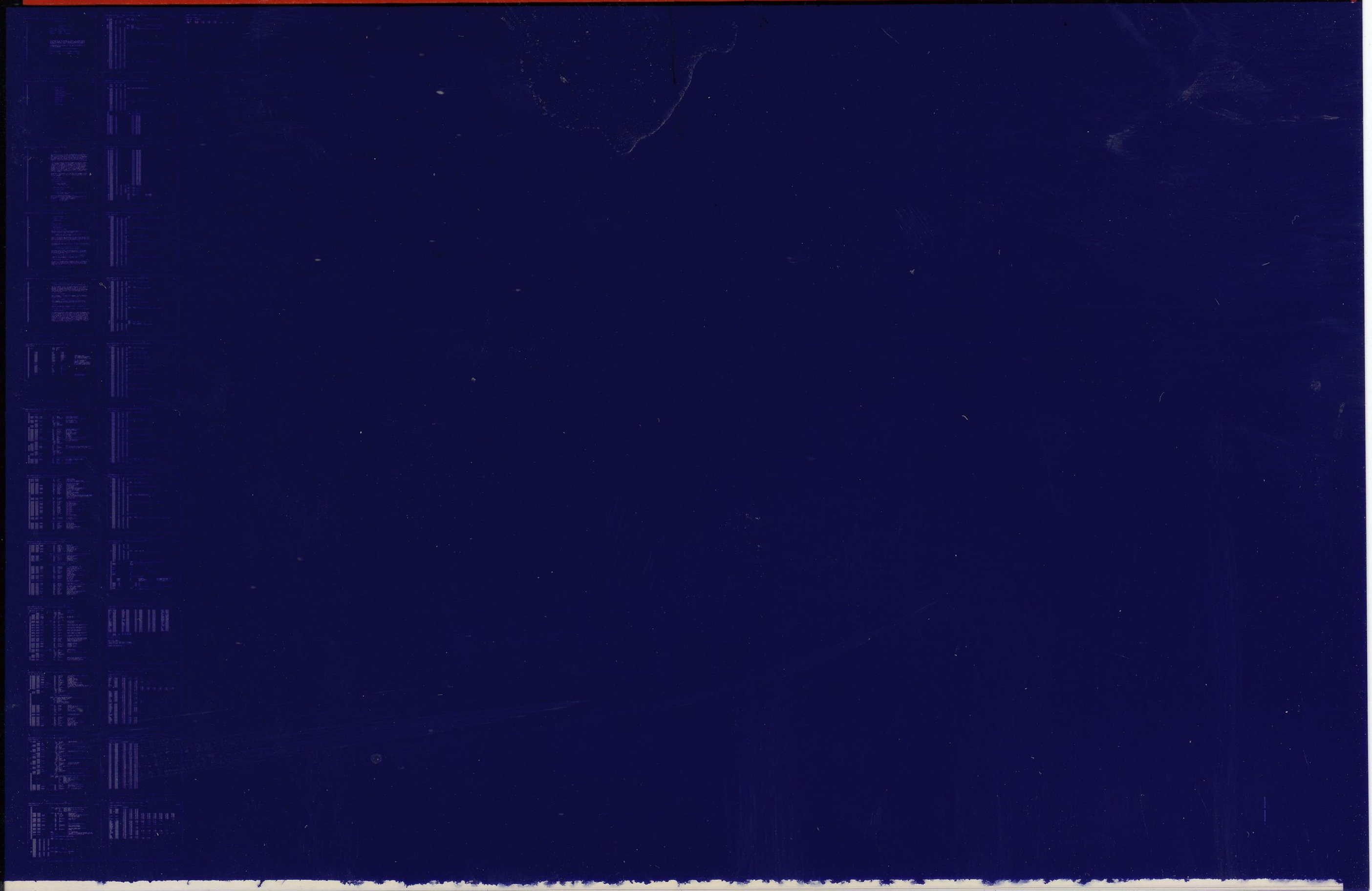


KDJ11-B

EEPROM ITA LANG LDR  
COEEEA0

AH-FF25A-MC  
1 OF 1 JUL 1985  
COPYRIGHT © 1985

**digital**  
MADE IN USA



A ::  
1  
COEEEE EPROM ITA LANG LDR

MACRO Y05.02 Saturday 16-Feb-85 22:46 Page 1

.TITLE COEEEE EPROM ITA LANG LDR

.REM &

IDENTIFICATION  
-----

PRODUCT CODE: AC-FF24A-MC  
PRODUCT NAME: COEEEO EPROM ITA LANG LDR  
PRODUCT DATE: FEBRUARY, 1985  
MAINTAINER: DIAGNOSTIC ENGINEERING

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 DOCUMENT.

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1985 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

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

47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70

TABLE OF CONTENTS

- 1. PROGRAM ABSRACT
- 2. SYSTEM REQUIREMENTS
- 3. LOADING AND STARTING PROCEDURES
- 4. SPECIAL ENVIRONMENTS
- 5. PROGRAM OPTIONS
- 6. EXECUTION TIMES
- 7. ERROR INFORMATION
- 8. EXAMPLES
- 9. PROGRAM DESCRIPTION

72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128

## 1. PROGRAM ABSTRACT

The KDJ11-B is a PDP-11 CPU that incorporates the J11 chip set as the heart of the processor. It is a quad height Q22 bus module. The KDJ11-B has two on-board ROM's. One of them, the 16-bit addressable ROM, contains the self-test and the boot codes. The other ROM, the 8-bit addressable one, contains the base area with hardware selection parameters, optional bootstraps, optional UFD (User Friendly Diagnostic) system description area, and optional foreign language text.

On units to be shipped to non-English speaking countries, a dummy or "null" language is loaded into the EEPROM. The purpose of this is to disable English language error messages when the system is first installed. If and when the system passes its internal self tests, the user will be instructed to run a UFD (User Friendly Diagnostics) package which will be part of a "country kit" for each separate language. The UFD package will use the local language for the particular country and, in addition, will load diagnostic and error messages in the local language into the EEPROM, so each subsequent power-up or reboot will have diagnostic and error messages in the user's own language.

The purpose of this program is to load the local language into the EEPROM. If it detects an error, the program will attempt to restore the "old" language, if any and will print a message informing the user of that fact.

## 2. SYSTEM REQUIREMENTS

### Hardware Requirements

To run successfully this utility needs:

1. KDJ11-B CPU module
2. console terminal
3. at least 28K of memory

## 3. LOADING AND STARTING PROCEDURES

To start-up this program:

1. Boot XXDP.
2. Type "R NAME", where NAME is the name of the BIN or BIC file for this program.

The starting address of the program is 1000.

Note: if trying to restart the program in an arbitrary place after HALT on Break the following registers should be set up:

17777572=0	to disable memory management
17777520=1000	to clear diagnostic mode (bit 8), but still save HALT on Break
17777746=400	to flush the cache

130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186

#### 4. SPECIAL ENVIRONMENTS

The program is not APT compatible.

#### 5. PROGRAM OPTIONS

None.

#### 6. EXECUTION TIMES

The program runs in under 20 seconds.

#### 7. ERROR INFORMATION

##### 7.1 DEFECTIVE BYTE IN EEPROM

After each write, the Byte which should have been written is compared to the Byte in the proper location, and if it is not correct, the following error message is displayed:

EEPROM write error, PCR page n, address mmmmm.  
Data written qqq, data read rrr.

where n is the EEPROM page selected by the Page Control Register (PCR), mmmmm is the physical address of the bad byte in question, qqq is the byte value that was written out to the address and rrr what was read back in after the write. (should be identical to qqq)

##### 7.2 PROCESSOR NOT KDJ11-B

The program checks the type of CPU it is running on, which must be a KDJ11-B processor (MFPT returns 5 in r0). If not, the following message is printed:

Language area not supported by this processor.

##### 7.3 "OLD" BOOT ROM CODE, LANGUAGE AREA NOT SUPPORTED

The program checks to see if the ROM code version is 7.0 or later. Earlier versions do not support the language area in the EEPROM and would print garbage if one was loaded. The program prints the following message in that case:

Current Boot ROM version does not support language area.

In addition, the language bit in the setup area of the EEPROM is cleared, to prevent "garbage" from being printed.

##### 7.4 CHECKSUM ERROR IN SETUP AREA

The checksum in the setup area is checked to see if it contains a valid checksum. Also, bytes 6 and 103 (addresses 17765022 and 17765314, respectively) are checked to see if they contain 0 and 252 octal, respectively. If any of these conditions is not met, the following message is printed:

EEPROM checksum error in setup area.

187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232

No attempt is made to correct a checksum error.

