

1 .REM &

2
3
4
5 IDENTIFICATION
6 -----
7

8 PRODUCT CODE: AC-F650A-MC
9
10 PRODUCT NAME: CXTUAA0 TU-58 MODULE
11
12 PRODUCT DATE: JUNE 1979
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 DIGITAL EQUIPMENT CORPORATION

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

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: 1593
OCTAL WORDS: 3071
OCTAL BYTES: 6162

3.0 PASS DEFINITION

A NUMBER OF WR/RD BLOCK OPERATIONS AVERAGED TO ALLOW FOR MAX. 30-SECOND REWINDS PER TAPE.

4.0 EXECUTION TIME

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

5.0 CONFIGURATION PARAMETERS

DEFAULT PARAMETERS:

DVA: 176500 VCT: 300, BR1: 4 BR2: 0, DVC: 1

AT CONFIGURATION TIME SPECIFY:

DVC = 1 FOR TEST DRIVE 0 ONLY

89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144

DVC = 2 FOR TEST DRIVE 1 ONLY
DVC = 3 FOR TEST DRIVE 0 AND DRIVE 1

6.0 DEVICE OPTION SETUP

MAKE SURE SCRATCH TAPE CARTRIDGE IS IN PLACE AND WRITE ENABLED
FOR ALL APPROPRIATE DRIVES.

7.0 MODULE OPERATION

7.1 SUBMODULE DEFINITIONS

INIT: THIS ROUTINE PROVIDES RADIAL SERIAL "BREAK" TO
THE DEVICE, AND EXPECTS THE APPROPRIATE RESPONSE.
IT IS INVOKED UPON STARTUP AND AFTER CERTAIN
ERRORS. FAILURE TO INIT IS GROUNDS FOR MODULE
DROP.

SYSINI: SETS UP BOTH INPUT AND OUTPUT INTERRUPT ROUTINES'
VECTORS AND PRIORITIES. INIT'S DRIVE CONFIGURATION
AND VARIABLES.

DECOD: RETRIEVES NEXT FUNCTIONAL OP-CODE (RD, WR, ETC.)
FROM TEST TABLE IN PREPARATION FOR NEXT COMMUNICATION
PACKET (COMMAND OR DATA), THEN CREATES THE
CORRECT PACKET USING THAT OP CODE THEN
SETS UP THE EXPECTED RESPONSE PARAMETERS
FROM UNIT : THE EXPECTED RECEIVE PACKET BYTE
COUNT(S), AND THE EXPECTED FLAG BYTE(S).

GVNTAK: THIS ROUTINE SETS INTERRUPT ENABLE FOR SEND AND
RECEIVE, THEN ENTERS WAIT LOOP UNTIL PKSNT IS
ASSERTED. IT THEN WAITS ON PKRCV TO COMPLETE,
THEN RETURNS. THESE FLAGS ARE INDICITIVE OF
ENTIRE PACKET SENT OR RECEIVED, NOT A SINGLE
BYTE.

SNDHND: SEND INTERRUPT ROUTINE SENDS PACKET TO DEVICE.
ASSERTS PKSNT AT FINISH; DOES RTI.

RCVHND: RECEIVE INTERRUPT ROUTINE GETS RESPONSE FROM
UNIT (MAY BE MULTIPLE PACKETS) AND CHECKS 1ST
BYTE OF EACH PACKET TO DETERMINE IF IT IS WHAT
WAS EXPECTED IN ORDER TO ADJUST RECEIVE BYTE
COUNT IF NOT, CHECKS FOR EARLY END PACK
OR DLV "OVERRUN" (DATA LATE). ASSERTS PKRCVD
AT COMPLETION AND RTI'S SINCE UNITS RESPONSE IS
CONTINUOUS (NO HANDSHAKE) AND FAST (38.4K BAUD
MAX!).

145 CHKANS: BASED ON THE RESPONSE, THIS ROUTINE CALLS ROUTINES
146 TO CHECK CHECKSUM, CHECK END PACKET, AND/OR CHECK DATA.
147
148

149
150 7.2 DESCRIPTION OF TABLES, QUEUES, BUFFERS
151 -----
152

153 OPTBL: LIST OF ENCODED WORDS CONTAINING THE NEXT PACKET
154 TO SEND, WHICH DRIVE TO ENABLE, AND IF PACKET IS
155 COMMAND OR DATA.
156

157 SNDBUF: AREA WHERE PACKET IS BUILT AND SUBSEQUENTLY SENT
158 VIA INTERRUPT ROUTINE. CONTAINS 1 PACKET - BE
159 IT COMMAND OR DATA.
160

161 RCVBUF: AREA WHERE PACKET(S) ARE RECEIVED DURING INTERRUPT
162 SERVICE. MAY CONTAIN MAX OF: [DATA-DATA-DATA-
163 DATA-END] PACKETS IN SEQUENCE.
164

165 8.0 OPERATOR OPTIONS
166 -----
167

168 A. USE THE "MOD" COMMAND TO DUMP THE BUFFERS IN ORDER TO MAXIMIZE
169 ERROR INFORMATION.
170

171

```

172 000000'
173 000000'
174
175 ;
176 .LIST RTN
177 ;*****
178 REGIN:
179 MODNAM: .ASCII /TUAA / ;MODULE NAME.
180 XFLAG: .BYTE OPEN ;USED TO KEEP TRACK OF WBUFF USAGE
181 ADDR: 176500+0 ;1ST DEVICE ADDR.
182 VECTOR: 300+0 ;1ST DEVICE VECTOR.
183 BR1: .BYTE PRTY4+0 ;1ST BR LEVEL.
184 BR2: .BYTE PRTY0+0 ;2ND BR LEVEL.
185 DVID1: 2+1 ;DEVICE INDICATOR 1.
186 SR1: OPEN ;SWITCH REGISTER 1
187 SR2: OPEN ;SWITCH REGISTER 2
188 SR3: OPEN ;SWITCH REGISTER 3
189 SR4: OPEN ;SWITCH REGISTER 4
190 ;*****
191 STAT: 140000 ;STATUS WORD.
192 INIT: START ;MODULE START ADDR.
193 SPOINT: MODSP ;MODULE STACK POINTER.
194 PASCNT: 0 ;PASS COUNTER.
195 ICNT: 1 ;# OF ITERATIONS PER PASS=1
196 ICONT: 0 ;LOC TO COUNT ITERATIONS
197 SOFCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
198 HRDCNT: 0 ;LOC TO SAVE TOTAL HARD ERRORS
199 SOFPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
200 HRDPAS: 0 ;LOC TO SAVE HARD ERRORS PER PASS
201 SYSCNT: 0 ;# OF SYS ERRORS ACCUMULATED
202 RANNUM: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
203 CONFIG:
204 RES1: 0 ;RESERVED FOR MONITOR USE
205 RES2: 0 ;RESERVED FOR MONITOR USE
206 SVR0: OPEN ;LOC TO SAVE R0.
207 SVR1: OPEN ;LOC TO SAVE R1.
208 SVR2: OPEN ;LOC TO SAVE R2.
209 SVR3: OPEN ;LOC TO SAVE R3.
210 SVR4: OPEN ;LOC TO SAVE R4.
211 SVR5: OPEN ;LOC TO SAVE R5.
212 SVR6: OPEN ;LOC TO SAVE R6.
213 CSRA: OPEN ;ADDR OF CURRENT CSR.
214 SBADR: ;ADDR OF GOOD DATA, OR
215 ACSR: OPEN ;CONTENTS OF CSR.
216 WASADR: ;ADDR OF BAD DATA, OR
217 ASTAT: OPEN ;STATUS REG CONTENTS.
218 ERRTRYP: ;TYPE OF ERROR
219 ASB: OPEN ;EXPECTED DATA.
220 AWAS: OPEN ;ACTUAL DATA.
221 RSTRT: RESTRT ;RESTART ADDRESS AFTER END OF PASS
222 WDTO: OPEN ;WORDS TO MEMORY PER ITERATION
223 WDFR: OPEN ;WORDS FROM MEMORY PER ITERATION
224 INTR: OPEN ;# OF INTERRUPTS PER ITERATION
225 IDNUM: 17 ;MODULE IDENTIFICATION NUMBER=17
226 .REPT SPSIZ ;MODULE STACK STARTS HERE.
227 .NLIST

