

.REM J

-----  
IDENTIFICATION  
-----

PRODUCT CODE: AC-E875P-WC  
PRODUCT NAME: CXARABO AR-11 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 ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1974, 1978 DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT

ARA IS AN IOMOD THAT EXERCISES THE AR11. IT WILL RUN IN ONE OF TWO COMPLETELY INDEPENDENT MODES.

A.) WITHOUT A G5036 WRAPAROUND TEST MODULE CONNECTED TO THE AR11.

B.) WITH A G5036 WRAPAROUND TEST MODULE CONNECTED TO THE AR11.

=NOTE=  
IF A BC08R CABLE IS USED  
THE BERG AT THE AR11 END SHOULD  
BE CONNECTED A TO A2 AND THE BERG AT THE G5036  
END SHOULD BE CONNECTED A TO VV.

THE PROGRAM AUTOMATICALLY ENTERS THE APPROPRIATE MODE BY TESTING FOR THE PRESENCE OF A WRAPAROUND MODULE AT RUN TIME.

2. REQUIREMENTS

HARDWARE: AR11

STORAGE: ARA REQUIRES:

1. DECIMAL WORDS: 1125
2. OCTAL WORDS: 02145
3. OCTAL BYTES: 4312

3. PASS DEFINITION

IN WRAPAROUND MODE, ONE PASS OF THE ARA MODULE CONSISTS OF SEQUENCING THROUGH 8 COMBINATIONS OF RMS AND PEAK NOISE CALCULATIONS ONE TIME FOR A TOTAL OF 81,920. CONVERSIONS.

IN NON-WRAPAROUND MODE, ONE PASS CONSISTS OF DISPLAYING EACH OF THE 16 A/D CHANNELS SEQUENTIALLY FOR A TOTAL OF 102,400 CONVERSIONS.

4. EXECUTION TIME

ONE PASS OF ARA RUNNING ALONE TAKES APPROXIMATELY 1 MINUTE.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DVA: 1, VCT: 1, BR1: 6, BR2: 4

REQUIRED PARAMETERS:

ARAB DECAY11 SYSTEM EXERCISER MODULE  
XARABO.P11 12-OCT-78 11:45

MACY11 30A(1052) 12-OCT-78 16:17 PAGE 4

SEQ 0003

ADDRESS AND VECTOR MUST BE SPECIFIED AT CONFIGURATION OR RUN TIME.

6. DEVICE/OPTION SETUP

TO RUN IN NON-WRAPAROUND MODE, DO NOT CONNECT THE G5036 WRAPAROUND MODULE SO THAT THE EXERCISER WILL NOT SENSE ITS PRESENCE AND WILL ENTER THE NON-WRAPAROUND MODE CORRECTLY.

TO RUN IN THE WRAPAROUND MODE, INSURE THAT THE G5036 WRAPAROUND MODULE IS CONNECTED PROPERLY TO THE AR11 AND IS OPERATING CORRECTLY BY RUNNING THE STAND ALONE WRAPAROUND DIAGNOSTIC. THIS WILL INSURE THAT THE EXERCISER WILL BE ABLE TO SENSE THE PRESENCE OF THE G5036 AND AUTOMATICALLY ENTER THE WRAPAROUND MODE.

7. MODULE OPERATION

AT THE BEGINNING OF THE DEC-X RUN, ARA WILL PRINTOUT ONE DOUBLE LINE MESSAGE TO INDICATE WHETHER IT IS RUNNING IN WRAPAROUND MODE OR NON-WRAPAROUND MODE. THIS TYPEOUT IS NOT AN ERROR.

IN NON-WRAPAROUND MODE, ARA DOES NOT MAKE ANY ERROR CHECKS ITSELF. PROPER OPERATION IS ASCERTAINED BY VISUAL EXAMINATION OF THE DISPLAY AND THROUGH THE MONITOR'S ERROR REPORTING CAPABILITY. DATA IS ACQUIRED AND DISPLAYED FROM EACH OF THE 16 A/D CHANNELS SEQUENTIALLY WITH THE CHANNEL NUMBER DISPLAYED IN THE CENTER BOTTOM.

IN WRAPAROUND MODE, NOISE CALCULATIONS ARE MADE FROM STATISTICAL ANALYSIS OF LARGE NUMBERS OF SAMPLES AND COMPARED TO LIMITS IN THE FOLLOWING SEQUENCE:

- A. A/D RMS NOISE USING FINE X WRAPAROUND DAC.
- B. A/D RMS NOISE USING FINE Y WRAPAROUND DAC.
- C. V DAC RMS NOISE.
- D. X DAC RMS NOISE.
- E. A/D PEAK NOISE USING FINE X WRAPAROUND DAC.
- F. A/D PEAK NOISE USING FINE Y WRAPAROUND DAC.
- G. V DAC PEAK NOISE.
- H. X DAC PEAK NOISE.
- I. END PASS.

8. OPERATION OPTIONS

LOCATIONS ARMLIM, DRMLIM, APKLM AND DPKLM CAN BE MODIFIED TO CHANGE THE MAXIMUM ALLOWABLE NOISE LIMITS. IF THE ACTUAL RMS AND PEAK NOISE FIGURES FOR THE AR11 RUNNING IN A SYSTEM ENVIRONMENT ARE DESIRED, THESE 4 LOCATIONS MAY BE CHANGED TO 000000 IN WHICH CASE ALL NOISE CALCULATIONS WILL BE TYPED OUT.

9. NON-STANDARD PRINTOUTS

A. A NON-ERROR MESSAGE AT THE BEGINNING OF THE DEC-X RUN WILL BE PRINTED:

AR11 RUNNING IN WRAPAROUND/NON-WRAPAROUND MODE

B. IF ARA FINDS EXCESSIVE ANALOG NOISE, IT REPORTS IT IN A MSGN CALL:

(EXAMPLE)  
A/D PEAK NOISE = 2.57 LSB (LIMIT = 2.00 LSB)

FOLLOWED BY AN ERRORN CALL WHICH PRINTS OUT 6 LOCATIONS CONTAINING POWER SUPPLY VOLTAGE INFORMATION:

AVP14V SPP14V AVN14V SPN14V AVP5HQ SPP5HQ

DEFINED AS FOLLOWS:

AVP14V (AVERAGE VOLTAGE LEVEL ON +14V H.Q. SUPPLY)

THE AVERAGE OF 512 CONVERSIONS TAKEN ON CH.#3 (+2.5V) SCALED DOWN FROM THE +14 VOLT SUPPLY.

IF THE OCTAL NUMBER IN THIS COLUMN IS LESS THAN 1704 THEN THE +14 VOLT SUPPLY IS TOO LOW FOR NORMAL OPERATION.

VOLTAGE OCTAL PRINTOUT  
>+13.9 001777  
13.75 001772  
13.5 001761  
13.25 001747  
13.0 001737  
12.75 001725  
12.5 001714  
12.25 001704

(LOWER LIMIT FOR NORMAL OPERATION)

SPP14V (COUNT SPREAD ON +14 VOLT H.Q. SUPPLY)

THE DIFFERENCE BETWEEN THE HIGHEST AND LOWEST CONVERSION OF 512 TAKEN ON CH.#3 (+2.5V) SCALED DOWN +14V.

IF THE OCTAL NUMBER PRINTED IN THIS COLUMN IS GREATER THAN 000010, THEN THE +14 VOLT SUPPLY IS CONSIDERED TOO NOISY FOR ACCURATE ANALOG OPERATION.

OCTAL PRINTOUT PEAK TO PEAK NOISE  
000000-000004 <100 MILLIVOLTS NOISE (TYPICAL LEVEL)  
>000010 >200 MILLIVOLTS NOISE (EXCESSIVELY NOISY)

-----  
AVN14V

(AVERAGE VOLTAGE ON THE -14 VOLT H.Q. SUPPLY)  
THE AVERAGE OF 512 CONVERSIONS TAKEN ON CH.#4 (-2.5V)  
SCALED DOWN FROM THE -14V. SUPPLY  
AN OCTAL PRINTOUT IN THIS COLUMN OF GREATER THAN  
000053 INDICATES THE -14 VOLT SUPPLY IS LOW.

