UNITED STATES DEPARTMENT OF THE INTERIOR

GEOLOGICAL SURVEY

OCEAN BOTTOM INSTRUMENT PACKAGE (OBIP) SOFTWARE: MODIFICATION II

by

U. S. Geological Survey[1]

Open-file Report 84-842

tw and

This report is preliminary and has not been reviewed for conformity with U.S. Geological Survey editorial standards. Use of tradenames is for purposes of identification only and does not constitute any endorsement by the USGS.

1. Woods Hole, Mass.

1984

Ocean Bottom Instrument Package (OBIP) software: Modification II

This open-file publication is a listing of the modified software program for the U.S. Geological Survey's Ocean Bottom Instrument Package (OBIP). This listing supercedes the listing contained in Open-File Report 84-267: Software Listings and Major Electronic Circuits and Components of the Ocean Bottom Instrument Package (OBIP).

```
; ---- ASSEMBLY CONTROL ----
FALSE   EQU     0               ;
TRUE    EQU     NOT FALSE       ;
;
;
; ---- WINDOW CONSTANTS ----
BIT0    EQU     01H             ; \
BIT1    EQU     02H             ;  \
BIT2    EQU     04H             ;   \
BIT3    EQU     08H             ;    \ Use as values for
BIT4    EQU     10H             ;    / bit masks
BIT5    EQU     20H             ;   /
BIT6    EQU     40H             ;  /
BIT7    EQU     80H             ; /
;
RECLEN  EQU     2000H           ; Physical rec len (excl hdr)
MAXB    EQU     50H             ; MAXIMUM INPUT BUFFER SIZE
RSTCC   EQU     4               ; CARTRIDGE CONTROLLER RESET LINE ON PORT B
                                ; CLEAR TO RESET,SET TO ALLOW OPERATION
;
;       ASCII EQUATES
CR      EQU     0DH             ; Carriage return
LF      EQU     0AH             ; Line feed
BS      EQU     08H             ; BACKSPACE
DEL     EQU     7FH             ; DELETE
SPC     EQU     20H             ; SPACE
CTLX    EQU     18H             ;CTLX
CTLU    EQU     15H             ;CTLU
CTLZ    EQU     1AH             ;CTLZ (STRING TERMINATOR)
;
;
; ---- PORT ASSIGNMENTS ----
ANAPRT  EQU     010H            ; Analog board STA & Threshold
AVGPRT  EQU     ANAPRT+1        ; Analog board Feedback control
ADPORT  EQU     018H            ; A-D board base address
FMPORT  EQU     28H             ; FM board base address
INTPRT  EQU     0BBH            ; NSC-800 interrupt mask port

        AD_INT_EN       EQU     8       ;RSTA FROM A/D
        T1_INT_EN       EQU     4       ;TIMER 1 INTERUPT ENABLE
        RSTC_EN         EQU     2       ;RSTC NOT USED
        EARLY_TERM      EQU     1       ;EARLY TERMINATOR BUTTON ENABLE

CPORT   EQU     0F0H            ; Cartridge controller base
MA      EQU     CPORT           ; Mode argument
PA      EQU     CPORT+1         ; Position argument
CA      EQU     CPORT+2         ; Command argument
DA      EQU     CPORT+3         ; Data argument,WRITE DATA ONLY
PS      EQU     CPORT+1         ; Port status
DSIS    EQU     CA              ; DRIVE/INTERFACE STATUS PORT
PWRPRT  EQU     0FFH            ; Power Interface board PORT
;
```

```
; ---- MEMORY MAPPED I/O ----
TMEM    EQU     2000H           ; RTC base address
CTEST   EQU     TMEM            ; TEST ONLY
SECX.1  EQU     TMEM+1          ; TENTHS OF SECONDS
SEC     EQU     TMEM+2          ; UNITS OF SECONDS
TENSEC  EQU     TMEM+3          ; TENS OF SECONDS
MIN     EQU     TMEM+4          ; UNITS OF MINUTES
TENMIN  EQU     TMEM+5          ; TENS OF MINUTES
HRS     EQU     TMEM+6          ; UNITS OF HRS.
TENHRS  EQU     TMEM+7          ; TENS OF HRS.
DAYS    EQU     TMEM+8          ; UNIT DAYS
TENDAY  EQU     TMEM+9          ; TENS OF DAYS
DAYWK   EQU     TMEM+10         ; DAY OF WEEK
MTH     EQU     TMEM+11         ; UNIT MONTHS
TENMTH  EQU     TMEM+12         ; TENS OF MONTHS
YRS     EQU     TMEM+13         ; YR STATUS REGISTER (WRITE ONLY)
STPST   EQU     TMEM+14         ; STOP START REGISTER
INTSTAT EQU     TMEM+15         ; CLOCK INTERUPT STATUS REGISTER
;END CLOCK ADDRESS EQUATES
;*************************************************************************
;NSC810 ADDRESS EQUATES
NSCIOT  EQU     3080H           ; NSC-810 I/O-Timer base
PBDATA  EQU     NSCIOT+01H      ; Port B data reg
PCDATA  EQU     NSCIOT+02H      ; Port C data reg
PBDDIR  EQU     NSCIOT+05H      ; Port B data direction reg
PCDDIR  EQU     NSCIOT+06H      ; Port C data direction reg
NSCMDR  EQU     NSCIOT+07H      ; Mode definition reg
PBCLRB  EQU     NSCIOT+09H      ; Port B bit clear reg
PCCLRB  EQU     NSCIOT+0AH      ; Port C bit clear reg
PBSETB  EQU     NSCIOT+0DH      ; Port B bit set reg
PCSETB  EQU     NSCIOT+0EH      ; Port C bit set reg
TMR0    EQU     NSCIOT+10H      ;TIMER 0 REGS
TMR0LS  EQU     NSCIOT+10H      ; Timer 0 lsb
TMR0MS  EQU     NSCIOT+11H      ; Timer 0 msb
TMR1    EQU     NSCIOT+12H      ;TIMER 1 REGS
TMR1LS  EQU     NSCIOT+12H      ; Timer 1 lsb
TMR1MS  EQU     NSCIOT+13H      ; Timer 1 msb
STOPT0  EQU     NSCIOT+14H      ; Timer 0 stop
STRT0   EQU     NSCIOT+15H      ; Timer 0 start
STOPT1  EQU     NSCIOT+16H      ; Timer 1 stop
STRT1   EQU     NSCIOT+17H      ; Timer 1 start
CMDT0   EQU     NSCIOT+18H      ; Timer 0 command reg
CMDT1   EQU     NSCIOT+19H      ; Timer 1 command reg
;
;
;       MEMORY EQUATES
;
NSCRAM  EQU     NSCIOT-80H      ; NSC-810 ram area
BUFMEM  EQU     8000H           ; 32K aquisition buffer
SCRATCH EQU     4000H           ; SCRATCH PAD RAM
DBGRAM  EQU     0A000H          ; DEBUG RAM
HDRRAM  EQU     0E000H          ; RAM FOR STORING ASCII INPUT PRIOR TO START
;
```

```
;SPECIAL EQUATES
XMRZ    EQU     DBGRAM+3H       ;
DBPXM   EQU     DBGRAM+06H
;*************************************************************************
;       RAMTEST VARIABLES(ONLY USED DURING RAMTEST AND BEFORE)
STACK   EQU     NSCRAM+50H      ; Run time stack CHANGE TO SYSRAM+1000H
ADDRESS EQU     NSCRAM+5AH      ; [1] STORAGE FOR BAD RAM ADDRESS
RAMST   EQU     NSCRAM+5BH      ; [2] START OF RAM TO TEST
PATRN   EQU     NSCRAM+5DH      ; [1] RAM TEST PATTERN
ENDRAM  EQU     NSCRAM+5EH      ; [2] END OF RAM TO BE TESTED
;*************************************************************************
;       SERIES/EXPERIMENT VARIABLES
;
IO      EQU     NSCRAM          ;BASE IO PORT FOR AD(1 LESS THAN FIRST)
NCX2    EQU     NSCRAM+1        ;# OF CHANNELS * 2 (B REG)
NCX2IO  EQU     IO              ;COMBINED FOR INI INSTRUCTION
SERTYP  EQU     NCX2IO+2        ;SERIES TYPE
EXPN    EQU     SERTYP+1        ;NO. OF EXPERIMENTS IN SERIES
STRTTB  EQU     EXPN+1          ;START TIME FOR SERIES
STOPTB  EQU     STRTTB+5        ;STOP TIME FOR SERIES
BUFSIZ  EQU     STOPTB+5        ;BUFFER SIZE IN 8K BLOCKS(NO. OF RECORDS)
PESAMPS EQU     BUFSIZ+1        ;POST EVENT SAMPLES
BSTART  EQU     PESAMPS+2       ;HIGH ORDER BUFFER START ADDRESS
MSAMPS  EQU     BSTART+1        ;# OF SAMPLES(MAX)
OFFSET  EQU     MSAMPS+2        ;WINDOW OFFSET IN SECONDS
PERIOD  EQU     OFFSET+1        ;WINDOW PERIOD IN MINUTES
ADVAL   EQU     PERIOD+1        ;VALUE FOR A/D PORT(SAMPLE RATE CODE)
ANVAL   EQU     ADVAL+1         ;VALUE FOR ANALOG PORT(STA&THRSH)

;END OF SERIES PARAMETERS,START WORKING PARAMETERS

ENDPARA EQU     ANVAL+1         ;THIS MARKS END OF SERIES PARAMS
NOPARA  EQU     LOW(ENDPARA-NSCRAM)     ;AMOUNT OF PARAMETER STORAGE
PESAMS  EQU     ENDPARA         ;POST EVENT FOR VERIFY
THRSHSV EQU     PESAMS+1        ;THRESHOLD FOR VERIFY
STASAV  EQU     THRSHSV+1       ;SHORT TERM AVERAGE FOR VERIFY
SRATE   EQU     STASAV+1        ;SAMPLING RATE FOR VERIFY
HDRBUF  EQU     SRATE+1         ;POINTER TO ASCII HEADER
GPCTRL  EQU     HDRBUF+2        ;GEN PURPOSE BCD CTR LOW 2 DIGITS
GPCTRH  EQU     GPCTRL+1        ;GEN PURPOSE BCD CTR HIGH 2 DIGITS
NSAMPS  EQU     GPCTRH+1        ;# OF SAMPLES(WORKING)
BUFPTR  EQU     NSAMPS+2        ;DATA ACQUISITION BUFFER POINTER
BSZSAV  EQU     BUFPTR+2        ;BUFF. SIZE FOR VERIFY
PARBUF  EQU     BSZSAV+1        ;STORAGE FOR DE DURING PARAM ENTRY&MOVE
WBUFSAV EQU     PARBUF+2        ;STORAGE FOR HL DURING WRITE TO TAPE
WBSTART EQU     WBUFSAV+2       ;HIGH ORDER BUFFER START ADDRESS FOR WRITE
RDBUF   EQU     WBSTART+1       ;BUFFER STORAGE FOR READ POINTER(2)
LSTCP   EQU     RDBUF+2         ;LAST VALID COMP. FOR COMPARE ROUTINE(2)(DE)
LSN     EQU     LSTCP+2         ;1BYTE,LAST SERIES# +1
TIMSAVE EQU     LSN+1           ;10 BYTES TO SAVE IN CASE OF ENTRY ERROR
```

```
;               SCRATCH PAD RAM EQUATES
NSTACK  EQU     SCRATCH+200H    ; OPERATING STACK AFTER RAM TEST
TSTORE  EQU     NSTACK+2        ; TEMPORARY STORAGE
TSTFLG  EQU     TSTORE+2        ; FLAG FOR RAM TEST
RECTIME EQU     TSTFLG+2        ; LENGTH OF RECORDING,MINUTES,SECONDS
SHIFTER EQU     RECTIME+2       ; USED TO DIVIDE TO GET ABOVE VALUE
ESECT   EQU     SHIFTER+2       ; SAVE SECTOR COUNT FOR WRITE ERROR
EBUFSAV EQU     ESECT+1         ; SAVE BUFFER PTR FOR WRITE ERROR
T1INT   EQU     EBUFSAV+2       ; T1 INTERUPT VECTOR
NDUM    EQU     T1INT+2         ; DUMMY END
DS      EQU     NDUM            ; [1] stored drive status
IS      EQU     DS+1            ; [1] stored interface status
IMA     EQU     DS+2            ; [1] mode arg copy
IPA     EQU     DS+3            ; [1] pos. arg copy
ICA     EQU     DS+4            ; [1] cmd. arg copy
GCODE   EQU     DS+5            ; [1] analog board gain code
SRCODE  EQU     DS+6            ; [1] Sample rate value for A-D board
TDBUF   EQU     TZBUF+51H       ;
PDIGE   EQU     CLRSCR-9CH      ;

ELAPSED_MIN     EQU     SCRATCH+300H    ;ELAPSED MINUTE COUNT

BASAD   EQU     SCRATCH+350H    ;BASE ADDRESS
XAG     EQU     BASAD+22H       ;DIAGN. ADD.
IBUFF   EQU     SCRATCH+400H    ; ASCII INPUT BUFFER
NCI     EQU     IBUFF+MAXB+1    ; NO. OF CHARACTERS IN BUFFER
SERBUF  EQU     SCRATCH+1000H   ;BEGINNING OF SERIES PARAM. STORAGE
ESERBUF EQU     SERBUF+9*NOPARA ;END OF SERIES PARAMETER STORAGE
EXPBUF  EQU     ESERBUF+1       ;BEGINNING OF EXPERIMENT BUFFER
SERPTR  EQU     EXPBUF          ;POINTER TO NEXT SERIES PARAMS
CSN     EQU     SERPTR+2        ;CURRENT SERIES NO.
CEXPN   EQU     CSN+1           ;CURRENT EXP# IN BCD
SAMPTIM EQU     CEXPN+1         ;TIME OF BEGINNING OF RECORD(15 BYTES ALLOWED)
CSECX1  EQU     SAMPTIM                 ;TENTHS OF SECONDS
CSEC    EQU     SAMPTIM+1               ;SECONDS
CTENSEC EQU     SAMPTIM+2               ; TENS OF SECONDS
CMIN    EQU     SAMPTIM+3               ; UNITS OF MINUTES
CTENMIN EQU     SAMPTIM+4               ; TENS OF MINUTES
CHRS    EQU     SAMPTIM+5               ; UNITS OF HRS.
CTENHRS EQU     SAMPTIM+6               ; TENS OF HRS.
CDAYS   EQU     SAMPTIM+7               ; UNIT DAYS
CTENDAY EQU     SAMPTIM+8               ; TENS OF DAYS
CDAYWK  EQU     SAMPTIM+9               ; DAY OF WEEK
CMTH    EQU     SAMPTIM+10              ; UNIT MONTHS
CTENMTH EQU     SAMPTIM+11              ; TENS OF MONTHS
CYR     EQU     SAMPTIM+12              ;YEAR(PACKED BCD)
CTIMER  EQU     SAMPTIM+13              ;TIMER (LO BYTE FIRST,2 BYTES)
CTIMLO  EQU     CTIMER                  ;LO BYTE OF TIMER
CTIMHI  EQU     SAMPTIM+14              ;HI BYTE OF TIMER
;
;EACH TIMER COUNT(TIMER IN HEX) REPRESENTS 1/16,384.375 SEC(ABOUT .061 MS)
DATABLOCK       EQU     SAMPTIM+15      ;NO. OF SECTORS TO WRITE(128 BYTES)
EEXPBUF EQU     DATABLOCK+1             ;END OF TAPE DATA IF 0FFH
```

```
; ---- MEMORY IMAGE OF CLOCK IO ----
MTMEM   EQU     5200H           ; RTC base address
MCTEST  EQU     MTMEM           ; TEST ONLY
MSECX.1 EQU     MTMEM+1         ; TENTHS OF SECONDS
MSEC    EQU     MTMEM+2         ; UNITS OF SECONDS
MTENSEC EQU     MTMEM+3         ; TENS OF SECONDS
MMIN    EQU     MTMEM+4         ; UNITS OF MINUTES
MTENMIN EQU     MTMEM+5         ; TENS OF MINUTES
MHRS    EQU     MTMEM+6         ; UNITS OF HRS.
MTENHRS EQU     MTMEM+7         ; TENS OF HRS.