```

```

228 .WORD 0
229 .LIST
230 .ENDR
231 MODSP:
232 ;*****

```

```

233 ;SBTTL MODULE DATA SECTION
234 ;----- FLAGS --- ETC -----
235
236 000224* 000000 PKSENT: WORD ;PACKET SENT
237 000226* 000000 PKRCVD: WORD ;PACKET RECEIVED
238 000230* 000000 LASTOP: WORD ;1-PASS-THROUGH-TABLE FLAG
239 000232* 000000 FSTFLG: WORD ;EXPECTING_FLAG_BYTE FLAG
240 000234* 000000 RCSP: WORD ;RCV STATUS REG. ADDR.
241 000236* 000000 RCDP: WORD ;RCV DATA BUFFER ADDR.
242 000240* 000000 XMSR: WORD ;TRANSMIT STATUS REG. ADDR.
243 000242* 000000 XMDB: WORD ;TRANSMIT DATA BUFFER ADDR.
244 000244* 000000 DVIDX: WORD ;DRIVE CONFIGURATION
245 ;-----
246 ;----- OTHER STORAGE -----
247
248
249 000246* 000000 SYSTAT: WORD ;SYSTEM STATUS WORD
250 ;BIT3=OUT-TO-LUNCH ERROR (CHKANS)
251 ;BIT1=SELF TEST SENT
252 ;BIT0=CHKSUM ODD
253 000250* 000000 DRV: WORD ;CURRENT DRIVE NUMBER
254 000252* 000002 ASDRV: BLKW 2 ;ASCII OF CURRENT DRIVE NUMBER
255 000256* 000000 WORD 0
256 000260* 000000 WORD 0
257
258 000262* 000002 ASRDBT: BLKW 2 ;ASCII OF # OF BAD BYTES
259 000266* 000000 WORD 0
260 000270* 000000 WORD 0
261
262 000272* 000000 REC: WORD ;RECORD NUMBER
263 000274* 000000 PATRN: WORD ;DATA PATTERN
264 000276* 000000 SUCCS: WORD ;SUCCESS CODE OF END PACKET
265 000300* 000000 PKPTR: WORD ;TOP OF CURRENT PACKET (ADDR.)
266 000302* 000000 XSPTR: WORD ;POINTER TO NEXT X$FLG OR X$CNT
267 000304* 000000 TBLTOP: WORD ;TOP OF OP TABLE
268 000306* 000000 TBLPTR: WORD ;POINTER TO OP CODE IN OPTBL
269 000310* 000000 SMDPTR: WORD ;POINTER TO XMIT BUFFER
270 000312* 000000 SMDCNT: WORD ;NUMBER OF BYTES REMAINING TO SEND
271 000314* 000000 RCVPTR: WORD ;POINTER IN RECEIVE BUFFER
272 000316* 000000 RCPKNT: WORD ;# OF PACKETS LEFT DURING RECIEVE
273 000320* 000000 PKCHKD: WORD ;# OF PACKETS BEING CHECKED IN CHKANS
274 000322* 000000 RVCNT: WORD ;ACTUAL COUNT USED IN RECEIVE
275 000324* 000000 RCBCNT: WORD ;COUNT USED IN CHECKANS
276 000326* 000000 X$PKNT: WORD ;EXPECTED # OF PACKETS TOTAL
277 000330* 000000 X$FLG: WORD ;THE EXPECTED FLAG BYTE OF 1ST PACKET
278 000332* 000000 X$CNT: WORD ;AND IT'S TOTAL BYTE COUNT
279 000334* 000010 BLKW 8 ;MULTIPLE PACKET EXPECTED FLAGS AND COUNTS
280 000334* 000000 ERROP: WORD ;THE TABLE OP CODE ON ERROR
281 000336* 000000 ERRFLG: WORD ;THE EXPECTED FLAG ON ERROR (@X$PTR)
282 000360* 000000 ERRER: WORD ;ERROR FLAG
283 000362* 000000 ERRWD: WORD ;CONTENTS OF DLV RCDP ON DLV ERROR
284 000364* 000000 ATPAK: WORD ;@PKPTR ON ERROR-1ST 2 BYTES
285 000366* 000000 ATPAK2: WORD ;@PKPTR+2 ON ERROR-NEXT 2 BYTES
286 000370* 000000 BDRYTS: WORD ;NUMBER OF BAD DATA BYTES (MAX. 128.)
287
288 ;-----

```

```

289
290 000372* 000000 BRKWD: WORD 0 ;2 NULL BYTES
291 000374* 004 INITWD: BYTE R$INIT ;INIT BYTE
292 000375* 000 R$CNT: BYTE 0 ;R$CNT EXPECTED HERE
293 ;-----
294 ;----- GENERAL EQUATES -----
295
296
297 000100 EI = 100
298 000100 DI = 100
299 000000 D$DATA = 0
300 000001 DTAFL1 = 1
301 000010 DTAFL2 = 10
302 000011 DTAFL3 = 11
303 000100 DTAFL4 = 100
304 100000 CMNDFL = 100000
305 000000 DRV0 = 0
306 000400 DRV1 = 400
307 ;-----
308 ;----- PROTOCOL EQUATES -----
309
310 ;FLAG BYTE CODES:
311
312
313
314 000002 R$CMND = 2
315 000020 R$CNT = 20
316 000020 R$XON = 20
317 000023 R$XOFF = 23
318 000004 R$INIT = 4
319 000001 R$DATA = 1
320 000002 R$END = R$CMND
321
322 ;SIZES:
323
324 000016 P$NSZ = 16 ;END PACK SIZE
325 000012 R$MSZ = 12 ;MESSAGE SIZE
326 000204 P$DASZ = 132 ;DATA PACK SIZE
327 000222 R$DNSZ = R$DASZ+R$NSZ ;DATA PLUS END PACK SIZE
328 000016 R$SNSZ = R$MSZ+4 ;SEND SIZE OF MESSAGE PACK
329
330 ;OP CODES:
331
332 000100 R$END = 100
333 000003 R$WR = 3
334 000002 R$RD = 2
335 000005 R$SFK = 5
336 000000 R$NOP = 0
337 000001 R$MIT = 1
338 000007 R$SLF = 7
339
340 ;SUCCESS CODES
341
342 000000 F$OK = 0 ;GOOD SUCCESS
343 177720 F$ARO = -40 ;BAD COMMAND
344 177767 F$NCR = -0 ;NO CART.

```

345 177770
 346 177776
 347 177740
 348 000001
 349 177765
 350 177737
 351 177720
 352 177711
 353 177757
 354 177777
 355 177757
 356 177757
 357 177757
 358
 359

ESNONX = -R, ;NO DRIVE
 ESPART = -2, ;E.O.M.
 ESSK = -32, ;SEEK ERROR
 ESTRY = 1 ;PETRIES
 ES#LOC = -11, ;WRITE PROT.
 ES#OMO = -33, ;MOTOR STOPPED
 ESCMD = -48, ;#BAD COMMAND
 ESREC = -55, ;BAD RECORD # SENT
 ESCK5 = -17, ;DATA CHECK ERROR
 ES5LF = -1, ;SELF TEST FAIL
 ESCKSM = ESCK5
 ES#R = ESCK5
 ES#RD = ESCK5

;------

360
 361
 362 000376° 100007
 363 000400° 100003
 364 000402° 000001
 365 000404° 000010
 366 000406° 000011
 367 000410° 000100
 368 000412° 100002
 369 000414° 100407
 370 000416° 100403
 371 000420° 000401
 372 000422° 000410
 373 000424° 000411
 374 000426° 000500
 375 000430° 100402
 376 000432° 177777
 377
 378
 379 000434° 100407
 380
 381
 382
 383

;------ TEST COMMAND TABLE -----

OPTBL1 .WORD R##SLFICMNDFLIDRV0
 .WORD R##WRICMNDFLIDRV0
 .WORD D\$DATAIDTAFL1IDRV0
 .WORD D\$DATAIDTAFL2IDRV0
 .WORD D\$DATAIDTAFL3IDRV0
 .WORD D\$DATAIDTAFL4IDRV0
 .WORD R##RDICMNDFLIDRV0
 OPTBL2 .WORD R##SLFICMNDFLIDRV1
 .WORD R##WRICMNDFLIDRV1
 .WORD D\$DATAIDTAFL1IDRV1
 .WORD D\$DATAIDTAFL2IDRV1
 .WORD D\$DATAIDTAFL3IDRV1
 .WORD D\$DATAIDTAFL4IDRV1
 .WORD R##RDICMNDFLIDRV1
 .WORD -1

DR1OP1 .WORD R##SLFICMNDFLIDRV1

;------

;------ TABLE FOR EXTENDED ERROR INFO -----

ERRTBL1 .WORD ERROP ;OP CODE IN COMMAND TABLE (OPTBL)
 .WORD REC ;THE RECORD NUMBER
 .WORD PATRN ;THE DATA PATTERN
 .WORD RCPTR ;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-BITS)
 .WORD ERRWD ;CONTENTS OF DLV RCDB IF DLV ERROR
 ERRTBL2 .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 ;

