FINAL REPORT ON THE DIGITAL AUDIO CAPTURE AND IDENTIFICATION SYSTEM (DACIS)

FOR THE REMOTE ENVIRONMENTAL SENSING PROGRAM

AUGUST 1998

Protagoras N. Cutchis
Space Department
Johns Hopkins University Applied Physics laboratory

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	August 1998	Final Report

4. TITLE AND SUBTITLE
Final Report on the Digital Audio Capture and Identification System (DACIS). SERDP: Advanced Biotelemetry for Resource Management on Military Lands (CS-759) , Part II

5. FUNDING NUMBERS
N/A

6. AUTHOR(S)
Protagoras N. Cutchis

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Johns Hopkins University Applied Physics Laboratory

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
SERDP
901 North Stuart St. Suite 303
Arlington, VA 22203

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
N/A

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
Approved for public release: distribution is unlimited.	A

13. ABSTRACT *(Maximum 200 Words)*
The Digital Audio Capture and Recognition System (DACIS) is designed to be an animal mounted (eventually bird borne) system which records and analyzes animal noises automatically. While the DACIS is fairly complex, it is actually an advanced acoustic sensor from a system point of view. The goal of the prototype system is to be able to identify with a high degree of reliability 4 animal calls from a single or multiple species. A very limited number of tests were performed on the DACIS with four types of wolf calls that were named growls, howls, whines and barks. A threshold value of 17 was found to be optimal yielding no false positives and 25% false negatives I early testing. Later tests showed up to 50% false negatives but still no false positives with a threshold of 20 or lower.

14. SUBJECT TERMS	15. NUMBER OF PAGES
SERDP, DACIS, biotelemetry, biodiversity, natural resource management, wildlife	140
	16. PRICE CODE N/A

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
unclass	unclass	unclass	UL

JOHNS HOPKINS
UNIVERSITY

Applied Physics Laboratory
11100 Johns Hopkins Road
Laurel MD 20723-6099
240-228-5000 / Washington
443-778-5000 / Baltimore

Please refer to:
CMI-98-040

31 August 1998

Institute for Advanced Science and Technology in Medicine

To: WS Seegar

From: PN Cutchis, M. D.

Subject: Final Report on the DACIS (Digital Audio Capture and Identification System)

Enclosures:
(1) Schematics of DACIS Processor Board
(2) Assembly Language Software for DACIS unit
(3) BASIC Software for PC Communication/Test Software
(4) Raw and analyzed data from DACIS for wolf calls
(5) Article from September 1997 NASA conference on DACIS

INTRODUCTION AND GENERAL DESCRIPTION OF THE DIGITAL AUDIO CAPTURE AND RECOGNITION SYSTEM (DACIS)

The DACIS system is designed to be an animal mounted (eventually bird borne) system which records and analyzes animal noises automatically. While the DACIS is fairly complex, it is actually an advanced acoustic sensor from a system point of view. The goal of the prototype system described here is to be able to identify with a high degree of reliability 4 animal calls from a single or multiple species. The system is based on an 8-bit MC68HC811 microprocessor that contains EEPROM, an 8-bit A/D converter and a serial interface, all of which are used in the DACIS design. The prototype board is 1.95" square and weighs 18.2 grams. This version incorporates a socket, which increase the size and weight of the board. The final design, which has been laid out, is 1.85" square and should weigh less than 15 grams.

The present design captures 5 seconds of audio at 6,000 samples per second to yield a theoretical high frequency capture of 3 kHz. The system then performs 10 digital filters on each of ten 0.5 second time epochs to yield a matrix of 100 time/frequency parameters. These parameters are then normalized to account for volume level differences and then compared to a stored table using a pattern-matching algorithm.

The DACIS unit, while in the listen mode, only has the microphone, 2-stage amplifier, and threshold comparator circuit operating. In this mode, the unit consumes only 900-1300 μA of current at 5 VDC. If the volume level exceeds a preset threshold (adjustable on the board), then the processor is awoken from its STOP (low power) mode and starts the digital 5 second recording. While recording and performing the analysis, the DACIS system consumes approximately 16 mA and this analysis takes 17 minutes per animal call.

SOFTWARE DESCRIPTION

The assembly language software for the DACIS includes subroutines that perform the following major tasks:

1) Interrupt trigger of processor
2) Digital 5 second recording
3) 10 frequency band digital band-pass filters
4) Subroutine for normalization of filter outputs
5) Pattern matching against four stored templates
6) Communication with Dahlgren furnished ARGOS PTT host microprocessor

The software for the DACIS system was written entirely in assembly language and the test software for the host PC was written in BASIC. The software initializes memory locations and other parameters and then executes the STOP command that halts all microprocessor functions including the oscillator for maximum power conservation. The IRQ (Interrupt ReQuest) line, which is pulled low by the comparator when the audio volume exceeds a preset threshold, awakens the processor from the stopped state. The processor then records 5 seconds of audio at 6,000 samples per second. Then, the software, in the routine DIGIFILT (for digital filter) performs 10 FIR (Finite Impulse Response) filters at frequencies from 250 Hz to 2500 Hz in 250 Hz increments. The filters are 23 stage. This number was determined from simulations run on a PC to obtain suitable fall off without too much filter overlap. The 10 filters are run on each of 10 time epochs which are approximately 0.5 seconds in length (the exact length is 0.512 seconds to ease software design).

After completing the digital filters, the 100 parameters obtained are normalized to compensate for differences in volume level during recording. The next routine is the pattern match routine which performs the least squares fit to the 100 template parameters for each of the four stored templates. The formula used is:

$$x = \sqrt{\frac{\sum \left(a_{ij} - c_{ij}\right)^2}{100}}$$

where a_{ij} and c_{ij} are the filter outputs and template parameters respectively. The last routine that was going to be implemented was simple threshold detection for positive identification of the calls. It was decided to not implement that routine (although it is coded into the DACIS processor) and instead to transfer all four correlation values from the above equation for each call to the Dahlgren unit.

There can be timing problems with missed bytes if the PC utilized is a very new fast processor. Adjusting timing loops in the PC code can compensate for this.

HARDWARE DESCRIPTION

The design of the hardware is complete. The MC68HC11 microprocessor contains internal RAM (Random Access memory), EEPROM (Electrically Erasable Read Only Memory), a UART (Universal Asynchronous Receiver Transmitter) and A/D (Analog to Digital)converter subsystems on-chip which are used in this design.

A two-stage amplifier built from an LM358AM dual operational amplifier amplifies the microphone's signal. Then, the signal is fed into an LM339 voltage comparator. When the sound level exceeds a threshold, which is set by an on board trimmer potentiometer, the IRQ line of the processor is brought low and the processor awakens from its low power "sleep" state and starts recording. The memory is a 128K byte by 8 static RAM of which all 128K can be addressed in the present hardware. However, the present software only requires about 32K.

Two versions of the DACIS processor were laid out and 2 prototypes of one version were fabricated. One version has no socket for the microprocessor and the second version, which is slightly larger, has a socket. The sizes are 1.85" square and 1.95" square respectively. The version with the socket version has been fabricated since this version allows easy removal of the microprocessor for software changes.

POWER SUPPLY REQUIREMENTS

The DACIS board requires regulated 5 VDC +/-500mV as there is no on board voltage regulation. The current consumption varies somewhat from unit to unit but is between 900 μA and 1250 μA during the listen period and 15 mA and 34 mA during the record and signal analysis periods. The DACIS system, other than audio "awakening" has no internal method to completely shut itself off. It shuts down all of its own internal timers while in the "stop" mode to save power and therefore has no reference of how much time has passed. Therefore, in the Dahlgren application, the Dahlgren unit, which does have a real-time clock, will be responsible for time keeping. The maximum number of calls to be analyzed over any set time period can be programmed by having the Dahlgren processor shut down power to the entire DACIS unit until further analyses are desired.

DAHLGREN INTERFACE

The details of the Dahlgren interface were not confirmed until August of 1998. The interface functions as follows. There are 2 serial lines (one in each direction), 0-5 volts at 9,600 baud, 8 bits, 1 stop bit and no parity between the two processors. The DACIS will keep the transmit line at high impedance until it is given permission to transmit data (see below). There are 2 additional handshaking lines, again one in each direction between the two processors. After analyzing a call, the DACIS will raise its line high to signal the Dahlgren processor that it has data available for transfer. When it is ready, the Dahlgren processor will raise its line to the DACIS high signaling that it is prepared for data transfer. The DACIS will immediately enable the transmitter and await an ASCII "S" from the Dahlgren unit over the serial link. The DACIS will wait essentially forever in an infinite loop until it sees this "S" or an "R" for resend. The DACIS will then transfer 4 bytes that correspond to the four template matches. After transferring the data once, the DACIS will wait for about 0.5 seconds to see if an "R" or "S" is sent to request that the data be sent again. If the DACIS does not receive an "R" or "S" within the 0.5 seconds it will reset and wait to analyze the next call. If it does it will wait again for another 0.5 seconds. The unit can be kept in this mode, resending the same data forever if requests are continually sent within 0.5 second windows.

The last portion of the Dahlgren interface is that the Dahlgren processor will have control over the DACIS power lines so that if too many analyses are being conducted, the Dahlgren unit can shut down the DACIS board temporarily to save system battery power.

The wiring connections on the ribbon cable to the DACIS board are as follows:

BLUE:	Ground
GREEN:	Dahlgren to DACIS control line
YELLOW:	DACIS to Dahlgren control line
ORANGE:	Serial data output
RED:	Serial data input
BROWN:	+5VDC

TRIGGER SOUND LEVEL ADJUSTMENTS

The gain of the two stage audio amplifier is fixed at a level that was found to be suitable for most anticipated wildlife environments. The goal was to obtain reasonable gain without the amplifier saturating. However, the trigger threshold is adjustable by adjusting a small trimmer potentiometer on the back (opposite side from the processor) of the board. The setting of this potentiometer is quite "touchy". All of the trimmers were set at APL during testing to a level that seemed appropriate. However, without actual field data to indicate most likely volume levels for a specific species, the level set at APL may need further adjustment. If the level is set to be too sensitive, the unit will trigger too frequently on spurious sounds. Therefore, it is recommended that the unit be set slightly to the less sensitive side of what may be perceived as optimum. The setting found at APL to be best yields a voltage of 1.5 volts at pin 4 of the LM339 comparator.

TEST RESULTS

A very limited number of tests were performed on the DACIS with four types of wolf calls that were named growls, howls, whines and barks. The howls and barks frequently blend together. This is one of several factors which indicate that from an acoustic standpoint, wolves appear not have been an "easy" first target animal for the DACIS system. Wolves were chosen because of the availability of acoustic data and because they are large enough to carry a large collar with the Dahlgren Argos/GPS unit.

The data were first analyzed on a PC in Excel with digital filter outputs transferred across a serial link from the DACIS. This was done so that actual wolf calls, as recorded through the DACIS microphone and recorded by its processor could be used. The threshold value, which can now be selected at the analysis station, can be moved up or down so as to reduce false positives or false negatives or the total error rate. As the spreadsheets show in Enclosure 4, a threshold value of 17 was found to be optimal yielding no false positives and 25% false negatives in early testing. Later tests showed up to 50% false negatives but still no false positives with a threshold of 20 or lower.

An additional analog data tape received in August 1998 provided few if any wolf calls of quality worthy of conducting additional testing. Therefore, after discussions with the sponsor, it was decided not to conduct further analyses. It was agreed upon to instead load the existing template parameters into the DACIS and to complete the software for the Dahlgren interface.

Protagoras Cutchis, M.D.

PN Cutchis, M.D.

PNC:cco

JOHNS HOPKINS
UNIVERSITY
Applied Physics Laboratory
Laurel MD 20723-6099

Internal Distribution:

PN Cutchis (w/enclosures)
B Henke (w/enclosures)
JC Murphy (w/enclosures)
WS Seegar (w/enclosures)
CMI Files (w/enclosures)

Enclosure 1
Schematics of DACIS Processor Board

NOTES:

1 WHERE APPLICABLE, UNLESS OTHERWISE SPECIFIED:

ALL RESISTANCES ARE IN OHMS

ALL CAPACITANCES ARE IN FARADS

ALL VOLTAGES ARE DC

THE SYMBOL _ REPRESENTS A SPACE

( ) SHEET TO SHEET REFERENCES

2 THE FOLLOWING ARE THE LAST REF DESIGNATIONS USED:

C13, MK1, R18, U8, Y1

3 THE FOLLOWING ARE UNUSED REF DESIGNATIONS:

NONE

4 UNLESS SHOWN ON THE ACTUAL CIRCUITRY, ALL POWER CONNECTIONS FOR INTEGRATED CIRCUITS WILL BE REPRESENTED ON THE "IC POWER/GROUND CHART"

D

C

B

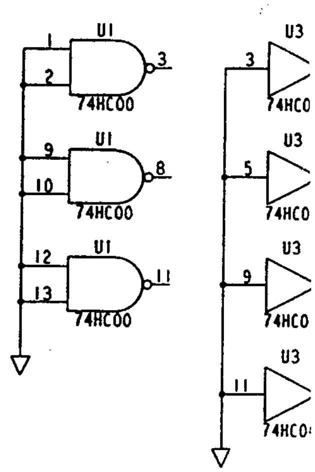
A

IC POWER/GROUND CHART

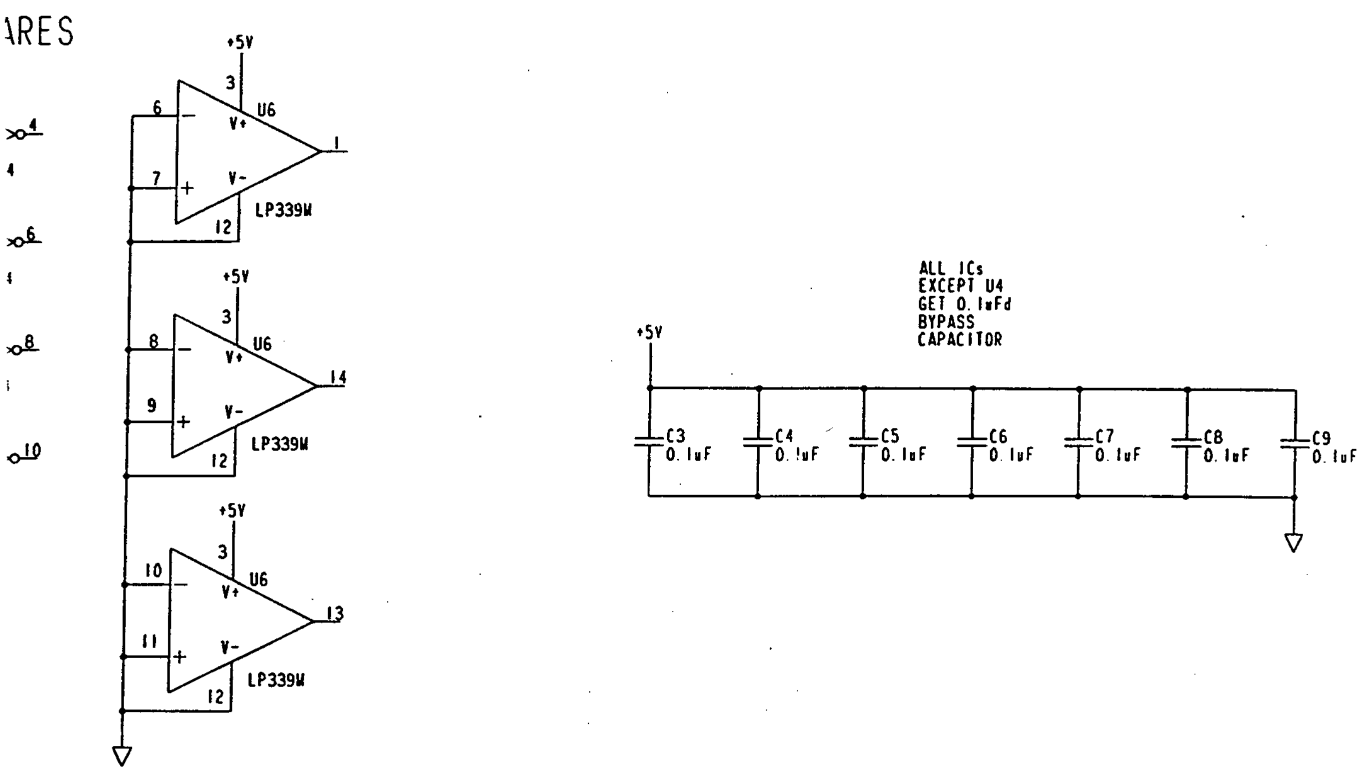
+5V

+5V pin		REF NO	PART NO	GENERIC NO		GND pin
14	+V	U1	MM74HC00M	74HC00	6	7
14	+V	U3	MM74HC04M	74HC04	6	7
20	+V	U8	MM74HC373WM	74HC373	6	10
		U5	LM358AM	LM358A		
		U6	LP339M	LP339		
		U7	MC68HC811E2CFN	MC68HC811E2		
		U4	S-8054HN-CB-X	S-805		
32	+V	U2	SRM20100LMT70	SRM20100LMT70	6	16

