATED BY THE HOST.

```

```

* CALLING PARAMTERS: JSR DEMOVE

```

```

AA 96 A7 A DEMOVE LDA ABFLAG
AC 84 43 A ANDA #$43 ;MASK AB BITS
AE 81 43 A CMPA #$43 ;HAS A/B BEEN GIVEN?
E0 27 08 02BA BEQ DEM020 ;YES

```

```

*
* ERROR - A/B ALIGNMENT NOT GIVEN

```

```

E2 C6 35 A LDB #UNDFAB
E4 ED 015B F DEMERR JSR ERROR
E7 0F A8 A CLR DEMODE ;RESET MODE (D,E)
E9 39 RTS

```

```

*
BA 86 FF A DEM020 LDA #$FF
EC 97 A8 A STA DEMODE ;SET TO (D,E) MODE
BE 7E 01E5 F JMF XYMOVE

```

AD-A152 687

MANUFACTURING METHODS AND TECHNOLOGY FOR DIGITAL FAULT
ISOLATION OF HYBRI. (U) HUGHES AIRCRAFT CO FULLERTON CA
GROUND SYSTEMS GROUP 01 MAR 82 MAC-FR-82-12-193
DRAH01-81-D-A002

2/2

UNCLASSIFIED

F/G 9/5

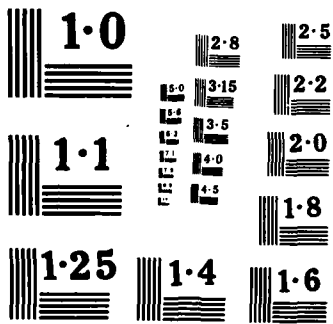
NL



END

FILMED

DTIC



```

00929      *
00930      * AB HYBRID ALIGNMENT COORDINATES
      *
00931      *
00932      * THIS ROUTINE IS TRIGGERED WHEN THE A/B
00933      * ALIGNMENT INFORMATION IS SENT BY THE HOST
00934      * FORMAT:
00935      *
00936      * OR
00937      *
00938      *
00939P 02C1 10BE 01AC A ABALIG LDY #EXEUF+2
00940P 02C5 C6 05 A LDB #5
00941P 02C7 BD 00CA P JSR VALNUM ;VALID NUMBER
00942P 02CA 24 12 02DE BCC ABAL10 ;YES
00943      *
00944      * ERROR - INVALID A/B ARGUMENT
00945      *
00946P 02CC 8E 01AA A ABERR0 LDX #EXBUFR
00947P 02CF A6 84 A LDA ,X ;GET FIRST CHAR
00948P 02D1 81 41 A CMPA #41 ;IS IT A ALIGNMENTS
00949P 02D3 27 04 02D9 BEQ ABERR1 ;YES
00950      *
00951      * ERROR- INVALID B ARGUMENT
00952      *
00953P 02D5 C6 33 A LDB #INVARG
00954P 02D7 20 02 02DB BRA ABERR
00955P 02D9 C6 33 A ABERR1 LDB #INVARG
00956P 02DB 7E 015B P ABERR JMP ERROR
00957      *
00958P 02DE BD 0000 A ABAL10 JSR CASCB ;CONVERT X COORD
00959P 02E1 00BF A FDB NUMBUF
00960P 02E3 05 A FCB 5
00961P 02E4 00C5 A FDB TEMPX
00962      *
00963P 02E6 8E 01AA A LDX #EXBUFR
00964P 02E9 A6 84 A LDA ,X
00965P 02EB A1 A0 A CMPA ,Y+ ;DO ARGUMENTS MATCH?
00966P 02ED 26 06 02F5 BNE ABAL20 ;NO, ERROR
00967      *
00968P 02EF A6 A0 A LDA ,Y+ ;GET NEXT CHAR
00969P 02F1 81 59 A CMPA #59 ;IS IT Y
00970P 02F3 27 04 02F9 BEQ ABAL30
00971      *
00972      * ERROR -COMMAND SYNTAX ERROR
00973      *
00974P 02F5 C6 30 A ABAL20 LDB #UNKCMD
00975P 02F7 20 E2 02DB BRA ABERR
00976      *
00977P 02F9 C6 05 A ABAL30 LDB #5
00978P 02FB BD 00CA P JSR VALNUM ;VALID Y COORD?
00979P 02FE 25 CC 02CC BCS ABERR0 ;NO
00980      *
00981P 0300 BD 0000 A JSR CASCB ;CONVERT E TO ASCII
00982P 0303 00BF A FDB NUMBUF
00983P 0305 05 A FCB 5
00984P 0306 00C7 A FDB TEMPY
00985      *
00986P 0308 8E 01AA A LDX #EXBUFR

```

