

```

1      .REM  &
2
3
4
5      IDENTIFICATION
6      -----
7
8      PRODUCT CODE:      AC-F650B-MC
9
10     PRODUCT NAME:      CXTUAB0 TU-58 MODULE
11
12     PRODUCT DATE:      FEBRUARY 1982
13
14     MAINTAINER:        DEC/X11 SUPPORT GROUP
15
16     THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE
17     WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT
18     BY DIGITAL EQUIPMENT CORPORATION.  DIGITAL EQUIPMENT
19     CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT
20     MAY APPEAR IN THIS MANUAL.
21
22     THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE
23     PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER
24     SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITALS
25     COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY
26     OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.
27
28     DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR
29     THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS
30     NOT SUPPLIED BY DIGITAL.
31
32     COPYRIGHT (C) 1979,1982 DIGITAL EQUIPMENT CORPORATION

```

```

34     .SBTTL  REVISION HISTORY
35
36     HISTORY
37     -----
38
39     1-JUNE-82      INITIAL RELEASE      XTUAA0
40     1-MARCH-82     SECOND RELEASE       XTUAB0
41
42     CHANGES TO XTUAA0
43     -----
44
45
46     1)  THIS REVISION ALLOWS THE USER TO EXECUTE TAPE READS AND WRITES
47         IN MODIFIED RADIAL SERIAL PROTOCOL.  THIS FEATURE IS OPTIONAL AND
48         IS SELECTED BY SETTING SWITCH REGISTER TWO, BIT ZERO TO A ONE (SEE
49         SECTION 5, CONFIGURATION PARAMETERS).  THE MODIFICATIONS TO THE
50         PROGRAM MAY BE REFERENCED IN THE COMMENT FIELD BY 'REV.0'.
51
52
53     2)  THIS REVISION ALLOWS THE USER TO SELECT 128 BLOCK/BYTE MODE.
54         THIS FEATURE IS OPTIONAL AND IS SELECTED BY SETTING SWITCH REGISTER
55         THREE, BIT ZERO TO A ONE (SEE CONFIGURATION PARAMETERS, SECTION 5).
56         THE MODIFICATIONS TO THE PROGRAM MAY BE REFERENCED IN THE COMMENT
57         FIELD BY 'REV.1'.
58
59
60     3)  THIS REVISION IS IMPLEMENTED TO MAXIMIZE THROUGHPUT OF THE
61         READ AND WRITE FUNCTIONS.  THREE CHANGES ARE MADE TO ACHIEVE
62         THIS GOAL:
63
64         A)  THE DIAGNOSE COMMAND WILL BE EXECUTED ON THE FIRST PASS ONLY.
65             DRIVE ZERO, IF SELECTED, AND THEN DRIVE ONE IF SELECTED.
66             THE DIAGNOSE COMMAND WILL NOT RUN AGAIN UNLESS THE MODULE IS
67             RESTARTED.
68
69         B)  WRITE/READ TESTING WILL BE PERFORMED ON SEQUENTIAL BLOCKS OF
70             TAPE.  THIS WILL REDUCE THE AMOUNT OF TIME IT TAKES TO SEEK,
71             SINCE THE NEXT BLOCK TO WRITE IS ADJACENT TO THE PREVIOUS ONE.
72
73         C)  THE PASS COUNT IS MODIFIED SO THAT A PASS EQUALS EIGHT DATA
74             TRANSFERS IN '512 BLOCK' MODE, AND TWENTY-SIX DATA TRANSFERS IN
75             '128 BLOCK MODE'.  THIS WILL CAUSE THE PASS COUNT TO BE INCREMENTED
76             APPROXIMATELY ONCE A MINUTE.  THE REASON FOR THIS IS BECAUSE THE
77             MONITOR RELOCATES THE PROGRAM AFTER EVERY PASS.  THEREFORE THE
78             NUMBER OF TRANSFERS PER PASS WILL INCREASE THROUGHPUT IN THE
79             SYSTEM.  THE MODIFICATIONS TO THE PROGRAM MAY BE REFERENCED
80             IN THE COMMENT FIELD BY 'REV.2'.
81
82
83
84     4)  A TEST HAS BEEN ADDED TO VERIFY THE 'ROOT' COMMAND.  THIS IS
85         ACCOMPLISHED BY USING THE RECOMMENDED INITIALIZATION SEQUENCE OF:
86
87         A)  SEND BREAK FOR 7 CHARACTER TIMES
88         B)  RESET BREAK BIT
89         C)  DUMP THE RECEIVE DATA REGISTER
90         D)  SEND TWO INIT CHARACTERS

```

91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109

E) WAIT FOR CONTINUE TO BE RECIEVED
F) SEND 'BOOT' FLAG

THE PROGRAM THEN READS BLOCK ZERO OF THE SELECTED TAPE DRIVE. THIS TEST IS NOT INTENDED TO BE A BOOT. IT IS INTENDED TO VERIFY THE OPERATION OF THE BOOT COMMAND SEQUENCE. THE 'BOOT' COMMAND WILL RUN UNDER 'RSP' OR 'MRSP', AND 512 BYTES/BLOCK OR 128 BYTES/BLOCK MODE. THIS TEST WILL RUN ONLY IF SWITCH REGISTER FOUR IS SET TO A ONE. THIS TEST TRANSMITS THE BOOT COMMAND SEQUENCE WITH INTERRUPTS DISABLED. IT DOES RECIEVE THE BOOTED DATA WITH INTERRUPTS ENABLED. SINCE THE BOOT COMMAND DOES NOT USE HANDSHAKING, IT IS POSSIBLE FOR 'DATA LATE' OR 'OVERRUN' ERRORS TO OCCUR. THE RISK OF THIS ERROR TYPE INCREASES IN SYSTEM WITH MANY INPUT/OUTPUT DEVICES. IN A SYSTEM OF THIS TYPE, THE BOOT COMMAND TEST MAY BE ELIMINATED VIA SWITCH REGISTER FOUR. THE MODIFICATIONS TO THE PROGRAM MAY BE REFERENCED BY 'REV.3' IN THE COMMENT FIELD.

111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167

1.0 ABSTRACT

TUA IS AN IOMOD THAT EXERCISES A TU58 CONTROLLER CONFIGURED WITH EITHER 1 OR 2 DRIVES. THE ACTUAL DEVICE INTERFACE TO THE PDP-11 BUS IS COMPATIBLE WITH THE DL11 FAMILY, AND HOST<-->DEVICE COMMUNICATIONS IS IMPLEMENTED AS SPECIFIED BY RADIAL SERIAL PROTOCOL.

THE EXERCISE ITSELF WILL CONSIST OF A SEQUENCE OF WRITE (RANDOM DATA) OPERATIONS FOLLOWED BY READ SAME BLOCK FOR RANDOM BLOCKS ON TAPE, WITH RETRIES (IF NECESSARY) PERFORMED AT THE DEVICE LEVEL.

ALL ISOLATABLE ERRORS (DATA CHECK OR COMMUNICATIONS) WILL BE REPORTED TO THE CONSOLE DEVICE. UPON OCCURRANCE OF NON-FATAL ERROR, THE DEVICE IS RE-INITIALIZED ANND THE TEST SEQUENCE RE-STARTED AT THE FIRST OPERATION (SELF TEST).

2.0 REQUIREMENTS

HARDWARE: ONE DL11 COMPATIBLE INTERFACE TU58 CONTROLLER WITH AT LEAST ONE DRIVE (DRIVE 0).

STORAGE:
DECIMAL WORDS: 3716
OCTAL WORDS: 3502
OCTAL BYTES: 7204

3.0 PASS DEFINITION

A PASS IS DEFINED BY THE NUMBER OF DATA TRANSFERS.

512 BYTE/BLOCK MODE: 8 READ/WRITE TRANSFERS PER PASS
128 BYTE/BLOCK MODE: 26 READ/WRITE TRANSFERS PER PASS

THE USER MAY EXPECT APPROXIMATELY ONE PASS PER MINUTE, DEPENDING ON OTHER I/O DEVICES BEING TESTED IN THE SYSTEM, AND WHETHER 'RSP' OR 'MRSP' PROTOCOL IS IMPLEMENTED.

4.0 EXECUTION TIME

VARIES WITH COMMUNICATIONS RATE BUT SHOULD BE LESS THAN ONE MINUTE AT 9600 BAUD OR GREATER.

5.0 CONFIGURATION PARAMETERS

```

168 -----
169
170 DEFAULT PARAMETERS:
171
172     DVA: 176500   VCT: 300,   BR1: 4   BR2: 0,   DVC: 1
173
174 AT CONFIGURATION TIME SPECIFY:
175
176     DVC = 1 FOR TEST DRIVE 0 ONLY
177     DVC = 2 FOR TEST DRIVE 1 ONLY
178     DVC = 3 FOR TEST DRIVE 0 AND DRIVE 1
179
180     SR2 = 0 FOR RADIAL SERIAL PROTOCOL
181     SR2 = 1 FOR MODIFIED RADIAL SERIAL PROTOCOL (REVISION 0)
182
183     SR3 = 0 FOR 512 BLOCK PER BYTE MODE / WITH 512 BLOCKS
184     SR3 = 1 FOR 128 BLOCK PER BYTE MODE / WITH 2048 BLOCKS (REVISION 1)
185
186     SR4 = 0 EXECUTES 'DIAGNOSE' COMMAND ON FIRST PASS ONLY
187     SR4 = 1 EXECUTES 'DIAGNOSE' AND 'BOOT' COMMANDS ON PASS ONE ONLY
188
189
190 6.0  DEVICE OPTION SETUP
191 -----
192
193     MAKE SURE SCRATCH TAPE CARTRIDGE IS IN PLACE AND WRITE ENABLED
194     FOR ALL APPROPRIATE DRIVES.
195
196
197 7.0  MODULE OPERATION
198 -----
199
200
201 7.1  SUBMODULE DEFINITIONS
202 -----
203
204     INIT:      THIS ROUTINE PROVIDES RADIAL SERIAL "BREAK" TO
205                THE DEVICE, AND EXPECTS THE APPROPRIATE RESPONSE.
206                IT IS INVOKED UPON STARTUP AND AFTER CERTAIN
207                ERRORS. FAILURE TO INIT IS GROUNDS FOR MODULE
208                DROP.
209
210     SYSINI:    SETS UP BOTH INPUT AND OUTPUT INTERRUPT ROUTINES'
211                VECTORS AND PRIORITIES. INIT'S DRIVE CONFIGURATION
212                AND VARIABLES.
213
214     DECOD:     RETRIEVES NEXT FUNCTIONAL OP-CODE (RD, WR, ETC.)
215                FROM TEST TABLE IN PREPARATION FOR NEXT COMMUNICATION
216                PACKET (COMMAND OR DATA), THEN CREATES THE
217                CORRECT PACKET USING THAT OP CODE THEN
218                SETS UP THE EXPECTED RESPONSE PARAMETERS
219                FROM UNIT : THE EXPECTED RECEIVE PACKET BYTE
220                COUNT(S), AND THE EXPECTED FLAG BYTE(S).
221
222     GVNTRK:    THIS ROUTINE SETS INTERRUPT ENABLE FOR SEND AND
223                RECEIVE, THEN ENTERS WAIT LOOP UNTIL PKCNT IS
224                ASSERTED. IT THEN WAITS ON PKRCV TO COMPLETE,

```

```

225     THEN RETURNS. THESE FLAGS ARE INDICATIVE OF
226     ENTIRE PACKET SENT OR RECEIVED, NOT A SINGLE
227     BYTE.
228
229     SNDRND:    SEND INTERRUPT ROUTINE SENDS PACKET TO DEVICE.
230                ASSERTS PKCNT AT FINISH; DOES RTI.
231
232     RCVHND:    RECEIVE INTERRUPT ROUTINE GETS RESPONSE FROM
233                UNIT (MAY BE MULTIPLE PACKETS) AND CHECKS 1ST
234                BYTE OF EACH PACKET TO DETERMINE IF IT IS WHAT
235                WAS EXPECTED IN ORDER TO ADJUST RECEIVE BYTE
236                COUNT IF NOT. CHECKS FOR EARLY END PACK
237                OR DLV "OVERRUN" (DATA LATE). ASSERTS PKRCVD
238                AT COMPLETION AND RTI'S SINCE UNITS RESPONSE IS
239                CONTINUOUS (NO HANDSHAKE) AND FAST (38.4K BAUD
240                MAX!).
241
242     CHKANS:    BASED ON THE RESPONSE, THIS ROUTINE CALLS ROUTINES
243                TO CHECK CHECKSUM, CHECK END PACKET, AND/OR CHECK DATA.
244
245
246 7.2  DESCRIPTION OF TABLES, QUEUES, BUFFERS
247 -----
248
249     OPTBL:     LIST OF ENCODED WORDS CONTAINING THE NEXT PACKET
250                TO SEND, WHICH DRIVE TO ENABLE, AND IF PACKET IS
251                COMMAND OR DATA.
252
253     SNDBUF:     AREA WHERE PACKET IS BUILT AND SUBSEQUENTLY SENT
254                VIA INTERRUPT ROUTINE. CONTAINS 1 PACKET - BE
255                IT COMMAND OR DATA.
256
257     RCVBUF:     AREA WHERE PACKET(S) ARE RECEIVED DURING INTERRUPT
258                SERVICE. MAY CONTAIN MAX OF: [DATA-DATA-DATA-
259                DATA-END] PACKETS IN SEQUENCE.
260
261
262 8.0  OPERATOR OPTIONS
263 -----
264
265     A.  USE THE "MOD" COMMAND TO DUMP THE BUFFERS IN ORDER TO MAXIMIZE
266         ERROR INFORMATION.
267
268
269
270
271
272
273
274 000000      IOMOD <TUAB >,176500,300,4,0,2,1,17
      000000      MODULE 140000,TUAB ,176500,300,4,0,2,1,17
      ;          .TITLE TUAB DEC/X11 SYSTEM EXERCISER MODULE
      ;          DDXCOM VERSION 6      23-MAY-78
      ;          .LIST BIN
      ;*****
      BEGIN:
      000000      MODNAM: .ASCII /TUAB / ;MODULE NAME.
      000000      124      125      101

