UCID- 17955

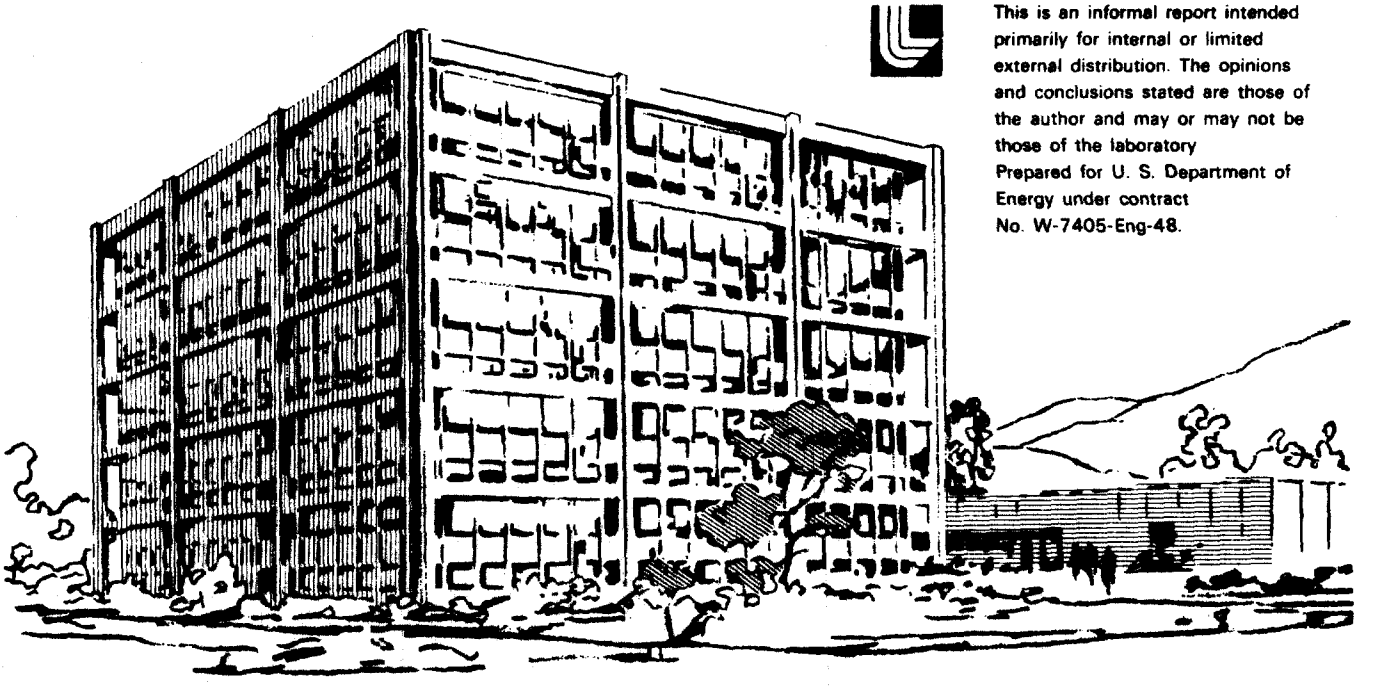
Lawrence Livermore Laboratory

REMOTE SYSTEM CONSOLE

CIRCULATION COPY
SUBJECT TO RECALL
IN TWO WEEKS

T. H. Gong

September 25, 1978



This is an informal report intended primarily for internal or limited external distribution. The opinions and conclusions stated are those of the author and may or may not be those of the laboratory
Prepared for U. S. Department of Energy under contract No. W-7405-Eng-48.

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This report has been reproduced
directly from the best available copy.