#### 7.5 DIFFERENCES BETWEEN UFD "QUIET" MODE AND "STANDALONE" MODE

When this program is run in UFD "Quiet" mode (which will usually be the case) none of the error messages will appear. If no error is detected, no messages whatsoever are printed. If any error is detected, the program will attempt to restore the UFD and language areas to the state they were in when the program was started. If the restoration was successful, the following message is printed in the user's language:

Unable to load <language>

where <language> is the name of the language. If the restoration was not successful, or there was no local language, the following message is printed.

Unable to load <language> - reverting to U.S. English

where <language> is as above. The program then clears the bit in the EEPROM setup area selecting a local language which means that the ROM English will be used from now on.

#### 8. EXAMPLES

After booting XXDP+ and running the program, no message should appear, just the XXDP dot prompt ( . )

If a problem occurred, one of the messages in section 7 should appear.

#### 9. PROGRAM DESCRIPTION

The program consists of a body of code which loads the language into the local language area of the EEPROM. The routine that performs the write first checks the current value of the byte to be written and if it is the same, no write is performed. This is done to extend the life of the EEPROM. The write routine also checks the value in the EEPROM after the write to insure it was written correctly. After a successful run, no message appears, after an unsuccessful attempt to write any of the bytes in the EEPROM, one of the messages in section 7 appears. If run under UFD "Quiet" mode, no message is printed if the program was successful, otherwise one of the messages in 7.5 appear. In both cases, the XXDP prompt appears.

&

## PROGRAM CONSTANTS

234		.SBTTL	PROGRAM CONSTANTS		
235	000000	.ENABL	ABS		
236		.NLIST	MD,CND		
237		.LIST	ME		
238					
239	177520	BCSR	=	177520	
240	177522	PCR	=	177522	
241	177522	PCRLB	=	177522	
242	165000	E2PROM	=	165000	
243	165316	E2PAR	=	E2PROM+316	;E2PROM PARITY BYTE
244	165006	E2LLB	=	E2PROM+6	;LOCAL LANGUAGE BIT IN E2PROM
245	166000	ENDE2R	=	E2PROM+1000	;LAST ADDRESS OF E2PROM+2
246	173002	RMVTST	=	173002	;WORD TO TEST ROM VERSION NUMBER
247	025370	DELAY	=	11000.	
248	000140	LNGHDR	=	140	;I.D. OF A LANGUAGE AREA
249	000040	UFDHDR	=	040	;I.D. OF A UFD BLOCK
250	000002	RETRY	=	2	;NUMBER OF ATTEMPTS TO WRITE A
251					;BYTE IN E2PROM BEFORE GIVING UP
252	000004	MAXERR	=	4	;NO. OF ERRORS ALLOWED IN LOCAL
253					;LANGUAGE TEXT BEFORE QUITTING
254	177524	BDR	=	177524	
255	000015	CR	=	15	
256	000012	LF	=	12	
257	000200	BIT7	=	200	
258	000100	BIT6	=	100	
259	000011	tab	=	11	
260	000010	backsp	=	10	
261	000040	space	=	40	
262	000033	esc	=	33	
263					
264	001667	ROMSZ	=	FLEND-TEXT	;SIZE IN BYTES OF TEXT TO BE
265					;LOADED INTO EEPROM
266					
277					
298					

CHECK FOR CERTAIN EXCEPTIONS FIRST

```

310                                     .SBTTL CHECK FOR CERTAIN EXCEPTIONS FIRST
311
312         001000                       .=1000
313
314 001000 005037 177522      START: CLR      @#PCR           ;SELECT PAGE 0 OF EEPROM
315 001004 013746 177520      MOV      @#BCSR,-(SP)      ;SAVE OLD BCSR VALUE
316 001010 112737 000067 177520  MOVB     #67,@#BCSR       ;WRITE ENABLE THE E2PROM & ENABLE ROM
317
318 001016 000007      MFPT           ;GET PROCESSOR TYPE
319 001020 020027 000005      CMP      R0,#5           ;CHECK TO SEE IF ORION
320 001024 001404      BEQ      1$             ;YES - CONTINUE
321 001026      .TYPMSG #FMSG2      ;FIELD-SERVICE MESSAGE
          000001      .NARG      NARGS
          000027      .NTYPE     NTYPE,#FMSG2
          001026 012700 002563      MOV      #FMSG2,R0
          001032 104003      EMT      3
322 001034 000443      BR      99$
323
324 001036 012700 165000      1$:  MOV      #E2PROM,R0      ;STARTING ADDRESS TO CHECKSUM
325 001042 005001      CLR      R1             ;INITIALIZE CHECKSUM
326 001044 012703 000151      MOV      #105.,R3       ;NO. OF BYTES TO CKSUM
327 001050 012005      201$: MOV      (R0)+,R5       ;GET A BYTE
328 001052 042705 177400      BIC      #177400,R5     ;NO BUS NOISE, THANK YOU.
329 001056 060501      ADD      R5,R1         ;ACCUMULATE CHECKSUM
330 001060 077305      SOB     R3,201$       ;CONTINUE TILL DONE
331 001062 105701      TSTB    R1             ;IS CKSUM 0?
332 001064 001007      BNE     202$          ;NO, ERROR
333 001066 105737 165022      TSTB    @#E2PROM+22    ;BYTE TO TEST FOR VALID ROM, SHOULD BE 0
334 001072 001004      BNE     202$          ;NO, ERROR
335 001074 123727 165314 000252  CMPB    @#E2PROM+314,#252 ;BYTE TO TEST FOR VALID ROM
336 001102 001404      BEQ     300$          ;GO TO NEXT CHECK IF OK
337 001104      202$: .TYPMSG #FMSG4      ;FIELD SERVICE MESSAGE
          000001      .NARG      NARGS
          000027      .NTYPE     NTYPE,#FMSG4
          001104 012700 002737      MOV      #FMSG4,R0
          001110 104003      EMT      3
338 001112 000414      BR      99$           ;QUIT
339 001114 005067 001304      300$: CLR      OLDSIZ       ;SET FLAG THAT ROM EXISTS, CURRENTLY NO LANGUAGE
340 001120 012737 000016 177522  MOV      #7*2,@#PCR     ;SEL. LAST PAGE OF 2K E2PROM, PGO OF ROM
341 001126 023727 173002      CMP      @#RMVTST,(PC)+ ;SEE IF ROM VER. 7 OR LATER (CAN SUPPORT LANGUAGE AREA)
342 001132 000250      CLN
343 001134 001405      BEQ     2$            ;YES - CONTINUE
344 001136      .TYPMSG #FMSG3
          000001      .NARG      NARGS
          000027      .NTYPE     NTYPE,#FMSG3
          001136 012700 002644      MOV      #FMSG3,R0
          001142 104003      EMT      3
345 001144 000167 000636      99$:  JMP      QUIT1
346
347                                     .SBTTL SAVE OLD LANGUAGE/UPD AREA IN CASE IT MUST BE RESTORED
348
349 001150 012700 165776      2$:  MOV      #ENDE2R-2,R0  ;LAST ADDRESS (CKSUM) OF E2PROM
350 001154 012701 000005      MOV      #5,R1         ;NO. OF BYTES IN HEADER TO CHECKSUM
351 001160 010005      MOV      R0,R5         ;SAVE ADDRESS
352 001162 005003      CLR      R3
353 001164 111004      4$:  MOVB     (R0),R4       ;GET A BYTE
354 001166 060403      ADD     R4,R3         ;ACCUMULATE CHECKSUM

