

UC Davis

UC Davis Previously Published Works

Title

A multichannel electronic monitor of acoustic behaviors, and software to parse individual channels

Permalink

<https://escholarship.org/uc/item/38m8n16d>

Journal

Journal of Neuroscience Methods, 133(1-2)

ISSN

0165-0270

Authors

Hedrick, Ann V
Mulloney, B

Publication Date

2004-02-01

Peer reviewed

A multichannel electronic monitor of acoustic behaviors, and software to parse individual channels

Ann V. Hedrick*, Brian Mulloney

Neurobiology, Physiology, and Behavior, University of California, Davis, CA 95616-8719, USA

Received 17 July 2003; received in revised form 11 October 2003; accepted 17 October 2003

Abstract

We designed an electronic device to monitor calling behavior in male crickets. The device and its associated software can record to disk the activity of as many as 16 individuals simultaneously. The data recorded contain detailed information about the temporal structure of each individual's calls. The temporal resolution achieved with an ordinary PC is good enough to detect and record the occurrence of every pulse of sound from each cricket. The resulting data files are efficient and compact, and so the system is appropriate for experiments lasting many days.

© 2003 Elsevier B.V. All rights reserved.

Keywords: Acoustic communication; Cricket; Individual differences; Temporal structure

1. Introduction

Female choice of mates and male strategies during courtship and mating are currently topics of considerable interest to evolutionary and behavioral biologists (Jennions and Petrie, 1997; Kokko et al., 2003; Tallamy et al., 2002). In many acoustic insects and anurans, males call to attract sexually receptive females. Species-specificity is encoded in the temporal patterns of these calls (Gerhardt and Huber, 2002). In addition, variation among males in the temporal properties of calls may be used by females to choose particular males as mates (Gerhardt and Huber, 2002). Therefore, detailed records on the calling activity of individual males may help us understand the processes by which females choose mates and males compete for mates.

To document individual differences in the calling behavior of male field crickets (*Gryllus integer*), we designed a device to monitor calling activity continuously. In *G. integer*, male song consists of trains of chirps, each chirp consisting of two or three sound pulses (each approximately 10 ms long at 25 °C) separated by pauses of approximately 30 ms (Hedrick and Weber, 1998; Hedrick et al., 2002). Males sing in trains of chirps, and differ from one another in the length of these

trains (calling-bout durations). Pioneering work that used 5 min tape recordings of the calls of different males found that calling-bout durations are heritable, and females prefer calls with longer bouts (Hedrick, 1986, 1988).

As part of a project to study individual differences in calling behavior of male *G. integer* in greater detail and for longer periods, and to discover daily and weekly patterns of calling, we wanted to monitor many males simultaneously and continuously. The device and software described here were designed to do this. This device was constructed with 16 parallel channels, and so could handle as many as 16 males, each calling undisturbed in acoustic isolation. The output from the device is a 16-bit word, which changes whenever the sounds produced by any of the animals changes. This word is read by a parallel port in a PC, and trains of these words are stored along with the time at which the word was read. The array of words is later parsed into separate files for each animal that list the times at which each chirp started and stopped. The temporal resolution of the device is limited by the grain of the PC system's clock, but is good enough to detect every chirp in a train, or the omission of a single chirp. Prior efforts to monitor the acoustic activity of crickets have produced useful devices (Bertram and Johnson, 1998; Kidder and Sakaluk, 1989), but the temporal resolution achieved by the device we describe here is a significant advance. This system should be equally useful for monitoring the activity of any acoustic animal.

* Corresponding author. Tel.: +1-530-754-7611; fax: +1-530-752-5582.

E-mail address: avhedrick@ucdavis.edu (A.V. Hedrick).

2. Methods

2.1. Hardware

To monitor the calling of as many as 16 crickets at the same time, we constructed a 16-channel device that reported sounds recorded by up to 16 microphones, each of which was connected to an independent, variable-threshold circuit that produced a TTL-level high whenever a sound occurred.

The detection circuitry for each channel (Fig. 1) consists of a Kobitone omnidirectional electret condenser microphone connected to an automatic gain-control IC amplifier. The output of this amplifier is connected to a tunable tone-decoder that sets the channel's bandwidth and center-frequency. The decoder's output connects to a voltage-comparator that produces a TTL-high whenever a sound in the right frequency range lasts longer than the minimum set by the decoder. The gain adjustment sets the sensitivity of the channel, and the Filter adjustment sets the channel's optimum frequency. The delay adjustment sets the minimum duration of each output pulse, and can be adjusted so that the output is brief enough to signal each sound pulse or long enough to signal each chirp, and not the pulses that constitute the chirp. The output of each channel is equipped with an LED that illuminates whenever a sound is detected; this visual cue lets the user tune each channel's gain and delay. The schematic for one channel (Fig. 2) shows the details of the circuit. We constructed 16 of these on one board using perforated circuit board and wire-wrap connections.

A parallel latched-output buffer then assembles the output from the 16 comparators into a 16-bit data word (Fig. 2). This word represents the state of each channel at the instant when the buffer is latched and read by the computer. This word is read periodically by a CIO-DIO 24 parallel port (Omega Engineering, Stamford, CT, USA) in a PC, connected to the monitor by a 37-pin D-connector.

