

KDJ11-B

EEPROM GER LANG LDR
COEEDA0

AH-FF23A-MC
1 OF 1 JUL 1985
COPYRIGHT © 1985

digital
MADE IN USA

A : :
1

COEEDA EEPROM GER LANG LDR

MACRO Y05.02 Saturday 16-Feb-85 13:55 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 COEEDA EEPROM GER LANG LDR

.REM E

IDENTIFICATION

PRODUCT CODE: AC-FF22A-MC
PRODUCT NAME: COEEDA0 EEPROM GER 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 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 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 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$: CLR OLDSIZ ;SET FLAG THAT ROM EXISTS, CURRENTLY NO LANGUAGE
340 001120 012737 000016 177522 MOV #7*2,@#PCR ;SEL. LAST PAGE OF 2K E2PROM, PGO OF ROM
341 001126 023727 173002 CMP @#RMVST,(PC)+ ;SEE IF ROM VER. 7 OR LATER (CAN SUPPORT LANGUAGE AREA)
342 001132 000250 CLN
343 001134 001405 BEQ 2$ ;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 005121  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  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 005121  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/UPD AREA IN CASE IT MUST BE RESTORED

```

412 001420 005067 001002          CLR    UFDSIZ          ;NO UFD AREA
413 001424 012702 005121          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 003135  WRLANG: MOV    #TEXT,R0 ;ADDRESS OF BEGINNING OF TEXT
425 001466 005001          CLR    R1            ;INIT CHECKSUM
426 001470 112002          25$:  MOVB   (R0)+,R2    ;READ A BYTE
427 001472 160201          SUB    R2,R1         ;ACCUMULATE CHECKSUM
428 001474 020027 005113          CMP    R0,#CKSUM    ;FINISHED ALL TEXT ?
429 001500 001373          BNE   25$           ;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 001764          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 005121          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 003135          40$:  MOV    #TEXT,R2   ;ADDRESS OF EEPROM LANGUAGE TEXT
447 001546 012701 001764          MOV    #ROMSZ,R1    ;BYTES TO MOVE
448 001552 112205          50$:  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 003232          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 005112          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 003145          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 003135          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 005121      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$:  MOV    #RETRY,R3     ;WRITE A LOCATION
561 002172 010510      MOV    R5,(R0)      ;11 MS WAIT
562 002174 012704 025370      MOV    #DELAY,R4    ;WASTE TIME
563 002200 077401      SOB    R4,.          ;SEE IF IT TOOK
564 002202 120510      CMPB   R5,(R0)      ;YES, ALL OKAY
565 002204 001442      BEQ    10$           ;IF AT FIRST YOU DON'T SUCCEED...
566 002206 077307      SOB    R3,1$        ;PCR PAGE OF BAD BYTE
567 002210 113704 177522      MOVB   @#PCRLB,R4   ;CONVERT TO PAGE #
568 002214 106204      ASRB   R4            ;CONVERT TO OCTAL
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
                    .TYPMSG  #FMSG1      ;PRINT OUT FIRST PART OF MESSAGE
                    .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
                    .TYPMSG  #FMSG1C    ;PRINT OUT LAST 3 DIGITS OF NUMBER & MESSAGE
                    .NARG     NARGS