```

SAVE OLD LANGUAGE/UFD AREA IN CASE IT MUST BE RESTORED

```

355 001170 005740      TST      -(R0)      ;CORRECT ADDRESS
356 001172 077104      SOB      R1,4$     ;LOOP FOR 5 BYTES
357 001174 105703      TSTB    R3         ;IF NOT ZERO, NO LANGUAGE LOADED
358 001176 001131      BNE     WRLANG    ;NON-EXISTANT OR CORRUPTED LANGUAGE - SKIP
359
360 001200 014504      MOV      -(R5),R4  ;HIGH BYTE OF BYTE COUNT
361 001202 014546      MOV      -(R5),-(SP) ;LOW BYTE OF BYTE COUNT
362 001204 110466 000001  MOVB    R4,1(SP)  ;SET UPPER BYTES OF SIZE
363 001210 042704 177437  BIC     #177437,R4 ;EXTRACT ID CODE
364 001214 012601      MOV      (SP)+,R1  ;GET SIZE BACK
365 001216 042701 160000  BIC     #160000,R1 ;R1 NOW CONTAINS SIZE OF BLOCK IN BYTES
366 001222 062701 000005  ADD     #5,R1     ;ADD BYTE COUNT FOR HEADER BLOCK
367 001226 120427 000040  CMPB   R4,#UFDHDR ;SEE IF IT IS A UFD BLOCK
368 001232 001013      BNE     LANG      ;NO, CHECK FOR A LANGUAGE
369 001234 010104      MOV      R1,R4    ;SAVE SIZE
370 001236 012702 005007  MOV     #BUFF,R2  ;ADDRESS OF SAVE BUFFER
371 001242 004767 000666  CALL   MOVROM    ;MOVE UFD AREA TO MEMORY
372 001246 001105      BNE     WRLANG    ;BAD CKSUM, QUIT
373
374
375
376 001250 010167 001150  MOV     R1,OLDSIZ ;NOTE - R3 CONTAINS CHECKSUM OF BLOCK AND HEADER
377 001254 010167 001146  MOV     R1,UFDSIZ ;HOWEVER THE CHECKSUM OF HEADER IS ALREADY KNOWN
378 001260 000500      BR      WRLANG    ;TO BE 0 SO R3 IS A VALID CHECK OF UFD BLOCK
379
380 001262 120427 000140  LANG:  CMPB   R4,#LNGHDR ;IS THIS A LANGUAGE HEADER?
381 001266 001075      BNE     WRLANG    ;NO - QUIT
382 001270 010167 001130  MOV     R1,OLDSIZ ;SAVE SIZE FOR NOW
383 001274 062701 000005  ADD     #5,R1     ;ADD SIZE OF (POSSIBLE) UFD HEADER
384 001300 004767 001036  CALL   ROMADR    ;SET UP PCR AND RO
385 001304 005003      CLR     R3        ;INITIALIZE CKSUM
386 001306 004767 001002  CALL   REAROM    ;GET A BYTE
387 001312 004767 000776  CALL   REAROM    ;GET A BYTE
388 001316 004767 000772  CALL   REAROM    ;GET A BYTE
389 001322 010546      MOV     R5,-(SP)  ;SAVE LOW BYTE OF SIZE FOR LATER
390 001324 004767 000764  CALL   REAROM    ;GET A BYTE
391 001330 110566 000001  MOVB   R5,1(SP)  ;SAVE HIGH BYTE OF SIZE AND ID
392 001334 004767 000754  CALL   REAROM    ;GET A BYTE
393 001340 116600 000001  MOVB   1(SP),R0  ;GET I.D.
394 001344 012601      MOV     (SP)+,R1  ;GET SIZE
395 001346 105703      TSTB   R3        ;SEE IF VALID CKSUM
396 001350 001025      BNE     1$       ;NO - WE HAVE LANGUAGE ONLY.
397
398 001352 042700 177437  BIC     #177437,R0 ;GET ID ONLY
399 001356 120027 000040  CMPB   R0,#UFDHDR ;IS THIS A UFD BLOCK?
400 001362 001020      BNE     1$       ;NO, IGNORE IT.
401
402
403
404 001364 042701 160000  BIC     #160000,R1 ;GET RID OF ID
405 001370 062701 000005  ADD     #5,R1     ;SIZE OF HEADER
406 001374 010104      MOV     R1,R4    ;BYTE COUNT TO MOVE
407 001376 010167 001024  MOV     R1,UFDSIZ ;SAVE UFD SIZE
408 001402 066701 001016  ADD     OLDSIZ,R1 ;ADD SIZE OF LANGUAGE AREA
409 001406 012702 005007  MOV     #BUFF,R2  ;MEMORY ADDRESS TO SAVE TO
410 001412 004767 000516  CALL   MOVROM    ;SAVE UFD AREA
411 001416 001404      BEQ     2$       ;YES, IT IS VALID, CONTINUE

```

SAVE OLD LANGUAGE/UFD AREA IN CASE IT MUST BE RESTORED

```

412 001420 005067 001002          CLR      UFDSIZ          ;NO UFD AREA
413 001424 012702 005007          1$: MOV     #BUFF,R2      ;RESET R2
414 001430 016701 000770          2$: MOV     OLDSIZ,R1     ;SIZE OF LANGUAGE AREA
415 001434 010104                MOV     R1,R4            ;BYTES TO MOVE
416 001436 066767 000764 000760  ADD     UFDSIZ,OLDSIZ    ;OLDSIZ IS THE TOTAL SIZE
417 001444 004767 000464          CALL    MOVROM          ;SAVE LANGUAGE AREA
418 001450 001404                BEQ     WRLANG          ;LANGUAGE IS GOOD
419 001452 005067 000746          CLR     OLDSIZ         ;NO LANGUAGE
420 001456 005067 000744          CLR     UFDSIZ         ;NO UFD AREA
421
422          ;GENERATE CHECKSUM FOR FOREIGN LANGUAGE TEST FILE & WRITE TO THE MEMORY IMAGE
423
424 001462 012700 003120  WRLANG: MOV    #TEXT,R0    ;ADDRESS OF BEGINNING OF TEXT
425 001466 005001          CLR     R1             ;INIT CHECKSUM
426 001470 112002          25$: MOVB   (R0)+,R2      ;READ A BYTE
427 001472 160201          SUB     R2,R1          ;ACCUMULATE CHECKSUM
428 001474 020027 005001          CMP     R0,#CKSUM      ;FINISHED ALL TEXT ?
429 001500 001373          BNE    25$            ;NO-CONTINUE
430 001502 110110          MOVB   R1,(R0)         ;WRITE THE CHECKSUM
431
432          .SBTTL  LOAD LOCAL LANGUAGE INTO E2PROM
433
434          ;WRITE UFD & LOCAL LANGUAGE BLOCKS
435
436 001504 016701 000716          MOV     UFDSIZ,R1      ;GET THE LENGTH OF THE UFD
437 001510 062701 001667          ADD     #ROMSZ,R1     ;... & THE TEXT AREA
438 001514 004767 000622          JSR    PC,ROMADR      ;COMPUTE E2PROM PAGE AND ADDR
439 001520 016701 000702          MOV     UFDSIZ,R1     ;SIZE OF UFD AREA TO SAVE
440 001524 001406                BEQ     40$            ;NO UFD AREA - SKIP
441 001526 012702 005007          MOV     #BUFF,R2      ;ADDRESS OF BEGINNING OF UFD AREA
442 001532 112205          35$: MOVB   (R2)+,R5     ;GET SOME DATA
443 001534 004767 000126          CALL    E2WRIT        ;GO WRITE IT
444 001540 077104          SOB    R1,35$         ;FINISHED UFD?
445          ;YES-DO LANGUAGE
446 001542 012702 003120          40$: MOV     #TEXT,R2    ;ADDRESS OF EEPROM LANGUAGE TEXT
447 001546 012701 001667          MOV     #ROMSZ,R1     ;BYTES TO MOVE
448 001552 112205          50$: MOVB   (R2)+,R5     ;GET SOME DATA
449 001554 004767 000106          CALL    E2WRIT        ;WRITE A BYTE
450 001560 077104          SOB    R1,50$         ;ARE WE DONE?
451          ;YES - EXIT
452 001562 112705 000200          MOVB   #BIT7,R5       ;TURN ON LOCAL LANGUAGE BIT IN
453          ;SETUP AREA, THEN EXIT
454
455 001566 105037 177522  EXIT:  CLRB   @#PCRLB    ;SELECT PAGE 0
456 001572 012700 165006          MOV     #E2LLB,R0     ;E2PROM WORD CONTAINING LOCAL LANG. BIT
457 001576 111001          MOVB   (R0),R1
458 001600 142701 177577          BICB   #+CBIT7,R1    ;GET CURRENT LOCAL LANGUAGE BIT
459 001604 120501          CMPB   R5,R1          ;SEE IF BIT ALREADY CORRECT
460 001606 001415          BEQ    EXIT1          ;YES, JUST RETURN
461 001610 112701 000200          MOVB   #BIT7,R1      ;LOCAL LANGUAGE BIT
462 001614 111005          MOVB   (R0),R5       ;GET OLD WORD AGAIN
463 001616 074105          XOR    R1,R5         ;FLIP THE BIT
464 001620 004767 000336          CALL    WRBYTE        ;CHANGE LOCAL LANGUAGE BIT IN E2PROM
465 001624 001006          BNE    EXIT1          ;WOULD NOT WRITE, JUST GIVE UP
466 001626 012700 165316          MOV     #E2PAR,R0     ;ADDRESS OF CKSUM BYTE
467 001632 111005          MOVB   (R0),R5       ;GET OLD CKSUM BYTE
468 001634 074105          XOR    R1,R5         ;CORRECT THE CKSUM