VOLTAGE  
<-13.9  
000000  
000006  
13.75  
000017  
13.25  
000031  
13.0  
000041  
12.75  
000053 (LOWER LIMIT)  
000064

-----  
SPN14V

(COUNT SPREAD ON -14 VOLT SUPPLY)  
THE DIFFERENCE BETWEEN THE HIGHEST AND LOWEST CONVERSION  
OF 512 TAKEN ON CH.#4 (-2.5V) SCALED DOWN -14V.  
IF THE OCTAL PRINTOUT IN THIS COLUMN IS GREATER THAN  
000010, THEN THE -14 VOLTS IS TOO NOISY FOR ACCURATE  
ANALOG OPERATION.

OCTAL PRINTOUT PEAK TO PEAK NOISE  
000000-000004 <100 MILLIVOLTS NOISE (TYPICAL)  
>000010 >200 MILLIVOLTS NOISE (EXCESSIVE NOISE)

-----  
AVP5HQ

(AVERAGE VOLTAGE LEVEL ON +5V H.Q.)  
THE AVERAGE OF 512 CONVERSIONS TAKEN ON CH.#17 (+4V)  
SCALED DOWN FROM THE +5 H.Q. VOLTAGE.  
AN OCTAL PRINTOUT OF LESS THAN 001321 OR GREATER  
THAN 001524 IN THIS COLUMN INDICATES THE +5HQ IS OUTSIDE  
OF ITS NORMAL OPERATING RANGE.

VOLTAGE  
5.7  
5.1  
5.0  
4.9  
4.8  
4.7  
4.6  
4.5  
4.4  
OCTAL PRINTOUT (UPPER LIMIT)  
001524  
001504  
001484  
001464  
001443  
001423  
001402  
001382  
001361  
001341  
001321 (LOWER LIMIT)

-----  
SPPSHQ (COUNT SPREAD ON +5V H.Q.)

THE DIFFERENCE BETWEEN THE HIGHEST AND LOWST CONVERSION  
OF 512 TAKEN ON CH.#17 (+4V) SCALED DOWN +5HQ.

AN OCTAL PRINTOUT IN THIS COLUMN OF GREATER THAN 00030  
IS REGARDED AS NOISY AND 00040 (200 MILLIVOLTS NOISE)  
IS HIGHLY QUESTIONABLE.

C. A MSG STATING "WRAPAROUND ERROR" FOLLOWED BY A DROPPED MESSAGE  
INDICATES THAT A/D SENSE A WRAPAROUND MODULE BUT THAT  
THROUGH A MALFUNCTION OR MALADJUSTMENT ALL ZERES  
OR ALL ONES WAS RETURNED BY THE SUCCESSIVE  
APPROXIMATION ROUTINE. RUN THE STAND ALONE WRAPAROUND  
DIAGNOSTIC TO DETERMINE WHY THE A/D VALUE WAS OUT OF  
THE RANGE OF THE DAC.

```

000000-
000000-
000000- 051101 041101 040
000005- 000
000006- 000001
000010- 000001
000012- 300
000013- 300
000014- 000001
000016- 000000
000020- 000000
000022- 000000
000024- 000000
000026- 140000
000030- 000224-
000032- 000224-
000034- 000000
000036- 000001
000040- 000000
000042- 000000
000044- 000000
000046- 000000
000050- 000000
000052- 000000
000054- 000000
000056- 000000
000060- 000000
000062- 000000
000064- 000000
000066- 000000
000070- 000000
000072- 000000
000074- 000000
000076- 000000
000100- 000000
000102-
000102- 000000
000104-
000104- 000000
000106-
000106- 000000
000110- 000000
000112- 000224-
000114- 000000
000116- 000000
000120- 000000
000122- 000133
000040-

]
IONOD <ARAB> 1,1,6,4,1,133
MODULE 140000 ARAB 1,1,6,4,1,133
TITLE ARAB DEC/X11 SYSTEM EXERCISER MODULE
DDXCON VERSION 6 23-MAY-78
LIST
*****
BEGIN:
MODNAM: -ASCII /ARAB / ;MODULE NAME.
XFLAG: ;BYTE OPEN ;USED TO KEEP TRACK OF WBUFF USAGE
ADDR: ;+0 ;1ST DEVICE ADDR
VECTOR: ;+0 ;1ST DEVICE VECTOR.
BR1: -BYTE PRTV6+0 ;1ST BR LEVEL-
BR2: -BYTE PRTV4+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.
INIT: START ;MODULE START ADDR.
SPOINT: MODSP ;MODULE STACK POINTER.
PASCNT: 0 ;PASS COUNTER
ICOUNT: 1 ;# OF ITERATIONS PER PASS=1
LCOUNT: 0 ;LOC TO COUNT ITERATIONS
SRFCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
HRDCNT: 0 ;LOC TO SAVE TOTAL HARD ERRORS
SOPPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
HRDPAS: 0 ;LOC TO SAVE HARD ERRORS PER PASS
SYSCNT: 0 ;# OF SYS ERRORS ACCUMULATED
RANNUM: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
CONFIG:
RES1: 0 ;RESERVED FOR MONITOR USE
RES2: 0 ;RESERVED FOR MONITOR USE
RES3: 0 ;RESERVED FOR MONITOR USE
SVR0: OPEN ;LOC TO SAVE R0.
SVR1: OPEN ;LOC TO SAVE R1.
SVR2: OPEN ;LOC TO SAVE R2.
SVR3: OPEN ;LOC TO SAVE R3.
SVR4: OPEN ;LOC TO SAVE R4.
SVR5: OPEN ;LOC TO SAVE R5.
SVR6: OPEN ;LOC TO SAVE R6.
CSRA: OPEN ;ADDR OF CURRENT CSR.
SBADR: ;ADDR OF GOOD DATA, OR
ACSR: OPEN ;CONTENTS OF CSR.
WASADR: ;ADDR OF BAD DATA, OR
ASTAT: OPEN ;STATUS REG CONTENTS.
ERRTYP: ;TYPE OF ERROR
ASB: OPEN ;EXPECTED DATA.
AWAS: OPEN ;ACTUAL DATA.
RSTRT: RSTRT ;RESTART ADDRESS AFTER END OF PASS
WDTO: OPEN ;WORDS TO MEMORY PER ITERATION
WDFR: OPEN ;WORDS FROM MEMORY PER ITERATION
INTR: OPEN ;# OF INTERRUPTS PER ITERATION
IDNUM: 133 ;MODULE IDENTIFICATION NUMBER=133
-REPT SPSIZ ;MODULE STACK STARTS HERE.

