

# Un keyer per CW senza ...fronzoli

di Stefano Barbanti

IV3LZQ



## Introduzione

Questo articolo presenta un semplice tasto elettronico per CW (keyer) realizzato intorno al microcontrollore PIC 16F84A, che come circuiteria trae ispirazione (ed è in parte compatibile per il programma del micro) con l'ottimo keyer presentato da IW4AA, PierLuigi Felletti sul numero di marzo 2002 di Radiokit.

La realizzazione attuale è mirata ad avere semplicissimo keyer, senza memorie né fronzoli strani, dotato però della manipolazione secondo modo IAMBIC B (la pigrizia cosa fa fare...!) e con alcune caratteristiche, a mio avviso,

desiderabili per l'utilizzo in spedizioni o field day.

Il tutto è nato dalla ricerca di un keyer da affiancare all'ormai immancabile PC con l'interfaccia per CW e il relativo Logger per il contest o per la spedizione, con collegamento in parallelo sul circuito di keying CW del ricetrasmittitore tramite jack audio stereo a due ingressi e relativo adattatore da jack grande a piccolo.

Il keyer con il tasto "in parallelo" al PC per generare il CW da garanzia di continuità nelle operazioni in caso di PC temporaneamente "piantato" o per "diligamenti" in CW con il corrispondente.

## Caratteristiche del Keyer

La semplicità e la praticità di uso e la pigrizia, unite all'esperienza di utilizzo in field day e mini spedizioni hanno guidato l'individuazione delle seguenti caratteristiche:

- Manipolazione IAMBIC in modalità B.
- Interruttore di Tuning per il trasmettitore.
- Interruttore per la selezione della funzionalità delle palette (Paddle), ossia per poter scegliere se generare i punti con la palette di sinistra e le linee con quella di destra o viceversa, e con selezione effettuabile al volo.
- Assenza di memorie e/o strampalati interruttori per memorizzazioni/invio/configurazione.
- Massima semplicità circuitale.
- Alimentazione da batterie (3 x 1,5V stilo o ministilo o pila piatta da 4,5V).
- Assenza di sidetone.
- Programma del micro con la massima semplicità possibile e facilmente modificabile (per i più volenterosi !)

## IAMBIC Mode B:

La scelta di scrivermi da solo il programma è nata proprio per soddisfare il primo punto, ossia la manipolazione IAMBIC in modalità B, infatti gli altri keyer che ho trovato in giro e su questa rivista non la prevedevano. Ma vediamo cosa prevede tale modalità: quando entrambe le palette



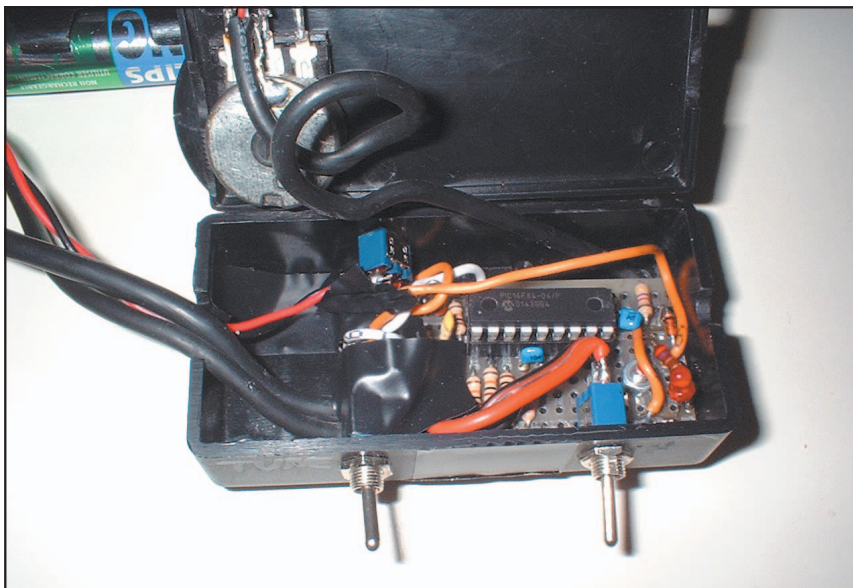


Fig. 3

del paddle sono attivate il keyer genera una sequenza alternata di punti/linee o viceversa a seconda della sequenza di pressione da parte dell'operatore, e al rilascio delle stesse palette genera un punto o una linea a seconda che l'ultimo carattere generato fosse una linea o un punto rispettivamente; dal punto di vista pratico, questa caratteristica che può sembrare apparentemente strampalata, comporta una facilità e una comodità estrema (la pigrizia di cui all'inizio...) di manipolazione, il più dei caratteri sono ottenibili con una sola e opportuna pressione delle palette.

Tale "caratteristica" di manipolazione è solitamente integrata nei keyer "entrocontenuti" negli apparecchi delle ultime generazioni (ad es. su IC 706, IC756, IC718 e sul FT1000, solo per citarne alcuni da me provati) dotati di micro che li controlla e fino a qualche decina di anni fa era una prerogativa quasi esclusiva dei keyer realizzati con il famoso circuito integrato della Curtiss.