```

## LOAD LOCAL LANGUAGE INTO E2PROM

```

469 001636 004767 000320          CALL   WRBYTE          ;UPDATE E2ROM
470
471 001642          EXIT1:  .FRCTYP #CRLF          ;COMPLETE LINE
                        .NARG  NARGS
                        .NTYPE  NTYPE,#CRLF
                        MOV     #CRLF,R0
                        EMT     44
472 001642 012700 002560          BICB   #60,(SP)        ;BE SURE ROM IS DISABLED
473 001650 142716 000060          MOV    (SP)+,@#BCSR   ;RESTORE BCSR
474 001654 012637 177520          CLR    @#PCR          ;
475 001660 005037 177522          RTS    PC
476
477 001666 004767 000270          E2WRIT: CALL  WRBYTE          ;WRITE THE BYTE TO E2PROM
478 001672 001431          BEQ   3$              ;OK THIS TIME
479 001674 005267 000522          INC   WERR            ;FLAG BAD BYTE
480
481 001700 026727 000516 000004          CMP    WERR,#MAXERR   ;CHECK TO SEE IF PAST THE MAXIMUM ERROR
482 001706 003036          BGT   QUIT            ;LIMIT OF BAD BYTES ALLOWED
483
484 001710 020227 003215          CMP    R2,#M001       ;CHECK TO SEE IF ERROR IS IN MESSAGE
485 001714 101433          BLOS  QUIT            ;BYTE COUNT (MUST BE CORRECT)
486
487 001716 020227 005000          CMP    R2,#MEND1      ;CHECK TO BE SURE DICTIONARY AND UFD
488 001722 101030          BHI   QUIT            ;BLOCKS ARE NOT CORRUPTED
489
490 001724 132705 000140          BITB  #140,R5         ;CHECK TO SEE IF IT SHOULD BE A CONTROL
491 001730 001425          BEQ   QUIT            ;CODE (POSSIBLY DICTIONARY ENTRY)
492
493 001732 132710 000140          BITB  #140,(R0)       ;IF CONTROL CODE (DICTIONARY REFERENCE
494 001736 001422          BEQ   QUIT            ;PERHAPS) CALL IT QUILTS
495
496 001740 111004          MOVB  (R0),R4         ;WE WILL LIVE WITH THIS ERROR, CORRECT
497 001742 116703 003033          MOVB  CKSUM,R3        ;THE CHECKSUM TO ACCOUNT FOR NEW VALUE
498 001746 060503          ADD   R5,R3          ;CANCEL OUT WHAT WAS SUPPOSED TO BE
499 001750 160403          SUB   R4,R3          ;CORRECT FOR ERRONEOUS VALUE
500 001752 110367 003023          MOVB  R3,CKSUM       ;PUT BACK CORRECTED VALUE
501
502 001756 062700 000002          3$:   ADD   #2,R0        ;INCREMENT LOCATION
503 001762 020027 166000          CMP   R0,#ENDE2R     ;FINISHED THIS PAGE ?
504 001766 001005          BNE   10$            ;NO-RETURN
505 001770 012700 165000          MOV   #E2PROM,R0     ;YES-RESET ADDRESS
506 001774 062737 000002 177522          ADD   #2,@#PCR       ;INCREMENT PCR TO NEXT PAGE
507 002002 000207          10$:  RETURN
508
509 002004 005726          QUIT:  TST   (SP)+     ;CORRECT STACK
510 002006 032737 000100 000052  QUIT1: BIT   #BIT6,@#52     ;SEE IF UFD QUIET
511 002014 001403          BEQ   5$              ;NO
512 002016          .FRCTYP #MSG000     ;MESSAGE FOR USER IN HIS OWN LANGUAGE
                        .NARG  NARGS
                        .NTYPE  NTYPE,#MSG000
                        MOV     #MSG000,R0
                        EMT     44
513 002016 012700 003006          002016 012700 003006
514 002022 104044          002022 104044
515
516 002024 016701 000374          5$:   MOV   OLDSIZ,R1
517 002030 100704          BMI   EXIT1          ;ERROR WAS NOT ORION OR CKSUM ERROR, DO NOT
518
519 002032 001427          BEQ   40$            ;TRY TO CLEAR LANGUAGE BIT
520 002034 004767 000302          JSR   PC,ROMADR      ;IF NO OLD LANGUAGE TO RESTORE
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
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600

```

LOAD LOCAL LANGUAGE INTO E2PROM

```

518 002040 012702 005007      MOV      #BUFF,R2      ;STARTING ADDRESS OF OLD LANGUAGE TEXT
519 002044 112205      10$:  MOVB   (R2)+,R5      ;GET A BYTE
520 002046 004767 000110      CALL   WRBYTE        ;WRITE IT OUT
521 002052 001017      BNE    40$           ;IF ERROR, GIVE UP
522 002054 062700 000002      ADD    #2,R0         ;INCREMENT LOCATION
523 002060 020027 166000      CMP    R0,#ENDE2R    ;FINISHED THIS PAGE ?
524 002064 001005      BNE    20$           ;NO-CONTINUE
525 002066 012700 165000      MOV    #E2PROM,R0    ;YES-RESET ADDRESS
526 002072 062737 000002 177522  ADD    #2,@#PCR      ;INCREMENT PCR TO NEXT PAGE
527 002100 077117      20$:  SOB    R1,10$        ;LOOP UNTIL DONE
528 002102 026767 000320 000314  CMP    UFDISZ,OLDSIZ ;IF THE SAME THEN NO LANGUAGE
529 002110 001254      BNE    EXIT1         ;IF LANGUAGE, LEAVE E2PROM LANG. BIT AS IT WAS
530 002112 005005      40$:  CLR    R5            ;TURN OFF LOCAL LANGUAGE BIT IN E2PROM
531 002114 036737 175760 000052  BIT    BIT6,@#52     ;SEE IF UFD QUIET
532 002122 001621      BEQ    EXIT          ;NO
533 002124      .FRCTYP #MSG001
      .NARG  NARGS
      .NTYPE NTYPE,#MSG001
      MOV    #MSG001,R0
      EMT    44
534 002132 000615      BR     EXIT          ;AND CALL IT A DAY
535
536      .SBTTL PROGRAM SUBROUTINES
537
538      ;MOVROM - MOVE BYTES FROM EEPROM TO MEMORY
539      ;ENTRY- R1 = STARTING ADDRESS IN EEPROM (# OF BYTES FROM END)
540      ;       R2 = ADDRESS OF MEMORY BUFFER
541      ;       R4 = # OF BYTES TO MOVE
542      ;EXIT  R1 - UNCHANGED
543      ;       R2 - UPDATED MEMORY ADDRESS
544      ;       R3 = (BYTE) 0 IF VALTD CKSUM
545      ;       "Z" FLAG SET IF CKSUM VALID
546
547 002134 010403      MOVROM: MOV    R4,R3      ;SAVE R4
548 002136 004767 000200      CALL   ROMADR        ;LOAD PCR AND R0 WITH LANGUAGE START AREA
549 002142 010304      MOV    R3,R4         ;RESTORE BYTE COUNT
550 002144 005003      CLR    R3            ;INIT CHECKSUM
551 002146 004767 000142      5$:  CALL   REAROM        ;GET A BYTE
552 002152 110522      MOVB   R5,(R2)+      ;SAVE IT
553 002154 077404      SOB    R4,5$         ;LOOP TILL DONE
554 002156 105703      TSTB   R3            ;IS CHECKSUM GOOD?
555 002160 000207      RETURN
556
557 002162 120510      WRBYTE: CMPB   R5,(R0)  ;IS THE NEW DATA DIFFERENT ?
558 002164 001452      BEQ    10$           ;NO-DO NOT WRITE OVER
559
560 002166 012703 000002      1$:  MOV    #RETRY,R3
561 002172 010510      MOV    R5,(R0)      ;WRITE A LOCATION
562 002174 012704 025370      MOV    #DELAY,R4    ;11 MS WAIT
563 002200 077401      SOB    R4,.          ;WASTE TIME
564 002202 120510      CMPB   R5,(R0)      ;SEE IF IT TOOK
565 002204 001442      BEQ    10$           ;YES, ALL OKAY
566 002206 077307      SOB    R3,1$        ;IF AT FIRST YOU DON'T SUCCEED...
567 002210 113704 177522      MOVB   @#PCRLB,R4   ;PCR PAGE OF BAD BYTE
568 002214 106204      ASRB   R4            ;CONVERT TO PAGE #
569 002216 062704 000060      ADD    #'0,R4       ;CONVERT TO OCTAL
570 002222 110467 000237      MOVB   R4,FMSG1A    ;STORE IT FOR PRINTING