```

```

000003      102      040
000005      000
000006      176500
000010      000300
000012      200
000013      000
000014      000003
000016      000000
000020      000000
000022      000000
000024      000000

XFLAG: .BYTE OPEN ;USED TO KEEP TRACK OF WBUFF USAGE
ADDR: 176500+0 ;1ST DEVICE ADDR.
VECTOR: 300+0 ;1ST DEVICE VECTOR.
BR1: .BYTE PRTY4+0 ;1ST BR LEVEL.
BR2: .BYTE PRTY0+0 ;2ND BR LEVEL.
DVID1: 2+1 ;DEVICE INDICATOR 1.
SR1: OPEN ;SWITCH REGISTER 1
SR2: OPEN ;SWITCH REGISTER 2
SR3: OPEN ;SWITCH REGISTER 3
SR4: OPEN ;SWITCH REGISTER 4
;*****
STAT: 140000 ;STATUS WORD.
INIT: START ;MODULE START ADDR.
SPOINT: MODSP ;MODULE STACK POINTER.
PASCNT: 0 ;PASS COUNTER.
ICOUNT: 1 ;# OF ITERATIONS PER PASS=1
ICOUNT: 0 ;LOC TO COUNT ITERATIONS
SOFCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
HRDCNT: 0 ;LOC TO SAVE TOTAL HARD ERRORS
SOFPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
HRDPAS: 0 ;LOC TO SAVE HARD ERRORS PER PASS
SYSCNT: 0 ;# OF SYS ERRORS ACCUMULATED
RANNUM: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
CONFIG:
RES1: 0 ;RESERVED FOR MONITOR USE
RES2: 0 ;RESERVED FOR MONITOR USE
SVR0: OPEN ;LOC TO SAVE R0.
SVR1: OPEN ;LOC TO SAVE R1.
SVR2: OPEN ;LOC TO SAVE R2.
SVR3: OPEN ;LOC TO SAVE R3.
SVR4: OPEN ;LOC TO SAVE R4.
SVR5: OPEN ;LOC TO SAVE R5.
SVR6: OPEN ;LOC TO SAVE R6.
CSRA: OPEN ;ADDR OF CURRENT CSR.
SBADR: ;ADDR OF GOOD DATA, OR
ACSR: OPEN ;CONTENTS OF CSR.
WASADR: ;ADDR OF BAD DATA, OR
ASTAT: OPEN ;STATUS REG CONTENTS.
ERRTYP: ;TYPE OF ERROR
ASB: OPEN ;EXPECTED DATA.
AWAS: OPEN ;ACTUAL DATA.
RSTRT: RSTRT ;RESTART ADDRESS AFTER END OF PASS
WOTO: OPEN ;WORDS TO MEMORY PER ITERATION
WDFR: OPEN ;WORDS FROM MEMORY PER ITERATION
INTR: OPEN ;# OF INTERRUPTS PER ITERATION
IDNUM: 17 ;MODULE IDENTIFICATION NUMBER=17
      .REPT SPSIZ ;MODULE STACK STARTS HERE.
      .NLIST
      .WORD 0
      .LIST
      .ENDR

000224      MODSP:
;*****

```

```

276
290      .SBTTL MODULE DATA SECTION
291      ;---- FLAGS -- ETC -----
292 000224 000000 PKSENT: .WORD ;PACKET SENT
293 000226 000000 PKRCVD: .WORD ;PACKET RECEIVED
294 000230 000000 LASTOP: .WORD ;1_PASS_THROUGH_TABLE FLAG
295 000232 000000 FSTFLG: .WORD ;EXPECTING_FLAG_BYTE FLAG
296 000234 000000 RCSR: .WORD ;RCV STATUS REG. ADDR.
297 000236 000000 RCDB: .WORD ;RCV DATA BUFFER ADDR.
298 000240 000000 XMSR: .WORD ;TRANSMIT STATUS REG. ADDR.
299 000242 000000 XMDB: .WORD ;TRANSMIT DATA BUFFER ADDR.
300 000244 000000 DVIDX: .WORD ;DRIVE CONFIGURATION
301
302
303      ;---- OTHER STORAGE -----
304
305 000246 000000 SYSTAT: .WORD ;SYSTEM STATUS WORD
306 ;BIT3=OUT-TO-LUNCH ERROR (CHKANS)
307 ;BIT1=SELF TEST SENT
308 ;BIT0=CHKSUM ODD
309 000250 000000 DRV: .WORD ;CURRENT DRIVE NUMBER
310 000252 .BLKW 2 ;ASCII OF CURRENT DRIVE NUMBER
311 000256 000000 .WORD 0
312 000260 000000 .WORD 0
313
314 000262 ASBDBT: .BLKW 2 ;ASCII OF # OF BAD BYTES
315 000266 000000 .WORD 0
316 000270 000000 .WORD 0
317
318 000272 000000 REC: .WORD ;RECORD NUMBER
319 000274 000000 PATRN: .WORD ;DATA PATTERN
320 000276 000000 SUCCS: .WORD ;SUCCESS CODE OF END PACKET
321 000300 000000 PKPTR: .WORD ;TOP OF CURRENT PACKET (ADDR.)
322 000302 000000 XSPTR: .WORD ;POINTER TO NEXT X$FLG OR X$CNT
323 000304 000000 TBLTOP: .WORD ;TOP OF OP TABLE
324 000306 000000 TBLPTR: .WORD ;POINTER TO OP CODE IN OPTBL
325 000310 000000 SNDPTR: .WORD ;POINTER TO XMIT BUFFER
326 000312 000000 SND CNT: .WORD ;NUMBER OF BYTES REMAINING TO SEND
327 000314 000000 RCVPTR: .WORD ;POINTER IN RECEIVE BUFFER
328 000316 000000 RCPKNN: .WORD ;# OF PACKETS LEFT DURING RECIEVE
329 000320 000000 PKCHKD: .WORD ;# OF PACKETS BEING CHECKED IN CHKANS
330 000322 000000 RVCNT: .WORD ;ACTUAL COUNT USED IN RECEIVE
331 000324 000000 RCB CNT: .WORD ;COUNT USED IN CHECKANS
332 000326 000000 X$PKNN: .WORD ;EXPECTED # OF PACKETS TOTAL
333 000330 000000 X$FLG: .WORD ;THE EXPECTED FLAG BYTE OF 1ST PACKET
334 000332 000000 X$CNT: .WORD ;AND IT'S TOTAL BYTE COUNT
335 000334 .BLKW 8. ;MULTIPLE PACKET EXPECTED FLAGS AND COUNTS
336 000354 000000 ERROP: .WORD ;THE TABLE OP CODE ON ERROR
337 000356 000000 ERRFLG: .WORD ;THE EXPECTED FLAG ON ERROR (X$SPTR)
338 000360 000000 ERRER: .WORD ;ERROR FLAG
339 000362 000000 ERRWD: .WORD ;CONTENTS OF DLV RCDB ON DLV ERROR
340 000364 000000 ATPAK: .WORD ;#PKPTR ON ERROR-1ST 2 BYTES
341 000366 000000 ATPAK2: .WORD ;#PKPTR+2 ON ERROR-NEXT 2 BYTES
342 000370 000000 BDBYTS: .WORD ;NUMBER OF BAD DATA BYTES (MAX. 128.)
343 000372 000000 M$RSP: .WORD 0 ;*** REV.0= 0 FOR RSP / 1 FOR MRSP
344 000374 000000 B$SIZE: .WORD 0 ;*** REV.1= 0 FOR 512 / 1 FOR 128 BLK/BYTE MODE
345 000376 000000 B$TEST: .WORD 0 ;*** REV.3= FLAG FOR BOOT TEST

```

```

346 000400 001000      BLKSIZ: ,WORD 512.      ;*** REV.1- ADDED FOR 128/512 BYTES/BLOCK MODE
347 000402 000000      DSFLAG: ,WORD 0      ;*** REV.1- ADDED FOR 128/512 BLOCK MODE
348 000404 000000      PSDATA: ,WORD 0      ;*** REV.1- ADDED FOR 1/4 DATA PACK FLAG
349
350
351
352 000406 000000      BRKWD: ,WORD 0      ;2 NULL BYTES
353 000410 004        INITWD: ,BYTE R$INIT    ;INIT BYTE
354 000411 000        ,BYTE 0      ;R$CONT EXPECTED HERE
355
356
357
358
359 000100      E1 = 100
360 000100      D1 = 100
361 000000      D$DATA = 0
362 000001      DTAFL1 = 1
363 000010      DTAFL2 = 10
364 000011      DTAFL3 = 11
365 000100      DTAFL4 = 100
366 100000      CMNOFL = 100000
367 000000      DRV0 = 0
368 000400      DRV1 = 400
369 000001      BIT0 = 1
370 000002      BIT1 = 2
371 000004      BIT2 = 4
372 000010      BIT3 = 10
373 000020      BIT4 = 20
374 000040      BIT5 = 40
375 000100      BIT6 = 100
376 000200      BIT7 = 200
377 000400      BIT8 = 400
378 001000      BIT9 = 1000
379 002000      BIT10 = 2000
380 004000      BIT11 = 4000
381 010000      BIT12 = 10000
382 020000      BIT13 = 20000
383 040000      BIT14 = 40000
384 100000      BIT15 = 100000
385
386
387
388
389
390
391
392 000002      R$CMND = 2
393 000020      R$CONT = 20
394 000020      R$XON = 20
395 000020      R$CONT = 20
396 000023      R$XOFF = 23
397 000010      R$BOOT = 10
398 000004      R$INIT = 4
399 000001      R$DATA = 1
400 000002      R$END = R$CMND
401
402
;SIZES:

```

```

403
404 000016      R$NDSZ = 14.      ;END PACK SIZE
405 000012      R$MSIZ = 10.      ;MESSAGE SIZE
406 000204      R$DASZ = 132.      ;DATA PACK SIZE
407 000222      R$DNSZ = R$DASZ+R$NDSZ ;DATA PLUS END PACK SIZE
408 000016      R$SNSZ = R$MSIZ+4   ;SEND SIZE OF MESSAGE PACK
409 000001      P$DAT1 = 1.      ;*** REV.1- ONE DATA PACK FOR 128. MODE
410 000004      P$DAT4 = 4.      ;*** REV.1- FOUR DATA PACKS FOR 512. MODE
411
412
413
414 000100      R$SEND = 100
415 000003      R$SWR = 3
416 000002      R$SRD = 2
417 000005      R$SEK = 5
418 000000      R$SNOP = 0
419 000001      R$SNIT = 1
420 000007      R$SLF = 7
421 000010      R$BOO = 10
422
423
424
425 000000      E$OK = 0      ;GOOD SUCCESS
426 177720      E$ABO = -48.    ;BAD COMMAND
427 177767      E$NCRT = -9.    ;NO CART.
428 177770      E$NONX = -8.    ;NO DRIVE
429 177776      E$PART = -2.    ;E.O.M.
430 177740      E$SK = -32.     ;SEEK ERROR
431 000001      E$TRY = 1      ;RETRIES
432 177765      E$WLOC = -11.   ;WRITE PROT.
433 177737      E$NOMO = -33.   ;MOTOR STOPPED
434 177720      E$CMD = -48.    ;BAD COMMAND
435 177711      E$REC = -55.    ;BAD RECORD # SENT
436 177757      E$CKS = -17.   ;DATA CHECK ERROR
437 177777      E$SLF = -1.    ;SELF TEST FAIL
438 177757      E$CKSM = E$CKS
439 177757      E$WR = E$CKS
440 177757      E$RD = E$CKS
441
442

