

# Interfaccia Echolink di terza generazione

Una scheda "intelligente" in grado di prendere "decisioni"

2ª parte

Con opportuna regolazione, che dipende anche dalla posizione dei due potenziometri di livello BF e VOX, è possibile ridurre la probabilità che il sistema attivi la trasmissione lato internet quando è interessato il sistema locale radio da un semplice colpo di portante (ad esempio s'impegna l'ingresso di un ponte al solo scopo di verificare se si è in copertura per mezzo della coda). Se per i colpi di portante è possibile fare qualcosa anche sul programma Echolink dal menù del VOX, per i codici DTMF, a parte la soppressione BF, non si può nulla. Resta ben inteso che il grado d'efficacia delle due funzioni precedenti dipende sia da una corretta calibrazione di tutto il sistema sia dalla natura del segnale. Risultati decisamente affidabili, per quel che concerne la soppressione del colpo di portante, sono scaturiti da prove su ripetitori con coda consentendo essi un **netto distacco** tra il colpo iniziale e quello finale. Ovvio che **sostenuti** rumori ambientali alle spalle dell'operatore faranno **comunque** agganciare il sistema, anche se egli tace (per ora con un PIC è ancora difficile riconoscere le parole!). Proprio per limitare questo, durante l'invio di ogni codice DTMF si è rivelata altamente efficace la tecnica

dell'oscuramento BF per 4 secondi (pari al tempo di permanenza sulla finestra annunciators di Echolink dei codici visualizzati).

## Funzionamento COR

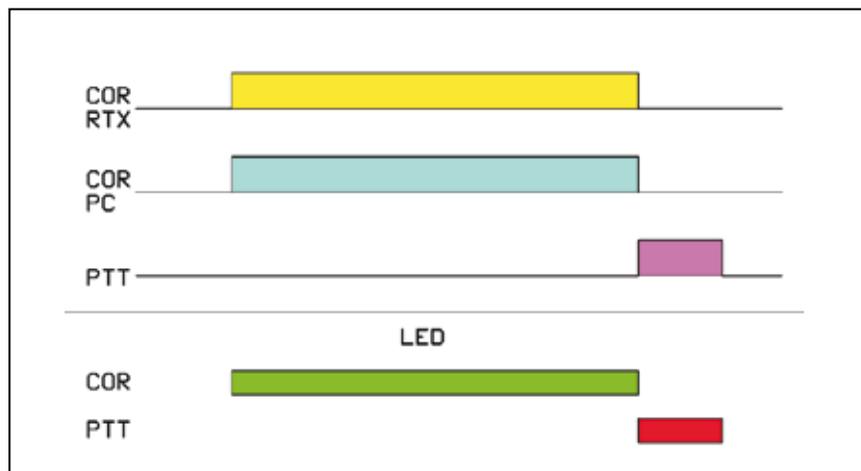
È il sistema classico dove il comando della trasmissione verso internet è attivato su ordine dell'apposita linea dell'RTX che fornisce un opportuno segnale durante una ricezione radio. Il comportamento dell'interfaccia è visibile in fig. 5. Non vi sono grosse novità se non il fatto che anche qui è possibile usufruire "dell'effetto sgancio" alla fine della ricezione. Non ci sono tem-

pistiche da regolare. Se s'intende far lavorare l'interfaccia **esclusivamente** in questa modalità, non essendo in tale processo interessata la BF, la linea di ritardo può anche essere omessa. Allo scopo, nello schema (e sullo stampato), ho predisposto due punti che vanno connessi fra loro per garantire la continuità del percorso BF in assenza di tale circuito. Se invece si vogliono sperimentare le varie configurazioni dell'interfaccia la si monti pure regolando il ritardo al valore minimo.