;------

401

```

402          ,SBTTL MAIN PROGRAM CONTROL
403 000500° 016706 177326 START: MOV SPOINT,SP ;THE INIT OF INITS
404 000504° 016767 177304 MOV DVID1,DVIDX ;COPY DRIVE CONFIGURATION
405 000512° 126737 177404 000041 CMPB IDNUM,#41 ;ARE WE THE LOAD DEVICE?
406 000520° 001016 BNE 2$ ;NO
407 000522° 105737 000040 TSTB R#40 ;YES-DRIVE 0?
408 000526° 001004 BNE 1$ ;NO
409 000530° 042767 000001 177506 BIC #RIT0,DVIDX ;YES-DESELECT
410 000536° 000407 BP 2$ ;CONTINUE
411 000540° 122737 000001 000040 1$: CMPB #1,#40 ;NOT DR0,DR1?
412 000546° 001003 BNE 2$ ;NO-SOME OTHER DR
413 000550° 042767 000002 177466 BIC #RIT1,DVIDX ;YES DESELECT
414 000556° 004767 0000310 2$: CALL SYSINI ;SET VECTOR GET CONFIG.
415 000562° 000240 RESTRT: NOP ;MAIN LOOP
416 000564° 104417 000000° RANDB,BEGIN
417 000570° 016767 177260 177474 MOV RANNUM,REC ;GET RECORD #
418 000576° 042767 177000 177466 BIC #177000,REC ;SIZE IT FOR TU58
419
420 000604° 104417 000000° RANDB,BEGIN
421 000610° 016767 177240 177456 MOV RANNUM,PATRN ;PROCURE DATA
422 000616° 042767 177400 177450 BIC #177400,PATRN ;1-BYTE IT
423
424 000624° 004767 000456 CALL DECOD ;INTERPRET OPERATION FROM TABLE; MAKE PACKET
425
426 000630° 004767 001652 RSTRT2: CALL CLRBUF ;ZERO RECEIVE BUFFER
427 000634° 004767 000100 CALL GVNATK ;SEND PACKET, RECEIVE RESPONSE
428 000640° 004767 001666 CALL CHKANS ;INTERPRET RESPONSE
429 000644° 005767 177510 TST ERRER ;ERROR?
430 000650° 001413 BEQ 1$ ;NO
431
432 000652° 016767 177426 177426 3$: MOV TBLTOP,TBLPTR ;YES, -->1ST OPERATION.
433 000660° 005067 177474 CLR ERRER ;RESET ERROR FLAG
434 000664° 005067 177356 CLR SYSTAT ;RESET SYSTEM STATUS
435 000670° 005067 177212 CLR ERRTYP ;RESET ERROR TYPE
436 000674° 004767 001372 CALL DOBRK ;INIT DEVICE
437 000700° 004767 000402 1$: CALL DECOD ;DECODE NEXT OPERATION
438 000704° 005767 177320 TST LASTOP ;DONE ONE TABLE'S WORTH?
439 000710° 001747 BEQ RSTRT2 ;NOT YET
440 000712° 104413 000000° ENDIT$,BEGIN ;SIGNAL END OF ITERATION.
441 ;MONITOR SHALL TEST END OF PASS
442 000716° 000721 BR RESTRT ;KEEP ON
443
444 000720° 042777 000100 177306 DROP: BIC #DI,ORCSR ;SHUT OFF
445 000726° 042777 000100 177304 BIC #DI,EXMSP ;ALL INTERRUPTS
446 000734° 104410 000000° END$,BEGIN ;
447

```

```

448          ,SBTTL GVNATK / MAIN COMMUNICATIONS LOOPS
449
450 000740° 012767 000001 177264 GVNATK: MOV #1,FSTFLG ;SET 1ST-BYTE FLAG
451 000746° 005067 177252 CLR PKSENT ;PACK NOT SENT YET
452 000752° 005067 177250 CLR PKRCVD ;PACK NOT RECEIVED YET
453 000756° 005067 177314 CLR SUCCS ;NO SUCCESS CODE YET
454 000762° 016767 177344 177332 MOV X$CNT,RCVCNT ;COPY # BYTES TO RECIEVE
455 000770° 016767 177332 177320 MOV X$PKNM,RCPKNM ;COPY # PACKETS TO RCV
456 000776° 012767 000330° 177276 MOV X$FLG,X$PTR ;COPY 1ST FLAG BYTE ADDR, POINTER
457 001004° 012767 005122° 177302 MOV #RCVBUF,RCVPTR ;-->TOP OF RECEIVE BUFF
458 001012° 012767 004714° 177270 MOV #TRBUF,SNDPTR ;-->TOP OF XMIT BUFFER
459 001020° 052777 000100 177212 BIS #EI,EXMSR ;SET UP TO SEND
460 001026° 052777 000100 177200 BIS #EI,ORCSR ;AND RECEIVE
461
462 001034° 104407 000000° 1$: BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR,...
463 001040° 104407 000000° BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
464 001044° 005767 177134 TST PKSENT ;DONE SENDING?
465 001050° 001771 BEQ 1$ ;NO
466 001052°
467 001052° 104407 000000° 2$: BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR,...
468 001056° 104407 000000° BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
469 001062° 005767 177140 TST PKRCVD ;DONE RECEIVING?
470 001066° 001771 BEQ 2$ ;NO
471 001070° 000207 RTS PC
472

```



```

473          ,SMTTL  SYSINI / SYSTEM (MODULE) INITIALIZATION
474
475 001072* 016700 176712  SYSINI:  MOV  VECTOR,R0      ;SET UP VECTORS:
476 001076* 012720 002074*  MOV  #PCVHND,(R0)+    ;RECEIVE ROUTINE
477 001102* 116720 176704  MOVB  BR1,(R0)+      ;AND PRIORITY
478 001106* 105720  TSTR  (R0)+          ;-->XMIT VECTOR ADDR
479 001110* 012720 002034*  MOV  #SNDHND,(R0)+    ;SET UP SEND ROUTINE ADDR
480 001114* 116710 176672  MOVB  BR1,#P0        ;AND PRIORITY
481 001120* 016700 176662  MOV  ADDR,R0         ;GET DEVICE ADDRESSES:
482 001124* 010067 177104  MOV  P0,PCSR        ;
483 001130* 005720  TST  (R0)+          ;
484 001132* 010067 177100  MOV  R0,RCDB         ;
485 001136* 005720  TST  (R0)+          ;
486 001140* 010067 177074  MOV  R0,XMSR        ;
487 001144* 005720  TST  (R0)+          ;
488 001146* 010067 177070  MOV  P0,XMDB        ;
489 001152* 032767 000001 177064  BIT  #BIT0,DVIDX     ;DR0 SELECTED?
490 001160* 001416  BEQ  1$              ;NO
491 001162* 012767 000376* 177114  MOV  #OPTBL,TBLTOP   ;YES-TABLE STARTS AT DR0 OPERATIONS
492 001170* 012767 177777 177216  MOV  #-1,OPTBL2     ;PRESET AS NO DR1
493 001176* 032767 000002 177040  BIT  #BIT1,DVIDX     ;SELECTED HOWEVER?
494 001204* 001420  BEQ  SYSSET        ;NO-ALL SET
495 001206* 016767 177222 177200  MOV  DR1OP,OPTBL2   ;YES-INSERT OP CODE
496 001214* 000414  BR   SYSSET        ;CONTINUE
497 001216* 032767 000002 177020 181 BIT  #BIT1,DVIDX     ;HERE=NO DR0...DR1?
498 001224* 001002  BNE  2$              ;YES-CONTINUE
499 001226* 004767 177466  CALL  DROP           ;NO DRIVES=NO WAY
500 001232* 012767 000414* 177044 2$  MOV  #OPTBL2,TBLTOP   ;JUST DR1-TOP IS DR1 START
501 001240* 016767 177170 177146  MOV  DR1OP,OPTBL2   ;INSERT OP CODE
502 001246* 016767 177032 177032  SYSSET: MOV  TBLTOP,TBLPTR ;NOW SET POINTER TO TOP.
503 001254* 004767 001012  CALL  DOBRK         ;INIT UNIT
504 001260* 005067 177074  CLR  ERRER         ;AND ERROR NUMBER
505 001264* 005067 177072  CLR  ERPHD         ;AND ERROR WORD
506 001270* 005067 177074  CLR  BOBYTS       ;AND BAD BYTES
507 001274* 005067 176746  CLR  SYSTAT       ;AND SYSTEM STATUS
508 001300* 005067 176602  CLR  ERRTP        ;AND ERROR TYPE
509 001304* 000207  RTS  PC

