

.REM 8

IDENTIFICATION

PRODUCT CODE: AC-E721H-MC

PRODUCT NAME: CXDJAH0 DEC/X11 DJ11 MODULE

DATE: FEB 1979

MAINTAINER: DEC/X11 SUPPORT GROUP

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITALS COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE OR EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1973,1979 DIGITAL EQUIPMENT CORPORATION

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39

40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93

1. ABSTRACT

THIS MODULE EXERCISES ANY COMBINATION OF ONE TO SIXTEEN LINES ON ANY COMBINATION OF DEVICES. TO FOUR DJ11 ASYNCHRONOUS TRANSMISSIONS ON EACH LINE. DJ11 UNDER INTERRUPT BUFFER. THE DATA FOR 8-BIT SUBSECTS OF THE SELECTED LINES. DJ11 UNABLE TO RECEIVE TRANSMISSIONS. THE FULL LINE WHEN DISABLED. THE LINES ARE ACTIVELY PAUSED. EACH PASS IS EXERCISED AND WILL BE EXERCISED BELOW FOR NECESSARY CHANGES REQUIRED FOR OTHER OPERATIONS. WATCHDOG TIMER IS ALSO RUN BY THE MODULE TO ALLOW DETECTION OF THE "HUNG" CONDITION (AN EXPECTED INTERRUPT THAT FAILED TO OCCUR). THE "HUNG" TIME IS SET TO RUN LONGER THAN THE LONGEST POSSIBLE MODULE PASS TIME, TO PREVENT A REAILY LOADED SYSTEM WHEN THE DEVICE TAKES A LONG TIME TO TIMEOUT (I.E. 10-15 MINUTE MESSAGE OUTPUT AND HUNG MODULE IS DETECTED, THIS PREVENTS A "LOST" INTERRUPT. PERMANENTLY UNABLE TO RELOCATE DUE TO A "LOST" INTERRUPT.

2. REQUIREMENTS

HARDWARE: DJ11 SIXTEEN LINE ASYNCHRONOUS COMMUNICATION DEVICE.
STORAGE: DJ11 REQUIREMENTS:
1. OCTAL WORDS: 1154
2. OCTAL BYTES: 4202
3. OCTAL BYTES: 4405

3. PASS DEFINITION

ONE PASS OF THE DJA MODULE CONSISTS OF TRANSMITTING, RECEIVING, AND CHECKING 8320 CHARACTERS PER DJ11.

4. EXECUTION TIMES

Worst Case (1 LINE ONLY ON EACH OF 4 DJ11'S AT 75 BAUD) TAKES APPROXIMATELY 3 MINUTES PER PASS. RUNNING ALL LINES (DEFAULT CASE) WILL CAUSE SEVERAL PASSES PER MINUTE TO BE COMPLETED.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:
DEVICE COUNT(DVID1): 1

REQUIRED PARAMETERS:
FIRST DEVICE ADDRESS: THE FIRST DEVICE REGISTER ADDRESS

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
145
146
147
148
149
150
151

MUST BE SPECIFIED.
VECTOR: THE VECTOR ADDRESS OF THE FIRST DJ11 MUST BE GIVEN
DVID1: IF MORE THAN 1 UNIT IS BEING RUN, VALUE MUST BE GIVEN
(4 UNITS MAX)

6. DEVICE/OPTION SETUP

ALL SIXTEEN LINES OPERATIONAL (DEFAULT) OR ALL SELECTED LINES OPERATIONAL
(IF NOT ALL LINES OPERATED).

7. MODULE OPERATION

A. INITIALIZE FIRST VALUES TO BE TRANSMITTED PRO EACH LINE
(SET TO ZERO).

B. INITIALIZE INTERNAL QUEUES AND PARAMETERS, SETUP LINKAGE OF DEVICE
VECTORS TO LINKAGE TABLE, AND INITIALIZE SELECTED DJ11-S
VIA SETTING "MOS CLEAR".

C. SET TRANSMITTER INTERRUPT ENABLE AND SILO FULL INTERRUPT ENABLE,
THEN SET TRANSMITTER ENABLE FOR ALL SELECTED LINES ON EACH
SELECTED DJ11. COUNT DJ11-S SELECTED AS THEY GET TURNED ON.

D. INITIALIZE WATCHDOG TIMER, WHICH RUNS VIA A SET OF "BREAK"
LOOPS. THE LOOP MONITORS THE COUNT OF ACTIVE DJ11-S. IF THE DJ11
COUNT GOES TO ZERO BEFORE THE TIMER TIMES OUT, AN END OF PASS CALL
IS MADE. (WHICH WILL EVENTUALLY OUT TO BE RESTARTED
AT STEP 8).
COUNT GOES TO ZERO BEFORE THE DJ11 ACTIVE
COUNT CODES TO ZERO. A "DEVICE HUNG" MESSAGE IS OUTPUT AND
THE MODULE IS DROPPED.

E. WHEN A TRANSMIT INTERRUPT OCCURS, TRANSMIT INTERRUPT ENABLE IS
TURNED OFF FOR THAT DJ11. UNIT IDENTIFICATION IS SAVED IN THE
MODULE. TRANSMIT Q-WORDS, AND THE MODULE Q-WORDS. WHEN THE MONITOR
RETURNS TO THE LINE, TRANSMIT Q-WORDS, AND THE CHARACTER BEING TRANSMITTED
WAS SAID TO OCCUR. THEN TRANSMIT Q-WORDS, AND THE CHARACTER BEING TRANSMITTED
RUN UNTIL ALL OTHER LINES HAVE BEEN DISABLED FROM BEING TRANSMITTED
(ACTUALLY, ON LINES DISABLED UNTIL THE NEXT 65 CHARACTER
HAVE BEEN TRANSMITTED). IF ALL LINES ARE DISABLED, CHARACTER
TRANSMITTER SERVICE IS COMPLETED. TRANSMITTER SERVICE THEN EXITS.

