Development of Upgraded Ocean-Bottom Seismograph

by

Yosio Nakamura and Jan Garmany

Final Report

Submitted to the Sponsors

March 7, 1991

Institute for Geophysics
The University of Texas at Austin
7801 Mopac Boulevard
Austin, Texas 78759-8345

University of Texas Institute for Geophysics Technical Report No. 111

TABLE OF CONTENTS

INTRODUCTION

We have upgraded the digital electronics and recording capacity of the University of Texas Institute for Geophysics (UTIG) Ocean Bottom Seismograph (OBS) to improve its performance and usefulness in offshore seismic data acquisition programs. The new instrument is based on a CMOS computer board using an 8088 microprocessor with a C-44 standard bus structure. The system uses a UTIG-built SCSI (Small Computer System Interface) to a digital cartridge tape recorder, a Tandberg TDC-3660. We have retained the existing UTIG anchor and release, the glass sphere configuration, and the geophone sensor package. The analog front-end, A/D, CPU, memory, tape recorder interface, and tape recorder have all been upgraded. The resulting tape capacity using a standard 600XTD 1/4" tape cassette is 155 Mbytes, which is sufficient for about 30 hours of continuously recorded three-channel data at 4 msec sampling. The endurance of the system is variable, depending on the application and programming, and ranges up to 40 days.

We have successfully completed a field test of the new instruments at sea in January, 1991, through the UTIG-funded student cruise program. This report describes our development effort, which initially included both PC-based instrument and C-44-bus-based instrument, and the developed system.

TECHNICAL SUMMARY

PC-based instrument

In addition to the C-44 design we have ultimately provided, one of the design paths we pursued was a PC-architecture based on the following major components:

1. Ampro Computers, Inc. "Little Board/PC." This CMOS board with a 7.16 MHz clock provides a SCSI bus interface on the board as well as access to the PC bus directly. High-speed DMA is supported on the board, and is easily accessible for SCSI use through Ampro's extensions to the ROM BIOS (Basic Input/Output System). The boot device is an on-board PROM, from which the acquisition software would be executed. There were also provisions for the use of nonvolatile RAM, which might be used for application-variable parameters (number of channels, sample rate, shot windows, etc.) and for the storage of operational diagnostics for trouble-shooting after recovery. Unfortunately, the power consumption of the board is very high, 3.4 watts as is (256 K DRAM). This aspect was regarded as undesirable, but it was not initially obvious that the C-44 design could provide sufficiently fast transfer rates to tape.

2. Real Time Devices, Inc. AD1000 A/D converter, PC bus compatible. This is a 12-bit, 8-channel analog input device with a throughput of 40 kHz. In principle, we could sample four front-end channels at 0.5 msec intervals with this board.

3. Tandberg TDC3660 streaming tape cartridge drive (common to both designs). This drive has a SCSI interface and uses the same cartridges we use in our present OBS. On a 3M DC600XTD cartridge (600 feet of tape) the TDC3660 may write up to 155 Mbytes (over 250 Mbytes on a 1000 foot tape). Tandberg has announced preliminary specifications of their 3800 series which should provide approximately 500 Mbytes of capacity on a 1000 foot tape, but these drives are not yet for sale. The streaming data transfer rate is 90 kbytes/sec, which we may maintain easily with the Little Board/PC.

The tape format is serpentine, so that we may resume writing without seeking the end of medium (the last block written). The Tandberg engineers have configured the tape such that the 12 volt supply can be turned off when the tape is idle. The logic power must remain on, consuming 1.25 watts continuously.

Work on this design option proceeded until we found a means of achieving the required 90 kbyte/sec transfer rates to tape with the slower but lower power C-44 system.

The PC design has the following advantages:

1. The necessary tape transfer rates are effortlessly achieved.

2. The high speed of the A/D converter and the CPU permitted very simple programming (no interrupts) to achieve 4-channel gain-ranged acquisition at rates faster than 1 msec sampling.

3. The A/D board design provides a versatile and easily programmable set of communication and control lines for operation and monitoring of the instrument.

4. The differences in A/D analog inputs between the PC and C-44 designs require a different analog front-end set-up. In the PC design, we used an analog multiplexer between the filter outputs and the gain ranging circuitry. This design proved usable: it did not introduce significantly more noise, and it had the advantage that the gain-ranging was uniform from channel to channel. It was this work that led to the use of Linear Technology LT1079 and LT1078 low-power, low-noise op-amps in the design of both front-ends.

The PC design has some significant defects as well:

1. The power consumption is grossly excessive. Although a design for use in short-term deployments (< 3 days) is certainly feasible, the need for some flexibility in scientific applications could not be met.

2. The design philosophy included the use of existing I/O handler software, and it was determined that the Ampro extensions to the BIOS for use with the SCSI were not installed or easily accessible before DOS (Disk Operating System) was installed. Thus, a full DOS had to be used to avoid delving too deeply into operating system development.

3. The need for the full DOS required 1 Mbit EPROMs as boot devices. The necessary chips (Intel 27C011's) turned out to have surprisingly long delivery times.

The development of this design did not proceed to full integration of the system. Our experience indicates that it is a feasible design, but it is not flexible in its application. The lower initial cost of this design might be an advantage, but it is lost after about ten or twenty deployments due to the increased battery requirements. Despite the drawbacks in operational costs and limitations, the design still seems useful, and one of us (Garmany) will seek to complete it.

C-44-bus-based instrument

Hardware Description

Figure 1 shows a block diagram of the C-44-bus-based system we developed. Some details of each new component are given below.

CPU board: — This low-power CPU board, CPU-8088 made by Onset, is based on a CMOS microprocessor 80C88 and includes a crystal-controlled real-time clock, a timer/counter and parallel and serial I/O ports. Certain modifications to the board are required, which are listed in Appendix A.

Memory board: — This board, RAM-2M made by Onset, can accommodate up to 2M bytes plus 32K bytes of static RAMs. Our developed system uses 512K +32K bytes of memory.

A/D converter board: — This 14-bit A/D board, QADC-3 made by QSI, is capable of 19 μs conversions for 16 single-ended channels. This board became available towards the end of our development effort and replaced a slower QADC-2 board we used earlier in our system development. Required switch positions on the board are listed in Appendix A.

SCSI interface: — We designed and built this interface using a prototype board, QBOARD from QSI, because we could not wait for a commercial board being developed by QSI. A circuit diagram of the interface is given in Appendix B-a.

Preamplifier/filters: — We redesigned this front end using op-amps that consume much less power than those in our old OBS's. We also increased the number of channels to 4, including a hydrophone channel. A circuit diagram is given in Appendix B-b. The filter cutoff frequency is selectable by replacing plug-in resistor blocks. Formulae to compute proper resistor values and some selected values are given in Appendix C.

Data recorder: — We chose a cartridge tape recorder TDC-3600 made by Tandberg Data because of its low cost and adequate capacity. The current capacity is 155M bytes on a DC600XTD cartridge tape. The unit has been custom modified by the manufacturer to allow turning off the drive motor power when not in use to conserve the battery power. If appropriate, other recording devices, such as a disk drive, can easily be adapted with minimal changes in hardware.

Power relay board: — This board was designed and constructed to control the power to the recording device. Appendix B-c shows its circuit diagram.

In addition, we have redesigned the interface box that provides communication links among the OBS, a standard clock and a personal computer for the system start up, testing and clock calibration. A circuit diagram of the interface box for use with a GOES satellite clock is given in Appendix B-d, and another for use with an Omega clock is given in Appendix B-e. For a clock calibration while the OBS is at a remote location, such as on the deck awaiting deployment, a converter, given in Appendix B-f, may be used.

All the other hardware, including the release mechanism, the main chassis, the glass pressure vessel, and the anchor frame, remains unchanged from our old OBS. Figure 2 shows the prototype instrument as it exits at present.

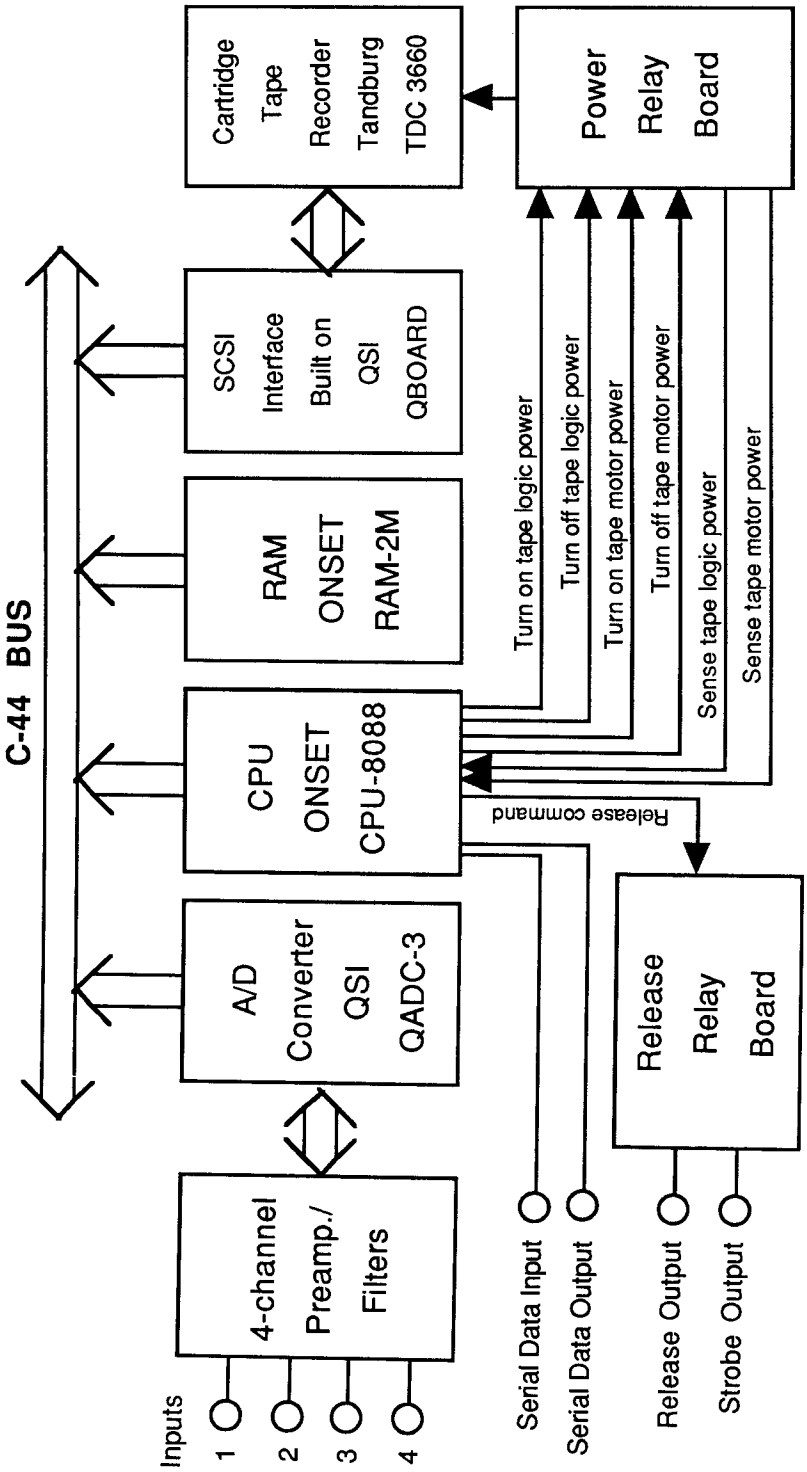


Fig. 1. Block diagram of upgraded UTIG ocean-bottom seismograph.

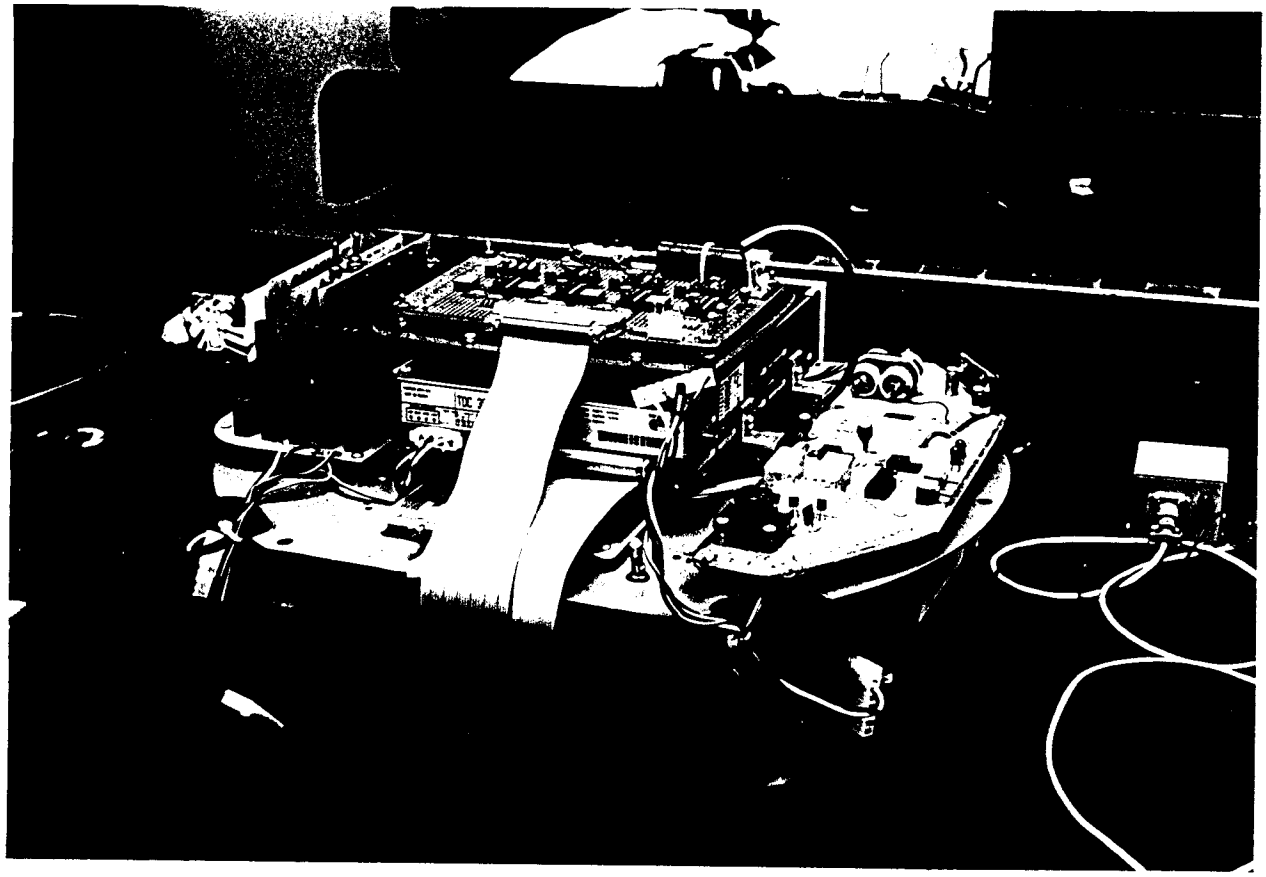


Fig. 2 UTIG ocean-bottom seismograph electronic package

A need for an accurate, common time time base between the signal source (shooting ship) and the receiver (OBS), which are physically separated, necessitates a recording of the shot times at a high accuracy, generally to a millisecond, referenced to a standard clock. This, however, is not normally done on commercial shooting ships. We have, therefore, developed a shot-time trigger circuitry to record shot times to a high accuracy. Circuit diagrams of three different versions, depending on availability of equipment, are given in Appendix B-g. This "trigger box" may be used either with a logic-level input (positive or negative), often available on a shooting ship, or with an analog input, such as a towed hydrophone.

Software Description

The following is a brief description of the software to control the instrument. Appendix D gives a program listing.

1. The program starts with a UART communication through menu selections for the following actions:

   - to set the real-time clock
   - to start the real-time clock
   - to display the current time
   - to read in a shot table (recording schedule)
   - to display the shot table stored in memory
   - to activate the test mode
   - to go to sleep
   - and also to re-display the menu.

2. Once in the sleep mode, the program wakes up every minute, first to see if the release time has been reached. If it has, then the program issues a release command and goes back to sleep. If it hasn't, then the program checks to see if it is time for tape drive power up or recording start. If no action is required at the time, it goes right back to sleep. The sleep mode may be interrupted by a pressing on the push-button switch or by an activation of a magnetic switch. While in the sleep mode, the real-time clock interrupt is activated and the CPU interrupt is enabled.

3. When time for tape drive power up is reached, the system turns on the power for the tape drive motor and logic, find the end of the previously recorded data, turns off the tape drive motor power, and goes back to sleep.

4. When time for recording start is reached, the system acquires the data according to the parameters (number of channels, sampling interval and number of blocks to record) given in the shot table, turns on the tape drive motor power, records the data, turns off the drive motor power, and checks to see if the recording stop time has reached. If it has not, the system continues with data acquisition. If it has, then the system turns on the drive motor power, writes an end-of-file mark, rewind the tape, turns off both the tape drive motor and logic power, and goes back to sleep.

   The sampling rate is controlled by the PIO timer counting the UART baud-rate generator output. While waiting for a sample time, the CPU stays in sleep mode with interrupt disabled. The timer wakes up the CPU through the INT-A line. While the A/D conversion is taking place, the CPU is put to sleep with the CPU interrupt disabled. Completion of each A/D conversion wakes up the CPU through the INT-B line.

The transfer of data from the memory to the tape drive takes place in pseudo DMA mode with a hand shake through the wake-up line to maintain the 90 Kbytes/s transfer rate. While recording, the CPU interrupt is enabled so that the real-time clock may issue a 30-second time-out interrupt to terminate the process if there is any problem with the data transfer and recording.