2.2. Software

To ensure that any change on any of the 16 channels would be detected once an experiment began, our program read the parallel port continually, without attempting to process the information. To make the data files as compact as possible and to eliminate redundant information, the program compared the most recently read word with the word read the previous time. If they were identical, this meant that no change had occurred so no action was called for, and the program looped to read the port again. If the two words differed, this meant that a change had occurred, so the new word was stored in an array along with the time at which the change was detected. The new word was saved for the next comparison, and the program looped to read the port again.

In a given experiment, not all channels were necessarily used. Changes in the output of unused channels had no meaning, so at the start of the experiment the user specified which channels were to be used. From this list of channels, a 16-bit word was constructed that served as a mask that insured that only changes in the pertinent channels would be detected.

The program that monitored the state of these 16 channels, called "BigBro", was written in Microsoft Basic Version 6, and incorporated drivers for the parallel port that were supplied by the port's manufacturer, Omega Engineering (Stamford, CT, USA). It runs both on an MS DOS computer and in a DOS window using the Windows 98 operating system. Key features of the code that contribute to its performance are described briefly here.

2.2.1. Construction of the channel-mask

This procedure asks the user for the channels to be monitored, and verifies that the user's answers are plausible. The channel-mask itself is a 16-bit word that is initially zero, so all of its bits are low. The procedure then uses a logical OR operation to set high the bit that corresponds to the

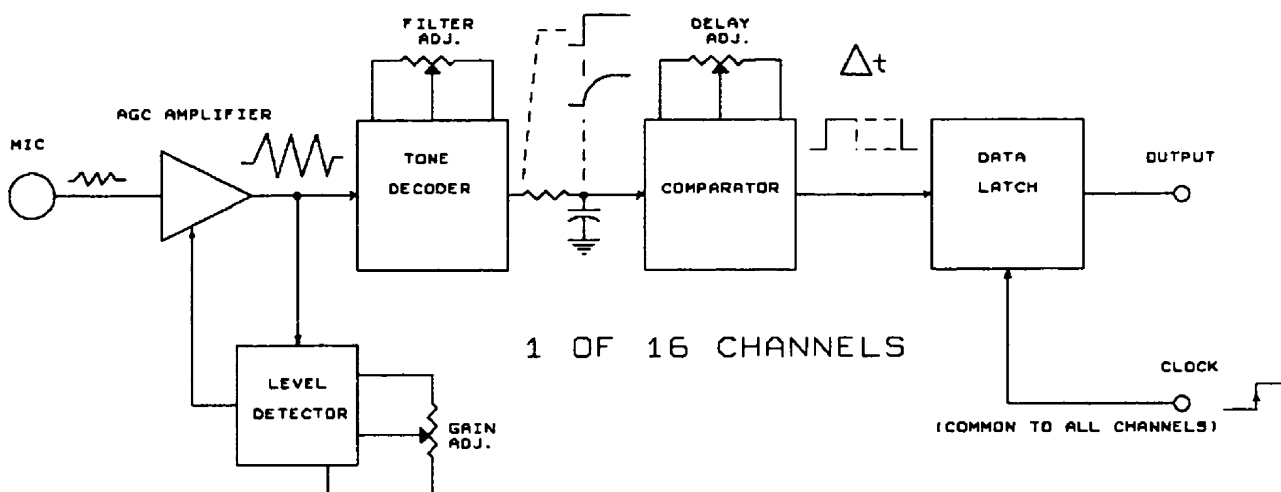


Fig. 1. Block diagram of one channel of the monitor.