```

```

510 001306* 027727 176774 177777  DECOD:  CMP  #TBLPTR,-1    ;LAST OP CODE PROCESSED?
511 001314* 001007  BNE  1$              ;NO
512 001316* 016767 176762 176762  MOV  TBLTOP,TBLPTR   ;YES-POINT TO TOP
513 001324* 005267 176700  INC  LASTOP         ;SET FLAG
514 001330* 000167 000476  JMP  DECRET         ;EXIT
515 001334* 012700 004714* 181  MOV  #TRBUF,R0       ;POINT TO TOP OF TRANSMIT BUFFER
516 001340* 005777 176742  TST  #TBLPTR        ;IS IT COMMAND?
517 001344* 100460  BMI  DECCMD        ;YES
518 001346* 012700 004714*  MOV  #TRBUF,R0       ;NO-THEREFORE IS DATA PACK!
519 001352* 112710 000001  MOVB  #P$DATA,@R0    ;FLAG BYTE
520 001356* 112760 000200 000001  MOVB  #128,,1(R0)    ;BYTE COUNT
521 001364* 012702 000200  MOV  #128,,R2       ;FOR DATA INSERTION COUNT
522 001370* 005720  TST  (R0)+          ;GET BY THE BYTE COUNT
523 001372* 116720 176676 281  MOVB  PATRN,(R0)+    ;AND INSERT DATA
524 001376* 005302  DEC  R2              ;UNTIL
525 001400* 001374  BNE  2$              ;COMPLETED
526 001402* 012767 000204 176702  MOV  #132,,SNDCNT   ;TOTAL COUNT FOR SENDING
527 001410* 012701 000202  MOV  #130,,R1       ;COUNT FOR CHECKSUM ROUTINE
528 001414* 012700 004714*  MOV  #TRBUF,R0       ;POINT TO TOP AGAIN
529 001420* 004767 002072  CALL  CHKSUM        ;CALC. THE CHECKSUM
530 001424* 010110  MOV  R1,(R0)         ;RETURNED IN R1 WITH R0 AS POINTER
531 001426* 032777 000100 176652  BIT  #DTAFL4,@TBLPTR ;LAST DATA PACKET?
532 001434* 001412  BEQ  3$              ;NO, EXPECT CONTINUE, ETC.
533 001436* 012767 000002 176664  MOV  #R$END,X$FLG   ;YES, EXPECT END PACK
534 001444* 012767 000016 176660  MOV  #R$NDSZ,X$CNT  ;THIS BIG
535 001452* 012767 000001 176646  MOV  #1,X$PKNM      ;AND ONLY 1 PACKET RESPONSE
536 001460* 000554  BR   DECRET1       ;EXIT
537 001462* 012767 000020 176640 381  MOV  #R$CONT,X$FLG  ;HERE NOT LAST PACKET
538 001470* 012767 000001 176634  MOV  #1,X$CNT       ;EXPECT; CONTINUE; 1 BYTE
539 001476* 012767 000001 176622  MOV  #1,X$PKNM      ;1 PACKET
540 001504* 000542  BR   DECRET1       ;EXIT
541 001506* 000240  DECCMD:  NOP                ;HERE WE BUILD A COMMAND PACKET!
542 001510* 112710 000002  MOV  #P$CMND,@R0     ;FLAG BYTE
543 001514* 112760 000012 000001  MOVB  #R$MSIZ,1(R0)  ;MESSAGE SIZE
544 001522* 117600 176560 000002  MOVB  #TBLPTR,2(R0) ;THE OP CODE
545 001530* 105000 000003  CLR  3(R0)         ;NOT VERIFY
546 001534* 105000 000004  CLR  4(R0)         ;PRESET TO DRIVE 0
547 001540* 005067 176504  CLR  DRV          ;DITTO
548 001544* 032777 000400 176534  BIT  #BIT8,@TBLPTR  ;WAS DRIVE 1, THOUGH?
549 001552* 001404  BEQ  4$              ;NO
550 001554* 105200 000004  INCB  4(R0)         ;YES-SELECT DR1
551 001560* 005267 176464  INC  DRV          ;DITTO
552 481  ;*****
553  ;CONVERT DRV TO ASCII AND
554  ;STORE AT ASDRV
555  BTODS,BEGIN,DRV,ASDRV
556 001564* 104421 000000* 000250*
557 001572* 000252*
558  ;*****
559 001574* 105000 000005  CLR  5(R0)         ;NO MODIFIERS
560 001600* 005000 000006  CLR  6(R0)         ;NO SEQUENCE #
561 001604* 012760 001000  MOV  #512,,8,(R0)   ;USF STANDARD BLOCK SIZE
562 001612* 016760 176454 000012  MOV  #C,10,(R0)    ;INSERT RECORD NUMBER
563 001620* 012701 000012  MOV  #P$MSIZ,R1     ;MESSAGE PACKET SIZE
564 001624* 005721  TST  (R1)+          ;PLUS BYTE COUNT AND FLAG
565 001626* 012767 000016 176456  MOV  #P$SNSZ,SNDCNT ; SIZE TO SEND

```

```

566 001634* 004767 001656          CALL  CHKSUM          ;GET CHECKSUM, RETURNED
567 001640* 010110          MOV    R1,R0          ;IN R1 WITH R0-->WHERE
568                                ;THE CHECKSUM WORD GOES
569 001642* 042767 000002 176376    BIC   #BIT1,SYSTAT   ;NOT SELF TEST PRESET
570 001650* 122777 000007 176430    CMPR  #R6$SLF,@TBLPTR ;WAS THE OPERATION SELF TEST?
571 001656* 001015          BNE   1$              ;NO
572 001660* 052767 000002 176360    RIS   #BIT1,SYSTAT   ;YES-SELF TEST FLAG SET
573 001666* 012767 000002 176434    MOV   #R0FND,X$FLG   ;YES-EXPECT END PACK
574 001674* 012767 000016 176430    MOV   #R$NDSZ,X$CNT  ;OF PROPER SIZE
575 001702* 012767 000001 176416    MOV   #1,X$PKNM      ;AND ONLY 1 PACKET
576 001710* 000440          BR    DECRT1         ;EXIT
577 001712* 122777 000003 176366    1$:  CMPB  #R$SWR,@TBLPTR ;NOT SELF TEST-WAS IT WRITE?
578 001720* 001012          BNE   2$              ;NO
579 001722* 012767 000020 176400    MOV   #R$CONT,X$FLG  ;YES-EXPECT CONTINUE
580 001730* 012767 000001 176374    MOV   #1,X$CNT       ;AND 1 BYTE
581 001736* 012767 000001 176362    MOV   #1,X$PKNM      ;AND 1 PACKET
582 001744* 000422          BR    DECRT1         ;EXIT
583 001746* 005000          2$:  CLR   R0            ;HERE TRANSACTION MUST BE READ
584 001750* 012703 000330*        MOV   #X$FLG,R3      ;GET ADDR. OF X$FLG
585 001754* 012723 000001 176204    3$:  MOV   #R0DATA,(R3)+ ;INSERT TOTAL OF
586 001760* 012723 000204        MOV   #132,(R3)+    ;4 DATA FLAG BYTES EXPECTED,
587 001764* 005200          INC   R0              ;SANDWICHED WITH
588 001766* 022700 000004        CMP   #4,,R0        ;THE COUNT OF
589 001772* 001370          BNE   3$              ;EACH
590 001774* 012723 000002        MOV   #R$END,(R3)+  ;THEN THE END PACK
591 002000* 012713 000016        MOV   #R$NDSZ,(R3) ;AND ITS SIZE
592 002004* 012767 000005 176314    MOV   #5,X$PKNM      ;CORRECT TO 5 EXPECTED PACKS
593
594 002012* 017767 176270 176334    DECRT1: MOV @TBLPTR,ERROR ;COPY OP CODE IN CASE ERROR
595 002020* 062767 000002 176260    ADD  #2,TBLPTR      ;UPDATE POINTER FOR NEXT OPERATION
596 002026* 005067 176176          CLR   LASTOP        ;INDICATE STILL IN TABLE
597 002032*
598 002032* 000207          DECRET: RTS         PC