SP



5301-1851



BASIC

PARTS LIST

PH CUTCHIS
J.A. DAVIS
PH CUTCHIS
PH CUTCHIS

THE JOHNS HOPKINS UNIVERSITY
APPLIED PHYSICS LABORATORY
JOHNS HOPKINS ROAD LAUREL, MD 20723-6099

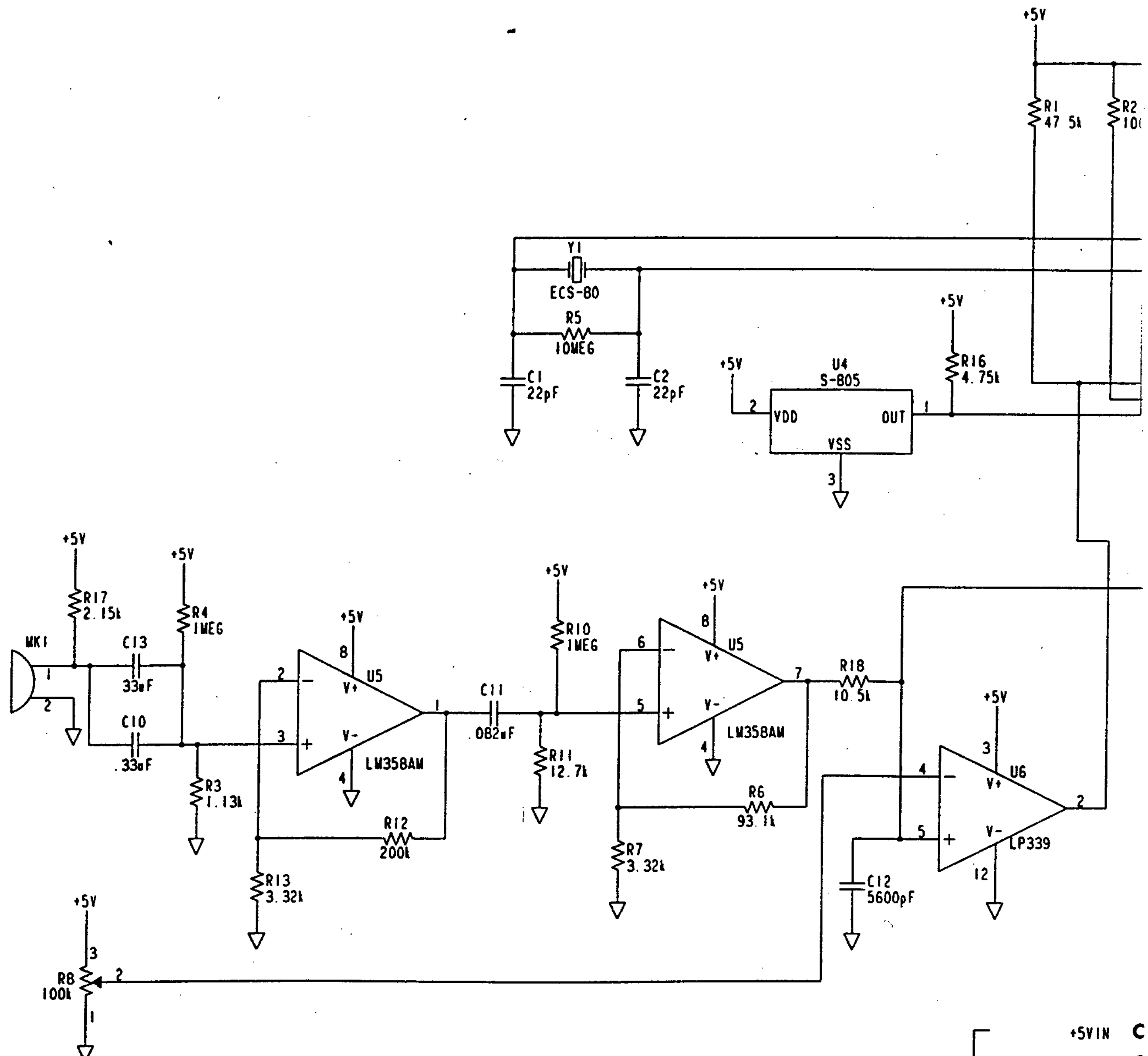
SCHEMATIC
TRIGGERED BIRD #2 AUDIO CAPTURE SYSTEM WITH 128K MEMORY

CAGE CODE	SIZE	DWG NO.	REV
88898	D	5301-1851	0

SCALE NONE — SHEET 1 OF 2

MENTOR GRAPHICS V8.2

5301-1850



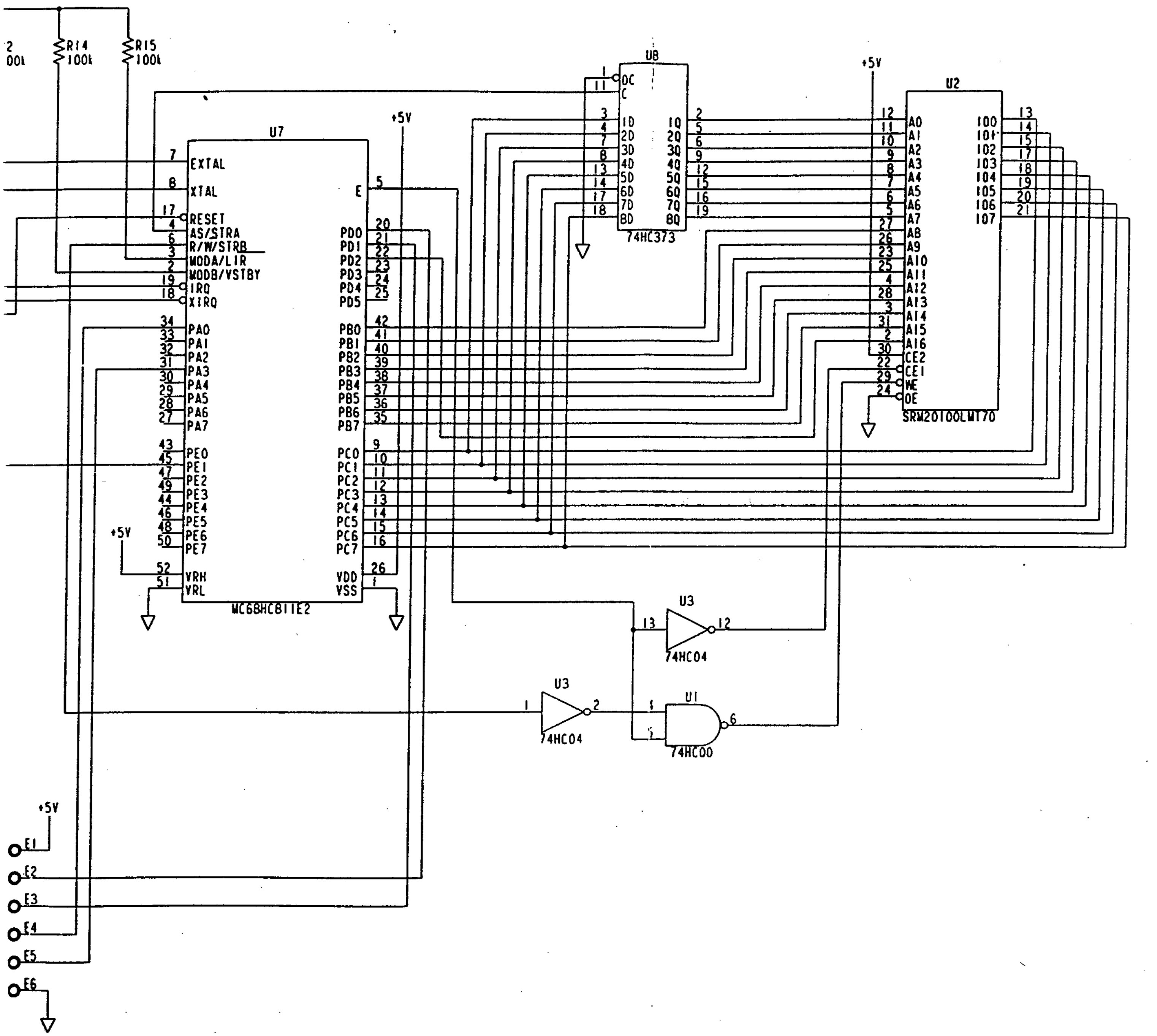
6 PLATED THROUGH HOLES OR OTHER MEANS TO ATTACH 6 CONDUCTOR RIBBON CABLE NEAR BOARD EDGE