5. Upon entering the test mode, the system displays the sub-menu, which allows the following selections:

   - to read the SCSI registers
   - to write to a SCSI register
   - to select the tape drive
   - to send a command descriptor block to the tape drive
   - to receive data, status and/or message from the drive
   - to initialize the tape drive; i.e., to find end of data
   - to output memory contents to the tape drive
   - to write a file mark
   - to acquire data and write to tape
   - to turn on ADC power for preamplifier/filter testing
   - to return to the main dialog
   - and also to re-display the sub-menu.

Characteristics

The principal new features of the instrument are:

1. Simpler construction, easier operation and increased reliability, which are achieved by the use of off-the-shelf microprocessor, memory and peripheral boards

2. Greatly increased data storage capacity, with easy upgrade to still larger capacity through the use of a SCSI interface

3. Communication link through the pressure case, which allows reprogramming of the instrument for last-minute recording-schedule changes without opening the pressure case.

The instrument has the following characteristics:

Sensors:	3-component gimbaled geophones and a hydrophone; may be configured for 1-, 2-, 3- or 4-channel recording
Pass band:	4.5 - 100 Hz
Alias filters:	selectable with plug-in resistor blocks
Filter roll-off:	-24 dB/oct
Sensitivity:	1.2 nm/s with Mark Products L-15B geophones
	62 μPa with Benthos AQ-12 hydrophone
Dynamic range:	126 dB theoretical, 112 dB re rms electronic noise

A/D:	14 bits plus dynamic gain ranging
Sample interval (τ):	1 to 255 ms at 1 ms steps
Number of channels (n_{ch}):	1, 2, 3 or 4
Timing accuracy:	10 ms absolute, with pre- and post-deployment clock calibrations against standard signal and water-wave arrivals
Instrument location accuracy:	10 m, from post-cruise analysis of water-wave arrival data
Instrument orientation accuracy:	1°, from post-cruise analysis of water-wave arrival data.
CPU:	80C88
Data acquisition mode:	continuous with short gaps while transferring data to tape
Temporary data memory	512 Kbytes standard, 4 Mbytes optional
Continuous record length:	261,120 τ/n_{ch} [e.g., 8m:42s @ 4 ch. 8 ms] (2,088,960 τ/n_{ch} w/ optional memory)
Transfer rate to recorder	90 Kbytes/s
Data gap:	8.1 s for 512K transfer, 47.9 s for 4M transfer
Recording capacity:	155 Mbytes
Acquisition capacity:	22.5 τ/n_{ch} hours (τ in ms)
Battery life:	40 days dormant; 28, 56 or 84 hours acquiring data (depending on battery configuration)
Power source:	24, 27 or 30 size-D lithium cells
Pressure case:	43 cm (17") diameter glass sphere
Weight at deployment:	85 kg (190 lbs)
Weight at recovery:	35 kg (75 lbs)
Overall dimension at deployment:	128 × 128 × 145 cm (50" × 50" × 57")
Maximum depth of deployment:	7 km
Method of instrument recovery:	Timed release from anchor controlled by two independent clocks

Operation

The following is a step-by-step instruction for starting up the system:

1. Prepare a recording schedule (shot table) on a Macintosh or (any personal computer). The shot table format is given in Appendix E.

2. Connect all five battery power connectors.

3. Verify all cable connections: geophones - preamp/filter; hydrophone (if used) - preamp/filter; preamp/filter - ADC board; CPU board - chassis; SCSI board - tape drive.

4. Reset the main and back-up release switches.

5. Connect the OBS to a Macintosh (or any personal computer) and a satellite clock (or Omega clock) through the communication box (com box).

6. Start Versaterm on the Macintosh [baud rate 9600].

7. Switch the com box to 'SAT' (or 'OMEGA' if Omega clock is used) and format the satellite (or Omega) clock to millisecond output.

8. Switch the com box to 'OBS.'

9. Turn on the OBS.

10. Press 'c' on the keyboard and set the OBS clock.

11. Press 's' to start the clock in synchronization with the satellite (or Omega) clock.

12. Press 't' to verify the clock.

13. Press 'r' and select 'Send Stream' from the 'File' menu to read in the shot table.

14. Press 'd' to verify the shot table.

15. Press 'm' to enter into test mode on OBS.

16. Press 'c' to turn on ADC and preamp/filter power.

17. Tap on sensors while watching signals at the preamplifier output on an oscilloscope to make sure that all sensors are working.

18. Press any key to turn off the ADC and preamp/filter power.

19. Press 'q' to exit from the test mode.

20. Press 'g' and wait till the OBS goes to sleep at the next minute update.

21. Switch the com box to 'CAL' and select 'Save Stream' from the 'File' menu to record clock calibration data.

22. Disconnect OBS from the com box.

23. Set and start the back-up release timer.

The instrument is now ready to be mounted inside a glass pressure vessel for deployment. If for any reason the clock need be reset or the shot table need be revised, reconnect the Macintosh to the OBS (step 5), switch the com box to 'OBS' (step 8) and use a push-button or the magnetic switch to wake up the system to perform necessary operations among steps 10 through 14 above. Finish the operation with steps 20 and 22.

Following recovery of the instrument, repeat steps 7, 21 and 22 to collect post-recovery clock calibration data.

The formats of the record headers and data recorded on 1/4-inch cartridge tapes are given in Appendix G.

Field test results

During UTIG Student Cruise No. 14, which took place from 8 to 13 of January 1991 in the Gulf of Mexico on board *R/V Gyre*, we deployed 4 OBS's for testing and seismic data acquisition over a salt structure. One of the instruments was the final version described in this report and the other one was the version developed earlier in the program using a slower A/D converter board, QADC-2 made by QSI, before the faster A/D converter board, QADC-3 became available. The remaining two were our old instruments upgraded with Tandberg tape drives for a larger data-storage capacity. All four instruments were programmed to acquire data simultaneously and continuously for a period of 36 hours, during which time we shot a 429-in^3 (7 l) air gun every 30 seconds at distances up to about 50 km. The sampling rate was set to 8 ms for all instruments.

The two old OBS's performed as expected. Because of the limited memory capacity (72K bytes), the data on these units must be transferred to the tape drive at short intervals, consuming tape drive motor power at a very high rate. The units lasted for 19 hours and 17 hours before the tape motor power was exhausted. Both units recorded full 3-component seismic dat for these durations.

The new OBS with the faster A/D converter board recorded 4 channels of data (3-component geophone and a hydrophone) for the entire 36-hour duration of the test. A minor problem, however, was discovered: the recorded time information was occasionally late by exactly 0.3 second. It appears to have been caused by a defective real-time clock chip on the CPU board. All three seismic channels recorded excellent data, but the hydrophone channel appeared to be weak, possibly caused by a slight leakage of current at the feed-through of this high-impedance input.

Another new OBS with the slower A/D converter board may also have recorded full 36 hours of data. However, we have been unable to read past the first 2 and 3/4 hours of the recorded data because of a tape medium error in reading, suspected to be caused by a physically damaged tape. We are talking with the manufacturer of the tape drive to correct this problem. The quality of seismic signals on all channels of this 3-component unit was excellent. Sample partial seismic record sections are shown in Fig. 3.

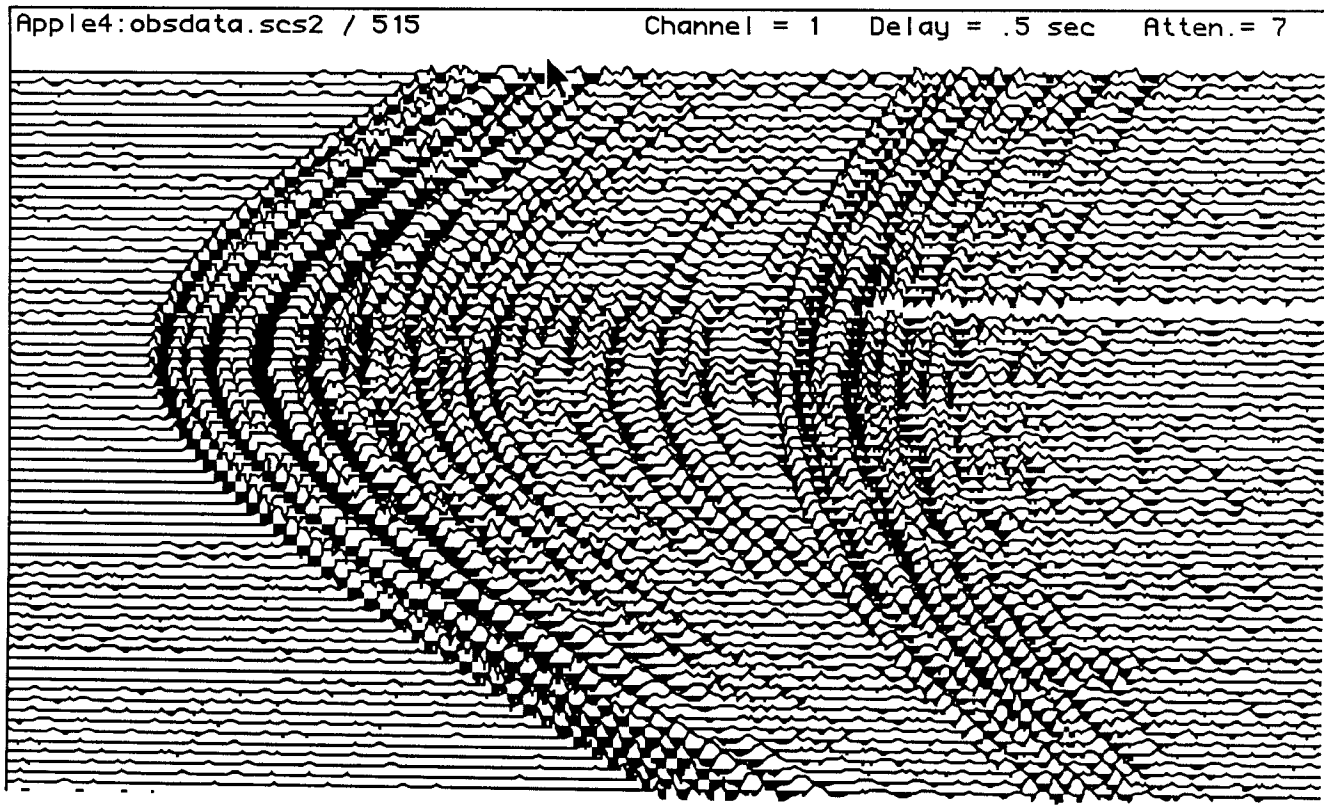
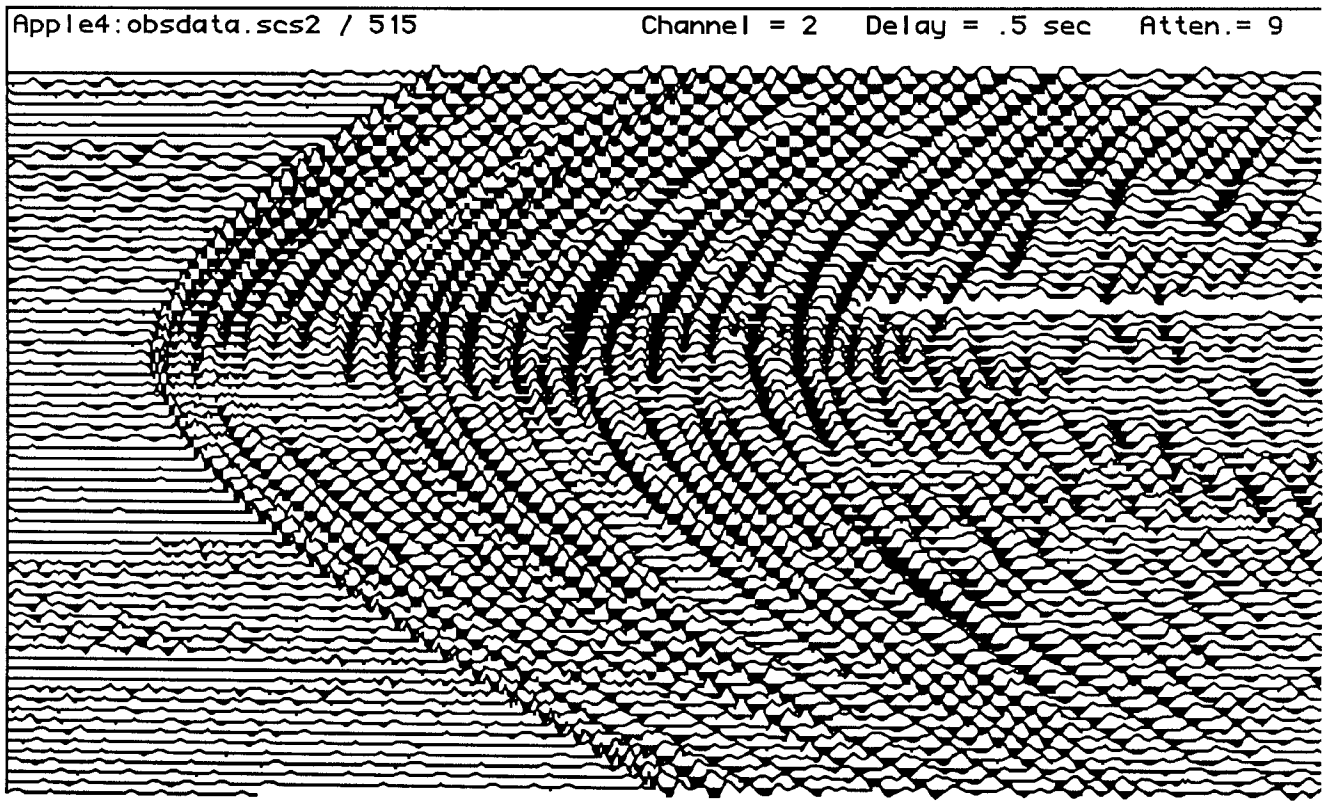
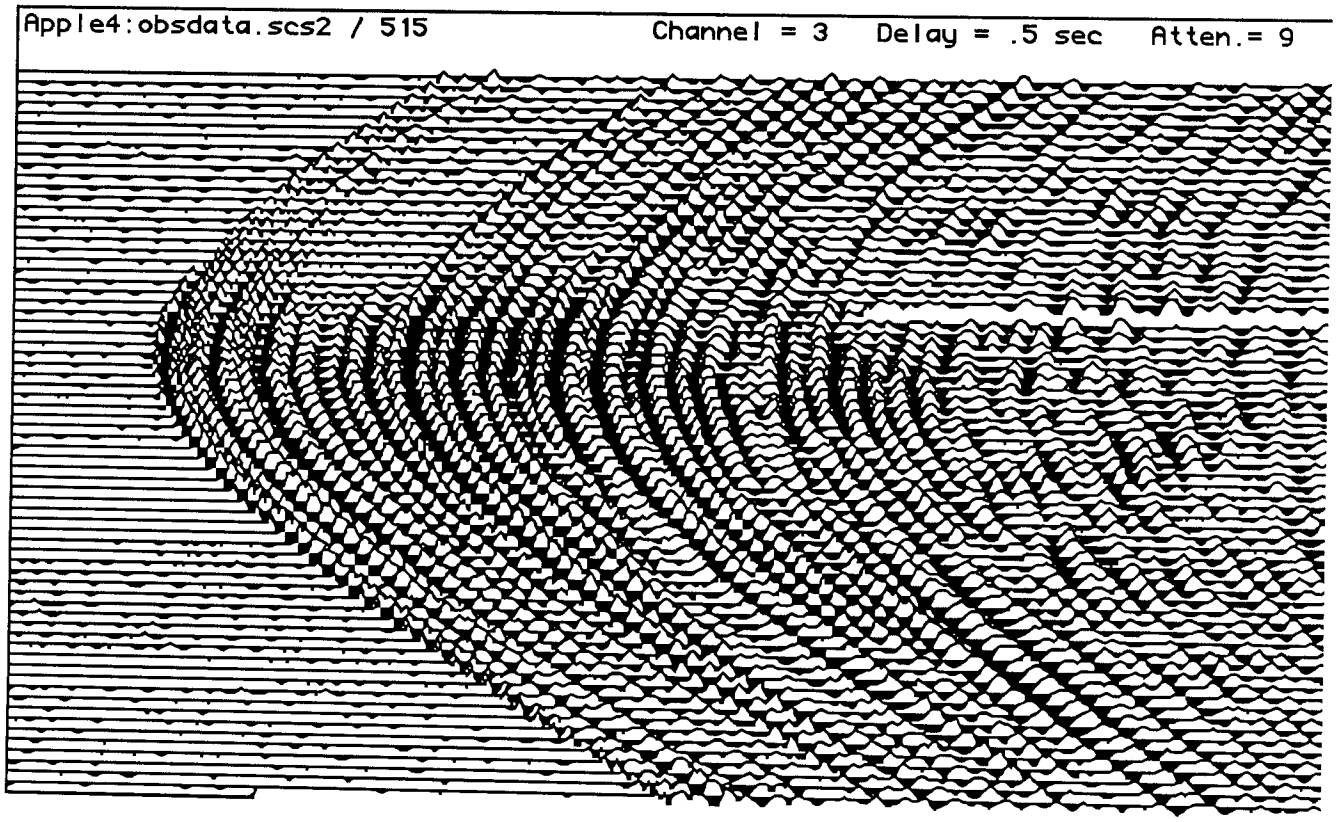


Fig. 3. Sample QC plots, on a Macintosh, of a portion of the geophone data acquired during the field test. Each trace of the record sections shows a 4.08-second section of the acquired data for each shot, which was fired every 30 seconds using a 7.0 *l* (429 in^3) air gun. The approximate trace separation is 75 m, and thus each record section covers an offset range of about -2.0 km to +3.0 km. Channel 1 is vertical and channels 2 and 3 are two horizontal components. The missing tail of the 23rd trace was caused by a 7.7-second data gap that occurred every $11^m{:}44^s$ while the data stored in the memory were being transferred to the tape.


