

KWEB DEC/X11 SYSTEM EXERCISER MODULE
XKWEB0.P11 12-OCT-78 12:07

MACY11 30A(1052) 12-OCT-78 16:47 PAGE 2

SEQ 0001

-REM-

IDENTIFICATION

PRODUCT CODE: AC-E920R-MC
PRODUCT NAME: CXKWEB0 KWV11K MODULE
PRODUCT DATE: SEPTEMBER 1978
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) 1976,1978 DIGITAL EQUIPMENT CORPORATION

1.0 ABSTRACT

THE KWE IS AN IOMOD THAT EXERCISES THE KVV11K REAL TIME CLOCK. ON START IT EXERCISES THE CSRS AND PRESET BUFFERS OF THE CLOCK. THEN AND FOR ALL RESTARTS, IT EXERCISES THE CLOCK AT EACH ONE OF ITS BASIC RATES.

2.0 REQUIREMENTS

HARDWARE: ONE KVV11

STORAGE:: KWE REQUIRES:

1. DECIMAL WORDS: 502
2. OCTAL WORDS: 0766
3. OCTAL BYTES: 1754

3.0 PASS DEFINITION

ONE PASS OF THE KWE MODULE CONSISTS OF GENERATING INTERRUPTS FOR ONE SECOND AT EACH CLOCK RATE, UNTIL 60 SECONDS HAVE ELAPSED.

4.0 EXECUTION TIME

ONE PASS OF THE KWE MODULE RUNNING ALONE TAKES APPROXIMATELY ONE MINUTE.

5.0 CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 170420, VECTOR 440, BRI: 4

DEVcnt: 1, SRI: 0

REQUIRED PARAMETERS:
NONE

6.0 DEVICE/OUTPUT SET-UP:

NONE

7.0 MODULE OPERATION

TEST SEQUENCE:

1. (START) BIT EXERCISE CSR, PRESET REGISTER OF CLOCK.
2. (RESTART) COUNT TESTS USING INTERRUPTS COUNT INTERRUPTS WILL OCCUR IN ONE SECOND AND ADVANCE THE TEST TO THE NEXT RATE.

AFTER A RATE HAS BEEN SELECTED, A CHECK IS MADE TO SEE IF THE OPERATOR HAS INHIBITED THAT RATE FROM TEST. IF NOT, CONTROL IS TRANSFERRED TO THE PARTICULAR RATE ROUTINE (LISTED BELOW). EACH RATE ROUTINE MUST PRELOAD THE BUFFER REGISTER OF THE CLOCK TO THE COUNT THAT WILL CAUSE IT TO INTERRUPT IN ONE SECOND. AFTER THE BUFFER IS LOADED, THE CSR IS LOADED WITH THE PROPER BITS THAT SELECT THE RATE.

- A. COUNT TEST CLOCK A RATE 1MHZ.
- B. CLOCK A RATE: 100KHZ
- C. CLOCK A RATE: 10KHZ
- D. CLOCK A RATE: 1KHZ
- E. CLOCK A RATE: 100HZ
- F. CLOCK A RATE: LINE FREQ.
- G. CLOCK A RATE: PSEUDO RANDOM (1 OF 3 RATES)

8.0 OPERATION OPTIONS

VALID SRI VALUES

SRI BIT	ENABLE/DISABLE	FUNCTION
0	0	ENABLE TESTING 1MHZ
	1	DISABLE TESTING 1MHZ
1	0	ENABLE TESTING 100KHZ
	1	DISABLE TESTING 100KHZ
2	0	ENABLE TESTING 10KHZ
	1	DISABLE TESTING 10KHZ
3	0	ENABLE TESTING 1KHZ
	1	DISABLE TESTING 1KHZ
4	0	ENABLE TESTING 100HZ
	1	DISABLE TESTING 100HZ
5	0	*ENABLE TESTING RANDOM
	1	DISABLE TESTING RANDOM
6	0	ENABLE TESTING LINE FREQ
	1	DISABLE TESTING LINE FREQ

*NOTE: IF RANDOM RATE SELECTED, THEN AN SRI BIT DISABLING A PARTICULAR RATE WILL BE IGNORED.