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information
P.O. Box 62, Oak Ridge, TN 37831
Prices available from (615) 576-8401, FTS 626-8401

Available to the public from the
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd.,
Springfield, VA 22161

CONTENTS

Abstract 1
Introduction 1
Software Description 2
System Startup 2
I/O Request 2
Restriction 3
Software Installation 3
References 15

REMOTE SYSTEM CONSOLE

ABSTRACT

This report describes a simple method of implementing a remote system console on a remote computer using the Hewlett Packard DS/1000 communication software package on the HP 21MX-E series computer.

INTRODUCTION

The Hewlett Packard RTE-MII operating system[1,2] requires a system console for logging all system-related messages. The LLL Seismic Network project found this console to be unacceptable for our unmanned remote computers, which are located in tunnels not easily accessible. This report describes a simple software package written in LLL SPL language. The package will transmit all messages destined for the system console back to the central computer, which is located in Livermore, California.

This report assumes that the reader is well-versed in using the RTE operating system.

SOFTWARE DESCRIPTION

The software package consists of four modules--DVA00, LSEND, LSTUQ, and STRTM. DVA00 is an I/O driver, which is a replacement for the HP-supplied terminal I/O driver. The driver will intercept all messages and pass them on to program LSEND.

LSEND will use the HP-supplied DS/1000[3,4] subprogram DMESG to transmit the message back to the central computer.

SYSTEM STARTUP

The I/O driver DVA00 must be initialized before the first I/O write request. This is done by program STRTM when RTE-M starts running. The program will perform a control I/O call to logical unit (LU) 1 passing the location of the subsystem global area (SSGA) variable SSGLC.

SSGLC is a lock word used to synchronize the program LSEND and the I/O driver DVA00. The variable will contain the first word address (FWA) of the equipment table (EQT) entry for the driver when an I/O request is made. This variable will put the driver in a busy state until LSEND has transmitted the message to the central computer. The variable will be reset to zero after the message is transmitted to allow DVA00 to process the next request, if any.

I/O REQUEST

When a I/O request is performed, the RTE-M system will execute the standard I/O call procedure until it reaches driver DVA00. In the driver, SSGLC will be set to the first word address of the EQT entry for the driver. The driver will then schedule program LSEND to transmit the message to the central computer. The driver will then set the EQT word 4 for time-out processing and EQT word 15 for a time-out of 50 ms.

After the time-out, the driver is reentered in the continuation/completion section. If SSGLC is not zero, EQT word 15 is reset to the 50-ms time-out value, and a continuation exit is taken. If SSGLC is zero, EQT word 15 is set to zero and a completion exit is taken.

If the driver is busy (SSGLC $\neq$ 0), all I/O requests will be either buffered or queued.

RESTRICTION

This software package will not run under any RTE operating system with a dynamic memory mapping system, because the data buffer may not be mapped in with the program LSEND. The message length must not be over 72 bytes long.

SOFTWARE INSTALLATION

To install the software package perform the following steps:

1. Add module %DVA00 to the relocation of the system modules.
2. Set the buffer limits to 128,256.
3. Assign DVA00 to EQT# 2 as follows:

   25,DVA00, B

   The select code should be one that is not assigned to any device.
4. For device reference table (DRT) entry, enter 2,0 for logical unit 1.
5. Add module %LSTUQ to the relocation section of SSGA.
6. Relocate program LSEND as a regular application program.
7. Modify the startup program, STRTM, to pass the location of variable SSGLC to driver DVA00.