                    .NTYPE    NTYPE,#FMSG1C
                    MOV      #FMSG1C,R0
                    EMT      3
                    MOV      @(SP)+,R0  ;GET BYTE AT ROM ADDRESS
                    BIC      #177400,R0  ;GET RID OF BUS NOISE
                    .ITOA     ,#DUMMY2   ;CONVERT TO OCTAL
                    .NARG     NARGS
                    .NTYPE    NTYPE,#DUMMY2
                    MOV      #DUMMY2,R1
                    EMT      30
                    .TYPMSG  #FMSG1D    ;PRINT LOWER 3 BYTES & REST OF MESSAGE
                    .NARG     NARGS
                    .NTYPE    NTYPE,#FMSG1D
                    MOV      #FMSG1D,R0
                    EMT      3
                    CLZ
                    10$: RETURN      ;COULDN'T DO IT, SET ERROR FLAG
574 002244 042705 177400
575 002250      000002
                    000005
                    002250 010500
                    000027
                    002252 012701 002526
                    002256 104030
576 002260      000001
                    000027
                    002260 012700 002531
                    002264 104003
577 002266 013600
578 002270 042700 177400
579 002274      000002
                    000027
                    002274 012701 002551
                    002300 104030
580 002302      000001
                    000027
                    002302 012700 002554
                    002306 104003
581 002310 000244
582 002312 000207
583
584 ;REAROM - READS A BYTE FROM E2PROM ADDRESS (R0)+ INTO R5. AUTOMATICLY ADJUSTS
585 ;PCRLB. UPDATES CKSUM IN R3
586 ;
587 ; ENTRY - R0 ADDRESS IN ROM TO READ FROM
588 ; R3 PARTIAL CKSUM
589 ; PCRLB CORRECT VALUE FOR BYTE TO READ
590 ; EXIT R0 ADDRESS OF NEXT BYTE
591 ; R3 UPDATED CKSUM
592 ; R5 BYTE READ
593 ; PCRLB CORRECT VALUE FOR NEXT BYTE
594 002314 012005 REAROM: MOV      (R0)+,R5    ;GET A BYTE & UPDATE ADDR. BY 2
595 002316 060503 ADD      R5,R3      ;UPDATE CKSUM
596 002320 020027 166000 CMP      R0,#ENDE2R ;SEE IF WE SHOULD SWITCH PAGES
597 002324 001005 BNE     10$        ;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 /
002433 122 117 115
002436 040 167 162
002441 151 164 145
002444 040 145 162
002447 162 157 162
002452 054 040 120
002455 103 122 040
002460 160 141 147
002463 145 040
639 002465 130 054 040 FMSG1A: .ASCII /X, address /
002470 141 144 144
002473 162 145 163
002476 163 040
640 002500 FMSG1B: .BLKB 6 ;FOR ADDRESS
641 002506 015 012 104 .ASCIIZ <CR><LF>/Data written /
002511 141 164 141
002514 040 167 162
002517 151 164 164
002522 145 156 040
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				
653	003006	015	104	145
	003011	165	164	163
	003014	143	150	145
	003017	040	115	145
	003022	154	144	165
	003025	156	147	145
	003030	156	040	153
	003033	157	145	156
	003036	156	145	156
	003041	040	156	151
	003044	143	150	164
	003047	040	147	145
	003052	154	141	144
	003055	145	156	040
	003060	167	145	162
	003063	144	145	156
	003066	056	000	
654	003070	040	055	040
	003073	132	165	162
	003076	165	145	143
	003101	153	040	172
	003104	165	040	145
	003107	156	147	154
	003112	151	163	143
	003115	150	145	156
	003120	040	115	145
	003123	154	144	165
	003126	156	147	145
	003131	156	056	015
	003134	000		

.SBTTL TRANSLATED LOADER ERROR MESSAGES

MSG000: .ASCIZ <CR>!Deutsche Meldungen koennen nicht geladen werden.!