```

00987P 030B A6 80 A LDA ,X+
00988P 030D 81 41 A CMPA #541 ;WAS THIS 'A' ALIGNMEN
00989P 030F 26 10 0321 BNE ABAL50 ;NO
00990 *
00991P 0311 9A A7 A ORA ABFLAG ;SET A FLAG TO INDIC
00992P 0313 97 A7 A STA ABFLAG ;THAT A DATA RECEIVES
00993P 0315 9E C5 A LDX TEMPX ;SET A COORDINATE
00994P 0317 BF 0100 A STX RAX
00995P 031A 9E C7 A LDX TEMPY
00996P 031C BF 0102 A STX RAY
00997P 031F 20 0E 032F BRA ABAL60
00998 *
00999P 0321 9A A7 A ABAL50 ORA ABFLAG ;INDICATE 'B'ALIG
01000P 0323 97 A7 A STA ABFLAG ;WAS SET
01001P 0325 9E C5 A LDX TEMPX
01002P 0327 EF 0104 A STX RBX
01003P 032A 9E C7 A LDX TEMPY
01004P 032C EF 0106 A STX REY
01005 84 43 A ABAL60 ANDA #543
01006P 032F 81 43 A CMPA #543
01007P 0331 21 03 A BEQ ABSET
01008P 0333 39 A RTS
01009P 0334 32 A NOP
01010P 0337 12 A NOP

```

```

01012 *
01013 * NAME: ABSET
01014 *
01015 * FUNCTION: THIS SUBROUTINE WILL ALLOW THE
01016 * THE USER TO SET THE A/B ALIGNME
01017 * VALUES MANUALLY.
01018 *
01019 * CALLING SEQUENCE: JSR ABSET
01020 *
01021 * UPON RETURN: A/B VALUES SET IF PROC
01022 * WAS NOT ABORTED.
01023 *
01024P 0338 BD 0000 A ABSET JSR ONLTE
01025P 033E 30 A FCB LABLGN
01026 *
01027P 033C 9E 36 A LDX CX
01028P 033E 9F 94 A STX CXSAVE ;SAVE CURRENT POS
01029P 0340 9E 39 A LDX CY
01030P 0342 9F 96 A STX CYSAVE
01031P 0344 96 A4 A LDA ZDIR
01032P 0346 97 77 A STA ZSAVE
01033 *
01034P 0348 D6 A7 A LDB ABFLAG ;HAS A VAL ALREADY BEE
01035P 034A C5 01 A BITB #1
01036P 034C 27 09 0357 BEQ ABS015 ;NO
01037 *
01038P 034E BE 0100 A LDX RAX ;GO TO CUURENT VALUES
01039P 0351 10BE 0102 A LDY RAY
01040P 0355 20 1A 0371 BRA ABS020
01041 *
01042 * LEARN THE A REFERENCE POINT
01043 *
01044P 0357 BD 0000 A ABS015 JSR ENTER
01045P 035A 04AF P FDB ENTAX
01046P 035C 17 A FCB 23
01047P 035D 05 A FCB 5 ;ENTER AX COORD
01048 *
01049P 035E 26 7A 03DA BNE ABS060 ;USER ABORTED
01050P 0360 109F 8C A STY SREFAX ;SAVE REF AX
01051 *
01052P 0363 BD 0000 A JSR ENTER
01053P 0366 04CB P FDB ENTAY
01054P 0368 17 A FCB 23
01055P 0369 05 A FCB 5 ;ENTER REF AY
01056 *
01057P 036A 26 6E 03DA BNE ABS060 ;USER ABORTED
01058P 036C 109F 8E A STY SREFAY ;SAVE REF AY
01059 *
01060 * LOCATE THE A POSITION AND ALLOW TO JOG
01061 *
01062P 036F 9E 8C A LDX SREFAX
01063P 0371 BD 011D P ABS020 JSR LOCATE ;LOCATE POSITION
01064P 0374 25 50 03C6 BCS ABS050 ;USER ABORTED
01065 *
01066P 0376 9F 8C A STX SREFAX ;HOLD NEW A
01067P 0378 109F 8E A STY SREFAY
01068 *
01069 * NOW DO THE SAME FOR THE B POINT

```

