

**RS03/04**

DEVICE ROUTINE (MPG)  
MD-11-DTRSA-A

EP-DTRSA-A-DL-A  
COPYRIGHT © 1976  
FICHE 1 OF 1

NOV 1976  
**digital**  
MADE IN U.S.A.



.REM %

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DTRSA-A  
 PRODUCT NAME: RH11/RH70 - RS03/RS04 DEVICE  
 ROUTINE FOR MPG  
 DATE: JULY 1976  
 MAINTAINED BY: DIAGNOSTIC ENGINEERING  
 AUTHOR: A. W. LEIGH

COPYRIGHT (C) 1976  
 DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN DEC.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

DEC ASSUMES NO RESPONSIBILITY FOR USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

%

MAINDEC-11-DTRSA-A  
DTRSA.A.P11  
MACY11 27(732) 24-SEP-76 14:12 PAGE 1  
SEQ 0065

CO1

MAINDEC-11-DTRSA-A RH11/RH70 - RSD3/RSD4 DEVICE ROUTINE FOR MPG  
DTRSA.A.P11 REVISION HISTORY

MACY11 27(732) 24-SEP-76 14:12 PAGE 2

SEQ 0066

11-00-10-01

.SBTTL REVISION HISTORY

⋮

JUL 76 DTRSA-A INITIAL RELEASE AS A FULL SUPPORT  
DEVICE ROUTINE FOR THE RSD3/RSD4 DISK.



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106

.SBTTL STANDARD DEVICE ROUTINE TABLE  
.TITLE MAINDEC-11-DTRSA-A RH11/RH70 - RS03/RS04 DEVICE ROUTINE FOR MPG

;REVISION 'A'

;FILENAME OF "TRSA0.MPG" ON MPG/XXDP MEDIA

;MACY11: DTRSA?,DTRSA?/CRF:SYM/DOC=DTRSA?.P11

;LNKX11: DTRSA?.MPG/B:0=DTRSA?/E

;PAPER TAPE: PUNCH DTRSA?.MPG/FILE:ELEV

000000'

.CSECT RJP11  
.DSABL GBL

;PROGRAM EQUALS

↑TIME=30000.

NRTRK=64.

NRSEC=64.

PS=177776

REGNUM=14.

STSLUP=1011

XXXX=0

;TIME COUNT FOR INTERRUPTS

;NR OF TRACKS

;NR OF SECTORS

;PSW

;NR OF DEVICE REGISTERS

;LOOP CONTROL FOR "STSTAT" ROUTINE

;VALUE TO BE TAYLORED BY DEVICE ROUTINE

072460  
000100  
000100  
177776  
000016  
001011  
000000

;DEVICE REGISTGER NAMES

RPCS1=0

RPWC=2

RPBA=4

RPDA=6

RPCS2=10

RPDS=12

RPER=14

RPER1=RPER

RPAS=16

RPLA=20

RPDB=22

RPMR=24

RPDT=26

RPBAE=30

RPCS3=32

000000  
000002  
000004  
000006  
000010  
000012  
000014  
000014  
000016  
000020  
000022  
000024  
000026  
000030  
000032

;RSCS1 DEVICE BIT EQUATES

↑RE=40000

MCPE=20000

DVA=4000

PSEL=2000

PORT=PSEL

IE=100

GO=1

040000  
020000  
004000  
002000  
002000  
000100  
000001

;RSCS2 DEVICE BIT EQUATES

NED=10000

CLR=40

PARITY=20

BAI=10

010000  
000040  
000020  
000010



107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135

100000  
040000  
010000  
000400  
  
040000  
  
000001  
000002  
000010  
  
000010  
000030  
000050  
000060  
000070  
  
100000  
002000  
000200  
000100  
000002  
000001

:RSDS DEVICE BIT EQUATES  
ATA=100000  
ERR=40000  
MOL=10000  
DPR=400  
  
:RSER DEVICE BIT EQUATES  
UNS=40000  
  
:SYSTEM FLAG-WORD BIT DEFINITIONS  
MMVER=1  
USMTPS=2  
CPU70=10  
  
:DISK COMMAND CODES  
DCCODE=10  
SCODE=30  
WCCODE=50  
WCODE=60  
RCODE=70  
  
:DEVICE ROUTINE (DFLGWD) FLAG BITS  
WAITMD=100000  
CORFLG=2000  
ANYIOI=200  
CMDISU=100  
DOTERM=2  
IOERR=1

:WAIT MODE, 0=WAIT  
:CORRECTION MODE, 0=CORON  
:ANY I/O HAS BEEN ISSUED  
:I/O COMMAND HAS BEEN ISSUED  
:PROCESS I/O TERMINATION  
:ERROR ON CURRENT I/O



137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181

000000' 010132  
000002' 000000  
000004' 000000  
000006' 000000  
000006' 000006  
000010' 000000  
000012' 000003  
000014' 000000  
000016' 000000  
000020' 000000  
000022' 000000  
000024' 172040  
000026' 000204  
000030' 000240  
000032' 000000  
000034' 001300  
000036' 001350  
000040' 001764  
000042' 001154  
000044' 001700  
000046' 000000  
000050' 000000  
000052' 000000  
000054' 000000  
000056' 000000  
000060' 000000  
000062' 000000  
000064' 000000  
000066' 000000  
000070' 000000  
000072' 000000  
000074' 000000  
000076' 000000  
000100' 000000  
000102' 000014  
000104' 000136  
000106' 000246  
000110' 000444  
000112' 000542  
000114' 000716

LOCZ: .WORD DVREND-  
DFLGWD: .WORD 0  
HEAD: .WORD 0  
TRAK=HEAD  
SECT: .WORD 0  
RTRY: .WORD 3  
SIZE: .WORD 0  
ERRI: .WORD 0  
DREGAD: .WORD 172040  
IVCTAD: .WORD 204  
PSWD: .WORD 240  
CIOSY: .WORD 0  
CUPGER: .WORD 0  
ULIST: .WORD 0  
CLIST: .WORD 0  
BINASC: .WORD 0  
BTASLZ: .WORD 0  
DECASC: .WORD 0  
CSYSFW: .WORD 0  
SETVEC: .WORD 0  
CLRVEC: .WORD 0  
TSTVEC: .WORD 0  
RTNINT: .WORD 0  
GETBYT: .WORD 0  
PUTBYT: .WORD 0  
DVREGS-  
DVCMD5-  
DVPKTE-  
DVMVTE-  
DVCPT-  
DVIWST-

: THE FOLLOWING TABLE IS IN THE STANDARDIZED FORMAT REQUIRED  
: TO INTERFACE WITH MPG.

: DEVICE ROUT SIZE IN BYTES  
: DEVICE ROUT FLAGWORD  
: NOT USED  
: HEAD # (0 THRU 18.)  
: SECTOR # (0 THRU 21./19.)  
: # OF RETRY ATTEMPTS  
: INTERFACE WORD # 5 (NOT USED)  
: INTERFACE WORD # 6 (NOT USED)  
: # OF BYTES TRANSFERRED / UNIMAP FLG  
: ERROR ON LAST I/O INDICATOR  
: FIRST DEVICE REGISTER ADR  
: INTERRUPT VECTOR ADR  
: INT PROC STATUS WORD (BR 5)  
: NOT USED  
: HOUSEKEEPING ROUT REL ADR  
: REPORT ROUT REL ADR  
: KILL ROUT REL ADR  
: DATA ERROR COUNTER REL ADR  
: TIME OUT ERROR ROUT REL ADR  
: I/O BUSY BRANCH ADR  
: DEVICE ERROR BRANCH ADR  
: USER MODE PRINT ROUTINE BRANCH ADR  
: CMND MODE PRINT ROUTINE BRANCH ADR  
: CONVERT BINARY TO ASCII ROUT BR ADR  
: CONVERT BINARY TO DECIMAL ASCII BR ADR  
: CONVERT PACKED DECIMAL TO ASCII BR ADR  
: MPG SYSTEM FLAGWORD ADR  
: SET INT VECT ROUT BR ADR  
: CLEAR INT VECTOR ROUT BR ADR  
: TEST INT VECTOR ROUT BR ADR  
: RETURN FROM INT ROUT BR ADR  
: GET DATA BYTE ROUT BR ADR  
: PUT DATA BYTE ROUT BR ADR  
: ADR OF DEVICE REGISTER NAMES  
: ADR OF DEVICE FUNCTIONS  
: ADR OF PACK TBL EXTENSION  
: ADR OF MODEL VECTOR TBL EXTEN.  
: ADR OF COMPILER TBL EXTEN.  
: ADR OF DEV INTERFACE WD SYM TBL



183 .SBTTL COMPILER TABLES & CONSTANT AREAS

```

184
185
186 000116' 051522 030503
187 000122' 000000
188 000124' 051522 041527
189 000130' 000002
190 000132' 051522 040502
191 000136' 000004
192 000140' 051522 040504
193 000144' 000006
194 000146' 051522 031103
195 000152' 000010
196 000154' 051522 051504
197 000160' 000012
198 000162' 051522 051105
199 000166' 000014
200 000170' 051522 051501
201 000174' 000016
202 000176' 051522 040514
203 000202' 000020
204 000204' 051522 041104
205 000210' 000022
206 000212' 051522 051115
207 000216' 000024
208 000220' 051522 052104
209 000224' 000026
210 000226' 051522 042501
211 000232' 000030
212 000234' 051522 031503
213 000240' 000032
214      000242'
215
216
217 000242'      120      211
218 000244' 002660
219 000246'      130      211
220 000250' 002704
221 000252'      376      000
222 000254' 001736
223 000256'      375      000
224 000260' 001712
225 000262'      374      000
226 000264' 001122
227 000266'      373      000
228 000270' 001116
229 000272'      370      211
230 000274' 002710
231 000276'      365      000
232 000300' 002002
233 000302'      364      000
234 000304' 002046
235 000306'      363      000
236 000310' 002724
237 000312'      362      000
238 000314' 001542

```

```

DVREGS: .ASCII /RSC1/
        .WORD 0
        .ASCII /RSWC/
        .WORD 2
        .ASCII /RSBA/
        .WORD 4
        .ASCII /RSDA/
        .WORD 6
        .ASCII /RSC2/
        .WORD 10
        .ASCII /RSDS/
        .WORD 12
        .ASCII /RSER/
        .WORD 14
        .ASCII /RSAS/
        .WORD 16
        .ASCII /RSLA/
        .WORD 20
        .ASCII /RSDB/
        .WORD 22
        .ASCII /RSMR/
        .WORD 24
        .ASCII /RSDT/
        .WORD 26
        .ASCII /RSAE/
        .WORD 30
        .ASCII /RSC3/
        .WORD 32

```

DVREGGE= .

```

DVCMDs: .BYTE 120,211
        .WORD READ-
        .BYTE 130,211
        .WORD WRITE-
        .BYTE 376,0
        .WORD NOWAIT-
        .BYTE 375,0
        .WORD WAIT-
        .BYTE 374,0
        .WORD REPORT-
        .BYTE 373,0
        .WORD REPORT-
        .BYTE 370,211
        .WORD WRCK-
        .BYTE 365,0
        .WORD CRESET-
        .BYTE 364,0
        .WORD DRESET-
        .BYTE 363,0
        .WORD SEARCH-
        .BYTE 362,0
        .WORD STEPUP-

```

```

:VALID DEVICE REGISTER NAMES &
:THEIR POSITIONS RELATIVE TO
:THE DEVICE REGISTERS BASE ADDRESS.

```

```

:VALID DEVICE FUNCTIONS
:FLAG BYTE:
:BIT 7 = NPR DEV
:BIT 3 = MASSBUS DEV
:BIT 0 = 2 WORDS FOR ADR
:(18 BIT ADRS)

```



239	000316'	361	000			.BYTE	361,0
240	000320'	001622				.WORD	STEPDN-
241	000322'	351	000			.BYTE	351,0
242	000324'	001676				.WORD	APORT-
243	000326'	350	000			.BYTE	350,0
244	000330'	001702				.WORD	BPORT-
245	000332'	345	000			.BYTE	345,0
246	000334'	001706				.WORD	ODD-
247	000336'	344	000			.BYTE	344,0
248	000340'	001712				.WORD	EVEN-
249	000342'	337	000			.BYTE	337,0
250	000344'	001716				.WORD	BAION-
251	000346'	336	000			.BYTE	336,0
252	000350'	001722				.WORD	BAIOFF-
253	000352'	177777				.WORD	177777
254							
255	000354'	047516	040527	052111	DVPKTE:	.ASCII	/NOWAIT/
256	000362'	376	000			.BYTE	376,0
257	000364'	020040	040527	052111		.ASCII	/WAIT/
258	000372'	375	000			.BYTE	375,0
259	000374'	052123	052101	051525		.ASCII	/STATUS/
260	000402'	374	000			.BYTE	374,0
261	000404'	047503	047525	051524		.ASCII	/COUNTS/
262	000412'	373	000			.BYTE	373,0
263	000414'	020040	051112	045503		.ASCII	/WRCK/
264	000422'	370	000			.BYTE	370,0
265	000424'	051103	051515	052105		.ASCII	/CRESET/
266	000432'	365	000			.BYTE	365,0
267	000434'	051103	051515	052105		.ASCII	/DRESET/
268	000442'	364	000			.BYTE	364,0
269	000444'	042523	051112	044103		.ASCII	/SEARCH/
270	000452'	363	000			.BYTE	363,0
271	000454'	052123	050106	050125		.ASCII	/STEPUP/
272	000462'	362	000			.BYTE	362,0
273	000464'	052123	050105	047104		.ASCII	/STEPDN/
274	000472'	361	000			.BYTE	361,0
275	000474'	040440	047520	052122		.ASCII	/APORT/
276	000502'	351	000			.BYTE	351,0
277	000504'	041040	047520	052122		.ASCII	/BPORT/
278	000512'	350	000			.BYTE	350,0
279	000514'	020040	047440	042104		.ASCII	/ODD/
280	000522'	345	000			.BYTE	345,0
281	000524'	020040	053105	047105		.ASCII	/EVEN/
282	000532'	344	000			.BYTE	344,0
283	000534'	041040	044501	047117		.ASCII	/BAION/
284	000542'	337	000			.BYTE	337,0
285	000544'	040502	047511	043106		.ASCII	/BAIOFF/
286	000552'	336	000			.BYTE	336,0
287							
288	000554'	000376	001064		DVMVTE:	.WORD	376,MSFMT1-LOCZ
289	000560'	000375	001064			.WORD	375,MSFMT1-LOCZ
290	000564'	000374	001064			.WORD	374,MSFMT1-LOCZ
291	000570'	000373	001064			.WORD	373,MSFMT1-LOCZ
292	000574'	000370	001065			.WORD	370,MSFMT2-LOCZ
293	000600'	000365	001064			.WORD	365,MSFMT1-LOCZ
294	000604'	000364	001064			.WORD	364,MSFMT1-LOCZ

;TABLE TERMINATOR

;PACK TABLE EXTENSION

;MODEL VECTOR TABLE EXTEN.