```

```
444
445
446 000412 100007
447 000414 100010
448 000416 100407
449 000420 100410
450 000422 177777
451 000424 100003
452 000426 000001
453 000430 000010
454 000432 000011
455 000434 000100
456 000436 100002
457 000440 100403
458 000442 000401
459 000444 000410
460 000446 000411
461 000450 000500
462 000452 100402
463 000454 177777
464
465
466 000456 100403
467
468
469
470
471 000460 000354'
472 000462 000272'
473 000464 000274'
474 000466 000314'
475 000470 000356'
476 000472 000300'
477 000474 000410'
478 000476 000362'
479 000500 000322'
480 000502 000324'
481 000504 000326'
482 000506 000330'
483 000510 000332'
484 000512 000320'
485 000514 000364'
486 000516 000366'
487 000520 177777
488
489

/----- TEST COMMAND TABLE -----
SELFTT: .WORD R$SLF:CMNDL:DRV0
        .WORD R$BOOT:CMNDL:DRV0
        .WORD R$SLF:CMNDL:DRV1
        .WORD R$BOOT:CMNDL:DRV1
        .WORD -1
OPTBL0: .WORD R$SWR:CMNDL:DRV0
        .WORD D$DATA:DTAFL1:DRV0
        .WORD D$DATA:DTAFL2:DRV0
        .WORD D$DATA:DTAFL3:DRV0
        .WORD D$DATA:DTAFL4:DRV0
        .WORD R$SRD:CMNDL:DRV0
OPTBL1: .WORD R$SWR:CMNDL:DRV1
        .WORD D$DATA:DTAFL1:DRV1
        .WORD D$DATA:DTAFL2:DRV1
        .WORD D$DATA:DTAFL3:DRV1
        .WORD D$DATA:DTAFL4:DRV1
        .WORD R$SRD:CMNDL:DRV1
        .WORD -1

DR1OP: .WORD R$SWR:CMNDL:DRV1
/-----
/----- TABLE FOR EXTENDED ERROR INFO -----
ERRTBL: .WORD ERROP          ;OP CODE IN COMMAND TABLE (OPTBL)
        .WORD REC           ;THE RECORD NUMBER
        .WORD PATRN         ;THE DATA PATTERN
        .WORD RCVPTR        ;BYTE POINTER IN RECIEVE BUFFER
        .WORD ERRFLG        ;THE EXPECTED FLAG BYTE OF CURRENT PACK
        .WORD PKPTR         ;POINTER TO TOP OF CURRENT PACKET
        .WORD INITWD        ;RESPONSE OF INIT OP (BIT15-BIT8)
        .WORD ERRWD         ;CONTENTS OF DLV RCDB IF DLV ERROR
ERRTB2: .WORD RCVCNT         ;# BYTES UNRECEIVED, IF ANY
        .WORD RCBCNT        ;# BYTES EXPECTED CURRENT PACKET
        .WORD X$PKNM        ;# OF PACKETS EXPECTED
        .WORD X$FLG         ;THE EXPECTED FLAG BYTE OF 1ST PACKET
        .WORD X$CNT         ;THE EXPECTED COUNT OF 1ST PACKET
        .WORD PKCHKD        ;# OF CURRENT PACKET BEING CHECKED
        .WORD ATPAK         ;1ST AND 2ND BYTES OF CURRENT PACKET
        .WORD ATPAK2        ;3RD AND 4TH BYTES OF CURRENT PACKET
        .WORD -1
/-----
```

```
491
492
493 000522 016706 177304
494 000526 016767 177262 177510
495 000534 126737 177362 000041
496 000542 001016
497 000544 105737 000040
498 000550 001004
499 000552 042767 000001 177464
500 000560 000407
501 000562 122737 000001 000040
502 000570 001003
503 000572 042767 000002 177444
504 000600 004767 000364
505 000604 005067 177462
506 000610 000240
507 000612 005267 177454
508
509
510 000616 005767 177552
511 000622 001404
512 000624 042767 174000 177440
513 000632 000403
514 000634 042767 177000 177430
515
516 000642
517 000642 104417 000000'
518 000646 016767 177202 177420
519 000654 042767 177400 177412
520
521 000662 004767 000764
522
523 000666 004767 002332
524 000672 004767 000102
525 000676 004767 002346
526 000702 005767 177452
527 000706 001413
528
529 000710 016767 177370 177370
530 000716 005067 177436
531 000722 005067 177320
532 000726 005067 177154
533 000732 004767 002052
534 000736 000240
535 000740 004767 000706
536 000744 005767 177260
537 000750 001746
538 000752 104413 000000'
539 000756 000714
540
541 000760 042777 000100 177246
542 000766 042777 000100 177244
543 000774 104410 000000'
544
545

.SBTTL MAIN PROGRAM CONTROL
START: MOV SPOINT,SP ;THE INIT OF INITS
      MOV DVID1,DVIDX ;COPY DRIVE CONFIGURATION
      CMPB IDNUM,0#41 ;ARE WE THE LOAD DEVICE?
      BNE 2S ;NO
      TSTB #40 ;YES-DRIVE 0?
      BNE 1S ;NO
      BIC #BIT0,DVIDX ;YES-DESELECT
      BR 2S ;CONTINUE
1S: CMPB #1,0#40 ;NOT DR0,DR1?
    BNE 2S ;NO-SOME OTHER DR
    BIC #BIT1,DVIDX ;YES DESELECT
2S: CALL SYSINI ;SET VECTOR GET CONFIG.
    CLR REC ;*** REV.2- SEQUENTIAL RECORDS
    RESTRT: NOP ;MAIN LOOP
    INC REC ;*** REV.2- SEQUENTIAL RECORDS FOR READ
           ;*** REV.2- AND WRITE'S.
           ;*****
           TST B$SIZE ;*** REV.1- IF 512 BYTE MODE
           BEQ 1S ;*** REV.1- THEN,
           BIC #174000,REC ;*** REV.1- ELSE, REC= 0 TO 2047 BLOCKS
           BR 2S ;*** REV.1-
1S: BIC #177000,REC ;*** REV.1- RECORD= 0 TO 512 BLOCKS
   ;*****
2S: RANDS,BEGIN
    MOV RANNUM,PATRN ;PROCURE DATA
    BIC #177400,PATRN ;1-BYTE IT
    CALL DECOD ;INTERPRET OPERATION FROM TABLE; MAKE PACKET
RSTR2: CALL CLRBUF ;ZERO RECEIVE BUFFER
      CALL GVNTAK ;SEND PACKET, RECEIVE RESPONSE
      CALL CHKANS ;INTERPRET RESPONSE
      TST ERREH ;ERROR?
      BEQ 1S ;NO
3S: MOV IBLTOP,TBLPTR ;YES, -->1ST OPERATION.
    CLR ERRER ;RESET ERROR FLAG
    CLR SYSTAT ;RESET SYSTEM STATUS
    CLR ERRTPY ;RESET ERROR TYPE
    CALL DOBRK ;INIT DEVICE
1S: NOP
    CALL DECOD ;DECODE NEXT OPERATION
    TST LASTOP ;DONE ONE TABLE'S WORTH?
    BEQ RSTR2 ;NOT YET
    ENDTB,BEGIN ;SIGNAL END OF ITERATION.
    BR RESTRT ;MONITOR SHALL TEST END OF PASS
           ;KEEP ON
/-----
DROP: BIC #DI,0RCR ;SHUT OFF
      BIC #DI,0XMSR ;ALL INTERRUPTS
      ENDS,BEGIN
/-----
```

