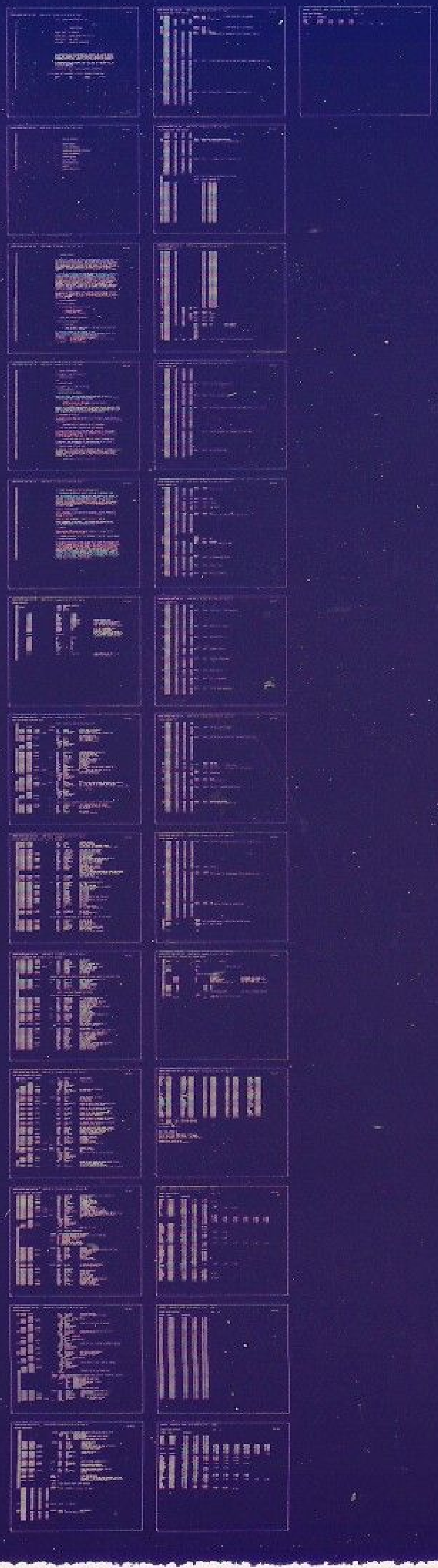
KDJ11-B

EEPROM SWED LANG LDR
COEEGB0

AH-FF29B-MC
1 OF 1 OCT 1985

MADE IN USA



```
                    .TITLE  COEEGB EEPROM SWED LANG LDR


                                                            .REM &

                                   IDENTIFICATION
                                   --------------

                    PRODUCT CODE:  AC-FF28B-MC

                    PRODUCT NAME:  COEEGBO EEPROM SWED LANG LDR

                    PRODUCT DATE:  JUNE, 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
```

TABLE OF CONTENTS

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 desciption 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 seperate 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
```

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 mmmmmm.
        Data written qqq, data read rrr.
```

where n is the EEPROM page selected by the Page Control Register (PCR), mmmmmm 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.
```

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 succesful 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.

```
                                         .SBTTL  PROGRAM CONSTANTS
000000                                   .ENABL  ABS
                                         .NLIST  MD,CND
                                         .LIST   ME

         177520                          BCSR    =       177520
         177522                          PCR     =       177522
         177522                          PCRLB   =       177522
         165000                          E2PROM  =       165000
         165316                          E2PAR   =       E2PROM+316              ;E2PROM PARITY BYTE
         165006                          E2LLB   =       E2PROM+6                ;LOCAL LANGUAGE BIT IN E2PROM
         166000                          ENDE2R  =       E2PROM+1000             ;LAST ADDRESS OF E2PROM+2
         173002                          RMVTST  =       173002                  ;WORD TO TEST ROM VERSION NUMBER
         025370                          DELAY   =       11000.
         000140                          LNGHDR  =       140                     ;I.D. OF A LANGUAGE AREA
         000040                          UFDHDR  =       040                     ;I.D. OF A UFD BLOCK
         000002                          RETRY   =       2                       ;NUMBER OF ATTEMPTS TO WRITE A
                                                                                 ;BYTE IN E2PROM BEFORE GIVING UP
         000004                          MAXERR  =       4                       ;NO. OF ERRORS ALLOWED IN LOCAL
                                                                                 ;LANGUAGE TEXT BEFORE QUITTING
         177524                          BDR     =       177524
         000015                          CR      =       15
         000012                          LF      =       12
         000200                          BIT7    =       200
         000100                          BIT6    =       100
         000011                          tab     =       11
         000010                          backsp  =       10
         000040                          space   =       40
         000033                          esc     =       33

         001551                          ROMSZ   =       FLEND-TEXT              ;SIZE IN BYTES OF TEXT TO BE
                                                                                 ;LOADED INTO EEPROM


```

```
                                                .SBTTL  CHECK FOR CERTAIN EXCEPTIONS FIRST

        001000                          .=1000

001000  005037  177522                  START:  CLR     @#PCR           ;SELECT PAGE 0 OF EEPROM
001004  013746  177520                          MOV     @#BCSR,-(SP)    ;SAVE OLD BCSR VALUE
001010  112737  000067  177520                  MOVB    #67,@#BCSR      ;WRITE ENABLE THE E2PROM & ENABLE ROM

001016  000007                                  MFPT                    ;GET PROCESSOR TYPE
001020  020027  000005                          CMP     R0,#5           ;CHECK TO SEE IF ORION
001024  001404                                  BEQ     1$              ;YES - CONTINUE
001026                                          .TYPMSG #FMSG2          ;FIELD-SERVICE MESSAGE
        000001                                  .NARG   NARGS
        000027                                  .NTYPE  NTYPE,#FMSG2
001026  012700  002563                            MOV   #FMSG2,R0
001032  104003                                    EMT   3
001034  000443                                  BR      99$

001036  012700  165000                  1$:     MOV     #E2PROM,R0      ;STARTING ADDRESS TO CHECKSUM
001042  005001                                  CLR     R1              ;INITIALIZE CHECKSUM
001044  012703  000151                          MOV     #105.,R3        ;NO. OF BYTES TO CKSUM
001050  012005                          201$:   MOV     (R0)+,R5        ;GET A BYTE
001052  042705  177400                          BIC     #177400,R5      ;NO BUS NOISE, THANK YOU.
001056  060501                                  ADD     R5,R1           ;ACCUMULATE CHECKSUM
001060  077305                                  SOB     R3,201$         ;CONTINUE TILL DONE
001062  105701                                  TSTB    R1              ;IS CKSUM 0?
001064  001007                                  BNE     202$            ;NO, ERROR
001066  105737  165022                          TSTB    @#E2PROM+22     ;BYTE TO TEST FOR VALID ROM, SHOULD BE 0
001072  001004                                  BNE     202$            ;NO, ERROR
001074  123727  165314  000252                  CMPB    @#E2PROM+314,#252       ;BYTE TO TEST FOR VALID ROM
001102  001404                                  BEQ     300$            ;GO TO NEXT CHECK IF OK
001104                                  202$:   .TYPMSG #FMSG4          ;FIELD SERVICE MESSAGE
        000001                                  .NARG   NARGS
        000027                                  .NTYPE  NTYPE,#FMSG4
001104  012700  002737                            MOV   #FMSG4,R0
001110  104003                                    EMT   3
001112  000414                                  BR      99$             ;QUIT
001114  005067  001304                  300$:   CLR     OLDSIZ          ;SET FLAG THAT ROM EXISTS, CURRENTLY NO LANGUAGE
001120  012737  000016  177522                  MOV     #7*2,@#PCR      ;SEL. LAST PAGE OF 2K E2PROM, PG0 OF ROM
001126  023727  173002                          CMP     @#RMVTST,(PC)+  ;SEE IF ROM VER. 7 OR LATER (CAN SUPPORT LANGUAGE AREA)
001132  000250                                  CLN
001134  001405                                  BEQ     2$              ;YES - CONTINUE
001136                                          .TYPMSG #FMSG3
        000001                                  .NARG   NARGS
        000027                                  .NTYPE  NTYPE,#FMSG3
001136  012700  002644                            MOV   #FMSG3,R0
001142  104003                                    EMT   3
001144  000167  000636                  99$:    JMP     QUIT1


