```
;
; This submit file will build the talking ethel firmware. It
; assumes a configuration of 4 double density diskettes labelled
; F0 to F3. In drive 0 should be the diskette labelled
; 'SYSTEM #0'. In drive 2 should be the diskette
; labeled 'SYSTEM #1' that is supplied with the transmittal package.
; In drive 3 should be placed the diskette labeled 'WORKING #1', also
; supplied in the transmittal package. Drive 1 should contain
; the diskette labeled 'SOURCE #1'. Stop now to verify this:

;
;  build MIP, Boot, etc.
;
consol /:f1:gen1.lst
submit :f1:gen1
copy :f1:*.obj to :f3: b
consol /:vo:
;
; now build KAOS and EDL
; insert the diskette labelled "'SOURCE #2 into Drive 1.

consol /:f1:gen2.lst
submit :f1:gen2
copy :f1:edl.obj to :f3: b
copy :f1:kaos.lib to :f3: b
consol /:vo:
;
; now build the power up confidence test and DLL
; insert the diskette labelled SOURCE #3 into Drive 1.

consol /:f1:gen3.lst
submit :f1:gen3
copy :f1:puconf.obj to :f3: b
copy :f1:dll.obj to :f3: b
consol /:vo:
;
; now link all of the firmware together and then locate it
;
consol /:f3:roml1.lst
submit :f3:roml1
;
; convert it to hex format
;
:f2:oh86 :f3:cbrom to :f3:cbrom.hex
consol /:vo:
;
; generation of the object module is now complete. To generate the
; Proms, see the installation instructions.
;
; now generate the external data link library. Place the diskette
; labeled 'Source #4' into drive 1 and the diskette labeled
; 'release #1' into drive 2.
;
; WARNING: THIS ROUTINE FORMATS THE DISKETTE IN DRIVE 2. MAKE SURE
;          THAT SYSTEM#1 DISKETTE HAS BEEN REMOVED.
```

```
consol /:f1:gen4.lst
submit :f1:gen4
consol /:vo:
;
; Now place the diskette labelled 'RELEASE #2' into drive 2. This
; diskette will be formatted and a copy of the firmware object
; and hex will be placed on it.

format :f2:ethel.rls
copy :f3:upm to :f2:
copy :f3:cbrom to :f2:
copy :f3:cbrom.hex to :f2:
copy :f3:promp1.csd to :f2:
;
; the generation of ethel firmware and software is now complete
;
```

```
LOC  OBJ                  LINE     SOURCE

                            51     ;
0000 ??                     52     CQMIPDEVCNT      DB      ?               ; NUMBER OF DEVICES KNOWN TO THIS ONE
0001 ??                     53     CQTHISDEVICE     DB      ?               ; DEV ID OF THIS DEVICE
0002 (8                     54     CQMIPIDSBASES    DW      8 DUP (?)       ; ORDER OF THESE THREE IS IMPORTANT
     ????
     )
0012 (126                   55     CQMIPDEVICEINFO  DB      126 DUP (?)     ; NINE DEVICES
     ??
     )
0090 (10                    56     CQMIPDEVTCENTRY  DB      10 DUP (?)
     ??
     )
                            57
                            58             PUBLIC  PORTTOMAILBOX,NOSUBSYSTEMS,SUBSYSTEMLIST
009A (24                    59     PORTTOMAILBOX    DW      24 DUP (?)
     ????
     )
00CA ??                     60     NOSUBSYSTEMS     DB      ?
00CB (5                     61     SUBSYSTEMLIST    DW      5 DUP (?)
     ????
     )
                            62
                            63             PUBLIC  SENDMSG,SENDRESULT,SENDSTATE,SENDDEVICE
                            64     ;
                            65     ; ORDER HERE IS CRITICAL
                            66     ;
00D5 ??                     67     SENDDEVICE       DB      ?
00D6 ??                     68     SENDSTATE        DB      ?
00D7 ??                     69     SENDRESULT       DB      ?
00D8 (2                     70     SENDMSG          DW      2 DUP (?)
     ????