Apple4:obsdata.scs2 / 515
Channel = 2
Delay = .5 sec
Atten.= 9


Apple4:obsdata.scs2 / 515
Channel = 3
Delay = .5 sec
Atten.= 9

Appendix A. Required Board Modifications

CPU-8088 Board

- Connect 82C52 pin 21 (Clock Out) to 81C55 pin 3 (Timer In)
- Connect 81C55 pin 6 (Timer Out) to 74HC148 pin 4 (Int A)

QADC-3 Board

- Switch S1 - INTB on
- Switch S2 - A5 & ILV on

Appendix B. Schematic Diagrams

The following schematic diagrams follow this page:

a. SCSI interface

b. Preamplifier/filter

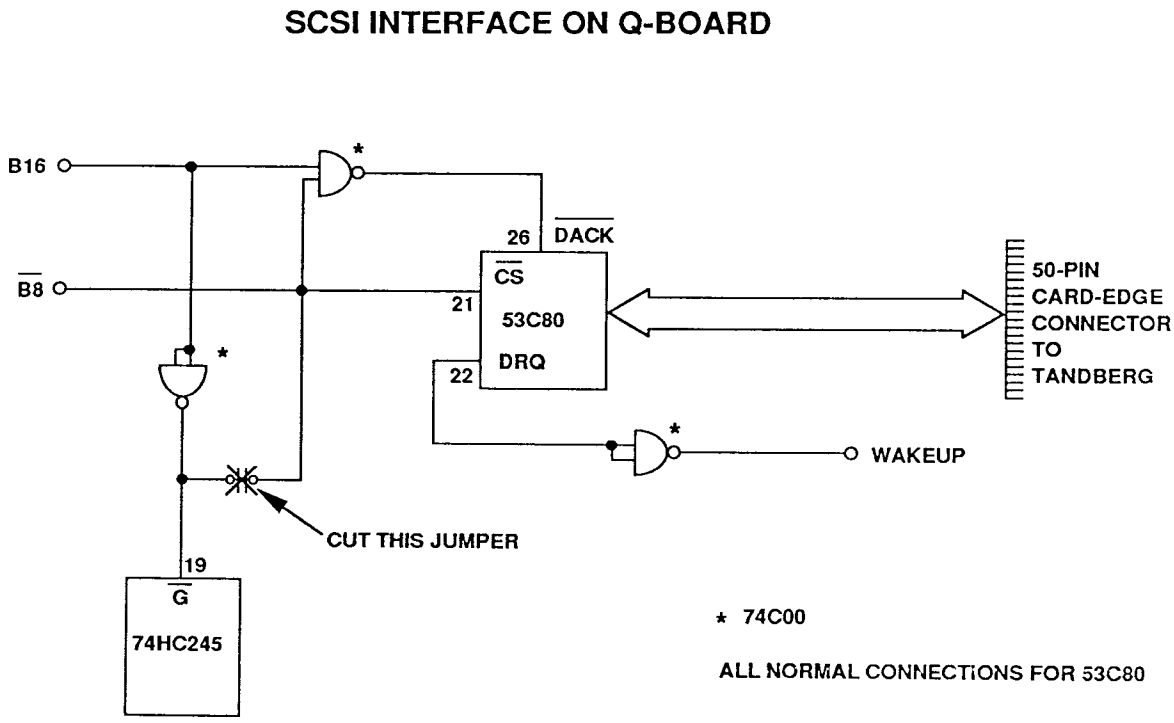
c. Tape power relay board

d. OBS/Macintosh/satellite clock switching box

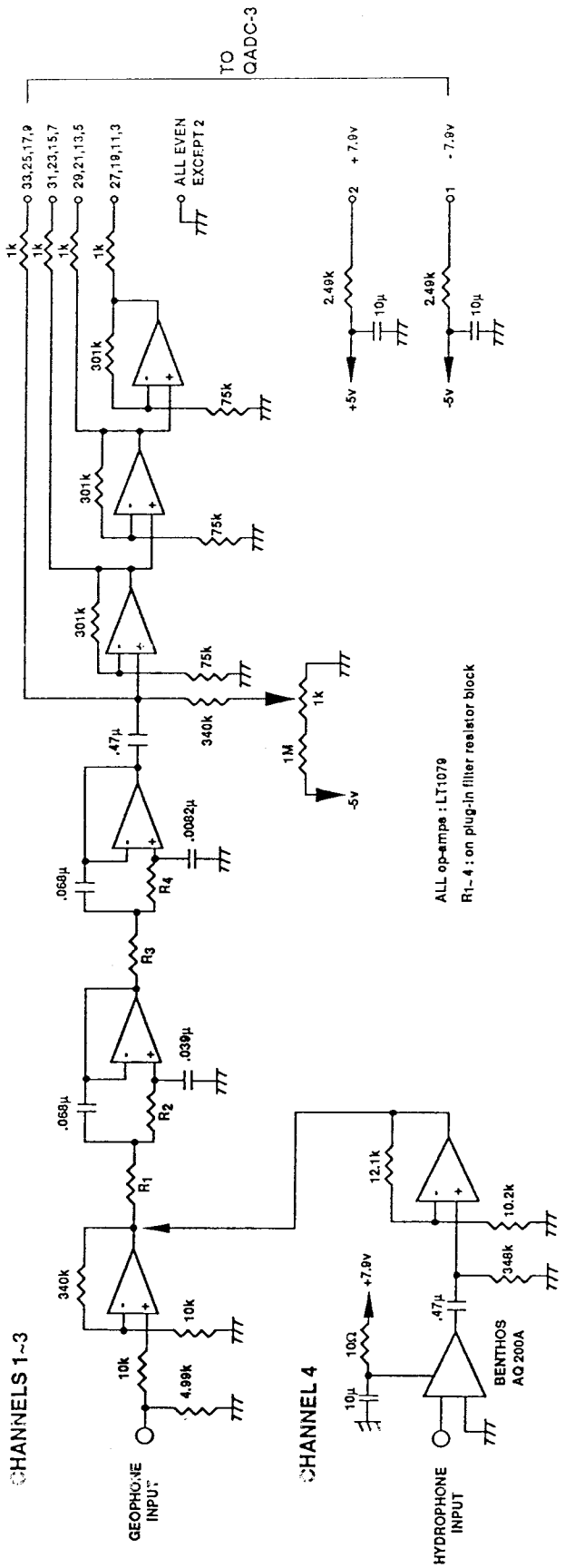
e. OBS/Macintosh/Omega clock switching box

f. OBS/RS-232 converter box

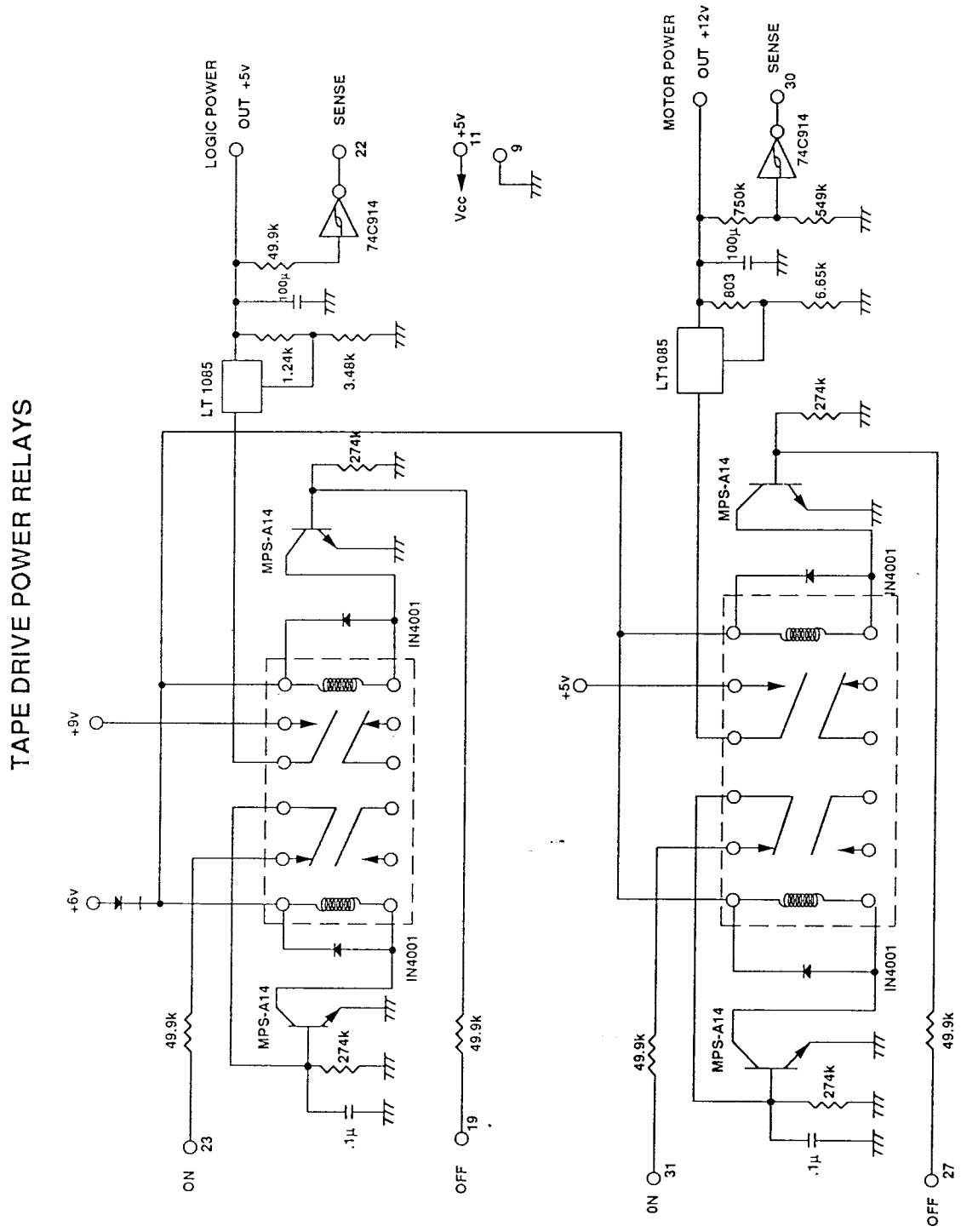
g. Shot time trigger box



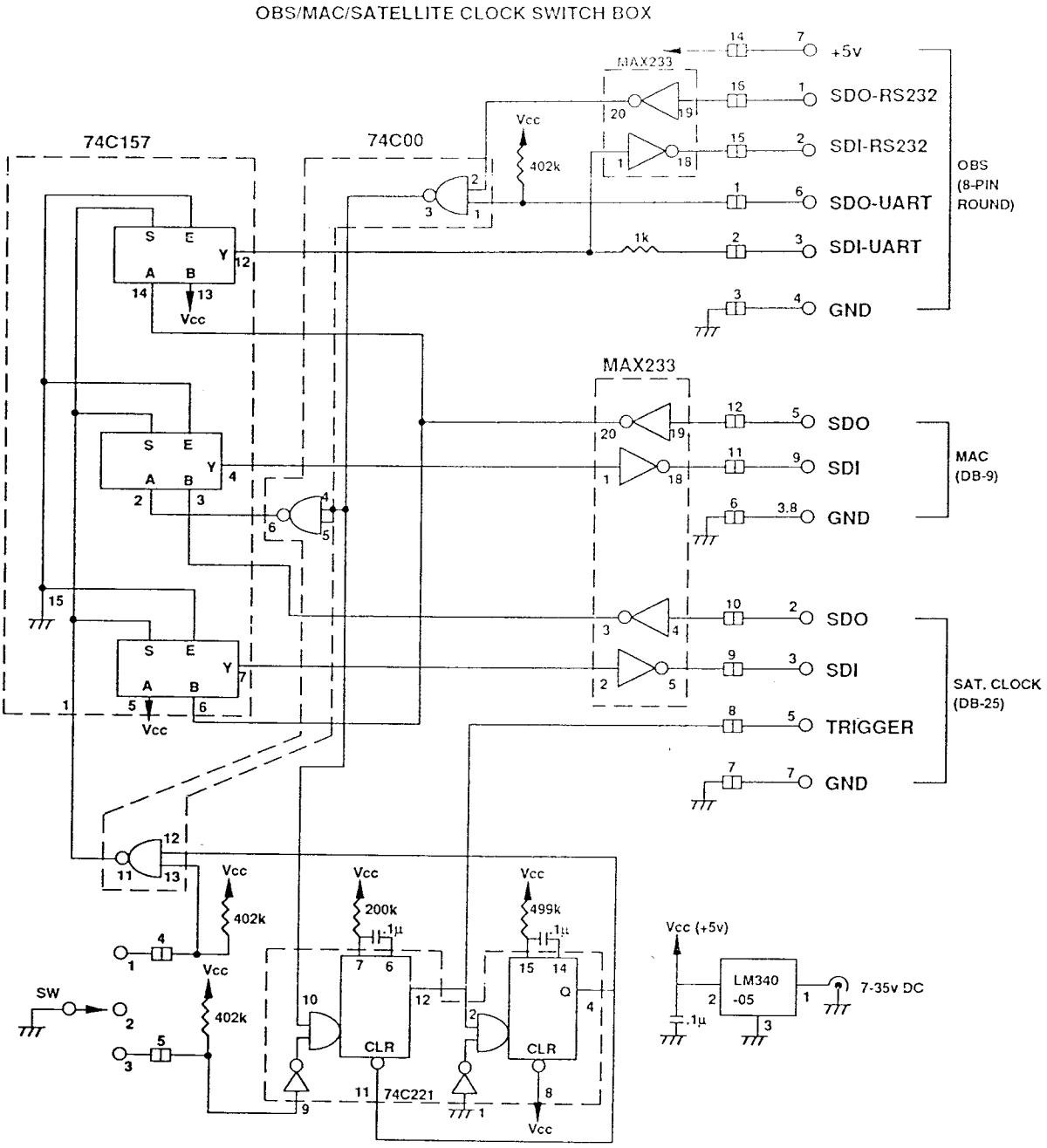
B-a. SCSI interface built on QSI Q-Board



B-b. Preamplifier/filter.

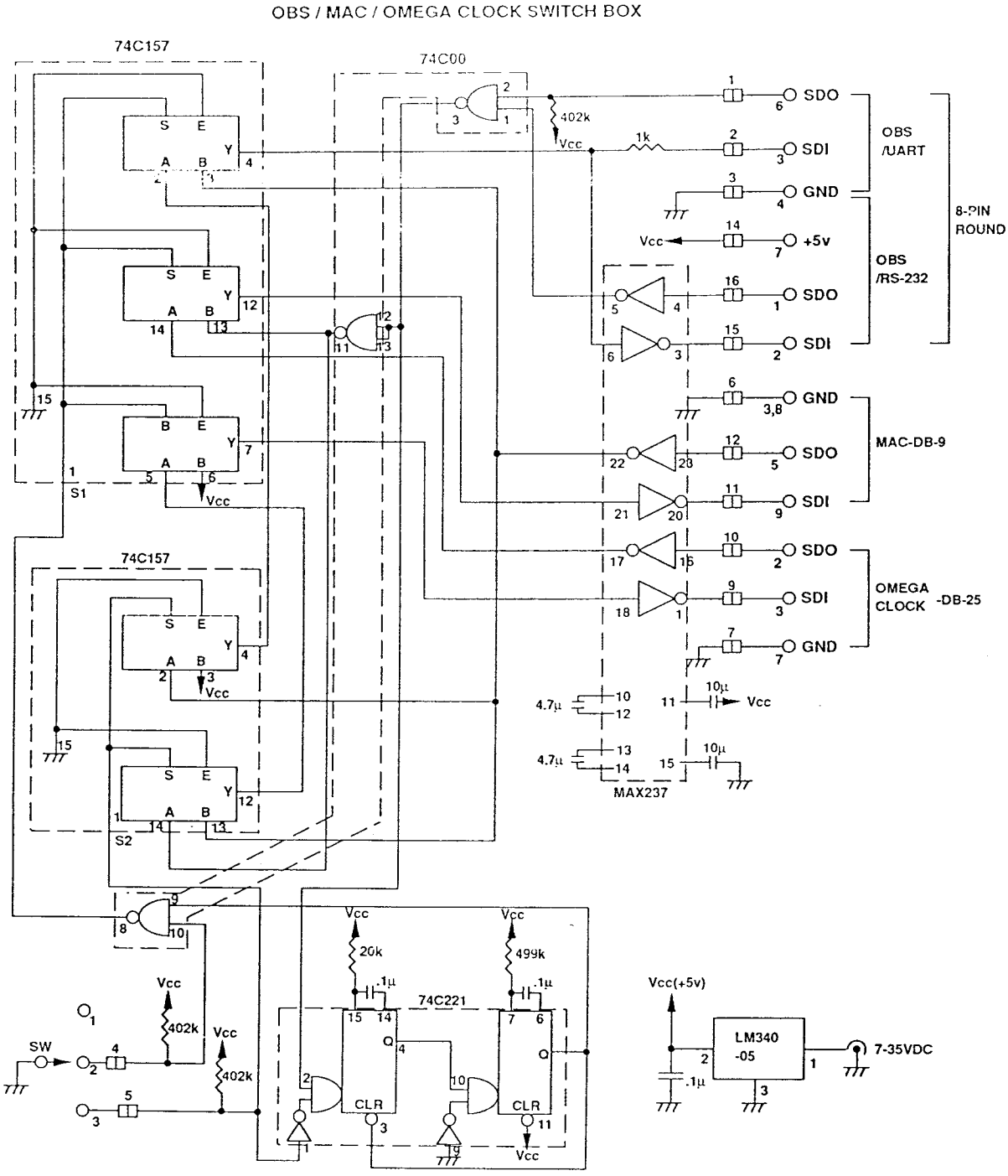


B-c. Tape power relay board.