```

01070
01071P 037B D6 A7 A LDB ABFLAG ;IS B VALUE KNOWN?
C 2P 037D C5 DC A BITB #2
01073P 037F 27 09 03BA BEQ ABS030 ;NO
01074
01075P 0381 BE 0104 A LDX RBX ;GO TOKNOWN LOCATION
01076P 0384 10BE 0106 A LDY RBY
01077P 0388 20 1A 03A4 BRA ABS040
01078
01079 * LEARN THE B REFERENCE POINT
01080
01081P 038A BD 0000 A ABS030 JSR ENTER
01082P 038D 04E1 P FDE ENTBX
01083P 038F 17 A FCB 23
01084P 0390 05 A FCB 5 ;ENTER BX COORD
01085
01086P 0391 26 33 03C6 BNE ABS050 ;USER ABORTED
01087P 0393 109F 90 A STY SREFBX ;SAVE NEW REF BX
01088
01089P 0396 ED 0000 A JSR ENTER
01090P 0399 04FA P FDE ENTBY
01091P 039B 17 A FCB 23
01092P 039C 05 A FCB 5 ;ENTER BY COORD
01093
01094P 039D 26 27 03C6 BNE ABS050 ;USER ABORTED
01095P 039F 109F 92 A STY SREFBY ;SAVE NEW REF BY
01096
01097 * LOCATE THE B REFERENCE POINT
01098
01099P 03A2 9E 90 A LDX SREFBX
01100P 03A4 BD 011D P ABS040 JSR LOCATE ;LOCATE B
01101P 03A7 25 1D 03C6 BCS ABS050 ;USER ABORTED
01102
01103P 03A9 BF 0104 A STX REX ;UPDATE TO NEW POS
01104P 03AC 10BF 0106 A STY RBY
01105
01106P 03B0 9E 8C A LDX SREFAX
01107P 03B2 109E 8E A LDY SREFAY
01108P 03B5 BF 0100 A STX RAX
01109P 03B8 10BF 0102 A STY RAY
01110
01111 * TRANSFORM THE NEW REFERENCE POINTS
01112
01113P 03BC 4F CLRA
01114P 03BD BD 0000 A JSR CREFD
01115
01116P 03C0 D6 A7 A LDB ABFLAG
01117P 03C2 CA 43 A ORB #43 ;SET TO AB KNOWN
01118P 03C4 D7 A7 A STB ABFLAG
01119
01120 * RETURN TO PREVIOUS POSITION
01121
01122P 03C6 9E 94 A ABS050 LDX CXSAVE
01123P 03C8 109E 96 A LDY CYSAVE
01124P 03CB 9F 3C A STX NX
01125P 03CD 109F 3F A STY NY ;UPDATE NEXT POS
01126P 03D0 BD 0000 A JSR MVPOS
01127

```


01128P	0303	0D	77	A	TST	ZSAVE	!WAS PROBE UP
01129P	0305	27	03	03DA	BEQ	ABS060	!YES
30P	0307	BD	01CE	P	JSR	DNPROB	
01131P	03DA	BD	0000	A	ABS060	JSR	OFFLYE
01132P	03DD		30	A	FCB	LABLGN	!TURN OFF AB LIGHT
01133P	03DE	39			ABABRT	RTS	
01134							
01135							

```

01137          *
01138          * HOME THE PROBE
01139          *
01140          * RETURN THE PROBE TO ITS HOME POSITION
01141          *
01142
01143P 03DF B0 01C2 P HOPROB JSR UPALWY ;RAISE THE PROBE
01144P 03E2 C6 03 A LDE #3
01145P 03E4 B0 0000 A JSR ZTABLE MOVE TO (0,0)
01146P 03E7 39 RTS
01147
01148          *
01149          * UNLOAD THE HYBRID
01150          *
01151P 03E8 7E 0000 A UNLOAD JMP UNLDKY ;MOVE TO PARK POS
01152          *
01153          * LOAD THE HYBRID
01154          *
01155P 03EB 7E 0000 A LOADHY JMP LOADKY

```