MSG001: .ASCIZ ! - Zurueck zu englischen Meldungen.!<CR>

.SBTTL START OF AREA TO BE LOADED INTO E2PROM

.SBTTL Deutsch LANGUAGE TEXT

```

TEXT:  .BYTE  M001-TEXT
        .BYTE  M002-M001
        .BYTE  M003-M002
        .BYTE  M004-M003
        .BYTE  M005-M004
        .BYTE  M006-M005
        .BYTE  M007-M006
        .BYTE  M010-M007
        .BYTE  M011-M010
        .BYTE  M012-M011
        .BYTE  M013-M012
        .BYTE  M014-M013
        .BYTE  M015-M014
        .BYTE  M016-M015
        .BYTE  M017-M016
        .BYTE  M020-M017
        .BYTE  M021-M020
        .BYTE  M022-M021

```

655				
656				
657				
658				
659	003135	075		
660	003136	010		
661	003137	002		
662	003140	006		
663	003141	010		
664	003142	007		
665	003143	002		
666	003144	002		
667	003145	002		
668	003146	000		
669	003147	000		
670	003150	000		
671	003151	000		
672	003152	000		
673	003153	000		
674	003154	000		
675	003155	050		
676	003156	030		

Deutsch LANGUAGE TEXT

677	003157	020				.BYTE	M023-M022	
678	003160	121				.BYTE	M024-M023	
679	003161	021				.BYTE	M025-M024	
680	003162	001				.BYTE	M026-M025	
681	003163	046				.BYTE	M027-M026	
682	003164	007				.BYTE	M030-M027	
683	003165	011				.BYTE	M031-M030	
684	003166	013				.BYTE	M032-M031	
685	003167	002				.BYTE	M033-M032	
686	003170	063				.BYTE	M034-M033	
687	003171	000				.BYTE	M035-M034	
688	003172	001				.BYTE	M036-M035	
689	003173	000				.BYTE	M037-M036	
690	003174	002				.BYTE	M040-M037	
691	003175	033				.BYTE	M041-M040	
692	003176	000				.BYTE	M042-M041	
693	003177	014				.BYTE	M043-M042	
694	003200	035				.BYTE	M044-M043	
695	003201	046				.BYTE	M045-M044	
696	003202	037				.BYTE	M046-M045	
697	003203	023				.BYTE	M047-M046	
698	003204	033				.BYTE	M050-M047	
699	003205	030				.BYTE	M051-M050	
700	003206	031				.BYTE	M052-M051	
701	003207	022				.BYTE	M053-M052	
702	003210	025				.BYTE	M054-M053	
703	003211	016				.BYTE	M055-M054	
704	003212	022				.BYTE	M056-M055	
705	003213	074				.BYTE	M057-M056	
706	003214	012				.BYTE	M060-M057	
707	003215	000				.BYTE	M061-M060	
708	003216	010				.BYTE	M062-M061	
709	003217	002				.BYTE	M063-M062	
710	003220	013				.BYTE	M064-M063	
711	003221	041				.BYTE	M065-M064	
712	003222	003				.BYTE	M066-M065	
713	003223	026				.BYTE	M067-M066	
714	003224	061				.BYTE	M070-M067	
715	003225	010				.BYTE	M071-M070	
716	003226	003				.BYTE	M072-M071	
717	003227	076				.BYTE	M073-M072	
718	003230	002				.BYTE	M074-M073	
719	003231	037				.BYTE	MEND1-M074	
720	003232	104	145	165	M001:	.ASCIZ	!Deutsch!	
	003235	164	163	143				
	003240	150	000					
721	003242	077	000		M002:	.ASCIZ	!?!	
722	003244	110	111	114	M003:	.ASCIZ	!HILFE!	
	003247	106	105	000				
723	003252	125	122	114	M004:	.ASCIZ	!URLADEN!	
	003255	101	104	105				
	003260	116	000					
724	003262	114	111	123	M005:	.ASCIZ	!LISTEN!	
	003265	124	105	116				
	003270	000						
725	003271	177	000		M006:	.ASCIZ	<177>	;Setup command
726	003273	177	000		M007:	.ASCIZ	<177>	;Map command

Deutsch LANGUAGE TEXT

727	003275	177	000		M010:	.ASCIZ <177>		;Test command
728	003277				M011:			
729	003277				M012:			
730	003277				M013:			
731	003277				M014:			
732	003277				M015:			
733	003277				M016:			
734	003277				M017:			
735	003277	107	145	162	M020:	.ASCII !Geraetenname Geraetenummern Beschreibung!<CR>		
	003302	141	145	164				
	003305	145	156	141				
	003310	155	145	040				
	003313	107	145	162				
	003316	141	145	164				
	003321	145	156	165				
	003324	155	155	145				
	003327	162	156	040				
	003332	102	145	163				
	003335	143	150	162				
	003340	145	151	142				
	003343	165	156	147				
	003346	015						
736	003347	114	141	144	M021:	.ASCII !Ladeprogramme auflisten!<CR>		
	003352	145	160	162				
	003355	157	147	162				
	003360	141	155	155				
	003363	145	040	141				
	003366	165	146	154				
	003371	151	163	164				
	003374	145	156	015				
737	003377	123	171	163	M022:	.ASCII !Systemstart von !		
