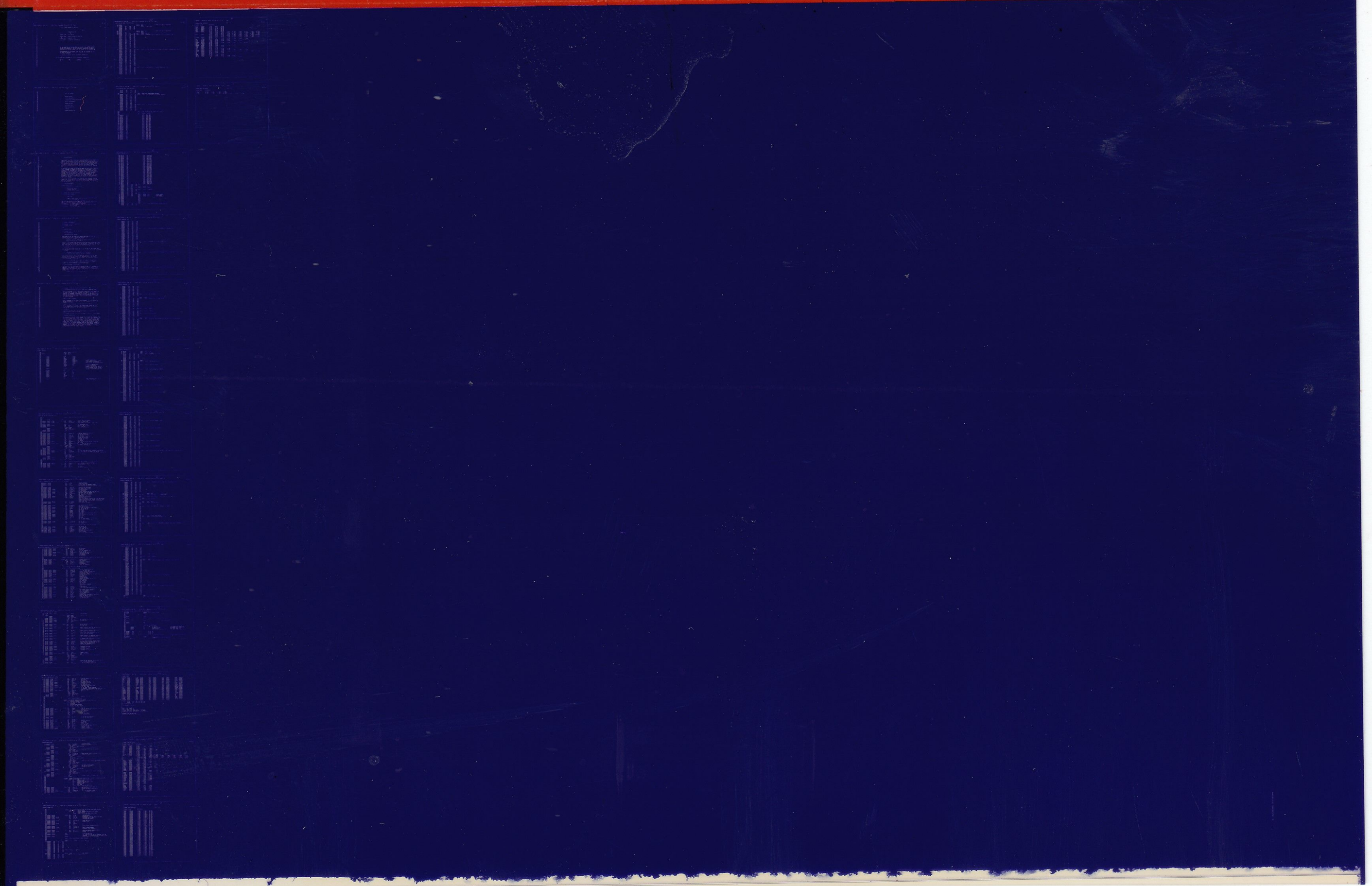


KDJ11-B

EEPROM FR LANG LDR
COEECA0

AH-FF21A-MC
1 OF 1 JUL 1985
COPYRIGHT © 1985

digital
MADE IN USA



9 W
A ::
1

COEECA EEPROM FR LANG LDR

MACRO Y05.02 Wednesday 06-Mar-85 15:24 Page 1

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

.TITLE COEECA EEPROM FR LANG LDR

.REM E

IDENTIFICATION

PRODUCT CODE: AC-FF20A-MC
PRODUCT NAME: COEECAO EEPROM FR 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	

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 OPTIUNS
- 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 memmory 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 message 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.

