

# I/8 - una macchina Arduino per il radioamatore

CW e DTMF senza problemi

di Marco Ibridi I4IBR



**A**rduino è nome noto a chi si occupa di elettronica ed informatica: è stato il primo hardware *open-source* ed oramai è capostipite affermato di tutta una serie di cloni basati sulla famiglia di microcontrollori ad 8 bit Amtel AVR (ATmega328, ATmega2560 etc...). Lasciamo, però, i dettagli al sito ufficiale del progetto Arduino<sup>(1)</sup> e vediamo come sia possibile utilizzare praticamente Arduino in una stazione di radioamatore. Verrà qui proposta una vera e propria *macchina* che potrà essere assemblata in più moduli ottenendo, al massimo delle sue potenzialità, le seguenti funzioni:

- decodificatore CW
- frequenzimetro per toni sub-audio (max 299 Hz)
- decodifica DTMF
- decodifica codice I/8 DTMF
- beacon CW

Va subito chiarito che pur essendo possibile, quantomeno in via teorica, utilizzare un algoritmo FFT sull'hardware di Arduino ciò, per gli scopi che si prefigge questo progetto, non è praticabile: le risorse di calcolo a disposizione sono veramente limitate ed anche caricando un codice già compilato in altro ambiente ed utilizzando tabelle precalcolate di seni e coseni si va poco più in là degli **sketch** (è così chiamato il software utente di Arduino, ovvero il programma che l'utente

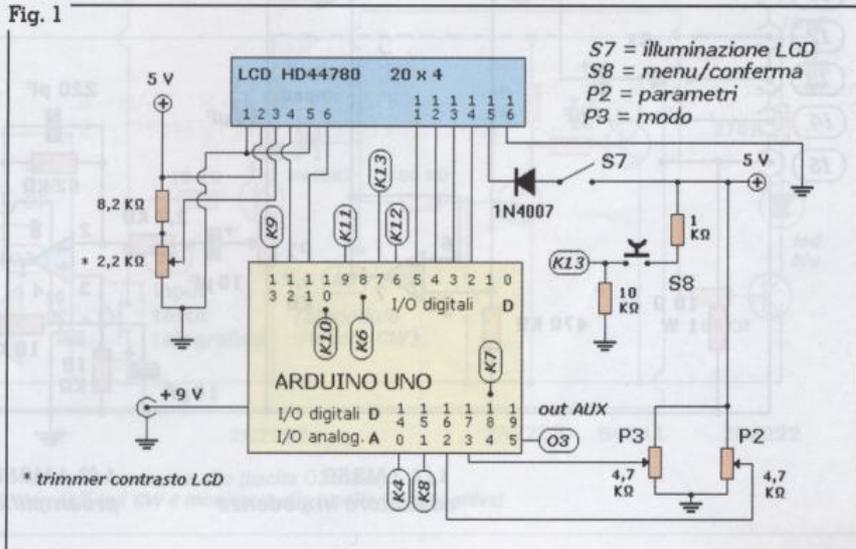
sviluppa per le proprie necessità) demo che si trovano in rete e che realizzano semplici analizzatori nel dominio della frequenza, a volte non in tempo reale ma da files audio; questo, almeno, allo stato attuale dell'arte. La necessità, quindi, di processare segnali audio richiede un supporto hardware adeguato e tradizionale: qui utilizzeremo operazionali per realizzare un modem CW ed un integrato specializzato nella decodifica DTMF; tutto materiale comunissimo e reperibile anche in questi tempi di elevata miniaturizzazione che tanto fa disperare chi ancora si diverte con il saldatore in mano.

Iniziamo con lo schema dei collegamenti del modulo Arduino

che, in questo progetto, sarà costituito dall'hardware **ARDUINO UNO**.