     )
                            71
----                        72     DATA    ENDS
                            73
                            74     ;end CMXmipdefs;
                            75             END;

ASSEMBLY COMPLETE, NO ERRORS FOUND
-submit :f1:gen1
-delete :f1:*.obj,:f1:*.lst,:f1:*.lib,:f1:*.bak
 :F1:BOOTA.OBJ, DELETED
 :F1:BOOTB.OBJ, DELETED
 :F1:INIT.OBJ, DELETED
 :F1:TABLE.OBJ, DELETED
 :F1:OBJLST.OBJ, DELETED
 :F1:INTASK.OBJ, DELETED
 :F1:MIPSND.OBJ, DELETED
 :F1:RQPROC.OBJ, DELETED
 :F1:MIPDEF.OBJ, DELETED
 :F1:MIP.OBJ, DELETED
 :F1:BOOTA.LST, DELETED
 :F1:INIT.LST, DELETED
 :F1:BOOTB.LST, DELETED
 :F1:TABLE.LST, DELETED
 :F1:OBJLST.LST, DELETED
 :F1:INTASK.LST, DELETED
 :F1:MIPSND.LST, DELETED
 :F1:RQPROC.LST, DELETED
 :F1:MIPDEF.LST, DELETED
 :F1:GEN1.LST, CAN'T DELETE OPEN FILE
 :F1:*.LIB, NO SUCH FILE
```

```
-run
ISIS-II RUN 8086, V1.1
>;
>; built boot etc
>;
>:f2:asm86 :f1:boota.a86 debug
SERIES-III 8C86/8C87/8088 MACRO ASSEMBLER, V1.0

ASSEMBLY COMPLETE, NO ERRORS FOUND
>:f2:plm86 :f1:bootb.p86 optimize(3) compact debug rom

SERIES-III PL/M-86 COMPILER X19A
PL/M-86 COMPILATION COMPLETE.      0 WARNINGS,     0 ERRORS

>:f2:asm86 :f1:init.a86 debug
SERIES-III 8C86/8C87/8088 MACRO ASSEMBLER, V1.0

ASSEMBLY COMPLETE, NO ERRORS FOUND
>:f2:asm86 :f1:table.a86 debug
SERIES-III 8C86/8C87/8088 MACRO ASSEMBLER, V1.0

ASSEMBLY COMPLETE, NO ERRORS FOUND
>:f2:asm86 :f1:objlst.a86 debug
SERIES-III 8C86/8087/8088 MACRO ASSEMBLER, V1.0

ASSEMBLY COMPLETE, NO ERRORS FOUND
>;
>; build MIP
>;
>:f2:asm86 :f1:intask.a86 debug
SERIES-III 8C86/8087/8088 MACRO ASSEMBLER, V1.0

ASSEMBLY COMPLETE, NO ERRORS FOUND
>:f2:asm86 :f1:mipsnd.a86 debug
SERIES-III 8C86/8087/8088 MACRO ASSEMBLER, V1.0

ASSEMBLY COMPLETE, NO ERRORS FOUND
>:f2:asm86 :f1:rqproc.a86 debug
SERIES-III 8086/8087/8088 MACRO ASSEMBLER, V1.0

ASSEMBLY COMPLETE, NO ERRORS FOUND
>:f2:asm86 :f1:mipdef.a86 debug
SERIES-III 8086/8087/8088 MACRO ASSEMBLER, V1.0

ASSEMBLY COMPLETE, NO ERRORS FOUND
>;
>; link MIP together
>;
>:f2:link86 :f1:mipsnd.obj,:f1:intask.obj,:f1:rqproc.obj,:f1:mipdef.obj  &