Comunque ho trovato degli schemi su degli Handbook della ARRL (ad es. su quello del 1994) riportanti schemi con la medesima caratteristica IAMBIC B sia con il sopracitato integrato che realizzati con componenti di-

creti con quattro o cinque circuiti integrati "digitali".

Tuttavia ormai tale integrato non è più facilmente reperibile, e con l'ormai abbordabile costo dei microcontrollori PIC, la scelta per la realizzazione è stata inevitabile.

Dotazioni e caratteristiche HW del keyer:

- L'interruttore di "sintonia" o "Tune" è davvero comodo, per la ricerca di accordi con il lineare o con l'accordatore di antenna.

- L'interruttore per il cambio di "mano" di manipolazione è sicuramente utile per le operazioni in multioperatore, dove ciascuno dei grafisti ha una sua diversa abitudine per la posizione dei punti e delle linee del keyer, il cambio può avvenire al volo ossia senza dover spegnere e riaccendere il circuitino; è utile per provare dei tasti "altrui", difatti non ho mai trovato due tasti cablati nello stesso modo per la posizione punti/linee.

- Potenziometro per la regolazione della velocità: variando la frequenza di clock del micro varia di conseguenza la velocità di manipolazione, ed è davvero comodo.

- Circuito di keying immune alla RFI.

- Interruttore per l'alimentazione "salvabatterie".

- Alimentazione con tre comuni pile stilo o ministilo da 1,5V in serie o da batteria piatta da 4,5V.

### Lo schema elettrico

Lo schema elettrico è riportato in Fig. 2, e dopo le spiegazioni di cui sopra non necessita di molti commenti.

Il tutto ruota intorno al microcontrollore Microchip PIC 16F84A o equivalente, opportunamente programmato e ad una manciata di componenti reperibili ovunque.

Ovviamente dovrete poi essere dotati di un programmatore di PIC e del software su PC per programmare il micro; di schemi di programmatori se ne trovano su Internet in gran quantità, o potrete richiedere l'ausilio di un vostro amico già attrezzato in tal senso, mentre il programma per il keyer lo potrete scaricare dal sito Internet di Radiokit, come descritto più avanti.

Se avete difficoltà nel reperire il micro o a programmarvelo da soli, posso fornirvi una copia già programmata, previo rimborso spese.

### La realizzazione

Vista la semplicità e l'esiguità del numero dei componenti si consiglia vivamente di realizzare il tutto con una basetta millefori con il passo per gli integrati e si raccomanda di dotare il micro di zoccolo (9+9 pin) per ovvi motivi, si veda la Fig.2

Prima di inserire il micro, una volta terminato il montaggio, applicate l'alimentazione verificando l'assenza di corti circuiti e con l'ausilio di un tester, misurate e verificate la correttezza della polarità e la presenza della tensione di alimentazione (min. 2.0 ... max. 5.5V, mi raccomando!) sui piedini dello zoccolo dell'integrato, Pin 14 (+Vdd) e Pin 5 (0V) aiutandovi con lo schema di Fig.1.

Vi consiglio inoltre di provare anche a connettere il tasto e a verificare che premendo la palette





Fig. 4

dei punti o quella delle linee, la tensione sul piedino relativo vada a zero con il tasto premuto, Pin 2 e Pin 3; lo stesso dicasi per i vari interruttori di sintonia e di cambio mano di manipolazione provandoli aperti e chiusi; difatti grazie alle resistenze di pull-up se il relativo ingresso non è attivato lo stesso è tenuto al valore dell'alimentazione dalla resistenza stessa.

Si può provare prima anche il circuito di keying CW per la radio: collegate all'ingresso CW l'uscita TX\_CW\_key e TX\_CW\_gnd del keyer, con Jack Stereo o mono da 3,5mm (per il più delle radio) e cavo schermato, e con la radio in modo CW, applicando con un ponticello la tensione di alimentazione al pin 18 il ricetrasmittitore dovrà andare in trasmissione.

A questo punto, verificata la correttezza del cablaggio e del montaggio, togliere l'alimentazione e inserire il micro, mi raccomando di rispettarne il corretto verso di inserzione, pena la distruzione dell'integrato stesso.

Il circuito è dunque pronto all'uso; accendendo, con il micro inserito e premendo una delle palette la radio deve passare in trasmissione emettendo una serie di punti o di linee.

Se il tutto va' in trasmissione, continua, come accendete, niente paura, significa solo che avete lasciato inserito l'interruttore di "Tune"; riportandolo nella posizione di aperto la radio dovrebbe ritornare in ricezione.

Verificate poi se la posizione per generare punti e linee per il tasto è quella a cui siete abituati, viceversa attivate l'interruttore di cambio mano.

Il tutto poi troverà posto in un piccolo contenitore plastico, si veda la Fig.4.