```

57 *
58 * THE FOLLOWING SET OF SUBROUTINES WILL GATH
59 * THE REQUESTED INFORMATION AND STORE IT IN
60 * THE OUTPUT QUEUE. WHEN THE PROBER BECOMES
61 * A TALKER, THE INFORMATION WILL BE SENT TO
62 * CONTROLLER VIA IEEE-488 PROTOCOL
63 *
64 *
65 *
66 * SEND (X,Y)POSITION TO CONTROLLER
67 *
68 * UPON REQUEST OF A 'SP' (SEND X,Y POSITIO
69 * THIS ROUTINE WILL FETCH THE CURRENT X AN
70 * POSITION OF THE PROBE, CONVERT IT TO ASC
71 * ANDSTORE THE INFORMATION IN THE OUTPUT Q
72 *
73 * CALLING SEQUENCE: JSR XYSEND
74 *
75P 03EE BD 0000 A XYSEND JSR CBASC ;CONVERT CUR X TO ASCI
76P 03F1 0036 A FDE CX
77P 03F3 01D4 A FDE CXYPOS+1
78P 03F5 05 A FCB 5
79 *
80P 03F6 BD 0000 A JSR CBASC ;CONVERT CUR Y TO ASCI
81P 03F9 0039 A FDE CY
82P 03FE 01DA A FDE CXYPOS+7
83P 03FD 05 A FCB 5
84 *
85 * STORE INFO INTO OUTPUT QUEUE
86 *
87P 03FE 108E 01D3 A LDY #CXYPOS ;POINTER TO ASCII INFO
88 *
89P 0402 8E 0466 A XYS010 LDX #OUTBCF
90P 0405 E6 A0 A XYS020 LDE ,Y+ ;GET CHAR
91P 0407 BD 0065 F JSR PUTCHR ;FIT IN BUFFER?
92P 040A 25 06 0412 BCS XYS050 ;NO
93 *
94P 040C C1 0A A CMPE #STRTRM ;LAST CHAR?
95P 040E 26 F5 0405 BNE XYS020 ;NO
96 *
97P 0410 0C AE A INC OUTCNT ;YES, SET OUT COUNT
98P 0412 39 XYS050 RTS
99

```

GE 032 HY11CMDP.SA:0 COMAND COMMAND PROCESSOR

```

201      *
202      * SEND (D,E) POSITION TO CONTROLLER
      *
203      *
204      * UPON REQUEST OF A 'SD' (SEND D,E POSITIO
205      * THIS ROUTINE WILL FETCH THE CURRENT (X,Y
206      * POSITION AND CONVERT IT TO A 4 BYTE SIGN
207      * RELATIVE BINARY NUMBER.
208      *
209      * CALLING SEQUENCE: JSR DESEND
210      *
211P 0413 96 A7 A DESEND LDA ABFLAG ;HAS AB ALIGN BEEN GIV
212P 0415 81 47 A CMPA ##47 ;A+B+OFFSET KNOWN
213P 0417 27 05 041E BEQ DES010 ;YES
214      *
215      * ERROR - NOT VALID BECAUSE A/E ALIGNMENT
216      * HAS NOT BEEN GIVEN BY HOST YET.
217      *
218P 0419 C6 35 A LDE #UNDFAB;
219P 041E 7E 015B F JMP ERROR
220      *
221P 041E 4F DES010 CLRA ;USE REF 1
222P 041F 8D 0000 A JSR CDEV ;CONVERT X,Y TO D,E
223P 0422 86 2E A LDA ##2E ;ASSUME POSITIVE
224P 0424 0D 41 A TST DD ;IS VALUE OF D POS?
225P 0426 2A 09 0431 BPL DES020 ;YES
226      *
227      * NEGATE DD TO MAKE A POSITIVE VALUE
228      *
229P 0428 CC 0000 A LDD #0
230P 042E 93 41 A SUBD DD
231P 042D DD 41 A STD DD
232      *
233P 042F 86 2D A LDA ##2D ;MAKE - SIGN
234P 0431 E7 01E2 A DES020 STA CDEPOS+1 ;SAVE SIGN
235P 0434 BD 0000 A JSR CBASC ;CONVERT DD TO ASCII
236P 0437 0041 A FDB DD
237P 0439 01E3 A FDB CDEPOS+2
238P 043E 05 A FCB 5
239      *
240      * NOW DO THE SAME FOR THE EE COORDINATE
241      *
242P 043C 86 2E A LDA ##2E ;ASSUME POS
243P 043E 0D 45 A TST EE ;IS EE POS?
244P 0440 2A 09 044E BPL DES050 ;YES
245      *
246      * NEGATE TO MAKE IT POSITIVE
247      *
248P 0442 CC 0000 A LDD #0
249P 0445 93 45 A SUBD EE
250P 0447 DD 45 A STD EE
251      *
252P 0449 86 2D A LDA ##2D ;MAKE SIGN NEG
253P 044B E7 01E9 A DES050 STA CDEPOS+8 ;SAVE SIGN
254      *
255      * CONVERT EE POSITION TO ASCII
256      *
257P 044E BD 0000 A JSR CBASC
258P 0451 0045 A FDB EE

```

E 033 HY11CMDP.SA:0 COMAND COMMAND PROCESSOR

59P	0453	01EA	A	FDB	CDEPOS+9	
60P	0455	05	A	FCB	5	
61						*
62P	0456	108E	01E1	A	LDY	#CDEPOS ;POINTER TO ASCII INFO
63P	045A	7E	0402	P	JMP	XY6010