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

001150  012700  165776                  2$:     MOV     #ENDE2R-2,R0    ;LAST ADDRESS (CKSUM) OF E2PROM
001154  012701  000005                          MOV     #5,R1           ;NO. OF BYTES IN HEADER TO CHECKSUM
001160  010005                                  MOV     R0,R5           ;SAVE ADDRESS
001162  005003                                  CLR     R3              ;
001164  111004                          4$:     MOVB    (R0),R4         ;GET A BYTE
001166  060403                                  ADD     R4,R3           ;ACCUMULATE CHECKSUM
```

```
001170  005740                          TST     -(R0)           ;CORRECT ADDRESS
001172  077104                          SOB     R1,4$           ;LOOP FOR 5 BYTES
001174  105703                          TSTB    R3              ;IF NOT ZERO, NO LANGUAGE LOADED
001176  001131                          BNE     WRLANG          ;NON-EXISTANT OR CORRUPTED LANGUAGE - SKIP

001200  014504                          MOV     -(R5),R4        ;HIGH BYTE OF BYTE COUNT
001202  014546                          MOV     -(R5),-(SP)     ;LOW BYTE OF BYTE COUNT
001204  110466  000001                  MOVB    R4,1(SP)        ;SET UPPER BYTES OF SIZE
001210  042704  177437                  BIC     #177437,R4      ;EXTRACT ID CODE
001214  012601                          MOV     (SP)+,R1        ;GET SIZE BACK
001216  042701  160000                  BIC     #160000,R1      ;R1 NOW CONTAINS SIZE OF BLOCK IN BYTES
001222  062701  000005                  ADD     #5,R1           ;ADD BYTE COUNT FOR HEADER BLOCK
001226  120427  000040                  CMPB    R4,#UFDHDR      ;SEE IF IT IS A UFD BLOCK
001232  001013                          BNE     LANG            ;NO, CHECK FOR A LANGUAGE
001234  010104                          MOV     R1,R4           ;SAVE SIZE
001236  012702  004655                  MOV     #BUFF,R2        ;ADDRESS OF SAVE BUFFER
001242  004767  000666                  CALL    MOVROM          ;MOVE UFD AREA TO MEMORY
001246  001105                          BNE     WRLANG          ;BAD CKSUM, QUIT
                                                                ;NOTE - R3 CONTAINS CHECKSUM OF BLOCK AND HEADER
                                                                ;HOWEVER THE CHECKSUM OF HEADER IS ALREADY KNOWN
                                                                ;TO BE 0 SO R3 IS A VALID CHECK OF UFD BLOCK
001250  010167  001150                  MOV     R1,OLDSIZ       ;SAVE TOTAL SIZE
001254  010167  001146                  MOV     R1,UFDSIZ       ;SAVE SIZE OF UFD AREA
001260  000500                          BR      WRLANG

001262  120427  000140          LANG:   CMPB    R4,#LNGHDR      ;IS THIS A LANGUAGE HEADER?
001266  001075                          BNE     WRLANG          ;NO - QUIT
001270  010167  001130                  MOV     R1,OLDSIZ       ;SAVE SIZE FOR NOW
001274  062701  000005                  ADD     #5,R1           ;ADD SIZE OF (POSSIBLE) UFD HEADER
001300  004767  001036                  CALL    ROMADR          ;SET UP PCR AND R0
001304  005003                          CLR     R3              ;INITIALIZE CKSUM
001306  004767  001002                  CALL    REAROM          ;GET A BYTE
001312  004767  000776                  CALL    REAROM          ;GET A BYTE
001316  004767  000772                  CALL    REAROM          ;GET A BYTE
001322  010546                          MOV     R5,-(SP)        ;SAVE LOW BYTE OF SIZE FOR LATER
001324  004767  000764                  CALL    REAROM          ;GET A BYTE
001330  110566  000001                  MOVB    R5,1(SP)        ;SAVE HIGH BYTE OF SIZE AND ID
001334  004767  000754                  CALL    REAROM          ;GET A BYTE
001340  116600  000001                  MOVB    1(SP),R0        ;GET I.D.
001344  012601                          MOV     (SP)+,R1        ;GET SIZE
001346  105703                          TSTB    R3              ;SEE IF VALID CKSUM
001350  001025                          BNE     1$              ;NO - WE HAVE LANGUAGE ONLY.

001352  042700  177437                  BIC     #177437,R0      ;GET ID ONLY
001356  120027  000040                  CMPB    R0,#UFDHDR      ;IS THIS A UFD BLOCK?
001362  001020                          BNE     1$              ;NO, IGNORE IT.

                                ;WE HAVE BOTH A LANGUAGE AREA AND A UFD BLOCK.  SAVE THE UFD BLOCK.

001364  042701  160000                  BIC     #160000,R1      ;GET RID OF ID
001370  062701  000005                  ADD     #5,R1           ;SIZE OF HEADER
001374  010104                          MOV     R1,R4           ;BYTE COUNT TO MOVE
001376  010167  001024                  MOV     R1,UFDSIZ       ;SAVE UFD SIZE
001402  066701  001016                  ADD     OLDSIZ,R1       ;ADD SIZE OF LANGUAGE AREA
001406  012702  004655                  MOV     #BUFF,R2        ;MEMORY ADDRESS TO SAVE TO
001412  004767  000516                  CALL    MOVROM          ;SAVE UFD AREA
001416  001404                          BEQ     2$              ;YES, IT IS VALID, CONTINUE
```