+5VIN
SDI
SDO
HANDSHKIN
HANDSHKOUT
GND

NOT ARCHIVED

7 6 5

REV	BY & DATE	DESCRIPTION	CHECK & DATE	APPROVED & DATE



CAGE CODE	SIZE	DWG NO.	REV
88898	D	5301-1851	b
SCALE NONE		DO NOT SCALE PRINT	SHEET 2

CAD SYSTEM MENTOR GRAPHICS V8.2

CAD FILE 5301-1850

4 | 3

C

B

A

Enclosure 2
Assembly Language Software
For DACIS Processor

```
      ; WOLF-MOUNTED DIGITAL AUDIO CAPTURE AND IDENTIFICATION
      ; SYSTEM (DACIS) FOR THE REST PROGRAM
      ; THIS PROGRAM IS DESIGNED FOR THE DAHLGREN INTERFACE
      ; **** ALL 10 DIGITAL FREQUENCY FILTERS WITH 10 TIME EPOCHS ***
      ;           WITH AUDIO CAPTURED THROUGH THE MICROPHONE
      ;
      ; VERSION: C:\BIRD97\WOLF987.ASM
      ;
      ; THIS VERSION UTILIZES A 128 KBYTE MEMORY (SRM20100LMT)
      ; & IMPLEMENTS THE LP339 COMPARATOR AUDIO THRESHOLD TRIGGER
      ; CREATED:       THURSDAY 8/27/98  2:25 PM
      ; LAST REVISION: THURSDAY 8/27/98  2:50 PM
      ; 1) CHANGED FROM IRQ TO XIRQ TO ALLOW STOP MODE USE 4/8/96
      ; 2) CHANGED TO OPERATE ON 4.9152 MHz XTAL (BAUD=00100001)
      ; 3) COMPLETED ASCII SUBROUTINE 7/18/96
      ; 4) REMAPPED INTERNAL ROM & RAM TO 8K AND 9K BOUNDARIES
      ; 5) CHANGED MEAN TO 50 TO ALLOW 5:1 DYNAMIC RANGE IN FILTER BANDS
      ; 6) REMOVED DATA TRANSFER CODE FROM DIGIFILT FOR FILTER OUTPUT 8/19/98
      ; 7) REMOVED DATA TRANSFER CODE FROM NORMALIZE 8/19/98
      ; 8) ADDED HSIN AND HSOUT  FOR DATA TRANSFER 8/19/98 (NO PA0 INTERRUPT)
      ; 9) DUMP ALL 8 BYTES OF MATCH DATA FOR TEST ONLY: ADDED 8/25/98
      ;     LAST REVIEWED: THURSDAY 8/27/98
                            DEFSEG  DIG,    START=0F800H
                            SEG     DIG
      ;----------------------------------------------------------------------
      ;ROM ADDRESSES (REGISTERS ETC.)
=103D     INIT        EQU    $103D
=8000     PORTAADR    EQU    $8000
=8003     PORTCADR    EQU    PORTAADR+3
=8004     PORTBADR    EQU    PORTCADR+1
=8007     PORTCDDR    EQU    PORTBADR+3
=8008     PORTDADR    EQU    PORTCDDR+1
=8009     PORTDDDR    EQU    PORTDADR+1
=800E     TCNT        EQU    PORTDDDR+5
=8016     TOC1        EQU    TCNT+8
=8018     TOC2        EQU    TOC1+2
=8021     TCTL2       EQU    TOC2+9
=8022     TMSK1       EQU    TCTL2+1
=8023     TFLG1       EQU    TMSK1+1
```

```
=8024     TMSK2       EQU     TFLG1+1
=8026     PACTL       EQU     TMSK2+2
=802B     BAUD        EQU     PACTL+5
=802C     SCCR1       EQU     BAUD+1
=802D     SCCR2       EQU     SCCR1+1
=802E     SCSR        EQU     SCCR2+1
=802F     SCDR        EQU     SCSR+1
=8030     ADCTL       EQU     SCDR+1
=8031     ADR1        EQU     ADCTL+1
=8032     ADR2        EQU     ADR1+1
=8033     ADR3        EQU     ADR2+1
=8034     ADR4        EQU     ADR3+1
=8039     OPTION      EQU     ADR4+5
        ;-----------------------------------------------------------------
        ;RAM ADDRESSES
        ;VARIABLE         REL. ADD.    ABS. ADD.     FUNCTION
=9000     COUNT       EQU   $9000          ;0
=9001     TEMP1       EQU   COUNT+1        ;1
=9002     TEMP2       EQU   TEMP1+1        ;2
=9003     TEMP3       EQU   TEMP2+1        ;3
=9004     TIMINT      EQU   TEMP3+1        ;4&5       SAMPLE TIME INT IN uS
=9006     NUMINT      EQU   TIMINT+2       ;6         NUMBER OF PLAYBACK REPEATS
=9007     MEMPAGE     EQU   NUMINT+1       ;7         MEMPAGE=0 OR 1
=9008     ENDDATA     EQU   MEMPAGE+1      ;8 & 9     ADDRESS FOR END OF DATA
=900A     EPOCH       EQU   ENDDATA+2      ;10        ADDRESS FOR EPOCH # (1-10)
=900B     INSTRUCT    EQU   EPOCH+1        ;11        SINGLE BYTE INSTRUCTION
=900C     FILPAR      EQU   INSTRUCT+1     ;12&13     PARAMETER INDEX
=900E     AVERAGE     EQU   FILPAR+2       ;14&15     AVERAGE VALUE OF FILTER OUT
=9010     FILTAMP     EQU   AVERAGE+2      ;16-115    100 FILTER OUTPUTS
=9074     NUMCALL1    EQU   FILTAMP+100    ;116       NUMBER OF TYPE 1 CALLS
=9075     NUMCALL2    EQU   NUMCALL1+1     ;117       NUMBER OF TYPE 2 CALLS
=9076     NUMCALL3    EQU   NUMCALL2+1     ;118       NUMBER OF TYPE 3 CALLS
=9077     NUMCALL4    EQU   NUMCALL3+1     ;119       NUMBER OF TYPE 4 CALLS
=9078     FREQ        EQU   NUMCALL4+1     ;120       FILTER BEING USED (0-9)
=9079     FILTSTART   EQU   FREQ+1         ;121-122   START ADDRESS OF FILTER
=907B     SIGN        EQU   FILTSTART+2    ;123       BIT7=SIGN OF MULT RESULT
=907C     MEGAADDL    EQU   SIGN+1         ;124-125   MEGA ADDITTION REGIS-LOW
=907E     MEGAADDH    EQU   MEGAADDL+2     ;126-127   MEGA ADDITION REGIS-HIGH
=9080     PCLOADFLG   EQU   MEGAADDH+2     ;128       PC LOADING FLAG
```

```
=9081                 79 TESTADDRS  EQU   PCLOADFLG+1  ;129&130    TEST ADDRESS FOR PC DATA LOAD
=9083                 80 MACROADDL  EQU   TESTADDRS+2  ;131&132
=9085                 81 MACROADDH  EQU   MACROADDL+2  ;133&134
=9087                 82 STARTDATA  EQU   MACROADDH+2  ;135&136    START OF AUDIO DATA SEGMENT
=9089                 83 MATCHQUAL1 EQU   STARTDATA+2  ;137&138    QUALITY OF MATCH 1
=908B                 84 MATCHQUAL2 EQU   MATCHQUAL1+2 ;139&140    QUALITY OF MATCH 2
=908D                 85 MATCHQUAL3 EQU   MATCHQUAL2+2 ;141&142    QUALITY OF MATCH 3
=908F                 86 MATCHQUAL4 EQU   MATCHQUAL3+2 ;143&144    QUALITY OF MATCH 4
=9091                 87 STOREADDR  EQU   MATCHQUAL4+2 ;147&148    FILTER OUTPUT
=9093                 88 INITIAL    EQU   STOREADDR+2  ;149        0=INITIAL RECORD
=9094                 89 PATTERN    EQU   INITIAL+1    ;150-151    PATTERN NUMBER (0-3)
=9096                 90 CALLSDONE  EQU   PATTERN+2    ;152-153    NUMBER OF PATTERNS ANALYZED
=9098                 91 DAHLFLAG   EQU   CALLSDONE+2  ;154        FLAG FOR DAHLGREN DATA SENT
=F448                 92 MATCHDATA  EQU    -3000       ;           RAM MATCH DATA
                   93 ;-------------------------------------------------------------------
                   94 ; RESET AND INTERRUPT VECTORS
=FFFE                 95                 ORG     0FFFEH        ;RESET VECTOR
07FE& 0000&           96                 FDB     DIGREC
=FFF2                 97                 ORG     0FFF2H        ;IRQ VECTOR
07F2& 01D4&           98                 FDB     RECORD5
=FFD6                 99                 ORG     0FFD6H        ;RECEIVE DATA INTERUPT
07D6& 0079&          100                 FDB     DATAREC
=FFF6                101                 ORG     0FFF6H        ;SWI VECTOR
07F6& 01D4&          102                 FDB     RECORD5
=FFF8                103                 ORG     0FFF8H        ;ILLEGAL OPCODE TRAP
07F8& 0000&          104                 FDB     DIGREC
                     105 ;-------------------------------------------------------------------
=F800                106                 ORG     0F800H
=0000&               107 DIGREC          EQU     $
0000& 0F             108                 SEI                  ;DISABLE INTERRUPTS
0001& 86 98          109                 LDAA    #98H         ;REMAP RAM AND REGISTERS
0003& B7 103D        110                 STAA    INIT         ;RAM STRT=$9000 REGS=$8000
0006& 86 00          111                 LDAA    #0           ;SET TIMER DIVIDER TO E/1
0008& B7 8024        112                 STAA    TMSK2        ;MUST BE DONE IN 1ST 64CLKS
000B& 8E 90FF        113                 LDS     #90FFH       ;LOAD STACK POINTER
000E& 86 88          114                 LDAA    #10001000B   ;CONFIGURE PORTA BITS 3 & 7
0010& B7 8026        115                 STAA    PACTL        ;AS OUTPUT TO D/A CONVERTER
0013& 86 04          116                 LDAA    #00000100B   ;CONFIGURE BITS D0&D1 AS INPUT
0015& B7 8009        117                 STAA    PORTDDDR     ;WILL BOTH BE HIGH IMPED FOR DAHLG
```

```
0018& B6 8039                        LDAA   OPTION       ;USE EXTERN XTAL CLK & POWER-UP
001B& 8A 80                          ORAA   #10000000B   ;A/D CONVERTER (BIT 7)
001D& B7 8039                        STAA   OPTION
0020& 86 00                          LDAA   #00000000B   ;SELECT 8 DATA BITS
0022& B7 802C                        STAA   SCCR1
0025& B6 8008                        LDAA   PORTDADR     ;SET TRANSMIT LINE HIGH=OFF
0028& 84 FB                          ANDA   #11111011B   ;THIS OUTPUT IS INVERTED
002A& B7 8008                        STAA   PORTDADR
002D& 86 21                          LDAA   #00100001B   ;SET BAUD RATE TO 9600
002F& B7 802B                        STAA   BAUD         ;FOR 4.9152 MHz CRYSTAL
0032& 86 FF                          LDAA   #255         ;START A/D WARM-UP DELAY
0034& C6 FF            CNT2          LDAB   #255
0036& 5A               CNT1          DECB
0037& 26 FD                          BNE    CNT1
0039& 4A                             DECA
003A& 26 F8                          BNE    CNT2         ;END A/D POWER-UP DELAY
003C& 86 05                          LDAA   #5
003E& B7 9006                        STAA   NUMINT
0041& CC 0000          NOTRIG        LDD    #0
0044& B7 8000                        STAA   PORTAADR
0047& FD 9096                        STD    CALLSDONE
004A& FD 9074                        STD    NUMCALL1     ;ZERO OUT ALL CALLS IDENTIFIED
004D& FD 9076                        STD    NUMCALL3
0050& FD 900C                        STD    FILPAR
0053& B7 9080                        STAA   PCLOADFLG
0056& B7 900A                        STAA   EPOCH        ;START WITH EPOCH=0
0059& B7 9078                        STAA   FREQ         ;ZERO OUT FILTER NUMBER
005C& B7 9093                        STAA   INITIAL
005F& 86 01                          LDAA   #00000001B   ;ENABLE PA0/IC3 INTERRUPT
0061& B7 8022                        STAA   TMSK1
0064& B7 8021                        STAA   TCTL2        ;CAPTURE RISING EDGES ONLY
0067& BD 046E&                       JSR    FULLSEC
006A& BD 046E&                       JSR    FULLSEC
                       ;--------------------------------------------------------------------
006D& 8E 90FF          NEWSTART      LDS    #90FFH       ;RELOAD STACK
0070& 07                             TPA
0071& 84 7F                          ANDA   #01111111B   ;CLEAR "S" BIT TO ENABLE
0073& 06                             TAP                 ;STOP MODE (TESTED 8/27/98=OK)
0074& 0E                             CLI                 ;ALLOW INTERRUPTS NOW
```

```
0075& CF                          STOP
0076& 01             TRIGGER      NOP
0077& 20 F4                       BRA     NEWSTART        ;ALLOW MULTIPLE ANALYSES
                     ;-----------------------------------------------------------------------
                     ;MOST OF THE FOLLOWING COMMAND INTERPRETERS/DIRECTORS HAVE BEEN
                     ; COMMENTED OUT ON 8/20/98 AS THEY WERE FOR TEST USE ONLY
0079& 0F             DATAREC      SEI
007A& 86 FF                       LDAA    #0FFH           ;CLEAR FLAG TO PREVENT
007C& B7 8023                     STAA    TFLG1           ;CONTINUOUS INTERRUPTS
007F& B6 802E                     LDAA    SCSR            ;CLEAR RDRF FLAG
0082& B6 802F                     LDAA    SCDR
0085& B7 900B                     STAA    INSTRUCT
                     ;            LDAB    PCLOADFLG
                     ;            BEQ     INTERP
                     ;            JSR     INSINE2
                     ;            RTI
                     ;INTERP
                     ;            CMPA    #47H            ;CHECK SEND ID INSTRU="G"
                     ;            BNE     NEXTINT1
                     ;            JSR     ECHOID
                     ;            RTI
                     ;NEXTINT1    CMPA    #49H            ;"I" =INPUT SINE WAVE FROM PC
                     ;            BNE     NEXTINT2
                     ;            JSR     INSINE
                     ;            RTI
                     ;NEXTINT2    CMPA    #4FH            ;"O" = DUMP AMPLITUDE
                     ;            BNE     NEXTINT3
                     ;            JSR     DUMPAMP
                     ;            RTI
                     ;NEXTINT3    CMPA    #4EH            ;"N"=WATERFALL DUMP
                     ;            BNE     NEXTINT4
                     ;            JSR     CHCKNORM
                     ;            RTI
                     ;NEXTINT4    CMPA    #44H            ;"D"=DIGITAL FILTER
                     ;            BNE     NEXTINT5
                     ;            JMP     DIGIFILT
                     ;NEXTINT5    CMPA    #43H            ;"C"=CONFIRM DUMP
                     ;            BNE     NEXTINT6
                     ;            JSR     CONFIRM
```

```
;NEXTINT6       CMPA    #50H            ;"P"=CHECK MATCH
;               BNE     NEXTINT7
;               JSR     SENDMATCH
;MOVED THESE INTERPRETERS DIRECTLY INTO DAHLGREN ROUTINE
;               CMPA    #53H            ;"S"=SEND DATA
;               JSR     DAHLGREN
;               CMPA    #52H            ;"R"=RESEND DATA
;NEXTINT7       RTI
;-------------------------------------------------------------------
;THE FOLOWING ROUTINE ALLOWED DOWNLOADS OF SINE WAVE DATA FROM A PC
;TO TEST THE DIGITAL FILTER BANKS AND WAS COMMENTED OUT ON 8/24/98
;INSINE         LDAA    #1              ;MADE THIS A SUBROUTINE
;               STAA    PCLOADFLG       ;SO EXIT BY RTS ONLY!
;               LDD     #0
;               STD     TESTADDRS
;               RTS
;INSINE2        LDX     TESTADDRS
;               LDAA    INSTRUCT
;               STAA    0,X
;               INX
;               STX     TESTADDRS
;               CPX     #30720          ;CHECK FOR 30720 SAMPLES
;               BNE     ENDSINE
;               CLR     PCLOADFLG
;ENDSINE        RTS
;-------------------------------------------------------------------
;THE FOLLOWING SUBROUTINE WAS FOR TEST ONLY AND WAS
;COMMENTED OUT ON 8/24/98
;SENDMATCH      LDD     #MATCHQUAL1     ;SEND FIRST BYTE
;               JSR     ASCII
;               LDD     #MATCHQUAL1     ;SEND SECOND BYTE
;               JSR     ASCII
;               LDD     #0
;               STD     FILPAR
;               RTS
;-------------------------------------------------------------------
;THE FOLLOWING ROUTINE WAS FOR TEST PURPOSES ONLY AND WAS
;COMMENTED OUT ON 8/24/98
;CONFIRM        LDD     TESTADDRS
```

```
                ;               CPD     #30720
                ;               BEQ     SENDX
                ;               LDD     #30750
                ;               SUBD    TESTADDRS
                ;               ADDB    #30H
                ;               JSR     TRANBYTE2
                ;SENDX          LDAB    #58H            ;OUTPUT "X" FOR SUCCESSFUL LOAD
                ;               JSR     TRANBYTE2
                ;               RTS
                ;-------------------------------------------------------------------
                ;THE ABS SUBROUTINES ADDS THE FILTERED SAMPLE'S ABSOLUTE VALUES
0088& FC 907E   ABS             LDD     MEGAADDH        ;IF H IS - SO IS TOTAL
008B& 2B 0B                     BMI     MAKEPLUS
008D& 27 02                     BEQ     CHECKLOW
008F& 20 19                     BRA     MACROADD
0091& FC 907C   CHECKLOW        LDD     MEGAADDL
0094& 2B 02                     BMI     MAKEPLUS
0096& 20 12                     BRA     MACROADD
0098& CC 0000   MAKEPLUS        LDD     #0
009B& B3 907E                   SUBD    MEGAADDH
009E& FD 907E                   STD     MEGAADDH
00A1& CC 0000                   LDD     #0
00A4& B3 907C                   SUBD    MEGAADDL
00A7& FD 907C                   STD     MEGAADDL
00AA& FC 907C   MACROADD        LDD     MEGAADDL
00AD& F3 9083                   ADDD    MACROADDL
00B0& 29 05                     BVS     MEGAINC2
00B2& FD 9083                   STD     MACROADDL
00B5& 20 0F                     BRA     ADDHI
00B7& 83 8000   MEGAINC2        SUBD    #-32768         ;STORE VALIDATED RESULT
00BA& FD 9083                   STD     MACROADDL
00BD& FC 9085                   LDD     MACROADDH
00C0& C3 0001                   ADDD    #1
00C3& FD 9085                   STD     MACROADDH
00C6& FC 907E   ADDHI           LDD     MEGAADDH
00C9& F3 9085                   ADDD    MACROADDH
00CC& FD 9085                   STD     MACROADDH
00CF& CC 0000                   LDD     #0
00D2& FD 907E                   STD     MEGAADDH
```

```
00D5& FD 907C                   STD     MEGAADDL
00D8& 39                        RTS
                 ;---------------------------------------------------------------
00D9& CC 900F     CHCKNORM      LDD     #AVERAGE+1     ;SEND AVERAGE VALUE FIRST
00DC& BD 0105&                  JSR     ASCII
00DF& FC 900C                   LDD     FILPAR
00E2& 1A 83 0065                CPD     #101
00E6& 26 06                     BNE     ENDROUT3
00E8& CC 0000                   LDD     #0
00EB& FD 900C                   STD     FILPAR
00EE& 39          ENDROUT3      RTS
                 ;---------------------------------------------------------------
00EF& CC F448     DUMPAMP       LDD     #MATCHDATA
00F2& BD 0105&                  JSR     ASCII
00F5& FC 900C                   LDD     FILPAR
00F8& 1A 83 000A                CPD     #10
00FC& 26 06                     BNE     ENDROUT2
00FE& CC 0000                   LDD     #0
0101& FD 900C                   STD     FILPAR
0104& 39          ENDROUT2      RTS
                 ;-----------------------------------------------------------------
                 ;THE FOLLOWING ROUTINE WAS TO DUMP OUT THE STORED FILTER PARAMETERS
                 ;AND IS FOR TEST PURPOSES ONLY.  THIS ROUTINE WAS COMMENTED OUT
                 ;ON 8/24/98
                 ;DUMPPARAM     LDD     #FILT01WGHT
                 ;              JSR     ASCII
                 ;              LDD     FILPAR          ;PARAMETERS=10X23=230
                 ;              CPD     #230
                 ;              BNE     ENDROUT1
                 ;              LDD     #0
                 ;              STD     FILPAR
                 ;ENDROUT1      RTS
                 ;-----------------------------------------------------------------
0105& F3 900C     ASCII         ADDD    FILPAR
0108& 8F                        XGDX
0109& FC 900C                   LDD     FILPAR
010C& C3 0001                   ADDD    #1
010F& FD 900C                   STD     FILPAR
0112& E6 00       NEXTBYTE8     LDAB    0,X
```

```
0114& C1 00                        CMPB    #0
0116& 2D 07                        BLT     SENDNEG
0118& 86 00                        LDAA    #0
011A& FD 9001                      STD     TEMP1
011D& 20 22                        BRA     SENDNUM
011F& C6 2D            SENDNEG     LDAB    #45
0121& BD 0184&                     JSR     TRANBYTE2
0124& E6 00                        LDAB    0,X
0126& C1 80                        CMPB    #-128
0128& 26 0A                        BNE     OK128
012A& C6 31                        LDAB    #31H
012C& BD 0184&                     JSR     TRANBYTE2
012F& CC 001C                      LDD     #28
0132& 20 1E                        BRA     DIV10
0134& F7 9001          OK128       STAB    TEMP1
0137& C6 00                        LDAB    #0
0139& F0 9001                      SUBB    TEMP1
013C& 86 00                        LDAA    #0
013E& FD 9001                      STD     TEMP1           ;D=POSITIVE 1-128
0141& 1A 83 0064       SENDNUM     CPD     #100
0145& 2D 0B                        BLT     DIV10
0147& C6 31                        LDAB    #31H
0149& BD 0184&                     JSR     TRANBYTE2
014C& FC 9001                      LDD     TEMP1
014F& 83 0064                      SUBD    #100
0152& CE 000A          DIV10       LDX     #10
0155& 02                           IDIV
0156& FD 9001                      STD     TEMP1
0159& 8F                           XGDX                    ;QUOTIENT IN D
015A& C3 0030                      ADDD    #30H
015D& BD 0184&                     JSR     TRANBYTE2
0160& FC 9001                      LDD     TEMP1
0163& CB 30                        ADDB    #30H
0165& BD 0184&                     JSR     TRANBYTE2
0168& C6 20                        LDAB    #20H            ;SEND SPACE AFTER EACH
016A& BD 0184&                     JSR     TRANBYTE2       ;ASCII VALUE
016D& 39                           RTS
                       ;--------------------------------------------------------------
016E& 18 CE 0000       ECHOID      LDY     #0
```

```
0172& CE 04EB&        352                 LDX     #ID
0175& E6 00           353 NEXTBYTE7       LDAB    0,X
0177& BD 0184&        354                 JSR     TRANBYTE2
017A& 08              355                 INX
017B& 18 08           356                 INY
017D& 18 8C 0011      357                 CPY     #17
0181& 26 F2           358                 BNE     NEXTBYTE7
0183& 39              359                 RTS
                      360 ;-----------------------------------------------------------------------
0184& B6 802E         361 TRANBYTE2             LDAA        SCSR        ;CLEAR SCSR
0187& F7 802F         362                       STAB        SCDR        ;BYTE TO BE SENT LOADED
018A& B6 802E         363 CHECKTRANS            LDAA        SCSR
018D& 84 80           364                       ANDA        #10000000B
018F& 27 F9           365                       BEQ         CHECKTRANS
0191& 39              366                       RTS
                      367 ;-----------------------------------------------------------------------
0192& 18 CE 0003      368 TRANSDELAY            LDY         #0003H      ;THIS ROUTINE PROVIDES 250mS
0196& CE 3DEB         369 NEXT31                LDX         #03DEBH     ;DELAY FOR TRANSMIT LINE
0199& 09              370 NEXT30                DEX
019A& 26 FD           371                       BNE         NEXT30
019C& 18 09           372                       DEY
019E& 26 F6           373                       BNE         NEXT31
01A0& 39              374                       RTS
                      375 ;-----------------------------------------------------------------------
01A1& B6 8000         376 PORTTOGL              LDAA        PORTAADR
01A4& 88 FF           377                       EORA        #11111111B
01A6& B7 8000         378                       STAA        PORTAADR
01A9& B6 8008         379                       LDAA        PORTDADR
01AC& 88 FF           380                       EORA        #11111111B
01AE& B7 8008         381                       STAA        PORTDADR
01B1& 39              382                       RTS
                      383 ;-----------------------------------------------------------------------
01B2& 86 80           384 FULLSEC2              LDAA        #10000000B       ;LIGHT FAR LED
01B4& B7 8000         385                       STAA        PORTAADR
01B7& 18 CE 000A      386                       LDY         #000AH
01BB& CE AACC         387 NEXTX11               LDX         #0AACCH
01BE& 09              388 NEXTX10               DEX
01BF& 26 FD           389                       BNE         NEXTX10
01C1& 18 09           390                       DEY
```

```
01C3& 26 F6                      BNE      NEXTX11
01C5& B6 8000                    LDAA     PORTAADR
01C8& 84 7F                      ANDA     #01111111B    ;TURN OFF FAR LED
01CA& B7 8000                    STAA     PORTAADR
01CD& BD 046E&                   JSR      FULLSEC       ;NO MORE INTERUPTS FOR
01D0& BD 046E&                   JSR      FULLSEC       ;ABOUT 2 SECONDS
01D3& 3B                         RTI
                ;-----------------------------------------------------------------------
                ;THIS INTERRUPT HANDLER RECORDS 5 SECONDS OF AUDIO AFTER THRESHOLD
                ;IS REACHED
                ;THE FOLLOWING THREE LINES LIT A TEST LED DURING RECORDING
                ;AND WERE COMMENTED OUT ON 8/27/98
                ;                LDAA     PORTAADR
                ;                ORAA     #00001000B    ;LIGHT INDICATOR LED
                ;                STAA     PORTAADR
01D4& B6 8008   RECORD5          LDAA     PORTDADR
01D7& 84 FB                      ANDA     #11111011B    ;ZERO OUT RAM A16=PD2
01D9& B7 8008                    STAA     PORTDADR
01DC& 86 01                      LDAA     #00000001B    ;SET FOR SINGLE CONVERSION
01DE& B7 8030                    STAA     ADCTL         ;START CONVERSIONS
01E1& CE 0000                    LDX      #0            ;START AT ADDRESS=0
01E4& 86 01     NEXTSAMP3        LDAA     #00000001B    ;2 CYC SET FOR SINGLE CONVERSION
01E6& B7 8030                    STAA     ADCTL         ;4 CYC START CONVERSIONS
                ;                                       SUBTOTAL = 6 CYCLES
01E9& B6 8030   CHECKAD2         LDAA     ADCTL         ;4CYC CHECK CONVERSION COMPLETE
01EC& 84 80                      ANDA     #10000000B    ;2CYC
01EE& 27 F9                      BEQ      CHECKAD2      ;3CYC
                ;                                       ABOVE CYC 9 CYCLES DON'T COUNT
01F0& B6 8031                    LDAA     ADR1          ;4 CYC
01F3& 2C 06                      BGE      SUB128        ;3 CYC CONVERT 0-255->-128 TO 127
                ;                                       ;SUBTOTAL CYCLES=13
01F5& 8B 7F                      ADDA     #127          ;2 CYC
01F7& 8B 01                      ADDA     #1            ;2 CYC
01F9& 20 06                      BRA      CONT522       ;3 CYC SUBTOTAL THIS ROUT=20
01FB& 80 7F     SUB128           SUBA     #127          ;2 CYC
01FD& 80 01                      SUBA     #1            ;2 CYC
01FF& 20 00                      BRA      CONT522       ;3 CYC SUBTOTAL THIS ROUT=20
0201& A7 00     CONT522          STAA     0,X           ;4 CYC SUBTOT=24
0203& 08                         INX                    ;3 CYC SUBTOT=27
```

```
0204& 8C 7800          430                  CPX     #30720       ;4 CYC SUBTOT=31
0207& 27 09            431                  BEQ     ENDREC       ;3 CYC SUBTOT=34  NEED 77CYC
0209& 86 07            432                  LDAA    #7           ;2 CYC SUBTOT=36
020B& 4A               433 MAINDELAY        DECA                 ;2 CYC
020C& 26 FD            434                  BNE     MAINDELAY    ;3 CYC LOOP=5 5x7=35 CYC ST=71
020E& 20 00            435                  BRA     ADDELAY      ;2 CYC SUBTOT=74
0210& 20 D2            436 ADDELAY          BRA     NEXTSAMP3    ;3 CYC SUBTOT=76
0212& B6 8000          437 ENDREC           LDAA    PORTAADR
0215& 84 F7            438                  ANDA    #11110111B
0217& B7 8000          439                  STAA    PORTAADR     ;TURN OFF LED
021A& BD 021E&         440                  JSR     DIGIFILT
021D& 3B               441                  RTI
                       442 ;-------------------------------------------------------------
021E& CC 0000          443 DIGIFILT         LDD     #0
0221& FD 907C          444                  STD     MEGAADDL      ;ZERO OUT REGISTERS
0224& FD 907E          445                  STD     MEGAADDH      ;FOR DIGITAL FILTER SUMS
0227& FD 9083          446                  STD     MACROADDL
022A& FD 9085          447                  STD     MACROADDH
022D& B6 900A          448                  LDAA    EPOCH         ;LOAD ENDING ADDRESS OF DATA
0230& C6 0C            449                  LDAB    #12           ;
0232& 3D               450                  MUL                   ;D=0-108 SO B=0-108
0233& 17               451                  TBA
0234& 5F               452                  CLRB                  ;D=(0-108)*256= 0-27,648
0235& FD 9087          453                  STD     STARTDATA
0238& C3 0BE8          454                  ADDD    #3048         ;D=3071+(N*3072)-23
023B& FD 9008          455                  STD     ENDDATA
023E& 18 FE 9087       456                  LDY     STARTDATA     ;START SAMPLE DATA IN Y REG
0242& C6 17            457                  LDAB    #23           ;23 STAGES PER FILTER
0244& B6 9078          458                  LDAA    FREQ          ;FREQ= 0-9
0247& 3D               459                  MUL                   ;D=0 TO 207
0248& C3 04FC&         460                  ADDD    #FILT01WGHT
024B& FD 9079          461                  STD     FILTSTART     ;FILTER PARAMETER START IN D
024E& 8F               462                  XGDX                  ;X REG HAS FILT START
024F& 7F 9000          463                  CLR     COUNT
0252& A6 00            464 NEXTELEM         LDAA    0,X
0254& 18 E6 00         465                  LDAB    0,Y
0257& BD 0416&         466                  JSR     SIGNMULT