ε

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 MOV #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$: MOV #E2PROM, R0 ;STARTING ADDRESS TO CHECKSUM
325 001042 0C5001 CLR R1 ;INITIALIZE CHECKSUM
326 001044 012703 000151 MOV #105., R3 ;NO. OF BYTES TO CKSUM
327 001050 012005 201$: 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$: MOV (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 005165  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 R0
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  ;WE HAVE BOTH A LANGUAGE AREA AND A UFD BLOCK.  SAVE THE UFD BLOCK.
405 001370 062701 000005  BIC     #160000,R1 ;GET RID OF ID
406 001374 010104      ADD     #5,R1      ;SIZE OF HEADER
407 001376 010167 001024  MOV     R1,R4      ;BYTE COUNT TO MOVE
408 001402 066701 001016  MOV     R1,UFDSIZ  ;SAVE UFD SIZE
409 001406 012702 005165  ADD     OLDSIZ,R1  ;ADD SIZE OF LANGUAGE AREA
410 001412 004767 000516  MOV     #BUFF,R2   ;MEMORY ADDRESS TO SAVE TO
411 001416 001404      CALL   MOVROM      ;SAVE UFD AREA
                     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 005165      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 003076  WRLANG:  MOV      #TEXT,R0      ;ADDRESS OF BEGINNING OF TEXT
425 001466 005001              CLR      R1      ;INIT CHECKSUM
426 001470 112002      25$:     MOVB     (R0)+,R2      ;READ A BYTE
427 001472 160201              SUB      R2,R1      ;ACCUMULATE CHECKSUM
428 001474 020027 005157      CMP      R0,#CKSUM      ;FINISHED ALL TEXT ?
429 001500 001373              BNE     25$         ;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 002067      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 005165      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 003076      40$:     MOV      #TEXT,R2      ;ADDRESS OF EEPROM LANGUAGE TEXT
447 001546 012701 002067      MOV      #ROMSZ,R1      ;BYTES TO MOVE
448 001552 112205      50$:     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 001650 142716 000060          BICB  #60,(SP)          ;BE SURE ROM IS DISABLED
473 001654 012637 177520          MOV   (SP)+,@#BCSR     ;RESTORE BCSR
474 001660 005037 177522          CLR   @#PCR            ;
475 001664 000207          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 003173          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 005156          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 QUIT
495
496 001740 111004          MOVB  (R0),R4         ;WE WILL LIVE WITH THIS ERROR, CORRECT
497 001742 116703 003211          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 003201          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 002024 016701 000374          5$:  MOV   OLDSIZ,R1
514 002030 100704          BMI   EXIT1          ;ERROR WAS NOT ORION OR CKSUM ERROR, DO NOT
515
516 002032 001427          BEQ   40$            ;TRY TO CLEAR LANGUAGE BIT
517 002034 004767 000302          JSR   PC,ROMADR      ;IF NO OLD LANGUAGE TO RESTORE
                        ;COMPUTE STARTING ADDRESS OF OLD LANG IN E2PROM

```

LOAD LOCAL LANGUAGE INTO E2PROM