```
001420  005067  001002                   CLR     UFDSIZ          ;NO UFD AREA
001424  012702  004655          1$:      MOV     #BUFF,R2        ;RESET R2
001430  016701  000770          2$:      MOV     OLDSIZ,R1       ;SIZE OF LANGUAGE AREA
001434  010104                           MOV     R1,R4           ;BYTES TO MOVE
001436  066767  000764  000760           ADD     UFDSIZ,OLDSIZ   ;OLDSIZ IS THE TOTAL SIZE
001444  004767  000464                   CALL    MOVROM          ;SAVE LANGUAGE AREA
001450  001404                           BEQ     WRLANG          ;LANGUAGE IS GOOD
001452  005067  000746                   CLR     OLDSIZ          ;NO LANGUAGE
001456  005067  000744                   CLR     UFDSIZ          ;NO UFD AREA

                                ;GENERATE CHECKSUM FOR FOREIGN LANGUAGE TEST FILE & WRITE TO THE MEMORY IMAGE

001462  012700  003104          WRLANG:  MOV     #TEXT,R0        ;ADDRESS OF BEGINNING OF TEXT
001466  005001                           CLR     R1              ;INIT CHECKSUM
001470  112002                  25$:     MOVB    (R0)+,R2        ;READ A BYTE
001472  160201                           SUB     R2,R1           ;ACCUMULATE CHECKSUM
001474  020027  004647                   CMP     R0,#CKSUM       ;FINISHED ALL TEXT ?
001500  001373                           BNE     25$             ;NO-CONTINUE
001502  110110                           MOVB    R1,(R0)         ;WRITE THE CHECKSUM

                                         .SBTTL  LOAD LOCAL LANGUAGE INTO E2PROM

                                ;WRITE UFD & LOCAL LANGUAGE BLOCKS

001504  016701  000716                   MOV     UFDSIZ,R1       ;GET THE LENGTH OF THE UFD
001510  062701  001551                   ADD     #ROMSZ,R1       ;... & THE TEXT AREA
001514  004767  000622                   JSR     PC,ROMADR       ;COMPUTE E2PROM PAGE AND ADDR
001520  016701  000702                   MOV     UFDSIZ,R1       ;SIZE OF UFD AREA TO SAVE
001524  001406                           BEQ     40$             ;NO UFD AREA - SKIP
001526  012702  004655                   MOV     #BUFF,R2        ;ADDRESS OF BEGINNING OF UFD AREA
001532  112205                  35$:     MOVB    (R2)+,R5        ;GET SOME DATA
001534  004767  000126                   CALL    E2WRIT          ;GO WRITE IT
001540  077104                           SOB     R1,35$          ;FINISHED UFD?
                                                                 ;YES-DO LANGUAGE
001542  012702  003104          40$:     MOV     #TEXT,R2        ;ADDRESS OF EEPROM LANGUAGE TEXT
001546  012701  001551                   MOV     #ROMSZ,R1       ;BYTES TO MOVE
001552  112205                  50$:     MOVB    (R2)+,R5        ;GET SOME DATA
001554  004767  000106                   CALL    E2WRIT          ;WRITE A BYTE
001560  077104                           SOB     R1,50$          ;ARE WE DONE?
                                                                 ;YES - EXIT
001562  112705  000200                   MOVB    #BIT7,R5        ;TURN ON LOCAL LANGUAGE BIT IN
                                                                 ;SETUP AREA, THEN EXIT

001566  105037  177522          EXIT:    CLRB    @#PCRLB         ;SELECT PAGE 0
001572  012700  165006                   MOV     #E2LLB,R0       ;E2PROM WORD CONTAINING LOCAL LANG. BIT
001576  111001                           MOVB    (R0),R1
001600  142701  177577                   BICB    #^CBIT7,R1      ;GET CURRENT LOCAL LANGUAGE BIT
001604  120501                           CMPB    R5,R1           ;SEE IF BIT ALREADY CORRECT
001606  001415                           BEQ     EXIT1           ;YES, JUST RETURN
001610  112701  000200                   MOVB    #BIT7,R1        ;LOCAL LANGUAGE BIT
001614  111005                           MOVB    (R0),R5         ;GET OLD WORD AGAIN
001616  074105                           XOR     R1,R5           ;FLIP THE BIT
001620  004767  000336                   CALL    WRBYTE          ;CHANGE LOCAL LANGUAGE BIT IN E2PROM
001624  001006                           BNE     EXIT1           ;WOULD NOT WRITE, JUST GIVE UP
001626  012700  165316                   MOV     #E2PAR,R0       ;ADDRESS OF CKSUM BYTE
001632  111005                           MOVB    (R0),R5         ;GET OLD CKSUM BYTE
001634  074105                           XOR     R1,R5           ;CORRECT THE CKSUM
```

```
001636  004767  000320                      CALL     WRBYTE          ;UPDATE E2ROM

001642                           EXIT1:     .FRCTYP  #CRLF           ;COMPLETE LINE
        000001                              .NARG    NARGS
        000027                              .NTYPE   NTYPE,#CRLF
001642  012700  002560                        MOV    #CRLF,R0
001646  104044                                EMT    44
001650  142716  000060                      BICB     #60,(SP)        ;BE SURE ROM IS DISABLED
001654  012637  177520                      MOV      (SP)+,@#BCSR    ;RESTORE BCSR
001660  005037  177522                      CLR      @#PCR           ;
001664  000207                              RTS      PC

001666  004767  000270           E2WRIT:    CALL     WRBYTE          ;WRITE THE BYTE TO E2PROM
001672  001431                              BEQ      3$              ;OK THIS TIME
001674  005267  000522                      INC      WERR            ;FLAG BAD BYTE

001700  026727  000516  000004              CMP      WERR,#MAXERR    ;CHECK TO SEE IF PAST THE MAXIMUM ERROR
001706  003036                              BGT      QUIT            ;LIMIT OF BAD BYTES ALLOWED

001710  020227  003201                      CMP      R2,#M001        ;CHECK TO SEE IF ERROR IS IN MESSAGE
001714  101433                              BLOS     QUIT            ;BYTE COUNT (MUST BE CORRECT)

001716  020227  004646                      CMP      R2,#MEND1       ;CHECK TO BE SURE DICTIONARY AND UFD
0C1722  101030                              BHI      QUIT            ;BLOCKS ARE NOT CORRUPTED

001724  132705  000140                      BITB     #140,R5         ;CHECK TO SEE IF IT SHOULD BE A CONTROL
001730  001425                              BEQ      QUIT            ;CODE (POSSIBLY DICTIONARY ENTRY)

001732  132710  000140                      BITB     #140,(R0)       ;IF CONTROL CODE (DICTIONARY REFERENCE
001736  001422                              BEQ      QUIT            ;PERHAPS) CALL IT QUITS

001740  111004                              MOVB     (R0),R4         ;WE WILL LIVE WITH THIS ERROR, CORRECT
001742  116703  002701                      MOVB     CKSUM,R3        ;THE CHECKSUM TO ACCOUNT FOR NEW VALUE
001746  060503                              ADD      R5,R3           ;CANCEL OUT WHAT WAS SUPPOSED TO BE
001750  160403                              SUB      R4,R3           ;CORRECT FOR ERRONEOUS VALUE
001752  110367  002671                      MOVB     R3,CKSUM        ;PUT BACK CORRECTED VALUE