See Figs. 1 through 5 for detailed listing of modules and installation examples.

```
ØØØ1
ØØØ2          NAME DVAØØ (Ø) "SYSTEM MESSAGE PROCESSOR DRIVER "
ØØØ3   !
ØØØ4   ! . . .    SYSTEM MESSAGE PROCESSOR DRIVER.  PUT EQT LOCATION INTO
ØØØ5   !          LMESG, THEN SCHEDULE LSEND TO PROCESS AND SEND
ØØØ6   !          THE MESSAGE TO CENTRAL
ØØØ7   !
ØØØ8          ASSEMBLE ["EXT $LIST"]
ØØØ9   !
ØØ1Ø          LET LSEND (3) BE INTEGER  ! CONTAIN NAME OF MESSAGE TRANSMITTER MODULE
ØØ11          LET EQT1 BE INTEGER       ! EQT FWA
ØØ12          LET EQT4 BE INTEGER  ! USE TO SET TIME-OUT BIT
ØØ13          LET EQT6 BE INTEGER  ! USE FOR EQUEST CODE CHECK
ØØ14          LET EQT7 BE INTEGER  ! BUFFER ADDRESS
ØØ15          LET EQT8 BE INTEGER  ! BUFFER LENGTH
ØØ16          LET EQT15 BE INTEGER  ! TIME-OUT COUNTER
ØØ17          LET I BE INTEGER
ØØ18          LET TIOUT BE INTEGER
ØØ19          LET SSGLC BE INTEGER
ØØ2Ø   !
ØØ21          INITIALIZE LSEND TO "LSEND "
ØØ22          INITIALIZE EQT1 TO 166ØK
ØØ23          INITIALIZE EQT4 TO 1Ø1663K
ØØ24          INITIALIZE EQT6 TO 1Ø1665K
ØØ25          INITIALIZE EQT7 TO 1Ø1666K
ØØ26          INITIALIZE EQT8 TO 1Ø1667K
ØØ27          INITIALIZE EQT15 TO 1Ø1774K
ØØ28          INITIALIZE TIOUT TO -5Ø
ØØ29          INITIALIZE SSGLC TO Ø
ØØ3Ø   !
ØØ31   ! . . .    INITIALIZATION, IF MESSAGE BUFFER IS NOT BUSY,
ØØ32   !          MOVE EQT1 LOCATION TO BUFFER AND SCHEDULE LSEND.
ØØ33   !
ØØ34   IAØØ: SUBROUTINE GLOBAL, DIRECT
ØØ35          IF ($EQT6 AND 3) = 3 THEN [SSGLC _ $EQT7; $SSGLC _ 1; GO TO IA2Ø]
ØØ36          IF ($EQT6 AND 3) # 2 THEN [.A. _ 1; RETURN]
ØØ37          IF SSGLC = Ø THEN [.A. _ 4;  RETURN]
ØØ38   !
ØØ39   ! . . .    SAVE EQT LOCATION IN LOCK WORD FOR LSEND PROGRAM
ØØ4Ø   !
ØØ41          $SSGLC _ $EQT1
ØØ42          ASSEMBLE ["JSB $LIST"; \
ØØ43             "OCT 2Ø1"; \
ØØ44             "DEF LSEND"]
ØØ45   !
ØØ46   IA2Ø:
ØØ47          $EQT15 _ TIOUT
ØØ48          $EQT4 _ $EQT4 OR 1ØØØØK
ØØ49          .A. _ Ø
ØØ5Ø          RETURN
ØØ51          END IAØØ
ØØ52   !
ØØ53   !
ØØ54   ! . . .    ENTER CONTINUATION SECTION BECAUSE OF TIME OUT,
ØØ55   !          CHECK TO SEE IF UNLOCK. IF NOT CONTINUE TO WAIT
ØØ56   !
ØØ57   !
ØØ58   CAØØ: SUBROUTINE GLOBAL, DIRECT
ØØ59   !
ØØ6Ø          IF $SSGLC # Ø THEN \
ØØ61             [$EQT15 _ TIOUT; \
ØØ62             ASSEMBLE ["ISZ CAØØ"]; \
ØØ63             RETURN]
ØØ64   !
ØØ65   !
ØØ66   ! . . .    CLEAR TIME-OUT
ØØ67   !
ØØ68          $EQT15 _ Ø
ØØ69          .A. _ Ø
ØØ7Ø          RETURN
ØØ71          END CAØØ
ØØ72   END DVAØØ
```