F. WHEN 65 CHARACTERS HAVE BEEN TRANSMITTED ON A GIVEN DJ11, THE
RECEIVER BUFFER (SILO) IS FULL. INTERRUPT INTERRUPT Q-WORDS. A
MODULE IDENTIFICATION Q-WORD IS RECEIVED. THE PIPELINES WHEN THE
UNIT FOR RECEIVING THE CHARACTER IS RECEIVED. THE INTERRUPT Q-WORD
THE COUNT BUFFER AND CHECKS THE SILO. THE INTERRUPT Q-WORD
SOFTWARE BUFFER AND CHECKS THE SILO. THE INTERRUPT Q-WORD
TRANSMITTER SERVICE IS COMPLETED. TRANSMITTER SERVICE THEN EXITS.

152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206

DECREMENTED. IF NOT ALL DONE THE RECEIVER DATA TABLE IS RESYNCD TO MATCH THE TRANSMITTER TABLE AND THE DJII IS REENABLED TO TRANSMIT ANOTHER CHARACTER. WHEN THE DJII ACTIVE COUNT GOES TO 0 THE WATCHDOG TIMER WILL DETECT IT AND ISSUE AN END OF PASS (SEE STEP).

THE SELECTED LINES GET REENABLED. IF 65 (DECIMAL) CHARACTERS

8. OPERATION OPTIONS

- A. TO MAKE THE TIME INTERVAL BETWEEN ENDPAS STATEMENTS LONGER OR SHORTER, EITHER INCREASE OR DECREASE THE "ENDSEL" WORD.
- B. TO EXERCISE A CONFIGURATION OF LINES OTHER THAN ALL SIXTEEN AT ONCE, CHANGE THE APPROPRIATE BITS OF THE "XMTSEL" WORD FOR THE SELECTED DJII AND THE SAME BITS OF THE "RMTSEL" WORD. A ONE INDICATES THAT THE LINE IS TO BE EXERCISED, A ZERO INDICATES THAT THE LINE IS OFF.

C. MEANING OF SRI

- 1. DATA SET BIT 1 (SRI=000004)
DATA SET BIT 2 (SRI=000002)
DATA SET BIT 3 (SRI=000001)
DATA SET BIT 4 (SRI=000000)
- 2. CHECKSUM COUNT IS USED TO DETERMINE SRI SETTING
ITERATION COUNT IS USED TO DETERMINE SRI SETTING
FOR 75 BAUD SET BIT 5
FOR 114.5 BAUD SET BIT 6
FOR 150 BAUD SET BIT 7
FOR 300 BAUD SET BIT 8
FOR 600 BAUD SET BIT 9
FOR 1200 BAUD SET BIT 10
FOR 1600 BAUD SET BIT 11
FOR 2400 BAUD SET BIT 12
FOR 4800 BAUD SET BIT 13
FOR 9600 BAUD SET BIT 14
OR NO BITS (5-15=0)

9. NON STANDARD PRINTOUTS

THERE ARE 2 SPECIAL MESSAGES WHICH MAY BE PRINTED OUT. THEY ARE:

"UNIT DROPPED" - THIS INDICATES THAT A DJII WHICH HAS JUST PREVIOUSLY REPORTED A FAILURE HAS BEEN DROPPED. DJII'S ARE DROPPED IF BUSY CLEAR FAILS TO CLEAR WHEN THEY ARE INITIALIZED.

"UNIT HUNG" - THIS INDICATES THAT THE MODULE'S WATCHDOG TIMER HAS TIMED OUT BEFORE THE DJII DEVILOUS ACTIVE COUNT WENT TO ZERO. THIS OCCURS WHEN AN EXPECTED TRANSMITTER OR RECEIVER INTERRUPT FAILS TO OCCUR ON ONE OR MORE DJII'S.

```

208 000000-
209 000000-
210
211
212
213
214
215
216 000000- 045104 044101 040
217 000000-
218 000000-
219 000000-
220 000000-
221 000000-
222 000000-
223 000000-
224 000000-
225 000000-
226 000000-
227 000000-
228 000000-
229 000000-
230 000000-
231 000000-
232 000000-
233 000000-
234 000000-
235 000000-
236 000000-
237 000000-
238 000000-
239 000000-
240 000000-
241 000000-
242 000000-
243 000000-
244 000000-
245 000000-
246 000000-
247 000000-
248 000000-
249 000000-
250 000000-
251 000000-
252 000000-
253 000000-
254 000000-
255 000000-
256 000000-
257 000000-
258 000000-
259 000000-
260 000000-
261 000000-
262 000000-
263 000000-

```

IOMOD <DJAH> 1,165,1,0,36
MODULE 140000 DJAH 1,165,1,0,36
DJAH DEC/X11 SYSTEM EXERCISER MODULE
VERSION 6 23-MAY-78
LIST BIN

REGN: *****
MODNAM: .ASCII (DJAH / ;MODULE NAME
XFLAG: .BYTE OPEN ;USED TO KEEP TRACK OF WBUFF USAGE
ADDR: 1=0 ;1ST DEVICE ADDR
VECTOR: 1=0 ;1ST DEVICE VECTOR.
BR1: .BYTE PRY5+0 ;1ST BR LEVEL.
BR2: .BYTE PRY+0 ;2ND BR LEVEL.
NTR101: ***** ;SWITCH REGISTER 1.
SR1: OPEN ;SWITCH REGISTER 2
SR2: OPEN ;SWITCH REGISTER 3
SR3: OPEN ;SWITCH REGISTER 4
SR4: OPEN ;SWITCH REGISTER 5