MDAYS   EQU     MTMEM+8         ; UNIT DAYS
MTENDAY EQU     MTMEM+9         ; TENS OF DAYS
MDAYWK  EQU     MTMEM+10                ; DAY OF WEEK
MMTH    EQU     MTMEM+11                ; UNIT MONTHS
MTENMTH EQU     MTMEM+12                ; TENS OF MONTHS
MYRS    EQU     MTMEM+13                ; YR STATUS REGISTER (WRITE ONLY)
MSTPST  EQU     MTMEM+14                ; STOP START REGISTER
MINTSTAT        EQU     MTMEM+15                ; CLOCK INTERUPT STATUS REGISTER
;END CLOCK ADDRESS EQUATES
TIMEN   EQU     MINTSTAT+1              ; 5 BYTES FOR CLOCK IN PBCD
OTENMTH EQU     TIMEN+5
;*****************************************************************************
```

```
;TAPE DRIVE EQUATES
READC           EQU     1
WRITEC          EQU     2
WRITEFMC        EQU     3
FWDSPCRECC      EQU     4
FWDSPCFILC      EQU     5
REVSPCRECC      EQU     6
REVSPCFILC      EQU     7
CURSTATC        EQU     8
SETRECLENC      EQU     9
WRITEWCHC       EQU     10
RECSERMASC      EQU     11
REWINDC         EQU     40H
RWREADC         EQU     81H             ;RAM READ
BREADC          EQU     41H
BWRITEC         EQU     42H
BWRITEFMC       EQU     43H
;
;MASKS
MAN             EQU     60H     ;NORMAL MA EXCLUDING TRACK BITS(0&1)
TRACKMA         EQU     3       ;MASK FOR TRACK
MSMA            EQU     0E0H    ;MASK SEARCH MA EXCL. TRACK BITS
COMSTATM        EQU     30H     ;COMMAND STATUS MASK FOR IS CHECK
EOTM            EQU     4       ;EOT MASK FOR DS
BOTM            EQU     8       ;BOT MASK FOR DS
;
BUF16   EQU     5300H
B16US   EQU     BUF16
B16NAME EQU     BUF16+1
B16EXT  EQU     BUF16+9
B16LB   EQU     BUF16+13
B16RC   EQU     BUF16+15
BSERNO  EQU     B16NAME+1
BEXPNH  EQU     B16NAME+6
BEXPNL  EQU     B16NAME+7
;
TIPBUF  EQU     5310H
;
;*************************************************************************
```

```
*******************************************************************************

                              MEMORY MAP

HEX ADDRESS

0000
          SYSTEM PROM
1FFF

2000
          REAL TIME CLOCK
200F

2010
          NOTHING
2FFF


3000
          NSC810 RAM
307F


3080
          NSC810 I/O
30??


4000
          SCRATCH PAD RAM
5FFF


6000
          NOTHING
7FFF


8000
        ACQUISITION BUFFER RAM
FFFF
)
```

```
;LAST UPDATE 08/13/84
;*************************************************************************
                          _______________________
                         |                       |
FROM SERIAL IN,PB7 )----|1 PC3/TG         VCC 40|---(+5 VOLTS
DIV.CLK,U14,P1,4040,Q12)|2 PC4/T1 IN  PC2/NSTB 39|--->U26,P12/ALT.ADDR.(SETHI OR XX CHIP)
SYSTEM CLOCK)-----------|3 T0 IN        PC1/BF 38|--->PWR SAVE DIS. IF LOW
RESET OUT,NSC800 )------|4 RESET      PC0/NINT 37|--->DIS. NRSTA IF HI,U27,P1
NRSTB & 8 INPUT NAND(---|5 PC5/T1 OUT      PB7 36|---(SERIAL IN,NRSTC,8 INPUT NAND
PB6 & 8 INPUT NAND (----|6 T0 OUT          PB6 35|---(T0 OUT
A7 )--------------------|7 IOT/NM          PB5 34|---(NRSTA
MEMORY DECODE LOGIC )---|8 CE              PB4 33|---(NINTR
NRD NSC 800 )-----------|9 NRD             PB3 32|---(SYNCHR. IN(CONNECT)
NWR NSC 800 )-----------|10 NWR  NSC 810   PB2 31|--->S-100,P65,N CARTRIDGE RESET
ALE NSC 800 )-----------|11 ALE            PB1 30|--->
                   /---|12 AD0             PB0 29|--->SERIAL OUT
                  / ---|13 AD1             PA7 28|---\
                 /  ---|14 AD2             PA6 27|--- \
ADDRESS/DATA    /   ---|15 AD3             PA5 26|---  \
BUS NSC800      \   ---|16 AD4             PA4 25|---   \    NOT USED
                 \  ---|17 AD5             PA3 24|---   /    PRESENTLY
                  \ ---|18 AD6             PA2 23|---  /
                   \---|19 AD7             PA1 22|--- /
GRND )-----------------|20 VSS             PA0 21|---/
                       |_______________________|
;
;*************************************************************************
;                         NSC 810 EQUATES
;
;*************************************************************************
OUTP    EQU     01H             ; DEFINE AS OUTPUT
INP     EQU     00H             ; DEFINE AS INPUT
HI      EQU     01H             ; DEFINE AS HIGH
LO      EQU     00H             ; DEFINE AS LOW
;*************************************************************************
;               PORT A DIRECTION
;
PA0     EQU     INP             ;
PA1     EQU     OUTP SHL 1      ; SECONDS PULSE OUT
PA1D    EQU     LO SHL 1
PA2     EQU     OUTP SHL 2      ; MINUTES PULSE OUT
PA2D    EQU     LO SHL 2
PA3     EQU     INP SHL 3       ;
PA4     EQU     INP SHL 4
PA5     EQU     INP SHL 5
PA6     EQU     INP SHL 6       ;SATELLITE CLOCK IN
PA7     EQU     INP SHL 7
ADIR    EQU     PA0 OR PA1 OR PA2 OR PA3 OR PA4 OR PA5 OR PA6 OR PA7
                                ; PORT A DIRECTION
ADAT    EQU     PA1D OR PA2D    ; PORT A DATA
;
```

```
;*****************************************************************************
;               PORT B DIRECTION
;
PB0     EQU     OUTP            ;SERIAL OUT,INVERTED
PB0D    EQU     HI              ;SPACE
PB1     EQU     OUTP SHL 1      ;FORMERLY USED FOR REEL-TO-REEL
PB1D    EQU     HI SHL 1        ;FORMER VALUE
PB2     EQU     OUTP SHL 2      ;NOT CARTRIDGE RESET
PB2D    EQU     LO SHL 2        ;RESET CARTRIDGE
PB3     EQU     INP SHL 3       ;TIME SYNC IN(MUST BE CONNECTED)
PB4     EQU     INP SHL 4       ;FROM NOT INTR
PB5     EQU     INP SHL 5       ;FROM NOT RSTA
PB6     EQU     INP SHL 6       ;FROM T0 OUT
PB7     EQU     INP SHL 7       ;SERIAL IN,INVERTED(ALSO CONNECTED TO NRSTC)
;
BDAT    EQU     PB0D OR PB1D OR PB2D    ;PORT B DATA WORD
BDIR    EQU     PB0 OR PB1 OR PB2 OR PB3 OR PB4 OR PB5 OR PB6 OR PB7
                                ; PORT B DIRECTION
;*****************************************************************************
;WARNING: IF TIMER1 IS USED,OR TIMER0 OPERATED IN MODES 2,3,OR 4,OR ANYTHING
;OTHER THAN MODE0 SELECTED FOR NSC810,PORT C PINS ARE UNSAFE FOR GENERAL
;PURPOSE IO.IF POSSIBLE THEY SHOULD NOT BE USED.
;*****************************************************************************
;               PORT C DIRECTION
;
PC0     EQU     OUTP            ;NOT RSTA DISABLE
PC0D    EQU     HI              ;DISABLE RSTA INITIALLY(WILL CHANGE)
PC1     EQU     OUTP SHL 1      ;DISABLE POWER SAVE
PC1D    EQU     LO SHL 1        ;DISABLE POWER SAVE INITIALLY
PC2     EQU     OUTP SHL 2      ;REMAP!? ADDRESSES!?
PC2D    EQU     HI SHL 2        ;DO NOT REMAP
PC3     EQU     INP SHL 3       ;TIMER GATE,CONNECTED TO SERIAL IN AND NRSTC
PC4     EQU     INP SHL 4       ;TIMER 1 INPUT,CONNECTED TO 4040
PC5     EQU     OUTP SHL 5      ;TIMER 1 OUTPUT,CONNECTED TO NRSTB AND NAND
PC5D    EQU     HI SHL 5        ;DON'T MESS UP RSTB
;
CDAT    EQU     PC0D OR PC1D OR PC2D OR PC5D
                                ;PORT C DATA
CDIR    EQU     PC0 OR  PC1 OR PC2 OR PC3 OR PC4 OR PC5
                                ; PORT C DIRECTION
;*****************************************************************************
;               NSC 810 MODES
;
MODE0   EQU     00H             ; PORT A BASIC IO
MODE1   EQU     01H             ; STROBED INPUT MODE(AFFECTS A AND C)
MODE2   EQU     03H             ; STROBED OUTPUT MODE """"""""""""""""
MODE3   EQU     07H             ; TRISTATE STROBED OUTPUT """""""""""
```

```
;*************************************************************************
;               TIMER MODES
;
TMODE0  EQU     00H             ; KILL TIMER
TMODE1  EQU     01H             ; EVENT COUNTER MODE
TMODE2  EQU     02H             ; STOPWATCH EVENT TIMER
TMODE3  EQU     03H             ; EVENT TIMER WITH RESET
TMODE4  EQU     04H             ; ONE SHOT
TMODE5  EQU     05H             ; SQUARE WAVE
TMODE6  EQU     06H             ; PULSE GENERATOR
;*************************************************************************
;               TIMER PRESCALER
;
PRE1    EQU     0H              ; NO PRESCALE
PRE2    EQU     8H              ; DIVIDE BY 2
PRE64   EQU     18H             ;DIVIDE BY 64,TIMER 0 ONLY(SEE NSC810 COOKBOOK)
;*************************************************************************
;               TIMER READ/WRITE MODE
;
BIT8T   EQU     20H             ; SINGLE BYTE READ/WRITE MODE(HI OR LOW)
BIT16T  EQU     0H              ; 16 BIT TIMER. READ OR WRITE LOW BYTE FIRST
;*************************************************************************
;               TIMER GATE CONTROL
;
GPOLH   EQU     0H                      ; GATE INPUT ACTIVE HIGH(PC3).THIS IS THE
                                        ; COMMON GATE FOR BOTH TIMERS.
GPOLL   EQU     40H                     ; GATE INPUT ACTIVE LOW
;*************************************************************************
;               TIMER OUTPUT CONTROL
;
;
OUTPOLH EQU     80H             ; TIMER OUTPUT ACTIVE HIGH. T1=PC5, T0=PIN 6
OUTPOLL EQU     0H              ; TIMER OUTPUT ACTIVE LOW
;*************************************************************************
;               NSC 810 ADDRESS EQUATES
;
NSCIOT  EQU     3080H           ; NSC-810 I/O-Timer base
BUF19   EQU     CNOT+30H        ;
ABRA    EQU     4350H           ;
DABRA   EQU     ABRA+22H        ;
NSCRAM  EQU     NSCIOT-80H      ; NSC-810 ram area
```

```
;               ADDRESSES RELATIVE TO NSCIOT(IN IX REG.)
PADADD  EQU     0H      ; PORT A DATA REG
PBDADD  EQU     01H     ; Port B data reg
PCDADD  EQU     02H     ; Port C data reg
PADIR   EQU     04H     ; PORT A DATA DIRECTION REG
PBDIR   EQU     05H     ; Port B data direction reg
PCDIR   EQU     06H     ; Port C data direction reg
CMDRAD  EQU     07H     ; Mode definition reg
PACADD  EQU     08H     ; PORT A BIT CLEAR REG
PBCADD  EQU     09H     ; Port B bit clear reg
PCCADD  EQU     0AH     ; Port C bit clear reg
PASADD  EQU     0CH     ; PORT A BIT SET REG
PBSADD  EQU     0DH     ; Port B bit set reg
PCSADD  EQU     0EH     ; Port C bit set reg
T0LSB   EQU     10H     ; Timer 0 lsb
T0MSB   EQU     11H     ; Timer 0 msb
T1LSB   EQU     12H     ; Timer 1 lsb
T1MSB   EQU     13H     ; Timer 1 msb
T0STOP  EQU     14H     ; Timer 0 stop
T0STRT  EQU     15H     ; Timer 0 start
T1STOP  EQU     16H     ; Timer 1 stop
T1STRT  EQU     17H     ; Timer 1 start
T0CMD   EQU     18H     ; Timer 0 command reg
T1CMD   EQU     19H     ; Timer 1 command reg
XAMCH   EQU     BUF19+2 ;

;END OF ADDRESS EQUATES FOR NSC 810
;*************************************************************************
; END OF NSC 810 EQUATES.
```

```
;LAST UPDATE 01/03/84 DHH
;*************************************************************************
                    POWER BOARD PORT
                     ________________
                    |                |
S-100,P36/D0--------)---|1 A0      VDD 16|---
S-100,P35/D1--------)---|2 A1       CL 15|---(----GRND
GRND----------------)---|3 A2       NE 14|---(----FROM U1 PIN 8(SEE BELOW)
TO Q4 CONT. CART+5V-(---|4 Q0  4724  D 13|---(----DATA BIT 7,S100,P90
TO LCD CTR CLOCK----(---|5 Q1       Q7 12|---)NOT USED
TO LCD CTR NRSET----(---|6 Q2       Q6 11|---)NOT USED
TO Q5 CONT. CART+/-24V(-|7 Q3       Q5 10|---)NOT USED
                     ---|8 VSS      Q4  9|---)NOT USED
                        |________________|

NOTES:CL AND NE HIGH,ALL OUTPUTS LOW
      CL HI,NE LO,ACTS AS A DEMULTIPLEXER
      IF NE AND CL ARE BOTH LOW,ADDRESSABLE LATCH
NB:   IN THIS MODE,WHICH IS THE WAY IT IS SET UP,A TRANSIENT WRONG ADDRESS
WILL OCCUR IF MORE THAN ONE ADDRESS LINE CHANGES WHILE NE IS LOW.THIS IS
GUARANTEED TO HAPPEN GIVEN THIS SET-UP.P L E A S E SEE NATIONAL BOOK.THIS
CAN CAUSE ALL KINDS OF GARBAGE ANY TIME THIS PORT IS ADDRESSED.ALSO PLEASE
OBSERVE THAT IT MIGHT REDUCE POWER CONSUMPTION TO CUT OFF THE 12V TO THE
REGULATOR RATHER THAN THE 5V OUTPUT TO THE CARTRIDGE RECORDER,GIVEN THAT
ELECTROLYTICS LEAK.
PLEASE NOTE ALSO THAT THE "EXTRES" LINE SHOULD HAVE AN R/C DEBOUNCE
;*************************************************************************
                             FUNCTIONS:
;*************************************************************************
                  OUT FF  ,___  PULLS NE LOW VIA ADDRESS NAND(U2) NAND
                                NWR AND SOUT.NOTE THAT THERE ARE A LOT OF
                                PINS UNUSED SO AN RC COULD BE ADDED TO
                                THESE GATES TO REMOVE GLITCHES NOTED ABOVE


                  DATA BIT 7    CONTROLS WHETHER ADDRESSED OUTPUT WILL GO HI
                                OR LOW.NOTE THAT UNUSED Q OUTPUTS COULD BE USED
                                TO LATCH ONE ADDRESS LINE.

