Consumer Appliance, Widget, Gadget

Mosquito!

Author: *Mark Walker*
Hillsboro, Oregon
email: mmwatcgi@aracnet.com

INTRODUCTION:

There have been many insect repellers used over time, with very few giving good results.

Liquids have personal and environmental impacts and many of the electronic versions are either very bothersome or not very effective.

This extremely simple circuit and approach is actually based on scientific data having to do with the personal life of the common mosquito. The natural enemy of the mosquito is the very fast and effective dragon fly, which can pluck a mosquito from the air before it can hide. Over many generations this fact has been imprinted on these nasty little critters and any time they hear anything that sounds like their mortal enemy, they immediately hide, land, stop moving, freeze up and almost stop breathing to avoid detection.

This means that if we can realistically duplicate the sound of the dragon fly wings in motion, we can convince the ubiquitous mosquito, at least any within hearing distance, to go hide and to not bother us. This works very effectively, is environmentally friendly and in use, is almost unnoticeable, since the dragon fly wings we are simulating are not very noisy to begin with.

APPLICATION OPERATION:

This circuit is extremely simple and very low power. The crystal earphone used with those old crystal radio kits works perfectly in this application, since it draws essentially no power and makes perfect ticking sounds with very simple drive. Note that I am not speaking of the transistor radio earphone that can have impedances as low as 4 ohms, but the crystal earphone. If that is hard to find, any small piezo noise maker can also be used. The clock of the PIC12C508 in this circuit is a low frequency RC circuit since crystals would draw more power than necessary. The RC circuit can be set to a lower frequency than even the typical watch crystal frequency to reduce the total drain to a level that allows months or even years of operation from a lithium or alkaline battery, making the entire unit extremely compact and convenient.

The wing beat of the typical dragon fly appears as bursts of clicks at about 30 Hz as the dragon fly spurts around chasing its dinner. Tests have shown that even a continuous thirty Hz beat is very effective at convincing mosquitoes to hide, so this implementation that creates bursts of about thirty Hz approximately two thirds of a second in length, with about a third second between them, is more effective yet. However, for the more energetic, improvements would include a slight randomizing of both the frequency and the length of the bursts, with the frequency slightly higher at the end of each burst than at the beginning.

Note: The timers are left essentially separate to allow for easy changes to this code to randomize a bit on each number. Doing so makes the result even less bothersome to listen to, as not quite repetitious is less annoying then exactly repetitious. Normal noises will mask this under typical use.

This project can easily lead into discussions of various techniques to produce random or pseudo-random numbers within the framework of a fun and practical little project.

APPENDIX A: SOURCE CODE:

The wing beat portion of the code is shown here, since the setup is straight forward and repetitiously available.

```
Set up notes for the ‘508;

Set up for external RC, weak pullups off, internal MCLR, watch dog timer off, no wake up on change.
Set rest of lines as outputs.
Set registers (EQU) for:

GPIO
WINGTIME
BURSTCNT
PAUSECNT

;Wing beat code
             movlw      .10             ;loads for number in burst
             movwf      BURSTCNT        ;

LOOP1ST      bsf        GPIO,GP2        ;does a tic sound
             movlw      .9              ;preloads counter
             movwf      WINGTIME        ;

LOOP2ND      decfsz     WINGTIME,F      ;runs loop
             goto       LOOP2ND         ;

             bcf        GPIO,GP2        ;does next tic sound
             movlw      .9              ;preloads counter again
             movwf      WINGTIME        ;

LOOP3RD      decfsz     WINGTIME,F      ;runs this loop
             goto       LOOP3RD         ;
             decfsz     BURSTCNT,F      ;counts number in burst
             goto       LOOP1ST

             movlw      .10             ;reloads this counter
             movwf      BURSTCNT        ;

             movlw      .100            ;loads pause timer
             movwf      PAUSECNT        ;

LOOP4TH      decfsz     PAUSECNT,F      ;runs quiet pause
             goto       LOOP4TH         ;

             goto       LOOP1ST         ;does it all again, and again
```