## Funzionamento MIX

È la collaborazione fra i due sistemi precedenti nata con l'intento di colmare una lacuna del programma Echolink. Il VOX ha il vantaggio di non impegnare la linea internet se non alla presenza di un reale segnale BF (e quindi mandare inutilmente in trasmissione tutti i sistemi connessi); il COR, dal canto suo, trova il punto forte nel fatto che è insensibile alle pause del parlato che potrebbero causare degli strappi. Ecco quindi questa realizzazione "ibrida": la **prima parte** del funzionamento è **VOX**, la **seconda COR**, si osservi la fig. 6. Fino a che non sussiste **una valida BF** la trasmissione lato inter-

Fig. 5



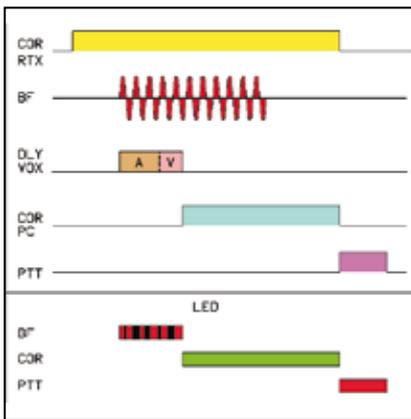


Fig. 6

net **non avviene** a prescindere dallo stato della linea COR radio. All'arrivo di questo segnale si attiva tale invio e da questo momento è presa in considerazione la **sola linea del COR radio** mantenendo la trasmissione PC fino al cessare della ricezione RTX. Quindi niente impegni inutili e nessuna interruzione durante le pause della voce; anche qui è possibile simulare lo sgancio ponte. Si noti che, avendo in comune con il sistema VOX alcune tempistiche, queste potranno/dovranno essere variate secondo le necessità.

### Funzione silenziamento

In poche parole, dopo una trasmissione radio, questa funzione impedisce per un certo tempo al PC di considerare un'eventuale ricezione di BF dall'RTX. Se si è connessi all'uscita di un ripetitore, infatti, si acquisirebbe la coda di quest'ultimo che potrebbe attivare la trasmissione lato internet, in particolare in presenza di toni di fine trasmissione. A parte il fastidio, se all'altro capo del collegamento opera un sistema simile si attiva un ping-pong infinito. Il problema, diversamente dalla seconda generazione, è qui risolto via **software**. Essendo Echolink sul computer impostato in **modalità "COR"**, basta non inviare tale segnale al PC a prescindere dallo stato radio a monte. Alla durata di tale insensibilità provvede con tempo programmabile un apposito timer il quale

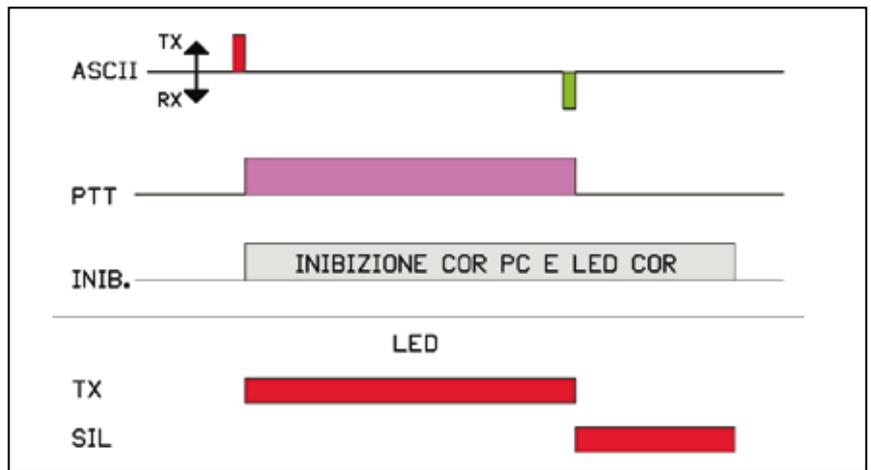


Fig. 7

fornisce anche un segnale per l'accensione del led SIL. Si osservi la fig. 7. Com'è possibile vedere, l'inibizione della linea COR PC e del led del COR avviene contestualmente alla ricezione del comando PTT mentre il tempo SIL scorre solo all'effettiva cessazione di tale trasmissione. Per le code "mute" con quest'interfaccia di terza generazione il problema è inesistente ma consiglio comunque di usare la funzione silenziamento. A volte, nel funzionamento COR, a causa del punto di prelievo fortunoso del segnale dall'RTX, dopo la cessazione di una trasmissione radio può capitare di ritrovarsi sulla linea un **segnale spurio** (con conseguente trasmissione lato internet). In questo caso agire su SIL può eliminare completamente l'inconveniente.