001756  062700  000002           3$:        ADD      #2,R0           ;INCREMENT LOCATION
001762  020027  166000                      CMP      R0,#ENDE2R      ;FINISHED THIS PAGE ?
001766  001005                              BNE      10$             ;NO-RETURN
001770  012700  165000                      MOV      #E2PROM,R0      ;YES-RESET ADDRESS
001774  062737  000002  177522              ADD      #2,@#PCR        ;INCREMENT PCR TO NEXT PAGE
002002  000207                   10$:       RETURN

002004  005726                   QUIT:      TST      (SP)+           ;CORRECT STACK
002006  032737  000100  000052   QUIT1:     BIT      #BIT6,@#52      ;SEE IF UFD QUIET
002014  001403                              BEQ      5$              ;NO
002016                                      .FRCTYP  #MSG000         ;MESSAGE FOR USER IN HIS OWN LANGUAGE
        000001                              .NARG    NARGS
        000027                              .NTYPE   NTYPE,#MSG000
002016  012700  003006                        MOV    #MSG000,R0
002022  104044                                EMT    44
002024  016701  000374           5$:        MOV      OLDSIZ,R1
002030  100704                              BMI      EXIT1           ;ERROR WAS NOT ORION OR CKSUM ERROR, DO NOT
                                                                     ;TRY TO CLEAR LANGUAGE BIT
002032  001427                              BEQ      40$             ;IF NO OLD LANGUAGE TO RESTORE
002034  004767  000302                      JSR      PC,ROMADR       ;COMPUTE STARTING ADDRESS OF OLD LANG IN E2PROM
```

```
 002040 012702  004655                   MOV     #BUFF,R2         ;STARTING ADDRESS OF OLD LANGUAGE TEXT
 002044 112205                   10$:    MOVB    (R2)+,R5         ;GET A BYTE
 002046 004767  000110                   CALL    WRBYTE           ;WRITE IT OUT
 002052 001017                           BNE     40$              ;IF ERROR, GIVE UP
 002054 062700  000002                   ADD     #2,R0            ;INCREMENT LOCATION
 002060 020027  166000                   CMP     R0,#ENDE2R       ;FINISHED THIS PAGE ?
 002064 001005                           BNE     20$              ;NO-CONTINUE
 002066 012700  165000                   MOV     #E2PROM,R0       ;YES-RESET ADDRESS
 002072 062737  000002  177522           ADD     #2,@#PCR         ;INCREMENT PCR TO NEXT PAGE
 002100 077117                   20$:    SOB     R1,10$           ;LOOP UNTIL DONE
 002102 026767  000320  000314           CMP     UFDSIZ,OLDSIZ    ;IF THE SAME THEN NO LANGUAGE
 002110 001254                           BNE     EXIT1            ;IF LANGUAGE, LEAVE E2PROM LANG. BIT AS IT WAS
 002112 005005                   40$:    CLR     R5               ;TURN OFF LOCAL LANGUAGE BIT IN E2PROM
 002114 036737  175760  000052           BIT     BIT6,@#52        ;SEE IF UFD QUIET
 002122 001621                           BEQ     EXIT             ;NO
 002124                                  .FRCTYP #MSG001
        000001                           .NARG   NARGS
        000027                           .NTYPE  NTYPE,#MSG001
 002124 012700  003045                     MOV   #MSG001,R0
 002130 104044                             EMT   44
 002132 000615                           BR      EXIT             ;AND CALL IT A DAY


                                         .SBTTL  PROGRAM SUBROUTINES


                                 ;MOVROM - MOVE BYTES FROM EEPROM TO MEMORY
                                 ;ENTRY- R1 = STARTING ADDRESS IN EEPROM (# OF BYTES FROM END)
                                 ;       R2 = ADDRESS OF MEMORY BUFFER
                                 ;       R4 = # OF BYTES TO MOVE
                                 ;EXIT   R1 - UNCHANGED
                                 ;       R2 - UPDATED MEMORY ADDRESS
                                 ;       R3 = (BYTE) 0 IF VALID CKSUM
                                 ;       "Z" FLAG SET IF CKSUM VALID

 002134 010403                   MOVROM: MOV     R4,R3            ;SAVE R4
 002136 004767  000200                   CALL    ROMADR           ;LOAD PCR AND R0 WITH LANGUAGE START AREA
 002142 010304                           MOV     R3,R4            ;RESTORE BYTE COUNT
 002144 005003                           CLR     R3               ;INIT CHECKSUM
 002146 004767  000142           5$:     CALL    REAROM           ;GET A BYTE
 002152 110522                           MOVB    R5,(R2)+         ;SAVE IT
 002154 077404                           SOB     R4,5$            ;LOOP TILL DONE
 002156 105703                           TSTB    R3               ;IS CHECKSUM GOOD?
 002160 000207                           RETURN

 002162 120510                   WRBYTE: CMPB    R5,(R0)          ;IS THE NEW DATA DIFFERENT ?
 002164 001452                           BEQ     10$              ;NO-DO NOT WRITE OVER

 002166 012703  000002                   MOV     #RETRY,R3
 002172 010510                   1$:     MOV     R5,(R0)          ;WRITE A LOCATION
 002174 012704  025370                   MOV     #DELAY,R4        ;11 MS WAIT
 002200 077401                           SOB     R4,.             ;WASTE TIME
 002202 120510                           CMPB    R5,(R0)          ;SEE IF IT TOOK
 002204 001442                           BEQ     10$              ;YES, ALL OKAY
 002206 077307                           SOB     R3,1$            ;IF AT FIRST YOU DON'T SUCCEED...
 002210 113704  177522                   MOVB    @#PCRLB,R4       ;PCR PAGE OF BAD BYTE
 002214 106204                           ASRB    R4               ;CONVERT TO PAGE #
 002216 062704  000060                   ADD     #'0,R4           ;CONVERT TO OCTAL
 002222 110467  000237                   MOVB    R4,FMSG1A        ;STORE IT FOR PRINTING
```

```
002226  010046                                  MOV     R0,-(SP)        ;SAVE ROM ADDRESS
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
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
002244  042705  177400                          BIC     #177400,R5      ;MAKE SURE R5 IS POSITIVE AND A BYTE
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
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
002266  013600                                  MOV     @(SP)+,R0       ;GET BYTE AT ROM ADDRESS
002270  042700  177400                          BIC     #177400,R0      ;GET RID OF BUS NOISE
002274                                          .ITOA   ,#DUMMY2        ;CONVERT TO OCTAL
        000002                                  .NARG   NARGS
        000027                                    .NTYPE        NTYPE,#DUMMY2
002274  012701  002551                            MOV   #DUMMY2,R1
002300  104030                                    EMT   30
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
002310  000244                                  CLZ                     ;COULDN'T DO IT, SET ERROR FLAG
002312  000207                          10$:    RETURN