```

518 002040 012702 005165      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    UFDSIZ,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$:  CLR    R5         ;TURN OFF LOCAL LANGUAGE BIT IN E2PROM
531 002114 036737 175760 000052  BIT    BIT6,@#52   ;SEE IF UFD QUIET
532 002122 001621      BEQ    EXIT        ;NO
533 002124      .FRCTYP #MSG001
      .NARG  NARGS
      .NTYPE NTYPE,#MSG001
      MOV    #MSG001,R0
      EMT    44
534 002132 000615      BR    EXIT        ;AND CALL IT A DAY
535
536      .SBTTL  PROGRAM SUBROUTINES
537
538      ;MOVROM - MOVE BYTES FROM EEPROM TO MEMORY
539      ;ENTRY- R1 = STARTING ADDRESS IN EEPROM (# OF BYTES FROM END)
540      ;       R2 = ADDRESS OF MEMORY BUFFER
541      ;       R4 = # OF BYTES TO MOVE
542      ;EXIT  R1 - UNCHANGED
543      ;       R2 - UPDATED MEMORY ADDRESS
544      ;       R3 = (BYTE) 0 IF VALID CKSUM
545      ;       "Z" FLAG SET IF CKSUM VALID
546
547 002134 010403      MOVROM: MOV    R4,R3      ;SAVE R4
548 002136 004767 000200      CALL   ROMADR     ;LOAD PCR AND R0 WITH LANGUAGE START AREA
549 002142 010304      MOV    R3,R4     ;RESTORE BYTE COUNT
550 002144 005003      CLR    R3        ;INIT CHECKSUM
551 002146 004767 000142      5$:  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$:  MOV    #RETRY,R3  ;WRITE A LOCATION
561 002172 010510      MOV    R5,(R0)   ;11 MS WAIT
562 002174 012704 025370      MOV    #DELAY,R4 ;WASTE TIME
563 002200 077401      SOB    R4, .     ;WASTE TIME
564 002202 120510      CMPB  R5,(R0)   ;SEE IF IT TOOK
565 002204 001442      BEQ    10$       ;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
                    000002      .NARG      NARGS
                    000027      .NTYPE     NTYPE,#FMSG1B
                    002230 012701 002500      MOV      #FMSG1B,R1
                    002234 104030      EMT      30
573 002236              .TYPMSG   #FMSG1            ;PRINT OUT FIRST PART OF MESSAGE
                    000001      .NARG      NARGS
                    000027      .NTYPE     NTYPE,#FMSG1
                    002236 012700 002430      MOV      #FMSG1,R0
                    002242 104003      EMT      3
574 002244 042705 177400      BIC      #177400,R5        ;MAKE SURE R5 IS POSITIVE AND A BYTE
575 002250              .ITOA      R5,#DUMMY1        ;CONVERT TO OCTAL
                    000002      .NARG      NARGS
                    000005      .NTYPE     NTYPE,R5
                    002250 010500      MOV      R5,R0
                    000027      .NTYPE     NTYPE,#DUMMY1
                    002252 012701 002526      MOV      #DUMMY1,R1
                    002256 104030      EMT      30
576 002260              .TYPMSG   #FMSG1C          ;PRINT OUT LAST 3 DIGITS OF NUMBER & MESSAGE
                    000001      .NARG      NARGS
                    000027      .NTYPE     NTYPE,#FMSG1C
                    002260 012700 002531      MOV      #FMSG1C,R0
                    002264 104003      EMT      3
577 002266 013600      MOV      @((SP)+,R0        ;GET BYTE AT ROM ADDRESS
578 002270 042700 177400      BIC      #177400,R0        ;GET RID OF BUS NOISE
579 002274              .ITOA      ,#DUMMY2          ;CONVERT TO OCTAL
                    000002      .NARG      NARGS
                    000027      .NTYPE     NTYPE,#DUMMY2
                    002274 012701 002551      MOV      #DUMMY2,R1
                    002300 104030      EMT      30
580 002302              .TYPMSG   #FMSG1D          ;PRINT LOWER 3 BYTES & REST OF MESSAGE
                    000001      .NARG      NARGS
                    000027      .NTYPE     NTYPE,#FMSG1D
                    002302 012700 002554      MOV      #FMSG1D,R0
                    002306 104003      EMT      3
581 002310 000244      CLZ                          ;COULDN'T DO IT, SET ERROR FLAG
582 002312 000207      10$: RETURN
583
584 ;REAROM - READS A BYTE FROM E2PROM ADDRESS (R0)+ INTO R5. AUTOMATICLY ADJUSTS
585 ;PCRLB. UPDATES CKSUM IN R3
586 ; ENTRY - R0 ADDRESS IN ROM TO READ FROM
587 ; R3 PARTIAL CKSUM
588 ; PCRLB CORRECT VALUE FOR BYTE TO READ
589 ; EXIT R0 ADDRESS OF NEXT BYTE
590 ; R3 UPDATED CKSUM
591 ; R5 BYTE READ
592 ; PCRLB CORRECT VALUE FOR NEXT BYTE
593
594 002314 012005      REAROM: MOV      (R0)+,R5        ;GET A BYTE & UPDATE ADDR. BY 2
595 002316 060503      ADD      R5,R3          ;UPDATE CKSUM
596 002320 020027 166000      CMP      R0,#ENDE2R      ;SEE IF WE SHOULD SWITCH PAGES
597 002324 001005      BNE     10$            ;NO
598 002326 012700 165000      MOV      #E2PROM,R0      ;YES - GO TO START OF PAGE
599 002332 062737 000002 177522      ADD      #2,@#PCR        ;ADVANCE A PAGE
600 002340 000207      10$: RETURN
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	115 MSG000: .ASCIZ <CR>!IMPOSSIBLE DE CHARGER FRANCAIS!
	003011	120	117	123
	003014	123	111	102
	003017	114	105	040
	003022	104	105	040
	003025	103	110	101
	003030	122	107	105
	003033	122	040	106
	003036	122	101	116
	003041	103	101	111
	003044	123	000	
654	003046	040	055	040 MSG001: .ASCIZ ! - RETOUR A L'ANGLAIS.!<CR>
	003051	122	105	124
	003054	117	125	122
	003057	040	101	040
	003062	114	047	101
	003065	116	107	114
	003070	101	111	123
	003073	056	015	000
655				.SBTTL START OF AREA TO BE LOADED INTO E2PROM
656				
657				.SBTTL FRANCAIS LANGUAGE TEXT
658				
659	003076	075		TEXT: .BYTE M001-TEXT
660	003077	011		.BYTE M002-M001
661	003100	002		.BYTE M003-M002
662	003101	005		.BYTE M004-M003
663	003102	013		.BYTE M005-M004
664	003103	006		.BYTE M006-M005
665	003104	002		.BYTE M007-M006
666	003105	002		.BYTE M010-M007
667	003106	002		.BYTE M011-M010
668	003107	000		.BYTE M012-M011
669	003110	000		.BYTE M013-M012
670	003111	000		.BYTE M014-M013
671	003112	000		.BYTE M015-M014
672	003113	000		.BYTE M016-M015
673	003114	000		.BYTE M017-M016
674	003115	000		.BYTE M020-M017
675	003116	041		.BYTE M021-M020
676	003117	054		.BYTE M022-M021
677	003120	042		.BYTE M023-M022
678	003121	133		.BYTE M024-M023
679	003122	012		.BYTE M025-M024
680	003123	001		.BYTE M026-M025
681	003124	037		.BYTE M027-M026
682	003125	007		.BYTE M030-M027
683	003126	011		.BYTE M031-M030
684	003127	015		.BYTE M032-M031
685	003130	002		.BYTE M033-M032
686	003131	071		.BYTE M034-M033
687	003132	000		.BYTE M035-M034