### Time-Out

Chi ha implementato interfacce della seconda generazione

con i software reperibili in internet, avrà di sicuro corso almeno una volta il rischio di arrostitire la parte radio. Spesse volte tali interfacce (anche di quelle commerciali) rimangono bloccate in trasmissione radio su comando del PC malfunzionante a prescindere che sia in uso il sistema ASCII oppure la linea dedicata. Ho quindi implementato un Time-Out per scongiurare questa eventualità. Vi sono due opzioni: il **blocco totale** (fig. 8) o l'**auto-ripristino** (fig. 9) dell'interfaccia. Il comportamento iniziale è comune: all'attivazione della trasmissione parte anche il conteggio del Time-Out (a tale scopo è dedicato **esclusivamente** uno dei tre timer hardware presenti nel microcontrollore). Se entro il tempo impostato non arriva il comando di ricezione si ha l'intervento: in un caso c'è la cessazione della trasmissione radio ed il blocco totale dell'interfaccia (segnalato dal perenne lampeggio del led TX), nell'altro è invece si-

Fig. 8

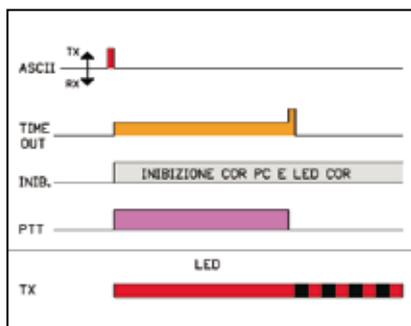
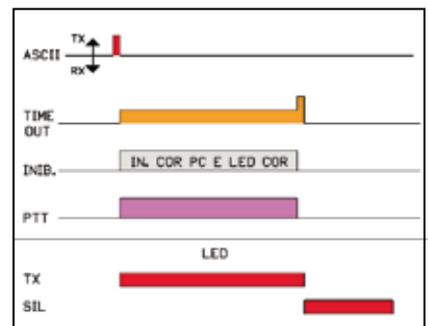


Fig. 9



mulato in loco un comando ASCII di ricezione e l'interfaccia passa nella configurazione di riposo (effettuando tutte le relative procedure come un vero comando). Ovvio che finché il PC non si ripristina anch'esso l'interfaccia resta in stand-by. Anche se sono previste due opzioni, **valutate bene l'uso dell'autoripristino** perché se il PC si blocca, fornendo **continuamente** il comando di trasmissione sulla linea ASCII, **il time-out non serve più a nulla.**

### Funzionamento VOX come seconda generazione

Attivando Echolink e l'interfaccia in **modalità VOX**, con l'impostazione su quest'ultima degli stessi parametri presenti sul programma, il comportamento sarà da seconda generazione ma in più fornendo la visualizzazione con i led BF, COR, PTT e DTMF come una terza generazione. Con i livelli regolati giustamente il sincronismo sarà perfetto (può esserci un'ovvia difformità di comportamento sulle portanti mute e codici DTMF). Anche in questo caso la linea di ritardo non serve e può benissimo essere omessa, oppure, come già detto per il COR, regolata al minimo. Tale configurazione è valida però solo per un link perché nel caso del ponte con coda, essendo selezionato sul programma Echolink il funzionamento VOX, sarebbe ignorata la linea del COR in configurazione "inibita". Per questo motivo ho lasciato il led SIL che funziona in sincronismo con il timer del silenziamento. Basta collegarsi in parallelo ad esso (oppure direttamente al pin 17 porta A0 di U1) con un circuito che interrompe la linea BF per eseguire il mute ed ottenere la **compatibilità totale** con la seconda generazione (anche qui niente linea di ritardo, ovviamente). Non essendo questo uso previsto dallo spirito del progetto riporto tale informazione solo a scopo informativo per chi volesse divertirsi a fare prove o creare altri tipi d'interfacce sfruttando questo software.