Come visibile in fig. 1, viene utilizzato un display di tipo HD44780 da 4 righe di 20 caratteri ciascuna; i due potenziometri selezionano le opzioni di menù ed i parametri operativi, sostituendo i normali commutatori ed offrendo una maggiore *apertura* al sistema: opzioni future saranno disponibili senza dover cambiare commutatori o rifare pannelli; inoltre, volendo implementare solo una delle funzionalità proposte, basterà sostituire il potenziometro P3 con un trimmer regolandolo sulla voce di menù attivata e realizzando solo il modulo hardware necessario.

Fig. 1



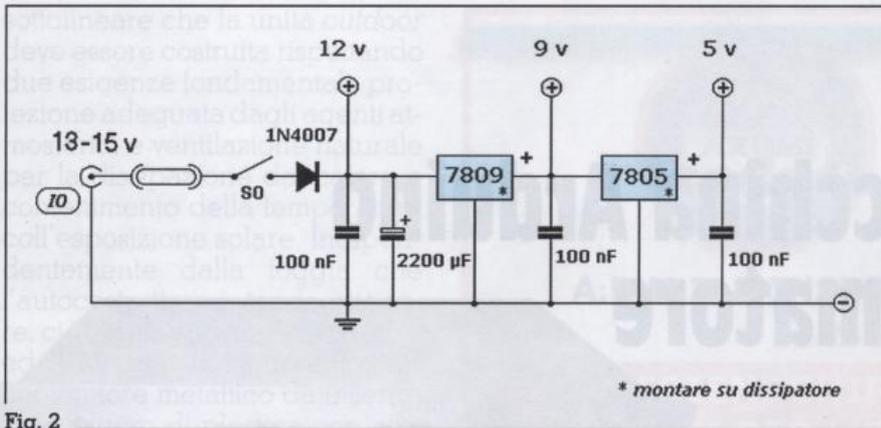
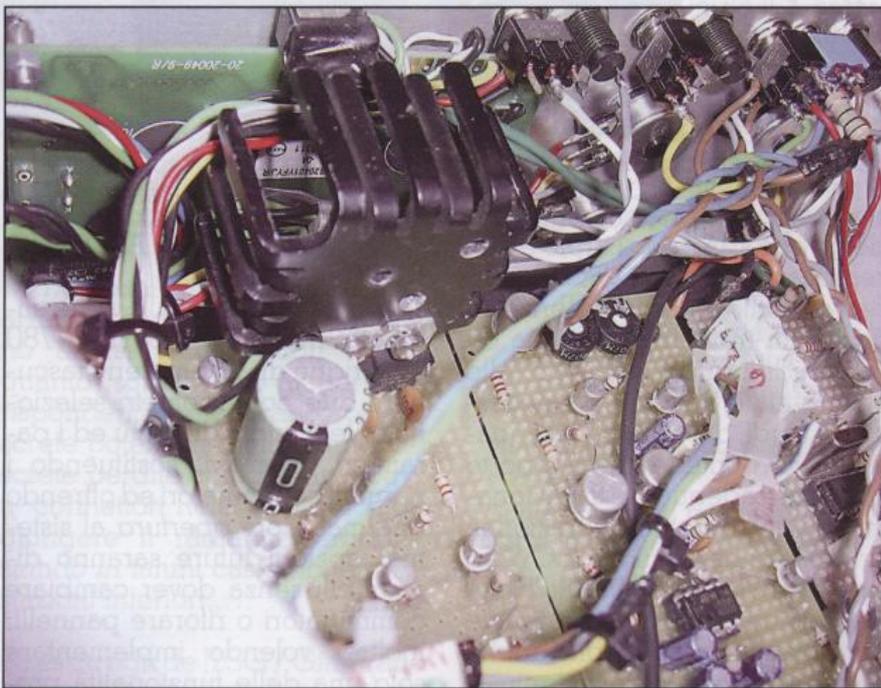


Fig. 2



L'alimentazione, fig.2, è del tipo tradizionale e mette a disposizione 12, 9 e 5 V in continua partendo da un'alimentazione a 13/15 volt. E' consigliato un modesto dissipatore, specialmente se si userà continuamente l'illuminazione del display (S7).