                                        ;REAROM - READS A BYTE FROM E2PROM ADDRESS (R0)+ INTO R5. AUTOMATICLY ADJUSTS
                                        ;PCRLB.  UPDATES CKSUM IN R3
                                        ;       ENTRY - R0      ADDRESS IN ROM TO READ FROM
                                        ;               R3      PARTIAL CKSUM
                                        ;               PCRLB   CORRECT VALUE FOR BYTE TO READ
                                        ;       EXIT    R0      ADDRESS OF NEXT BYTE
                                        ;               R3      UPDATED CKSUM
                                        ;               R5      BYTE READ
                                        ;               PCRLB   CORRECT VALUE FOR NEXT BYTE

002314  012005                          REAROM: MOV     (R0)+,R5        ;GET A BYTE & UPDATE ADDR. BY 2
002316  060503                                  ADD     R5,R3           ;UPDATE CKSUM
002320  020027  166000                          CMP     R0,#ENDE2R      ;SEE IF WE SHOULD SWITCH PAGES
002324  001005                                  BNE     10$             ;NO
002326  012700  165000                          MOV     #E2PROM,R0      ;YES - GO TO START OF PAGE
002332  062737  000002  177522                  ADD     #2,@#PCR                ;ADVANCE A PAGE
002340  000207                          10$:    RETURN

```

```

                                   ;ROMADR - CALCULATE PAGE OFFSET FROM END OF ROM GIVEN SIZE IN BYTES
                                   ;        ENTRY - R1       SIZE IN BYTES
                                   ;        EXIT -  R0       INITIAL ADDRESS FOR FIRST BYTE IN ROM
                                   ;                R1       SIZE IN BYTES
                                   ;                PCRLB    CORRECT VALUE FOR FIRST BYTE IN ROM

 002342  010100                    ROMADR: MOV     R1,R0           ;COPY BYTE COUNT
 002344  010105                            MOV     R1,R5           ;SECOND COPY
 002346  072527  177770                    ASH     #-8.,R5         ;DIVIDE BYTE COUNT BY 256. BYTE PAGES
 002352  012704  000010                    MOV     #7+1,R4         ;LAST PAGE IN 2 K PART + 1
 002356  160504                            SUB     R5,R4           ;STARTING PAGE NUMBER

 002360  042700  177400                    BIC     #177400,R0      ;LEAVE ONLY BITS 7:0
 002364  006300                            ASL     R0              ;DOUBLE VALUE
 002366  001003                            BNE     20$
 002370  012700  165000                    MOV     #E2PROM,r0      ;
 002374  000406                            BR      30$             ;IF 0

 002376  005400                    20$:    NEG     R0              ;MAKE STARTING ADDRESS BITS 8:0
 002400  042700  177000                    BIC     #177000,R0      ;
 002404  052700  165000                    BIS     #E2PROM,R0      ;MAKE A E2PROM ADDRESS
 002410  005304                            DEC     R4              ;DECREMENT PAGE NUMBER BY 1

 002412  006304                    30$:    ASL     R4              ;MAKE PAGE NUMBER CORRECT FOR PCR
 002414  110437  177522                    MOVB    R4,@#PCRLB      ;CORRECT PAGE IN PCRLB
 002420  000207                            RTS     PC              ;RETURN

 002422  000000                    WERR:   0                       ;FLAG FOR BAD BYTE
 002424  177777                    OLDSIZ: -1                      ;>0 - SIZE IN BYTES OF OLD LANGUAGE, 0 IF NO
                                                                   ;LANGUAGE, -1 IF E2PROM MAY BE BAD/NONEXISTANT
 002426  000000                    UFDSIZ: 0                       ;SIZE IN BYTES OF OLD UFD AREA

                                   .SBTTL  "FIELD SERVICE MODE" ERROR MESSAGES

                                   .ENABL LC
 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
 002465     130     054     040    FMSG1A: .ASCII  /X, address /
 002470     141     144     144
 002473     162     145     163
 002476     163     040
 002500                            FMSG1B: .BLKB   6               ;FOR ADDRESS
 002506     015     012     104            .ASCIZ  <CR><LF>/Data written /
 002511     141     164     141
 002514     040     167     162
 002517     151     164     164
 002522     145     156     040
 002525     000
```

```
    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
        0C2627     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
```

```
      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       123       171    MSG000: .ASCIZ  <CR>!Systemet kan ej ladda Svenska!
      003011      163       164       145
      003014      155       145       164
      003017      040       153       141
      003022      156       040       145
      003025      152       040       154
      003030      141       144       144
      003033      141       040       123
      003036      166       145       156
      003041      163       153       141
      003044      000
654   003045      040       055       040    MSG001: .ASCIZ  ! - ]tergOr till U.S. English.!<CR>
      003050      135       164       145
      003053      162       147       175
      003056      162       040       164
      003061      151       154       154
      003064      040       125       056
      003067      123       056       040
      0C3072      105       156       147
      003075      154       151       163
      003100      150       056       015
      003103      000
655                                          .SBTTL  START OF AREA TO BE LOADED INTO E2PROM
656
657                                          .SBTTL  Svenska LANGUAGE TEXT
658
659   003104      075                        TEXT:   .BYTE   M001-TEXT
660   003105      010                                .BYTE   M002-M001
661   003106      002                                .BYTE   M003-M002
662   003107      006                                .BYTE   M004-M003
663   003110      006                                .BYTE   M005-M004
664   003111      005                                .BYTE   M006-M005
665   003112      002                                .BYTE   M007-M006
666   003113      002                                .BYTE   M010-M007
667   003114      002                                .BYTE   M011-M010
668   003115      000                                .BYTE   M012-M011
669   003116      000                                .BYTE   M013-M012
670   003117      000                                .BYTE   M014-M013
671   003120      000                                .BYTE   M015-M014
672   003121      000                                .BYTE   M016-M015
673   003122      000                                .BYTE   M017-M016
674   003123      000                                .BYTE   M020-M017
675   003124      037                                .BYTE   M021-M020
676   003125      025                                .BYTE   M022-M021
677   003126      026                                .BYTE   M023-M022
678   003127      117                                .BYTE   M024-M023
679   003130      021                                .BYTE   M025-M024
680   003131      001                                .BYTE   M026-M025
681   003132      020                                .BYTE   M027-M026
682   003133      004                                .BYTE   M030-M027
683   003134      010                                .BYTE   M031-M030
684   003135      012                                .BYTE   M032-M031
```