```

```

000224-
-NLIST
-WORD 0
-LIST
-ENDR
MODSP:
*****

```



ARABO-P11 12-OCT-78 11:45
334 000234 016700 177556
335 000230 010067 003236
337 000234 005720
338 000236 010067 003232
339 000242 005720
340 000244 010067 003226
341 000250 005720
342 000252 010067 003222
343 000256 005720
344 000260 010067 003216
345 000264 005720
346 000266 010067 003212
347 000272 005720
348 000274 010067 003206
349 000300 005720
350 000302 010067 003202
351
352 000306
353
354
355 000306 104421 000000 004302
356 000314 003514
357
358 000316 116767 003174 002550
359 000324 116767 003167 002544
361 000332 116767 003162 002537
362
363
364 000340 104421 000000 004306
365 000346 003514
366
367 000350 116767 003142 002530
369 000356 116767 003135 002524
370 000364 116767 003130 002517
371
372
373 000372 104421 000000 004304
374 000400 003514
375
376 000402 116767 003110 002510
378 000410 116767 003103 002504
379 000416 116767 003076 002477
380
381
382 000434 104421 000000 004310
383 000432 003514
384
385 000434 116767 003056 002470
387 000442 116767 003051 002464
388 000450 116767 003044 002457

START:
RSTRT: MOV ADDR,R0 ;SETUP REGISTER ADDRESSES
MOV R0,ADSR
TST (R0)+
MOV R0,ADRR
TST (R0)+
MOV R0,CLKSR
TST (R0)+
MOV R0,CLKBR
TST (R0)+
MOV R0,DPSR
TST (R0)+
MOV R0,XDAC
TST (R0)+
MOV R0,YDAC
TST (R0)+
MOV R0,CLKCNT

LIMIT: ;\*\*\*\*\*
;CONVERT ARMLIM TO ASCII AND
;STORE AT DECIM
BTODS,BEGIN,ARMLIM,DECIM
;\*\*\*\*\*
MOV DECIM+2,P7
MOV DECIM+3,P7+2
MOV DECIM+4,P7+3
;CONVERT APKLM TO ASCII AND
;STORE AT DECIM
BTODS,BEGIN,APKLM,DECIM
;\*\*\*\*\*
MOV DECIM+2,P8
MOV DECIM+3,P8+2
MOV DECIM+4,P8+3
;CONVERT DRMLIM TO ASCII AND
;STORE AT DECIM
BTODS,BEGIN,DRMLIM,DECIM
;\*\*\*\*\*
MOV DECIM+2,P9
MOV DECIM+3,P9+2
MOV DECIM+4,P9+3
;CONVERT DPKLM TO ASCII AND
;STORE AT DECIM
BTODS,BEGIN,DPKLM,DECIM
;\*\*\*\*\*
MOV DECIM+2,P10
MOV DECIM+3,P10+2
MOV DECIM+4,P10+3

ARABO-P11 12-OCT-78 11:45
389 000456 016700 177326
390 000462 012720 000550
391 000466 016710 000550
392 000472 005067 000104
393 000476 012777 000120 002766
394 000504 012777 000001 002770
395 000512 104407 000000
396 000516 104407 000000
397 000522 005767 002764
398 000526 001413
399 000530 012767 000020 177356
400 000536 012767 000020 177352
401 000544 000167 000416
402 000550 005267 002736
403 000554 000002
404
405 000556 012767 000200 177334
406 000564 012767 003340 177324
407 000572 104403 000000 003410
408 000600 012777 000100 002664
409 000606 005077 002822
410 000612 012703 003522
411 000616 012767 000576 177266
412 000624 012777 000110 002650
413 000632 012777 000102 002636
414 000640 016700 177144
415 000644 012720 000726
416 000650 005720
417 000652 012720 000720
418 000656 116720 177130
419 000662 105720
420 000664 012720 000736
421 000670 116710 177117
422 000674 005740
423 000676 016701 000262
424 000702 016777 000254
425 000710 005277 002562 002570
426 000714 104400 000000
427 000720 005277 002546
428 000724 000002
429 000726 017777 002542 002552
430 000734 000002
431 000736
432
433 000736 000004 000000 000744
434
435 000744 062777 000020 002532
436 000752 005777 002526
437 000756 001351
438 000760 012710 001006
439 000764 012705 000012
440 000770 010304

SENSE: MOV VECTOR,R0 ;SET VECTOR AND PSW FOR DSP. INTERRUPT
MOV #SAINT,(R0)+
MOV BR1,@R0
CLR #12,@ADSR ;CLEAR WRAPAROUND INDICATOR
MOV #1,@DPSR ;EXTERNAL START AND INTERRUPT ENABLE
MOV #1,@DPSR ;INTENSIFY
BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR....
BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
TST WRPFLC ;SEE IF A/D INTERRUPTED
BREQ NOWRAP ;16. WORDS TO MEM/ITERATION
MOV #16,@WDT0 ;16. WORDS FROM MEM/ITERATION
MOV #16,@WDFR
JMP WDFR
SAINT: INC WRPFLC ;FLAG THAT INTERRUPT HAS OCCURRED
RTI

NOWRAP: MOV #128,@INTR ;128. INTERRUPTS/ITERATION
MOV #1760,@WDFR ;1760 WORDS FROM MEM
MSGNS,BEGIN,MSG9 ;ASCII MESSAGE CALL WITH COMMON HEADER
MOV #100,@ADSR ;SET INTERRUPT ENABLE
CLR @XDAC ;INIT TO LEFT SIDE
MOV #CARGEN,R3 ;INIT POINTER TO CHARACTER GENERATOR
MOV #SWEEP,RSTRT ;MOVE RESTART FOR ENDPASS
MOV #10,@DPSR ;INTENSIFY ON LOAD Y AND INTERRUPT ENABLE
MOV #102,@CLKSR ;INTERUPT ENABLE AND 10USEC. RATE
MOV VECTOR,R0 ;USE R0 TO SET UP VECTORS
TST #NAINT,(R0)+ ;A/D VECTOR
MOV (R0)+
MOV #NCINT,(R0)+ ;CLOCK VECTOR
MOV BR1,(R0)+ ;STATUS
TSTR (R0)+
MOV #NDINT,(R0)+ ;DISPLAY VECTOR
MOV BR2,@R0 ;STATUS
TST -(R0) ;BACK TO DISPLAY VECTOR
MOV #PRIME,R1 ;NUMBER OF FRAMES PER CHANNEL
CLR @CLKBR ;SET INTERVAL
EXITS,BEGIN ;START CLOCK RUNNING
RTI @ADSR,@YDAC ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
RTI ;START AN A/D CONVERSION
NDINT: RTI

;-----
;BIROS,BEGIN,IS ;QUEUE UP TO CONTINUE AT IS AND RTI
;-----
1S: ADD #16,@XDAC ;MOVE RIGHT 16
TST @XDAC ;TEST FOR END OF SCREEN
BNE PRIME ;DO NEXT POINT IF NOT AT END OF SCREEN
MOV #NDINT2,@R0 ;CHANGE VECTOR FOR DISPLAYING CH.#
MOV #10,@R5 ;LOOP COUNTER
MOV R3,R4 ;WORKING POINTER

```

441 000772- 012477 002506
442 000776- 012477 002504
443 001006- 104400 000000-
444
445
446 001006- 000004 000000- 001014-
447
448 001014- 005305
449 001016- 001365
450 001020- 032777 004000 002444
451 001026- 012477 002402
452 001030- 012761 004222-
453 001034- 012705 000012-
454 001040- 012710 001060-
455 001044- 012477 002434
456 001050- 012477 002432
457 001054- 104400 000000-
458 001060-
459
460 001060- 000004 000000- 001066-
461
462 001066- 005305
463 001070- 001365
464 001072- 012710 000736-
465 001076- 005077 002402-
466 001102- 005301
467 001104- 001471
468 001106- 001479 000400 002356
469 001114- 012702 002352
470 001120- 042702 174377
471 001124- 006202
472 001126- 006202
473 001130- 006202
474 001132- 010203
475 001134- 006202
476 001136- 006202
477 001140- 006203
478 001142- 002703 003522-
479 001144- 002703 002320 000100
480 001150- 001250 000000-
481 001156- 104413
482
483 001162- 000200
484 001164- 000156

```

CHNOM: MOV (R4)+,BXDAC ;LOAD X  
MOV (R4)+,BXDAC ;LOAD Y AND INTENSIFY  
EXIT\$,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

NDINT2: ;-----  
;PIRQS,BEGIN,IS ; QUEUE UP TO CONTINUE AT IS AND RTI  
IS: DEC R5 ;COUNT THROUGH LOOP  
BNE CHNOM ;BRANCH IF NOT DONE  
BIT #BIT11, R4DSR ;CH.#10 OR GREATER?  
BEQ CHEXIT ;BRANCH IF NOT  
MOV #D10, R4 ;SET UP TO DISPLAY TEN  
MOV #10, R5 ;LOOP COUNTER  
CHTEN: MOV #NDINT3, R0 ;CHANGE VECTOR AGAIN  
MOV (R4)+,BXDAC ;LOAD X  
MOV (R4)+,BXDAC ;LOAD Y AND INTENSIFY  
EXIT\$,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