>>            to :f1:mip.obj map print (:f1:mip.mp1) name(KAOS_MIP)
SERIES-III 8086 LINKER, V1.0
WARNING 12: UNRESOLVED SYMBOLS
>exit
-:F0:SUBMIT RESTORE :F1:GEN1.CS(:F3:GETHEL.CS,4,64)
-copy :f1:*.obj to :f3: b
COPIED :F1:BOOTA.OBJ TO :F3:BOOTA.OBJ
COPIED :F1:BOOTB.OBJ TO :F3:BOOTB.OBJ
COPIED :F1:INIT.OBJ TO :F3:INIT.OBJ
COPIED :F1:TABLE.OBJ TO :F3:TABLE.OBJ
COPIED :F1:OBJLST.OBJ TO :F3:OBJLST.OBJ
COPIED :F1:INTASK.OBJ TO :F3:INTASK.OBJ
COPIED :F1:MIPSND.OBJ TO :F3:MIPSND.OBJ
COPIED :F1:RQPROC.OBJ TO :F3:RQPROC.OBJ
COPIED :F1:MIPDEF.OBJ TO :F3:MIPDEF.OBJ
COPIED :F1:MIP.OBJ TO :F3:MIP.OBJ
```

```
-submit :f1:gen2
-DELETE :F1:KAOS.LIB,:f1:*.bak,:f1:*.lst,:f1:*.obj
 :F1:KAOS.LIB, DELETED
 :F1:*.BAK, NO SUCH FILE
 :F1:EDL.LST, DELETED
 :F1:SYSGEN.LST, DELETED
 :F1:NUCLUS.LST, DELETED
 :F1:CLOCK.LST, DELETED
 :F1:CREATE.LST, DELETED
 :F1:GEN2.LST, CAN'T DELETE OPEN FILE
 :F1:EDL.OBJ, DELETED
 :F1:SYSGEN.OBJ, DELETED
 :F1:NUCLUS.OBJ, DELETED
 :F1:CLOCK.OBJ, DELETED
 :F1:CREATE.OBJ, DELETED
-RUN
ISIS-II RUN 8086, V1.1
>:F2:ASM86 :F1:EDL.SRC DEBUG
SERIES-III 8086/8087/8088 MACRO ASSEMBLER, V1.0

ASSEMBLY COMPLETE, NO ERRORS FOUND
>:F2:ASM86 :F1:SYSGEN.SRC
SERIES-III 8086/8087/8088 MACRO ASSEMBLER, V1.0

ASSEMBLY COMPLETE, NO ERRORS FOUND
>:F2:ASM86 :F1:NUCLUS.SRC DEBUG
SERIES-III 8086/8087/8088 MACRO ASSEMBLER, V1.0

ASSEMBLY COMPLETE, NO ERRORS FOUND
>:F2:ASM86 :F1:CLOCK.SRC DEBUG
SERIES-III 8086/8087/8088 MACRO ASSEMBLER, V1.0

ASSEMBLY COMPLETE, NO ERRORS FOUND
>:F2:ASM86 :F1:CREATE.SRC DEBUG
SERIES-III 8086/8087/8088 MACRO ASSEMBLER, V1.0

ASSEMBLY COMPLETE, NO ERRORS FOUND
>:F2:LIB86
SERIES-III 8086 LIBRARIAN V1.0
*CREATE :F1:KAOS.LIB
*ADD :F1:NUCLUS.OBJ TO :F1:KAOS.LIB
*ADD :F1:CREATE.OBJ TO :F1:KAOS.LIB
*ADD :F1:CLOCK.OBJ TO :F1:KAOS.LIB
*EXIT
>EXIT
-:F0:SUBMIT RESTORE :F1:GEN2.CS(:F3:GETHEL.CS,5,105)
-copy :f1:edl.obj to :f3: b
COPIED :F1:EDL.OBJ TO :F3:EDL.OBJ
-copy :f1:kaos.lib to :f3: b
COPIED :F1:KAOS.LIB TO :F3:KAOS.LIB
-consol ,:vo:
```

```
delete :f1:*.bak,:f1:*.obj,:f1:*.lst
run
:f2:asm86 :f1:puconf.com debug
:F2:ASM86 :f1:dll.src debug