                  DATA BITS 0,1 SELECT ADDRESS

                  (0)  00       5V TO CARTRIDGE
                  (1)  01       LCD CLOCK LINE
                  (2)  10       LCD COUNTER RESET LINE
                  (3)  11       24V TO CARTRIDGE
;*************************************************************************
;
;                   POWER PORT EQUATES
;*************************************************************************
OFF5V   EQU     00      ;BYTE TO TURN OFF 5V TO CARTRIDGE RECORDER
ON5V    EQU     80H     ;BYTE TO TURN ON 5V TO CARTRIDGE RECORDER
LCDCLL  EQU     01H     ;SET LCD CLOCK LINE LOW
LCDCLH  EQU     81H     ;SET LCD CLOCK HI
LCDRST  EQU     02H     ;PULL RESET LOW(ACTIVE)ON LCD CTR
```

```
NLCDR   EQU     82H     ;RESET HI ON LCD(NOT RESET)
OFF24V  EQU     03H     ;TURN OFF 24V TO CARTRIDGE RECORDER
ON24V   EQU     83H     ;TURN ON 24V TO CARTRIDGE RECORDER
;*****************************************************************************
CQCHEK  EQU     TZBUF+53H       ;
QBASA   EQU     TDBUF+4         ;
CATHD   EQU     CQCHEK+3        ;
;*****************************************************************************
;
```

```
;LAST REVISED 5/06/84
;*******************************************************************************

;LOAD
;THIS INSTRUCTION ALLOWS ANY REGISTER PAIR TO BE LOADED FROM ANY OTHER
;REGISTER PAIR WITHIN THE SAME REGISTER SET
;TWO BYTES,21 TSTATES
LOAD    MACRO   REG1,REG2
        PUSH    REG2
        POP     REG1
        ENDM

;
;STA
;THIS INSTRUCTION ALLOWS THE DIRECT STORAGE OF ANY REGISTER OR IMMEDIATE VALUE
;IN A SPECIFIED MEMORY LOCATION.NO REGISTERS AFFECTED.
;6 BYTES FOR REGISTER,38 TSTATES
;7 BYTES FOR IMMEDIATE,41 TSTATES
STA     MACRO   ADDR,ARG
        PUSH    AF
        LD      A,ARG
        LD      ADDR,A
        POP     AF
        ENDM

;
;LDR
;THIS INSTRUCTION ALLOWS THE DIRECT LOADING OF ANY REGISTER FROM ANY
;SPECIFIED MEMORY LOCATION.NO REGISTERS AFFECTED.
;6 BYTES, 38 T STATES
LDR     MACRO   REG,ADDR
        PUSH    AF
        LD      A,ADDR
        LD      REG,A
        POP     AF
        ENDM
;THIS MACRO DELAYS N HUNDRED T-STATES
;
DELAY   MACRO   NHUNDRED
        PUSH    BC              ;11 T --SAVE B REGISTER
        LD      B,NHUNDRED-1    ; 7 T --SET UP B FOR DJNZ IN DELAYR
        CALL    DELAYR  ;17 T --CALL DELAY SUBROUTINE
        POP     BC              ;10 T --RESTORE B AND CONTINUE
        ENDM                    ;END OF MACRO
;
;MACRO USES 45 TSTATES.DELAYR WASTES 55 TSTATES AND CALLS A 100 T-STATE
;DELAY N-1 TIMES FOR A TOTAL DELAY OF N HUNDRED TSTATES.  NOTE THAT MAX.
;DELAY FOR THIS MACRO IS 25.6 MS.  DELAY TIMES ARE BASED ON T-STATE=1 MICRO-
;SEC WHICH IS ONLY APPROXIMATE.  APPROXIMATION IS NOT USED FOR CRITICAL DELAYS.;
;*******************************************************************************
;MACRO TO DO GEN. INFO INPUT
;PRINT   MACRO   STRING
        PUSHALL                 ;SAVE ALL REGS
        LD      HL,STRING       ;LOAD HL WITH PARTICULAR MESSAGE
        CALL    WPRA            ;CALL SUBROUTINE TO PRINT MESSAGE+?
```

```
        LD      HL,STRING       ;SET UP TO MOVE TO BUFFER
        CALL    WPRB            ;SUB TO MOVE QUESTION+RESPONSE TO BUFFER
        POPALL                  ;RESTORE REGS
        ENDM
;*****************************************************************************
;
PRINT   MACRO   STRING
        PUSH    HL              ; Usage:
        LD      HL,STRING       ;               PRINT DIAGMS
        CALL    SNDMES  ;       DIAGMS: DB 'MESSAGE',0
        POP     HL              ; Note: ALTERS NOTHING
        ENDM
;
;
pushall macro                   ;
        push    af              ;
        push    bc              ;
        push    de              ;
        push    hl              ;
        endm
;
popall  macro                   ;
        pop     hl              ;
        pop     de              ;
        pop     bc              ;
        pop     af              ;
        endm
;
;*****************************************************************************
PRINTREG        MACRO   STRING
        LD      HL,STRING&M
        CALL    SNDMES
        LD      HL,(P&STRING)
        CALL    LADR
        CALL    CRLF
        ENDM
;
MSG     MACRO   STRING
STRING&M::      DB      '&STRING&= '
        DB      0
        ENDM
;
;*****************************************************************************
;THIS MACRO PRINTS WHATEVER FOLLOWS(IMMEDIATE)
;NO QUOTES ARE REQUIRED,NO REGS ALTERED
;SHOULD ENCLOSE IN L+R ANGLE BRACKETS ()
;
IPRINT  MACRO   STRING
        LOCAL   ABC,BCD
        PUSH    HL
        LD      HL,ABC
        CALL    SNDMES
        POP     HL
        JR      BCD
ABC:    DB      '&STRING&'
```

```
        DB      CR,LF,0
BCD:
        ENDM
```

```
;.Z80
NAME ('OBIP')
TITLE   OBIP SOFTWARE
SUBTTL  PAGE ZERO INTERRUPT AND RST VECTORS
;
INCLUDE W.EQU
INCLUDE     WNSC810.EQU
INCLUDE WPWRPRT.EQU
MACLIB  MAC

        ORG     0000H
;
;*************************************************************************
;   PROCESSOR STARTS HERE ON RESET. INTERRUPT AND REFRESH REGS ARE       *
;   CLEARED. ALL INTERRUPTS ARE DISABLED (HARDWARE DI INSTRUCTION). INTERRUPT *
;   CONTROL REGISTER IS SET TO 01, WHICH ENABLES "NOT INTR" AND MASKS OFF *
;   "NOT" RSTA, RSTB, RSTC. REMEMBER -- NO INTERRUPT CAN WORK UNLESS BOTH *
;   AN EI INSTRUCTION HAS BEEN ISSUED AND THERE IS A ONE IN THE APPROPIATE *
;   SPOT IN THE INTERRUPT MASK REGISTER, WHICH IS A WRITE ONLY REGISTER   *
;   ADDRESSED AS AN OUTPUT PORT (LOWER 4 BITS ONLY) BY AN OUT BBH OR EQUIV. *
;   INSTRUCTION. THE FOLLOWING VALUES ENABLE THE CORRESPONDING INT. LINES: *
;         08H               ENABLES RSTA                                  *
;         04H               ENABLES RSTB                                  *
;         02H               ENABLES RSTC                                  *
;         01H               ENABLES INTR                                  *
;   NOTE THAT 0FH ENABLES ALL INT. LINES                                  *
;   8080 INTERRUPT MODE IS AUTOMATICALLY SELECTED ON RESET                *
;*************************************************************************
        RESTART 0       (11000111)!
START:  DI                      ; NOT NECESSARY
        JP      MAIN            ;
;
;       END OF RESET CODE
;*************************************************************************
THESE ARE THE 8 NORMAL Z80 RESTART LOCATIONS. IN 8080 MODE , THESE
CONSTITUTE A MEANS FOR EXTERNAL DEVICES TO FORCE A CALL TO ONE OF 8 LOCATIONS
WITH ONE INSTRUCTION PUT ON THE BUS DURING AN INTACK CYCLE. ALSO CAN BE USED
AS A VERY SHORT CALL VIA RST INSTRUCTION.
NOTE THAT A DI INSTRUCTION IS AUTOMATICALLY EXECUTED
SEE NATIONAL NSC 800 PAGE 4-17. (ALSO STANDARD ON 8080 AND Z80)
;*************************************************************************
!
;       RESTART 1       (11001111)
        ASEG
        ORG     0008H
        RET
;
;*************************************************************************
;NOTE: THIS SHOULD BE ALTERED SINCE THERE IS NO REASONABLE WAY
;WE CAN GET THE CONSOLE TO TRIGGER INTERRUPT 1. MONITOR SHOULD USE
;RST7 AND THE TERMINATOR SHOULD USE RSTB OR RSTC. (MONITOR REQUIRES
;BOTH AN INSTRUCTION AND A HARDWARE RESTART.)
;
;*************************************************************************
```

```
;
;       RESTART 2       (11010111)
        ASEG
        ORG     0010H
        RET
;
;       RESTART 3       (11011111)
        ASEG
        ORG     0018H           ;
        RET
;
;       RESTART 4       (11100111)
        ASEG
        ORG     0020H
        RET
;
;       RESTART 5       (11101111)
        ASEG
        ORG     0028H
        RET
;
;       RESTART C
        ASEG
        ORG     002CH
        RET
;
;       RESTART 6       (11110111)
        ASEG
        ORG     0030H
        RET
;
        RESTART B
        ASEG
        ORG     0034H
        LD      HL,(T1INT)
        JP      (HL)
;
        RESTART 7       (11111111)
                                ;IN MODE 1,WILL COME HERE ON NOT INTR GOING LOW
        ASEG
        ORG     0038H
        JP      ETERM           ; EARLY TERMINATION ROUTINE
NOTE THAT THIS REQUIRES HARDWARE OR MODE1 INTERRUPTS!
*****************************************************************************
        NSC     800 SPECIAL INTERRUPT LOCATIONS
        IF INT. ENABLED AND MASK OK WILL COME HERE AS SHOWN.
        NO INSTRUCTION NEEDED ON BUS-- JUST PULL THE APPROPIATE PIN LOW.
 RSTC AND RSTB
 THESE TWO INTERRUPTS ARE NOT USED.
 THESE HAVE NOW BEEN MADE RETURNS FOR SAFETY.7/16/83
*****************************************************************************
*****************************************************************************
                                RESTART A
```

```
;THIS INTERRUPT IS GENERATED ON THE A/D BOARD.IT IS CONTROLLED BY A NUMBER
;OF CPU BOARD OUTPUTS.THE NSC 810 PORT C PIN 0 CONTROLS AN OR GATE WHICH
;DISABLES THE INTERUPT COMING FROM THE A/D BOARD ON S-100 PIN 4 IF C-0 IS
;HIGH.
;THE A/D BOARD WILL NOT GENERATE AN INTERRUPT IF ITS FIRST
;INTERRUPT WAS NOT ACKNOWLEDGED.THE A/D INTERRUPT OCCURS IMMEDIATELY AFTER
;CONVERSION COMPLETE FOLLOWING A TIME-OUT OF THE SOFTWARE SETTABLE COUNTER
;WHICH GIVES THE APPROXIMATE SAMPLE(WILL HAVE ABOUT 200 MICROSECONDS OF SLOP)
;TIME.
;*************************************************************************
        ASEG
        ORG     003CH
AQUINT: EX      AF,AF'
        EXX
        LD      A,(XAMCH)
        CP      4DH
        JR      NZ,AQUINT
        LD      HL,(BUFPTR)
        LD      BC,(NCX2IO)
        LD      A,(BSTART)
        LD      D,A
        OR      H
        LD      H,A
;*************************************************************************
;C REGISTER IS THE IO PORT BASE ADDRESS-1:B REGISTER IS THE COUNTER
;SO FETCH DATA FROM IO 18 THROUGH 18 +2 TIMES Number Channels--NCX2 BYTES TOTAL
;*************************************************************************
AQLP:   INC     C               ; IO PORT=IO PORT + 1(FOR LOOP)
        INI                     ; BLOCK INPUT INSTR.
        JR      Z,ALLIN
        LD      A,H
        OR      D
        LD      H,A
        JR      AQLP
;
;*************************************************************************
;FETCH FROM IO PORT (C),STORE AT HL,DECR. B, INCR. HL .
;*************************************************************************
;
ALLIN:  LD      (BUFPTR),HL     ; SAVE BUFFER POINTER
;
;*************************************************************************
;TO MAINTAIN CIRCULAR 32K BUFFER,MAKE SURE HL ALWAYS HAS HIGH BIT SET.
;THUS FFFF INCREMENTS TO 0000,BUT THIS SETS IT BACK TO 8000H.
;*************************************************************************
        LD      HL,(NSAMPS)     ; LOAD # OF SAMPLES
        DEC     HL              ; JUST DONE ONE
        LD      (NSAMPS),HL     ; PUT SAMPLE COUNT BACK
;
        EXX                     ; BACK TO REGULAR REGS
        EX      AF,AF'          ; ACCUMULATOR AND FLAGS
        EI                      ; ENABLE INTERUPTS
        RET                     ; RETURN
;
; ---- INITIALIZE OPERATING PARAMETERS ----
```

```
;
MAIN:   IM      1               ;
        XOR     A               ; Clear accum.
        OUT     (INTPRT),A      ; DISABLE INTERUPTS
        LD      IX,NSCIOT       ; SETS UP THE IX REGISTER TO POINT TO THE
                                ; NSC 810
;
;                   SET-UP STACK
;
        LD      SP,STACK        ; SET STACK TOP -DATA PUSHED ON STACK FROM
;                                 304FH DOWN TO 3000H IN NSC 810 RAM
;
;                 INITIALIZE NSC 810
;
;*************************************************************************
;THE FOLLOWING CODE INITIALIZES THE NSC 810. IT SHOULD BE NOTED THAT AFTER
;RESET THE 810 IS IN THE FOLLOWING CONDITION:
;       1. ALL INTERNAL REGISTERS ARE ZEROED
;       2. ALL COUNTER/TIMERS ARE STOPPED AND RESET.
;       3. ALL IO PORTS GO TO HIGH Z INPUT MODE
;       4. THE RAM IS LEFT UNCHANGED
;*************************************************************************
;
INI810: LD(IX+PADADD),ADAT      ;INITIALIZE PORT A DATA
        LD(IX+PBDADD),BDAT      ;INIT PORT B DATA
        LD(IX+PCDADD),CDAT      ;INIT PORT C DATA
        LD(IX+PADIR),ADIR       ;INIT PORT A DIRECTION
        LD(IX+PBDIR),BDIR       ;INIT PORT B DIRECTION
        LD(IX+PCDIR),CDIR       ;INIT PORT C DIRECTION
        LD(IX+CMDRAD),MODE0     ;INIT NSC 810 MODE
T0DAT   EQU     TMODE1 OR PRE64 OR BIT16T OR GPOLH OR OUTPOLL
        LD(IX+T0CMD),T0DAT      ;INIT TIMER0 COMMAND REG
T1DAT   EQU     TMODE1 OR PRE1 OR BIT16T OR GPOLH OR OUTPOLL
        LD(IX+T1CMD),T1DAT      ;INIT TIMER1 COMMAND REG
;
        LD      (STOPT0),A
        LD      (IX+T0LSB),0FFH
        LD      (IX+T0MSB),03FH
;
;
        LD      (STOPT1),A
        LD      (IX+T1LSB),0FFH
        LD      (IX+T1MSB),59
;
;
        LD      HL,STAMB
        LD      A,(QBASA)
        CP      4FH
        JR      NZ,LTMR
        LD      HL,MINTMR
LTMR:   LD      (T1INT),HL
;*************************************************************************
;                   END NSC 810 INITIALIZATION
;*************************************************************************
;                   INITIALIZE EVENT DETECTOR
```

```
;
        LD      A,1             ; Set average high and
        OUT     (AVGPRT),A      ;  clear pending interrupts
;CANNOT DOCUMENT FURTHER WITHOUT SCHEMATIC AND EXPLANATION
;*************************************************************************
;                       INITIALIZE A/D BOARD
;*************************************************************************
;                   A/D BOARD HARDWARE SUMMARY
;
;THE A/D BOARD CONSISTS OF 4 CHANNELS OF QUASI 16 BIT A/D CONVERSION. THIS IS
;ACCOMPLISHED BY A 12 BIT A/D USING 4 BITS OF D/A AS A GAIN RANGING DEVICE.
;THESE 4 BITS OF D/A MAY BE SET BY SOFTWARE,BUT, IF SO,THEY ARE SENT TO ALL
;FOUR CHANNELS,WHILE THE HARDWARE GAIN SETTING OPTION WORKS ON EACH CHANNEL
;INDIVIDUALLY.