9.0 NON-STANDARD PRINTOUTS:

ALL PRINTOUTS HAVE THE STANDARD FORMATS DESCRIBED IN THE DEC/X11 DOCUMENT.

```

177 000000
178 000000
179
180
181
182
183 000000
184 000000 053513 041105 040
185 000005 000
186 000006 170420
187 000010 000440
188 000013 300
189 000014 300
190 000014 000001
191 000016 000000
192 000020 000000
193 000022 000000
194 000024 000000
195
196 000026 140000
197 000030 000254
198 000033 000224
199 000034 000000
200 000036 000074
201 000039 000000
202 000042 000000
203 000044 000000
204 000046 000000
205 000050 000000
206 000052 000000
207 000054 000000
208 000056 000000
209 000058 000000
210 000060 000000
211 000062 000000
212 000064 000000
213 000066 000000
214 000070 000000
215 000072 000000
216 000074 000000
217 000076 000000
218 000100 000000
219 000102 000000
220 000104 000000
221 000106 000000
222 000104 000000
223 000106 000000
224 000110 000000
225 000112 001152
226 000114 000000
227 000116 000000
228 000120 000000
229 000122 000122
230 000040
231
232

```

```

IDMOD <KWEB>,170420,440,6,6,60,,122
MODULE 140000,KWEB,170420,440,6,60,,122
; TITLE KWEB DEC/X11 SYSTEM EXERCISER MODULE
; DDXCOM VERSION 6 23-MAY-78
; LIST BIN
;*****
BEGIN:
MODNAM: .ASCII /KWEB / ;MODULE NAME
XPL6: .BYTE OPEN ;USE TO KEEP TRACK OF WBUF USAGE
ADDR: 170420+0 ;1ST DEVICE ADDR
VECTOR: 440+0 ;1ST DEVICE VECTOR.
BR1: .BYTE PRTY6+0 ;1ST BR LEVEL.
BR2: .BYTE PRTY6+0 ;2ND BR LEVEL.
DVID1: 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. ADDR
INIT: START ;MODULE START ADDR.
SPOINT: MODSP ;MODULE STACK POINTER.
PASCNT: 0 ;PASS COUNTER.
ICOUNT: 60. ;# OF ITERATIONS PER PASS=60.
SOFCNT: 0 ;LOC TO COUNT ITERATIONS
HRDCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
SOFPAS: 0 ;LOC TO SAVE TOTAL HARD ERRORS
HRDPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
SYSCNT: 0 ;LOC TO SAVE HARD ERRORS PER PASS
RANUM: 0 ;# OF SYS ERRORS ACCUMULATED
;HCLDS RANDOM # WHEN RAND MACRC 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, DR
ACSR: OPEN ;CONTENTS OF CSR.
WASADR: ;ADDR OF BAD DATA, DR
ASTAT: OPEN ;STATUS REG CONTENTS.
ERRTYP: ;TYPE OF ERROR.
ASB: OPEN ;EXPECTED DATA.
AWAS: OPEN ;ACTUAL DATA.
RSTRT: RESTRT ;RESTART ADDRESS AFTER END OF PASS
WDT0: OPEN ;WORDS TO MEMORY PER ITERATION
WDFR: OPEN ;WORDS FROM MEMORY PER ITERATION
INTR: OPEN ;# OF INTERRUPTS PER ITERATION
IDNUM: 122 ;MODULE IDENTIFICATION NUMBER=122
;REPT SPSIZ ;MODULE STACK STARTS HERE.
.NLIST