```

55
56
57 * SEND THE PROBER ERROR CODE
58 * THIS ROUTINE IS TRIGGERED ONLY WHEN TH
59 * CONTROLLER REQUESTS WHAT ERROR LAST OCCU
60 * THE CONTROLLER CAN DETECT A PROBER ERROR
61 * BY REQUESTING STATUS AND POLLING THE ERR
62 * BIT (BIT 2) OF THE STATUS BYTE. THIS ER
63 * BIT AND THE CODE BYTE WILL BE CLEARED AT
64 * EXEC LEVEL IF THE NEXT COMMAND FOLLOWING
65 * ERROR IS NOT EITHER A STATUS REQUEST OR
66 * THE "SE" (SEND PROBER ERROR REQUEST) COM
67 * AFTER THE ERROR REQUEST IS SENT THE ERRO
68 * BIT WILL BE CLEARED AND THE ERROR CODE S
69 * TO NO ERROR.
70
71
72
73
74
75
76
77
78
79
80
81 * CALLING PARAMETER: LDB PROBE
82 * JSR ERSE
83
84F 045D 34 04 A ERSEND FSHS B ;SAVE ERROR CODE
85F 045F 8E 0466 A LDX #OUTBCF ;INIT OUTPUT BCF
86F 0462 CC 04D4 A LDD #OUTBUF
87F 0465 108E 0539 A LDY #OUTEND
88F 0469 BD 0104 F JSR INTBCF
89
90F 046C 35 04 A PULS B ;RESTORE CODE
91F 046E 8E 0466 A LDX #OUTBCF
92F 0471 BD 0065 F JSR PUTCHR
93
94F 0474 BD 010E F JSR OTCRLF ;OUTPUT CR/LF
95F 0477 0C AE A INC OUTCNT ;SET FOR REQUEST
96F 0479 39 RTS
97
98
99 * FORM STATUS BYTE
100 * THIS ROUTINE WILL GATHER THE STATE OF SEVE
101 * FLAGS AND COMBINE THEM INTO A SINGLE BYTE
102 * AS FOLLOWS:
103
104 * STATUS BYTE CONFIGURATION = 01ZBOEMM
105 * WHERE:
106
107 * Z = (0/1) PROBE Z POSITION (UP/DOWN)
108 * B = (0/1) PROBE BUSY (NOT BUSY/BUSY)
109 * O = (0/1) ONLINE/OFFLINE FLAG
110 * E = (0/1) PROBER ERROR (NO ERROR/ERROR)
111 * MM = 00 - AUTOMATIC MODE
112 * 01 - MANUAL/STEP MODE
113 * 10 - TEST MODE
114 * 11 - IDLE MODE
115
116 * THIS ROUTINE IS TRIGGERED WHEN THE CONTROL
117 * SENDS THE "??" STATUS REQUEST COMMAND.
118
119 * CALLING PARAMTERS: JSR FORMST
120
121 * RETURN: A-REG CONTAINS STATUS BYTE
122

```

035 HY11CMDP.SA:0 COMAND COMMAND PROCESSOR

```

047A B6 40 A FORMST LDA #40 ;INIT THE STATUS BYTE
047C 34 02 A PSHS A ;WORK OFF OF STACK
047E D6 A4 A LDB ZDIR ;CURRENT Z DIRECTION
0480 C4 20 A ANDB #ZBIT
0482 EA E4 A ORB 0,S ;COMBINE
0484 E7 E4 A STB 0,S
*
0486 0D AF A TST INCNT ;DEVICE BUSY?
0488 27 06 0490 BEQ FORM10 ;NO
048A C6 10 A LDB #BBIT ;YES
048C EA E4 A ORB 0,S
048E E7 E4 A STB 0,S
*
0490 D6 9F A FORM10 LDB ONOFFL ;GET ONLINE/OFFLINE ST
0492 C4 18 A ANDB #OBIT+BBIT ;SET BUSY ALSO IF
0494 EA E4 A ORB 0,S ;IN OFFLINE
0496 E7 E4 A STB 0,S
*
0498 D6 A6 A LDB ESTATE ;CURRENT ERROR STATE
049A C4 04 A ANDB #EBIT
049C EA E4 A ORB 0,S
049E E7 E4 A STB 0,S
*
04A0 D6 A2 A LDB CURMOD ;CURRENT OPERATIONS MO
04A2 C4 03 A ANDB #MBIT
04A4 EA E4 A ORB 0,S
04A6 E7 E4 A STB 0,S
*
04AB 35 02 A PULS A ;GET FINAL RESULT
04AA 39 RTS

```

036 HY11CMDP.SA:0 COMAND COMMAND PROCESSOR

