

Susan M. Bertram^{*}, Luke A. Johnson, Jerome Clark, Carmenlita Chief
School of Life Sciences, Arizona State University, Tempe, Arizona, USA 85287-4501

An electronic acoustic recorder for quantifying total signaling time, duration, rate and magnitude in acoustically signaling insects

Received 05.11.2004, published 03.12.2004

APPENDIX I – DATA ACQUISITION SOFTWARE

Borland C software designed to run one or two EARs. The software communicates between the EAR(s) and the computer via one DIO24db/ct digital interface board.

```
/* ears.c */
/* EARs driver software*/
/* by Luke A. Johnson */
/* Requires: */
/* DIO24db/ct */
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <conio.h>
#include <dos.h>
#include <bios.h>
#define IRQ 5
```

```
unsigned char config, CurMic=15, Sample, status;
unsigned long LineCount = 0;
unsigned int SumA[16][8];
unsigned int SumB[16][8];
unsigned int Xlat[16]={7,15,6,14,5,13,4,12,3,11,2,10,1,9,0,8};
unsigned int SumCnt=7;
unsigned int i=0;
float NextUpdate=0;
unsigned short address = 0xEC40;
```

```
typedef enum {FALSE,TRUE} boolean;
```

```
void main()
{
    char filename[20];
```

^{*}corresponding author, e-mail: sbertram@asu.edu

```

FILE *outfile;
struct time t;
struct date d;
gettime(&t);
getdate(&d);

/* Initialize Digital Ports */
/* Port A: Output - Microphone select
   Port B: Input - Microphone sample
   Port C: Input - Microphone sample */
outportb(address+0x03,0x8b);

/* Initialize timer to 64Hz Square Clock */
outportb(address+0x13,0x37);
outportb(address+0x10,0x00); /* 0000 bcd */
outportb(address+0x10,0x00);
outportb(address+0x13,0x77);
outportb(address+0x11,0x05); /* 0005 bcd */
outportb(address+0x11,0x00);
outportb(address+0x13,0xb7);
outportb(address+0x12,0x25); /* 3125 bcd */
outportb(address+0x12,0x31);

sprintf(filename,"S%02d%02d.ccc",d.da_mon,d.da_day);

outfile = fopen(filename, "w");
fprintf(outfile,"%02d:%02d:%02d %d/%d/%d\n\n",
        t.ti_hour, t.ti_min, t.ti_sec,d.da_mon,d.da_day,d.da_year);
fprintf(outfile,"\nTime  A B C D E F G H A B C D E F G H A B C D E F G H A B
C D E F G H A B C D E F G H A B C D E F G H A B C D E F G H A B C
D E F G H A B C D E F G H A B C D E F G H A B C D E F G H A B C D
E F G H A B C D E F G H A B C D E F G H");
fprintf(outfile,"\n      A A A A A A A A A A A A A A A A A A A A A A A A
A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A");
fprintf(outfile,"\n      0 1 2 3 4 5 6 7 8 9 a b c d e f 0 1 2 3 4 5 6 7 8 9 a b c d e f 0 1 2
3 4 5 6 7 8 9 a b c d e f 0 1 2 3 4 5 6 7 8 9 a b c d e f 0 1 2 3 4 5 6 7 8 9 a b c d e f 0 1 2 3 4 5 6
7 8 9 a b c d e f 0 1 2 3 4 5 6 7 8 9 a b c d e f 0 1 2 3 4 5 6 7 8 9 a b c d e f");
fprintf(outfile,"\n%02d:%02d:%02d ",t.ti_hour, t.ti_min, t.ti_sec);

clrscr();
printf("Electronic Acoustic Recorder data file (%s)\nStarted %02d:%02d:%02d
%d/%d/%d\n\n",filename,
        t.ti_hour, t.ti_min, t.ti_sec,d.da_mon,d.da_day,d.da_year);
gotoxy(21,5);
printf("Box A");
gotoxy(57,5);
printf("Box B");

```

[illegible]

```

SumA[7][SumCnt]=0;
SumA[8][SumCnt]=0;
SumA[9][SumCnt]=0;
SumA[10][SumCnt]=0;
SumA[11][SumCnt]=0;
SumA[12][SumCnt]=0;
SumA[13][SumCnt]=0;
SumA[14][SumCnt]=0;
SumA[15][SumCnt]=0;
SumB[0][SumCnt]=0;
SumB[1][SumCnt]=0;
SumB[2][SumCnt]=0;
SumB[3][SumCnt]=0;
SumB[4][SumCnt]=0;
SumB[5][SumCnt]=0;
SumB[6][SumCnt]=0;
SumB[7][SumCnt]=0;
SumB[8][SumCnt]=0;
SumB[9][SumCnt]=0;
SumB[10][SumCnt]=0;
SumB[11][SumCnt]=0;
SumB[12][SumCnt]=0;
SumB[13][SumCnt]=0;
SumB[14][SumCnt]=0;
SumB[15][SumCnt]=0;
gettime(&t);
fprintf(outfile, "\n%2d:%02d:%02d ",
        t.ti_hour, t.ti_min, t.ti_sec);
if (t.ti_min >= NextUpdate) {
    NextUpdate = t.ti_min + 30;
    if (NextUpdate >= 60) {
        NextUpdate -= 60;
    }
    fclose(outfile);
    outfile = fopen(filename, "a");
}
} else
/* Select next mic */
outportb(address+0x00, Xlat[CurMic]);

/* Wait for next clock edge */
outportb(address+0x13, 0xe8); /* get status of counter A2 */
Sample = inportb(address+0x12) & 0x80;
do {
    outportb(address+0x13, 0xe8); /* get status of counter A2 */
    status = inportb(address+0x12) & 0x80;
} while (status == Sample);

/* Read Microphone */

```

```

        Sample = inportb(address+0x01);
        if (Sample&0x80) SumA[CurMic][1] += 1;
        if (Sample&0x40) SumB[CurMic][1] += 1;
        if (Sample&0x20) SumA[CurMic][3] += 1;
        if (Sample&0x10) SumB[CurMic][3] += 1;
        if (Sample&0x08) SumA[CurMic][5] += 1;
        if (Sample&0x04) SumB[CurMic][5] += 1;
        if (Sample&0x02) SumA[CurMic][7] += 1;
        if (Sample&0x01) SumB[CurMic][7] += 1;
        Sample = inportb(address+0x02);
        if (Sample&0x80) SumA[CurMic][0] += 1;
        if (Sample&0x40) SumB[CurMic][0] += 1;
        if (Sample&0x20) SumA[CurMic][2] += 1;
        if (Sample&0x10) SumB[CurMic][2] += 1;
        if (Sample&0x08) SumA[CurMic][4] += 1;
        if (Sample&0x04) SumB[CurMic][4] += 1;
        if (Sample&0x02) SumA[CurMic][6] += 1;
        if (Sample&0x01) SumB[CurMic][6] += 1;
    } while (kbhit()==0 || CurMic<15);

    gettimeofday(&t);
    getdate(&d);
    fprintf(outfile, "\n\nEnded %2d:%02d:%02d %d/%d/%d\n\n",
            t.ti_hour, t.ti_min, t.ti_sec, d.da_mon, d.da_day, d.da_year);
    printf("\n\nEnded %2d:%02d:%02d %d/%d/%d\n\n",
            t.ti_hour, t.ti_min, t.ti_sec, d.da_mon, d.da_day, d.da_year);

    fclose(outfile);
} /* End main program */

```