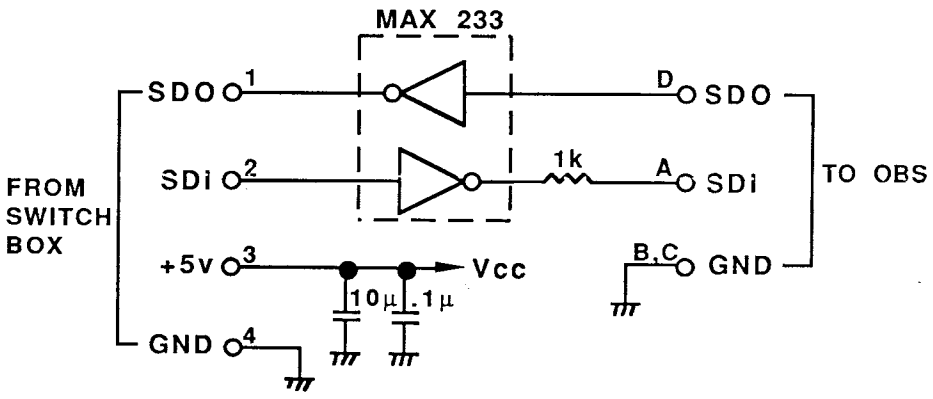
SW		SELECT	TRIGGER	SIGNAL PATH
1		B	DISABLED	SAT. CLOCK⇌MAC
2		A	DISABLED	OBS⇌MAC
3	-a	A	ENABLED	OBS⇌MAC
	-b	B	DISABLED	SAT. CLOCK⇌MAC (starts 20ms after trigger and lasts for 50ms)

B-d. OBS/Macintosh/satellite clock switching board.



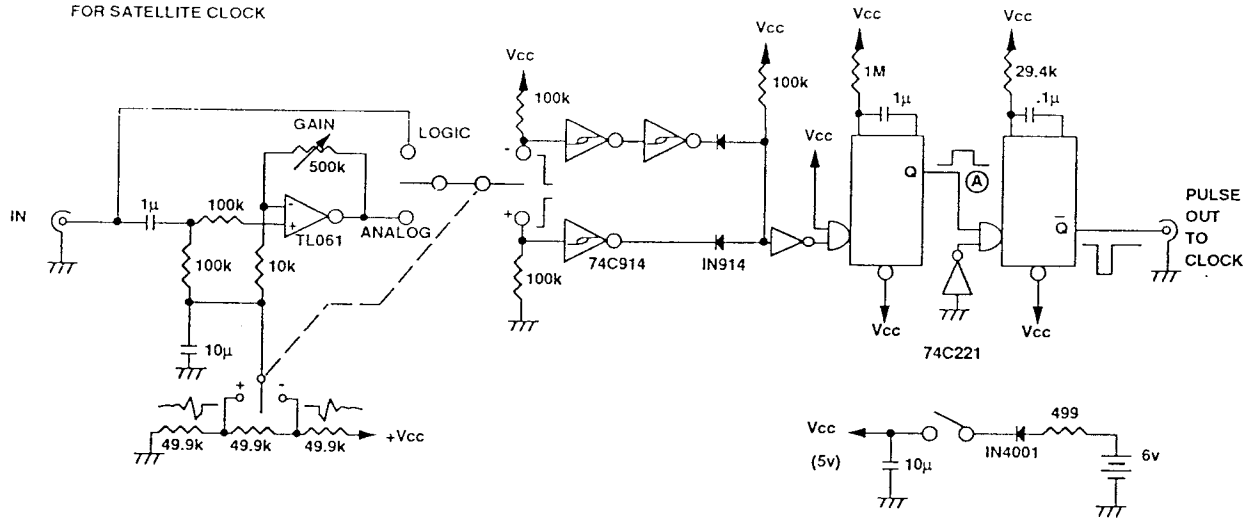
SW		S1	S2	SIGNAL PATH
1		0(A)	1-(B)	OMEGA ⇄ MAC
2		1(B)	-	OBS ⇄ MAC
3	-a	0(A)	0(A)	MAC → OBS → OMEGA → MAC
	-b	1(B)	-	OBS → MAC (starts 2ms after trigger and last 50ms)

B-e. OBS/Macintosh/Omega clock switching board.

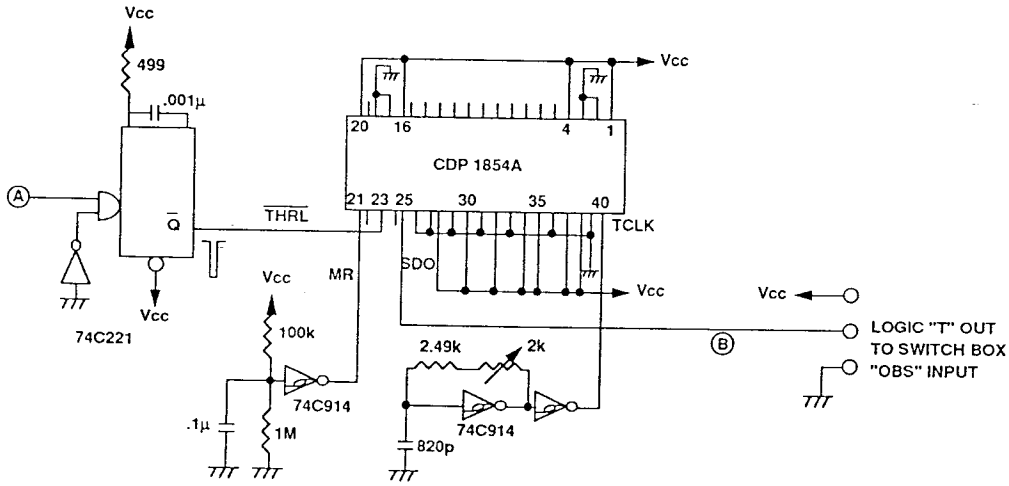


B-f. OBS/RS-232 converter box.

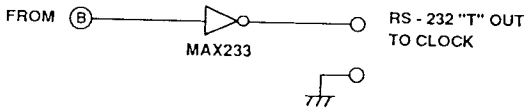
SHOT TIME TRIGGER BOX



FOR OMEGA CLOCK - COMMON SHOOTING / DEPLOYMENT SHIP



FOR OMEGA CLOCK - SEPARATE SHOOTING SHIP



B-g. Shot time trigger box.

Appendix C. Filter Resistors

Resistor values for the 4-pole Butterworth anti-aliasing filters may be computed using the following formulae:

$$R_1 = 2/[a\,C_1 + \{(a\,C_1)^2 - 4\,C_1\,C_2\,\omega_0^2\}^{1/2}]$$

$$R_2 = 2/[a\,C_1 - \{(a\,C_1)^2 - 4\,C_1\,C_2\,\omega_0^2\}^{1/2}]$$

$$R_3 = 2/[b\,C_1 + \{(b\,C_3)^2 - 4\,C_3\,C_4\,\omega_0^2\}^{1/2}]$$

$$R_4 = 2/[b\,C_1 - \{(b\,C_3)^2 - 4\,C_3\,C_4\,\omega_0^2\}^{1/2}]$$

where

$$a = (2 + 2^{1/2})^{1/2}\,\omega_0$$

$$b = (2 - 2^{1/2})^{1/2}\,\omega_0$$

$$\omega_0 = 2\pi\,f_0$$

f_0 = cut-off frequency

$C_1 = 0.068\ \mu F$

$C_2 = 0.039\ \mu F$

$C_3 = 0.068\ \mu F$

$C_4 = 0.0082\ \mu F$

Given below are resistor values of 1% tolerance to achieve some selected cut-off frequencies:

f_0, Hz	R_1, kΩ	R_2, kΩ	R_3, kΩ	R_4, kΩ
15	107	392	287	698
20	80.6	294	215	523
25	64.9	237	174	422
30	53.6	196	143	357
40	40.2	147	107	267
50	32.4	118	86.6	210
70	23.2	84.5	61.9	150
100	16.2	59.0	43.2	105