```

PROGRAM SUBROUTINES

```

571 002226 010046      MOV     R0,-(SP)           ;SAVE ROM ADDRESS
572 002230             .ITOA   ,#FMSG1B          ;CONVERT ROM ADDRESS TO OCTAL
                    .NARG   NARGS
                    .NTYPE  NTYPE,#FMSG1B
                    MOV     #FMSG1B,R1
                    EMT     30
002230 012701 002500      .TYPMSG #FMSG1           ;PRINT OUT FIRST PART OF MESSAGE
002234 104030          .NARG   NARGS
                    .NTYPE  NTYPE,#FMSG1
                    MOV     #FMSG1,R0
                    EMT     3
573 002236             BIC     #177400,R5        ;MAKE SURE R5 IS POSITIVE AND A BYTE
                    .ITOA   R5,#DUMMY1        ;CONVERT TO OCTAL
                    .NARG   NARGS
                    .NTYPE  NTYPE,R5
                    MOV     R5,R0
                    .NTYPE  NTYPE,#DUMMY1
                    MOV     #DUMMY1,R1
                    EMT     30
002236 012700 002430      .TYPMSG #FMSG1C          ;PRINT OUT LAST 3 DIGITS OF NUMBER & MESSAGE
002242 104003          .NARG   NARGS
                    .NTYPE  NTYPE,#FMSG1C
                    MOV     #FMSG1C,R0
                    EMT     3
574 002244 042705 177400  MOV     @(SP)+,R0        ;GET BYTE AT ROM ADDRESS
575 002250             BIC     #177400,R0        ;GET RID OF BUS NOISE
                    .ITOA   ,#DUMMY2         ;CONVERT TO OCTAL
                    .NARG   NARGS
                    .NTYPE  NTYPE,#DUMMY2
                    MOV     #DUMMY2,R1
                    EMT     30
002250 010500          .TYPMSG #FMSG1D          ;PRINT LOWER 3 BYTES & REST OF MESSAGE
002252 012701 002526      .NARG   NARGS
                    .NTYPE  NTYPE,#FMSG1D
                    MOV     #FMSG1D,R0
                    EMT     3
002256 104030          CLZ
576 002260             10$: RETURN
                    ;REAROM - READS A BYTE FROM E2PROM ADDRESS (R0)+ INTO R5. AUTOMATICLY ADJUSTS
                    ;PCRLB. UPDATES CKSUM IN R3
                    ; ENTRY - R0 ADDRESS IN ROM TO READ FROM
                    ; R3 PARTIAL CKSUM
                    ; PCRLB CORRECT VALUE FOR BYTE TO READ
                    ; EXIT R0 ADDRESS OF NEXT BYTE
                    ; R3 UPDATED CKSUM
                    ; R5 BYTE READ
                    ; PCRLB CORRECT VALUE FOR NEXT BYTE
002260 012700 002531      REAROM: MOV     (R0)+,R5        ;GET A BYTE & UPDATE ADDR. BY 2
002264 104003          ADD     R5,R3           ;UPDATE CKSUM
577 002266 013600          CMP     R0,#ENDE2R      ;SEE IF WE SHOULD SWITCH PAGES
578 002270 042700 177400  BNE     10$           ;NO
579 002274             MOV     #E2PROM,R0      ;YES - GO TO START OF PAGE
002274 012701 002551      ADD     #2,@#PCR        ;ADVANCE A PAGE
002300 104030          10$: RETURN
002302 012700 002554      ;
002306 104003          ;
581 002310 000244          ;
582 002312 000207          ;
583
584
585
586
587
588
589
590
591
592
593
594 002314 012005          ;
595 002316 060503          ;
596 002320 020027 166000  ;
597 002324 001005          ;
598 002326 012700 165000  ;
599 002332 062737 000002 177522 ;
600 002340 000207          ;
601

```

PROGRAM SUBROUTINES

```

602
603 ;ROMADR - CALCULATE PAGE OFFSET FROM END OF ROM GIVEN SIZE IN BYTES
604 ; ENTRY - R1 SIZE IN BYTES
605 ; EXIT - R0 INITIAL ADDRESS FOR FIRST BYTE IN ROM
606 ; R1 SIZE IN BYTES
607 ; PCRLB CORRECT VALUE FOR FIRST BYTE IN ROM
608
609 002342 010100 ROMADR: MOV R1,R0 ;COPY BYTE COUNT
610 002344 010105 MOV R1,R5 ;SECOND COPY
611 002346 072527 177770 ASH #8.,R5 ;DIVIDE BYTE COUNT BY 256. BYTE PAGES
612 002352 012704 000010 MOV #7+1,R4 ;LAST PAGE IN 2 K PART + 1
613 002356 160504 SUB R5,R4 ;STARTING PAGE NUMBER
614
615 002360 042700 177400 BIC #177400,R0 ;LEAVE ONLY BITS 7:0
616 002364 006300 ASL R0 ;DOUBLE VALUE
617 002366 001003 BNE 20$
618 002370 012700 165000 MOV #E2PROM,r0 ;
619 002374 000406 BR 30$ ;IF 0
620
621 002376 005400 20$: NEG R0 ;MAKE STARTING ADDRESS BITS 8:0
622 002400 042700 177000 BIC #177000,R0 ;
623 002404 052700 165000 BIS #E2PROM,R0 ;MAKE A E2PROM ADDRESS
624 002410 005304 DEC R4 ;DECREMENT PAGE NUMBER BY 1
625
626 002412 006304 30$: ASL R4 ;MAKE PAGE NUMBER CORRECT FOR PCR
627 002414 110437 177522 MOVB R4,#PCRLB ;CORRECT PAGE IN PCRLB
628 002420 000207 RTS PC ;RETURN
629
630 002422 000000 WERR: 0 ;FLAG FOR BAD BYTE
631 002424 177777 OLDSIZ: -1 ;>0 - SIZE IN BYTES OF OLD LANGUAGE, 0 IF NO
632 ;LANGUAGE, -1 IF E2PROM MAY BE BAD/NONEXISTANT
633 002426 000000 UFDSIZ: 0 ;SIZE IN BYTES OF OLD UFD AREA
634
635 .SBTTL "FIELD SERVICE MODE" ERROR MESSAGES
636
637 .ENABL LC
638 002430 105 105 120 FMSG1: .ASCII /EEPROM write error, PCR page /
639 002433 122 117 115
640 002436 040 167 162
641 002441 151 164 145
642 002444 040 145 162
643 002447 162 157 162
644 002452 054 040 120
645 002455 103 122 040
646 002460 160 141 147
647 002463 145 040
648 002465 130 054 040 FMSG1A: .ASCII /X, address /
649 002470 141 144 144
650 002473 162 145 163
651 002476 163 040
652 002500 FMSG1B: .BLKB 6 ;FOR ADDRESS
653 002506 015 012 104 .ASCIIZ <CR><LF>/Data written /
654 002511 141 164 141
655 002514 040 167 162
656 002517 151 164 164
657 002522 145 156 040
658 002525 000

```

"FIELD SERVICE MODE" ERROR MESSAGES

642	002526				DUMMY1: .BLKB 3	;3 UPPER BYTES NOT TO BE PRINTED
643	002531				FMSG1C: .BLKB 3	
644	002534	054	040	104	.ASCIZ /, Data read /	
	002537	141	164	141		
	002542	040	162	145		
	002545	141	144	040		
	002550	000				
645	002551				DUMMY2: .BLKB 3	;3 UPPER BYTES NOT TO BE PRINTED
646	002554				FMSG1D: .BLKB 3	
647	002557	056			.ASCII ./	
648	002560	015	012	000	CRLF: .ASCIZ <CR><LF>	
649	002563	114	141	156	FMSG2: .ASCIZ /Language Area not supported on this processor./<CR><LF>	
	002566	147	165	141		
	002571	147	145	040		
	002574	101	162	145		
	002577	141	040	156		
	002602	157	164	040		
	002605	163	165	160		
	002610	160	157	162		
	002613	164	145	144		
	002616	040	157	156		
	002621	040	164	150		
	002624	151	163	040		
	002627	160	162	157		
	002632	143	145	163		
	002635	163	157	162		
	002640	056	015	012		
	002643	000				
650	002644	103	165	162	FMSG3: .ASCIZ /Current boot ROM version does not support language area./<CR><LF>	
	002647	162	145	156		
	002652	164	040	142		
	002655	157	157	164		
	002660	040	122	117		
	002663	115	040	166		
	002666	145	162	163		
	002671	151	157	156		
	002674	040	144	157		
	002677	145	163	040		
	002702	156	157	164		
	002705	040	163	165		
	002710	160	160	157		
	002713	162	164	040		
	002716	154	141	156		
	002721	147	165	141		
	002724	147	145	040		
	002727	141	162	145		
	002732	141	056	015		
	002735	012	000			
651	002737	103	150	145	FMSG4: .ASCIZ /Checksum error in EEPROM setup area./<CR><LF>	
	002742	143	153	163		
	002745	165	155	040		
	002750	145	162	162		
	002753	157	162	040		
	002756	151	156	040		
	002761	105	105	120		
	002764	122	117	115		
	002767	040	163	145		