;
;*************************************************************************
;
;
;
;
; ---- Force A-D gain to zero
;
        ld      a,0ch           ; Select 128 Hz and force
        out     (adport),A      ;  manual gain of zero
        ld      a,0             ; Select 128 Hz and
        out     (adport),a      ;  gain ranging mode
;CANNOT DOCUMENT FURTHER WITHOUT SCHEMATIC AND EXPLANATION
;*************************************************************************
;                       INITIALIZE POWER BOARD
;*************************************************************************
;
;THE FOLLOWING CODE INITIALIZES THE POWER BOARD
;CARTRIDGE RECORDER IS TURNED OFF
;
;*************************************************************************
PIBINIT:        CALL    OFFCART ;KILL CARTRIDGE POWER
;POWER INTERFACE BOARD INITIALIZED
;*************************************************************************
TERMWAI:LD      A,(PBDATA)
        RLA
        JP      NC,TERMWAI
;WAIT FOR TERMINAL TO TURN ON(BAUD RATE FIXED AT 1200 BAUD)
;*************************************************************************
;
        LD      A,0C5H
        LD      C,A
        PUSH    BC
        POP     AF
        CALL    NZ,BADPRO
        CALL    NC,BADPRO
        CALL    P,BADPRO
        CALL    PO,BADPRO
        XOR     A
        LD      C,A
```

```
        PUSH    BC
        POP     AF
        CALL    Z,BADPRO
        CALL    C,BADPRO
        CALL    M,BADPRO
        CALL    PE,BADPRO
        LD      A,40H
        ADD     A,A
        CALL    PO,BADPRO
        RLCA
        CALL    NC,BADPRO
        JR      REGTEST
BADPRO: PRINT   BADPR
        POP     HL
        JP      DBPXM
REGTEST:        LD      A,0FFH
        CALL    SHIFTREG
        CALL    NZ,BADPRO
        EXX
        CALL    SHIFTREG
        CALL    NZ,BADPRO
        EX      AF,AF'
        LD      A,B
        LD      L,A
        EX      AF,AF'
        CP      L
        CALL    NZ,BADPRO
        XOR     A
        CALL    SHIFTREG
        CALL    NZ,BADPRO
        EXX
        CALL    SHIFTREG
        CALL    NZ,BADPRO
        EX      AF,AF'
        LD      A,B
        LD      L,A
        EX      AF,AF'
        CP      L
        CALL    NZ,BADPRO
        LD      IX,0
        LD      SP,0FFFEH
        EX      (SP),IX
        LD      IY,0FFFFH
        EX      (SP),IY
        PUSH    IY
        POP     HL
        XOR     A
        CP      H
        CALL    NZ,BADPRO
        CP      L
        CALL    NZ,BADPRO
        EX      (SP),IX
        PUSH    IX
        POP     HL
        LD      A,0FFH
```

```
        CP      H
        CALL    NZ,BADPRO
        CP      L
        CALL    NZ,BADPRO
        XOR     A
        LD      SP,0FFFFH
        SBC     HL,SP
        CALL    NZ,BADPRO
        LD      SP,0000
        ADC     HL,SP
        LD      SP,STACK
        CP      H
        CALL    NZ,BADPRO
        CP      L
        CALL    NZ,BADPRO
        LD      IX,NSCIDT
;
7810:   PRINT   CNOT
        STA     (TSTFLG),0H
        PRINT   CLRSCR
        CALL    SETCLK
NSC810: LD      HL,NSCRAM
NSCFILL:LD      A,L
        XOR     B
        LD      (HL),A
        INC     HL
        LD      A,L
        CP      80H             ;END OF NSC810 RAM
        JP      NZ,NSCFILL
;
        LD      HL,NSCRAM
;
NSCTEST:LD      A,L
        XOR     B
        CP      (HL)
        JP      NZ,NSCBAD
        INC     HL
        LD      A,L
        CP      80H
        JP      NZ,NSCTEST
        INC     B
        JP      NZ,NSC810
NSCOK:  PRINT   OKNSC
        JP      TCON

NSCBAD: LD      (ADDRESS),HL
        LD      (PATRN),A
        CALL    RAMER

TCON:   LD      SP,STACK
        LD      A,(TSTFLG)
        CP      99H
        JP      Z,TESTED
        LD      A,(TIMEN)
        LD      (STRTTB),A
```

```
        CALL    INIT_HDR_RAM
        LD      HL,4000H
        CALL    CLR2K
        WPRINT  TSCR
        CALL    SKIPPER
        JR      Z,TESTB
        PRINT   CLRSCR
        PRINT   TRAM
        CALL    CRLF
        LD      HL,4000H
        LD      (RAMST),HL
        LD      BC,60H
        LD      (ENDRAM),BC
        CALL    RAMTST
        CALL    RAMOKM
        LD      HL,4000H
        CALL    CLR2K           ;CLEAR RAM AFTER TEST
;*****************************************************************************
;       SCRATCH PAD TEST COMPLETE,SAFE TO USE
;*****************************************************************************
TESTB:  WPRINT  TSTB?
        CALL    SKIPPER
        JR      Z,TSKIP
        PRINT   CLRSCR
        PRINT   TZBUF
        PRINT   TRAM
        LD      HL,8000H
        LD      (RAMST),HL
        LD      BC,0H
        LD      (ENDRAM),BC
        CALL    RAMTST
        CALL    RAMOKM
        CALL    INIT_HDR_RAM
        STA     (TSTFLG),99H
TESTED:
TSKIP:  CALL    GCLOCK
        LD      A,(STRTTB)
        LD      (TIMEN),A
        CALL    CONT
        CALL    CLEAR
        LD      SP,NSTACK
        LD      HL,0
        LD      DE,BASAD
        LD      C,20H
ABLP:   XOR     A
        LD      B,A
ADDLP:  ADD     A,M
        INC     HL
        DJNZ    ADDLP
        LD      (DE),A
        INC     DE
        DEC     C
        JR      NZ,ABLP
        LD      HL,(HDRBUF)
        PUSH    HL
```

```
        LD      HL,NSCRAM
        CALL    CLR128
        POP     HL
        LD      (HDRBUF),HL
        WPRINT  DEPL
        WPRINT  INSTR
        WPRINT  CHSCI
        WPRINT  CRUISE
        WPRINT  SPHERE
        WPRINT  LAT
        WPRINT  LONG
        PRINT   FEG
        LD      HL,FEG
        LD      DE,(HDRBUF)
        CALL    MOVLP
        CALL    CHANNEL
        PRINT   FED
        LD      HL,FED
        LD      DE,(HDRBUF)
        CALL    MOVLP
        CALL    CHANNEL
        LD      A,CTLZ
        LD      DE,(HDRBUF)
        CALL    INS
        CALL    CONT
        CALL    CLEAR
        LD      HL,HDRRAM
PBACK:  LD      A,(HL)
        CP      CTLZ
        JR      Z,ISOK
        CALL    PBYT
        INC     HL
        JR      PBACK
ISOK:   CALL    OK?
        PUSH    PSW
        CALL    CRLF
        POP     PSW
        JP      NZ,TCON
        LD      DE,ABRA
        LD      HL,KA
        PUSH    IX
        LD      IX,DABRA
        LD      B,1FH
ABALP:  LD      C,M
        LD      (IX),C
        LD      A,(DE)
        CP      C
        JP      NZ,BPA
        INC     IX
        INC     HL
        INC     DE
        DJNZ    ABALP
        POP     IX
        CALL    BTAPE
        CALL    PARAM
```

```
EXECU:
        IPRINT  (IS THIS A TEST? (Y/N))
        CALL    GBYTNE
        CP      'Y'
        JR      Z,YTEST
        PRINT   UNPLUGM
DISCON: LD      A,(PBDATA)
        RLA
        JR      C,DISCON
        LD      B,20
SLOWPOKE:
        PRINT   BEEP
        DELAY   256
        LD      A,(PBDATA)
        RLA
        JR      C,SLOWPOKE
        DJNZ    SLOWPOKE
YTEST:  CALL    WGPHDR
        LD      A,(BUF19)
        CP      41H
        JR      NZ,LTMR1
        LD      HL,MINTMR
LTMR1:  LD      (T1INT),HL
        JP      CONTROL
ENDPRO:
        CALL    CRLF
        IPRINT  (PROG OVR)
        CALL    W_EOD
        DI
        LD      HL,STAMB
        LD      (STOPT1),A
        LD      (IX+T1LSB),0FFH
        LD      (IX+T1MSB),00
        LD      A,(TDBUF)
        CP      48H
        JR      NZ,NRDX
        LD      HL,TESTTIMER
NRDX:   LD      (T1INT),HL
        LD      A,(IX+T0LSB)
        LD      A,(IX+T0MSB)
        LD      HL,PBDATA
WAIT_FOR:
        BIT     6,M
        JR      NZ,WAIT_FOR
        LD      (STRT1),A
        LD      A,T1_INT_EN OR EARLY_TERM
        OUT     (INTPRT),A
        EI
LOOPER:
        JR      LOOPER
*****************************************************************************
SUBROUTINES
CHANNEL:        WPRINT  CH1
        WPRINT  CH2
        WPRINT  CH3
```

```
        WPRINT  CH4
        RET
;
SHIFTREG:       LD      B,A
        LD      C,B
        LD      D,C
        LD      E,D
        LD      H,E
        LD      L,H
        CP      L
        RET
;
INIT_HDR_RAM:
        LD      HL,HDRRAM
        LD      (HDRBUF),HL
        CALL    CLR2K
        RET
;
```

```
;
SETCLK:PUSHALL
       PRINT   SETM
       LD      HL,TIMEN
       CALL    TIMINP
       LD      E,M
       LD      D,0
       LD      A,16
       CALL    DIVIDE
       LD      E,B
       LD      A,4
       CALL    DIVIDE
       LD      E,10H
       INC     B
YRLOOP: SRA    E
       DJNZ    YRLOOP
       STA     (MYRS),E
;
LCM:   XOR     A
       LD      (STPST),A
       LD      (MTMEM),A
       LD      (MTMEM+14),A
       LD      (MTMEM+15),A
       LD      (OTENMTH),A
       LD      DE,MTMEM+12
       LD      HL,TIMEN+1
       CALL    MOVT
       DEC     DE
       LD      B,3
TLX:   CALL    MOVT
       DJNZ    TLX
MCLOCK: LD     HL,MTMEM
       LD      DE,TMEM
       LD      BC,16
       LDIR
       LD      IX,NSCIOT
SETCLK2:CALL   CONT
       PUSH    HL
       LD      HL,CATHD
       LD      A,4EH
       CP      M
       JP      NZ,STAMB
       POP     HL
       PRINT   CLRSCR          ;CALL CLEAR SCREEN
       PRINT   CLKPRM          ;SAT CLOCK PROMPT
       LD      A,(IX+0)        ;READ PORT "A" BIT 6
       BIT     6,A             ;TEST BIT 6
       JR      Z,SCCON         ;SAT. CLK. CONNECTED ?
       PRINT   SATNC           ;NO! SAT. CLK. NOT. CONN.
       JR      CLK2            ;
SCCON:  PRINT  SATC            ;print sat clk. conn.
CLK2:  CALL    GCLOCK          ;print time of projected start
       CALL    TIMOUT
       CALL    GBYT            ;test for control B from
```

```
        CP      02              ;console
        JR      Z,SKIP          ;start clock if cntl "b"
        CP      CR              ;carriage return
        JR      NZ,SETCLK2      ;repeat procedure is screw-up
        LD      A,(IX+0)        ;read port "a"
        BIT     6,A             ;test bit 6
        JR      NZ,SETCLK2      ;must have sat clk. connected
CLKLOO: LD      A,(IX+0)        ;read port "a"
        BIT     6,A             ;stay in loop until bit 6
        JR      Z,CLKLOO        ;goes high
SKIP:   DELAY   256             ;DELAY .1 SEC FOR NATIONAL DISASTER
        DELAY   256             ;this delays compensates for the
        DELAY   256             ;fact that the 58174 always
        DELAY   256             ;starts @ .1 seconds
        DELAY   30
        call    d27t
        LD      A,1             ;load acc. with 1 to start
        LD      (TMEM+14),A     ;real time clock
:satellite start secuence complete,START TIMERS
        IPRINT  (SYNCHRONIZING TIMERS)
        CALL    SECRD
        CALL    SECINC
SECLP:  CALL    SECRD
        CP      E
        JR      NZ,SECLP        ;TIMERS SYNCHED WITH TRANSITION
        LD      (STRT0),A
        LD      (STRT1),A
        POPALL
        RET
;
TESTTIMER:
        EXX
        EX      AF,AF'
        LD      L,(IX+T0LSB)
        LD      H,(IX+T0MSB)
        LD      (IX+PASADD),2
        CALL    GCLOCK
        PUSH    HL
        LD      HL,TIMEN
        CALL    TIMOUT
        POP     HL
        CALL    PHREG
        CALL    CRLF
        LD      A,(IX+T1LSB)
        LD      A,(IX+T1MSB)
        LD      (IX+PACADD),0FFH
        EX      AF,AF'
        EXX
        EI
        RET
;
MINTMR:
        EXX
        EX      AF,AF'
        LD      A,(IX+T1LSB)
```

```
        LD      A,(IX+T1MSB)
        LD      HL,ELAPSED_MIN
        LD      A,M
        ADD     A,1
        DAA
        LD      M,A
        EX      AF,AF'
        EXX
        EI
        RET
;
GCLOCK: PUSHALL
GCLOCK1:LD      HL,TMEM+1
        LD      DE,MTMEM+1
        LD      BC,12
        LDIR
        LD      HL,MTMEM+1
        CALL    NOFS
        JR      Z,GCLOCK1
PREPT:  LD      HL,TIMEN+1
        LD      DE,MTMEM+12
MONIN:  CALL    DIGIN
        CALL    DIGIN
        DEC     DE
        LD      B,3
RESIN:  INC     HL
        CALL    DIGIN
        CALL    DIGIN
        DJNZ    RESIN
        LD      A,(MTENMTH)
        LD      HL,OTENMTH
        CP      M
        LD      HL,TIMEN
        JR      NC,OLDYR
        INC     M
OLDYR:  LD      (OTENMTH),A
        LD      HL,CQCHEK
        LD      A,4DH
        CP      M
        JR      NZ,GCLOCK
        POPALL
        RET
;
MOVT:   RLD
        LD      (DE),A
        DEC     DE
        XOR     A
        RLD
        LD      (DE),A
        DEC     DE
        INC     HL
        XOR     A
        RET
;
DIGIN:  LD      A,(DE)
```

```
        AND     0FH
        RLD
        DEC     DE
        RET
;
TCHECK:
        PUSHALL
        LD      A,(CEXPN)
        CP      1
        JR      NZ,NOT_EXP1
        DI
        STA     (ELAPSED_MIN),0
        LD      (STOPT1),A
        LD      HL,STRTTB
        LD      DE,TIMEN
NOT_YET:
        CALL    SECRD
        CALL    GCLOCK
        CALL    CTIM
        JR      C,NOT_YET
        LD      A,(OFFSET)
        CP      0
        JR      Z,NOW
        LD      E,A
OFFSET_WAIT:
        CALL    SECRD
        LD      B,A
        LD      A,(TENSEC)
        SLA     A
        SLA     A
        SLA     A
        SLA     A
        OR      B
        CP      E
        JR      NZ,OFFSET_WAIT
NOW:
        LD      (STRT1),A
AGIN:   CALL    TIMREC
        POPALL
        EI
        RET
;
NOT_EXP1:
        LD      A,(PERIOD)
        LD      HL,ELAPSED_MIN
        CP      M
        INC     HL
        LD      M,A
        DEC     HL
        JR      NZ,NOT_EXP1
        LD      M,0
        LD      HL,PDIGE
        LD      A,44H
        CP      M
        JP      NZ,STAMB
```

```
        JR      AGIN
;
NOTSTOP:
        DI