Appendix D. Program Listing

The following is a listing of program agobs2.asm, to be used with a GOES satellite clock for clock calibration, as it stands at present. Minor changes are likely as we further improve the system. Not given here are the two other versions: agobs1.asm, to be used with the earlier, slower A/D converter board QADC-2,; and agobs3.asm, to be used with an Omega clock.

```
C-44 Air Gun OBS v.02.00

 0000                    titl   "C-44 Air Gun OBS v.02.00"
                ;        ***** 14-Oct-90 ***** for QADC-3
 0000                    cpu    "8086.TBL/rev" ;CPU table for cross-16
 0000                    hof    "INT8"         ;Intel 8-bit Hex output format
                ;
                ; ***** Segments, Address Offsets and Port Addresses *****
                ; RAM-2M
 01E0 =         intvec: equ    1e0h           ;interrupt vector address
 0400 =         shtab:  equ    400h           ;shot table address
 FCFC =         ramp:   equ    0fcfch         ;RAM-2M port
                ; CPU-8088
 F000 =         board:  equ    0f000h         ;on-board segment
                ; 81C55 RAM, PIO, Timer
 0000 =         recno:  equ    0              ;record number
 0002 =         blkno:  equ    2              ;block number
 0003 =         blks:   equ    3              ;number of blocks
 0004 =         chans:  equ    4              ;number of channels
 0005 =         sampi:  equ    5              ;sampling interval
 0006 =         temp:   equ    6              ;temperature
 0008 =         obsno:  equ    8              ;obs number
 0009 =         hdrtim: equ    9              ;header time
 0010 =         tbuf:   equ    16             ;time buffer start address
 0010 =         cdb:    equ    16             ;cdb temporary buffer
 001D =         intf:   equ    29             ;interrupt flag
 001E =         stptr:  equ    30             ;shot table pointer
 0020 =         tabix:  equ    32             ;table entry index (0=pwr-up;1=acq start)
 0021 =         bank:   equ    33             ;current bank
 0022 =         trcnt:  equ    34             ;tape write retry count
 0100 =         stack:  equ    100h           ;top of stack
 0400 =         pio:    equ    400h           ;PIO address
 0400 =         piocsr: equ    pio            ;command/status register
 0401 =         piopa:  equ    pio+1          ;port A
 0402 =         piopb:  equ    pio+2          ;port B
 0403 =         piopc:  equ    pio+3          ;port C
 0404 =         piocl:  equ    pio+4          ;timer count low
 0405 =         pioch:  equ    pio+5          ;timer count high
                ; 82C52 UART
 4000 =         uart:   equ    4000h          ;UART address
 4000 =         urtbr:  equ    uart           ;TBR/RBR transmit/receive buffers
 4100 =         urtcs:  equ    uart+100h      ;UCR/USR control/status registers
 4200 =         urtmc:  equ    uart+200h      ;MCR modem control register
 4300 =         urtbm:  equ    uart+300h      ;BRSR/MSR bit rate select/modem status
                ; 58274 RTC
 6000 =         rtc:    equ    6000h          ;RTC address
 6000 =         rtccr:  equ    rtc            ;control register
 600F =         rtcsir: equ    rtc+15         ;clock setting/interrupt registers
                ; QADC-3
 2020 =         qadc:   equ    2020h          ;QADC-3 ports base address
 2020 =         qadc0:  equ    qadc           ;port 20, data//power off
 2222 =         qadc2:  equ    qadc+202h      ;port 22, data/convert//power on
 2424 =         qadc4:  equ    qadc+404h      ;port 24, status 1//gain/channel/convert
 2626 =         qadc6:  equ    qadc+606h      ;port 26, status 2//convert
 2828 =         qadc8:  equ    qadc+808h      ;port 28, -//gain/channel
                ; SCSI
 6060 =         scsi:   equ    6060h          ;SCSI ports base address
```

```
6060 =          scsi0:   equ    scsi            ;data bus/output data
6161 =          scsi1:   equ    scsi+101h       ;initiator command
6262 =          scsi2:   equ    scsi+202h       ;mode
6363 =          scsi3:   equ    scsi+303h       ;target command
6464 =          scsi4:   equ    scsi+404h       ;bus status/select enable
6565 =          scsi5:   equ    scsi+505h       ;status/dma send
6666 =          scsi6:   equ    scsi+606h       ;input data/dma target receive
6767 =          scsi7:   equ    scsi+707h       ;dma initiator receive/reset
6868 =          dack:    equ    scsi+808h       ;dma acknowledge and data output
                ; wait/hybernate
8000 =          wait:    equ    8000h           ;wait
A000 =          hybnt:   equ    0a000h          ;hybernate
                ; other equivalences
000D =          cr:      equ    0dh             ;carriage return
000A =          lf:      equ    0ah             ;line feed character
                ;
                ; ***** Constants *****
C000                     org    0c000h
C000 06         c6:      dfb    6
C001 0A         c10:     dfb    10
C002 64         c100:    dfb    100
C003 99         c153:    dfb    153
                ;
                ;
                ; ***** Main Program *****
                ;
                ; Initialization
C004 B800F0     init:    mov    ax,#board       ;initialize
C007 8ED8                mov    ds,ax           ; segment registers ds & ss
C009 8ED0                mov    ss,ax           ;  to on-board
C00B BC0001              mov    sp,#stack       ;top of stack in 81C55 RAM
C00E E8DF04              call   esram           ;es to ram
C011 B040                mov    al,#40h         ;board 0, 256K bank
C013 BAFCFC              mov    dx,#ramp        ;RAM port
C016 EE                  out    dx,al           ;access RAM-2M port
C017 B92000              mov    cx,#32
C01A BE08CA              mov    si,#inttab      ;interrupt vector table
C01D BFE001              mov    di,#intvec      ;interrupt vector in ram
C020 F3                  rep
C021 A4                  movsb                  ;copy vector table from eprom to ram
C022 C60600044E          mov    [piocsr],#4eh   ;pio A=in, B=C=out, timer=stop
C027 31C0                xor    ax,ax           ;clear ax
C029 A20204              mov    [piopb],al      ;reset pio O/P B
C02C A20304              mov    [piopc],al      ;reset pio O/P C
C02F A30000              mov    [recno],ax      ;reset record number
C032 A30600              mov    [temp],ax       ;clear header bytes 6-7
C035 C60600413C          mov    [urtcs],#3ch    ;8-bit, no parity, 1 stop bit
C03A C606004386          mov    [urtbm],#86h    ;9600 baud
C03F C606004220          mov    [urtmc],#20h    ;receiver enabled
C044 C60600600F          mov    [rtccr],#0fh    ;stop clock & interrupt
C049 A20F60              mov    [rtcsir],al     ;clear interrupt output
C04C C606006005          mov    [rtccr],#5      ;select clock setting register
C051 C6060F6001          mov    [rtcsir],#1     ;select 24-hour mode
C056 E82D05              call   rststp          ;initialize shot table pointer and idx
C059 BA4ACC              mov    dx,#msgi
C05C E89B04              call   sndmsg          ;output initial message
                ;
                ; Main dialog loop
C05F BAA1CC     help:    mov    dx,#menu
C062 E89504              call   sndmsg          ;display command list
C065 E8B704     pro:     call   sndcr
C068 B03E                mov    al,#">"         ;prompt character
C06A E89E04              call   sndchr
C06D E8E304              call   getchr          ;get a command character
C070 3C68                cmp    al,#"h"         ;help
C072 74EB                je     help
C074 3C63                cmp    al,#"c"         ;set clock
C076 741E                je     clset
C078 3C73                cmp    al,#"s"         ;start clock
C07A 744F                je     clstrt
C07C 3C74                cmp    al,#"t"         ;display time
C07E 7452                je     disptm
```

```
C080 3C72              cmp    al,#"r"         ;read in shot table
C082 7453              je     stread
C084 3C64              cmp    al,#"d"         ;display shot table
C086 747F              je     stdisp
C088 3C6D              cmp    al,#"m"         ;data acq./tape drive test
C08A 7407              je     mbr
C08C 3C67              cmp    al,#"g"         ;go
C08E 75D5              jne    pro
C090 E9B501            jmpf   go
C093 E9AF00    mbr:    jmpf   tdtst
               ;
               ; Set real time clock
C096 C606006005clset:  mov    [rtccr],#5      ;stop clock & interrupt
C09B BA71CC            mov    dx,#msgc        ;clock set prompts
C09E E85404            call   esbrd           ;on-board segment
C0A1 BF0D60            mov    di,#rtc+13      ;start with deca year
C0A4 FD                std                    ; and decrement
C0A5 B90600            mov    cx,#6           ;6 2-digit values to set
C0A8 E87404    cloop:  call   sndcr           ;crlf
C0AB E84C04            call   sndmsg          ;prompt
C0AE E8B604            call   getnm           ;10's digit
C0B1 AA                stosb
C0B2 E8B204            call   getnm           ;1's digit
C0B5 AA                stosb
C0B6 E2F0              loop   cloop
C0B8 8A450B            mov    al,11[di]       ;year
C0BB 2403              and    al,#3           ;leap year indix
C0BD D0E0              shlb   al,1
C0BF D0E0              shlb   al,1
C0C1 FEC0              incb   al              ;set 24-hour mode
C0C3 A20F60            mov    [rtcsir],al
C0C6 E82704            call   esram           ;reset es
C0C9 EB9A              jmp    pro
               ;
               ; Start real time clock
C0CB C606006001clstrt: mov    [rtccr],#1      ;start clock
C0D0 EB93              jmp    pro
               ;
               ; Display time
C0D2 E86E05    disptm: call   tmdisp          ;display clock time
C0D5 EB8E              jmp    pro
               ;
               ; Read in shot table
C0D7 BA91CE    stread: mov    dx,#msgr
C0DA E81D04            call   sndmsg          ;output mesage
C0DD E88D04            call   gethx           ;obs number
C0E0 A20800            mov    [obsno],al
C0E3 BF0004            mov    di,#shtab       ;shot table address
C0E6 FC                cld                    ;increment di with store
C0E7 E8DE04            call   timein          ;release time
C0EA E8DB04    rloop:  call   timein          ;wake up time
C0ED 3C24              cmp    al,#"$"         ;end of table?
C0EF 740B              je     rend
C0F1 E8D404            call   timein          ;start time
C0F4 E8E904            call   parin           ;recording parameters
C0F7 E8CE04            call   timein          ;stop time
C0FA EBEE              jmp    rloop
C0FC B00F      rend:   mov    al,#0fh
C0FE AA                stosb
C0FF AA                stosb
C100 AA                stosb
C101 E88204            call   rststp          ;reset shot table pointer and index
C104 E95EFF            jmpf   pro
               ;
               ; Display shot table
C107 BAAFCE    stdisp: mov    dx,#msgt0
C10A E8ED03            call   sndmsg          ;obs no. msg
C10D A00800            mov    al,[obsno]
C110 E82904            call   sndhx
C113 BE0004            mov    si,#shtab       ;shot table address
C116 FC                cld                    ;increment si with load
C117 BAC2CE            mov    dx,#msgt1
```

```
C11A E8DD03             call   sndmsg          ;release time msg
C11D E8E504             call   disptt          ;display release time
C120 BAD4CE    sloop:   mov    dx,#msgt2
C123 E8D403             call   sndmsg          ;wake-up time msg
C126 E8DC04             call   disptt          ;display wake-up time
C129 E8CE03             call   sndmsg          ;record start time msg
C12C E8D604             call   disptt          ;display record start time
C12F E8C803             call   sndmsg          ;parameter msg
C132 E8E304             call   dispp           ;display parameters
C135 E8C203             call   sndmsg          ;stop time msg
C138 E8CA04             call   disptt          ;display stop time
C13B 26                 seg    es              ;segment override
C13C 8A04               mov    al,[si]
C13E 3C0F               cmp    al,#0fh         ;last entry?
C140 75DE               jne    sloop           ; no
C142 E920FF             jmpf   pro
                ;
                ; Tape drive test
C145 BA36CD    tdtst:   mov    dx,#menum
C148 E8AF03             call   sndmsg          ;display drive test menu
C14B C6061D00FE         mov    [intf],#0feh    ;direction in case of tape problem
C150 E8CC03    prom:    call   sndcr
C153 B02A               mov    al,#"*"
C155 E8B303             call   sndchr          ;prompt
C158 E8F803             call   getchr          ;get a command character
C15B 3C68               cmp    al,#"h"         ;help
C15D 74E6               je     tdtst
C15F 3C72               cmp    al,#"r"         ;read registers
C161 7431               je     rdreg
C163 3C77               cmp    al,#"w"         ;write to register
C165 7443               je     wtreg
C167 3C73               cmp    al,#"s"         ;select tape drive
C169 744B               je     tsel
C16B 3C70               cmp    al,#"p"         ;put cdb
C16D 7465               je     put
C16F 3C67               cmp    al,#"g"         ;get data/status/message
C171 747F               je     get
C173 3C69               cmp    al,#"i"         ;initialize drive
C175 7444               je     tini
C177 3C6F               cmp    al,#"o"         ;output to drive
C179 7445               je     tout
C17B 3C66               cmp    al,#"f"         ;write file mark
C17D 744C               je     tfm
C17F 3C61               cmp    al,#"a"         ;test acquisition
C181 744D               je     tacqs
C183 3C63               cmp    al,#"c"         ;preamp check
C185 744B               je     packs
C187 3C71               cmp    al,#"q"         ;quit drive test
C189 75C5               jne    prom
                ;
C18B E89205             call   tmpoff          ;turn off tape motor power
C18E E8B705             call   tlpoff          ;turn off tape logic power
C191 E9D1FE             jmpf   pro             ;return to main menu
                ;
                ; Read SCSI registers
C194 30C0      rdreg:   xor    al,al
C196 E8CB04    rrlp:    call   screg           ;get scsi register address
C199 50                 push   ax
C19A E88B03             call   sndsp           ;space
C19D EC                 in     al,dx
C19E E89B03             call   sndhx           ;hex output of register content
C1A1 58                 pop    ax
C1A2 FEC0               incb   al
C1A4 3C08               cmp    al,#8
C1A6 75EE               jne    rrlp            ;repeat for 8 registers
C1A8 EBA6               jmp    prom
                ;
                ; Write to SCSI register
C1AA E8BA03    wtreg:   call   getnm           ;get register number
C1AD E8B404             call   screg           ;get scsi register address
C1B0 E8BA03             call   gethx           ;get hex byte
C1B3 EE                 out    dx,al           ;write to scsi register
```

```
C1B4 EB9A               jmp    prom
                ;
                ; Test select tape drive
C1B6 E8B805     tsel:   call   tdsel
C1B9 EB95               jmp    prom
                ;
                ; Test initialize drive
C1BB E8D304     tini:   call   ctinie
C1BE EB90               jmp    prom
                ;
                ; Test output to drive
C1C0 E8AA03     tout:   call   gethx          ;get number of blocks in hex
C1C3 A20300             mov    [blks],al
C1C6 E80707             call   write
C1C9 EB85               jmp    prom
                ;
                ; Test write file mark
C1CB E8E007     tfm:    call   wreof
C1CE EB80               jmp    prom
                ;
C1D0 EB44       tacqs:  jmp    tacq
C1D2 EB60       packs:  jmp    pack
                ;
                ; Put CDB
C1D4 BF1000     put:    mov    di,#cdb        ;CDB address
C1D7 E81B03             call   esbrd          ; on board
C1DA FC                 cld
C1DB B90600             mov    cx,#6          ;6-byte CDB
C1DE 51         ptlp:   push   cx
C1DF E88B03             call   gethx          ;get hex byte
C1E2 AA                 stosb                 ;put in cdb
C1E3 59                 pop    cx
C1E4 E2F8               loop   ptlp
C1E6 E80703             call   esram          ;reset es
C1E9 BE1000             mov    si,#cdb
C1EC E8E005             call   sndcdb         ;output cdb
C1EF E95EFF             jmpf   prom
                ;
                ; Get data/status/message
C1F2 BA6464     get:    mov    dx,#scsi4      ;bus status
C1F5 EC                 in     al,dx
C1F6 A820               test   al,#20h        ;REQ?
C1F8 7503               jnz    grq            ; yes
C1FA E953FF     jprm:   jmpf   prom           ; no
C1FD A804       grq:    test   al,#4          ;input?
C1FF 74F9               jz     jprm           ; no
C201 D0E8               shrb   al,1
C203 D0E8               shrb   al,1
C205 2407               and    al,#7          ;isolate phase
C207 BA6363             mov    dx,#scsi3      ; and put in tgt cmd register
C20A EE                 out    dx,al
C20B E81A03             call   sndsp          ;output space
C20E E89906             call   getbyt         ;get a byte
C211 E82803             call   sndhx          ;output in hex
C214 EBDC               jmp    get            ;repeat till no input REQ
                ;
                ; Test data acquisition
C216 E84E03     tacq:   call   getnm          ;get number of channels
C219 A20400             mov    [chans],al
C21C E84803             call   getnm          ;get sampling interval
C21F E83A01             call   sampic         ;convert to interval count
C222 E84803             call   gethx          ;get number of blocks
C225 A20300             mov    [blks],al
C228 BA79CF             mov    dx,#msgaq
C22B E8CC02             call   sndmsg
C22E E84F01             call   record         ;acquire data
C231 E91CFF             jmpf   prom
                ;
                ; ADC power on for preamp check
C234 BA24CF     pack:   mov    dx,#msgpa      ;adc power-on message
C237 E8C002             call   sndmsg
C23A BA2222             mov    dx,#qadc2      ;qadc power-on port
```

```
C23D EE                 out   dx,al          ;adc power on
C23E E81203             call  getchr         ;hold till keyboard input
C241 BA2020             mov   dx,#qadc0      ;qadc power-off port
C244 EE                 out   dx,al          ;adc power off
C245 E908FF             jmpf  prom
                ;
                ; Go
C248 C606006003go:      mov   [rtccr],#3     ;select rtc interrupt register
C24D C6060F600F         mov   [rtcsir],#0fh  ;60 sec repeated interrupt
C252 BE0360     glp1:   mov   si,#rtc+3      ;point to deca sec
C255 B90300             mov   cx,#3          ;test last 3 digits
C258 E84D03             call  clkud          ;wait for clock update
C25B 803C00     glp2:   cmp   [si],#0
C25E 75F2               jne   glp1           ;wait till 00.0 sec
C260 4E                 decw  si
C261 E2F8               loop  glp2
C263 C606006000         mov   [rtccr],#0     ;start rtc interrupt
C268 E8D803             call  tmdisp         ;display current time
C26B BA55CF             mov   dx,#msggn      ;Good night.
C26E E88902             call  sndmsg
C271 C606004202         mov   [urtmc],#2     ;stop uart
C276 C6061D0000g1:      mov   [intf],#0
C27B FB                 sti                  ;enable cpu interrupt
C27C A00080             mov   al,[wait]      ;go to sleep
C27F 90                 nop
C280 90                 nop
C281 803E1D0000         cmp   [intf],#0      ;test interrupt flag
C286 7CC0               jl    go             ;acq. return - reset rtc interrupt
C288 7FEC               jg    g1             ;non-acq. return - go back to sleep
C28A FA                 cli                  ;disable interrupt while in dialog
C28B C606004220         mov   [urtmc],#20h   ;restart uart
C290 E8E703             call  dly30          ;wait for uart to settle
C293 E8AD03             call  tmdisp         ;display current time
C296 BA66CF             mov   dx,#msgwu      ;wakeup message
C299 E85E02             call  sndmsg
C29C E9C6FD             jmpf  pro
                ;
                ;
                ; ***** Interrupt Service Routines *****
                ;
                ; key interrupt service
C29F CF         keyint: iret
                ;
                ; rtc interrupt service
C2A0 A00060     rtcint: mov   al,[rtccr]     ;clear rtc interrupt flag
C2A3 803E1D0000         cmp   [intf],#0
C2A8 7D03               jge   wkup           ;regular wake up if intf=0
C2AA E9F501             jmpf  twprb          ; otherwise tape write problem
C2AD C606004200wkup:    mov   [urtmc],#0     ;start uart with receiver disabled
C2B2 E88E03             call  tmdisp         ;display current time
C2B5 BF0004             mov   di,#shtab      ;release time (mmddhhmm)
C2B8 E89203             call  tmchk          ;check if release time reached
C2BB 752B               jne   rels
C2BD 8B3E1E00           mov   di,[stptr]     ;shot table entry
C2C1 E88903             call  tmchk          ;check time
C2C4 7415               je    nyst           ;not yet
C2C6 803E200000         cmp   [tabix],#0     ;power up or data acquisition?
C2CB 7530               jnz   dataq          ;data acquisition
C2CD E8BB03             call  ctinit         ;initialize tape drive
C2D0 7509               jnz   nyst           ;initialization failed, try minute later
C2D2 83061E0008         add   [stptr],#8     ;advance shot table pointer
C2D7 FE062000           incb  [tabix]        ;and index
C2DB E84102     nyst:   call  sndcr
C2DE FE061D00   nar:    incb  [intf]         ;intf>0 for non-acq. interrupt return
C2E2 C606004202rtir:    mov   [urtmc],#2     ;stop uart
C2E7 CF                 iret
                ;
                ; Release
C2E8 C606030422rels:    mov   [piopc],#22h   ;release on (pc1 & 5 - pins 15 & 16)
C2ED BAB2CF             mov   dx,#msgrl
C2F0 E80702             call  sndmsg
C2F3 E87E03             call  dly10          ;after 15+10 ms
```

```
C2F6 C606030400          mov    [piopc],#0      ;release off
C2FB EBE1                jmp    nar
                ;
                ; Data acquisition
C2FD BA79CF     dataq:   mov    dx,#msgaq
C300 E8F701              call   sndmsg
C303 C606006001          mov    [rtccr],#1      ;stop rtc interrupt
C308 83061E0008          add    [stptr],#8      ;advance pointer to parameters
C30D 8B3E1E00            mov    di,[stptr]      ;point to shot parameters
C311 FC                  cld
C312 26                  seg    es
C313 8A05                mov    al,[di]         ;number of channels
C315 A20400              mov    [chans],al
C318 47                  incw   di
C319 26                  seg    es
C31A 8A05                mov    al,[di]         ;sampling interval
C31C E83D00              call   sampic          ;convert to interval count
C31F 47                  incw   di
C320 26                  seg    es
C321 8A05                mov    al,[di]         ;number of blocks
C323 A20300              mov    [blks],al
C326 83061E0004          add    [stptr],#4      ;advance pointer to stop time
C32B C6061D00FF          mov    [intf],#0ffh    ;intf=-1 for aquisition
C330 E84D00     rcdlp:   call   record          ;acquire a record
C333 8B3E1E00            mov    di,[stptr]      ;point to stop time
C337 E81303              call   tmchk           ;check for stop time
C33A 74F4                je     rcdlp           ;continue with next record
C33C E80403              call   tmdisp          ;display current time
C33F BA95CF              mov    dx,#msgac       ;completion message
C342 E8B501              call   sndmsg
C345 E86606              call   wreof           ;write eof
C348 E89806              call   rewind          ;rewind tape
C34B E8D203              call   tmpoff          ;turn off drive motor power
C34E E8F703              call   tlpoff          ;turn off tape logic power
C351 83061E0008          add    [stptr],#8      ;advance pointer to next power-up
C356 C606200000          mov    [tabix],#0      ;clear table index
C35B CF                  iret
                ;
                ; sampling interval count
C35C A20500     sampic:  mov    [sampi],al
C35F 88C3                mov    bl,al           ;save si temporarily in bl
C361 F62603C0            mulb   [c153]          ;si x 153
C365 93                  xchg   ax,bx           ;save in bx and get si
C366 F62600C0            mulb   [c6]            ;si x 6
C36A 050500              add    ax,#5           ;+5
C36D F63601C0            divb   [c10]           ;/10
C371 30E4                xor    ah,ah           ;clear remainder
C373 01D8                add    ax,bx
C375 80CCC0              or     ah,#0c0h        ;set to continuous pulse mode
C378 A20404              mov    [piocl],al
C37B 88260504            mov    [pioch],ah
C37F C3                  ret
                ;
                ; Acquire a record
C380 BA2222     record:  mov    dx,#qadc2       ;qadc power-on port
C383 EE                  out    dx,al           ;adc power on
C384 BA2424              mov    dx,#qadc4       ;status 1 port
C387 EC         cals:    in     al,dx           ;get status 1
C388 D0C0                rolb   al,1            ;cal bit in c
C38A 73FB                jnc    cals            ;wait for calibration to start
C38C EC         calf:    in     al,dx           ;get status 1
C38D D0C0                rolb   al,1            ;cal bit in c
C38F 72FB                jc     calf            ;wait for calibration to finish
C391 B090                mov    al,#90h         ;bipolar, single-ended, channel 0
C393 BA2828              mov    dx,#qadc8       ;gain/channel port
C396 EE                  out    dx,al           ;select ch 1 low gain
C397 C606210040          mov    [bank],#40h     ;start from bank 0
C39C E8F301              call   rdclk           ;read clock
C39F FF060000            incw   [recno]         ;update record number
C3A3 C6060200 01         mov    [blkno],#1      ;initialize block number
C3A8 BE1C00              mov    si,#tbuf+12     ;decimal time address
C3AB BF0900              mov    di,#hdrtim      ;header time address
```

```
C3AE FD                 std
C3AF B90600             mov     cx,#6