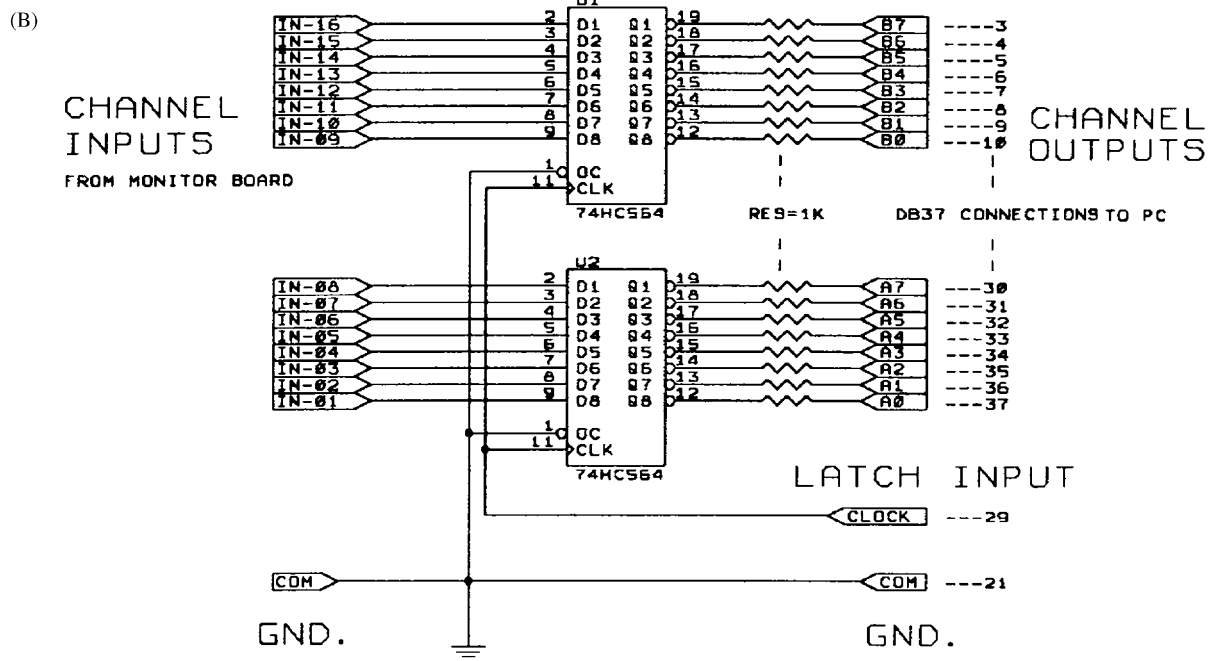
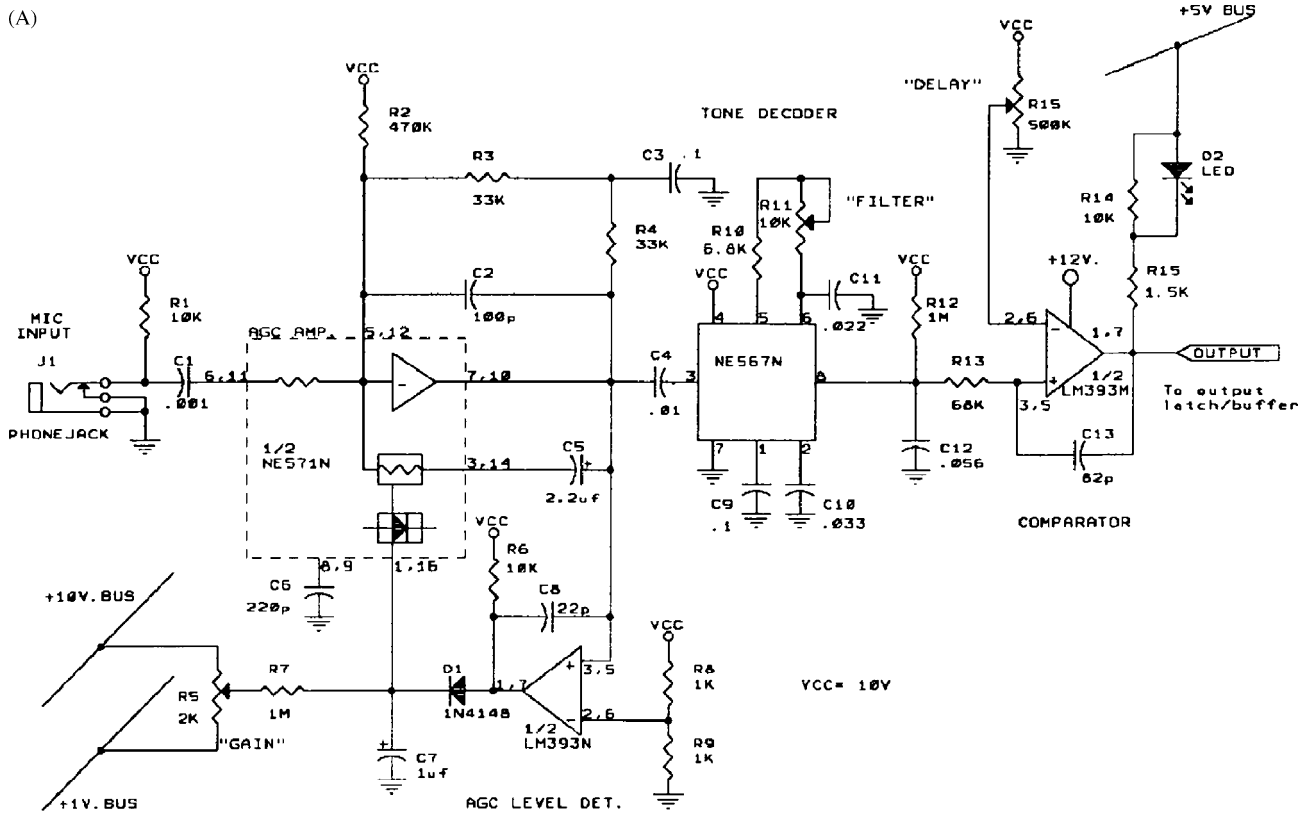


Fig. 2. (A) Circuit diagram of one channel of the monitor. (B) Circuit diagram of the parallel port that constructs the 16-bit data word and is read by the CIO-DIO 24 board in the computer.

specified channel (Fig. 3). The procedure also forces the user to identify the animals on that channel, an identification that will be used later to label the data files for that animal.