```

```

599                                .SBTTL / INTERRUPT SERVICE ROUTINES /
600
601 002034* 117777 176250 176200    SNDHND: MOVB  @SNDPTR,@XMDB ;OUTPUT BYTE
602 002042* 005267 176242          INC   SNDPTR        ;-->NEXT
603 002046* 005367 176240          DEC   SMDCNT        ;ANY MORE?
604 002052* 001005          BNE   1$            ;YES
605 002054* 005267 176144          INC   PKSNT         ;NO-SET FLAG
606 002060* 042777 000100 176152    BIC   #DI,@XMSR    ;AND NO MORE INTS
607 002066* 000002          RTI                ;FAST AND FURIOUS
608
609 002070* 042777 000100 176136    RCVHND: BIC   #DI,@RCR  ;KILL ANY NESTING
610 002076* 017767 176134 176256    MOV   @RCDB,ERRWD   ;GET BYTE+ERROR
611 002104* 116777 176252 176202    MOVB  ERRWD,@RCVPTR ;SAVE BYTE IN BUFFER
612 002112* 005767 176244          TST  ERRWD          ;ERROR?
613 002116* 100003          BPL  RCVOK         ;NO
614 002120* 005267 176102          INC  PKRCVD        ;YES SET RCVD FLAG
615 002124* 000461          BR   RCVRET        ;EXIT
616 002126* 005067 176230          CLR  ERRWD         ;NO-CLEAR ERROR
617 002132* 005767 176074          TST  FSTFLG        ;IS THIS A 1ST BYTE?
618 002136* 001423          BEQ  1$            ;NO
619 002140* 005067 176066          CLR  FSTFLG        ;YES-CLEAR
620 002144* 127777 176132 176142    CMPB  @X$PTR,@RCVPTR ;WAS IT THE FLAG WE EXPECTED?
621 002152* 001415          BEQ  1$            ;YES-CONTINUE
622 002154* 012767 000001 176140    MOV   #1,RCVCNT     ;OOPS-PRESET FOR EXIT
623 002162* 012767 000001 176126    MOV   #1,RCPKNM     ;IS LAST PACKET, TOO.
624 002170* 127727 176120 000002    CMPB  @RCVPTR,#R$END ;WAS IT A PREMATURE END PACK?
625 002176* 001003          BNE  1$            ;NO WAY
626 002200* 012767 000016 176114    MOV   #R$NDSZ,RCVCNT ;YES USE CORRECT COUNT
627 002206* 005267 176102          1$:  INC  RCVPTR       ;-->NEXT SLOT
628 002212* 005367 176104          DEC  RCVCNT        ;MORE OF THIS PACKET?
629 002216* 001016          BNE  2$            ;YES
630 002220* 005367 176072          DEC  RCPKNM        ;NO-ANY MORE PACKETS?
631 002224* 001417          BEQ  3$            ;NO
632 002226* 062767 000006 176046    ADD  #6,X$PTR       ;YES-->NEXT EXPECTED COUNT
633 002234* 017767 176042 176060    MOV  @X$PTR,RCVCNT  ;GET NEW COUNT
634 002242* 162767 000002 176032    SUB  #2,X$PTR       ;-->NEXT EXPECTED FLAG
635 002250* 005267 175756          INC  FSTFLG        ;SAY IT'S 1ST BYTE
636 002254* 052777 000100 175752    2$:  RIS  #FI,@RCR     ;OK TO RE-ENABLE INTS.
637 002262* 000402          BR   RCVRET        ;QUIT; QUICKLY
638 002264* 005267 175736          3$:  INC  PKRCVD      ;HEP! LAST PACKET RCV'D
639
640 002270* 000002          RCVRET: RTI        ;EXIT

