

RK06

DEVICE ROUTINE (MFG)
MD-11-DTR6A-A

EP-DTR6A-A-DL-A
COPYRIGHT © 1976

NOV 1976
digital
MADE IN U.S.A.

The left side of the page contains a grid of 50 small, illegible tables or diagrams arranged in 10 rows and 5 columns. Each cell in the grid appears to contain a small table with multiple columns and rows of text, but the text is too small and faded to be legible. The tables are organized in a regular grid pattern, with some cells containing what might be flowcharts or diagrams alongside the text.

.REM %

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DTR6A-A
PRODUCT NAME: RK611 - RK06 DISK 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-DTR6A-A
DTR6AA.P11

CO1

MAINDEC-11-DTR6A-A RK611 - RK06 DEVICE ROUTINE FOR MPG
DTR6AA.P11 REVISION HISTORY

MACY11 27(732) 24-SEP-76 14:11 PAGE 2

SEQ 0002

44
45
46

.SBTTL REVISION HISTORY

;
;

JUL 76 DTR6A-A INITIAL RELEASE AS A FULL SUPPORT
DEVICE ROUTINE FOR THE RK06 DISK.

MAINDEC-11-DTR6A-A RK611 - RK06 DEVICE ROUTINE FOR MPG
DTR6AA.P11 STANDARD DEVICE ROUTINE TABLE

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

.SBTTL STANDARD DEVICE ROUTINE TABLE
.TITLE MAINDEC-11-DTR6A-A RK611 - RK06 DEVICE ROUTINE FOR MPG
;REVISION 'A'
;FILENAME OF "TR6AAD.MPG" ON MPG/XODP MEDIA
;MACY11: DTR6AA,DTR6AA/CRF:SYM/DOC=DTR6AA.P11
;LNKX11: DTR6AA.MPG/B:D=DTR6AA/E
;PAPER TAPE: PUNCH DTR6AA.MPG/FILE:ELEV

000000'

.CSECT RJP11
.DSABL GBL

::EQUIVALENT STATEMENTS

:PROGRAM EQUALS
BUSMAP=170200
ITIME=30000.
NRWORD=256.
NRBYTE=2*NRWORD
NRCYL=411.
NRHEAD=3
PS=177776
REGNUM=16.
SDMAX=3
STSLUP=2412
XXXX=0

:UNIBUS ADDR OF BUS-MAP REGS
:INTERRUPT TIME-COUNT
:NR WORDS IN ONE SECTOR
:NR BYTES IN ONE SECTOR
:NR OF CYLINDERS
:NR OF HEADS
:PSW
:NR OF DEVICE REGISTERS
:HIGHEST VALUE OF SELECT-DRIVE CODE
:LOOP CONTROL FOR "STSTAT" ROUTINE
:VALUE TO BE TAYLORED BY DEVICE ROUTINE

:DEVICE REGISTGER NAMES

RPCS1=0
RPMC=2
RPGA=4
RPGA=6
RPCS2=10
RPOS=12
RPER=14
RPER1=RPER
RPS=16
RPDC=20
RKNL=22
RPOB=24
RPMR1=26
RPEC1=30
RPEC2=32
RPMR2=34
RPMR3=36

:RKCS1 DEVICE BIT EQUATES

CCLR=100000
DI=40000
SPAR=20000
CFMT=10000

100000
040000
020000
010000

MAINDEC-11-DTR6A-A RK611 - RK06 DEVICE ROUTINE FOR MPG
DTR6AA.P11 STANDARD DEVICE ROUTINE TABLE

104	004000	CTO=4000
105	000200	RDY=200
106	000100	IE=100
107	000001	GO=1
108		
109		:RKCS2 DEVICE BIT EQUATES
110	100000	DLT=100000
111	040000	MCE=40000
112	020000	UPE=20000
113	010000	NED=10000
114	004000	NEH=4000
115	002000	PGE=2000
116	001000	MDS=1000
117	000400	UFE=400
118	000200	OR=200
119	000040	SCLR=40
120	000020	BAI=20
121	000010	RLS=10
122		
123		:RKDS DEVICE BIT EQUATES
124	100000	SVAL=100000
125	040000	DSC=40000
126	004000	WRL=4000
127	000200	DRDY=200
128	000100	VV=100
129	000040	DROT=40
130	000020	DSL=20
131	000010	ACLO=10
132	000001	DRA=1
133		
134		:RKER DEVICE BIT EQUATES
135	100000	DCX=100000
136	040000	UNS=40000
137	020000	OPI=20000
138	010000	DTE=10000
139	004000	MLE=4000
140	002000	IDAE=2000
141	001000	COE=1000
142	000400	HVAC=400
143	000200	BSE=200
144	000100	ECH=100
145	000040	DTYE=40
146	000020	FHTE=20
147	000010	DRPAR=10
148	000004	NXF=4
149	000002	SKI=2
150	000001	ILF=1
151		
152		:RKMR1 DEVICE BIT EQUATES
153	000020	PAT=20
154		
155		:SYSTEM FLAG-WORD BIT DEFINITIONS
156	000001	HWVER=1
157	000002	USMTPS=2
158	000010	CPU70=10
159	000040	UNIMAP=40

160			
161		:DISK COMMAND CODES	
162	000001	SDCODE=1	
163	000003	PACODE=3	
164	000005	DCCODE=5	
165	000007	UCODE=7	
166	000011	SSCODE=11	
167	000013	RCCODE=13	
168	000015	OCODE=15	
169	000017	SCODE=17	
170	000021	RCODE=21	
171	000023	MCODE=23	
172	000025	RHCODE=25	
173	000027	MHCODE=27	
174	000031	MCCODE=31	
175			
176		:DEVICE ROUTINE (DFLGND) FLAG BITS	
177	100000	WAITMD=100000	:WAIT MODE, 0=WAIT
178	002000	CORFLG=2000	:CORRECTION MODE, 0=CORON
179	000200	ANYIOI=200	:ANY I/O HAS BEEN ISSUED
180	000100	CHDISU=100	:I/O COMMAND HAS BEEN ISSUED
181	001000	VVFLG= 1000	:VV MODE - 0 = VVON
182	000400	SPINFL= 400	:SPIN COMMAND IN PROGRESS
183	000040	SNOVTO= 40	:TIMEOUT ON SWITCH OVER
184	000020	SNOIER= 20	:ERROR ON INT FOR SWITCH OVER
185	000010	SNOVER= 10	:ACQUIRING DISK - SWITCH OVER
186	000002	DOTERM=2	:PROCESS I/O TERMINATION
187	000001	IOERR=1	:ERROR ON CURRENT I/O
188			
189		:MEMORY MANAGEMENT EQUATES	
190	172350	KPAR4=172350	
191	172310	KPDR4=172310	
192	100000	P4CONS=100000	
193	077406	PDRCON=077406	
194			
195		:DIAGNOSTIC FLAG EQUATES	
196	000001	DRIB=1	:DRIVE IS BUSY
197	000002	NEE=2	:NO ERRORS EXPECTED
198	000004	NDC=4	:NEED DRIVE CLEAR
199	000010	NRC=10	:NEED RE-CALIBRATE
200	000020	NCC=20	:NEED CONTROLLER/DRIVE CLEAR
201	000040	PBSH=40	:PRINT BAD SECTOR HEADING

203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249

000000' 015544
000002' 000000
000004' 000000
000006' 000000
000010' 000000
000012' 000003
000014' 000000
000016' 000000
000020' 000001
000022' 000000
000024' 177440
000026' 000210
000030' 000240
000032' 000000
000034' 002042
000036' 002122
000040' 002730
000042' 001700
000044' 002456
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' 000152
000106' 000346
000110' 000714
000112' 001076
000114' 001406

MPG INTERFACE

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

LOCZ: .WORD DVREND-.
DFLGND: .WORD 0
CYL: .WORD 0
HEAD: .WORD 0
SECT: .WORD 0
RTRY: .WORD 3
MSGA: .WORD 0
MSGB: .WORD 0
SIZE: .WORD 1
ERRI: .WORD 0
DREGAD: .WORD 177440
IVCTAD: .WORD 210
PSMD: .WORD 240

CIBSY: .WORD 0
CUPGER: .WORD 0
ULIST: .WORD 0
CLIST: .WORD 0
BINASC: .WORD 0
BTASLZ: .WORD 0
DECASC: .WORD 0
CSYSFM: .WORD 0
SETVEC: .WORD 0
CLRVEC: .WORD 0
TSTVEC: .WORD 0
RTNINT: .WORD 0
GETBYT: .WORD 0
PUTBYT: .WORD 0

.WORD DVREGS--
.WORD DVCMD5--
.WORD DVPKTE--
.WORD DVMVTE--
.WORD DVCPT5--
.WORD DVIWST--

:DEVICE ROUT SIZE IN BYTES
:DEVICE ROUT FLAGWORD
:CYLINDER # (0 THRU 410.)
:HEAD # (0 THRU 2.)
:SECTOR # (0 THRU 21./19.)
:# OF RETRY ATTEMPTS
:SAVE MSG A
:SAVE MSG B
:# 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
:CMD 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

.SBTTL COMPILER TABLES & CONSTANT AREAS

COMPILER TABLES & CONSTANT AREAS

```

100000 000116 045522 030503
100001 000122 000000
100002 000124 045522 041527
100003 000130 000002
100004 000132 045522 040502
100005 000136 000004
100006 000140 045522 040504
100007 000144 000006
100008 000146 045522 031103
100009 000152 000010
100010 000154 045522 051504
100011 000160 000012
100012 000162 045522 051105
100013 000166 000014
100014 000170 045522 051501
100015 000174 000016
100016 000176 045522 041504
100017 000202 000020
100018 000204 047516 052524
100019 000210 000022
100020 000212 045522 041104
100021 000216 000024
100022 000220 045522 030515
100023 000224 000026
100024 000226 045522 047520
100025 000232 000030
100026 000234 045522 040520
100027 000240 000032
100028 000242 045522 031115
100029 000246 000034
100030 000250 045522 031515
100031 000254 000036
100032 000256 000256
100033 000256 120 201
100034 000260 004130
100035 000262 130 201
100036 000264 004156
100037 000266 376 000
100038 000270 003154
100039 000272 375 000
100040 000274 003130
100041 000276 374 000
100042 000300 001660
100043 000302 373 000
100044 000304 001654
100045 000306 372 000
100046 000310 004446
100047 000312 371 000

```

```

DVREGS: .ASCII /RKC1/ ;VALID DEVICE REGISTER NAMES &
         .WORD 0 ;THEIR POSITIONS RELATIVE TO
         .ASCII /RKWC/
         .WORD 2
         .ASCII /RKBA/
         .WORD 4
         .ASCII /RKDA/
         .WORD 6
         .ASCII /RKC2/
         .WORD 10
         .ASCII /RKDS/
         .WORD 12
         .ASCII /RKER/
         .WORD 14
         .ASCII /RKAS/
         .WORD 16
         .ASCII /RKDC/
         .WORD 20
         .ASCII /NOTU/
         .WORD 22
         .ASCII /RKDB/
         .WORD 24
         .ASCII /RKM1/
         .WORD 26
         .ASCII /RKPO/
         .WORD 30
         .ASCII /RKPA/
         .WORD 32
         .ASCII /RKM2/
         .WORD 34
         .ASCII /RKM3/
         .WORD 36
DVREG= .

```

```

DVCMDS: .BYTE 120,201 ;VALID DEVICE FUNCTIONS
         .WORD READ-
         .BYTE 130,201 ;FLAG BYTE:
         .WORD WRITE-.
         .BYTE 376,0 ;BIT 7 = NPR DEV
         .WORD NOWAIT-. ;BIT 3 = MASSBUS DEV
         .BYTE 375,0 ;BIT 0 = 2 WORDS FOR ADR
         .WORD WAIT-. ; (18 BIT ADRS)
         .BYTE 374,0
         .WORD REPORT-.
         .BYTE 373,0
         .WORD REPORT-.
         .BYTE 372,0
         .WORD SEEK-.
         .BYTE 371,0

```


MAINDEC-11-DTR6A-A RK611 - RK06 DEVICE ROUTINE FOR MPG
DTR6AA.P11 COMPILER TABLES & CONSTANT AREAS

307	000314'	004510				.WORD	SELDRI-
308	000316'	370	201			.BYTE	370,201
309	000320'	004406				.WORD	WRCK-
310	000322'	367	000			.BYTE	367,0
311	000324'	004356				.WORD	RDMD-
312	000326'	366	201			.BYTE	366,201
313	000330'	004364				.WORD	WRMD-
314	000332'	365	000			.BYTE	365,0
315	000334'	003220				.WORD	CRESET-
316	000336'	364	000			.BYTE	364,0
317	000340'	003342				.WORD	DRESET-
318	000342'	363	000			.BYTE	363,0
319	000344'	003260				.WORD	SRESET-
320	000346'	362	000			.BYTE	362,0
321	000350'	002452				.WORD	STEPUP-
322	000352'	361	000			.BYTE	361,0
323	000354'	002620				.WORD	STEPON-
324	000356'	360	000			.BYTE	360,0
325	000360'	004420				.WORD	OFFSET-
326	000362'	357	000			.BYTE	357,0
327	000364'	003332				.WORD	SPIN-
328	000366'	356	000			.BYTE	356,0
329	000370'	003320				.WORD	UNLOAD-
330	000372'	355	000			.BYTE	355,0
331	000374'	004416				.WORD	RECAL-
332	000376'	354	000			.BYTE	354,0
333	000400'	003324				.WORD	PAKACK-
334	000402'	353	000			.BYTE	353,0
335	000404'	075160				.WORD	BADSEC-
336	000406'	352	000			.BYTE	352,0
337	000410'	003322				.WORD	REL-
338	000412'	347	000			.BYTE	347,0
339	000414'	003120				.WORD	FMT22-
340	000416'	346	000			.BYTE	346,0
341	000420'	003124				.WORD	FMT20-
342	000422'	345	000			.BYTE	345,0
343	000424'	003030				.WORD	000-
344	000426'	344	000			.BYTE	344,0
345	000430'	003034				.WORD	EVEN-
346	000432'	337	000			.BYTE	337,0
347	000434'	003040				.WORD	BAION-
348	000436'	336	000			.BYTE	336,0
349	000440'	003044				.WORD	BAIOFF-
350	000442'	335	000			.BYTE	335,0
351	000444'	003050				.WORD	CORON-
352	000446'	334	000			.BYTE	334,0
353	000450'	003054				.WORD	COROFF-
354	000452'	177777				.WORD	177777
355							
356	000454'	047516	040527	052111	DVPKTE:	.ASCII	/NOWAIT/
357	000456'	376	000			.BYTE	376,0
358	000464'	010040	040527	052111		.ASCII	/WAIT/
359	000472'	375	000			.BYTE	375,0
360	000474'	052123	052101	051525		.ASCII	/STATUS/
361	000502'	374	000			.BYTE	374,0
362	000504'	047503	047125	051524		.ASCII	/COUNTS/

;TABLE TERMINATOR

;PACK TABLE EXTENSION

MAINDEC-11-DTR6A-A RK611 - RK06 DEVICE ROUTINE FOR MPG
DTR6AA.P11 COMPILER TABLES & CONSTANT AREAS

363	000512'	373	000		.BYTE	373,0
364	000514'	020040	042523	045505	.ASCII	/SEEK/
365	000522'	372	000		.BYTE	372,0
366	000524'	042523	042114	044522	.ASCII	/SELDRI/
367	000532'	371	000		.BYTE	371,0
368	000534'	020040	051127	045503	.ASCII	/WRCK/
369	000542'	370	000		.BYTE	370,0
370	000544'	020040	042122	042110	.ASCII	/RDHD/
371	000552'	367	000		.BYTE	367,0
372	000554'	020040	051127	042110	.ASCII	/WRHD/
373	000562'	366	000		.BYTE	366,0
374	000564'	051103	051505	052105	.ASCII	/CRESET/
375	000572'	365	000		.BYTE	365,0
376	000574'	051104	051505	052105	.ASCII	/DRESET/
377	000602'	364	000		.BYTE	364,0
378	000604'	051123	051505	052105	.ASCII	/SRESET/
379	000612'	363	000		.BYTE	363,0
380	000614'	052123	050105	050125	.ASCII	/STEPUP/
381	000622'	362	000		.BYTE	362,0
382	000624'	052123	050105	047104	.ASCII	/STEPDN/
383	000632'	361	000		.BYTE	361,0
384	000634'	043117	051506	052105	.ASCII	/OFFSET/
385	000642'	360	000		.BYTE	360,0
386	000644'	020040	050123	047111	.ASCII	/SPIN/
387	000652'	357	000		.BYTE	357,0
388	000654'	047125	047514	042101	.ASCII	/UNLOAD/
389	000662'	356	000		.BYTE	356,0
390	000664'	051040	041505	046101	.ASCII	/RECAL/
391	000672'	355	000		.BYTE	355,0
392	000674'	040520	040513	045503	.ASCII	/PAKACK/
393	000702'	354	000		.BYTE	354,0
394	000704'	040502	051504	041505	.ASCII	/BADSEC/
395	000712'	353	000		.BYTE	353,0
396	000714'	020040	051040	046105	.ASCII	/REL/
397	000722'	352	000		.BYTE	352,0
398	000724'	043040	052115	031062	.ASCII	/FMT22/
399	000732'	347	000		.BYTE	347,0
400	000734'	043040	052115	030062	.ASCII	/FMT20/
401	000742'	346	000		.BYTE	346,0
402	000744'	020040	047440	042104	.ASCII	/000/
403	000752'	345	000		.BYTE	345,0
404	000754'	020040	053105	047105	.ASCII	/EVEN/
405	000762'	344	000		.BYTE	344,0
406	000764'	041040	044501	047117	.ASCII	/BAION/
407	000772'	337	000		.BYTE	337,0
408	000774'	040502	047511	043106	.ASCII	/BAIOFF/
409	001002'	336	000		.BYTE	336,0
410	001004'	041440	051117	047117	.ASCII	/CORON/
411	001012'	335	000		.BYTE	335,0
412	001014'	047503	047522	043106	.ASCII	/COROFF/
413	001022'	334	000		.BYTE	334,0
414						
415	001024'	000376	001570		DVMVTE: .WORD	376,MSFMT1-LOCZ
416	001030'	000375	001570		.WORD	375,MSFMT1-LOCZ
417	001034'	000374	001570		.WORD	374,MSFMT1-LOCZ
418	001040'	000373	001570		.WORD	373,MSFMT1-LOCZ

;MODEL VECTOR TABLE EXTEN.

419	001044'	000372	001570	.WORD	372,MSFMT1-LOCZ
420	001050'	000371	001605	.WORD	371,MSFMT5-LOCZ
421	001054'	000370	001571	.WORD	370,MSFMT2-LOCZ
422	001060'	000367	001576	.WORD	367,MSFMT3-LOCZ
423	001064'	000366	001605	.WORD	366,MSFMT5-LOCZ
424	001070'	000365	001570	.WORD	365,MSFMT1-LOCZ
425	001074'	000364	001570	.WORD	364,MSFMT1-LOCZ
426	001108'	000363	001570	.WORD	363,MSFMT1-LOCZ
427	001104'	000362	001605	.WORD	362,MSFMT5-LOCZ
428	001110'	000361	001605	.WORD	361,MSFMT5-LOCZ
429	001114'	000360	001605	.WORD	360,MSFMT5-LOCZ
430	001120'	000357	001570	.WORD	357,MSFMT1-LOCZ
431	001124'	000356	001570	.WORD	356,MSFMT1-LOCZ
432	001130'	000355	001570	.WORD	355,MSFMT1-LOCZ
433	001134'	000354	001570	.WORD	354,MSFMT1-LOCZ
434	001140'	000353	001570	.WORD	353,MSFMT1-LOCZ
435	001144'	000352	001570	.WORD	352,MSFMT1-LOCZ
436	001150'	000347	001570	.WORD	347,MSFMT1-LOCZ
437	001154'	000346	001570	.WORD	346,MSFMT1-LOCZ
438	001160'	000345	001570	.WORD	345,MSFMT1-LOCZ
439	001164'	000344	001570	.WORD	344,MSFMT1-LOCZ
440	001170'	000337	001570	.WORD	337,MSFMT1-LOCZ
441	001174'	000336	001570	.WORD	336,MSFMT1-LOCZ
442	001200'	000335	001570	.WORD	335,MSFMT1-LOCZ
443	001204'	000334	001570	.WORD	334,MSFMT1-LOCZ

COMPILER TABLE EXTENSION

444	001210'	003	376	DVCPT:	.BYTE	3,376	;NO WAIT
445	001212'	004537	000012		.WORD	4537,10.	
446	001216'	003	375		.BYTE	3,375	;WAIT
447	001220'	004537	000012		.WORD	4537,10.	
448	001224'	004	374		.BYTE	4,374	;STATUS
449	001226'	004537	000012	001002	.WORD	4537,10.,1002	
450	001234'	004	373		.BYTE	4,373	;COUNTS
451	001236'	004537	000012	001001	.WORD	4537,10.,1001	
452	001244'	003	372		.BYTE	3,372	;SEEK
453	001246'	004537	000012		.WORD	4537,10.	
454	001252'	004	371		.BYTE	4,371	;SELECT DRIVE
455	001254'	004537	000012	000000	.WORD	4537,10.,0	
456	001262'	006	370		.BYTE	6,370	;WRITE CHECK DATA
457	001264'	004537	000012	000000	.WORD	4537,10.,0,2,2	
458	001272'	000002	000002				
459	001276'	005	367		.BYTE	5,367	;READ HEADER
460	001300'	004537	000012	000000	.WORD	4537,10.,0,2	
461	001306'	000002					
462	001310'	006	366		.BYTE	6,366	;WRITE HEADER
463	001312'	004537	000012	000204	.WORD	4537,10.,132.,2,2	
464	001320'	000002	000002				
465	001324'	003	365		.BYTE	3,365	;CONTROL RESET
466	001326'	004537	000012		.WORD	4537,10.	
467	001332'	003	364		.BYTE	3,364	;DRIVE RESET
468	001334'	004537	000012		.WORD	4537,10.	
469	001340'	003	363		.BYTE	3,363	;SUBSYSTEM RESET
470	001342'	004537	000012		.WORD	4537,10.	