NDINT3: ;-----  
;PIRQS,BEGIN,IS ; QUEUE UP TO CONTINUE AT IS AND RTI  
IS: DEC R5 ;COUNT THROUGH LOOP  
BNE CHTEN ;BRANCH IF NOT DONE  
CHEXIT: MOV #NDINT, R0 ;RESTORE DISPLAY VECTOR  
MOV #BXDAC ;BACK TO LEFT SIDE OF SCREEN  
DEC R1 ;COUNT FRAMES  
BNE PRIME ;DO ANOTHER FRAME IF NOT DONE WITH CHANNEL  
ADD #400, R4DSR ;SELECT NEXT CHANNEL  
MOV #R4, R2 ;FETCH STATUS  
BIC #1437, R2 ;ISOLATE LOWER 3 BITS OF CH.#  
ASR R2  
ASR R2  
ASR R2, R3 ;R2 = CH.# X 32  
;R3 = CH.# X 40  
ASR R2  
ASR R2, R3 ;R2 = CH.# X 8  
;R3 = CH.# X 40  
ADD #CARGEN, R3 ;R3 = POINTER TO CORRESPONDING CHARACTER  
CMP #ADSR, #100 ;DONE ALL 16 CHANNELS?  
BNE SWEEP ;DO NEXT CHANNEL IF NOT DONE  
ENDITS,BEGIN ;SIGNAL END OF ITERATION.  
;MONITOR CHANNEL TEST END OF PASS  
;CLOCK INTERVAL  
;FRAMES PER CHANNEL

CLCNT: 200  
FRMCNT: 110.

```

485 001166- 104403 000000- 003422- WRAP:
486 001194- 012477 000002 002274 MSGN$,BEGIN,MSG10 ;ASCII MESSAGE CALL WITH COMMON HEADER
487 001202- 016700 176602 ;SET CLOCK FOR LOS. RATE, SINGLE INTERVAL
488 001206- 012720 002300- ;SET VECTOR AND PSW
489 001212- 116710 176574
490 001216- 012767 001224- 176666 MOV #R1, R0
MOV #ADRMS1, RSTR ;SET RESTART ADDRESS FOR WRAPAROUND MODE
;CALCULATE A/D RMS NOISE USING FINE X WRAPAROUND DAC.
491
492
493
494
495 001224- 012700 000657
496 001230- 012777 004140 002234 ADRMS1: MOV #431, R0 ;R0 = 84.13% OF 512 CONVERSIONS
MOV #410, R4DSR ;CH.#10 CLK. ST. AND INTERRUPT ENABLE
MOV #BXDAC, R1 ;R1 = ADDRESS OF SAR DAC
JSR PC, SAR ;GET DAC VALUE THAT PRODUCES 16/84 SPLIT
MOV #R1, R5 ;SAVE LEFT BOUNDARY
MOV #R1, R0 ;R0 = 15.87% OF 512 CONVERSIONS
JSR PC, SAR ;GET DAC VALUE THAT PRODUCES 84/16 SPLIT
SUB #R1, R5 ;CHANGE TO 15.87% OF 512 CONVERSIONS
MOV #R5, R5 ;R5 = BREADTH OF NOISE @ 68% AREA
CMP #R5, RMLM ;SAVE FOR DAC NOISE CALCULATIONS
BLE ADRMS2 ;< OR = RMS LIMIT?
;IF WITHIN LIMIT THEN CONTINUE AT ADRMS2
JSR PC, ERCON ;GET ERROR PARAMETERS
497
498 001300- 104403 000000- 003250- MSGN$,BEGIN,MSG1 ;ASCII MESSAGE CALL WITH COMMON HEADER
499
500 001306- 012767 000031 176572 MOV #31, ERRTYP ;A/D NOISE ERROR
***** ;A/D RMS NOISE EXCEEDED LIMIT
;*****
501
502 001314- 104405 000000- 003440- HDRRS$,BEGIN,VOLTS ;A/D RMS NOISE EXCEEDED LIMIT
***** ;*****
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520 001322- 012700 000657
521 001326- 012777 003540 002136 ADRMS2: MOV #431, R0 ;R0 = 84.13% OF 512 CONVERSIONS
MOV #3540, R4DSR ;CH.#7, CLK. ST. AND INTERRUPT ENABLE
MOV #BXDAC, R1 ;R1 = SAR DAC
JSR PC, SAR ;GET DAC VALUE THAT PRODUCES 16/84 SPLIT
MOV #R1, R5 ;SAVE DAC VALUE
MOV #R1, R0 ;R0 = 15.87% OF 512 CONVERSIONS
JSR PC, SAR ;GET DAC VALUE THAT PRODUCES 84/16 SPLIT
SUB #R1, R5 ;CHANGE TO 15.87% OF 512 CONVERSIONS
MOV #R5, R5 ;R5 = RMS NOISE X 2
CMP #R5, RMLM ;SAVE FOR DAC NOISE CALCULATIONS
BLE ADRMS2 ;< OR = RMS LIMIT?
;BRANCH IF NO ERROR
JSR PC, ERCON ;GET ERROR PARAMETERS
522
523
524 001352- 004767 000662
525 001356- 161105
526 001360- 010567 002710
527 001364- 020567 002712
528 001370- 003413
529 001372- 004767 001002
530
531 001376- 104403 000000- 003264- MSGN$,BEGIN,MSG2 ;ASCII MESSAGE CALL WITH COMMON HEADER
532
533 001404- 012767 000031 176474 MOV #31, ERRTYP ;A/D NOISE ERROR
***** ;A/D RMS NOISE EXCEEDED LIMIT
;*****
534
535 001412- 104405 000000- 003440- HDRRS$,BEGIN,VOLTS ;A/D RMS NOISE EXCEEDED LIMIT
***** ;*****
536
537

```

```

538
539
540
541 001420 012700 000657
542 001424 012777 002140 002040
543 001432 012777 001000 002046
544 001440 016701 002040
545 001444 004767 000570
546 001450 011105
547 001452 012700 000121
548 001456 004767 000556
549 001464 161105 002540
550 001470 020567 002500
551 001474 003413 002610
552 001476 004767 000676
553
554 001502 104403 000000 003300
555
556 001510 012767 000032 176370
557
558 001516 104405 000000 003440
559
560
561
562
563
564
565
566 001524 012700 000657
567 001530 012777 002140 001734
568 001536 012777 001000 001740
569 001544 016701 001736
570 001550 004767 000464
571 001552 012700 000121
572 001556 004767 000452
573 001564 161105 002500
574 001570 166705 002504
575 001600 003413
576 001602 004767 000572
577
578 001606 104403 000000 003314
579
580 001614 012767 000032 176264
581
582 001622 104405 000000 003440
583
584

```

```

585
586
587
588 001630 012700 000775
589 001634 012777 004140 001630
590 001642 016701 001636
591 001646 004767 000366
592 001652 011105
593 001654 012700 000003
594 001656 004767 000354
595 001664 161105 002404
596 001666 010567 002410
597 001676 003413
598 001700 004767 000474
599
600 001704 104403 000000 003330
601
602 001712 012767 000031 176166
603
604 001720 104405 000000 003440
605
606
607
608
609
610
611 001726 012700 000775
612 001732 012777 003540 001532
613 001740 016701 001542
614 001744 004767 000270
615 001750 011105
616 001752 012700 000003
617 001756 004767 000256
618 001762 161105 002310
619 001764 010567 002312
620 001770 020567
621 001774 003413
622 001776 004767 000376
623
624 002002 104403 000000 003344
625
626 002010 012767 000031 176070
627
628 002016 104405 000000 003440
629