C3B2 AC         htlp:   lodsb
C3B3 F62601C0           mulb    [c10]           ;convert two decimal digits
C3B7 0204               add     al,[si]         ; to binary
C3B9 8805               mov     [di],al         ;  and put in header
C3BB 4E                 decw    si
C3BC 47                 incw    di
C3BD E2F3               loop    htlp            ;for year, month, day, hour and min.
C3BF 8A04               mov     al,[si]
C3C1 8805               mov     [di],al         ;ds
C3C3 FC                 cld
C3C4 B80040             mov     ax,#4000h
C3C7 8EC0               mov     es,ax           ;initial output segment
C3C9 31FF               xor     di,di           ;initial header output offset
C3CB A00004             mov     al,[piocsr]     ;clear timer flag
C3CE C6060004CE         mov     [piocsr],#0ceh ;start timer
C3D3 BAFCFC     bnklp:  mov     dx,#ramp        ;RAM-2M port
C3D6 A02100             mov     al,[bank]
C3D9 EE                 out     dx,al           ;select bank
C3DA 89F8       blklp:  mov     ax,di           ;display block address
C3DC B104               mov     cl,#4
C3DE D3C0               rolw    ax,cl
C3E0 0430               add     al,#"0"
C3E2 E82601             call    sndchr
C3E5 31F6               xor     si,si           ;header address
C3E7 B91000             mov     cx,#16
C3EA F3                 rep
C3EB A4                 movsb                   ;transfer 16-byte header
C3EC A00080     aqloop: mov     al,[wait]       ;go to sleep
C3EF 90                 nop                     ;and wait for timer interrupt
C3F0 A00004             mov     al,[piocsr]     ;reset timer interrupt
C3F3 E84200             call    smple
C3F6 F7C7F00F           test    di,#0ff0h       ;end of block?
C3FA 75F0               jnz     aqloop          ; no, continue
C3FC A00200             mov     al,[blkno]      ; yes
C3FF 3A060300           cmp     al,[blks]       ;last block?
C403 7420               je      aqcmpl          ; yes, acquisition complete
C405 FE060200           incb    [blkno]         ; no, update block no.
C409 81FF0000           cmp     di,#0           ;end of segment?
C40D 75CB               jne     blklp           ; no, continue with next block
C40F 8CC0               mov     ax,es           ; yes
C411 F7D8               negw    ax
C413 050090             add     ax,#9000h
C416 8EC0               mov     es,ax           ;change segment
C418 3D0050             cmp     ax,#5000h       ;second segment of the bank?
C41B 74BD               je      blklp           ; yes, continue with next block in new segment
C41D 8306210010         add     [bank],#10h     ; no, update bank
C422 E9AEFF             jmpf    bnklp
C425 C60600044Eaqcmpl:  mov     [piocsr],#4eh   ;stop timer
C42A BA2020             mov     dx,#qadc0       ;power-off port
C42D EE                 out     dx,al           ;turn off adc power
C42E E81202             call    tmdisp          ;display current time
C431 E8AD00             call    rstsb           ;reset es and bank
C434 E89904             call    write           ;output data to tape
C437 C3                 ret
                ;
                ; Smpling with adc
C438 30C9       smple:  xor     cl,cl           ;clear channel count
C43A BA2626     loopad: mov     dx,#qadc6       ;convert port
C43D EE                 out     dx,al           ;start convertion for low gain
C43E BA2020             mov     dx,#qadc0       ;data port
C441 A20080             mov     [wait],al       ;put to wait mode
C444 90                 nop
C445 EC                 in      al,dx           ;get high byte
C446 88C4               mov     ah,al
C448 EC                 in      al,dx           ;get low byte
C449 2D0080             sub     ax,#8000h       ;remove offset
C44C 7902               jns     l1              ;if negative
C44E F7D8               negw    ax              ; take abs
C450 86C4       l1:     xchg    ah,al
C452 D1C0               rolw    ax,1
```

```
C454 D1C0              rolw   ax,1           ;/16
C456 25FF01            and    ax,#1ffh       ;9 msb
C459 BE46CA            mov    si,#gtab       ;gain table
C45C 01C6              add    si,ax
C45E 8A3C              mov    bh,[si]        ;obtain proper gain
C460 88F8              mov    al,bh
C462 30E4              xor    ah,ah
C464 BE46CC            mov    si,#ctab       ;channel table
C467 01C6              add    si,ax
C469 88C8              mov    al,cl          ;data channel
C46B D0E0              shlb   al,1           ;in bits 1-2
C46D 0A04              or     al,[si]        ;add gain channel bits
C46F BA2828            mov    dx,#qadc8      ;gain/channel port
C472 EE                out    dx,al          ;select proper adc channel
C473 FEC1              incb   cl             ;advance channel count
C475 3A0E0400          cmp    cl,[chans]     ;channels complete?
C479 7C02              jl     l2             ;no
C47B 30C9              xor    cl,cl          ;yes, reset channel count
C47D 88CD      l2:     mov    ch,cl
C47F D0E5              shlb   ch,1           ;prepare for next data channel
C481 80CD90            or     ch,#90h        ;with bipolar, single-ended, low-gain
C484 BA2424            mov    dx,#qadc4      ;status 1//gain/channel/convert port
C487 EC        l3:     in     al,dx          ;get status 1
C488 2408              and    al,#8          ;test end-of-track bit
C48A 75FB              jne    l3             ;wait for input signal acquisition
C48C 88E8              mov    al,ch          ;next low-gain channel
C48E EE                out    dx,al          ;select next channel and start conversion
C48F A20080            mov    [wait],al      ;put to wait mode
C492 90                nop
C493 BA2020            mov    dx,#qadc0      ;data port
C496 EC                in     al,dx          ;get high byte
C497 88C4              mov    ah,al
C499 EC                in     al,dx          ;get low byte and start conversion
C49A 08F8              or     al,bh          ;add gain bits
C49C AB                stosw                 ;store result
C49D 08C9              or     cl,cl          ;last channel?
C49F 7599              jne    loopad         ;no, sample next channel
C4A1 C3                ret
                ;
                ; Tape write problem - abort
C4A2 C606006001twprb:  mov    [rtccr],#1     ;stop rtc interrupt
C4A7 E83700            call   rstsb          ;reset segment reg. & bank
C4AA E8B602            call   rstsc          ;reset SCSI bus
C4AD BC0001            mov    sp,#stack      ;reset stack pointer
C4B0 BAA7D0            mov    dx,#msgto      ;timeout abort message
C4B3 E84400            call   sndmsg
C4B6 803E1D00FF        cmp    [intf],#0ffh
C4BB 7403              je     rtng
C4BD E990FC            jmpf   prom           ;return to 2nd menu prompt
C4C0 832E1E000Crtng:   sub    [stptr],#12    ;roll back shot table pointer
C4C5 E98AFD            jmpf   glp1           ;go back to sleep
                ;
                ; ADC interrupt
C4C8 CF         adcint: iret
                ;
                ; Sampling interrupt
C4C9 CF         smpint: iret
                ;
                ; Misdirected interrupts
C4CA B032       md2int: mov    al,#"2"
C4CC EB0A               jmp    bimsg
C4CE B033       md3int: mov    al,#"3"
C4D0 EB06               jmp    bimsg
C4D2 B036       md6int: mov    al,#"6"
C4D4 EB02               jmp    bimsg
C4D6 B037       md7int: mov    al,#"7"
C4D8 E83000     bimsg:  call   sndchr
C4DB BAC4CF             mov    dx,#msgmi
C4DE E81900             call   sndmsg
                ;
                ;
                ; ***** Subroutines *****
```

```
                        ;
                        ; Reset ds(on-board), es(RAM-2M) and bank(0)
C4E1 E80C00     rstsb:  call    esram           ;set es to ram
C4E4 BAFCFC             mov     dx,#ramp        ;RAM port
C4E7 B040               mov     al,#40h         ;board 0, 256K bank
C4E9 EE                 out     dx,al           ;access RAM-2M port at bank 0
C4EA B800F0             mov     ax,#board
C4ED 8ED8               mov     ds,ax           ;set ds to on-board
C4EF C3                 ret
                        ;
                        ; Set es to ram
C4F0 31C0       esram:  xor     ax,ax
C4F2 8EC0       es1:    mov     es,ax
C4F4 C3                 ret
                        ;
                        ; Set es to on-board
C4F5 B800F0     esbrd:  mov     ax,#board
C4F8 EBF8               jmp     es1
                        ;
                        ; Send a message (@dx) out to UART
C4FA 87D6       sndmsg: xchg    si,dx
C4FC 8A04       mloop:  mov     al,[si]         ;get a character
C4FE 46                 incw    si
C4FF 3C24               cmp     al,#"$"         ;terminator?
C501 7405               je      mout            ;yes
C503 E80500             call    sndchr          ;output a character
C506 EBF4               jmp     mloop
C508 87F2       mout:   xchg    dx,si
C50A C3                 ret
                        ;
                        ; Send a character (in al) out to UART
C50B A20040     sndchr: mov     [urtbr],al      ;send a character
C50E A00041     swt:    mov     al,[urtcs]      ;get status
C511 D0C0               rolb    al,1            ;any data left over?
C513 7305               jnc     sc1             ;no
C515 A00040             mov     al,[urtbr]      ;yes, discard it
C518 EBF4               jmp     swt
C51A 2440       sc1:    and     al,#40h         ;complete?
C51C 74F0               je      swt             ;wait till complete
C51E C3                 ret
                        ;
                        ; Send CR & LF out ot UART
C51F B00D       sndcr:  mov     al,#cr          ;CR
C521 E8E7FF             call    sndchr
C524 B00A               mov     al,#lf          ;LF
C526 EBE3               jmp     sndchr
                        ;
                        ; Send space
C528 B020       sndsp:  mov     al,#" "
C52A EBDF               jmp     sndchr
                        ;
                        ; Send numeric character
C52C 0430       sndnm:  add     al,#"0"         ;convert to ASCII
C52E EBDB               jmp sndchr
                        ;
                        ; Send space or numeral
C530 08C9       sndsn:  or      cl,cl
C532 75F8               jne     sndnm           ;leading non-zero
C534 08C0               or      al,al
C536 74F0               je      sndsp           ;zero - send space
C538 FEC1               incb    cl              ;set non-zero flag
C53A EBF0               jmp     sndnm           ;non-zero - send numeral
                        ;
                        ; Send hex characters
C53C 50         sndhx:  push    ax
C53D B104               mov     cl,#4
C53F D2F8               sarb    al,cl
C541 E80500             call    hexout
C544 58                 pop     ax
C545 E80100             call    hexout
C548 C3                 ret
C549 240F       hexout: and     al,#0fh
```

```
C54B 3C09               cmp    al,#9
C54D 7EDD               jle    sndnm
C54F 0407               add    al,#7
C551 EBD9               jmp    sndnm
                ;
                ; Get a character (in al) from UART
C553 A00041     getchr: mov    al,[urtcs]      ;get status
C556 D0C0               rolb   al,1            ;data ready?
C558 73F9               jnc    getchr          ;no, wait till data ready
C55A A00040             mov    al,[urtbr]      ;get the character
C55D 3C74               cmp    al,#"t"
C55F 7405               je     grtn            ;no echo if "t"
C561 50                 push   ax
C562 E8A6FF             call   sndchr          ;echo
C565 58                 pop    ax
C566 C3         grtn:   ret
                ;
                ; Get a character and convert to numeric
C567 E8E9FF     getnm:  call   getchr          ;get a character
C56A 2C30               sub    al,#"0"         ;convert to binary
C56C C3                 ret
                ;
                ; Get hex characters and convert to binary
C56D E80C00     gethx:  call   hexin
C570 B104               mov    cl,#4
C572 D2E0               shlb   al,cl
C574 88C1               mov    cl,al
C576 E80300             call   hexin
C579 08C8               or     al,cl
C57B C3                 ret
C57C E8E8FF     hexin:  call   getnm
C57F 3C09               cmp    al,#9
C581 7E02               jle    hxrtn
C583 2C27               sub    al,#39
C585 C3         hxrtn:  ret
                ;
                ; Reset shot table pointer
C586 C7061E0008rststp:  mov    [stptr],#shtab+8      ;first shot group
C58C C606200000         mov    [tabix],#0      ; reset index
C591 C3                 ret
                ;
                ; Read clock
C592 BE0D60     rdclk:  mov    si,#rtc+13      ;start with deca year
C595 BF1C00             mov    di,#tbuf+12     ;time buffer in RAM
C598 E85AFF             call   esbrd           ;es on board
C59B FD                 std                    ; and decrement
C59C B90D00             mov    cx,#13          ;for 13 bytes
C59F E80600             call   clkud           ;wait for clock update
C5A2 F3                 rep
C5A3 A4                 movsb                  ;get clock data
C5A4 E849FF             call   esram           ;reset es to ram
C5A7 C3                 ret
                ;
                ; Wait for clock update
C5A8 A00060     clkud:  mov    al,[rtccr]      ;reset control reg
C5AB A00060     wtupd:  mov    al,[rtccr]
C5AE 2408               and    al,#8           ;test for DCF
C5B0 74F9               je     wtupd           ;wait for update
C5B2 C3                 ret
                ;
                ; Display clock time
C5B3 B054       dispck: mov    al,#"T"
C5B5 E853FF             call   sndchr          ;signal start of time data
C5B8 BE1C00             mov    si,#tbuf+12     ;start with deca year
C5BB B90D00             mov    cx,#13          ; and for 13 bytes
C5BE AC         dloop:  lodsb                  ;get a byte
C5BF E86AFF             call   sndnm           ;and output
C5C2 E2FA               loop   dloop
C5C4 E861FF             call   sndsp
C5C7 C3                 ret
                ;
                ; Table time input
```

```
C5C8 E854FF    timein: call   sndcr
C5CB E885FF    tloop:  call   getchr          ;get a character
C5CE 3C24              cmp    al,#"$"
C5D0 740D              je     tirtn           ;return if end of table
C5D2 3C20              cmp    al,#" "
C5D4 74F5              je     tloop           ;skip if space
C5D6 3C0D              cmp    al,#cr
C5D8 7405              je     tirtn           ;return if cr
C5DA 2C30              sub    al,#"0"         ;convert to binary
C5DC AA                stosb                  ;and store
C5DD EBEC              jmp    tloop
C5DF C3        tirtn:  ret
               ; Table parameter input
C5E0 E83CFF    parin:  call   sndcr
C5E3 31D2      ploop:  xor    dx,dx           ;clear dx
C5E5 E86BFF    ploop2: call   getchr          ;get a character
C5E8 3C2C              cmp    al,#","
C5EA 7410              je     pout            ;if comma, store parameter
C5EC 3C0D              cmp    al,#cr
C5EE 7411              je     pout2           ;if cr, store par. and return
C5F0 2C30              sub    al,#"0"         ;convert to binary
C5F2 86D0              xchg   al,dl           ;save in dl and get dl
C5F4 F62601C0          mulb   [c10]           ;multiply previous value by 10
C5F8 01C2              add    dx,ax           ;and add to new digit
C5FA EBE9              jmp    ploop2
C5FC 88D0      pout:   mov    al,dl
C5FE AA                stosb                  ;save binary byte
C5FF EBE2              jmp    ploop
C601 89D0      pout2:  mov    ax,dx
C603 AB                stosw                  ;save binary word
C604 C3                ret
               ;
               ; Dipsplay shot table time
C605 B90400    disptt: mov    cx,#4
C608 E81DFF    dtloop: call   sndsp
C60B 26                seg    es              ;override segment es for input
C60C AD                lodsw                  ;get two bytes
C60D E81CFF            call   sndnm           ;output first numeral
C610 88E0              mov    al,ah
C612 E817FF            call   sndnm           ;output second numeral
C615 E2F1              loop   dtloop          ;repeat for 4 sets of numerals
C617 C3                ret
               ;
               ; Display shot table parameters
C618 E80700    dispp:  call   decb            ;output sample rate
C61B E80400            call   decb            ;output number of channels
C61E 26                seg    es
C61F AD                lodsw                  ;get two-byte number of blocks
C620 EB04              jmp    decpr           ;and output
C622 26        decb:   seg    es
C623 AC                lodsb                  ;get a byte
C624 30E4              xor    ah,ah           ;clear ah
C626 30C9      decpr:  xor    cl,cl           ;clear none-zero flag
C628 F63602C0          divb   [c100]          ;100's digit
C62C E801FF            call   sndsn           ;space or numeral
C62F 88E0              mov    al,ah           ;remainder
C631 30E4              xor    ah,ah           ;clear ah
C633 F63601C0          divb   [c10]           ;10's digit
C637 E8F6FE            call   sndsn
C63A 88E0              mov    al,ah           ;1's digit
C63C E8EDFE            call   sndnm
C63F E8E6FE            call   sndsp
C642 C3                ret
               ;
               ; Time display
C643 E84CFF    tmdisp: call   rdclk           ;read clock
C646 E86AFF            call   dispck          ;display clock time
C649 E82800            call   dly10           ;wait 10 ms for possible clock output
C64C C3                ret
               ;
               ; time check - zero flag cleared if time reached or passed
C64D BE0B60    tmchk:  mov    si,#rtc+11      ;current time (mmddhhmm)
```

```
C650 FC                 cld                     ;mo->min
C651 30C0               xor     al,al           ;clear flag
C653 B90800             mov     cx,#8           ;up to 8 digits to test
C656 A6         tclp:   cmpsb                   ;check time
C657 7C08               jl      ny              ;not yet
C659 7F04               jg      psd             ;passed
C65B 4E                 decw    si
C65C 4E                 decw    si
C65D E2F7               loop    tclp
C65F FEC0       psd:    incb    al              ;time reached or passed, set flag
C661 08C0       ny:     or      al,al           ;set/reset zero flag
C663 C3                 ret
                ;
                ; SCSI register I/O address
C664 BA6060     screg:  mov     dx,#scsi
C667 00C6               add     dh,al
C669 00C2               add     dl,al
C66B C3                 ret
                ;
                ; 25 us delay
C66C 51         wt25:   push    cx
C66D B90200             mov     cx,#2
C670 E2FE       lp25:   loop    lp25
C672 59                 pop     cx
C673 C3                 ret
                ;
                ; 10 ms delay
C674 51         dly10:  push cx
C675 B94D01             mov     cx,#333
C678 EB04               jmp     dllp
                ; 30 ms delay
C67A 51         dly30:  push    cx
C67B B9E703             mov     cx,#999
C67E 50         dllp:   push    ax
C67F D40A               aam
C681 58                 pop     ax
C682 E2FA               loop    dllp            ;30.3 us/loop
C684 59                 pop     cx
C685 C3                 ret
                ;
                ; 2 sec delay
C686 51         dly2s:  push    cx
C687 31C9               xor     cx,cx
C689 EBF3               jmp     dllp
                ;
                ;
                ; ***** Cartridge Tape Driver Subroutines *****
                ;
                ; Initialize cartrige tape drive
C68B BA66CF     ctinit: mov     dx,#msgwu
C68E E869FE             call    sndmsg
C691 E82800     ctinie: call    tmpon           ;turn on tape motor power
C694 7525               jnz     cirtn           ;no motor power
C696 E85600             call    tlpon           ;turn on tape logic power
C699 7520               jnz     cirtn           ;no logic power
C69B BA3ED0             mov     dx,#msgit
C69E E859FE             call    sndmsg
C6A1 E8BF00             call    rstsc           ;reset SCSI bus
C6A4 E8DFFF             call    dly2s           ;wait for 2 s
C6A7 E8C700             call    tdsel           ;select tape drive
C6AA 750F               jnz     cirtn           ;select error
C6AC E8F100             call    turdy           ;take care of unit attenstion
C6AF E8BF00             call    tdsel           ;select drive again
C6B2 7507               jnz     cirtn           ;select error
C6B4 E8F800             call    enddat          ;skip to end of data
C6B7 7502               jnz     cirtn           ;failed to find end of data
C6B9            pwoff:; call    tmpoff          ;turn off tape motor power
C6B9 30C0               xor     al,al           ;clear error flag
C6BB C3         cirtn:  ret
                ;
                ; Turn on tape drive motor power
C6BC B90A00     tmpon:  mov     cx,#10          ;retry count
```

```
C6BF C606020402mplp:    mov    [piopb],#2      ;motor power on (pb1 - pin 31)
C6C4 E8B3FF             call   dly30           ;wait 30 ms
C6C7 F606010402         test   [piopa],#2      ;sense motor power (pa1 - pin 30)
C6CC 750A               jnz    m1              ;not on
C6CE BADCCF             mov    dx,#msgpu
C6D1 E826FE             call   sndmsg
C6D4 30C0               xor    al,al           ;on, clear error flag
C6D6 EB11               jmp    tprtn
C6D8 C606020408m1:      mov    [piopb],#8      ;motor power off (pb3 - pin 27)
C6DD E89AFF             call   dly30           ;wait 30 ms
C6E0 E2DD               loop   mplp            ;try again
C6E2 BAC4D0             mov    dx,#msge1       ;error 1 message (no motor power)
C6E5 E812FE     em1:    call   sndmsg
C6E8 42         sf1:    incw   dx              ;set error flag
C6E9 C606020400tprtn:   mov    [piopb],#0      ;clear tape power command
C6EE C3                 ret
                ;
                ; Turn on tape drive logic power
C6EF B90A00     tlpon:  mov    cx,#10          ;retry count
C6F2 C606020420lplp:    mov    [piopb],#20h    ;logic power on (pb5 - pin 23)
C6F7 E880FF             call   dly30           ;wait 30 ms
C6FA F606010420         test   [piopa],#20h    ;sense logic power (pa5 - pin 22)
C6FF 750A               jnz    lg1             ;not on
C701 BAF4CF             mov    dx,#msglu
C704 E8F3FD             call   sndmsg
C707 30C0               xor    al,al           ;on, clear error flag
C709 EBDE               jmp    tprtn
C70B C606020480lg1:     mov    [piopb],#80h    ;logic power off (pib7 - pin 19)
C710 E867FF             call   dly30           ;wait 30 ms
C713 E2DD               loop   lplp            ;try again
C715 BADBD0             mov    dx,#msge2       ;error 2 message (no logic power)
C718 E8DFFD     em2:    call   sndmsg
C71B E80200             call   tmpoff          ;turn off tape motor power
C71E EBC8               jmp    sf1             ;set error flag and return
                ;
                ; Turn off tape drive motor power
C720 F606010420tmpoff:  test   [piopa],#20h    ;logic power on?
C725 7503               jnz    n1              ;no
C727 E85CFF             call   dly2s           ;if logic power on, wait 2 s
C72A B93629     n1:     mov    cx,#10550       ;200 ms time out count
C72D C606020408molp:    mov    [piopb],#8      ;motor power off (pb3 - pin 27)
C732 F606010402         test   [piopa],#2      ;sense motor power (pa1 - pin 30)
C737 7507               jnz    n2              ;power is now off
C739 E2F2               loop   molp            ;wait till power off, 18.76 us/loop
C73B BAF2D0             mov    dx,#msge3       ;error 3 message (motor power left on)
C73E EB03               jmp    n3
C740 BA0CD0     n2:     mov    dx,#msgpd
C743 E8B4FD     n3:     call   sndmsg
C746 EBA1               jmp    tprtn
                ;
                ; Turn off tape drive logic power
C748 B93629     tlpoff: mov    cx,#10550       ;200 ms time out count
C74B C606020480lolp:    mov    [piopb],#80h    ;logic power off (pb3 - pin 27)
C750 F606010420         test   [piopa],#20h    ;sense logic power (pa5 - pin 22)
C755 7507               jnz    n4              ;power is now off
C757 E2F2               loop   lolp            ;wait till power off, 18.76 us/loop
C759 BA09D1             mov    dx,#msge4       ;error 4 message (logic power left on)
C75C EBE5               jmp    n3
C75E BA25D0     n4:     mov    dx,#msgld
C761 EBE0               jmp    n3
                ;
                ; Reset SCSI bus
C763 B080       rstsc:  mov    al,#80h         ;assert RST
C765 BA6161             mov    dx,#scsi1       ; to initiator command register
C768 EE                 out    dx,al
C769 B008               mov    al,#8
C76B FEC8       dlp1:   decb   al
C76D 75FC               jnz    dlp1            ;40 us delay
C76F EE                 out    dx,al           ;deassert RST
C770 C3                 ret
                ;
                ; Select tape drive
```

```
C771 E88400    tdsel:  call   bfphs          ;get bus-free phase
C774 7518              jnz    dserr          ;no bus-free phase
C776 B041              mov    al,#41h        ;initiator(6) and target(0) ID's
C778 BA6060            mov    dx,#scsi0      ; to output data register
C77B EE                out    dx,al
C77C B005              mov    al,#5          ;assert SEL & DATA BUS
C77E BA6161            mov    dx,#scsi1      ; to initiator command register
C781 EE                out    dx,al
C782 31C9              xor    cx,cx          ;timeout count
C784 BA6464    tolp1:  mov    dx,#scsi4      ;current scsi bus status
C787 EC                in     al,dx
C788 A840              test   al,#40h        ;BSY set?
C78A 7507              jnz    sel            ; yes - selected
C78C E2F6              loop   tolp1          ;550 ms timeout loop
C78E BA20D1    dserr:  mov    dx,#msge5      ;error 5 message (drive select error)
C791 EB85              jmp    em2            ;turn off moter, set flag and return
C793 BA59D0    sel:    mov    dx,#msgds
C796 E861FD            call   sndmsg
C799 30C0              xor    al,al          ;deassert SEL & DATA BUS
C79B BA6161            mov    dx,#scsi1      ; in initiator command register