472	001346'	004	362		.BYTE	4,362		;STEP UP
473	001350'	004537	000012	000000	.WORD	4537,10.,0		
474	001356'	004	361		.BYTE	4,361		;STEP DOWN
475	001360'	004537	000012	000000	.WORD	4537,10.,0		
476	001366'	004	360		.BYTE	4,360		;OFFSET
477	001370'	004537	000012	000000	.WORD	4537,10.,0		
478	001376'	003	357		.BYTE	3,357		;START SPINDLE
479	001400'	004537	000012		.WORD	4537,10.		
480	001404'	003	356		.BYTE	3,356		;UNLOAD
481	001406'	004537	000012		.WORD	4537,10.		
482	001412'	003	355		.BYTE	3,355		;RECALIBRATE
483	001414'	004537	000012		.WORD	4537,10.		
484	001420'	003	354		.BYTE	3,354		;PACK ACKNOWLEDGE
485	001422'	004537	000012		.WORD	4537,10.		
486	001426'	003	353		.BYTE	3,353		;PRINT LIST OF BAD SECTORS
487	001430'	004537	000012		.WORD	4537,10.		
488	001434'	003	352		.BYTE	3,352		;RELEASE
489	001436'	004537	000012		.WORD	4537,10.		
490	001442'	003	347		.BYTE	3,347		;FORMAT 22
491	001444'	004537	000012		.WORD	4537,10.		
492	001450'	003	346		.BYTE	3,346		;FORMAT 20
493	001452'	004537	000012		.WORD	4537,10.		
494	001456'	003	345		.BYTE	3,345		;ODD
495	001460'	004537	000012		.WORD	4537,10.		
496	001464'	003	344		.BYTE	3,344		;EVEN
497	001466'	004537	000012		.WORD	4537,10.		
498	001472'	003	337		.BYTE	3,337		;BAI ON
499	001474'	004537	000012		.WORD	4537,10.		
500	001500'	003	336		.BYTE	3,336		;BAI OFF
501	001502'	004537	000012		.WORD	4537,10.		
502	001506'	003	335		.BYTE	3,335		;CORRECTION ON
503	001510'	004537	000012		.WORD	4537,10.		
504	001514'	003	334		.BYTE	3,334		;CORRECTION OFF
505	001516'	004537	000012		.WORD	4537,10.		

DEVICE INTERFACE WORD SYMBOL TABLE

507								
508								
509								
510	001522'	054503	020114	DVIWST:	.ASCII	/CYL /		
511	001526'	000004			.WORD	DEVIW1		
512	001530'	042510	042101		.ASCII	/HEAD/		
513	001534'	000006			.WORD	DEVIW2		
514	001536'	042523	052103		.ASCII	/SECT/		
515	001542'	000010			.WORD	DEVIW3		
516	001544'	052122	054522		.ASCII	/RTRY/		
517	001550'	000012			.WORD	DEVIW4		
518	001552'	051515	040507		.ASCII	/MSGA/		
519	001556'	000014			.WORD	DEVIW5		
520	001560'	051515	041107		.ASCII	/MSGB/		
521	001564'	000016			.WORD	DEVIW6		
522	001566'	177777			.WORD	177777		;END OF TABLE

MODEL STATEMENT TABLE EXTENSION

523								
524								
525								
526								
527	001570'	000		MSFMT1:	.BYTE	0		;NO OPERAND

MAINDEC-11-DTR6A-A RK611 - RK06 DEVICE ROUTINE FOR MPG
DTR6AA.P11 COMPILER TABLES & CONSTANT AREAS

```

528 001571' 377 052101 000377 MSFMT2: .ASCIZ <377>/AT/<377>
529 001576' 044777 052116 177517 MSFMT3: .ASCIZ <377>/INTO/<377>
    001604' 000
    001605' 377 000 MSFMT5: .BYTE 377,0 ;ONE OPERAND
    001610' .EVEN
;DEVICE ROUTINE CONSTANTS
    001610'
    001610' MSKIPST= .
    000000 ISTAT= . ;STORAGE FOR DEV REG'S AT INT
    001610' 000000 ICS1: .WORD 0 ;STORAGE FOR DISK REGISTERS
    001612' 000000 IRKWC: .WORD ; AS SAVED AT INTERRUPT TIME
    001614' 000000 IRKBA: .WORD
    001616' 000000 .WORD
    001620' 000000 ICS2: .WORD
    001622' 000000 IRKDS: .WORD
    001624' 000000 IRKER: .WORD
    001626' 000000 IRKAS: .WORD
    001630' 000000 000000 .WORD 0,0
    001634' 000000 IRKDB: .WORD
    001636' 000000 .WORD
    001640' 000000 IRKPO: .WORD
    001642' 000000 IRKPA: .WORD
    001644' 000000 000000 .WORD 0,0
    001650' 000020 CSTAT: .BLKW 16. ;DEV REG CURRENT VALUES STORAGE
    001710' 000000 COUNTS:
    001710' 000000 BYRD: .WORD 0 ;BYTES READ COUNT
    001712' 000000 .WORD
    001714' 000000 BYWR: .WORD ;BYTES WRITTEN COUNT
    001716' 000000 .WORD
    001720' 000000 BYCK: .WORD ;BYTES CHECKED COUNT
    001722' 000000 .WORD
    001724' 000000 RDCNT: .WORD ;READ CMD COUNT
    001726' 000000 WRCNT: .WORD ;WRITE CMD COUNT
    001730' 000000 CKCNT: .WORD ;CHECK CMD COUNT
    001732' 000000 SKCNT: .WORD ;SEEK/SEARCH CMD COUNT
    001734' 000000 MISCNT: .WORD ;MISC. CMD COUNT
    001736' 000000 ERRCNT: .WORD ;DEVICE ERRORS COUNT
    001740' 000000 CECCER: .WORD ;CORRECTABLE ECC ERRORS
    001742' 000000 DATAER: .WORD ;DATA/OPERATOR ERRORS COUNT
    001744' 000000 DLTCNT: .WORD ;DATA LATE ERRORS
    001746' 000000 DTECNT: .WORD ;DRIVE TIMING ERRORS
    001750' 000000 HRCNT: .WORD ;HEADER VRC ERRORS
    001752' 000000 FERCNT: .WORD ;FORMAT ERRORS
    001754' 000000 DCKCNT: .WORD ;DATA CHECK ERRORS
    001756' 000000 WCECNT: .WORD ;WRITE CHECK ERRORS
    001760' 000000 RETRYS: .WORD ;# OF RETRIES ON I/O CMDS
    001762' 000000 INTCNT: .WORD 0 ;INTERRUPTS COUNT
    001764' CNTEND= . ;END OF COUNT TABLE
    000026 CNTNUM=CNTEEND-COUNTS/2 ;NR OF COUNTERS
    001764' 000000 ERRADR: .WORD 0 ;CURR ADR IN USER PROG
    001766' 000000 CNTADR: .WORD 0 ;ADR OF BYTE COUNT TOTALS

```

583	001770'	000000	CURFLG: .WORD	0	:FLAG WORD OF CURR CMND
584	001772'	000000	CURCMD: .WORD	0	:CURR CMND CODE
585	001774'	000000	CURADR: .WORD	0	:CURR BUS ADDRESS
586	001776'	000000			
587	002000'	000000	CURCNT: .WORD	0	:NEG WORD CNT FOR CURR CMND
588	002002'	000000	CURPBC: .WORD	0	:POSITIVE BYTE CNT FOR CURR CMND
589	002004'	000000	FINCNT: .WORD	0	:FINAL WORD CNT (RPMC)
590	002006'	000000	CURRTY: .WORD	0	:CURR RETRY COUNT
591	002010'	000000	RTRYIP: .WORD	0	:RETRY IN PROGRESS FLAG
592	002012'	000000	CURPSW: .WORD	0	:PSW STORAGE AREA
593		002014'	HSKPEN= .		
594					
595	002014'	000000	RPCS1V: .WORD	0	:BASE VALUE FOR RPCS1 REG
596	002016'	000000	RPCS2V: .WORD	0	:BASE VALUE FOR RPCS2 REG
597	002020'	000000	RPMR1V: .WORD	0	:BASE VALUE FOR RPMR1 REG
598	002022'	000000	DIAFLG: .WORD	0	
599	002024'	000000	LUPCNT: .WORD	0	:COUNT LOOPS
600	002026'		PATCH: .REPT	20.	:PATCH AREA
601			.WORD	0	
602			.ENDR		

604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659

.SBTTL RK06 SUPPORT ROUTINES ENTERED FROM MPG

...SUPPORT ROUTINES ENTERED FROM MPG

;DEVICE ROUTINE HOUSEKEEPING

```

:JSR   RS,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 RPS1 VALUE
:INITIALIZE RPS2 VALUE
:INITIALIZE RPMR1 VALUE
:INITIALIZE DIAGNOSTIC FLAGS
: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

```

```

175706 HSKEEP: MOV      #3,RTRY \
                CLR      RPS1V
                CLR      RPS2V
                CLR      RPMR1V
                CLR      DIAFLG
                TST      (RS)+
                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      RS

```

;RK06 REPORT ROUTINE

```

:JSR   RS,REPORT          S/R CALL
:WORD  FLAGWORD          FLAGWORD
                               BIT 15 = CHND MODE CALL
                               BIT  9 = PROG STMT CALL
                               BIT  1 = DO STATUS REPORT
                               BIT  0 = DO COUNTS REPORT

: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

```

```

010106 REPORT: JSR      R0,SAVREG
010134        JSR      PC,SUPTAD
                MOV      (RS),R4
                BIT      @2,R4
                BEQ      5$
                JSR      RS,STSTAT
                WORD     CSTAT-
5$:  BIT      @177776,R4
                BNE      15$
                MOV      PC,R0
                ADD      @COUNTS-,R0
                MOV      @CNTNUM,R1
10$: TST      (R0)+
                BNE      15$
                DEC      R1

```

660	002234'	001374			BNE	10\$:CK'ED ALL WORDS? (Y,N-10\$)
661	002236'	000477			BR	DVREX		:GO TO EXIT -- ALL CNTS ARE 0'S
662	002240'	004767	010174	15\$:	JSR	PC,DEVID		:DISPLAY DEVICE T.D.
663	002244'	032704	000002		BIT	#2,R4		:DOING STATUS DISPLAY?
664	002250'	001432			BEQ	DISCNT		:Y,N-DISCNT
665	002252'	010700			MOV	PC,R0		:SET UP ADR OF REG'S AT
666	002254'	062700	177334		ADD	#1STAT-,R0		:LAST INT
667	002260'	012701	000020		MOV	#REGNUM,R1		:SET UP # OF REG'S
668	002264'	005720		20\$:	TST	(R0)+		:ALL REG'S = 0?
669	002266'	001003			BNE	30\$:N,Y-40\$
670	002270'	005301			DEC	R1		
671	002272'	001374			BNE	20\$		
672	002274'	000407			BR	40\$		
673	002276'	004567	010462	30\$:	JSR	RS,PRINT		:ISSUE 'AT LAST INT' MSG
674	002302'	010765			.WORD	ATMSG-		
675	002304'	000014			.WORD	12		
676	002306'	004567	010216		JSR	RS,DISPST		:GO DISPLAY STATUS AT LAST INT
677	002312'	177276			.WORD	1STAT-		
678	002314'	004567	010444	40\$:	JSR	RS,PRINT		:ISSUE 'CURRENTLY' MSG
679	002320'	010763			.WORD	CLMSG-		
680	002322'	000012			.WORD	10		
681	002324'	004567	010200		JSR	RS,DISPST		:GO DISPLAY CURRENT STATUS
682	002330'	177320			.WORD	CSTAT-		
683	002332'	004767	010356		JSR	PC,PTIWD		:GO DISPLAY INFO WORDS
684	002336'	032704	000001	DISCNT:	BIT	#1,R4		:DISPLAY COUNTS?
685	002342'	001431			BEQ	RPTEND		:Y,N-RPTEND
686	002344'	012700	000026		MOV	#CNTNUM,R0		:SET UP # OF WORDS
687	002350'	010701			MOV	PC,R1		:SET UP ADR OF CNTS
688	002352'	062701	177336		ADD	#COUNTS-,R1		
689	002356'	010702			MOV	PC,R2		:SET UP TBL ADR
690	002360'	062702	000066		ADD	#REPTBL-,R2		
691	002364'	012267	000012	RPTLP:	MOV	(R2)+,RPTBAS		:MOV MSG ADR TO S/R LINKAGE
692	002370'	004067	007676		JSR	RD,SAVEG		:SAVE ALL REG'S
693	002374'	011100			MOV	(R1),R0		:GET CURRENT COUNT
694	002376'	004577	175454		JSR	RS,#BINASC		:CONVERT IT TO ASCII
695	002402'	000000		RPTBAS:	.WORD	XXXX		
696	002404'	004067	007676		JSR	RD,RESREG		:RESTORE REG'S
697	002410'	005721			TST	(R1)+		:POINT AT NXT CNT
698	002412'	005300			DEC	R0		:DONE ALL WORDS?
699	002414'	001363			BNE	RPTLP		:Y,N-RPTLP
700	002416'	004567	010342		JSR	RS,PRINT		:GO ISSUE COUNTS MSG
701	002422'	011014			.WORD	CNTSMG-		
702	002424'	000502			.WORD	CNTSEN-CNTSMG		
703	002426'	004567	010332	RPTEND:	JSR	RS,PRINT		:ISSUE "END OF REPORT" MSG
704	002432'	010663			.WORD	RENDMG-		
705	002434'	177761			.WORD	-15		
706	002436'	004067	007644	DVREX:	JSR	RD,RESREG		:RESTORE REGISTERS
707	002442'	005725			TST	(RS)+		:SET UP RETURN POINT
708	002444'	000205			RTS	RS		:EXIT IN-LINE
709								
710								
711	002446'	011050		REPTBL:	.WORD	BCMRD-RPTBAS		
712	002450'	011056			.WORD	BCMRD+6-RPTBAS		
713	002452'	011072			.WORD	BCMRD-RPTBAS		
714	002454'	011100			.WORD	BCMRD+6-RPTBAS		
715	002456'	011116			.WORD	BCMRD-RPTBAS		


```

716 002460' 011124 .WORD BCMCK+6-RPTBAS
717 002462' 011151 .WORD CHDCRD-RPTBAS
718 002464' 011164 .WORD CHDCLR-RPTBAS
719 002466' 011177 .WORD CHDCCK-RPTBAS
720 002470' 011215 .WORD CHDCSK-RPTBAS
721 002472' 011232 .WORD CHDCMS-RPTBAS
722 002474' 011260 .WORD CNTERR-RPTBAS
723 002476' 011301 .WORD CNTCEC-RPTBAS
724 002480' 011322 .WORD CNTDER-RPTBAS
725 002482' 011350 .WORD CNTDLY-RPTBAS
726 002484' 011364 .WORD CNTDTE-RPTBAS
727 002486' 011401 .WORD CNTHCR-RPTBAS
728 002490' 011420 .WORD CNTFER-RPTBAS
729 002492' 011434 .WORD CNTDCK-RPTBAS
730 002494' 011453 .WORD CNTMCE-RPTBAS
731 002496' 011502 .WORD CNTRTY-RPTBAS
732 002498' 011530 .WORD CNTINT-RPTBAS

```

; TIMEOUT ERROR ROUTINE

```

733
734
735 ; JSR RS, TOUTER S/R CALL
736
737
738
739 002522' 004067 007544 TOUTER: JSR RO, SAVREG ; SAVE ALL REGISTERS
740 002524' 004767 007576 JSR PC, SUPTAD ; SET UP RPCS1 & PROG TBL ADR'S
741 002526' 032767 000400 175242 BIT @SPINFL, DFLGMD ; IS SPIN OR RECAL IN PROGRESS
742 002530' 001024 BNE CKSPIN ; YES
743 002532' 004567 007576 JSR RS, STSTAT ; STORE CURRENT STATUS
744 002534' 177102 .WORD CSTAT-
745 002536' 004567 006404 JSR RS, TVECT ; CK IF I HAVE VECTOR CONTROL
746 002538' 000404 BR 105 ; BR IF I DON'T
747 002540' 142714 000100 BICB @100, (R4) ; RESET INT ENABLE
748 002542' 004767 006346 JSR PC, RINTV ; RESET THE INTERRUPT VECTOR
749 002544' 042713 000010 105: BIC @INT4IOT, (R3) ; RESET WAITING FOR I/O FLG
750 002546' 004567 006420 JSR RS, ERACS1 ; ISSUE I/O TIMEOUT ERROR MSG
751 002548' 003062 .WORD IOT0-ERMBAS
752 002550' 004067 007502 JSR RO, RESREG ; RESTORE REGISTERS
753 002552' 012605 MOV (SP)+, RS ; REMOVE RETURN ADR
754 002554' 000177 175236 JHP @CUPGER ; GO TO ERROR EXIT
755
756 002612' 032777 000002 175244 CKSPIN: BIT @USMTPS, @CSYSFW ; NEED TO USE MTPS INST?
757 002614' 001007 BNE 325 ; N Y-325
758 002616' 113767 177776 177162 MOVB @RPS, CURPSM ; SAVE CURRENT PRIORITY
759 002618' 152737 000340 177776 BICB @340, @RPS ; SET PRIORITY TO ?
760 002620' 000404 BR 345 ; GO SET UP UNIT #
761 002622' 106767 177146 325: MFPS CURPSM ; SAVE CURR PRIORITY
762 002624' 106427 000340 MTPS @340 ; SET PRIORITY TO ?
763 002626' 016400 000010 345: MOV RPCS2(R4), RO ; GET CURR UNIT #
764 002628' 010001 MOV RO, R1 ; SAVE IT
765 002630' 042700 177747 BIC @177747, RO ; RESET UNIT # & OTHER BITS
766 002632' 156300 000035 BICB PCURDV(R3), RO ; SET IN MY UNIT #
767 002634' 110064 000010 MOVB RO, RPCS2(R4) ; MOVE UNIT # TO RK611
768 002636' 005005 CLR RS
769 002638' 004767 010242 JSR PC, SD ; SELECT MY DRIVER
770 002640' 016400 000012 MOV RPOS(R4), RO ; GET DRIVE'S STATUS
771 002642' 032700 000200 BIT @DRDY, RO ; IS DRIVE BACK ON LINE?

```

```

772 002710' 001405          BEQ      40S          :NO, WAIT LONGER
773 002712' 042713 000010 38S: BIC      8WT4IOT, (R3) :RESET WAITING FOR I/O TERM FLAG
774 002715' 042767 000400 175056 BIC      8SPINFL, DFLCMD :RESET SPIN IN PROGRESS FLAG
775 002724' 012763 005670 000030 40S: MOV      83000, 8PYCNT(R3) :RESTORE I SECOND I/O COUNT
776 002732' 110164 000010          MOVB     R1, 8PCS2(R4) :RESTORE ORIG UNIT #
777 002736' 032777 000002 175120 BIT      8USHTPS, 8CSYSFW :NEED TO USE MTPS INST?
778 002744' 001004          BNE      50S          :N, Y-50S
779 002746' 116737 177040 177776 MOVB     CURPSW, 88PS :RESTORE ORIG PRIORITY
780 002754' 000402          BR       60S          :GO TO EXIT
781 002756' 106467 177030          MTPS     CURPSW :RESTORE ORIG PRIORITY
782 002762' 004067 007320          JSR      80, RESREG :RESTORE REGISTERS
783 002766' 000205          RTS      RS          :RETURN IN-LINE
784
785
786                                     ;KILL USER PROGRAM ROUTINE
787
788                                     :JSR      RS, KILL      S/R CALL
789                                     :R3 MUST CONTAIN PROG TBL ADR
790                                     :DESTROYS 80, R1
791
792 002770' 016701 175030          KILL:  MOV     DREG80, R1 :GET DEV REG ADR
793 002774' 004567 006160          JSR     RS, IVECT :DO I HAVE VECTOR CONTROL?
794 003000' 000407          BR     KILLEX :BR IF I DON'T
795 003002' 132711 000100          BITB   8100, (R1) :IS INT ENABLE SET?
796 003006' 001402          BEQ    10S :Y, N-10S
797 003010' 142711 000100          BICB   8100, (R1) :RESET INT ENABLE
798 003014' 004767 006114          JSR    PC, RINTV :RESET INT VECTOR INFO
799 003020' 000205          KILLEX: RTS     RS :EXIT IN-LINE

```

.SBTTL RK06 NON-I/O FUNCTION ROUTINES

NON-INTERRUPT I/O FUNCTION ROUTINES

;"STEPUP" FUNCTION ROUTINE

```

801
802
803
804
805
806
807
808
809
810
811
812 003022' 004767 000104
813 003026' 012567 000076
814 003032' 004767 000252
815 003036' 010067 174742
816 003042' 010167 174740
817 003046' 010267 174736
818 003052' 005767 000052
819 003056' 001402
820 003060' 005367 000044
821 003064' 004767 000220
822 003070' 020067 174710
823 003074' 103406
824 003076' 020027 000632
825 003102' 001011
826 003104' 020127 000002
827 003110' 001006
828 003112' 005067 174666
829 003116' 005067 174664
830 003122' 005067 174662
831 003126' 000205
832 003130' 000000
833
834
835 003132' 016700 174646
836 003136' 016701 174644
837 003142' 016702 174642
838 003146' 012704 000633
839 003152' 012703 000024
840 003156' 032767 010000 176630
841 003164' 001002
842 003166' 012703 000026
843 003172' 000207
844
845
846
847
848
849
850
851 003174' 004767 177732
852 003200' 012567 177724
853 003204' 004767 000146
854 003210' 020067 174570
855 003214' 101014
856 003216' 010067 174562

```

;"STEPDN" FUNCTION ROUTINE

```

848
849
850
851
852
853
854
855
856

```

857	003222'	010167	174560		MOV	R1, HEAD	
858	003226'	010267	174556		MOV	R2, SECT	
859	003232'	020027	000632		CMP	R0, #NRCYL-1	: IS ENDING CYL=MAX CYL
860	003236'	001023			BNE	STP1	: NO, EXIT
861	003240'	020127	000002		CMP	R1, #NRHEAD-1	: IS ENDING HEAD=MAX HEAD
862	003244'	001020			BNE	STP1	: NO, EXIT
863	003246'	012767	000632	174530	MOV	#NRCYL-1, CYL	: YES, FORCE WRAP-AROUND
864	003254'	012767	000002	174524	MOV	#NRHEAD-1, HEAD	
865	003262'	005067	174522		CLR	SECT	
866	003266'	005767	177636		TST	STEIV	: IS OFFSET=0
867	003272'	001002			BNE	STP31	: NO
868	003274'	005367	174506		DEC	HEAD	: YES, CORRECT HEAD VALUE
869	003300'	004767	177626		JSR	PC, STPCOM	
870	003304'	000737		STP31:	BR	STP2	: START OVER FROM TOP DOWN
871	003306'	000205		STP1:	RTS	RS	
872							
873	003310'	066702	177614		UPSUB:	ADD	STEIV, R2
874	003314'	020203		UP4:	CMP	R2, R3	: GET NEW SECT
875	003316'	103403			BLO	UP1	: IS SECT IN RANGE
876	003320'	160302			SUB	R3, R2	: YES
877	003322'	005201			INC	R1	: NO, REDUCE SECT
878	003324'	000773			BR	UP4	: INCREMENT HEAD
879	003326'	020127	000003	UP1:	CMP	R1, #NRHEAD	: IS HEAD IN RANGE
880	003332'	103404			BLO	UP2	: YES
881	003334'	162701	000003		SUB	#NRHEAD, R1	: NO, REDUCE HEAD
882	003340'	005200			INC	R0	: INCREMENT CYL
883	003342'	000771			BR	UP1	
884	003344'	020004		UP2:	CMP	R0, R4	: IS CYL IN RANGE
885	003346'	103402			BLO	UP3	: YES, EXIT
886	003350'	160400			SUB	R4, R0	: NO, REDUCE CYL
887	003352'	000774			BR	UP2	
888	003354'	000207		UP3:	RTS	PC	
889							
890	003356'	166702	177546		DWNSUB:	SUB	STEIV, R2
891	003362'	020203		DWN4:	CMP	R2, R3	: GET NEW SECT
892	003364'	103403			BLO	DWN1	: IS SECT IN RANGE
893	003366'	060302			ADD	R3, R2	: YES
894	003370'	005301			DEC	R1	: NO, INCREASE SECT
895	003372'	000773			BR	DWN4	: DECREMENT HEAD
896	003374'	020127	000003	DWN1:	CMP	R1, #NRHEAD	: IS HEAD IN RANGE
897	003400'	103404			BLO	DWN2	: YES
898	003402'	062701	000003		ADD	#NRHEAD, R1	: INCREASE HEAD
899	003406'	005300			DEC	R0	: DECREMENT CYL
900	003410'	000771			BR	DWN1	
901	003412'	020004		DWN2:	CMP	R0, R4	: IS CYL IN RANGE?
902	003414'	103402			BLO	DWN3	: YES, EXIT
903	003416'	060400			ADD	R4, R0	: NO, INCREASE CYL
904	003420'	000774			BR	DWN2	
905	003422'	000207		DWN3:	RTS	PC	
906							
907							
908							
909							
910							
911							
912	003424'	042767	100000	174350	WAIT:	BIC	#WAITMD, DFLGWD