FRANCAIS LANGUAGE TEXT

688	003133	001				.BYTE	M036-M035	
689	003134	000				.BYTE	M037-M036	
690	003135	002				.BYTE	M040-M037	
691	003136	024				.BYTE	M041-M040	
692	003137	000				.BYTE	M042-M041	
693	003140	014				.BYTE	M043-M042	
694	003141	017				.BYTE	M044-M043	
695	003142	034				.BYTE	M045-M044	
696	003143	027				.BYTE	M046-M045	
697	003144	026				.BYTE	M047-M046	
698	003145	033				.BYTE	M050-M047	
699	003146	025				.BYTE	M051-M050	
700	003147	027				.BYTE	M052-M051	
701	003150	025				.BYTE	M053-M052	
702	003151	025				.BYTE	M054-M053	
703	003152	021				.BYTE	M055-M054	
704	003153	015				.BYTE	M056-M055	
705	003154	140				.BYTE	M057-M056	
706	003155	012				.BYTE	M060-M057	
707	003156	000				.BYTE	M061-M060	
708	003157	010				.BYTE	M062-M061	
709	003160	002				.BYTE	M063-M062	
710	003161	013				.BYTE	M064-M063	
711	003162	041				.BYTE	M065-M064	
712	003163	003				.BYTE	M066-M065	
713	003164	024				.BYTE	M067-M066	
714	003165	075				.BYTE	M070-M067	
715	003166	010				.BYTE	M071-M070	
716	003167	003				.BYTE	M072-M071	
717	003170	123				.BYTE	M073-M072	
718	003171	003				.BYTE	M074-M073	
719	003172	041				.BYTE	MEND1-M074	
720	003173	106	122	101	M001:	.ASCIZ	!FRANCAIS!	
	003176	116	103	101				
	003201	111	123	000				
721	003204	077	000		M002:	.ASCIZ	!?!	
722	003206	101	111	104	M003:	.ASCIZ	!AIDE!	
	003211	105	000					
723	003213	103	110	101	M004:	.ASCIZ	!CHARGEMENT!	
	003216	122	107	105				
	003221	115	105	116				
	003224	124	000					
724	003226	114	111	123	M005:	.ASCIZ	!LISTE!	
	003231	124	105	000				
725	003234	177	000		M006:	.ASCIZ	<177>	:Setup command
726	003236	177	000		M007:	.ASCIZ	<177>	:Map command
727	003240	177	000		M010:	.ASCIZ	<177>	:Test command
728	003242				M011:			
729	003242				M012:			
730	003242				M013:			
731	003242				M014:			
732	003242				M015:			
733	003242				M016:			
734	003242				M017:			
735	003242	125	116	111	M020:	.ASCII	!UNITES PERIPHERIQUES DESCRIPTION!<CR>	
	003245	124	105	123				
	003250	040	120	105				

FRANCAIS LANGUAGE TEXT

	003253	122	111	120	
	003256	110	105	122	
	003261	111	121	125	
	003264	105	123	040	
	003267	104	105	123	
	003272	103	122	111	
	003275	120	124	111	
	003300	117	116	015	
736	003303	124	101	120	M021: .ASCII !TAPEZ LES NOMS DES PROGRAMMES DE CHARGEMENT!<CR>
	003306	105	132	040	
	003311	114	105	123	
	003314	040	116	117	
	003317	115	123	040	
	003322	104	105	123	
	003325	040	120	122	
	003330	117	107	122	
	003333	101	115	115	
	003336	105	123	040	
	003341	104	105	040	
	003344	103	110	101	
	003347	122	107	105	
	003352	115	105	116	
	003355	124	015		
737	003357	103	110	101	M022: .ASCII !CHARGEMENT DU SYSTEME A PARTIR DE !
	003362	122	107	105	
	003365	115	105	116	
	003370	124	040	104	
	003373	125	040	123	
	003376	131	123	124	
	003401	105	115	105	
	003404	040	101	040	
	003407	120	101	122	
	003412	124	111	122	
	003415	040	104	105	
	003420	040			
738	003421	015	104	105	M023: .ASCII <CR>!DESCRIPTION DE LA COMMANDE!<CR><CR>!CHARGEMENT CHARGEZ E!
	003424	123	103	122	
	003427	111	120	124	
	003432	111	117	116	
	003435	040	104	105	
	003440	040	114	101	
	003443	040	103	117	
	003446	115	115	101	
	003451	116	104	105	
	003454	015	015	103	
	003457	110	101	122	
	003462	107	105	115	
	003465	105	116	124	
	003470	040	103	110	
	003473	101	122	107	
	003476	105	132	040	
	003501	105			
739	003502	124	040	114	.ASCII !T LANCEZ LE SYSTEME A PARTIR DE !<CR>!LISTE!<TAB>! !
	003505	101	116	103	
	003510	105	132	040	
	003513	114	105	040	
	003516	123	131	123	