In fig. 3 abbiamo la parte comune a più moduli: è il modulo preamplificatore / adattatore d'impedenza.

Sono disponibili cinque linee di input audio di cui una, la quinta, con carico resistivo; i trimmer da 22 kΩ verranno regolati *una tantum* in fase di taratura iniziale mentre il potenziometro P1 sarà posto sul pannello frontale e regolerà il livello del segnale audio per tutti i moduli.

Segue, fig.4, il modem CW; questo modulo trasforma il segnale audio in ingresso in uno stato logico TTL compatibile con l'input digitale di Arduino. Nello schema è riportata, pure, un'uscita open collector (O1) attivata dal modulo monitor e che è utilizzata dalla funzione **Beacon CW** o dalla manipolazione manuale sul tasto telegrafico (I6).

In fig.5 il modulo complementare al modem CW: il monitor. Questo modulo è l'**oscillofono** che, oltre a rendere udibile una eventuale manipolazione dal tasto locale (nello schema in I6),

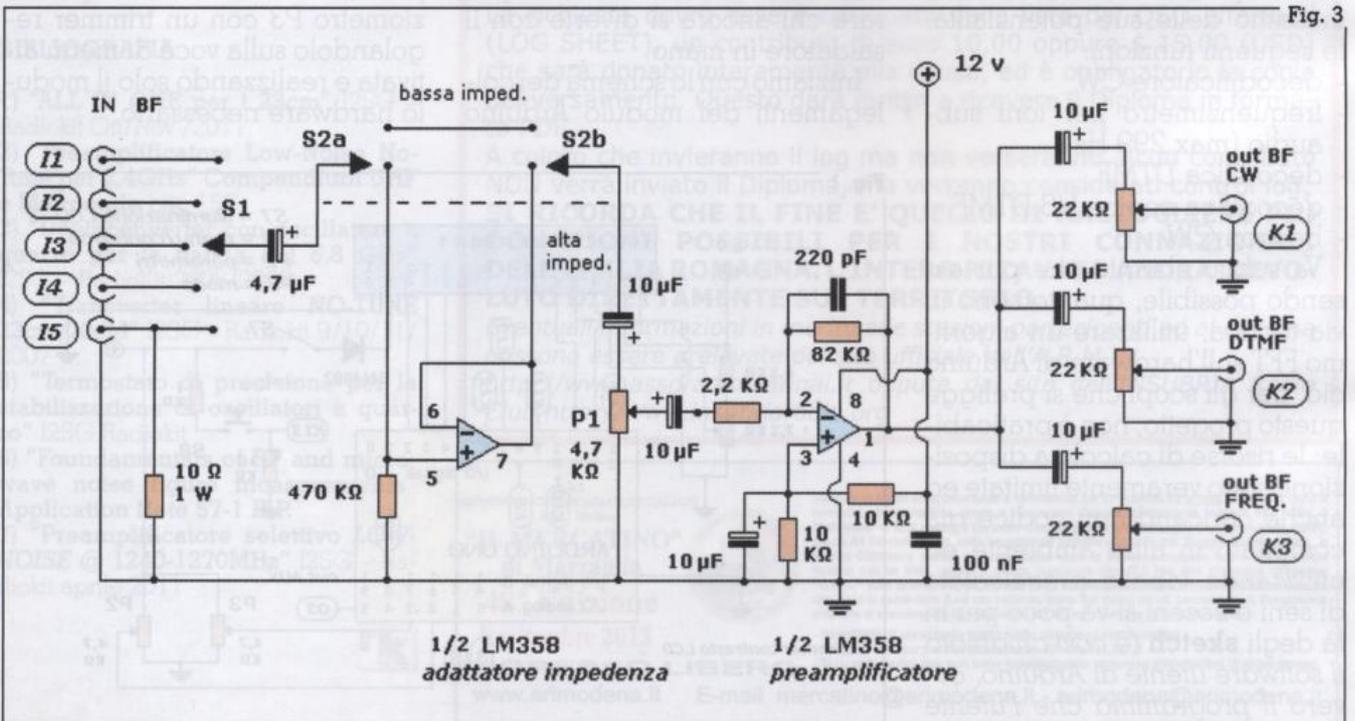
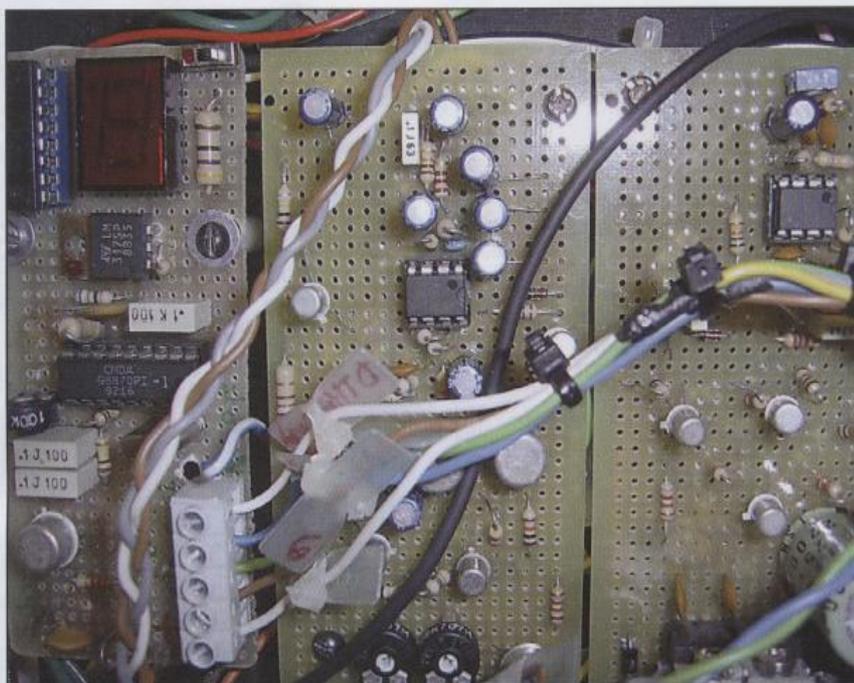
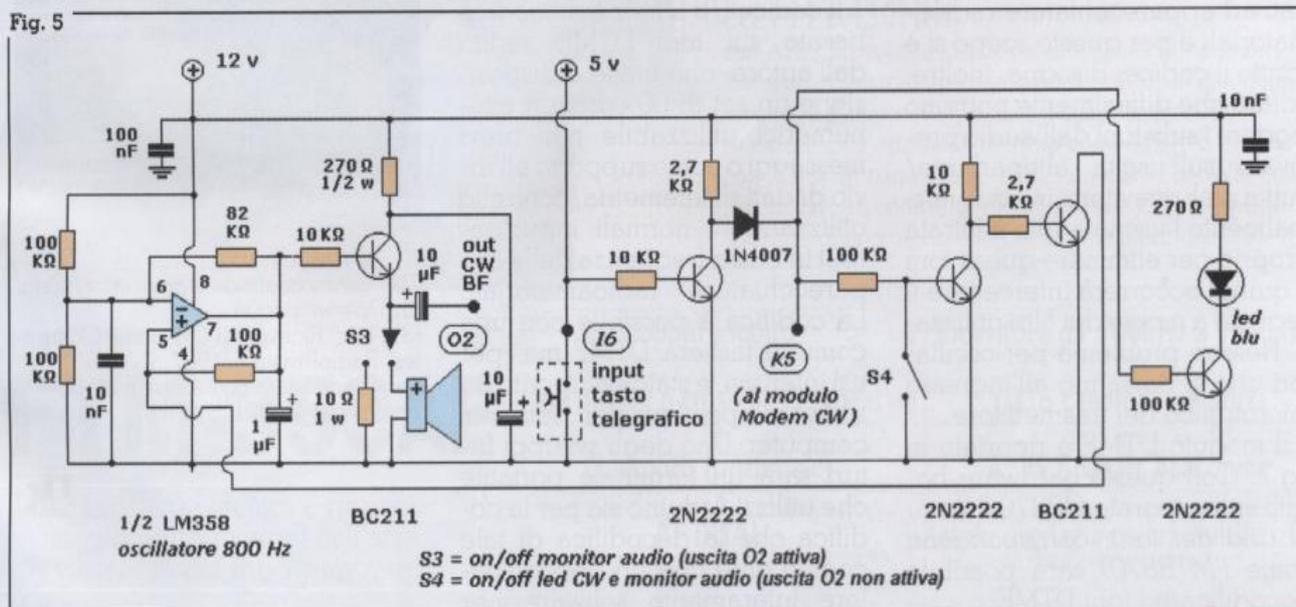
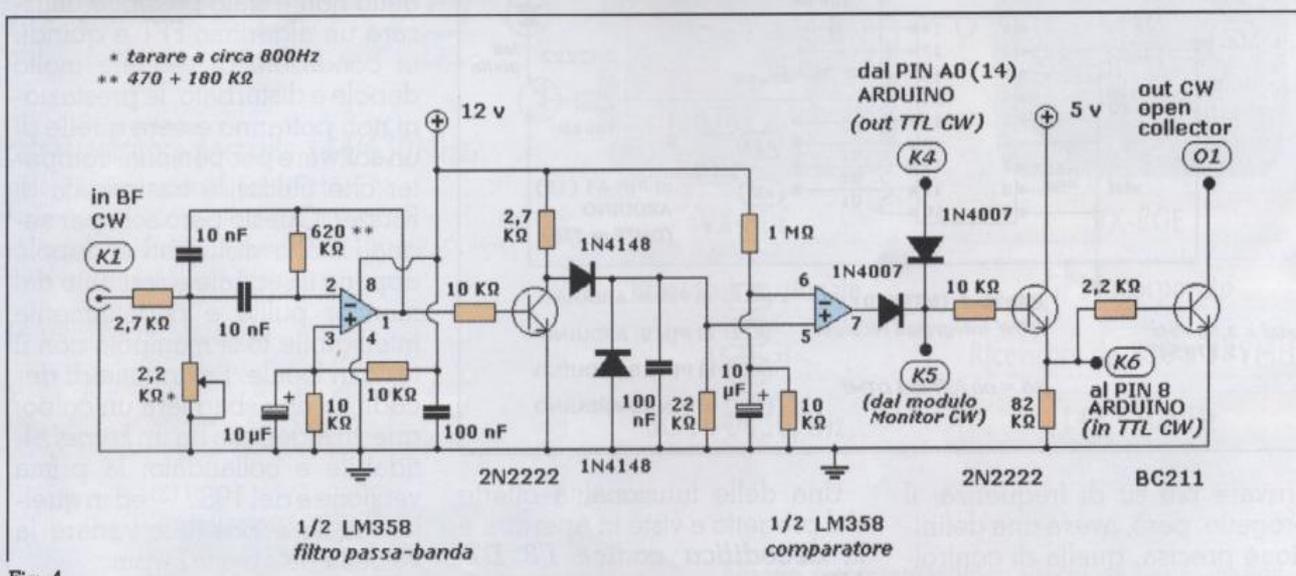