;"WAIT" FUNCTION ROUTINE

;JSR RS, WAIT

FUNCTION CALL

;RESET THE "NOWAIT" FLAG

H02

MAINDEC-11-DTR6A-A RK611 - RK06 DEVICE ROUTINE FOR MPG
DTR6AA.P11 RK06 NON-I/O FUNCTION ROUTINES

MACY11 27(732) 24-SEP-76 14:11 PAGE 7-2

SEQ 0020

```
913 003432' 004767 005232 JSR PC,CKDBSY ;WAIT IF BUSY & DO TERMINATION
914 003436' 004767 005472 JSR PC,RINTV ;RESET THE INTERRUPT VECTOR
915 003442' 000205 RTS RS ;EXIT IN-LINE
916
917
918 ;"NOWAIT" FUNCTION ROUTINE
919 ;JSR RS,NOWAIT FUNCTION CALL
920
921 003444' 052767 100000 174330 NOWAIT: BIS #WAITMD,DFLGMD ;SET THE "NOWAIT" FLAG
922 003452' 000205 RTS RS ;EXIT IN-LINE
923
924 ;"ODD" FUNCTION ROUTINE
925 ;JSR RS,ODD FUNCTION CALL
926
927 003454' 042767 000020 176336 ODD: BIC #PAT,RPMR1V ;SELECT ODD PARITY
928 003462' 000205 RTS RS ;EXIT IN-LINE
929 ;"EVEN" FUNCTION ROUTINE
930 ;JSR RS,EVEN FUNCTION CALL
931
932 003464' 052767 000020 176326 EVEN: BIS #PAT,RPMR1V ;SELECT EVEN PARITY
933 003472' 000205 RTS RS ;EXIT IN-LINE
934
935 ;"BAION" FUNCTION ROUTINE
936 ;JSR RS,BAION FUNCTION CALL
937
938 003474' 052767 000020 176314 BAION: BIS #BAI,RPCS2V ;SET THE BAI BIT
939 003502' 000205 RTS RS ;EXIT IN-LINE
940
941 ;"BAIOFF" FUNCTION ROUTINE
942 ;JSR RS,BAIOFF FUNCTION CALL
943
944 003504' 042767 000020 176304 BAIOFF: BIC #BAI,RPCS2V ;RESET THE BAI BIT
945 003512' 000205 RTS RS ;EXIT IN-LINE
946 ;"CORON" FUNCTION ROUTINE
947 ;JSR RS,CORON FUNCTION CALL
948
949 003514' 042767 002000 174260 CORON: BIC #CORFLG,DFLGMD ;RESET THE COR INH FLAG
950 003522' 000205 RTS RS ;EXIT IN-LINE
951
952 ;"COROFF" FUNCTION ROUTINE
953 ;JSR RS,COROFF FUNCTION CALL
954
955 003524' 052767 002000 174250 COROFF: BIS #CORFLG,DFLGMD ;SET THE COR INH FLAG
956 003532' 000205 RTS RS ;EXIT IN-LINE
957
958
```

```
969                                     ;"FMT22" FUNCTION ROUTINE
970
971                                     ;JSR   RS,FMT22           FUNCTION CALL
972
973 003534' 042767 010000 176252 FMT22: BIC   #CFMT,RPCS1V       ;SELECT 22 SECTORS
974 003542' 000205                RTS   RS                     ;EXIT IN-LINE
975
976                                     ;"FMT20" FUNCTION ROUTINE
977
978                                     ;JSR   RS,FMT20           FUNCTION CALL
979
980
981 003544' 052767 010000 176242 FMT20: BIS   #CFMT,RPCS1V       ;SELECT 20 SECTORS
982 003552' 000205                RTS   RS                     ;EXIT IN-LINE
```

```

984                                     .SBTTL  RK06 NON-INTERRUPT TYPE I/O FUNCTION ROUTINES
985
986
987                                     ;"CRESET" FUNCTION ROUTINE
988
989                                     ;JSR    R5,CRESET          FUNCTION CALL
990
991 003554' 004767 005110      CRESET: JSR    PC,CKDBSY          ;GO CK IF DEV IS BUSY
992 003560' 005267 176150      INC    MISCNT          ;ADD 1 TO MISC. CMND CNT
993 003564' 005067 174232      CLR    ERRI           ;RESET THE ERROR INDICATOR
994 003570' 005000      CLR    RD             ;INITIALIZE TIME OUT COUNT
995 003572' 052714 100000      BIS    #CCLR,(R4)     ;SET THE CCLR BIT
996 003576' 032714 000200      CRE1: BIT    #RDY,(R4) ;IS CONTROLLER NOW READY?
997 003602' 001007      BNE    ZOS           ;YES
998 003604' 005300      DEC    RD             ;TIMED OUT?
999 003606' 100773      BMI    CRE1          ;NOT YET
1000 003610' 004567 005374      JSR    R5,ERRCS      ;GO ISSUE CRESET TIMEOUT ERROR
1001 003614' 003013      .WORD  CRT0-ERMBAS
1002 003616' 000177 174226      JMP    @CUPGER
1003 003622' 000205      ZOS:  RTS    R5      ;EXIT IN-LINE TO USER'S PROG
1004
1005
1006                                     ;"SRESET" FUNCTION ROUTINE
1007
1008                                     ;JSR    R5,SRESET          FUNCTION CALL
1009
1010 003624' 004767 005040      SRESET: JSR    PC,CKDBSY ;GO CK IF DEV IS BUSY
1011 003630' 005267 176100      INC    MISCNT        ;COUNT MISC CMNDS
1012 003634' 005067 174162      CLR    ERRI         ;CLEAR ERROR INDICATOR
1013 003640' 005000      CLR    RD           ;PERFORM SRESET
1014 003642' 052764 000040 000010 BIS    #SCLR,RPCS2(R4)
1015 003650' 004767 007306      JSR    PC,DRVDLY
1016 003654' 032714 000200      SRES1: BIT    #RDY,(R4) ;IS CONTROLLER READY?
1017 003660' 001007      BNE    SRES2        ;YES
1018 003662' 005300      DEC    RD           ;NO, ALARM AN ERROR
1019 003664' 100773      BMI    SRES1
1020 003666' 004567 005316      JSR    R5,ERRCS
1021 003672' 003031      .WORD  SR10-ERMBAS
1022 003674' 000177 174150      JMP    @CUPGER
1023 003700' 000205      SRES2: RTS    R5
1024
1025                                     ;"DRESET" FUNCTION ROUTINE
1026
1027                                     ;JSR    R5,DRESET          FUNCTION CALL
1028
1029 003702' 012702 000005      DRESET: MOV    #DCCODE,R2 ;SETUP DRESET CMND CODE
1030 003706' 000436      BR     NOICOM        ;GO TO NO INT CMND COM PROCESSING
1031
1032                                     ;"UNLOAD" FUNCTION ROUTINE
1033
1034                                     ;JSR    R5,UNLOAD          FUNCTION CALL
1035
1036
1037 003710' 012702 000007      UNLOAD: MOV    #UCODE,R2 ;GET UNLOAD CODE
1038 003714' 000433      BR     NOICOM        ;GO TO NO INT CMND COM PROCESSING
1039

```

```

1040 ;"SPIN" FUNCTION ROUTINE
1041
1042 ;JSR RS,SPIN
1043
1044 003716' 012702 000011 SPIN: MOV #SSCODE,R2 ;GET SPIN CODE
1045 003722' 000430 BR NOICOM ;GO TO NO INT CMND COM PROCESSING
1046
1047 ;"PAKACK" FUNCTION ROUTINE
1048
1049 ;JSR RS,PAKACK FUNCTION CALL
1050
1051 003724' 012702 000003 PAKACK: MOV #PACODE,R2 ;GET PACK ACK CODE
1052 003730' 000425 BR NOICOM ;GO TO NO INT CMND COM PROCESSING
1053
1054 ;"REL" FUNCTION ROUTINE
1055
1056 ;JSR RS,REL ;FUNCTION CALL
1057
1058 003732' 004767 004732 REL: JSR PC,CKDBSY ;GO CK IF DEV IS BUSY
1059 003736' 005267 175772 INC MISCNT ;COUNT MISC CMNDS
1060 003742' 005067 174054 CLR ERRI ;CLEAR ERROR COUNT
1061 003746' 005000 CLR RO
1062 003750' 052764 000010 000010 REL1: BIS #RLS,RPCS2(R4) ;PERFORM REL
1063 003756' 032714 000200 BIT #RDY,(R4) ;IS CONTROLLER READY?
1064 003762' 001007 BNE REL2 ;YES
1065 003764' 005300 DEC RO
1066 003766' 100773 BMI REL1
1067 003770' 004567 005214 JSR RS,ERRCS ;NO, ALARM ERROR
1068 003774' 003047 .WORD RT0-ERMBAS
1069 003776' 000177 174046 JMP @CUPGER
1070 004002' 000205 REL2: RTS RS
1071
1072 ;NON INTERRUPT I/O FUNCTION COMMON PROCESSING
1073
1074 ;RS = USER PROGRAM RETURN ADR
1075 ;R4 = RPCS1 ADR
1076 ;R3 = PROG TBL ADR
1077 ;R2 = FUNCTION'S COMMAND CODE
1078
1079 004004' 004767 004660 NOICOM: JSR PC,CKDBSY ;GO CK IF DEV IS BUSY
1080 004010' 005267 175720 INC MISCNT ;ADD 1 TO MISC. CMND CNT
1081 004014' 004767 000174 NOI1: JSR PC,ACQHSK ;HOUSEKEEP THE DISK
1082 004020' 032764 000100 000012 BIT #VV,RPDS(R4) ;IS THE VV BIT SET?
1083 004026' 001004 BNE 20$ ;N.Y-20$
1084 004030' 012714 000003 MOV #PACODE,(R4) ;ISSUE THE PACK ACKNOWLEDGE CMND
1085 004034' 004767 007122 JSR PC,DRVDLY
1086 004040' 056702 175750 20$: BIS RPCS1V,R2 ;MERGE CFMT/CDT
1087 004044' 052767 000300 173730 BIS #CMDISU+ANYIOI,DFLGWD ;SET CMND ISSUED FLAGS
1088 004052' 010214 MOV R2,(R4) ;ISSUE SPECIFIED CMND
1089 004054' 012700 000024 MOV #20.,RO ;SET UP DELAY CNT
1090 004060' 005300 30$: DEC RO ;DELAY FOR A FEW MICROSECONDS
1091 004062' 001376 BNE 30$
1092 004064' 004767 007072 JSR PC,DRVDLY
1093 004070' 032764 100000 000012 BIT #SVAL,RPDS(R4) ;IS RKDS VALID
1094 004076' 001002 BNE 14$ ;YES
1095 004100' 004767 007056 JSR PC,DRVDLY ;NO, DELAY

```


1096	004104'	032764	000001	000012	14S:	BIT	#DRA,RPDS(R4)	: IS DRIVE AVAILABLE
1097	004112'	001010				BNE	35S	: YES
1098	004114'	042767	000300	173660		BIC	#CMDISU,ANYIOI,DFLGND	: NO, WAIT FOR IT
1099	004122'	004577	173720			JSR	RS,@CIOBSY	: ALLOW OTHERS TO RUN
1100	004126'	004767	007116			JSR	PC,DRVCLR	: DO DRIVE CLEAR
1101	004132'	000730				BR	NO11	: REPEAT COMMAND
1102	004134'	005714			35S:	TST	(R4)	: IS 'CERR' ERROR BIT SET?
1103	004136'	100005				BPL	50S	: Y,N-50S
1104	004140'	004567	005044			JSR	RS,ERRCS	: REPORT NON-INT TERM ERROR
1105	004144'	003101				.WORD	NO1TER-ERMBAS	
1106	004146'	000177	173676			JMP	@CLPGR	: GO TO MPG'S ERROR RETURN POINT
1107	004152'	120227	000011		50S:	CMPB	R2,#SSCODE	: IS THIS THE SPIN COMMAND
1108	004156'	001015				BNE	60S	: NO
1109	004160'	052767	000400	173614		BIS	#SPINFL,DFLGND	: SET SPIN IN PROGRESS FLAG
1110	004166'	012763	005670	000030		MOV	#3000.,PTCNT(R3)	: INITIALIZE 1 SECOND T/O CNT
1111	004174'	005767	173602			TST	DFLGND	: "NOWAIT" FLAG SET?
1112	004200'	100404				BMI	60S	: N,Y-60S
1113	004202'	052713	000010			BIS	#WT4IOT,(R3)	: SET WAITING FOR I/O TERM FLAG
1114	004206'	004577	173634			JSR	RS,@CIOBSY	: RELEASE CNTRL UNTIL UNIT IS ON-LINE
1115	004212'	000205			60S:	RTS	RS	: EXIT TO USER PROG

;HOUSEKEEP DISK

;JSR PC,ACQHSK S/R CALL
;R5 = ADR AFTER USER PROG JSR
;R4 = RPCS1 ADR
;R3 = PROG TBL ADR
;DESTROYS R0

```

1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127 004214' 010146          ACQHSK: MOV      R1, -(SP)          ;SAVE R1
1128 004216' 042767 000561 173556 ACQRTY: BIC      @CNDISU!SPINFL!SMOVRTO!SMOIER!IOERR,DFLGWD ;HSKP FLAG BITS
1129 004224' 012763 072460 000030      MOV      @ITIME,PTOCNT(R3) ;SETUP INT TIMER
1130 004232' 116300 000035      MOVVB   PCURDV(R3),R0 ;GET MY UNIT #
1131 004236' 020027 000007      CMP      R0,#7 ;VALID UNIT #?
1132 004242' 101405          BLOS    10$ ;N.Y-10$
1133 004244' 005267 175472          INC     DATAER ;ADJ ERROR COUNTS
1134 004250' 005367 175462          DEC     ERRCNT
1135 004254' 000427          BR      ACQERR ;GO REPORT THE ERROR
1136 004256' 010001          10$:  MOV     R0,R1 ;GET DISPLACEMENT INTO
1137 004260' 006301          ASL     R1 ;THE ATA TABLE FOR
1138 004262' 060701          ADD     PC,R1 ;THIS UNIT #
1139 004264' 062701 000100          ADD     @ATATBL-. ,R1
1140 004270' 112167 000110          MOVVB   (R1)+,MYATA ;STORE ATA BIT MASKS FOR
1141 004274' 111167 000106          MOVVB   (R1),OTHATA ;THIS UNIT #
1142 004300' 056700 175512          BIS     RPCS2V,R0 ;SET BAI BIT IN UNIT #
1143 004304' 010064 000010          MOV     R0,RPCS2(R4) ;SETUP RKCS2
1144 004310' 016764 175504 000026          MOV     RPAR1V,RPMR1(R4) ;AND RPMR1
1145 004316' 010501          MOV     R5,R1 ;SELECT DRIVE
1146 004320' 005005          CLR     R5
1147 004322' 004767 006614          JSR     PC,SD
1148 004326' 010105          MOV     R1,R5
1149 004330' 012601          MOV     (SP)+,R1 ;RESTORE R1
1150 004332' 000207          RTS     PC ;EXIT IN-LINE
1151 004334' 010146          ACQERR: MOV     R1, -(SP) ;SAVE R1 & R2
1152 004336' 010246          MOV     R2, -(SP)
1153 004340' 004567 004644          JSR     R5,ERRCS ;STORE CURR STATUS & REPORT
1154 004344' 003562          .WORD  INVDVN-ERMBAS ;INVALID UNIT NR
1155 004346' 012602          MOV     (SP)+,R2 ;RESTORE R1 & R2
1156 004350' 012601          MOV     (SP)+,R1
1157 004352' 004577 173472          JSR     R5,@CUPGER ;GO TO MPG'S ERROR RETURN POINT
1158 004356' 004767 004306          JSR     PC,CKDBSY ;CHECK IF DISK IS BUSY
1159 004362' 000715          BR      ACQRTY
1160
1161 004364' 001 376          ATATBL: .BYTE 001,376
1162 004366' 002 375          .BYTE 002,375
1163 004370' 004 373          .BYTE 004,373
1164 004372' 010 367          .BYTE 010,367
1165 004374' 020 357          .BYTE 020,357
1166 004376' 040 337          .BYTE 040,337
1167 004400' 100 277          .BYTE 100,277
1168 004402' 200 177          .BYTE 200,177
1169
1170 004404' 000000          MYATA: .WORD 0
1171 004406' 000000          OTHATA: .WORD 0

```

.SBTTL RK06 INTERRUPT TYPE I/O FUNCTION ROUTINES

INTERRUPT TYPE I/O FUNCTION ROUTINES

;"READ" FUNCTION ROUTINE

```

:JSR      RS,READ      FUNCTION CALL
:..WORD   ADR          DATA ADDRESS (BITS 16 - 21)
:..WORD   ADR          DATA ADDRESS (BITS 0 - 15)
:..WORD   CNT          BYTE COUNT
:..WORD   DEV          (NOT USED)
    
```

```

1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187 004410' 012702 000121
1188 004414' 012701 000635
1189 004420' 004767 004244
1190 004424' 005267 175274
1191 004430' 010700
1192 004432' 062700 175260
1193 004436' 000167 000454
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204 004442' 012702 000123
1205 004446' 012701 000235
1206 004452' 004767 004212
1207 004456' 005267 175244
1208 004462' 010700
1209 004464' 062700 175232
1210 004470' 004067 005576
1211 004474' 016746 173304
1212 004500' 016746 173302
1213 004504' 016746 173300
1214 004510' 016567 000004 176412
1215 004516' 006267 176406
1216 004522' 042767 100000 176400
1217 004530' 012746 000010
1218 004534' 006267 176370
1219 004540' 005316
1220 004542' 001374
1221 004544' 005726
1222 004546' 032765 000777 000004
1223 004554' 001402
1224 004556' 005267 176346
1225 004562' 004767 176344
1226 004566' 005367 176336
1227 004572' 004567 176266
1228 004576' 026667 000004 173200
    
```

```

READ:  MOV      #RCODE! IE,R2
        MOV      #635,R1
RDCOM: JSR      PC,CKDBSY
        INC      RDCNT
        MOV      PC,R0
        ADD      #BYRD+2--,R0
        JMP      CMDCOM
    
```

```

:SETUP READ CODE
:SET UP CMND FLAG WORD
:GO CK IF DEV IS BUSY
:ADD 1 TO READ CMND CNT
:SET UP ADR OF BYTES READ CNT
:GO TO CMND COMMON PROCESSING
    
```

;"WRITE" FUNCTION ROUTINE

```

:JSR      RS,WRITE     FUNCTION CALL
:..WORD   ADR          DATA ADDRESS (BITS 16 - 21)
:..WORD   ADR          DATA ADDRESS (BITS 0 - 15)
:..WORD   CNT          BYTE COUNT
:..WORD   DEV          (NOT USED)
    
```

```

WRITE: MOV      #WCODE! IE,R2
        MOV      #235,R1
WRCOM: JSR      PC,CKDBSY
        INC      WRCNT
        MOV      PC,R0
        ADD      #BYWR+2--,R0
        JSR      R0,SAVREG
        MOV      CYL,-(SP)
        MOV      HEAD,-(SP)
        MOV      SECT,-(SP)
        MOV      4(R5),STEIV
        ASR      STEIV
        BIC      #100000,STEIV
        MOV      #8,-(SP)
WR3:   ASR      STEIV
        DEC      (SP)
        BNE     WR3
        TST     (SP)+
        BIT     #777,4(R5)
        BEQ     WR1
        INC     STEIV
WR1:   JSR      PC,STPCOM
        DEC     STEIV
        JSR     RS,STPWRT
        CMP     4(SP),CYL
    
```

```

:SETUP WRITE CODE
:SET UP CMND FLAG WORD
:GO CK IF DEV IS BUSY
:ADD 1 TO WRITE CMND CNT
:SET UP ADR OF BYTES WRITTEN CNT
:SAVE REGS
:SAVE STARTING DISK ADDR
:GET SIZE OF XFER TO WRITE
:DIVIDE BY 512 BYTES/SECTOR
: TO GET NR SECTORS
:IS THERE A PARTIAL SECTOR
:NO
:YES, USE ONE MORE SECT
:WILL THIS XFER DAMAGE LAST TRACK
:USE STEPUP TO GET ENDING ADDR
:DID CYL CHANGE
    
```

MAINDEC-11-DTR6A-A RK611 - RK06 DEVICE ROUTINE FOR MPG
DTR6AA.P11 RK06 INTERRUPT TYPE I/O FUNCTION ROUTINES

```

1239 004604' 001021      BNE      WR2           ;YES, ALARM
1230 004606' 026667 000002 173172  CMP      2(SP),HEAD  ;DID HEAD CHANGE
1231 004614' 001015      BNE      WR2           ;YES, ALARM
1232 004616' 021667 173166  CMP      (SP),SECT  ;DID SECT CHANGE
1233 004622' 001012      BNE      WR2           ;YES, ALARM
1234 004624' 012667 173160  MOV      (SP)+,SECT ;NO, PERFORM XFER
1235 004630' 012667 173152  MOV      (SP)+,HEAD ;AFTER RESTORE ADDR
1236 004634' 012667 173144  MOV      (SP)+,CYL  ;AND REGISTERS
1237 004640' 004067 005442  JSR      RD,RESREG
1238 004644' 000167 000246  JMP      CMDCOM      ;GO TO COMMON
1239 004650' 012667 173134      WR2:  MOV      (SP)+,SECT ;GET BAD ADDRESS AND
1240 004654' 012667 173126  MOV      (SP)+,HEAD
1241 004660' 012667 173120  MOV      (SP)+,CYL
1242 004664' 062706 000014  ADD      #12,SP      ;CORRECT STACK
1243 004670' 004567 004314  JSR      RS,ERACS   ;PRINT HAVE BAD ADDRESS
1244 004674' 003421      .WORD   PAHBA-ERMBAS
1245 004676' 000177 173146  JMP      @CUPGER    ;GO TO MPG ERROR RTN

; "RDHD" FUNCTION ROUTINE
;JSR      RS,RDHD    FUNCTION CALL
; .WORD   ADR        DATA ADDRESS (BITS 16 - 21)
; .WORD   CNT        BYTE COUNT

1246 004702' 012702 000125      RDHD: MOV      @RHCODE!IE,R2 ;SETUP READ-HEADER CODE
1247 004706' 012701 001626  MOV      #1626,R1    ;SET UP CMD FLAG WORD
1248 004712' 000642      BR      RDCOM       ;GO TO COMMON READ PROCESSING