2.2.2. Reading the parallel port

When BigBro begins, it first initializes the parallel port, constructs the channel-mask, gets descriptive information about the experiment from the user, and opens a file (#2 in

```

ChMask& = Zero
PRINT "          What channels do you want to monitor? (1..16)"

DO UNTIL Finished
  INPUT " Type a channel-number, or Enter to continue: ", ANSWER$
  IF ANSWER$ = "" AND ChanNum% = ZERO THEN
    BEEP
    PRINT " You must choose at least one Channel. Try again."
  ELSEIF ANSWER$ = "" THEN
    Finished = TRUE
  ELSEIF VAL(ANSWER$) < FRSTCHAN% OR VAL(ANSWER$) > LASTCHAN% THEN
    BEEP
    PRINT " That channel-number can't be right. Try again..."
  ELSE
    ChMask& = ChMask& OR (2 ^ (VAL(ANSWER$) - 1)) ' Construct the Mask
    ChanNum% = ChanNum% + 1 ' Increment the counter
    DO
      INPUT " What animal is that? ID Number (<= 5 characters): ", ID$
      IF ID$ = "" THEN
        BEEP
        PRINT " You must identify the animal somehow. Try again."
      ELSE
        ID(VAL(ANSWER$)) = ID$
        EXIT DO
      END IF
    LOOP
  END IF
  IF ChanNum% > LASTCHAN% THEN Finished = TRUE
LOOP
PRINT "          Channel-mask = "; ChMask&

```

Fig. 3. Procedure in BigBro to construct the logical mask used to hide unused channels.

the code example of Fig. 4) that will hold the data from the experiment. Upon a signal from the user, it starts a procedure to read the port and write the data to the file. This procedure uses the parameter OldWord& to test newly read data words

by using a logical XOR operation, and it uses a logical variable, StopRead, to control execution of the main loop.

The data files produced by BigBro are named automatically by a procedure that uses the date on which the

```

OldWord& = ZERO
StopRead = FALSE
PRINT " Here we go! Press ESC to stop." ' Pressing ESC sets StopRead TRUE

DO UNTIL StopRead
  ' Each pass through the loop begins by latching the data-ports
  ' reading the two 8-bit ports that contain data, and finally releasing
  ' the ports so they can continue to collect data. This code was
  ' provided by the CIO-DIO 24 board's manufacturer.

  ' These steps return two 8-bit words: FirstEight% (channels 1-8)
  ' and NextEight% (channels 9-16).

  ' Construct the new 16-bit dataword, and mask the meaningless bits.
  NewWord& = ((NextEight% * 256) + FirstEight%) AND ChMask&

  ' Look for something that changed
  IF (NewWord& XOR OldWord&) <> ZERO THEN
    ' Write the dataword and the time to file #2,
    ' Col 1 is dataword, Col 2 is TIMER word
    PRINT #2, USING "#####"; NewWord&,
    PRINT #2, USING " #####.###", TIMER

    ' Change the comparison for the next loop.
    OldWord& = NewWord&
  END IF
LOOP
CLOSE #2 ' Close the data file.

```

Fig. 4. Procedure in BigBro to read the data word and check for a change on any channel.

experiment began, a file number, and a specified extension to construct the name. This procedure checks that the name has not already been used; if the name exists, the file number is incremented and the check is repeated until a satisfactory name is constructed.

Using the basic TIMER function to measure time sets the grain of BigBro's temporal resolution. TIMER reads the computer's system clock, and reports the time to the nearest

16.6 ms. The procedure described here can readily detect changes that occur with shorter intervals, but cannot say more precisely when these changes occurred.

2.2.3. Parsing the resulting data file

The data file contains a list of 16-bit words and associated times that encode every change in sound that occurred during the experiment on each channel that was not deliberately

```

FOR I% = 0 TO 15          ' Set up bit-masks
  MASK(I% + 1) = 2 ^ I%
NEXT

FOR I% = First% TO Last%
  IF LEN(ID(I%)) <> ZERO THEN          ' There was something on this channel
    FileNum% = ZERO
    ReadHeader InFileName$, Error$    ' Open DataFile to read data
    MakFilename InFileName$, OutFile$, I%  ' Create TPD Filename

    Done = FALSE
    Starting% = TRUE
    FirstCall% = TRUE
    OldWord& = ZERO
    K% = 1

    DO                                ' Read and parse data file
      IF NOT EOF(1) THEN              ' DataWord1& is 16-bit data,
                                      ' DataWord2! is TIMER word
        INPUT #1, DataWord1&, DataWord2!

        IF Starting% THEN
          IF (DataWord1& AND MASK(I%)) <> ZERO THEN
            ' The first bit in the series was HIGH,
            ' so skip this time and check the next one
          ELSE
            ' The first bit was LOW, so get set
            ' for the first new start of a call.
            Starting% = FALSE
          END IF

          ELSEIF ((DataWord1& AND MASK(I%)) XOR (OldWord& AND MASK(I%))) <> ZERO THEN
            ' Mask BOTH words to look for changes
            ' If this is TRUE, the bit flipped
            ' Keep that time
            TIMEPOINT(K%) = DataWord2!
            OldWord = DataWord1&
            K% = K% + 1
          END IF
        ELSE
          Done = TRUE
        END IF

        IF (K% >= MaxNum% * 2) OR Done THEN          ' Check each time that
                                                    ' the arrays are not full
          CalculateTPD K%, TimeRow%, FirstCall%
          WriteTPD InFileName$, OutFile$, TimeRow%, I%
          REDIM TIMEPOINT(MaxNum% * 2) AS SINGLE, TIME(MaxNum%) AS SINGLE
          REDIM PERIOD(MaxNum%) AS SINGLE, DURATION(MaxNum%) AS SINGLE
          IF FirstCall% THEN FirstCall% = FALSE
          K% = 1
        END IF
      LOOP UNTIL Done
    CLOSE                                ' Close the data file so it can be read sequentially again
  END IF
NEXT

```

Fig. 5. Procedure in Mom to parse each animal's own data from the file produced by BigBro.

masked out. The activity of individual animals, however, must be extracted from the data by examining the particular bit in each data word that corresponds to the channel that recorded that animal's singing. This parsing is accomplished by a second program, "Mom" (BigBro's interpreter), that opens a data file, reads the header that described the experiment and the list of channels recorded, and then reads the list of data words and times to see if and when the animal sang. It creates an output file specific to that channel/animal that is a list of the times at which each pulse of sound began, its duration, and the period from the start of this pulse to the start of the next pulse. These output files are named automatically by a procedure that uses the animal's identification number, originally entered by the user who ran BigBro, and the name of the data file.