```

```
630 ;CALCULATE Y DAC PEAK NOISE USING COARSE Y, FINE X WRAPAROUND DACS.  
631 YPEAK: MOV #509,R0 ;.6% SPLIT  
632 MOV #3140,@ADSR ;CH.#6, CLK. ST. AND INTERRUPT ENABLE  
633 MOV #1000,@VDAC ;SET Y DAC FOR MID-RANGE  
634 MOV @DAC,R1 ;R1 = ADDRESS OF SAR DAC  
635 JSR PC,SAR ;GET DAC VALUE THAT PRODUCES SPLIT  
636 MOV @R1,R5 ;R5 = LEFT BOUNDARY  
637 MOV @R1,R5 ;R5 = LEFT BOUNDARY  
638 MOV @R3,R0 ;CHANGE SPLIT FOR RIGHT BOUNDARY  
639 JSR PC,SAR ;GET DAC VALUE THAT PRODUCES SPLIT  
640 MOV @R1,R5 ;A/D PLUS Y DAC NOISE  
641 SUB @R1,R5 ;SUBTRACT OUT A/D NOISE  
642 CMP R5,@PLIM ;< OR = DAC PEAK NOISE LIMIT?  
643 BLE @PEAK ;BRANCH IF NO ERROR  
644 JSR PC,ERCOM ;GET ERROR INFO  
645  
646 MSGNS,BEGIN,MSG7 ;ASCII MESSAGE CALL WITH COMMON HEADER  
647  
648 MOV #32,ERRTYP ;D/A ERROR  
649 *****  
650 HDRERS,BEGIN,VOLTS ;Y-D/A PEAK NOISE EXCEEDED LIMIT  
651 *****  
652  
653 ;CALCULATE X DAC PEAK NOISE USING COARSE X, FINE Y WRAPAROUND DACS.  
654 XPEAK: MOV #509,R0 ;.6% SPLIT  
655 MOV #2540,@ADSR ;CH.#5, CLK. ST. AND INTERRUPT ENABLE  
656 MOV #1000,@XDAC ;SET X DAC TO MID-RANGE  
657 MOV @DAC,R1 ;R1 = SAR DAC  
658 JSR PC,SAR ;GET DAC VALUE THAT PRODUCES SPLIT  
659 MOV @R1,R5 ;R5 = LEFT BOUNDARY  
660 MOV @R1,R5 ;R5 = LEFT BOUNDARY  
661 MOV @R3,R0 ;CHANGE SPLIT FOR RIGHT BOUNDARY  
662 JSR PC,SAR ;GET DAC VALUE THAT PRODUCES SPLIT  
663 MOV @R1,R5 ;A/D PLUS X DAC PEAK TO PEAK NOISE  
664 SUB @R1,R5 ;TAKE AWAY A/D NOISE  
665 CMP R5,@PLIM ;< OR = DAC PEAK NOISE LIMIT?  
666 BLE @PEAK ;BRANCH IF WITHIN LIMIT  
667 JSR PC,ERCOM ;GET ERROR PARAMETERS  
668  
669 MSGNS,BEGIN,MSG8 ;ASCII MESSAGE CALL WITH COMMON HEADER  
670  
671 MOV #32,ERRTYP ;D/A ERROR  
672 *****  
673 HDRERS,BEGIN,VOLTS ;X-D/A PEAK NOISE EXCEEDED LIMIT  
674 *****  
675  
676 DONE: ENDS,BEGIN ;SIGNAL END OF ITERATION  
677 ;MONITOR SHALL TEST END OF PASS  
678
```

```
680 ;FIND DAC VALUE WHICH PRODUCES THE NUMBER OF HIGH COUNTS DEFINED IN R0  
681 ;USING SUCCESSIVE APPROXIMATION AND WRAPAROUND DAC DEFINED IN R1.  
682 SAR: MOV #1000,R2 ;R2 = MSB OF DAC  
683 CLR R1 ;GET RID OF "ONES"  
684 BIT: ADD R2,@R1 ;TRY THIS BIT  
685 CLR R3 ;INIT HIGH COUNT  
686 MOV #512,R4 ;R4 = # OF SAMPLES IN A BURST  
687 JSR PC,RANDY ;GET A RANDOM NUMBER  
688 MOV @R4,@CLKBR ;PUT IT IN CLOCK PRESET REGISTER  
689 INC @CLKBR ;START CLOCK RUNNING  
690 EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.  
691  
692 WAIT: PIRQS,BEGIN,15 ;-----  
693 ; QUEUE UP TO CONTINUE AT 15 AND RTI  
694  
695  
696 15: CMP @ADSR,#1000 ;> OR = 1000?  
697 BHI #4 ;BRANCH IF < 1000  
698 INC R4 ;COUNT IF > OR = 1000  
699 BNE CONV ;COUNT THROUGH BURST  
700 CMP R3,R0 ;BRANCH IF NOT DONE  
701 BGT #4 ;HIGH COUNT > # OF 512 CONVERSIONS?  
702 BLE #4 ;BRANCH TO LEAVE BIT IN  
703 SUB R2,@R1 ;TAKE BIT OUT  
704 ASR R2 ;NEXT BIT  
705 BNE BIT ;AND GO RESOLVE IT IF NOT DONE  
706 TST @R1 ;CHECK FOR ALL "ZEROS"  
707 BRPERR ;BRANCH IF INVALID RESULT  
708 MOV @R1,#1777 ;CHECK FOR ALL "ONES"  
709 BRPERR ;BRANCH IF INVALID RESULT  
710 RTS PC  
711  
712 ;VOLTAGE NEEDED TO PRODUCE # OF HIGH COUNTS SPECIFIED IS OUT OF THE  
713 ;RANGE OF THE WRAPAROUND DAC.  
714 ;CLEAR INTERRUPTS, PRINT MESSAGE AND DROP MODULE.  
715 WRPERR: CLR @ADSR  
716 CLR @ADPSR  
717 CLR @CLKBR  
718 MSGNS,BEGIN,MSG11 ;ASCII MESSAGE CALL WITH COMMON HEADER  
719 ENDS,BEGIN ;OUT OF THE DAC'S RANGE  
720
```

```
721 ;CONVTRT NOISE RESULTS TO DECIMAL AND
722 ;MEASURE AR11'S POWER SUPPLY VOLTAGES AND CALCULATE AVERAGE AND SPREAD
723
724 002400* 016767 001066 175472 ERCON: MOV ANSR,CSRA ;LOAD HEADER FOR ERROR CALL
725 002406* 017767 001060 175466 MOV BADSR,ACSR
726 002414* 016767 175408 175482 MOV STAT,STAT
727 002422* 010546 MOV R5,-(SP) ;STACK BINARY NOISE VALUE
728 ;*****
729 ;CONVERT DECIM TO ASCII AND
730 ;STORE AT
731 002424* 104421 000000* 003514* BTODS,BEGIN,DECIM,
732 002432* 000000
733 ;*****
734 002434* 116767 001056 000502 MOVB DECIM+2,VALUE ;MOVE CONVERTED VALUES TO ASCII BUFFER
735 002442* 116767 001051 000476 MOVB DECIM+3,VALUE+2
736 002450* 116767 001044 000471 MOVB DECIM+4,VALUE+3
737 002458* 012777 002636* 175324 MOV #140,R3 ;SETUP VECTOR FOR AVERAGING ROUTINE
738 002464* 012777 001540 001000 JSR #140,R3 ;CH=#3,CLK,ST,AND INTERRUPT ENABLE
739 002472* 012703 001740 ;R3 = EXPECTED VALUE
740 002476* 004767 000074 JSR PC,AVER ;GET AVERAGE VALUE AND COUNT SPREAD
741 002480* 004767 000074 MOV R0,AVP14V ;AVERAGE +14 VOLTS
742 002506* 010167 000746 JSR R0,AVP14V ;COUNT SPREAD ON +14V
743 002512* 012777 002140 000752 MOV #140,BADSR ;CH=#4,CLK,ST,AND INTERRUPT ENABLE
744 002524* 012703 000040 JSR #40,R3 ;R3 = EXPECTED VALUE
745 002528* 004767 000074 JSR PC,AVER ;GET AVERAGE AND SPREAD
746 002530* 010167 000726 MOV R0,AVP14V ;AVERAGE -14V
747 002534* 010167 000724 JSR R0,AVP14V ;COUNT SPREAD ON -14V
748 002546* 012777 027540 000724 MOV #140,BADSR ;CH=#17,CLK,ST,UNIPOLAR AND I. F.
749 002548* 012703 000020 JSR #400,R3 ;R3 = EXPECTED VALUE
750 002552* 004767 000020 JSR PC,AVER ;GET AVERAGE AND SPREAD
751 002556* 010087 000704 MOV R0,AVP5HQ ;AVERAGE +5HQ
752 002560* 010167 000702 MOV R1,APP5HQ ;COUNT SPREAD ON +5HQ
753 002564* 010167 002300* 175214 MOV #AINT,AVPCTOR ;SET VECTOR BACK FOR SAR
754 002574* 000207 RTS PC
```

```
755 ;GET THE AVERAGE OF 512 SAMPLES AND THEIR COUNT SPREAD
756
757 002576* 005000 AVER: CLR R0 ;INIT R0 FOR RUNNING SUM
758 002600* 012704 001000 MOV #512,R4 ;BURST OF 512 CONVERSIONS
759 002604* 012701 100000 MOV #BIT15,R1 ;INIT HIGH COUNT TO NEGATIVE #
760 002610* 012702 040000 MOV #BIT14,R2 ;INIT LOW COUNT TO POSITIVE #
761 002614* 004767 001000 ECON: JSR PC,RANDY ;GET A RANDOM NUMBER
762 002620* 016777 000166 000652 MOV R0,RCLKR ;PUT IT IN CLOCK PRESET REGISTER
763 002626* 005277 000644 INC RCLKR ;START CLOCK RUNNING
764 002632* 104400 000000* ERINT: EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
765 002636*
766 ;-----
767 002636* 000004 000000* 002644* ;PIRQS,BEGIN,IS ; QUEUE UP TO CONTINUE AT IS AND RTI
768 ;-----
769 002644* 057700 000624 1S: ADD @ADDR,R0 ;R0 = RUNNING SUM OF DIFFERENCES
770 002650* 160300 SUB R3,R0 ;OFFSET BY EXPECTED VALUE
771 002652* 027701 000616 CMP @ADDR,R1 ;HIGHER THAN PREVIOUS HIGH?
772 002656* 003402 BLE 2S ;BRANCH IF NOT A NEW HIGH
773 002660* 017701 000610 MOV @ADDR,R1 ;NEW HIGH RESULT
774 002664* 027702 000604 2S: CMP @ADDR,R2 ;LOWER THAN PREVIOUS LOW?
775 002670* 002002 BGE 3S ;BRANCH IF NO
776 002672* 017702 000576 3S: MOV @ADDR,R2 ;NEW LOW RESULT
777 002676* 005304 DEC R4 ;COUNT THROUGH BURST
778 002700* 001345 BNE ECON ;DO ANOTHER SAMPLE IF NOT DONE
779 002702* 000300 SWAB R0 ;DIVIDE SUM OF DIFFERENCES BY 256
780 002704* 110000 MOV R0,R0 ;SIGN EXTEND
781 002706* 006200 ASR R0 ;DIVIDE BY 2 MORE
782 002710* 005500 ADD R0,R0 ;ROUND UP FOR > 1/2
783 002712* 060300 ADD R3,R0 ;OFFSET AVERAGE BY EXPECTED RESULT
784 002714* 160201 SUB R2,R1 ;R1 = COUNT SPREAD
785 002716* 000207 RTS PC
786
787 ;GENERATE A RANDOM NUMBER
788
789
790 002720* 066767 000070 000064 RANDY: ADD RNB,RNA
791 002726* 066767 000064 000056 ADD RNC,RNA
792 002734* 005567 000052 ADC RNB,RNA
793 002740* 066767 000046 000046 ADD RNC,RNB
794 002746* 066767 000044 000046 ADD RNB,RNC
795 002754* 005567 000034 ADC RNC,RNB
796 002760* 066767 000026 000030 ADD RNA,RNC
797 002766* 066767 000022 000022 ADD RNB,RNC
798 002774* 005567 000016 ADC RNC,RNC
799 003000* 126727 000006 000300 CMP RNA,#300 ;CHECK FOR SMALL NEGATIVE NUMRERS
800 003006* 101344 BHI RANDY ;GET ANOTHER RANDOM NUMBER
801 003010* 000207 RTS PC
802 003012* 063311 RNB: 063241
803 003014* 142315 RNC: 142315
804 003016* 127623 RNC: 127623
```

```

XARABO.P11 12-OCT-78 11:45
805
806
807 003020 040445 042057 000040 P1: .ASCIZ "%A/D "
808 003026 040445 042055 040457 P2: .ASCIZ "%X-D/A "
809 003034 000040
810 003036 054445 042055 040457 P3: .ASCIZ "%V-D/A "
811 003044 000040
812 003046 046577 020123 000 P4: .ASCIZ "RMS "
813 003053 000
814 003054 042520 045501 000040 P5: .ASCIZ "PEAK "
815 003062 047516 051511 020105 P6: .ASCIZ "NOISE = "
816 003070 020075 000
817 003073 000
818 003074 027060 032462 046040 P7: .ASCIZ "0.25 LSB)"
819 003102 041123 000051
820 003106 027062 030060 046040 P8: .ASCIZ "2.00 LSB)"
821 003114 041123 000051
822 003120 027060 032462 046040 P9: .ASCIZ "0.25 LSB)"
823 003126 041123 000051
824 003132 027062 030060 046040 P10: .ASCIZ "2.00 LSB)"
825 003140 041123 000051
826 003144 027130 054130 046040 VALUE: .ASCIZ "X.XX LSB (LIMIT = "
827 003152 041123 024040 044514
828 003160 044070 020124 020075
829 003166 000
830 003167 000
831 003170 040445 030522 020061 P11: .ASCIZ "%AR11 RUNNING IN "
832 003176 052527 047116 047111
833 003177 052527 047111 000040
834 003212 047516 026516 000 P12: .ASCIZ "NON-"
835 003217 000
836 003220 051127 050101 051101 P13: .ASCIZ "WRAPAROUND "
837 003226 051127 042116 000440
838 003234 047515 042504 000 P14: .ASCIZ "MODE"
839 003241 000
840 003250 003105 047522 000122 P15: .ASCIZ "ERROR"
841 003252 003046 P1
842 003254 003062 P4
843 003256 003144 P6
844 003256 003144 VALUE
845 003262 177777 P7
846 003262 177777 P9
847 003264 003020 MSG2: P1
848 003266 003040 P4
849 003266 003040 P6
850 003272 003144 VALUE
851 003274 003074 P7
852 003276 177777 P7