```

1258
 1259
 1260
 1261
 1262
 1263
 1264
 1265
 1266
 1267
 1268
 1269
 1270
 1271
 1272
 1273
 1274
 1275
 1276
 1277
 1278
 1279
 1280
 1281
 1282
 1283
 1284
 1285
 1286
 1287
 1288
 1289
 1290
 1291
 1292
 1293
 1294
 1295
 1296
 1297
 1298
 1299
 1300
 1301
 1302
 1303
 1304
 1305
 1306
 1307
 1308
 1309
 1310
 1311
 1312
 1313

;"WRHD" FUNCTION ROUTINE

```

      ;JSR    RS,WRHD      FUNCTION CALL
      ;.WORD  ADR         DATA ADDRESS (BITS 16 - 21)
      ;.WORD  ADR         DATA ADDRESS (BITS 0 - 15)
      ;.WORD  CNT         BYTE COUNT

WRHD:  MOV    @WRCODE!IE,R2 ;SETUP WRITE-HEADER CODE
      MOV    @236,R1       ;SET UP CMD FLAG WORD
      BR     WRCCOM        ;GO TO WRITE COMMON PROCESSING
  
```

;"WRCK" FUNCTION ROUTINE

```

      ;JSR    RS,WRCK      FUNCTION CALL
      ;.WORD  ADR         DATA ADDRESS (BITS 16 - 21)
      ;.WORD  ADR         DATA ADDRESS (BITS 0 - 15)
      ;.WORD  CNT         BYTE COUNT

WRCK:  MOV    @WRCODE!IE,R2 ;SETUP WRITE-CHECK CODE
      MOV    @236,R1       ;SET UP CMD FLAG WORD
      JSR    PC,CKDBSY    ;GO CK IF DEV IS BUSY
      INC   CKCNT        ;ADD 1 TO CHECK CMD COUNT
      MOV   PC,R0        ;SET UP ADR OF BYTES
      ADD   @BYCK+2-.,R0 ;CHECKED COUNT
      BR    WRCCOM        ;GO TO CMD COM PROCESSING
  
```

;"SEEK" FUNCTION ROUTINE

```

      ;JSR    RS,SEEK      FUNCTION CALL

SEEK:  MOV    @SCODE!IE,R2 ;SETUP SEEK CODE
      MOV    @2060,R1     ;SET UP CMD FLAG WORD
SKCOM: JSR    PC,CKDBSY    ;GO CK IF DEV IS BUSY
      INC   SKCNT        ;ADD 1 TO SEEK CMD COUNT
      BR    WRCCOM        ;GO TO CMD COMMON PROCESSING
  
```

;"OFFSET" FUNCTION ROUTINE

```

      ;JSR    RS,OFFSET    FUNCTION CALL
      ;.WORD  VALUE       BINARY OFFSET VALUE

OFFSET: MOV    @OCODE!IE,R2 ;SETUP OFFSET CODE
      MOV    @2140,R1     ;SET UP CMD FLAG WORD
      BR    SKCOM        ;GO TO SEEK CMD COM PROCESSING
  
```

;"RECAL" FUNCTION ROUTINE

```

      ;JSR    RS,RECAL     FUNCTION CALL

RECAL: MOV    @RCCODE!IE,R2 ;SETUP RECALIBRATE CODE
      MOV    @2040,R1     ;SET UP CMD FLAG WORD
      BR    SKCOM        ;GO TO SEEK CMD COM PROCESSING
  
```

;"SELDRI" FUNCTION ROUTINE

MAINDEC-11-DTR6A-A RK611 - RK06 DEVICE ROUTINE FOR MPG
DTR6AA.P11 RK06 INTERRUPT TYPE I/O FUNCTION ROUTINES

```

1314                                     :JSR   RS,SELDRI
1315                                     :.WORD 0          CODE TO SELECT MESSAGE FROM DRIVE
1316
1317 005024' 004767 005274           SELDRI: JSR   PC,SUPTAD           :SETUP R3 & R4
1318 005030' 004767 003634           JSR   PC,CKOBSY          :CHECK IF DISK IS BUSY
1319 005034' 004767 177154           JSR   PC,ACOMSK          :HOUSEKEEP DISK
1320 005040' 005267 174670           INC   MISCNT             :COUNT MISC COMMANDS
1321 005044' 012546                   MOV   (RS)+,-(SP)        :GET CODE
1322 005046' 100416                   BHI  SEL1                :CODE IS BAD
1323 005050' 021627 000003           CMP   (SP),#SDMAX        :IS CODE IN RANGE?
1324 005054' 101013                   BHI  SEL1                :NO
1325 005056' 010502                   MOV   RS,R2              :YES, SAVE RETURN ADDRESS
1326 005060' 012605                   MOV   (SP)+,RS          :GET CODE AND SELECT
1327 005062' 004767 006054           JSR   PC,SD              :MSG A
1328 005066' 016467 000034 172720   MOV   RPAR2(R4),MSGA     :MSG B
1329 005074' 016467 000036 172714   MOV   RPAR3(R4),MSGB
1330 005102' 000202                   RTS   R2
1331 005104' 004567 004106           SEL1: JSR   RS,ERRCS1     :PRINT INVALID MSG-SEL
1332 005110' 003575                   .WORD ISDMC-ERMBAS
1333 005112' 000177 172732           JMP   @CUPGER           :GO TO MPG'S ERROR RTN

```

; 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

; CMD FLAGWORD FORMAT:

;BIT 12 = 10000 = WRND COMMAND
;BIT 11 = 4000 = RECALIBRATE COMMAND
;BIT 10 = 2000 = TWO INTERRUPTS EXPECTED
;BIT 9 = 1000 = RDND COMMAND
;BIT 8 = 400 = DO ECC CORRECTION IF ALLOWED
;BIT 7 = 200 = PERFORM RETRIES ON CMD
;BIT 6 = 100 = OFFSET COMMAND
;BIT 5 = 040 = CMD TERMINATES WITH ATA
;BIT 4 = 020 = SET UP CYL/HD/SECT #
;BIT 3 = 010 = INCREMENT BYTE COUNTS
;BIT 2 = 004 = DATA TRANSFER CMD
;BIT 1 = 002 = 3 ARGUMENT CMD
;BIT 0 = 001 = 4 ARGUMENT CMD

1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390

```

005116 010067 174644      CMDCOM:  MOV   R0,CNTADR          ;SAVE ADR OF BYTE COUNT
005122 004767 177066      CMD1:   JSR   PC,ACQHSK         ;HOUSEKEEP THE DISK
005126 010167 174636          MOV   R1,CURFLG           ;SAVE FLAGND FOR TERMINATION
005132 056702 174656          BIS   RPCS1V,R2          ;MERGE CFMT
005136 010267 174630          MOV   R2,CURCMD         ;SAVE CURR CMD CODE
005142 032701 000003          BIT   #3,R1             ;THIS CMD HAVE BUS ADR & MD CNT?
005146 001440          BEQ   10$              ;Y,N-10$
005150 012567 174620          MOV   (R5)+,CURADR      ;STORE 2 WORD BUS ADR
005154 032701 001000          BIT   #1000,R1         ;IS THIS A RDND CMD
005160 001002          BNE   5$              ;YES
005162 012567 174610          MOV   (R5)+,CURADR+2
005166 012500          5$:   MOV   (R5)+,R0          ;GET BYTE COUNT
005170 032701 010000          BIT   #10000,R1        ;IS THIS A WRND CMD
005174 001405          BEQ   7$              ;NO
005176 032702 010000          BIT   #CFMT,R2         ;SHOULD USE 20 SECTORS
005202 001402          BEQ   7$              ;NO
005204 012700 000170          MOV   #120,R0          ;YES, GET NR BYTES OF HEADERS
005210 032701 001000          7$:   BIT   #1000,R1        ;IS THIS A RDND
005214 001002          BNE   8$              ;YES
005216 042700 000001          BIC   #1,R0            ;MAKE SURE ITS EVEN
005222 010067 174554          8$:   MOV   R0,CURPBC      ;SAVE POSITIVE BYTE CNT
005226 000241          CLC                    ;MAKE IT A WORD COUNT
005230 006000          POR   R0
005232 005400          NEG   R0
005234 010067 174540          MOV   R0,CURCNT        ;MAKE IT NEGATIVE
005240 032701 000001          BIT   #1,R1            ;SAVE IT
005244 001401          BEQ   10$              ;THERE A 4TH WORD?
005246 005725          TST   (R5)+            ;Y,N-10$
005250 032764 000100 000012 10$:  BIT   #VV,RPDS(R4)     ;BYPASS IT
005256 001004          BNE   20$              ;IS THE VV BIT SET?
                          ;N,Y-20$

```

```

1391 005260' 012714 000003      MOV      #SPCODE, (R4)      ;ISSUE PACK ACKNOWLEDGE CMND
1392 005264' 004767 005672      JSR      PC, DRVDLY
1393 005270' 004767 000176      JSR      PC, SUIORG      ;GO SET UP REGS FOR I/O
20S:                               ;THIS THE OFFSET CMND?
1394 005274' 032701 000100      BIT      #100, R1
1395 005300' 001403      BEQ      30S      ;Y, N-30S
1396 005302' 012500      MOV      (R5)+, R0      ;RETRIEVE OFFSET VALUE
1397 005304' 010064 000016      MOV      R0, RPAS(R4)      ;SETUP OFFSET
1398 005310' 016767 172476 174470 30S:  MOV      RTRY, CURRTY      ;INITIALIZE RETRY COUNT
1399 005316' 005067 174466      CLR      RTRYIP      ;CLEAR RETRY IN PROGRESS FLAG
1400 005322' 012767 003111 003552      MOV      #IOTERM-ERMBAS, INTEAD ;INIT TERMINATION ERROR MSG
1401 005330' 052713 000010      BIS      #MT4IOT, (R3)      ;SET WAITING FOR I/O TERM FLAG
1402 005334' 052767 000302 172440      BIS      #CMDISU!ANYIOI!DOTERM, DFLGMD ;SET CMND ISSUED FLAGS
1403 005342' 010214      MOV      R2, (R4)      ;ISSUE SPECIFIED CMND
1404 005344' 004767 005612      JSR      PC, DRVDLY      ;WAIT FOR DRIVE RESPONSE
1405 005350' 032767 000001 174444      BIT      #DRIB, DIAFLG      ;IS DRIVE BUSY?
1406 005356' 001413      BEQ      60S      ;NO
1407 005360' 042767 000001 174434      BIC      #DRIB, DIAFLG      ;YES, REMOVE COMMAND FROM MPG
1408 005366' 042767 000302 172406      BIC      #CMDISU!ANYIOI!DOTERM, DFLGMD
1409 005374' 004577 172446      JSR      RS, #CIBSY      ;ALLOW OTHERS TO RUN
1410 005400' 004767 005644      JSR      PC, DRVCLR      ;DO DRIVE CLEAR
1411 005404' 000646      BR      CMND1      ;REPEAT COMMAND
1412 005406' 032701 004000      60S:  BIT      #4000, R1      ;IS THIS A RECAL CMND
1413 005412' 001406      BEQ      61S      ;NO
1414 005414' 052767 000400 172360      BIS      #SPINFL, DFLGMD      ;YES, FLAG WAIT FOR RDY
1415 005422' 012763 005670 000030      MOV      #3000, PTOCNT(R3)      ;SETUP INT TIMER
1416 005430' 005767 172346      61S:  TST      DFLGMD      ;"NOWAIT" BIT SET?
1417 005434' 100011      BPL      40S      ;Y, N-40S
1418 005436' 042713 000010      BIC      #MT4IOT, (R3)      ;RESET WAITING FOR I/O TERM
1419 005442' 032701 004000      BIT      #4000, R1      ;IS THIS THE RECAL CMND
1420 005446' 001410      BEQ      50S      ;NO
1421 005450' 042767 000400 172324      BIC      #SPINFL, DFLGMD      ;YES
1422 005456' 000404      BR      50S      ;GO TO EXIT
1423 005460' 004577 172362      40S:  JSR      RS, #CIBSY      ;WAIT FOR I/O TO COMPLETE
1424 005464' 004767 003316      JSR      PC, PROCTM      ;GO PROCESS TERMINATION
1425 005470' 000205      50S:  RTS      ;EXIT IN-LINE TO USER PROG
1426
1427
1428
1429
1430                               ;SET UP DEVICE REGS FOR I/O
1431
1432                               ;JSR      PC, SUIORG      S/R CALL
1433
1434                               ;R4 = RPCS1 ADR
1435                               ;R3 = PROG TBL ADR
1436                               ;R2 = CMND CODE
1437                               ;R1 = CMND FLAGWORD
1438
1439                               ;DESTROYS R0
1440
1441 005472' 032701 000020      SUIORG: BIT      #20, R1      ;NEED TO SET UP CYL/HEAD/SECT?
1442 005476' 001412      BEQ      10S      ;Y, N-10S
1443 005500' 016764 172300 000020      MOV      CYL, RPDC(R4)      ;LOAD CYL #
1444 005506' 016746 172274      MOV      HEAD, -(SP)      ;GET HEAD #
1445 005512' 000316      SWAB      (SP)      ;PUT IN CORRECT BIT POSITION
1446 005514' 116716 172270      MOV      SECT, (SP)      ;SET IN SECT #

```


1447	005520	012664	000006		MOV	(SP)+,RPDA(R4)	:LOAD HEAD & SECT #'S
1448	005524	032701	000004	10S:	BIT	#4,R1	:DATA XFER CMD?
1449	005530	001414			BEQ	30S	:Y,N-30S
1450	005532	016700	174236		MOV	CURADR,R0	:GET HIGH BITS OF ADR
1451	005536	042700	177774		BIC	#177774,R0	:RESET BITS ABOVE A17
1452	005542	000300			SWAB	R0	:ALIGN BITS A16 & A17
1453	005544	050002			BIS	R0,R2	:SET THEM INTO CMD CODE WORD
1454	005546	016764	174224	000004	MOV	CURADR+2,RPBA(R4)	:LOAD BITS 0-15 OF ADR
1455	005554	016764	174220	000002	MOV	CURCNT,RPWC(R4)	:LOAD WORD COUNT
1456	005562	000207		30S:	RTS	PC	:EXIT IN-LINE

```

1458                                     ;"BADSEC" FUNCTION ROUTINE
1459
1460                                     ;JSR   RS,BADSEC
1461
1462 005564' 004767 003100      BADSEC: JSR   PC,CKDBSY      ;CHECK IF DISK IS BUSY
1463 005570' 005267 174140      INC   MISCNT
1464 005574' 004767 176414      JSR   PC,ACOMSK    ;HOUSEKEEP DISK
1465 005600' 010700      MOV   PC,R0        ;GET ADDR OF BYRD
1466 005602' 062700 174110      ADD   #BYRD+2-,R0
1467 005606' 010067 174154      MOV   R0,CNTADR
1468 005612' 016702 174176      MOV   RPCSIV,R2    ;SETUP RKCSI
1469 005616' 052702 000121      BIS   #RCODE!IE,R2 ; WITH READ COMMAND
1470 005622' 010267 174144      MOV   R2,CURCMD    ;SAVE COMMAND
1471 005626' 016300 000020      MOV   PRDIOA(R3),R0 ; IS RDIO LARGE ENOUGH
1472 005632' 016301 000016      MOV   PRDIOA(R3),R1 ; FOR ONE SECTOR
1473 005636' 160100      SUB   R1,R0
1474 005640' 020027 001000      CMP   R0,#NRBYTE
1475 005644' 103005      BHS   BAD1         ;YES
1476 005646' 004567 003344      JSR   R5,ERRCSI   ;NO, ALARM
1477 005652' 003635      .WORD
1478 005654' 000177 172170      JMP   3CUPGER
1479 005660' 032777 000001 172176  BAD1: BIT   #MVER,3CSYSFW ;USING MEMMGMT
1480 005666' 001422      BEQ   BAD11        ;NO
1481 005670' 032777 000040 172166      BIT   #UNIMAP,3CSYSFW ;USING UNIBUS MAP
1482 005676' 001407      BEQ   BAD12        ;NO
1483 005700' 016367 000244 174066      MOV   PRDIOV(R3),CURADR ;YES
1484 005706' 016367 000246 174062      MOV   PRDIOV+2(R3),CURADR+2
1485 005714' 000414      BR   BAD13
1486 005716' 016367 000240 174050  BAD12: MOV   PRDIOX(R3),CURADR
1487 005724' 016367 000242 174044      MOV   PRDIOX+2(R3),CURADR+2
1488 005732' 000405      BR   BAD13
1489 005734' 005067 174034      BAD11: CLR   CURADR
1490 005740' 016367 000016 174030      MOV   PRDIOA(R3),CURADR+2
1491 005746' 012767 000400 174024  BAD13: MOV   #NRWORD,CURCNT ;SETUP WORD COUNT
1492 005754' 005467 174020      NEG   CURCNT
1493 005760' 032764 000100 000012      BIT   #VV,RPDS(R4) ;IS VOLUME VALID
1494 005766' 001004      BNE   BAD14        ;YES
1495 005770' 012714 000003      MOV   #PACODE,(R4)
1496 005774' 004767 005162      JSR   PC,DRVDLY
1497 006000' 012767 000632 171776  BAD14: MOV   #NRCYL-1,CYL ;INIT ADDRESS
1498 006006' 012767 000002 171772      MOV   #NRHEAD-1,HEAD
1499 006014' 032767 010000 173772      BIT   #CFMT,RPCSIV
1500 006022' 001003      BNE   BAD141
1501 006024' 005067 171760      CLR   SECT
1502 006030' 000403      BR   BAD142
1503 006032' 012767 000001 171750  BAD141: MOV   #1,SECT
1504 006040' 042767 000001 171734  BAD142: BIC   #IOERR,DFLGWD
1505 006046' 012701 000635      MOV   #635,R1     ;GET COMMAND FLAG
1506 006052' 010167 173712      MOV   R1,CURFLG   ;AND SAVE
1507 006056' 004767 177410      JSR   PC,SUIORG   ;SET UP REGS
1508 006062' 005064 000016      CLR   RPAS(R4)
1509 006066' 016767 171720 173712      MOV   RTRY,CURTRY
1510 006074' 005067 173710      CLR   RTRYIP
1511 006100' 012767 003111 002774      MOV   #IOTERM-ERMBAS,INTEAD
1512 006106' 052713 000010      BIS   #WT4IOT,(R3) ;SET WAITING FLAG
1513 006112' 052767 000102 171662      BIS   #CMDISU!.DOTERM,DFLGWD

```

1514	006120'	010214			MOV	R2,(R4)		;ISSUE READ DATA
1515	006122'	004767	005034		JSR	PC,DRVDLY		
1516	006126'	032767	000001	173666	BIT	#DRIB,DIAFLG		;IS DRIVE BUSY
1517	006134'	001411			BEQ	BAD2		;NO
1518	006136'	042767	000001	173656	BIC	#DRIB,DIAFLG		;YES
1519	006144'	042767	000302	171630	BIC	#CMDISU!ANYIOI!DOTERM,DFLGND		
1520	006152'	004577	171670		JSR	RS,ACIOBSY		;ALLOW OTHERS TO RUN
1521	006156'	000730			BR	BAD142		;TRY AGAIN
1522	006158'	004577	171662		JSR	RS,ACIOBSY		;WAIT FOR XFER COMPLETE
1523	006164'	042767	000002	171610	BIC	#DOTERM,DFLGND		
1524	006172'	032767	000001	171602	BIT	#IOERR,DFLGND		;DID ERRORS OCCUR
1525	006200'	001414			BEQ	BAD3		;NO
1526	006202'	062767	000002	171600	ADD	#2,SECT		;YES, GET NEXT SECTOR
1527	006210'	026727	171574	000026	CMP	SECT,#22.		
1528	006216'	002710			BLT	BAD142		
1529	006220'	004567	002764		JSR	RS,ERRCS		;CAN NOT READ LAST TRACK
1530	006224'	003310			.WORD	CALT-ERMBAS		
1531	006226'	000177	171616		JMP	ACUPGER		;GO TO MPG'S ERROR RTN
1532	006230'	004767	000450		JSR	PC,CLRCOD		;CLEAR OUTPUT BUFFER FOR MSG
1533	006236'	004567	004636		JSR	RS,PRINTX		;PRINT HEADING
1534	006240'	002224			.WORD	BSH-PRIX1		
1535	006244'	000030			.WORD	BSHX-BSH		
1536	006246'	016304	000016		MOV	PRDIOA(R3),R4		;GET RDIO ADDR
1537	006250'	012400			MOV	(R4)+,R0		;GET CARTRIDGE #
1538	006254'	004577	171576		JSR	RS,ABINASC		; AND CONVERT TO ASCII
1539	006260'	007140			.WORD	PNUM2-		
1540	006262'	012400			MOV	(R4)+,R0		
1541	006264'	004577	171566		JSR	RS,ABINASC		
1542	006270'	007122			.WORD	PNUM1-		
1543	006272'	004567	004602		JSR	RS,PRINTX		;PRINT CARTRIDGE #
1544	006276'	002254			.WORD	PNUM-PRIX1		
1545	006300'	000024			.WORD	PNUMX-PNUM		
1546	006302'	005724			TST	(R4)+		;IGNORE UNUSED WORD OF DATA
1547	006304'	012400			MOV	(R4)+,R0		;IS PACK AN ALIGNMENT ONE
1548	006306'	001407			BEQ	BAD321		;NO
1549	006310'	004767	004010		JSR	PC,SUPTAD		;SETUP R3 & R4
1550	006314'	004567	002676		JSR	RS,ERRCS1		;YES, ALARM
1551	006320'	004053			.WORD	TIAP-ERMBAS		
1552	006322'	000177	171522		JMP	ACUPGER		;GO TO MPG ERROR RTN
1553	006326'	004767	000354		JSR	PC,CLRCOD		;CLR OUTPUT BUFFER FOR MSG
1554	006332'	010703			MOV	PC,R3		;GET ADDR OF OUTPUT BUFFER
1555	006334'	062703	005706		ADD	#COOFLD-.,R3		
1556	006340'	005067	000032		CLR	BAD32X		;FLAG NO BAD-SECTORS
1557	006344'	005067	173454		CLR	LIPCNT		
1558	006350'	012400			MOV	(R4)+,R0		;GET NEXT WORD OF DATA
1559	006352'	100512			BMI	BAD33		;DONE
1560	006354'	010367	000016		MOV	R3,BAD32X		;PACK CYL
1561	006360'	010701			MOV	PC,R1		;GET ADDR OF ASCII STORAGE
1562	006362'	062701	000014		ADD	#BAD32X-.,R1		
1563	006366'	160167	000004		SUB	R1,BAD32X		; AND MAKE RELATIVE
1564	006372'	004577	171460		JSR	RS,ABINASC		;CONVERT CYL TO ASCII
1565	006376'	000000			.WORD	XXX		;STORE RELATIVE ADDR HERE
1566	006400'	062703	000006		ADD	#6,R3		;GET ADDR OF NEXT VALUE
1567	006404'	112723	000054		MOVB	#,(R3)+		;PACK COMMA
1568	006410'	116400	000001		MOVB	1(R4),R0		;PACK TRACK
1569	006414'	010367	000016		MOV	R3,BAD32Y		;GET REL ADDR FOR DATA

```

1570 006420' 010701          MOV      PC,R1
1571 006422' 062701 000014  ADD      #BAD32Y--,R1
1572 006426' 160167 000004  SUB      R1,BAD32Y
1573 006432' 004577 171420  JSR      R5,JBINASC          ; CONVERT TRACK TO ASCII
1574 006436' 000000          .WORD   XXXX                ; STORE REL ADDR HERE
1575 006440' 062703 000006  ADD      #6,R3
1576 006444' 112723 000054  MOV      #' ,(R3)+          ; PACK COMMA
1577 006450' 012400          MOV      (R4)+,R0          ; PACK SECTOR
1578 006452' 042700 177400  BIC      #177400,R0        ; REMOVE TRACK
1579 006456' 010367 000016  MOV      R3,BAD32Z        ; GET REL ADDR FOR DATA
1580 006462' 010701          MOV      PC,R1
1581 006464' 062701 000014  ADD      #BAD32Z--,R1
1582 006470' 160167 000004  SUB      R1,BAD32Z
1583 006474' 004577 171356  JSR      R5,JBINASC          ; CONVERT SECTOR TO ASCII
1584 006500' 000000          .WORD   XXXX
1585 006502' 062703 000006  ADD      #6,R3
1586 006506' 112723 000040  MOV      #' ,(R3)+          ; PACK
1587 006512' 112723 000040  MOV      #' ,(R3)+          ; SPACES
1588 006516' 005267 173302  INC      LUPCNT
1589 006522' 026727 173276 000003  CMP      LUPCNT,#3        ; IS LINE RDY FOR PRINTING
1590 006530' 001307          BNE      BAD32Z            ; NO
1591 006532' 005767 177640  TST      BAD32X            ; WERE ANY BAD SEC FOUND
1592 006536' 001447          BEQ      BAD4              ; NO
1593 006540' 032767 000040 173254  BIT      #PBSH,DIAFLG      ; SHOULD PRINT HEADING
1594 006546' 001007          BNE      BAD32Z            ; NO
1595 006550' 052767 000040 173244  BIS      #PBSH,DIAFLG      ; YES
1596 006556' 004567 004316  JSR      R5,PRINTX          ; PRINT HEADING
1597 006562' 002300          .WORD   BSHEAD-PRIX1
1598 006564' 000024          .WORD   BSHEX-BSHEAD
1599 006566' 004567 004306  BAD32Z: JSR      R5,PRINTX          ; PRINT ONE LINE
1600 006572' 001112          .WORD   COOFLD-PRIX1
1601 006574' 000102          .WORD   66
1602 006576' 000653          BR       BAD321
1603 006600' 005767 177572  BAD33: TST      BAD32X          ; WERE BAD SEC FOUND
1604 006604' 001424          BEQ      BAD4              ; NO
1605 006606' 032767 000040 173206  BIT      #PBSH,DIAFLG      ; SHOULD PRINT HEADING
1606 006614' 001007          BNE      BAD331            ; NO
1607 006616' 052767 000040 173176  BIS      #PBSH,DIAFLG      ; YES
1608 006624' 004567 004250  JSR      R5,PRINTX          ; PRINT HEADING
1609 006630' 002300          .WORD   BSHEAD-PRIX1
1610 006632' 000024          .WORD   BSHEX-BSHEAD
1611 006634' 004567 004240  BAD331: JSR      R5,PRINTX          ; YES, PRINT ONE LINE
1612 006640' 001112          .WORD   COOFLD-PRIX1
1613 006642' 000102          .WORD   66
1614 006644' 004567 004230  JSR      R5,PRINTX          ; PRINT END OF BAD SEC
1615 006650' 002367          .WORD   E0BS-PRIX1
1616 006652' 177753          .WORD   E0BS-E0BSX
1617 006654' 000404          BR       BAD41
1618 006656' 004567 004216  BAD4:  JSR      R5,PRINTX          ; PRINT "NONE"
1619 006662' 002324          .WORD   NONE-PRIX1
1620 006664' 177771          .WORD   NONE-NONEX
1621 006666' 004767 003432  BAD41: JSR      PC,SUPTAD          ; SETUP R3 & R4
1622 006672' 004767 002110  JSR      PC,PROCTM         ; PROCESS TERMINATION
1623 006676' 042767 000040 173116  BIC      #PBSH,DIAFLG
1624 006704' 000205          RTS
1625