A fundamental definition in Mom's algorithm is that a low bit (0) means that there was no sound, and a high bit (1) means that there was sound. The procedure defines sounds as starting when the bit under scrutiny changes

from low to high, and ending when the bit next changes from high to low. Therefore, at the start of each experiment and following each interruption in recording, it is important to see if the bit is low, and if it is not, to wait until it goes low before starting to record. If this is done, then in an ordered list of times, odd-numbered times mark the start of a sound and even-numbered times mark the end of a sound.

Mom's central procedure calls four other procedures: ReadHeader, MakFilename, CalculateTPD, and WriteTPD. ReadHeader is a procedure that opens the data file created by BigBro, reads the description of the experiment and the list of animal identification numbers, and leaves the data file open and ready to be read sequentially. MakFilename constructs the name of the output file from the animal's identification number, CalculateTPD calculates the time, period, and duration associated with each pulse of sound, and WriteTPD writes the list of times, periods, and durations to the output file named for that animal. These procedures

```

SUB CalculateTPD (Row%, TimeRow%, FirstCall%)
    ' Given that TIMEPOINT() becomes ZERO at Midnight (a BASIC TIMER convention),
    ' this interprets ZERO as the start of the next day.
    ' Assume that no TIMEPOINT() is meaningless, since the bit DID flip
    ' Assume odd times are starts of sound (bit high), even times are stops (bit low)
    STATIC Midnight

    C% = 0
    IF FirstCall% THEN Midnight! = 0!

    IF Row% MOD 2 <> ZERO THEN
        Row% = Row% - 1
        ' last bit was high, but omit it since
        ' we might have missed the end of the sound
    END IF

    FOR I% = 1 TO (Row% - 2) STEP 2
        C% = C% + 1
        TIME(C%) = TIMEPOINT(I%) + Midnight!

        IF TIMEPOINT(I%) > TIMEPOINT(I% + 2) THEN
            Midnight! = Midnight! + 86400!
            ' The clock struck 24:00
            ' 86400 is 24 hours in BASIC
            PERIOD(C%) = (TIMEPOINT(I% + 2) + 86400!) - TIMEPOINT(I%)
        ELSE
            PERIOD(C%) = TIMEPOINT(I% + 2) - TIMEPOINT(I%)
        END IF

        IF TIMEPOINT(I%) > TIMEPOINT(I% + 1) THEN
            DURATION(C%) = (TIMEPOINT(I% + 1) + 86400!) - TIMEPOINT(I%)
        ELSE
            DURATION(C%) = TIMEPOINT(I% + 1) - TIMEPOINT(I%)
        END IF
    NEXT I%

    TimeRow% = C%
END SUB

```

Fig. 6. Procedure in Mom to calculate the time, period, and duration of each animal's songs.

were adapted from those first described in Mulloney and Hall (1987).

In Mom's central procedure (Fig. 5), ID() is the list of animal identification numbers called by ReadHeader, Mask() is the list of channel-masks used to isolate data from each channel, and TimePoint(), Time(), Period(), and Duration() are dynamic arrays that hold the results until they are written to the output file. The procedure checks that the arrays are not full, and if they are full, calls WriteTPD and reinitializes the arrays. The logical variable FirstCall% sees that subsequent calls to WriteTPD append additional results to the existing file. The variables First% and Last% control how many times the data file is opened and parsed, and #1 is the identifier of the data file opened by ReadHeader.

2.2.4. Calculating times, periods, and durations

The calculation of these parameters from an ordered list of increasing times would be trivial except for an assumption made in the program and a feature of the Microsoft Basic TIMER function. The program assumes that the animal was

initially silent, so that odd-numbered time-points mark the start of a sound and even-numbered time-points mark the end of a sound. The parsing algorithm in Mom checks that the animal was actually silent at the beginning of the recording session and, if not, waits until it goes silent before starting to calculate anything.

The TIMER function counts upward from midnight until the following midnight, at which time it resets to zero. Therefore, in experiments that run overnight or longer, the list of times at which BigBro detected a change will include these apparent reversals. The procedure CalculateTPD looks for and corrects these features of the list (Fig. 6).

For non-commercial use, the complete source code for BigBro and Mom is available from the authors upon request.

3. Results

We will illustrate the temporal structure of a sample of one cricket's calling song, and the monitor's ability to detect that structure.