### Software e funzionamento operativo

Il programma è lungo, articolato e complesso (per questo genere di PIC, s'intende!); poiché da buon Radioamatore io di solito sbadiglio e mi annoio di fronte ad interminabili pagine d'illustrazione software (e penso accada anche ad altri), mi limiterò all'essenziale. Potete acquisire il file pronto per essere caricato nella memoria del micro sia sul sito di RadioKit sia su <http://www.webalice.it/max4000/r3valsusa>. A livello operativo il micro inizia effettuando le configurazioni richieste: porte analogiche e digitali, timer, interrupt, gestione seriale ecc. Successivamente controlla la pressione del pulsante P1 per entrare eventualmente nel menù taratura e settaggi del quale si parlerà più avanti. Trovando invece tale pulsante a riposo è compiuto un test sull'efficienza del collegamento seriale fra interfaccia e PC: se l'esito è regolare lampeggerà in sincronismo con il flusso dei dati il led PTT e sullo schermo del monitor, nella finestra Echolink degli annunciatori, apparirà la scritta di fig. 10 e poco dopo anche un numero (fig. 11). Questa cifra, moltiplicata per due, fornisce il tempo di pausa, in millisecondi, tra uno scambio dati ASCII e quello successivo; tale valore, anche se rilevato con estrema precisione, è puramente indicativo (circa  $32 \times 2 = 64$  ms nell'esempio). Esso, infatti, dipende da diverse situazioni tra le quali lo stato di lavoro del programma Echolink nel momento dell'acquisizione e la presenza o meno di un adattatore seriale-USB al posto della connessione diretta alla porta COM. Il valore risultante può quindi variare anche di molte unità nell'ambito di sistemi diversi e di qualche unità se riferito sempre allo stesso hardware.

Se tutto è a posto dopo la scomparsa di queste scritte s'illumina

il led verde del COR ed al suo spegnimento l'interfaccia diventa operativa. A questo punto, a seconda della configurazione prevista, è attivata la procedura di lavoro. Circa metà del programma è speso per la gestione del menù taratura e settaggi. Durante il funzionamento in tale menù (ma a volte anche nella configurazione operativa) le porte subiscono numerosi cambi di funzioni (ingresso/uscita, analogico/digitale ecc.), per questo l'hardware relativo è stato **accuratamente studiato** per non creare **situazioni di conflitto**. Questo spiega la presenza d'alcuni componenti aggiuntivi (resistenze) che altrimenti non avrebbero senso di esistere. Anche nella parte software quest'eventualità è in agguato proprio in virtù del fatto che il programma Echolink su PC e l'interfaccia "ragionano" in modo autonomo. Inoltre, per garantire il tempo di corretta visualizzazione sulla finestra "annunciatori", ove necessario, la ricezione DTMF è momentaneamente disabilitata. Si ricorda in proposito che il led DTMF rileva solo la regolare decodifica del bitono, l'acquisizione effettiva è indipendente e demandata al programma sul micro.

Questo stratagemma è obbligatorio perché la permanenza dei simboli in questa finestra Echolink è **temporizzata** e quindi è possibile scrivere nuovamente senza sovrapposizioni solo dopo la scomparsa di quanto precede. Quando necessario (molto spesso) le routine sono ottimizzate ed il programma, per non oltrepassare la memoria disponibile, è stato scritto interamente in assembler. Per dare un'idea concreta su tale lavoro basti pensare che la stampa del sorgente con il programma MpLab occupa quasi 30 pagine, l'occupazione effettiva di memoria è di circa 1,6K contro i 170 byte di una seconda generazione (spesi in buona parte ad emulare la porta seriale). Per lo sviluppo si è fatto largo uso della tecnica di programmazione "in circuit" e per il debug è stato utilizzato l'MpLab SIM, in altre pa-