C79E EE                out    dx,al
C79F C3                ret
                ;
                ; Test if tape drive ready
C7A0 BE28CA    turdy:  mov    si,#urcdb      ;test unit ready CDB
C7A3 E82900            call   sndcdb         ;send command to tape drive
C7A6 7506              jnz    turtn          ;phase mismatch
C7A8 BB0100            mov    bx,#1          ;timeout count
C7AB E8D600            call   status         ;get status
C7AE C3        turtn:  ret
                ;
                ; Skip to end of data
C7AF BE2ECA    enddat: mov    si,#edcdb      ;space to end of data CDB
C7B2 E81A00            call   sndcdb         ;send command to tape drive
C7B5 750F              jnz    ederr          ;phase mismatch
C7B7 BB9501            mov    bx,#405        ;timeout count (5 min)
C7BA E8C700            call   status         ;get status
C7BD 7507              jnz    ederr          ;failed to find end of data
C7BF BA6BD0            mov    dx,#msged      ;end of data message
C7C2 E835FD            call   sndmsg
C7C5 C3                ret
C7C6 BA36D1    ederr:  mov    dx,#msge6      ;error 6 message (failed to find eod)
C7C9 E8BAFE            call   dly2s          ;wait 2 sec
C7CC E949FF            jmpf   em2            ;turn off moter, set flag and return
                ;
                ; Send a command descriptor block to tape drive
C7CF E88B00    sndcdb: call   cmdphs         ;command phase?
C7D2 746B              jz     pmm            ; no, phase mismatch
C7D4 B90600            mov    cx,#6          ;byte count
                ;
                ; Send command or data [address in si; byte count in cx]
C7D7 E8E400    sndcd:  call   reqas          ;wait for REQ
C7DA BA6565            mov    dx,#scsi5      ;bus and status register
C7DD EC                in     al,dx
C7DE A808              test   al,#8          ;phase matched?
C7E0 745D              jz     pmm            ; no, phase mismatched
C7E2 FC                cld
C7E3 AC                lodsb                 ;command/data byte
C7E4 BA6060            mov    dx,#scsi0      ; to output data register
C7E7 EE                out    dx,al
C7E8 B019              mov    al,#19h        ;assert ACK, BSY & DATA BUS
C7EA BA6161            mov    dx,#scsi1      ; in initiator command register
C7ED EE                out    dx,al
C7EE E8D600            call   reqda          ;wait for no REQ
C7F1 BA6161            mov    dx,#scsi1
C7F4 EE                out    dx,al          ;deassert ACK & DATA BUS
C7F5 E2E0              loop   sndcd
C7F7 C3                ret
                ;
                ; Get bus-free phase
C7F8 31C9      bfphs:  xor    cx,cx          ;timeout count
C7FA BA6464    tolp2:  mov    dx,#scsi4      ;current SCSI bus status
```

```
C7FD EC                 in      al,dx
C7FE 24FE               and     al,#0feh        ;bus-free?
C800 7505               jnz     bf1             ; no
C802 BA6363             mov     dx,#scsi3
C805 EE                 out     dx,al           ;clear mode register
C806 C3                 ret
C807 A820       bf1:    test    al,#20h         ;REQ active?
C809 750E               jnz     bf2             ; yes, proceed
C80B E2ED               loop    tolp2           ;0.87 s timeout loop
C80D E82CFD             call    sndhx
C810 E815FD             call    sndsp
C813 BA54D1             mov     dx,#msge7       ;error 7 message (no bus-free phase)
C816 E9CCFE             jmpf    eml             ;set flag and return
C819 D0E8       bf2:    shrb    al,1
C81B D0E8               shrb    al,1            ;shift right 2
C81D 2407               and     al,#7           ;isolate phase bits
C81F BA6363             mov     dx,#scsi3
C822 EE                 out     dx,al           ; place phase bits in mode register
C823 BA6060             mov     dx,#scsi0       ; for the next step
C826 A801               test    al,#1           ;input?
C828 7405               jz      bf3             ; no, output
C82A EC                 in      al,dx           ; yes, get it (and discard it)
C82B B018               mov     al,#18h         ; and assert ACK & BSY
C82D EB03               jmp     bf4
C82F EE         bf3:    out     dx,al           ;output 0
C830 B019               mov     al,#19h         ; and assert ACK, BSY & DATA BUS
C832 BA6161     bf4:    mov     dx,#scsi1       ; in initiator command register
C835 EE                 out     dx,al
C836 E88E00             call    reqda           ;wait for no REQ
C839 BA6161             mov     dx,#scsi1
C83C EE                 out     dx,al           ;deassert ACK
C83D EBB9               jmp     bfphs           ;repeat
                ;
                ; Phase mismatch message
C83F BA71D1     pmm:    mov     dx,#msge8       ;error 8 message (phase mismatch)
C842 E8B5FC             call    sndmsg
C845 BA6363             mov     dx,#scsi3       ;target command register
C848 EC                 in      al,dx
C849 E8E0FC             call    sndnm
C84C E8D9FC             call    sndsp
C84F BA6464             mov     dx,#scsi4       ;bus status register
C852 EC                 in      al,dx
C853 E8E6FC             call    sndhx
C856 30C0       pmx:    xor al,al               ;z for phase mismatch
C858 C3                 ret
                ;
                ; Test for data-out phase
C859 30C0       dophs:  xor     al,al           ;none for data-out phase
C85B EB0A               jmp     phstst          ;test for data-out phase
                ;
                ; Test for command phase
C85D B002       cmdphs: mov     al,#2           ;C/D for command phase
C85F EB06               jmp     phstst          ;test for command phase
                ;
                ; Test for status phase
C861 B003       stphs:  mov     al,#3           ;C/D & I/O for status phase
C863 EB02               jmp     phstst          ;test for correct phase
                ;
                ; Test for message-in phase
C865 B007       miphs:  mov     al,#7           ;MSG, C/D & I/O for message-out phse
                ;
                ; Test phase
C867 BA6363     phstst: mov     dx,#scsi3
C86A EE                 out     dx,al           ;phase bits in target command register
C86B 31C9               xor     cx,cx           ;timeout count
C86D BA6464     tolp3:  mov     dx,#scsi4
C870 EC                 in      al,dx           ;current SCSI bus status
C871 24FE               and     al,#0feh        ;bus free?
C873 740E               jz      ptrtn           ; yes, return
C875 A820               test    al,#20h         ;REQ?
C877 7504               jnz     ph1             ; yes
C879 E2F2               loop    tolp3           ;0.74 s timeout loop
```

```
C87B EBD9               jmp    pmx             ;mismatched (z)
C87D BA6565     ph1:    mov    dx,#scsi5
C880 EC                 in     al,dx           ;bus and status register
C881 2408               and    al,#8           ;phase matched?
C883 C3         ptrtn:  ret                    ;nz if matched, z if mismatched
                ;
                ; Get status and message from drive
C884 E8DAFF     status: call   stphs           ;test for status phase
C887 7410               jz     spmm            ;phase mismatch
C889 E81E00             call   getbyt          ;get status
C88C 50                 push   ax              ;save status
C88D E8D5FF     msgip:  call   miphs           ;test for message-in phase
C890 7410               jz     mpmm            ;phase mismatch
C892 E81500             call   getbyt          ;get message and discard
C895 58         pops:   pop    ax              ;restore status
C896 240A               and    al,#0ah         ;error?
C898 C3                 ret
C899 4B         spmm:   decw   bx              ;timeout count down
C89A 75E8               jnz    status
C89C E8A0FF             call   pmm             ;send phase mismatch message
C89F FEC0               incb   al              ;phase mismatch, code=01
C8A1 C3                 ret
C8A2 4B         mpmm:   decw   bx              ;timeout count down
C8A3 75E8               jnz    msgip
C8A5 E897FF             call   pmm             ;send phase mismatch message
C8A8 EBEB               jmp    pops            ;restore status and return
                ;
                ; Get a byte
C8AA BA6060     getbyt: mov    dx,#scsi0
C8AD EC                 in     al,dx           ;read current SCSI data
C8AE 50                 push   ax              ;save the data
C8AF B018               mov    al,#18h         ;assert ACK & BSY
C8B1 BA6161             mov    dx,#scsi1       ; in initiator command register
C8B4 EE                 out    dx,al
C8B5 E80F00             call   reqda           ;wait for no REQ
C8B8 BA6161             mov    dx,#scsi1
C8BB EE                 out    dx,al           ;deassert ACK
C8BC 58                 pop    ax
C8BD C3                 ret
                ;
                ; Wait for REQ assertion
C8BE BA6464     reqas:  mov    dx,#scsi4
C8C1 EC                 in     al,dx           ;get current SCSI bus status
C8C2 A820               test   al,#20h         ;REQ asserted?
C8C4 74F8               jz     reqas           ;wait for REQ
C8C6 C3                 ret
                ;
                ; Wait for REQ deassertion
C8C7 BA6464     reqda:  mov    dx,#scsi4
C8CA EC                 in     al,dx           ;get current SCSI bus status
C8CB 2420               and    al,#20h         ;REQ deasserted?
C8CD 75F8               jnz    reqda           ;wait for no REQ
C8CF C3                 ret
                ;
                ; Send data out to tape drive in psudo DMA mode
C8D0 C606220003write:   mov    [trcnt],#3      ; 3 trials in case of problem
C8D5 C606006003         mov    [rtccr],#3      ;select rtc interrupt register
C8DA C6060F6006         mov    [rtcsir],#6     ;30 sec timeout (single interrupt)
C8DF            wtlp:;  call   tmpon           ;turn on tape motor power
                ;       jnz    w1              ; power on failed
C8DF E88FFE             call   tdsel           ;select tape drive
C8E2 7403               jz     w2
C8E4 E9B200     w1:     jmpf   wrerr           ; drive not selected
C8E7 BE34CA     w2:     mov    si,#wrcdb       ;write CDB
C8EA BF1000             mov    di,#cdb         ;temporary CDB buffer
C8ED E805FC             call   esbrd           ;set es to on-board
C8F0 FC                 cld
C8F1 B90600             mov    cx,#6           ;6-byte cdb
C8F4 F3                 rep
C8F5 A4                 movsb                  ;copy write CDB to cdb buffer
C8F6 B740               mov    bh,#40h         ;start with bank 0
C8F8 A00300             mov    al,[blks]       ;number of input (4K) blocks
```

```
C8FB 88C3              mov   bl,al          ;block counter
C8FD 30E4              xor   ah,ah
C8FF D1E0              shlw  ax,1
C901 D1E0              shlw  ax,1
C903 D1E0              shlw  ax,1           ;x8
C905 88261300          mov   [cdb+3],ah     ;number of output (0.5K) blocks
C909 A21400            mov   [cdb+4],al     ; in cdb
C90C BE1000            mov   si,#cdb        ;send write command
C90F E8BDFE            call  sndcdb         ; to tape drive
C912 75D0              jnz   wl             ; phase mismatch
C914 E842FF            call  dophs          ;test for data-out phase
C917 74CB              jz    wl             ; phase mismatch
C919 C606006000        mov   [rtccr],#0     ;start rtc interrupt
C91E FB                sti                  ;enable cpu interrupt
C91F 31F6              xor   si,si          ;data start offset
C921 B80040            mov   ax,#4000h      ;first segment
C924 8EC0              mov   es,ax
C926 B009              mov   al,#9          ;assert BSY & DATA BUS
C928 BA6161            mov   dx,#scsi1      ; in initiator command register
C92B EE                out   dx,al
C92C B002              mov   al,#2          ;DMA mode
C92E BA6262            mov   dx,#scsi2      ; to mode register
C931 EE                out   dx,al
C932 BA6565            mov   dx,#scsi5
C935 EE                out   dx,al          ;start DMA send
C936 88F8      wrlp1:  mov   al,bh
C938 BAFCFC            mov   dx,#ramp
C93B EE                out   dx,al          ;select bank
C93C BA6868            mov   dx,#dack       ;prepare for psudo-DMA output
C93F B90004    wrlp2:  mov   cx,#400h       ;4-byte count for a 4K block
C942 26        wrlp3:  seg   es
C943 AC                lodsb                ;get a data byte
C944 A20080            mov   [wait],al      ;wait for DRQ interrupt (wake up)
C947 90                nop
C948 EE                out   dx,al          ;output data to tape
C949 26                seg   es
C94A AC                lodsb                ;second
C94B A20080            mov   [wait],al      ; byte
C94E 90                nop                  ;  output
C94F EE                out   dx,al
C950 26                seg   es
C951 AC                lodsb                ;third
C952 A20080            mov   [wait],al      ; byte
C955 90                nop                  ;  output
C956 EE                out   dx,al
C957 26                seg   es
C958 AC                lodsb                ;fourth
C959 A20080            mov   [wait],al      ; byte
C95C 90                nop                  ;  output
C95D EE                out   dx,al
C95E E2E2              loop  wrlp3          ;39.25 us/4 bytes = 100 KB/s max.
C960 FECB              decb  bl             ;block count count down
C962 7419              jz    wrcmpl         ;tape write complete
C964 81FE0000          cmp   si,#0          ;end of segment?
C968 75D5              jne   wrlp2          ; no, continue with next block
C96A 8CC0              mov   ax,es          ; yes, switch segment
C96C F7D8              negw  ax
C96E 050090            add   ax,#9000h
C971 8EC0              mov   es,ax
C973 3D0050            cmp   ax,#5000h      ;second segment?
C976 74C7              je    wrlp2          ; yes, continue
C978 80C710            add   bh,#10h        ; no, advance to next bank
C97B EBB9              jmp   wrlp1
C97D 30C0      wrcmpl: xor   al,al          ;turn off
C97F BA6161            mov   dx,#scsi1      ; assert DATA BUS
C982 EE                out   dx,al          ;  in initiator command register
C983 BA6262            mov   dx,#scsi2      ; and DMA MODE
C986 EE                out   dx,al          ;  in mode register
C987 E857FB            call  rstsb          ;reset segment registers and bank
C98A FA                cli                  ;disable cpu interrupt
C98B C606006001        mov   [rtccr],#1     ;stop rtc interrupt
C990 BB1B00            mov   bx,#27         ;20 s timeout
```

```
C993 E8EEFE             call   status          ;get status
C996 7501               jnz    wrerr           ;no status
                ;       call   tmpoff          ;turn off tape motor power
C998 C3         wret:   ret
C999 E845FB     wrerr:  call   rstsb           ;reset segment registers and bank
C99C BA84D1             mov    dx,#msge9       ;error 9 message (write error)
C99F E858FB             call   sndmsg
C9A2 FE0E2200           decb   [trcnt]         ;count down retry count
C9A6 74F0               je     wret            ;retry count exceeded
C9A8 E8E6FC             call   ctinie          ;reinitialize tape drive
C9AB E931FF             jmpf   wtlp            ; and try again
                ;
                ; Write file mark
C9AE C606220003wreof:   mov    [trcnt],#3      ;at most 3 retrys
C9B3 E806FD     wflp:   call   tmpon           ;turn on drive motor power
C9B6 E8B8FD             call   tdsel           ;select tape drive
C9B9 7517               jnz    eferr           ; drive not selected
C9BB BE3ACA             mov    si,#fmcdb       ;write file mark CDB
C9BE E80EFE             call   sndcdb          ;send command to tape drive
C9C1 750F               jnz    eferr           ; phase mismatch
C9C3 BB0E00             mov    bx,#14          ;10 s timeout
C9C6 E8BBFE             call   status          ;get status and message
C9C9 7507               jnz    eferr           ;eof write error
C9CB BA85D0             mov    dx,#msgfm       ;eof message
C9CE E829FB             call   sndmsg
                ;       call   tmpoff          ;turn off drive motor power
C9D1 C3         fret:   ret
C9D2 BA9ED1     eferr:  mov    dx,#msge10      ;error 10 message (eof error)
C9D5 E822FB             call   sndmsg
C9D8 FE0E2200           decb   [trcnt]         ;count down retry count
C9DC 74F3               je     fret            ;retry count exceeded
C9DE E8B0FC             call   ctinie          ;reinitialize tape drive
C9E1 EBD0               jmp    wflp            ; and try again
                ;
                ; Resind tape
C9E3 E8D6FC     rewind: call   tmpon           ;turn on drive motor power
C9E6 E888FD             call   tdsel           ;select tape drive
C9E9 7516               jne    rwerr           ;drive not selected
C9EB BA9AD0             mov    dx,#msgrw       ;rewinding message
C9EE E809FB             call   sndmsg
C9F1 BE40CA             mov    si,#rwcdb       ;rewind CDB
C9F4 E8D8FD             call   sndcdb          ; to tape drive
C9F7 7508               jne    rwerr           ;phase mismatch
C9F9 BBF300             mov    bx,#243         ;3 min timeout
C9FC E885FE             call   status          ;get status and message
C9FF 7406               je     rwrtn
CA01 BAB7D1     rwerr:  mov    dx,#msge11      ;error 11 message (rwd error)
CA04 E8F3FA             call   sndmsg
CA07 C3         rwrtn:  ret
                ;
                ; ***** Tables and Messages *****
                ;
                ; Interrupt vector table
CA08            inttab:
CA08 C9C400F0           dfb    low smpint,high smpint,low board,high board
CA0C C8C400F0           dfb    low adcint,high adcint,low board,high board
CA10 CAC400F0           dfb    low md2int,high md2int,low board,high board
CA14 CEC400F0           dfb    low md3int,high md3int,low board,high board
CA18 A0C200F0           dfb    low rtcint,high rtcint,low board,high board
CA1C 9FC200F0           dfb    low keyint,high keyint,low board,high board
CA20 D2C400F0           dfb    low md6int,high md6int,low board,high board
CA24 D6C400F0           dfb    low md7int,high md7int,low board,high board
                ;
                ; SCSI command descriptor blocks
CA28 0000000000urcdb:   dfb    0,0,0,0,0,1     ;test unit ready, linked
CA2E 1103000000edcdb:   dfb    11h,3,0,0,0,0   ;space to end of data
CA34 0A01000008wrcdb:   dfb    0ah,1,0,0,8,0   ;write 8 blocks
CA3A 1000000001fmcdb:   dfb    10h,0,0,0,1,0   ;write 1 file mark
CA40 0100000000rwcdb:   dfb    1,0,0,0,0,0     ;rewind
                ;
                ; Gain table
CA46 0000000101gtab:    dfb    0,0,0,1,1,1,1,1,1,1,1,1,1,1,2       ;0-15
```

```
CA56 0202020202          dfb    2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2      ;16-31
CA66 0202020202          dfb    2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2      ;32-47
CA76 0202020202          dfb    2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2      ;48-63
CA86 0202020202          dfb    2,2,2,2,2,2,2,2,2,2,2,3,3,3,3,3      ;64-79
CA96 0303030303          dfb    3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3      ;80-95
CAA6 0303030303          dfb    3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3      ;96-111
CAB6 0303030303          dfb    3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3      ;112-127
CAC6 0303030303          dfb    3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3      ;128-143
CAD6 0303030303          dfb    3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3      ;144-159
CAE6 0303030303          dfb    3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3      ;160-175
CAF6 0303030303          dfb    3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3      ;176-191
CB06 0303030303          dfb    3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3      ;192-207
CB16 0303030303          dfb    3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3      ;208-223
CB26 0303030303          dfb    3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3      ;224-239
CB36 0303030303          dfb    3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3      ;240-255
CB46 0303030303          dfb    3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3      ;256-271
CB56 0303030303          dfb    3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3      ;272-287
CB66 0303030303          dfb    3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3      ;288-303
CB76 0303030303          dfb    3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3      ;304-319
CB86 0303030303          dfb    3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3      ;320-335
CB96 0303030303          dfb    3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3      ;336-351
CBA6 0303030303          dfb    3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3      ;352-367
CBB6 0303030303          dfb    3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3      ;368-383
CBC6 0303030303          dfb    3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3      ;384-399
CBD6 0303030303          dfb    3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3      ;400-415
CBE6 0303030303          dfb    3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3      ;416-431
CBF6 0303030303          dfb    3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3      ;432-447
CC06 0303030303          dfb    3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3      ;448-463
CC16 0303030303          dfb    3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3      ;464-479
CC26 0303030303          dfb    3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3      ;480-495
CC36 0303030303          dfb    3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3      ;496-511
                ;
                ; Channel selection table
CC46 99919890   ctab:    dfb    99h,91h,98h,90h

                ; Messages
CC4A 0D0A0A07   msgi:    dfb    cr,lf,lf,7
CC4E 2A2A2A2043          dfb    "*** C-44 Air Gun OBS v.01.00 ***"
CC6E 0D0A24              dfb    cr,lf,"$"
CC71 596561723Amsgc:     dfb    "Year:  $"
CC79 4D6F6E7468          dfb    "Month: $"
CC81 4461793A20          dfb    "Day:   $"
CC89 486F75723A          dfb    "Hour:  $"
CC91 4D696E3A20          dfb    "Min:   $"
CC99 5365633A20          dfb    "Sec:   $"
CCA1 0D0A       menu:    dfb    cr,lf
CCA3 68202D2048          dfb    "h - Help",cr,lf
CCAD 63202D2053          dfb    "c - Set clock",cr,lf
CCBC 73202D2053          dfb    "s - Start clock",cr,lf
CCCD 74202D2044          dfb    "t - Display time",cr,lf
CCDF 72202D2052          dfb    "r - Read in shot table",cr,lf
CCF7 64202D2044          dfb    "d - Display shot table",cr,lf
CD0F 6D202D2054          dfb    "m - Test data acq./tape drive",cr,lf
CD2E 67202D2047          dfb    "g - Go $"
CD36 0D0A       menum:   dfb    cr,lf
CD38 6820202020          dfb    "h      - Help",cr,lf
CD46 7220202020          dfb    "r      - Read SCSI registers",cr,lf
CD63 776E686820          dfb    "wnhh   - Write hh to register n",cr,lf
CD83 7320202020          dfb    "s      - Select tape drive",cr,lf
CD9E 70682E2E2E          dfb    "ph...  - Put CDB",cr,lf
CDAF 6720202020          dfb    "g      - Get data/status/message",cr,lf
CDD0 6920202020          dfb    "i      - Initialize drive",cr,lf
CDEA 6F62622020          dfb    "obb    - Output bb(hex) blocks of data",cr,lf
CE11 6620202020          dfb    "f      - Write file mark",cr,lf
CE2A 6163736262          dfb    "acsbb  - Acquire c chs @s ms, bb blks.",cr,lf
CE51 6320202020          dfb    "c      - ADC power on for preamp check",cr,lf
CE78 7120202020          dfb    "q      - Quit drive test $"
CE91 0D0A       msgr:    dfb    cr,lf
CE93 506C656173          dfb    "Please send a shot table.",cr,lf,"$"
CEAF 0D0A0A4F42msgt0:    dfb    cr,lf,lf,"OBS number:    $"
CEC2 0D0A0A5265msgt1:    dfb    cr,lf,lf,"Release time: $"
CED4 0D0A0A5761msgt2:    dfb    cr,lf,lf,"Wake-up time: $"
```

```
CEE6 0D0A537461          dfb    cr,lf,"Start time:   $"
CEF7 2020202050          dfb    "    Parameters (ch,si,bl): $"
CF13 0D0A53746F          dfb    cr,lf,"Stop time:    $"
CF24 0720204144msgpa:    dfb    7,"  ADC power turned on"
CF3A 20202D2048                 dfb            "  - Hit any key when done.$"
CF55 072020476Fmsggn:    dfb    7,"  Good night.",cr,lf,"$"
CF66 0720204869msgwu:    dfb    7,"  Hi! I am awake.$"
CF79 0720205374msgaq:    dfb    7,"  Start data acquisition",cr,lf,"$"
CF95 0720204461msgac:    dfb    7,"  Data acquisition complete$"
CFB2 0720204F42msgrl:    dfb    7,"  OBS released",cr,lf,"$"
CFC4 07204D6973msgmi:    dfb    7," Misdirected interrupt$"
CFDC 0D0A074472msgpu:    dfb    cr,lf,7,"Drive motor power on$"
CFF4 0D0A074472msglu:    dfb    cr,lf,7,"Drive logic power on$"
D00C 0D0A074472msgpd:    dfb    cr,lf,7,"Drive motor power off$"
D025 0D0A074472msgld:    dfb    cr,lf,7,"Drive logic power off$"
D03E 0D0A07496Emsgit:    dfb    cr,lf,7,"Initializing tape drive$"
D059 0D0A074472msgds:    dfb    cr,lf,7,"Drive selected$"
D06B 0D0A07536Bmsged:    dfb    cr,lf,7,"Skipped to end of data$"
D085 0D0A074669msgfm:    dfb    cr,lf,7,"File mark written$"
D09A 0D0A075265msgrw:    dfb    cr,lf,7,"Rewinding$"
D0A7 0D0A075461msgto:    dfb    cr,lf,7,"Tape write time out abort$"
D0C4 0D0A074E6Fmsge1:    dfb    cr,lf,7,"No tape motor power$"
D0DB 0D0A074E6Fmsge2:    dfb    cr,lf,7,"No tape logic power$"
D0F2 0D0A074D6Fmsge3:    dfb    cr,lf,7,"Motor power left on$"
D109 0D0A074C6Fmsge4:    dfb    cr,lf,7,"Logic power left on$"
D120 0D0A074472msge5:    dfb    cr,lf,7,"Drive not selected$"
D136 0D0A074661msge6:    dfb    cr,lf,7,"Failed to find end of data$"
D154 0D0A074E6Fmsge7:    dfb    cr,lf,7,"No bus-free mode obtained$"
D171 0D0A075068msge8:    dfb    cr,lf,7,"Phase mismatch $"
D184 0D0A074461msge9:    dfb    cr,lf,7,"Data write unsucessful$"
D19E 0D0A07454Fmsge10:   dfb    cr,lf,7,"EOF write unsucessful$"
D1B7 0D0A074661msge11:   dfb    cr,lf,7,"Failed to rewind$"
                ;
                ; ***** Reset Bootstrap *****
                ;
DFF0                     org    0dff0h
DFF0 EA04C000F0reset:    jmpf   init,board
0000                     end
```