        CALL    SECRD
        CALL    GCLOCK
        EI
        LD      HL,TIMEN
        LD      DE,STOPTB
        CALL    CTIM
        RET
;
SECRD:
        LD      A,(TMEM+2)
        AND     0FH
        CP      0FH
        JR      NZ,SECRD
NOTOVR:
        LD      A,(TMEM+2)
        AND     0FH
        CP      0FH
        JR      Z,NOTOVR
        RET
;
SECINC: INC     A
        DAA
        AND     0FH
        LD      E,A
        RET
;
TIMREC:
        PUSHALL
RCLOCK: LD      DE,SAMPTIM
        LD      HL,TMEM+1
        LD      BC,12
        LDIR
        LD      HL,SAMPTIM
        CALL    NOFS
        JR      Z,RCLOCK
        LD      A,(CTENMTH)
        LD      HL,OTENMTH
        CP      M
        LD      HL,TIMEN
        JR      NC,COLDYR
        INC     M
COLDYR: LD      (OTENMTH),A
        LD      A,M
        LD      (CYR),A
        LD      L,(IX+T0LSB)
        LD      H,(IX+T0MSB)
        LD      (CTIMER),HL
        POPALL
        RET
;
NOFS:   PUSH    HL
```

```
        PUSH    BC
        LD      C,0FH
        LD      B,12
CHCLLP: XOR     A
        OR      M
        AND     C
        CP      C
        JR      Z,RREAD
        INC     HL
        DJNZ    CHCLLP
RREAD:  POP     BC
        POP     HL
        RET
;
CTIM:
        PUSH    BC
        PUSH    DE
        PUSH    HL
        LD      B,5
CTIMLP: LD      A,(DE)
        CP      M
        INC     HL
        INC     DE
        JR      C,BIGGER
        JR      NZ,BIGGER
        DJNZ    CTIMLP
BIGGER: POP     HL
        POP     DE
        POP     BC
        RET
```

```
;
PARAM:
SKGI:   LD      A,30H
        LOAD    IY,AF
        LD      DE,SERBUF
        LD      (PARBUF),DE
NSERIES:LOAD    AF,IY
        INC     A
        LOAD    IY,AF
        CP      3AH
        JR      NZ,REPENT
        CALL    CLEAR
        RET
REPENT: CALL    CLEAR
        PRINT   SN
        CALL    LOP1
        PRINT   WE
        PRINT   QUES
        CALL    GBYT
        LD      B,A
        CALL    GBYTNE
        LD      A,B
        CP      'T'
        JR      NZ,EVENT
        LD      (SERTYP),A
        JR      NCHAN
EVENT:  CP      'E'
        JR      NZ,NOMORE
        LD      (SERTYP),A
        JR      NCHAN
NOMORE: CP      CR
        JR      NZ,REPENT
        RET
NCHAN:  PRINT   NCAP
        LD      HL,NCX2
        CALL    INDAT
        JR      C,NCHAN
        LD      A,4
        CP      M
        LD      HL,ID
        CALL    INDAT
        JR      C,BCHAN
        LD      A,(NCX2)
        DEC     M
        ADD     A,M
        CP      5
        JR      NC,NCHAN
        SLA     (HL)
        LD      A,ADPORT-1
        ADD     A,M
        LD      M,A
        INC     HL
        SLA     (HL)
SIZ:    CALL    CRLF
```

```
        PRINT   BSIZP
        LD      HL,BUFSIZ
        CALL    INDAT
        JR      C,BSIZ
        LD      A,M
        LD      HL,BSTART
;
        CP      1
        JR      C,BSIZ
        JR      NZ,BSIZ2
        LD      DE,2000H
        LD      (HL),0E0H
        JR      SIZDON
BSIZ2:  CP      2
        JR      NZ,BSIZ4
        LD      DE,4000H
        LD      (HL),0C0H
        JR      SIZDON
BSIZ4:  CP      4
        JR      NZ,BSIZ
        LD      DE,8000H
        LD      (HL),D
;NOW COMPUTE # OF SAMPLES
SIZDON:CALL     BSZR
        LD      A,(NCX2)
        CALL    DIVIDE
        LD      (MSAMPS),DE
SAMRAT: CALL    CRLF
        PRINT   SINTV
        PRINT   QUES
        CALL    GBYT
        CP      34H
        JR      NZ,NOT4
        STA     (ADVAL),1
        JR      NEXP
NOT4:   CP      38H
        JR      NZ,SAMRAT
        STA     (ADVAL),5
NEXP:   CALL    GBYTNE
        CALL    SRR
        CALL    CRLF
        CALL    RTIME
        CALL    CRLF
REPN:   LD      A,(SERTYP)
        CP      'E'
        JR      NZ,WNEXP
        PRINT   MAX
WNEXP:  PRINT   NEXPP
        LOAD    AF,IY
        CALL    PBYT
        LD      HL,EXPN
        CALL    INDAT
        JR      C,REPN
        CALL    SAVTIME
RETIM:  PRINT   TIMNOW
```

```
        CALL    GCLOCK
        LD      HL,TIMEN
        CALL    TIMOUT
        PRINT   STFS
        CALL    LOP1
        LD      HL,STRTTB
        CALL    TIMINP
        LD      HL,TIMEN
        CALL    GCLOCK
        LD      HL,STRTTB
        LD      DE,TIMEN
        CALL    CTIM
        JR      C,NFST
        JR      BT
NFST:   LD      DE,STOPTB
FST:    CALL    CTIM
        JR      C,TSOK
BT:     IPRINT  (IMPROPER START TIME)
        JR      RETIM
;
TSOK:   PRINT   STOP
        PRINT   TFS
        CALL    LOP1
        LD      HL,STOPTB
        CALL    TIMINP
        LD      DE,STRTTB
        CALL    CTIM
        JR      C,TOK
        IPRINT  (IMPROPER STOP TIME)
        JP      RETIM
TOK:    LD      A,(SERTYP)
        CP      'E'
        JP      NZ,WINDAT
PES:    PRINT   PESP
        LD      HL,PESAMS
        CALL    INDAT
        JR      C,PES
        LD      A,M
        LD      DE,(MSAMPS)
        PUSH    DE
        POP     HL
        OR      A
        RR      D
        RR      E
        CP      50H
        JR      Z,PESDON
        OR      A
        RR      D
        RR      E
        CP      25H
        JR      Z,PESDON
        CP      75H
        JR      NZ,PES1
        SBC     HL,DE
        EX      DE,HL
```

```
        JR      PESDON
PES1:   CP      87H
        JR      C,PES
        JR      NZ,PES
        RR      D
        RR      E
        OR      A
        SBC     HL,DE
        EX      DE,HL
PESDON: LD      (PESAMPS),DE
SHORTERM:
        PRINT   STA
        LD      HL,STASAV
        PRINT   QUES
        CALL    GETLN
        LD      DE,IBUFF+1
        CALL    BCDIN
        JR      C,SHORTERM
        LD      A,M
        CP      5H
        JR      NZ,P10
        LD      A,10H
        JR      SHOREND
P10:    CP      10H
        JR      NZ,P25
        LD      A,20H
        JR      SHOREND
P25:    CP      25H
        JR      NZ,P50
        LD      A,40H
        JR      SHOREND
P50:    CP      50H
        JR      NZ,SHORTERM
        LD      A,80H
SHOREND:        LD      (ANVAL),A
THRESHOLD:
        PRINT   THRSH
        LD      HL,THRSHSV
        CALL    INDAT
        JR      C,THRESHOLD
        LD      A,M
        CP      6H
        JR      NZ,DB12
        LD      A,1
        JR      THREND
DB12:   CP      12H
        JR      NZ,DB18
        LD      A,2
        JR      THREND
DB18:   CP      18H
        JR      NZ,DB24
        LD      A,4
        JR      THREND
DB24:   CP      24H
        JR      NZ,THRESHOLD
```

```
        LD      A,8
THREND: LD      B,A
        LD      A,(ANVAL)
        OR      B
        LD      (ANVAL),A
        JP      SPITBACK
WINDAT:
WOFFSET:PRINT   WOFF
        PRINT   WOFF1
        LD      HL,OFFSET
        CALL    INDAT
        JR      C,WOFFSET
        LD      A,M
        CP      60H
        JR      NC,WOFFSET
WPERIOD:PRINT   WPER
        PRINT   WPER1
        CALL    MINALP
        OR      30H
        CALL    PBYT
        PRINT   WPER2
        PUSH    BC
        LD      HL,PERIOD
        CALL    INDAT
        POP     BC
        JR      C,WPERIOD
        LD      A,M
        CP      B
        JR      C,WPERIOD
SPITBACK:
        CALL    CRLF
        LD      A,(BUF19)
        CP      41H
        JP      NZ,STAMB
        CALL    CONT
        CALL    SPITP
        LD      A,(SERTYP)
        CP      'E'
        JR      NZ,WINDSPIT
PEST0:  PRINT   PESPP
        PRINT   EQUALS
        LD      HL,PESAMS
        CALL    PBCD
        LD      A,87H
        CP      M
        JR      NZ,PESTY
        LD      A,'.'
        CALL    PBYT
        LD      A,'5'
        CALL    PBYT
PESTY:  LD      A,'%'
        CALL    PBYT
        CALL    CRLF
SHORTSPIT:
        PRINT STA1
```

```
        LD      A,'.'
        CALL    PBYT
        LD      HL,STASAV
        CALL    PBCD
        PRINT   SECOND
        CALL    CRLF
THRESHSPIT:
        PRINT   THRSH1
        LD      HL,THRSHSV
        CALL    PBCD
        PRINT   DBP
        CALL    CRLF
        JR      OKPARAM
WINDSPIT:
        PRINT   WOFF
        PRINT   EQUALS
        LD      HL,OFFSET
        CALL    PBCD
        PRINT   SECOND
        CALL    CRLF
;
        PRINT   WPER
        PRINT   EQUALS
        LD      HL,PERIOD
        CALL    PBCD
        PRINT   MINUTE
        CALL    CRLF
;
OKPARAM:CALL    OK?
        JR      Z,PARSTOR
        CALL    RESTIME
        JP      REPENT
PARSTOR:LD      DE,(PARBUF)
        LD      HL,NSCRAM
PARMOV: LDI
        LD      A,L
        CP      LOW(ENDPARA)
        JR      NZ,PARMOV
        LD      (PARBUF),DE
        JP      NSERIES
        RET
;
SPITP:  PUSHALL
        CALL    BSZR
        CALL    SRR
        CALL    CLEAR
        PRINT   SN
        CALL    LOP1
        LD      A,(SERTYP)
        CP      'E'
        JR      NZ,SWIND
        PRINT   EVENTP
        JR      SPIT2
SWIND:  PRINT   WPRMT
SPIT2:  PRINT   NEXPP
```

```
        CALL    LOP2
        LD      HL,EXPN
        CALL    PBCD
        CALL    CRLF
        PRINT   STFS
        CALL    LOP2
        LD      HL,STRTTB
        CALL    TIMOUT
        PRINT   STOP
        PRINT   TFS
        CALL    LOP2
        LD      HL,STOPTB
        CALL    TIMOUT
        CALL    CRLF
        PRINT   ACHAN
        LD      A,(NCX2)
        RRCA
        LD      B,A
        LD      A,(IO)
        SUB     ADPORT-1
        OR      A
        RRCA
        INC     A
        OR      30H
PCHLO:  CALL    PBYT
        PUSH    AF
        LD      A,","
        CALL    PBYT
        POP     AF
        INC     A
        DJNZ    PCHLO
        CALL    BSPCR
        CALL    CRLF
SPITBSIZ:
        PRINT   BSIZP1
        LD      HL,BSZSAV
        CALL    PBCD
        LD      A,'K'
        CALL    PBYT
        CALL    CRLF
SAMRATS:        PRINT SINTV1
        LD      A,(SRATE)
        CALL    PBYT
        PRINT   MILSEC
        CALL    CRLF
        CALL    RTIME
        CALL    CRLF
        POPALL
        RET

;
BSZR:   PUSH    AF
        PUSH    BC
        LD      A,(BUFSIZ)
        LD      B,A
        XOR     A
```

```
BZLP:   ADD     A,B
        DAA
        DJNZ    BZLP
        LD      (BSZSAV),A
        POP     BC
        POP     AF
        RET
;
SRR:    PUSH    AF
        LD      A,(ADVAL)
        CP      1
        JR      NZ,N4MS
        STA     (SHIFTER),8
        STA     (SRATE),34H
        JR      ENDSRR
N4MS:   STA     (SHIFTER),7
        STA     (SRATE),38H
ENDSRR: POP     AF
        RET
;
SAVTIME:
        PUSHALL
        LD      BC,10
        LD      DE,TIMSAVE
        LD      HL,STRTTB
        LDIR
        POPALL
        RET
;
RESTIME:
        PUSHALL
        LD      BC,10
        LD      DE,STRTTB
        LD      HL,TIMSAVE
        LDIR
        POPALL
        RET

;
TIMNOW:
        DB      'TIME NOW:   ',0
;
```

```
;
RAMTST:
        PUSH    BC
        PUSH    HL
        CALL    ZCTR
        LD      BC,(ENDRAM)
RAM:    LD      HL,(RAMST)
FILL:   LD      A,L
        XOR     H
        XOR     B
        LD      (HL),A
        INC     HL
        LD      A,H
        CP      C
        JR      NZ,FILL
;
TESTE:  LD      HL,(RAMST)
;
TEST:   LD      A,L
        XOR     H
        XOR     B
        CP      (HL)
        JR      NZ,BADRAM
        INC     HL
        LD      A,H
        CP      C
        JR      NZ,TEST
        CALL    INCSCR
        INC     B
        JR      NZ,RAM
;
RAMOK:  XOR     A
        JR      RAMRET
;
RAMOKM:PRINT    OKRAM
        RET
;
BADRAM: LD      (ADDRESS),HL
        LD      (PATRN),A
        CALL    RAMER
RAMRET:
        POP     HL
        POP     BC
        RET
;
RAMER:  PRINT   RAMER1
        LD      HL,(ADDRESS)
        CALL    PHREG
        CALL    CRLF
        PRINT   RAMER2
        LD      A,(PATRN)
        CALL    HEXOUT
        CALL    CRLF
        PRINT   RAMER3
```

```
        CALL    GBYT
        CP      CR
        RET     Z
        JP      0000H
;
TESTTAP:
        PUSH    BC
        PUSH    HL
        LD      HL,CQCHEK
        LD      A,4DH
        CP      M
        JR      NZ,RAMER
        LD      BC,(ENDRAM)
        JR      TESTE
;
PHREG:
        LD      A,H             ;GET HIGH TWO DIGITS
        CALL    HEXOUT          ;PRINT THEM
        LD      A,L             ;GET LOW TWO DIGITS
HEXOUT: PUSH    AF              ;SAVE THE LOW DIGIT
        RRCA                    ;PUT HIGH NIBBLE INTO BITS 0-3
        RRCA
        RRCA
        RRCA
        CALL    PCD             ;PRINT SINGLE DIGIT
        POP     AF              ;GET THE LOW DIGIT
PCD:    AND     0FH             ;
        ADD     A,90H           ;
        DAA
        ADC     A,40H
        DAA
        JP      PBYT
;
DIVIDE: OR      A
        JR      NZ,OKTODV
        SCF
        RET
OKTODV: LD      C,A
        LD      B,16
        XOR     A
DLOOP:  SLA     E
        RL      D
        RL      A
        CP      C
        JR      C,NOINC
        INC     E
        SUB     C
NOINC:  DJNZ    DLOOP
        LD      B,A
NOINC2: RET
;
HTBCD:
        PUSH    BC
        PUSH    AF
        LD      B,M
```

```
        XOR     A
        cp      b
        JR      Z,HTRET
        XOR     A                       ;MAKE SURE ALL FLAGS CLEAR
BCDL:   INC     A
        DAA
        JR      C,HTRET
LT99:   DJNZ    BCDL
        LD      M,A
HTRET:  POP     AF
        POP     BC
        RET
;
;THIS SUBROUTINE PROVIDES A PAUSE BEFORE CLEARING SCREEN
CONT:   PRINT   CONTIN
        CALL    GBYTNE
        RET
;
;THIS SUBROUTINE SKIPS A RAMTEST
SKIPPER:        LD      HL,IBUFF
        LD      A,'N'
        CP      M
        RET
;
OK?:    CALL    CRLF
        PRINT   OKQ                     ;WAS IT OK?