STAT: 140000 ;STATUS WORD
INIT: START ;MODULE START ADDR.
SPDINT: MODSP ;MODULE STACK POINTER.
PASCNT: 0 ;PASS COUNTER.
ICOUNT: 0 ;# OF ITERATIONS PER PASS=0
SOPCNT: 0 ;LOC TO COUNT ITERATIONS
HRDCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
SOPPAS: 0 ;LOC TO SAVE TOTAL HARD ERRORS
VSCNT: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
RANNUM: 0 ;LOC TO SAVE HARD ERRORS PER PASS
CONFIC: ***** ;% OF SVS ERRORS ACCUMULATED
RESA: 0 ;RESERVED FOR MONITOR USE
RESB: 0 ;RESERVED FOR MONITOR USE
RESR: 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
ASAT: OPEN ;STATUS REG CONTENTS.
ERRTP: ***** ;TYPE OF ERROR
ASB: OPEN ;EXPECTED DATA.
AWAS: ***** ;ACTUAL DATA.
RSTR: RSTRPT ;RESTART ADDRESS AFTER END OF PASS
WDT0: OPEN ;WORDS TO MEMORY PER ITERATION
WDT1: OPEN ;WORDS FROM MEMORY PER ITERATION
INTR: OPEN ;% OF INTERRUPTS PER ITERATION
IDNUM: 36 ;MODULE IDENTIFICATION NUMBER=36
.REPT SPSIZ ;MODULE STACK STARTS HERE.

```

264
265
266
267
268 000224-
269
270
271
272
273 000224- 004567 003276
274 000230- 004567
275 000236- 000800 002734
276 000240- 004567 003262
277 000244- 000010
278 000246- 004567 002720
279 000250- 004567
280 000254- 004567 003246
281 000260- 000020
282 000262- 004567 002704
283 000266- 000020
284 000270- 004567 003232
285 000274- 004567
286 000276- 004567 002670
287 000302- 000030
288
289 000000-
290 000000-
291 000304- 000040
292
293 000404- 000100
294
295 000604- 000404
296
297 001614- 000004
298
299 001624- 000004
300
301 001634- 000020
302 001634- 000020
303 001676- 000000
304
305 001700- 000101 000101 000101
306 001706- 000101
307
308
309
310 001710- 000101 000101 000101
311 001716- 000101
312
313
314
315 001720- 000200 000200 000200
316 001726- 000200
317
318
319 001730- 177777 177777 177777 177777

```

.NLIST
.WORD 0
.LIST
.ENDR
MODSP: *****
;*****
;SOME VARIABLES AND CONSTANTS UNIQUE TO THIS ROUTINE
DJLINK: JSR R5,RCVINT ;UNIT 0 RECEIVER
JSR R5,XMTINT ;UNIT 0 TRANSMITTER
JSR R5,RCVINT ;UNIT 1 RECEIVER
JSR R5,XMTINT ;UNIT 1 TRANSMITTER
JSR R5,RCVINT ;UNIT 2 RECEIVER
JSR R5,XMTINT ;UNIT 2 TRANSMITTER
JSR R5,RCVINT ;UNIT 3 RECEIVER
JSR R5,XMTINT ;UNIT 3 TRANSMITTER
OPEN=0
XMTTBL: .BLKW 32. ;A TABLE FOR RECORDING CHARACTERS WHICH HAVE
;BEEN TRANSMITTED
RCVTBL: .BLKW 64. ;A TABLE FOR RECORDING CHARACTERS WHICH YOU
;EXPECT TO RECEIVE
SILO: .BLKW 260. ;A BUFFER AREA TO STORE THE DATA RECORDED IN
;THE DJ'S RECEIVER SILO
XQ: .BLKW 4. ;A QUEUE FOR SAVING THE DEVICE OFFSET WHILE A
;TRANSMIT INTERRUPT IS IN "PIRQ"
RQ: .BLKW 4. ;A QUEUE FOR SAVING THE DEVICE OFFSET WHILE A
;RECEIVER INTERRUPT IS IN "PIRQ"
ERRQ: .BLKW 16.
ERRQ: OPEN
ERRQ: OPEN
XCNT: .WORD 101,101,101,101 ;NUMBER OF CHARACTERS LEFT TO BE TRANSMITTED DURING
;THIS INTERRUPT. A WORD FOR EACH DEVICE
RCNT: .WORD 101,101,101,101 ;NUMBER OF CHARACTERS TO BE PROCESSED BY THE
;RECEIVE ROUTINE. A WORD FOR EACH DEVICE
ENDCNT: .WORD 200,200,200,200 ;NUMBER OF "65 CHARACTER CHUNKS"
;TO BE PROCESSED BY THE MODULE PER PASS
XMTSEL: .WORD 177777,177777,177777,177777 ;LINES TO BE RUN

```

320 001736 177777
321 001740 177777 177777 177777 XMITEN: .WORD 177777,177777,177777,177777 ;WORD USED TO FILL TRANSMITTER COMMAND
322 001746 177777 ;REGISTER 16XXX2, CORRESPONDS TO
323 ;A ONE FOR EACH ACTIVE LINE
324
325 001750 000000 000000 000000 XNITSV: .WORD 0,0,0,0 ;SAVE LINES ACTIVATED IN CASE OF HANG
326 001756 000000
327
328
329
330
331
332
333 001760 001614 XPNTR: XQ ;POINTERS USED FOR THE BOOKKEEPING OF
334 001762 001614 XPNTR: XQ ;THE TRANSMIT QUEUE
335 001764 001624 XPNTR: XQ ;POINTERS USED FOR THE BOOKKEEPING OF
336 001766 001624 XPNTR: XQ ;THE RECEIVE QUEUE
337
338
339
340 001770 000000 I: 000 ;TEMPORARY STORAGE LOCATION USED DURING RECEIVE DATA CHE
341 001772 000000 SELMKT: 1 ;MASK TO CHECK FOR UNIT SELECTED
342 001774 000000 SELRPT: 0
343 001776 100000 EXPLIN: 100000 ;USED DURING INITIALIZATION, THE ONE REPRESENTS THE
344 ;DONE BIT IN THE EXPECTED RECEIVE DATA TABLE
345 002000 000200 ENDSSEL: 200 ;REFILL OF "ENDSEL" WORD AFTER ENDPASS
346 002002 000191 CNTSEL: 101 ;REFILL OF "RCNT" AND "XCNT" AFTER AN INTERRUPT SERVICE
347 ;BY VARIABLE SINCE 65. CHARS READ TO GET SILO FULL INT
348 ;HOLDS VECTOR ADDRESS DURING SETUP ERROR PRINTOUT
349
350 002004 000000 VCT: OPEN ;HOLDS LINK ADDRESS DURING SETUP ERROR PRINTOUT
351 002006 000000 LINK: OPEN
352 002010 000000 SCRSR: OPEN
353 002014 000000 ENDFLG: OPEN
354 002016 000000 TIME: 110 ;NUMBER OF DJ11'S STILL RUNNING
355 002020 000000 TIME: 0 ;WATCHDOG TIMER OUTER LOOP VALUE
356 002022 000000 MASK: 0
357 002024 000000 CHAR: 0
358
359
360 002026 017000 TMBTBL: 17000 ;DATA BIT LENGTH (5-9) MASK
361 002030 011700 ;TEMP HOLDING FOR MASK & ITERATION COUNT &
362 002032 006000 ;DATA COMPRESSION DURING TEST.
363 002034 006000 ;8500 BPS VARIABLE
364 002036 006000 ;4800 BPS ITERATION
365 002038 006000 ;2400 BPS COUNTS
366 002040 006000 ;1800 BPS PER
367 002042 006000 ;1200 BPS LINE
368 002044 006000 ;900 BPS SPEEDS
369 002046 006000 ;350 BPS
370 002048 006000 ;300 BPS
371 002050 006000 ;134.5 BPS
372 002052 006000 ;75 BPS
373
374
375 ;MODULE INITIALIZATION--THIS ROUTINE INITIALIZES THE DJ11 ASYNCHRONOUS
376 ;XMITTER AND RECEIVER, POINTING INTERRUPT VECTORS TO INTERRUPT HANDLERS (VIA A
377 ;LINKAGE TABLE) AND INITIALIZING PARAMETERS AND CONTROL REGISTERS
378 START: MOV #65, INTR ;65 INTERRUPTS/ITERATION
379 MOV #64, WDT0 ;64 WORDS TO MEM/ITERATION
380 MOV #64, WDFR ;64 WORDS FROM MEM/ITERATION