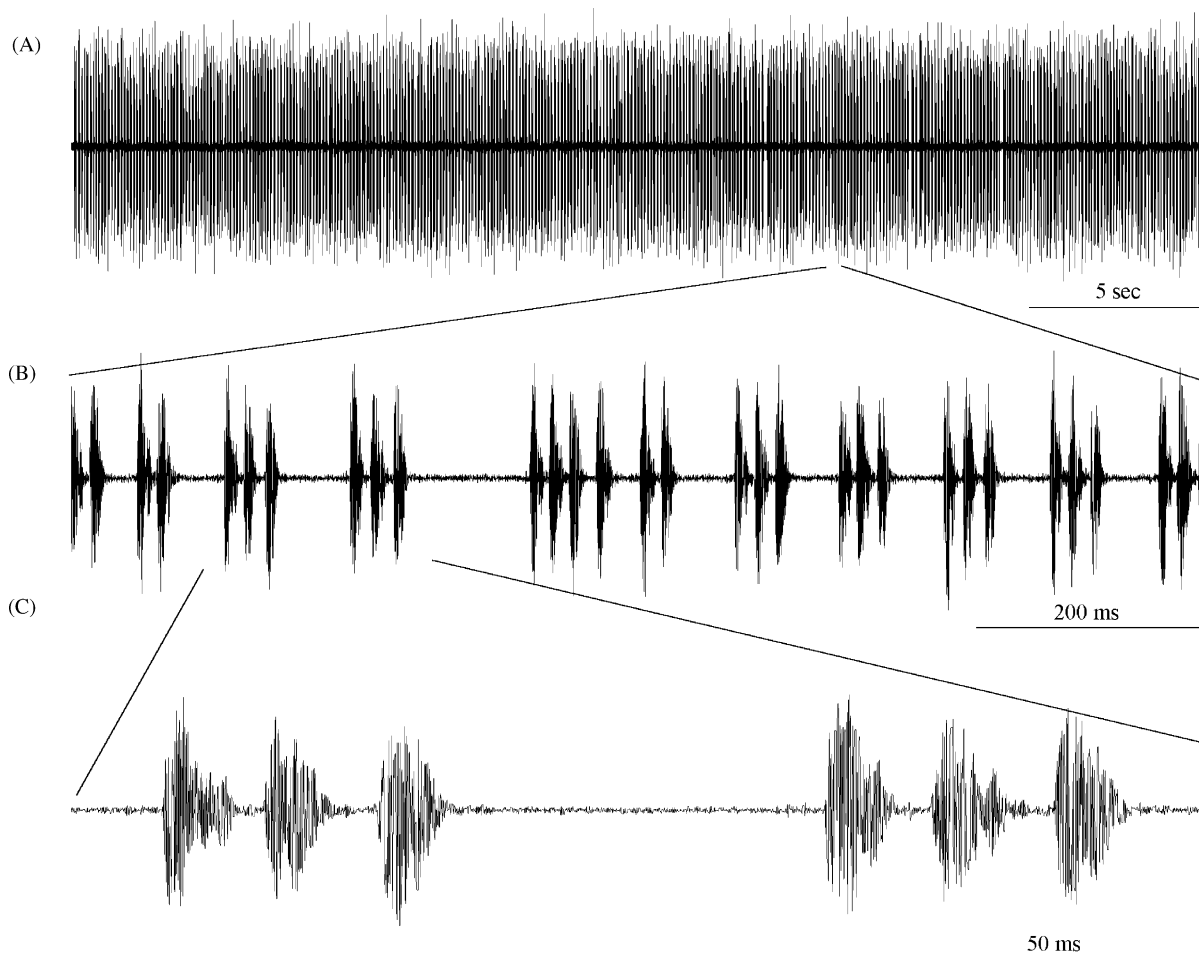


Fig. 7. Temporal structure of a male cricket's calling song. During a bout of singing (A), individual pulses of sound (C) with a carrier frequency of 4.5 kHz are clustered to form chirps (B).

3.1. What cricket songs look like

Male crickets, *G. integer*, make calling songs that vary in duration and in certain internal parameters (Fig. 7). Female crickets choose between different calling males on the basis of these songs, and demonstrate a preference for long-lasting songs that are not interrupted (Hedrick, 1986). In acoustic terms, these songs are constructed of a series of pulses of sound (Fig. 7C). Within each pulse, most energy is at the carrier frequency, about 4.5 kHz (Hedrick and Weber, 1998). In this species, the songs also have a fine-scale structure that females use to distinguish the songs of conspecific males from those of other species (Hedrick and Weber, 1998). Pulses are grouped into clusters, “chirps”, of two or three (rarely four) pulses, separated from other chirps by an interval longer than the interval between pulses within each chirp (Fig. 7B).

3.2. The performance of the monitor

The output from each channel could record the start of each pulse or each chirp, depending on the tuning of the delay potentiometer (Fig. 8). When the delay was set approximately to the duration of a chirp, the output went high at the

start of each chirp, independent of the number of pulses in the chirp (Fig. 8A). When the delay was set approximately to the duration of a pulse of sound, the output went high at the start of each pulse, and gave an accurate record of each pulse’s occurrence. The temporal grain of the system clock in a PC is 16.6 ms (see Section 2), so two pulses that began in less than 16 ms would be recorded as two high events with the same time.

Once the monitor was tuned and started, animals housed in separate containers with food and water and acoustically isolated from one another could be left undisturbed for several days. The records from these experiments had many features that we expected: the males sang mostly in the evening, and were relatively quiet during the day. However, they also had unexpected features, particularly the individual differences in patterns of activity, and in the long durations of calling achieved by some males (Fig. 9). The monitor also proved useful in characterizing the individual reactions of calling males to a disturbance, such as that created by a predator (e.g. a toad). When presented with a predator stimulus (a low-frequency “thumping” on the lid of the boxes that held their cages), males invariably stopped calling, but individual males took different amounts of time to begin calling again (Hedrick, 2000).