```

```

1626 ;SUBROUTINE TO ZERO BUFFER USED TO CONTAIN ASCII OF BAD SECTOR INFORMATION
1627
1628 006706' 012700 000041 CLRC0D: MOV #33, R0
1629 006712' 010701 MOV PC, R1 ;GET ADDR OF BUFFER
1630 006714' 062701 005326 ADD #C0DFLD-., R1
1631 006720' 005021 CLRC1: CLR (R1)+
1632 006722' 005300 DEC R0
1633 006724' 001375 BNE CLRC1
1634 006726' 000207 RTS PC

```

```

1636 .SBTTL RK06 INTERRUPT SERVICE ROUTINE
1637
1638 006730' 004067 003336 RHPINT: JSR R0, SAVREG ;SAVE ALL REGISTERS
1639 006734' 004567 003404 JSR R5, S1STAT ;GO STORE ALL DEV REG'S
1640 006740' 172650 .WORD ISTAT-
1641 006742' 005267 173014 INC INTCNT ;ADD 1 TO INTERRUPT CNT
1642 006746' 004767 003352 JSR PC, SUPTAD ;SET UP PROG TBL & RPCS1 ADR'S
1643 006752' 004767 004204 JSR PC, DRVDLY
1644 006756' 005767 172640 TST IRKDS ;IS REGISTER VALID?
1645 006762' 100420 BMI RHP1 ;YES
1646 006764' 032701 002000 BIT #2000,R1 ;IS THIS 2ND INTERRUPT
1647 006770' 001406 BEQ RHP11 ;NO, ALARM
1648 006772' 042701 002000 BIC #2000,R1 ;YES, SELECT DRIVE
1649 006776' 005005 CLR R5
1650 007000' 004767 004136 JSR PC, SD
1651 007004' 000407 BR RHP1 ;CHECK DI & ATN BITS
1652 007006' 012767 003370 002066 RHP11: MOV #NSVAL-ERMBAS, INTEAD ;ALARM
1653 007014' 052767 000004 173000 BIS #NDC, DIAFLG ;FLAG NEED DRIVE CLR
1654 007022' 000467 BR RHP6
1655 007024' 032767 000001 172570 RHP1: BIT #DRA, IRKDS ;IS DRIVE AVAILABLE?
1656 007032' 001006 BNE RHP2 ;YES
1657 007034' 042713 000010 BIC #WT4IOT (R3) ;CLEAR WAITING FOR I/O
1658 007040' 052767 000001 172754 BIS #DRIB, DIAFLG ;FLAG DRIVE IS BUSY
1659 007046' 000554 BR RHP62 ;EXIT UNTIL LATER
1660 007050' 016701 172714 RHP2: MOV CURFLG, R1 ;GET THIS CMD'S FLGWD
1661 007054' 005714 TST (R4) ;IS 'CERR' BIT SET?
1662 007056' 100554 BMI RHP3 ;YES
1663 007060' 016700 172542 MOV IRKAS, R0 ;GET ATA BITS
1664 007064' 000300 SWAB R0
1665 007066' 032701 000040 BIT #40, R1 ;IS COMMAND SUPPOSED TO SET ATA?
1666 007072' 001422 BEQ RHP24 ;NO, CHECK EXTRA ATA
1667 007074' 032767 040000 172506 BIT #DI, ICS1 ;IS DI SET?
1668 007102' 001007 BNE RHP23 ;YES, CHECK FOR ATA BIT
1669 007104' 032701 002000 BIT #2000, R1 ;IS ANOTHER INTERRUPT TO ARRIVE
1670 007110' 001133 BNE RHP62 ;YES, WAIT FOR 2ND INT
1671 007112' 012767 003157 001762 MOV #NODI-ERMBAS, INTEAD ;NO, SETUP ERROR MSG
1672 007120' 000430 BR RHP6
1673 007122' 036700 175256 RHP23: BIT MYATA, R0 ;IS MY ATA SET?
1674 007126' 001016 BNE RHP211 ;YES, TEST FOR ERRORS
1675 007130' 012767 003137 001744 MOV #NOATA-ERMBAS, INTEAD ;NO, ALARM
1676 007136' 000421 BR RHP6
1677 007140' 036700 175240 RHP24: BIT MYATA, R0 ;IS MY ATA SET?
1678 007144' 001412 BEQ RHP21 ;NO, TEST FOR ERRORS
1679 007146' 012767 003176 001726 MOV #UXPATA-ERMBAS, INTEAD ;YES, ALARM
1680 007154' 052767 000004 172640 BIS #NDC, DIAFLG ;FLAG NEED DRIVE CLR
1681 007162' 000407 BR RHP6
1682 007164' 052767 000004 172630 RHP211: BIS #NDC, DIAFLG ;FLAG NEED DRIVE CLR
1683 007172' 052767 000002 172622 RHP21: BIS #NEE, DIAFLG ;FLAG NO ERRORS EXPECTED
1684 007200' 000506 BR RHP313
1685
1686 ;EXIT
1687
1688 007202' 052767 000001 170572 RHP6: BIS #IOERR, DFLGWD ;SET TERM I/O ERROR FLAG
1689 007210' 042713 000010 RHP61: BIC #WT4IOT (R3) ;RESET WAITING FOR I/O TERM
1690 007214' 042767 000002 172600 BIC #NEE, DIAFLG
1691 007222' 032767 000004 172572 BIT #NDC, DIAFLG ;SHOULD DO A DRIVE CLR

```

```

1692 007230' 001405          BEQ      RHP612          ;NO
1693 007232' 004767 004012  JSR      PC,DRVCLR      ;YES
1694 007236' 042767 000004 172556  BIC      #NDC,DIAFLG
1695 007244' 032767 000010 172550  RHP612: BIT      #NRC,DIAFLG      ;DO I NEED A RECAL
1696 007252' 001426          BEQ      RHP613          ;NO
1697 007254' 016446 000010  MOV      RPCS2(R4),-(SP) ;YES
1698 007260' 052764 000040 000010  BIS      #SCLR,RPCS2(R4)
1699 007266' 004767 003670  JSR      PC,DRVDLY
1700 007272' 012664 000010  MOV      (SP)+,RPCS2(R4)
1701 007276' 012714 000013  MOV      #RCCODE,(R4)      ; OUTPUT RECAL
1702 007302' 052767 000400 170472  BIS      #SPINFL,DFLGWD    ;PREPARE FOR INT TIMEOUT
1703 007310' 042767 000010 172504  BIC      #NRC,DIAFLG      ; TO TEST DRY BIT
1704 007316' 012763 005670 000030  MOV      #3000,PTCNT(R3)  ; WHEN HEADS ARE HOME
1705 007324' 052713 000010  BIS      #WT410T,(R3)     ;FLAG STAY IN MPG UNTIL DONE
1706 007330' 032767 000020 172464  RHP613: BIT      #NCC,DIAFLG ;DO NEED CONTROL CLR
1707 007336' 001407          BEQ      RHP611          ;NO
1708 007340' 042767 000020 172454  BIC      #NCC,DIAFLG      ;YES
1709 007346' 004767 003634  JSR      PC,CTRLCL
1710 007352' 004767 003672  JSR      PC,DRVCLR
1711 007356' 032767 000004 172404  RHP611: BIT      #4,CURFLG  ;THIS A DATA TRANSFER CMND?
1712 007364' 001403          BEQ      RHP614          ;NO
1713 007366' 016467 000002 172410  MOV      RPMC(R4),FINCNT  ;SAVE FINAL WORD COUNT
1714 007374' 042714 000100  RHP614: BIC      #IE,(R4)  ;REMOVE IE
1715 007400' 004067 002702  RHP62:  JSR      RD,RESREG  ;RESTORE ALL REGISTERS
1716 007404' 000177 170464  JMP      #RTNINT         ;EXIT FROM INTERRUPT
1717
1718 ;ANALYZE ERROR CONDITION
1719
1720 007410' 052767 000020 172404  RHP3:  BIS      #NCC,DIAFLG  ;FLAG NEED CCLR
1721 007416' 032767 010420 172200  RHP313: BIT      #DTE!HVRC!FMTE,IRKER ;DID OTHER ERRORS OCCUR?
1722 007424' 001141          BNE      CKRTRY         ;YES, CHECK RETRIES
1723 007426' 032767 100000 172170  BIT      #DCK,IRKER     ;DID DATA ERROR OCCUR?
1724 007434' 001402          BEQ      RHP315          ;NO
1725 007436' 000167 000530  JMP      CKCORR
1726 007442' 032767 024000 172140  RHP315: BIT      #CTO!SPAR,ICS1  ;DID CTO OR SPAR ERRORS OCCUR?
1727 007450' 001402          BEQ      RHP31          ;NO
1728 007452' 000167 000472  JMP      HARDER         ;YES
1729 007456' 032767 037400 172134  RHP31:  BIT      #UFE!MDS!PGE!NEM!NED!UPE,ICS2 ;ANY RKCS2 ERRORS?
1730 007464' 001402          BEQ      RHP311          ;NO
1731 007466' 000167 000456  JMP      HARDER         ;YES
1732 007472' 032767 140000 172120  RHP311: BIT      #DLT!WCE,ICS2  ;DID DLT OR WCE OCCUR?
1733 007500' 001113          BNE      CKRTRY         ;YES, CHECK RETRIES
1734 007502' 032767 004070 172112  BIT      #ACLO!DSL!DROT!WRL,IRKDS ;ANY RKDS ERRORS?
1735 007510' 001402          BEQ      RHP312          ;NO
1736 007512' 000167 000432  JMP      HARDER         ;YES
1737 067357          RKERBT=ILF!SKI!NXF!DRPAR!DYE!ECH!BSE!COE!IDAE!WLE!OPI!UNS
1738 007516' 032767 067357 172100  RHP312: BIT      #RKERBT,IRKER ;ANY RKER ERRORS?
1739 007524' 001402          BEQ      RHP314          ;NO
1740 007526' 000167 000416  JMP      HARDER         ;YES
1741 007532' 032767 000002 172262  RHP314: BIT      #NEE,DIAFLG  ;ARE ERRORS EXPECTED
1742 007540' 001010          BNE      RHP4           ;NO
1743 007542' 012767 003340 001332  MOV      #LINKERR-ERMBAS,INTEAD ;SETUP FOR UNKNOWN ERROR
1744 007550' 052767 000020 172244  BIS      #NCC,DIAFLG      ;FLAG NEED CCLR
1745 007556' 000167 177420  JMP      RHP6
1746
1747 ;SERVICE RDHD COMMAND

```

1748										
1749	007562	032701	001000		RHP4:	BIT	#1000,R1			: IS THIS A RMD COMMAND
1750	007566	001610				BEQ	RHP61			: NO, FINISHED
1751	007570	032767	000200	172022		BIT	#OR,ICS2			: IS DATA OUTPUT RDY
1752	007576	001005				BNE	RHP41			: YES, GET HEADER DATA
1753	007600	012767	003521	001274		MOV	#ONRRH-ERMBAS,INTEAD			: NO, ALARM
1754	007606	000167	177370			JMP	RHP6			
1755	007612	016700	172156		RHP41:	MOV	CURADR,RO			: GET STORAGE ADDRESS
1756	007616	016701	172012			MOV	IRKOB,R1			: GET 1ST HEADER WORD
1757	007622	004777	170252			JSR	PC,@PUTBYT			: LET MPG STORE IT
1758	007626	000301				SWAB	R1			
1759	007630	004777	170244			JSR	PC,@PUTBYT			
1760	007634	016401	000024			MOV	RPOB(R4),R1			: GET 2ND HEADER WORD
1761	007640	004777	170234			JSR	PC,@PUTBYT			: LET MPG STORE IT
1762	007644	000301				SWAB	R1			
1763	007646	004777	170226			JSR	PC,@PUTBYT			
1764	007652	016401	000024			MOV	RPOB(R4),R1			: GET 3RD HEADER WORD
1765	007656	004777	170216			JSR	PC,@PUTBYT			: LET MPG STORE IT
1766	007662	000301				SWAB	R1			
1767	007664	004777	170210			JSR	PC,@PUTBYT			
1768	007670	062767	000006	172014		ADD	#6,BYRD+2			: COUNT NR BYTES READ
1769	007676	005567	172006			ADC	BYRD			
1770	007702	062767	000006	170110		ADD	#6,SIZE			
1771	007710	005367	172066			DEC	CURPBC			: SHOULD I GET MORE HEADERS
1772	007714	003002				BGT	RHP42			: YES
1773	007716	000167	177266			JMP	RHP61			: NO
1774	007722	010067	172046		RHP42:	MOV	RO,CURADR			: SAVE STORAGE ADDR
1775	007726	000476				BR	CKR1			: YES
1776										
1777										
1778										: DETERMINE WHICH ERROR SHOULD BE RETRIED
1779	007730	005767	172054		CKRTRY:	TST	RTRYIP			: ALREADY DONE RETRIES ON THIS CMND?
1780	007734	001054				BNE	55\$: N,Y-55\$
1781	007736	005767	172044			TST	CURTRY			: ARE RETRIES SPECIFIED?
1782	007742	001502				BEQ	HARDER			: Y,N-HARDER
1783	007744	032767	100000	171646		BIT	#DLT,ICS2			: DLT ERROR?
1784										
1785	007752	001403				BEQ	42\$: Y,N-42\$
1786	007754	005267	171764			INC	DLTCNT			: ADD 1 TO DLT COUNT
1787	007760	000442				BR	55\$: GO CK RETRY COUNT
1788	007762	032767	010000	171634	42\$:	BIT	#DTE,IRKER			: DTE ERROR?
1789	007770	001403				BEQ	44\$: Y,N-44\$
1790	007772	005267	171750			INC	DTECNT			: ADD 1 TO DTE COUNT
1791	007776	000433				BR	55\$: GO CK RETRY COUNT
1792	010000	032767	000400	171616	44\$:	BIT	#HVRC,IRKER			: HVRC ERROR?
1793	010006	001403				BEQ	46\$: Y,N-46\$
1794	010010	005267	171734			INC	HCRCNT			: ADD 1 TO HCRC COUNT
1795	010014	000424				BR	55\$: GO CK RETRY COUNT
1796	010016	032767	000020	171600	46\$:	BIT	#FMTE,IRKER			: FER ERROR?
1797	010024	001403				BEQ	50\$: NO
1798	010026	005267	171720			INC	FERCNT			: ADD 1 TO FER COUNT
1799	010032	000415				BR	55\$: GO CK RETRY COUNT
1800	010034	032767	100000	171562	50\$:	BIT	#DCK,IRKER			: DCK ERROR?
1801	010042	001403				BEQ	52\$: Y,N-52\$
1802	010044	005267	171704			INC	DCKCNT			: ADD 1 TO DCK COUNT
1803	010050	000406				BR	55\$: GO CK RETRY COUNT

1804	010052'	032757	040000	171540	52S:	BIT	#WCE,ICS2	:WCE ERROR?
1805	010060'	001402				BEQ	55S	:Y N-55S
1806	010062'	005267	171670			INC	WCECNT	:ADD 1 TO WCE COUNT
1807	010066'	005367	171714		55S:	DEC	CURTRY	:DECREMENT RETRY COUNT
1808	010072'	100004				BPL	60S	:CNT EXHAUSTED? (Y N-60S)
1809	010074'	012767	003215	001000		MOV	#RTYEXH-ERMBAS,INTEAD	:SET UP EXHAUSTED RETRIES ERR MSG ADR
1810	010102'	000431				BR	JSETER	:GO TO ERROR EXIT
1811	010104'	005267	171650		60S:	INC	RETRY5	:ADD 1 TO RETRY TOTAL CNT
1812	010110'	005267	171674			INC	RTRYIP	:SET RETRY IN PROGRESS FLAG
1813	010114'	004767	003066			JSR	PC,CTRLCL	:CLEAR CONTROLLER
1814	010120'	004767	003124			JSR	PC,DRVCLR	:CLEAR DRIVE
1815	010124'	016702	171642		CKR1:	MOV	CURCMD,R2	:GET CURR CMD IN R2
1816	010130'	004767	175336			JSR	PC,SUIORG	:SET UP DEV REGS
1817	010134'	012763	072460	000030		MOV	#ITIME,PTCNT(R3)	:MSKP T/O COUNT
1818	010142'	010214				MOV	R2,(R4)	:RE-ISSUE THE ORIG CMD
1819	010144'	000167	177230			JMP	RMP62	:GO TO INT EXIT
1820								
1821	010150'	032767	002002	171446	HARDER:	BIT	#IDAE!SKI,IRKER	:DID ADDRESSING ERROR OCCUR
1822	010156'	001403				BEQ	JSETER	:NO
1823	010160'	052767	000010	171634		BIS	#NRC,DIAFLG	:YES, FLAG NEED RECAL
1824	010166'	000167	177010		JSETER:	JMP	RMP6	:GO TO ERROR EXIT
1825								
1826	010172'	032767	002000	167602	CKCORR:	BIT	#CORFLG,DFLGD	:CORON STATEMENT IN EFFECT?
1827	010200'	001253				BNE	CKRTRY	:Y N-CKRTRY
1828	010202'	032714	010000			BIT	#CFMT,(R4)	:IN 18 BIT WORD MODE?
1829	010206'	001250				BNE	CKRTRY	:N Y-CKRTRY
1830	010210'	032764	000100	000014		BIT	#ECH,RPER(R4)	:IS THIS A HARD ERROR
1831	010216'	001244				BNE	CKRTRY	:YES, RETRY BUT NO CORRECTION
1832	010220'	005764	000032			TST	RPEC2(R4)	:ECC BIT PATTERN = 0?
1833	010224'	001004				BNE	70S	:Y N-70S
1834	010226'	012767	003237	000646		MOV	#INVPAT-ERMBAS,INTEAD	:SET UP INV BIT PATTERN ERR MSG ADR
1835	010234'	000754				BR	JSETER	:GO TO ERROR EXIT
1836	010236'	026427	000030	010041	70S:	CHP	RPEC1(R4),#10041	:ECC BIT POSITION TOO BIG?
1837	010244'	101404				BLOS	80S	:Y N-80S
1838	010246'	012767	003263	000626		MOV	#INVPOS-ERMBAS,INTEAD	:SET UP INV BIT POSITION ERR MSG ADR
1839	010254'	000744				BR	JSETER	:GO TO ERROR EXIT
1840	010256'	016405	000002		80S:	MOV	RPMC(R4),R5	:GET REMAINING NEG WORD CNT
1841	010262'	006305				ASL	R5	:MAKE IT A BYTE CNT
1842	010264'	066705	171512			ADD	CURPBC,R5	:ADD IN ORIG POS BYTE CNT
1843	010270'	001617				BEQ	CKRTRY	:ANY BYTES READ? (Y N-CKRTRY)
1844	010272'	005001				CLR	R1	:RESET DATA START DISPL VALUE
1845	010274'	012746	001000			MOV	#512,-(SP)	:INITIALIZE BLOCK LENGTH
1846	010300'	020516			85S:	CHP	R5,(SP)	:REACHED START OF BAD BLOCK?
1847	010302'	101403				BLOS	90S	:N Y-90S
1848	010304'	061601				ADD	(SP),R1	:ADD BLK LNTH TO DATA DISPL
1849	010306'	151605				SUB	(SP),R5	:REDUCE BYTE READ CNT
1850	010310'	000773				BR	85S	:GO CK IT NOW
1851	010312'	005726			90S:	TST	(SP)+	:CORRECT STACK
1852	010314'	016400	000030			MOV	RPEC1(R4),R0	:GET ECC BIT POSITION VALUE
1853	010320'	005300				DEC	R0	:ADJ IT FOR SHIFTING
1854	010322'	010002				MOV	R0,R2	:SET UP A SHIFT COUNT
1855	010324'	042702	177760			BIC	#177760,R2	:ISOLATE SHIFT CNT BITS
1856	010330'	040200				BIC	R2,R0	:CLEAR SHIFT CNT BITS IN BIT POSITION
1857	010332'	006200				ASR	R0	:CONVERT BIT POSITION TO A WORD
1858	010334'	006200				ASR	R0	:DISPL INTO THE DATA IN THIS
1859	010336'	006200				ASR	R0	:BLOCK