```

```

233
234
235 000224
236
237

```

```

;WORD 0
;LIST
;ENDR
MODSP:
;*****

```

```

238 ;MODULE REQUIRED REGISTERS - SET UP BY THIS MODULE.
239
240
241 000224* 170420 ASR: -WORD 170420 ;CLOCK A STATUS REG.
242 000226* 170422 ABR: -WORD 170422 ;CLOCK A BUFFER REG.
243
244 000230* 000440 AVECT1: -WORD 440 ;CLOCK A INTERRUPT VECTOR.
245 000232* 000442 AVECT2: -WORD 442
246
247 000234* 000444 BVECT1: -WORD 444 ;CLOCK INTERRUPT VECTOR.
248 000236* 000446 BVECT2: -WORD 446
249
250
251 000240* 000001 RATEP: -WORD 1 ;POINTS TO CURRENT RATE
252 000242* 000000 OFF: -WORD 0 ;OFFSET TO TAKE US TO RATE ROUTINE
253 000244* 000000 RANA: -WORD 0 ;RANDOM NUMBER.
254 000246* 000000 RANB: -WORD 0 ;RANDOM NUMBER.
255 000250* 000000 AIPLG: -WORD 0 ;FLAG TO SHOW THAT CLOCK A HAS INTERRUPTED.
256 000252* 000000 TRY: 0 ;*****
257
258 000254* 012767 000007 177636 START: MOV #7,INTR ;7 INTERRUPTS/ITERATION
259
260 000262* 016767 177520 177734 MOV ADDR,ASR ;GET BASE ADDR.
261 000270* 016767 177514 177732 MOV VECTOR,AVECT ;GET BASE VECTOR ADDR.
262
263
264 000276* 016767 177722 177722 MOV ASR,ABR ;NOW WE'RE GONNA FIX
265 000304* 062767 000002 177714 ADD #2,ABR ;ALL CLOCK ADDRESSES BASED ON ASR.
266
267 000312* 016700 177712 MOV AVECT,R0 ;NOW FIX VECTOR ADDRESSES
268 000316* 062700 000004 ADD R4,R0
269 000322* 010067 177706 MOV #4,R0
270 000326* 016767 177676 MOV AVECT,AVECT2
271 000334* 062767 000002 177670 ADD #2,AVECT2
272 000342* 016767 177666 MOV BVECT,BVECT2
273 000350* 062767 000002 177660 ADD #2,BVECT2
274
275 ;*LOGIC TEST #1 BE SURE A CLOCK EXISTS AT THE
276 ;*SPECIFIED ADDR. IF NO CLOCK THEN A
277 ;*DEC/X11 SYS ERROR WILL OCCUR.
278 ;*
279
280 000356* 005777 177642 LOG1: TST @ASR ;ADDRESS THE CLOCK. IF SYS ERROR
281 ;OCCURS, THEN CLOCK DID NOT
282 ;RETURN SLAVE-SYN WHEN
283 ;ADDRESSED.
284
285 ;*LOGIC TEST #2. MAKE SURE CLOCK CSR BITS
286 ;*14,11,6,5,2, AND 0 CAN BE SET + CLEARED.
287 ;*
288
289 000362* 012767 044125 177514 LOG2: MOV #044125,ASTAT ;GENERATE + RECORD PATTERN TO BE USED.
290 000370* 012767 177510 177514 MOV ASTAT,ASR ;SET THEM IN CSR OF CLOCK A.
291 000376* 017767 177622 177476 MOV @ASR,ACSR ;READ THEM BACK
292
293