	003402	164	145	155				
	003405	163	164	141				
	003410	162	164	040				
	003413	166	157	156				
	003416	040						
738	003417	015	113	157	M023:	.ASCII <CR>!Kommando Beschreibung!<CR><CR>!URLADEN System von Ger!		
	003422	155	155	141				
	003425	156	144	157				
	003430	040	040	102				
	003433	145	163	143				
	003436	150	162	145				
	003441	151	142	165				
	003444	156	147	015				
	003447	015	125	122				
	003452	114	101	104				
	003455	105	116	040				
	003460	040	040	123				
	003463	171	163	164				
	003466	145	155	040				
	003471	166	157	156				
	003474	040	107	145				
	003477	162						
739	003500	141	145	164		.ASCII !aet laden und starten!<CR>!LISTEN !		
	003503	040	154	141				
	003506	144	145	156				
	003511	040	165	156				

Deutsch LANGUAGE TEXT

	003514	144	040	163	
	003517	164	141	162	
	003522	164	145	156	
	003525	015	114	111	
	003530	123	124	105	
	003533	116	040	040	
	003536	040	040		
740	003540	015	114	141	M024: .ASCII <CR>!Ladeversuch mit !
	003543	144	145	166	
	003546	145	162	163	
	003551	165	143	150	
	003554	040	155	151	
	003557	164	040		
741	003561	057			M025: .ASCII '/'
742	003562	104	162	165	M026: .ASCII !Druecken Sie die Wagenruecklauf-taste: !
	003565	145	143	153	
	003570	145	156	040	
	003573	123	151	145	
	003576	040	144	151	
	003601	145	040	127	
	003604	141	147	145	
	003607	156	162	165	
	003612	145	143	153	
	003615	154	141	165	
	003620	146	164	141	
	003623	163	164	145	
	003626	072	040		
743	003630	106	145	150	M027: .ASCII !Fehler !
	003633	154	145	162	
	003636	040			
744	003637	040	101	144	M030: .ASCII ! Adresse !
	003642	162	145	163	
	003645	163	145	040	
745	003650	124	145	163	M031: .ASCII !Test laeuft!
	003653	164	040	154	
	003656	141	145	165	
	003661	146	164		
746	003663	060	055		M032: .ASCII /0-/
747	003665	015	107	145	M033: .ASCII <CR>!Geben Sie ein Kommando ein und druecken Sie <WR>: !
	003670	142	145	156	
	003673	040	123	151	
	003676	145	040	145	
	003701	151	156	040	
	003704	113	157	155	
	003707	155	141	156	
	003712	144	157	040	
	003715	145	151	156	
	003720	040	165	156	
	003723	144	040	144	
	003726	162	165	145	
	003731	143	153	145	
	003734	156	040	123	
	003737	151	145	040	
	003742	074	127	122	
	003745	076	072	040	
748	003750				M034:
749	003750	011			M035: .BYTE TAB

Deutsch LANGUAGE TEXT

750	003751				M036:	
751	003751	015	040		M037:	.BYTE CR,SPACE
752	003753	122	117	115	M040:	.ASCII !ROM-Urladung wird gestartet!
	003756	055	125	162		
	003761	154	141	144		
	003764	165	156	147		
	003767	040	167	151		
	003772	162	144	040		
	003775	147	145	163		
	004000	164	141	162		
	004003	164	145	164		
753	004006				M041:	
754	004006	015	115	145	M042:	.ASCII <CR>!Meldung 06!<CR>
	004011	154	144	165		
	004014	156	147	040		
	004017	060	066	015		
755	004022	114	141	165	M043:	.ASCII !Laufwerk nicht betriebsbereit!