Fig. 10



Fig. 11



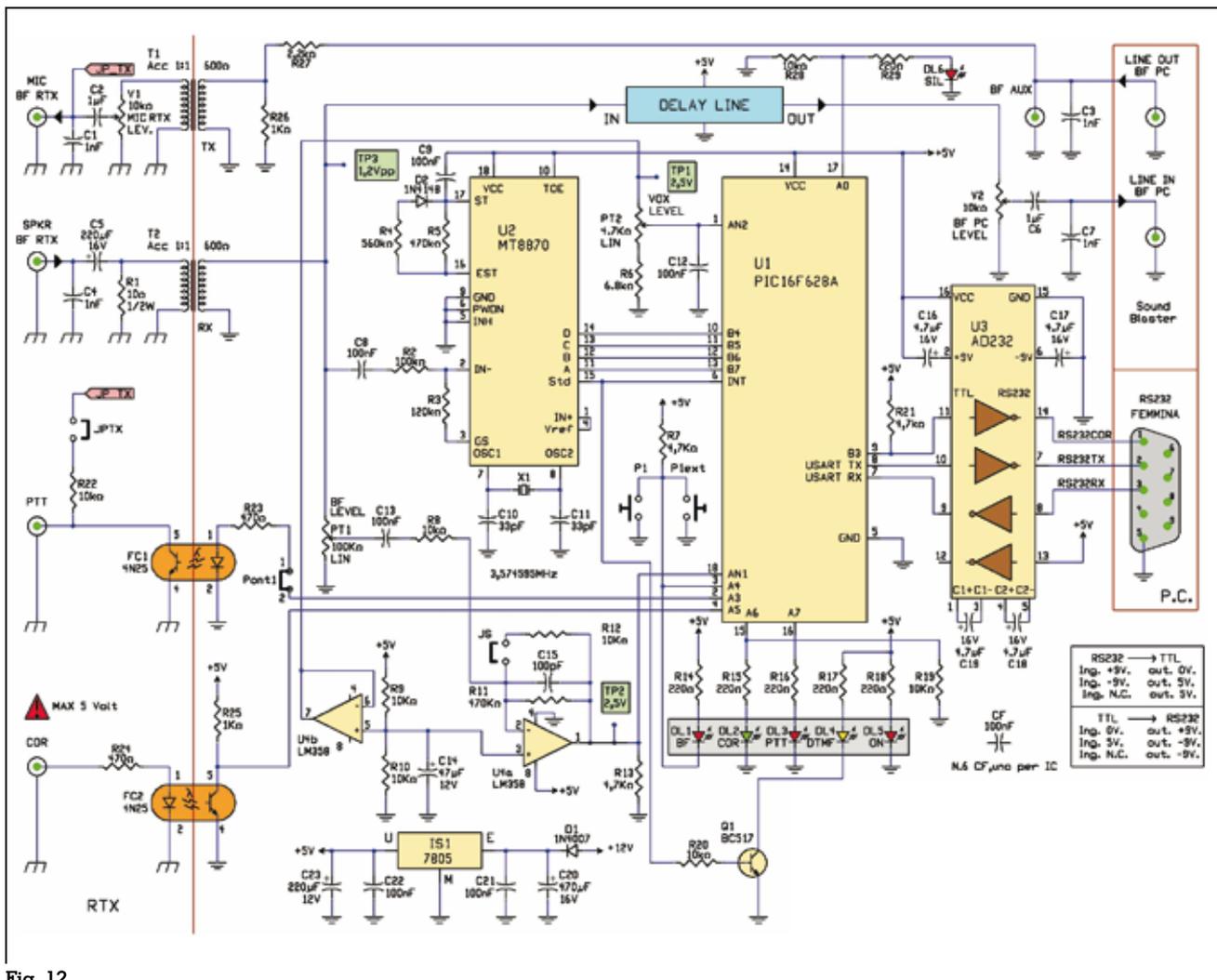


Fig. 12

role il simulatore software in dotazione.

### Lo schema interfaccia

Il circuito è composto principalmente da cinque circuiti integrati, un transistor, due fotoaccoppiatori e due trasformatori d'isolamento. Si osservi lo schema in fig. 12 partendo dall'alto a sinistra.