FIG. 1. Listing of I/O driver DVA00.

```
ØØØ1
ØØØ2        NAME LSEND (17, 4Ø) "SYSTEM MESSAGE PROCESSOR"
ØØØ3  !
ØØØ4  ! . . .    FORM MESSAGE AND SEND IT TO CENTRAL
ØØØ5  !
ØØØ6        LET DMESG BE SUBROUTINE, EXTERNAL
ØØØ7        LET EXEC BE SUBROUTINE, EXTERNAL
ØØØ8        LET SSGLC  BE INTEGER, EXTERNAL
ØØØ9        LET CNODE BE INTEGER      ! CENTRAL CPU NODE NUMBER
ØØ1Ø        LET CT BE INTEGER         ! MESSAGE LENGTH
ØØ11  !
ØØ12        INITIALIZE CNODE TO 99
ØØ13  !
ØØ14  LSEND:
ØØ15  !
ØØ16  !
ØØ17        CT _ $(SSGLC + 7)
ØØ18        IF CT < -72 THEN CT _ -72
ØØ19        IF CT > 36 THEN CT _ 36
ØØ2Ø  !
ØØ21  ! . . .    TRANSMIT MESSAGE TO CENTRAL
ØØ22  !
ØØ23  L2Ø:
ØØ24        CALL DMESG (CNODE, $$(SSGLC + 6), CT)
ØØ25        GO TO L4Ø ! ERROR RETURN
ØØ26  !
ØØ27  ! . . .    CLEAR LOCK
ØØ28  !
ØØ29  L3Ø:
ØØ3Ø        SSGLC _ Ø
ØØ31  !
ØØ32        CALL EXEC (6)
ØØ33  !
ØØ34  !
ØØ35  ! . . .    SAVE CONTENT OF A AND B REGISTERS
ØØ36  !
ØØ37  !
ØØ38  L4Ø:
ØØ39        A _ .A.
ØØ4Ø        B _ .B.
ØØ41  !     GO TO L2Ø
ØØ42        GO TO L3Ø
ØØ43        END LSEND
```

FIG. 2. Listing of LSEND.

```
ØØØ1
ØØØ2        NAME LSTUQ (3Ø) "SYSTEM MESSAGE PROCESSOR SSGA"
ØØØ3  !
ØØØ4  ! . . .    LOCK WORD AND CONTAIN LOCATION OF EQT TABLE FWA
ØØØ5  !          WHEN NON-ZERO
ØØØ6  !
ØØØ7        LET SSGLC BE INTEGER, GLOBAL
ØØØ8        INITIALIZE SSGLC TO Ø
ØØØ9        END LSTUQ
```