```

```

294 000404* 026767 177474 177470 CMP ASTAT,ACSR ;DID THEY ALL SET?
295 000412* 001415 BEQ #25 ;YES - GO TO NEXT TEST.
296 000414* 104407 000000 BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR....
297 000420* 104407 000000 BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
298
299 000424* 016767 177574 177446 1S: MOV ASR,CSRA ;RECORD CSR'S ADDR
300 000432* 012767 000025 177446 MOV #25,ERRTYP ;BIT STUCK
301
302 000440* 104405 000000* 000000 ;*****
303 ;***** PATTERN 044125 FAILED
304 ;*****
305
306 000446* 005077 177552 2S: CLR @ASR ;TRY CLEARING THE BITS
307 000452* 017767 177546 177422 MOV @ASR,ACSR ;READ IT BACK.
308 000460* 001417 BEQ LOG3 ;IF ZERO CSR GOOD.
309 000466* 104407 000000* BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR....
310 000472* 005067 177406 3S: CLR ASTAT ;THEN CONTINUE AT NEXT INSTRUCTION.
311 000476* 016767 177522 177374 MOV ASR,CSRA ;EXPECT ZERO CSR.
312 000504* 012767 000025 177374 MOV #25,ERRTYP ;RECORD CSR'S ADDR.
313 ;*****
314 ;***** PATTERN 044125 FAILED
315 ;*****
316
317 ;*LOGIC TEST #3. MAKE SURE CLOCK CSR BITS
318 ;*13,5,3, AND 1 CAN BE SET + CLEARED.
319 ;*
320
321 000520* 012767 020052 177356 LOG3: MOV #020052,ASTAT ;GENERATE + RECORD PATTERN TO BE USED.
322 000526* 016777 177352 177470 MOV ASTAT,ASR ;SET THEM IN CSR OF CLOCK A.
323 000534* 017767 177464 177340 MOV @ASR,ACSR ;READ THEM BACK
324 000542* 026767 177336 177342 CMP ASTAT,ACSR ;DID THEY ALL SET?
325 000550* 001415 BEQ #25 ;YES - GO TO NEXT TEST.
326 000552* 104407 000000* BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR....
327 000556* 104407 000000* BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
328
329 000562* 016767 177436 177310 1S: MOV ASR,CSRA ;RECORD CSR'S ADDR.
330 000570* 012767 000025 177310 MOV #25,ERRTYP ;BIT STUCK
331
332 000576* 104405 000000* 000000 ;*****
333 ;***** PATTERN 020052 FAILED
334 ;*****
335
336 000604* 005077 177414 2S: CLR @ASR ;TRY CLEARING THE BITS
337 000610* 017767 177410 177264 MOV @ASR,ACSR ;READ IT BACK.
338 000616* 001417 BEQ LOG3 ;IF ZERO CSR GOOD.
339 000620* 104407 000000* BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR....
340 000624* 104407 000000* BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
341 000630* 005067 177250 3S: CLR ASTAT ;EXPECT ZERO CSR.
342 000634* 016767 177364 177236 MOV ASR,CSRA ;RECORD CSR'S ADDR.
343 000642* 012767 000025 177236 MOV #25,ERRTYP ;BIT STUCK
344 ;*****
345 ;***** PATTERN 020052 FAILED
346 ;*****
347
348 000650* 104405 000000* 000000 ;*****
349 ;*****
350
351 ;*LOGIC TEST #4. MAKE SURE CLOCK BUFFER REG