295	000610'	000363	001064	.WORD	363,MSFMT1-LOCZ
296	000614'	000362	001072	.WORD	362,MSFMT5-LOCZ
297	000620'	000361	001072	.WORD	361,MSFMT5-LOCZ
298	000624'	000351	001064	.WORD	351,MSFMT1-LOCZ
299	000630'	000350	001064	.WORD	350,MSFMT1-LOCZ
300	000634'	000345	001064	.WORD	345,MSFMT1-LOCZ
301	000640'	000344	001064	.WORD	344,MSFMT1-LOCZ
302	000644'	000337	001064	.WORD	337,MSFMT1-LOCZ
303	000650'	000336	001064	.WORD	336,MSFMT1-LOCZ

...  
DVCPT: COMPILER TABLE EXTENSION

308	000654'	003	376	.BYTE	3,376	;NO WAIT
309	000656'	004537	000012	.WORD	4537,10.	
310	000662'	003	375	.BYTE	3,375	;WAIT
311	000664'	004537	000012	.WORD	4537,10.	
312	000670'	004	374	.BYTE	4,374	;STATUS
313	000672'	004537	000012	.WORD	4537,10.,1002	
314	000700'	004	373	.BYTE	4,373	;COUNTS
315	000702'	004537	000012	.WORD	4537,10.,1001	
316	000710'	006	370	.BYTE	6,370	;WRITE CHECK DATA
317	000712'	004537	000012	.WORD	4537,10.,0,2,2	
	000720'	000002	000002			
318	000724'	003	365	.BYTE	3,365	;CONTROL RESET
319	000726'	004537	000012	.WORD	4537,10.	
320	000732'	003	364	.BYTE	3,364	;DRIVE RESET
321	000734'	004537	000012	.WORD	4537,10.	
322	000740'	003	363	.BYTE	3,363	;SEARCH
323	000742'	004537	000012	.WORD	4537,10.	
324	000746'	004	362	.BYTE	4,362	;STEP UP
325	000750'	004537	000012	.WORD	4537,10.,0	
326	000756'	004	361	.BYTE	4,361	;STEP DOWN
327	000760'	004537	000012	.WORD	4537,10.,0	
328	000766'	003	351	.BYTE	3,351	;A PORT
329	000770'	004537	000012	.WORD	4537,10.	
330	000774'	003	350	.BYTE	3,350	;B PORT
331	000776'	004537	000012	.WORD	4537,10.	
332	001002'	003	345	.BYTE	3,345	;ODD
333	001004'	004537	000012	.WORD	4537,10.	
334	001010'	003	344	.BYTE	3,344	;EVEN
335	001012'	004537	000012	.WORD	4537,10.	
336	001016'	003	337	.BYTE	3,337	;BAI ON
337	001020'	004537	000012	.WORD	4537,10.	
338	001024'	003	336	.BYTE	3,336	;BAI OFF
339	001026'	004537	000012	.WORD	4537,10.	

...  
DVIWST: DEVICE INTERFACE WORD SYMBOL TABLE

344	001032'	051124	045501	.ASCII	/TRAK/
345	001036'	000006		.WORD	DEVIW2
346	001040'	042510	042101	.ASCII	/HEAD/
347	001044'	000006		.WORD	DEVIW2
348	001046'	042523	052103	.ASCII	/SECT/
349	001052'	000010		.WORD	DEVIW3

```

350 001054' 052122 054522 .ASCII /RTRY/
351 001060' 000012 .WORD DEVIW4
352 001062' 177777 .WORD 177777 ;END OF TABLE
353
354
355 ;
356 ; MODEL STATEMENT TABLE EXTENSION
357 001064' 000 MSFMT1: .BYTE 0
358 001065' 377 052101 000377 MSFMT2: .ASCIZ <377>/AT/<377>
359 001072' 377 000 MSFMT5: .BYTE 377,0
360 .EVEN
361
362
363 ;DEVICE ROUTINE CONSTANTS & EQUATES
364
365
366 001074' HSKPST= .
367 001074' ISTAT= . ;STORAGE FOR DEV REG'S AT INT
368 001074' 000000 000000 000000 .WORD 0,0,0,0,0,0,0,0
001102' 000000 000000 000000
001110' 000000 000000 000000
369 001114' 000000 000000 000000 .WORD 0,0,0,0,0,0
001122' 000000 000000 000000
370
371 001130' 000016 CSTAT: .BLKW 14. ;DEV REG CURRENT VALUES STORAGE
372 001164' COUNTS:
373 001164' 000000 BYRD: .WORD 0 ;BYTES READ COUNT
374 001166' 000000 .WORD 0
375 001170' 000000 BYWR: .WORD 0 ;BYTES WRITTEN COUNT
376 001172' 000000 .WORD 0
377 001174' 000000 BYCK: .WORD 0 ;BYTES CHECKED COUNT
378 001176' 000000 .WORD 0
379 001200' 000000 RDCNT: .WORD 0 ;READ CMND COUNT
380 001202' 000000 WRCNT: .WORD 0 ;WRITE CMND COUNT
381 001204' 000000 CKCNT: .WORD 0 ;CHECK CMND COUNT
382 001206' 000000 SKCNT: .WORD 0 ;SEARCH CMND COUNT
383 001210' 000000 CTLCNT: .WORD 0 ;CONTROLLER CLEAR COUNT
384 001212' 000000 DRVCNT: .WORD 0 ;DRIVE CLEAR CMND COUNT
385 001214' 000000 ERRCNT: .WORD 0 ;DEVICE ERRORS COUNT
386 001216' 000000 DATAER: .WORD 0 ;DATA/OPERATOR ERRORS COUNT
387 001220' 000000 DLT CNT: .WORD 0 ;DATA LATE ERRORS
388 001222' 000000 DTECNT: .WORD 0 ;DRIVE TIMING ERRORS
389 001224' 000000 DCKCNT: .WORD 0 ;DATA CHECK ERRORS
390 001226' 000000 WCECNT: .WORD 0 ;WRITE CHECK ERRORS
391 001230' 000000 RETRYS: .WORD 0 ;# OF RETRIES ON I/O CMNDS
392 001232' 000000 INTCNT: .WORD 0 ;INTERRUPTS COUNT
393 001234' CNTEND=.
394 000024 CNTNUM=CNTEEND-COUNTS/2 ;SIZE OF COUNT TABLE
395
396 001234' 000000 ERRADR: .WORD 0 ;CURR ADR IN USER PROG
397 001236' 000000 CNTADR: .WORD 0 ;ADR OF BYTE COUNT TOTALS
398 001240' 000000 CURFLG: .WORD 0 ;FLAG WORD OF CURR CMND
399 001242' 000000 CURCMD: .WORD 0 ;CURR CMND CODE
400 001244' 000000 CURADR: .WORD 0 ;CURR BUS ADDRESS
401 001246' 000000 .WORD 0
402 001250' 000000 CURCNT: .WORD 0 ;NEG WORD CNT FOR CURR CMND

```



K01

403	001252'	000000	FINCNT: .WORD	0	:FINAL WORD CNT (RPWC)
404	001254'	000000	CURRY: .WORD	0	:CURR RETRY COUNT
405	001256'	000000	RTRYIP: .WORD	0	:RETRY IN PROGRESS FLAG
406		001260'	HSKPEN= .		
407					
408	001260'	000000	RPCS1V: .WORD	0	:BASE VALUE FOR RSCS1 REG
409					
410	001262'	000000	RPCS2V: .WORD	0	:BASE VALUE FOR, RPCS2 REG
411					
412					
413					
414					
415					
416	001264'		PATCH: .REPT	20..	:PATCH AREA
417			.WORD	0	
418			.ENDR		

420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475

.SBTTL RS03/RS04 SUPPORT ROUTINES ENTERED FROM MPG

;DEVICE ROUTINE HOUSEKEEPING

```

;JSR   R5,HSKEEP           S/R CALL
;.WORD 0 OR 1              0 = DO HSKP PER OPSW
;                                           1 = UNCOND. DO HSKP
;R2 = PROG'S OPSW
;DESTROYS R0,R1

;INIT # OF RETRY ATTEMPTS
;INITIALIZE RPCS1 VALUE
;INITIALIZE RPCS2 VALUE
;UNCONDITIONALLY DO HSKP?
;N,Y-10$
;OPSW SPECIFY EACH PASS HSKP?
;Y,N-30$
;SET UP FIRST WD ADR

;SET UP # OF WORDS
;HSKP ALL NECESSARY AREAS

;EXIT IN-LINE
    
```

```

176450 HSKEEP: MOV #3,RTRY
CLR RPCS1V
CLR RPCS2V
TST (R5)+
BNE 10$
BIT #HSKPEP,R2
BNE 30$
10$: MOV PC,R0
ADD #HSKPST-.,R0
MOV #HSKPEN-HSKPST/2,R1
20$: CLR (R0)+
DEC R1
BNE 20$
30$: RTS R5
    
```

;RS03/RS04 REPORT ROUTINE

```

;JSR   R5,REPORT          S/R CALL
;.WORD FLGWD              FLAGWORD
;                                           BIT 15 = CMND MODE CALL
;                                           BIT 9 = PROG STMT CALL
;                                           BIT 1 = DO STATUS REPORT
;                                           BIT 0 = DO COUNTS REPORT
    
```

```

004272 REPORT: JSR R0,SAVREG
004320 JSR PC,SUPTAD
011504 MOV (R5),R4
032704 BIT #2,R4
001403 BEQ 5$
004567 JSR R5,STSTAT
177476 .WORD CSTAT-
032704 5$: BIT #177776,R4
001012 BNE 15$
010700 MOV PC,R0
062700 ADD #COUNTS-.,R0
012701 MOV #CNTNUM,R1
005720 10$: TST (R0)+
001003 BNE 15$
005301 DEC R1
001374 BNE 10$
000477 BR DVREX
004767 15$: JSR PC,DEVID
032704 BIT #2,R4
001432 BEQ DISCNT
    
```

```

;SAVE REG'S R0 - R5
;SET UP PROG TBL ADR IN R3
;GET FLAGWORD
;GOING TO DO STATUS DISPLAY?
;Y,N-5$
;GO STORE STATUS REG'S

;DISPLAYING CNTS AT END OF
;PROG PASS? (Y,N-15$)
;SET UP ADR OF CNTS

;GET # OF CNT WORDS
;THIS CNT WORD = 0?
;Y,N-15$
;DECR WORD CNT
;CK'ED ALL WORDS? (Y,N-10$)
;GO TO EXIT -- ALL CNTS ARE 0'S
;DISPLAY DEVICE I.D.
;DOING STATUS DISPLAY?
;Y,N-DISCNT
    
```



476	001500'	010700		MOV	PC,R0		;SET UP ADR OF REG'S AT
477	001502'	062700	177372	ADD	#ISTAT-. ,R0		;LAST INT
478	001506'	012701	000016	MOV	#REGNUM,R1		;SET UP # OF REG'S
479	001512'	005720		20\$: TST	(R0)+		;ALL REG'S = 0?
480	001514'	001003		BNE	30\$		;N,Y-40\$
481	001516'	005301		DEC	R1		
482	001520'	001374		BNE	20\$		
483	001522'	000407		BR	40\$		
484	001524'	004567	004710	30\$: JSR	R5,PRINT		;ISSUE 'AT LAST INT' MSG
485	001530'	005031		.WORD	ATMSG-		
486	001532'	000014		.WORD	12.		
487	001534'	004567	004442	JSR	R5,DISPST		;GO DISPLAY STATUS AT LAST INT
488	001540'	177334		.WORD	ISTAT-		
489	001542'	004567	004672	40\$: JSR	R5,PRINT		;ISSUE 'CURRENTLY' MSG
490	001546'	005027		.WORD	CURMSG-		
491	001550'	000012		.WORD	10.		
492	001552'	004567	004424	JSR	R5,DISPST		;GO DISPLAY CURRENT STATUS
493	001556'	177352		.WORD	CSTAT-		
494	001560'	004767	004616	JSR	PC,PRTIWD		;GO DISPLAY INFO WORDS
495	001564'	032704	000001	DISCNT: BIT	#1,R4		;DISPLAY COUNTS?
496	001570'	001431		BEQ	RPTEND		;Y,N-RPTEND
497	001572'	012700	000024	MOV	#CNTNUM,R0		;SET UP # OF WORDS
498	001576'	010701		MOV	PC,R1		;SET UP ADR OF CNTS
499	001600'	062701	177364	ADD	#COUNTS-. ,R1		
500	001604'	010702		MOV	PC,R2		;SET UP TBL ADR
501	001606'	062702	000066	ADD	#REPTBL-. ,R2		
502	001612'	012267	000012	RPTLP: MOV	(R2)+,RPTBAS		;MOV MSG ADR TO S/R LINKAGE
503	001616'	004067	004062	JSR	R0,SAVREG		;SAVE ALL REG'S
504	001622'	011100		MOV	(R1),R0		;GET CURRENT COUNT
505	001624'	004577	176226	JSR	R5,JBINASC		;CONVERT IT TO ASCII
506	001630'	000000		RPTBAS: .WORD	XXXX		
507	001632'	004067	004062	JSR	R0,RESREG		;RESTORE REG'S
508	001636'	005721		TST	(R1)+		;POINT AT NXT CNT
509	001640'	005300		DEC	R0		;DONE ALL WORDS?
510	001642'	001363		BNE	RPTLP		;Y,N-RPTLP
511	001644'	004567	004570	JSR	R5,PRINT		;GO ISSUE COUNTS MSG
512	001650'	005060		.WORD	CNTSMG-		
513	001652'	000445		.WORD	CNTSEN-CNTSMG		
514	001654'	004567	004560	RPTEND: JSR	R5,PRINT		;ISSUE "END OF REPORT" MSG
515	001660'	004727		.WORD	RENDMG-		
516	001662'	177761		.WORD	-15.		
517	001664'	004067	004030	DVREX: JSR	R0,RESREG		;RESTORE REGISTERS
518	001670'	005725		TST	(R5)+		;SET UP RETURN POINT
519	001672'	000205		RTS	R5		;EXIT IN-LINE
520							
521							
522	001674'	005114		REPTBL: .WORD	BCMRD-RPTBAS		
523	001676'	005122		.WORD	BCMRD+6-RPTBAS		
524	001700'	005137		.WORD	BCMWR-RPTBAS		
525	001702'	005145		.WORD	BCMWR+6-RPTBAS		
526	001704'	005163		.WORD	BCMCK-RPTBAS		
527	001706'	005171		.WORD	BCMCK+6-RPTBAS		
528	001710'	005216		.WORD	CMDCRD-RPTBAS		
529	001712'	005232		.WORD	CMDCWR-RPTBAS		
530	001714'	005247		.WORD	CMDCK-RPTBAS		
531	001716'	005265		.WORD	CMDCK+6-RPTBAS		

```

532 001720' 005301 .WORD CMDDRV-RPTBAS
533 001722' 005316 .WORD CMDCMS-RPTBAS
534 001724' 005344 .WORD CNTCEC-RPTBAS
535 001726' 005365 .WORD CNTDER-RPTBAS
536 001730' 005413 .WORD CNTDLT-RPTBAS
537 001732' 005427 .WORD CNTHCE-RPTBAS
538 001734' 005443 .WORD CNTDCK-RPTBAS
539 001736' 005462 .WORD CNTWCE-RPTBAS
540 001740' 005511 .WORD CNTRTY-RPTBAS
541 001742' 005537 .WORD CNTINT-RPTBAS
    
```

542  
543  
544  
545

;TIMEOUT ERROR ROUTINE

546  
547  
548  
549

```

;JSR R5,TOUTER S/R CALL
TOUTER: JSR R0,SAVREG ;SAVE ALL REGISTERS
        JSR PC,SUPTAD ;SET UP RPCS1 & PROG TBL ADR'S
        JSR R5,STSTAT ;STORE CURRENT STATUS
        .WORD CSTAT-
        JSR R5,TVECT ;CK IF I HAVE VECTOR CONTROL
        BR 10$ ;BR IF I DON'T
        BICB #100,(R4) ;RESET INT ENABLE
        JSR PC,RINTV ;RESET THE INTERRUPT VECTOR
10$: BIC #WT4IOT,(R3) ;RESET WAITING FOR I/O FLG
        JSR R5,ERRCSI ;ISSUE I/O TIMEOUT ERROR MSG
        .WORD IOTO-ERMBAS
        JSR R0,RESREG ;RESTORE REGISTERS
        MOV (SP)+,R5 ;REMOVE RETURN ADR
        JMP @CUPGER ;GO TO ERROR EXIT
    
```