```

```

XARABO.P11 12-OCT-78 11:45
853 003300 003036 MSG3: P3
854 003302 003040 P4
855 003304 003040 P6
856 003306 003144 VALUE
857 003310 003120 P9
858 003314 177777
859 003314 003026 MSG4: 177777
860 003316 003046 P4
861 003320 003062 P6
862 003324 003144 VALUE
863 003324 003144 P9
864 003326 177777
865 003330 003020 MSG5: 177777
866 003332 003054 P1
867 003334 003062 P5
868 003336 003144 VALUE
869 003340 003106 P8
870 003342 177777
871 003344 003020 MSG6: 177777
872 003346 003054 P5
873 003350 003062 P6
874 003352 003144 VALUE
875 003354 003106 P9
876 003356 177777
877 003360 003036 MSG7: P3
878 003362 003054 P5
879 003364 003062 P6
880 003366 003144 VALUE
881 003370 003120 P10
882 003372 177777
883 003374 003026 MSG8: 177777
884 003376 003054 P2
885 003400 003062 P6
886 003402 003144 VALUE
887 003404 003106 P9
888 003406 177777
889 003410 003170 MSG9: 177777
890 003412 003212 P11
891 003414 003220 P12
892 003416 003234 P13
893 003420 177777
894 003422 003170 MSG10: 177777
895 003424 003220 P14
896 003426 003234 P13
897 003430 177777
898 003432 003220 MSG11: 177777
899 003434 003170 P15
900 003436 177777 P15

