ARRL Periodicals Archive - Search Results

A membership benefit of ARRL and the ARRL Technical Information Service

ARRL Members: You may print a copy for personal use. Any other use of the information requires permission (see Copyright/Reprint Notice below).

Need a higher quality reprint or scan? Some of the scans contained within the periodical archive were produced with older imaging technology. If you require a higher quality reprint or scan, please contact the ARRL Technical Inforamtion Service for assistance. Photocopies are $3 for ARRL members, $5 for non-members. For members, TIS can send the photocopies immediately and include an invoice. Non-members must prepay. Details are available at www.arrl.org/tis or email photocopy@arrl.org.

***QST* on DVD:** Annual DVDs are available for recent publication years. For details and ordering information, visit www.arrl.org/benefits.

Non-Members: Get access to the ARRL Periodicals Archive when you join ARRL today at www.arrl.org/join. For a complete list of membership benefits, visit www.arrl.org/benefits.

QST Issue: Oct 1998

Title: Creating a PIC-Based IDer (sidebar to Using PIC Microcontrollers in Amateur Radio Projects)

Click Here to Report a Problem with this File

Table 2

An Assembly Language Program for a PIC Morse Code Generator

```
            list    p=16f84
              radix   hex
              __config  0x3FF1
dahlen      equ     d'99'  ;  <==controls code speed..
                            ;increase
ditlen      equ     d'33'  ;  <==(though not over 255) for
            org     0x000       ;slower code, decrease for
            MOVLW   00          ;faster code. Make sure the
            MOVWF   0A          ;top number is three times
                                ;the bottom.
            GOTO    start
            NOP
            NOP
time        MOVF    10,W
            BTFSC   03,2
            GOTO    endtime
uptop       MOVLW   01
            MOVWF   0D
upagn       CLRF    0C
doagn       DECFSZ  0C,F
            GOTO    doagn
            DECFSZ  0D,F
            GOTO    upagn
            MOVLW   4A
            MOVWF   0C
upone       DECFSZ  0C,F
            GOTO    upone
            DECFSZ  10,F
            GOTO    uptop
endtime  RETLW  00

dah         MOVLW   01      ;subroutine to do a dah
            MOVWF   0E
agn3dah MOVLW           dahlen
            SUBWF   0E,W
            BTFSC   03,0
            GOTO    enddah
            BSF     06,1    ;turn on pin B1
            MOVLW   01
            MOVWF   10
            CALL    time    ;wait 1 millisecond
            BCF     06,1    ;turn off pin B1
            MOVLW   01
            MOVWF   10
            CALL    time    ;wait 1 millisecond
            MOVF    0E,W
            INCF    0E,F
            GOTO    agn3dah ;loop up to do it again.
enddah      MOVLW   3C      ;add a small delay
            MOVWF   10
            CALL    time
            RETLW   00

dit         MOVLW   01      ;subroutine to do a dit
            MOVWF   0E
agn3dit     MOVLW   ditlen
            SUBWF   0E,W
            BTFSC   03,0
            GOTO    enddit
            BSF     06,1    ;turn on pin B1
            MOVLW   01
            MOVWF   10
            CALL    time    ; wait 1 millisecond
            BCF     06,1    ;turn off pin B1
            MOVLW   01
            MOVWF   10
          CALL      time    ;wait 1 millisecond
            MOVF    0E,W
            INCF    0E,F
            GOTO    agn3dit ;loop up and do it again.
enddit      MOVLW   3C      ;add a small delay
            MOVWF   10
            CALL    time
            RETLW   00

lspace              0xB4    ;subroutine to make a
            MOVLW           ;letter space
            MOVWF   10
            CALL    time
            RETLW   00

start       CLRF    04      ;MAIN PROGRAM
                            ;STARTS HERE
            MOVLW   0xFD
            TRIS    6

top         CALL    dit     ;modify this code to
                            ;xmit the
            CALL    dah     ;CW you want to send
            CALL    dah
            CALL    lspace  ;lspace is a pause for
                            ;the space between
            CALL    dit     ;letters. Include it after
                            ;each letter.
            CALL    dit     ;As it appears here, the
                            ;IDer will
            CALL    dah     ;transmit W2FS
            CALL    dah
            CALL    dah
            CALL    lspace
            CALL    dit
            CALL    dit
            CALL    dah
            CALL    dit
            CALL    lspace
            CALL    dit
            CALL    dit
            CALL    dit

            MOVLW   01
            MOVWF   0E
                            ;this code programs the
                            ;delay between IDs.
loop        MOVLW   d'150'  ;<= 4 x this number =
                            ;number of seconds
            SUBWF   0E,W    ; between IDs when
                            ;using a 4 MHz resona-
                            ;tor.
            BTFSC   03,0
            GOTO    bottom
            MOVLW   10
            MOVWF   0F
again       MOVLW   0xFA
            MOVWF   10
            CALL    time
            DECFSZ  0F,F
            GOTO    again
            MOVF    0E,W
            INCF    0E,F
            GOTO    loop
bottom      GOTO    top
            END
```

Creating a PIC-Based IDer

Table 2 contains the assembly-language code that makes a 16F84 PIC generate Morse code. The code is designed to work with the circuit of Figure 1, using a ceramic resonator or a 3.5795-MHz colorburst crystal. Audio output is obtained at pin RB1. The program is designed to be easy to modify to insert your choice of call sign, Morse code speed and transmit interval. To change the Morse code speed, alter the lines labeled "dahlen" and "ditlen." To change the ID interval, change the number between the apostrophes in the line labeled "loop." Change the lines that start with the label "top" to change the call sign (or other text) that is to be transmitted by the chip. Follow the pattern shown in the program and make sure you put an "lspace" after the end of each letter. If you plan to transmit text other than just a call sign, you can do this by adding extra "lspace" instructions to create longer delays between the words. When altering this code, be sure to keep the label "top" on the same line as the first code element to be sent.

If you type in the program yourself, you may leave out the comments on each line by dropping the semicolon and the text that follows it. Blank lines can either be left in or deleted as you prefer. Alternatively, you can download this file from the ARRL's FTP site and edit it. The program may look long, but it only uses about 10% of the program capacity of a 16F84.