025A& BD 03E7&         467                  JSR     MEGAADD
025D& 7C 9000          468                  INC     COUNT
```

```
0260& B6 9000                         LDAA    COUNT
0263& 81 17                           CMPA    #23
0265& 27 05                           BEQ     SUMDATA       ;THIS DATA SAMPLE OF THE
0267& 08                              INX                   ;OUTPUT NOW TO BE SUMMED
0268& 18 08                           INY
026A& 20 E6                           BRA     NEXTELEM
026C& BD 0088&      SUMDATA           JSR     ABS
026F& 7F 9000                         CLR     COUNT
0272& 18 BC 9008                      CPY     ENDDATA
0276& 27 0C                           BEQ     RESETSAMP       ;DONE WITH THIS FILTER
0278& FE 9079                         LDX     FILTSTART     ;RESET FILTER ADDRESS
027B& 18 8F                           XGDY                  ;GET Y INTO D REG
027D& 83 0015                         SUBD    #21           ;0-22 SUB 21 INCS BY 1
0280& 18 8F                           XGDY                  ;NEW ADDRESS IN Y
0282& 20 CE                           BRA     NEXTELEM
0284& F6 900A       RESETSAMP         LDAB    EPOCH        ;STORE RESULT
0287& 86 0A                           LDAA    #10
0289& 3D                              MUL                  ;D=0-90->B=0-90
028A& FB 9078                         ADDB    FREQ         ;B=0-99
028D& C3 9010                         ADDD    #FILTAMP     ;ADD OFFSET
0290& FD 9091                         STD     STOREADDR
0293& FC 9085                         LDD     MACROADDH
0296& CE 0014                         LDX     #20          ;DIV BY 20 SHOULD KEEP ALL
0299& 02                              IDIV                 ;RESULTS<255 MAXIMUM!
029A& 8F                              XGDX                 ;GET QUOTIENT IN D REGISTER
029B& FE 9091                         LDX     STOREADDR
029E& E7 00                           STAB    0,X
                    ;------------------------------------------------------------------
                    ;NEXT SECTION COMMENTED OUT ON 8/19/98 WAS FOR TEST DUMP OF
                    ;RAW DIGITAL FILTER OUTPUT DATA FOR TEST PURPOSES ONLY
                    ;NEXTBYTE1         LDD     #MACROADDL    ;OUTPUT 4 DATA BYTES
                    ;                 JSR     ASCII
                    ;                 LDD     FILPAR
                    ;                 CPD     #4
                    ;                 BEQ     RESFIL6
                    ;                 BRA     NEXTBYTE1
                    ;RESFIL6           LDD     #0
                    ;                 STD     FILPAR
                    ;------------------------------------------------------------------
```

```
02A0& B6 9078                        LDAA    FREQ
02A3& 4C                             INCA
02A4& B7 9078                        STAA    FREQ
02A7& 81 0A                          CMPA    #10
02A9& 27 03                          BEQ     RESET815      ;DONE WITH ALL 10 FREQUENCIES
02AB& 7E 021E&                       JMP     DIGIFILT
02AE& 7F 9078         RESET815       CLR     FREQ          ;RESET FREQUENCY
02B1& 7C 900A                        INC     EPOCH
02B4& B6 900A                        LDAA    EPOCH
02B7& 81 0A                          CMPA    #10           ;REGULAR VALUES=0-9
02B9& 27 03                          BEQ     MAJORRES
02BB& 7E 021E&                       JMP     DIGIFILT
02BE& 7F 900A         MAJORRES       CLR     EPOCH
02C1& BD 02FC&                       JSR     NORMALIZE     ;NORMALIZE DATA
02C4& BD 034C&                       JSR     MATCH         ;CALCULATED CORELLATIONS
                      ;SKIP MATCH DECISION IN "FINAL" SUBROUTINE AND INSTEAD
                      ;SEND ALL FOUR CORELLATION COEFFICIENTS TO THE DAHLGREN UNIT
                      ;              JSR     FINAL         ;DO FINAL ANALYSIS AND STORE
02C7& BD 047D&                       JSR     DAHLGREN      ;AWAIT TRANSFER OPPORTUNITY
02CA& 39                             RTS
                      ;-------------------------------------------------------------------------
                      ;AS OF AUGUST, 1998, THIS ROUTINE WILL NOT BE USED.  ALL FOUR
                      ;CORRELATION VALUES WILL BE SENT OUT TO THE DAHLGREN PROCESSOR
02CB& FC 9089         FINAL          LDD     MATCHQUAL1    ;DIFF=0 TO 25,600 (100*256)
02CE& 1A B3 0771&                    CPD     THRESHOLD
02D2& 2E 03                          BGT     CHECK2
02D4& 7C 9074                        INC     NUMCALL1
02D7& FC 908B         CHECK2         LDD     MATCHQUAL2
02DA& 1A B3 0771&                    CPD     THRESHOLD
02DE& 2E 03                          BGT     CHECK3
02E0& 7C 9075                        INC     NUMCALL2
02E3& FC 908D         CHECK3         LDD     MATCHQUAL3
02E6& 1A B3 0771&                    CPD     THRESHOLD
02EA& 2E 03                          BGT     CHECK4
02EC& 7C 9076                        INC     NUMCALL3
02EF& FC 908F         CHECK4         LDD     MATCHQUAL4
02F2& 1A B3 0771&                    CPD     THRESHOLD
02F6& 2E 03                          BGT     DONEFINL
02F8& 7C 9077                        INC     NUMCALL4
```

```
02FB& 39             DONEFINL         RTS
                     ;-------------------------------------------------------------------------
                     ;SET AVERAGE VALUE TO 38 BY MULTIPLYING ALL VALUES BY C=38/AVERAGE
                     ;NOTE: AVERAGE OF 50 CAUSES PROBLEMS WITH DYNAMIC RANGE OF FILTER OUTPTS
02FC& CC 0000        NORMALIZE        LDD    #0
02FF& B7 9000                         STAA   COUNT
0302& FD 907C                         STD    MEGAADDL
0305& FD 907E                         STD    MEGAADDH
0308& CE 9010                         LDX    #FILTAMP
030B& 86 00          SUMMORE          LDAA   #0
030D& E6 00                           LDAB   0,X              ;VALUE= 0-256 IN D REG
030F& FD 9001                         STD    TEMP1
0312& BD 03E7&                        JSR    MEGAADD
0315& 08                              INX
0316& 7C 9000                         INC    COUNT
0319& B6 9000                         LDAA   COUNT
031C& 81 64                           CMPA   #100
031E& 26 EB                           BNE    SUMMORE
0320& FC 907C                         LDD    MEGAADDL         ;VALUE=0-25,600 (100X256)
0323& CE 0064                         LDX    #100             ;AVERAGE=(50*100)/100
0326& 02                              IDIV                    ;X REG=0-256
0327& FF 900E                         STX    AVERAGE          ;STORE AVERAGE
032A& 7F 9000                         CLR    COUNT
032D& 18 CE 9010                      LDY    #FILTAMP
0331& 18 A6 00       MORENORM         LDAA   0,Y              ;D REG NOW HAS VALUE X 256
0334& C6 26                           LDAB   #38              ;AIM FOR AVERAGE=38
0336& 3D                              MUL                     ;D REG NOW HAS + VALUE X 38
0337& FE 900E                         LDX    AVERAGE          ;NOW HAS 0 TO 32,768
033A& 02                              IDIV                    ;QUOTIENT IN X REG=0-256
033B& 8F                              XGDX                    ;QUOTIENT IN D
033C& 18 E7 00                        STAB   0,Y              ;STORE NORMALIZED VALUE
033F& 18 08                           INY
0341& 7C 9000                         INC    COUNT
0344& B6 9000                         LDAA   COUNT
0347& 81 64                           CMPA   #100
0349& 26 E6                           BNE    MORENORM
                     ;-------------------------------------------------------------------------
                     ;THE FOLLOWING SECTION WAS COMMENTED OUT ON 8/19/98
                     ;THIS SECTION HAD TRANSFERRED THE NORMALIZED FILTER OUPUTS TO THE
```

```
                 ;PC AND WAS FOR TEST PURPOSES ONLY
                 ;OUTNORM           JSR    FULLSEC          ;ONE SEC DELAY BETWEEN BYTES
                 ;                  LDD    #FILTAMP         ;ADDRESS=FILPAR+#FILTAMP
                 ;                  JSR    ASCII            ;SENT BY ASCII ROUTINE
                 ;                  LDAB   #20H
                 ;                  JSR    TRANBYTE2        ;SEND A SPACE
                 ;                  LDD    FILPAR
                 ;                  CPD    #100             ;CHECK FOR 100 PARAMETERS
                 ;                  BEQ    ENDNORM          ;SENT
                 ;                  BRA    OUTNORM
                 ;ENDNORM           LDD    #0
                 ;                  STD    FILPAR
034B& 39                            RTS
                 ;-------------------------------------------------------------------
                 ;THIS ROUTINE WILL DO THE PATTERN MATCH
                 ;STORES ALL 8 BYTES OF MATCHQUAL DATA
034C& 7F 9000    MATCH              CLR    COUNT
034F& CC 0000                       LDD    #0
0352& FD 9094                       STD    PATTERN
0355& FD 907C                       STD    MEGAADDL
0358& FD 907E                       STD    MEGAADDH
035B& 18 CE 05E1&                   LDY    #MATCHEPOCH1T1   ;START OF TEMPLATE DATA
035F& CE 9010    SUMMER1            LDX    #FILTAMP         ;START OF NORMALIZED FILTER
0362& 18 E6 00   SUMPAT             LDAB   0,Y              ;TEMPLATE DATA INTO LOW BYTE
0365& 86 00                         LDAA   #0               ;ZERO OUT HIGH BYTE
0367& FD 9001                       STD    TEMP1
036A& E6 00                         LDAB   0,X              ;OUTPUT DATA INTO LOW BYTE
036C& 86 00                         LDAA   #0               ;ZERO OUT HIGH BYTE
036E& B3 9001                       SUBD   TEMP1            ;RESULT IN D REG
0371& FD 9001                       STD    TEMP1            ;STORE RESULT NOW
0374& 2B 02                         BMI    PLUSIT
0376& 20 09                         BRA    ADDIT
0378& CC 0000    PLUSIT             LDD    #0               ;NEGATE RESULT
037B& B3 9001                       SUBD   TEMP1            ;SO THAT ITS ALWAYS POS.
037E& FD 9001                       STD    TEMP1
0381& B6 9002    ADDIT              LDAA   TEMP2            ;LOW BYTE (0-255)
0384& F6 9002                       LDAB   TEMP2            ;LOW BYTE (0-255)
0387& 3D                            MUL
0388& FD 9001                       STD    TEMP1            ;STORE SQUARED RESULT
```

```
038B& BD 03E7&                        JSR    MEGAADD
038E& 7C 9000                         INC    COUNT
0391& 08                              INX
0392& 18 08                           INY
0394& B6 9000                         LDAA   COUNT
0397& 81 64                           CMPA   #100
0399& 27 02                           BEQ    NEXTPATT
039B& 20 C5                           BRA    SUMPAT
039D& 7F 9000         NEXTPATT        CLR    COUNT            ;SUM=0-25,500=>MEGAADDL
03A0& FC 907C                         LDD    MEGAADDL
03A3& CE 0064                         LDX    #100             ;AVERAGE=0-255
03A6& 02                              IDIV
03A7& 8F                              XGDX                    ;PLACE RESULT INTO D REG
03A8& FD 907C                         STD    MEGAADDL
03AB& FC 907E                         LDD    MEGAADDH
03AE& CE 0064                         LDX    #100
03B1& 02                              IDIV
03B2& 8F                              XGDX
03B3& FD 907E                         STD    MEGAADDH
03B6& CC 9089         STOREMATCH      LDD    #MATCHQUAL1
03B9& F3 9094                         ADDD   PATTERN          ;CORRECT ADDRESS IN D REG
03BC& F3 9094                         ADDD   PATTERN          ;2 BYTES PER MATCH
03BF& 8F                              XGDX                    ;CORRECT ADDRESS INTO X
03C0& FC 907E                         LDD    MEGAADDH
03C3& E7 00                           STAB   0,X              ;STORE LOW BYTE
03C5& 08                              INX
03C6& FC 907C                         LDD    MEGAADDL
03C9& E7 00                           STAB   0,X              ;STORE LOW BYTE
03CB& FC 9094                         LDD    PATTERN
03CE& C3 0001                         ADDD   #1
03D1& FD 9094                         STD    PATTERN
03D4& 1A 83 0004                      CPD    #4
03D8& 27 0C                           BEQ    ENDMATCH
03DA& CC 0000                         LDD    #0
03DD& FD 907C                         STD    MEGAADDL
03E0& FD 907E                         STD    MEGAADDH
03E3& 7E 035F&                        JMP    SUMMER1          ;DO NEXT PATTERN
03E6& 39              ENDMATCH        RTS
                      ;----------------------------------------------------------------------
```

```
03E7& FC 9001        MEGAADD       LDD    TEMP1
03EA& F3 907C                      ADDD   MEGAADDL
03ED& 29 04                        BVS    MEGAINC
03EF& FD 907C                      STD    MEGAADDL
03F2& 39                           RTS
03F3& 2B 11          MEGAINC       BMI    FLIPIT1          ;RESULT WAS+/LOOKS -
03F5& 83 8000                      SUBD   #-32768          ;STORE VALIDATED RESULT
03F8& FD 907C                      STD    MEGAADDL
03FB& FC 907E                      LDD    MEGAADDH
03FE& 83 0001                      SUBD   #1
0401& FD 907E                      STD    MEGAADDH
0404& 20 0F                        BRA    ENDMEG
0406& 83 8000        FLIPIT1       SUBD   #-32768
0409& FD 907C                      STD    MEGAADDL
040C& FC 907E                      LDD    MEGAADDH
040F& C3 0001                      ADDD   #1
0412& FD 907E                      STD    MEGAADDH
0415& 39             ENDMEG        RTS
                     ;----------------------------------------------------------------------
0416& B7 9001        SIGNMULT      STAA   TEMP1
0419& F7 9002                      STAB   TEMP2
041C& F8 9001                      EORB   TEMP1
041F& C4 80                        ANDB   #10000000B
0421& F7 907B                      STAB   SIGN
0424& F6 9001                      LDAB   TEMP1
0427& 2B 07                        BMI    NEGATE
0429& F6 9002        TESTIT2       LDAB   TEMP2
042C& 2B 15                        BMI    NEGATE1
042E& 20 24                        BRA    MULT1
0430& C1 80          NEGATE        CMPB   #-128
0432& 26 07                        BNE    GOON81
0434& 86 7F                        LDAA   #127
0436& B7 9001                      STAA   TEMP1
0439& 20 EE                        BRA    TESTIT2
043B& 86 00          GOON81        LDAA   #0
043D& 10                           SBA                  ;RESULT ACCUM A
043E& B7 9001                      STAA   TEMP1
0441& 20 E6                        BRA    TESTIT2
0443& C1 80          NEGATE1       CMPB   #-128
```

```
0445& 26 07                         BNE    GOON81A
0447& 86 7F                         LDAA   #127
0449& B7 9002                       STAA   TEMP2
044C& 20 06                         BRA    MULT1
044E& 86 00         GOON81A         LDAA   #0
0450& 10                            SBA
0451& B7 9002                       STAA   TEMP2
0454& B6 9001       MULT1           LDAA   TEMP1          ;0-127
0457& F6 9002                       LDAB   TEMP2          ;0-127
045A& 3D                            MUL                   ;0-16,129
045B& FD 9001                       STD    TEMP1
045E& B6 907B                       LDAA   SIGN
0461& 26 01                         BNE    NEGATEF
0463& 39                            RTS
0464& CC 0000       NEGATEF         LDD    #0
0467& B3 9001                       SUBD   TEMP1
046A& FD 9001                       STD    TEMP1
046D& 39                            RTS
                    ;-----------------------------------------------------------------------
046E& 18 CE 000A    FULLSEC         LDY         #000AH
0472& CE AACC       NEXT11          LDX         #0AACCH
0475& 09            NEXT10          DEX
0476& 26 FD                         BNE         NEXT10
0478& 18 09                         DEY
047A& 26 F6                         BNE         NEXT11
047C& 39                            RTS
                    ;--------------------------------------------------------------------------
                    ;THIS ROUTINE SENDS THE DATA TO THE DAHLGREN PROCESSOR UPON REQUEST
                    ;1) IT IS EXECUTED AS AN INFINITE LOOP AFTER AN AUDIO ANALYSIS
                    ;2) IT FIRST CHECKS PA0 (HANDSHAKE IN) TO SEE IF IT IS HIGH
                    ;3) IF IT IS HIGH, THE DACIS ENABLES THE SERIAL OUPUT LINE
                    ;4) THE DACIS THEN WAITS AND LOOKS FOR AN "S" (SEND DATA)
                    ;5) THE DACIS THEN SENDS THE FOUR SERIAL SINGLE BYTE VALUES CORRESPONDING
                    ;   TO THE FOUR TEMPLATE CORRELATIONS FOR THIS WOLF CALL
                    ;6) THE DACIS THEN LOOPS FOR APPROXIMATELY ONE HALF SECOND CHECKING TO
                    ;   SEE IF THE DAHLGREN PROCESSOR HAS SENT AN "R" FOR RESEND THE DATA
                    ;7) AFTER THE ONE SECOND PASSES, THE ROUTINE RESETS COUNTERS AND
                    ;   EXECUTES AN RTI
                    ;8) THIS ROUTINE MODIFIED TO SEND ALL 8 BYTES OF MATCH DATA
```

```
047D& 7F 9098        DAHLGREN     CLR    DAHLFLAG        ;CLEAR DAHLGREN FLAG
0480& B6 8000                     LDAA   PORTAADR        ;UNTIL DAT SENT ONCE
0483& 8A 08                       ORAA   #00001000B      ;SET PA3 HIGH
0485& B7 8000                     STAA   PORTAADR
0488& B6 8000        DAHLGREN2    LDAA   PORTAADR
048B& 84 01                       ANDA   #00000001B      ;CHECK BIT PA0
048D& 26 02                       BNE    SENDDAHL1
048F& 20 F7                       BRA    DAHLGREN2       ;CHECK HNDSHK IN AGAIN
0491& 86 0C          SENDDAHL1    LDAA   #00001100B      ;ENABLE TRANS/REC
0493& B7 802D                     STAA   SCCR2           ;NO TRANS. OR REC. INT.
0496& CE 6C81        SENDDAHL2    LDX    #27777        ;27,777 X 18uS =0.5 SEC
0499& B6 802E        SENDDAHL3    LDAA   SCSR          ;4 CYC
049C& 84 20                       ANDA   #00100000B    ;2 CYC  CHECK RECEIVE DATA
049E& 26 0C                       BNE    CHECKCOMM1    ;3 CYC  REGISTER FULL
04A0& 09                          DEX                  ;3 CYC
04A1& 27 02                       BEQ    CHECKFLAG1    ;3 CYC
04A3& 20 F4                       BRA    SENDDAHL3     ;3 CYC TRY AGAIN
                     ; TOTAL= 18 CYCLES
04A5& B6 9098        CHECKFLAG1   LDAA   DAHLFLAG        ;CHECK FOR DATA SENT ONCE
04A8& 26 2E                       BNE    ENDAHLGREN      ;QUIT IF SENT ONCE
04AA& 20 EA                       BRA    SENDDAHL2       ;MUST TRANSMIT ONCE!
04AC& B6 802F        CHECKCOMM1   LDAA   SCDR            ;GET DATA INTO A REG
04AF& F6 802E                     LDAB   SCSR            ;CLEAR RDRF FLAG
04B2& 81 53                       CMPA   #53H            ;"S"=SEND DATA
04B4& 27 06                       BEQ    SEND4BYTES
04B6& 81 52                       CMPA   #52H            ;"R"=RESEND DATA
04B8& 27 02                       BEQ    SEND4BYTES
04BA& 20 DA                       BRA    SENDDAHL2       ;CONTINUE LOOKING
04BC& 86 01          SEND4BYTES   LDAA   #1
04BE& B7 9098                     STAA   DAHLFLAG
04C1& 18 CE 0004                  LDY    #4
04C5& CE 9089                     LDX    #MATCHQUAL1     ;FIRST MATCH BYTE
04C8& 08                          INX
04C9& E6 00          SNDMORE1     LDAB   0,X             ;LOAD DATA INTO B REG
04CB& BD 0184&                    JSR    TRANBYTE2       ;SEND DATA
04CE& 08                          INX
04CF& 08                          INX
04D0& 18 09                       DEY
04D2& 27 C2                       BEQ    SENDDAHL2       ;DO IT AGAIN?
```

```
04D4& 20 F3                                     BRA      SNDMORE1
04D6& 20 00                                     BRA      ENDAHLGREN
04D8& 86 00              ENDAHLGREN             LDAA     #00000000B       ;DISABLE TRANS/REC
04DA& B7 802D                                   STAA     SCCR2            ;NO TRANS. OR REC. INT.
04DD& 7F 9098                                   CLR      DAHLFLAG         ;NOT NEEDED HERE BUT CLR
04E0& B6 8000                                   LDAA     PORTAADR         ;ANYWAY
04E3& 84 F7                                     ANDA     #11110111B       ;SET PA3 BACK LOW
04E5& B7 8000                                   STAA     PORTAADR
04E8& 7E 006D&                                  JMP      NEWSTART         ;RESTART FOR NEXT CALL
                         ;-------------------------------------------------------------------
04EB& 42 49 52 44 42     ID                FCB  42H,49H,52H,44H,42H,4FH,52H,4EH,45H,20H ;BIRDBORNE
04F0& 4F 52 4E 45 20
04F5& 55 4E 49 54 20                       FCB  55H,4EH,49H,54H,20H,20H,31H             ;UNIT__1
04FA& 20 31
                         ; DIGITAL FILTER WEIGHTS STORED HERE
                         ; FOR THE FOLLOWING CENTER FREQUENCIES:
                         ; 250Hz,500Hz,750Hz,1000Hz,1250Hz,1500Hz,1750Hz,2000Hz,2250Hz,2500Hz
04FC& AB AE B9 CB E3     FILT01WGHT        FCB  -85,-82,-71,-53,-29,0,30,60,87,108,122
0501& 00 1E 3C 57 6C
0506& 7A
0507& 7F 7A 6C 57 3C                       FCB  127,122,108,87,60,30,0,-29,-53,-71,-82,-85
050C& 1E 00 E3 CB B9
0511& AE AB
0513& 4B 2E FF CB A0     FILT02WGHT        FCB  75,46,-1,-53,-96,-115,-103,-61,0,62,109,127
0518& 8D 99 C3 00 3E
051D& 6D 7F
051F& 6D 3E 00 C3 99                       FCB  109,62,0,-61,-103,-115,-96,-53,-1,46,75
0524& 8D A0 CB FF 2E
0529& 4B
052A& C2 00 46 69 4D     FILT03WGHT        FCB  -62,0,70,105,77,-1,-84,-122,-88,0,89,127
052F& FF AC 86 A8 00
0534& 59 7F
0536& 59 00 A8 86 AC                       FCB  89,0,-88,-122,-84,-1,77,105,70,0,-62
053B& FF 4D 69 46 00
0540& C2
0541& 2B D1 9C CB 36     FILT04WGHT        FCB  43,-47,-100,-53,54,114,59,-61,-124,-63,63,127
0546& 72 3B C3 84 C1
054B& 3F 7F
054D& 3F C1 84 C3 3B                       FCB  63,-63,-124,-61,59,114,54,-53,-100,-47,43
```

```
0552& 72 36 CB 9C D1
```



```
0557& 2B
0558& E9 51 46 CB 95      804 FILT05WGHT       FCB  -23,81,70,-53,-107,0,114,60,-88,-109,32,127
055D& 00 72 3C A8 93
0562& 20 7F
0564& 20 93 A8 3C 72      805                  FCB  32,-109,-88,60,114,0,-107,-53,70,81,-23
0569& 00 95 CB 46 51
056E& E9
056F& FF A2 00 69 FF      806 FILT06WGHT       FCB  -1,-94,0,105,-1,-115,0,121,-1,-126,0,127
0574& 8D 00 79 FF 82
0579& 00 7F
057B& 00 82 FF 79 00      807                  FCB  0,-126,-1,121,0,-115,-1,105,-94,-1
0580& 8D FF 69 A2 FF
0585& 16 51 B9 CB 6A      808 FILT07WGHT       FCB  22,81,-71,-53,106,-1,-115,60,87,-109,-33,127
058A& FF 8D 3C 57 93
058F& DF 7F
0591& DF 93 57 3C 8D      809                  FCB  -33,-109,87,60,-115,-1,106,-53,-71,81,22
0596& FF 6A CB B9 51
059B& 16
059C& D4 D1 63 CB C9      810 FILT08WGHT       FCB  -44,-47,99,-53,-55,114,-60,-61,123,-63,-64,127
05A1& 72 C4 C3 7B C1
05A6& C0 7F
05A8& C0 C1 7B C3 C4      811                  FCB  -64,-63,123,-61,-60,114,-55,-53,99,-47,-44
05AD& 72 C9 CB 63 D1
05B2& D4
05B3& 3D FF B9 69 B2      812 FILT09WGHT       FCB  61,-1,-71,105,-78,0,83,-122,87,-1,-90,127
05B8& 00 53 86 57 FF
05BD& A6 7F
05BF& A6 FF 57 86 53      813                  FCB  -90,-1,87,-122,83,0,-78,105,-71,-1,61
05C4& 00 B2 69 B9 FF
05C9& 3D
05CA& B4 2E FF CB 5F      814 FILT10WGHT       FCB  -76,46,-1,-53,95,-115,102,-61,0,62,-110,127
05CF& 8D 66 C3 00 3E
05D4& 92 7F
05D6& 92 3E 00 C3 66      815                  FCB  -110,62,0,-61,102,-115,95,-53,-1,46,-76
05DB& 8D 5F CB FF 2E
05E0& B4
                          816 ;
                          817 ; PARAMETERS FOR CORRELATION MATCHING ALGORITHM STORED HERE
```

```
                    818 ;FOR ANIMAL CALL.TYPE1
```



```
                    819 ;WOLF HOWL COMPOSITE TEMPLATE
05E1& 7E 74 55 1E 15    820 MATCHEPOCH1T1    FCB  126,116,85,30,21,18,17,7,12,9
05E6& 12 11 07 0C 09
05EB& 84 79 53 28 14    821 MATCHEPOCH2T1    FCB  132,121,83,40,20,18,9,8,12,10
05F0& 12 09 08 0C 0A
05F5& 71 80 5C 22 16    822 MATCHEPOCH3T1    FCB  113,128,92,34,22,18,11,11,13,9
05FA& 12 0B 0B 0D 09
05FF& 77 77 4B 21 14    823 MATCHEPOCH4T1    FCB  119,119,75,33,20,17,10,8,13,9
0604& 11 0A 08 0D 09
0609& 7B 6D 48 1E 12    824 MATCHEPOCH5T1    FCB  123,109,72,30,18,17,10,6,12,9
060E& 11 0A 06 0C 09
0613& 82 6D 42 1A 14    825 MATCHEPOCH6T1    FCB  130,109,66,26,20,18,14,7,13,9
0618& 12 0E 07 0D 09
061D& 64 60 45 1A 13    826 MATCHEPOCH7T1    FCB  100,96,69,26,19,18,8,6,12,9
0622& 12 08 06 0C 09
0627& 55 56 3F 19 12    827 MATCHEPOCH8T1    FCB  85,86,63,25,18,16,6,6,12,7
062C& 10 06 06 0C 07
0631& 4A 3F 2B 18 11    828 MATCHEPOCH9T1    FCB  74,63,43,24,17,15,6,5,12,5
0636& 0F 06 05 0C 05
063B& 53 3E 31 1B 11    829 MATCHEPOCH10T1   FCB  83,62,49,27,17,15,5,4,12,6
0640& 0F 05 04 0C 06
                        830 ;FOR ANIMAL CALL TYPE2
                        831 ;WOLF GROWL COMPOSITE TEMPLATE
0645& 6C 5C 36 1C 1B    832 MATCHEPOCH1T2    FCB  108,92,54,28,27,25,29,23,18,9
064A& 19 1D 17 12 09
064F& 74 5E 44 22 1E    833 MATCHEPOCH2T2    FCB  116,94,68,34,30,29,36,25,20,12
0654& 1D 24 19 14 0C
0659& 7F 5F 39 1F 1D    834 MATCHEPOCH3T2    FCB  127,95,57,31,29,23,20,14,17,10
065E& 17 14 0E 11 0A
0663& 68 53 31 1C 1B    835 MATCHEPOCH4T2    FCB  104,83,49,28,27,24,27,17,17,10
0668& 18 1B 11 11 0A
066D& 62 54 39 1D 19    836 MATCHEPOCH5T2    FCB  98,84,57,29,25,24,23,20,19,10
0672& 18 17 14 13 0A
0677& 66 48 2F 1C 1A    837 MATCHEPOCH6T2    FCB  102,72,47,28,26,21,15,12,16,8
067C& 15 0F 0C 10 08
0681& 5A 33 26 1A 16    838 MATCHEPOCH7T2    FCB  90,51,38,26,22,19,15,10,16,8
0686& 13 0F 0A 10 08
068B& 53 5B 37 20 1A    839 MATCHEPOCH8T2    FCB  83,91,55,32,26,21,19,13,18,13
```

```
0690& 15 13 0D 12 0D
```



```
0695& 7E 50 2F 1D 1C       840 MATCHEPOCH9T2    FCB  126,80,47,29,28,20,19,14,16,9
069A& 14 13 0E 10 09
069F& 65 4A 34 1D 19       841 MATCHEPOCH10T2   FCB  101,74,52,29,25,23,21,18,17,9
06A4& 17 15 12 11 09
                           842 ;FOR ANIMAL CALL TYPE3
                           843 ;WOLF BARK COMPOSITE TEMPLATE
06A9& 3E 3B 3D 30 3E       844 MATCHEPOCH1T3    FCB  62,59,61,48,62,40,49,37,20,13
06AE& 28 31 25 14 0D
06B3& 3F 3F 30 28 2E       845 MATCHEPOCH2T3    FCB  63,63,48,40,46,29,42,30,17,10
06B8& 1D 2A 1E 11 0A
06BD& 3A 3B 33 2F 33       846 MATCHEPOCH3T3    FCB  58,59,51,47,51,30,40,26,16,11
06C2& 1E 28 1A 10 0B
06C7& 41 4E 33 36 30       847 MATCHEPOCH4T3    FCB  65,78,51,54,48,30,43,28,18,12
06CC& 1E 2B 1C 12 0C
06D1& 35 42 2C 38 2B       848 MATCHEPOCH5T3    FCB  53,66,44,56,43,24,27,19,13,8
06D6& 18 1B 13 0D 08
06DB& 30 36 32 31 32       849 MATCHEPOCH6T3    FCB  48,54,50,49,50,27,31,23,15,10
06E0& 1B 1F 17 0F 0A
06E5& 30 39 24 24 23       850 MATCHEPOCH7T3    FCB  48,57,36,36,35,23,27,20,14,8
06EA& 17 1B 14 0E 08
06EF& 32 3E 34 2E 30       851 MATCHEPOCH8T3    FCB  50,62,52,46,48,31,37,32,18,11
06F4& 1F 25 20 12 0B
06F9& 33 49 2C 26 25       852 MATCHEPOCH9T3    FCB  51,73,44,38,37,24,28,22,14,8
06FE& 18 1C 16 0E 08
0703& 43 5E 33 2C 2A       853 MATCHEPOCH10T3   FCB  67,94,51,44,42,28,33,26,13,10
0708& 1C 21 1A 0D 0A
                           854 ;FOR ANIMAL CALL TYPE 4
                           855 ;WOLF WHINE COMPOSITE TEMPLATE
070D& 46 66 24 29 1A       856 MATCHEPOCH1T4    FCB  70,102,36,41,26,39,38,14,16,10
0712& 27 26 0E 10 0A
0717& 4A 6B 22 2B 1E       857 MATCHEPOCH2T4    FCB  74,107,34,43,30,41,63,129,22,13
071C& 29 3F 81 16 0D
0721& 40 1E 14 15 10       858 MATCHEPOCH3T4    FCB  64,30,20,21,16,15,8,7,13,6
0726& 0F 08 07 0D 06
072B& 41 37 1B 18 13       859 MATCHEPOCH4T4    FCB  65,55,27,24,19,21,17,9,15,7
0730& 15 11 09 0F 07
0735& 5B 5E 2D 20 1C       860 MATCHEPOCH5T4    FCB  91,94,45,32,28,40,48,23,17,11
073A& 28 30 17 11 0B
```

```
073F& AB A6 4E 2F 37      861 MATCHEPOCH6T4   FCB  171,166,78,47,55,54,54,56,27,22
0744& 36 36 38 1B 16
```



```
0749& 46 2A 1A 22 1D      862 MATCHEPOCH7T4   FCB  70,42,26,34,29,25,26,17,19,11
074E& 19 1A 11 13 0B
0753& 41 21 18 16 11      863 MATCHEPOCH8T4   FCB  65,33,24,22,17,15,10,8,13,8
0758& 0F 0A 08 0D 08
075D& 34 1B 16 17 12      864 MATCHEPOCH9T4   FCB  52,27,22,23,18,36,16,17,30,21
0762& 24 10 11 1E 15
0767& 4C 26 15 16 14      865 MATCHEPOCH10T4  FCB  76,38,21,22,20,19,15,15,19,23
076C& 13 0F 0F 13 17
0771& 0B B8               866 THRESHOLD       FCB  0BH,0B8H                        ;THRESHOLD=3000
                          867 ;TOP OF EEPROM IS 07BFH=1983 BYTES
                          868 ;-----------------------------------------------------------------------------
                          869                     END
```