	004025	146	167	145		
	004030	162	153	040		
	004033	156	151	143		
	004036	150	164	040		
	004041	142	145	164		
	004044	162	151	145		
	004047	142	163	142		
	004052	145	162	145		
	004055	151	164			
756	004057	104	141	164	M044:	.ASCII !Datentraeger kann nicht geladen werden!
	004062	145	156	164		
	004065	162	141	145		
	004070	147	145	162		
	004073	040	153	141		
	004076	156	156	040		
	004101	156	151	143		
	004104	150	164	040		
	004107	147	145	154		
	004112	141	144	145		
	004115	156	040	167		
	004120	145	162	144		
	004123	145	156			
757	004125	113	145	151	M045:	.ASCII !Keine Platte/Diskette eingelegt!
	004130	156	145	040		
	004133	120	154	141		
	004136	164	164	145		
	004141	057	104	151		
	004144	163	153	145		
	004147	164	164	145		
	004152	040	145	151		
	004155	156	147	145		
	004160	154	145	147		
	004163	164				
758	004164	113	145	151	M046:	.ASCII !Kein Band eingelegt!
	004167	156	040	102		
	004172	141	156	144		
	004175	040	145	151		
	004200	156	147	145		
	004203	154	145	147		
	004206	164				

Deutsch LANGUAGE TEXT

759	004207	123	164	145	M047:	.ASCII	!Steuermodul nicht vorhanden!
	004212	165	145	162			
	004215	155	157	144			
	004220	165	154	040			
	004223	156	151	143			
	004226	150	164	040			
	004231	166	157	162			
	004234	150	141	156			
	004237	144	145	156			
760	004242	114	141	165	M050:	.ASCII	!Laufwerk nicht vorhanden!
	004245	146	167	145			
	004250	162	153	040			
	004253	156	151	143			
	004256	150	164	040			
	004261	166	157	162			
	004264	150	141	156			
	004267	144	145	156			
761	004272	125	156	147	M051:	.ASCII	!Ungueltige Geraetenummer !
	004275	165	145	154			
	004300	164	151	147			
	004303	145	040	107			
	004306	145	162	141			
	004311	145	164	145			
	004314	156	165	155			
	004317	155	145	162			
	004322	040					
762	004323	125	156	147	M052:	.ASCII	!Ungueltiges Geraet!
	004326	165	145	154			
	004331	164	151	147			
	004334	145	163	040			
	004337	107	145	162			
	004342	141	145	164			
763	004345	106	145	150	M053:	.ASCII	!Fehler im Steuermodul!
	004350	154	145	162			
	004353	040	151	155			
	004356	040	123	164			
	004361	145	165	145			
	004364	162	155	157			
	004367	144	165	154			
764	004372	114	141	165	M054:	.ASCII	!Laufwerkfehler!
	004375	146	167	145			
	004400	162	153	146			
	004403	145	150	154			
	004406	145	162				
765	004410	015	015	125	M055:	.ASCII	<CR><CR>!Umladung laeuft !
	004413	162	154	141			
	004416	144	165	156			
	004421	147	040	154			
	004424	141	145	165			
	004427	146	164	040			
766	004432	015	123	151	M056:	.ASCII	<CR>!Siehe Systemhandbuch, Abschnitt Fehlersuche und -behebung!
	004435	145	150	145			
	004440	040	123	171			
	004443	163	164	145			
	004446	155	150	141			
	004451	156	144	142			
	004454	165	143	150			

Deutsch LANGUAGE TEXT

	004457	054	040	101		
	004462	142	163	143		
	004465	150	156	151		
	004470	164	164	040		
	004473	106	145	150		
	004476	154	145	162		
	004501	163	165	143		
	004504	150	145	040		
	004507	165	156	144		
	004512	040	055	142		
	004515	145	150	145		
	004520	142	165	156		
767	004523	147	015	015		
768	004526	033	133	062	M057:	.ASCII !g!<CR><CR> .ASCII <ESC>/[2J/ ;Erase screen
	004531	112				
769	004532	033	133	065		.ASCII <ESC>/[5;0H/ ;Set cursor to line 5 and col 1
	004535	073	060	110		
770	004540				M060:	
771	004540	115	145	154	M061:	.ASCII !Meldung !