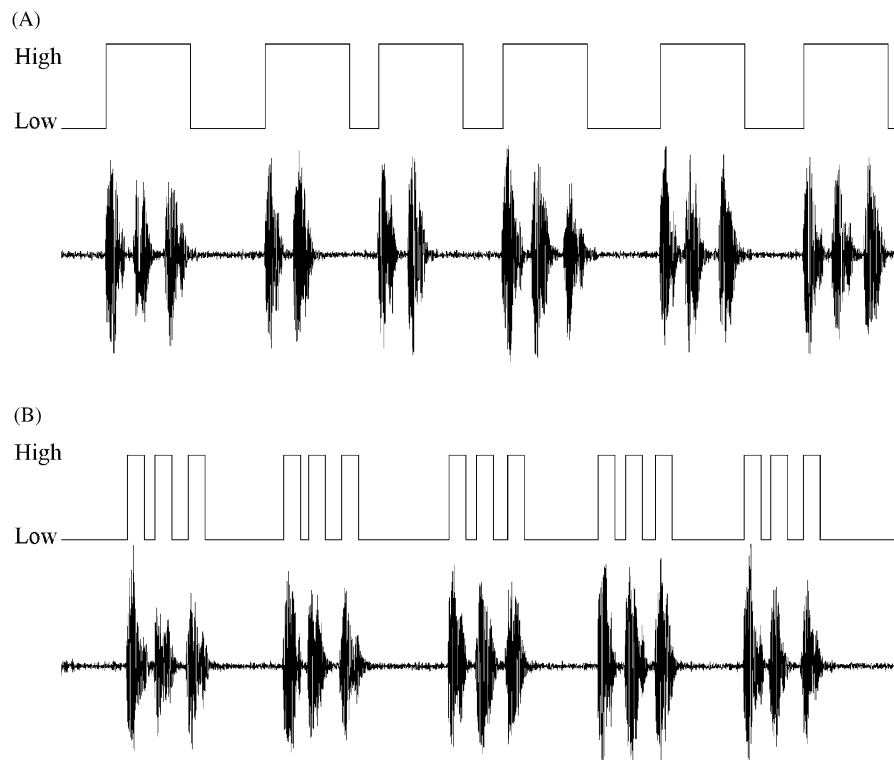


Fig. 8. Examples of the monitor’s output when it is tuned to different delays. Each panel shows a 1.5-s sequence of a cricket call recorded simultaneously with the TTL output of the channel monitoring that cricket. (A) During this sequence, the delay was tuned to 50 ms, and the output recorded chirps. These TTL-high events are each of 50 ms duration, but the first pulse of each chirp triggers a new event, and so the output signals the occurrence of each chirp. (B) During this sequence, the delay was tuned to 10 ms, so the output signals the occurrence of each pulse. All other parameters were the same as in (A).