FRANCAIS LANGUAGE TEXT

	003521	124	105	115	
	003524	105	040	101	
	003527	040	120	101	
	003532	122	124	111	
	003535	122	040	104	
	003540	105	040	015	
	003543	114	111	123	
	003546	124	105	011	
	003551	040	040	040	
740	003554	015	101	124	M024: .ASCII <CR>!ATTENDEZ !
	003557	124	105	116	
	003562	104	105	132	
	003565	040			
741	003566	057			M025: .ASCII '/'
742	003567	101	120	120	M026: .ASCII !APPUYEZ SUR LA TOUCHE RETOUR : !
	003572	125	131	105	
	003575	132	040	123	
	003600	125	122	040	
	003603	114	101	040	
	003606	124	117	125	
	003611	103	110	105	
	003614	040	122	105	
	003617	124	117	125	
	003622	122	040	072	
	003625	040			
743	003626	105	122	122	M027: .ASCII !ERREUR !
	003631	105	125	122	
	003634	040			
744	003635	040	101	104	M030: .ASCII ! ADRESSE !
	003640	122	105	123	
	003643	123	105	040	
745	003646	124	105	123	M031: .ASCII !TEST EN COURS!
	003651	124	040	105	
	003654	116	040	103	
	003657	117	125	122	
	003662	123			
746	003663	060	055		M032: .ASCII /0-/
747	003665	015	105	116	M033: .ASCII <CR>!ENTREZ UNE COMMANDE PUIS APPUYEZ SUR LA TOUCHE RETOUR : !
	003670	124	122	105	
	003673	132	040	125	
	003676	116	105	040	
	003701	103	117	115	
	003704	115	101	116	
	003707	104	105	040	
	003712	120	125	111	
	003715	123	040	101	
	003720	120	120	125	
	003723	131	105	132	
	003726	040	123	125	
	003731	122	040	114	
	003734	101	040	124	
	003737	117	125	103	
	003742	110	105	040	
	003745	122	105	124	
	003750	117	125	122	
	003753	040	072	040	
748	003756				M034:

FRANCAIS LANGUAGE TEXT

749	003756	011			M035:	.BYTE	TAB
750	003757				M036:		
751	003757	015	040		M037:	.BYTE	CR,SPACE
752	003761	114	101	116	M040:	.ASCII	!LANCEMENT DE LA ROM !
	003764	103	105	115			
	003767	105	116	124			
	003772	040	104	105			
	003775	040	114	101			
	004000	040	122	117			
	004003	115	040				
753	004005				M041:		
754	004005	015	115	105	M042:	.ASCII	<CR>!MESSAGE 06!<CR>
	004010	123	123	101			
	004013	107	105	040			
	004016	060	066	015			
755	004021	125	116	111	M043:	.ASCII	!UNITE PAS PRETE!
	004024	124	105	040			
	004027	120	101	123			
	004032	040	120	122			
	004035	105	124	105			
756	004040	120	105	122	M044:	.ASCII	!PERIPHERIQUE NON CHARGEABLE !
	004043	111	120	110			
	004046	105	122	111			
	004051	121	125	105			
	004054	040	116	117			
	004057	116	040	103			
	004062	110	101	122			
	004065	107	105	101			
	004070	102	114	105			
	004073	040					
757	004074	111	114	040	M045:	.ASCII	!IL N'Y A PAS DE DISQUE !
	004077	116	047	131			
	004102	040	101	040			
	004105	120	101	123			
	004110	040	104	105			
	004113	040	104	111			
	004116	123	121	125			
	004121	105	040				
758	004123	111	114	040	M046:	.ASCII	!IL N'Y A PAS DE BANDE !
	004126	116	047	131			
	004131	040	101	040			
	004134	120	101	123			
	004137	040	104	105			
	004142	040	102	101			
	004145	116	104	105			
	004150	040					
759	004151	111	114	040	M047:	.ASCII	!IL N'Y A PAS DE CONTROLEUR,!
	004154	116	047	131			
	004157	040	101	040			
	004162	120	101	123			
	004165	040	104	105			
	004170	040	103	117			
	004173	116	124	122			
	004176	117	114	105			
	004201	125	122	054			
760	004204	111	114	040	M050:	.ASCII	!IL N'Y A PAS D'UNITE !
	004207	116	047	131			

FRANCAIS LANGUAGE TEXT

	004212	040	101	040	
	004215	120	101	123	
	004220	040	104	047	
	004223	125	116	111	
	004226	124	105	040	
761	004231	115	101	125	M051: .ASCII !MAUVAIS NUMERO D'UNITE !
	004234	126	101	111	
	004237	123	040	116	
	004242	125	115	105	
	004245	122	117	040	
	004250	104	047	125	
	004253	116	111	124	
	004256	105	040		
762	004260	115	101	125	M052: .ASCII !MAUVAIS PERIPHERIQUE !
	004263	126	101	111	
	004266	123	040	120	
	004271	105	122	111	
	004274	120	110	105	
	004277	122	111	121	
	004302	125	105	040	
763	004305	105	122	122	M053: .ASCII !ERREUR DU CONTROLEUR !
	004310	105	125	122	
	004313	040	104	125	
	004316	040	103	117	
	004321	116	124	122	
	004324	117	114	105	
	004327	125	122	040	
764	004332	105	122	122	M054: .ASCII !ERREUR DE L'UNITE!
	004335	105	125	122	
	004340	040	104	105	
	004343	040	114	047	
	004346	125	116	111	
	004351	124	105		
765	004353	015	015	103	M055: .ASCII <CR><CR>!CHARGEMENT !
	004356	110	101	122	
	004361	107	105	115	
	004364	105	116	124	
	004367	040			
766	004370	015	123	111	M056: .ASCII <CR>!SI VOUS AVEZ BESOIN D'AIDE, CONSULTEZ LE CHAPITRE SUR LE!
	004373	040	126	117	
	004376	125	123	040	
	004401	101	126	105	
	004404	132	040	102	
	004407	105	123	117	
	004412	111	116	040	
	004415	104	047	101	
	004420	111	104	105	
	004423	054	040	103	
	004426	117	116	123	
	004431	125	114	124	
	004434	105	132	040	
	004437	114	105	040	
	004442	103	110	101	
	004445	120	111	124	
	004450	122	105	040	
	004453	123	125	122	
	004456	040	114	105	