Enclosure 3
BASIC Software for PC

```
REM                          BIRDBORN PC TEST PROGRAM
REM                               WOLFTEST.BAS
REM                          TESTS DAHLGREN INTERFACE
REM
REM                       WRITTEN BY: PROTAGORAS CUTCHIS
REM
REM                    VERSION 1.2 - WEDNESDAY-AUGUST 26, 1998
REM                               TIME 3:40 PM
REM
REDU: DIM DIGITS%(20)
    ERRORS = 0
      OPEN "COM1:9600,N,8,1" FOR RANDOM AS #1 LEN = 256
    REM CONFIGURE RS232 PORT
    GOSUB INITDATE
    REM DEFINE FUNCTION KEYS ...
    KEY ON
    KEY 10, "QUIT"
     KEY(10) ON
     ON KEY(10) GOSUB SEEYA
HELLO: REM PRINT HELLO SCREEN ...
     CLS
     PRINT "PRESS 'F10' KEY AT ANY TIME TO QUIT THE PROGRAM "
     PRINT "      IF THE PROGRAM IS WAITING FOR YOU TO ENTER A NUMBER";
     PRINT " AND YOU WANT TO QUIT"
     PRINT "      PRESS ENTER AFTER PRESSING THE 'F10' KEY"
     LOCATE 7, 1
     GOTO TAGSTUF
STALL: REM CAUSES PROGRAMS TO WAIT 'TIMEDELAY' SECONDS TO ALLOW RECEIVER
       REM OR THE RS232 INTERFACE TO COMPLETE AN ACTION
       FOR XX = 1 TO 100
       FOR YY = 1 TO 20
       NEXT YY
       NEXT XX
            ' ADDS SMALL DELAY BEFORE CONTINUING TO ENSURE CURRENTTIMER
            ' IS GREATER THAN NOW
       RETURN

WEIGHT14: CURRENTTIMER = TIMER
          IF CURRENTTIMER > 86399.5 THEN GOTO WEIGHT14
          IF CURRENTTIMER < NOW THEN
               REM THIS IS TRUE ONLY WHEN TIMER RESETS TO 0 AT MIDNIGHT
               NOW = 0
          END IF
       IF CURRENTTIMER - NOW < TIMEDELAY THEN GOTO WEIGHT14
RETURN
```

```
REM @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
INITDATE: REM CONVERTS DATE AND TIME TO FILE NAME
        '
        ' FILE NAME  =   DDD_HHMM.DAT      WHERE
        ' DDD = DAY OF THE YEAR (1-365). AND HHMM = HOURS-MINUTES
          DAZE$ = DATE$
          DAY = VAL(MID$(DAZE$, 4, 2))
        MONTH = VAL(MID$(DAZE$, 1, 2))
        YEARS = VAL(MID$(DAZE$, 7, 4))
        IF MONTH = 1 THEN DAYNUM = 0
        IF MONTH = 2 THEN DAYNUM = 31
        IF MONTH = 3 THEN DAYNUM = 59
        IF MONTH = 4 THEN DAYNUM = 90
        IF MONTH = 5 THEN DAYNUM = 120
        IF MONTH = 6 THEN DAYNUM = 151
        IF MONTH = 7 THEN DAYNUM = 181
        IF MONTH = 8 THEN DAYNUM = 212
        IF MONTH = 9 THEN DAYNUM = 243
        IF MONTH = 10 THEN DAYNUM = 273
        IF MONTH = 11 THEN DAYNUM = 304
        IF MONTH = 12 THEN DAYNUM = 334
        IF YEARS MOD 4 = 0 AND DAYNUM >= 59 THEN DAYNUM = DAYNUM + 1
        DAYNUM = DAYNUM + DAY
        DAYNUMS$ = MID$(STR$(DAYNUM), 2, 3)
        HOUR$ = MID$(TIME$, 1, 2)
        MINS$ = MID$(TIME$, 4, 2)
        FILENAME$ = DAYNUMS$ + "_" + HOUR$ + MINS$
RETURN
REM @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
TAGSTUF:    PRINT "IT IS RECOMMENDED THAT THE SYSTEM ID BE CHECKED"
            PRINT "TO CONFIRM SYSTEM COMMUNICATIONS"
            PRINT "DO YOU WISH TO CONFIRM SYSTEM ID?"
            PRINT "ENTER Y/N"
TRYAGAIN1: TAGVAR$ = INKEY$
            IF TAGVAR$ = "Y" OR TAGVAR$ = "y" THEN GOTO GETID
            IF TAGVAR$ = "N" OR TAGVAR$ = "n" THEN CLS : GOTO NEXTTEST1
            GOTO TRYAGAIN1
GETID:      GOSUB GETID2
            GOTO NEXTTEST1
GETID2:     INNY$ = INPUT$(LOC(1), #1)
            EFLINE$ = "G"
            PRINT #1, EFLINE$
            GOSUB STALL
            RS232 = 99
            INNY$ = INPUT$(LOC(1), #1)
            RS232 = 11
```

```
           PRINT INNY$
           RETURN
NEXTTEST1: NORMFLAG = 0
           PRINT "TEST OPTIONS ARE:"
           PRINT "  1) DUMP ALL FILTER WEIGHTS"
           PRINT "  2) DO A SINGLE TIME EPOCH FREQUENCY TEST"
           PRINT "  3) DO A MULTI-EPOCH FREQUENCY TEST"
           PRINT "  4) RECONFIRM UNIT ID CODE"
           PRINT "  5) TEST SIGNED MULTIPLY ROUTINE"
           PRINT "  6) TEST DIGITAL MEGA ADDITION ROUTINE"
           PRINT "  7) TEST A MULTI-TIME EPOCH MULTI-FREQUENCY FILTER RESPONSE"
           PRINT "  8) TEST THE NORMALIZATION ROUTINE"
           PRINT "  9) TEST DIGITAL FILTERS AND NORMALIZATION WITH ACTUAL RECORDED DATA"
           PRINT "  A) ANALYZE A RECORDING AND DUMP NORMALIZED PARAMETERS"
           PRINT "  B) DO A MATCH AGAINST A RECORDED REFERENCE"
           PRINT "  C) MATCH 2 STORED DATA FILES"
           PRINT "  D) TEST INTERFACE WITH DAHLGREN GPS/ARGOS UNITS"
           PRINT "  E) TEST DAHLGREN INTERFACE AND TRANSFER 8 BYTES"
           PRINT "ENTER 1-9 OR A-E"
TRYAGAIN2: TAGVAR$ = INKEY$
           IF TAGVAR$ = "1" THEN CLS : GOTO DUMPWGHT
           IF TAGVAR$ = "2" THEN CLS : GOTO FILTTEST
           IF TAGVAR$ = "3" THEN CLS : GOTO WATERFALL
           IF TAGVAR$ = "4" THEN CLS : GOSUB GETID2: GOTO NEXTTEST1
           IF TAGVAR$ = "5" THEN CLS : GOTO MULTIPLY
           IF TAGVAR$ = "6" THEN CLS : GOTO MEGATEST
           IF TAGVAR$ = "7" THEN CLS : GOTO MULTTEST
           IF TAGVAR$ = "8" THEN CLS : GOTO NORMAL
           IF TAGVAR$ = "9" THEN CLS : GOTO AUDIO
           IF TAGVAR$ = "A" OR TAGVAR$ = "a" THEN CLS : GOTO ANALYZE1
           IF TAGVAR$ = "B" OR TAGVAR$ = "b" THEN CLS : GOTO CAPTURE
           IF TAGVAR$ = "C" OR TAGVAR$ = "c" THEN GOTO MATCH1
           IF TAGVAR$ = "D" OR TAGVAR$ = "d" THEN CLS : DAHLFLAG = 0: GOTO DAHLGREN
           IF TAGVAR$ = "E" OR TAGVAR$ = "e" THEN CLS : DAHLFLAG = 1: GOTO DAHLGREN
           GOTO TRYAGAIN2
MATCH1:    PRINT "ENTER FIRST FILE NAME."
           INPUT FILE1$
           PRINT "ENTER SECOND FILE NAME."
           INPUT FILE2$
           OPEN FILE1$ FOR INPUT AS #3
           FOR L = 1 TO 100
           INPUT #3, A(L)
           NEXT L
           CLOSE #3
           OPEN FILE2$ FOR INPUT AS #3
           FOR L = 1 TO 100
```

```
            INPUT #3, B(L)
            NEXT L
            CLOSE #3
            DIFF1 = 0: DIFF2 = 0
            FOR L = 1 TO 100
            DIFF1 = DIFF1 + ABS(A(L) - B(L))
            NEXT L
            FOR L = 1 TO 9
            FOR K = 0 TO 9
            INDEX = L + (K * 10)
            DIFF2 = DIFF2 + ABS(A(INDEX) - B(INDEX))
            NEXT K
            NEXT L
            PRINT "MATCH DIFFERENCE BASED ON STRAIGHT"
            PRINT "100 PARAMETER TIME/FREQ DISTRIBUTION IS: "; DIFF1
            PRINT " "
            PRINT "MATCH DIFFERENCE BASED ON 20 PARAMETER FREQ/VARIANCE"
            PRINT 'DISTRIBUTION IS: ";DIFF2
TA911A:     TAGVAR$ = INKEY$
            IF TAGVAR$ = " " THEN GOTO NEXTTEST1
            GOTO TA911A
ANALYZE1:   NORMFLAG = 1
            PRINT "START CAPTURE"
            PRINT "START TIME IS: "; TIME$: TSTRT = 3600 * VAL(LEFT$(TIME$, 2)) + 60 * VAL(MID$(TIME$, 4, 2)) +
VAL(RIGHT$(TIME$, 2))
            GOTO GETFILT8
            PRINT "WILL NOW WAIT FOR 12 MINUTES AND QUERY"
CHCKTIME1:  IF 3600 * VAL(LEFT$(TIME$, 2)) + 60 * VAL(MID$(TIME$, 4, 2)) + VAL(RIGHT$(TIME$, 2)) >= TSTRT + 800 THEN GOTO
NORMAL
            GOTO CHCKTIME1
DAHLGREN:   PRINT "HIT Y KEY AND ENTER WHEN YELLOW LIGHT STAYS LIT"
            PRINT "THIS WILL BE APPROXIMATELY 15 MINUTES AFTER CAPTURE"
GETY1:      TAGVAR$ = INKEY$
            IF TAGVAR$ = "Y" OR TAGVAR$ = "y" THEN GOTO DAHL2
            GOTO GETY1
DAHL2:      PRINT " "
            PRINT " "
            PRINT "HIT MOMENTARY PUSHBUTTON SWITCH ON PROTOTYPE"
            PRINT "AND HOLD DOWN WHILE HITTING X AND ENTER."
GETX1:      TAGVAR$ = INKEY$
            IF TAGVAR$ = "X" OR TAGVAR$ = "x" THEN GOTO DAHL3
            FOR L = 1 TO 200
            NEXT L
            GOTO GETX1
DAHL3:      PRINT "OK"
            INNY$ = INPUT$(LOC(1), #1)
```

```
            EFLINE$ = "S"
            PRINT #1, EFLINE$
            RS232 = 99
            FOR L = 1 TO 200
            NEXT L
            INNY$ = INPUT$(LOC(1), #1)
            RS232 = 11
            EFLINE$ = "R"
            PRINT #1, EFLINE$
            RS232 = 99
            FOR L = 1 TO 500
            NEXT L
            INNY2$ = " "
            INNY2$ = INPUT$(LOC(1), #1)
            PRINT "ON SEND REQUEST:   DATA=: ";
            CHAR$ = LEFT$(INNY$, 1)
            GOSUB ASCIIHEX
            PRINT A$; " ";
            CHAR$ = MID$(INNY$, 2, 1)
            GOSUB ASCIIHEX
            PRINT A$; " ";
            CHAR$ = MID$(INNY$, 3, 1)
            GOSUB ASCIIHEX
            PRINT A$; " ";
            IF DAHLFLAG = 0 THEN GOTO RIGHT1
            CHAR$ = MID$(INNY$, 4, 1)
            GOSUB ASCIIHEX
            PRINT A$; " ";
            CHAR$ = MID$(INNY$, 5, 1)
            GOSUB ASCIIHEX
            PRINT A$; " ";
            CHAR$ = MID$(INNY$, 6, 1)
            GOSUB ASCIIHEX
            PRINT A$; " ";
            CHAR$ = MID$(INNY$, 7, 1)
            GOSUB ASCIIHEX
            PRINT A$; " ";
RIGHT1:     CHAR$ = RIGHT$(INNY$, 1)
            GOSUB ASCIIHEX
            PRINT A$; " ";
            PRINT "  BYTES="; LEN(INNY$)
RETRAN1:    PRINT "ON RESEND REQUEST: DATA=: ";
            CHAR$ = LEFT$(INNY2$, 1)
            GOSUB ASCIIHEX
            PRINT A$; " ";
            CHAR$ = MID$(INNY2$, 2, 1)
```

```
            GOSUB ASCIIHEX
            PRINT A$; " ";
            CHAR$ = MID$(INNY2$, 3, 1)
            GOSUB ASCIIHEX
            PRINT A$; " ";
            IF DAHLFLAG = 0 THEN GOTO RIGHT2
            CHAR$ = MID$(INNY2$, 4, 1)
            GOSUB ASCIIHEX
            PRINT A$; " ";
            CHAR$ = MID$(INNY2$, 5, 1)
            GOSUB ASCIIHEX
            PRINT A$; " ";
            CHAR$ = MID$(INNY2$, 6, 1)
            GOSUB ASCIIHEX
            PRINT A$; " ";
            CHAR$ = MID$(INNY2$, 7, 1)
            GOSUB ASCIIHEX
            PRINT A$; " ";
RIGHT2:     CHAR$ = RIGHT$(INNY2$, 1)
            GOSUB ASCIIHEX
            PRINT A$; " ";
            PRINT "  BYTES="; LEN(INNY2$); ""
            GOTO DAHLGREN
CAPTURE:    PRINT "FIRST RECORD A REFERENCE ANIMAL CALL, THEN"
            PRINT "RECORD A TEST CALL.  HAVE THESE TWO THINGS BEEN"
            PRINT "BEEN COMPLETED? (Y/N)"
IN95:       TAGVAR$ = INKEY$
            IF TAGVAR$ = "N" OR TAGVAR$ = "n" THEN PRINT "THEN DO THEM NOW": PRINT " ": GOTO CAPTURE
            IF TAGVAR$ = "Y" OR TAGVAR$ = "y" THEN GOTO CAPT2
            GOTO IN95
CAPT2:      PRINT "START TIME IS: "; TIME$
            INNY$ = INPUT$(LOC(1), #1)
            PRINT "HOST PC NOW AWAITING ANALYSIS COMPLETION"
            PRINT "ESTIMATED TIME IS 17 MINUTES FROM RECORDING"
            PRINT "OF TEST CALL"
            TEMP1$ = ""
GETMORE95:  INNY$ = INPUT$(LOC(1), #1)
            IF INNY$ <> "" AND LEN(INNY$) > 4 THEN PRINT "ANALYSIS COMPLETED, WILL NOW REQUEST RESULTS": GOTO GETEM1
            IF INNY$ <> "" THEN TEMP1$ = TEMP1$ + INNY$
            IF LEN(TEMP1$) >= 6 THEN INNY$ = TEMP1$: GOTO GETEM1
            GOSUB STALL
            GOTO GETMORE95
GETEM1:     PRINT CHR$(7): GOSUB STALL: PRINT CHR$(7): PRINT INNY$
            GOSUB EVAL2
            PRINT "MATCH PARAMETER IS: "; CALCADD
            PRINT " ": PRINT "HIT THE SPACE BAR TO CONTINUE"
```

```
INSPACE1:  TAGVAR$ = INKEY$
           IF TAGVAR$ = " " THEN GOTO NEXTTEST1
           GOTO INSPACE1
NORMAL:    PRINT "YOU MUST PERFORM A FULL MULTI-FREQUENCY"
           PRINT "MULTI-EPOCH TEST FIRST.  HAS THIS BEEN DONE YET? (Y/N)"
TA1:       TAGVAR$ = INKEY$
           IF TAGVAR$ = "Y" OR TAGVAR$ = "y" THEN GOTO NORM2
           IF TAGVAR$ = "N" OR TAGVAR$ = "n" THEN GOTO NEXTTEST1
           GOTO TA1
NORM2:     PRINT "DO YOU WISH TO WRITE THE DATA TO DISK?  (Y/N)"
TA911:     TAGVAR$ = INKEY$
           IF TAGVAR$ = "Y" OR TAGVAR$ = "y" THEN WRITFLAG = 1: GOTO NORM3
           IF TAGVAR$ = "N" OR TAGVAR$ = "n" THEN WRITFLAG = 0: GOTO NORM3
           GOTO TA911
NORM3:     IF WRITFLAG = 1 THEN GOSUB INITDATE: OPEN FILENAME$ + ".DAT" FOR OUTPUT AS #3
           SUM = 0
           MIN = 256: MAX = 0
           FOR L = 1 TO 101
           EFLINE$ = "N"
           INNY$ = INPUT$(LOC(1), #1)
           PRINT #1, EFLINE$;
WAIT103:   FOR J = 1 TO 200
           NEXT J
           INNY$ = INPUT$(LOC(1), #1)
           D = VAL(INNY$)
           IF D < 0 THEN D = D + 256
           IF L = 1 THEN PRINT "AVERAGE CALCULATED IN uP IS: "; D: GOTO CCB
           IF INT((L - 1) / 10) = (L - 1) / 10 THEN PRINT D
           IF WRITFLAG = 1 THEN WRITE #3, D
           IF INT((L - 1) / 10) <> (L - 1) / 10 THEN PRINT D;
           IF D < MIN THEN MIN = D
           IF D > MAX THEN MAX = D
           SUM = SUM + D
CCB:       NEXT L
           PRINT "SUM= "; SUM; " AVERAGE = "; SUM / 100
           PRINT "MAX= "; MAX; "   MIN= "; MIN; "  MAX:AVE= "; MAX / (SUM / 100)
           CLOSE #3
TA2:       TAGVAR$ = INKEY$
           IF TAGVAR$ = " " THEN GOTO NEXTTEST1
           GOTO TA2
MULTTEST:  CLS
           PRINT "TRANSFER FREQUENCY PATTERN TO DACIS PROCESSOR"
ENTAMP2:   PRINT "ENTER PEAK AMPLITUDE:": INPUT AMP
           IF AMP > 127 THEN PRINT "AMPLITUDE TOO LARGE, TRY AGAIN": GOTO ENTAMP2
           PRINT "ENTER START FREQUENCY": INPUT STARTFREQ
INPEND:    PRINT "ENTER END FREQ": INPUT ENDFREQ
```

```
            FOR L = 1 TO 3072
            PRINT L
            A = INT(AMP * SIN(2 * 3.14159 * FREQ * (L / 6000)))
            IF A < 0 THEN A = A + 256
            PRINT #1, CHR$(A);
            FOR K = 1 TO 3
            NEXT K
            NEXT L
            PRINT "DOWNLOAD FINISHED, WILL NOW CHECK FOR RECEPTION"
            EFLINE$ = "C"
            PRINT #1, EFLINE$;
            GOSUB STALL
            RS232 = 99
            GOTO STRTHERE:
AUDIO:      PRINT "DO YOU WANT TO PRINT PARAMETERS? (Y/N)";
TRY822:     TAGVAR$ = INKEY$
            IF TAGVAR$ = "Y" OR TAGVAR$ = "y" THEN PRNTFLAG = 1: PRINT "YES": LPRINT DATE$: GOTO WRITEFILE
            IF TAGVAR$ = "N" OR TAGVAR$ = "n" THEN PRNTFLAG = 0: PRINT "NO": GOTO WRITEFILE
            GOTO TRY822
WRITEFILE:  PRINT "DO YOU WANT TO WRITE DATA TO DISK? (Y/N)";
TRY888:     TAGVAR$ = INKEY$
            IF TAGVAR$ = "Y" OR TAGVAR$ = "y" THEN DISKFLAG = 1: PRINT "YES": GOTO AUDICONT
            IF TAGVAR$ = "N" OR TAGVAR$ = "n" THEN DISKFLAG = 0: PRINT "NO": GOTO AUDICONT
            GOTO TRY888
AUDICONT:   ANALNUM = 1
            PRINT "DO YOU WANT TO DO REPETITIVE AUTOMATIC ANALYSES? (Y/N)";
TRY420:     TAGVAR$ = INKEY$
            IF TAGVAR$ = "Y" OR TAGVAR$ = "y" THEN AUTOFLAG = 1: PRINT "YES": GOTO STARTAUD
            IF TAGVAR$ = "N" OR TAGVAR$ = "y" THEN AUTOFLAG = 0: PRINT "NO": GOTO STARTAUD
            GOTO TRY420
STARTAUD:   IF AUTOFLAG = 1 THEN GOTO GETFILT8
            PRINT "HAS AN AUDIO CAPTURE BEEN PERFORMED? (Y/N)"
IN829:      TAGVAR$ = INKEY$
            IF TAGVAR$ = "Y" OR TAGVAR$ = "y" THEN GOTO GETFILT8
            IF TAGVAR$ = "N" OR TAGVAR$ = "n" THEN PRINT "THEN DO ONE NOW": GOTO NEXTTEST1
            GOTO IN829
MULTRECV:   PRINT "DOWNLOAD FINISHED, WILL NOW CHECK FOR RECEPTION"
            EFLINE$ = "C"
            PRINT #1, EFLINE$;
            GOSUB STALL
            RS232 = 99
            INNY$ = INPUT$(LOC(1), #1)
            RS232 = 11
            IF INNY$ = "X" THEN PRINT "DOWNLOAD SUCCESSFUL": GOTO GETFILT8
            IF INNY$ <> "X" THEN PRINT "DOWNLOAD WAS NOT SUCCESSFUL"
GETFILT8:   DCNT = 1
```

```
            IF PRNTFLAG = 1 THEN LPRINT "FREQ       "; "250  500  750  1000 1250 1500 1750 2000 2250 2500"
            FOR JJ = 1 TO 10
            SUMFILT(JJ) = 0
            NEXT JJ
            GETTIMFLG = 1
            IF NORMFLAG = 1 THEN GOTO GETFILT9
GETFILT6:   GOTO GETFILT9
            EFLINE$ = "D"
            RS232 = 99
            PRINT #1, EFLINE$
            RS232 = 11
            FOR J = 1 TO 2000
            FOR K = 1 TO 10
            NEXT K
            NEXT J
            SUM1 = 0
GETFILT9:   INNY$ = ""
INNY$ = INPUT$(LOC(1), #1)
            IF LEN(INNY$) >= 12 AND RIGHT$(INNY$, 1) = " " THEN GOTO GOTSAMP
            IF LEN(INNY$) <> 0 THEN INFIRST$ = INFIRST$ + INNY$
            IF LEN(INFIRST$) >= 12 THEN INNY$ = INFIRST$: INFIRST$ = "": GOTO GOTSAMP