```

04AB 0000 A ZERO32 FDB $0,$0
04AF 45 A ENTAX FCC /ENTER AX POSITION: /
04C7 0D A FCC $0D
04C8 45 A ENTAY FCC /ENTER AY POSITION: /
04E0 0D A FCC $0D
04E1 45 A ENTBX FCC /ENTER BX POSITION: /
04F9 0D A FCC $0D
04FA 45 A ENTEY FCC /ENTER BY POSITION: /
0512 0D A FCC $0D
0513 4F A ONTRGM FCC /ON TARGET?/
051D 0D A FCC $0D
051E 4D A MOVMSG FCC /MOVE TO TEST POINT/
0530 0D A FCC $0D

```

*
* COMMAND TABLE
*

```

> 0531 4D A CMDTBL FCC /MS/
> 0533 FC35 A FDB GETMSG-* ;GET MESSAGE
> 0535 4D A FCC /M0/
> 0537 FC3A A FDB AUTPRE-* ;PROBE MODE TO AUTO
> 0539 4D A FCC /M1/
> 053E FC4E A FDB MANPRE-* ;PROBE MODE TO MANUAL
> 053D 4D A FCC /M2/
> 053F FC61 A FDB TSTPRE-* ;PROBE MODE TO TEST
> 0541 55 A FCC /UP/
> 0543 FC79 A FDB UPPROB-* ;RAISE PROBE
> 0545 44 A FCC /DN/
> 0547 FC87 A FDB DNPROB-* ;LOWER PROBE
> 0549 58 A FCC /X/
> 054A 00 A FCC DONTCR
> 054B FC9A A FDB XYMOVE-* ;MOVE TO X,Y
> 054D 44 A FCC /D/
> 054E 00 A FCC DONTCR
> 054F FD5B A FDB DEMOVE-* ;GO TO DE
> 0551 48 A FCC /HO/
> 0553 FE8C A FDB HOPROB-* ;HOME PROBE TABLE
> 0555 41 A FCC /AX/
> 0557 FD6A A FDB ABALIG-* ;A OF AB ALIGN
> 0559 42 A FCC /BX/
> 055E FD66 A FDB ABALIG-* ;B OF AB ALIGN
> 055D 55 A FCC /UL/
> 055F FE89 A FDB UNLOAD-* ;UNLOAD HYBRID
> 0561 4C A FCC /LD/
> 0563 FE88 A FDB LOADHY-* ;LOAD HYBRID

```

*
* STATUS REQUEST PROCESSORS
*