```

901 003440 003456  
 902 003442 003460  
 903 003444 003462  
 904 003446 003464  
 905 003450 003464  
 906 003452 003470  
 907 003454 177777  
 908 003456 000000  
 909 003460 000000  
 910 003462 000000  
 911 003464 000000  
 912 003466 000000  
 913 003470 000000  
 914 003472 000000  
 915 003474 000000  
 916 003476 000000  
 917 003500 000000  
 918 003502 000000  
 919 003504 000000  
 920 003506 000000  
 921 003510 000000  
 922 003512 000000  
 923 003514 000003  
 924 003522 001000 000004 001002  
 925 003524 000010 001002 000014  
 926 003536 001006 000000 001006  
 927 003544 000020 001012 000000  
 928 003552 001014 000020 001016  
 929 003560 000000 000000 000010  
 930 003566 001016 000014  
 931 003572 001004 000000 001005  
 932 003600 000010 001006 000000  
 933 003602 000012 001010 000000  
 934 003614 000004 001010 000010  
 935 003622 001010 000014 001010  
 936 003630 000020 001012 000000  
 937 003636 001012 000006  
 938 003642 001002 000000 001002  
 939 003650 000004 001006 000000  
 940 003656 001006 000010 001006  
 941 003664 000010 001012 000000  
 942 003672 001012 000010 001012  
 943 003700 000020 001016 000000  
 944 003706 001016 000014  
 945 003712 001002 000010 001002  
 946 003720 000020 001006 000000  
 947 003726 001006 000010 001006  
 948 003734 000020 001012 000000  
 949 003742 001012 000010 000014  
 950 003750 000020 001016 000004  
 951 003756 001016 000014

VOLTS: AVP14V  
 SPP14V  
 AVN14V  
 SPN14V  
 AVPSHO  
 SPPSHO  
 177777  
 AVP14V: OPEN  
 SPP14V: OPEN  
 AVN14V: OPEN  
 SPN14V: OPEN  
 AVPSHO: OPEN  
 SPPSHO: OPEN  
 ADBR: OPEN  
 ADDBR: OPEN  
 CLKSR: OPEN  
 CLRR: OPEN  
 DPSR: OPEN  
 XDAC: OPEN  
 YDAC: OPEN  
 CLKWT: OPEN  
 BRPFLG: OPEN  
 DECIM: -BLKW  
 CARGEN: -WORD

3 1002,4,1002,10,1002,14,1006,0,1006,20,1012,0,1012,20,1016,4,1016,10,1016  
 .WORD 1004,0,1005,15,1006,0,1010,0,1010,4,1010,10,1010,14,1010,20,1012,0,1014,  
 .WORD 1002,0,1002,4,1006,0,1006,10,1006,20,1012,0,1012,10,1012,20,1016,0,1016,  
 .WORD 1002,0,1002,20,1006,0,1006,10,1006,20,1012,0,1012,10,1012,20,1016,4,1016

952 003762 001002 000010 001002  
 953 003770 000014 001002 000020  
 954 003776 001006 000010 001012  
 955 004004 000000 001012 000004  
 956 004010 001010 000010 001012  
 957 004020 000014 001012 000020  
 958 004026 001016 000010  
 959 004032 001002 000002 001002  
 960 004040 001012 000014 000020  
 961 004048 001007 000006 001007  
 962 004054 000012 001010 000020  
 963 004062 001014 000002 001014  
 964 004070 000011 001016 000006  
 965 004076 001016 000020  
 966 004102 001002 000004 001002  
 967 004110 000010 001002 000014  
 968 004116 001006 000000 001006  
 969 004124 000010 001006 000020  
 970 004132 001012 000000 001012  
 971 004140 000010 001012 000020  
 972 004146 001016 000004  
 973 004152 001002 000020 001005  
 974 004160 000020 001010 000020  
 975 004166 001011 000000 001012  
 976 004174 000004 001013 000007  
 977 004202 001013 000020 001014  
 978 004210 000012 001015 000015  
 979 004216 001016 000020  
 980 004222 000765 000000 000762  
 981 004230 000765 000000 000765  
 982 004236 000765 000000 000765  
 983 004244 000004 000765 000010  
 984 004252 000765 000014 000765  
 985 004260 000000 000767 000000  
 986 004266 000771 000000  
 987  
 988  
 989  
 990 004272 000000  
 991 004274 000000  
 992 004276 000000  
 993 004300 000000  
 994  
 995  
 996  
 997 004302 000031  
 998 004304 000031  
 999 004306 000310  
 1000 004310 000310  
 1001 000001

ARMX: OPEN  
 ARMY: OPEN  
 APX: OPEN  
 APKY: OPEN

NOISE LIMITS IN 1/100THS LSB

ARMW: 25. ;A/D RMS NOISE LIMIT  
 DRMLM: 25. ;D/A RMS NOISE LIMIT  
 APKLM: 200. ;A/D PEAK NOISE LIMIT  
 DPKLM: 200. ;D/A PEAK NOISE LIMIT  
 .END

D10: .WORD 761,0,762,15,763,0,765,0,765,4,765,10,765,14,765,20,767,0,771,0

.WORD 1002,10,1002,14,1002,20,1006,10,1012,0,1012,4,1012,10,1012,14,1012,20,10  
 .WORD 1002,2,1002,12,1002,20,1007,0,1007,12,1010,20,1014,2,1014,11,1016,6,1016  
 .WORD 1002,4,1002,10,1002,14,1006,0,1006,10,1006,20,1012,0,1012,10,1012,20,101  
 .WORD 1002,20,1005,20,1010,20,1011,0,1012,4,1013,7,1013,20,1014,12,1015,15,101

ARAB DEC/11 SYSTEM EXERCISER MODULE  
 KARABO. P11 12-OCT-78 11:45

MACV11 30A(1052) 12-OCT-78 16:17 PAGE 26  
 CROSS REFERENCE TABLE -- USSR SYMBOLS

SEQ 0024

ACSR	000102R	316#	725*													
ADDR	000006R	282#	429	696	769	771	773	774	776	915#						
ADDR22=	001000	334#	335													
ADPK1	001630R	575	587#													
ADPK2	001636R	597	610#													
ADRMS1	001226R	491	495#													
ADRMS2	001322R	505	518#													
ADSR	003472R	336*	393*													
APKLI	004306R	611*	633*	408*	427*	450	468*	469	479	496*	519*	541*	565*	588*		
APKX	004276R	595*	641													
APKY	004300R	618*	665													
ARMIL	004302R	356*	504													
ARMY	004274R	526*	573													
ASB	000106R	320#														
ASTAT	000104R	318#	726*													
AVEP	002576R	740	745													
AVN14V	003462R	746*	903		757#											
AVP14V	003456R	741*	901													
AVPSHQ	003466R	751*	905													
AWAS	003466R	751*	905													
BEGIN	000000R	379#	356	365	374	383	395	396	407	426	433	443	446	457		
		460	481	486	508	512	531	535	564	568	578	582	600	604		
		463	627	646	650	670	674	678	691	694	719	720	731	731	764	
		451#	705													
BIT	002246R	685#														
BIT0	000001	334#														
BIT1	000002	334#														
BIT10	000000	334#														
BIT11	004000	334#	450													
BIT12	010000	334#														
BIT13	020000	334#														
BIT14	040000	334#														
BIT15	100000	334#	760													
BIT2	000004	334#	759													
BIT3	000010	334#														
BIT4	000100	334#														
BIT5	000040	334#														
BIT6	000100	334#														
BIT7	000200	334#														
BIT8	000400	334#														
BIT9	001000	334#														
BREAK	104407	334#	395	396												
BR1	000012R	284#	391	418	490											
BR2	000013R	285#	421													
BT0DS	104421	334#	365	374	383	731										
CARGEW	003522R	410	478													
CHRTAS	104442	334#	464#													
CHKI1	001042R	465	464#													
CHNUM	000772R	444#	449													
CHTEN	001044R	455#	463													
CLCMT	001522R	424*	483#	689*	762*	917#										
CLCNR	003500R	443*	474#													
CLRCNT	003510R	356*	921#													