        CALL    GBYT
        CP      'Y'
        RET
;
CLEAR:  PUSHALL
        LD      HL,CLRSCR
HD:     LD      A,M
        CP      8
        JR      Z,HDRET
        CALL    PBYT
        INC     HL
        JR      HD
HDRET:
        PRINT   CNOT
        CALL    CRLF
        POPALL
        RET
;
ETERM:
        IPRINT  (TERMINATED)
        CALL    CONT
        JP      ENDPRO
;
LOP2:   PUSH    AF
        LOAD    AF,IY
        CALL    PBYT
        PRINT   EQUALS
        POP     AF
        RET
```

```
;
TIMOUT:PUSHALL
       LOAD    DE,HL
       INC     HL
       CALL    PBCD
       LD      A,"/"
       CALL    PBYT
       INC     HL
       CALL    PBCD
       LD      A,"/"
       CALL    PBYT
       EX      DE,HL
       CALL    PBCD
       EX      DE,HL
       LD      A,' '
       CALL    PBYT
       INC     HL
       CALL    PBCD
       LD      A,':'
       CALL    PBYT
       INC     HL
       CALL    PBCD
       CALL    CRLF
       POPALL
       RET
;
RTIME:  PUSHALL
       PRINT   RTIMEP
       LD      DE,(MSAMPS)
       LDR     B,(SHIFTER)
SHLP:  SRA     D
       RR      E
       DJNZ    SHLP
       LD      A,60
       CALL    DIVIDE
       LD      HL,RECTIME
       LD      M,E
       INC     HL
       LD      M,B
       DEC     HL
       CALL    HTBCD
       CALL    PBCD
       PRINT   MINUTE
       INC     HL
       CALL    HTBCD
       CALL    PBCD
       PRINT   SECOND
       CALL    CRLF
       POPALL
       RET
;
MINALP: LD     A,(BUFSIZ)
       SLA     A
       OR      10H
       LD      B,A
```

```
        LD      A,(RECTIME+1)
        ADD     A,B
        DAA
        CP      59H
        LD      B,0
        JR      C,NOMIN
        INC     B
NOMIN:  INC     B
        LD      A,(RECTIME)
        ADD     A,B
        LD      B,A
        RET
;
;
INCSCR: PUSHALL
        LD      HL,GPCTRL
        LD      A,M
        INC     A
        DAA
        LD      M,A
        JR      NC,DISP
        INC     HL
        ADC     A,(HL)
        DAA
        LD      M,A
DISP:   LD      HL,GPCTRH
        CALL    PBCD
        DEC     HL
        CALL    PBCD
        LD      A,CR
        CALL    PBYT
        POPALL
        RET
;
PBCD:   PUSH    AF
        LD      A,30H
        RLD
        CALL    PBYT
        RLD
        CALL    PBYT
        RLD
        POP     AF
        RET
;
ZCTR:   PUSH    AF
        XOR     A
        LD      (GPCTRL),A
        LD      (GPCTRH),A
        POP     AF
        RET
;
BCDIN:  PUSH    BC
        LD      B,2
        LD      (HL),0
BCDLOP: LD      A,(DE)
```

```
        CALL    CONVAS
        JR      C,NVALIT
        RLD
        INC     DE
        DJNZ    BCDLOP
        LD      A,(DE)
        CALL    CONVAS
IOK:    INC     DE
        CCF
IBAD:   POP     BC
        RET
;
NVALIT: LD      A,B
        CP      2
        SCF
        JR      Z,IBAD
        JR      IOK
;
;
TIMINP: PUSHALL
        LD      (TSTORE),HL
REINT:  LD      HL,(TSTORE)
        PRINT   YMDHM
        PRINT   QUES
        CALL    GETLN
        LD      DE,IBUFF
        LD      A,(NCI)
        CP      12
        JR      C,REINT
        LD      B,5
TINLP:  CALL    BCDIN
        JR      C,REINT
        INC     HL
        DJNZ    TINLP
        LD      HL,(TSTORE)
        LD      A,83H
        CP      M
        JR      NC,REINT
        INC     HL
        LD      A,12H
        CP      M
        JR      C,REINT
        XOR     A
        CP      M
        JR      Z,REINT
        INC     HL
        LD      A,31H
        CP      M
        JR      C,REINT
        XOR     A
        CP      M
        JR      Z,REINT
        INC     HL
        LD      A,23H
        CP      M
```

```
        JR      C,REINT
        INC     HL
        LD      A,59H
        CP      M
        JR      C,REINT
        POPALL
        RET
;
;THIS SUBROUTINE CONVERTS ASCII TO BCD DIGIT,RETURNS WITH CARRY SET
;IF INVALID INPUT
CONVAS: SUB     30H
        RET     C               ;CARRY SET IF LESS THAN 30HEX
        CP      0AH             ;BETTER BE LESS THAN OR =9
        JR      NC,NOTVAL
        CCF                     ;COMPLEMENT CARRY(CLEAR IT)
        RET
NOTVAL: SCF
        RET
;
;
INDAT:  PRINT   QUES
        CALL    GETLN
        LD      DE,IBUFF
        CALL    BCDIN
        RET
;
LOP1:   PUSH    AF              ;
        LOAD    AF,IY
        CALL    PBYT
        CALL    CRLF
        POP     AF
        RET
;
CLR128: PUSH    AF
        PUSH    BC
        XOR     A
        LD      B,128
CLELOOP:
        LD      M,A
        INC     HL
        DJNZ    CLELOOP
        POP     BC
        POP     AF
        RET
;
;THIS SUBROUTINE CLEARS A 2K HEX BLOCK OF RAM
;CALL WITH HL SET TO BEGINNING OF RAM TO BE CLEARED
;NO REGS ALTERED
CLR2K:  PUSH    BC
        PUSH    HL
        LD      B,64
CLR2KL:
        CALL    CLR128
        DJNZ    CLR2KL
        POP     HL
```

```
        POP     BC
        RET
;
```

```
;
;
CONTIN: DB      'HIT ANY KEY TO CONTINUE',0
;
OKRAM:  DEFB    'THIS RAM IS OK'
        DEFB    CR
        DEFB    LF
        DEFB    0
;
RAMER1: DEFB    'BAD RAM @ HEX ADDRESS '
        DEFB    0
;
RAMER2: DEFB    'HEX PATTERN LOADED = '
        DEFB    0
;
RAMER3: DEFB    'HIT RETURN TO TEST NEXT RAM'
        DEFB    CR
        DEFB    LF
        DEFB    0
;
TRAM:   DEFB    'RAMTEST TAKES 256 ITERATIONS'
        DEFB    CR
        DEFB    LF
        DEFB    'ITERATION # IS SHOWN BELOW'
        DEFB    CR
        DEFB    LF
        DEFB    'Please wait'
        DEFB    CR
        DEFB    LF
        DEFB    0
;
OKNSC:  DEFB    'NSC RAM IS OK'
        DEFB    CR
        DEFB    LF
        DEFB    0
;
TZBUF:  DEFB    'TESTING DATA ACQUISITION BUFFER'
        DEFB    CR
        DEFB    LF
        DEFB    0
;
CNOT:   DB      '
        DB      '
        DB      '
        DB      0
;
CLRSCR:REPT     22
       DEFB     LF
       ENDM
       DEFB     CR
       DB       '                    U.S. GEOLOGICAL SURVEY 4-CHANNEL DATA RECORDER',CR,LF
       DB       8
       DB       CR
       REPT     13
       DEFB     LF
```

```
        ENDM
        DEFB    CR
        DEFB    0
;
TSCR:   DEFB    'TEST SCRATCH PAD RAM ',0
;
TSTB?:  DB      'TEST DATA BUFFER     ',0
;
QUES:   DB      ' ?      ',0
;
SINTV:  DB      'ENTER 4 or 8 ms.'
        DB      CR,LF
SINTV1:
        DB      'SAMPLING RATE  = '
        DEFB    0
;
THRSH:  DB      CR,LF,'ENTER 6, 12, 18, or 24 db.',CR,LF
THRSH1:
        DB      'THRESHOLD      = ',0
;
STA:    DB      CR,LF,'ENTER .05, .10, .25, or .50 SEC.',CR,LF
STA1:   DB      'STA TIME CONST = ',0
;
PESP:   DB      CR,LF
        DB      'ENTER 87.5, 75, 50, OR 25%',CR,LF
PESPP:  DB      'POST-EVENT SAMPLE (%)',0
;
DEPL:           DEFB    'DEPLOYMENT #    '
        DEFB    0
;
INSTR:          DEFB    'INSTRUMENT #    '
        DEFB    0
;
CHSCI:          DB      'CHIEF SCIENTIST '
        db      0
;
CRUISE: DB      'CRUISE #        '
        db      0
;
SPHERE: DB      'SPHERE #        '
        db      0
;
LAT:            DB      'LATITUDE        '
        db      0
;
LONG:           DB      'LONGITUDE       '
        db      0
;
FEG:    DB      'FRONT END GAIN'
        db      CR,LF,0
;
FED:    DB      'FRONT END DAMPING'
        db      CR,LF,0
;
CH1:            DB      'CHANNEL 1 '
```

```
        db      0
;
CH2:            DB      'CHANNEL 2 '
        db      0
;
CH3:            DB      'CHANNEL 3 '
        db      0
;
CH4:            DB      'CHANNEL 4 '
        db      0
;
OKQ:    DB      'IS THIS CORRECT (Y/N)',0
;
YMDHM:  DB      'YR/MTH/DAY/HR/MIN',0
;
ME:     DB      'TIMER OR EVENT MODE(T/E) ',0
;
NEXPP:  DB      '# OF RECORDS IN '
SN:     DB      'SERIES #',0
;
STFS:   DB      CR,LF,'START '
TFS:    DB      'TIME FOR SERIES #',0
;
STOP:   DB      'STOP  ',0
;
BSIZP:  DB      'ENTER 1,2, OR 4 blocks of 8K',CR,LF
BSIZP1:DB       'RECORD SIZE    = ',0
;
NCAP:   DB      CR,LF,'# OF CHANNELS (1-4)',0
;
BCHAP:  DB      'BASE CHANNEL  (1-4)',0
;
EQUALS:DB       '    = ',0
;
ACHAN:  DB      'Active Channel(s) = ',0
;
WPRMT:  DB      'Timer Mode',cr,lf,0
;
EVENTP: DB      'Event Mode',cr,lf,0
;
WOFF:   DB      CR,LF
        DB      'WINDOW OFFSET    ',0
WOFF1:  DB      '(0-59 sec.)',0
;
WPER:   DB      'PERIOD of RECORDS',0
woer1:  DB      '(',0
WPER2:  DB      '-99 min.)',0
;
MINUTE: DB      ' min. ',0
;
SECOND: DB      ' sec.',0
;
MILSEC: DB      ' ms.',0
;
DBP:    DB      ' db',0
```

```
;
RTIMEP: DB      'RECORD TIME  = ',0
;
MAX:    DB      'MAXIMUM ',0
;
SETM:   DB      'ENTER CURRENT TIME + 1 MINUTE',CR,LF,0
;
SATC:   DB      'SATELLITE CLOCK CONNECTED',CR,LF,0
;
SATNC:  DB      'SATELLITE CLOCK NOT CONNECTED',CR,LF,0
;
CLKPRM: DB      'HIT RETURN WHEN LESS THAN ONE MINUTE TO GO',CR,LF,0
;
BADPR:  DB      'BAD PROCESSOR',CR,LF,0
;
INTAPE: DB      'INSERT TAPE CARTRIDGE',CR,LF,0
;
REMTAPE:        DB      'REMOVE TAPE CARTRIDGE'
CRLFM:          DB      CR,LF,0
;
UNPLUGM:        DB      'UNPLUG TERMINAL FROM OBIP',CR,LF
                DB      'HEADER WILL BE WRITTEN ABOUT 5 SEC. LATER'
        DB      0
BEEP:   DB      7,0
;
BLEEP:  DB      7,1,1,1,7,0
```

```
;
;           SEND A MESSAGE TO THE TERMINAL
;     (THIS SUBROUTINE MUST BE HERE TO AVOID PROBLEMS WITH MACROS)
;
SNDMES:
        PUSH    AF
MESSND: LD      A,(HL)          ; Get byte at HL.
        CP      0               ; Test for terminater.
        JP      Z,SNDRET        ; Done if 0
        CALL    PBYT            ; Else print character.
        INC     HL              ; Point at next.
        JR      MESSND          ; and continue.
SNDRET: POP     AF
        RET
;
;THIS SUBROUTINE IS PART OF WPRINT MACRO
WPRA:   CALL    SNDMES  ;PRINT PARTICULAR MESSAGE
        LD      HL,QUES ;NOW GENERAL QUESTION
        CALL    SNDMES          ;IS PRINTED
        RET
;
;THIS SUBROUTINE IS PART OF WPRINT MACRO
WPRB:   LD      DE,(HDRBUF)
        CALL    MOVLP           ;MOVE QUESTION TO BUFFER
        CALL    GETLN           ;GET RESPONSE
        CALL    RMOV            ;PUT THAT IN BUFFER
        RET
;
;       WARNING: THESE SUBROUTINES ARE PART OF MACROS-REGISTERS ARE PRESERVED
;       IN MACROS,NOT IN SUBROUTINES!
RMOV:   LD      HL,IBUFF
        LD      B,0
        LD      A,(NCI)         ;GET NO. OF CHARACTERS
        LD      C,A             ;PUT IN C
        LD      DE,(HDRBUF)     ;GET WHERE IT'S GOING
        LDIR                    ;MOVE IT
        XOR     A               ;ZERO
INS:    LD      (DE),A          ;INSERT TERMINATOR
        INC     DE              ;ACCOUNT FOR INSERTION
        LD      (HDRBUF),DE     ;STORE POINTER TO HEADER
        RET
;
MOVLP:  LDI
        XOR     A
        CP      M
        JR      NZ,MOVLP
        LD      (HDRBUF),DE
        RET
;
CRLF:   PUSH    HL
        LD      HL,CRLFM
        CALL    SNDMES
        POP     HL
```

```
        RET
;*****************************************************************************
;DELAY SUBROUTINES
;
;*****************************************************************************
;
DELAYR:CALL     D87T            ;17T--THIS CALL INCLUDED IN 87T TIME
       DJNZ     DELAYR          ;13T--THIS+87=100 PER LOOP
       POP      BC              ;10T--WASTE TIME.NOTE THAT WE HAVE 8 WHEN WE
;                               FALL THROUGH
       PUSH     BC              ;11T--RESTORE
       NOP                      ;4T
       NOP                      ;A TOTAL OF 8+21+16+10=55 TSTATES
       NOP
       NOP
       RET                      ;10T--RETURN.SEE MACROS FOR MORE DOCUMENTATION
;
;*****************************************************************************
;
;       50 TSTATE DELAY
;CALL TO COME HERE=17 TSTATES
;
D50T:   EX      AF,AF'          ; 4T--SWITCH REGS SO AS TO SAVE FLAGS
        AND     0FFH            ; 7T--THIS ONLY SCREWS F'
        EX      AF,AF'          ; 4T--RESTORE FLAGS
        NOP                     ; 4T--WASTE TIME
        NOP                     ; 4T--4+4+4+4+7+10+CALL TO COME(17)=50
        RET                     ;10T
;
;*****************************************************************************
;
;       87 TSTATE DELAY
;
;CALL TO COME HERE=17 T
;
D87T:   JP      D77T            ;10T--WASTE TIME
D77T:   JP      D67T            ;10T--HERE DOWN=77 INCLUDING THE CALL IN
D67T:   JP      D57T            ;10T--AND SO ON
D57T:   JP      D47T
D47T:   JP      D37T
D37T:   JP      D27T
D27T:   RET                     ;10T--NOTE THAT THIS FITS WELL WITH DJNZ WHICH
;                               IS 13T
;
;*****************************************************************************
;
;       100 TSTATE DELAY
;
;CALL TO COME HERE=17 T
;
D100T:  CALL    D50T            ;THIS CALL IS INCLUDED IN 50T'S TIMING
        EX      AF,AF'          ;SAME TRICKS AS IN 50T
        AND     0FFH
        EX      AF,AF'
        NOP
```

```
        NOP
        RET
;
;
; ---- GET BYTE AND ECHO IF ACC =0 ROUTINE ----
;
;*************************************************************************
;
;       GET SET UP
;
GBYTNE: LD      A,1     ; NO ECHO
        JR      NE              ; JUMP OVER ECHO
GBYT:   LD      A,0             ; ECHO ENTRY
NE:     PUSH    BC              ; NO ECHO ENTRY
        PUSH    AF              ; SAVE FLAGS&ACC
FRAMERR:LD      B,8             ; Set index for number of bits to input.