Fig. 3



rigenera il segnale decodificato dal modem CW; è così possibile **sentire** se si è correttamente sintonizzata una emissione CW. L'uscita O2 mette a disposizione questo segnale di bassa frequenza per diffusione od ascolto in cuffia ma va ricordato che la nota è ottenuta da un oscillatore in onda quadra e quindi di scarsa purezza spettrale.

Il modulo frequenzimetro sub-audio è essenzialmente un passa-basso (fig. 6). Con questo modulo, utilizzando normalmente l'ingresso audio commutato in alta impedenza, avremo a disposizione un frequenzimetro per letture sino a 300 Hz. Questo è un limite voluto: nulla vieta di cambiare la linea di codice interessata, eliminare il passa-basso ed



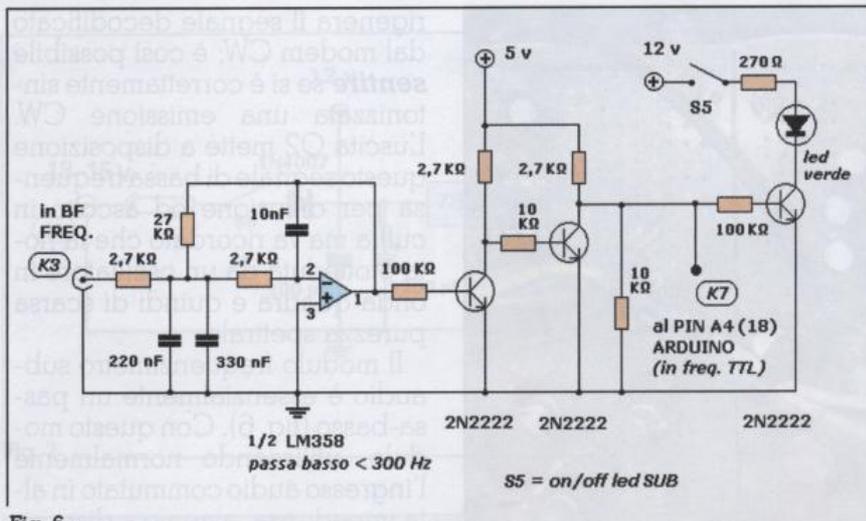


Fig. 6

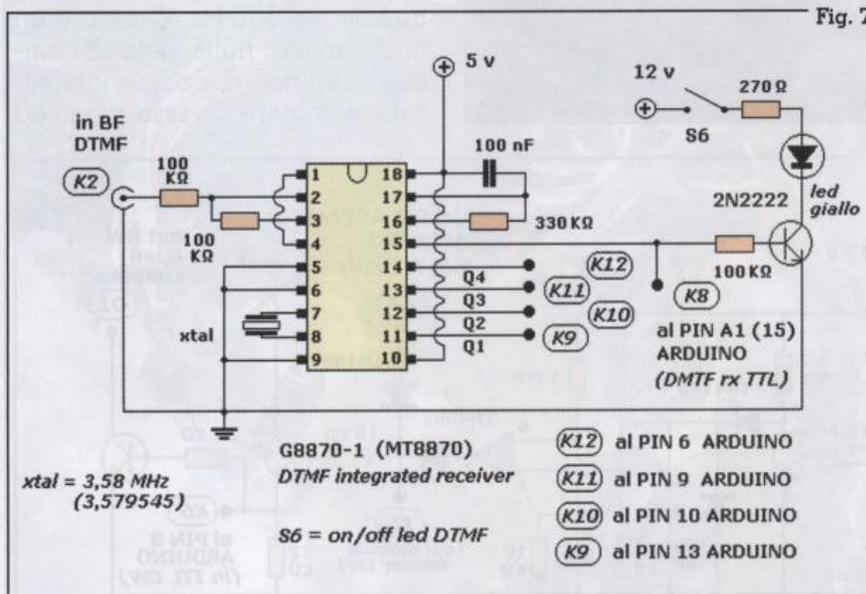


Fig. 7

arrivare più su di frequenza; il progetto, però, aveva una definizione precisa, quella di controllare i sub-toni di oscillatori destinati ad apparecchiature radioamatoriali e per questo scopo si è scritto il codice. Bisogna, inoltre, notare che difficilmente potremo leggere i sub-toni dall'audio prelevato sull'uscita altoparlante/cuffia del ricevitore in uso: normalmente l'uscita audio è filtrata proprio per eliminare questi toni e quindi occorrerà intercettare il segnale a monte dei filtri utilizzati. Nessun problema per oscillatori che si porranno all'ingresso microfonico del trasmettitore.