exit
```

```
-submit :f1:gen4
-;
-; generate the EDL80 library first
-;
-PLM80 :F1:CPYRT.SRC

ISIS-II PL/M-80 COMPILER V3.1
PL/M-80 COMPILATION COMPLETE.      0 PROGRAM ERROR(S)

-PLM80 :F1:VERS.SRC

ISIS-II PL/M-80 COMPILER V3.1
PL/M-80 COMPILATION COMPLETE.      0 PROGRAM ERROR(S)

-PLM80 :f1:mipdef.mod

ISIS-II PL/M-80 COMPILER V3.1
PL/M-80 COMPILATION COMPLETE.      0 PROGRAM ERROR(S)

-ASM80 :f1:intask.m80

ISIS-II 8080/8085 MACRO ASSEMBLER, V4.0

ASSEMBLY COMPLETE,   NO ERRORS
-ASM80 :f1:rqproc.m80

ISIS-II 8080/8085 MACRO ASSEMBLER, V4.0

ASSEMBLY COMPLETE,   NO ERRORS
-ASM80 :f1:mipsnd.m80

ISIS-II 8080/8085 MACRO ASSEMBLER, V4.0

ASSEMBLY COMPLETE,   NO ERRORS
-ASM80 :f1:miprcv.m80

ISIS-II 8080/8085 MACRO ASSEMBLER, V4.0

ASSEMBLY COMPLETE,   NO ERRORS
-LINK :f1:mipsnd.obj,:f1:miprcv.obj,:f1:intask.obj,:f1:rqproc.obj,  &
**       :f1:mipdef.obj to :f1:imip.obj
ISIS-II OBJECT LINKER V3.0
-PLM80 :F1:EDLLIB.SRC

ISIS-II PL/M-80 COMPILER V3.1
PL/M-80 COMPILATION COMPLETE.      0 PROGRAM ERROR(S)

-DELETE :F1:EDL80.LIB
 :F1:EDL80.LIB, DELETED
-LIB
ISIS-II LIBRARIAN V2.1
*CREATE :F1:EDL80.LIB
*ADD :F1:CPYRT.OBJ TO :F1:EDL80.LIB
*ADD :F1:VERS.OBJ TO :F1:EDL80.LIB
*ADD :F1:EDLLIB.OBJ TO :F1:EDL80.LIB
*ADD :F1:IMIP.OBJ TO :F1:EDL80.LIB
*EXIT
-FORMAT :F2:EDL80.LIB
NON-SYSTEM DISK
-COPY :F1:EDL80.LIB TO :F2:EDL80.LIB
COPIED :F1:EDL80.LIB TO :F2:EDL80.LIB
-;
-; now copy the examples to the release diskette.
```

```
-COPY :F1:GENEX.CSD TO :F2:
COPIED :F1:GENEX.CSD TO :F2:GENEX.CSD
-COPY :F1:EXAMPL.CSD TO :F2:
 :F1:EXAMPL.CSD, NO SUCH FILE
-COPY :F1:PRINSE.P80 TO :F2:
COPIED :F1:PRINSE.P80 TO :F2:PRINSE.P80
-COPY :F1:PRINIT.P80 TO :F2:
COPIED :F1:PRINIT.P80 TO :F2:PRINIT.P80
-COPY :F1:PRLIB.P80 TO :F2:
COPIED :F1:PRLIB.P80 TO :F2:PRLIB.P80
-COPY :F1:REPRIN.P80 TO :F2:
COPIED :F1:REPRIN.P80 TO :F2:REPRIN.P80
-COPY :F1:XMX.P80 TO :F2:
COPIED :F1:XMX.P80 TO :F2:XMX.P80
-;
-; now compile the examples and link them together.
-;
-SUBMIT :F1:GENEX
-;
-; This submit file will generate the print server examples described in
-; the iSBC 550 software users guide.