;
;*************************************************************************
;
;       WE MUST DETECT THE BEGINNING OF THE START BIT-SO THE FIRST STEP IS TO
;       MAKE SURE THAT IT HAS NOT BEGUN!
;
MARK:   LD      A,(PBDATA)      ; GET INPUT AND MAKE SURE IT IS A MARK
        RLA                     ; INPUT TO CARRY(INPUT IS INVERTED)
        JP      NC,MARK         ; WAIT TILL MARK
;
;       NOW GET TRANSITION
;
GTSTRT: LD      A,(PBDATA)      ;13T-- Look for start bit.
        RLA                     ; 4T-- Move to carry flag.
        JP      C,GTSTRT        ;10T-- IF  start bit not present
                                ; THEN go back and look again.
;
;       MAXIMUM ERROR IN START BIT TIME IS 27 T-STATES
;
;*************************************************************************
;
;       FOR THIS SYSTEM THE CLOCK IS 1.0486*10^6
;       AND THE NO. OF T-STATES IN ONE BIT TIME FOR COMMON BAUD RATES ARE:
;
;       9600            109.23 T-STATES
;       4800            218.46
;       2400            436.92
;       1200            873.83
;        300           3495.30
;
;       FOR NOW WE ARE RUNNING AT 1200 BAUD, FIXED
;       1/2 BIT TIME = 437 TSTATES FOR ALL INTENTS AND PURPOSES
;
;*************************************************************************
;*************************************************************************
;
;       WE NOW HAVE A HYPOTHETICAL START BIT
;
;       SO NOW WAIT 1/2 BIT TIME AND SEE IF IT IS STILL THERE
```

```
;
;       TIME NOW=TRANSITION+14 T
        DELAY   4               ; MACRO TO DELAY 400 TSTATES
        JP      CHSTRT          ;10T
;       TIME NOW=TRANS.+424T(13T TO READ LINE MAKES 437)
CHSTRT: LD      A,(PBDATA)      ;13T-- Look for start bit again.
        RLA                     ; 4T-- Move it to carry flag.
        JP      C,GTSTRT        ;10T-- IF   start bit is gone, TRY AGAIN
;
;*******************************************************************************
;
;       PREPARE TO ECHO START BIT
;
;       TIME IS 451 TSTATES AFTER START BIT EDGE DETECTED
;
;       WE WILL NOW NEED TO DELAY ABOUT 1 BIT TIME SO THAT WE CONTINUE TO HIT
;       THE MIDDLE OF EACH BIT AND SO THAT THE BITS WE ECHO ARE THE RIGHT
;       LENGTH.
;
        LD      A,1             ; 7T-- Prepare A as start bit
        CALL    D50T            ;50T-- DELAY 50T STATES
        LD      (PBSETB),A      ;13T-- and send it.(NO LONGER)
        CALL    D27T            ;27T-- WAIT 27 T
;
;*******************************************************************************
;
;       TIME IS NOW 111 T-STATES AFTER MIDDLE OF BIT
;       763 TO GO BEFORE WE READ NEXT,847 (IDEALLY) BEFORE NEXT ECHO
;
;*******************************************************************************
XMIT:   DELAY   7               ; Delay 700 T-STATES
        CALL    D50T            ; DELAY 50 MORE
;763-750=13 WHICH IS HOW LONG IT TAKES TO READ
        LD      A,(PBDATA)      ;13T Get receive bit 7 in A.
;                               ;84T TO ECHO
        RLCA                    ; 4T Rotate into carry flag AND BIT 0
        RR      C               ; 8T then into C.
        AND     1               ; 7T Mask other bits AND SET FLAGS FOR TEST
        LD      A,1             ; 7T Prepare A to set or reset line.
;                               ;FLAGS UNALTERED FROM THE AND ABOVE
        NOP                     ; 8 MORE TO ADD
        NOP
;50 T-STATES TO GO .LESS 13T TO DO IT AND 10T FOR THE JUMP LEAVES 27
        CALL    D27T            ; 27T SO DELAY IT
        JP      Z,GTCLRB        ; 10T TEST FLAG AND GOTO APPROPIATE ECHO
;
;*******************************************************************************
;       THESE ROUTINES NO LONGER DO AN ECHO--LEFT HERE FOR TIMING
;       (IF ECHO DESIRED PLEASE RESTORE PBCLRB WHERE APPROPIATE)
;
;
;       ECHO A ONE              NOTE THAT BOTH INPUT AND OUTPUT ARE INVERTED
;                               SO THAT THIS IS REALLY A ZERO
;
GTSETB: LD      (PBSETB),A      ; 13T SET OUTPUT HIGH
```

```
        JP      GTREST          ; 10T AND CONTINUE,KEEPING TIMING THE SAME
;
;       23 T-STATES FOR THIS BRANCH
;*******************************************************************************
;
;       THESE ROUTINES NO LONGER DO AN ECHO--LEFT HERE FOR TIMING
;       (IF ECHO DESIRED PLEASE RESTORE PBCLRB WHERE APPROPIATE)
;
GTCLRB: LD      (PBSETB),A      ; 13T SET OUTPUT LOW
        JP      GTREST          ; 10T and keep timing constant.
;
;*******************************************************************************
;
GTREST: NOP                     ; 4T--DELAY
        DJNZ    XMIT            ;13T Go till all bits in.
;
;TIMING:SINCE LAST READ=4+8+7+7+8+27+10+13+10+4+13=111
;       SINCE LAST WRITE=10+4+13=27
;       NOTE TIMING PRESERVED THROUGH XMIT LOOP(SEE VALUES AT START OF LOOP)
;
                                ; 8T WHEN WE FALL THROUGH
;TIMING:SINCE LAST READ=111-13+8=106
;       SINCE LAST WRITE=27-13+8=22
;
;*******************************************************************************
;
        DELAY   7
        CALL    D47T
        NOP
        NOP
;
;*******************************************************************************
;TIMING=755+106=861 SINCE LAST READ(+13 FOR READ=874)
;TIMING=755+22=777 SINCE LAST WRITE
;*******************************************************************************
;
        LD      A,(PBDATA)      ;13T Get receive bit 7 in A.
        RLA                     ; 4T TO CARRY
        JP      NC,FRAMERR      ;10T FRAMING ERROR IF NOT A STOP BIT
;
;*******************************************************************************
;
;TIMING=777+27=804 SINCE LAST WRITE
;
;*******************************************************************************
;
        CALL    D50T            ;50T
        LD      A,1             ; 7T Prepare A as stop bit.
        LD      (PBSETB),A      ;13T Echo stop bit AT 874T
;
;*******************************************************************************
        DELAY   9               ;TIMING NO LONGER CRITICAL-IT JUST MUST BE
                                ;LONG ENOUGH
        LD      A,C             ; Transfer assembled byte to A
;*******************************************************************************
```

```
;
        CP      6               ;
        JP      Z,BADRET        ;
        POP     AF              ; RESTORE FLAGS&ACC
        PUSH    AF              ; PROTECT FLAGS, RETRIEVE ACC
        CP      0               ; ECHO REQUIRED IF ZERO
        JP      Z,ECHO          ;

ERET:   POP     AF              ; NO ECHO NEEDED
        LD      A,C             ; Transfer assembled byte to A
        POP     BC              ; RESTORE BC
        RET                     ; and return to caller.
;*******************************************************************************
BADRET: POP     AF
        POP     BC
        JP      XMRZ
;
;*******************************************************************************
ECHO:   LD      A,C             ;GET CHARACTER
        CP      BS              ;BACKSPACE?
        JP      Z,BACK          ;DO BACKSPACE ROUTINE
        CP      DEL             ;DEL=BS FOR US
        JP      Z,BACK          ;
        CP      CR              ;CARRIAGE RETURN?
        JP      Z,CRLFR         ;DO CR,LF ECHO BUT KEEP CR
        CP      SPC
        JP      C,ERET          ;DO NOT ECHO OTHER CONTROL CODES
        CALL    PBYT            ;NORMAL ECHO
        JP      ERET
;*******************************************************************************
CRLFR:  CALL    CRLF            ;CRLF ROUTINE IN MONITOR
        JP      ERET
;*******************************************************************************
BACK:   LD      C,BS            ; MAKE SURE DEL & BS SAME
        JP      ERET            ; LET CALLER DECIDE WHAT TO DO(NO ECHO)
;*******************************************************************************
GETLN:  PUSHALL
GLN1:   LD      HL,IBUFF        ;ASCII INPUT BUFFER
        LD      C,0             ;NO. OF CHARACTERS
        LD      B,MAXB          ;MAX BUFFER LENGTH
GLN2:   CALL    GBYT            ;GET CHARACTER
        CP      BS              ;BACKSPACE?
        JP      NZ,NOBS         ;NO
        LD      A,B
        CP      MAXB
        JP      Z,GLN1          ;IGNORE AT BEGINNING
        CALL    BSPCR
        INC     B               ;RESTORE CHARACTER COUNT
        DEC     C               ;NO. OF CHARACTERS
        DEC     HL
        JR      GLN2            ;TRY AGAIN
NOBS:   CP      CTLX            ;CONTROL X
        JP      Z,CTLXR         ;DO IT IF SO
        CP      CTLU            ;CONTROL U
        JP      Z,CTLXR
```

```
        CP      CR
        JR      Z,NOTST
        CP      SPC
        JR      C,GLN2          ;NO OTHER CONTROL CHARACTERS RECOGNIZED
NOTST:  LD      (HL),A          ;STORE CHAR
        INC     HL
        INC     C
        CP      CR
        JP      Z,GLEND         ;DONE SO EXIT PROPERLY
        DJNZ    GLN2
        JP      CTLXR           ;INPUT TOO LONG,MUST BE ERROR
;*****************************************************************************
;
glend:  LD      (HL),LF         ;ADD LINE FEED
        INC     C               ;INC CHARACTER COUNT
        LD      A,C             ;SAVE CHARACTER COUNT
        LD      (NCI),A         ;
        POPALL
        RET
;*****************************************************************************
;
CTLXR:  LD      A,C             ;GET CHARACTER COUNT
        CP      0               ;MAKE SURE NOT ZERO
        JR      Z,GLN1          ;NO ACTION NEEDED
        LD      B,C
BLOOP:  CALL    BSPCR           ;CLEAN UP SCREEN
        DJNZ    BLOOP
        JR      GLN1
;*****************************************************************************

BSPCR:  LD      A,BS            ;BS/SPC/BS
        CALL    PBYT
        LD      A,SPC
        CALL    PBYT
        LD      A,BS
        CALL    PBYT
        RET
;
        .PA
;*****************************************************************************
;       SEND CHARARACTER TO TERMINAL
;*****************************************************************************
;       GET READY
;
PBYT:   PUSH    AF
        PUSH    BC
        LD      C,A             ; 4T Move output byte to C register.
        LD      B,8             ; Word length
        LD      A,1             ; Prepare A to send bits,
;
;*****************************************************************************
;       SEND START BIT
;
        LD      (PBCLRB),A      ; and send start bit
;
```

```
;*************************************************************************
;
;       TIMING=874 T-STATES TO END OF NEXT SEND
;
;*************************************************************************
;
        NOP                     ; 4T
        NOP                     ; 4T
        NOP                     ; 4T
        NOP                     ; 4T
        CALL    D27T            ;27T
        DELAY   8               ;800T
;
;*************************************************************************
;               TIMING=874-843=31 TO SEND
;*************************************************************************
;
PUTLP:  RRC     C               ; 8T--PUT BIT INTO CARRY
        JP      NC,PTCLRB       ;10T--IF A ZERO THERE IS NO CARRY.REST IS LIKE
                                ; 6BYT ABOVE
;
;*************************************************************************
;       FROM PUTLP TO SEND=10+13=31
PTSETB: LD      (PBSETB),A      ;13T
        JP      PUTRST          ;10T
;       TIMING=23T FOR THIS BRANCH
;*************************************************************************
;       FROM PUTLP TO SEND=10+13=31
PTCLRB: LD      (PBCLRB),A      ;13T
        JP      PUTRST          ;10T
;       TIMING=23 T FOR THIS BRANCH
;*************************************************************************
;       TIMING=874-10=864 T-STATES TO SEND
PUTRST: DELAY   8               ;800T
        JP      DLY1            ;10T
DLY1:   JP      DLY2            ;10T
DLY2:   DJNZ    PUTLP           ;13T LOOP TILL DONE
;       TIMING=864-833=31       TIMING PRESERVED IN LOOP
                                ; 8T FALLING THROUGH
;*************************************************************************
;       TIMING=31+13-8=36
        AND     0FFH            ; 7T    CLEAR CARRY,DELAY
        NOP                     ; 4T    DELAY
        NOP                     ; 4T    DELAY
        NOP                     ; 4T    DELAY
        NOP                     ; 4T    DELAY
;TIMING=36-23=13         JUST TIME TO DO IT
        LD      (PBSETB),A      ;13T    STOPBIT
        DELAY   9               ;TIMING NOT CRITICAL NOW
        POP     BC
        POP     AF
        RET                     ; and return to caller.