```

```

376 002076 016767 175712 177670 MOV DVID1, SELECT
377 002078 000304 000304 MOV #XMITBL, R4
378 1$: CMP #R4+, #RCVTBL ;CLEAR XMT TABLE
379 BLD #R4+, #RCVTBL
380 MOV #R4+, #RCVTBL
381 MOV #R4+, #RCVTBL
382 BIC #R4+, #RCVTBL
383 MOV #R4+, #RCVTBL
384 2$: MOV #TMBTBL, R3
385 TST #R3
386 BEQ #R3, #R3 ;IS THIS IT?
387 TST #R3 ;YES LEAVE NOW
388 BR #R3+ ;POINT TO NXT IN TBL
389 MOV #R3, #R3 ;TRY AGAIN
390 3$: MOV #R3, #R3 ;SET UP COUNT
391 CLR #R3
392 MOV #R3, #R3
393 MOV #R3, #R3
394 MOV #R3, #R3
395 ADD #R3, #R3
396 DEC #R3
397 BNE #R3, #R3
398 TST #R3
399 BNE #R3, #R3
400 4$: MOV #R3, #R3
401 MOV #R3, #R3
402 MOV #R3, #R3
403 ADD #R3, #R3
404 DEC #R3
405 BNE #R3, #R3
406 TST #R3
407 BNE #R3, #R3
408 5$: MOV #R3, #R3
409 MOV #R3, #R3
410 MOV #R3, #R3
411 ADD #R3, #R3
412 ADD #R3, #R3
413 ADD #R3, #R3
414 ADD #R3, #R3
415 BR #R3
416 9$: MOV #R3, #R3
417 MOV #R3, #R3
418 MOV #R3, #R3
419 MOV #R3, #R3
420 INCB #R3
421 BIC #R3, #R3
422 CMP #R3, #R3
423 BLD #R3, #R3
424 MOV #R3, #R3
425 TST #R3
426 BEQ #R3, #R3
427 MOV #R3, #R3
428 BIC #R3, #R3
429 ASR #R3
430 ASR #R3
431 BNE #R3, #R3
432 MOV #R3, #R3