FRANCAIS LANGUAGE TEXT

767	004461	040	104	105		.ASCII	! DEPANNAGE!<CR>!DU GUIDE DE L'UTILISATEUR.!<CR>
	004464	120	101	116			
	004467	116	101	107			
	004472	105	015	104			
	004475	125	040	107			
	004500	125	111	104			
	004503	105	040	104			
	004506	105	040	114			
	004511	047	125	124			
	004514	111	114	111			
	004517	123	101	124			
	004522	105	125	122			
	004525	056	015				
768	004527	015				.ASCII	<CR>
769	004530	033	133	062	M057:	.ASCII	<ESC>/[2J/ ;Erase screen
	004533	112					
770	004534	033	133	065		.ASCII	<ESC>/[5;0H/ ;Set cursor to line 5 and col 1
	004537	073	060	110			
771	004542				M060:		
772	004542	115	105	123	M061:	.ASCII	!MESSAGE !
	004545	123	101	107			
	004550	105	040				
773	004552	015	015		M062:	.BYTE	CR,CR
774	004554	015	015	113	M063:	.ASCII	<CR><CR>/KDJ11-B >/
	004557	104	112	061			
	004562	061	055	102			
	004565	040	076				
775	004567	015	105	122	M064:	.ASCII	<CR>!ERREUR DE CHARGEMENT EN EEPROM !<CR>
	004572	122	105	125			
	004575	122	040	104			
	004600	105	040	103			
	004603	110	101	122			
	004606	107	105	115			
	004611	105	116	124			
	004614	040	105	116			
	004617	040	105	105			
	004622	120	122	117			
	004625	115	040	015			
776	004630	010	040	010	M065:	.BYTE	BACKSP,SPACE,BACKSP
777	004633	015	115	101	M066:	.ASCII	<CR>!MAUVAISE COMMANDE !<CR>
	004636	125	126	101			
	004641	111	123	105			
	004644	040	103	117			
	004647	115	115	101			
	004652	116	104	105			
	004655	040	015				
778	004657	015	015	114	M067:	.ASCII	<CR><CR>!LES COMMANDES DISPONIBLES SONT AIDE, CHARGEMENT, ET !
	004662	105	123	040			
	004665	103	117	115			
	004670	115	101	116			
	004673	104	105	123			
	004676	040	104	111			
	004701	123	120	117			
	004704	116	111	102			
	004707	114	105	123			
	004712	040	123	117			
	004715	116	124	040			

FRANCAIS LANGUAGE TEXT

	004720	101	111	104	
	004723	105	054	040	
	004726	103	110	101	
	004731	122	107	105	
	004734	115	105	116	
	004737	124	054	040	
	004742	105	124	040	
779	004745	114	111	123	.ASCII !LISTE. !
	004750	124	105	056	
	004753	040			
780	004754	101	104	122	M070: .ASCII !ADRESSE !
	004757	105	123	123	
	004762	105	040		
781	004764	040	075	040	M071: .ASCII / = /
782	004767	124	101	120	M072: .ASCII !TAPEZ LES NUMEROS DU PERIPHERIQUE ET DE L'UNITE ET !
	004772	105	132	040	
	004775	114	105	123	
	005000	040	116	125	
	005003	115	105	122	
	005006	117	123	040	
	005011	104	125	040	
	005014	120	105	122	
	005017	111	120	110	
	005022	105	122	111	
	005025	121	125	105	
	005030	040	105	124	
	005033	040	104	105	
	005036	040	114	047	
	005041	125	116	111	
	005044	124	105	040	
	005047	105	124	040	
783	005052	015	101	120	.ASCII <CR>!APPUYEZ SUR LA TOUCHE RETOUR : !
	005055	120	125	131	
	005060	105	132	040	
	005063	123	125	122	
	005066	040	114	101	
	005071	040	124	117	
	005074	125	103	110	
	005077	105	040	122	
	005102	105	124	117	
	005105	125	122	040	
	005110	072	040		
784	005112	011	040	040	M073: .ASCII <TAB>! !
785	005115	015	104	105	M074: .ASCII <CR>!DEBUT DU CHARGEMENT AUTOMATIQUE!<CR>
	005120	102	125	124	
	005123	040	104	125	
	005126	040	103	110	
	005131	101	122	107	
	005134	105	115	105	
	005137	116	124	040	
	005142	101	125	124	
	005145	117	115	101	
	005150	124	111	121	
	005153	125	105	015	
786	005156				MEND1:
787					.SBTTL NULL DICTIONARY BLOCK, CHECKSUM AND LANGUAGE HEADER
788	005156				wb:

NULL DICTIONARY BLOCK, CHECKSUM AND LANGUAGE HEADER

```

789 005156      001            ENGRD: .BYTE    ENDBLK-ENGRD
790 005157                    ENDBLK:
791
792
793 005157                    WEND:
794
795 005157      000            CKSUM: .byte    0            ;checksum
796
797
798 005160                    MEND:                            ;END OF NULL TEXT
799
800 005160                    ME:
801 005160                    WE:
802
803                    ;FOREIGN LANGUAGE HEADER
804
805                    000002            B1            =            WE-WB&377                    ;DICTIONARY BYTE COUNT 7:0
806                    000000            B2            =            WE-WB&17400/256.            ;DICTIONARY BYTE COUNT 10:8
807                    000062            B3            =            MEND-text&377                ;TEXT BYTE COUNT 7:0
808                    000144            B4            =            MEND-text&017400/256.!140    ;TEXT BYTE COUNT 12:8 & ID=011
809
810 005160      002                    .BYTE    B1
811 005161      000                    .BYTE    B2
812 005162      062                    .BYTE    B3
813 005163      144                    .BYTE    B4
814 005164      150                    .BYTE    -<B1+B2+B3+B4>&377            ;THIS BYTE IS HEADER CHECKSUM
815
816 005165                    FLEND:
817 005165                    BUFF:                    ;TEMPORARY SAVE AREA FOR OLD AREA
818                    001000                    .END            START

```


Symbol table

BACKSP= 000010	FLEND 005165	M010 003240	M042 004005	M074 005115
BCSR = 177520	FMSG1 002430	M011 003242	M043 004021	NARGS = 000001
BDR = 177524	FMSG1A 002465	M012 003242	M044 004040	NTYPE = 000027
BIT6 = 000100	FMSG1B 002500	M013 003242	M045 004074	OLDSIZ 002424
BIT7 = 000200	FMSG1C 002531	M014 003242	M046 004123	PCR = 177522
BUFF 005165	FMSG1D 002554	M015 003242	M047 004151	PCRLB = 177522
B1 = 000002	FMSG2 002563	M016 003242	M050 004204	QUIT 002004
B2 = 000000	FMSG3 002644	M017 003242	M051 004231	QUIT1 002006
B3 = 000062	FMSG4 002737	M020 003242	M052 004260	REAROM 002314
B4 = 000144	LANG 001262	M021 003303	M053 004305	RETRY = 000002
CKSUM 005157	LF = 000012	M022 003357	M054 004332	RMVTST= 173002
CR = 000015	LNGHDR= 000140	M023 003421	M055 004353	ROMADR 002342
CRLF 002560	MAXERR= 000004	M024 003554	M056 004370	ROMSZ = 002067
DELAY = 025370	ME 005160	M025 003566	M057 004530	SPACE = 000040
DUMMY1 002526	MEND 005160	M026 003567	M060 004542	START 001000
DUMMY2 002551	MEND1 005156	M027 003626	M061 004542	TAB = 000011
ENDBLK 005157	MOVROM 002134	M030 003635	M062 004552	TEXT 003076
ENDE2R= 166000	MSG000 003006	M031 003646	M063 004554	UFDHDR= 000040
ENGWRD 005156	MSG001 003046	M032 003663	M064 004567	UFDSIZ 002426
ESC = 000033	M001 003173	M033 003665	M065 004630	WB 005156
EXIT 001566	M002 003204	M034 003756	M066 004633	WE 005160
EXIT1 001642	M003 003206	M035 003756	M067 004657	WEND 005157
E2LLB = 165006	M004 003213	M036 003757	M070 004754	WERR 002422
E2PAR = 165316	M005 003226	M037 003757	M071 004764	WRBYTE 002162
E2PROM= 165000	M006 003234	M040 003761	M072 004767	WRLANG 001462
E2WRIT 001666	M007 003236	M041 004005	M073 005112	

. ABS. 005165 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: 00:00:36.06
 OEECA0.BIC,COEECA0/CR/-SP=COEECA0

SYMBOL CROSS REFERENCE

CREF V02

SYMBOL	VALUE	REFERENCES
BACKSP	= 000010	#5-260 6-776 6-776
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	005165	6-370 6-409 6-413 6-441 6-518 #6-817
B1	= 000002	#6-805 6-810 6-814
B2	= 000000	#6-806 6-811 6-814
B3	= 000062	#6-807 6-812 6-814
B4	= 000144	#6-808 6-813 6-814
CKSUM	005157	6-428 6-497 *6-500 #6-795
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 6-738 6-739 6-740 6-747 6-751 6-754
		6-754 6-765 6-765 6-765 6-766 6-767 6-767 6-768 6-773 6-774
		6-774 6-774 6-775 6-775 6-775 6-777 6-777 6-778 6-778 6-783
		6-785 6-785
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	005157	6-789 #6-790
ENDE2R	= 166000	#5-245 6-349 6-503 6-523 6-596
ENGWRD	005156	#6-789 6-789
ESC	= 000033	#5-262 6-769 6-770
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	005165	5-264 #6-816
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	005160	#6-800
MEND	005160	#6-798 6-807 6-808
MEND1	005156	6-487 6-719 #6-786
MOVROM	002134	6-371 6-410 6-417 #6-547
MSG000	003006	6-512 6-512 #6-653
MSG001	003046	6-533 6-533 #6-654
M001	003173	6-484 6-659 6-660 #6-720
M002	003204	6-660 6-661 #6-721