1860	010340'	060001		ADD	R0,R1	:ADD IT TO DATA DISPL VALUE
1861	010342'	020167	171434	CMP	R1,CURPBC	:CORRECTION WITHIN USER'S DATA?
1862	010346'	103026		BHIS	1108	:Y,N-1108
1863	010350'	005267	171364	INC	CECCER	:ADD 1 TO CORRECTABLE ECC ERROR CNT
1864	010354'	016405	000032	MOV	RPEC2(R4),R5	:GET ECC BIT PATTERN
1865	010360'	005046		CLR	-(SP)	:CLEAR SHIFT INTO WORD
1866	010362'	005302		DEC	R2	:DECREMENT SHIFT COUNT
1867	010364'	002403		BLT	1008	:FINISHED SHIFTING? (N,Y-1008)
1868	010366'	006305		ASL	R5	:DO A BIT SHIFT ON 2 WORDS
1869	010370'	006116		ROL	(SP)	
1870	010372'	000773		BR	958	:GO CK SHIFT CNT
1871	010374'	004767	000052	JSR	PC,DOCORR	:CORRECT 1ST WORD OF PAIR
1872	010400'	012605		MOV	(SP)+,R5	:GET 2ND CORRECTION WORD
1873	010402'	052701	000002	ADD	R2,R1	:INCR DATA DISPL VALUE
1874	010406'	020167	171370	CMP	R1,CURPBC	:STILL WITHIN HIS DATA?
1875	010412'	103002		BHIS	1058	:Y,N-1058
1876	010414'	004767	000032	JSR	PC,DOCORR	:CORRECT 2ND WORD OF PAIR
1877	010420'	004767	001700	JSR	PC,SUPTAD	:SET UP R3 & R4 AGAIN
1878	010424'	005764	000002	TST	RPMC(R4)	:ANY MORE DATA TO READ?
1879	010430'	001002		BNE	1158	:Y,N-1158
1880	010432'	000167	176552	JMP	RHP61	:GO CLEAR WAIT FLG & EXIT
1881	010436'	004767	002544	JSR	PC,CTRLCL	:ISSUE CTRL CLR CMD
1882	010442'	016714	171324	MOV	CURCMD,(R4)	:RESUME ORIG CMD
1883	010446'	000167	176726	JMP	RHP62	:GO TO INT EXIT

:DO ECC DATA CORRECTION

```

:JSR PC,DOCORR      S/R CALL
:
:R1 = DATA WORD DISPLACEMENT
:R5 = BIT PATTERN

```

1894	010452'	016703	171316	DOCORR:	MOV	CURADR,R3	:GET DATA'S ABS STARTING ADR
1895	010456'	016704	171314		MOV	CURADR+2,R4	
1896	010462'	060104			ADD	R1,R4	:ADD DATA DISPL INTO IT
1897	010464'	005503			ADC	R3	
1898	010466'	032777	000001		BIT	SMYVER,ACSYSFW	:RUNNING UNDER MEM MGMT?
1899	010474'	001460			BEG	208	:Y,N-208
1900	010476'	032777	000040		BIT	SMUNMAP,ACSYSFW	:IS UNIBUS MAP USED?
1901	010504'	001430			BEG	58	:NO
1902	010506'	010446			MOV	R4, -(SP)	:YES, GET UPPER 5 BITS
1903	010510'	006203			ASR	R3	:OF VIRTUAL ADDRESS
1904	010512'	006016			ROR	(SP)	
1905	010514'	006203			ASR	R3	:MERGE BIT 16 WITH BITS15-13
1906	010516'	006016			ROR	(SP)	:PUT BIT 17 IN CARRY
1907	010520'	006216			ASR	(SP)	:MERGE BIT 17 IN WITH BITS16-13
1908	010522'	042716	101777		BIC	0101777,(SP)	:POSITION SUCH THAT IS X4
1909	010526'	000316			SWAB	(SP)	:REMOVE ANY SIGN BIT
1910	010530'	062716	170200		ADD	08USMAP,(SP)	:MOVE 5 BITS TO LOWER BYTE
1911	010534'	010446			MOV	R4, -(SP)	:GET BASE ADR OF REG
1912	010536'	017604	000002		MOV	22(SP),R4	:SAVE ORIGINAL OFFSET
1913	010542'	062766	000002	000002	ADD	22(SP),R3	:GET CONTENTS OF MAP REGS
1914	010550'	017603	000002		MOV	22(SP),R3	
1915	010554'	042716	160000		BIC	0160000,(SP)	:MERGE IN ORIGINAL OFFSET

```

1916 010560' 062604          ADD      (SP)+,R4
1917 010562' 005503          ADC      R3
1918 010564' 005726          TST     (SP)+
1919 010566' 010400          SS:     MOV     R4,R0
1920 010570' 042704 177700   BIC     @177700,R4
1921 010574' 052704 100000   BIS     @P4CONS,R4
1922 010600' 012702 000006   MOV     @6,R2
1923 010604' 006203          10S:    ASR     R3
1924 010606' 006000          ROR     R0
1925 010610' 005302          DEC     R2
1926 010612' 001374          BNE     10S
1927 010614' 013746 172350   MOV     @#KPAR4,-(SP)
1928 010620' 013746 172310   MOV     @#KPDY4,-(SP)
1929 010624' 010037 172350   MOV     R0,@#KPAR4
1930 010630' 012737 077406 172310   MOV     @PDRCON,@#KPDY4
1931 010636' 011400          20S:    MOV     (R4),R0
1932 010640' 040514          BIC     R5,(R4)
1933 010642' 040005          BIC     R0,R5
1934 010644' 050514          BIS     R5,(R4)
1935 010646' 032777 000001 167210   BIT     @#MVER,@CSYSFW
1936 010654' 001404          BEQ     30S
1937 010656' 012637 172310   MOV     (SP)+,@#KPDY4
1938 010662' 012637 172350   MOV     (SP)+,@#KPAR4
1939 010666' 000207          30S:    RTS     PC

```

```

;CORRECT STACK
;SAVE LOW 16 BITS IN WORK AREA
;ISOLATE UP TO 32 WORD OFFSET
;SET ADR TO SELECT PAGE 4
;SET UP SHIFT CNT
;SHIFT ADR 1 BIT

;GOT HI 16 BITS IN 1 WORD?
;Y,N-10S
;SAVE PAGE 4 STUFF

;SET PAGE 4 TO USER'S AREA

;GET DATA WORD READ
;RESET PATTERN'S BITS IN WORD READ
;RESET WORD READ BITS IN PAT WORD
;SET REMAINING SINGLE BITS BACK IN
;RUNNING UNDER MEM MGMT?
;Y,N-30S
;RESTORE PAGE 4'S REGS

;EXIT IN-LINE

```

.SBTTL SUBROUTINES FOR RK06 FUNCTION ROUTINES

;CHECK IF DEVICE IS BUSY AND WAIT IF IT IS

```

:JSR PC,CKDBSY S/R CALL
:DESTROYS R0,R3,R4
:ON EXIT: R3 = PROG TBL ADR
:         R4 = RPCS1 ADR
    
```

1961	010670	004767	001430		CKDBSY:	JSR	PC,SUPTAD		;SET UP PROG TBL & RPCS1 ADR'S
1962	010674	032714	000100		10S:	BIT	81E,(R4)		;INT ENABLE ON?
1963	010700	001403				BEQ	20S		;Y,N-20S
1964	010702	004577	167140		15S:	JSR	R5,2C10BSY		;RELEASE CONTROL
1965	010706	000772				BR	10S		;GO CK AGAIN
1966	010710	032767	000400	167064	20S:	BIT	8SPINFL,DFLGMD		;DO I HAVE AN SPIN IN PROGRESS?
1967	010716	001403				BEQ	25S		;Y,N-25S
1968	010720	052713	000010			BIS	8MT4IOT,(R3)		;SET WAITING FOR I/O TERM
1969	010724	000766				BR	15S		;GO RELEASE CONTROL
1970	010726	032767	000002	167046	25S:	BIT	800TERM,DFLGMD		;HAVE TO PROCESS PREV TERMINATION?
1971	010734	001403				BEQ	30S		;Y,N-30S
1972	010736	004767	000044			JSR	PC,PROCTM		;GO PROCESS TERMINATION
1973	010742	000754				BR	10S		;GO CK INT ENABLE AGAIN
1974	010744	016767	167056	000012	30S:	MOV	IVCTAD,40S		;STORE INT VECTOR ADR
1975	010752	016767	167052	000006		MOV	PSWD,45S		;STORE PROC STATUS WORD
1976	010760	004577	167102			JSR	R5,2SETVEC		;GO SET UP INTERRUPT VECTOR
1977	010764	000000			40S:	.WORD	X00X		;INT VECTOR ADR
1978	010766	000000			45S:	.WORD	X00X		;PSW
1979	010770	175740				.WORD	RHPINT-		;REL INT ROUT ADR
1980	010772	010567	170766			MOV	R5,ERRADR		;SAVE CURR USER STMT ADR
1981	010776	162767	000004	170760		SUB	84,ERRADR		
1982	011004	000207				RTS	PC		;EXIT IN-LINE

;PROCESS TERMINATION OF PREVIOUS I/O FUNCTION

```

:JSR PC,PROCTM S/R CALL
:R3 = PROG TABLE ADR
:DESTROYS R0
    
```

1984	011006	010746			PROCTM:	MOV	R1,-(SP)		;SAVE R1 & R2
1985	011010	010246				MOV	R2,-(SP)		
1986	011012	042767	000002	166762		BIC	800TERM,DFLGMD		;RESET PROCESS TERMINATION FLAG
1987	011020	032767	000010	170742		BIT	810,CURFLG		;INCR BYTE COUNT?
1988	011024	001417				BEQ	20S		;Y,N-20S
1989	011030	016700	170744			MOV	CURCNT,R0		;GET INITIAL WORD CNT
1990	011034	005400				NEG	R0		;MAKE IT POSITIVE AGAIN
1991	011036	016701	170742			MOV	FINCNT,R1		;GET FINAL WORD CNT
1992	011042	100001				BPL	10S		;IS IT NEGATIVE? (Y,N-10S)
1993	011044	005401				NEG	R1		;MAKE IT POSITIVE
1994	011046	160100			10S:	SUB	R1,R0		;SUB REMAINING CNT FROM INITIAL CNT
1995	011050	006300				ASL	R0		;MAKE IT A BYTE CNT
1996	011052	010067	166742			MOV	R0,SIZE		;STORE # OF BYTES ACTUALLY XFERRED

```

1997 011056' 016701 170704      MOV      CNTADR,R1      ;GET ADR OF BYTE CNT TOTALS
1998 011052' 050011      ADD      RD,(R1)       ;ADD IN THIS CNT
1999 011054' 005541      RDC      -(R1)         ;UPDATE MOST SIGNF WORD OF CNT
2000 011066' 032767 000001 166706 20S: BIT      @IOERR,DFLGMD ;WAS THERE AN ERROR?
2001 011074' 001412      BEQ      PROCEX        ;Y N-PROCEX
2002 011076' 004567 000132      JSR      RS,ERRIS      ;GO ISSUE I/O TERMINATION
2003 011102' 003111      INTEAD: .WORD      IOTERM-ERMBAS ;ERROR MSG
2004 011104' 004767 000024      JSR      PC,RINTV     ;RESET THE INT VECTOR
2005 011110' 012602      MOV      (SP)+,R2     ;RESTORE R1 & R2
2006 011112' 012601      MOV      (SP)+,R1
2007 011114' 004577 166730      JSR      RS,@CUPGER   ;GO TO MPG ERR RETN POINT
2008 011120' 000207      RTS      PC           ;RETURN IN-LINE
2009 011122' 004767 000006      PROCEX: JSR      PC,RINTV ;GO RESET INT VECTOR
2010 011126' 012602      MOV      (SP)+,R2     ;RESTORE R1 & R2
2011 011130' 012601      MOV      (SP)+,R1
2012 011132' 000207      RTS      PC           ;EXIT IN-LINE
2013
2014
2015      ;RESET INTERRUPT VECTOR S/R
2016
2017      ;JSR      PC,RINTV      S/R CALL
2018      ;R3 MUST CONTAIN PROG TBL ADR
2019      ;DESTROYS RD
2020
2021 011134' 004567 000020      RINTV: JSR      RS,TVECT ;GO CK IF I HAVE VECTOR CONTROL
2022 011140' 000406      BR      RINTEX        ;BR IF I DON'T
2023 011142' 016767 166660 000004      MOV      IVCTAD,IOS   ;GET CURR INT VECT ADR
2024 011150' 004577 166714      JSR      RS,@CLAVEC   ;GO HAVE MPG CLEAR IT
2025 011154' 000000      IOS:   .WORD      XXXX
2026 011156' 000207      RINTEX: RTS      PC   ;EXIT IN-LINE
2027
2028      ;TEST INTERRUPT VECTOR S/R
2029
2030
2031      ;JSR      RS,TVECT      S/R CALL
2032      ;BR      LABEL        EXECUTED IF NOT SAME
2033      ;R3 MUST CONTAIN PROG TBL ADR
2034      ;DESTROYS RD
2035
2036 011160' 016767 166642 000010      TVECT: MOV      IVCTAD,20S ;GET CURR INT VECT ADR
2037 011166' 016346 000004      MOV      PFWADR(R3),-(SP) ;STORE FLGMD ADR TO IDENTIFY ME
2038 011172' 004577 166674      JSR      RS,@TSTVEC   ;DO I HAVE VECTOR CONTROL?
2039 011176' 000000      20S:   .WORD      XXXX ;MPG WILL TELL ME SINCE I CAN'T
2040 011200' 175530      .WORD      RHPINT-    ;GET AT LOWER MEM IF MEM MGMT
2041 011202' 000401      BR      TVECTX        ;BR IF I DONT'T HAVE CNTRL
2042 011204' 005725      TST      (RS)+        ;BYPASS BR INST IN S/R CALL
2043 011206' 000205      TVECTX: RTS      RS   ;EXIT IN-LINE

```

;ERROR INFORMATION DISPLAY S/R

```

2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055 011210' 004567 001130      ERRCS: JSR      R5,ERRCS      ;STORE CURR STATUS
2056 011214' 170434          .WORD      CSTAT-          ;STORE ADR OF CURR STATUS
2057 011216' 012767 170020 000404 ERRCS1: MOV      @CSTAT-ERSTAD,ERSTAD
2058 011224' 012767 170206 000212      MOV      @CSTAT-EBSBAS,EBSTAT
2059 011232' 000406          BR          ERRCOM      ;GO TO COMMON POINT
2060 011234' 012767 167760 000366 ERRIS: MOV      @ISTAT-ERSTAD,ERSTAD
2061 011242' 012767 170146 000174      MOV      @ISTAT-EBSBAS,EBSTAT
2062 011250' 012567 000134      ERRCOM: MOV     (R5)+,ERMBAS
2063 011254' 005267 170456          INC      ERRCNT      ;STORE MSG ADR
2064 011260' 012767 000001 166534      MOV      @ERRI          ;ADD 1 TO ERROR CNT
2065 011266' 032763 000400 000002      BIT      @OERCK,POPSW(R3) ;SET THE ERROR INDICATOR
2066 011274' 001004          BNE      25          ;SUPPOSED TO DO ERROR CHECKING?
2067 011276' 032763 020000 000002      BIT      @PRNER,POPSW(R3) ;Y,N-25
2068 011304' 001402          BEQ      45          ;ERROR PRINTING INHIBITED?
2069 011306' 000167 000404          JMP      ERREX      ;N,Y-45
2070 011312' 010446          45: MOV      R4,-(SP)    ;GO TO EXIT
2071 011314' 010546          MOV      R5,-(SP)    ;SAVE R4 & R5
2072 011316' 005004          CLR      R4          ;SET USER MODE PRINT FLAG
2073 011320' 004767 001114      JSR      PC,DEVID    ;DISPLAY DEVICE I.D.
2074 011324' 032767 000100 166450      BIT      @CNDISU,DFLGMD ;HAS THE CND BEEN ISSUED?
2075 011332' 001005          BNE      65          ;N,Y-65
2076 011334' 004567 001424      JSR      R5,PRINT    ;PRINT THE "BEFORE ISSUING I/O" MSG
2077 011340' 003004          .WORD    BEFIO-     ;
2078 011342' 000030          .WORD    24         ;
2079 011344' 000404          BR          85          ;GO CALC MSG LENGH
2080 011346' 004567 001412      65: JSR      R5,PRINT    ;PRINT THE "AFTER ISSUING I/O" MSG
2081 011352' 003022          .WORD    AFTIO-     ;
2082 011354' 000027          .WORD    23         ;
2083 011356' 010700          85: MOV      PC,R0      ;GET START ADR OF ERROR MSG
2084 011360' 062700 000030      ADD      @ERMBAS--,R0
2085 011364' 061000          ADD      (R0),R0
2086 011366' 012701 177777      MOV      @-1,R1      ;INITIALIZE MSG LENGTH
2087 011372' 005201          105: INC      R1        ;ADD 1 TO MSG LENGTH
2088 011374' 105720          TSTB    (R0)+        ;MSG TERMINATOR?
2089 011376' 001375          BNE      105        ;Y,N-105
2090 011400' 010167 000006      MOV      R1,ERMBAS+2 ;STORE MSG LENGTH
2091 011404' 004567 001354      JSR      R5,PRINT    ;PRINT ERROR MSG SPECIFIED
2092 011410' 000000          ERMBAS: .WORD    XXXX ;
2093 011412' 000000          .WORD    XXXX ;
2094 011414' 026727 177770 003562      CMP      ERMBAS,@INVDVN-ERMERS ;INVALID UNIT # MSG OR HIGHER?
2095 011422' 103105          BHIS    ERRSNM      ;N,Y-ERRSNM
2096 011424' 010701          MOV      PC,R1      ;GET ADR OF CODE AREA IN ERR MSG
2097 011426' 062701 002614      ADD      @CODFLD--,R1
2098 011432' 010700          MOV      PC,R0      ;SET UP ADR OF ERROR CODE TBL
2099 011434' 062700 000264      ADD      @ERCOTB--,R0
2100 011440' 010702          MOV      PC,R2      ;SET UP ADR OF STORED DEV REG'S

```

MAINDEC-11-DTR6A-A RK611 - RK06 DEVICE ROUTINE FOR MPG
DTR6AA.P11 SUBROUTINES FOR RK06 FUNCTION ROUTINES

2101	011442'	062702			EBSBAS: ADD	(PC)+,R2
2102	011444'	170206			EBSTAT: .WORD	CSTAT-EBSBAS
2103	011446'	012767	000015	000146	MOV	#13.,70\$
2104	011454'	012746	000100		MOV	#64.,-(SP)
2105	011460'	012205			15\$: MOV	(R2)+,R5
2106	011462'	000305			17\$: SWAB	R5
2107	011464'	112004			20\$: MOVB	(R0)+,R4
2108	011466'	005704			TST	R4
2109	011470'	001445			BEQ	60\$
2110	011472'	122704	000377		CMPB	#377,R4
2111	011476'	001770			BEQ	15\$
2112	011500'	122704	000376		CMPB	#376,R4
2113	011504'	001766			BEQ	17\$
2114	011506'	032704	000100		BIT	#100,R4
2115	011512'	001403			BEQ	30\$
2116	011514'	131005			BITB	(R0),R5
2117	011516'	001407			BEQ	40\$
2118	011520'	000402			BR	35\$
2119	011522'	131005			30\$: BITB	(R0),R5
2120	011524'	001004			BNE	40\$
2121	011526'	042704	177770		35\$: BIC	#177770,R4
2122	011532'	060400			ADD	R4,R0
2123	011534'	000753			BR	20\$
2124	011536'	042704	177770		40\$: BIC	#177770,R4
2125	011542'	020416			CMF	R4,(SP)
2126	011544'	101017			BHI	60\$
2127	011546'	060467	000050		ADD	R4,70\$
2128	011552'	005304			DEC	R4
2129	011554'	005200			INC	R0
2130	011556'	021627	000100		CMF	(SP),#64.
2131	011562'	001403			BEQ	50\$
2132	011564'	112721	000054		MOVB	#',(R1)+
2133	011570'	005316			DEC	(SP)
2134	011572'	112021			50\$: MOVB	(R0)+,(R1)+
2135	011574'	005316			DEC	(SP)
2136	011576'	005304			DEC	R4
2137	011600'	001374			BNE	50\$
2138	011602'	000730			BR	20\$
2139	011604'	005004			60\$: CLR	R4
2140	011606'	022627	000100		CMF	(SP)+,#64.
2141	011612'	001404			BEQ	80\$
2142	011614'	004567	001144		JSR	R5,PRINT
2143	011620'	002404			.WORD	DKMSG-
2144	011622'	000116			70\$: .WORD	78.
2145	011624'	004567	000700		80\$: JSR	R5,DISPST
2146	011630'	000000			ERSTAD: .WORD	XXXX
2147	011632'	004767	001056		JSR	PC,PTIWD
2148	011636'	016300	000022		ERRSNM: MOV	PSACST(R3),R0
2149	011642'	111001			110\$: MOVB	(R0),R1
2150	011644'	026067	000004	170112	CMF	4(R0),ERRADR
2151	011652'	001402			BEQ	120\$
2152	011654'	060100			ADD	R1,R0
2153	011656'	000771			BR	110\$
2154	011660'	005720			120\$: TST	(R0)+
2155	011662'	010701			MOV	PC,R1
2156	011664'	062701	002332		ADD	#STNUM-.,R1

```

:INITIALIZE MSG LENGTH
:INITIALIZE CODE FIELD CNT
:GET NEXT DEV REG WORD
:GET DESIRED BYTE IN LOW BYTE
:GET FLAG & LENGTH BYTE
:END OF THE CODE TBL?
N,Y-60$
:GO TO NXT DEV REG WORD?
N,Y-15$
:GO TO NXT BYTE IN DEV REG WORD?
N,Y-17$
:BIT VALUE OF 0 = AN ERROR CONDITION?
Y,N-30$
:THIS BIT RESET IN DEV REG BYTE?
N,Y-40$
:GO TO NXT TBL ENTRY
:THIS ERROR BIT SET IN DEV REG BYTE?
N,Y-40$
:ISOLATE ENTRY LENGTH
:POINT AT NXT CODE TBL ENTRY
:GO CK FOR NXT CODE
:ISOLATE I.D. NAME LENGTH + 1
:ENOUGH ROOM FOR NAME?
Y,N-60$
:ADJ MSG LENGTH FOR NAME
:ADJ FOR BIT MASK CHAR
:POINT PAST BIT MASK
:FIRST ERROR CODE IN MSG?
N,Y-50$
:MOVE COMMA TO MSG
:ADJ REMAINING ROOM IN MSG
:MOVE ERROR CODE TO MSG
:ADJ REMAINING ROOM IN MSG
:MOVED ALL NAME CHARS?
Y,N-50$
:GO CK FOR MORE ERROR BITS
:SET USER MODE PRINT
:ANY ERROR CODES PUT IN MSG?
Y,N-80$
:GO ISSUE ERROR BITS MSG

:DISPLAY DEVICE REG'S

:DISPLAY CYL,HEAD,SECT VALUES
:GET ADR OF SRC STMTS
:SAVE STMT LENGTH
:ERROR OCCUR ON THIS STMT?
N,Y-120$
:POINT AT NXT STMT
:GO CK NXT STMT
:SET UP ADR OF STMT & DATA
:SET UP DATA OUTPUT ADR