## "FIELD SERVICE MODE" ERROR MESSAGES

002772	164	165	160	
002775	040	141	162	
003000	145	141	056	
003003	015	012	000	
652				.SBTTL TRANSLATED LOADER ERROR MESSAGES
653 003006	015	111	155	MSG000: .ASCIZ <CR>!Impossibilita' di caricare Italiano!
003011	160	157	163	
003014	163	151	142	
003017	151	154	151	
003022	164	141	047	
003025	040	144	151	
003030	040	143	141	
003033	162	151	143	
003036	141	162	145	
003041	040	111	164	
003044	141	154	151	
003047	141	156	157	
003052	000			
654 003053	040	055	040	MSG001: .ASCIZ ! - ritornare alla versione inglese.!<CR>
003056	162	151	164	
003061	157	162	156	
003064	141	162	145	
003067	040	141	154	
003072	154	141	040	
003075	166	145	162	
003100	163	151	157	
003103	156	145	040	
003106	151	156	147	
003111	154	145	163	
003114	145	056	015	
003117	000			
655				.SBTTL START OF AREA TO BE LOADED INTO E2PROM
656				
657				.SBTTL Italiano LANGUAGE TEXT
658				
659 003120	075			TEXT: .BYTE M001-TEXT
660 003121	011			.BYTE M002-M001
661 003122	002			.BYTE M003-M002
662 003123	006			.BYTE M004-M003
663 003124	006			.BYTE M005-M004
664 003125	006			.BYTE M006-M005
665 003126	002			.BYTE M007-M006
666 003127	002			.BYTE M010-M007
667 003130	002			.BYTE M011-M010
668 003131	000			.BYTE M012-M011
669 003132	000			.BYTE M013-M012
670 003133	000			.BYTE M014-M013
671 003134	000			.BYTE M015-M014
672 003135	000			.BYTE M016-M015
673 003136	000			.BYTE M017-M016
674 003137	000			.BYTE M020-M017
675 003140	040			.BYTE M021-M020
676 003141	032			.BYTE M022-M021
677 003142	025			.BYTE M023-M022
678 003143	121			.BYTE M024-M023
679 003144	022			.BYTE M025-M024
680 003145	001			.BYTE M026-M025

Italiano LANGUAGE TEXT

681	003146	032				.BYTE	M027-M026	
682	003147	007				.BYTE	M030-M027	
683	003150	013				.BYTE	M031-M030	
684	003151	022				.BYTE	M032-M031	
685	003152	002				.BYTE	M033-M032	
686	003153	052				.BYTE	M034-M033	
687	003154	000				.BYTE	M035-M034	
688	003155	001				.BYTE	M036-M035	
689	003156	000				.BYTE	M037-M036	
690	003157	002				.BYTE	M040-M037	
691	003160	031				.BYTE	M041-M040	
692	003161	000				.BYTE	M042-M041	
693	003162	016				.BYTE	M043-M042	
694	003163	021				.BYTE	M044-M043	
695	003164	027				.BYTE	M045-M044	
696	003165	016				.BYTE	M046-M045	
697	003166	017				.BYTE	M047-M046	
698	003167	040				.BYTE	M050-M047	
699	003170	022				.BYTE	M051-M050	
700	003171	036				.BYTE	M052-M051	
701	003172	026				.BYTE	M053-M052	
702	003173	037				.BYTE	M054-M053	
703	003174	022				.BYTE	M055-M054	
704	003175	024				.BYTE	M056-M055	
705	003176	070				.BYTE	M057-M056	
706	003177	012				.BYTE	M060-M057	
707	003200	000				.BYTE	M061-M060	
708	003201	012				.BYTE	M062-M061	
709	003202	002				.BYTE	M063-M062	
710	003203	013				.BYTE	M064-M063	
711	003204	032				.BYTE	M065-M064	
712	003205	003				.BYTE	M066-M065	
713	003206	025				.BYTE	M067-M066	
714	003207	046				.BYTE	M070-M067	
715	003210	012				.BYTE	M071-M070	
716	003211	003				.BYTE	M072-M071	
717	003212	072				.BYTE	M073-M072	
718	003213	002				.BYTE	M074-M073	
719	003214	037				.BYTE	MEND1-M074	
720	003215	111	164	141	M001:	.ASCIZ	!Italiano!	
	003220	154	151	141				
	003223	156	157	000				
721	003226	077	000		M002:	.ASCIZ	!?!	
722	003230	107	125	111	M003:	.ASCIZ	!GUIDA!	
	003233	104	101	000				
723	003236	101	126	126	M004:	.ASCIZ	!AVVIO!	
	003241	111	117	000				
724	003244	114	111	123	M005:	.ASCIZ	!LISTA!	
	003247	124	101	000				
725	003252	177	000		M006:	.ASCIZ	<177>	:Setup command
726	003254	177	000		M007:	.ASCIZ	<177>	:Map command
727	003256	177	000		M010:	.ASCIZ	<177>	:Test command
728	003260				M011:			
729	003260				M012:			
730	003260				M013:			
731	003260				M014:			
732	003260				M015:			

## Italiano LANGUAGE TEXT

733	003260				M016:	
734	003260				M017:	
735	003260	104	151	163	M020:	.ASCII !Dispositivi!<TAB>!Unita'!<TAB><TAB>!Descrizione!<CR>
	003263	160	157	163		
	003266	151	164	151		
	003271	166	151	011		
	003274	125	156	151		
	003277	164	141	047		
	003302	011	011	104		
	003305	145	163	143		
	003310	162	151	172		
	003313	151	157	156		
	003316	145	015			
736	003320	105	154	145	M021:	.ASCII !Elenco programmi di avvio!<CR>
	003323	156	143	157		
	003326	040	160	162		
	003331	157	147	162		
	003334	141	155	155		
	003337	151	040	144		
	003342	151	040	141		
	003345	166	166	151		
	003350	157	015			
737	003352	101	166	166	M022:	.ASCII !Avvio del sistema da !
	003355	151	157	040		
	003360	144	145	154		
	003363	040	163	151		
	003366	163	164	145		
	003371	155	141	040		
	003374	144	141	040		
738	003377	015	103	157	M023:	.ASCII <CR>!Comando Descrizione!<CR><CR>!AVVIO Carica e avvia il!
	003402	155	141	156		
	003405	144	157	040		
	003410	040	104	145		
	003413	163	143	162		
	003416	151	172	151		
	003421	157	156	145		
	003424	015	015	101		
	003427	126	126	111		
	003432	117	040	040		
	003435	040	040	103		
	003440	141	162	151		
	003443	143	141	040		
	003446	145	040	141		
	003451	166	166	151		
	003454	141	040	151		
	003457	154				
739	003460	040	163	151		.ASCII ! sistema dal dispositivo!<CR>!LISTA!<TAB>! !
	003463	163	164	145		
	003466	155	141	040		
	003471	144	141	154		
	003474	040	144	151		
	003477	163	160	157		
	003502	163	151	164		
	003505	151	166	157		
	003510	015	114	111		
	003513	123	124	101		
	003516	011	040			

## Italiano LANGUAGE TEXT

740	003520	015	111	156	M024:	.ASCII	<CR>!In fase di prova !
	003523	040	146	141			
	003526	163	145	040			
	003531	144	151	040			
	003534	160	162	157			
	003537	166	141	040			
741	003542	057			M025:	.ASCII	'/'
742	003543	120	162	145	M026:	.ASCII	!Premere il tasto RITORNO: !
	003546	155	145	162			
	003551	145	040	151			
	003554	154	040	164			
	003557	141	163	164			
	003562	157	040	122			
	003565	111	124	117			
	003570	122	116	117			
	003573	072	040				
743	003575	105	162	162	M027:	.ASCII	!Errore !
	003600	157	162	145			
	003603	040					
744	003604	040	151	156	M030:	.ASCII	! indirizzo !
	003607	144	151	162			
	003612	151	172	172			
	003615	157	040				
745	003617	124	145	163	M031:	.ASCII	!Test in esecuzione!
	003622	164	040	151			
	003625	156	040	145			
	003630	163	145	143			
	003633	165	172	151			
	003636	157	156	145			
746	003641	060	055		M032:	.ASCII	/0-/
747	003643	015	111	156	M033:	.ASCII	<CR>!Introdurre un comando e premere RITORNO: !
	003646	164	162	157			
	003651	144	165	162			
	003654	162	145	040			
	003657	165	156	040			
	003662	143	157	155			
	003665	141	156	144			
	003670	157	040	145			
	003673	040	160	162			
	003676	145	155	145			
	003701	162	145	040			
	003704	122	111	124			
	003707	117	122	116			
	003712	117	072	040			
748	003715				M034:		
749	003715	011			M035:	.BYTE	TAB
750	003716				M036:		
751	003716	015	040		M037:	.BYTE	CR,SPACE
752	003720	106	141	163	M040:	.ASCII	!Fase di avviamento da ROM!
	003723	145	040	144			
	003726	151	040	141			
	003731	166	166	151			
	003734	141	155	145			
	003737	156	164	157			
	003742	040	144	141			
	003745	040	122	117			
	003750	115					