564  
565  
566  
567

;KILL USER PROGRAM ROUTINE

568  
569  
570  
571

```

;JSR R5,KILL S/R CALL
;R3 MUST CONTAIN PROG TBL ADR
;DESTROYS R0,R1
KILL: MOV DREGAD,R1 ;GET DEV REG ADR
        JSR R5,TVECT ;DO I HAVE VECTOR CONTROL?
        BR KILLEX ;BR IF I DON'T
        BITB #100,(R1) ;IS INT ENABLE SET?
        BEQ 10$ ;Y,N-10$
        BICB #100,(R1) ;RESET INT ENABLE
10$: JSR PC,RINTV ;RESET INT VECTOR INFO
KILLEX: RTS ;EXIT IN-LINE
    
```

572  
573  
574  
575  
576  
577  
578  
579  
580



.SBTTL RS03/RS04 NON-I/O FUNCTION ROUTINES

;"STEPUP" FUNCTION ROUTINE

582										
583										
584										
585										
586										
587										
588										
589										
590	002056'	004767	000036	STEPUP:	JSR	PC,STPCOM				:DO COMMON SETUP
591	002062'	062502			ADD	(R5)+,R2				:ADD INCR VALUE TO SECT #
592	002064'	020203		STEPU2:	CMP	R2,R3				:IS SECT # IN RANGE?
593	002066'	103403			BLO	STEPU1				:YES
594	002070'	160302			SUB	R3,R2				:ADJ SECT # DOWNWARDS
595	002072'	005201			INC	R1				:ADD 1 TO HEAD #
596	002074'	000773			BR	STEPU2				:CHECK IT AGAIN
597	002076'	020104		STEPU1:	CMP	R1,R4				:IS TRAK # IN RANGE?
598	002100'	103402			BLO	STEPEX				:N,Y-STEPEX
599	002102'	160401			SUB	R4,R1				:ADJ TRACK # DOWNWARDS
600	002104'	000774			BR	STEPU1				:CHECK IT AGAIN
601	002106'			STEPEX:						
602	002106'	010167	175674		MOV	R1,HEAD				
603	002112'	010267	175672		MOV	R2,SECT				
604	002116'	000205			RTS	R5				:EXIT TO USER PROG
605										
606										
607	002120'			STPCOM:						
608	002120'	016701	175662		MOV	TRAK,R1				:GET STARTING TRACK
609	002124'	016702	175660		MOV	SECT,R2				: AND SECTOR
610	002130'	012704	000100		MOV	#NRTRK,R4				:GET NR TRACKS
611	002134'	012703	000100		MOV	#NRSEC,R3				: AND NR SECTORS
612	002140'	000207			RTS	PC				

;"STEPDN" FUNCTION ROUTINE

613										
614										
615										
616										
617										
618										
619										
620	002142'	004767	177752	STEPDN:	JSR	PC,STPCOM				:DO COMMON SETUP
621	002146'	162502			SUB	(R5)+,R2				:SUB DECR FACTOR FROM SECT #
622	002150'	020203		STEPD2:	CMP	R2,R3				:IS SECT # IN RANGE?
623	002152'	103403			BLO	STEPD1				:YES
624	002154'	060302			ADD	R3,R2				:ADJ SECT # UPWARDS
625	002156'	005301			DEC	R1				:DECR HEAD # BY 1
626	002160'	000773			BR	STEPD2				:CHECK IT AGAIN
627	002162'	020104		STEPD1:	CMP	R1,R4				:IS TRACK # IN RANGE?
628	002164'	103750			BLO	STEPEX				:N,Y-STEPEX
629	002166'	060401			ADD	R4,R1				:ADJ IT UPWARDS
630	002170'	000774			RTS	PC				:CHECK IT AGAIN

;"WAIT" FUNCTION ROUTINE

631										
632										
633										
634										
635										
636										
637	002172'	042767	100000	175602	WAIT:	BIC	#WAITMD,DFLGWD			:RESET THE "NOWAIT" FLAG

```

638 002200' 004767 002024      JSR    PC,CKDBSY      ;WAIT IF BUSY & DO TERMINATION
639 002204' 004767 002276      JSR    PC,RINTV      ;RESET THE INTERRUPT VECTOR
640 002210' 000205              RTS     RS            ;EXIT IN-LINE
641
642
643      ;"NOWAIT" FUNCTION ROUTINE
644
645      ;JSR    RS,NOWAIT      FUNCTION CALL
646
647 002212' 052767 100000 175562 NOWAIT: BIS    #WAITMD,DFLGWD    ;SET THE "NOWAIT" FLAG
648 002220' 000205              RTS     RS            ;EXIT IN-LINE
649
650
651      ;"APORT" FUNCTION ROUTINE
652
653      ;JSR    RS,APORT      FUNCTION CALL
654
655 002222' 042767 002000 177030 APORT: BIC    #PORT,RPCSIV    ;RESET THE PORT BIT
656 002230' 000205              RTS     RS            ;EXIT IN-LINE
657
658
659      ;"BPORT" FUNCTION ROUTINE
660
661      ;JSR    RS,BPORT      FUNCTION CALL
662
663 002232' 052767 002000 177020 BPORT: BIS    #PORT,RPCSIV    ;SET THE PORT BIT TO B PORT
664 002240' 000205              RTS     RS            ;EXIT IN-LINE
665
666
667      ;"ODD" FUNCTION ROUTINE
668
669      ;JSR    RS,ODD        FUNCTION CALL
670
671 002242' 042767 000020 177012 ODD:  BIC    #PARITY,RPCS2V    ;RESET THE PARITY BIT
672 002250' 000205              RTS     RS            ;EXIT IN-LINE

```



```

674                                     ;"EVEN" FUNCTION ROUTINE
675
676                                     ;JSR   RS,EVEN           FUNCTION CALL
677
678 002252' 052767 000020 177002 EVEN:  BIS   #PARITY,RPCS2V      ;SET THE PARITY BIT FOR EVEN
679 002260' 000205                                     RTS   RS                 ;EXIT IN-LINE
680
681
682                                     ;"BAION" FUNCTION ROUTINE
683
684                                     ;JSR   RS,BAION          FUNCTION CALL
685
686
687 002262' 052767 000010 176772 BAION:  BIS   #BAI,RPCS2V        ;SET THE BAI BIT
688 002270' 000205                                     RTS   RS                 ;EXIT IN-LINE
689
690
691                                     ;"BAIOFF" FUNCTION ROUTINE
692
693                                     ;JSR   RS,BAIOFF         FUNCTION CALL
694
695
696 002272' 042767 000010 176762 BAIOFF: BIC   #BAI,RPCS2V        ;RESET THE BAI BIT
                                RTS   RS                 ;EXIT IN-LINE

```





```

752                                     ;HOUSEKEEP THE DISK
753
754                                     ;JSR   PC,ACQHSK       S/R CALL
755
756                                     ;R5 = ADR AFTER USER PROG JSR
757                                     ;R4 = RPCS1 ADR
758                                     ;R3 = PROG TBL ADR
759
760                                     ;DESTROYS R0
761
762 002462' 010146          ACQHSK: MOV   R1,-(SP)           ;SAVE R1
763 002464'                ACQRTY:                ;
764 002464' 012763 072460 000030          MOV   #ITIME,PTCNT(R3)      ;SETUP INTERRUPT TIMER
765 002472' 042767 000100 175302          BIC   #CMDISU,DFLGWD      ;HMKP FLAG BITS
766 002500' 116300 000035                MOVVB PCURDV(R3),R0      ;GET MY UNIT #
767 002504' 020027 000007                CMP   R0,#7              ;VALID UNIT #?
768 002510' 101411                BLOS  10$                ;N,Y-10$
769 002512' 012767 003134 000336          MOV   #INVDVN-ERMBAS,ACQEAD ;SET UP ADR OF INV UNIT # ERR MSG
770 002520' 005267 176472                INC   DATAER           ;COUNT OPERATOR ERROR
771 002524' 005367 176464                DEC   ERRCNT
772 002530' 000167 000312                JMP   ACQERR            ;GO REPORT THE ERROR
773 002534' 010001                10$: MOV   R0,R1          ;GET DISPLACEMENT INTO
774 002536' 006301                ASL   R1                ;THE ATA TABLE FOR
775 002540' 060701                ADD   PC,R1             ;THIS UNIT #
776 002542' 062701 000336          ADD   #ATATBL-. ,R1     ;STORE ATA BIT MASKS FOR
777 002546' 112167 000346          MOVVB (R1)+,MYATA      ;THIS UNIT #
778 002552' 111167 000344          MOVVB (R1),OTHATA     ;SET PAT & BAI BITS IN UNIT #
779 002556' 056700 176500          BIS   RPCS2V,R0        ;HMKP 1ST TIME FLAG
780 002562' 005001                CLR   R1                ;MOVE UNIT # TO RH11
781 002564' 010064 000010                12$: MOV   R0,RPCS2(R4) ;SELECT THE DRIVE
782 002570' 005714                TST   (R4)              ;NON-EXISTENT DRIVE?
783 002572' 032764 010000 000010          BIT   #NED,RPCS2(R4)   ;NO
784 002600' 001413                BEQ   GOTDSK           ;FIRST TIME?
785 002602' 005701                TST   R1                ;Y,N-16$
786 002604' 001005                BNE   16$              ;RESET FIRST TIME
787 002606' 005201                INC   R1                ;DO RH11 ERROR CLEAR
788 002610' 112764 000100 000001          MOVVB #100,1(R4)      ;GO LOAD UNIT # AGAIN
789 002616' 000762                BR    12$              ;SET UP ADR OF NON-EXIST DRIVE ERR MSG
790 002620' 012767 002716 000230          16$: MOV   #NONEXD-ERMBAS,ACQEAD ;GO REPORT THE ERROR
791 002626' 000507                BR    ACQERR           ;SETUP INTERRUPT TIMER
792 002630' 012763 072460 000030          GOTDSK: MOV  #ITIME,PTCNT(R3) ;DISK ON-LINE?
793 002636' 032764 010000 000012          BIT   #MOL,RPDS(R4)   ;N,Y-40$
794 002644' 001004                BNE   40$              ;SET UP ADR OF DISK OFFLINE ERR MSG
795 002646' 012767 002774 000202          MOV   #OFFLIN-ERMBAS,ACQEAD ;GO REPORT THE ERROR
796 002654' 000474                BR    ACQERR           ;RESET ERROR LOOP CNT
797 002656' 005001                40$: CLR   R1          ;IS 'SC' ERROR BIT SET?
798 002660' 005714                50$: TST   (R4)        ;Y,N-100$
799 002662' 100062                BPL   100$            ;IS THERE AN UNSAFE ERROR?
800 002664' 032764 040000 000014          BIT   #UNS,RPER1(R4)  ;N,Y-80$
801 002672' 001033                BNE   80$              ;5TH TIME THRU ON THIS ERROR?
802 002674' 020127 000005                CMP   R1,#5           ;Y,N-60$
803 002700' 001004                BNE   60$              ;SET UP ADR OF INITIATION ERR MSG
804 002702' 012767 002750 000146          MOV   #INITDE-ERMBAS,ACQEAD ;GO REPORT THE ERROR
805 002710' 000456                BR    ACQERR           ;ADD 1 TO ERROR LOOP CNT
806 002712' 005201                60$: INC   R1          ;GET ATA REG
807 002714' 016400 000016          MOV   RPAS(R4),R0

```

808	002720'	105700			TSTB	RO		: IS THERE AN ATTENTION?
809	002722'	001432			BEQ	90\$		: Y,N-90\$
810	002724'	036700	000172		BIT	OTHATA,RO		: MY ATA ONLY?
811	002730'	001427			BEQ	90\$		: N,Y-90\$
812	002732'	036700	000162		BIT	MYATA,RO		: MY ATA LINE ALSO?
813	002736'	001006			BNE	70\$		: N,Y-70\$
814	002740'	010064	000016		MOV	RO,RPAS(R4)		: RESET OTHER ATA LINES
815	002744'	112764	000100	000001	MOVB	#100,1(R4)		: DO RH11 ERROR CLEAR
816	002752'	000742			BR	50\$		: GO CK 'SC' AGAIN
817	002754'	010064	000016	70\$:	MOV	RO,RPAS(R4)		: RESET OTHER ATA LINES
818	002760'	000413			BR	90\$		: GO CLEAR MY DRIVE
819	002762'	005767	175034	80\$:	TST	ERRI		: ERROR ON PREVIOUS I/O?
820	002766'	001010			BNE	90\$		: N,Y-90\$
821	002770'	032767	000200	175004	BIT	#ANYIOI,DFLGWD		: ANY I/O CMNDS BEEN ISSUED?
822	002776'	001404			BEQ	90\$		: Y,N-90\$
823	003000'	012767	002741	000050	MOV	#INITUS-ERMBAS,ACQEAD		: SET UP ADR OF UNSAFE ERROR MSG
824	003006'	000417			BR	ACQERR		: GO REPORT THE ERROR
825	003010'	005067	175006	90\$:	CLR	ERRI		: RESET PREV I/O ERR FLAG
826	003014'	052767	000200	174760	BIS	#ANYIOI,DFLGWD		: SET FLAG TO PREVENT LOOP
827	003022'	012714	040011		MOV	#TRE!DCCODE!GO,(R4)		: DO RH11 & DRIVE CLEAR
828	003026'	000714			BR	50\$		: GO CK 'SC' AGAIN
829	003030'	005067	174766	100\$:	CLR	ERRI		: HSKP ERROR INDICATOR
830	003034'	042767	000001	174740	BIC	#IOERR,DFLGWD		: & FLAGS
831	003042'	012601			MOV	(SP)+,R1		: RESTORE R1
832	003044'	000207			RTS	PC		: EXIT IN-LINE
833								
834	003046'	010146			ACQERR: MOV	R1,-(SP)		: SAVE R1 & R2
835	003050'	010246			MOV	R2,-(SP)		
836	003052'	004567	001504		JSR	RS,ERRCS		: STORE CURR STATUS & REPORT
837	003056'	002750			ACQEAD: .WORD	INITDE-ERMBAS		: THE ERROR
838	003060'	012602			ACQERC: MOV	(SP)+,R2		: RESTORE R1 & R2
839	003062'	012601			MOV	(SP)+,R1		
840	003064'	004577	174760		JSR	RS,ACUPGER		: GO TO MPG'S ERROR RETURN POINT
841	003070'	004767	001134		ACQGDK: JSR	PC,CKDBSY		: GO CK IF DEV IS BUSY NOW
842	003074'	000167	177364		JMP	ACQRTY		: GO TRY AGAIN
843								
844								
845	003100'	001	376		ATATBL: .BYTE	001,376		
846	003102'	002	375		.BYTE	002,375		
847	003104'	004	373		.BYTE	004,373		
848	003106'	010	367		.BYTE	010,367		
849	003110'	020	357		.BYTE	020,357		
850	003112'	040	337		.BYTE	040,337		
851	003114'	100	277		.BYTE	100,277		
852	003116'	200	177		.BYTE	200,177		
853								
854	003120'	000000			MYATA: .WORD	0		
855	003122'	000000			OTHATA: .WORD	0		





```

910 ;"SEARCH" FUNCTION ROUTINE
911
912 ;JSR R5,SEARCH FUNCTION CALL
913
914 003234' 012702 000131 SEARCH: MOV #SCODE!IE!GO,R2 ;GET SEARCH COMMAND
915 003240' 012701 000060 MOV #060,R1 ;SET UP CMND FLAG WORD
916 003244' 004767 000760 JSR PC,CKDBSY ;SEE IF DEVICE IS BUSY
917 003250' 005267 175732 INC SKCNT ;COUNT SEARCHES
918 003254' 000400 BR CMDCOM ;GO TO CMND COMMON PROCESSING
919
    
```



; INTERRUPT TYPE I/O FUNCTION COMMON PROCESSING ROUTINE

;R4 = ADR OF RPCS1 DEV REG  
;R3 = PROG TBL ADR  
;R2 = COMMAND CODE  
;R1 = COMMAND FLAG WORD  
;R0 = ADR OF BYTE COUNT, IF APPLICABLE

; CMND FLAGWORD FORMAT:

;BIT 7 = 200 = PERFORM RETRIES ON CMND  
;BIT 6 = 100 = NOT USED  
;BIT 5 = 040 = CMND TERMINATES WITH ATA  
;BIT 4 = 020 = SET UP HEAD/SECT #  
;BIT 3 = 010 = INCREMENT BYTE COUNTS  
;BIT 2 = 004 = DATA TRANSFER CMND  
;BIT 1 = 002 = 3 ARGUMENT CMND  
;BIT 0 = 001 = 4 ARGUMENT CMND

921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976

```

003256' 010067 175754      CMDCOM: MOV      R0,CNTADR      ;SAVE ADR OF BYTE COUNT
003262' 004767 177174      JSR      PC,ACQHSK    ;ACQUIRE & HOUSEKEEP THE DISK
003266' 010167 175746      MOV      R1,CURFLG    ;SAVE FLAGWD FOR TERMINATION
003272' 056702 175762      BIS      RPCS1V,R2    ;SET PORT SELECT BIT IN CMND CODE
003276' 010267 175740      MOV      R2,CURCMD    ;SAVE CURR CMND CODE
003302' 032701 000003      BIT      #3,R1        ;THIS CMND HAVE BUS ADR & WD CNT?
003306' 001416              BEQ      10$          ;Y,N-10$
003310' 012567 175730      MOV      (R5)+,CURADR ;STORE 2 WORD BUS ADR
003314' 012567 175726      MOV      (R5)+,CURADR+2
003320' 012500              MOV      (R5)+,R0
003322' 000241              CLC
003324' 006000              ROR      R0
003326' 005400              NEG      R0
003330' 010067 175714      MOV      R0,CURCNT    ;MAKE IT NEGATIVE
003334' 032701 000001      BIT      #1,R1        ;SAVE IT
003340' 001401              BEQ      10$          ;THERE A 4TH WORD?
003342' 005725              TST      (R5)+        ;Y,N-10$
003344'                  ;BYPASS IT
10$:
003344' 004767 000070      JSR      PC,SUIORG    ;GO SET UP REGS FOR I/O
003350' 016767 174436 175676 MOV      RTRY,CURRTY  ;INITIALIZE RETRY COUNT
003356' 005067 175674      CLR      RTRYIP        ;CLEAR RETRY IN PROGRESS FLAG
003362' 012767 003025 001064 MOV      #IOTERM-ERMBAS,INTEAD ;INIT TERMINATION ERROR MSG
003370' 052767 000002 174404 BIS      #DOTERM,DFLGWD ;SET THE "PROCESS TERMINATION" FLAG
003376' 052713 000010      BIS      #WT4IOT,(R3) ;SET WAITING FOR I/O TERM FLAG
003402' 052767 000300 174372 BIS      #CMDISU+ANYIOI,DFLGWD ;SET CMND ISSUED FLAGS
003410' 010214              MOV      R2,(R4)      ;ISSUE SPECIFIED CMND
003412' 005767 174364      TST      DFLGWD      ;"NOWAIT" BIT SET?
003416' 100003              BPL      40$          ;Y,N-40$
003420' 042713 000010      BIC      #WT4IOT,(R3) ;RESET WAITING FOR I/O TERM
003424' 000404              BR       50$          ;GO TO EXIT
003426' 004577 174414      40$: JSR      R5,ACIOBSY  ;WAIT FOR I/O TO COMPLETE
003432' 004767 000722      JSR      PC,PROCTM    ;GO PROCESS TERMINATION
003436' 000205      50$: RTS      R5      ;EXIT IN-LINE TO USER PROG

```

```

977
978
979
980
981
982
983
984
985
986
987
988
989
990 003440' 032701 000020      SUIORG: BIT      #20,R1      ;NEED TO SET UP HEAD/SECT?
991 003444' 001411              BEQ      10$      ;Y,N-10$
992 003446' 016746 174334      MOV      HEAD,-(SP) ;GET HEAD #
993 003452' 000316              SWAB     (SP)      ;PUT IN CORRECT BIT POSITION
994 003454' 006216              ASR      (SP)
995 003456' 006216              ASR      (SP)
996 003460' 056716 174324      BIS      SECT,(SP) ; AND MERGE WITH SECTOR
997 003464' 012664 000006      MOV      (SP)+,RPDA(R4) ;LOAD HEAD & SECT #'S
998 003470' 032701 000004      10$: BIT      #4,R1 ;DATA XFER CMND?
999 003474' 001423              BEQ      30$      ;Y,N-30$
1000 003476' 016700 175542      MOV      CURADR,RO ;GET HIGH BITS OF ADR
1001 003502' 042700 177774      BIC      #177774,RO ;RESET BITS ABOVE A17
1002 003506' 000300              SWAB     RO ;ALIGN BITS A16 & A17
1003 003510' 050002              BIS      RO,R2 ;SET THEM INTO CMND CODE WORD
1004 003512' 016764 175530 000004 MOV      CURADR+2,RPBA(R4) ;LOAD BITS 0-15 OF ADR
1005 003520' 032777 000010 174336 BIT      #CPU70,ACSYSFW ;RUNNING ON AN 11/70?
1006 003526' 001403              BEQ      20$      ;Y,N-20$
1007 003530' 016764 175510 00003^ MOV      CURADR,RPBAE(R4) ;MOVE BITS A16-A21 TO ADR EXT
1008 003536' 016764 175506 000002 20$: MOV      CURCNT,RPWC(R4) ;LOAD WORD COUNT
1009 003544' 000207              30$: RTS      PC ;EXIT IN-LINE
    
```



```

1011 .SBTTL RS03/RS04 INTERRUPT SERVICE ROUTINE
1012
1013
1014 003546' 004067 002132 RHPINT: JSR RD, SAVREG ;SAVE ALL REGISTERS
1015 003552' 004567 002200 JSR RS, STSTAT ;GO STORE ALL DEV REG'S
1016 003556' 175316 .WORD ISTAT-
1017 003560' 005267 175446 INC INTCNT ;ADD 1 TO INTERRUPT CNT
1018 003564' 004767 002146 JSR PC, SUPTAD ;SET UP PROG TBL & RPCS1 ADR'S
1019 003570' 016701 175444 MOV CURFLG, R1 ;GET THIS CMND'S FLGWD
1020 003574' 005714 TST (R4) ;IS 'SC' BIT SET?
1021 003576' 100426 BMI CKSC ;N, Y-CKSC
1022 003600' 032701 000040 BIT #40, R1 ;CMND SUPPOSED TO SET ATA?
1023 003604' 001406 BEQ CLRWTF ;Y, N-CLRWTF
1024 003606' 012767 003053 000640 MOV #NOATA-ERMBAS, INTEAD ;SET UP NO ATA ERR MSG ADR
1025 003614' 052767 000001 174160 SETERR: BIS #IOERR, DFLGWD ;SET THE TERMINATION I/O ERR FLAG
1026 003622' 042713 000010 CLRWTF: BIC #WT4IOT, (R3) ;RESET WAITING FOR I/O TERM
1027 003626' 032767 000004 175404 BIT #4, CURFLG ;THIS A DATA TRANSFER CMND?
1028 003634' 001403 BEQ INTEX ;Y, N-INTEX
1029 003636' 016467 000002 175406 MOV RPWC(R4), FINCNT ;SAVE FINAL WORD COUNT
1030 003644' 004067 002050 INTEX: JSR RD, RESREG ;RESTORE ALL REGISTERS
1031 003650' 000177 174220 JMP @RTNINT ;EXIT FROM INTERRUPT
1032
1033 003654' 032714 020000 CKSC: BIT #MCPE, (R4) ;MCPE ERROR SET?
1034 003660' 001402 BEQ 28$ ;N, Y-HARDER
1035 003662' 000167 000310 JMP HARDER
1036 003666' 032714 040000 28$: BIT #TRE, (R4) ;TRE ERROR BIT SET?
1037 003672' 001031 BNE ERRFND ;N, Y-ERRFND
1038 003674' 032764 040000 000012 BIT #ERR, RPDS(R4) ;ERROR SUMMARY BIT SET?
1039 003702' 001025 BNE ERRFND ;N, Y-ERRFND
1040 003704' 016400 000016 MOV RPAS(R4), RO ;GET ATA REG
1041 003710' 036700 177204 BIT MYATA, RO ;MY ATA BIT SET?
1042 003714' 001011 BNE 40$ ;N, Y-40$
1043 003716' 010064 000016 MOV RO, RPAS(R4) ;RESET OTHER ATA BITS
1044 003722' 032701 000004 BIT #4, R1 ;DATA XFER CMND?
1045 003726' 001335 BNE CLRWTF ;N, Y-CLRWTF
1046 003730' 036764 177164 000016 30$: BIT MYATA, RPAS(R4) ;MY ATA SET?
1047 003736' 001774 BEQ 30$ ;Y, N-30$
1048 003740' 032701 000040 40$: BIT #40, R1 ;CMND SUPPOSED TO SET ATA?
1049 003744' 001326 BNE CLRWTF ;N, Y-CLRWTF
1050 003746' 012767 003073 000500 MOV #UXPATA-ERMBAS, INTEAD ;SET UP UNEXP ATA ERR MSG ADR
1051 003754' 000717 BR SETERR ;GO TO ERROR EXIT
1052
1053 003756' 032764 037400 000010 ERRFND: BIT #037400, RPCS2(R4) ;UPE/NED/NEM/PGE/MXF/MDPE
1054 003764' 001104 BNE HARDER ;IN RPCS2? (N, Y-HARDER)
1055 003766' 032764 067017 000014 BIT #067017, RPER1(R4) ;UNS/OPI/WLE/IAE/AOE/PAR/RMR/
1056 003774' 001100 BNE HARDER ;ILR/ILF IN RPER1? (N, Y-HARDER)
1057 003776' 032764 140000 000010 BIT #140000, RPCS2(R4) ;DLT OR WCE IN RPCS2?
1058 004004' 001004 BNE CKRTRY ;N, Y-CKRTRY
1059 004006' 032764 010000 000014 BIT #010000, RPER1(R4) ;DTE IN RPER1?
1060 004014' 001500 BEQ CKCORR ;Y, N-CKCORR
1061 004016' 005767 175234 CKRTRY: TST RTRYIP ;ALREADY DONE RETRIES ON THIS CMND?
1062 004022' 001036 BNE 55$ ;N, Y-55$
1063 004024' 005767 175224 TST CURRTY ;ARE RETRIES SPECIFIED?
1064 004030' 001462 BEQ HARDER ;Y, N-HARDER
1065 004032' 032764 100000 000010 BIT #100000, RPCS2(R4) ;DLT ERROR?
1066 004040' 001403 BEQ 42$ ;Y, N-42$

```





```

1102 .SBTTL SUBROUTINES FOR RS03/RS04 FUNCTION ROUTINES
1103
1104
1105 ;CHECK IF DEVICE IS BUSY AND WAIT IF IT IS
1106
1107 ;JSR PC,CKDBSY S/R CALL
1108
1109 ;DESTROYS R0,R3,R4
1110 ;ON EXIT: R3 = PROG TBL ADR
1111 ; R4 = RPCS1 ADR
1112
1113 004230' 004767 001502 CKDBSY: JSR PC,SUPTAD ;SET UP PROG TBL & RPCS1 ADR'S
1114 004234' 016400 000010 10$: MOV RPCS2(R4),R0 ;GET CURR UNIT #
1115 004240' 010046 MOV RO,-(SP) ;SAVE IT
1116 004242' 042700 177747 BIC #177747,R0 ;RESET UNIT # & OTHER BITS
1117 004246' 156300 000035 BISB PCURDV(R3),R0 ;SET IN MY UNIT #
1118 004252' 110064 000010 MOV R0,RPCS2(R4) ;SELECT MY DRIVE
1119 004256' 011400 MOV (R4),R0 ;GET RPCS1 REG
1120 004260' 112664 000010 MOV (SP)+,RPCS2(R4) ;RESTORE ORIG UNIT #
1121 004264' 032700 000100 BIT #100,R0 ;INT ENABLE ON?
1122 004270' 001403 BEQ 20$ ;Y,N-20$
1123 004272' 004577 173550 15$: JSR R5,@CIOBSY ;RELEASE CONTROL
1124 004276' 000756 BR 10$ ;GO CK AGAIN
1125 004300' 20$:
1126 004300' 032767 000002 173474 BIT #DOTERM,DFLGWD ;SHOULD PROCESS PREV TERMINATION?
1127 004306' 001403 BEQ 30$ ;Y,N-30$
1128 004310' 004767 000044 JSR PC,PROCTM ;GO PROCESS TERMINATION
1129 004314' 000747 BR 10$ ;GO CK INT ENABLE AGAIN
1130 004316' 016767 173504 000012 30$: MOV IVCTAD,40$ ;STORE INT VECTOR ADR
1131 004324' 016767 173500 000006 MOV PSWD,45$ ;STORE PROC STATUS WORD
1132 004332' 004577 173530 JSR R5,@SETVEC ;GO SET UP INTERRUPT VECTOR
1133 004336' 000000 40$: .WORD XXXX ;INT VECTOR ADR
1134 004340' 000000 45$: .WORD XXXX ;PSW
1135 004342' 177204 .WORD RHPINT- ;REL INT ROUT ADR
1136 004344' 010567 174664 MOV R5,ERRADR ;SAVE CURR USER STMT ADR
1137 004350' 162767 000004 174656 SUB #4,ERRADR
1138 004356' 000207 RTS PC ;EXIT IN-LINE
1139
1140
1141 ;PROCESS TERMINATION OF PREVIOUS I/O FUNCTION
1142
1143 ;JSR PC,PROCTM S/R CALL
1144
1145 ;R3 = PROG TABLE ADR
1146
1147 ;DESTROYS R0
1148
1149 004360' 010146 PROCTM: MOV R1,-(SP) ;SAVE R1 & R2
1150 004362' 010246 MOV R2,-(SP)
1151 004364' 042767 000002 173410 BIC #DOTERM,DFLGWD ;RESET PROCESS TERMINATION FLAG
1152 004372' 032767 000010 174640 BIT #10,CURFLG ;INCR BYTE COUNT?
1153 004400' 001417 BEQ 20$ ;Y,N-20$
1154 004402' 016700 174642 MOV CURCNT,R0 ;GET INITIAL WORD CNT
1155 004406' 005400 NEG R0 ;MAKE IT POSITIVE AGAIN
1156 004410' 016701 174636 MOV FINCNT,R1 ;GET FINAL WORD CNT
1157 004414' 100001 BPL 10$ ;IS IT NEGATIVE? (Y,N-10$)