FIG. 3. Listing of LSTUQ.

```
0001
0002        NAME STRTM (17, 1) "RTE-M START-UP PROGRAM"
0003  !
0004  ! . . .    THIS PROGRAM INITIALIZE DS/1000 BY SCHEDULING LSTEN
0005  !          AND PASSING A BUFFER WHICH CONTAINS ANSWERS FOR THE
0006  !          INITIALIZATION SEQUENCE.  AFTERWARD LSTEN AND THIS
0007  !          PROGRAM ARE KILL BY CLEARING THE ID SEGMENT
0008  !
0009        LET EXEC BE SUBROUTINE, EXTERNAL
0010        LET POPEN BE SUBROUTINE, EXTERNAL
0011        LET PREAD BE SUBROUTINE, EXTERNAL
0012        LET ICONV BE SUBROUTINE, EXTERNAL
0013  !
0014        LET BUF (80) BE INTEGER       ! BUFFER FOR ANSWER DATA FOR LSTEN
0015        LET LEN BE INTEGER            ! WORD LENGTH OF ARRAY BUF
0016        LET CLASS BE INTEGER          ! CLASS NUMBER
0017        LET LSTEN (3) BE INTEGER      ! LSTEN PROGRAM NAME
0018        LET RECLEN BE INTEGER         ! RECORD LENGTH OF EACH ANSWER IN BUF
0019        LET SSGLC BE INTEGER, EXTERNAL
0020        LET MESSS BE FUNCTION, EXTERNAL
0021        LET DMM (3) BE INTEGER
0022        LET LNODE BE INTEGER, EXTERNAL
0023        LET TAGI (20) BE INTEGER, EXTERNAL
0024        LET CPARM (3) BE INTEGER
0025        LET PCB (4) BE INTEGER
0026        LET RSYN (3) BE INTEGER
0027        LET DCOFF (144) BE INTEGER, EXTERNAL
0028        LET ALIGN (3) BE INTEGER
0029        LET CAL (3) BE INTEGER
0030        LET MESS (16) BE INTEGER
0031  !
0032        INITIALIZE BUF TO "YES     ", \ 1000 CONNECTED?
0033           "NO      ", \ 3000 CONNECTED?
0034           "/D      ", \ # OF ACTIVE TRANSACTIO
0035           "2       ", \ ENABLED LU #
0036           "/E      ", \ LAST LU
0037           "0       ", \ NDT FILE NOT BEING USED
0038           "1       ", \ LOCAL CPU #
0039           "3       ", \ NUMBER OF NODES
0040           "1       ", \ CPU #, LU, TIMEOUT
0041           "99,2,30 ", \ CPU #, LU, TIMEOUT
0042           "201,2,30", \ CPU #, LU, TIMEOUT
0043           "OPERM   ", \
0044           "/E      ", \
0045           "SY      ", \ SECURITY CODE
0046           "/T      ", \ TIME ADJUSTMENT
0047           "200     ", \
0048           "100     ", \
0049           "10      ", \
0050           "0       ", \
0051           "/E      "  ! ALL DONE
0052  !
0053        INITIALIZE LEN TO 80
0054        INITIALIZE CLASS TO 0
0055        INITIALIZE LSTEN TO "LSTEN "
0056        INITIALIZE RECLEN TO 4
0057        INITIALIZE CPARM TO "CPARM "
0058        INITIALIZE NODE TO 99
0059        INITIALIZE RSYN TO "RSYN  "
0060        INITIALIZE ALIGN TO "ALIGN "
0061        INITIALIZE CAL TO "CAL   "
0062        INITIALIZE MESS TO "DS/1000 ERROR ", 3(0), " FROM STRTM "
0063        INITIALIZE DMM TO "DMM   "
0064  !
0065  STRTM:
0066  !
0067  ! . . .    PUT NODE NUMBER INTO LSTEN BUFFER TO INITIALIZE NODE
0068  !
0069        BUF (25), BUF (33) _ LNODE + 30060K
```

FIG. 4. Example of RTE-MII startup program.

```
!
! . . .    INITIALIZE DVA00 BY GIVING IT LOCATION OF SSGA VARIABLE SSGLC
!
      CALL EXEC (3, 1, @SSGLC)
!
! . . .    PUT DATA INTO CLASS VIA WRITE/READ
!
      CALL EXEC (20, 0, BUF, LEN, 0, 0, CLASS)
!
! . . .    SCHEDULE LSTEN WITH THREE PARAMETERS TO PERFORM CLASS GET
!
      CALL EXEC (9, LSTEN, -1, CLASS, RECLEN)
!
!
      SSGLC _ 0 ! DS/1000 IS INITIALIZE, RELEASE LOCK
!
!
! . . .    GET STARTUP PARAMETERS FROM CENTRAL
!
      TAGI (1) _ LNODE
      CALL POPEN (PCB, IERR, CPARM, NODE, TAGI)
      IF IERR # 0 THEN GO TO ERROR
!
! . . .    GET CALIBRATION FACTORS
!
      TAGI (1) _ LNODE
      CALL PREAD (PCB, IERR, DCOFF, 96, TAGI)
      IF IERR # 0 THEN GO TO ERROR
!
! . . .    NEED TO PERFORM ALIGNMENT?
!
      IF TAGI (1) = 0 THEN GO TO L2000
!
L1000:
      CALL EXEC (9, ALIGN)
!
! . . .    NEED TO PERFORM CALIBRATION?
!
L2000:
      IF TAGI (2) = 0 THEN GO TO L3000
!
      CALL EXEC (9, CAL, 60)
!
! . . .    TIME SYN
!
L3000:
      CALL EXEC (9, RSYN)
!
! . . .    START DMM
!
      CALL EXEC (10, DMM)
!
      CALL EXEC (6)
!
! . . .    ERROR
!
ERROR:
      CALL ICONV (IERR, MESS (8))
      CALL EXEC (2, 1, MESS, 16)
      TAGI (1), TAGI (2) _ 0
      GO TO L1000
      END STRTM
```