```
    685 003136      002                                .BYTE    M033-M032
    686 003137      050                                .BYTE    M034-M033
    687 003140      000                                .BYTE    M035-M034
    688 003141      001                                .BYTE    M036-M035
    689 003142      000                                .BYTE    M037-M036
    690 003143      002                                .BYTE    M040-M037
    691 003144      027                                .BYTE    M041-M040
    692 003145      000                                .BYTE    M042-M041
    693 003146      017                                .BYTE    M043-M042
    694 003147      015                                .BYTE    M044-M043
    695 003150      021                                .BYTE    M045-M044
    696 003151      036                                .BYTE    M046-M045
    697 003152      024                                .BYTE    M047-M046
    698 003153      023                                .BYTE    M050-M047
    699 003154      020                                .BYTE    M051-M050
    700 003155      026                                .BYTE    M052-M051
    701 003156      015                                .BYTE    M053-M052
    702 003157      021                                .BYTE    M054-M053
    703 003160      027                                .BYTE    M055-M054
    704 003161      021                                .BYTE    M056-M055
    705 003162      060                                .BYTE    M057-M056
    706 003163      012                                .BYTE    M060-M057
    707 003164      000                                .BYTE    M061-M060
    708 0C3165      013                                .BYTE    M062-M061
    709 003166      002                                .BYTE    M063-M062
    710 003167      013                                .BYTE    M064-M063
    711 003170      031                                .BYTE    M065-M064
    712 003171      003                                .BYTE    M066-M065
    713 003172      025                                .BYTE    M067-M066
    714 003173      060                                .BYTE    M070-M067
    715 003174      007                                .BYTE    M071-M070
    716 003175      003                                .BYTE    M072-M071
    717 003176      066                                .BYTE    M073-M072
    718 003177      001                                .BYTE    M074-M073
    719 003200      035                                .BYTE    MEND1-M074
    720 003201      123     166     145     M001:      .ASCIZ   !Svenska!
        003204      156     163     153
        003207      141     000
    721 003211      077     000             M002:      .ASCIZ   !?!
    722 003213      110     112     133     M003:      .ASCIZ   !HJ[LP!
        003216      114     120     000
    723 003221      114     101     104     M004:      .ASCIZ   !LADDA!
        003224      104     101     000
    724 003227      126     111     123     M005:      .ASCIZ   !VISA!
        003232      101     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      105     156     150     M020:      .ASCII   !Enhet!<TAB>!Enhetsnummer!<TAB>!Beskrivning!<CR>
        003245      145     164     011
        003250  105     156     150
        003253  145     164     163
        003256  156     165     155
        003261  155     145     162
        003264  011     102     145
        003267  163     153     162
        003272  151     166     156
        003275  151     156     147
        003300  015
    736 003301  126     151     163  M021:   .ASCII  !Visa startprogrammen!<CR>
        003304  141     040     163
        003307  164     141     162
        003312  164     160     162
        003315  157     147     162
        003320  141     155     155
        003323  145     156     015
    737 003326  123     164     141  M022:   .ASCII  !Startar systemet frOn !
        003331  162     164     141
        003334  162     040     163
        003337  171     163     164
        003342  145     155     145
        003345  164     040     146
        003350  162     175     156
        0C3353  040
    738 003354  015     113     157  M023:   .ASCII  <CR>!Kommando Beskrivning!<CR><CR>!LADDA!<TAB>! L|ser in och !
        003357  155     155     141
        003362  156     144     157
        003365  040     102     145
        003370  163     153     162
        003373  151     166     156
        003376  151     156     147
        003401  015     015     114
        003404  101     104     104
        003407  101     011     040
        003412  114     173     163
        003415  145     162     040
        003420  151     156     040
        003423  157     143     150
        003426  040
    739 003427  163     164     141          .ASCII  !startar systemet frOn enheten!<CR>!VISA!<TAB>! !
        003432  162     164     141
        003435  162     040     163
        003440  171     163     164
        003443  145     155     145
        003446  164     040     146
        003451  162     175     156
        003454  040     145     156
        003457  150     145     164
        003462  145     156     015
        003465  126     111     123
        003470  101     011     040
    740 003473  015     111     156  M024:   .ASCII  <CR>!Inl|sning pOgOr !
        003476  154     173     163
        003501  156     151     156
        003504  147     040     160
        003507  175     147     175
        003512  162     040
    741 003514     057                 M025:    .ASCII  '/'
    742 003515     124     162     171 M026:    .ASCII  !Tryck p0 <ret>: !
        003520     143     153     040
        003523     160     175     040
        003526     074     162     145
        003531     164     076     072
        003534     040
    743 003535     106     145     154 M027:    .ASCII  !Fel !
        003540     040
    744 003541     040     141     144 M030:    .ASCII  ! adress !
        003544     162     145     163
        003547     163     040
    745 003551     124     145     163 M031:    .ASCII  !Test p0g0r!
        003554     164     040     160
        003557     175     147     175
        003562     162
    746 003563     060     055         M032:    .ASCII  /0-/
    747 003565     015     123     153 M033:    .ASCII  <CR>!Skriv ett kommando och tryck p0 <ret>: !
        003570     162     151     166
        003573     040     145     164
        003576     164     040     153
        003601     157     155     155
        003604     141     156     144
        003607     157     040     157
        003612     143     150     040
        003615     164     162     171
        003620     143     153     040
        003623     160     175     040
        003626     074     162     145
        003631     164     076     072
        003634     040
    748 003635                         M034:
    749 003635     011                 M035:    .BYTE   TAB
    750 003636                         M036:
    751 003636     015     040         M037:    .BYTE   CR,SPACE
    752 003640     123     164     141 M040:    .ASCII  !Startar laddning av ROM!
        003643     162     164     141
        003646     162     040     154
        003651     141     144     144
        003654     156     151     156
        003657     147     040     141
        003662     166     040     122
        003665     117     115
    753 003667                         M041:
    754 003667     015     115     145 M042:    .ASCII  <CR>!Meddelande 06!<CR>
        003672     144     144     145
        003675     154     141     156
        003700     144     145     040
        003703     060     066     015
    755 003706     105     156     150 M043:    .ASCII  !Enhet ej klar!
        003711     145     164     040
        003714     145     152     040
        003717     153     154     141
        003722     162
    756 003723     115     145     144 M044:    .ASCII  !Media ej laddbart!
        003726     151     141     040
        003731     145     152     040
```

```
     003734      154     141     144
     003737      144     142     141
     003742      162     164
 757 003744      111     156     147  M045:   .ASCII  !Inget media i laddningsenheten!
     003747      145     164     040
     003752      155     145     144
     003755      151     141     040
     003760      151     040     154
     003763      141     144     144
     003766      156     151     156
     003771      147     163     145
     003774      156     150     145
     003777      164     145     156
 758 004002      111     156     147  M046:   .ASCII  !Inget band i enheten!
     004005      145     164     040
     004010      142     141     156
     004013      144     040     151
     004016      040     145     156
     004021      150     145     164
     004024      145     156
 759 004026      123     164     171  M047:   .ASCII  !Styrenhet finns ej,!
     004031      162     145     156
     004034      150     145     164
     0C4037      040     146     151
     004042      156     156     163
     004045      040     145     152
     004050      054
 760 004051      105     156     150  M050:   .ASCII  !Enheten finns ej!
     004054      145     164     145
     004057      156     040     146
     004062      151     156     156
     004065      163     040     145
     004070      152
 761 004071      117     147     151  M051:   .ASCII  !Ogiltigt enhetsnummer !
     004074      154     164     151
     004077      147     164     040
     004102      145     156     150
     004105      145     164     163
     004110      156     165     155
     004113      155     145     162
     004116      040
 762 004117      117     147     151  M052:   .ASCII  !Ogiltig enhet!
     004122      154     164     151
     004125      147     040     145
     004130      156     150     145
     004133      164
 763 004134      106     145     154  M053:   .ASCII  !Fel i styrenheten!
     004137      040     151     040
     004142      163     164     171
     004145      162     145     156
     004150      150     145     164
     004153      145     156
 764 004155      106     145     154  M054:   .ASCII  !Fel p0 laddningsenheten!
     004160      040     160     175
     004163      040     154     141
     004166      144     144     156
     004171      151     156     147
```