```

```

1158 004416' 005401          NEG      R1          ;MAKE IT POSITIVE
1159 004420' 160100          10$: SUB      R1,R0      ;SUB REMAINING CNT FROM INITIAL CNT
1160 004422' 006300          ASL      R0          ;MAKE IT A BYTE CNT
1161 004424' 010067 173370  MOV      R0,SIZE     ;STORE # OF BYTES ACTUALLY XFERRD
1162 004430' 016701 174602  MOV      CNTADR,R1   ;GET ADR OF BYTE CNT TOTALS
1163 004434' 060011          ADD      R0,(R1)    ;ADD IN THIS CNT
1164 004436' 005541          ADC      -(R1)      ;UPDATE MOST SIGNF WORD OF CNT
1165 004440' 032767 000001 173334 20$: BIT      @IOERR,DFLGWD ;WAS THERE AN ERROR?
1166 004446' 001412          BEQ      PROCEX     ;Y N-PROCEX
1167 004450' 004567 000132          JSR      RS,ERRIS   ;GO ISSUE I/O TERMINATION
1168 004454' 003025          INTEAD: .WORD     IOTERM-ERMBAS ;ERROR MSG
1169 004456' 004767 000024          JSR      PC,RINTV  ;RESET THE INT VECTOR
1170 004462' 012602          MOV      (SP)+,R2  ;RESTORE R1 & R2
1171 004464' 012601          MOV      (SP)+,R1
1172 004466' 004577 173356          JSR      RS,@CUPGER ;GO TO MPG ERR RETN POINT
1173 004472' 000207          RTS      PC        ;RETURN IN-LINE
1174 004474' 004767 000006          PROCEX: JSR      PC,RINTV ;GO RESET INT VECTOR
1175 004500' 012602          MOV      (SP)+,R2  ;RESTORE R1 & R2
1176 004502' 012601          MOV      (SP)+,R1
1177 004504' 000207          RTS      PC        ;EXIT IN-LINE
1178
1179
1180          ;RESET INTERRUPT VECTOR S/R
1181
1182          ;JSR      PC,RINTV      S/R CALL
1183          ;R3 MUST CONTAIN PROG TBL ADR
1184          ;DESTROYS R0
1185
1186 004506' 004567 000020          RINTV: JSR      RS,TVECT ;GO CK IF I HAVE VECTOR CONTROL
1187 004512' 000406          BR      RINTEX     ;BR IF I DON'T
1188 004514' 016767 173306 000004          MOV      IVCTAD,10$ ;GET CURR INT VECT ADR
1189 004522' 004577 173342          JSR      RS,@CLAVEC ;GO HAVE MPG CLEAR IT
1190 004526' 000000          10$: .WORD     XXXX
1191 004530' 000207          RINTEX: RTS      PC   ;EXIT IN-LINE
1192
1193
1194          ;TEST INTERRUPT VECTOR S/R
1195
1196          ;JSR      RS,TVECT      S/R CALL
1197          ;BR      LABEL        EXECUTED IF NOT SAME
1198          ;R3 MUST CONTAIN PROG TBL ADR
1199          ;DESTROYS R0
1200
1201 004532' 016767 173270 000010          TVECT: MOV      IVCTAD,20$ ;GET CURR INT VECT ADR
1202 004540' 016346 000004          MOV      PFWADR(R3),-(SP) ;STORE FLGWD ADR TO IDENTIFY ME
1203 004544' 004577 173322          JSR      RS,@TSTVEC ;DO I HAVE VECTOR CONTROL?
1204 004550' 000000          20$: .WORD     XXXX ;MPG WILL TELL ME SINCE I CAN'T
1205 004552' 176774          .WORD     RHPINT-. ;GET AT LOWER MEM IF MEM MGMNT
1206 004554' 000401          BR      TVECTX     ;BR IF I DON'T HAVE CNTRL
1207 004556' 005725          TST      (R5)+     ;BYPASS BR INST IN S/R CALL
1208 004560' 000205          TVECTX: RTS      RS   ;EXIT IN-LINE
    
```



```

1210                                     ;ERROR INFORMATION DISPLAY S/R
1211
1212                                     ;JSR   R5,ERRCS           S/R CALL FOR CURR STATUS
1213                                     ;JSR   R5,ERRCS1        S/R CALL FOR CURR STATUS W/O STORING
1214                                     ;JSR   R5,ERRIS        S/R CALL FOR INT STATUS
1215                                     ;.WORD MSGADR-ERMBAS   REL ADR OF ERROR MSG
1216
1217                                     ;R3 = PROG TABLE ADR
1218                                     ;DESTROYS R0,R1,R2
1219
1220 004562' 004567 001170      ERRCS: JSR   R5,STSTAT           ;STORE CURR STATUS
1221 004566' 174342          .WORD CSTAT-
1222 004570' 012767 173706 000424 ERRCS1: MOV   #CSTAT-ERSTAD,ERSTAD ;STORE ADR OF CURR STATUS
1223 004576' 012767 174114 000212      MOV   #CSTAT-EBSBAS,EBSTAT
1224 004604' 000406          BR    ERRCOM
1225 004606' 012767 173652 000406 ERRIS: MOV   #ISTAT-ERSTAD,ERSTAD ;GO TO COMMON POINT
1226 004614' 012767 174060 000174      MOV   #ISTAT-EBSBAS,EBSTAT ;STORE ADR OF LAST INT STATUS
1227 004622' 012567 000134          ERRCOM: MOV   (R5)+,ERMBAS
1228 004626' 005267 174362          INC   ERRCNT           ;STORE MSG ADR
1229 004632' 012767 000001 173162      MOV   #1,ERRI         ;ADD 1 TO ERROR CNT
1230 004640' 032763 000400 000002      BIT   #DOERCK,POPSW(R3) ;SET THE ERROR INDICATOR
1231 004646' 001004          BNE   2$              ;SUPPOSED TO DO ERROR CHECKING?
1232 004650' 032763 020000 000002      BIT   #PRONER,POPSW(R3) ;Y,N-2$
1233 004656' 001402          BEQ   4$              ;ERROR PRINTING INHIBITED?
1234 004660' 000167 000424          JMP   ERREX          ;N,Y-4$
1235 004664' 010446          4$:  MOV   R4,-(SP)      ;GO TO EXIT
1236 004666' 010546          MOV   R5,-(SP)      ;SAVE R4 & R5
1237 004670' 005004          CLR   R4
1238 004672' 004767 001170          JSR   PC,DEVID       ;SET USER MODE PRINT FLAG
1239 004676' 032767 000100 173076      BIT   #CMDISU,DFLGWD ;DISPLAY DEVICE I.D.
1240 004704' 001005          BNE   6$              ;HAS THE CMND BEEN ISSUED?
1241 004706' 004567 001526          JSR   R5,PRINT       ;N,Y-6$
1242 004712' 002652          .WORD BEFIO-         ;PRINT THE "BEFORE ISSUING I/O" MSG
1243 004714' 000030          .WORD 24
1244 004716' 000404          BR    8$
1245 004720' 004567 001514          6$:  JSR   R5,PRINT       ;GO CALC MSG LNGTH
1246 004724' 002670          .WORD AFTIO-         ;PRINT THE "AFTER ISSUING I/O" MSG
1247 004726' 000027          .WORD 23
1248 004730' 010700          8$:  MOV   PC,R0           ;GET START ADR OF ERROR MSG
1249 004732' 062700 000030          ADD   #ERMBAS-.,R0
1250 004736' 061000          ADD   (R0),R0
1251 004740' 012701 177777          MOV   #-1,R1
1252 004744' 005201          10$: INC   R1             ;INITIALIZE MSG LENGTH
1253 004746' 105720          TSTB (R0)+           ;ADD 1 TO MSG LENGTH
1254 004750' 001375          BNE   10$            ;MSG TERMINATOR?
1255 004752' 010167 000006          MOV   R1,ERMBAS+2   ;Y,N-10$
1256 004756' 004567 001456          JSR   R5,PRINT       ;STORE MSG LENGTH
1257 004762' 000000          ERMBAS: .WORD XXXX   ;PRINT ERROR MSG SPECIFIED
1258 004764' 000000          .WORD XXXX
1259 004766' 026727 177770 003134      CMP   ERMBAS,#INVDVN-ERMBAS ;INVALID UNIT # MSG OR HIGHER?
1260 004774' 103115          BHS   ERRSNM        ;N,Y-ERRSNM
1261 004776' 010701          MOV   PC,R1         ;GET ADR OF CODE AREA IN ERR MSG
1262 005000' 062701 002464          ADD   #CODFLD-.,R1
1263 005004' 010700          MOV   PC,R0         ;SET UP ADR OF ERROR CODE TBL
1264 005006' 062700 000304          ADD   #ERCDTB-.,R0
1265 005012' 010702          MOV   PC,R2         ;SET UP ADR OF STORED DEV REG'S

```

1266	005014'	062702			EBSBAS: ADD	(PC)+,R2		
1267	005016'	174114			EBSTAT: .WORD	CSTAT-EBSBAS		
1268	005020'	012767	000015	000166	MOV	#13.,70\$		: INITIALIZE MSG LENGTH
1269	005026'	012746	000100		MOV	#64.,-(SP)		: INITIALIZE CODE FIELD CNT
1270	005032'	012205			15\$: MOV	(R2)+,R5		: GET NEXT DEV REG WORD
1271	005034'	000305			17\$: SWAB	R5		: GET DESIRED BYTE IN LOW BYTE
1272	005036'	112004			20\$: MOVB	(R0)+,R4		: GET FLAG & LENGTH BYTE
1273	005040'	005704			TST	R4		: END OF THE CODE TBL?
1274	005042'	001455			BEQ	60\$		: N,Y-60\$
1275	005044'	122704	000377		CMPB	#377,R4		: GO TO NXT DEV REG WORD?
1276	005050'	001770			BEQ	15\$		: N,Y-15\$
1277	005052'	122704	000376		CMPB	#376,R4		: GO TO NXT BYTE IN DEV REG WORD?
1278	005056'	001766			BEQ	17\$		: N,Y-17\$
1279	005060'	032704	000040		BIT	#40,R4		: THIS AN 11/70 ONLY ERROR BIT?
1280	005064'	001405			BEQ	26\$		: NO
1281	005066'	032777	000010	172770	BIT	#CPU70,DCSYSFW		: RUNNING ON AN 11/70?
1282	005074'	001411			BEQ	35\$		: Y,N-35\$
1283	005076'	000400			BR	26\$		: GO CK ERROR BIT
1284	005100'	032704	000100		26\$: BIT	#100,R4		: BIT VALUE OF 0 = AN ERROR CONDITION?
1285	005104'	001403			BEQ	30\$		: Y,N-30\$
1286	005106'	131005			BITB	(R0),R5		: THIS BIT RESET IN DEV REG BYTE?
1287	005110'	001407			BEQ	40\$		: N,Y-40\$
1288	005112'	000402			BR	35\$		: GO TO NXT TBL ENTRY
1289	005114'	131005			30\$: BITB	(R0),R5		: THIS ERROR BIT SET IN DEV REG BYTE?
1290	005116'	001004			BNE	40\$		: N,Y-40\$
1291	005120'	042704	177770		35\$: BIC	#177770,R4		: ISOLATE ENTRY LENGTH
1292	005124'	060400			ADD	R4,R0		: POINT AT NXT CODE TBL ENTRY
1293	005126'	000743			BR	20\$		: GO CK FOR NXT CODE
1294	005130'	042704	177770		40\$: BIC	#177770,R4		: ISOLATE I.D. NAME LENGTH + 1
1295	005134'	020416			CMP	R4,(SP)		: ENOUGH ROOM FOR NAME?
1296	005136'	101017			BHI	60\$		: Y,N-60\$
1297	005140'	060467	000050		ADD	R4,70\$		: ADJ MSG LENGTH FOR NAME
1298	005144'	005304			DEC	R4		: ADJ FOR BIT MASK CHAR
1299	005146'	005200			INC	R0		: POINT PAST BIT MASK
1300	005150'	021627	000100		CMP	(SP),#64.		: FIRST ERROR CODE IN MSG?
1301	005154'	001403			BEQ	50\$		: N,Y-50\$
1302	005156'	112721	000054		MOVB	#,(R1)+		: MOVE COMMA TO MSG
1303	005162'	005316			DEC	(SP)		: ADJ REMAINING ROOM IN MSG
1304	005164'	112021			50\$: MOVB	(R0)+,(R1)+		: MOVE ERROR CODE TO MSG
1305	005166'	005316			DEC	(SP)		: ADJ REMAINING ROOM IN MSG
1306	005170'	005304			DEC	R4		: MOVED ALL NAME CHARS?
1307	005172'	001374			BNE	50\$		: Y,N-50\$
1308	005174'	000720			BR	20\$		: GO CK FOR MORE ERROR BITS
1309	005176'	005004			60\$: CLR	R4		: SET USER MODE PRINT
1310	005200'	022627	000100		CMP	(SP)+,#64.		: ANY ERROR CODES PUT IN MSG?
1311	005204'	001404			BEQ	80\$		: Y,N-80\$
1312	005206'	004567	001226		JSR	R5,PRINT		: GO ISSUE ERROR BITS MSG
1313	005212'	002234			.WORD	DKEMSG-		
1314	005214'	000116			70\$: .WORD	78.		
1315	005216'	004567	000760		80\$: JSR	R5,DISPST		: DISPLAY DEVICE REG'S
1316	005222'	000000			ERSTAD: .WORD	XXXX		
1317	005224'	004767	001152		JSR	PC,PRTIWD		: DISPLAY TRACK, SECT VALUES
1318	005230'	016300	000022		ERRSNM: MOV	PSACST(R3),R0		: GET ADR OF SRC STMENTS
1319	005234'	111001			110\$: MOVB	(R0),R1		: SAVE STMT LENGTH
1320	005236'	026067	000004	173770	CMP	4(R0),ERRADR		: ERROR OCCUR ON THIS STMT?
1321	005244'	001402			BEQ	120\$		: N,Y-120\$



```

1322 005246' 060100          ADD    R1,R0          ;POINT AT NXT STMT
1323 005250' 000771          BR     110$          ;GO CK NXT STMT
1324 005252' 005720          120$: TST    (R0)+      ;SET UP ADR OF STMT # DATA
1325 005254' 010701          MOV    PC,R1        ;SET UP DATA OUTPUT ADR
1326 005256' 062701 002162    ADD    #STMNUM-,R1
1327 005262' 004577 172574    JSR    R5,DEASC     ;CONVERT IT TO ASCII
1328 005266' 012767 020040 002150  MOV    #20040,STMNUM+4 ;SET 2 LOW DIGITS TO SPACES
1329 005274' 004567 001140    JSR    R5,PRINT     ;ISSUE STMT # MSG
1330 005300' 002130          .WORD  STMNUM-
1331 005302' 177762          .WORD  -14
1332 005304' 012605          MOV    (SP)+,R5     ;RESTORE R5 & R4
1333 005306' 012604          MOV    (SP)+,R4
1334 005310' 000205          ERREX: RTS    R5    ;EXIT IN-LINE
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353

```

;ERROR MESSAGE CODE TABLE

```

;377 = GO TO NEXT DEVICE REGISTER WORD
;376 = GO TO NEXT DEVICE REGISTER BYTE
;BYTE 0 CONTAINS FLAG BITS & I.D. NAME LENGTH
;
;BITS 0-2 = LENGTH OF BIT MASK + I.D. NAME
;BIT 3 = NOT USED
;BIT 4 = NOT USED
;BIT 5 = 11/70 ONLY ERROR BIT
;BIT 6 = BIT = 0 IS AN ERROR CONDITION
;
;BYTE 1 IS THE BIT MASK
;
;BYTES 2 THRU ? ARE THE BIT'S ASCII I.D.