La BF proveniente dal PC, attraverso una rete d'attenuazione (R26-27), passa nel trasformatore d'isolamento T1 e, dopo essere stata dosata dal trimmer V1 (indicativamente si ha come range di regolazione 0-200 mVpp con 1 Vpp di BF PC), va all'ingresso microfono dell'RTX per essere irradiata. Il condensatore C2 blocca la tensione continua presente sugli apparati palmari

per il circuito di trasmissione e la presa BF AUX è prevista per futuri sviluppi. Il segnale di ricezione passa prima attraverso C5 e successivamente trova la resistenza di carico R1. Fluisce quindi attraverso T2 e poi si dirama: da un lato va al PC, attraverso la delay line, dall'altro si porta al decoder DTMF U2 ed infine al potenziometro PT1 che dosa il livello BF da applicare al microcontrollore. Vediamo questi tre casi in dettaglio. Nel primo troncone la BF raggiunge V2 che regola il livello in ingresso alla sound blaster. Il condensatore C6 blocca l'eventuale tensione continua proveniente dalla scheda del PC, realtà spesso presente nei terminali Mic. In questo percorso è presente la linea di ritardo. Essa è rappresentata in uno schema a parte in modo da essere facilmente circoscritta nel

caso si decida di ometterla per i motivi già menzionati.

E' ovvio che qualora s'intenda seguire tale strada sarà **necessario congiungere** con un filo i punti IN ed OUT (presenti anche sullo stampato con apposite piazzole) per garantire il percorso della BF. Il secondo troncone, con C8-R2-R3, fa capo all'operazionale d'ingresso del decoder DTMF MT8870 (io ho utilizzato il KT3170, un compatibile che costa la metà). I valori sono tali da indurre una leggera amplificazione in modo da garantire un segnale forte ma comunque nel range di lavoro (con 1,2 Vpp di BF si ha 25dB di margine sul livello minimo decodificabile). Ai pin 16-17 fa capo la rete di decisione sulla nota DTMF: in parole povere imposta il tempo di riconoscimento e quello di rilascio. Rispetto alla classica confi-

gurazione utilizzata per tutte le stagioni, al fine di soddisfare le nostre specifiche esigenze radio-amatoriali, l'ho resa **asimmetrica** in modo di facilitare l'operazione alla presenza di rumore. Questo è possibile dosando accuratamente i valori di R4 ed R5 in funzione di C9. A garantire il risultato dell'operazione provvede D2, **non invertitelo** altrimenti il funzionamento sarà l'opposto! Le linee d'U2 responsabili del trasporto del codice in formato binario A-B-C-D fanno capo a B7-B6-B5-B4 (pin 13-12-11-10) del PortB di U1. Quella invece di presenza codice Std va a INT (pin 6) che non è un ingresso qualsiasi, ma bensì una porta programmabile come interrupt. Da tale collegamento è diramato il circuito di visualizzazione a led (DL4 DTMF - giallo) per mezzo di R0-Q1. Veniamo al terzo troncone della BF radio, in altre parole quello che fa capo a PT1. Per assolvere alla funzione VOX, al micro deve essere portato anche tale segnale ma purtroppo il livello a disposizione non è sufficiente. U4a pone rimedio con 33dB (47 volte) circa d'amplificazione (per la taratura, tramite il jumper JS, è possibile ridurla a 0dB) ed il potenziometro a monte permette di regolare il livello che fluirà successivamente in U1 attraverso la porta AN1 (pin 18). Considerato che il livello di partenza ai capi di PT1 è indicativamente circa 1,2 V<sub>pp</sub>, è facile intuire che, con un simile grado d'amplificazione, si raggiungano continuamente i limiti della dinamica dell'operazionale. A noi serve proprio questa situazione in modo da disporre d'un segnale ricco di fronti da elaborare, ma non basta: sempre per esigenze hardware/software in condizione d'assenza BF non vi deve essere zero volt all'ingresso d'U1, bensì 2,5V. A tale necessità provvede la rete R9-R10-C14 facente capo al pin 3 d'U4a. Essa è in comune con il pin 5 d'U4b il quale, in configurazione buffer, fornisce tale voltaggio anche a PT2 che, insieme a R6-C12 dosa il livello della soglia d'intervento del VOX che fa riferimento alla

porta AN2 (pin 1) d'U1. In teoria tale voltaggio sarebbe disponibile anche sul pin 4 d'U2, ma ho constatato tolleranze poco compatibili con quelle richieste dalla nostra applicazione.