            FOR R = 1 TO 500
            NEXT R
            GOTO GETFILT9
GOTSAMP:    IF GETTIMFLG = 1 AND DISKFLAG = 1 THEN GOSUB INITDATE
            IF GETTIMFLG = 1 THEN CLS : PRINT "DATE= "; DATE$; "  START="; TIME$; "  ANALYSIS NUMBER= "; ANALNUM: ANALNUM
= ANALNUM + 1
            IF GETTIMFLG = 1 THEN PRINT "FREQ       "; "250  500  750  1000 1250 1500 1750 2000 2250 2500"
            IF GETTIMFLG = 1 AND DISKFLAG = 1 THEN OPEN FILENAME$ + ".DAT" FOR OUTPUT AS #3
            IF GETTIMFLG = 1 AND DISKFLAG = 1 THEN PRINT #3, "DATE="; DATE$; "   TIME="; TIME$
            IF DISKFLAG = 1 AND GETTIMFLG = 1 THEN PRINT #3, "FREQ       "; "250  500  750  1000 1250 1500 1750 2000 2250
2500": GETTIMFLG = 0
            GOSUB EVAL1
            INDEX1 = DCNT - 10 * INT(DCNT / 10)
            N = INT((DCNT - 1) / 10) + 1: M = DCNT - 10 * (N - 1)
            IF INT((DCNT - 1) / 10) = (DCNT - 1) / 10 THEN PRINT "EPOCH"; (DCNT + 9) / 10; " ";
            IF INT((DCNT - 1) / 10) = (DCNT - 1) / 10 AND PRNTFLAG = 1 THEN LPRINT "EPOCH"; (DCNT + 9) / 10; " ";
            IF INT((DCNT - 1) / 10) = (DCNT - 1) / 10 AND DISKFLAG = 1 THEN PRINT #3, "EPOCH"; (DCNT + 9) / 10; " ";
            IF INT((DCNT - 1) / 10) = (DCNT - 1) / 10 AND DCNT <> 91 THEN PRINT " ";
            A$ = STR$(INT(CALCADD / 100000)): ZZ = INT(CALCADD / 100000)
            SUMFILT(INDEX1) = SUMFILT(INDEX1) + ZZ
            MATRIX(N, M) = ZZ
            SUM1 = SUM1 + ZZ
            GOSUB MAKE4
            GOTO PRINT104
MAKE4:      IF LEN(A$) = 1 THEN A$ = A$ + "    "
```

```
            IF LEN(A$) = 2 THEN A$ = A$ + "   "
            IF LEN(A$) = 3 THEN A$ = A$ + "  "
            IF LEN(A$) = 4 THEN A$ = A$ + " "
            RETURN
PRINT104:   PRINT A$;
            IF PRNTFLAG = 1 THEN LPRINT A$;
            IF DISKFLAG = 1 THEN PRINT #3, A$;
            DCNT = DCNT + 1
            IF INT((DCNT - 1) / 10) = (DCNT - 1) / 10 THEN PRINT " "
            IF INT((DCNT - 1) / 10) = (DCNT - 1) / 10 AND PRNTFLAG = 1 THEN LPRINT " "
            IF INT((DCNT - 1) / 10) = (DCNT - 1) / 10 AND DISKFLAG = 1 THEN PRINT #3, " "
REM           IF DCNT = 101 THEN GOSUB FINDATA
REM           IF DCNT = 101 THEN PRINT "SUM= "; SUM1; "   AVERAGE= "; SUM1 / 100: GOTO CHCKSPACE
            IF DCNT = 101 THEN GOTO CHCKSPACE
            GOTO GETFILT9
CHCKSPACE:  PRINT "NORMALIZED PARAMETERS"
            IF PRNTFLAG = 1 THEN LPRINT "NORMALIZED PARAMETERS"
            IF DISKFLAG = 1 THEN PRINT #3, "NORMALIZED PARAMETERS"
            PRINT "FREQ       "; "250  500  750  1000 1250 1500 1750 2000 2250 2500"
            IF DISKFLAG = 1 THEN PRINT #3, "250  500  750  1000 1250 1500 1750 2000 2250 2500"
            IF PRNTFLAG = 1 THEN LPRINT "FREQ     "; "250  500  750  1000 1250 1500 1750 2000 2250 2500"
            DCNT = 1: SUM = 0
REDO1019:   INFIRST$ = ""
GET1017:    INNY$ = ""
GET1018:    INNY$ = INPUT$(LOC(1), #1)
            IF LEN(INNY$) = 0 THEN GOTO GET1017
            IF RIGHT$(INNY$, 1) <> " " THEN INFIRST$ = INFIRST$ + INNY$: GOTO GET1018
            INFIRST$ = INFIRST$ + INNY$: A$ = INFIRST$: INFIRST$ = ""
            IF LEFT$(A$, 1) = " " THEN GOTO REDO1019
            IF LEFT$(A$, 1) = "-" THEN A$ = STR$(256 + VAL(A$))
            IF LEFT$(A$, 1) = " " THEN A$ = RIGHT$(A$, LEN(A$) - 1)
            IF LEFT$(A$, 1) = " " THEN A$ = RIGHT$(A$, LEN(A$) - 1)
            SUM = SUM + VAL(A$)
            GOSUB MAKE4
            IF INT((DCNT - 1) / 10) = (DCNT - 1) / 10 THEN PRINT "EPOCH"; (DCNT + 9) / 10; " ";
            IF INT((DCNT - 1) / 10) = (DCNT - 1) / 10 AND PRNTFLAG = 1 THEN LPRINT "EPOCH"; (DCNT + 9) / 10; " ";
            IF INT((DCNT - 1) / 10) = (DCNT - 1) / 10 AND DISKFLAG = 1 THEN PRINT #3, "EPOCH"; (DCNT + 9) / 10; " ";
            IF INT((DCNT - 1) / 10) = (DCNT - 1) / 10 AND DCNT <> 91 THEN PRINT "  ";
            IF INT(DCNT / 10) = DCNT / 10 THEN PRINT A$
            IF INT(DCNT / 10) = DCNT / 10 AND PRNTFLAG = 1 THEN LPRINT A$
            IF INT(DCNT / 10) = DCNT / 10 AND DISKFLAG = 1 THEN PRINT #3, A$
            IF INT(DCNT / 10) <> DCNT / 10 THEN PRINT A$;
            IF INT(DCNT / 10) <> DCNT / 10 AND PRNTFLAG = 1 THEN LPRINT A$;
            IF INT(DCNT / 10) <> DCNT / 10 AND DISKFLAG = 1 THEN PRINT #3, A$;
            DCNT = DCNT + 1
            IF DCNT = 101 THEN GOTO IN1017
```

```
            GOTO GET1017
IN1017:     PRINT "AVERAGE IS:"; SUM / 100;
            IF PRNTFLAG = 1 THEN LPRINT "AVERAGE IS:"; SUM / 100
            IF DISKFLAG = 1 THEN PRINT #3, "AVERAGE IS:"; SUM / 100
            CLOSE #3
            IF AUTOFLAG = 1 THEN GOTO GETFILT8
IN10172:    TAGVAR$ = INKEY$
            IF TAGVAR$ = " " THEN GOTO NEXTTEST1
            GOTO IN10172
STRTHERE:   INNY$ = INPUT$(LOC(1), #1)
            IF INNY$ = "X" THEN PRINT "DOWNLOAD SUCCESSFUL"
            IF INNY$ <> "X" THEN PRINT "DOWNLOAD NOT SUCCESSFUL": PRINT INNY$, LEN(INNY$)
NEXTFREQ:   PRINT "AMPLITUDE= "; AMP; "      INPUT FREQUENCY= "; FREQ
            IF WRITDISK = 1 THEN PRINT #2, "AMPLITUDE= "; AMP; "     INPUT FREQUENCY="; FREQ
            RS232 = 11
            EFLINE$ = "D": SECONDS = 3600 * VAL(LEFT$(TIME$, 2)) + 60 * VAL(MID$(TIME$, 4, 2)) + VAL(RIGHT$(TIME$, 2))
            PRINT #1, EFLINE$;
            FOR J = 1 TO 3000
            FOR K = 1 TO 10
            NEXT K
            NEXT J
            RS232 = 99
GETFILT:    FREQREC = 0
GETFILT2:   INNY$ = ""
GETFILT3:   INNY$ = INPUT$(LOC(1), #1)
   '12=MINIMUM TRANSMISSION PER FILTER OUTPUT
            IF LEN(INNY$) >= 12 AND RIGHT$(INNY$, 1) = " " THEN GOTO PRINTDATA
            IF LEN(INNY$) <> 0 THEN INFIRST$ = INFIRST$ + INNY$
            IF LEN(INFIRST$) >= 12 THEN INNY$ = INFIRST$: INFIRST$ = "": GOTO PRINTDATA
            FOR L = 1 TO 500
            NEXT L
            GOTO GETFILT2
PRINTDATA:  GOSUB EVAL1
            FILTTIME = 3600 * VAL(LEFT$(TIME$, 2)) + 60 * VAL(MID$(TIME$, 4, 2)) + VAL(RIGHT$(TIME$, 2)) - SECONDS
            PRINT "TIME TO COMPLETE FILTER= "; INT(FILTTIME / 60); ":"; FILTTIME - 60 * INT(FILTTIME / 60); "    ";
            IF WRITDISK = 1 THEN PRINT #2, "TIME TO COMPLETE FILTER= "; INT(FILTTIME / 60); ":"; FILTTIME - 60 *
INT(FILTTIME / 60); "    ";
            FREQREC = FREQREC + 1: ACTFREQ = 250 * FREQREC
            PRINT "FREQ="; ACTFREQ; "Hz  OUTPUT= "; CALCADD
            IF WRITDISK = 1 THEN PRINT #2, "FREQ= "; ACTFREQ; " Hz  OUTPUT= "; CALCADD
WAIT1:      IF DOALL = 1 AND FREQREC = 10 THEN GOTO CHECKALL
            IF FREQREC = 10 AND INKEY$ = " " THEN CLOSE #2: GOTO NEXTTEST1
            IF FREQREC <> 10 THEN GOTO GETFILT2
            GOTO WAIT1
CHECKALL:   IF FREQ = 2500 THEN GOTO NEXTTEST1
            PRINT #2, "      "
```

```
            FREQ = FREQ + 250: GOTO DUMPSINE
MEGATEST:   EFLINE$ = "B"
            CLS
MEGATEST2:  PRINT "THIS TEST WILL SUM ALL INTEGERS BETWEEN THE START"
            PRINT " AND ENDING VALUES AND RETURN THE RESULT"
            PRINT "ENTER START VALUE:": INPUT S1$: S1 = VAL(S1$)
            PRINT "ENTER END VALUE:": INPUT S2$: S2 = VAL(S2$)
            RESULT = ((S1 + S2) * (ABS(S2 - S1) + 1)) / 2
            IF S1 < S2 THEN LS1 = S1
            IF S1 < S2 THEN GS1 = S2
            IF S2 < S1 THEN LS1 = S2
            IF S2 < S1 THEN GS1 = S1
            IF LS1 >= 0 THEN LESSER1 = INT(LS1 / 256)
            IF LS1 >= 0 THEN LESSER2 = LS1 - 256 * LESSER1
            IF GS1 >= 0 THEN GREATER1 = INT(GS1 / 256)
            IF GS1 >= 0 THEN GREATER2 = GS1 - 256 * GREATER1
            IF LS1 < 0 THEN LESSER1 = 128 + INT((LS1 + 32768) / 256)
            IF LS1 < 0 THEN LESSER2 = (LS1 + 32768) - 256 * (LESSER1 - 128)
            IF GS1 < 0 THEN GREATER1 = 128 + INT((GS1 + 32768) / 256)
            IF GS1 < 0 THEN GREATER2 = (GS1 + 32768) - 256 * (GREATER1 - 128)
            INNY$ = INPUT$(LOC(1), #1)
            PRINT #1, EFLINE$
            FOR DELAY = 1 TO 400
            NEXT DELAY
            PRINT #1, CHR$(LESSER1)
            PRINT #1, CHR$(LESSER2)
            PRINT #1, CHR$(GREATER1)
            PRINT #1, CHR$(GREATER2)
            GOSUB STALL
            RS232 = 99
            INNY$ = INPUT$(LOC(1), #1)
            PRINT INNY$, LEN(INNY$)
            RS232 = 11
            GOSUB EVAL1
            GOTO DONE820
EVAL2:      FOR L = 1 TO 5
            IF MID$(INNY$, L, 1) = " " THEN LOHIBYTE = VAL(LEFT$(INNY$, L)): GOTO NEXT991
            NEXT L
NEXT991:    LOLOBYTE = VAL(RIGHT$(INNY$, LEN(INNY$) - L))
            A = 256 * LOHIBYTE + LOLOBYTE
            IF LOHIBYTE >= 0 AND LOLOBYTE < 0 THEN A = 256 * LOHIBYTE + (LOLOBYTE - 256)
            IF LOHIBYTE < 0 AND LOLOBYTE < 0 THEN A = 256 * (LOHIBYTE + 256) + (LOLOBYTE + 256)
            IF LOHIBYTE < 0 AND LOLOBYTE >= 0 THEN A = 256 * LOHIBYTE + (LOLOBYTE - 256)
            CALCADD = A
            RETURN
EVAL1:      FOR L = 1 TO 5
```

```
            IF MID$(INNY$, L, 1) = " " THEN LOHIBYTE = VAL(LEFT$(INNY$, L)): GOTO NEXT891
            NEXT L
NEXT891:    IF LEFT$(INNY$, 1) = "0" AND MID$(INNY$, 2, 1) = "0" THEN LOHIBYTE = 0
            FOR K = L + 1 TO L + 5
            IF MID$(INNY$, K, 1) = " " THEN LOLOBYTE = VAL(MID$(INNY$, L + 1, K - L)): GOTO NEXT892
            NEXT K
NEXT892:    IF MID$(INNY$, L + 1, 1) = "0" AND MID$(INNY$, L + 2, 1) = "0" THEN LOLOBYTE = 0
            FOR L = K + 1 TO K + 5
            IF MID$(INNY$, L, 1) = " " THEN HIHIBYTE = VAL(MID$(INNY$, K + 1, L - K)): GOTO NEXT81X
            NEXT L
NEXT81X:    IF MID$(INNY$, K + 1, 1) = "0" AND MID$(INNY$, K + 2, 1) = "0" THEN HIHIBYTE = 0
             HILOBYTE = VAL(RIGHT$(INNY$, LEN(INNY$) - L))
            GOTO NOPRINT
            PRINT INNY$; "  "; LOHIBYTE; "  "; LOLOBYTE; "  "; HIHIBYTE; "  "; HILOBYTE
NOPRINT:    A = 256 * LOHIBYTE + LOLOBYTE
            B = 256 * HIHIBYTE + HILOBYTE
            IF LOHIBYTE >= 0 AND LOLOBYTE < 0 THEN A = LOHIBYTE * 256 + (LOLOBYTE + 256)
            IF HIHIBYTE >= 0 AND HILOBYTE < 0 THEN B = HIHIBYTE * 256 + (HILOBYTE + 256)
            IF LOHIBYTE < 0 AND LOLOBYTE < 0 THEN A = ((LOHIBYTE + 256) * 256) + (LOLOBYTE + 256) - 65536
            IF HIHIBYTE < 0 AND HILOBYTE < 0 THEN B = ((HIHIBYTE + 256) * 256) + (HILOBYTE + 256) - 65536
            IF LOHIBYTE < 0 AND LOLOBYTE >= 0 THEN A = ((LOHIBYTE + 256) * 256) + LOLOBYTE - 65536
            IF HIHIBYTE < 0 AND HILOBYTE >= 0 THEN B = ((HIHIBYTE + 256) * 256) + HILOBYTE - 65536
            CALCADD = B * 32768 + A
            RETURN
DONE820:    PRINT "A="; A; "B="; B
            PRINT "CALCULATED SUM="; RESULT
            PRINT "PROCESSOR SUM="; CALCADD
            GOTO NEXTTEST1
MULTIPLY:   PRINT "ENTER VALUE 1=-128 TO +127"
            INPUT A1
            IF A1 < -128 OR A1 > 127 THEN GOTO MULTIPLY
            X1 = A1
            IF A1 < 0 THEN A1 = A1 + 256
            PRINT A1, STR$(A1), LEN(STR$(A1)), RIGHT$(STR$(A1), 2), VAL(STR$(A1))
A2AGAIN:    PRINT "ENTER SECOND VALUE =-128 TO +127"
            INPUT A2
            IF A2 < -128 OR A2 > 127 THEN GOTO A2AGAIN
            X2 = A2
            IF A2 < 0 THEN A2 = A2 + 256
            INNY$ = INPUT$(LOC(1), #1)
            EFLINE$ = "M"
            PRINT #1, EFLINE$
            FOR DELAY = 1 TO 300
            NEXT DELAY
            PRINT #1, CHR$(A1)
            FOR DELAY = 1 TO 300
```

```
          NEXT DELAY
          PRINT #1, CHR$(A2)
          FOR DELAY = 1 TO 300
          NEXT DELAY
          RS232 = 99
          INNY$ = INPUT$(LOC(1), #1)
          RS232 = 11
          FOR L = 1 TO 4
          IF MID$(INNY$, L, 1) = " " THEN HIGHBYTE = VAL(LEFT$(INNY$, LEN(INNY$) - L))
          NEXT L
          IF LEFT$(INNY$, 1) = "0" AND MID$(INNY$, 2, 1) = "0" THEN HIGHBYTE = 0
          FOR L = 1 TO 8
          IF MID$(INNY$, L, 1) = " " THEN LOWBYTE = VAL(RIGHT$(INNY$, LEN(INNY$) - L)): GOTO PRINTMUL
          NEXT L
PRINTMUL: IF LOWBYTE < 0 THEN LOWBYTE = LOWBYTE + 256
          PRINT INNY$, HIGHBYTE, LOWBYTE
          PRINT "MULTIPLIED RESULT IS:"; HIGHBYTE * 256 + LOWBYTE
          PRINT "EXPECTED RESULT IS: "; X1 * X2
          GOTO NEXTTEST1
DUMPWGHT: INNY$ = INPUT$(LOC(1), #1)
          FOR L = 1 TO 10
          LINER$ = "FILTER"
          IF L < 10 THEN LINER$ = "FILTER "
          PRINT LINER$; L; " ";
          FOR K = 1 TO 23
          EFLINE$ = "F"
          PRINT #1, EFLINE$
          FOR DELAY = 1 TO 300
          NEXT DELAY
          RS232 = 99
          INNY$ = INPUT$(LOC(1), #1)
          RS232 = 11
          IF LEN(INNY$) = 3 THEN INNY$ = " " + INNY$
          IF LEN(INNY$) = 2 THEN INNY$ = "  " + INNY$
          IF LEN(INNY$) = 1 THEN INNY$ = "   " + INNY$
          PRINT INNY$;
          IF K <> 11 THEN PRINT ",";
          NEXT K
          PRINT " "
          NEXT L
          GOTO NEXTTEST1
SPECTRUM: INNY$ = INPUT$(LOC(1), #1)
          PRINT "FREQUENCY     POWER"
          FOR L = 1 TO 10
          FREQ = 250 * L
          PRINT "   "; FREQ; "      ";
```

```
           EFLINE$ = "O"
           PRINT #1, EFLINE$
           FOR DELAY = 1 TO 300
           NEXT DELAY
           RS232 = 99
           INNY$ = INPUT$(LOC(1), #1)
           RS232 = 11
           POWER = VAL(INNY$)
           IF POWER < 0 THEN POWER = POWER + 256
           PRINT POWER
           NEXT L
           GOTO NEXTTEST1
WATERFALL: PRINT "FREQ.  EPOCH1 EPOCH2 EPOCH3 EPOCH4 EPOCH5 EPOCH6 EPOCH7 EPOCH8 EPOCH9 EPOCH10"
           FOR L = 1 TO 10
           FREQ = 250 * L
           A$ = STR$(FREQ)
           IF LEN(A$) < 5 THEN A$ = A$ + " "
           PRINT A$; " ";
           EFLINE$ = "W"
           FOR J = 1 TO 10
           PRINT #1, EFLINE$
           FOR DELAY = 1 TO 300
           NEXT DELAY
           RS232 = 99
           INNY$ = INPUT$(LOC(1), #1)
           RS232 = 11
           POWER = VAL(INNY$)
           IF POWER < 0 THEN POWER = POWER + 256
           POW$ = STR$(POWER)
           IF LEN(POW$) = 3 THEN POW$ = POW$ + "  "
           IF LEN(POW$) = 4 THEN POW$ = POW$ + " "
           IF LEN(POW$) = 2 THEN POW$ = POW$ + "   "
           IF LEN(POW$) = 1 THEN POW$ = POW$ + "    "
           PRINT POW$; "  ";
           NEXT J
           PRINT " "
           NEXT L
           GOTO NEXTTEST1
FINDATA:   PRINT "AVERAGE   ";
           FOR M = 1 TO 10
           SUMROW = 0
           FOR N = 1 TO 10
           SUMROW = SUMROW + MATRIX(N, M)
           NEXT N
           A$ = STR$(INT(SUMROW / 10))
           GOSUB MAKE4
```

```
            PRINT A$;
            NEXT M
            PRINT " "
            PRINT "ABSDEV    "
            FOR M = 1 TO 10
            ABSDEV = 0
            FOR N = 1 TO 10
            SUMROW = SUMROW + MATRIX(N, M)
            NEXT N
            ROWAVE = SUMROW / 10
            FOR N = 1 TO 10
            ABSDEV = ABSDEV + ABS(MATRIX(N, M) - ROWAVE)
            NEXT N
            A$ = STR$(INT(ABSDEV / 10))
            GOSUB MAKE4
            PRINT A$;
            NEXT M
            PRINT " "
            RETURN
ASCII:      FOR L = 0 TO 255
            IF CHR$(L) = CHAR$ THEN VALUE = L: A$ = STR$(VALUE): RETURN
            NEXT L
            RETURN
ASCIIHEX:   FOR L = 0 TO 255
            IF CHR$(L) = CHAR$ THEN VALUE = L: GOTO CONVERTH
            NEXT L
            RETURN
CONVERTH:   A = INT(VALUE / 16)
            IF A > 9 THEN GOSUB CONVERTH1: GOTO SKIPIT1
            C$ = STR$(A)
SKIPIT1:    A$ = C$:
            B = VALUE - (16 * A)
            IF B > 9 THEN GOSUB CONVERTH1: GOTO SKIPIT2
            C$ = STR$(B)
SKIPIT2:    A$ = A$ + RIGHT$(C$, 1) + "H"
            RETURN
CONVERTH1:  IF A = 10 THEN C$ = "A"
            IF A = 11 THEN C$ = "B"
            IF A = 12 THEN C$ = "C"
            IF A = 13 THEN C$ = "D"
            IF A = 14 THEN C$ = "E"
            IF A = 15 THEN C$ = "F"
            RETURN
SEEYA:      PRINT "PROGRAM ABORTED"
            END
            RETURN
```