	004543	144	165	156		
	004546	147	040			
772	004550	015	015		M062:	.BYTE CR,CR
773	004552	015	015	113	M063:	.ASCII <CR><CR>/KDJ11-B >/
	004555	104	112	061		
	004560	061	055	102		
	004563	040	076			
774	004565	015	106	145	M064:	.ASCII <CR>!Fehler im EEPROM-Urladeprogramm!<CR>
	004570	150	154	145		
	004573	162	040	151		
	004576	155	040	105		
	004601	105	120	122		
	004604	117	115	055		
	004607	125	162	154		
	004612	141	144	145		
	004615	160	162	157		
	004620	147	162	141		
	004623	155	155	015		
775	004626	010	040	010	M065:	.BYTE BACKSP,SPACE,BACKSP
776	004631	015	125	156	M066:	.ASCII <CR>!Ungueltiges Kommando!<CR>
	004634	147	165	145		
	004637	154	164	151		
	004642	147	145	163		
	004645	040	113	157		
	004650	155	155	141		
	004653	156	144	157		
	004656	015				
777	004657	015	015	115	M067:	.ASCII <CR><CR>!Moegliche Kommandos: HILFE, URLADEN und LISTEN.!
	004662	157	145	147		
	004665	154	151	143		
	004670	150	145	040		
	004673	113	157	155		
	004676	155	141	156		
	004701	144	157	163		
	004704	072	040	110		
	004707	111	114	106		
	004712	105	054	040		
	004715	125	122	114		

Deutsch LANGUAGE TEXT

	004720	101	104	105	
	004723	116	040	165	
	004726	156	144	040	
	004731	114	111	123	
	004734	124	105	116	
	004737	056			
778	004740	101	144	162	M070: .ASCII !Adresse !
	004743	145	163	163	
	004746	145	040		
779	004750	040	075	040	M071: .ASCII / = /
780	004753	107	145	142	M072: .ASCII !Geben Sie Geraetenname und -nummer ein und druecken Sie <WR>!
	004756	145	156	040	
	004761	123	151	145	
	004764	040	107	145	
	004767	162	141	145	
	004772	164	145	156	
	004775	141	155	145	
	005000	040	165	156	
	005003	144	040	055	
	005006	156	165	155	
	005011	155	155	145	
	005014	162	040	145	
	005017	151	156	040	
	005022	165	156	144	
	005025	040	144	162	
	005030	165	145	143	
	005033	153	145	156	
	005036	040	123	151	
	005041	145	040	074	
	005044	127	122	076	
781	005047	072	040		.ASCII !: !
782	005051	011	011		M073: .ASCII <TAB><TAB>
783	005053	015	101	165	M074: .ASCII <CR>!Automatische Urladung beginnt!<CR>
	005056	164	157	155	
	005061	141	164	151	
	005064	163	143	150	
	005067	145	040	125	
	005072	162	154	141	
	005075	144	165	156	
	005100	147	040	142	
	005103	145	147	151	
	005106	156	156	164	
	005111	015			
784	005112				MEND1:
785					.SBTTL NULL DICTIONARY BLOCK, CHECKSUM AND LANGUAGE HEADER
786	005112				wb:
787	005112	001			ENGWRD: .BYTE ENDBLK-ENGWRD
788	005113				ENDBLK:
789					
790					
791	005113				WEND:
792					
793	005113	000			CKSUM: .byte 0 ;checksum
794					
795					
796	005114				MEND:
797					;END OF NULL TEXT

NULL DICTIONARY BLOCK, CHECKSUM AND LANGUAGE HEADER

```

798 005114                    ME:
799 005114                    WE:
800
801                    ;FOREIGN LANGUAGE HEADER
802
803                    000002      B1        =      WE-WB&377                    ;DICTIONARY BYTE COUNT 7:0
804                    000000      B2        =      WE-WB&17400/256.               ;DICTIONARY BYTE COUNT 10:8
805                    000357      B3        =      MEND-text&377                   ;TEXT BYTE COUNT 7:0