```

> 0565 53 A FCC /SP/
> 0567 FE87 A FDB XYSEND-* ;SEND X,Y POSITION
> 0569 53 A FCC /SD/
> 056B FEAB A FDB DESEND-* ;SEND D,E POSITION
> 056D 53 A FCC /SE/
> 056F FEED A FDB ERSEND-* ;SEND ERROR CODE
> 0571 FF A FCC EOTCHR

```

END

TOTAL ERRORS 00000--00000
 TOTAL WARNINGS 00000--00000

P 03DE ABABRT 01133*
 P 02DE ABAL10 00942 00958*
 P 02F5 ABAL20 00966 00974*
 P 02F9 ABAL30 00970 00977*
 P 0321 ABAL50 00989 00999*
 P 032F ABAL60 00997 01006*
 P 0337 ABAL70 01007 01010*
 P 02C1 ABALIG 00939*01394 01396
 P 02DB ABERR 00954 00956*00975
 P 02CC ABERR0 00946*00979
 P 02D9 ABERR1 00949 00955*
 00A7 ABFLAG 00159*00768 00912 00991 00992 00999 01000 01034 01071
 01116 01118 01211
 P 0357 ABS015 01036 01044*
 P 0371 ABS020 01040 01063*
 P 038A ABS030 01073 01081*
 P 03A4 ABS040 01077 01100*
 P 03C6 ABS050 01064 01086 01094 01101 01122*
 P 03DA ABS060 01049 01057 01129 01131*
 DP 0338 ABSET 00012 01024*
 0085 APUERR 00048*
 0000 AUTMOD 00088*00601 00608 00861
 0171 AUTPRB 00010 00600*01376
 0186 AUTPRT 00602 00612*00641 00668
 0086 AXISER 00049*
 00A0 AXISIN 00066*
 0090 AXISTO 00052*
 0032 BADCTR 00035*00267
 0010 BBIT 00095*01333 01338
 0004 BOFFST 00107*00769
 0024 BUFLN 00104*00213 00220
 R CASCB 00017*00781 00815 00958 00981
 R CBASC 00018*01175 01180 01235 01257
 01E1 CDEPOS 00170*01234 01237 01253 01259 01262
 R CDEV 00019*01222
 P 0531 CMDTBL 00251 01373*
 P 0009 CMIN15 00215*00217
 P 0014 CMIN20 00221*00230
 P 0025 CMIN35 00235*00239
 P 0031 CMIN37 00223 00237 00241*
 P 0030 CMIN40 00227 00251*
 P 005F CMIN50 00255 00272*
 P 0061 CMIN60 00268 00273*
 P 0064 CMIN70 00266 00274*
 DP 0000 CMINTP 00008 00211*
 R CREFD 00020*01009 01114
 00A2 CURMOD 00139*00600 00609 00628 00637 00656 00665 00690 00712
 00759 00860 00875 01347
 0036 CX 00016*00607 00635 00663
 0094 CXSAVE 00171*01028 01122
 R CXYPF 00018*00853
 01D3 CXYPOS 00169*01177 01182 01187

0039 CY 00115*00534 00882 01029 01181
 0096 CYSAVE 00172*01030 01123
 0041 DD 00118*00829 00833 00834 00840 00844 01224 01230 01231
 01236
 P 02BA DEMO20 00915 00924*
 P 02B4 DEMERR 00766 00773 00802 00811 00920*
 00A8 DEMODE 00160*00431 00789 00822 00892 00921 00925
 P 02AA DEMOVE 00912*01390
 P 041E DES010 01213 01221*
 P 0431 DES020 01225 01234*
 P 044B DES050 01244 01253*
 P 0413 DESEND 01211*01408
 DP 01CE DNPROB 00011 00711*00890 01130 01384
 P 01E0 DNRTN 00692 00714 00723*
 0000 DONTCR 00105*00369 01386 01389
 0087 DSKERR 00050*
 R DSPL 00020*00517 00524 00579
 0004 EBIT 00097*01343
 0045 EE 00119*00830 00837 00838 00846 00850 01243 01249 01250
 01258
 P 04AF ENTAX 01045 01357*
 P 04C8 ENTAY 01053 01359*
 P 04E1 ENTBX 01082 01361*
 P 04FA ENTBY 01090 01363*
 R ENTER 00020*01044 01052 01081 01089
 00FF EOTCHR 00106*00392 01411
 P 015E ERROR 00273 00561*00673 00724 00920 00956 01219
 0165 ERRTN 00563 00569*
 045D ERSEND 01284*01410
 00A6 ESTATE 00158*00262 00567 01342
 01AA EXBUFR 00155*00212 00219 00252 00580 00775 00939 00946 00963
 00986
 R FLSHYN 00021*00520
 P 0490 FORM10 01332 01337*
 DP 047A FORMST 00009 01324*
 R FPSUB 00019*
 P 009A GETC30 00337 00342*
 DP 0086 GETCHR 00008 00222 00236 00329*
 P 0168 GETMSG 00579*01374
 DP 03DF HOPROB 00011 01143*01392
 0003 IDLMOB 00085*00660 00691 00713 00760
 0081 IEEEEER 00044*
 045E INBCF 00152*00221 00235
 00AF INCNT 00154*01331
 DP 0104 INTBCF 00009 00472*01288
 0033 INVARG 00036*00810 00953 00955
 0031 INVCHR 00034*00801
 R JOG 00019*00528 00872
 0030 LABLGN 00029*01025 01132
 0045 LAUTO 00025*00611
 0083 LIMERR 00046*