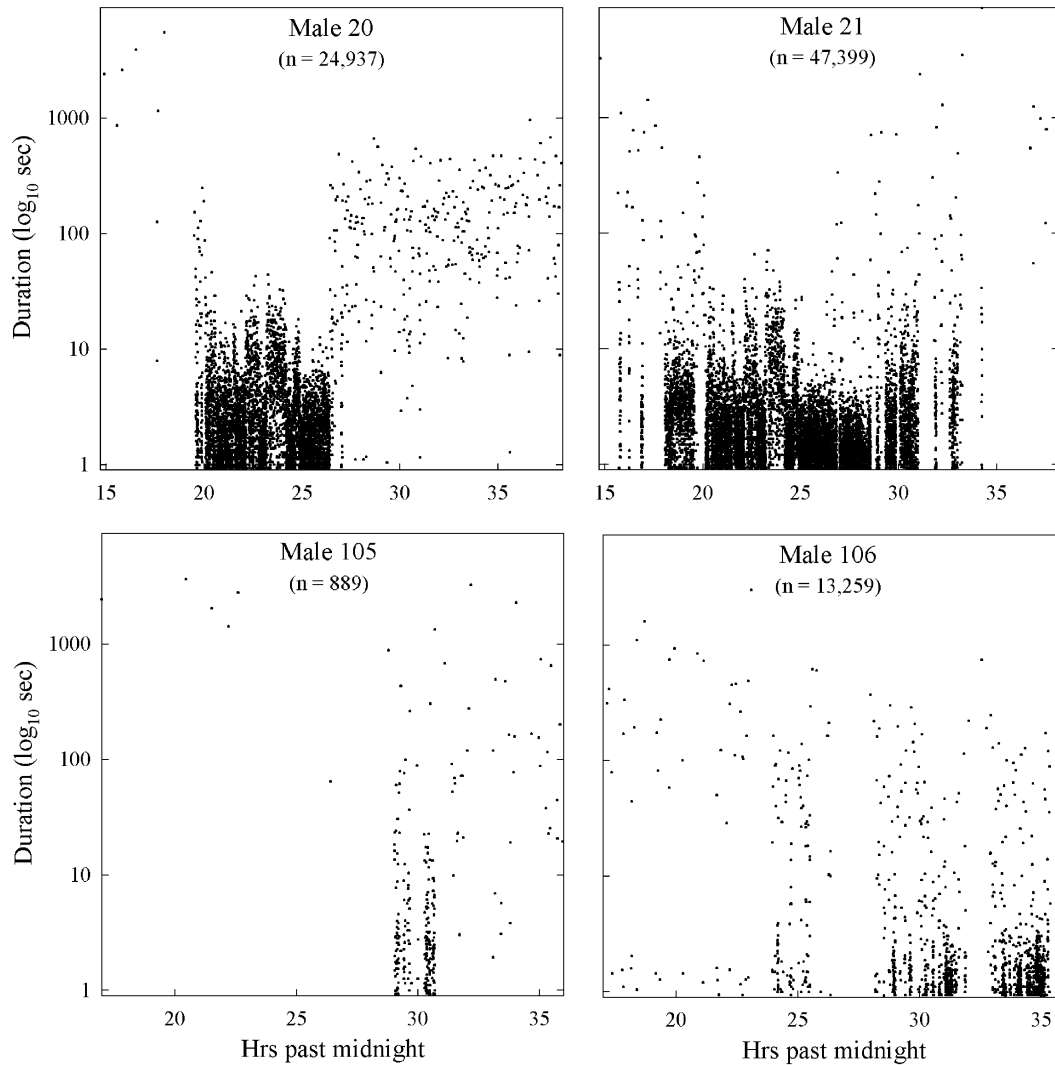


Fig. 9. Durations of calling-bouts from four male crickets, beginning at 15:00h and extending over 24 h. Males 105 and 106 were recorded simultaneously; the rest were recorded on different days.

4. Discussion

The device and supporting software that we have described provides an effective method to measure quantitatively the temporal details of singing in as many as 16 undisturbed animals simultaneously for many days. The practical limit on the length of a recording session arises from the animals' need for fresh food and water. If this were tried by recording the sounds themselves onto tape or disk, the data would be far more voluminous, and the temporal features that were our original interest would still have to be extracted by replaying the tapes repeatedly, once for each animal. In contrast, the data files produced by BigBro are compact, and the parsing algorithm in Mom is efficient. For example, the file for seven males monitored for 48 h was 88 Mb. Some aspects of acoustic communication are necessarily lost, for example loudness and changes in carrier frequency, but even with the comparatively coarse time clock provided

by the Basic TIMER function, BigBro can record every pulse.

4.1. Features of the circuit and software

The device as constructed is designed to be most sensitive to sounds near 4.5 kHz, the carrier frequency of the calling songs of these crickets. By substituting appropriate capacitors (see the NE567 data sheet, Phillips Electronics), this tuning could be altered to suit sounds ranging from 0.1 to >50 kHz. Readers who choose to construct the device should probably substitute the more recent and more widely available SA571 circuit for the NE571 named in the circuit diagram; they are pin-for-pin compatible.

Some applications will require a clock more fine-grained than that provided by the TIMER function. One useful modification of the software would be to use a second board with an accurate programmable clock, perhaps a 9513A, that can

provide times accurate to less than a millisecond, and to substitute for each of the TIMER calls in BigBro a function to read that clock.

Acknowledgements

This research was supported by NSF Grant 0076484 to A.V. Hedrick, and by NSF Grants IBN 97281 and 0091284 to B. Mulloney.

References

- Bertram S, Johnson L. An electronic technique for monitoring the temporal aspects of acoustic signals of captive organisms. *Bioacoustics* 1998;9:107–18.
- Gerhardt HC, Huber F. *Acoustic communication in insects and anurans: common problems and diverse solutions*. Chicago: University of Chicago Press; 2002.
- Hedrick AV. Female preferences for male calling bout duration in a field cricket. *Behav Ecol Sociobiol* 1986;19:73–7.
- Hedrick AV. Female choice and the heritability of attractive male traits—an empirical study. *Am Nat* 1988;132:267–76.
- Hedrick AV, Weber T. Variance in female responses to the fine structure of male song in the field cricket, *Gryllus integer*. *Behav Ecol* 1998;9:582–91.
- Hedrick AV. Crickets with extravagant mating songs compensate for predation risk with extra caution. *Proc R Soc Lond B: Biol Sci* 2000;267:671–5.
- Hedrick AV, Perez D, Lichti N, Yew J. Temperature preferences of male field crickets (*Gryllus integer*) alter their mating calls. *J Comp Physiol A Neuroethol Sens Neural Behav Physiol* 2002;188:799–805.
- Jennions MD, Petrie M. Variation in mate choice and mating preferences: a review of causes and consequences. *Biol Rev* 1997;72:283–327.
- Kidder GW, Sakaluk SK. Simple and inexpensive electronic recording device for automatic recording and analysis of insect acoustical activity. *Florida Entomol* 1989;72:642–9.
- Kokko H, Brooks R, Jennions MD, Morely J. The evolution of mate choice and mating biases. *R Soc Lond* 2003;270:653–64.
- Mulloney B, Hall WM. The PD programs: a method for the quantitative description of motor patterns. *J Neurosci Methods* 1987;19:47–59.
- Tallamy DW, Powell BE, McClafferty JA. Male traits under cryptic female choice in the spotted cucumber beetle (Coleoptera: Chrysomelidae). *Behav Ecol* 2002;13:511–8.