## Italiano LANGUAGE TEXT

753	003751				M041:	
754	003751	015	115	145	M042:	.ASCII <CR>!Messaggio 06!<CR>
	003754	163	163	141		
	003757	147	147	151		
	003762	157	040	060		
	003765	066	015			
755	003767	125	156	151	M043:	.ASCII !Unita' non pronta!
	003772	164	141	047		
	003775	040	156	157		
	004000	156	040	160		
	004003	162	157	156		
	004006	164	141			
756	004010	123	165	160	M044:	.ASCII !Supporto non caricabile!
	004013	160	157	162		
	004016	164	157	040		
	004021	156	157	156		
	004024	040	143	141		
	004027	162	151	143		
	004032	141	142	151		
	004035	154	145			
757	004037	115	141	156	M045:	.ASCII !Manca il disco!
	004042	143	141	040		
	004045	151	154	040		
	004050	144	151	163		
	004053	143	157			
758	004055	115	141	156	M046:	.ASCII !Manca il nastro!
	004060	143	141	040		
	004063	151	154	040		
	004066	156	141	163		
	004071	164	162	157		
759	004074	125	156	151	M047:	.ASCII !Unita' di controllo inesistente,!
	004077	164	141	047		
	004102	040	144	151		
	004105	040	143	157		
	004110	156	164	162		
	004113	157	154	154		
	004116	157	040	151		
	004121	156	145	163		
	004124	151	163	164		
	004127	145	156	164		
	004132	145	054			
760	004134	125	156	151	M050:	.ASCII !Unita' inesistente!
	004137	164	141	047		
	004142	040	151	156		
	004145	145	163	151		
	004150	163	164	145		
	004153	156	164	145		
761	004156	116	165	155	M051:	.ASCII !Numero dell'unita' non valido !
	004161	145	162	157		
	004164	040	144	145		
	004167	154	154	047		
	004172	165	156	151		
	004175	164	141	047		
	004200	040	156	157		
	004203	156	040	166		
	004206	141	154	151		
	004211	144	157	040		

## Italiano LANGUAGE TEXT

762	004214	104	151	163	M052:	.ASCII	!Dispositivo non valido!
	004217	160	157	163			
	004222	151	164	151			
	004225	166	157	040			
	004230	156	157	156			
	004233	040	166	141			
	004236	154	151	144			
	004241	157					
763	004242	105	162	162	M053:	.ASCII	!Errore dell'unita' di controllo!
	004245	157	162	145			
	004250	040	144	145			
	004253	154	154	047			
	004256	165	156	151			
	004261	164	141	047			
	004264	040	144	151			
	004267	040	143	157			
	004272	156	164	162			
	004275	157	154	154			
	004300	157					
764	004301	105	162	162	M054:	.ASCII	!Errore nell'unita'!
	004304	157	162	145			
	004307	040	156	145			
	004312	154	154	047			
	004315	165	156	151			
	004320	164	141	047			
765	004323	015	015	106	M055:	.ASCII	<CR><CR>!Fase di avviamento!
	004326	141	163	145			
	004331	040	144	151			
	004334	040	141	166			
	004337	166	151	141			
	004342	155	145	156			
	004345	164	157				
766	004347	015	103	157	M056:	.ASCII	<CR>!Consultare la sezione sui guasti nel Manuale operativo!
	004352	156	163	165			
	004355	154	164	141			
	004360	162	145	040			
	004363	154	141	040			
	004366	163	145	172			
	004371	151	157	156			
	004374	145	040	163			
	004377	165	151	040			
	004402	147	165	141			
	004405	163	164	151			
	004410	040	156	145			
	004413	154	040	115			
	004416	141	156	165			
	004421	141	154	145			
	004424	040	157	160			
	004427	145	162	141			
	004432	164	151	166			
	004435	157					
767	004436	015				.ASCII	<CR>
768	004437	033	133	062	M057:	.ASCII	<ESC>/[2J/ ;Erase screen
	004442	112					
769	004443	033	133	065		.ASCII	<ESC>/[5;0H/ ;Set cursor to line 5 and col 1
	004446	073	060	110			
770	004451				M060:		

## Italiano LANGUAGE TEXT

771	004451	115	145	163	M061:	.ASCII	!Messaggio !
	004454	163	141	147			
	004457	147	151	157			
	004462	040					
772	004463	015	015		M062:	.BYTE	CR,CR
773	004465	015	015	113	M063:	.ASCII	<CR><CR>/KDJ11-B >/
	004470	104	112	061			
	004473	061	055	102			
	004476	040	076				
774	004500	015	105	162	M064:	.ASCII	<CR>!Errore di avvio di EPROM!<CR>
	004503	162	157	162			
	004506	145	040	144			
	004511	151	040	141			
	004514	166	166	151			
	004517	157	040	144			
	004522	151	040	105			
	004525	120	122	117			
	004530	115	015				
775	004532	010	040	010	M065:	.BYTE	BACKSP,SPACE,BACKSP
776	004535	015	103	157	M066:	.ASCII	<CR>!Comando non valido.!<CR>
	004540	155	141	156			
	004543	144	157	040			
	004546	156	157	156			
	004551	040	166	141			
	004554	154	151	144			
	004557	157	056	015			
777	004562	015	015	111	M067:	.ASCII	<CR><CR>!I comandi sono Guida, Avvio e Lista.!
	004565	040	143	157			
	004570	155	141	156			
	004573	144	151	040			
	004576	163	157	156			
	004601	157	040	107			
	004604	165	151	144			
	004607	141	054	040			
	004612	101	166	166			
	004615	151	157	040			
	004620	145	040	114			
	004623	151	163	164			
	004626	141	056				
778	004630	111	156	144	M070:	.ASCII	!Indirizzo !
	004633	151	162	151			
	004636	172	172	157			
	004641	040					
779	004642	040	075	040	M071:	.ASCII	/ = /
780	004645	111	156	164	M072:	.ASCII	!Introdurre numero unita' e dispositivo e premere RITORNO: !
	004650	162	157	144			
	004653	165	162	162			
	004656	145	040	156			
	004661	165	155	145			
	004664	162	157	040			
	004667	165	156	151			
	004672	164	141	047			
	004675	040	145	040			
	004700	144	151	163			
	004703	160	157	163			
	004706	151	164	151			
	004711	166	157	040			

Italiano LANGUAGE TEXT

004714	145	040	160	
004717	162	145	155	
004722	145	162	145	
004725	040	122	111	
004730	124	117	122	
004733	116	117	072	
004736	040			
781 004737	011	011		M073: .ASCII <TAB><TAB>
782 004741	015	106	141	M074: .ASCII <CR>!Fase di avviamento automatico!<CR>
004744	163	145	040	
004747	144	151	040	
004752	141	166	166	
004755	151	141	155	
004760	145	156	164	
004763	157	040	141	
004766	165	164	157	
004771	155	141	164	
004774	151	143	157	
004777	015			
783 005000				MEND1:
784				.SBTTL NULL DICTIONARY BLOCK, CHECKSUM AND LANGUAGE HEADER
785 005000				wb:
786 005000	001			ENGWRD: .BYTE ENDBLK-ENGWRD
787 005001				ENDBLK:
788				
789				
790 005001				WEND:
791				
792 005001	000			CKSUM: .byte 0 ;checksum
793				
794				
795 005002				MEND: ;END OF NULL TEXT
796				
797 005002				ME:
798 005002				WE:
799				
800				;FOREIGN LANGUAGE HEADER
801				
802	000002			B1 = WE-WB&377 ;DICTIONARY BYTE COUNT 7:0
803	000000			B2 = WE-WB&17400/256. ;DICTIONARY BYTE COUNT 10:8
804	000262			B3 = MEND-text&377 ;TEXT BYTE COUNT 7:0
805	000143			B4 = MEND-text&017400/256.!140 ;TEXT BYTE COUNT 12:8 & ID=011
806				
807 005002	002			.BYTE B1
808 005003	000			.BYTE B2
809 005004	262			.BYTE B3
810 005005	143			.BYTE B4
811 005006	351			.BYTE -<B1+B2+B3+B4>&377 ;THIS BYTE IS HEADER CHECKSUM
812				
813 005007				FLEND:
814 005007				BUFF: ;TEMPORARY SAVE AREA FOR OLD AREA
815	001000			.END START