```

```
;
COMMAND MACRO   COMVAL
        PUSH    AF
        LD      A,COMVAL
        CALL    SCOMMAND
        POP     AF
        ENDM
;THIS SUBROUTINE TURNS OFF POWER TO CARTRIDGE RECORDER
;
OFFCART:        PUSH    AF              ;SAVE FLAGS &ACC
                LD      A,OFF24V        ;TURN OFF 24VOLTS
                OUT     (PWRPRT),A      ;DO IT
                DELAY   256             ;25.6 MILLI-SEC DELAY
                LD      A,OFF5V         ;NOW KILL THE 5 VOLTS
                OUT     (PWRPRT),A      ;DO IT
                LD      A,RSTCC         ;SET UP FOR CARTRIDGE CONTROLLER RESET
                LD      (PBCLRB),A      ;LOW=RESET,HIGH=RUN
                POP     AF              ;RESTORE FLAGS&ACC
                RET                     ;DONE,SO GO HOME
;POWER TO CARTRIDGE DRIVE IS OFF,RESET LINE TO CART. CONTR. IS RESET
;*****************************************************************************
;THIS SUBROUTINE TURNS ON POWER TO CARTRIDGE RECORDER
;
ONCART: PUSH    AF              ;SAVE FLAGS&ACC
                LD      A,ON5V          ;TURN ON CONTROLLER FIRST
                OUT     (PWRPRT),A      ;DO IT
                LD      A,RSTCC         ;SET UP FOR CARTRIDGE CONTROLLER RESET
                LD      (PBCLRB),A      ;LOW=RESET,HIGH=RUN
                delay   30              ;3 ms delay
                LD      (PBSETB),A      ;CARTRIDGE CONTROLLER RESET
                DELAY   100             ;10 MILLI-SEC DELAY
                LD      A,ON24V         ;NOW 24 VOLTS(OR 12)
                OUT     (PWRPRT),A      ;DO IT
                DELAY   256             ;STABILIZE DRIVE
        ;RESET CONTROLLER AGAIN IN CASE DRIVE CAME UP SCREWY
                LD      A,RSTCC         ;SET UP FOR CARTRIDGE CONTROLLER RESET
                LD      (PBCLRB),A      ;LOW=RESET,HIGH=RUN
                delay   10              ;1 ms delay
                LD      (PBSETB),A      ;CARTRIDGE CONTROLLER RESET
                DELAY   256             ;STABILIZE EVERYTHING
                LD      B,0FFH
NOTREADY:
                COMMAND CURSTATC
                DELAY   20
                LD      A,(IS)
                AND     COMSTATM
                JR      Z,READY
                DJNZ    NOTREADY
READY:          POP     AF              ;RESTORE FLAGS&ACC
                RET                     ;DONE ,GO HOME
;CARTRIDGE RECORDER HAS POWER,IS RESET
;*****************************************************************************
;
BTAPE:  CALL    ONCART
        STA     (IMA),MAN
```

```
        CALL    BLANKB16
BTAP1:  COMMAND CURSTATC
        LD      A,(IS)
        CP      92H
        JR      Z,NOTAPE
        PRINT   REMTAPE
        JR      BTAP1
NOTAPE: PRINT   INTAPE
        COMMAND CURSTATC
        LD      A,(IS)
        CP      92H
        JR      Z,NOTAPE
        LD      A,(DS)
        CP      99H
        JR      Z,BTAP2
        AND     1
        JR      NZ,BTAP1
        IPRINT  (WRITE PROTECTED)
        JR      NOTAPE
BTAP2:  CALL    OFFCART         ;SO SAVE THE POORLY DESIGNED REGULATORS
        WPRINT  TSTTAPM
        CALL    SKIPPER
        RET     Z
TSTTAP:
        LD      BC,0FFA0H
        LD      (ENDRAM),BC
        LD      HL,8000H
        LD      (RAMST),HL
        CALL    RAMTST
        XOR     A
        LD      (WBSTART),A
        STA     (DATABLOCK),40H
        LD      (WBUFSAV),HL
        CALL    ONCART
        CALL    WRITED
        LD      HL,8000H
        CALL    CLR2K
        XOR     A
        OUT     (PA),A
        LD      (RDBUF),HL
        COMMAND REVSPCRECC
        CALL    READIT
        CALL    TESTTAP
        JR      Z,COK
        PRINT   NOCOMPM
        JR      TSTTAP
COK:    PRINT   COMPSM
        CALL    OFFCART
        CALL    CONT
        RET
TAPECHK:
        PUSHALL
        LD      HL,0
        LD      DE,XAG
        LD      C,1FH
```

```
XL_P:   XOR     A
        LD      B,A
ADDX:   ADD     A,M
        INC     HL
        DJNZ    ADDX
        EX      DE,HL
        CP      M
        JP      NZ,STAMB
        EX      DE,HL
        INC     DE
        DEC     C
        JR      NZ,XL_P
        POPALL
        RET
W_EOD:
        CALL    ONCART
        CALL    SENDMA
        COMMAND WRITEFMC
        LD      A,(IS)
        AND     COMSTATM
        JR      Z,W_EOD1
        LD      HL,W_EOD
        PUSH    HL
        JP      ABORT
W_EOD1: COMMAND WRITEFMC
        LD      A,(IS)
        AND     COMSTATM
        JR      Z,W_EOD2
        LD      HL,W_EOD1
        PUSH    HL
        JP      ABORT
W_EOD2: CALL    OFFCART
        CALL    TAPECHK
        RET
WBPHDR:
        XOR     A
        LD      (WBSTART),A
        STA     (DATABLOCK),40H-2
        LD      HL,HDRRAM
        LD      (WBUFSAV),HL
        CALL    BLANKB16
        LD      HL,HEADER
        CALL    MNAME
        CALL    ONCART
        CALL    WRITED
        CALL    OFFCART
        CALL    BLANKB16
        LD      HL,BDTAHDR
        CALL    MNAME
;CLEAR RAM
        LD      HL,HDRRAM
        CALL    CLR2K
        RET
WDR:    CALL    ONCART
        CALL    TAPECHK
```

```
        CALL    NOBLOCKS
        LD      A,(BSTART)
        LD      (WBSTART),A
        CALL    FIXNOS
        CALL    WRITED
        CALL    OFFCART
        RET
;
SCOMMAND:       OUT     (CA),A
        CALL    READSTAT
GETDSIS:        PUSH    AF
        IN      A,(DSIS)
        LD      (DS),A
        CALL    READSTAT
        IN      A,(DSIS)
        LD      (IS),A
        POP     AF
        RET
;
READSTAT:       IN      A,(PS)
                RRCA
                RRCA
                JR      NC,READSTAT
                RET
;
WRITESTAT:      IN      A,(PS)
        RRCA
        JR      NC,WRITESTAT
        RET
;
READIT: CALL    SENDMA
        COMMAND READC
        LD      A,(IS)
        CP      0C0H
        JP      NZ,ABORT
        LD      C,DSIS
        CALL    READSTAT
        IN      E,(C)
        CALL    READSTAT
        IN      D,(C)
        LD      HL,TIPBUF
READTIP:
        DEC     DE
        CALL    READSTAT
        INI
        BIT     4,L
        JR      NZ,READTIP
        LD      HL,(RDBUF)
READLP:
        DEC     DE
        BIT     7,D
        JR      NZ,DONEOK
        CALL    READSTAT
        INI
        JR      READLP
```

```
DONEOK: RET
;
WBLOCK: PUSH    AF
        PUSH    BC
        LD      C,DA
WLOOP:  CALL    WRITESTAT
        OUTI
        JR      Z,WEND
        LD      A,(WBSTART)
        OR      H
        LD      H,A
        JR      WLOOP
WEND:   POP     BC
        POP     AF
        RET
;
WRITE16:PUSHALL
        LD      HL,BUF16
        LD      B,16
        LD      A,(WBSTART)
        STA     (WBSTART),00
        CALL    WBLOCK
        LD      (WBSTART),A
        POPALL
        RET
;
WRITE128:       PUSHALL
        LD      HL,(WBUFSAV)
        LD      B,128
        CALL    WBLOCK
        LD      (WBUFSAV),HL
        POPALL
        RET
;
BLANKB16:       PUSHALL
        LD      BC,16
        LD      HL,BLANK16
        LD      DE,BUF16
        LDIR
        POPALL
        RET
;
MNAME:  PUSHALL
        LD      DE,B16NAME
        LD      BC,8
        LDIR
        POPALL
        RET
;
FIXNOS: PUSHALL
        LD      A,(CSN)
        OR      30H
        LD      (BSERNO),A
        LD      HL,CEXPN
        LD      A,30H
```

```
        RLD
        LD      (BEXPNH),A
        RLD
        LD      (BEXPNL),A
        RLD                     ;RESTORE CEXPN
        POPALL
        RET
;
WRITEPAR:       PUSHALL
        LD      HL,SERBUF
        LD      (WBUFSAV),HL
        LD      A,(WBSTART)
        STA     (WBSTART),0
        LD      B,2
WRPARA: CALL    WRITE128
        DJNZ    WRPARA
        LD      (WBSTART),A
        POPALL
        RET
;
WRITED:
        CALL    SENDMA
        LD      A,(DATABLOCK)
        LD      (ESECT),A
        LD      HL,(WBUFSAV)
        LD      (EBUFSAV),HL
        CP      41H
        JR      C,LASTBLOCK
        SUB     40H
        LD      (DATABLOCK),A
        LD      B,40H
        CALL    RECSIZ
        CALL    WRITE16
BWR:    CALL    WRITE128
        DJNZ    BWR
        CALL    WRITEIT
        JR      WRITED
;
LASTBLOCK:      LD      B,A
        CALL    RECSIZ
        CALL    SETLDF
        CALL    WRITE16
LDB:    CALL    WRITE128
        DJNZ    LDB
        CP      40H
        JR      Z,NPB
        ADD     A,2
        CP      40H
        CALL    NZ,DBPXM        ;ERROR
        CALL    WRITEPAR
NPB:    CALL    WRITEIT
        RET
;
NOBLOCKS:
        PUSH    AF
```

```
        PUSH    BC
        LD      A,(BUFSIZ)
        LD      B,A
        XOR     A
ADDREC: ADD     A,40H
        DJNZ    ADDREC
        SUB     2
        LD      (DATABLOCK),A
        POP     BC
        POP     AF
        RET
;
;THIS SUBROUTINE SENDS THE MA
SENDMA: PUSH    AF
        LD      A,(IMA)
        OUT     (MA),A
        POP     AF
        RET
;
RECSIZ: PUSH    AF
        XOR     A
        LD      (B16LB),A
;       LD      A,B
        LD      A,40H
        LD      (B16RC),A
        POP     AF
        RET
;
SETLDF: STA     (B16LB),1
        RET
WRITEIT:
        COMMAND WRITEWCHC
BOTFIX: LD      A,(IS)
        AND     COMSTATM
        JR      NZ,WABORT
        LD      A,(DS)
        AND     BOTM
        JR      Z,NOTBOT
        COMMAND BWRITEC
        JR      BOTFIX
NOTBOT: CALL    EOTCH
        COMMAND CURSTATC
        CALL    EOTCH
        RET
;
WABORT: LD      HL,WRECOVER
        PUSH    HL
;
ABORT:  PRINT   ABORTM
        LD      A,(IS)
        AND     0FH
        SLA     A
        LD      C,A
        LD      B,0
        LD      HL,JTABLEB
```

```
        ADD     HL,BC
        LD      E,M
        INC     HL
        LD      D,M
        EX      DE,HL
        JP      (HL)
;
WRECOVER:
        CALL    OFFCART
        CALL    ONCART
        LD      A,(ESECT)
        LD      (DATABLOCK),A
        LD      HL,(EBUFSAV)
        LD      (WBUFSAV),HL
        POP     HL
        JP      WRITED
;
JTABLEB:
        DW      FLAG
        DW      PROTECTED
        DW      NODRIVE
        DW      NORESP
        DW      UNKNOWN
        DW      UNKNOWN
        DW      FMVER
        DW      TABORT
        DW      RFHER
        DW      RFCRCC
        DW      RFSHORT
        DW      RFBVP
        DW      WFRAW
        DW      WF
        DW      RFFMD
        DW      UNKNOWN
;
FLAG:
        IPRINT  (CODE 0 FLAG COND.)
        RET
;
PROTECTED:
        IPRINT  (CODE 1 WRITE PROTECTED)
        RET
;
NODRIVE:
        IPRINT  (CODE 2 NO DRIVE OR TAPE)
        RET
;
NORESP:
        IPRINT  (CODE 3 DRIVE DID NOT DO IT)
        RET
;
FMVER:
        IPRINT  (CODE 6 FILE MARK VER. ERR.)
        RET
;
```

```
TABORT:
        CALL    EOTCH
        JR      Z,NOEOT
        LD      A,(IMA)
        AND     TRACKMA
        OR      30H
        PRINT   TRACKNM
        CALL    PBYT
        CALL    CRLF
        RET
NOEOT:  IPRINT  (CODE 7 ABORT BEFORE DONE)
        RET
;
RFHER:
        IPRINT  (CODE 8 H-E-R)
        RET
;
RFCRCC:
        IPRINT  (CODE 9 BAD CRCC )
        RET
;
RFSHORT:
        IPRINT  (CODE 10 SHORT REC.)
        RET
;
RFBVP:
        IPRINT  (CODE 11 BAD V. PAR.)
        RET
;
WFRAW:
        IPRINT  (CODE 12 R-A-W ERROR)
        RET
;
WF:
        IPRINT  (CODE 13 WRITE FAIL)
        RET
;
RFFMD:
        IPRINT  (CODE 14 FILE MARK DET.)
        RET
;
UNKNOWN:
        IPRINT  (CODES 4,5,15 IMPROPER ABORT CODES)
        RET
;
EOTCH:  LD      A,(DS)
        AND     EOTM
        RET     Z
        CALL    EOT
        OR      0FFH            ;SET FLAG
        RET
;
EOT:    PUSH    AF
        LD      A,(IMA)
        AND     TRACKMA
```

```
        INC     A
        CP      4
        JR      Z,END_OF_TAPE
        OR      MAN
        LD      (IMA),A
DO_IT_AGAIN:
        CALL    SENDMA
        COMMAND BWRITEFMC
        LD      A,(IS)
        AND     COMSTATM
        JR      Z,EOT_DONE
        CALL    OFFCART
        CALL    ONCART
        JR      DO_IT_AGAIN
EOT_DONE:
        POP     AF
        RET
;
END_OF_TAPE:
        CALL    CRLF
        IPRINT  (OUT OF TAPE)
        JP      ENDPRO

;
BLANK16:        DB      0,20H,20H,20H,20H,20H,20H,20H,20H
                DB      20H,20H,20H,0,0,0,0
HEADER:         DB      'GPHEADER'
BDTAHDR:        DB      'S0EXPN00'
ABORTM:         DB      'ABORT ',0
NOCOMPM:        DB      'BAD TAPE OR DRIVE',CR,LF,0
TSTTAPM:        DB      'TEST TAPE ',0
COMPSM:         DB      'TAPE DRIVE OK',CR,LF,0
TRACKNM:        DB      CR,LF
                DB      'TRACK #',0
```

```
;
CONTROL:
        LOAD    AF,IY
        AND     0FH
        LD      (LSN),A
        LD      A,1
        LD      (CSN),A
        OR      30H
        LOAD    IY,AF
        LD      HL,SERBUF
        LD      (SERPTR),HL
;
NEWSER: DI
        LD      A,1
        OUT     (INTPRT),A
        LD      (STOPT1),A
        EI
        STA     (CEXPN),1
        CALL    LDSER
        CALL    SPITP
        LD      A,(SERTYP)
        CP      'E'
        JR      Z,EVENTX
        CP      'T'
        JR      Z,WINDOW
        JP      ENDPRO
;
EVENTX: CALL    INCSER
        JR      NEWSER
;
WINDOW: CALL    INITW
        PRINT   EXPM
        LD      HL,CEXPN
        CALL    PBCD
        PRINT   TM
        LD      A,(IMA)
        AND     TRACKMA
        OR      30H
        CALL    PBYT
        LD      A,CR
        CALL    PBYT
        CALL    TCHECK
        CALL    DOIT
        CALL    NEXTW
        JR      NC,WINDOW       ;
        CALL    INCSER
        JR      NEWSER
;
LDSER:
        PUSHALL
        LD      HL,(SERPTR)
        LD      BC,NOPARA
        LD      DE,NSCRAM
        LDIR
```

```
        LD      (SERPTR),HL
        POPALL
        RET
;
INITW:  XOR     A
        LD      L,A             ;
        LD      A,(BSTART)
        LD      H,A
        LD      (BUFPTR),HL
        LD      (WBUFSAV),HL    ;
        LD      HL,(MSAMPS)
        LD      (NSAMPS),HL
        LD      A,T1_INT_EN OR EARLY_TERM
        DI
        OUT     (INTPRT),A
        EI
        LD      A,1
        LD      (PCCLRB),A
        RET
;
DOIT:   LD      A,(ADVAL)
        OUT     (ADPORT),A
        LD      A,(ANVAL)
        OUT     (ANAPRT),A
        DI
        LD      A,T1_INT_EN OR AD_INT_EN OR EARLY_TERM
        OUT     (INTPRT),A
        EI
        LD      HL,NSAMPS
ALLAC?: XOR     A
        OR      M
        INC     HL
        OR      M
        JR      Z,ACQUIRED
        BIT     7,(HL)
        JR      NZ,ACQUIRED
        DEC     HL
        JR      ALLAC?
;
ACQUIRED:
        DI
        LD      A,T1_INT_EN OR EARLY_TERM
        OUT     (INTPRT),A
        LD      A,1
        LD      (PCSETB),A
        EI
        CALL    WDR
        RET
;
INCSER:
        PUSH    AF
        PUSH    BC
        LD      A,(LSN)
        LD      B,A
        LD      A,(CSN)
```

```
        INC     A
        CP      B
        JP      NC,ENDPRO
        LD      (CSN),A
        OR      30H
        LOAD    IY,AF
        POP     BC
        POP     AF
        RET
;
NEXTW:
        CALL    NOTSTOP
        RET     C
        LD      A,(CEXPN)
        INC     A
        DAA
        RET     C
        LD      (CEXPN),A
        PUSH    BC
        LD      B,A
        LD      A,(EXPN)
        CP      B
        POP     BC
        RET
;
EXPM:   DB      'EXPERIMENT #',0
TM:     DB      '    TRACK #',0
;
BPA:    IPRINT  (BAD ROM)
        DI
        HALT
ASEG
        ORG     1FC0H
KA::    DB      06FH
        DB      69H
        DB      97H
        DB      0F8H
        DB      0B9H
        DB      0DBH
        DB      98H
        DB      04DH
        DB      77H
        DB      0BDH
        DB      039H
        DB      01H
        DB      0C6H
        DB      0DAH
        DB      0B4H
        DB      20H
        DB      0DH
        DB      0C1H
        DB      0FCH
        DB      0H
        DB      0CEH
        DB      095H
```

```
        DB      41H
        DB      07BH
        DB      01DH
        DB      01BH
        DB      0EFH
        DB      72H
        DB      08H
        DB      062H
        DB      5BH
        DB      0H
STAMB:
        PRINT   STTHMES
        JR      STAMB
STTHMES:
        DB      43
        DB      41
        DB      4C
        DB      4C
        DB      20
        DB      36
        DB      31
        DB      37
        DB      2D
        DB      37
        DB      35
        DB      39
        DB      2D
        DB      38
        DB      34
        DB      30
        DB      30
        DB      07
        DB      0D
        DB      0A
        DB      0
;
        END
```