```
    004174   163   145   156
    004177   150   145   164
    004202   145   156
    004204   015   015   114  M055:   .ASCII  <CR><CR>!Laddning pOgOr !
    004207   141   144   144
    004212   156   151   156
    004215   147   040   160
    004220   175   147   175
    004223   162   040
    004225   015   123   145  M056:   .ASCII  <CR>!Se avsnitt fels)kning i Handledning f)r hj!lp!
    004230   040   141   166
    004233   163   156   151
    004236   164   164   040
    004241   146   145   154
    004244   163   174   153
    004247   156   151   156
    004252   147   040   151
    004255   040   110   141
    004260   156   144   154
    004263   145   144   156
    004266   151   156   147
    004271   040   146   174
    004274   162   040   150
    004277   152   173   154
    004302   160
    004303   015   015               .ASCII  <CR><CR>
    004305   033   133   062  M057:  .ASCII  <ESC>/[2J/        ;Erase screen
    004310   112
    004311   033   133   065          .ASCII  <ESC>/[5;0H/      ;Set cursor to line 5 and col 1
    004314   073   060   110
    004317                   M060:
    004317   115   145   144  M061:  .ASCII  !Meddelande !
    004322   144   145   154
    004325   141   156   144
    004330   145   040
    004332   015   015         M062:  .BYTE   CR,CR
    004334   015   015   113  M063:  .ASCII  <CR><CR>/KDJ11-B >/
    004337   104   112   061
    004342   061   055   102
    004345   040   076
    004347   015   106   145  M064:  .ASCII  <CR>!Fel vid EEPROM laddning!<CR>
    004352   154   040   166
    004355   151   144   040
    004360   105   105   120
    004363   122   117   115
    004366   040   154   141
    004371   144   144   156
    004374   151   156   147
    004377   015
    004400   010   040   010  M065:  .BYTE   BACKSP,SPACE,BACKSP
    004403   015   106   145  M066:  .ASCII  <CR>!Felaktigt kommando.!<CR>
    004406   154   141   153
    004411   164   151   147
    004414   164   040   153
    004417   157   155   155
    004422   141   156   144
    004425   157   056   015
    777 004430     015      015      124  M067:    .ASCII  <CR><CR>!Tillg|ngliga kommandon: Hj|lp, Ladda och Visa.!
        004433     151      154      154
        004436     147      173      156
        004441     147      154      151
        004444     147      141      040
        004447     153      157      155
        004452     155      141      156
        004455     144      157      156
        004460     072      040      110
        004463     152      173      154
        004466     160      054      040
        004471     114      141      144
        004474     144      141      040
        004477     157      143      150
        004502     040      126      151
        004505     163      141      056
    778 004510     101      144      162  M070:    .ASCII  !Adress !
        004513     145      163      163
        004516     040
    779 004517     040      075      040  M071:    .ASCII  / = /
    780 004522     123      153      162  M072:    .ASCII  !Skriv enhet och enhetsnummer tryck d|refter p0 <ret>: !
        004525     151      166      040
        004530     145      156      150
        004533     145      164      040
        004536     157      143      150
        004541     040      145      156
        004544     150      145      164
        004547     163      156      165
        004552     155      155      145
        004555     162      040      164
        004560     162      171      143
        004563     153      040      144
        004566     173      162      145
        004571     146      164      145
        004574     162      040      160
        004577     175      040      074
        004602     162      145      164
        004605     076      072      040
    781 004610     011                     M073:    .BYTE   TAB
    782 004611     015      123      164  M074:    .ASCII  <CR>!Startar automatisk laddning!<CR>
        004614     141      162      164
        004617     141      162      040
        004622     141      165      164
        004625     157      155      141
        004630     164      151      163
        004633     153      040      154
        004636     141      144      144
        004641     156      151      156
        004644     147      015
    783 004646                             MEND1:
    784                                    .SBTTL  NULL DICTIONARY BLOCK, CHECKSUM AND LANGUAGE HEADER
    785 004646                             wb:
    786 004646     001                     ENGWRD: .BYTE   ENDBLK-ENGWRD
    787 004647                             ENDBLK:
    788
    789
    790 004647                             WEND:
```

```

004647      000              CKSUM:  .byte   0        ;checksum


004650                       MEND:                              ;END OF NULL TEXT

004650                       ME:
004650                       WE:

                             ;FOREIGN LANGUAGE HEADER

         000002              B1      =       WE-WB&377                   ;DICTIONARY BYTE COUNT 7:0
         000000              B2      =       WE-WB&17400/256.            ;DICTIONARY BYTE COUNT 10:8
         000144              B3      =       MEND-text&377               ;TEXT BYTE COUNT 7:0
         000143              B4      =       MEND-text&017400/256.!140   ;TEXT BYTE COUNT 12:8 & ID=011

004650      002                      .BYTE   B1
004651      000                      .BYTE   B2
004652      144                      .BYTE   B3
004653      143                      .BYTE   B4
004654      067                      .BYTE   -<B1+B2+B3+B4>&377          ;THIS BYTE IS HEADER CHECKSUM

004655                       FLEND:
004655                       BUFF:           ;TEMPORARY SAVE AREA FOR OLD AREA
         001000                      .END    START
```

```
Symbol table

BACKSP= 000010          FLEND   004655          M010    003240          M042    003667          M074    004611
BCSR  = 177520          FMSG1   002430          M011    003242          M043    003706          NARGS = 000001
BDR   = 177524          FMSG1A  002465          M012    003242          M044    003723          NTYPE = 000027
BIT6  = 000100          FMSG1B  002500          M013    003242          M045    003744          OLDSIZ  002424
BIT7  = 000200          FMSG1C  002531          M014    003242          M046    004002          PCR   = 177522
BUFF    004655          FMSG1D  002554          M015    003242          M047    004026          PCRLB = 177522
B1    = 000002          FMSG2   002563          M016    003242          M050    004051          QUIT    002004
B2    = 000000          FMSG3   002644          M017    003242          M051    004071          QUIT1   002006
B3    = 000144          FMSG4   002737          M020    003242          M052    004117          REAROM  002314
B4    = 000143          LANG    001262          M021    003301          M053    004134          RETRY = 000002
CKSUM   004647          LF    = 000012          M022    003326          M054    004155          RMVTST= 173002
CR    = 000015          LNGHDR= 000140          M023    003354          M055    004204          ROMADR  002342
CRLF    002560          MAXERR= 000004          M024    003473          M056    004225          ROMSZ = 001551
DELAY = 025370          ME      004650          M025    003514          M057    004305          SPACE = 000040
DUMMY1  002526          MEND    004650          M026    003515          M060    004317          START   001000
DUMMY2  002551          MEND1   004646          M027    003535          M061    004317          TAB   = 000011
ENDBLK  004647          MOVROM  002134          M030    003541          M062    004332          TEXT    003104
ENDE2R= 166000          MSG000  003006          M031    003551          M063    004334          UFDHDR= 000040
ENGWRD  004646          MSG001  003045          M032    003563          M064    004347          UFDSIZ  002426
ESC   = 000033          M001    003201          M033    003565          M065    004400          WB      004646
EXIT    001566          M002    003211          M034    003635          M066    004403          WE      004650
EXIT1   001642          M003    003213          M035    003635          M067    004430          WEND    004647
E2LLB = 165006          M004    003221          M036    003636          M070    004510          WERR    002422
E2PAR = 165316          M005    003227          M037    003636          M071    004517          WRBYTE  002162
E2PROM= 165000          M006    003234          M040    003640          M072    004522          WRLANG  001462
E2WRIT  001666          M007    003236          M041    003667          M073    004610