Per la gestione seriale ho utilizzato l'apposito hardware interno che fa capo alle porte USART RX (pin 7) ed USART TX (pin 8). Poiché, per scelta del programmatore d'Echolink, il segnale di COR non può essere inserito nel flusso ASCII (e come si sa tale programma è **blindato** per volontà del suo creatore **K1RFD**), ho dovuto dedicare un'apposita linea a tale funzione, ovvero B3 (pin 9). Questi segnali saranno traslati di livello da un classico circuito facente capo ad uno dei molteplici integrati della serie 232 (U3: MAX232, AD232 ecc.). La linea A3 (pin 2) si occupa di fornire tramite il ponticello Pont1 al fotoaccoppiatore FC1 il segnale di PTT (ponticello attualmente da cablare, serve solo per eventuali futuri sviluppi). All'uscita di quest'ultimo è presente una resistenza (R22) inseribile tramite JP-TX nel circuito di BF RTX al fine di comandare in trasmissione gli apparati che funzionano con questa modalità. Ad A5 (pin 4) è invece applicato il segnale di provenienza COR radio debitamente fotoaccoppiato da FC2.

Anche se la sigla nominale di questi due componenti è 4N25, il circuito ha lavorato correttamente anche con altri tipi. Attenzione al segnale d'ingresso che deve essere sufficiente per far accendere il led interno di FC2 ma non deve superare i 5 V altrimenti è necessario aumentare il valore di R24 (per es. con 12 V - 1 k $\Omega$ ). Come già accennato in precedenza è possibile disporre, per i motivi già noti, di un segnale "silenzioso" che pilota il relativo led (DL6 SIL - rosso): esso è fornito da A0 (pin 17). Dovendo assolvere anche ad altri compiti, i led del COR (DL2 COR - verde) e del PTT (DL3 PTT - rosso) sono pilotati direttamente dal micro tramite rispettivamente A6 (pin 15) e A7 (pin 16). Per ultima ho lasciato la descrizione del led BF (DL1 BF - rosso) il quale sparti-

sce la sua porta A4 (pin 2) con il pulsante sito sulla scheda P1. Quest'ultimo, se trovato premuto all'accensione dell'interfaccia, permette di entrare nel menù taratura e settaggi; per comodità, volendo, è possibile aggiungere uno esterno (P1ext). Tale porta subisce quindi dei cambi di direzione nel funzionamento secondo le necessità. Certi piedini hanno delle specifiche funzioni (USART, interrupt, analogiche/digitali, open-drain, tensioni interne di riferimento, scambi ingressi/uscite ecc.) e spesso nel menù settaggi si hanno corpose modifiche di configurazione con funzioni purtroppo interlacciate tra loro. Esistono quindi dei componenti appositi che garantiscono questi scambi senza incidenti (R7-R19-R21-R28). Come trasformatori d'isolamento vanno bene qualunque modello 600 ohm 1:1 ma, se di dimensioni fisiche diverse, bisogna tenerne conto per lo stampato. La soluzione più rapida è lasciare il circuito così com'è e montarli su un'apposita basetta autonoma che poi si collegherà a quella principale in modo volante (niente paura, siamo in bassa frequenza). In alcuni punti dei circuiti BF vi sono dei condensatori ceramici con funzione di by-pass RF (C1-C3-C4-C7). Per ultimo una curiosità: avete notato che U2 ha il suo bel quarzo mentre U1 non ha nulla? È una delle bellezze di questo modello di PIC che dispone internamente di un oscillatore di precisione a 4 MHz che, se usato, permette di recuperare due porte bidirezionali altrimenti adibite al quarzo esterno (B-B7); un'altra porta (solo in ingresso) si ottiene rinunciando al reset esterno (A5). Poiché nell'interfaccia queste tre linee sono utilizzate come porte, bisogna prestare attenzione a fornire le informazioni esatte nel caso di programmazione del micro con sistemi che gestiscono in proprio il "configuration word register".

*(Continua)*