## Symbol table

BACKSP=	000010	FLEND	005007	M010	003256	M042	003751	M074	004741
BCSR	= 177520	FMSG1	002430	M011	003260	M043	003767	NARGS	= 000001
BDR	= 177524	FMSG1A	002465	M012	003260	M044	004010	NTYPE	= 000027
BIT6	= 000100	FMSG1B	002500	M013	003260	M045	004037	OLDSIZ	002424
BIT7	= 000200	FMSG1C	002531	M014	003260	M046	004055	PCR	= 177522
BUFF	005007	FMSG1D	002554	M015	003260	M047	004074	PCRLB	= 177522
B1	= 000002	FMSG2	002563	M016	003260	M050	004134	QUIT	002004
B2	= 000000	FMSG3	002644	M017	003260	M051	004156	QUIT1	002006
B3	= 000262	FMSG4	002737	M020	003260	M052	004214	REAROM	002314
B4	= 000143	LANG	001262	M021	003320	M053	004242	RETRY	= 000002
CKSUM	005001	LF	= 000012	M022	003352	M054	004301	RMVTST	= 173002
CR	= 000015	LNGHDR	= 000140	M023	003377	M055	004323	ROMADR	002342
CRLF	002560	MAXERR	= 000004	M024	003520	M056	004347	ROMSZ	= 001667
DELAY	= 025370	ME	005002	M025	003542	M057	004437	SPACE	= 000040
DUMMY1	002526	MEND	005002	M026	003543	M060	004451	START	001000
DUMMY2	002551	MEND1	005000	M027	003575	M061	004451	TAB	= 000011
ENDBLK	005001	MOVROM	002134	M030	003604	M062	004463	TEXT	003120
ENDE2R	= 166000	MSG000	003006	M031	003617	M063	004465	UFDHDR	= 000040
ENGWRD	005000	MSG001	003053	M032	003641	M064	004500	UFDSIZ	002426
ESC	= 000033	M001	003215	M033	003643	M065	004532	WB	005000
EXIT	001566	M002	003226	M034	003715	M066	004535	WE	005002
EXIT1	001642	M003	003230	M035	003715	M067	004562	WEND	005001
E2LLB	= 165006	M004	003236	M036	003716	M070	004630	WERR	002422
E2PAR	= 165316	M005	003244	M037	003716	M071	004642	WRBYTE	002162
E2PROM	= 165000	M006	003252	M040	003720	M072	004645	WRLANG	001462
E2WRIT	001666	M007	003254	M041	003751	M073	004737		

. ABS. 005007 000 (RW,I,GBL,ABS,OVR)  
 000000 001 (RW,I,LCL,REL,CON)

Errors detected: 0

## \*\*\* Assembler statistics

Work file reads: 0  
 Work file writes: 0  
 Size of work file: 8553 Words ( 34 Pages)  
 Size of core pool: 19402 Words ( 74 Pages)  
 Operating system: RSX-11M/PLUS (Under VAX/VMS)

Elapsed time: 00:00:14.44  
 OEEEA0.BIC,COEEEA0/CR/-SP=COEEEA0

SYMBOL CROSS REFERENCE

CREF V02

SYMBOL	VALUE	REFERENCES	CREF	V02						
BACKSP	= 000010	#5-260	6-775	6-775						
BCSR	= 177520	#5-239	6-315	*6-316	*6-473					
BDR	= 177524	#5-254								
BIT6	= 000100	#5-258	6-510	6-531						
BIT7	= 000200	#5-257	6-452	6-458	6-461					
BUFF	005007	6-370	6-409	6-413	6-441	6-518	#6-814			
B1	= 000002	#6-802	6-807	6-811						
B2	= 000000	#6-803	6-808	6-811						
B3	= 000262	#6-804	6-809	6-811						
B4	= 000143	#6-805	6-810	6-811						
CKSUM	005001	6-428	6-497	*6-500	#6-792					
CR	= 000015	#5-255	6-641	6-648	6-649	6-650	6-651	6-653	6-654	6-735
		6-736	6-738	6-738	6-738	6-739	6-740	6-747	6-751	6-754
		6-754	6-765	6-765	6-766	6-767	6-772	6-772	6-773	6-773
		6-774	6-774	6-776	6-776	6-777	6-777	6-782	6-782	
CRLF	002560	6-471	6-471	#6-648						
DELAY	= 025370	#5-247	6-562							
DUMMY1	002526	6-575	6-575	#6-642						
DUMMY2	002551	6-579	6-579	#6-645						
ENDBLK	005001	6-786	#6-787							
ENDE2R	= 166000	#5-245	6-349	6-503	6-523	6-596				
ENGWRD	005000	#6-786	6-786							
ESC	= 000033	#5-262	6-768	6-769						
EXIT	001566	#6-455	6-532	6-534						
EXIT1	001642	6-460	6-465	#6-471	6-514	6-529				
E2LLB	= 165006	#5-244	6-456							
E2PAR	= 165316	#5-243	6-466							
E2PROM	= 165000	#5-242	5-243	5-244	5-245	6-324	6-333	6-335	6-505	6-525
		6-598	6-618	6-623						
E2WRIT	001666	6-443	6-449	#6-477						
FLEND	005007	5-264	#6-813							
FMSG1	002430	6-573	6-573	#6-638						
FMSG1A	002465	*6-570	#6-639							
FMSG1B	002500	6-572	6-572	#6-640						
FMSG1C	002531	6-576	6-576	#6-643						
FMSG1D	002554	6-580	6-580	#6-646						
FMSG2	002563	6-321	6-321	#6-649						
FMSG3	002644	6-344	6-344	#6-650						
FMSG4	002737	6-337	6-337	#6-651						
LANG	001262	6-368	#6-380							
LF	= 000012	#5-256	6-641	6-648	6-649	6-650	6-651			
LNGHDR	= 000140	#5-248	6-380							
MAXERR	= 000004	#5-252	6-481							
ME	005002	#6-797								
MEND	005002	#6-795	6-804	6-805						
MEND1	005000	6-487	6-719	#6-783						
MOVROM	002134	6-371	6-410	6-417	#6-547					
MSG000	003006	6-512	6-512	#6-653						
MSG001	003053	6-533	6-533	#6-654						
M001	003215	6-484	6-659	6-660	#6-720					
M002	003226	6-660	6-661	#6-721						
M003	003230	6-661	6-662	#6-722						

## SYMBOL CROSS REFERENCE

CREF V02

SYMBOL	VALUE	REFERENCES		
M004	003236	6-662	6-663	#6-723
M005	003244	6-663	6-664	#6-724
M006	003252	6-664	6-665	#6-725
M007	003254	6-665	6-666	#6-726
M010	003256	6-666	6-667	#6-727
M011	003260	6-667	6-668	#6-728
M012	003260	6-668	6-669	#6-729
M013	003260	6-669	6-670	#6-730
M014	003260	6-670	6-671	#6-731
M015	003260	6-671	6-672	#6-732
M016	003260	6-672	6-673	#6-733
M017	003260	6-673	6-674	#6-734
M020	003260	6-674	6-675	#6-735
M021	003320	6-675	6-676	#6-736
M022	003352	6-676	6-677	#6-737
M023	003377	6-677	6-678	#6-738
M024	003520	6-678	6-679	#6-740
M025	003542	6-679	6-680	#6-741
M026	003543	6-680	6-681	#6-742
M027	003575	6-681	6-682	#6-743
M030	003604	6-682	6-683	#6-744
M031	003617	6-683	6-684	#6-745
M032	003641	6-684	6-685	#6-746
M033	003643	6-685	6-686	#6-747
M034	003715	6-686	6-687	#6-748
M035	003715	6-687	6-688	#6-749
M036	003716	6-688	6-689	#6-750
M037	003716	6-689	6-690	#6-751
M040	003720	6-690	6-691	#6-752
M041	003751	6-691	6-692	#6-753
M042	003751	6-692	6-693	#6-754
M043	003767	6-693	6-694	#6-755
M044	004010	6-694	6-695	#6-756
M045	004037	6-695	6-696	#6-757
M046	004055	6-696	6-697	#6-758
M047	004074	6-697	6-698	#6-759
M050	004134	6-698	6-699	#6-760
M051	004156	6-699	6-700	#6-761
M052	004214	6-700	6-701	#6-762
M053	004242	6-701	6-702	#6-763
M054	004301	6-702	6-703	#6-764
M055	004323	6-703	6-704	#6-765
M056	004347	6-704	6-705	#6-766
M057	004437	6-705	6-706	#6-768
M060	004451	6-706	6-707	#6-770
M061	004451	6-707	6-708	#6-771
M062	004463	6-708	6-709	#6-772
M063	004465	6-709	6-710	#6-773
M064	004500	6-710	6-711	#6-774
M065	004532	6-711	6-712	#6-775
M066	004535	6-712	6-713	#6-776
M067	004562	6-713	6-714	#6-777



MACRO CROSS REFERENCE

CREF V02

MACRO NAME	REFERENCES
.FRCTY	05-299 6-471 6-512 6-533
.ITOA	05-278 6-572 6-575 6-579
.TYPMS	05-267 6-321 6-337 6-344 6-573 6-576 6-580