TUAB DEC/X11 SYSTEM EXERCISER M MACRO M1113 02-MAR-82 08:30 PAGE 9
GVNTAK / MAIN COMMUNICATIONS LOOPS
SES 0014

Address	Disassembly	Hex	Comment
584	SETUP		
585	SETUP		
586	SETUP		
587	SETUP		
588	SETUP		
589	SETUP		
590	SETUP		
591	SETUP		
592	SETUP		
593	SETUP		
594	SETUP		
595	SETUP		
596	SETUP		
597	SETUP		
598	SETUP		
599	SETUP		
600	SETUP		
601	SETUP		
602	SETUP		
603	SETUP		
604	SETUP		
605	SETUP		
606	SETUP		
607	SETUP		
608	SETUP		
609	SETUP		
610	SETUP		
611	SETUP		
612	SETUP		
613	SETUP		
614	SETUP		
615	SETUP		
616	SETUP		
617	SETUP		
618	SETUP		
619	SETUP		
620	SETUP		
621	SETUP		
622	SETUP		
623	SETUP		
624	SETUP		
625	SETUP		
626	SETUP		
627	SETUP		
628	SETUP		
629	SETUP		
630	SETUP		
631	SETUP		
632	SETUP		
633	SETUP		
634	SETUP		
635	SETUP		
636	SETUP		
637	SETUP		
638	SETUP		
639	SETUP		
640	SETUP		
641	SETUP		
642	SETUP		
643	SETUP		
644	SETUP		
645	SETUP		
646	SETUP		
647	SETUP		
648	SETUP		
649	SETUP		
650	SETUP		
651	SETUP		
652	SETUP		
653	SETUP		
654	SETUP		
655	SETUP		
656	SETUP		
657	SETUP		
658	SETUP		
659	SETUP		
660	SETUP		
661	SETUP		
662	SETUP		
663	SETUP		
664	SETUP		
665	SETUP		
666	SETUP		
667	SETUP		
668	SETUP		
669	SETUP		
670	SETUP		
671	SETUP		
672	SETUP		
673	SETUP		
674	SETUP		
675	SETUP		
676	SETUP		
677	SETUP		
678	SETUP		
679	SETUP		
680	SETUP		
681	SETUP		
682	SETUP		
683	SETUP		
684	SETUP		
685	SETUP		
686	SETUP		
687	SETUP		
688	SETUP		
689	SETUP		
690	SETUP		
691	SETUP		
692	SETUP		
693	SETUP		
694	SETUP		
695	SETUP		
696	SETUP		
697	SETUP		
698	SETUP		
699	SETUP		
700	SETUP		

```
640 001452 032767 000001 176342      BIT      #BIT0,SR3      ;*** REV.1- IF 512. MODE
641 001460 001425      BEQ      28      ;*** REV.1- THEN-
642 001462 005267 176706      INC      #SSIZE      ;*** REV.1- ELSE, SET 128. BLOCK FLAG
643 001466 012767 000200 176704      MOV      #128,,BLKSIZ      ;*** REV.1- AND, SET BLOCK SIZE
644 001474 012767 000001 176702      MOV      #PSDAT1,PSDATA      ;*** REV.1- AND, EXPECT ONE DATA PACK
645 001502 012767 000001 176672      MOV      #DTAFL1,DSFLAG      ;*** REV.1- AND, SET ONE PACK FLAG
646 001510 050024      BIS      R0,(R4)+      ;*** REV.1- SET FLAG FOR SKIPPING PACKETS
647 001512 005302      DEC      R2      ;*** REV.1-
648 001514 001375      BNE      108:      ;*** REV.1-
649 001516 050025      BIS      R0,(R5)+      ;*** REV.1- SET FLAG FOR DRIVE 1
650 001520 005303      DEC      R3      ;*** REV.1-
651 001522 001375      BNE      208:      ;*** REV.1-
652 001524 012767 000032 176304      MOV      #26,,ICONT      ;*** REV.2- NO. OF READ/WITES PER PASS
653 001532 000422      BR      458      ;*** REV.1- PARAMETERS FOR 128. MODE COMPLETED
654 001534 012767 001000 176636 28:      MOV      #512,,BLKSIZ      ;*** REV.1- PARAMETERS FOR 512. MODE
655 001542 012767 000004 176634      MOV      #PSDAT4,PSDATA      ;*** REV.1- EXPECT FOUR DATA PACKS
656 001550 012767 000100 176624      MOV      #DTAFL4,DSFLAG      ;*** REV.1- SET FOUR DATA PACK FLAG
657 001556 040024      BIC      R0,(R4)+      ;*** REV.1- RESET FLAG FOR SKIPPING PACKETS
658 001560 005302      DEC      R2      ;*** REV.1-
659 001562 001375      BNE      408:      ;*** REV.1-
660 001564 040025      BIC      R0,(R5)+      ;*** REV.1- RESET FLAG FOR DRIVE 1
661 001566 005303      DEC      R3      ;*** REV.1-
662 001570 001375      BNE      408:      ;*** REV.1-
663 001572 012767 000010 176236 458:      MOV      #8,,ICONT      ;*** REV.2- NO. OF READ/WITES PER PASS
664 001600 005067 176566      CLR      MRSP      ;*** REV.0- RESET MRSP FLAG
665 001604 032767 000001 176206      BIT      #BIT0,SR2      ;*** REV.0- IF NOT MRSP,
666 001612 001402      BEQ      508:      ;*** REV.0- THEN,
667 001614 005267 176552      INC      MRSP      ;*** REV.0- ELSE, SET MRSP FLAG
668 001620      508:
669
670      ;*****
671
672 001620 004767 001164 38:      CALL      DOBRK      ;INIT UNIT
673 001624 005067 176530      CLR      ERRER      ;AND ERROR NUMBER
674 001630 005067 176526      CLR      ERRWD      ;AND ERROR WORD
675 001634 005067 176530      CLR      BDBYTS      ;AND BAD BYTES
676 001640 005067 176402      CLR      SYSTAT      ;AND SYSTEM STATUS
677 001644 005067 176236      CLR      ERRTP      ;AND ERROR TYPE
678 001650 000207      RTS      PC
679
```

```
681
682 001652 027727 176430 177777 DECOD: CMP      #TBLPTR,-1      ;LAST OP CODE PROCESSED?
683 001660 001007      BNE      18      ;NO
684 001662 016767 176416 176416      MOV      TBLTOP,TBLPTR      ;YES-POINT TO TOP
685 001670 005267 176334      INC      LASTOP      ;SET FLAG
686 001674 000167 000612      JMP      DECRET      ;EXIT
687 001700 012700 005670' 18:      MOV      #TRBUF,R0      ;POINT TO TOP OF TRANSMIT BUFFER
688      ;*****
689 001704 032777 040000 176374      BIT      #BIT14,@TBLPTR      ;*** REV.1.2- SKIP THIS DATA PACK ?
690 001712 001402      BEQ      108:      ;*** REV.1.2- NO, CONTINUE
691 001714 000167 000600      JMP      NOPACK      ;*** REV.1.2- YES, GET NEXT PACK.
692 001720      108:
693      ;*****
694 001720 005777 176362      TST      @TBLPTR      ;IS IT COMMAND?
695 001724 100462      BMI      DECCMD      ;YES
696 001726 012700 005670'      MOV      #TRBUF,R0      ;NO-THEREFORE IS DATA PACK:
697 001732 112710 000001      MOV      #RSDATA,@R0      ;FLAG BYTE
698 001736 112760 000200 000001      MOV      #128,,1(R0)      ;BYTE COUNT
699 001744 012702 000200      MOV      #128,,R2      ;FOR DATA INSERTION COUNT
700 001750 005720      TST      (R0)+      ;GET BY THE BYTE COUNT
701 001752 116720 176316 28:      MOV      PATRN,(R0)+      ;AND INSERT DATA
702 001756 005302      DEC      R2      ;UNTIL
703 001760 001374      BNE      28:      ;COMPLETED
704 001762 012767 000204 176322      MOV      #132,,SND CNT      ;TOTAL COUNT FOR SENDING
705 001770 012701 000202      MOV      #130,,R1      ;COUNT FOR CHECKSUM ROUTINE
706 001774 012700 005670'      MOV      #TRBUF,R0      ;POINT TO TOP AGAIN
707 002000 004767 002230      CALL      CHKSUM      ;CALC. THE CHECKSUM
708 002004 010110      MOV      R1,(R0)      ;RETURNED IN R1 WITH R0 AS POINTER
709 002006 036777 176370 176272      BIT      DSFLAG,@TBLPTR      ;*** REV.1- 128/512 BLOCK MODE
710 002014 001413      BEQ      38      ;NO, EXPECT CONTINUE, ETC.
711 002016 012767 000002 176304      MOV      #RSEND,XSFLG      ;YES, EXPECT END PACK
712 002024 012767 000016 176300      MOV      #RSENDZ,XSCNT      ;THIS BIG
713 002032 012767 000001 176266      MOV      #1,XSPKNM      ;AND ONLY 1 PACKET RESPONSE
714 002040 000167 000412      JMP      DECRET      ;EXIT
715 002044 012767 000020 176256 38:      MOV      #RSCONT,XSFLG      ;HERE NOT LAST PACKET
716 002052 012767 000001 176252      MOV      #1,XSCNT      ;EXPECT: CONTINUE; 1 BYTE
717 002060 012767 000001 176240      MOV      #1,XSPKNM      ;1 PACK
718 002066 000167 000364      JMP      DECRET      ;EXIT
719
720 002072 000240      DECCMD: NOP      ;HERE WE BUILD A COMMAND PACKET:
721      ;*****
722 002074 122777 000010 176204      CMPB      #RBOOT,@TBLPTR      ;*** REV.3- IS IT A BOOT COMMAND ?
723 002102 001004      BNE      88      ;*** REV.3- NO, PROCEED AS USUAL
724 002104 004767 002520      JSR      PC,BOOT      ;*** REV.3- YES, TRY BOOT IN WHATEVER PROTOCOL
725 002110 000167 000404      JMP      NOPACK      ;*** REV.3- EXIT, THIS COMMAND ISN'T A PACKET.
726 002114      88:
727      ;*****
728 002114 112710 000002      MOV      #RSCMD,@R0      ;FLAG BYTE
729 002120 112760 000012 000001      MOV      #RMSIZ,1(R0)      ;MESSAGE SIZE
730 002126 117760 176154 000002      MOV      #TBLPTR,2(R0)      ;THE OP CODE
731      ;*****
732 002134 105060 000003      CLRB      3(R0)      ;*** REV.1- PRESET MODIFIER
733 002140 005767 176230      TST      BSSIZE      ;*** REV.1- IF 128 BLOCK/BYTE MODE
734 002144 001403      BEQ      108:      ;*** REV.1- THEN,
735 002146 152760 000200 000003      BISB      #BIT7,3(R0)      ;*** REV.1- ELSE, SET MODIFIER BIT
736 002154      108:
737      ;*****
```



```

738 002154 105060 000004      CLR B 4(R0)      ;PRESET TO DRIVE 0
739 002160 005067 176064      CLR DRV          ;DITTO
740 002164 032777 000400 176114 BIT #BIT8,@TBLPTR ;WAS DRIVE 1, THOUGH?
741 002172 001404              BEQ 4S              ;NO
742 002174 105260 000004      INCB 4(R0)          ;YES-SELECT DR1
743 002200 005267 176044      INC DRV             ;DITTO
744 002204              4S:
;*****
;CONVERT DRV TO ASCII AND
;STORE AT ASDRV
002204 104421 000000' 000250' BTOD$,BEGIN,DRV,ASDRV
002212 000252'
;*****
745 ;*****
746 002214 105060 000005      CLR B 5(R0)      ;*** REV.0- PRESET SWITCH
747 002220 005767 176146      TST #MRSP          ;*** REV.0- IF NOT MRSP,
748 002224 001403              BEQ 5S              ;*** REV.0- THEN,
749 002226 152760 000010 000005 BISB #BIT3,5(R0) ;*** REV.0- ELSE, SET MRSP SWITCH
750 002234 005060 000006      CLR 6(R0)          ; RESET SEQUENCE NUMBER
751 002240 016760 000010 000010 MOV #LKSI2,B.(R0) ;*** REV.1- SETUP FOR 128/512 BLOCK SIZE
752 ;*****
753 002246 016760 176020 000012 MOV REC,10.(R0) ;INSERT RECORD NUMBER
754 002254 012701 000012      MOV #RMSIZ,R1      ;MESSAGE PACKET SIZE
755 002260 005721              TST (R1)+          ;PLUS BYTE COUNT AND FLAG
756 002262 012767 000016 176022 MOV #RSSNSZ,SND CNT ; SIZE TO SEND
757 002270 004767 001740      CALL CHKSUM        ;GET CHECKSUM, RETURNED
758 002274 010110              MOV R1,@R0         ;IN R1 WITH R0-->WHERE
759 ;THE CHECKSUM WORD GOES
760 002276 042767 000002 175742 BIC #BIT1,SYSTAT ;NOT SELF TEST PRESET
761 002304 122777 000007 175774 CMPB #RSSSLF,@TBLPTR ;WAS THE OPERATION SELF TEST?
762 002312 001016              BNE 1S              ;NO
763 002314 052767 000002 175724 BIS #BIT1,SYSTAT ;YES-SELF TEST FLAG SET
764 002322 012767 000002 176000 MOV #RSEND,XSFLG ;YES-EXPECT END PACK
765 002330 012767 000016 175774 MOV #RNSDSZ,XSCNT ;OF PROPER SIZE
766 002336 012767 000001 175762 MOV #1,XSPKNM ;AND ONLY 1 PACKET
767 002344 000167 000106      JMP DECR1          ;EXIT
768 002350 122777 000003 175730 1S: CMPB #RSSWR,@TBLPTR ;NOT SELF TEST-WAS IT WRITE?
769 002356 001013              BNE 2S              ;NO
770 002360 012767 000020 175742 MOV #RSCONT,XSFLG ;YES-EXPECT CONTINUE
771 002366 012767 000001 175736 MOV #1,XSCNT ;AND 1 BYTE
772 002374 012767 000001 175724 MOV #1,XSPKNM ;AND 1 PACKET
773 002402 000167 000050      JMP DECR1          ;EXIT
774 002406 005000              2S: CLR R0          ;HERE TRANSACTION MUST BE READ
775 002410 012703 000330'      MOV #XSFLG,R3 ;GET ADDR. OF XSFLG
776 002414 012723 000001 175736 MOV #R6DATA,(R3)+ ;INSERT TOTAL OF
777 002420 012723 000204      MOV #132.,(R3)+ ; 14 DATA FLAG BYTES EXPECTED,
778 002424 005200              INC R0             ;SANDWICHED WITH
779 002426 026700 175752      CMP #R6DATA,R0 ;*** REV.1- 1/4 DATA PACKETS EXPECTED
780 002432 001370              BNE 3S              ;EACH
781 002434 012723 000002      MOV #RSEND,(R3)+ ;THEN THE END PACK
782 002440 012713 000016      MOV #RNSDSZ,(R3) ;AND ITS SIZE
783 002444 016767 175734 175654 MOV #R6DATA,XSPKNM ;*** REV.1- EXPECT 2 OR 5 PACKETS
784 002452 005267 175650      INC XSPKNM         ;*** REV.1-
785
786 002456 017767 175624 175670 DECR1: MOV @TBLPTR,ERROP ;*** REV.0- COPY OP CODE IN CASE ERROR
787 002464 005067 175540      CLR LASTOP       ;*** REV.0- INDICATE STILL IN TABLE
788 002470 005777 175612      TST @TBLPTR        ;*** REV.0- IF, PACKET IS DATA

```

```

789 002474 100002              BPL 1S              ;*** REV.0- THEN, JUST EXIT
790 002476 004767 002340      JSR PC,HND SHK      ;*** REV.0- ELSE, 'MRSP CONT.', THANKYOU
791 002502 062767 000002 175576 1S: ADD #2,TBLPTR ;*** REV.0- UPDATE POINTER FOR NEXT OPERATION
792 002510 000207              RTS PC
793 002512 004767 002324      DECRET: JSR PC,HND SHK ;*** REV.0- 'MRSP CONT.' TO SATISFY CONTROLLER
794 002516 000207              RTS PC
795 002520 062767 000002 175560 NOPACK: ADD #2,TBLPTR ;*** REV.1- UPDATE POINTER FOR NEXT OPERATION
796 002526 000167 177120      JMP DECOD          ;*** REV.1- DECODE NEXT PACKET
797
798

```

```

800
801
802 .SBTTL / INTERRUPT SERVICE ROUTINES /
803 002532 117777 175552 175502 SNDHND: MOVB @SNDPTR,@XMDB ;OUTPUT BYTE
804 002540 005267 175544 INC SNDCNT ;-->NEXT
805 002544 005367 175542 DEC SNDCNT ;ANY MORE?
806 002550 001005 BNE 1$ ;YES
807 002552 005267 175446 INC PKCNT ;NO-SET FLAG
808 002556 042777 000100 175454 BIC #DI,@XMSR ;AND NO MORE INTS
809 002564 000002 1$ RTI ;FAST AND FURIOUS
810
811 002566 042777 000100 175440 RCVHND: BIC #DI,@RCR ;KILL ANY NESTING
812 002574 017767 175436 175560 MOV @RCDB,ERRWD ;GET BYTE+ERROR
813 002602 116777 175554 175504 MOVB @RCDB,ERRWD ;SAVE BYTE IN BUFFER
814 002610 005767 175546 IST ERRWD ;ERROR?
815 002614 100004 BPL RCVOK ;NO
816 002616 005267 175404 INC PKRCVD ;YES SET RCVD FLAG
817 002622 000240 NOP
818 002624 000461 BR RCVERR ;EXIT WITH NO HANSHAKING
819 002626 005067 175530 RCVOK: CLR ERRWD ;NO-CLEAR ERROR
820 002632 005767 175374 TST FSTFLG ;IS THIS A 1ST BYTE?
821 002636 001423 BEQ 1$ ;NO
822 002640 005067 175366 CLR FSTFLG ;CLEAR
823 002644 127777 175432 175442 CMPB @XSPTR,@RCVPTR ;WAS IT THE FLAG WE EXPECTED?
824 002652 001415 BEQ 1$ ;YES-CONTINUE
825 002654 012767 000001 175440 MOV #1,RCVCNT ;OOPS-PRESET FOR EXIT
826 002662 012767 000001 175426 MOV #1,RCPKNM ;IS LAST PACKET, TOO.
827 002670 127777 175420 000002 CMPB @RCVPTR,#RSEND ;WAS IT A PREMATURE END PACK?
828 002676 001003 BNE 1$ ;NO WAY
829 002700 012767 000016 175414 MOV #RSEND,RCVCNT ;YES USE CORRECT COUNT
830 002706 005267 175402 1$ INC RCVPTR ;-->NEXT SLOT
831 002712 005367 175404 DEC RCVCNT ;MORE OF THIS PACKET?
832 002716 001016 BNE 2$ ;YES
833 002720 005367 175372 DEC RCPKMN ;NO-ANY MORE PACKETS?
834 002724 001417 BEQ 3$ ;NO
835 002726 062767 000006 175346 ADD #6,XSPTR ;YES-->NEXT EXPECTED COUNT
836 002734 017767 175342 175360 MOV @XSPTR,RCVCNT ;GET NEW COUNT
837 002742 162767 000002 175332 SUB #2,XSPTR ;-->NEXT EXPECTED FLAG
838 002750 005267 175256 INC FSTFLG ;SAY IT'S 1ST BYTE
839 002754 052777 000100 175252 2$ BIS #EI,@RCR ;OK TO RE-ENABLE INTS.
840 002762 000403 BR RCVRET ;QUIT; QUICKLY
841 002764 005267 175236 3$ INC PKRCVD ;HERE LAST PACKET RCV'D
842 002770 000002 RCVERR: RTI ;*** REV.0- EXIT WITH NO MRSP HANDSHAKE
843
844 002772 000240 RCVRET: NOP
845 002774 005767 175376 TST B$TEST ;*** REV.3- IF BOOT TEST
846 003000 001002 BNE 1$ ;*** REV.3- THEN, NO HANDSHAKE.
847 003002 004767 002034 JSR PC,HNDSHK ;*** REV.0- IF MRSP, XMIT A 'CONTINUE'
848 003006 000002 1$ RTI ;EXIT
849