```

```

        641                ,SBTTL  UTILITY SUBROUTINES
        642
        643 002272' 000240 DOBRK: NOP                ;SEND INIT TO DEVICE
        644 002274' 105067 176075 CLRB          INITWD+1        ;START FRESH
        645 002300' 042777 000100 175732 BIC          #DI,0XMSR      ;KILL SEND
        646 002306' 042777 000100 175720 BIC          #DI,0PCSR      ;KILL RECEIVE
        647 002314' 052777 000001 175716 BIS          #BIT0,0XMSR     ;SET "BREAK"
        648 002322' 012704 000010          MOV          #0,,R4        ;SET COUNT
        649 002326'
        650 002326' 104407 000000' 18: BREAK$,BEGIN      ;TEMPORARY RETURN TO MONITOR,...
        651 002332' 104407 000000' BREAK$,BEGIN      ;THEN CONTINUE AT NEXT INSTRUCTION.
        652 002336' 105777 175676 TSTB         0XMSR        ;READY?
        653 002342' 100371 BPL          18          ;NO
        654 002344' 116777 176022 175670 MOVB        BRKWD,0XMDB    ;SEND NULL
        655 002352' 005304 DEC          R4         ;MORE NULLS TO SEND?
        656 002354' 001364 BNE          18          ;YES
        657 002356' 042777 000001 175654 BIC          #BIT0,0XMSR   ;NO-CLEAR "BREAK"
        658 002364' 017704 175646 MOV          #PCDB,R4     ;CLEAR POSSIBLE GARBAGE
        659 002370' 012704 000002          MOV          #2,R4        ;NUMBER OF "INITS" TO SEND
        660 002374'
        661 002374' 104407 000000' 28: BREAK$,BEGIN      ;TEMPORARY RETURN TO MONITOR,...
        662 002400' 104407 000000' BREAK$,BEGIN      ;THEN CONTINUE AT NEXT INSTRUCTION.
        663 002404' 105777 175630 TSTB         0XMSR        ;READY TO SEND?
        664 002410' 100371 BPL          28          ;NO
        665 002412' 116777 175756 175622 MOVB        INITWD,0XMDB  ;YES-SEND
        666 002420' 005304 DEC          R4         ;MORE?
        667 002422' 001364 BNE          28          ;YES-CONTINUE
        668 002424'
        669 002424' 104407 000000' 38: BREAK$,BEGIN      ;TEMPORARY RETURN TO MONITOR,...
        670 002430' 104407 000000' BREAK$,BEGIN      ;THEN CONTINUE AT NEXT INSTRUCTION.
        671 002434' 105777 175574 TSTB         0PCSR        ;READY TO RECEIVE?
        672 002440' 100371 BPL          38          ;NO
        673 002442' 117767 175570 175725 MOVB        #PCDB,INITWD+1 ;YES-GET RESPONSE
        674 002450' 126727 175721 000020 CMPB        INITWD+1,#RCONT ;IS CONTINUE?
        675 002456' 001412 BEQ          48          ;YES
        676 002460' 004767 001126 CALL         ERSET        ;NO-PREPARE EXTENDED ERROR INFO.
        677 002464' 012767 000034 175414 MOV          #34,ERRTYP    ;CODE FOR "CAN NOT INITIALIZE"
        678
        679 002472' 104405 000000' 000436' HRDR$,BEGIN,ERRTBL      ;*****
        680 ;*****
        681 ;*****
        682 002500' 004767 176214 CALL DROP      ;FATAL
        683 002504'
        684 002504' 000207 48: RTS          PC
    
```

```

        685                ;MODULE TO ZERO THE 542, BYTE RECEIVE BUFFER "RCVBUF"
        686
        687 002506' 000240 CLRBUF: NOP                ;ZEROS RECEIVE BUFFER
        688 002510' 012703 005122' MOV          #RCVBUF,R3    ;GET ADDRESS
        689 002514' 012704 001036 MOV          #542,,R4     ;AND THE SIZE
        690 002520' 005023 18: CLR          (R3)+        ;CLEAR
        691 002522' 102704 000002 SUB          #2,R4         ;2 BYTES LESS
        692 002526' 001374 BNE          18          ;RR IF MORE
        693 002530' 000207 RTS          PC
    
```

```

;MODULE TO INTERPPRET ANY RESPONSE FROM DEVICE.
694
695
696 002532* 000240
697 002534* 005067 175560
698 002540* 012700 005122*
699 002544* 016702 175556
700 002550* 012703 000330*
701 002554* 010301
702 002556* 062701 000002
703 002562* 010067 175512
704 002566* 005267 175526
705 002572* 011167 175526
706 002576* 004767 001010
707 002602* 121013
708 002604* 001037
709 002606* 121027 000020
710 002612* 001500
711
712 002614* 016704 175504
713 002620* 005744
714 002622* 004767 001046
715 002626* 103012
716 002630* 104403 000000* 004556*
717
718 002636* 104405 000000* 000436*
719
720 002644* 012767 177777 175506
721 002652* 000475
722 002654* 122710 000002
723 002660* 001003
724 002662* 004767 000240
725 002666* 000467
726 002670* 122710 000001
727 002674* 001056
728 002676* 004767 001032
729 002702* 000444
730
731
732 002704* 000240
733 002706* 122710 000004
734 002712* 001012
735 002714* 104403 000000* 004570*
736
737 002722* 104405 000000* 000436*
738
739 002730* 012767 177777 175422
740 002736* 000443
741
742 002740* 122710 000002
743 002744* 001032
744
745 002746* 012704 000016
746 002752* 005744
747 002754* 004767 000714
748
749 002760* 103403

CHKANS: NOP ;EVALUATE THE TOTAL RESPONSE FROM UNIT;
CLR PKCHKD ;# OF PACKS INTO TRANSACTION=0.
MOV #RCVBUF,R0 ;GET BUFFER ADDR.
MOV X#PKNM,R2 ;AND # OF PACKETS EXPECTED
MOV X#FLG,R3 ;ADDRESS OF 1ST EXPECTED FLAG
MOV R3,R1 ;COPY TO R1
ADD #2,P1 ;NOW R1=>X#BCNT FOR 1ST PACKET
MOV R0,PKPTR ;HERE R0=>TOP OF PACKET
INC PKCHKD ;ONE MORE PACK INTO RESPONSE
MOV #R1,RCBCNT ;COPY COUNT
CALL ERRSET ;PREP FOR ERROR
CMPB #R0,R3 ;1ST BYTE=EXPECTED?
BNE 58 ;UH OH...BRANCH
CMPB #R0,#R#CONT ;OK, IS IT 1 BYTE?
BEQ 78 ;YES...ONTO NEXT PACK
MOV RCBCNT,R4 ;NO, SO > 1 BYTE (NEVER EXPECT INIT)
TST -(R4) ;SO THEN USE EXPECTED COUNT
CALL CKCKSM ;ADJUST FROM RECIEVE COUNT TO CHECKSUM CNT
BCC 28 ;CHECK THE CHECKSUM
MSGNS,BEGIN,CHKSMR ;NO CARRY...NO INCORRECT
;ASCII MESSAGE CALL WITH COMMON HEADER
;*****
HDRS,BEGIN,ERRTBL ;CHECKSUM ERROR
;*****
MOV #-1,ERRER ;ERROR FLAG
BR CHKRET ;EXIT
CMPB #R#END,(R0) ;END PAK?
BNE 38 ;NO
CALL CHKSUC ;YES-SEE IF SUCCESS CODE VALID
BR CHKRET ;EXIT
CMPB #R#DATA,#R0 ;DATA PAK?
BNE OTLERR ;NO
CALL COMPAR ;YES-CHECK DATA
BR 78 ;ON TO NEXT PACK

58: NOP ;HERE CHECKS UNEXPECTED RESPONSE
CMPB #R#INIT,#R0 ;INIT?
BNE 68 ;NO
MSGNS,BEGIN,RCINIT ;ASCII MESSAGE CALL WITH COMMON HEADER
;*****
HDRS,BEGIN,ERRTBL ;INIT RECIEVED
;*****
MOV #-1,ERRER ;FATAL
BR CHKRET

68: CMPB #R#END,(R0) ;END?
BNE OTLERR ;NO-OUT TO LUNCH

78: MOV #R#NDSZ,R4 ;YES-TOTAL SIZE MINUS
TST -(R4) ;TWO (THE CHKSUM)
CALL CKCKSM ;CHECK IT

BCS 108 ;OOPS
    
```

```

750 002762* 004767 000140
751 002766* 000427
752
753 002770*
754 002770* 104403 000000* 004556*
755
756 002776* 104405 000000* 000436*
757
758 003004* 012767 177777 175346
759 003012* 000415
760
761 003014* 005302
762 003016* 001413
763 003020* 066700 175300
764 003024* 022121
765 003026* 022323
766 003030* 000654
767
768 003032* 012767 177777 175320
769 003040* 052767 000010 175200
770 003046* 005767 175310
771 003052* 001407
772 003054* 104403 000000* 004616*
773
774 003062* 104405 000000* 000436*
775
776 003070* 000415
777
778 003072* 032767 000010 175146
779 003100* 001411
780 003102* 042767 000010 175136
781 003110* 104403 000000* 004602*
782
783 003116* 104405 000000* 000436*
784
785 003124*
786 003124* 000207

108: CALL CHKSUC ;OK,NOW TEST SUC. CODE
BR CHKRET ;EXIT

108: MSGNS,BEGIN,CHKSMR ;ASCII MESSAGE CALL WITH COMMON HEADER
;*****
HDRS,BEGIN,ERRTBL ;CHECKSUM ERROR
;*****
MOV #-1,ERRER ;FATAL
BR CHKRET ;EXIT

78: DEC R2 ;NO. OF PACKETS LEFT TO CHECK
BEQ CHKRET ;ALL DONE
ADD RCBCNT,R0 ;POINT TO NEXT PACKET
CMP (R1)+,(R1)+ ;POINT TO NEXT EXPECTED COUNT
CMP (R3)+,(R3)+ ;AND EXPECTED FLAG
BR 18 ;TRY ANOTHER,THEY'RE SMALL

OTLERR: MOV #-1,ERRER ;HERE=STRANGE RESPONSE
BIS #BIT3,SYSTAT ;SET O.T.L. FLAG
CHKRET: TST ERPWD ;DLV ERROR?
BEQ 98 ;NO-BRANCH
MSGNS,BEGIN,DLERR ;ASCII MESSAGE CALL WITH COMMON HEADER
;*****
HDRS,BEGIN,ERRTBL ;DLV ERROR
;*****
BR 108 ;EXIT

98: BIT #BIT3,SYSTAT ;WAS THERE OTL ERROR?
BEQ 108 ;NO-EXIT
BIC #BIT3,SYSTAT ;YES-RESET AND PRINT;
MSGNS,BEGIN,OTL ;ASCII MESSAGE CALL WITH COMMON HEADER
;*****
HDRS,BEGIN,ERRTBL ;UNKNOWN FLAG BYTE
;*****

108: RTS PC
    
```

```

;MODULE TO INTERPRET THE SUCCESS CODE OF AN END PACKET
787
788
789 003126 000240 CHKSUC: NOP ;SEE IF VALID SUCCESS CODE:
790 003130 010067 175144 MOV R0,PKPTR ;COPY TOP OF PACKET POINT
791 003134 116067 000003 175134 MOVR 3(R0),SUCCS ;GET SUCCESS CODE
792 003142 122760 000000 000003 CMPR #EOK,3(R0) ;GOOD SUCCESS?
793 003150 001561 BEQ 126 ;YES-EXIT
794 003152 004767 000434 CALL ERRSET ;POTENTIAL HARD ERROR PREP.
795
796 003156 122760 000001 000003 CMPB #ESTRY,3(R0) ;OK BUT RETRIES?
797 003164 001007 BNE 208 ;NO
798 003166 104403 000000 004654 MSGN$,BEGIN,RTY ;ASCII MESSAGE CALL WITH COMMON HEADER
799
800 003174 104406 000000 000436 $OFER$,BEGIN,ERRTBL ;RETRIES OCCURRED
801
802 003202 000544 BR 128 ;YES-EXIT
803 003204 122760 177737 000003 208: CMPB #ENOMO,3(R0) ;NO MOTOR?
804 003212 001010 BNE 18 ;NO...
805 003214 104403 000000 004702 MSGN$,BEGIN,NOMOT ;ASCII MESSAGE CALL WITH COMMON HEADER
806
807 003222 104405 000000 000436 HRDR$,BEGIN,ERRTBL ;"NO MOTOR" ERROR
808
809 003230 004767 175464 CALL DROP ;FATAL-COULD UNWIND CASSETTE
810 003234 122760 177757 000003 18: CMPB #ECSK,3(R0) ;CHECKSUM ERROR?
811 003242 001007 BNE 138 ;NO
812 003244 104403 000000 004632 MSGN$,BEGIN,DACHK ;ASCII MESSAGE CALL WITH COMMON HEADER
813
814 003252 104405 000000 000436 HRDR$,BEGIN,ERRTBL ;READ RETRIES FAILED
815
816 003260 000515 BR 128 ;YES-EXIT
817 003262 032767 000002 174756 138: BIT #BIT1,SYSTAT ;SELF TEST?
818 003270 001413 BEQ 38 ;NO
819
820 003272 105760 000003 TSTB 3(R0) ;NEGATIVE IF FAIL
821 003276 100106 BPL 128 ;OK
822 003300 104403 000000 004624 MSGN$,BEGIN,SLFOTL ;ASCII MESSAGE CALL WITH COMMON HEADER
823
824 003306 104405 000000 000436 HRDR$,BEGIN,ERRTBL ;SELF TEST FAIL
825
826 003314 004767 175400 CALL DROP ;FATAL UNEQUIVICALLY
827
828 003320 122760 177740 000003 30: CMPB #ESK,3(R0) ;SEEK ERROR?
829 003326 001007 BNE 48 ;NO
830 003330 012767 000051 174550 MOV #51,ERRTYP ;YES.
831
832 003336 104405 000000 000436 HRDR$,BEGIN,ERRTBL ;SEEK ERROR
833
834 003344 000460 BR 108 ;EXIT-ERROR
835 003346 122760 177767 000003 48: CMPB #ENCRT,3(R0) ;NO CART?
836 003354 001007 BNE 58 ;NO
837 003356 104403 000000 004516 MSGN$,BEGIN,NOCART ;ASCII MESSAGE CALL WITH COMMON HEADER
838
839 003364 104405 000000 000436 HRDR$,BEGIN,ERRTBL ;NO MEDIA
840
841 003372 000445 BR 108 ;EXIT -ERROR
842