```

```

350 ;*PATTERN 125252 CAN BE SET + CLEARED.
351 ;*
352
353 000656* 012767 125252 177220 LOG4: MOV #125252,ASTAT ;GENERATE + RECORD PATTERN TO BE USED.
354 000664* 016777 177214 177334 MOV ASTAT,@ABR ;SET THEM IN BUFFER REG OF CLOCK.
355 000672* 037467 177330 177202 MOV @ABR,ACSR ;READ THEM BACK
356 000700* 012767 177200 177174 CMP ASTAT,ACSR ;DID THEY ALL SET?
357 000706* 001415 BREAKS,BEGIN ;YES - GO TO NEXT TEST.
358 000710* 104407 000000* BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR...
359 000714* 104407 000000* BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
360
361 000720* 016767 177302 177152 1S: MOV ABR,CSRA ;RECORD BUFFER REG ADDR.
362 000726* 012767 000025 177152 MOV #25,ERRTYP ;BIT STUCK
363 ***** ;*****
364 000734* 104405 000000* 000000 HRDERS,BEGIN,NULL ;:BUFFER REG PATTERN 125252 FAILED
365 ***** ;*****
366
367 000742* 005077 177260 177126 2S: CLR @ABR ;TRY CLEARING THE BITS
368 000746* 017767 177254 177126 MOV @ABR,ACSR ;READ IT BACK.
369 000754* 001417 BEQ LOGS ;IF ZERO BUFFER GOOD.
370 000756* 104407 000000* BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR...
371 000762* 005067 177112 000000* BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
372
373 000772* 016767 177230 177100 3S: MOV ABR,CSRA ;EXPECT ZERO BUFFER.
374 001000* 012767 000025 177100 MOV #25,ERRTYP ;RECORD ADDR. OF BUFFER REG.
375 ***** ;*****
376 001006* 104405 000000* 000000 HRDERS,BEGIN,NULL ;:BUFFER REG FAILED TO CLEAR
377 ***** ;*****
378
379 ;*
380 ;*LOGIC TEST #5. MAKE SURE CLOCK BUFFER REG
381 ;*PATTERN 052525 CAN BE SET + CLEARED.
382 ;*
383
384 001014* 012767 052525 177062 LOG5: MOV #052525,ASTAT ;GENERATE + RECORD PATTERN TO BE USED.
385 001022* 016777 177056 177176 MOV ASTAT,@ABR ;SET THEM IN BUFFER OF CLOCK A.
386 001030* 017767 177172 177044 MOV @ABR,ACSR ;READ THEM BACK
387 001036* 036767 177042 177036 CMP ASTAT,ACSR ;DID THEY ALL SET?
388 001044* 001415 BREAKS,BEGIN ;YES - GO TO NEXT TEST.
389 001046* 104407 000000* BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR...
390 001052* 104407 000000* BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
391
392 001056* 016767 177144 177014 1S: MOV ABR,CSRA ;RECORD BUFFER REG ADDR.
393 001064* 012767 000025 177014 MOV #25,ERRTYP ;BIT STUCK
394 ***** ;*****
395 001072* 104405 000000* 000000 HRDERS,BEGIN,NULL ;:BUFF REG PATTERN 052525 FAILED
396 ***** ;*****
397
398 001100* 005077 177122 176770 2S: CLR @ABR ;TRY CLEARING THE BITS
399 001104* 017767 177116 176770 MOV @ABR,ACSR ;READ IT BACK.
400 001112* 001417 BEQ LOGS ;IF ZERO BUFFER GOOD.
401 001114* 104407 000000* BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR...
402 001120* 104407 000000* BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
403
404 001130* 005067 176754 176742 3S: CLR ASTAT ;EXPECT ZERO BUFFER.
405 001136* 012767 000025 176742 MOV ABR,CSRA ;RECORD BUFFER REG A ADDR.
406 ***** ;*****
407 MOV #25,ERRTYP ;BIT STUCK