```

```

851
852 .SBTTL UTILITY SUBROUTINES
853
854 003010 000240 DOBRK: NOP ;SEND INIT TO DEVICE
855 003012 105067 175373 CLRB INITWD+1 ;START FRESH
856 003016 042777 000100 175214 BIC #DI,@XMSR ;KILL SEND
857 003024 042777 000100 175202 BIC #DI,@RCR ;KILL RECEIVE
858 003032 052777 000001 175200 BIS #BIT0,@XMSR ;SET "BREAK"
859 003040 012704 000001 MOV #8,,R4 ;SET COUNT
860 003044
861 003044 104407 000000' 1$ BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR....
862 003050 104407 000000' BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
863 003054 105777 175160 TSTB @XMSR ;READY?
864 003060 100371 BPL 1$ ;NO
865 003062 116777 175320 175152 MOVB @RCDB,INITWD+1 ;SEND NULL
866 003070 005304 DEC R4 ;MORE NULLS TO SEND?
867 003072 001364 BNE 1$ ;YES
868 003074 042777 000001 175136 BIC #BIT0,@XMSR ;NO-CLEAR "BREAK"
869 003102 017704 175130 MOV @RCDB,R4 ;CLEAR POSSIBLE GARBAGE
870 003106 012704 000002 MOV #2,R4 ;NUMBER OF "INTS" TO SEND
871 003112
872 003112 104407 000000' 2$ BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR....
873 003116 104407 000000' BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
874 003122 105777 175112 TSTB @XMSR ;READY TO SEND?
875 003126 100371 BPL 2$ ;NO
876 003130 116777 175254 175104 MOVB INITWD,@XMDB ;YES-SEND
877 003136 005304 DEC R4 ;MORE?
878 003140 001364 BNE 2$ ;YES-CONTINUE
879 003142
880 003142 104407 000000' 3$ BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR....
881 003146 104407 000000' BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
882 003152 105777 175056 TSTB @RCR ;READY TO RECEIVE?
883 003156 100371 BPL 3$ ;NO
884 003160 117767 175052 175223 MOVB @RCDB,INITWD+1 ;YES-GET RESPONSE
885 003166 126727 175217 000020 CMPB INITWD+1,#RSCNT ;IS CONTINUE?
886 003174 001412 BEQ 4$ ;YES
887 003176 004767 001126 CALL ERRSET ;NO-PREPARE EXTENDED ERROR INFO.
888 003202 012767 000034 174676 MOV #34,ERRTYP ;CODE FOR "CAN NOT INITIALIZE"
889
890 003210 104405 000000' 000460' HRDRS,BEGIN,ERRTBL ;CAN'T INIT
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```

```

897                                     ;MODULE TO INTERPRET ANY RESPONSE FROM DEVICE.
898
899 003250 000240                                     CHKANS: NOP                                     ;EVALUATE THE TOTAL RESPONSE FROM UNIT:
900 003252 005067 175042                               CLR                                     ;# OF PACKS INTO TRANSACTION=0.
901 003256 012700 006156'                               MOV #RCVBUF,R0                               ;GET BUFFER ADDR.
902 003262 016702 175040                               MOV XSPKMM,R2                               ;AND # OF PACKETS EXPECTED
903 003266 012703 000330'                               MOV #XSLG,R3                               ;ADDRESS OF 1ST EXPECTED FLAG
904 003272 010301                                     MOV R3,R1                                   ;COPY TO R1
905 003274 062701 000002                               ADD #2,R1                                   ;NOW R1-->X$BCNT FOR 1ST PACKET
906 003300 010067 174774 18:                               MOV R0,PKPTR                               ;HERE R0-->TOP OF PACKET
907 003304 005267 175010                               INC PKCHKD                               ;ONE MORE PACK INTO RESPONSE
908 003310 011167 175010                               MOV @R1,RCBCNT                             ;COPY COUNT
909 003314 004767 001010                               CALL ERRSET                               ;PREP FOR ERROR
910 003320 121013                               CMPB @R0,@R3                               ;1ST BYTE=EXPECTED?
911 003322 001037                               BNE 5$                                     ;UH OH...BRANCH
912 003324 121027 000020                               CMPB @R0,@R$CONT                          ;OK, IS IT 1 BYTE?
913 003330 001500                               BEQ 7$                                     ;YES...ONTO NEXT PACK
914
915 003332 016704 174766                               MOV RCBCNT,R4                               ;NO, SO > 1 BYTE (NEVER EXPECT INIT!)
916 003336 005744                               TST -(R4)                                  ;SO THEN USE EXPECTED COUNT
917 003340 004767 001046                               CALL CKCKSM                               ;ADJUST FROM RECIEVE COUNT TO CHECKSUM CNT
918 003344 103012                               BCC 2$                                     ;CHECK THE CHECKSUM
919 003346 104403 000000' 005532'                     MSGNS,BEGIN,CHKSMR                         ;NO CARRY...NO INCORRECT
920                                     ;*****
921 003354 104405 000000' 000460'                     HRDRS,BEGIN,ERRTBL                       ;ASCII MESSAGE CALL WITH COMMON HEADER
922                                     ;*****
923 003362 012767 177777 174770                     MOV #-1,ERRER                             ;CHECKSUM ERROR
924 003370 000475                                     BR CHKRET                                  ;ERROR FLAG
925 003372 122710 000002 28:                               CMPB #R$END,(R0)                          ;EXIT
926 003376 001003                                     BNE 3$                                     ;END PAK?
927 003400 004767 000240                               CALL CHKSUC                               ;NO
928 003404 000467 000001 38:                               BR CHKRET                                  ;YES-SEE IF SUCCESS CODE VALID
929 003406 122710 000001                               CMPB #R$DATA,@R0                          ;EXIT
930 003412 001056                                     BNE 0TLERR                               ;DATA PAK?
931 003414 004767 001032                               CALL COMPAR                               ;NO
932 003420 000444                                     BR 7$                                     ;YES-CHECK DATA
933                                     ;ON TO NEXT PACK
934
935 003422 000240 58:                               NOP                                     ;HERE CHECKS UNEXPECTED RESPONSE
936 003424 122710 000004                               CMPB #R$INIT,@R0                          ;INIT?
937 003430 001012                                     BNE 6$                                     ;NO
938 003432 104403 000000' 005544'                     MSGNS,BEGIN,RCINIT                       ;ASCII MESSAGE CALL WITH COMMON HEADER
939 003440 104405 000000' 000460'                     HRDRS,BEGIN,ERRTBL                       ;*****
940                                     ;*****
941 003446 012767 177777 174704                     MOV #-1,ERRER                             ;INIT RECIEVED
942 003454 000443                                     BR CHKRET                                  ;FATAL
943
944 003456 122710 000002 68:                               CMPB #R$END,(R0)                          ;END?
945 003462 001032                                     BNE 0TLERR                               ;NO-OUT TO LUNCH
946
947 003464 012704 000016                               MOV #R$NDSZ,R4                             ;YES-TOTAL SIZE MINUS
948 003470 005744                                     TST -(R4)                                  ;TWO (THE CHKSUM)
949 003472 004767 000714                               CALL CKCKSM                               ;CHECK IT
950
951 003476 103403                                     BCS 10$                                   ;YES-CHECK DATA
952 003500 004767 000140                               CALL CHKSUC                               ;OK,NOW TEST SUC. CODE

```

```

950 003504 000427                                     BR CHKRET                                  ;EXIT
951
952 003506 108:                               003506 104403 000000' 005532'
953 003514 104405 000000' 000460'                     MSGNS,BEGIN,CHKSMR                         ;ASCII MESSAGE CALL WITH COMMON HEADER
954                                     ;*****
955 003522 012767 177777 174630                     HRDRS,BEGIN,ERRTBL                       ;CHECKSUM ERROR
956                                     ;*****
957 003530 000415                               MOV #-1,ERRER                             ;FATAL
958 003532 005302                                     BR CHKRET                                  ;EXIT
959 003534 001413 78:                               DEC R2                                     ;NO. OF PACKETS LEFT TO CHECK
960 003536 066700 174562                               BEQ CHKRET                                  ;ALL DONE
961 003542 022121                               ADD RCBCNT,R0                             ;POINT TO NEXT PACKET
962 003544 022323                               CMP (R1)+,(R1)+                           ;POINT TO NEXT EXPECTED COUNT
963 003546 000654                               CMP (R3)+,(R3)+                           ;AND EXPECTED FLAG
964 003550 012767 177777 174602                               BR 1$                                     ;TRY ANOTHER,THEY'RE SMALL
965 003556 052767 000010 0TLERR:                               MOV #-1,ERRER                             ;HERE=STRANGE RESPONSE
966 003564 005767 174572                               BIS #BIT3,SYSTAT                          ;SET O.T.L. FLAG
967 003570 001407                               CHKRET: TST ERRWD                          ;DLV ERROR?
968 003572 104403 000000' 005572'                     BEQ 9$                                     ;NO-BRANCH
969                                     MSGNS,BEGIN,DLERR                         ;ASCII MESSAGE CALL WITH COMMON HEADER
970                                     ;*****
971 003600 104405 000000' 000460'                     HRDRS,BEGIN,ERRTBL                       ;DLV ERROR
972                                     ;*****
973 003606 000415                               BR 10$                                    ;EXIT
974
975 003610 032767 000010 174430 98:                               BIT #BIT3,SYSTAT                          ;WAS THERE 0TL ERROR?
976 003616 001411                               BEQ 10$                                    ;NO-EXIT
977 003620 042767 000010 174420                               BIC #BIT3,SYSTAT                          ;YES-RESET AND PRINT:
978 003626 104403 000000' 005556'                     MSGNS,BEGIN,0TL ;ASCII MESSAGE CALL WITH COMMON HEADER
979                                     ;*****
980 003634 104405 000000' 000460'                     HRDRS,BEGIN,ERRTBL                       ;UNKNOWN FLAG BYTE
981                                     ;*****
982 003642 000207 108:                               RTS PC

```

```
980
981
982          ;MODULE TO INTERPRET THE SUCCESS CODE OF AN END PACKET
983 003644 000240          CHKSUC: NOP          ;SEE IF VALID SUCCESS CODE:
984 003646 010067 174426  MOV          R0,PKPTR    ;COPY TOP OF PACKET POINT
985 003652 116067 000003 174416  MOVB      3(R0),SUCCS  ;GET SUCCESS CODE
986 003660 122760 000000 000003  CMPB      #ESOK,3(R0) ;GOOD SUCCESS?
987 003666 001561          BEQ          12$      ;YES-EXIT
988 003670 004767 000434  CALL          ERRSET    ;POTENTIAL HARD ERROR PREP.
989
990 003674 122760 000001 000003  CMPB      #ESTRY,3(R0) ;OK BUT RETRIES?
991 003702 001007          BNE          20$      ;NO
992 003704 104403 000000' 005630' MSGNS$,BEGIN,RTRY  ;ASCII MESSAGE CALL WITH COMMON HEADER
993          003712 104406 000000' 000460' ;*****
          SOFERS$,BEGIN,ERRTBL ;RETRIES OCCURRED
          ;*****
994 003720 000544          BR          12$      ;YES-EXIT
995 003722 122760 177737 000003 20$: CMPB      #ENOMO,3(R0) ;NO MOTOR?
996 003730 001010          BNE          1$      ;NO...
997 003732 104403 000000' 005656' MSGNS$,BEGIN,NOMOT  ;ASCII MESSAGE CALL WITH COMMON HEADER
998          003740 104405 000000' 000460' ;*****
          HRDRS$,BEGIN,ERRTBL ;"NO MOTOR" ERROR
          ;*****
999 003746 004767 175006  CALL          DROP    ;FATAL-COULD UNWIND CASSETTE
1000 003752 122760 177757 000003 1$: CMPB      #ESCK,3(R0) ;CHECKSUM ERROR?
1001 003760 001007          BNE          13$      ;NO
1002 003762 104403 000000' 005606' MSGNS$,BEGIN,DACHK  ;ASCII MESSAGE CALL WITH COMMON HEADER
1003          003770 104405 000000' 000460' ;*****
          HRDRS$,BEGIN,ERRTBL ;READ RETRIES FAILED
          ;*****
1004 003776 000515          BR          12$      ;YES-EXIT
1005 004000 032767 000002 174240 13$: BIT          #BIT1,SYSTAT ;SELF TEST?
1006 004006 001413          BEQ          3$      ;NO
1007
1008 004010 105760 000003  TSTB          3(R0)      ;NEGATIVE IF FAIL
1009 004014 100106          BPL          12$      ;OK
1010 004016 104403 000000' 005600' MSGNS$,BEGIN,SLFOTL  ;ASCII MESSAGE CALL WITH COMMON HEADER
1011          004024 104405 000000' 000460' ;*****
          HRDRS$,BEGIN,ERRTBL ;SELF TEST FAIL
          ;*****
1012 004032 004767 174722  CALL          DROP    ;FATAL UNEQUIVICALLY
1013
1014 004036 122760 177740 000003 3$: CMPB      #ESCK,3(R0) ;SEEK ERROR?
1015 004044 001007          BNE          4$      ;NO
1016 004046 012767 000051 174032  MOV          #51,ERRTYP ;YES.
1017          004054 104405 000000' 000460' ;*****
          HRDRS$,BEGIN,ERRTBL ;SEEK ERROR
          ;*****
1018 004062 000460          BR          10$      ;EXIT-ERROR
1019 004064 122760 177767 000003 4$: CMPB      #ENCRT,3(R0) ;NO CART?
1020 004072 001007          BNE          5$      ;NO
1021 004074 104403 000000' 005472' MSGNS$,BEGIN,NOCART  ;ASCII MESSAGE CALL WITH COMMON HEADER
1022          004102 104405 000000' 000460' ;*****
          HRDRS$,BEGIN,ERRTBL ;NO MEDIA
          ;*****
1023 004110 000445          BR          10$      ;EXIT -ERROR
1024
```