```

```

2157 011670' 004577 166166 JSR R5,JOECASC ;CONVERT IT TO ASCII
2158 011674' 012767 020040 002320 MOV #20040,STNUM+4 ;SET 2 LOW DIGITS TO SPACES
2159 011702' 004567 001056 JSR R5,PRINT ;ISSUE STMT # MSG
2160 011706' 002300 .WORD STNUM-
2161 011710' 177762 .WORD -14
2162 011712' 012605 MOV (SP)+,R5 ;RESTORE R5 & R4
2163 011714' 012604 MOV (SP)+,R4
2164 011716' 000205 ERREX: RTS ;EXIT IN-LINE
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182

```

;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 = RPO4 ONLY ERROR BIT
;BIT 4 = RPOS/RPO6 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.

```

```

2183 011720' 100005 042503 051122 ERCDTB: .ASCII <005><200>/CERR/ ;RPCS1 - BYTE 1
2184 011726' 020005 050123 051101 .ASCII <005><040>/SPAR/
2185 011734' 004004 052103 117 .ASCII <004><010>/CTO/
2186 011741' 377 .BYTE 377
2187 011742' 377 .BYTE 377
2188 011743' 377 .BYTE 377
2189 011744' 377 .BYTE 377
2190 011745' 004 042200 052114 .ASCII <004><200>/DLT/ ;RPCS2 - BYTE 1
2191 011752' 040004 041527 105 .ASCII <004><100>/MCE/
2192 011757' 004 052440 042520 .ASCII <004><040>/UPE/
2193 011764' 010004 042516 104 .ASCII <004><020>/MED/
2194 011771' 004 047010 046505 .ASCII <004><010>/NEM/
2195 011776' 002004 043520 105 .ASCII <004><004>/PGE/
2196 012003' 004 046402 051504 .ASCII <004><002>/MDS/
2197 012010' 000404 043125 105 .ASCII <004><001>/UFE/
2198 012015' 377 .BYTE 377
2199 012016' 004004 051127 114 .ASCII <004><010>/MRL/ ;RPDS - BYTE 1
2200 012023' 376 .BYTE 376
2201 012024' 100105 051104 054504 .ASCII <105><200>/DRDY/ ;RPDS - BYTE 0
2202 012032' 040103 053126 .ASCII <103><100>/VV/
2203 012036' 020005 051104 052117 .ASCII <005><040>/DROT/
2204 012044' 010004 051504 114 .ASCII <004><020>/DSL/
2205 012051' 005 040410 046103 .ASCII <005><010>/ACLO/
2206 012057' 377 .BYTE 377
2207 012060' 100004 041504 113 .ASCII <004><200>/DCK/ ;RPER1 - BYTE 1
2208 012065' 004 052500 051516 .ASCII <004><100>/UNS/
2209 012072' 020004 050117 111 .ASCII <004><040>/OPT/
2210 012077' 004 042020 042524 .ASCII <004><020>/DTE/
2211 012104' 004004 046127 105 .ASCII <004><010>/MLE/

```


MAINDEC-11-DTR6A-A RK611 - RK06 DEVICE ROUTINE FOR MPG
DTR6AA.P11 SUBROUTINES FOR RK06 FUNCTION ROUTINES

2212	012111'	005	044404	040504	.ASCII	<005><004>/IDAE/	
	012116'	105					
2213	012117'	004	041402	042517	.ASCII	<004><002>/COE/	
2214	012124'	000405	053110	041522	.ASCII	<005><001>/MVRC/	
2215	012132'	376			.BYTE	376	
2216	012133'	004	041200	042523	.ASCII	<004><200>/BSE/	;RPER1 - BYTE 0
2217	012140'	040004	041505	110	.ASCII	<004><100>/ECH/	
2218	012145'	005	042040	054524	.ASCII	<005><040>/DTYE/	
	012152'	105					
2219	012153'	005	043020	052115	.ASCII	<005><020>/FMTE/	
	012160'	105					
2220	012161'	006	042010	050122	.ASCII	<006><010>/DRPAR/	
	012166'	051101					
2221	012170'	002004	054116	106	.ASCII	<004><004>/NXF/	
2222	012175'	004	051402	044513	.ASCII	<004><002>/SKI/	
2223	012202'	000404	046111	106	.ASCII	<004><001>/ILF/	
2224	012207'	377			.BYTE	377	
2225	012210'	100005	052101	033501	.ASCII	<005><200>/ATA7/	;RPAS - BYTE 0
2226	012216'	040005	052101	033101	.ASCII	<005><100>/ATA6/	
2227	012224'	020005	052101	032501	.ASCII	<005><040>/ATA5/	
2228	012232'	010005	052101	032101	.ASCII	<005><020>/ATA4/	
2229	012240'	004005	052101	031501	.ASCII	<005><010>/ATA3/	
2230	012246'	002005	052101	031101	.ASCII	<005><004>/ATA2/	
2231	012254'	001005	052101	030501	.ASCII	<005><002>/ATA1/	
2232	012262'	000405	052101	030101	.ASCII	<005><001>/ATA0/	
2233	012270'	000			.BYTE	0	;TABLE TERMINATOR
2234	012272'				.EVEN		

```

2236 .SBTTL SUBROUTINES FOR RK06 DEVICE ROUTINE
2237
2238
2239
2240 ;SAVE REGISTERS R0 THRU R5
2241 ;JSR R0,SAVREG S/R CALL
2242
2243 SAVREG: MOV R1,-(SP) ;SAVE R0 THRU R5
2244 012272' 010146 MOV R2,-(SP)
2245 012274' 010246 MOV R3,-(SP)
2246 012276' 010346 MOV R4,-(SP)
2247 012300' 010446 MOV R5,-(SP)
2248 012302' 010546 MOV R0,PC ;EXIT IN-LINE
2249 012304' 010007
2250
2251 ;RESTORE REGISTERS R0 THRU R5
2252 ;JSR R0,RESREG S/R CALL
2253
2254 RESREG: TST (SP)+ ;RESTORE R5 THRU R0
2255 012306' 005726 MOV (SP)+,R5
2256 012310' 012605 MOV (SP)+,R4
2257 012312' 012604 MOV (SP)+,R3
2258 012314' 012603 MOV (SP)+,R2
2259 012316' 012602 MOV (SP)+,R1
2260 012320' 012601 RTS R0 ;EXIT IN-LINE
2261 012322' 000200
2262
2263 ;SET PROGRAM'S PROG TABLE ADR IN R3 & RPCS1 ADR IN R4
2264 ;JSR PC,SUPTAD S/R CALL
2265
2266 SUPTAD: MOV PC,R3 ;SET UP LOCATION ZERO ADR
2267 012324' 010703 ADD #LOCZ--,R3
2268 012326' 062703 165452 SUB -2(R3),R3 ;SUBTRACT PROG TBL LENGTH
2269 012332' 166303 177776 MOV DREGAD,R4 ;GET DEV REG BASE ADR
2270 012336' 016704 165462 RTS PC ;EXIT IN-LINE
2271 012342' 000207
2272
2273 ;STORE DEVICE'S STATUS REGISTERS
2274 ;JSR R5,STSTAT S/R CALL
2275 ;WORD STADR- REL STORAGE ADR
2276 ;DESTROYS R0,R1,R2
2277
2278 STSTAT: MOV R5,R1 ;GET REL STORAGE ADR & MAKE
2279 012344' 010501 ADD (R5)+,R1 ;IT ABSOLUTE
2280 012346' 062501 MOV DREGAD,R0 ;GET DEV REG ADR
2281 012350' 016700 165450 MOV R0,-(SP) ;SET UP ADR OF RPCS2 REG
2282 012354' 010046 ADD #RPCS2,(SP) ;FOR LATER USE
2283 012356' 062716 000010 MOV #STSLUP,R2 ;SETUP TWO LOOP COUNTS
2284 012362' 012702 002412 10S: MOV (R0)+,(R1)+ ;STORE DEV REG
2285 012366' 012021 DECB R2 ;FINISHED WITH THIS GROUP OF REGS?
2286 012370' 105302 BNE 10S ;Y,N-10S
2287 012372' 001375 SWAB R2 ;SET UP NEXT LOOP CNT
2288 012374' 000302
2289
2290
2291

```

2292	012376'	001417			BEG	30\$:DONE 2 PASSES? (N,Y-30\$)
2293	012400'	010746			MOV	PC, -(SP)		:SET UP CURRENT STATUS
2294	012402'	062716	167246		ADD	#CSTAT-, (SP)		:STORAGE ADR
2295	012406'	020126			CMP	R1, (SP)+		:STORING STATUS FOR INTERRUPT?
2296	012410'	101005			BHI	15\$:Y,N-15\$
2297	012412'	032736	000200		BIT	#OR, 2(SP)+		:IS OUTPUT READY?
2298	012416'	001403			BEG	20\$:Y,N-20\$
2299	012420'	012021			MOV	(R0)+, (R1)+		:STORE RPDB'S CONTENTS
2300	012422'	000761			BR	10\$:GO DO SECOND PASS
2301	012424'	005726		15\$:	TST	(SP)+		:TAKE UNUSED ADR OFF STACK
2302	012426'	062700	000002	20\$:	ADD	#2, R0		:BYPASS READ OF RPDB
2303	012432'	005021			CLR	(R1)+		:SET ITS STORAGE TO 0'S
2304	012434'	000754			BR	10\$:GO DO SECOND PASS
2305	012436'	000205		30\$:	RTS	RS		:EXIT IN-LINE
2306								
2307								:DISPLAY DEVICE I.D. & UNIT #
2308								
2309								:JSR PC, DEVID S/R CALL
2310								
2311								:R3 MUST CONTAIN PROG TBL ADR
2312								:DESTROYS R0, R1, R2
2313								
2314	012440'	012767	000026	000056	DEVID:	MOV #22, DEVIML		:INITIALIZE TO NORMAL MSG LENGH
2315	012446'	116300	000035		MOVB	PCURDV(R3), R0		:GET CURR UNIT #
2316	012452'	020027	000007		CMP	R0, #7		:VALID UNIT #?
2317	012456'	101007			BHI	DEVIIV		:Y,N-DEVIIV
2318	012460'	004577	165374		JSR	RS, #BTASLZ		:CONVERT # TO DECIMAL ASCII
2319	012464'	000674			.WORD	UNASCI-		
2320	012466'	016767	000672	000664	MOV	UNASCI+4, UNASCI		:MOVE ASCII # TO 1ST TWO DIGITS
2321	012474'	000410			BR	DEVIPR		:GO ISSUE MSG
2322	012476'	012767	000032	000020	DEVIIV:	MOV #26, DEVIML		:SET UP ERR COND MSG LENGH
2323	012504'	042700	177400		BIC	#177400, R0		:RESET HIGH BYTE
2324	012510'	004577	165342		JSR	RS, #BINASC		:CONVERT BINARY # TO ASCII
2325	012514'	000644			.WORD	UNASCI-		
2326	012516'	004567	000242		DEVIPR:	JSR RS, PRINT		:GO ISSUE UNIT # MSG
2327	012522'	000612			.WORD	UNITMG-		
2328	012524'	000026			DEVIML:	.WORD 22.		
2329	012526'	000207			RTS	PC		:EXIT IN-LINE

```

2331                                     ;TAILOR STATUS MSG & PRINT IT
2332
2333                                     ;JSR   R5,DISPST           S/R CALL
2334                                     ;WORD  STATADR-         REL ADR OF STATUS DATA
2335                                     ;DESTROYS R0,R1,R2
2336
2337 012530' 010346          DISPST: MOV   R3,-(SP)           ;SAVE R3
2338 012532' 010503          MOV   R5,R3           ;GET REL DATA ADR
2339 012534' 062503          ADD   (R5)+,R3       ;MAKE IT ABS
2340 012536' 010546          MOV   R5,-(SP)       ;SAVE R5
2341 012540' 010705          MOV   PC,R5           ;SET UP ADR OF REG NAMES IN ASCII
2342 012542' 062705          ADD   #DVRGMS- ,R5   ;
2343 012544' 012746          MOV   #REGNUM,-(SP) ;SET UP # OF REGISTERS TO DISPLAY
2344 012552' 012700          10$: MOV   #3,R0         ;SET UP 3 REG LOOP CNT
2345 012556' 010701          MOV   PC,R1         ;POINT AT REG NAME IN MSG
2346 012560' 062701          ADD   #DVRGMS- ,R1   ;
2347 012564' 012521          15$: MOV   (R5)+,(R1)+ ;MOVE REG NAME TO MSG
2348 012566' 012521          MOV   (R5)+,(R1)+ ;
2349 012570' 005725          TST   (R5)+         ;POINT TO NEXT NAME
2350 012572' 062701          ADD   #10.,R1       ;POINT TO NEXT FIELD IN MSG
2351 012576' 005300          DEC   R0             ;DONE 3 REGS?
2352 012600' 001371          BNE   15$           ;Y N-15$
2353 012602' 012300          MOV   (R3)+,R0      ;CONVERT OCTAL REGISTER CONTENTS
2354 012604' 004577          JSR   R5,ABINASC    ;FOR 3 REGISTERS TO ASCII
2355 012610' 000564          .WORD DVRDT1-      ;AND PLACE IN THE MSG
2356 012612' 012300          MOV   (R3)+,R0      ;
2357 012614' 004577          JSR   R5,ABINASC    ;
2358 012620' 000572          .WORD DVRDT2-      ;
2359 012622' 012300          MOV   (R3)+,R0      ;
2360 012624' 004577          JSR   R5,ABINASC    ;
2361 012630' 000600          .WORD DVRDT3-      ;
2362 012632' 012767          000050 000034      MOV   #40.,30$     ;INITIALIZE MSG LENGTH TO 3 REGS
2363 012640' 162716          000003          SUB   #3,(SP)       ;DECR REGISTER CNT
2364 012644' 100005          BPL   25$           ;< 3 REGS? (Y,N-25$)
2365 012646' 162767          000016 000020 20$: SUB   #14.,30$     ;SHORTEN MSG LENGTH BY 1 REG
2366 012654' 005216          INC   (SP)         ;INCR NEG REG CNT
2367 012656' 100773          BMI   20$           ;CNT BACK TO 0? (Y,N-20$)
2368 012660' 010346          25$: MOV   R3,-(SP)   ;SAVE REG DATA PNTR
2369 012662' 016603          000006          MOV   6(SP),R3     ;RESTORE PROG TBL ADR
2370 012666' 004567          000072          JSR   R5,PRINT     ;GO PRINT THE MSG
2371 012672' 000474          .WORD DVRGMS-      ;
2372 012674' 000050          30$: .WORD 40.      ;
2373 012676' 012603          MOV   (SP)+,R3     ;RESTORE REG DATA PNTR
2374 012700' 005716          TST   (SP)         ;MORE REGS TO GO?
2375 012702' 001323          BNE   10$           ;N,Y-10$
2376 012704' 005726          TST   (SP)+       ;REMOVE CNT FROM STACK
2377 012706' 012605          MOV   (SP)+,R5     ;RESTORE R5 & R3
2378 012710' 012603          MOV   (SP)+,R3     ;
2379 012712' 000205          RTS   R5           ;EXIT IN-LINE

```

```

2381                                     ;DISPLAY CYL/HEAD/SECT WORDS' VALUES
2382
2383                                     ;JSR   PC,PRTIWD      S/R CALL
2384                                     ;DESTROYS R0,R1,R2
2385
2386 012714' 016700 165064 PRTIWD: MOV   CYL,R0      ;GET CYL VALUE
2387 012720' 004577 165132 JSR   R5,@BINASC ;CONVERT ITS VALUE TO ASCII
2388 012724' 001221 .WORD IFCYL-
2389 012726' 016700 165054 MOV   HEAD,R0    ;GET & CONVERT HEAD VALUE
2390 012732' 004577 165120 JSR   R5,@BINASC
2391 012736' 001224 .WORD IFHEAD-
2392 012740' 016700 165044 MOV   SECT,R0   ;GET & CONVERT SECT VALUE
2393 012744' 004577 165106 JSR   R5,@BINASC
2394 012750' 001227 .WORD IFSECT-
2395 012752' 004567 000006 JSR   R5,PRINT  ;PRINT MSG WITH THEIR VALUES
2396 012756' 001162 .WORD INFOMG-
2397 012760' 000045 .WORD 37
2398 012762' 000207 RTS          ;EXIT IN-LINE
2399
2400
2401                                     ;ISSUE MSG TO LIST DEVICE SUBROUTINE
2402
2403                                     ;JSR   R5,PRINT      S/R CALL
2404                                     ;.WORD MSGADR-      REL ADR OF MSG
2405                                     ;.WORD BYTCNT       MSG BYTE CNT (IF NEGATIVE,
2406                                     ;                  RESET PRT DEV DEDICATED.)
2407
2408                                     ;R3 = PROG TBL ADR
2409                                     ;R4 = FLAGWORD -- IF NEGATIVE, USE CMND MODE PRINT
2410                                     ;DESTROYS R0,R1,R2
2411 012764' 010500 PRINT: MOV   R5,R0      ;GET MSG ADR & MAKE IT ABS
2412 012766' 062500 ADD   (R5)+,R0
2413 012770' 012501 MOV   (R5)+,R1    ;GET BYTE COUNT
2414 012772' 005704 TST   R4          ;USE CMND MODE PRINT?
2415 012774' 100030 BPL   40$        ;Y,N-40$
2416 012776' 010702 MOV   PC,R2      ;SET UP LINK INFO ADR
2417 013000' 062702 000040 ADD   #20$--,R2
2418 013004' 160200 SUB   R2,R0      ;MAKE MSG ADR REL
2419 013006' 010022 MOV   R0,(R2)+  ;STORE MSG ADR
2420 013010' 010112 MOV   R1,(R2)   ;STORE MSG'S BYTE COUNT
2421 013012' 100001 BPL   10$        ;CNT NEG? (Y,N-10$)
2422 013014' 005412 NEG   (R2)       ;MAKE IT POSITIVE
2423 013016' 016367 000006 000240 10$: MOV   PASCIN(R3),PROGM ;STORE PROG'S # IN MSG
2424 013024' 004577 165024 JSR   R5,@CLIST  ;ISSUE PROG #
2425 013030' 000232 .WORD PNMSG-
2426 013032' 000005 .WORD 5
2427 013034' 004577 165014 JSR   R5,@CLIST  ;ISSUE MSG SPECIFIED
2428 013040' 000000 20$: .WORD XXXX
2429 013042' 000000 .WORD XXXX
2430 013044' 004577 165004 JSR   R5,@CLIST  ;ISSUE A <CR> & <LF>
2431 013050' 000440 .WORD CRLF-
2432 013052' 000002 .WORD 2
2433 013054' 000410 BR    PRTEX     ;GO TO EXIT
2434 013056' 010067 000010 40$: MOV   R0,50$   ;STORE MSG'S ABS ADR
2435 013062' 010167 000006 MOV   R1,60$   ;STORE ITS BYTE CNT
2436 013066' 004577 164760 JSR   R5,@JLIST ;GO TO MPG TO ISSUE THE MSG

```

```

437 013072' 000000
438 013074' 000000
439 013076' 000205
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
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492

```

```

508: .WORD XXXX
608: .WORD XXXX
PRTEX: RTS RS ;EXIT IN-LINE

:SUBROUTINE TO PRINT ON THE LIST DEVICE
:CALL AS FOLLOWS
JSR RS,PRINTX
.WORD MSGADR-PRIX1
.WORD BYTE COUNT
RETURN
:THIS SUBROUTINE DOES NOT DESTROY R3 & R4

PRINTX: MOV R4,-(SP)
MOV R3,-(SP)
JSR PC,SUPTAD ;SETUP R3
MOV R0,R4 ;AND R4 FOR USER MODE
MOV (R5)+,PRIX1 ;RELOCATE ADDRESSES
MOV (R5)+,PRIX2
JSR RS,PRINT ;GO TO MPG PRINT

PRIX1: .WORD XXXX
PRIX2: .WORD XXXX
MOV (SP)+,R3
MOV (SP)+,R4
RTS RS

:SUBROUTINE TO SELECT A DRIVE
:CALL AS FOLLOWS
PUT MESSAGE SELECT CODE IN RS (R4 MUST = BASE ADDR)
JSR PC,SD
RETURN

SD: BIS RS,RPMR1(R4) ;SETUP MESSAGE SELECT CODE
MOV R5,SDCODE,RPCS1(R4) ;SELECT DRIVE
JSR PC,DRVDLY ;WAIT FOR DRIVE
RTS PC

:SUBROUTINE TO WAIT FOR DRIVE DELAY TIME
:CALL WITH JSR PC,DRVDLY

DRVDLY: MOV RS,-(SP)
MOV R5,R5
DRV1: BIT R5,R5 ;IS CONTROLLER READY
BNE DRV2 ;YES, EXIT
DEC RS
BNE DRV1
DRV2: MOV (SP)+,RS
RTS PC ;DEFAULT IF CONTRL NOT RDY

:SUBROUTINE TO ISSUE CONTROLLER CLEAR
:SUBROUTINE SAVES AND RESTORES RKCS2, RKBA, AND RKDA
:R4 MUST HAVE BASE ADDRESS OF DISK ADDRESSES
:CALL WITH JSR PC,CTRLCL

CTRLCL: MOV RPCS2(R4),-(SP) ;SAVE DISK PARAMATERS
MOV RPBA(R4),-(SP)
MOV RPDA(R4),-(SP)

```

```

177214
000000
000010
000006
177634

```

```

000000

```

```

000140
000200

```

```

;SAVE DISK PARAMATERS

```

MAINDEC-11-DTR6A-A RK611 - RK06 DEVICE ROUTINE FOR MPG
DTR6AA.P11 SUBROUTINES FOR RK06 DEVICE ROUTINE

013222	052714	100000
013226	004767	177730
013230	012664	000006
013236	012664	000004
013242	012664	000010
013246	000207	
013250	012714	000005
013254	004767	177702
013260	000207	

```

      BIS      @CCLR,(R4)          ;CLEAR CONTROLLER
      JSR      PC,DRVCLY
      MOV      (SP)+,RPOA(R4)     ;RESTORE DISK PARAMATERS
      MOV      (SP)+,RPBA(R4)
      MOV      (SP)+,RPCS2(R4)
      RTS      PC