ARAB DEC/11 SYSTEM EXERCISER MODULE  
 KARABO. P11 12-OCT-78 11:45

MACV11 30A(1052) 12-OCT-78 16:17 PAGE 27  
 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0025

CLCSR	003476R	340*	413*	425*	487*	690*	718*	763*	916#							
COMFIG	000056R	304#														
CONV6	00726R	688#	700													
CSRA	000100R	334#	724*													
DATCKS=	104411	334#														
DATERE=	104404	334#														
DECTH	003514R	336														
DHFE	002234R	329	359	360												
DPLIM	004502R	337	677#	388	351	365	368	369	370	374	377	378	379	383		
DPSR	004502R	383	666	1000#												
DRML	004304R	374	394*	514	717*	918#										
DVID1	000014R	286#	550		998#											
DIO	004222R	452														
EDCN	004102R	751#	980#													
ENDITS=	104413	334#	778													
ENDS	104410	334#	790	678												
ERCOM	002400R	596	529													
ERINT	002636R	737	755*	552	576	598	621	644	668	724#						
ERRTPP	000106R	319#	533*	533*	556*	598	621	644	668	724#						
EXITS	104400	334#	426	443	457	580*	602*	625*	648*	672*						
FRMCNT	001164R	423#	484#													
GTPAS	104415	334#														
GWBVS	104414	334#														
HRDCNT	000044R	299#														
HRDERS	104405	334#	512	535	558	582	604	627	650	674						
HRDPLS	000050R	301#														
ICONT	000036R	267#														
ICOUNT	000040R	267#														
IDNUM	000122R	326#														
IINT	000030R	292#														
INTR	000120R	325#	405*													
LIMIN	000306R	352#														
NAP22S	104416	334#														
MODNAM	000000R	280#														
MODSP	000224R	294	332#													
MSCNS	104403	334#	407	486	508	531	554	578	600	623	646	670	719			
MSCS	104402	334#														
MSG1	003250R	508	841#													
MSG10	003422R	486	894#													
MSG11	003432R	719	898#													
MSG2	003264R	521	847#													
MSG3	003300R	521	859#													
MSG4	003314R	578	865#													
MSG5	003330R	600	871#													
MSG6	003448R	623	878#													
MSG7	003360R	624	883#													
MSG8	003374R	670	889#													
MSG9	003410R	407	894#													
NAINT	000126R	415	429#													
NCTN	000726R	420	431#													
NDINT	000736R	420	431#	464												



NULL = 000000	334#																	
OPEN = 000000	281	287	288	289	290	307	308	309	310	311	312	313	314					
	316	318	320	321	323	324	325	334#	908	909	910	911	914					
	913	914	915	916	917	918	919	920#	921	922	990	991	992					
UTDAS = 104420	334#																	
PASCNT = 000034R	295#																	
PTQ3 = 000004	334#	433	446	460	694	767												
POPSP = 005726	334#																	
POPSP2 = 022626	334#																	
PRIME = 000702R	424#	437	467															
PRTV = 000000	334#																	
PRTV0 = 000000	334#																	
PRTV1 = 000040	334#																	
PRTV2 = 000100	334#																	
PRTV3 = 000140	334#																	
PRTV4 = 000200	285	334#																
PRTV5 = 000240	334#																	
PRTV9 = 000300	284	334#																
PRTV7 = 000340	334#																	
PS = 177476	334#																	
PSW = 177776	334#																	
PUSH = 005746	334#																	
PUSH2 = 074466	334#																	
P10 = 003020R	801#	841	847	865	871													
P10 = 003132R	386*	387*	388*	824#	881	887												
P11 = 003170R	831#	889	894															
P12 = 003200R	830	830																
P13 = 003240R	834#	890	895	898														
P14 = 003234R	838#	892	896															
P15 = 003242R	840#	899																
P16 = 003272R	834#	856																
P17 = 003036R	810#	853																
P4 = 003046R	812#	842	848	854	860													
P5 = 003054R	814#	866	872	878	884													
P6 = 003062R	816#	843	846	855	861	867	873	879	885									
P7 = 003064R	352*	361*	361*	618*	845	851												
P8 = 003106R	368*	369*	370*	820#	869	875												
P9 = 003120R	377*	378*	379*	822#	857	863												
RANDY = 003120R	698	761	790#	800														
RANUM = 000054R	303#																	
RESTRT = 000224R	322	335#																
RES2 = 000066R	302#																	
RNA = 003012R	689	762	790*	791*	792*	793	796	799	802#									
RNB = 003014R	790	793*	794*	795*	797	803#												
RNC = 000056R	791	794	796*	797*	798*	804#												
RSTRT = 000112R	322#	411*	491*															
SAINT = 000550R	390	402*																
SAR = 002240R	498	501	521	524	544	547	568	571	590	593	613	616	636					
SADR = 000102R	632#	660	663	683#														
SENSE = 000456R	389#																	
SOPCNT = 000042R	298#																	
SOPERS = 104406	334#																	

SOPPAS = 000046R	300#																	
SPR14V = 003464R	747*	904	911#															
SPD1MT = 000032R	294#																	
SPF14V = 003460R	742*	902	909#															
SPP5HQ = 003470R	752*	906	913#															
SPS1Z = 000040	327																	
SR1 = 000016R	287#																	
SR2 = 000020R	288#																	
SR3 = 000022R	289#																	
SR4 = 000024R	290#																	
START = 000224R	293	334#																
STRT = 000020R	292	726																
SVR0 = 000062R	307#																	
SVR1 = 000064R	308#																	
SVR2 = 000066R	309#																	
SVR3 = 000070R	310#																	
SVR4 = 000072R	311#																	
SVR5 = 000074R	312#																	
SVR6 = 000076R	313#																	
SWEEP = 000676R	411	423#	480															
SYSCNT = 000052R	302#																	
TRPDFD = 000022	334#																	
VALUE = 003144R	734*	735*	736*	826#	844	850	856	862	868	874	880	886						
VECTOR = 000010R	283	289	414	488	737*	753*												
VOLTS = 003440R	512	535	558	582	604	627	650	674	901#									
WAINT = 002300R	489	692#	753															
WASADR = 000104R	317#																	
WDR = 000116R	323#																	
WDT0 = 000114R	322#	400*	406*															
WRAP = 001166R	401	485#																
WRPFRR = 002352R	707	709	716#															
WRPFLC = 003512R	392*	397	402*															
XDAC = 003504R	346*	409*	435*	922#	441*	455*	465*	497	543	566*	589	635	658*					
	919#			436														
XFLAG = 000005R	281#																	
XPEAK = 002130R	643	656#																
XRMS = 001524R	593	564#																
YDAC = 003506R	348*	429*	442*	456*	520	542*	567	612	634*	659	920#							
YPEAK = 002024R	620	632#																
YRMS = 001420R	528	540#																
	697	702	923#															

- ARS. 000000 000  
 004312 001