10.0 Appendices

Appendix A - Assembly Language Program to read data from the parallel port into memory and then store it on disk.

```
INCDIR  "DH1:DEVPAC/INCLUDE/"       ;include file directory
INCLUDE "hardware/cia.i"            ;
INCLUDE "hardware/custom.i"         ;include files
INCLUDE "exec/exec_lib.i"           ;
INCLUDE "libraries/dos_lib.i"       ;
OPT  P+                             ;check for position
                                    ;independent code
_ciaa     equ  $bfe001              ;base address of
_ciab     equ  $bfd000              ;8520 chips and
_custom   equ  $dff000              ;custom chips
bufsize   equ  255600               ;size of buffer
port      equ  $bfe101              ;parallel port address
ctrl      equ  $bfe301              ;control register address

start:    movem.l   a0-a6,-(sp)     ;save all registers
          movem.l   d0-d7,-(sp)     ;on the stack
          lea  dosname(pc),a1       ;open DOS library
          moveq     #0,d0
          CALLEXEC OpenLibrary
          tst.l     d0
          beq  error
          lea  _DOSBase(pc),a0      ;store DOS library's
          move.l    d0,(a0)         ;base address
          lea  window_def(pc),a0
```

```
        move.l    a0,d1           ;open a window
        move.l    #1005,d2
        CALLDOS   Open
        tst.l     d0
        beq  error
        lea  conhandle(pc),a0     ;store window's i.d.
        move.l    d0,(a0)
        move.l    #bufsize,d0     ;allocate   memory   for
        move      #0,d1           ;buffer
        CALLEXEC AllocMem
        lea  buffer_address(pc),a0
        move.l    d0,(a0)         ;store buffer address
        move.b    #0,(ctrl)       ;set port for input
        lea _custom,a3            ;pointers to custom
        lea _ciab,a4              ;chips and 8520 chip in
        move.b ciacra(a4),d0      ;registers
        and.b #%11000000,d0       ;set up timer in
        or.b #%00001000,d0        ;one-shot mode
        move.b d0,ciacra(a4)      ;
        move.b #%01111111,ciaicr(a4)  ;clear all interrupts
        move.b #$31,ciatalo(a4)   ;561 counts =$0231
        move.b #$02,ciatahi(a4)
        bsr  print_time           ;print start time in window
        lea  buffer_address(pc),a0
        move.l    (a0),a1         ;buffer address in a1
        move.l    a1,a2
        add.l     #bufsize,a2     ;end of buffer address in a2
next:   btst.b    #0,ciaicr(a4)   ;wait for count zero
```

```
        beq.s     next              ;
        bset.b    #0,ciacra(a4)     ;restart timer
        move.b    (port),d1         ;get data from port
        move.b    d1,(a1)+          ;store data in buffer
        cmp.l     a1,a2             ;check for end of buffer
        bne  next
        bsr  print_time             ;print end time in window
save:   move.l    #1006,d2          ;new file mode
        lea  filename(pc),a0
        move.l    a0,d1             ;pointer to filename
        CALLDOS   Open              ;open file
        lea  filehd(pc),a0
        move.l    d0,(a0)
        move.l    d0,d1
        lea  buffer_address(pc),a0
        move.l    (a0),d2
        move.l    #bufsize,d3
        CALLDOS Write               ;write buffer to disk
        lea  filehd(pc),a0
        move.l    (a0),d1
        CALLDOS   Close             ;close file
mouse:  btst #6,($bfe001)           ;check for mouse button
        bne  mouse
        lea  conhandle(pc),a0
        move.l    (a0),d1
        CALLDOS   Close             ;close window
        move.l    #bufsize,d0
        lea  buffer_address(pc),a0
```

```
            move.l    (a0),a1
            CALLEXEC FreeMem          ;deallocate buffer memory
            move.l    #0,d0
            movem.l   (sp)+,d0-d7     ;get registers from stack
            movem.l   (sp)+,a0-a6
            rts                       ;return
            error:                    ;error routine
            movem.l   (sp)+,d0-d7     ;get registers from stack
            movem.l   (sp)+,a0-a6
            rts                       ;terminate program
print_time:
            lea  command(pc),a0
            move.l    a0,d1           ;time subroutine
            clr.l     d2
            lea  conhandle(pc),a0     ;calls the AmigaDOS date
            move.l    (a0),d3         ;command to print the date
            CALLDOS   Execute         ;and time
            rts
dosname:       dc.b "dos.library",0,0   ;data and storage
               even                     ;areas are defined here
_DOSBase:           ds.l 1
buffer_address:     ds.l 1
filehd:             ds.l 1
conhandle:          ds.l 1
filename:           dc.b "dh1:images/Temp",0,0
command:            dc.b "ram:date",0
window_def:         dc.b "CON:20/20/300/150/Meteosat Data",0
                    end
```