```

```

843 003374 122760 177720 000003 58: CMPB #ECMD,3(R0) ;NO UNDERSTAND HOST?
844 003402 001436 BEQ 118 ;YES
845 003404 122760 177770 000003 68: CMPB #ENONX,3(R0) ;NON EXISTENT UNIT
846 003412 001432 BEQ 118 ;YES
847 003414 122760 177765 000003 78: CMPB #EWLOC,3(R0) ;WRITE LOCKED
848 003422 001007 BNE 88 ;NO
849 003424 104403 000000 004530 MSGN$,BEGIN,WRPRO ;ASCII MESSAGE CALL WITH COMMON HEADER
850
851 003432 104405 000000 000436 HRDR$,BEGIN,ERRTBL ;WRITE LOCKED
852
853 003440 000422 BR 108 ;EXIT-ERROR
854 003442 122760 177776 000003 88: CMPB #E$PART,3(R0) ;PARTIAL OP?
855 003450 001413 BEQ 118 ;YES
856 003452 122760 177711 000003 98: CMPB #E$REC,3(R0) ;WRONG RECORD
857 003460 001407 BEQ 118 ;EXIT
858 003462 104403 000000 004542 MSGN$,BEGIN,SUCOTL ;ASCII MESSAGE CALL WITH COMMON HEADER
859
860 003470 104405 000000 000436 HRDR$,BEGIN,ERRTBL ;UNKNOWN SUCCESS CODE
861
862 003476 000403 BR 108 ;EXIT -ERROR
863
864 003500 104403 000000 004666 MSGN$,BEGIN,BDSUC ;ASCII MESSAGE CALL WITH COMMON HEADER
865 003506 012767 177777 174644 108: MOV #-1,ERRER ;OPERATION FATAL
866 003514
867 003514 000207 RTS PC

```

```

868 ;CHKSUM / FORM THE PACKET CHECKSUM
869 ;++
870 ; ENTER;
871 ; R0 POINTS TO PACKET
872 ; P1 = BYTES IN PACKET
873 ;
874 ;
875 ; EXIT;
876 ; R1 = CHECKSUM
877 ; R0 POINTS TO ADDRESS FOR PACKET CHECKSUM
878 ;==
879
880 003516*          CHKSUM:
881 003516* 010346      MOV    R3, -(SP)
882 003520* 010246      MOV    R2, -(SP)
883 003522* 042767 000001 174516  BIC    #R10,SYSTAT ;"CHECKSUM IS ODD" BIT
884 003530* 032701 000001          BIT    #R10,R1 ;AN ODD # OF BYTES?
885 003534* 001403          BEQ    16 ;NO
886 003536* 052767 000001 174502  BIS    #R10,SYSTAT ;JUST USED AS TEMP STORAGE
887
888 003544* 006001          18:   ROR    R1 ;/2 FOR WORDS
889
890 003546* 005003          20:   CLR    R3
891
892 003550* 062003          38:   ADD    (R0)+,R3 ;FORM SUM
893 003552* 005503          ADC    R3 ;WITH CARRY
894 003554* 005301          DEC    R1 ;ONE LESS WORD
895 003556* 001374          BNE    38 ;BR IF MORE LEFT
896
897 003560* 032767 000001 174460  BIT    #R10,SYSTAT ;WAS IT ODD
898 003566* 001405          BEQ    48 ;NO
899 003570* 112002          MOVB  (R0)+,R2 ;YES-COPY TO R2
900 003572* 042702 177400          BIC    #177400,R2 ;SIGN UN-EXTEND
901 003576* 060203          ADD    R2,R3 ;ADD LAST BYTE
902 003600* 005503          ADC    R3 ;AND CARRY JUST IN CASE
903
904 003602* 010301          48:   MOV    R3,R1 ;RETURN IT IN CORRECT PLACE
905 003604* 012602          MOV    (SP)+,R2
906 003606* 012603          MOV    (SP)+,R3
907 003610* 000207          RTS    PC

```

```

908 ;MODULE TO FILL ERRRTBL WITH EXTENDED ERROR INFORMATION
909
910 003612* 016767 174416 174260  ERRSET: MOV    RCSR,CSRA ;COPY CSR
911 003620* 017767 174410 174254      MOV    #RCSR,ACSR ;AND CONTENTS
912 003626* 016767 174444 174250      MOV    SUCCS,ACSTAT ;AND SUCCESS IF APPLICABLE
913 003634* 017767 174442 174514      MOV    #R0PKPTR,ERRFLG ;COPY EXPECTED FLAGBYTE
914 003642* 017767 174432 174514      MOV    #R0PKPTR,ATPAK ;1ST AND 2ND BYTES OF ACTUAL PACK
915 003650* 062767 000002 174422      ADD    #2,PKPTR ;POINTER INC.
916 003656* 017767 174416 174502      MOV    #R0PKPTR,ATPAK2 ;COPY 3RD AND 4TH BYTE
917 003664* 162767 000002 174406      SUB    #2,PKPTR ;CORRECT POINTER
918 003672* 000207          RTS    PC

```

```

919 ;MODULE TO CHECK THE CHKSUM OF A RECEIVED PACKET
920
921 ;++
922 ; ENTER:
923 ; R4 = PACKET BYTE COUNT
924 ; R0 POINTS TO TOP OF PACKET
925 ;
926 ; CALLS: "CHKSUM"
927 ; TO FORM CHECKSUM,
928 ;
929 ; EXIT:
930 ; CARRY SET = ERROR
931 ; RESET = NO ERROR
932 ;--
933
934 CKCKSM:
935     MOV     R1,=(SP)
936     MOV     R0,=(SP)
937     MOV     R4,R1
938     CALL    CHKSUM
939 ;HERE R0 --> XMITTED CHKSUM, R1=CHKSUM CALC'D
940
941
942     CMPB   (R0)+,R1
943     BNE    Z8
944
945     SWAB   R1
946
947     CMPB   (R0)+,R1
948     BNE    Z8
949     CLC
950     BR     Z8
951     SEC
952
953     MOV     (SP)+,R0
954     MOV     (SP)+,R1
955     RTS    PC

```

```

956 ;COMPAR / COMPARE PATTERN WRITTEN TO DATA RECV'D
957
958 COMPARI:
959     MOV     R0,=(SP)
960     MOV     R4,=(SP)
961     MOV     R1,=(SP)
962     CLR     BDBYTS
963     MOV     R0,PKPTR
964     MOV     PKPTR,R4
965     INC     R4
966     MOV     R4,R1
967     BIC     #177400,R1
968
969     INC     R4
970     CMPB   PATRN,(R4)+
971     BEQ    Z8
972     TST    BDBYTS
973     BNE    Z8
974     DEC     R4
975     MOV     R4,ANAS
976     MOV     R4,WASADR
977     INC     R4
978     INC     BDBYTS
979     DEC     R1
980     BNE    Z8
981     TST    BDBYTS
982     BEQ    Z8
983     MOV     R4,ACSR
984     MOV     #PATRN,SBADR
985     MOV     PATRN,ASE
986
987     DATER$,BEGIN
988     ;*****
989     ;*****
990     ;*****
991     ;*****
992     BTOD$,BEGIN,BDBYTS,A$BDBT
993
994     MSGN$,BEGIN,BADBY
995
996     MOV     (SP)+,R1
997     MOV     (SP)+,R4
998     MOV     (SP)+,R0
999     PTS    PC