```

432 002410- 005767 177360
433 002414- 001787
434 002416- 032767 177760 177350
435 002424- 001402
436 002426-
437 002435- 104419 000000
438 002440- 012767 001614- 177320
439 002446- 012767 001624- 177310
440 002454- 012767 001634- 177304
441 002462- 012767 001634- 177204
442 002470- 012767 001634- 177200
443 002476- 012767
444 002506- 015670 005219-
445 002512- 016700 175270
446 002516- 012767 177246
447 002518- 016700 177242
448 002524- 012767
449 002534- 010120
450 002536- 175250
451 002544- 000006
452 002552- 175234
453 002560- 000006
454 002564- 001334
455 002574- 014704 000012
456 002600- 032715 000020
457 002604- 001456
458 002608- 104407 000000
459 002616- 005307
460 002622- 001367
461 002626- 010167 177160
462 002632- 010567 177152
463 002636- 010567 177136
464 002644- 010567 177130
465 002650- 011567 175226
466 002654- 005915
467 002658- 012767 000016 175222
468 002664- 104405 000000- 000000
469 002672- 104403 000000- 004350
470 002700- 016787 177110 177110
471 002710- 001002 177062
472 002714- 104410 000000-
473 002720- 016700 177052
474 002724- 016700 177054
475 002730- 016705 177054
RESTRT: TST SELECT ;ANY DJ'S SELECTED?
BEQ 15 ;BR IF NO
BIT 15 ;UNITS OTHER THAN 0-3 SELECTED?
BEQ 25 ;NO- OK
1\$: EMVS,BEGIN ;UNIT SELECTION (DVID1) INCORRECT
MOV R2,XQ,XPNW ;SETUP QUEUE POINTERS
MOV R3,XQ,XPNTR
MOV R4,XQ,XPNTR
MOV R5,XQ,XPNTR
MOV R6,XQ,XPNTR
MOV R7,XQ,XPNTR
MOV R8,XQ,XPNTR
MOV R9,XQ,XPNTR
MOV R10,XQ,XPNTR
MOV R11,XQ,XPNTR
MOV R12,XQ,XPNTR
MOV R13,XQ,XPNTR
MOV R14,XQ,XPNTR
MOV R15,XQ,XPNTR
CLR ENDPLG
LDJ LINK,R1 ;GET ADDRESS OF DJ'S LINKING TABLE
MOV VECTOR,R0 ;GET VECTOR ADDRESS
MOV ADDR,R5 ;GET CSR ADDRESS
NXTDEV: BIT SELMSK,SELECT ;HAS THIS DEVICE BEEN SELECTED?
BIT R5 ;NO- BRANCH
MOV R1,(R0)+ ;YES- LOAD RCV INT VECTOR
CLR R1 ;LOAD RCV BR
MOV R1,(R0)+ ;LOAD XMT INT VECTOR
CLR R1 ;LOAD XMT BR
ADD R6,R1
JSR PC,TBLSNL ;SYNCH RCV AND XMT TABLES
MOV R1,(R5) ;CLEAR DJ BUFFER AND QARTS
MOV R2,(R4) ;SETUP WAIT LOOP COUNTER
MOV R3,(R5) ;BUSY CLEAR GONE LOW YET?
BEQ 115 ;YES- BRANCH
BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR...
BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
DEC R4 ;WAIT LOOP COUNTER
MOV R1,LINK ;LOOP IF NOT TIMED OUT
MOV R0,VCT ;SAVE INFO DURING ERROR (CAN BE DONE
MOV R5,SCSR ;SINCE THIS IS A SEQUENTIAL LOOP
BIT SELMSK,SELECT ;AND INFO NOT USED ELSEWHERE)
MOV R6,CSR ;CLEAR UNIT SELECTED BIT - DROP UNIT
MOV R7,ACSR ;SETUP FOR ERROR PRINTOUT
CLR R8 ;DISABLE UNIT
MOV R9,ERRTYP ;BUSY STUCK

HDRS\$,BEGIN,HUNG ;BUSY CLEAR DIDN'T CLEAR IN TIME

MSGNS\$,BEGIN,DROPH ;ASCII MESSAGE CALL WITH COMMON HEADER
;UNIT DROPPED
MOV TIME,TIMER ;RESET TIMER DUE TO PRINTOUT DELAY
TST SELECT ;ANY DJ'S STILL SELECTED?
BEQ 15 ;YES- BRANCH
EMVS,BEGIN ;ALL UNITS DROPPED
12\$: MOV R1,LINK,R1
MOV R2,VCT,R0
MOV R3,SCSR,R5