```

```

1354 005312' 100003 041523          ERCDTB: .ASCII <003><200>/SC/          ;RPCS1 - BYTE 1
1355 005316' 040004 051124 105      .ASCII <004><100>/TRE/
1356 005323' 005 046440 050103      .ASCII <005><040>/MCPE/
1357 005330' 105
1357 005331' 104 042010 040526      .ASCII <104><010>/DVA/
1358 005336' 376
1359 005337' 104 051200 054504      .ASCII <104><200>/RDY/
1360 005344' 377
1361 005345' 377
1362 005346' 377
1363 005347' 377
1364 005350' 100004 046104 124      .ASCII <004><200>/DLT/          ;RPCS2 - BYTE 1
1365 005355' 004 053500 042503      .ASCII <004><100>/WCE/
1366 005362' 020004 050125 105      .ASCII <004><040>/UPE/
1367 005367' 004 047020 042105      .ASCII <004><020>/NED/
1368 005374' 004004 042516 115      .ASCII <004><010>/NEM/
1369 005401' 004 050004 042507      .ASCII <004><004>/PGE/
1370 005406' 001004 054115 106      .ASCII <004><002>/MXF/
1371 005413' 005 046401 050104      .ASCII <005><001>/MDPE/
1372 005420' 105
1372 005421' 377
1373 005422' 100004 052101 101      .ASCII <004><200>/ATA/          ;RPDS - BYTE 1
1374 005427' 004 042500 051122      .ASCII <004><100>/ERR/
1375 005434' 020004 044520 120      .ASCII <004><040>/PIP/

```

1376	005441'	104	046420	046117	.ASCII	<104><020>/MOL/	
1377	005446'	000504	050104	122	.ASCII	<104><001>/DPR/	
1378	005453'	376			.BYTE	376	
1379	005454'	100104	051104	131	.ASCII	<104><200>/DRY/	;RPDS - BYTE 0
1380	005461'	377			.BYTE	377	
1381	005462'	100004	041504	113	.ASCII	<004><200>/DCK/	;RPER1 - BYTE 1
1382	005467'	004	052500	051516	.ASCII	<004><100>/UNS/	
1383	005474'	020004	050117	111	.ASCII	<004><040>/OPT/	
1384	005501'	004	042020	042524	.ASCII	<004><020>/DTE/	
1385	005506'	004004	046127	105	.ASCII	<004><010>/WLE/	
1386	005513'	004	044404	042501	.ASCII	<004><004>/IAE/	
1387	005520'	001004	047501	105	.ASCII	<004><002>/AOE/	
1388	005525'	376			.BYTE	376	
1389	005526'	004004	040520	122	.ASCII	<004><010>/PAR/	
1390	005533'	004	051004	051115	.ASCII	<004><004>/RMR/	
1391	005540'	001004	046111	122	.ASCII	<004><002>/ILR/	
1392	005545'	004	044401	043114	.ASCII	<004><001>/ILF/	
1393	005552'	377			.BYTE	377	
1394	005553'	376			.BYTE	376	
1395	005554'	100005	052101	033501	.ASCII	<005><200>/ATA7/	;RPAS - BYTE 0
1396	005562'	040005	052101	033101	.ASCII	<005><100>/ATA6/	
1397	005570'	020005	052101	032501	.ASCII	<005><040>/ATA5/	
1398	005576'	010005	052101	032101	.ASCII	<005><020>/ATA4/	
1399	005604'	004005	052101	031501	.ASCII	<005><010>/ATA3/	
1400	005612'	002005	052101	031101	.ASCII	<005><004>/ATA2/	
1401	005620'	001005	052101	030501	.ASCII	<005><002>/ATA1/	
1402	005626'	000405	052101	030101	.ASCII	<005><001>/ATA0/	
1403	005634'	377			.BYTE	377	
1404	005635'	377			.BYTE	377	
1405	005636'	377			.BYTE	377	
1406	005637'	377			.BYTE	377	
1407	005640'	377			.BYTE	377	
1408	005641'	377			.BYTE	377	
1409	005642'	100044	050101	105	.ASCII	<044><200>/APE/	;RPCS3 - BYTE 1 (11/70 ONLY)
1410	005647'	046	042100	042520	.ASCII	<046><100>/DPEOW/	
1411	005654'	053517					
1411	005656'	020046	050104	042505	.ASCII	<046><040>/DPEEW/	
1412	005664'	127					
1412	005665'	046	053420	042503	.ASCII	<046><020>/WCEOW/	
1413	005672'	053517					
1413	005674'	004046	041527	042505	.ASCII	<046><010>/WCEEW/	
1414	005702'	127					
1414	005703'	000			.BYTE	0	;TABLE TERMINATOR
1415					.EVEN		



```

1417          .SBTTL  SUBROUTINES FOR RS03/RS04 DEVICE ROUTINE
1418
1419
1420
1421          ;SAVE REGISTERS R0 THRU R5
1422
1423          ;JSR    RO,SAVREG          S/R CALL
1424
1425 SAVREG: MOV    R1,-(SP)            ;SAVE R0 THRU R5
1426        MOV    R2,-(SP)
1427        MOV    R3,-(SP)
1428        MOV    R4,-(SP)
1429        MOV    R5,-(SP)
1430        MOV    RO,PC              ;EXIT IN-LINE
1431
1432
1433          ;RESTORE REGISTERS R0 THRU R5
1434
1435          ;JSR    RO,RESREG          S/R CALL
1436
1437 RESREG: TST    (SP)+              ;RESTORE R5 THRU R0
1438        MOV    (SP)+,R5
1439        MOV    (SP)+,R4
1440        MOV    (SP)+,R3
1441        MOV    (SP)+,R2
1442        MOV    (SP)+,R1
1443        RTS    RO                  ;EXIT IN-LINE
1444
1445
1446          ;SET PROGRAM'S PROG TABLE ADR IN R3 & RPCS1 ADR IN R4
1447
1448          ;JSR    PC,SUPTAD          S/R CALL
1449
1450 SUPTAD: MOV    PC,R3              ;SET UP LOCATION ZERO ADR
1451        ADD    #LOCZ--,R3
1452        SUB    -2(R3),R3           ;SUBTRACT PROG TBL LENGTH
1453        MOV    DREGAD,R4         ;GET DEV REG BASE ADR
1454        RTS    PC                  ;EXIT IN-LINE
1455
1456
1457          ;STORE DEVICE'S STATUS REGISTERS
1458
1459          ;JSR    R5,STSTAT          S/R CALL
1460        ;.WORD  STADR-             REL STORAGE ADR
1461        ;DESTROYS R0,R1,R2
1462
1463 STSTAT: MOV    R5,R1              ;GET REL STORAGE ADR & MAKE
1464        ADD    (R5)+,R1           ;IT ABSOLUTE
1465        MOV    DREGAD,RO         ;GET DEV REG ADR
1466        MOV    RO,-(SP)         ;SET UP ADR OF RPCS2 REG
1467        ADD    #RPCS2,(SP)       ;FOR LATER USE
1468        MOV    #STSLUP,R2       ;SETUP TWO LOOP COUNTS
1469        BIT    #CPU70,ACSYSFW   ;RUNNING ON AN 11/70?
1470        BEQ    10$,             ;Y,N-10$
1471        ADD    #1000,R2         ;ALLOW FOR 2 MORE REGS
1472        MOV    (RO)+,(R1)+      ;STORE DEV REG

```

```

1473 006016' 105302          DECB   R2          ;FINISHED WITH THIS GROUP OF REGS?
1474 006020' 001375          BNE   10$         ;Y,N-10$
1475 006022' 000302          SWAB  R2          ;SET UP NEXT LOOP CNT
1476 006024' 001417          BEQ   30$         ;DONE 2 PASSES? (N,Y-30$)
1477 006026' 010746          MOV   PC, -(SP)  ;SET UP CURRENT STATUS
1478 006030' 062716 173100  ADD   #CSTAT-.,(SP) ;STORAGE ADR
1479 006034' 020126          CMP   R1,(SP)+  ;STORING STATUS FOR INTERRUPT?
1480 006036' 101005          BHI   15$         ;Y,N-15$
1481 006040' 032736 000200  BIT   #200,2(SP)+ ;OUTPUT READY SET IN RPCS2?
1482 006044' 001403          BEQ   20$         ;Y,N-20$
1483 006046' 012021          MOV   (R0)+,(R1)+ ;STORE RSDB CONTENTS
1484 006050' 000761          BR    10$        ;GO DO SECOND PASS
1485 006052' 005726          15$: TST   (SP)+  ;TAKE UNUSED ADR OFF STACK
1486 006054' 062700 000002  20$: ADD   #2,R0  ;BYPASS READ OF RPOB
1487 006060' 005021          CLR  (R1)+      ;SET ITS STORAGE TO 0'S
1488 006062' 000754          BR    10$        ;GO DO SECOND PASS
1489 006064' 000205          30$: RTS   R5    ;EXIT IN-LINE
1490
1491
1492          ;DISPLAY DEVICE I.D. & UNIT #
1493
1494          ;JSR   PC,DEVID      S/R CALL
1495
1496          ;R3 MUST CONTAIN PROG TBL ADR
1497          ;DESTROYS R0,R1,R2
1498
1499 006066' 012700 031460  DEVID: MOV   #'03,R0  ;INITIALIZE TO RS03
1500 006072' 032763 000020 000032  BIT   #20,PMDLCD(R3) ;IS IT AN RS04?
1501 006100' 001402          BEQ   10$        ;NO
1502 006102' 012700 032060  10$: MOV   #'04,R0  ;YES
1503 006106' 010067 000522          MOV   R0,UNITMG+6 ;TAILOR DEV ID MESSAGE
1504 006112' 012767 000026 000056  MOV   #26,DEVIML  ;INITIALIZE TO NORMAL MS
1505 006120' 116300 000035          MOVB  PCURDV(R3),R0 ;GET CURR UNIT #
1506 006124' 020027 000007          CMP   R0,#7     ;VALID UNIT #?
1507 006130' 101007          BHI   DEVIIV    ;Y,N-DEVIIV
1508 006132' 004577 171722          JSR   R5,2BTASLZ ;CONVERT # TO DECIMAL ASCII
1509 006136' 000514          .WORD UNASCII-
1510 006140' 016767 000512 000504  MOV   UNASCII+4,UNASCII ;MOVE ASCII # TO 1ST TWO DIGITS
1511 006146' 000410          BR    DEVIPR    ;GO ISSUE MSG
1512 006150' 012767 000032 000020  DEVIIV: MOV   #26,DEVIML ;SETUP ERROR COND MSG LNGTH
1513 006156' 042700 177400          BIC   #177400,R0 ;RESET HIGH BYTE
1514 006162' 004577 171670          JSR   R5,2BINASC ;CONVERT BINARY # TO ASCII
1515 006166' 000464          .WORD UNASCII-
1516 006170' 004567 000244  DEVIPR: JSR   R5,PRINT ;GO ISSUE UNIT # MSG
1517 006174' 000432          .WORD UNITMG-
1518 006176' 000026          DEVIML: .WORD 22.
1519 006200' 000207          RTS   PC        ;EXIT IN-LINE

```



```

1521                                     ;TAILOR STATUS MSG & PRINT IT
1522
1523                                     ;JSR   R5,DISPST           S/R CALL
1524                                     ;WORD  STATADR-         REL ADR OF STATUS DATA
1525                                     ;DESTROYS R0,R1,R2
1526
1527 006202' 010346          DISPST: MOV   R3,-(SP)           ;SAVE R3
1528 006204' 010503          MOV   R5,R3           ;GET REL DATA ADR
1529 006206' 062503          ADD   (R5)+,R3       ;MAKE IT ABS
1530 006210' 010546          MOV   R5,-(SP)       ;SAVE R5
1531 006212' 010705          MOV   PC,R5          ;SET UP ADR OF REG NAMES IN ASCII
1532 006214' 062705          ADD   #DVRGMS-.,R5
1533 006220' 012746          MOV   #REGNUM-2,-(SP) ;SETUP # OF REG TO DISPL
1534 006224' 032777          BIT   #CPU70,#CSYSFW ;RUNNING ON AN 11/70?
1535 006232' 001402          BEQ   10$           ;Y,N-10$
1536 006234' 062716          ADD   #2,(SP)       ;MAKE IT 14 REGISTERS
1537 006240' 012700          MOV   #3,R0         ;SET UP 3 REG LOOP CNT
1538 006244' 010701          MOV   PC,R1         ;POINT AT REG NAME IN MSG
1539 006246' 062701          ADD   #DVRGMG-.,R1
1540 006252' 012521          MOV   (R5)+,(R1)+   ;MOVE REG NAME TO MSG
1541 006254' 012521          MOV   (R5)+,(R1)+
1542 006256' 005725          TST   (R5)+         ;POINT TO NEXT NAME
1543 006260' 062701          ADD   #10.,R1       ;POINT TO NEXT FIELD IN MSG
1544 006264' 005300          DEC   R0            ;DONE 3 REGS?
1545 006266' 001371          BNE   15$           ;Y,N-15$
1546 006270' 012300          MOV   (R3)+,R0      ;CONVERT OCTAL REGISTER CONTENTS
1547 006272' 004577          JSR   R5,#BINASC    ;FOR 3 REGISTERS TO ASCII
1548 006276' 000370          .WORD DVRDT1-      ;AND PLACE IN THE MSG
1549 006300' 012300          MOV   (R3)+,R0
1550 006302' 004577          JSR   R5,#BINASC
1551 006306' 000376          .WORD DVRDT2-
1552 006310' 012300          MOV   (R3)+,R0
1553 006312' 004577          JSR   R5,#BINASC
1554 006316' 000404          .WORD DVRDT3-
1555 006320' 012767          MOV   #40.,30$     ;INITIALIZE MSG LENGTH TO 3 REGS
1556 006326' 162716          SUB   #3,(SP)       ;DECR REGISTER CNT
1557 006332' 100005          BPL   25$           ;< 3 REGS? (Y,N-25$)
1558 006334' 162767          SUB   #14.,30$     ;SHORTEN MSG LENGTH BY 1 REG
1559 006342' 005216          INC   (SP)          ;INCR NEG REG CNT
1560 006344' 100773          BMI   20$           ;CNT BACK TO 0? (Y,N-20$)
1561 006346' 010346          MOV   R3,-(SP)     ;SAVE REG DATA PNTR
1562 006350' 016603          MOV   6(SP),R3     ;RESTORE PROG TBL ADR
1563 006354' 004567          JSR   R5,PRINT     ;GO PRINT THE MSG
1564 006360' 000300          .WORD DVRGMG-
1565 006362' 000050          .WORD 40.
1566 006364' 012603          MOV   (SP)+,R3     ;RESTORE REG DATA PNTR
1567 006366' 005716          TST   (SP)         ;MORE REGS TO GO?
1568 006370' 001323          BNE   10$           ;N,Y-10$
1569 006372' 005726          TST   (SP)+        ;REMOVE CNT FROM STACK
1570 006374' 012605          MOV   (SP)+,R5     ;RESTORE R5 & R3
1571 006376' 012603          MOV   (SP)+,R3
1572 006400' 000205          RTS   R5           ;EXIT IN-LINE

```

```

1574 ;DISPLAY TRACK/SECT WORDS' VALUES
1575
1576 ;JSR PC,PRTIWD S/R CALL
1577 ;DESTROYS R0,R1,R2
1578
1579 PRTIWD:
1580 006402' 016700 171400 MOV HEAD,R0 ;GET & CONVERT TRACK VALUE
1581 006406' 004577 171444 JSR R5,@BINASC
1582 006412' 000771 .WORD IFHEAD-.
1583 006414' 016700 171370 MOV SECT,R0 ;GET & CONVERT SECT VALUE
1584 006420' 004577 171432 JSR R5,@BINASC
1585 006424' 000775 .WORD IFSECT-.
1586 006426' 004567 000006 JSR R5,PRINT ;PRINT MSG WITH THEIR VALUES
1587 006432' 000743 .WORD INFOMG-.
1588 006434' 000031 .WORD 25.
1589 006436' 000207 RTS PC ;EXIT IN-LINE
1590
1591 ;ISSUE MSG TO LIST DEVICE SUBROUTINE
1592
1593 ;JSR R5,PRINT S/R CALL
1594 ;.WORD MSGADR-. REL ADR OF MSG
1595 ;.WORD BYTCNT MSG BYTE CNT (IF NEGATIVE,
1596 ; RESET PRT DEV DEDICATED.)
1597 ;R3 = PROG TBL ADR
1598 ;R4 = FLAGWORD -- IF NEGATIVE, USE CMND MODE PRINT
1599 ;DESTROYS R0,R1,R2
1600
1601 PRINT:
1602 006440' 010500 MOV R5,R0 ;GET MSG ADR & MAKE IT ABS
1603 006442' 062500 ADD (R5)+,R0
1604 006444' 012501 MOV (R5)+,R1 ;GET BYTE COUNT
1605 006446' 005704 TST R4 ;USE CMND MODE PRINT?
1606 006450' 100030 BPL 40$ ;Y,N-40$
1607 006452' 010702 MOV PC,R2 ;SET UP LINK INFO ADR
1608 006454' 062702 000040 ADD #20$--,R2
1609 006460' 160200 SUB R2,R0 ;MAKE MSG ADR REL
1610 006462' 010022 MOV R0,(R2)+ ;STORE MSG ADR
1611 006464' 010112 MOV R1,(R2) ;STORE MSG'S BYTE COUNT
1612 006466' 100001 BPL 10$ ;CNT NEG? (Y,N-10$)
1613 006470' 005412 NEG (R2) ;MAKE IT POSITIVE
1614 006472' 016367 000006 000056 10$: MOV PASCIN(R3),PROGM ;STORE PROG'S # IN MSG
1615 006500' 004577 171350 JSR R5,@CLIST ;ISSUE PROG #
1616 006504' 000050 .WORD PNMMMSG-.
1617 006506' 000005 .WORD 5
1618 006510' 004577 171340 JSR R5,@CLIST ;ISSUE MSG SPECIFIED
1619 006514' 000000 20$: .WORD XXXX
1620 006516' 000000 .WORD XXXX
1621 006520' 004577 171330 JSR R5,@CLIST ;ISSUE A <CR> & <LF>
1622 006524' 000257 .WORD CRLF-.
1623 006526' 000002 .WORD 2
1624 006530' 000410 BR PRTEX ;GO TO EXIT
1625 006532' 010067 000010 40$: MOV R0,50$ ;STORE MSG'S ABS ADR
1626 006536' 010167 000006 MOV R1,60$ ;STORE ITS BYTE CNT
1627 006542' 004577 171304 JSR R5,@CLIST ;GO TO MPG TO ISSUE THE MSG
1628 006546' 000000 50$: .WORD XXXX
1629 006550' 000000 60$: .WORD XXXX