```

```

406 ***** ;*****
407 001144* 104405 000000* 000000 HRDERS,BEGIN,NULL ;:BUFFER REG FAILED TO CLEAR
408 ***** ;*****
409
410 001152* 012767 125252 177064 RESTRT: MOV #125252,RANA ;PRESET RANDOM NUMBER GENERATOR
411 001160* 012767 000100 177052 MOV #BIT6,RATEP ;FIRST PASS THRU LOOP, OFFSET=0, RATES WILL=1.
412 001166* 012777 001626 177034 MOV #INSERV,RAVECT ;SET UP CLOCK A'S INTER. VECTOR.
413 001174* 014402 176632 177030 MOV @RAVECT2 ;SET PRIORITY ON CLOCK A'S INTR.
414 001202* 006167 177032 LOOP: ROLB RATEP ;GET NEXT RATE.
415
416 BPL 1S ;IF NOT END THEN CONTINUE.
417 001212* 005067 177026 CLR OFF ;CLEAR THE OFFSET
418 001214* 017767 000001 177016 MOV #1,RATEP ;LOOK AT FIRST RATE
419 001222* 062767 000002 177012 1S: ADD #2,OFF
420 001230* 036767 177004 176560 BIT RATEP,SRI ;IS THIS RATE INHIBITED?
421 001236* 001361 RNE LOOP
422
423 CLR AIFLG ;CLR FLAG INDICATING CLOCK A HAS INTERRUPTED.
424 001244* 016701 176772 MOV OFF,R1 ;PICK UP OFFSET
425 001250* 000171 001254* JMP @LISTP(R1) ;GO SET THE RATE + START THE CLOCK.
426
427 ;
428 ;THE FOLLOWING (LISTP) ARE POINTERS TO VARIOUS RATE
429 ;ROUTINES. THEY ARE USED BY "LOOP". "LOOP" GENERATES
430 ;AN OFFSET OF A RATE WE WISH TO EXERCISE. THE OFFSET
431 ;IS STORED IN R1. WE INDEIX "LISTP" BY R1 (JMP @LISTP(R1))
432 ;TO GET THE ADDRESS OF THE RATE ROUTINE TO EXERCISE.
433 ;
434
435 LISTP: .WORD 1 ;
436 .WORD RATE0 ;:POINTER TO 1MHZ ROUTINE
437 .WORD RATE1 ;:POINTER TO 100KHZ ROUTINE
438 .WORD RATE2 ;:POINTER TO 10KHZ ROUTINE
439 .WORD RATE3 ;:POINTER TO 1KHZ ROUTINE
440 .WORD RATE4 ;:POINTER TO 100HZ ROUTINE
441 .WORD RATE5 ;:POINTER TO RANDOM ROUTINE
442 .WORD RATE6 ;:POINTER TO LINE FREQ. ROUTINE
443
444 ;
445 ;THE FOLLOWING (RATEAL) ARE THE PRESET VALUES THAT THE
446 ;VARIOUS RATE ROUTINES NEED. THEY ARE LOADED INTO
447 ;CLOCK A'S PRESET BUFFER. "RATEAL" IS INDEXED BY
448 ;AN OFFSET IN R1 BY THE RATE ROUTINES TO GET THE
449 ;PRESET VALUE
450 ;
451
452 RATEAL: .WORD 1 ;:OFFSET ZERO,NO RATE.
453 .WORD -50000. ;:VALUE FOR 1MHZ PRESET.
454 .WORD -50000. ;:PRESET VALUE FOR 100 KHZ
455 .WORD -10000. ;:PRESET VALUE FOR 10 KHZ
456 .WORD -1000. ;:PRESET VALUE FOR 1 KHZ
457 .WORD -100. ;:PRESET VALUE FOR 100 HZ
458 .WORD 0 ;:PRESET VALUE FOR RANDOM
459 .WORD -60. ;:PRESET VALUE FOR LINE FREQ.
460
461 ;