. ABS.  004655    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: 00:00:16.26
OEEGBO.BIC,COEEGBO/CR/-SP=COEEGBO
```

```
COEEGBO    CREATED BY  MACRO  ON 20-JUN-85 AT 11:59      PAGE 1
                                                                                      SEQ 0024
SYMBOL CROSS REFERENCE                                   CREF   V02

SYMBOL  VALUE             REFERENCES

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      004655       6-370     6-409     6-413     6-441     6-518    #6-814
B1      = 000002      #6-802     6-807     6-811
B2      = 000000      #6-803     6-808     6-811
B3      = 000144      #6-804     6-809     6-811
B4      = 000143      #6-805     6-810     6-811
CKSUM     004647       6-428     6-497    *6-500    #6-792
CR      = 000015      #5-255     6-641     6-648     6-649     6-650     6-651     6-653     6-654     6-735
                       6-736     6-738     6-738     6-738     6-739     6-740     6-747     6-751     6-754
                       6-754     6-765     6-765     6-766     6-767     6-767     6-772     6-772     6-773
                       6-773     6-774     6-774     6-776     6-776     6-777     6-777     6-782     6-782
CRLF      002560       6-471     6-471    #6-648
DELAY   = 025370      #5-247     6-562
DUMMY1    002526       6-575     6-575    #6-642
DUMMY2    002551       6-579     6-579    #6-645
ENDBLK    004647       6-786    #6-787
ENDE2R  = 166000      #5-245     6-349     6-503     6-523     6-596
ENGWRD    004646      #6-786     6-786
ESC     = 000033      #5-262     6-768     6-769
EXIT      001566      #6-455     6-532     6-534
EXIT1     001642       6-460     6-465    #6-471     6-514     6-529
E2LLB   = 165006      #5-244     6-456
E2PAR   = 165316      #5-243     6-466
E2PROM  = 165000      #5-242     5-243     5-244     5-245     6-324     6-333     6-335     6-505     6-525
                       6-598     6-618     6-623
E2WRIT    001666       6-443     6-449    #6-477
FLEND     004655       5-264    #6-813
FMSG1     002430       6-573     6-573    #6-638
FMSG1A    002465      *6-570    #6-639
FMSG1B    002500       6-572     6-572    #6-640
FMSG1C    002531       6-576     6-576    #6-643
FMSG1D    002554       6-580     6-580    #6-646
FMSG2     002563       6-321     6-321    #6-649
FMSG3     002644       6-344     6-344    #6-650
FMSG4     002737       6-337     6-337    #6-651
LANG      001262       6-368    #6-380
LF      = 000012      #5-256     6-641     6-648     6-649     6-650     6-651
LNGHDR  = 000140      #5-248     6-380
MAXERR  = 000004      #5-252     6-481
ME        004650      #6-797
MEND      004650      #6-795     6-804     6-805
MEND1     004646       6-487     6-719    #6-783
MOVROM    002134       6-371     6-410     6-417    #6-547
MSG000    003006       6-512     6-512    #6-653
MSG001    003045       6-533     6-533    #6-654
M001      003201       6-484     6-659     6-660    #6-720
M002      003211       6-660     6-661    #6-721
M003      003213       6-661     6-662    #6-722
```

```
SYMBOL CROSS REFERENCE                            CREF   V02                                     SEQ 0025

SYMBOL  VALUE           REFERENCES

M004       003221        6-662      6-663      #6-723
M005       003227        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       003301        6-675      6-676      #6-736
M022       003326        6-676      6-677      #6-737
M023       003354        6-677      6-678      #6-738
M024       003473        6-678      6-679      #6-740
M025       003514        6-679      6-680      #6-741
M026       003515        6-680      6-681      #6-742
M027       003535        6-681      6-682      #6-743
M030       003541        6-682      6-683      #6-744
M031       003551        6-683      6-684      #6-745
M032       003563        6-684      6-685      #6-746
M033       003565        6-685      6-686      #6-747
M034       003635        6-686      6-687      #6-748
M035       003635        6-687      6-688      #6-749
M036       003636        6-688      6-689      #6-750
M037       003636        6-689      6-690      #6-751
M040       003640        6-690      6-691      #6-752
M041       003667        6-691      6-692      #6-753
M042       003667        6-692      6-693      #6-754
M043       003706        6-693      6-694      #6-755
M044       003723        6-694      6-695      #6-756
M045       003744        6-695      6-696      #6-757
M046       004002        6-696      6-697      #6-758
M047       004026        6-697      6-698      #6-759
M050       004051        6-698      6-699      #6-760
M051       004071        6-699      6-700      #6-761
M052       004117        6-700      6-701      #6-762
M053       004134        6-701      6-702      #6-763
M054       004155        6-702      6-703      #6-764
M055       004204        6-703      6-704      #6-765
M056       004225        6-704      6-705      #6-766
M057       004305        6-705      6-706      #6-768
M060       004317        6-706      6-707      #6-770
M061       004317        6-707      6-708      #6-771
M062       004332        6-708      6-709      #6-772
M063       004334        6-709      6-710      #6-773
M064       004347        6-710      6-711      #6-774
M065       004400        6-711      6-712      #6-775
M066       004403        6-712      6-713      #6-776
M067       004430        6-713      6-714      #6-777
```

```
COEEGBO    CREATED BY  MACRO  ON 20-JUN-85 AT 11:59     PAGE 3
                                                                                                   SEQ 0026
SYMBOL CROSS REFERENCE                              CREF    V02

SYMBOL  VALUE           REFERENCES

M070      004510     6-714     6-715    #6-778
M071      004517     6-715     6-716    #6-779
M072      004522     6-716     6-717    #6-780
M073      004610     6-717     6-718    #6-781
M074      004611     6-718     6-719    #6-782
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   = 001551    #5-264     6-437     6-447
SPACE   = 000040    #5-261     6-751     6-775
START     001000    #6-314     6-815
TAB     = 000011    #5-259     6-735     6-735     6-738     6-739     6-749     6-781
TEXT      003104     5-264     6-424     6-446    #6-659     6-659     6-804     6-805
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        004646    #6-785     6-802     6-803
WE        004650    #6-798     6-802     6-803
WEND      004647    #6-790
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
```