```
1025 004112 122760 177720 000003 5$: CMPB      #ESCMD,3(R0) ;NO UNDERSTAND HOST?
1026 004120 001436          BEQ          11$      ;YES
1027 004122 122760 177770 000003 6$: CMPB      #ENONX,3(R0) ;NON EXISTENT UNIT
1028 004130 001432          BEQ          11$      ;YES
1029 004132 122760 177765 000003 7$: CMPB      #ESWLOC,3(R0) ;WRITE LOCKED
1030 004140 001007          BNE          8$      ;NO
1031 004142 104403 000000' 005504' MSGNS$,BEGIN,WRPRO  ;ASCII MESSAGE CALL WITH COMMON HEADER
1032          004150 104405 000000' 000460' ;*****
          HRDRS$,BEGIN,ERRTBL ;WRITE LOCKED
          ;*****
1033 004156 000422          BR          10$      ;EXIT-ERROR
1034 004160 122760 177776 000003 8$: CMPB      #ESPART,3(R0) ;PARTIAL OP?
1035 004166 001413          BEQ          11$      ;YES
1036 004170 122760 177711 000003 9$: CMPB      #ESREC,3(R0) ;WRONG RECORD
1037 004176 001407          BEQ          11$      ;EXIT
1038 004200 104403 000000' 005516' MSGNS$,BEGIN,SUCOTL  ;ASCII MESSAGE CALL WITH COMMON HEADER
1039          004206 104405 000000' 000460' ;*****
          HRDRS$,BEGIN,ERRTBL ;UNKNOWN SUCCESS CODE
          ;*****
1040 004214 000403          BR          10$      ;EXIT -ERROR
1041 004216          004216 104403 000000' 005642' 11$: MSGNS$,BEGIN,BDSUC  ;ASCII MESSAGE CALL WITH COMMON HEADER
1042 004224 012767 177777 174126 10$: MOV          #-1,ERRER ;OPERATION FATAL
1043 004232          004232 000207          12$: RTS          PC
1044
```

```

1046
1047
1048 ;CHKSUM / FORM THE PACKET CHECKSUM
1049
1050 ;++
1051 ; ENTER:
1052 ; R0 POINTS TO PACKET
1053 ; R1 = BYTES IN PACKET
1054 ;
1055 ; EXIT:
1056 ; R1 = CHECKSUM
1057 ; R0 POINTS TO ADDRESS FOR PACKET CHECKSUM
1058 ;--
1059 004234 010346 CHKSUM:
1060 004236 010246 MOV R3,-(SP)
1061 004240 042767 000001 174000 MOV R2,-(SP)
1062 004246 032701 000001 BIC #BIT0,SYSTAT ;"CHECKSUM IS ODD" BIT
1063 004252 001403 BEQ 1$ ;AN ODD # OF BYTES?
1064 004254 052767 000001 173764 BIS #BIT0,SYSTAT ;NO
;JUST USED AS TEMP STORAGE
1065
1066 004262 006001 1$: RDR R1 ;/2 FOR WORDS
1067
1068 004264 005003 2$: CLR R3
1069
1070 004266 062003 3$: ADD (R0)+,R3 ;FORM SUM
1071 004270 005503 ADC R3 ;WITH CARRY
1072 004272 005301 DEC R1 ;ONE LESS WORD
1073 004274 001374 BNE 3$ ;BR IF MORE LEFT
1074
1075 004276 032767 000001 173742 BIT #BIT0,SYSTAT ;WAS IT ODD
1076 004304 001405 BEQ 4$ ;NO
1077 004306 112002 MOVB (R0)+,R2 ;YES-COPY TO R2
1078 004310 042702 177400 BIC #177400,R2 ;SIGN UN-EXTEND
1079 004314 060203 ADD R2,R3 ;ADD LAST BYTE
1080 004316 005503 ADC R3 ;AND CARRY JUST IN CASE
1081
1082 004320 010301 4$: MOV R3,R1 ;RETURN IT IN CORRECT PLACE
1083 004322 012602 MOV (SP)+,R2
1084 004324 012603 MOV (SP)+,R3
1085 004326 000207 RTS PC
1086

```

```

1088
1089 ;MODULE TO FILL ERRTABL WITH EXTENDED ERROR INFORMATION
1090
1091 004330 016767 173700 173542 ERRSET: MOV RCSR,CSRA ;COPY CSR
1092 004336 017767 173672 173536 MOV @RCSR,ACSR ;AND CONTENTS
1093 004344 016767 173726 173532 MOV SUCCS,ASTAT ;AND SUCCESS IF APPLICABLE
1094 004352 017767 173724 173776 MOV @X6PTR,ERRFLG ;COPY EXPECTED FLAGBYTE
1095 004360 017767 173714 173776 MOV @PKPTR,ATPAK ;1ST AND 2ND BYTES OF ACTUAL PACK
1096 004366 062767 000002 173704 ADD #2,PKPTR ;POINTER INC.
1097 004374 017767 173700 173764 MOV @PKPTR,ATPAK2 ;COPY 3RD AND 4TH BYTE
1098 004402 162767 000002 173670 SUB #2,PKPTR ;CORRECT POINTER
1099 004410 000207 RTS PC

```

```

1101                ;MODULE TO CHECK THE CHKSUM OF A RECEIVED PACKET
1102
1103                ;++
1104                ; ENTER:
1105                ; R4 = PACKET BYTE COUNT
1106                ; R0 POINTS TO TOP OF PACKET
1107                ;
1108                ; CALLS: "CHKSUM"
1109                ; TO FORM CHECKSUM.
1110                ;
1111                ; EXIT:
1112                ; CARRY SET - ERROR
1113                ; RESET - NO ERROR
1114                ;--
1115
1116 004412          CKCKSM:
1117 004412 010146      MOV     R1, -(SP)
1118 004416 010046      MOV     R0, -(SP)
1119 004420 004767 177610 CALL    CHKSUM          ;COPY BYTE COUNT TO CORRECT
1120                                     ;REGISTER FOR CHKSUM AND CALL
1121
1122                ;HERE R0 --> XMITTED CHKSUM, R1=CHKSUM CALC'D
1123
1124 004424 122001      CMPB    (R0)+,R1          ;LOWER ORDER CHECK
1125 004426 001005      BNE     2$                ;NE COMPAREZ PAS!
1126
1127 004430 000301      SWAB    R1                ;OK-NOW FOR OTHER HALF:
1128
1129 004432 122001      CMPB    (R0)+,R1          ;HIGH ORDER CHECK
1130 004434 001002      BNE     2$                ;WRONG-BRANCH
1131 004436 000241      CLC                     ;NO ERROR FLAG
1132 004440 000401      BR     3$                ;EXIT-NO ERROR
1133 004442 000261      SEC                     ;ERROR FLAG
1134 004444          2$:
1135 004446 012600      MOV     (SP)+,R0
1136 004448 012601      MOV     (SP)+,R1
1137 004450 000207      RTS     PC

```

```

1138
1139                ;COMPAR / COMPARE PATTERN WRITTEN TO DATA RECV'D
1140
1141 004452          COMPAR:
1142 004452 010046      MOV     R0, -(SP)
1143 004454 010046      MOV     R4, -(SP)
1144 004456 010146      MOV     R1, -(SP)
1145 004460 005067 173704 CLR     BDBYTS          ;NONE WRONG YET
1146 004462 010067 173610 MOV     R0, PKPTR          ;R0->TOP OF PACKET
1147 004464 016704 173604 MOV     PKPTR, R4          ;GET TOP OF PACKET
1148 004466 005204      INC     R4                ;LOCATE COUNT
1149 004468 111401      MOVB    @R4, R1          ;GET IT
1150 004470 042701 177400 BIC     #177400, R1      ;SIGN UN-EXTEND
1151
1152 004504 005204      INC     R4                ;MUST TEST BYTE-WISE...
1153 004506 126724 173562 1$: CMPB    PATRN, (R4)+    ;-->FIRST DATA BYTE
1154 004512 001413      BEQ     2$                ;WAS WHAT WAS WRITTEN?
1155 004514 005767 173650 TST     BDBYTS          ;OK!
1156 004516 001006      BNE     3$                ;WRONG-FIRST ERROR?
1157 004518 005304      DEC     R4                ;NO, DETOUR 1ST ADDRESS SETUP
1158 004520 111467 173360 MOVB    @R4, AWAS          ;UN-AUTO INCREMENT
1159 004522 010467 173350 MOV     R4, WASADR          ;SAVE BAD BYTE
1160 004524 005204      INC     R4                ;AND SAVE ADDRESS
1161 004526 005267 173626 INC     BDBYTS          ;CORRECT POINTER
1162 004528 005301      INC     R1                ;ONE MORE BAD BYTE
1163 004530 001360      DEC     R1                ;MORE LEFT?
1164 004532 001360      BNE     1$                ;YES
1165 004534 005767 173616 TST     BDBYTS          ;ANY WRONG?
1166 004536 001422      BEQ     4$                ;NO. EXIT
1167 004538 016767 173512 173320 MOV     REC, ACSR          ;YES-GET RECORD #
1168 004540 012767 000274' 173312 MOV     #PATRN, SBADR          ;ADDR SHOULD BE
1169 004542 016767 173500 173310 MOV     PATRN, ASB          ;DATA SHOULD BE
1170
1171 004576 104404 000000'          ;*****
1172                                     ;DATERS, BEGIN          ;DATA ERROR!!!
1173                                     ;*****
1174                                     ;*****
1175                                     ;CONVERT BDBYTS TO ASCII AND
1176                                     ;STORE AT A$BDBT
1177
1178 004602 104421 000000' 000370' BTOD$, BEGIN, BDBYTS, A$BDBT
1179 004610 000262'
1180
1181 004612 104403 000000' 005620' ;*****
1182 MSGN$, BEGIN, BADBY          ;ASCII MESSAGE CALL WITH COMMON HEADER
1183
1184 004620 012601      MOV     (SP)+, R1
1185 004622 012604      MOV     (SP)+, R4
1186 004624 012600      MOV     (SP)+, R0
1187 004626 000207      RTS     PC
1188
1189                ;-----
1190
1191
1192

```

```

1179 .SBTTL BOOT / FUNCTIONAL BOOT TEST
1180
1181 ;*** REV.3- THIS TEST WILL EXECUTE IF BIT0 OF SWITCH REGISTER 4 IS SET TO
1182 ;*** REV.3- A ONE. THIS MODULE PERFORMS A TU58 BOOT. IT IS INTENDED TO VERIFY
1183 ;*** REV.3- THE BOOT COMMAND IN EITHER 'RSP' OR 'MRSP' PROTOCOLS. IT IS NOT
1184 ;*** REV.3- INTENDED TO BE A BOOT, BUT TO SIMPLY VERIFY THE BOOT COMMAND.
1185 ;*** REV.3- THE SUBROUTINE TRANSMITS WITH INTERRUPTS DISABLED AND RECIEVES
1186 ;*** REV.3- WITH INTERRUPTS ENABLED.
1187
1188 004630 000240 BOOT: NOP
1189 004632 005067 173540 CLR B$TEST ;RESET BOOT TEST FLAG
1190 004636 032767 000001 173160 BIT #BIT0,SR4 ;IF BIT ZERO OF SR4 IS RESET
1191 004644 001461 BEQ NOBOOT ;THEN, SKIP BOOT TEST
1192 004646 005267 173524 INC B$TEST ;ELSE, SET BOOT TEST FLAG
1193 ;1) ELSE, SET UP RECIEVER
1194 004652 005067 173350 CLR PKRCVD ;RESET RECIEVE PACK FLAG
1195 004656 012767 001000 173436 MOV #512,,RCVCNT ;EXPECT TO RECIEVE 512. BYTES
1196 004664 012767 000001 173424 MOV #1,RCPKNM ;IN ONE PACKET
1197 004672 012767 006156 173414 MOV #RCVBUF,RCVPTR ;AND PUT THE DATA HERE.
1198 004700 052777 000100 173326 BIS #E1,RCRSR ;AND ENABLE RECIEVER INTERRUPTS
1199
1200 ;2) TRANSFER THE BOOT COMMAND
1201 004706 016700 173374 MOV TBLPTR,R0 ;GET DRIVE NUMBER
1202 004712 116000 000001 MOV B 1(R0),R0
1203 004716 042700 177776 BIC #177776,R0
1204 004722 016701 173306 MOV RCSR,R1 ;R1= DEVICE ADDRESS
1205 004726 012777 000001 173304 MOV #1,@XMSR ;SET BREAK
1206 004734 005002 CLR R2 ;SEND NULLS
1207 004736 004767 000054 JSR PC,SEND8 ;DELAY 7 CHARACTERS
1208 004742 005077 173272 CLR @XMSR ;CLEAR BREAK
1209 004746 005777 173264 TST @RCDB ;DUMP RCV REG
1210 004752 012702 MOV (PC)+,R2 ;GET INIT, BOOT FLAGS
1211 004754 004 010 .BYTE RSINIT,R$BOOT
1212 004756 004767 000040 JSR PC,SEND2 ;SEND INIT, BOOT FLAGS
1213 004762 010002 MOV R0,R2
1214 004764 004767 000034 JSR PC,SEND1 ;SEND UNIT NUMBER
1215
1216 004770 000240 1$: NOP ;3) RECIEVE BOOTED DATA
1217 004772 104407 000000' BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR....
1218 004776 104407 000000' BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
1219 005002 005767 173220 TST PKRCVD ;RECIEVE COMPLETED ?
1220 005006 001770 BEQ 1$ ;CYCLE UNTIL DONE.
1221 005010 005067 173362 NOBOOT: CLR B$TEST ;RESET BOOT TEST FLAG
1222 005014 000207 RTS ;COMPLETED
1223
1224 005016 004717 SEND8: JSR PC,@PC ;SEND 8 CHAR
1225 005020 004717 JSR PC,@PC ;SEND 4
1226 005022 004717 JSR PC,@PC ;SEND 2
1227 005024 105777 173210 SEND1: TSTB @XMSR ;SEND 1
1228 005030 100375 BPL SEND1
1229 005032 110277 173204 MOV B R2,@XMDB ;SEND CHARACTER
1230 005036 000302 SWAB R2
1231 005040 000207 RTS PC

```