```

462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517

```

;THE FOLLOWING (RSAL) IS USED BY THE RANDOM
;RATE ROUTINE (RATE5). THEY ARE THE VALUES NEEDED
;TO BE PUT INTO THE CLOCK'S CSR FOR A PARTICULAR RATE.
RSAL:  .WORD  0           ;OFFSET ZERO, NO RATE.
        .WORD  113        ;1 MHZ, GO., MODE 1
        .WORD  123        ;100 KHZ, GO., MODE 1
        .WORD  131        ;10 KHZ, GO.
        .WORD  141        ;1 KHZ, GO.
        .WORD  151        ;100 HZ, GO.
        .WORD  171        ;LINE FREQ.
;
; *THIS ROUTINE PRESETS CLOCK A FOR
; *1 MHZ RATE   CLOCK A INTRs IN 1/20 SEC., 25 TIMES.
; *
RATE0:  CLR   @ASR          ;CLEAR CLOCK A.
        MOV   RATEAL(R1),@ASR ;PRESET COUNT IN CLOCK A.
        MOV   RSAL(R1),@ASR  ;START CLOCK A.
        EXITS,BEGIN        ;NOW WAIT FOR INTERRUPT.
                                ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
;
; *THIS ROUTINE PRESETS CLOCK A FOR
; *100 KHZ RATE  CLOCK A INTRs IN .5 SEC., TWICE.
; *
RATE1:  CLR   @ASR          ;CLEAR CLOCK A.
        MOV   RATEAL(R1),@ASR ;PRESET COUNT IN CLOCK A.
        MOV   RSAL(R1),@ASR  ;START CLOCK A.
        EXITS,BEGIN        ;NOW WAIT FOR INTERRUPT.
                                ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
;
; *THIS ROUTINE PRESETS CLOCK A FOR
; *10 KHZ RATE   CLOCK A INTRs IN 1.0 SEC.
; *
RATE2:  CLR   @ASR          ;CLEAR CLOCK A.
        MOV   RATEAL(R1),@ASR ;PRESET COUNT IN CLOCK A.
        MOV   RSAL(R1),@ASR  ;START CLOCK A.
        EXITS,BEGIN        ;NOW WAIT FOR INTERRUPT.
                                ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
;
; *THIS ROUTINE PRESETS CLOCK FOR
; *1 KHZ RATE    CLOCK A INTRs IN 1.0 SEC.
; *

```

518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573

```

; *
RATE3:  CLR   @ASR          ;CLEAR CLOCK A.
        MOV   RATEAL(R1),@ASR ;PRESET COUNT IN CLOCK A.
        MOV   RSAL(R1),@ASR  ;START CLOCK A.
        EXITS,BEGIN        ;NOW WAIT FOR INTERRUPT.
                                ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
;
; *THIS ROUTINE PRESETS CLOCK FOR
; *100 HZ RATE  CLOCK A INTRs IN 1.0 SEC.
; *
RATE4:  CLR   @ASR          ;CLEAR CLOCK A.
        MOV   RATEAL(R1),@ASR ;PRESET COUNT IN CLOCK A.
        MOV   RSAL(R1),@ASR  ;START CLOCK A.
        EXITS,BEGIN        ;NOW WAIT FOR INTERRUPT.
                                ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
;
; *THIS ROUTINE PRESETS CLOCK FOR
; *RANDOM RATES
RATE5:  JSR   PC,RANDOM      ;GET 2 RANDOM NUMBERS.
        BIC   #177771,RANA   ;MAKE NUMBER < 10.
        BIC   #177771,RANB   ;MAKE 2ND NUMBER < 10.
;
; *NUMBERS MUST BE 2, 4, OR 6
3$:     TST   RANA           ;IS NUMBER ZERO?
        BNE  4$,RANA        ;NO-GO AHEAD.
        ADD  #2,RANA        ;MAKE IT NON-ZERO
;
4$:     TST   RANB           ;IS NUMBER ZERO?
        BNE  5$,RANB        ;NO-GO AHEAD.
        ADD  #2,RANB        ;MAKE IT NON-ZERO.
;
5$:
RATE6:  CLR   @ASR          ;CLEAR CLOCK A.
        MOV   RANA,R1       ;RECORD THE OFFSET.
        MOV   R1,OFF        ;PRESET CLOCK A.
        MOV   RATEAL(R1),@ASR ;START CLOCK A.
        EXITS,BEGIN        ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
;
; *THIS ROUTINE PRESETS CLOCK FOR
; *LINE FREQ RATE  CLOCK A INTRs. IN 1.0 SEC
; *
; *
RATE6:  CLR   @ASR          ;CLEAR CLOCK A.
        MOV   RATEAL(R1),@ASR ;PRESET COUNT IN A.