FIG. 4. (continued)

```
GEN
OUTPUT ON P01000
MAP MODULES,GLOBALS ON "R0100
END
*
* TYPE OF SYSTEM?
2
*
* TBG CHNL?
15
*
* PRIV. INT?
12
*
* LWAM?
77777
*
* FWA BP?
21
*
*
* CHANGE ENTS?
.MPY,RP,100200
.DIV,RP,100400
.DLD,RP,104200
.DST,RP,104400
.FAD,RP,105000
.FSB,RP,105020
.FMP,RP,105040
.FDV,RP,105060
IFIX,RP,105100
FLOAT,RP,105120
.MBT,RP,105765
.MVW,RP,105777
*
* . . .    SELECT CODE FOR DVR62
*
ADSCC,AB,14
ADSCD,AB,13
END
*
* REL SYS MODS
LINKS IN CURRENT
REL %MSY2R::11
REL %MSC2::11
REL %MEX2::11
REL %MDI2::11
REL %MIO2::11
REL %MBU::11
REL %MTI::11
REL %MTS::11
*
*
REL %MOP::11
REL %MCL::11
REL %MRN::11
REL %MDMI::11
REL %MER::11
REL %DVA00::11
REL %DVA65::11
REL %DVA62::11
END
*
```

FIG. 5. Answer file example for generating RTE-MII system with remote system console.

```
0064  * # OF I/O CLASSES?
0065  32
0066  *
0067  * # OF RESOURCE NUMBERS?
0068  32
0069  *
0070  * BUFFER LIMITS (LOW,HIGH)?
0071  128,256
0072  *
0073  *
0074  * EQT TBL
0075  *
0076  *
0077  * . . .    EQT 1 - COMMUNICATION BOARD
0078  *
0079  10,DVA65,X=7,T=16
0080  *
0081  * . . .    EQT 2 - SYSTEM MESSAGE PROCESSOR
0082  *
0083  25,DVA00,B
0084  *
0085  *
0086  *
0087  * . . .    EQT 3 - UI CARD FOR ADC, ETC.
0088  *
0089  13,DVA62,T=500
0090  *
0091  END
0092  *
0093  * DRT TBL
0094  *
0095  * . . .    LU 1 - SYSTEM MESSAGE PROCESSOR
0096  *
0097  2,0
0098  *
0099  * . . .    LU 2 - COMMUNICATION CARD
0100  *
0101  1,0
0102  *
0103  * . . .    LU 3 - UI CARD FOR ADC, ETC.
0104  *
0105  3,0
0106  *
0107  END
0108  *
0109  * INT TBL
0110  *
0111  *
0112  *
0113  * . . .    COMMUNICATION CARD
0114  *
0115  10,PRG,QUEUE
0116  *
0117  * . . .    UI ADC DATA CHANNEL
0118  *
0119  13,EQT,3
0120  *
0121  * . . .    UI ADC COMMAND CHANNEL
0122  *
0123  14,EQT,3
0124  *
0125  *
0126  END
0127  *
0128  *
```