Il modulo DTMF è riportato in fig.7. Con questo hardware basato sull'integrato G8870-1 (o su di uno dei tanti corrispondenti come l'MT8870) sarà possibile decodificare i toni DTMF.

Una delle funzionalità offerte dal progetto e viste in apertura è la **decodifica codice I/8 DTMF**.

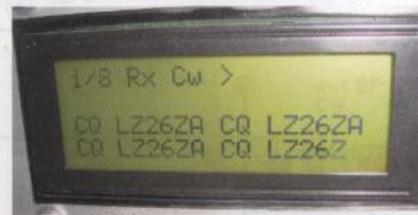
Il codice I/8 DTMF è un codice basato sui toni DTMF, scritto dall'autore, che mette a disposizione un set di 60 caratteri alfanumerici utilizzabile per brevi messaggi o come supporto all'invio di dati di telemetria/controllo utilizzando i normali input/output in bassa frequenza delle apparecchiature radioamatoriali. La codifica è possibile con una comune tastiera DTMF ma, per usi intensivi, è stato sviluppato un software specifico per personal computer. Uno degli sviluppi futuri sarà un terminale portatile che utilizzi Arduino sia per la codifica che la decodifica di tale codice oltre che un decodificatore interamente software per

personal computer. Le tabelle del codice I/8 DTMF come il software (**sketch**) per Arduino (collaudato con l'ide 0022) ed il software di codifica per piattaforma Windows (utilizzabile anche con **Ubuntu/Wine**) sono liberamente scaricabili all'indirizzo web in riferimento. (2)

Il **Beacon CW** è facilmente configurabile modificando l'opportuna riga del codice; lo sketch viene distribuito con la seguente stringa:

```
// messaggio beacon
String msg_beacon="vvv
de i4ibr op marco loca-
tor jn54ot ";
```

Alcune considerazioni vanno, infine, fatte riguardo alla decodifica dei segnali in CW; come si è detto non è stato possibile utilizzare un algoritmo FFT e quindi, in condizioni di segnale molto debole e disturbato, le prestazioni non potranno essere quelle di un software per personal computer che utilizzi la trasformata di Fourier. Questo però solo per segnali molto disturbati e deboli, appena il segnale è restituito dal monitor pulito e perfettamente intelligibile lo si manipola con il tasto in locale, l'algoritmo di decodifica non sbaglierà un colpo; questo algoritmo ha un kernel affidabile e collaudato: la prima versione è del 1987 (3) ed in quella attuale è possibile variare la velocità da 10 a 70 wpm.



- (1) [www.arduino.cc](http://www.arduino.cc)
- (2) [www.centrodata.com/i4ibr/main.php?p=wi\\_pag04](http://www.centrodata.com/i4ibr/main.php?p=wi_pag04)
- (3) i4ibr "Ricevere il CW con il Computer" RadioRivista 4/1987 [www.centrodata.com/i4ibr/main.php?p=wi\\_pag01](http://www.centrodata.com/i4ibr/main.php?p=wi_pag01)



### ERRATA CORRIGE

*Il diavolo ogni tanto ci mette la coda.....E stavolta l'ha infilata nella macchina Arduino pubblicata a pagina 31 di Radiokit di settembre. L'errore è nello schema elettrico di figura 5 a pagina 33. Qui di seguito pubblichiamo lo schema corretto chiedendo scusa ai Lettori.*