SYMBOL CROSS REFERENCE

CREF V02

SYMBOL	VALUE	REFERENCES
M003	003206	6-661 6-662 #6-722
M004	003213	6-662 6-663 #6-723
M005	003226	6-663 6-664 #6-724
M006	003234	6-664 6-665 #6-725
M007	003236	6-665 6-666 #6-726
M010	003240	6-666 6-667 #6-727
M011	003242	6-667 6-668 #6-728
M012	003242	6-668 6-669 #6-729
M013	003242	6-669 6-670 #6-730
M014	003242	6-670 6-671 #6-731
M015	003242	6-671 6-672 #6-732
M016	003242	6-672 6-673 #6-733
M017	003242	6-673 6-674 #6-734
M020	003242	6-674 6-675 #6-735
M021	003303	6-675 6-676 #6-736
M022	003357	6-676 6-677 #6-737
M023	003421	6-677 6-678 #6-738
M024	003554	6-678 6-679 #6-740
M025	003566	6-679 6-680 #6-741
M026	003567	6-680 6-681 #6-742
M027	003626	6-681 6-682 #6-743
M030	003635	6-682 6-683 #6-744
M031	003646	6-683 6-684 #6-745
M032	003663	6-684 6-685 #6-746
M033	003665	6-685 6-686 #6-747
M034	003756	6-686 6-687 #6-748
M035	003756	6-687 6-688 #6-749
M036	003757	6-688 6-689 #6-750
M037	003757	6-689 6-690 #6-751
M040	003761	6-690 6-691 #6-752
M041	004005	6-691 6-692 #6-753
M042	004005	6-692 6-693 #6-754
M043	004021	6-693 6-694 #6-755
M044	004040	6-694 6-695 #6-756
M045	004074	6-695 6-696 #6-757
M046	004123	6-696 6-697 #6-758
M047	004151	6-697 6-698 #6-759
M050	004204	6-698 6-699 #6-760
M051	004231	6-699 6-700 #6-761
M052	004260	6-700 6-701 #6-762
M053	004305	6-701 6-702 #6-763
M054	004332	6-702 6-703 #6-764
M055	004353	6-703 6-704 #6-765
M056	004370	6-704 6-705 #6-766
M057	004530	6-705 6-706 #6-769
M060	004542	6-706 6-707 #6-771
M061	004542	6-707 6-708 #6-772
M062	004552	6-708 6-709 #6-773
M063	004554	6-709 6-710 #6-774
M064	004567	6-710 6-711 #6-775
M065	004630	6-711 6-712 #6-776
M066	004633	6-712 6-713 #6-777

SYMBOL CROSS REFERENCE

CREF V02

SYMBOL	VALUE	REFERENCES									
M067	004657	6-713	6-714	#6-778							
M070	004754	6-714	6-715	#6-780							
M071	004764	6-715	6-716	#6-781							
M072	004767	6-716	6-717	#6-782							
M073	005112	6-717	6-718	#6-784							
M074	005115	6-718	6-719	#6-785							
NARGS	= 000001	#6-321	6-321	#6-337	6-337	#6-344	6-344	#6-471	6-471	#6-512	
		6-512	#6-533	6-533	#6-572	6-572	6-572	#6-573	6-573	#6-575	
		6-575	6-575	#6-576	6-576	#6-579	6-579	6-579	6-579	#6-580	6-580
NTYPE	= 000027	#6-321	6-321	#6-337	6-337	#6-344	6-344	#6-471	6-471	#6-512	
		6-512	#6-533	6-533	#6-572	6-572	#6-573	6-573	6-573	#6-575	6-575
		#6-575	6-575	#6-576	6-576	#6-579	6-579	#6-580	6-580	6-580	
OLDSIZ	002424	*6-339	*6-376	*6-382	6-408	6-414	*6-416	*6-419	6-513	6-528	
		#6-631									
PCR	= 177522	#5-240	*6-314	*6-340	*6-474	*6-506	*6-526	*6-599			
PCRLB	= 177522	#5-241	*6-455	6-567	*6-627						
QUIT	002004	6-482	6-485	6-488	6-491	6-494	#6-509				
QUIT1	002006	6-345	#6-510								
REAROM	002314	6-386	6-387	6-388	6-390	6-392	6-551	#6-594			
RETRY	= 000002	#5-250	6-560								
RMVTST	= 173002	#5-246	6-341								
ROMADR	002342	6-384	6-438	6-517	6-548	#6-609					
ROMSZ	= 002067	#5-264	6-437	6-447							
SPACE	= 000040	#5-261	6-751	6-776							
START	001000	#6-314	6-818								
TAB	= 000011	#5-259	6-739	6-749	6-784						
TEXT	003076	5-264	6-424	6-446	#6-659	6-659	6-807	6-808			
UFDHDR	= 000040	#5-249	6-367	6-399							
UFDSIZ	002426	*6-377	*6-407	*6-412	6-416	*6-420	6-436	6-439	6-528	#6-633	
WB	005156	#6-788	6-805	6-806							
WE	005160	#6-801	6-805	6-806							
WEND	005157	#6-793									
WERR	002422	*6-479	6-481	#6-630							
WRBYTE	002162	6-464	6-469	6-477	6-520	#6-557					
WRLANG	001462	6-358	6-372	6-378	6-381	6-418	#6-424				

MACRO CROSS REFERENCE CREF V02

MACRO NAME	REFERENCES
.FRCTY	#5-299 6-471 6-512 6-533
.ITOA	#5-278 6-572 6-575 6-579
.TYPMS	#5-267 6-321 6-337 6-344 6-573 6-576 6-580