Enclosure 4
Filter Output Data and Analysis of Wolf Calls

HOWL COMPOSITE										
	250	500	750	1000	1250	1500	1750	2000	2250	2500
EPOCH 1	126	116	85	29.5	21	18	16.75	6.5	12	9.25
EPOCH 2	132	120.5	82.5	39.5	19.75	17.75	8.5	7.75	12.25	10.25
EPOCH 3	113	127.5	91.5	34	22	18.25	11.25	11	12.75	9.25
EPOCH 4	118.75	118.75	74.75	32.5	19.5	17.25	9.5	8.25	12.75	9
EPOCH 5	122.75	109.25	71.5	29.5	18.25	16.75	10	5.75	12	9.25
EPOCH 6	129.75	109.25	65.5	25.75	20.25	17.75	14.25	6.5	12.75	9.25
EPOCH 7	100.25	96	68.75	25.75	19	17.75	8	6.25	12.25	8.5
EPOCH 8	85.25	85.5	63.25	25.25	18.5	16	6.25	6.25	12.25	7
EPOCH 9	74.5	63.25	42.75	23.5	16.5	15	5.5	5.25	12	5
EPOCH 10	82.75	61.75	48.5	27	16.5	15	4.5	4.25	12	6
SUM	1085	1007.75	694	292.25	191.25	169.5	94.5	67.75	123	82.75
STDV	21.26	23.54	15.41	4.91	1.78	1.22	3.85	1.87	0.33	1.69
GROWL COMPOSITE										
	250	500	750	1000	1250	1500	1750	2000	2250	2500
EPOCH 1	107.5	91.75	54	28	27	24.5	29.25	22.5	17.5	9.25
EPOCH 2	116.25	94.25	67.5	33.5	30.25	28.75	35.5	24.75	19.75	11.5
EPOCH 3	126.75	94.75	57	30.75	28.5	23	20.25	13.5	17.25	9.75
EPOCH 4	103.5	82.5	49	28	27.25	23.5	26.5	16.75	16.5	9.5
EPOCH 5	97.75	84.25	57.25	28.75	25	24	23.25	19.5	18.5	9.5
EPOCH 6	102	71.5	47.25	28.25	26.25	20.75	14.75	12	15.75	8.25
EPOCH 7	90.25	50.75	38.25	25.5	22	19.25	14.5	10	15.5	7.5
EPOCH 8	83.25	90.75	55.25	32.25	25.75	21.25	18.75	13.25	18.25	12.75
EPOCH 9	125.75	79.75	46.75	28.5	27.75	20.25	19	14	15.75	8.75
EPOCH 10	101	73.75	51.5	28.75	25.25	22.75	21	17.5	17	8.75
SUM	1054	814	523.75	292.25	265	228	222.75	163.75	171.75	95.5
STDV	14.16	13.52	7.85	2.32	2.24	2.71	6.57	4.74	1.37	1.54
BARK COMPOSITE										
	250	500	750	1000	1250	1500	1750	2000	2250	2500
EPOCH 1	62	59.25	61.25	48	61.5	39.75	48.75	36.5	20	13.25
EPOCH 2	63.25	62.5	47.5	40	45.75	28.75	42	30	17	9.75
EPOCH 3	57.5	58.5	50.5	47	50.5	30	40	26	16	10.5
EPOCH 4	64.5	77.5	51	54.25	47.5	29.75	42.75	27.75	17.5	11.5
EPOCH 5	53	66.25	44.25	56	42.75	23.5	26.75	18.75	13.25	8
EPOCH 6	48.25	53.75	49.75	48.5	50	27.25	30.75	22.5	15	9.75
EPOCH 7	47.5	56.75	36.25	35.75	35	22.5	26.75	20.25	13.75	7.75
EPOCH 8	49.75	61.75	51.5	45.5	48	30.5	36.5	31.75	17.75	11
EPOCH 9	51.25	73.25	44.25	37.75	36.5	23.75	28.25	22	13.5	8
EPOCH 10	67.25	94.25	51	43.75	42	27.75	32.5	26	13.25	9.75
SUM	564.25	663.75	487.25	456.5	459.5	283.5	355	261.5	157	99.25
STDV	7.38	12.24	6.47	6.60	7.61	4.94	25.00	5.54	2.33	1.74
WHINE COMPOSITE										
	250	500	750	1000	1250	1500	1750	2000	2250	2500
EPOCH 1	70	102.25	36.25	40.5	26	39.5	37.5	13.75	16	10.25
EPOCH 2	74	106.5	33.75	42.5	29.5	41	63	129	22	13
EPOCH 3	63.75	30	19.75	20.75	15.75	14.75	7.5	6.75	13	6.25
EPOCH 4	65.25	55.25	27.25	24	18.75	20.75	17.25	8.75	14.75	7
EPOCH 5	91.25	93.75	44.5	32	28	39.25	47.5	22.75	17	11
EPOCH 6	171	166	77.5	46.75	54.75	53.75	54	55.5	26.5	22.25
EPOCH 7	69.75	41.5	25.75	34.25	29	25	25.75	17	19.25	11
EPOCH 8	64.5	33.25	24	22	17	15.25	9.75	8.25	13	8
EPOCH 9	52	26.75	21.75	23.25	18.25	36.25	15.5	16.75	30	20.55
EPOCH 10	75.25	37.5	21	22	20	18.75	14.75	15	19.25	22.75
SUM	796.75	692.75	331.5	308	257	304.25	292.5	293.5	190.75	317
STDV	33.63	46.01	17.44	9.76	11.48	13.27	25.00	37.71	5.67	61.34

COMPARISON MATRIX				
	HOWL COMP.	GROWL COMP.	BARK COMP.	WHINE COMP
HOWL1	13.74056494	21.24445516	33.45437191	38.40712824
HOWL2	16.80887043	22.16059284	35.22939114	39.9940308
HOWL3	14.69347219	23.91959187	35.05909297	40.15504327
HOWL4	24.86761197	26.65148917	31.79158694	43.17218433
GROWL1	29.80181412	19.85414	29.29334395	42.67718946
GROWL2	18.01577434	14.64980802	27.15059852	30.23024148
GROWL3	22.19488961	14.35502961	24.54251413	37.0136799
GROWL4	17.44485956	16.59975527	24.74772717	36.97516058
BARK1	31.91814727	25.19080934	11.99979166	33.94521321
BARK2	32.46295004	26.74905372	19.02971361	41.93599289
BARK3	32.54901112	25.52580802	13.86001443	39.13537402
BARK4	33.95383815	26.61506857	17.09312727	34.49213679
WHINE1	36.71986009	33.99568906	35.50880173	2.739981752
WHINE2	36.41899401	33.21719246	34.08364711	1.103403825
WHINE3	36.21716616	32.8367458	33.29391836	2.225421308
WHINE4	35.71460661	32.577168	33.57655432	1.724093965
THRESHOLD				
17				
	% POS CORR.	FALSE NEG	%NEG CORR.	FALSE POS
	75	25	100	0
	TOTAL CORRECT RATE			
	93.75			

AVERAGE IS: 38.21 08-29-1997

FREQ	250	500	750	1000	1250	1500	1750	2000	2250	2500
EPOCH 1	651	648	757	169	139	106	52	50	69	66
EPOCH 2	621	879	904	175	120	112	49	47	69	79
EPOCH 3	543	816	600	174	126	98	59	41	72	51
EPOCH 4	596	617	356	232	109	91	59	39	69	-9396
EPOCH 5	702	639	240	233	107	87	41	34	69	50
EPOCH 6	686	692	245	137	100	85	50	35	71	42
EPOCH 7	605	595	241	129	100	85	51	28	69	45
EPOCH 8	7877	397	148	117	94	81	28	23	68	37
EPOCH 9	451	261	115	112	87	80	26	22	68	29
EPOCH 10	338	195	107	120	86	80	25	22	68	26

NORMALIZED PARAMETERS

FREQ	250	500	750	1000	1250	1750	2000	2250	2500	
EPOCH 1	121	121	140	30	25	19	09	08	12	12
EPOCH 2	115	164	167	31	22	20	08	08	12	14
EPOCH 3	100	152	111	31	23	18	11	07	13	08
EPOCH 4	110	115	66	42	19	15	11	06	12	08
EPOCH 5	131	118	44	42	19	15	07	06	12	08
EPOCH 6	127	128	45	25	18	15	08	06	12	07
EPOCH 7	112	110	44	23	18	15	08	04	12	07
EPOCH 8	100	73	26	20	17	14	04	03	12	06
EPOCH 9	83	47	20	20	15	14	04	03	12	04
EPOCH 10	66	35	19	22	15	14	03	03	12	04

08-29-1997

FREQ	250	500	750	1000	1250	1500	1750	2000	2250	2500
EPOCH 1	955	636	232	149	100	97	44	31	70	57
EPOCH 2	923	686	301	169	109	103	49	39	72	59
EPOCH 3	877	868	642	201	136	119	107	74	73	68
EPOCH 4	947	852	589	209	128	130	91	72	70	75
EPOCH 5	1107	932	523	185	111	125	125	51	71	79
EPOCH 6	1045	767	379	174	125	142	250	63	86	87
EPOCH 7	629	468	282	162	122	122	97	71	74	53
EPOCH 8	421	493	526	164	117	97	53	81	68	56
EPOCH 9	342	390	420	145	95	89	46	64	68	45
EPOCH 10	332	348	651	127	98	84	38	35	68	52

NORMALIZED PARAMETERS

FREQ	250	500	750	1000	1250	1750	2000	2250	2500	
EPOCH 1	145	97	35	22	15	14	06	04	10	08
EPOCH 2	140	104	46	25	16	15	07	06	11	09
EPOCH 3	133	132	98	30	20	18	16	11	11	10
EPOCH 4	144	130	89	31	19	19	13	11	10	11
EPOCH 5	169	142	79	28	16	19	19	07	10	12
EPOCH 6	159	117	57	26	19	21	38	09	13	13
EPOCH 7	96	71	43	24	18	18	14	10	11	08
EPOCH 8	64	75	80	25	17	14	08	12	10	08
EPOCH 9	52	59	64	22	14	13	07	09	10	06
EPOCH 10	50	53	99	19	14	13	05	05	10	07

9/3
~~08-29~~-1997

GROWL at 650

FREQ	250	500	750	1000	1250	1500	1750	2000	2250	2500
EPOCH 1	519	400	170	131	116	129	184	87	80	51
EPOCH 2	660	604	369	193	175	204	371	223	126	90
EPOCH 3	736	561	329	157	159	142	136	104	92	58
EPOCH 4	542	370	160	118	104	94	62	58	84	40
EPOCH 5	632	530	368	166	143	149	164	140	118	61
EPOCH 6	834	677	370	188	185	124	105	69	93	71
EPOCH 7	322	167	105	110	87	80	32	24	68	27
EPOCH 8	360	306	198	121	97	84	38	33	69	31
EPOCH 9	421	386	256	130	132	98	96	87	77	35
EPOCH 10	457	378	278	127	124	90	63	56	70	34

NORMALIZED PARAMETERS

FREQ	250	500	750	1000	1250	1500	1750	2000	2250	2500
EPOCH 1	103	79	34	24	22	24	36	17	15	09
EPOCH 2	131	120	73	38	34	40	73	44	24	17
EPOCH 3	146	111	65	31	31	27	26	19	18	10
EPOCH 4	107	73	31	23	19	18	11	10	15	07
EPOCH 5	125	105	73	32	27	28	32	27	23	11
EPOCH 6	165	134	73	36	36	23	20	13	18	13
EPOCH 7	64	32	20	20	17	15	05	03	13	05
EPOCH 8	72	60	39	23	18	15	06	06	13	05
EPOCH 9	83	77	51	24	26	18	18	17	14	06
EPOCH 10	90	74	55	24	23	17	11	10	13	06

AVERAGE IS: 38.79

9/2
~~08-29~~-1997

GROWN at 660 ON TAPE

FREQ	250	500	750	1000	1250	1500	1750	2000	2250	2500
EPOCH 1	496	512	365	151	149	140	183	167	104	63
EPOCH 2	351	195	184	127	106	94	53	42	71	35
EPOCH 3	447	239	196	130	101	98	93	65	87	56
EPOCH 4	493	293	199	137	128	125	124	89	91	60
EPOCH 5	327	159	104	109	90	80	38	27	68	25
EPOCH 6	328	159	103	110	86	80	38	33	69	28
EPOCH 7	441	185	134	122	105	101	5	58	77	37
EPOCH 8	1880	634	334	211	152	117	93	78	112	120
EPOCH 9	1014	375	195	156	118	101	69	52	84	71
EPOCH 10	577	278	182	154	126	106	73	59	83	48

NORMALIZED PARAMETERS

FREQ	250	500	750	1000	1250	1500	1750	2000	2250	2500
EPOCH 1	118	123	87	36	34	33	44	39	23	14
EPOCH 2	83	45	44	30	25	22	12	09	15	07
EPOCH 3	107	57	45	30	23	23	22	14	20	12
EPOCH 4	118	69	47	31	30	30	30	20	20	14
EPOCH 5	79	38	23	25	20	19	07	06	15	06
EPOCH 6	79	38	23	25	20	19	07	07	15	06
EPOCH 7	106	44	31	28	25	23	19	12	17	07
EPOCH 8	49	152	80	50	36	26	22	17	26	28
EPOCH 9	243	90	45	36	28	23	15	11	19	15
EPOCH 10	139	66	42	36	30	25	17	14	19	11

AVERAGE IS: 38.04

error!