```


ICONT	000036R	200#																			
ICOUNT	000040R	201#																			
IDNUM	000122R	230#																			
INE	001171R	602	605#																		
INIT	000030R	197#																			
INSERV	001626R	412																			
INTR	000120R	229#																			
LSTP	001154R	475																			
LOG1	000356R	281#																			
LOG2	000362R	291#																			
LOG3	000520R	307																			
LOG4	000556R	338																			
LOG5	001014R	369																			
LOOP	001202R	414#																			
MAP22S	= 104416	238#																			
MODNAM	000000R	184#																			
MODSP	000224R	198																			
MSGMS	= 104403	238#																			
MSGSS	= 104402	238#																			
MSG	= 104401	238#																			
NULL	= 000000	238#																			
OFF	= 000242R	252#	302	314	333	345	364	376	395	407											
OPEN	= 000000	185#	417*	419*	424	560*	587	592	213	214	215	216	217	218							
		222	191	192	193	194	211	212	213	214	215	216	217	218							
			185#	224	225	227	228	229	230#												
PTOAS	= 104420	238#																			
PASCNT	000034R	199#																			
PAROS	000004	238#																			
POPSP	= 005726	238#																			
POPSP2	= 022626	238#																			
PRTV	000000	238#																			
PRTV0	000000	238#																			
PRTV1	000040	238#																			
PRTV2	000100	238#																			
PRTV3	000140	238#																			
PRTV4	000200	238#																			
PRTV5	000240	238#																			
PRTV6	000300	188																			
PRTV7	000340	238#	189	238#																	
PS	= 177776	238#																			
PSM	= 005746	238#																			
PUSH	= 024646	238#																			
PUSH2	= 00724R	238#																			
RANA	00024R	238#																			
RANB	00024R	238#	410*	545*	549	551*	559	619	620*	621*											
RAND	001726R	543	546*	553	555*	619*	621	622*													
RANDS	= 104417	238#	619#																		
RANUM	000054R	202																			
RATEAL	000240R	251#																			
RATEP	000240R	251#	493	496	509	522	535	561	573												
RATE0	001332R	436	411*	414*	418*	420															
RATE1	001406R	438																			
RATE2	001426R	439																			
RATE3	001426R	440																			
RATE4	001452R	441																			
RATE5	001476R	441																			
RATE6	001602R	442																			

RESTR	001152R	226	400	410#																
RES1	000056R	209#																		
RES2	000060R	210#																		
RSAL	001314R	467#	484	497	510	523	536	562	574											
RSTR	000112R	226#																		
SBADR	000102R	219#																		
SOPCNT	000042R	202#																		
SOPERS	= 104406	238#																		
SOPPAS	000046R	204#																		
SPOINT	000032R	198#																		
SPSIZ	= 000040	1																		
SR1	000016R	191#																		
SR2	000020R	192#																		
SR3	000022R	193#																		
SR4	000024R	194#																		
START	000254R	197#																		
STAT	00026R	196#																		
SVRO	000022R	211#																		
SVR1	000064R	212#																		
SVR2	000066R	213#																		
SVR3	000070R	214#																		
SVR4	000072R	215#																		
SVR5	000074R	216#																		
SVR6	000076R	217#																		
SYSCNT	000052R	206#																		
TAPDFD	= 000022	238#																		
TRY	000252R	256#																		
VECTOR	000010R	187#																		
WASADR	000104R	221#																		
WDFR	000116R	228#																		
WDTO	000114R	227#																		
XFLAG	000005R	185#																		

. ABS. 000000 000
 001754 001

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

XKWEB0, XKWEB0/SOL/CRF:SYM=DDXCOM, XKWEB0
 RUN-TIME: 1 2 3 SECONDS
 RUN-TIME RATIO: 31/3=7.7
 CORE USED: 7K (13 PAGES)