FIG. 5. (continued)

```
Ø129  * #ID SEG?
Ø13Ø  2Ø
Ø131  *
Ø132  * START-UP PROG?
Ø133  *
Ø134  STRTM
Ø135  *
Ø136  * REL RES LIB
Ø137  REL %RLIB#(.ENTR)::11
Ø138  REL %MALRN::11
Ø139  END
Ø14Ø  *
Ø141  * REL SSGA
Ø142  REL %DSLB1(RES)::11
Ø143  REL %DSLB1(#REQU)::11
Ø144  REL %DSLB1(DRTEQ)::11
Ø145  REL %DSLB1(PGMAD)::11
Ø146  REL %LSTUQ::11
Ø147  REL %RSSGA::11
Ø148  REL %LNOD1::11
Ø149  END
Ø15Ø  *
Ø151  * # WDS IN COMM?
Ø152  Ø
Ø153  *
Ø154  * ALIGN AT NEXT PAGE?
Ø155  YES
Ø156  *
Ø157  *
Ø158  *
Ø159  * . . .    DS/1ØØØ PROGRAM QUEUE
Ø16Ø  *
Ø161  REL %QUEUE::11
Ø162  REL %MRNRQ::11
Ø163  END
Ø164  *
Ø165  Ø
Ø166  *
Ø167  * . . .    DS/1ØØØ PROGRAM GRPM
Ø168  *
Ø169  REL %GRPM::11
Ø17Ø  REL %MRNRQ::11
Ø171  END
Ø172  *
Ø173  Ø
Ø174  *
Ø175  * . . .    DS/1ØØØ PROGRAM RTRY
Ø176  *
Ø177  REL %RTRY::11
Ø178  END
Ø179  *
Ø18Ø  Ø
Ø181  *
Ø182  * . . .    DS/1ØØØ PROGRAM QCLM
Ø183  *
Ø184  REL %QCLM::11
Ø185  SEARCH %MSYLB::11
Ø186  SEARCH %RLIB#::11
Ø187  END
Ø188  *
Ø189  Ø
Ø19Ø  *
Ø191  * . . .    DS/1ØØØ PROGRAM UPLIN
Ø192  *
Ø193  REL %UPLIN::11
```

FIG. 5. (continued)

```
Ø194  REL %MRNRQ::11
Ø195  REL %$MMES::11
Ø196  END
Ø197  *
Ø198  Ø
Ø199  *
Ø2ØØ  *
Ø2Ø1  *
Ø2Ø2  * . . .    SYSTEM MESSAGE PROCESSOR MODULE
Ø2Ø3  *
Ø2Ø4  REL %LSEND::11
Ø2Ø5  REL %MRNRQ::11
Ø2Ø6  REL %MDRCT::11
Ø2Ø7  REL %MREIO::11
Ø2Ø8  REL %MCNMO::11
Ø2Ø9  SEARCH %DSLB2::11
Ø21Ø  SEARCH %DSLB1::11
Ø211  SEARCH %RLIB#::11
Ø212  END
Ø213  *
Ø214  Ø
Ø215  *
Ø216  *
Ø217  * . . .    A P P L I C A T I O N   P R O G R A M S   S T A R T
Ø218  *
Ø219  *                                H E R E
Ø22Ø  *
Ø221  *
Ø222  *
Ø223  *
Ø224  * . . .    DS/1ØØØ OPERATOR REQUEST MONITOR MODULE
Ø225  *
Ø226  REL %OPERM::11
Ø227  REL %$MMES::11
Ø228  SEARCH %DSLB1::11
Ø229  END
Ø23Ø  *
Ø231  Ø
Ø232  *
Ø233  *
Ø234  *
Ø235  *
Ø236  *
Ø237  * . . .    F R E E   P E R I O D   I N I T I A T O R
Ø238  *
Ø239  REL %PERID::11
Ø24Ø  END
Ø241  *
Ø242  Ø
Ø243  *
Ø244  * . . .    S T R G 2
Ø245  *
Ø246  REL %STR62::11
Ø247  REL %PACKT::11
Ø248  SEARCH %MSYLB::11
Ø249  SEARCH %RLIB#::11
Ø25Ø  END
Ø251  *
Ø252  Ø
Ø253  *
Ø254  *
Ø255  * . . .    R C E M
Ø256  *
Ø257  REL %RCEM::11
Ø258  REL %ICONV::11
```