GROWL SAMPLE#3

09-03-1997

FREQ	250	500	750	1000	1250	1500	1750	2000	2250	2500
EPOCH 1	543	386	172	119	104	100	74	65	87	44
EPOCH 2	749	627	454	198	188	175	201	163	130	79
EPOCH 3	726	573	281	156	140	105	70	83	84	56
EPOCH 4	323	170	106	110	9347	80	31	25	68	28
EPOCH 5	377	367	235	127	101	86	41	35	71	33
EPOCH 6	442	372	257	130	137	104	107	99	85	41
EPOCH 7	459	383	289	130	128	96	74	65	75	39
EPOCH 8	511	433	291	141	137	102	101	69	75	37
EPOCH 9	539	603	365	157	197	119	192	124	79	49
EPOCH 10	528	622	452	164	153	165	257	208	115	71

NORMALIZED PARAMETERS

FREQ	250	500	750	1000	1250	1500	1750	2000	2250	2500
EPOCH 1	103	74	32	22	19	19	13	12	16	07
EPOCH 2	144	120	87	38	35	32	38	30	24	15
EPOCH 3	139	110	53	29	26	20	12	08	15	10
EPOCH 4	62	32	20	20	16	15	05	03	12	05
EPOCH 5	72	70	44	24	19	16	07	06	12	06
EPOCH 6	84	70	49	24	25	20	20	19	15	07
EPOCH 7	88	73	55	24	24	17	13	12	13	06
EPOCH 8	98	83	55	26	25	19	19	12	13	06
EPOCH 9	103	116	69	30	38	22	36	22	15	08
EPOCH 10	101	119	87	31	29	31	49	39	21	12

AVERAGE IS: 37.6

EPOCH 9 522 09-03-1997

GROWL SAMPLE 4 ≃660 ON TAPE

FREQ	250	500	750	1000	1250	1500	1750	2000	2250	2500
EPOCH 1	461	392	281	131	145	101	110	100	74	37
EPOCH 2	468	403	291	128	119	94	91	75	72	36
EPOCH 3	501	445	282	144	151	104	91	62	73	36
EPOCH 4	554	676	427	167	191	138	262	151	85	58
EPOCH 5	502	535	391	152	153	145	205	171	109	65
EPOCH 6	347	195	191	126	107	96	57	39	71	35
EPOCH 7	448	240	206	136	101	98	97	64	87	54
EPOCH 8	494	296	207	134	-530	117	128	82	92	57
EPOCH 9	322	159	104	109	90	80	37	26	68	26
EPOCH 10	326	158	101	109	86	80	38	33	69	28

NORMALIZED PARAMETERS

FREQ	250	500	750	1000	1250	1500	1750	2000	2250	2500
EPOCH 1	106	91	63	30	33	22	24	22	16	07
EPOCH 2	107	92	66	28	27	21	19	16	16	07
EPOCH 3	115	101	65	33	34	22	21	13	16	07
EPOCH 4	127	156	98	38	44	31	60	34	19	12
EPOCH 5	115	123	89	34	34	33	47	39	24	15
EPOCH 6	79	44	44	28	24	21	12	09	15	07
EPOCH 7	103	54	47	30	22	22	21	13	19	12
EPOCH 8	114	68	47	30	24	25	28	18	21	12
EPOCH 9	74	36	22	24	19	18	07	06	15	06
EPOCH 10	74	36	22	24	19	18	07	07	15	06

AVERAGE IS: 38.8

Enclosure 5
Article Presented at NASA Technology 2007 Conference

MINIATURE LOW POWER DIGITAL AUDIO CAPTURE AND IDENTIFICATION SYSTEM

Protagoras N. Cutchis
Johns Hopkins University Applied Physics Laboratory
Johns Hopkins Road, Laurel, MD 20723
301-953-5013

William S. Seegar
U.S. Army Edgewood Research, Development and Engineering Center
Chemical and Biological Defense Command
Aberdeen Proving Ground
Aberdeen, MD 21010
410-671-2586

We report here on the development of a small low power Digital Audio Capture and Identification System (DACIS) for the automatic identification of animal calls supported by the Strategic Environmental Research and Development Program (SERDP). The system is light enough and low power enough to be attached to larger birds so that the bird can act as a sentinel animal for its own environment. The system is designed to be used with a Global Positioning System (GPS) and Argos Platform Transmitter Terminal (PTT) which will allow accurate location fixes and telemetry of data via satellite. After capture and identification, the time and location of identified sounds are periodically transmitted up to the Argos satellite for relay to the earth.

INTRODUCTION

In 1981 the Bird-borne program was initiated at the Johns Hopkins University, Applied Physics Laboratory (APL) under U.S. Army funding, to develop a capability to locate (i.e., track) and monitor small highly mobile organisms on a local, regional and global scale. Initially, the program focused on the development of small Platform Transmitter Terminals (PTTs) to be tracked via the French-U.S. Argos satellite system. Since the inception of the program, miniaturization has led to the fielding of transmitters that weigh less than one ounce and are capable of interfacing with an array of sensors.

The primary objective of the Bird-borne and the Remote Environmental Sensing Technology (REST) program (follow-up initiated in 1991) has been to develop a system for the remote tracking and monitoring of free ranging organisms that pose especially difficult field problems for study. Biotelemetry can be used to collect information from the environment surrounding the animal (temperature, humidity, altitude) as well as behavioral and physiological parameters (motion, core temperature and heart rate) of the organism. Biotelemetry has enabled scientists to accurately study behavior, home range, and habitat use of wildlife for basic research and the development of management plans for conservation.

However, for studies of free ranging organisms that travel long distances over extended periods and frequent inaccessible habitats due to geographic or boundary restrictions, such as military installations, space-based tracking and monitoring systems are advantageous. Remote tracking and monitoring systems can support effective study of these organisms and aid in identifying their range and critical habitat requirements for breeding, migration and wintering.

The planned uses for the acoustic system are to identify the prey on which predatory animals are feeding and identification of known calls of the animal on which the unit is mounted to identify behaviors correlated with those sounds.. While the identification of prey is a desirable goal, it was felt that the number of targets for identification would be too large for a proof of concept. Therefore, the present system is focusing on the latter.

The DOD has established requirements for environmental research, technology development and land management and supports a variety of programs such as Legacy and the Strategic Environmental Research and Development Program (SERDP). Biological studies designed to evaluate effects of military land use on natural resources pose unique and difficult problems because collecting biological data during military activities is required. Advanced technologies that allow remote tracking and monitoring of wildlife can alleviate many of these conflicts yet provide comprehensive data.

TECHNOLOGY DEVELOPMENT

Initial Birdborne Development

The Bird-borne effort to develop a space based tracking and monitoring capability started with a study to evaluate the critical engineering paths to build a satellite received transmitter to be used on free ranging birds. Requirements for the development of the first prototype satellite transmitter were, 1) identify a space based system for transmitter development, 2) develop a PTT under 200 grams, 3) allow for 270 days of operation, and 4) accommodate environmental, behavioral, and physiological sensors.

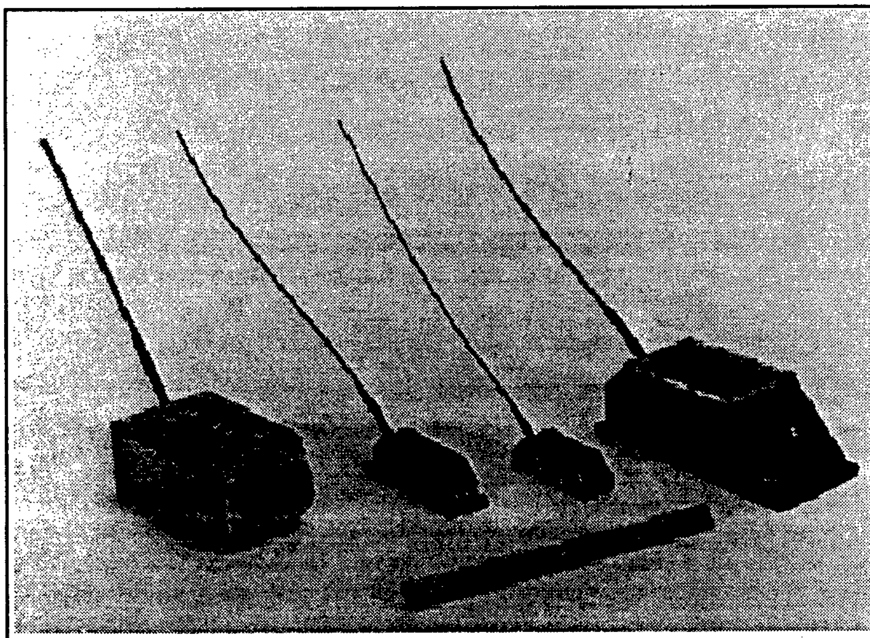


Figure 1: From Left to Right: Original 200 g Argos PTT, 30 and 20 g Nano PTTs and prototype solar powered GPS/PTT

The French operated Argos System fielded in the 1970s proved to be the basis for the development of a bird borne transmitter. The Argos system consists of receivers on the Tyros N series of National Oceanic and Atmospheric Administration (NOAA) satellites positioned in low (850 km) polar orbits. The PTTs weighed a kilogram or more and operated with primary batteries. The PTTs are positioned based on the Doppler shift, which is dependent on a highly stable frequency transmission at 104.6 megahertz.

The initial bird borne PTT had to be relatively small (<200g), to avoid adversely affecting bird flight. System Argos required PTTs to transmit a minimum of 1.0 watt. To meet this power requirement for transmission for 270 days required 500 grams of primary batteries which exceeded by more than a factor of 2 the maximum allowed mass. Therefore, we initially met this power requirement by using a solar array with rechargeable Ni/Cad batteries. This power a pack allowed for tracking duration of nearly 3 years.

During the past 15 years the electronics in the satellite transmitter have been continually miniaturized and provided new capabilities through the integration of microprocessors and mini-computers as shown in Figure 1. The newest experimental bird borne transmitters weigh 20 grams, which includes 3.5 grams of electronics, an 8 gram battery and the remainder in the container. The transmitter can interface with a variety of sensors to collect information from the environment surrounding the organism as well as collecting behavioral data. Many researchers are now applying PTTs to the study of birds [1] as well as other wide-ranging animals [2].

DIGITAL AUDIO CAPTURE AND IDENTIFICATION SYSTEM

There remain, however, many questions for research and important conservation issues that need to be addressed in a timely, effective manner, and would benefit from additional development of technology. We report here on the development of a bird borne transmitter that will incorporate a Digital Audio Capture and Identification System (DACIS) that will assist in the interpretation of acoustical information to link time and location to discrete animal behaviors. The initial goals of this system are to analyze and time tag 5 distinct animal sounds from the same species or different species and transmit that data, along with the location at which the call was heard, to the Argos satellite.

Acoustic analysis of various bird calls was performed at APL by producing "waterfall" plots. These show intensity of sound as a function of frequency and time. Plots of several species appear to indicate that a system which prepares a course version of these plots and compares them to a known template will be able to identify different species or different calls of the same species.

System Hardware

The block diagram of the entire electronics system of the digital audio capture and control circuit is shown in Figure 2. An audio trigger, which consists of a micropower operational amplifier and voltage comparator is always on and listening to the environment. When a preset threshold is detected, the microprocessor is turned on to start the digital recording. The design of the digital audio capture circuit centers on an MC68HC811 micro controller. This device was chosen because it has several system components on a single chip and small size and weight are critical in this system's design. The subsystems of the micro controller are the internal universal asynchronous receiver transmitter (UART), the internal timer, internal Read Only Memory (RAM), Electrically Erasable Programmable Read Only Memory (EEPROM), and an 8-bit Analog to Digital (A/D) converter. The A/D converter is used to sample the amplified signal from the electret microphone. The audio sample is then immediately stored in memory for future transmission.

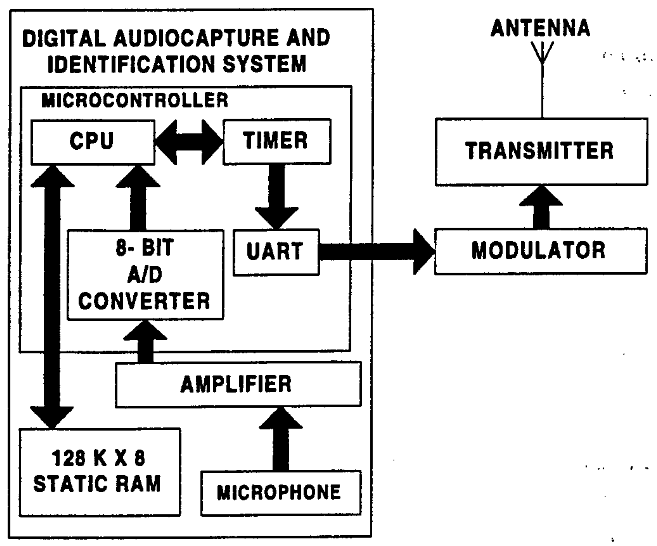


Figure 2: DACIS Block Diagram

The MC68HC811, which is an 8 bit micro controller, can only directly address 64K bytes of memory space. Therefore, page mode addressing is implemented using a separate output bit to control the highest address line (A16) of the memory. If more than 128K of memory is required in future versions, new circuitry will be added to control the chip select pins of the memory chip to avoid data bus clashes. There are enough spare gates and microprocessor output pins present on the existing design to increase memory to 512K bytes if desired.

The initial memory configuration, which used two 32 Kbyte memory chips was replaced with a single S-MOS systems 128K byte memory chip. The sampling rate, which can be easily changed in software, was set at 6,000 samples per second to yield reasonable quality audio playback. Initial experiments were conducted with a sampling rate of 2,700 samples per second and proved to yield marginal results for the intended system use. There is a direct trade off between sampling rate and total record time. At 6,000 samples per second and using 128K (actually 131,072 bytes) less 4,096 bytes for EEPROM (and the image of EEPROM in upper memory which is inaccessible in the present implementation), the total record time available is 126,976/6,000 ≈ 21.2 seconds. Presently, only 5 seconds is recorded after each triggering event.

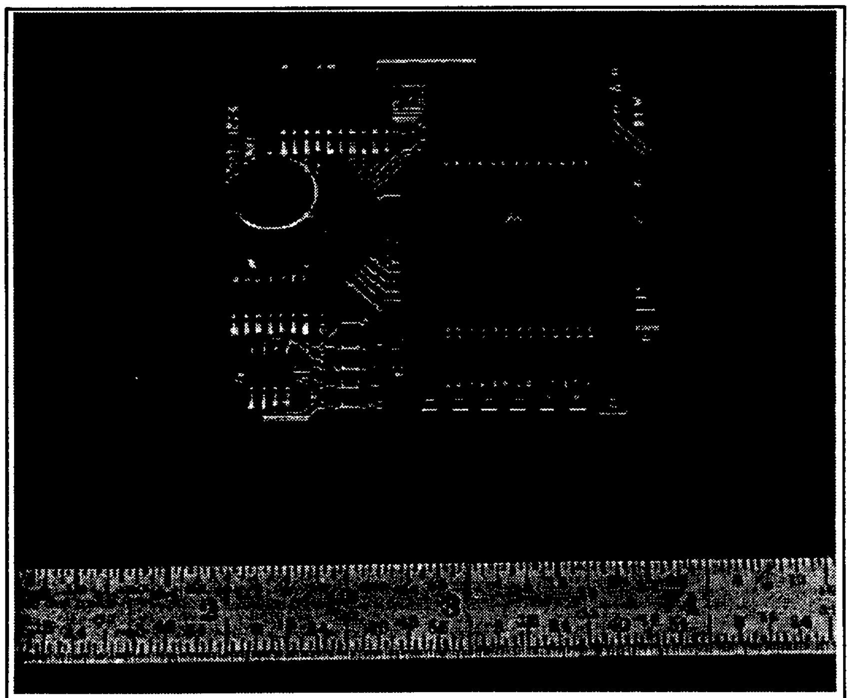


Figure 3: Prototype DACIS Board Showing Processor, RAM and other components

The microprocessor's on-chip UART is used to generate the serial data stream during transmission. The data rate is programmable and has been set to 9,600 bits per second due to requirements of the prototype modulator. The data rate is not infinitely flexible in that the rate is obtained from selectable divide ratios of the microprocessor's clock. The present data format is 8 data bits per word with one start bit, one stop bit, and no parity bit. Additionally, a 74HC4024 counter is used to divide the microprocessor clock by 8 to obtain a 153.6 KHz 50% duty cycle square wave which is 16 times the data rate. This signal is required by the modulator. Lastly, during transmission, a line called transmit is brought low to activate the transmitter. This line activates the transmitter approximately 200 msec before the data starts to allow the transmitter to stabilize. This portion of the system will also serve as the control for future features such as control of a Global Positioning System (GPS) receiver. The microprocessor will then reformat the data and transmit it through the Argos satellite.

All components used in the design are available in surface mount packages. The prototype board is shown in close up in Figure 3. The board measures approximately 2" by 2", weighs about 15 grams, and has a socket for the CPU for test purposes only. A slightly smaller version has been designed without the socket for later testing

SOFTWARE

A flow chart of the signal processing is shown in Figure 4. All of the software for the system was written in 6800 series assembly language. After power up, the initial section of code simply initializes the timer counter, baud rate resister, A/D converter and I/O pins. The audio capture portion of the software controls the internal A/D converter and stores each sample in RAM. The timing of audio sampling is controlled by interrupts from the internal timer. After all memory is filled, the program stops capturing audio and starts to process the recorded data.

The processing starts by implementing a digital 23 stage FIR (finite impulse response) filter on ten bands centered form 250 Hz to 2500 Hz in 250 Hz evenly spaced increments. The total recording of 5 seconds is divided into 10 epochs of 0.5 seconds each and the ten filters are performed on each of the ten epochs to yield 100 time/frequency parameters. This is a crude representation of the waterfall plot. The parameters are then normalized to account for differences in volume level which may disturb the pattern match.

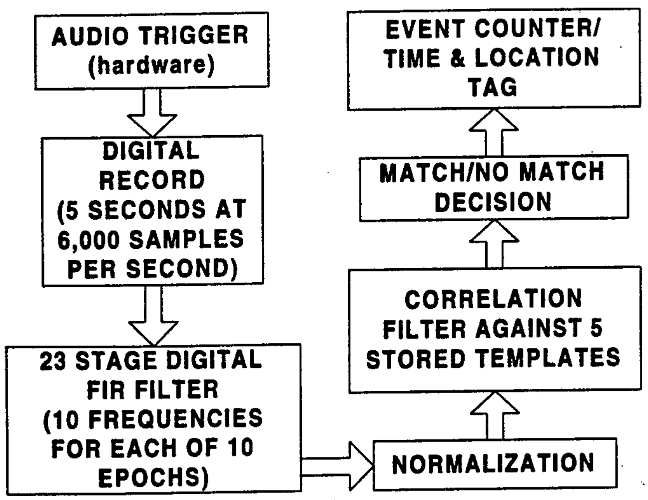


FIGURE 4: Signal Flow Diagram

The final processing stage is to match the 100 parameters against the 5 stored templates and the do the threshold detection. These two critical stages are currently under development and testing using wolf calls. The positively identified call information, along with the time and location, will be stored for later transmission. The exact scenario by which transmission will be started has not yet been determined. The possibilities include internal timer activation, reception of a transmitted signal to the animal mounted unit or some sort of proximity activation from a small device placed in or near the animals habitat. The last method has the advantage of allowing the receiver to be near the free ranging animal also so that the transmission distance is short. Presently, if the entire contents of the memory are sent at 9,600 baud, the total transmit time is about 132 seconds.

Global Positioning System Qualified Argos PTT

The Argos system is capable of giving locations to within ±150m anywhere on the surface of the earth but locations obtained from tiny low power (100mW) ARGOS beacons, mounted on the backs of birds often give locations in the range of ±2 km of the birds true locations. To achieve the highest grade Argos location at least 4 messages have to be received by the satellite over a period exceeding 420 seconds. The availability of small commercial GPS receiver modules have now made it possible to combine such a receiver with an Argos transmitter and field a package small enough to be carried by a goose size bird. By scheduling the collection of GPS locations throughout the day and storing these positions for later transmission via Argos as many as 20 GPS positions (±20 m) can be transmitted to the user in a single Argos message.

An Argos/GPS package, under development by Microwave Telemetry Inc., incorporates a commercially available GPS receiver, a microcontroller based data logger and a Microwave Telemetry NANO PTT. The data logger controls the GPS receiver and the collection of GPS data which is dependent on power availability from the solar charged power source. The data logger then sequences data transfer to the NANO PTT at times favorable to satellite availability. The prototype unit is now undergoing laboratory testing and weighs less than 200 grams.

DISCUSSION

The technology we have described is designed for use on free ranging animals to provide data on their locations, behavior, and environment. A GPS receiver, when integrated with an Argos PTT, will provide more accurate location data that can be collected at pre-designated times. The Argos system is dependent upon collecting frequency data on the PTT signal transmission to calculate a single time dependent location. With the use of a minicomputer integrated into the unit GPS positions can be collected according to a programmed schedule. This will increase our ability to locate free ranging organisms, and derive important facts regarding range and habitat use. With enhanced accuracy and greater numbers of locations, home range estimations, programs, and Geographic Information Systems can be used more effectively to relate animal movements to jurisdiction boundaries, habitat, and land use activity maps.

Animal sound identification data, in combination with time and location, will provide additional information relevant to natural resources. For example, the DACIS is designed to recognize animal vocalizations, thus allowing evaluation of animal behaviors and specific activities. By locating exact animal behaviors and linking them to specific habitat within the range of the organism valuable information can be collected on relationships among animals and micro-habitat components of their range. The real time and near real time components of new telemetry will allow more complete study of animal responses to a wide range of ecological variables. Time coded information on location, heading, altitude, speed, ambient temperature, humidity and other sensor data, can be displayed and analyzed relative to other geographically linked features such as geomorphology, ecological community, meteorology, and land use activities. Free ranging organisms tagged with animal track and monitor units act as a sentinel in the population. These sentinel individuals either moving alone or in herds or flocks, can reflect the activities of many organisms and enhance the biological data base dramatically.

Discussions with commercial companies have not yet identified any willing to fund the development of this system. Some have, however, indicated an interest in fabricating the unit after the proof of concept has been completed. A system which automatically trains the DACIS unit from animal sound recordings will likely be necessary to commercialize this unit. Presently, the setting of the 100 template parameters is a labor intensive task.

The use of a stand-off system to monitor and collect pertinent environmental data can be used very effectively in the evaluation of a variety of issues pertinent to many human activities. Noise effects on people as well as wildlife pose a significant problem in many areas and require a significant investment in time and money to resolve. Sensors capable of measuring environmental noise at the organism are in place and being used by the U.S. Air Force to monitor jet aircraft noise and by the Army to evaluate single event noises associated with weapons testing and training. The DOD is developing the capability to monitor the level of noise at the organism and the behavioral response to the event.

References

1. Fuller, M.R., W.S. Seegar, J.M. Marzluff, and B.A. Hoover. 1995a. Raptors, technological tools and conservation. Trans. 60th No. Am. Wildl. & Natur. Resour. Conf. pp. 131-141.

2. McConnell, B.J., C. Chambers, K.S. Nicholas, and M.A Fedak. 1992. Satellite tracking of grey seals (Halichoerus grypus). J. Zool., Lond. 226:271-282.