488 002734- 062705 000010
489 002740- 000433
490 002744- 012715 050405
491 002748- 010503 175032
492 002752- 008203
493 002756- 006203
494 002760- 001730- 001740-
495 002764- 001740- 001750-
496 002768- 001750- 000004-
497 002772- 009369
498 002776- 062705 000010
499 002780- 000406
500 002784- 062701 000014
501 002788- 062700 000010
502 002792- 002489 000010
503 002796- 002489 000010
504 002800- 026727 000020
505 002804- 026727 176732
506 002808- 026630
507 002812- 016787 176744 176744
508 002816- 005067 176742
509 002820-
510 002824- 104407 000000-
511 002828- 104407 000000-
512 002832- 005767 176720
513 002836- 001002
514 002840- 104413 000000-
515 003100- 005367 176714
516 003104- 001364
517 003108- 005367 176704
518 003112- 104403
519 003116- 000000- 004354-
520 003120- 012767 176636
521 003124- 032767 176632 176632
522 003128- 005914
523 003132- 000010
524 003136- 005367 176614
525 003140- 026727 176810 000020
526 003144- 002763
527 003148- 104410 000000-
528 003152-
529 003156-
530 003160-
531 003164-
532 003168-
533 003172- 010246
534 003176- 011502
535 003180- 066702 174604
536 003184- 042702 040000
537 003188- 015744 008046
538 003192- 026727 176540
539 003196- 026727 176534 001624-
540 003200- 001003
541 003204- 012767 001614- 176522
542 003208- 012608
543 003212- 012608
ADD #10,R5
BR #4
MOV R2,#0405,(R5) ;TURN ON RCVR AND XMIT ENABLE
MOV R3,R3
SUB ADDR,R3
ASR R3
MOV XMTSEL(R3),XMITEN(R3) ;GET OFFSET (UNIT # TIMES 2)
MOV XMITEN(R3),XMTSVR(R3) ;SAVE IN CASE OF HANG TO SHOW ACTIVE LINES
MOV XMITEN(R3),4(R5) ;TURN ON XMITR
INC ENDPLG ;COUNT UNITS RUNNING
ADD R1,R1
ADD R0,R0
ADD R0,R0
CMP SELMSK,#20 ;NEXT UNIT
BLT NXTDEV ;DONE?
CLR TIME,TIMER ;NO- LOOP
3\$: TIME1 ;YES- DELAY
BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR...
BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
TST ENDPLG ;DJ11'S DONE?
BNE 45 ;NO- BRANCH
ENDITS,BEGIN ;SIGNAL END OF ITERATION.
MONITOR ;MONITOR SHALL TEST END OF PASS
DEC TIME1 ;COUNT TIMER
BNE 35 ;TO CATCH
BNE 35 ;HUNG DJ11'S
MSGNS\$,BEGIN,HUNG ;ASCII MESSAGE CALL WITH COMMON HEADER
MOV ADDR,R5
MOV R1,SELMSK,SELECT
BIT R1,SELMSK,SELECT
BEQ 65 ;BRANCH IF NOT SELECTED
CLR R5 ;IF SELECTED CLEAR THE CSR
ADD R0,R5 ;NEXT CSR ADDRESS
ASR SELMSK,#20 ;NEXT UNIT SELECT BIT
BLT 5 ;ALL UNITS CHECKED?
ENDS,BEGIN ;DJ11 HUNG
;ROUTINE TO HANDLE TRANSMITTER INTERRUPTS
XMTINT: MOV R6,(SP) ;SAVE REG 2 ON STACK
MOV R7,(SP) ;GET OFFSET INTO R25
ADD ADDR,R7 ;GET DEVICE ADDRESS
BIT R5,XPNW ;TURN OFF XMT INT ENABLE WHILE YOU SERVICE
MOV R5,XPNW ;SAVE OFFSET IN QUEUE
CMP R5,#2 ;INCREMENT IN POINTER
CNE R5,#XQ+10 ;TIME TO WRAP?
MOV R2,XQ,XPNW ;YES
MOV R3,(SP)+,R2 ;RESTORE R2
MOV R5,(SP)+,R5 ;RESTORE R5

```

544 003242 000004 000000 003250
545 -----
546 PIRQS,BEGIN,XMTRTN } QUEUE UP TO CONTINUE AT XMTRTN AND RTI
547 -----
548 XMTRTN: MOV R3,R3 } GET OFFSET
549 ADD R2,XPNTR } INCREMENT OUT POINTER
550 CMP XPNTR,#RQ+10 } TIME TO WRAP?
551 BNE IS } NO
552 MOV R3,RQ,XPNTR } YES
553 ADD ADDR,R2 } GET READY- COPY OFFSET
554 TST (R2) } TO FORM DEVICE ADDRESS
555 BNE XMTOK } XMT READY BIT SET?
556 MOV R2,CSRA } OR IF OK
557 MOV R2,ACSR } STORE COMMAND REGISTER ADDRESS
558 MOV #1,ERRTYP } STORE CONTENTS OF COMMAND REGISTER
559 ***** } ILLEGAL INTERRUPT
560 HRDERS,BEGIN,NULL } XMIT READY NOT SET ON XMIT INTERRUPT
561 *****
562 MOV TIME,TIMER } RESET TIMER DUE TO MESSAGE PRINTOUT DELAY
563 BIS #040000,BCSRA } SET REENABLE TRANSMITTERS
564 EXITS,BEGIN } EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
565 -----
566 XMTOK: MOV R3,R0 } PREPARE A ONE WORD OFFSET
567 ASR R0 } BEGIN THE OFFSET WITH ONE SHIFT
568 ASL R0 } COMPLETE THE OFFSET WITH 2ND SHIFT
569 ASL R3 } OFFSET OF 20 TIMES UNIT #
570 MOV R3,R3 } PUT LINE # IN R4
571 BIC #177760,R4 } PURGE LEFTOVER DATA FROM R4
572 ADD XMTOK,(R4),CHAR } FORM UNIQUE LINE/CHAR OFFSET
573 BIC MASK,CHAR } COMP. HLD CHAR
574 MOV R3,CHAR } FOR MASKING
575 INCB XMTOK,(R4) } TRANSMIT CHARACTER
576 DECCNT } ADVANCE TO NEXT CHARACTER
577 } AVOID NEXT SECTION OF CODE
578 MOV #1,R1 } IF SENT CHARS NOT WRAPPED TO ZERO
579 BIC #177760,R4 } BIT POSITION OF THE ONE IS SAME AS XMT LINE #
580 BNE IS } LEAVE ONLY LINE # IN R4
581 ASL R1 } LINE 0 IS DONE, TURN IT OFF
582 MOV R1,XMITEN(R0) } SHIFT XMT BIT TO NEXT XMTTER LINE POSITION
583 BNE DECCNT } DECREMENT ACTUAL LINE NUMBER
584 BIC R1,XMITEN(R0) } BRANCH BACK IF NOT DONE
585 MOV XMTSEL(R0),XMITEN(R0) } TURN OFF THE INDIVIDUAL XMTTER LINE
586 DECCNT } ARE ALL LINES DONE NOW?
587 MOV XMTSEL(R0),XMITEN(R0) } IF YES, RESET ENABLE WORD
588 TST (R2) } DECREMENT CHARACTER COUNT
589 BNE XMTOUT } HAVE YOU SENT 65? BR IF YES
590 BR } BR IF YES
591 MOV XMTOUT } GO OUT AND WAIT FOR ANOTHER LINE TO INT
592 CLR R2 } TURN OFF XMTTER
593 MOV XMTSEL,R2 } RESET CHAR COUNT
594 EXITS,BEGIN } EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
595 -----
596 XMTOUT: BIS #040000,(R2) } SET REENABLE TRANSMITTERS
597 EXITS,BEGIN } EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