806                    000143      B4        =      MEND-text&017400/256.!140       ;TEXT BYTE COUNT 12:8 & ID=011
807
808 005114                    .BYTE    B1
809 005115                    .BYTE    B2
810 005116                    .BYTE    B3
811 005117                    .BYTE    B4
812 005120                    .BYTE    -<B1+B2+B3+B4>&377               ;THIS BYTE IS HEADER CHECKSUM
813
814 005121                    FLEND:
815 005121                    BUFF:       ;TEMPORARY SAVE AREA FOR OLD AREA
816                    001000      .END        START

```

Symbol table

BACKSP=	000010	FLEND	005121	M010	003275	M042	004006	M074	005053
BCSR	= 177520	FMSG1	002430	M011	003277	M043	004022	NARGS	= 000001
BDR	= 177524	FMSG1A	002465	M012	003277	M044	004057	NTYPE	= 000027
BIT6	= 000100	FMSG1B	002500	M013	003277	M045	004125	OLDSIZ	002424
BIT7	= 000200	FMSG1C	002531	M014	003277	M046	004164	PCR	= 177522
BUFF	005121	FMSG1D	002554	M015	003277	M047	004207	PCRLB	= 177522
B1	= 000002	FMSG2	002563	M016	003277	M050	004242	QUIT	002004
B2	= 000000	FMSG3	002644	M017	003277	M051	004272	QUIT1	002006
B3	= 000357	FMSG4	002737	M020	003277	M052	004323	REAROM	002314
B4	= 000143	LANG	001262	M021	003347	M053	004345	RETRY	= 000002
CKSUM	005113	LF	= 000012	M022	003377	M054	004372	RMVTST	= 173002
CR	= 000015	LNGHDR	= 000140	M023	003417	M055	004410	ROMADR	002342
CRLF	002560	MAXERR	= 000004	M024	003540	M056	004432	ROMSZ	= 001764
DELAY	= 025370	ME	005114	M025	003561	M057	004526	SPACE	= 000040
DUMMY1	002526	MEND	005114	M026	003562	M060	004540	START	001000
DUMMY2	002551	MEND1	005112	M027	003630	M061	004540	TAB	= 000011
ENDBLK	005113	MOVROM	002134	M030	003637	M062	004550	TEXT	003135
ENDE2R	= 166000	MSG000	003006	M031	003650	M063	004552	UFDHDR	= 000040
ENGWRD	005112	MSG001	003070	M032	003663	M064	004565	UFDSIZ	002426
ESC	= 000033	M001	003232	M033	003665	M065	004626	WB	005112
EXIT	001566	M002	003242	M034	003750	M066	004631	WE	005114
EXIT1	001642	M003	003244	M035	003750	M067	004657	WEND	005113
E2LLB	= 165006	M004	003252	M036	003751	M070	004740	WERR	002422
E2PAR	= 165316	M005	003262	M037	003751	M071	004750	WRBYTE	002162
E2PROM	= 165000	M006	003271	M040	003753	M072	004753	WRLANG	001462
E2WRIT	001666	M007	003273	M041	004006	M073	005051		

. ABS. 005121 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: 8558 Words (34 Pages)
 Size of core pool: 19402 Words (74 Pages)
 Operating system: RSX-11M/PLUS (Under VAX/VMS)

Elapsed time: 00:00:26.22
 OEEDA0.BIC,COEEDA0/CR/-SP=COEEDA0

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	005121	6-370	6-409	6-413	6-441	6-518	#6-815			
B1	= 000002	#6-803	6-808	6-812						
B2	= 000000	#6-804	6-809	6-812						
B3	= 000357	#6-805	6-810	6-812						
B4	= 000143	#6-806	6-811	6-812						
CKSUM	005113	6-428	6-497	*6-500	#6-793					
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	6-767	6-772	6-772	6-773
		6-773	6-774	6-774	6-776	6-776	6-777	6-777	6-783	6-783
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	005113	6-787	#6-788							
ENDE2R	= 166000	#5-245	6-349	6-503	6-523	6-596				
ENGWRD	005112	#6-787	6-787							