```

```

1001          ,SBTTL MESSAGE FORMAT
1002
1003 004112* 052045 032525 020070 MES1: ,ASCIZ 'RTU58 CONTROLLER '
1004 004120* 047503 052116 047522
1005 004126* 046114 051105 000040
1006 004134* 042040 044522 042526 MES2: ,ASCIZ ' DRIVE '
1007 004142* 000040
1008 004144* 042045 053114 042440 MES3: ,ASCIZ '%DLV ERROR %'
1009 004152* 051122 051117 022440
1010 004160* 000
1011 004161* 045 042523 043114 MES4: ,ASCIZ '%SELF TEST FAIL%'
1012 004166* 052040 051505 020124
1013 004174* 040506 046111 020045
1014 004202* 000
1015 004203* 045 042522 042101 MES5: ,ASCIZ '%READ RETRYS ON OP,%'
1016 004210* 051040 052105 054522
1017 004216* 020123 047117 047440
1018 004224* 027120 000045
1019 004230* 041445 042510 045503 MES7: ,ASCIZ '%CHECKSUM ERROR%'
1020 004236* 052523 020115 051105
1021 004244* 047522 022522 000
1022 004251* 045 042522 042503 MES8: ,ASCIZ '%RECEIVING INIT REQUEST%'
1023 004256* 053111 047111 020107
1024 004264* 047111 052111 051040
1025 004272* 050505 042525 052123
1026 004300* 000045
1027 004302* 051045 052105 044522 MES9: ,ASCIZ '%RETRIES FAILED ON OP,%'
1028 004310* 051505 043040 044501
1029 004316* 042514 020104 047117
1030 004324* 047440 027120 020045
1031 004332* 000
1032 004333* 045 047516 046440 MES10: ,ASCIZ '%NO MOTOR ACTION%'
1033 004340* 052117 051117 040440
1034 004346* 052103 047511 022516
1035 004354* 000
1036 004355* 045 047516 041440 MES11: ,ASCIZ '%NO CARTRIDGE%'
1037 004362* 051101 051124 042111
1038 004370* 042507 000045
1039 004374* 053445 044522 042524 MES12: ,ASCIZ '%WRITE PROTECTED%'
1040 004402* 050040 047522 042524
1041 004410* 052103 042105 000045
1042 004416* 052045 052117 046101 MES13: ,ASCIZ '%TOTAL BAD BYTES = '
1043 004424* 041040 042101 041040
1044 004432* 052131 051505 036440
1045 004440* 000040
1046 004442* 051440 041525 042503 MES18: ,ASCIZ ' SUCCESS CODE%'
1047 004450* 051523 041440 042117
1048 004456* 022505 000
1049 004461* 040 046106 043501 MES19: ,ASCIZ ' FLAG BYTE%'
1050 004466* 041040 052131 022505
1051 004474* 000
1052 004475* 045 000 MES20: ,ASCIZ '% '
1053 004477* 045 047125 047113 MES21: ,ASCIZ '%UNKNOWN '
1054 004504* 053517 020116 000
1055 004511* 102 042101 000040 MES22: ,ASCIZ '%BAD '
1056          ,EVEN
  
```

```

1057          ,SBTTL MESSAGE CONSTRUCTS
1058
1059 004516* 004112* NOCART: MES1 ;CARTRIDGE NOT INSERTED
1060 004520* 004134* MES2
1061 004522* 000252* A&DRV
1062 004524* 004355* MES11
1063 004526* 177777 -1
1064
1065 004530* 004112* WRPRO: MES1 ;CARTRIDGE WRITE PROTECTED
1066 004532* 004134* MES2
1067 004534* 000252* A&DRV
1068 004536* 004374* MES12
1069 004540* 177777 -1
1070
1071 004542* 004112* SUCOTL: MES1 ;COULD NOT INTERPRET SUCCESS CODE
1072 004544* 004134* MES2
1073 004546* 000252* A&DRV
1074 004550* 004477* MES21
1075 004552* 004442* MES18
1076 004554* 177777 -1
1077
1078 004556* 004112* CHKSMR: MES1 ;HOST CALCULATED CHECKSUM FOR
1079 004560* 004134* MES2 ;PACKET DOESN'T EQUAL TRANSMITTED ONE.
1080 004562* 000252* A&DRV
1081 004564* 004230* MES7
1082 004566* 177777 -1
1083
1084 004570* 004112* RCINIT: MES1 ;HOST RECEIVED "INIT" REQUEST
1085 004572* 004134* MES2
1086 004574* 000252* A&DRV
1087 004576* 004251* MES8
1088 004600* 177777 -1
1089
1090 004602* 004112* OTL: MES1 ;1ST BYTE OF PACKET UNINTERPRETABLE
1091 004604* 004134* MES2
1092 004606* 000252* A&DRV
1093 004610* 004477* MES21
1094 004612* 004461* MES19
1095 004614* 177777 -1
1096
1097 004616* 004112* DLERR: MES1 ;OVERRUN OR FRAMING ERROR IN RECEIVE
1098 004620* 004144* MES3
1099 004622* 177777 -1
1100
1101 004624* 004112* SLFOTL: MES1 ;DEVICE'S SELF TEST FAILED BUT COULD
1102 004626* 004161* MES4 ;STILL SEND AN INTELLIGENT PACKET (I)
1103 004630* 177777 -1
1104
1105 004632* 004112* DACHK: MES1 ;DEVICE RETRIED OPERATION 8 TIMES
1106 004634* 004134* MES2 ;AND FAILED
1107 004636* 000252* A&DRV
1108 004640* 004302* MFS9
1109 004642* 177777 -1
1110
1111 004644* 004416* BADRY: MES13 ;MESSAGE FOR THE TOTAL INCORRECT
1112 004646* 000262* A&BDBT ;DATA BYTES PER PACKET.
  
```


1113	004650	004475	MES20	
1114	004652	17777	-1	
1115				
1116	004654	004112	RTRY: MES1	;DEVICE RETRIED OPERATION SUCCESSFULLY
1117	004656	004134	MES2	
1118	004660	000252	A&DRV	
1119	004662	004203	MES5	
1120	004664	17777	-1	
1121				
1122	004666	004112	BDSUC: MES1	;SUCCESS CODE VALID BUT OPERATION FAILED
1123	004670	004134	MES2	
1124	004672	000252	A&DRV	
1125	004674	004511	MES22	
1126	004676	004442	MES18	
1127	004700	17777	-1	
1128				
1129	004702	004112	NOMOT: MES1	;DEVICE COULDN'T GET ANY MEANINGFUL SIGNAL
1130	004704	004134	MES2	FROM TAPE
1131	004706	000252	A&DRV	
1132	004710	004333	MES10	
1133	004712	17777	-1	

1134			.SBTTL	BUFFER AREAS:	
1135					
1136	004714	000204	TRBUF:	.BLKB 132.	;DATA PACK SIZE MAX.
1137	005120	000000		.WORD 0	
1138					
1139	005122	001036	RCVBUF:	.BLKB 4*R&DABZ+R&NDSZ	;4 DATA PACKS AND END
1140	006160	000000	LSTADR:	.WORD 0	
1141		000001	.END		

SR3	000022R	188*																		
SR4	000024R	189*																		
START	000500R	192	403*																	
STAT	000026R	191*																		
SUCCS	000276R	264*		453*	791*	912														
SUCCTL	004542R	858	1071*																	
SVR0	000052R	206*																		
SVR1	000054R	207*																		
SVR2	000056R	208*																		
SVR3	000070R	209*																		
SVR4	000072R	210*																		
SVR5	000074R	211*																		
SVR6	000076R	212*																		
SISCNT	000052R	201*																		
SISINI	001072R	414*	475*																	
SISSET	001246R	494	496	502*																
SYSTAT	000246R	249*	434*	507*	569*	572*	769*	778	780*	817	883*	886*	897							
TBLPTR	000306R	268*	432*	502*	510	512*	516	531	544	548	570	577	594	595*						
TBLTOP	000304R	267*	432	491*	500*	502	512													
TRBUF	004714R	458	515	518	528	1136*														
TRPDFD	000072	233*																		
VECTOR	000010R	182*	475																	
WASADR	000104R	216*	976*																	
WDPR	000116R	223*																		
WDTO	000114R	222*																		
WRPRO	004530R	849	1065*																	
XFLAG	000005R	180*																		
XMDB	000242R	243*	488*	601*	654*	665*														
XMSR	000240R	242*	445*	459*	486*	606*	645*	647*	652	657*	663									
XSCNT	000332R	278*	396	454	534*	538*	574*	580*												
XFLG	000330R	277*	395	456	533*	537*	573*	579*	584	700										
X&PKNM	000326R	276*	394	455	535*	539*	575*	581*	592*	699										
X&PTR	000302R	266*	456*	620	632*	633	634*	913												
.	006162R	254*	258*	279*	1136*	1139*														

. ABS. 000000 000
 006162 001

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

XTUAA0,XTUAA0/SOL/CRF:SYM=DDXCOM,XTUAA0
 RUN-TIME: 5 9 1 SECONDS
 RUN-TIME RATIO: 35/17=2.0
 CORE USED: 7K (13 PAGES)