```

```

600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655

```

```

} ROUTINE TO HANDLE RECEIVER INTERRUPTS
RCVRTN: MOV R3,(SP) } SAVE R2 ON STACK
ADD R2,(R5),R2 } GET OFFSET
ADDR,R2 } FORM DEVICE ADDRESS
BIC #010000,(R2) } TURN OFF INTERRUPTS
ADD R5,R2,R5 } SAVE OFFSET IN QUEUE
CMP XPNTR,#RQ+10 } INCREMENT WRITE POINTER
BNE IS } LINE TO WRAP?
MOV R3,RQ,XPNTR } YES
MOV R3,RQ,R2 } RESTORE R2
MOV (SP),R5 } RESTORE R5
-----
} QUEUE UP TO CONTINUE AT RCVRTN AND RTI
RCVRTN: MOV R3,R3 } GET OFFSET
ADD R2,XPNTR } INCREMENT READ POINTER
CMP XPNTR,#RQ+10 } TIME TO WRAP?
BNE IS } NO
MOV R3,RQ,XPNTR } YES
MOV R3,R0 } GET OFFSET AGAIN
ASL R0 } FORM A ONE WORD
INDEX } GET READY
MOV R3,R2 } TO FORM DEVICE ADDRESS
ADD ADDR,R2 } TO FORM DEVICE ADDRESS
BIT #020000,(R2) } SEE IF BUFFER SILO FULL SET
BNE CONT1 } CONTINUE IF OK
MOV R2,CSRA } CONTINUE IF OK
MOV R2,ACSR } STORE COMMAND REGISTER ADDRESS
MOV #1,ERRTYP } STORE CONTENTS OF COMMAND REGISTER
***** } ILLEGAL INTERRUPT
HRDERS,BEGIN,NULL } FALSE INTERRUPT
*****
MOV TIME,TIMER } RESET TIMER DUE TO MESSAGE PRINTOUT DELAY
BIS #010000,BCSRA } BETTER ENABLE RECEIVER BEFORE LEAVING
EXITS,BEGIN } EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
-----
CONT1: MOV R3,R4 } BEGIN TO FORM
ASL R4 } A NEW OFFSET
ASL R4 } INTO EXPECTED RECEIVE DATA TABLE
MOV R4,R5 } BETTER SAVE IT
MOV R4,R4 } SO YOU CAN FORM A SILO OFFSET
ASL R4 }
MOV R4,R4 }
ADD R4,R4 } OFFSET = 202(OCTAL) TIMES UNIT#
MOV R4,R4 } FORM SILO BASE OFFSET
MOV R4,R4 } REMEMBER FIRST SILO WORD ADDRESS
MOV R4,R4 } MOVE DATA FROM DJ TO SILO BUFFER
DEC R4 } DECREMENT # OF CHARACTERS RECEIVED
BNE NOTDUN } BR IF NOT RECEIVED 65 YET
MOV R4,R4 } RESET RECEIVE COUNT
MOV R4,R4 } FORM OFFSET INTO EXP RCV DATA TABLE
MOV R4,R4 } INTO EXP RCV DATA TABLE
MOV R4,R4 } (RCV) BASE ADDRESS FOR THIS UNIT)
MOV R4,R4 } GET LINE NUMBER FROM DATA WORD
NEXTCHR: MOV R5,R5 } MASK SO HAVE ONLY LINE #
BIC #177760,R5

```



```

656 004006 006305 ASL R5 ;MAKE LINE# EVEN WORD BOUNDARY
657 004010 006705 ADD R5 ;FORM EXPECTED DATA ADDRESS
658 004014 011567 MOV R5,CHAR ;-TEMP HLDING FOR BIT SIZING
659 004020 175776 BICB MASK,(R1)+ ;-CHECK ONLY STRAPPED BITS
660 004024 175775 CME CHAR,(R1)+ ;-COMPARE EXPECTED VS ACTUAL DATA
661 004028 082088 BNE DERRR ;BR IF BAD DATA
662 004032 185215 INCB DERRR ;INCREMENT EXPECTED DATA
663 004036 182088 CMP R1,R4 ;HAVE YOU FINISHED ENTIRE BUFFER?
664 004040 183256 BLO NXRCHR ;GO BACK IF NOT
665 004044 001720 DEC ENDCNT,(R0)
666 004048 001720 BGT LS
667 004052 175724 LDRSEL,ENDCNT,(R0)
668 004056 050405 BIC #050405,(R2) ;TURN OFF DJ11
669 004060 175724 DEC ENDFLC
670 004064 000000 EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
671 004068 000000 MOV R2,R5 ;COPY CSR ADDRESS TO R5
672 004072 000024 JSR PC,TBLSNC ;SYNCH RCV AND XMT TABLES
673 004076 050000 BIS #50000,(R2) ;NO- ENABLE SIO FULL INT AND XMT INT
674 004080 001740 MOV XMITEN,(R0),4(R2) ;TURN XMITTERS BACK ON
675 004084 001740 MOV XMITEN,(R0),XMITSV,(R0) ;SAVE INDICATOR OF LINES ACTIVATED
676 004088 000000 EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
677
678 ;SYNCHRONIZE RECEIVER AND TRANSMITTER TABLES FOR ONE UNIT
679 ;WHOSE CSR ADDRESS IS IN R5
680 004124 010546 TBLSNC: MOV R5,(SP) ;CSR ADDRESS
681 004128 006316 SUP ADDR,(SP)
682 004132 006316 ASL (SP),-(SP) ;GET OFFSET (UNIT # TIMES 20 OCTAL)
683 004136 006316 MOV (SP),-(SP) ;COPY OFFSET
684 004140 000404 ASL (SP),-(SP) ;OFFSET INTO RCV TABLE (UNIT # TIMES 40)
685 004144 000002 ADD RRCVTL,(SP)
686 004148 000002 ADD XMITTL,(2)(SP)
687 004152 000004 MOV R16,(SP)
688 004156 000004 1$: MOVWB #4(SP),#2(SP) ;COUNTER
689 ;MOVE XMIT TABLE CONTENTS INTO DATA
690 ;PORTION OF RCV TABLE
691 004164 005266 INC 4(SP)
692 004168 000002 ADD #4(SP)
693 004172 006216 DEC (SP)
694 004176 001366 BNE (SP)
695 004202 022626 CMP (SP)+,(SP)+ ;FIX SP
696 004206 000266 TST (SP)+
697 RTS PC
698
699 ;ERROR HANDLER FOR DATA ERRORS
700 DATERR: MOV R2,CSRA ;STORE COMMAND REGISTER ADDRESS
701 MOV R5,SBADR ;STORE ADDRESS OF TEST DATA
702 TST -(R1) ;RESET R1 BACK TO THE ADDRESS OF THE FAIL DATA
703 MOV R1,WASADR ;STORE ADDRESS OF XMITTED DATA
704 MOV (R1),AWAS ;STORE EXPECTED RECEIVED DATA
705 MOV #ERRQI,R2 ;STORE ACTUAL RECEIVED DATA AND RESET R1 TO RIGH
706 MOV ERROI,R2
707 BICB MASK,(R2)+ ;-MASK OUT STRAPPED OFF BITS
708 MOV R0,(R2)+
709 MOV R1,(R2)+
710 MOV R2,(R2)+
711 MOV R5,(R2)+