:SUBROUTINE TO ISSUE DRIVE CLEAR
:CALL WITH JSR PC,DRVCLR
:R4 MUST CONTAIN THE BASE ADDRESS OF THE DISK REGISTERS
DRVCLR: MOV      @DCCODE,(R4)     ;CLEAR DRIVE
      JSR      PC,DRVCLY
      RTS      PC

```

MAINDEC-11-DTR6A-A RK611 - RK06 DEVICE ROUTINE FOR MPG
DTR6AA.P11 RK06 MESSAGE STORAGE AREA

.SBTTL RK06 MESSAGE STORAGE AREA
: MESSAGE STORAGE AREA
:

.NLIST BEX

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
00

```

013262' 021520          PNMSG: .ASCII  /P8/
013264' 054130          PROGM: .ASCII  /XX/<011>
013267'      101 020124  040514  ATMSG: .ASCII  /AT LAST INT:/
013303'      103 051125  042522  CURMSG: .ASCII  /CURRENTLY:/
013315'      105 042116  047440  RENDMG: .ASCII  /END OF REPORT/<15><12>
                                .EVEN
013334' 025052 025052 045522  UNITNG: .ASCII  /###RK06 DISK UNIT: /
013360' 054130 054130 054130  UNASCI: .ASCII  /X0000X/
                                .EVEN
013376' 054130 054130 020075  DVRGNG: .ASCII  /X00X= /
013377' 054130 054130 054130  DVRT1: .ASCII  /X0000X X00X= /
013378' 054130 054130 054130  DVRT2: .ASCII  /X0000X X00X= /
013379' 054130 054130 054130  DVRT3: .ASCII  /X0000X/
013380' 054130 054130 035123  CNTSMG: .ASCII  /BYTES: RD= /
013381' 054130 054130 054130  BCHRD: .ASCII  /X00000000000X MR= /
013382' 054130 054130 054130  BCHNR: .ASCII  /X00000000000X/
013383' 054130 054130 054130  CALF: .ASCII  <015><012>
                                .EVEN
013384' 054130 054130 020075  BCNCK: .ASCII  <011><011>/CK= /
013385' 054130 054130 054130  CNDCRD: .ASCII  /X00000000000X/<015><012><011>/CMDS: RD= /
013386' 054130 054130 054130  CNDCHR: .ASCII  /X0000X MR= /
013387' 054130 054130 054130  CNDCK: .ASCII  /X0000X CK= /
013388' 054130 054130 054130  CNDCKX: .ASCII  /X0000X/<015><012><011><011>/SK= /
013389' 054130 054130 054130  CNDCKY: .ASCII  /X0000X MISC= /
013390' 054130 054130 054130  CNDCHS: .ASCII  /X0000X/<015><012><011>/ERRORS: DEV= /
013391' 054130 054130 054130  CNTERR: .ASCII  /X0000X CORR ECC= /
013392' 054130 054130 054130  CNTCEC: .ASCII  X0000X DATA/OPR= X
013393' 054130 054130 054130  CNTDUR: .ASCII  /X0000X/<015><012><011>/RETRYS: DLT= /
013394' 054130 054130 054130  CNTDLT: .ASCII  /X0000X DTE= /
013395' 054130 054130 054130  CNTDTE: .ASCII  /X0000X HVRC= /
014003' 054130 054130 054130  CNTHCR: .ASCII  /X0000X/<015><012><011><011>/FER= /
014004' 054130 054130 054130  CNTFER: .ASCII  /X0000X DCK= /
014005' 054130 054130 054130  CNTDCK: .ASCII  /X0000X/<015><012><011><011>/MCE= /
014006' 054130 054130 054130  CNTMCE: .ASCII  /X0000X/<015><012><011>/TOTAL RETRYS: /
014104' 054130 054130 054130  CNTRTY: .ASCII  /X0000X/<015><012><011>/INTERRUPTS: /
014110' 054130 054130 054130  CNTINT: .ASCII  /X0000X/
                                .EVEN
014140' 054503 036514  040     INFONG: .ASCII  /CYL= /
014145' 054130 054130 054130  IFCYL: .ASCII  /X0000X HEAD= /
014151' 054130 054130 054130  IFHEAD: .ASCII  /X0000X SECT= /
014177' 054130 054130 054130  IFSECT: .ASCII  /X0000X/
                                .EVEN
014206' 054130 047115 020124  STWING: .ASCII  /STANT # /
014216' 054130 054130 054130  STWNUM: .ASCII  /X0000X/
014224' 051109 047522 020122  DKEMSG: .ASCII  /ERROR BITS:/<015><012><011>
014242' 050109 047506 042522  COOFLO: .BLKB  66.
014344' 042502 047506 042522  BEFIO: .ASCII  'BEFORE ISSUING I/O CMND:'
014374' 043101 042524 020122  AFTIO: .ASCII  'AFTER ISSUING I/O CMND:'

```


014423'	122	047457	047440	CRT0:	.ASCIZ	'T/O ON CRESET'
014441'	122	047457	047440	SRT0:	.ASCIZ	'T/O ON SRESET'
014457'	122	047457	047440	RT0:	.ASCIZ	'T/O ON REL'
014512'	047457	047457	047440	IOTO:	.ASCIZ	'TIMEOUT V ₁ :/0'
014517'	116	047117	047455	NOITER:	.ASCII	'NON-INT V ₁ :/0'
014521'	111	047457	0512040	IOTERM:	.ASCIZ	'I/O TERMINATION ERROR'
014527'	111	047116	047440	NOATA:	.ASCIZ	'INT WITHOUT ATA'
014527'	111	047116	047440	NODI:	.ASCIZ	'INT WITHOUT DI'
014527'	047116	047116	047120	LDPATA:	.ASCIZ	'UNEXP ATA COND'
014527'	111	047116	047440	RTYEXH:	.ASCIZ	'EXHAUSTED RETRIES'
014527'	111	047116	047440	INVPAT:	.ASCIZ	'INV ECC BIT PATTERN'
014527'	111	047116	047440	INVPOS:	.ASCIZ	'INV ECC BIT POSITION'
014720'	047116	047116	047516	CLT:	.ASCIZ	'CAN NOT READ LAST TRACK'
014720'	047116	047116	047516	UNKERR:	.ASCIZ	'UNKNOWN ERROR CONDITION'
015000'	053123	046101	041040	NSVAL:	.ASCIZ	'SVAL BIT NOT SET IN RKDS'
015031'	123	047523	051107	PABSA:	.ASCIZ	'PROGRAM ABORT: HAVE DISK ADDRESS WHICH WOULD CHANGE LAST TRACK'
015131'	117	052123	052320	ONRPH:	.ASCIZ	'OUTPUT NOT READY ON RDMD COMMAND'

MESSAGES WHICH FOLLOW WILL NOT HAVE DEVICE REGISTERS PRINTED

015172'	047111	020126	047125	INVDVN:	.ASCIZ	'/INV UNIT &/'
015205'	123	046105	051104	ISDMC:	.ASCIZ	'SELDRI COMMAND HAS INVALID CODE'
015235'	122	044504	020117	RIOTS:	.ASCII	'RDIO IS TOO SMALL FOR "BADSEC" COMMAND' <15><12>
015315'	011	030465	024062		.ASCIZ	<11>'512(DECIMAL) BYTES ARE NEEDED'
015354'	040502	020104	042523	BSH:	.ASCII	'/BAD SECTORS (OCTAL DATA)/'
	015404			BSHX=.		
015404'	040520	045503	021440	PNUM:	.ASCII	'/PACK &/'
015412'	000006			PNUM1:	.BLKB	6
015420'	000006			PNUM2:	.BLKB	6
015426'	015	012			.BYTE	15,i2
015430'	015430			PNLPM:		
015430'	020040	041440	046131	BSHEAD:	.ASCII	' / CYL, HEAD, SECT/'
	015454			BSHDX=.		
015454'	047516	042516	005015	NONE:	.ASCII	'/NONE/<15><12><12>
	015463			NONEX=.		
015463'	124	044510	020123	TIAP:	.ASCIZ	'/THIS IS AN ALIGNMENT PACK/<15><12>
015517'	105	042116	047440	E0BS:	.ASCII	'/END OF BAD SECTORS/<15><12><12>
	015544'			E0BSX=.		
				.EVEN		
				.LIST	BEX	
015544'				DVREND=.		

2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652

.SBTTL FORMATS FOR PROGRAM & DEVICE ROUTINE TABLES

; PROGRAM TABLE FORMAT

000242

PTLGTH= 162. ;PROGRAM TABLE LENGTH - NON MEM MGMT VERSION OF MPG

; (PTLGTH= 212. ;PROGRAM TABLE LENGTH - MEM MGMT VERSION OF MPG)

000000

PFLGWD= +0. ;PROGRAM FLAG WORD - 1 WORD

000002

URSTOP= 2 : 1 = USER HAS STOPPED THIS PROGRAM
ERSTOP= 4 : 1 = AN ERROR HAS STOPPED THIS PROGRAM
MT4IOT= 10 : 1 = WAITING FOR I/O TERMINATION
CTPRIO= 20 : 1 = CONSOLE OR PRINTER I/O IN PROGRESS
SETDED= 40 : 1 = THIS PROG SET THE PRT DEV DEDICATED FLAG
OCPRES= 100 : 1 = OBJ CODE IS PRESENT
USELBM= 200 : 1 = THIS PROG USES THE UNIBUS MAP (MEM MGMT ONLY)
ACTIVE= 100000 : 1 = PROGRAM IS ACTIVE (SPECIFIED FOR EXECUTION)

000002

POPSM= +2. ;PROGRAM'S OPERATION SWITCHES - 1 WORD

100000

STONER= 100000 : 1 = STOP PROG EXECUTION UPON ERROR
CYCPRG= 40000 : 1 = CYCLE PROGRAM (ON CURRENT DEVICE)
PRONER= 20000 : 1 = DO NOT PRINT ON ERROR
BIT12= 10000 : 0 = NOT USED
BIT11= 4000 : 0 = NOT USED
CYCOVL= 2000 : 1 = CYCLE THE DEVICE LIST
GTNXTD= 1000 : 1 = CYCLE ON SAME DEVICE UPON ERROR
DOERCK= 400 : 1 = DON'T DO ERROR CHECKING
SOPER= 200 : 1 = DEVICE SPECIAL OPERATION
BIT6= 100 : 0 = NOT USED
DOIOT= 40 : 1 = DO NOT PERFORM I/O TIMEOUT
AUTORP= 20 : 1 = DO NOT AUTOMATICALLY DISPLAY COUNTS
AURPEP= 10 : 1 = AUTO DISPLAY COUNTS AT END OF FINAL PASS ONLY
HSKPEP= 4 : 1 = HOUSEKEEP COUNTS ONLY AT RUN COMMAND
PFBBOV= 2 : 1 = PRINT FIRST BAD BYTE ONLY ON VERIFY
NOCMP= 1 : 1 = DO NOT PRINT PROG COMPLETED MSG

000004

PFWADR= +4. ;PROGRAM FLAGWORD ADDRESS - 1 WORD

000006

PASCIN= +6. ;PROGRAM'S NUMBER IN ASCII - 1 WORD

000010

PNAME= +8. ;PROGRAM'S NAME IN ASCII - 6 BYTES

000016

PRDIOA= +14. ;ADDRESS OF READ I/O AREA - 1 WORD

000020

PWRIOA= +16. ;ADDRESS OF WRITE I/O AREA - 1 WORD

000022

PSRCST= +18. ;SOURCE STATEMENTS START ADDRESS - 1 WORD

000024

POBJST= +20. ;OBJECT CODE START ADDRESS - 1 WORD

000026

PLNGTH= +22. ;PROG AREA LENGTH (OBJ END MINUS PROG TBL START) - 1 WORD

000030

PTOCNT= +24. ;I/O TIMEOUT COUNT - 1 WORD

2663	000032	PMDLCD= +26.	;DEV ROUT MODEL # CODE - 1 WORD
2664	000034	PDPNTR= +28.	;CURRENT DEVICE NUMBER POINTER - 1 BYTE
2665	000035	PCURDV= +29.	;CURRENT DEVICE # - 1 BYTE
2666	000036	PDNMS= +30.	;DEVICE NUMBERS - 16 BYTES
2667	000056	PTEM0= +46.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2668	000060	PTEM1= +48.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2669	000062	PTEM2= +50.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2670	000064	PTEM3= +52.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2671	000066	PTEM4= +54.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2672	000070	PTEM5= +56.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2673	000072	PTEM6= +58.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2674	000074	PTEM7= +60.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2675	000076	PTEM8= +62.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2676	000078	PTEM9= +64.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2677	000100	PTEM10= +66.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2678	000102	PTEM11= +68.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2679	000104	PTEM12= +70.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2680	000106	PTEM13= +72.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2681	000110	PTEM14= +74.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2682	000112	PTEM15= +76.	;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2683	000114	Pnbr= +78.	;NUMBER OF BYTES TO TRANSFER ON MOVE (NBR) - 1 WORD
2684	000116	PSRC= +80.	;DATA SOURCE ADDRESS ON MOVE (SRC) - 1 WORD
2685	000120	PDST= +82.	;DATA DESTINATION ADDRESS ON MOVE (DST) - 1 WORD
2686	000122	PSTKCT= +84.	;# OF WORDS (X 2) SAVED OFF STACK - 1 WORD
2687	000124	PSTKSV= +86.	;STACK WORDS STORAGE AREA - 30 WORDS
2688	000126	PSVREG= +146.	;USER'S R0 THRU R5 REGISTERS STORAGE AREA - 6 WORDS
2689	000222	PUSRPC= +158.	;USER'S CURRENT PROGRAM COUNTER - 1 WORD
2690	000236		
2700			
2701			
2702			
2703			
2704			
2705			
2706			
2707			
2708			
2709			
2710			
2711			
2712			
2713			
2714			
2715			
2716			

```

2718                                     ;FOLLOWING ENTRIES (PRDIOX THRU PUBMAP) ARE ONLY IN MEM MGMT VERSION
2719
2720      000240      PRDIOX= +160. ;18/22 BIT ABSOLUTE ADDRESS OF READ I/O AREA - 2 WORDS)
2721
2722      000244      PRDIOV= +164. ;18 BIT VIRTUAL ADDRESS OF READ I/O AREA - 2 WORDS)
2723
2724      000250      PWRIOX= +168. ;18/22 BIT ABSOLUTE ADDRESS OF WRITE I/O AREA - 2 WORDS)
2725
2726      000254      PWRIOV= +172. ;18 BIT VIRTUAL ADDRESS OF WRITE I/O AREA - 2 WORDS)
2727
2728      ;(PUPARS= +176. ;STORAGE AREA FOR USER'S PAR'S 0 THRU 7 - 8 WORDS)
2729
2730      ;(PUPDRS= +192. ;STORAGE AREA FOR USER'S PDR'S 0 THRU 7 - 8 WORDS)
2731
2732      ;(PUBMAP= +208. ;1ST UNIBUS MAP REG # AND # OF REGS USED - 1 WORD)
2733
2734      ;END OF MEM MGMT ONLY ENTRIES
2735
2736      000240      PTSIZE= +160. ;PROGRAM TABLE SIZE IN BYTES - 1 WORD - NON MEM MGMT
2737
2738      ;(PTSIZE= +210. ;PROGRAM TABLE SIZE IN BYTES - 1 WORD - MEM MGMT VERSION)
2739
2740      000242      PTEND= +162. ;END OF PROGRAM TABLE - NON MEM MGMT VERSION
2741
2742      ;(PTEND= +212. ;END OF PROGRAM TABLE - MEM MGMT VERSION)

```

```

2744           ;      DEVICE ROUTINE TABLE
2745
2746           000116      DRTLTH= 78.      ;DEVICE ROUTINE TABLE LENGTH
2747           ;
2748           ;
2749           ;
2750           000000      DEVRSZ= +0.      ;DEVICE ROUTINE SIZE IN BYTES - 1 WORD
2751           ;
2752           000002      DEVFWD= +2.      ;DEVICE ROUTINE FLAGWORD - 1 WORD
2753           ;
2754           000004      DEVIW1= +4.      ;DEVICE INTERFACE WORD # 1 - 1 WORD
2755           ;
2756           000006      DEVIW2= +6.      ;DEVICE INTERFACE WORD # 2 - 1 WORD
2757           ;
2758           000010      DEVIW3= +8.      ;DEVICE INTERFACE WORD # 3 - 1 WORD
2759           ;
2760           000012      DEVIW4= +10.     ;DEVICE INTERFACE WORD # 4 - 1 WORD
2761           ;
2762           000014      DEVIW5= +12.     ;DEVICE INTERFACE WORD # 5 - 1 WORD
2763           ;
2764           000016      DEVIW6= +14.     ;DEVICE INTERFACE WORD # 6 - 1 WORD
2765           ;
2766           000020      DEVIW7= +16.     ;DEVICE INTERFACE WORD # 7 - 1 WORD (SIZE)
2767           ;
2768           000022      DEVIW8= +18.     ;DEVICE INTERFACE WORD # 8 - 1 WORD (ERR)
2769           ;
2770           000024      DEVDR= +20.     ;DEVICE REGISTERS ADDRESS - 1 WORD
2771           ;
2772           000026      DEVIV= +22.     ;DEVICE INTERRUPT VECTOR ADDRESS - 1 WORD
2773           ;
2774           000030      DEVRPS= +24.     ;DEVICE READ PROCESSOR STATUS WORD (BUS REQ) - 1 WORD
2775           ;
2776           000032      DEVWPS= +26.     ;DEVICE WRITE PROC STATUS WORD (BUS REQ) - 1 WORD
2777           ;
2778           000034      DHKPAD= +28.     ;DEVICE ROUT HOUSEKEEPING ROUT REL ENTRY ADR - 1 WORD
2779           ;
2780           000036      DERPAD= +30.     ;DEVICE ROUT REPORT ROUT REL ENTRY ADR - 1 WORD
2781           ;
2782           000040      DKILAD= +32.     ;DEVICE ROUT KILL ROUTINE REL ENTRY ADR - 1 WORD
2783           ;
2784           000042      DECTAD= +34.     ;DEVICE ROUT ERROR COUNTER REL ADR - 1 WORD
2785           ;
2786           000044      DTOEAD= +36.     ;DEVICE ROUT TIMEOUT ERR ROUT REL ENTRY ADR - 1 WORD
2787           ;
2788           000046      DEVI0B= +38.     ;DEVICE I/O BUSY BRANCH ADDRESS (CIOBSY) - 1 WORD
2789           ;
2790           000050      DEVDER= +40.     ;DEVICE ERROR BRANCH ADDRESS (CUPGER) - 1 WORD
2791           ;
2792           000052      DVUPRT= +42.     ;USER MODE PRINT BRANCH ADDRESS (ULIST) - 1 WORD
2793           ;
2794           000054      DVCPR= +44.     ;CMND MODE PRINT BRANCH ADDRESS (CLIST) - 1 WORD
2795           ;
2796           000056      DEVBTA= +46.     ;CONVERT BINARY TO ASCII BR ADR (BINASC) - 1 WORD
2797           ;
2798           000060      DVBTDA= +48.     ;CONVERT BINARY TO DECIMAL ASCII BR ADR (BTASLZ) - 1 WORD
2799

```

2800	000062	DVPDTA= +50.	; CONVERT PACKED DECIMAL TO ASCII BR ADR (DECASC) - 1 WORD
2801			
2802	000064	DVSFWD= +52.	; MPG SYSTEM FLAGWORD ADDRESS (CSYSFW) - 1 WORD
2803			
2804	000066	DVSVEC= +54.	; SET INTERRUPT VECTOR BR ADR (SETVEC) - 1 WORD
2805			
2806	000070	DVCVEC= +56.	; CLEAR INTERRUPT VECTOR BR ADR (CLRVEC) - 1 WORD
2807			
2808	000072	DVTVEC= +58.	; TEST INTERRUPT VECTOR BR ADR (TSTVEC) - 1 WORD
2809			
2810	000074	DVRINT= +60.	; RETURN FROM INTERRUPT BR ADR (RTNINT) - 1 WORD
2811			
2812	000076	DVGETB= +62.	; GET DATA BYTE BR ADR (GETBYT) - 1 WORD
2813			
2814	000100	DVPUTB= +64.	; PUT DATA BYTE BR ADR (PUTBYT) - 1 WORD
2815			
2816	000102	DEVSTP= +66.	; DEVICE ROUT REL SYMBOL TABLE POINTER - 1 WORD
2817			
2818	000104	DEVETP= +68.	; DEVICE ROUT REL ENTRY TABLE POINTER - 1 WORD
2819			
2820	000106	DVPTEP= +70.	; PACK TABLE EXTEN. REL POINTER - 1 WORD
2821			
2822	000110	DVVTEP= +72.	; VECTOR TABLE EXTEN. REL POINTER - 1 WORD
2823			
2824	000112	DVCTEP= +74.	; COMPILER TBL EXTEN. REL POINTER - 1 WORD
2825			
2826	000114	DVIWSP= +76.	; DEVICE INTERFACE WORD SYMBOL TBL REL POINTER - 1 WORD
2827			
2828	000116	DRTEND= +78.	; END OF DEVICE ROUTINE TABLE
2829			
2830			
2831	000001	.END	

M05

MAINDEC-11-DTR6A-A RK611 - RK06 DEVICE ROUTINE FOR MPG
 DTR6AA.P11 SYMBOL TABLE

MACY11 27(732) 24-SEP-76 14:11 PAGE 15-2

SEQ 0064

TIAAP	015463R	002	UNIMAP=	000040		UP3	003354R	002	WAITMD=	100000		WRCOM	004452R	002
TOUTER	002522R	002	UNITMG	013334R	002	UP4	003314R	002	WCCOE=	000031		WRHD	004714R	002
TSTVEC	000072R	002	UNKERR	014750R	002	URSTOP=	000002		WCE =	040000		WRITE	004442R	002
TVECT	011160R	002	UNLOAD	003710R	002	USEUBM=	000200		WCECNT	001756R	002	WRL =	004000	
TVECTX	011206R	002	UNS =	040000		USMTPS=	000002		WCODE =	000023		WR1	004562R	002
UCODE =	000007		UPE =	020000		UXPATA	014606R	002	WHCODE=	000027		WR2	004650R	002
UFE =	000400		UPSUB	003310R	002	VV =	000100		WLE =	004000		WR3	004534R	002
ULIST	000052R	002	UP1	003326R	002	VVFLG =	001000		WRCK	004726R	002	WT410T=	000010	
UNASCI	013360R	002	UP2	003344R	002	WAIT	003424R	002	WRCNT	001726R	002	XXXX =	000000	
.	= 015544R	002												
. ABS.	000000	000												
	000000	001												
RJP11	015544	002												

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

*,DTR6AA/NL:TOC/DOC=DTR6AA.P11
 RUN-TIME: 8 18 1 SECONDS
 RUN-TIME RATIO: 57/28=1.9
 CORE USED: 6K (11 PAGES)

DOCUMENT PAGES: 64

NOS

Session runtime 10 Seconds, 44 KCS, 254 disk reads, 3 disk writes, 64 pages

Date of test 31/12/1999 Mailer: 2.0.0 (1999) #####

#####567890123456789012 ##

000000011111111122222222333333334444444455555555666666667777777788888888999999990000000011111111112222222233312