```

1233 ;*** REV.0- THIS SUBROUTINE OUTPUT A 'MRSP' CONTINUE TO THE TU58
1234 ;*** REV.0- CONTROLLER. IT XMIT'S A CONTINUE ONLY IF RUNNING IN MRSP.
1235
1236 005042 005767 173324 HNDSHK: TST MRSP ;IF RADIAL SERIAL PROTOCOL
1237 005046 001406 BEQ 64$ ;THEN, NO HANDSHAKE REQUIRED
1238 005050 105777 173164 63$: TSTB @XMSR ;ELSE, WHEN XMIT READY,
1239 005054 100375 BPL 63$
1240 005056 112777 000020 173156 MOV B #X$CONT,@XMDB ;TRANSMIT AN 'MRSP' CONTINUE
1241 005064 000207 64$: RTS PC
1242
1243
1244

```

```

1246
1247
1248
1249 005066 045 124 125 MES1: .ASCIZ '%TU58 CONTROLLER '
      005071 065 070 040
      005074 103 117 116
      005077 124 122 117
      005102 114 114 105
      005105 122 040 000
1250 005110 040 104 122 MES2: .ASCIZ ' DRIVE '
      005113 111 126 105
      005116 040 000
1251 005120 045 104 114 MES3: .ASCIZ '%DLV ERROR %'
      005123 126 040 105
      005126 122 122 117
      005131 122 040 045
      005134 000
1252 005135 045 123 105 MES4: .ASCIZ '%SELF TEST FAIL%'
      005140 114 106 040
      005143 124 105 123
      005146 124 040 106
      005151 101 111 114
      005154 045 040 000
1253 005157 045 122 105 MES5: .ASCIZ '%READ RETRYS ON OP.%'
      005162 101 104 040
      005165 122 105 124
      005170 122 131 123
      005173 040 117 116
      005176 040 117 120
      005201 056 045 000
1254 005204 045 103 110 MES7: .ASCIZ '%CHECKSUM ERROR%'
      005207 105 103 113
      005212 123 125 115
      005215 040 105 122
      005220 122 117 122
      005223 045 000
1255 005225 045 122 105 MES8: .ASCIZ '%RECEIVING INIT REQUEST%'
      005230 103 105 111
      005233 126 111 116
      005236 107 040 111
      005241 116 111 124
      005244 040 122 105
      005247 121 125 105
      005252 123 124 045
      005255 000
1256 005256 045 122 105 MES9: .ASCIZ '%RETRIES FAILED ON OP.% '
      005261 124 122 111
      005264 105 123 040
      005267 106 101 111
      005272 114 105 104
      005275 040 117 116
      005300 040 117 120
      005303 056 045 040
      005306 000
1257 005307 045 116 117 MES10: .ASCIZ '%NO MOTOR ACTION%'
      005312 040 115 117
      005315 124 117 122
  
```

```

      005320 040 101 103
      005323 124 111 117
      005326 116 045 000
1258 005331 045 116 117 MES11: .ASCIZ '%NO CARTRIDGE%'
      005334 040 103 101
      005337 122 124 122
      005342 111 104 107
      005345 105 045 000
1259 005350 045 127 122 MES12: .ASCIZ '%WRITE PROTECTED%'
      005353 111 124 105
      005356 040 120 122
      005361 117 124 105
      005364 103 124 105
      005367 104 045 000
1260 005372 045 124 117 MES13: .ASCIZ '%TOTAL BAD BYTES = '
      005375 124 101 114
      005400 040 102 101
      005403 104 040 102
      005406 131 124 105
      005411 123 040 075
      005414 040 000
1261 005416 040 123 125 MES18: .ASCIZ ' SUCCESS CODE%'
      005421 103 103 105
      005424 123 123 040
      005427 103 117 104
      005432 105 045 000
1262 005435 040 106 114 MES19: .ASCIZ ' FLAG BYTE%'
      005440 101 107 040
      005443 102 131 124
      005446 105 045 000
1263 005451 045 000 MES20: .ASCIZ '% '
1264 005453 045 125 MES21: .ASCIZ '%UNKNOWN '
      005456 113 116 117
      005461 127 116 040
      005464 000
1265 005465 102 101 104 MES22: .ASCIZ 'BAD '
      005470 040 000
1266
1267
  
```



```

1269
1270      .SBTTL MESSAGE CONSTRUCTS
1271
1272 005472 005066'      NOCART: MES1      ;CARTRIDGE NOT INSERTED
1273 005474 005110'      MES2
1274 005476 000252'      A$DRV
1275 005500 005331'      MES11
1276 005502 177777      -1
1277
1278 005504 005066'      WRPRO: MES1      ;CARTRIDGE WRITE PROTECTED
1279 005506 005110'      MES2
1280 005510 000252'      A$DRV
1281 005512 005350'      MES12
1282 005514 177777      -1
1283
1284 005516 005066'      SUCOTL: MES1     ;COULD NOT INTERPRET SUCCESS CODE
1285 005520 005110'      MES2
1286 005522 000252'      A$DRV
1287 005524 005453'      MES21
1288 005526 005416'      MES18
1289 005530 177777      -1
1290
1291 005532 005066'      CHKSMR: MES1     ;HOST CALCULATED CHECKSUM FOR
1292 005534 005110'      MES2             ;PACKET DOESN'T EQUAL TRANSMITTED ONE.
1293 005536 000252'      A$DRV
1294 005540 005204'      MES7
1295 005542 177777      -1
1296
1297 005544 005066'      RCINIT: MES1     ;HOST RECEIVED "INIT" REQUEST
1298 005546 005110'      MES2             ;
1299 005550 000252'      A$DRV             ;/
1300 005552 005225'      MES8
1301 005554 177777      -1
1302
1303 005556 005066'      OTL: MES1        ;1ST BYTE OF PACKET UNINTERPRETABLE
1304 005560 005110'      MES2
1305 005562 000252'      A$DRV
1306 005564 005453'      MES21
1307 005566 005435'      MES19
1308 005570 177777      -1
1309
1310 005572 005066'      DLERR: MES1     ;OVERRUN OR FRAMING ERROR IN RECEIVE
1311 005574 005120'      MES3
1312 005576 177777      -1
1313
1314 005600 005066'      SLFOTL: MES1    ;DEVICE'S SELF TEST FAILED BUT COULD
1315 005602 005135'      MES4           ;STILL SEND AN INTELLIGENT PACKET (!)
1316 005604 177777      -1
1317
1318 005606 005066'      DACHK: MES1     ;DEVICE RETRIED OPERATION 8 TIMES
1319 005610 005110'      MES2           ;AND FAILED
1320 005612 000252'      A$DRV
1321 005614 005256'      MES9
1322 005616 177777      -1
1323
1324 005620 005372'      BADBY: MES13    ;MESSAGE FOR THE TOTAL INCORRECT
1325 005622 000262'      A$BDBT         ;DATA BYTES PER PACKET.

```

```

1326 005624 005451'      MES20
1327 005626 177777      -1
1328
1329 005630 005066'      RTRY: MES1     ;DEVICE RETRIED OPERATION SUCCESSFULLY
1330 005632 005110'      MES2
1331 005634 000252'      A$DRV
1332 005636 005157'      MES5
1333 005640 177777      -1
1334
1335 005642 005066'      BDSUC: MES1     ;SUCCESS CODE VALID BUT OPERATION FAILED
1336 005644 005110'      MES2
1337 005646 000252'      A$DRV
1338 005650 005465'      MES22
1339 005652 005416'      MES18
1340 005654 177777      -1
1341
1342 005656 005066'      NOMOT: MES1    ;DEVICE COULDN'T GET ANY MEANINGFUL SIGNAL
1343 005660 005110'      MES2           ;FROM TAPE
1344 005662 000252'      A$DRV
1345 005664 005307'      MES10
1346 005666 177777      -1

```

```

1348          .SBTTL  BUFFER AREAS:
1349
1350          005670          TRBUF:  .BLKW  132
1351          006154          .WORD  0
1352
1353          006156          RCVBUF:  .BLKB  4*RSNASZ+R&NDSZ  ;4 DATA PACKS AND END
1354          007214          LSTADR:  .WORD  0
1355          007216          PATCH:  .REPT  20
1356          .NLIST
1357          .WORD  0
1358          .LIST
1359          .ENDR
1360          000001          .END
  
```