```

```

712 004260 020227 001674 CMP R2,#ERRQ+32.
713 004264 103482 BLD LS
714 004268 001634 1$: MOV #ERRQ,R2
715 004272 175376 MOV R2,ERRQI
716 *****
717 004276 104404 000000 DATERS,BEGIN ;DATA ERROR!!!
718 *****
719 004302 175585 175506 MOV TIME,TIMER ;RESET TIMER DUE TO PRINTOUT DELAY
720 004306 175382 ERRO,R2
721 004310 012200 MOV (R2)+,R3
722 004314 012201 MOV (R2)+,R4
723 004318 012204 MOV (R2)+,R5
724 004322 016705 MOV (R2)+,R6
725 004326 001674 CMP R2,#ERRQ+32.
726 004330 063462 BLD R2,#ERRQ+32.
727 004334 012705 001634 MOV #ERRQ,R2
728 004338 175334 2$: MOV R2,ERRQ0
729 004342 016702 175332 MOV CSRA,R2
730 004346 000632 BR DONE ;NOW CONTINUE CHECKING DATA
731
732 004350 004360 DROP: DROP
733 004354 177777 -1
734 004358 004375 HUNG: HUNG
735 004362 177777 -1
736
737 004366 047125 052111 042040 DROP: .ASCIZ /UNIT DROPPED/
738 004370 047522 050120 042105
739 004374 000000
740 004378 000000 044516 020124 HUNG: .ASCIZ /UNIT HUNG/
741 004402 051110 043516 000
742 000001 .END

```


RES2	000960R	242#									
RPTTR	001176R	337#	441*	616	617*	619	620*				
RPTW	0011764R	337#	440*	605*	606*	607	608*				
RO	001624R	299#	337	338	440	441	609	618	620		
RSSFR	000112R	458#									
RSSADR	000102R	251#	701*								
CRSR	000111R	350#	470*	487							
SELECT	000171R	341#	446*	434	434*	471*	482*	523			
SPLMSK	0001772R	341#	448*	434	434*	471*	482*	523	527*	528	
SLO	000604R	295#	421	645							
SOPCMT	000042R	234#									
SOPERS	004406R	470#									
SOPAS	000746R	430#									
SPOINT	000332R	230#									
SPSIZ	000040	263	263								
SR1	000016R	221#	381	424	426						
SR2	000020R	224#									
SR3	000022R	222#									
SR4	000022R	222#									
START	000054R	259#	373#								
STAT	000026R	228#									
SVR0	000062R	243#									
SVR1	000064R	244#									
SVR2	000070R	242#									
SVR3	000070R	242#									
SVR4	000072R	244#									
SVR5	000074R	244#									
SVR6	000076R	244#									
SYSCNT	000052R	238#									
TBLSCNC	000052R	238#	672	680#							
TIME	002014R	355#	481	563	563*	634	719*				
TIMER	002016R	355#	481*	507*	518*	634*	719*				
TIMEL	002020R	355#	508*	516*							
THTBLK	000042R	339#	383								
TIPDFD	000042R	339#									
VCT	002504R	348#	459*	486							
VECTOR	000016R	219#	446								
WASADR	000104R	253#	703*								
WDFR	000114R	250#	375*	414*							
WDND	000114R	250#	374*	413*							
WNT	000108R	320#	393*	589*							
XFLAG	000005R	317#		595*							
XMITEN	001740R	322#	495*	496	497	586*	588*	674	675		
XMITSV	001750R	327#	496*	675*							
XMITNT	003172R	274#	278	282	286	533#					
XMTOK	003356R	556#	567#								
XMTOUT	003356R	556#	567#								
XMTRTN	003250R	545#	548#								
XMTSEL	001730R	309#	495	588							
XMTTBL	000304R	201#	377	574	577*	686					
XPNT	001750R	330#	439*	548*	549*	550					
XPNTM	001750R	330#	438*	537*	538*	550					
XQ	001614R	297#	336	337*	438*	439	557*				
XQ	001614R	297#	336	337*	438*	439	557*				
XQ	0044407D	291#	293#	295#	297#	299#	301#	541	550	552	

ARS. 000000 000
 004407 001

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

XDJAH0,XDJAH0/SOL/CRP:SYN=DDXCOM,XDJAH0
 RUN-TIME: 1 2 .3 SECONDS
 RUN-TIME RATIO: 2774=5.7
 CORE USED: 7K (13 PAGES)