```



K03

MAINDEC-11-DTRSA-A RH11/RH70 - RS03/RS04 DEVICE ROUTINE FOR MPG  
DTRSA.A.P11 SUBROUTINES FOR RS03/RS04 DEVICE ROUTINE

MACY11 27(732) 24-SEP-76 14:12 PAGE 12-4

SEQ 0100

1630 006552' 000205

PRTEX: RTS R5

;EXIT IN-LINE

```

1632 .SBTTL RS03/RS04 MESSAGE STORAGE AREA
1633
1634
1635 .NLIST BEX
1636
1637 .EVEN
1638 006554' 021520 PNMMSG: .ASCII /P#/
1639 006556' 054130 011 PROGMM: .ASCII /XX/<011>
1640 006561' 101 020124 040514 ATIMSG: .ASCII /AT LAST INT:/
1641 006575' 103 051125 042522 CURMSG: .ASCII /CURRENTLY:/
1642 006607' 105 042116 047440 RENDMG: .ASCII /END OF REPORT/<15><12>
1643 .EVEN
1644 006626' 025052 025052 051522 UNITMG: .ASCII /****RSXX DISK UNIT: /
1645 006652' 054130 054130 054130 UNASCI: .ASCII /XXXXXX/
1646 .EVEN
1647 006660' 054130 054130 020075 DVRGMG: .ASCII /XXXX= /
1648 006666' 054130 054130 054130 DVRDT1: .ASCII /XXXXXX XXXX= /
1649 006704' 054130 054130 054130 DVRDT2: .ASCII /XXXXXX XXXX= /
1650 006722' 054130 054130 054130 DVRDT3: .ASCII /XXXXXX/
1651 006730' 054502 042524 035123 CNTSMG: .ASCII /BYTES: RD= /
1652 006744' 054130 054130 054130 BCMRD: .ASCII /XXXXXXXXXXXXX WR = /
1653 006767' 130 054130 054130 BCMWR: .ASCII /XXXXXXXXXXXXX/
1654 007003' 015 012 CRLF: .ASCII <015><012>
1655 007005' 011 041411 036513 .ASCII <011><011>/CK= /
1656 007013' 130 054130 054130 BCMCK: .ASCII /XXXXXXXXXXXXX/<015><012><011>/CMNDS: RD= /
1657 007046' 054130 054130 054130 CMDCRD: .ASCII /XXXXXX WR = /
1658 007062' 054130 054130 054130 CMDCWR: .ASCII /XXXXXX CK = /
1659 007077' 130 054130 054130 CMDCCK: .ASCII /XXXXXX/<015><012><011><011>/SK= /
1660 007115' 130 054130 054130 CMDCSK: .ASCII /XXXXXX CLR= /
1661 007131' 130 054130 054130 CMDDRV: .ASCII /XXXXXX DCLR= /
1662 007146' 054130 054130 054130 CMDCMS: .ASCII /XXXXXX/<015><012><011>/ERRORS: DEV= /
1663 007174' 054130 054130 054130 CNTCEC: .ASCII *XXXXXX DATA/OPR= *
1664 007215' 130 054130 054130 CNTDER: .ASCII /XXXXXX/<015><012><011>/RETRY: DLT= /
1665 007243' 130 054130 054130 CNTDLT: .ASCII /XXXXXX DTE= /
1666 007257' 130 054130 054130 CNTHCE: .ASCII /XXXXXX DCK= /
1667 007273' 130 054130 054130 CNTDCK: .ASCII /XXXXXX/<015><012><011><011>/WCE= /
1668 007312' 054130 054130 054130 CNTWCE: .ASCII /XXXXXX/<015><012><011>/TOTAL RETRY: /
1669 007341' 130 054130 054130 CNTRTY: .ASCII /XXXXXX/<015><012><011>/INTERRUPTS: /
1670 007367' 130 054130 054130 CNTINT: .ASCII /XXXXXX/
1671 007375' 007375' CNTSEN= .
1672 007375' 124 040522 036513 INFOMG: .ASCII /TRAK= /
1673 007403' 130 054130 054130 IFHEAD: .ASCII /XXXXXX SECT= /
1674 007421' 130 054130 054130 IFSECT: .ASCII /XXXXXX/
1675 .EVEN
1676 007430' 052123 047115 020124 STMNMG: .ASCII /STMNT # /
1677 007440' 054130 054130 054130 STMNUM: .ASCII /XXXXXX/
1678 007446' 051105 047522 020122 DKEMSG: .ASCII /ERROR BITS: /<015><012><011>
1679 007464' 000100 CODFLD: .BLKB 64.
1680 007564' 042502 047506 042522 BEFIO: .ASCII 'BEFORE ISSUING I/O CMND:'
1681 007614' 043101 042524 020122 AFTIO: .ASCII 'AFTER ISSUING I/O CMND:'
1682 007643' 124 047457 047440 CRT0: .ASCIZ 'T/O ON CRESET'
1683 007661' 124 046511 047505 IOTO: .ASCIZ 'TIMEOUT ON I/O'
1684 007700' 047516 026516 054105 NONEXD: .ASCIZ 'NON-EXISTENT DRIVE'
1685 007723' 125 051516 043101 INITUS: .ASCII 'UNSAFE'
1686 007732' 051105 047522 020122 INITDE: .ASCIZ 'ERROR ON INITIATION'
1687 007756' 044504 045523 044440 OFFLIN: .ASCIZ 'DISK IS OFF-LINE'

```



1688	007777'	116	047117	044455	NOITER: .ASCII	'NON-INT '
1689	010007'	111	047457	052040	IOTERM: .ASCIZ	'I/O TERMINATION ERROR'
1690	010035'	111	052116	053440	NOATA: .ASCIZ	'INT WITHOUT ATA'
1691	010055'	125	042516	050130	UXPATA: .ASCIZ	'UNEXP ATA COND'
1692	010074'	054105	040510	051525	RTYEXH: .ASCIZ	'EXHAUSTED RETRIES'
1693	010116'	047111	020126	047125	INVDVN: .ASCIZ	'/INV UNIT #/'
1694		010132'			.EVEN	
1695					.LIST	BEX
1696						
1697	010132'				DVREND= .	

```

1699          .SBTTL FORMATS FOR PROGRAM & DEVICE ROUTINE TABLES
1700
1701          ; PROGRAM TABLE FORMAT
1702
1703          000242 PTLGTH= 162. ;PROGRAM TABLE LENGTH - NON MEM MGMNT VERSION OF MPG
1704
1705          ;(PTLGTH= 212. ;PROGRAM TABLE LENGTH - MEM MGMNT VERSION OF MPG)
1706
1707          000000 PFLGWD= +0. ;PROGRAM FLAG WORD - 1 WORD
1708
1709          000002 URSTOP= 2 ; 1 = USER HAS STOPPED THIS PROGRAM
1710          000004 ERSTOP= 4 ; 1 = AN ERROR HAS STOPPED THIS PROGRAM
1711          000010 WT4IOT= 10 ; 1 = WAITING FOR I/O TERMINATION
1712          000020 CTPRIO= 20 ; 1 = CONSOLE OR PRINTER I/O IN PROGRESS
1713          000040 SETDED= 40 ; 1 = THIS PROG SET THE PRT DEV DEDICATED FLAG
1714          000100 OCPRES= 100 ; 1 = OBJ CODE IS PRESENT
1715          000200 USEUBM= 200 ; 1 = THIS PROG USES THE UNIBUS MAP (MEM MGMNT ONLY)
1716          100000 ACTIVE= 100000 ; 1 = PROGRAM IS ACTIVE (SPECIFIED FOR EXECUTION)
1717
1718          000002 POPSW= +2. ;PROGRAM'S OPERATION SWITCHES - 1 WORD
1719
1720          100000 STONER= 100000 ; 1 = STOP PROG EXECUTION UPON ERROR
1721          040000 CYCPRG= 40000 ; 1 = CYCLE PROGRAM (ON CURRENT DEVICE)
1722          020000 PRONER= 20000 ; 1 = DO NOT PRINT ON ERROR
1723          010000 BIT12= 10000 ; 0 = NOT USED
1724          004000 BIT11= 4000 ; 0 = NOT USED
1725          002000 CYCDVL= 2000 ; 1 = CYCLE THE DEVICE LIST
1726          001000 GTNXTD= 1000 ; 1 = CYCLE ON SAME DEVICE UPON ERROR
1727          000400 DOERCK= 400 ; 1 = DON'T DO ERROR CHECKING
1728          000200 SPOPER= 200 ; 1 = DEVICE SPECIAL OPERATION
1729          000100 BIT6= 100 ; 0 = NOT USED
1730          000040 DOIOT= 40 ; 1 = DO NOT PERFORM I/O TIMEOUT
1731          000020 AUTORP= 20 ; 1 = DO NOT AUTOMATICALLY DISPLAY COUNTS
1732          000010 AURPEP= 10 ; 1 = AUTO DISPLAY COUNTS AT END OF FINAL PASS ONLY
1733          000004 HSKPEP= 4 ; 1 = HOUSEKEEP COUNTS ONLY AT RUN COMMAND
1734          000002 PFBBOV= 2 ; 1 = PRINT FIRST BAD BYTE ONLY ON VERIFY
1735          000001 NOCOMP= 1 ; 1 = DO NOT PRINT PROG COMPLETED MSG
1736
1737          000004 PFWADR= +4. ;*;PROGRAM FLAGWORD ADDRESS - 1 WORD
1738
1739          000006 PASCIN= +6. ;PROGRAM'S NUMBER IN ASCII - 1 WORD
1740
1741          000010 PNAME= +8. ;PROGRAM'S NAME IN ASCII - 6 BYTES
1742
1743          000016 PRDIOA= +14. ;ADDRESS OF READ I/O AREA - 1 WORD
1744
1745          000020 PWRIOA= +16. ;ADDRESS OF WRITE I/O AREA - 1 WORD
1746
1747          000022 PSRCST= +18. ;SOURCE STATEMENTS START ADDRESS - 1 WORD
1748
1749          000024 POBJST= +20. ;OBJECT CODE START ADDRESS - 1 WORD
1750
1751          000026 PLNGTH= +22. ;PROG AREA LENGTH (OBJ END MINUS PROG TBL START) - 1 WORD
1752
1753          000030 PTOCNT= +24. ;I/O TIMEOUT COUNT - 1 WORD
1754

```



1755	000032	PMDLCD= +26.	;DEV ROUT MODEL # CODE - 1 WORD
1756			
1757	000034	PDPNTR= +28.	;CURRENT DEVICE NUMBER POINTER - 1 BYTE
1758			
1759	000035	PCURDV= +29.	;CURRENT DEVICE # - 1 BYTE
1760			
1761	000036	PDNUMS= +30.	;DEVICE NUMBERS - 16 BYTES
1762			
1763	000056	PTEM0= +46.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1764			
1765	000060	PTEM1= +48.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1766			
1767	000062	PTEM2= +50.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1768			
1769	000064	PTEM3= +52.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1770			
1771	000066	PTEM4= +54.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1772			
1773	000070	PTEM5= +56.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1774			
1775	000072	PTEM6= +58.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1776			
1777	000074	PTEM7= +60.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1778			
1779	000076	PTEM8= +62.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1780			
1781	000100	PTEM9= +64.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1782			
1783	000102	PTEM10= +66.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1784			
1785	000104	PTEM11= +68.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1786			
1787	000106	PTEM12= +70.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1788			
1789	000110	PTEM13= +72.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1790			
1791	000112	PTEM14= +74.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1792			
1793	000114	PTEM15= +76.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1794			
1795	000116	PNBR= +78.	;NUMBER OF BYTES TO TRANSFER ON MOVE (NBR) - 1 WORD
1796			
1797	000120	PSRC= +80.	;DATA SOURCE ADDRESS ON MOVE (SRC) - 1 WORD
1798			
1799	000122	PDST= +82.	;DATA DESTINATION ADDRESS ON MOVE (DST) - 1 WORD
1800			
1801	000124	PSTKCT= +84.	;# OF WORDS (X 2) SAVED OFF STACK - 1 WORD
1802			
1803	000126	PSTKSV= +86.	;STACK WORDS STORAGE AREA - 30 WORDS
1804			
1805	000222	PSVREG= +146.	;USER'S R0 THRU R5 REGISTERS STORAGE AREA - 6 WORDS
1806			
1807	000236	PUSRPC= +158.	;USER'S CURRENT PROGRAM COUNTER - 1 WORD
1808			

1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834

;FOLLOWING ENTRIES (PRDIOX THRU PUBMAP) ARE ONLY IN MEM MGMNT VERSION

;(PRDIOX= +160. ;18/22 BIT ABSOLUTE ADDRESS OF READ I/O AREA - 2 WORDS)

;(PRDIOV= +164. ;18 BIT VIRTUAL ADDRESS OF READ I/O AREA - 2 WORDS)

;(PWRIOX= +168. ;18/22 BIT ABSOLUTE ADDRESS OF WRITE I/O AREA - 2 WORDS)

;(PWRIOV= +172. ;18 BIT VIRTUAL ADDRESS OF WRITE I/O AREA - 2 WORDS)

;(PUPARS= +176. ;STORAGE AREA FOR USER'S PAR'S 0 THRU 7 - 8 WORDS)

;(PUPDRS= +192. ;STORAGE AREA FOR USER'S PDR'S 0 THRU 7 - 8 WORDS)

;(PUBMAP= +208. ;1ST UNIBUS MAP REG # AND # OF REGS USED - 1 WORD)

;END OF MEM MGMNT ONLY ENTRIES

000240

PTSIZE= +160. ;PROGRAM TABLE SIZE IN BYTES - 1 WORD - NON MEM MGMNT

;(PTSIZE= +210. ;PROGRAM TABLE SIZE IN BYTES - 1 WORD - MEM MGMNT VERSION)