```

ACSR      000102R      DECOD      001652R      ICONT      000036R      PRTY1 = 000040      R&SLF= 000007
ADDR      000006R      DECRET     002512R      ICOUNT      000040R      PRTY2 = 000100      R&SWR = 000003
ADDR22= 001000      DECRET1     002456R      IDNUM       000122R      PRTY3 = 000140      R6 = 0000006
ASB       000106R      DI = 000100      INIT        000030R      PRTY4 = 000200      R7 = 0000007
ASTAT     000104R      DLERR     005572R      INITWD      000410R      PRTY5 = 000240      SBADR      000102R
ATPAK     000364R      DGBRK     003010R      INTR        000120R      PRTY6 = 000300      SELFTT     000412R
ATPAK2    000366R      DROP      000760R      LASTOP      000230R      PRTY7 = 000340      SEND1      005024R
AWAS      000110R      DRV       000250R      LSTADR      007214R      PS = 177776      SEND2      005022R
A&BDBT    000262R      DRV0 = 000000      MAP22$= 104416      PSW = 177776      SEND8      005016R
A&DRV     000252R      DRV1 = 000400      MES1        005066R      PUSH = 005746      SLFOU     005600R
BADBY     005620R      DR1OP     000456R      MES10       005307R      PUSH2 = 024646      SDCNT      000312R
BDBYTS    000370R      DTAF1L1= 000001      MES11       005331R      P&DATA     000404R      SDCNT      002532R
BDSUC     005642R      DTAF1L2= 000010      MES12       005350R      P&DAT1= 000001      SDCPTR     000310R
BEGIN     000000R      DTAF1L3= 000011      MES13       005372R      P&DAT4= 000004      SOFCNT     000042R
BIT0 = 000001      DTAF1L4= 000100      MES18       005416R      RAND$ = 104417      SOFERS= 104406
BIT1 = 000002      DVIDX     000244R      MES19       005435R      RANNUM     000054R      SOFPAS     000046R
BIT10 = 002000      DVID1     000014R      MES2        005110R      RCBCNT     000324R      SPOINT     000032R
BIT11 = 004000      D&DATA= 000000      MES20       005451R      RCDB       000236R      SPSIZ = 000040
BIT12 = 010000      D&FLAG   000402R      MES21       005453R      RCINIT     005544R      SR1 = 000016R
BIT13 = 020000      EI = 000100      MES22       005465R      RCPKNM     000316R      SR2 = 000020R
BIT14 = 040000      ENDIT$= 104413      MES3        005120R      RCSR       000234R      SR3 = 000022R
BIT15 = 100000      ENDS = 104410      MES4        005135R      RCVBUF     006156R      SR4 = 000024R
BIT2 = 000004      ERRER     000360R      MES5        005157R      RCVCNT     000322R      START      000522R
BIT3 = 000010      ERRFLG    000356R      MES7        005204R      RCVERR     002770R      STAT       000026R
BIT4 = 000020      ERROP     000354R      MES8        005225R      RCVHND     002566R      SUCCS      000276R
BIT5 = 000040      ERRSET    004330R      MES9        005256R      RCVOK      002626R      SUCOTL     005516R
BIT6 = 000100      ERRTRL    000460R      MODNAM      000000R      RCVPTR     000314R      SVR0 = 000062R
BIT7 = 000200      ERRTR2    000500R      MODSP       000224R      RCVRET     002772R      SVR1 = 000064R
BIT8 = 000400      ERRTRYP   000106R      MSG$ = 104403      REC        000272R      SVR2 = 000066R
BIT9 = 001000      ERRWD     000362H      MSG$ = 104402      R&STRT     000610R      SVR3 = 000070R
BLKSIZ    000400R      EXIT$ = 104400      MSG$ = 104401      RES1       000056R      SVR4 = 000072R
BOOT      004630R      E&ABO = 177720      MSG$ = 104400      RES2       000060R      SVR5 = 000074R
BREAK$= 104407      E&CK$ = 177757      M&RSP       000372R      R&STRT     000112R      SVR6 = 000076R
BRKWD     000406R      E&CK$M= 177757      NOBOOT      005010R      R&STRT2    000666R      SYS&CNT    000052R
BR1       000012R      E&CMD = 177720      NOCART      005472R      RTRY       005630R      SYSINI     001170R
BR2       000013R      E&NCRT= 177767      NOHAT       005656R      R&BOOT= 000010      SYSSET     001422R
BTOD$ = 104421      E&NOMO= 177737      NOPACK      002520R      R&CMND= 000002      SYSTAT     000246R
B&SIZE    000374R      E&NONX= 177770      NULL = 000000      R&CONT= 000020      TBLPTR     000306R
B&TEST    000376R      E&OK = 000000      OPTBL0      000424R      R&NASZ= 000204      TBLTOP     000304R
CDATA$= 104412      E&PART= 177776      OPTBL1      000440R      R&DATA= 000001      TRBUF      005670R
CHKANS     003250R      E&RD = 177757      OTL         005556R      R&DNSZ= 000222      TRPDFD= 000022
CHKRET     003564R      E&REC = 177711      OTLERR      003550R      R&END = 000002      VECTOR     000010R
CHKSMR     005532R      E&SK = 177740      OTOAS = 104420      R&INIT= 000004      WASADR     000104R
CHKSUM     003644R      E&SLF = 177777      PASCNT      000034R      R&MSIZ= 000012      WDFR       000116R
CKCKSM     004234R      E&STRY = 000001      PATCH       007216R      R&NDSZ= 000016      WDT0       000114R
CKCKSM     004412R      E&WLOC= 177765      PATRN       000274R      R&SNSZ= 000016      WRPRD      005504R
CLRBUF     003224R      E&WR = 177757      PIRQS = 000004      R&XOFF= 000023      XFLAG      000005R
CMNDFL= 100000      FSTFLG    000232R      PKCHKD      000320R      R&XON = 000020      XMDB       000242R
COMPAR     004452R      GETPAS= 104415      PKPIR       000300R      R&SBOO= 000010      XMSR       000240R
CONFIG     000056R      GWNTAK     001000R      PKRCVD      000226R      R&SEND= 000100      X&CNT      000332R
CSRA       000100R      GW&BFS= 104414      PKSNT       000224R      R&SNIT= 000001      X&CONT= 000020
DACHK     005606R      HND$HK     005042R      POPSP2= 005726      R&SNOP= 000000      X&FLG      000330R
DATCK$= 104411      HRDCNT     000044R      POPSP2= 022626      R&SRD = 000002      X&PKNM     000326R
DATER$= 104404      HRDR$= 104405      PRY = 000000      R&S&SEK= 000005      X&PTR      000302R
DECCMD     002072R      HRDPAS     000050R      PRTY0 = 000000
  
```

. ABS. 000000 000

007256 001
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 6204 WORDS (25 PAGES)
DYNAMIC MEMORY: 19446 WORDS (74 PAGES)
ELAPSED TIME: 00:02:44
CXTUAB.OBJ,CXTUAB.LST/CR/-SP=DDXCOM,MAC,CXTUAB,MAC

CXTUAB		CREATED BY		MACRO		ON 2-MAR-82 AT 08:31		PAGE 1											
SYMBOL		CROSS REFERENCE						CREF											
SYMBOL	VALUE	REFERENCES																	
ACSR	000102 R	#4-274	*16-1092	*18-1165															
ADDR	000006 R	#4-274	9-592																
ADDR22	= 001000	#4-274																	
ASB	000106 R	#4-274	*18-1167																
ASTAT	000104 R	#4-274	*16-1093																
ATPAK	000364 R	#5-340	6-485	*16-1095															
ATPAK2	000366 R	#5-341	6-486	*16-1097															
AWAS	000110 R	#4-274	*18-1157																
ABBOBT	000262 R	#5-314	18-1169	22-1325															
ABDRV	000252 R	#5-310	10-744	22-1274	22-1280	22-1286	22-1293	22-1299	22-1305	22-1320									
		22-1331	22-1337	22-1344															
BADBY	005620 R	18-1170	#22-1324																
BDBYTS	000370 R	#5-342	*9-675	*18-1144	18-1154	*18-1160	18-1163	18-1169											
BDSUC	005642 R	14-1041	#22-1335																
BEGIN	000000 R	#4-274	7-517	7-538	7-543	8-562	8-562	8-576	8-576	8-576	10-744								
		12-860	12-860	12-869	12-869	12-875	12-875	12-883	13-919	13-920									
		13-936	13-937	13-952	13-953	13-968	13-969	13-975	13-976	14-992									
		14-993	14-997	14-998	14-1002	14-1003	14-1010	14-1011	14-1017	14-1021									
		14-1022	14-1031	14-1032	14-1038	14-1039	14-1041	18-1168	18-1169	18-1170									
		19-1217	19-1217																
BIT0	= 000001	#4-274	#5-369	7-499	9-607	9-640	9-665	12-858	12-866	15-1061									
		15-1062	15-1064	15-1075	19-1190														
BIT1	= 000002	#4-274	#5-370	7-503	9-613	10-760	10-763	14-1005											
BIT10	= 002000	#4-274	#5-379																
BIT11	= 004000	#4-274	#5-380																
BIT12	= 010000	#4-274	#5-381																
BIT13	= 020000	#4-274	#5-382																
BIT14	= 040000	#4-274	#5-383	9-602	9-634	10-689													
BIT15	= 100000	#4-274	#5-384																
BIT2	= 000004	#4-274	#5-371																
BIT3	= 000010	#4-274	#5-372	10-749	13-965	13-972	13-974												
BIT4	= 000020	#4-274	#5-373																
BIT5	= 000040	#4-274	#5-374																
BIT6	= 000100	#4-274	#5-375																
BIT7	= 000200	#4-274	#5-376	10-735															
BIT8	= 000400	#4-274	#5-377	10-740															
BIT9	= 001000	#4-274	#5-378																
BLKSIZ	000400 R	#5-346	*9-643	*9-654	10-751														
BOOT	004630 R	10-724	#19-1188																
BREAK8	= 104407	#4-274	8-562	8-576	8-576	12-860	12-860	12-869	12-869										
		12-875	12-875	19-1217	19-1217														
BRKWD	000406 R	#5-352	12-863																
BR1	000012 R	#4-274	9-588	9-591															
BR2	000013 R	#4-274																	
BTODS	= 104421	#4-274	10-744	18-1169															
BSSIZE	000374 R	#5-344	7-510	*9-639	*9-642	10-733													
BSTEST	000376 R	#5-345	11-845	*19-1189	*19-1192	*19-1220													
CODATS	= 104412	#4-274																	
CHKANS	003250 R	7-525	#13-899																
CHKRET	003564 R	13-922	13-926	13-939	13-950	13-955	13-958	#13-966											
CHKSMR	005532 R	13-919	13-952	#22-1291															
CHKSUC	003644 R	13-925	13-949	#14-983															

SEQ 0038

SEQ 0039

SEQ 0040

CXTUAB CREATED BY MACRO ON 2-MAR-82 AT 08:31 PAGE 4
 SYMBOL CROSS REFERENCE CREF
 SEQ 0041

SYMBOL	VALUE	REFERENCES
M#RSP	000372 R	#5-343 *9-664 *9-667 10-747 20-1236
NOBOOT	005010 R	19-1191 #19-1220
NO CART	005472 R	14-1021 #22-1272
NOMOT	005656 R	14-997 #22-1342
NOPACK	002520 R	10-691 10-725 #10-795
NULL	= 000000	#4-274
OPEN	= 000000	4-274 4-274 4-274 4-274 4-274 4-274 4-274 4-274 4-274 4-274
OPTBL0	000424 R	#6-451 9-625 9-637
OPTBL1	000440 R	#6-457 *9-628 9-630 *9-631 9-638
OTL	005556 R	13-975 #22-1303
OTLERR	003550 R	13-928 13-942 #13-964
OTOA#	= 104420	#4-274
PASCMT	000034 R	#4-274
PATCH	007216 R	#23-1355
PATRN	000274 R	#5-319 6-473 *7-518 *7-519 10-701 18-1152 18-1166 18-1167
PIRQ#	= 000004	#4-274
PKCHKD	000320 R	#5-329 6-484 *13-900 *13-907
PKPTR	000300 R	#5-321 6-476 *13-906 *14-984 16-1095 *16-1096 16-1097 *16-1098 *18-1145
PKRCVD	000226 R	18-1146
PKSNT	000224 R	#5-293 *8-552 8-577 *11-816 *11-841 *19-1194 19-1218
POPSP	= 005726	#5-292 *8-551 8-563 *11-807
POPSP2	= 022626	#4-274
PRTY	= 000000	#4-274
PRTY0	= 000000	4-274 #4-274
PRTY1	= 000040	#4-274
PRTY2	= 000100	#4-274
PRTY3	= 000140	#4-274
PRTY4	= 000200	4-274 #4-274
PRTY5	= 000240	#4-274
PRTY6	= 000300	#4-274
PRTY7	= 000340	#4-274
PS	= 177776	#4-274
PSW	= 177776	#4-274
PUSH	= 005746	#4-274
PUSH2	= 024646	#4-274
P#DATA	000404 R	#5-348 *9-644 *9-655 10-779 10-783
P#DAT1	= 000001	#5-409 9-644
P#DATA4	= 000004	#5-410 9-655
RAND#	= 104417	#4-274 7-517
RANNUM	000054 R	#4-274 7-518
RCBCNT	000324 R	#5-331 6-480 *13-908 13-915 13-959
RCDB	000236 R	#5-297 *9-595 11-812 12-867 12-878 19-1209
RCINIT	005544 R	13-936 #22-1297
RCPKNM	000316 R	#5-328 *8-555 *11-826 *11-833 *19-1196
RCSR	000234 R	#5-296 7-541 *9-593 11-811 11-839 12-857 12-876 16-1091
RCVBUF	006156 R	16-1092 19-1198 19-1204
RCVCNT	000322 R	8-557 12-890 13-901 19-1197 #23-1353
RCVERR	002770 R	#5-330 6-479 *8-554 *11-825 *11-829 *11-831 *11-836 *19-1195
		11-818 #11-842

CXTUAB CREATED BY MACRO ON 2-MAR-82 AT 08:31 PAGE 5
 SYMBOL CROSS REFERENCE CREF
 SEQ 0042

SYMBOL	VALUE	REFERENCES
RCVHND	002566 R	9-587 #11-811
RCVOK	002626 R	11-815 #11-819
RCVPTR	000314 R	#5-327 6-474 *8-557 11-813 11-823 11-827 *11-830 *19-1197
RCVRET	002772 R	11-840 #11-844
REC	000272 R	#5-318 6-472 *7-505 *7-507 *7-512 *7-514 10-753 18-1165
RESTRT	000610 R	4-274 #7-506 7-539
RES1	000056 R	#4-274
RES2	000060 R	#4-274
RSTRT	000112 R	#4-274
RSTRT2	000666 R	7-523
RTRY	005630 R	14-992 #22-1329
R#BOOT	= 000010	#5-397 6-447 10-722 19-1211
R#CMND	= 000002	#5-392 5-400 10-728
R#CNT	= 000020	#5-393 10-715 10-770 12-879 13-912
R#DASZ	= 000204	#5-406 5-407 23-1353
R#DATA	= 000001	#5-399 10-697 13-927
R#DSZ	= 000222	#5-407
R#END	= 000002	#5-400 10-711 10-764 10-781 11-827 13-923 13-941
R#INIT	= 000004	5-353 #5-398 13-934 19-1211
R#MSIZ	= 000012	#5-405 5-408 10-729 10-754
R#NSZ	= 000016	#5-404 5-407 10-712 10-765 10-782 11-829 13-944 23-1353
R#NSZ	= 000016	#5-408 10-756
R#XOFF	= 000023	#5-396
R#XON	= 000020	#5-394
R#B00	= 000010	#5-421
R#END	= 000100	#5-414
R#NIT	= 000001	#5-419
R#NOP	= 000000	#5-418
R#RD	= 000002	#5-416 6-456 6-462 8-571
R#SEK	= 000005	#5-417
R#SLF	= 000007	#5-420 6-446 6-448 10-761
R#WR	= 000003	#5-415 6-451 6-457 10-768
R6	= 0000006	#4-274
R7	= 0000007	#4-274
SADR	000102 R	#4-274 *18-1166
SELFTT	000412 R	#6-446 9-603
SEND1	005024 R	19-1214 #19-1226 19-1227
SEND2	005022 R	19-1212 #19-1225
SEND8	005016 R	19-1207 #19-1223
SLFOTL	005600 R	14-1010 #22-1314
SND CNT	000312 R	#5-326 *10-704 *10-756 *11-805
SNDHND	002532 R	9-590 #11-803
SNDPTR	000310 R	#5-325 *8-558 11-803 *11-804
SOF CNT	000042 R	#4-274
SOFER#	= 104406	#4-274 14-993
SOFPAS	000046 R	#4-274
SPDINT	000032 R	#4-274 7-493
SPSIZ	= 000040	#1-7 4-274
SR1	000016 R	#4-274
SR2	000020 R	#4-274 9-665
SR3	000022 R	#4-274 9-640
SR4	000024 R	#4-274 19-1190

SEQ 0043

SEQ 0044