 P 03EB LOADHY 01155*01400
 LOADKY 00021*01155
 0128 LOC010 00517*00529
 0110 LOCATE 00512*01063 01100
 P 013F LOCRT1 00522 00531*
 P 0142 LOCRTN 00521 00533*
 0040 LSTEP 00027*00640

0042 LTEST 00026*00667
 0033 LUPDWN 00028*00698 00720
 I 0189 MANPRB 00010 00628*01378
 r 019E MANPRT 00630 00641*
 DP 009F MATCH 00012 00254 00362*
 P 00A4 MATCH2 00368*00377 00396
 P 00AE MATCH3 00370 00375*
 P 00B6 MATCH5 00373 00387*
 P 00C7 MATCH7 00393 00398*
 0003 MBIT 00098*01348
 0060 MOVEZ 00127*
 P 051E MOVMSG 00525 01367*
 00A0 MTCNT 00156*00363 00395
 DP 0148 MYPARK 00012 00543*
 R MVPOS 00018*00515 00548 00867 00887 01126
 0036 NOFSET 00039*00772
 00BF NUMBUF 00162*00443 00782 00816 00959 00982
 003C NX 00116*00512 00545 00825 00883 01124
 003F NY 00117*00513 00546 00826 00884 01125
 005E NZ 00120*
 0008 OBIT 00096*01338
 R OFFLTE 00017*00697 01131
 R ONLTE 00016*00610 00639 00666 00719 01024
 009F ONOFFL 00168*00562 01337
 P 0513 ONTRGM 00518 01365*
 0080 OPABRT 00043*
 DP 010E OTCRLF 00009 00485*01294
 0466 OUTBCF 00153*01189 01285 01291
 04D4 OUTBUF 00142*01286
 00AE OUTCNT 00141*01197 01295
 0539 OUTEND 00143*01287
 00A9 OVRFLW 00167*00211 00247 00265
 4E20 FARKX 00108*00543
 4E20 PARKY 00109*00544
 P 0187 PRBERR 00605 00633 00661 00672*
 00A3 PROBER 00138*00260 00261 00565
 E710 PRSWIT 00182*00891
 P 0074 PUTC10 00300 00305*
 P 007C PUTC20 00309*00343
 P 0081 PUTC30 00306 00313*00333
 DP 0065 PUTCHR 00008 00294*00488 00490 01191 01292
 0100 RAX 00122*00994 01038 01108
 0102 RAY 00123*00996 01039 01109
 0104 RBX 00124*01002 01075 01103
 0106 RBY 00125*01004 01076 01104
 DP 015F RERROR 00012 00565*
 R SFLOAT 00018*00832 00836
 00CF SRCPTR 00157*00364 00394
 008C SREFAX 00175*01050 01062 01066 01106
 008E SREFAY 00176*01058 01067 01107
 0090 SREFBX 00177*01087 01099
 0092 SREFBY 00178*01095
 0001 STPMOD 00087*00629 00636
 000A STRTRM 00103*00226 00238 01194
 R TCLRDP 00020*00582
 00C5 TEMPX 00163*00784 00820 00961 00993 01001
 00C7 TEMPY 00164*00818 00821 00984 00995 01003
 R TOUCHD 00017*00716

0084 TRSHER 00047*
 0002 TSTMOD 00086*00604 00632 00657 00664 00876
 L J1A0 TSTPRB 00010 00656*01380
 P 01B5 TBTRTN 00658 00668*
 0035 UNDFAB 00038*00919 01218
 0030 UNKCHD 00033*00272 00974
 0082 UNKINT 00045*
 R UNLDKY 00021*01151
 P 03E8 UNLOAD 01151*01398
 00B2 UNFSET 00128*00886 00888
 P 01C2 UPALNY 00514 00547 00694*01143
 DP 01BC UPPROB 00011 00689*00859 01382
 P 00CE VAL010 00427*00429
 P 00E4 VAL030 00432 00438 00442*
 P 00E6 VAL050 00435 00441 00443*
 P 00E9 VAL060 00444*00452
 P 00FF VALERR 00446 00448 00459*
 P 00CA VALNUM 00423*00777 00805 00941 00978
 009E VALSGN 00161*00425 00440 00455
 0034 WRGMOD 00037*00672 00723 00765
 00C4 XDSIGN 00165*00779 00841
 007B XOFFS 00129*
 P 01F0 XYM010 00761 00768*
 P 01FB XYM020 00770 00775*
 P 021C XYM025 00790 00796*
 P 0220 XYM030 00794 00801*
 P 0225 XYM040 00793 00797 00804*
 022C XYM050 00778 00810*
 0231 XYM060 00806 00813*
 P 024B XYM070 00823 00829*
 P 026B XYM080 00842 00846*
 P 0272 XYM090 00848 00852*
 P 0276 XYM100 00827 00859*
 P 029B XYM105 00862 00885*
 P 02A1 XYM110 00890*
 P 02A7 XYM120 00873 00877 00892*
 P 01E5 XYMOVE 00759*00926 01387
 P 0402 XYS010 01189*01263
 P 0405 XYS020 01190*01195
 P 0412 XYS050 01192 01198*
 P 03EE XYSEND 01175*01406
 00C9 YESIGN 00166*00813 00847
 007A YOFFS 00130*
 0020 ZBIT 00094*01327
 00A4 ZDIR 00137*00699 00718 01031 01326
 P 04AB ZERO32 01356*
 0077 ZSAVE 00173*01032 01128
 R ZTABLE 00019*00695 01145

END

FILMED

5-85

DTIC