000242

PTEND= +162. ;END OF PROGRAM TABLE - NON MEM MGMNT VERSION

;(PTEND= +212. ;END OF PROGRAM TABLE - MEM MGMNT VERSION)



```

1836           ;      DEVICE ROUTINE TABLE
1837
1838           000116      DRTLTH= 78.      ;DEVICE ROUTINE TABLE LENGTH
1839           ;
1840           000000      DEVRSZ= +0.      ;DEVICE ROUTINE SIZE IN BYTES - 1 WORD
1841           ;
1842           000002      DEVFWD= +2.      ;DEVICE ROUTINE FLAGWORD - 1 WORD
1843           ;
1844           000004      DEVIW1= +4.      ;DEVICE INTERFACE WORD # 1 - 1 WORD
1845           ;
1846           000006      DEVIW2= +6.      ;DEVICE INTERFACE WORD # 2 - 1 WORD
1847           ;
1848           000010      DEVIW3= +8.      ;DEVICE INTERFACE WORD # 3 - 1 WORD
1849           ;
1850           000012      DEVIW4= +10.     ;DEVICE INTERFACE WORD # 4 - 1 WORD
1851           ;
1852           000014      DEVIW5= +12.     ;DEVICE INTERFACE WORD # 5 - 1 WORD
1853           ;
1854           000016      DEVIW6= +14.     ;DEVICE INTERFACE WORD # 6 - 1 WORD
1855           ;
1856           000020      DEVIW7= +16.     ;DEVICE INTERFACE WORD # 7 - 1 WORD (SIZE)
1857           ;
1858           000022      DEVIW8= +18.     ;DEVICE INTERFACE WORD # 8 - 1 WORD (ERR)
1859           ;
1860           000024      DEVDRA= +20.     ;DEVICE REGISTERS ADDRESS - 1 WORD
1861           ;
1862           000026      DEVIVA= +22.     ;DEVICE INTERRUPT VECTOR ADDRESS - 1 WORD
1863           ;
1864           000030      DEVRPS= +24.     ;DEVICE READ PROCESSOR STATUS WORD (BUS REQ) - 1 WORD
1865           ;
1866           000032      DEVWPS= +26.     ;DEVICE WRITE PROC STATUS WORD (BUS REQ) - 1 WORD
1867           ;
1868           000034      DHKPAD= +28.     ;DEVICE ROUT HOUSEKEEPING ROUT REL ENTRY ADR - 1 WORD
1869           ;
1870           000036      DERPAD= +30.     ;DEVICE ROUT REPORT ROUT REL ENTRY ADR - 1 WORD
1871           ;
1872           000040      DKILAD= +32.     ;DEVICE ROUT KILL ROUTINE REL ENTRY ADR - 1 WORD
1873           ;
1874           000042      DECTAD= +34.     ;DEVICE ROUT ERROR COUNTER REL ADR - 1 WORD
1875           ;
1876           000044      DTOEAD= +36.     ;DEVICE ROUT TIMEOUT ERR ROUT REL ENTRY ADR - 1 WORD
1877           ;
1878           000046      DEVI0B= +38.     ;DEVICE I/O BUSY BRANCH ADDRESS (CIOBSY) - 1 WORD
1879           ;
1880           000050      DEVDER= +40.     ;DEVICE ERROR BRANCH ADDRESS (CUPGER) - 1 WORD
1881           ;
1882           000052      DVUPRT= +42.     ;USER MODE PRINT BRANCH ADDRESS (ULIST) - 1 WORD
1883           ;
1884           000054      DVCPRT= +44.     ;CMND MODE PRINT BRANCH ADDRESS (CLIST) - 1 WORD
1885           ;
1886           000056      DEVBTA= +46.     ;CONVERT BINARY TO ASCII BR ADR (BINASC) - 1 WORD
1887           ;
1888           000060      DV3TDA= +48.     ;CONVERT BINARY TO DECIMAL ASCII BR ADR (BTASLZ) - 1 WORD
1889
1890
1891

```

1892	000062	DVPDTA= +50.	; CONVERT PACKED DECIMAL TO ASCII BR ADR (DECASC) - 1 WORD
1893			
1894	000064	DVSFWD= +52.	; MPG SYSTEM FLAGWORD ADDRESS (CSYSFW) - 1 WORD
1895			
1896	000066	DVSVEC= +54.	; SET INTERRUPT VECTOR BR ADR (SETVEC) - 1 WORD
1897			
1898	000070	DVCVEC= +56.	; CLEAR INTERRUPT VECTOR BR ADR (CLRVEC) - 1 WORD
1899			
1900	000072	DVTVEC= +58.	; TEST INTERRUPT VECTOR BR ADR (TSTVEC) - 1 WORD
1901			
1902	000074	DVRINT= +60.	; RETURN FROM INTERRUPT BR ADR (RTNINT) - 1 WORD
1903			
1904	000076	DVGETB= +62.	; GET DATA BYTE BR ADR (GETBYT) - 1 WORD
1905			
1906	000100	DVPUTB= +64.	; PUT DATA BYTE BR ADR (PUTBYT) - 1 WORD
1907			
1908	000102	DEVSTP= +66.	; DEVICE ROUT REL SYMBOL TABLE POINTER - 1 WORD
1909			
1910	000104	DEVETP= +68.	; DEVICE ROUT REL ENTRY TABLE POINTER - 1 WORD
1911			
1912	000106	DVPTEP= +70.	; PACK TABLE EXTEN. REL POINTER - 1 WORD
1913			
1914	000110	DVVTEP= +72.	; VECTOR TABLE EXTEN. REL POINTER - 1 WORD
1915			
1916	000112	DVCTEP= +74.	; COMPILER TBL EXTEN. REL POINTER - 1 WORD
1917			
1918	000114	DVIWSP= +76.	; DEVICE INTERFACE WORD SYMBOL TBL REL POINTER - 1 WORD
1919			
1920	000116	DRTEND= +78.	; END OF DEVICE ROUTINE TABLE
1921			
1922			
1923	000001	.END	



ACQCAD	003056R	002	CNTDLT	007243R	002	DEVIW8=	000022	002	EBSTAT	005016R	002	MSFMT5	001072R	002
ACQERC	003060R	002	CNTEND=	001234R	002	DEVVPS=	000030	002	ERCDTB	005312R	002	MYATA	003120R	002
ACQERR	003046R	002	CNTHCE	007257R	002	DEVRSZ=	000000	002	ERMBAS	004762R	002	NED =	010000	
ACQGDK	003070R	002	CNTINT	007367R	002	DEVSTP=	000102	002	ERR =	040000		NOATA	010035R	002
ACQHSK	002462R	002	CNTNUM=	000024	002	DEVWPS=	000032	002	ERRADR	001234R	002	NOCOMP=	000001	
ACQRTY	002464R	002	CNTRTY	007341R	002	DFLGWD	000002R	002	ERRCNT	001214R	002	NOITER	007777R	002
ACTIVE=	100000		CNTSEN=	007375R	002	DHKPAD=	000034	002	ERRCOM	004622R	002	NONEXD	007700R	002
AFTIO	007614R	002	CNTSMG	006730R	002	DISCNT	001564R	002	ERRCS	004562R	002	NOWAIT	002212R	002
ANYIOI=	000200		CNTWCE	007312R	002	DISPST	006202R	002	ERRCSI	004570R	002	NRSEC =	000100	
APORT	002222R	002	CODFLD	007464R	002	DKEMSG	007446R	002	ERREX	005310R	002	NRTRK =	000100	
ATA =	100000		CORFLG=	002000	002	DKILAD=	000040	002	ERRFND	003756R	002	OCPRES=	000100	
ATATBL	003100R	002	COUNTS	001164R	002	DLTCNT	001220R	002	ERRI	000022R	002	ODD	002242R	002
ATIMSG	006561R	002	CPU70 =	000010	002	DOERCK=	000400	002	ERRIS	004606R	002	OFFLIN	007756R	002
AURPEP=	000010		CRESET	002302R	002	DOIOT =	000040	002	ERRSNM	005230R	002	OTHATA	003122R	002
AUTORP=	000020		CRLF	007003R	002	DOTERM=	000002	002	ERSTAD	005222R	002	PARITY=	000020	
BAI =	000010		CRT0	007643R	002	DPR =	000400	002	ERSTOP=	000004	002	PASCIN=	000006	
BAIOFF	002272R	002	CSTAT	001130R	002	DREGAD	000024R	002	EVEN	002252R	002	PATCH	001264R	002
BAION	002262R	002	CSYSFW	000064R	002	DRESET	002352R	002	FINCNT	001252R	002	PC =%	000007	
BCMCK	007013R	002	CTLCNT	001210R	002	DRTEND=	000116	002	GETBYT	000076R	002	PCURDV=	000035	
BCMRO	006744R	002	CTPRIO=	000020	002	DRTLTH=	000116	002	GO =	000001	002	PDNUMS=	000036	
BCMWR	006767R	002	CUPGER	000050R	002	DRVcnt	001212R	002	GOTDSK	002630R	002	PDPNTR=	000034	
BEFIO	007564R	002	CURADR	001244R	002	DTECNT	001222R	002	GTNXTD=	001000	002	PDST =	000122	
BINASC	000056R	002	CURCMD	001242R	002	DTOEAD=	000044	002	HARDER	004176R	002	PFBBOV=	000002	
BIT11 =	004000		CURCNT	001250R	002	DVA =	004000	002	HEAD	000006R	002	PFLGWD=	000000	
BIT12 =	010000		CURFLG	001240R	002	DVBTD=	000060	002	HSKEEP	001334R	002	PFWADR=	000004	
BIT6 =	000100		CURMSG	006575R	002	DVCMD5	000242R	002	HSKPEN=	001260R	002	PLNGTH=	000026	
BPORT	002232R	002	CURRTY	001254R	002	DVCPRT=	000054	002	HSKPEP=	000004	002	PMDLCD=	000032	
BTASLZ	000060R	002	CYCDVL=	002000	002	DVCPTE	000654R	002	HSKPST=	001074R	002	PNAME =	000010	
BYCK	001174R	002	CYCPRG=	040000	002	DVCTEP=	000112	002	IE =	000100	002	PNR =	000116	
BYRD	001164R	002	DATAER	001216R	002	DVCVEC=	000070	002	IFHEAD	007403R	002	PNMSG	006554R	002
BYWR	001170R	002	DCCODE=	000010	002	DVGETB=	000076	002	IFSECT	007421R	002	POBJST=	000024	
CIOSY	000046R	002	DCKCNT	001224R	002	DVIWSP=	000114	002	INFOMG	007375R	002	POPSW =	000002	
CKCNT	001204R	002	DECASC	000062R	002	DVIWST	001032R	002	INITDE	007732R	002	PORT =	002000	
CKCORR	004216R	002	DECTAD=	000042	002	DVMVTE	000554R	002	INITUS	007723R	002	PRDIOA=	000016	
CKOSY	004230R	002	DERPAD=	000036	002	DVPDTA=	000062	002	INTCNT	001232R	002	PRINT	006440R	002
CKRTRY	004016R	002	DEVBTA=	000056	002	DVPKTE	000354R	002	INTEAD	004454R	002	PROCEX	004474R	002
CKSC	003654R	002	DEVDER=	000050	002	DVPTEP=	000106	002	INTEX	003644R	002	PROCTM	004360R	002
CLIST	000054R	002	DEVORA=	000024	002	DVPUTB=	000100	002	INVDVN	010116R	002	PROGNM	006556R	002
CLR =	000040		DEVETP=	000104	002	DVRDT1	006666R	002	IOERR =	000001	002	PRONER=	020000	
CLRVEC	000070R	002	DEVFWD=	000002	002	DVRDT2	006704R	002	IOTERM	010007R	002	PRTEX	006552R	002
CLWTF	003622R	002	DEVID	006066R	002	DVRDT3	006722R	002	IOTO	007661R	002	PRTIWD	006402R	002
CMDCCK	007077R	002	DEVIIV	006150R	002	DVREGE=	000242R	002	ISTAT =	001074R	002	PS =	177776	
CMDCMS	007146R	002	DEVIML	006176R	002	DVREGS	000116R	002	ITIME =	072460	002	PSEL =	002000	
CMDCOM	003256R	002	DEVI08=	000046	002	DVREND=	010132R	002	IVCTAD	000026R	002	PSRC =	000120	
CMDCRD	007046R	002	DEVIPR	006170R	002	DVREX	001664R	002	JSETER	004212R	002	PSRCST=	000022	
CMDCSK	007115R	002	DEVIVA=	000026	002	DVRGMG	006660R	002	KILL	002024R	002	PSTKCT=	000124	
CMDCWR	007062R	002	DEVIW1=	000004	002	DVRINT=	000074	002	KILLEX	002054R	002	PSTKSV=	000126	
CMDDRV	007131R	002	DEVIW2=	000006	002	DVSFWD=	000064	002	LOCZ	000000R	002	PSVREG=	000222	
CMDISU=	000100		DEVIW3=	000010	002	DVSVEC=	000066	002	MCPE =	020000	002	PSWD	000030R	002
CNTADR	001236R	002	DEVIW4=	000012	002	DVTVEC=	000072	002	MMVER =	000001	002	PTEM0 =	000056	
CNTCEC	007174R	002	DEVIW5=	000014	002	DVUPRT=	000052	002	MOL =	010000	002	PTEM1 =	000060	
CNTDCK	007273R	002	DEVIW6=	000016	002	DVVTEP=	000110	002	MSFMT1	001064R	002	PTEM10=	000102	
CNTDER	007215R	002	DEVIW7=	000020	002	EBSBAS	005014R	002	MSFMT2	001065R	002	PTEM11=	000104	

PTEM12= 000106	RENDMG 006607R	002	RPMR = 000024		SKCNT 001206R	002	TVECTX 004560R	002
PTEM13= 000110	REPORT 001406R	002	RPTBAS 001630R	002	SP =%000006		ULIST 000052R	002
PTEM14= 000112	REPTBL 001674R	002	RPTEND 001654R	002	SPOPER= 000200		UNASCI 006652R	002
PTEM15= 000114	RESREG 005720R	002	RPTLP 001612R	002	STEPDN 002142R	002	UNITMG 006626R	002
PTEM2 = 000062	RETRYS 001230R	002	RPWC = 000002		STEPA1 002162R	002	UNS = 040000	
PTEM3 = 000064	RHPINT 003546R	002	RTNINT 000074R	002	STEPD2 002150R	002	URSTOP= 000002	
PTEM4 = 000066	RINTEX 004530R	002	RTRY 000012R	002	STEPEX 002106R	002	USEUBM= 000200	
PTEM5 = 000070	RINTV 004506R	002	RTRYIP 001256R	002	STEPUP 002056R	002	USMTPS= 000002	
PTEM6 = 000072	RPAS = 000016		RTYEXH 010074R	002	STEPU1 002076R	002	UXPATA 010055R	002
PTEM7 = 000074	RPBA = 000004		R0 =%000000		STEPU2 002064R	002	WAIT 002172R	002
PTEM8 = 000076	RPBAE = 000030		R1 =%000001		STMNMG 007430R	002	WAITMD= 100000	
PTEM9 = 000100	RPCS1 = 000000		R2 =%000002		STMNUM 007440R	002	WCCODE= 000050	
PTEND = 000242	RPCS1V 001260R	002	R3 =%000003		STONER= 100000		WCECNT 001226R	002
PTLGH= 000242	RPCS2 = 000010		R4 =%000004		STPCOM 002120R	002	WCODE = 000060	
PTOCNT= 000030	RPCS2V 001262R	002	R5 =%000005		STSLUP= 001011		WRCK 003204R	002
PTSIZE= 000240	RPCS3 = 000032		SAVREG 005704R	002	STSTAT 005756R	002	WRCNT 001202R	002
PUSRPC= 000236	RPDA = 000006		SCODE = 000030		SUIORG 003440R	002	WRITE 003154R	002
PUTBYT 000100R	RPDB = 000022		SEARCH 003234R	002	SUPTAD 005736R	002	WT4IOT= 000010	
PWRIOA= 000020	RPDS = 000012		SECT 000010R	002	TOUTER 001744R	002	XXXX = 000000	
RCODE = 000070	RPDT = 000026		SETDED= 000040		TRAK = 000006R	002	. = 010132R	002
. ABS. 000000		000						
		001						
RJP11 010132		002						

ERRORS DETECTED: 0  
DEFAULT GLOBALS GENERATED: 0

\*,DTRSAA/NL:TOC/DOC=DTRSAA.P11  
RUN-TIME: 5 12 1 SECONDS  
RUN-TIME RATIO: 24/19=1.2  
CORE USED: 5K (9 PAGES)

DOCUMENT PAGES: 45