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	005121	5-264	#6-814							
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	005114	#6-798								
MEND	005114	#6-796	6-805	6-806						
MEND1	005112	6-487	6-719	#6-784						
MOVROM	002134	6-371	6-410	6-417	#6-547					
MSG000	003006	6-512	6-512	#6-653						
MSG001	003070	6-533	6-533	#6-654						
M001	003232	6-484	6-659	6-660	#6-720					
M002	003242	6-660	6-661	#6-721						
M003	003244	6-661	6-662	#6-722						

SYMBOL CROSS REFERENCE

CREF V02

SYMBOL	VALUE	REFERENCES		
M004	003252	6-662	6-663	#6-723
M005	003262	6-663	6-664	#6-724
M006	003271	6-664	6-665	#6-725
M007	003273	6-665	6-666	#6-726
M010	003275	6-666	6-667	#6-727
M011	003277	6-667	6-668	#6-728
M012	003277	6-668	6-669	#6-729
M013	003277	6-669	6-670	#6-730
M014	003277	6-670	6-671	#6-731
M015	003277	6-671	6-672	#6-732
M016	003277	6-672	6-673	#6-733
M017	003277	6-673	6-674	#6-734
M020	003277	6-674	6-675	#6-735
M021	003347	6-675	6-676	#6-736
M022	003377	6-676	6-677	#6-737
M023	003417	6-677	6-678	#6-738
M024	003540	6-678	6-679	#6-740
M025	003561	6-679	6-680	#6-741
M026	003562	6-680	6-681	#6-742
M027	003630	6-681	6-682	#6-743
M030	003637	6-682	6-683	#6-744
M031	003650	6-683	6-684	#6-745
M032	003663	6-684	6-685	#6-746
M033	003665	6-685	6-686	#6-747
M034	003750	6-686	6-687	#6-748
M035	003750	6-687	6-688	#6-749
M036	003751	6-688	6-689	#6-750
M037	003751	6-689	6-690	#6-751
M040	003753	6-690	6-691	#6-752
M041	004006	6-691	6-692	#6-753
M042	004006	6-692	6-693	#6-754
M043	004022	6-693	6-694	#6-755
M044	004057	6-694	6-695	#6-756
M045	004125	6-695	6-696	#6-757
M046	004164	6-696	6-697	#6-758
M047	004207	6-697	6-698	#6-759
M050	004242	6-698	6-699	#6-760
M051	004272	6-699	6-700	#6-761
M052	004323	6-700	6-701	#6-762
M053	004345	6-701	6-702	#6-763
M054	004372	6-702	6-703	#6-764
M055	004410	6-703	6-704	#6-765
M056	004432	6-704	6-705	#6-766
M057	004526	6-705	6-706	#6-768
M060	004540	6-706	6-707	#6-770
M061	004540	6-707	6-708	#6-771
M062	004550	6-708	6-709	#6-772
M063	004552	6-709	6-710	#6-773
M064	004565	6-710	6-711	#6-774
M065	004626	6-711	6-712	#6-775
M066	004631	6-712	6-713	#6-776
M067	004657	6-713	6-714	#6-777

SYMBOL CROSS REFERENCE

CREF V02

SYMBOL	VALUE	REFERENCES									
M070	004740	6-714	6-715	#6-778							
M071	004750	6-715	6-716	#6-779							
M072	004753	6-716	6-717	#6-780							
M073	005051	6-717	6-718	#6-782							
M074	005053	6-718	6-719	#6-783							
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	#6-580	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	#6-575	6-575	
		#6-575	6-575	#6-576	6-576	#6-579	6-579	#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	= 001764	#5-264	6-437	6-447							
SPACE	= 000040	#5-261	6-751	6-775							
START	001000	#6-314	6-816								
TAB	= 000011	#5-259	6-749	6-782	6-782						
TEXT	003135	5-264	6-424	6-446	#6-659	6-659	6-805	6-806			
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	005112	#6-786	6-803	6-804							
WE	005114	#6-799	6-803	6-804							
WEND	005113	#6-791									
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				