FIG. 5. (continued)

```
SEARCH %DSLB3::11
SEARCH %DSLB2::11
SEARCH %DSLB1::11
SEARCH %MSYLB::11
SEARCH %RLIB#::11
END
*
Ø
*
*
* . . .    D M M
*
REL %DMM::11
SEARCH %MSYLB::11
REL %DMLWA::11
END
*
Ø
*
* . . .    RTEM START UP PROGRAM
*
REL %STRTM::11
REL %ICONV::11
SEARCH %DSLB3::11
SEARCH %DSLB2::11
SEARCH %DSLB1::11
SEARCH %MSYLB::11
SEARCH %RLIB#::11
END
*
Ø
*
* . . .    TIME SYNC
*
REL %RSYN::11
REL %ICONV::11
SEARCH %DSLB3::11
SEARCH %DSLB2::11
SEARCH %DSLB1::11
SEARCH %MSYLB::11
SEARCH %RLIB#::11
END
*
Ø
*
*
* . . .    DC OFFSET CALCULATOR
*
REL %DCCAL::11
END
*
Ø
*
*
* . . .    CALIBRATION
*
REL %CAL::11
REL %CAL1::11
REL %CALFN::11
REL %CALDI::11
REL %STORA::11
REL %CDIGF::11
REL %FP3::11
REL %CALDS::11
REL %ICONV::11
```

FIG. 5. (continued)

```
Ø324  REL %SQUZE::11
Ø325  SEARCH %DSLB3::11
Ø326  SEARCH %DSLB2::11
Ø327  SEARCH %DSLB1::11
Ø328  SEARCH %MSYLB::11
Ø329  SEARCH %RLIB#::11
Ø33Ø  REL %CALBU::11
Ø331  END
Ø332  *
Ø333  Ø
Ø334  *
Ø335  * . . .     ALIGNMENT
Ø336  *
Ø337  REL %ALIGN::11
Ø338  END
Ø339  *
Ø34Ø  Ø
Ø341  *
Ø342  *
Ø343  *
Ø344  *
Ø345  * . . .     DS/1ØØØ PROGRAM LSTEN TO INITIALIZE DS/1ØØØ
Ø346  *
Ø347  REL %LSTEN::11
Ø348  SEARCH %DSLB3::11
Ø349  REL %READF::11
Ø35Ø  REL %RW$UB::11
Ø351  REL %RWND$::11
Ø352  REL %PPASF::11
Ø353  SEARCH %MSYLB::11
Ø354  SEARCH %RLIB#::11
Ø355  END
Ø356  *
Ø357  Ø
Ø358  *
Ø359  END
Ø36Ø  *
Ø361  * . . .     SET LWAM
Ø362  *
Ø363  7Ø777
Ø364  *
Ø365  * ALIGN AT NEXT PAGE?
Ø366  YES
Ø367  *
Ø368  * SNAPSHOT?
Ø369  SNAP ON &RØ1ØØ
Ø37Ø  END
Ø371  *
```

FIG. 5. (continued)

REFERENCES

1. Hewlett Packard, RTE-M Programmer's Reference Manual, Hewlett Packard, Cupertino, Calif. (1977).

2. Hewlett Packard, RTE-M System Generation Reference Manual, Hewlett Packard, Cupertino, Calif. (1977).

3. Hewlett Packard, DS/1000 Programmer's Reference Manual, Part No. 91740-90002, Hewlett Parkard, Cupertino, Calif. (1977).

4. Hewlett Packard, DS/1000 Netword Manager's Manual--A Guide to Installing, Configuring and Using Your DS/1000 System, Hewlett Packard, Cupertino, Calif. (1977).

WTF