## Il programma del 16F84A

Questa parte è dedicata ai più curiosi, spero numerosi, che desiderano conoscere come può

Fig. 5

```

; ***** MAIN LOOP *****
NEXT                ; MAIN RETURN POINT FOR MAIN LOOP
BTFF ; RA_DOT_POS  ; CHECK IF SWITCH ON RA2, PIN 1 IS CLOSED FOR CHOOSING DOT & DASH POSITION
GOTO MAIN_B        ; DASH/DOT CONFIGURATION CHOSEN SO EXECUTE MAIN_B OTHERWISE EXECUTE MAIN_A

; ***** MAIN LOOP FOR PADDLE DOT/DASH INPUTS (MAIN_A) *****
MAIN_A              ; CONFIGURATION WITH RA3 FOR DOT AND RA4 FOR DASH

BTFFS RA_TUNE_SW   ; IS THE TUNE SWITCH PRESSED ? ; Bit Test f, Skip if Set = BTFFS, if not pressed is high due to pullup res.
GOTO TUNE          ; YES, SO EXECUTE TUNE
BCF RA_KEY        ; UNKEY TRANSMITTER

BTFFS FLAG_DAH_MEM ; WAS THE DAH MEMORY TRIGGERED ?
GOTO DASH         ; YES, SO SEND A DASH

BTFFS FLAG_DIT_MEM ; WAS THE DIT MEMORY TRIGGERED ?
GOTO DOT         ; YES, SO SEND A DOT

BTFFS RA_DAH_SW   ; IS THE DAH PADDLE PRESSED ?
GOTO DASH        ; YES, SO SEND A DASH

BTFFS RA_DIT_SW   ; IS THE DIT PADDLE PRESSED ?
GOTO DOT         ; YES, SO SEND A DOT

GOTO NEXT        ; LOOP UNTIL NEXT PADDLE OR TUNE SWITCH PRESS

; ***** END OF MAIN LOOP FOR PADDLE DOT/DASH INPUTS (MAIN_A) *****

```

In base alla posizione di S2 esegue due loop diversi, con la posizione punto/linea scambiata per il tasto.

Se la posizione di S2 è per la configurazione punto/linea (RA3/RA4) esegue MAIN\_A

Se S1 è chiuso, esegue TUNE (per mandare in trasmissione fisso la radio)

Se le flag per le memorie di carattere sono state attivate esegue una linea (DASH) o un punto (DOT), realizzando il modo IAMBIC B di keying.

essere realizzato un programma per realizzare un keyer, usando il linguaggio assembler (MPASM) per i microcontrollori PIC.

Il programma in oggetto, l'M-PASM, può essere scaricato gratuitamente dal sito della Microchip, così come il datasheet del 16F84A, che tra l'altro include anche tutti i dettagli sul linguaggio assembler del micro stesso, con l'elenco e la descrizione di tutte le istruzioni.

Vedasi appunto <http://www.microchip.com>.

Ovviamente esistono molti modi, sicuramente molto più eleganti e intelligenti del mio, per scrivere un programma per un microcontrollore in assembler (linguaggio macchina), ma sicuramente vanno a discapito della semplicità e della didattica. Ho trovato dei programmi decisamente complicati, che realizzano delle implementazioni con macchine a stati logici, di un OM tedesco, oppure bellissimi programmi per la gestione delle memorie e del sidetone quali quello di IW4AA, per citarne uno su tutti, a cui vi rimando per ulteriori approfondimenti.

Per semplicità vi riporto solo un pezzetto del programma in Assembler, vedi Fig.5, tanto per invogliarvi ad andare a visionare il listato completo; vi rimando quindi ai copiosi commenti del listato in assembler (file con estensione .asm) scaricabile dal sito di Radiokit, <http://www.edizionicec.it> nell'area DOWNLOAD, insieme al programma già compilato pronto per essere trasferito nel microcontrollore (file con estensione .hex).

## Conclusioni

Come al solito spero abbiate trovato di vostro gradimento e interesse l'articolo, e possiate usare numerosi il keyer in radio per tenere viva la nobile arte del CW.

Spero che l'utilizzo della manipolazione IAMBIC "B" possa rendere ancora più piacevole il vostro trasmettere in CW, vi abituerete in breve e poi non vi sembrerà possibile più farne a

meno, provare per credere.

All'inizio magari vi capiterà di generare qualche carattere in più del dovuto, rispetto a prima, specialmente se eravate abituati ad una manipolazione IAMBIC normale (ossia al modo A).

Come già accennato, se avete difficoltà nel reperire il micro o a programmarvelo da soli posso fornirvi un esemplare già programmato, ovviamente previo rimborso spese per il componente e per la spedizione.

Per richieste e o suggerimenti il mio indirizzo e-mail è sempre: [iv3lzq@libero.it](mailto:iv3lzq@libero.it) mentre il mio sito è: <http://www.qsl.net/iv3lzq>

A risentirci in aria, in CW si intende!

73 de Stefano, IV3LZQ

## Bibliografia

Articolo "Un PIC per il CW" di IW4AA, PierLuigi Felletti sul numero di Marzo 2002 di Radiokit;  
PIC16F84A - Data Sheet - Microchip;  
ARRL Handbook 1993 Edition - CW Keyers Section.