-;
-; first compile all modules
-;
-PLM80 :F1:PRINSE.P80

ISIS-II PL/M-80 COMPILER V3.1
PL/M-80 COMPILATION COMPLETE.      0 PROGRAM ERROR(S)

-PLM80 :F1:PRINIT.P80

ISIS-II PL/M-80 COMPILER V3.1
PL/M-80 COMPILATION COMPLETE.      0 PROGRAM ERROR(S)

-PLM80 :F1:PRLIB.P80

ISIS-II PL/M-80 COMPILER V3.1
PL/M-80 COMPILATION COMPLETE.      0 PROGRAM ERROR(S)

-PLM80 :F1:REPRIN.P80

ISIS-II PL/M-80 COMPILER V3.1
PL/M-80 COMPILATION COMPLETE.      0 PROGRAM ERROR(S)

-PLM80 :F1:XMX.P80

ISIS-II PL/M-80 COMPILER V3.1
PL/M-80 COMPILATION COMPLETE.      0 PROGRAM ERROR(S)

-;
-; now link and locate the print server
-;
-LINK :F1:PRINSE.OBJ, :F1:PRLIB.OBJ, :F1:PRINIT.OBJ, :F1:XMX.OBJ, &
**     PLM80.LIB, SYSTEM.LIB TO :F1:PRINSE.LNK
ISIS-II OBJECT LINKER V3.0
-LOCATE :F1:PRINSE.LNK
ISIS-II OBJECT LOCATER V3.0
-;
-; now link and locate the device asking for service
-;
-LINK :F1:REPRIN.OBJ, :F1:PRLIB.OBJ, :F1:PRINIT.OBJ, :F1:XMX.OBJ, &
**     PLM80.LIB, SYSTEM.LIB TO :F1:REPRIN.LNK
ISIS-II OBJECT LINKER V3.0
-LOCATE :F1:REPRIN.LNK
ISIS-II OBJECT LOCATER V3.0
```

```
-; generation complete
-;
-:FO:SUBMIT RESTORE :F1:GENEX.CS(:F1:GEN4.CS,7,13)
-;
-; put object and listings onto release diskette
-;
-COPY :F1:PRINSE TO :F2:
COPIED :F1:PRINSE TO :F2:PRINSE
-COPY :F1:REPRIN TO :F2:
COPIED :F1:REPRIN TO :F2:REPRIN
-COPY :F1:PRINSE.LST TO :F2:
COPIED :F1:PRINSE.LST TO :F2:PRINSE.LST
-COPY :F1:PRINIT.LST TO :F2:
COPIED :F1:PRINIT.LST TO :F2:PRINIT.LST
-COPY :F1:PRLIB.LST TO :F2:
COPIED :F1:PRLIB.LST TO :F2:PRLIB.LST
-COPY :F1:REPRIN.LST TO :F2:
COPIED :F1:REPRIN.LST TO :F2:REPRIN.LST
-COPY :F1:XMX.LST TO :F2:
COPIED :F1:XMX.LST TO :F2:XMX.LST
-;
-; ALL DONE
-:FO:SUBMIT RESTORE :F1:GEN4.CS(:F3:GETHEL.CSD,13,13)
-consol ,:vo:
```

```
:f%0:stfs
initialize :f%0:comm

v

debug = 1
test 5
exit
```

```
run
:f2:link86 :f3:boota.obj,:f3:bootb.obj,:f3:init.obj,:f3:mip.obj, &
           :f3:dll.obj, :f3:edl.obj, :f3:puconf.obj, &
           :f3:table.obj,:f3:kaos.lib,:f3:objlst.obj to :f3:cbrom.obj &
           print (:f3:cbrom.mp1) name(CB_ROM) map

:f2:loc86 :f3:cbrom.obj &
          ORDER (SM(data,stack,stacks,buffers,entrypts,code)) &
          ADDRESSES (GROUPS (dgroup(CF000OH), cgroup(CF00C0H) ),&
          SEGMENTS (entrypts(0FEC00H), data(0F0040H), buffers(0F8000H) )) &
          SEGSIZE(stack(C)) &
          NOINITCODE SC(3)  RESERVE (0 TO 0F003FH) START(bootstartentry) &
          bootstrap print(:f3:cbrom.mp2)
exit
:F2:CH86 :F3:CBROM TO :F3:CBROM.HEX
```

Prompt.cso ?

```
;
; procedure to program initial proms for talking ethel firmware
;
; drive 1 should contain 'WORKING #1'
;
:f1:LPM
2732
socket = 2
offset
;
; now enter the monitor and fill up the area starting at the
; number returned from 'offset'. The command is:
;       'Fx,y,FF'   where x = the offset, y = x+2000H
;
read 86hex file :f1:cbrom.hex into 2000h
;
; insert the low order prom (U59) into socket 2
;

program from 0 to 0fffh start 0
;
; insert the high order prom (U58) into socket 2
;

program from 1000h to 1fffh start 0
;
; the programming of the initial set of proms is complete. Information
; on the next step in contained in file 'INSTAL.HLP'
;

exit
```