Appendix E. Shot Table (Recording Schedule) Format

Format

Line 1:	OBS number, two digits
Line 2:	Release time in month, day, hour and minute, each of two digits and separated by a space
Line 3:	Wake-up time in the same format as above
Line 4:	Recording start time in the same format as above
Line 5:	Recording parameters (number of channels, sampling interval in ms*, and number of blocks) separated by commas
Line 6:	Recording stop time in the same format as above
Lines 7ff:	Repeat lines 3-6 for additional shot groups
Last line:	$

*The actual sampling interval is slightly different from exact milliseconds in most cases. Appendix F gives the formula to calculate the true sampling intervals with some examples.

Example

```
05              — OBS No. 5
07 01 23 45     — releases on July 1 at 23h:45m
06 30 11 56     — wakes up on June 30 at 11h:56m
06 30 12 00     — recording starts on June 30 at 12h:00m
3,8,64          — 3 channels, 8 ms sampling interval, 64 blocks
06 30 14 00     — recording ends on June 30 at 14h:00m
06 30 23 54     — wakes up again on June 30 at 23h:54m
07 01 00 00     — recording starts on July 1 at 00h:00m
4,1,128         — 4 channels, 1 ms sampling interval, 128 blocks
07 01 12 00     — recording ends on July 1 at 12h:00m
$               — end of shot table
```

Appendix F. Sampling Intervals

Sampling intervals are derived by counting down the clock rate of the baud-rate generator, and thus are slightly different from exact milliseconds in most cases. The actual sampling interval, τ_a in ms, can be calculated from a nominal sampling interval, τ_n in ms, by

$$\tau_a = \text{Int}(153.6\ \tau_n + 0.5)/153.6$$

Their values for τ_n = 1 to 25 ms are listed below:

τ_n, ms	τ_a, ms	Error
1	1.0026	0.260%
2	1.9987	-0.065%
3	3.0013	0.043%
4	3.9974	-0.065%
5	5.0000	0.000%
6	6.0026	0.043%
7	6.9987	-0.019%
8	8.0013	0.016%
9	8.9974	-0.029%
10	10.0000	0.000%
11	11.0026	0.024%
12	11.9987	-0.011%
13	13.0013	0.010%
14	13.9974	-0.019%
15	15.0000	0.000%
16	16.0026	0.016%
17	16.9987	-0.008%
18	18.0013	0.007%
19	18.9974	-0.014%
20	20.0000	0.000%
21	21.0026	0.012%
22	21.9987	-0.006%
23	23.0013	0.006%
24	23.9974	-0.011%
25	25.0000	0.000%

Appendix G. Header and Data Formats

Header Format

Header occupies the first 16 bytes of each 4K block.

Byte	Data
0	Record number, low byte
1	Record number, high byte
2	Block number
3	Number of blocks in this record
4	Number of channels
5	Sampling interval in ms
6	Not used
7	Not used
8	OBS number
9	Year
10	Month
11	Day
12	Hour
13	Minute
14	Second
15	1/10 second

All header entries are in binary.

Data Format

Data occupies the remaining 4080 bytes of each block.
Two bytes (16 bits) per data sample, multiplexed (i.e., channel 1, ch. 2, ..., ch. 1, ch. 2, ...).

Byte	Bits	Data
1st	6 msb	6 lsb of ADC output
1st	2 lsb	Exponent of power 5
2nd	all	8 msb of ADC output

ADC output is in offset binary with 2000h for zero input. To obtain amplitude referred to the highest gain, subtract 2000h from ADC and multiply by 5 raised to the power of the exponent.

Temporary Irregularity

For an unknown reason, the recorded data are shifted forward one bye, with the header for the first block starting at the second byte of the block and the last data byte of the block appearing at the first byte of the second block, and so on, with the very last byte of the last block missing.