

Elettronica Innovativa

Elettronica In

Mensile di elettronica applicata, attualità scientifica, novità tecnologiche.

124



**Chiave
DTMF**

**Tabellone
segnapunti
radiocomandato**



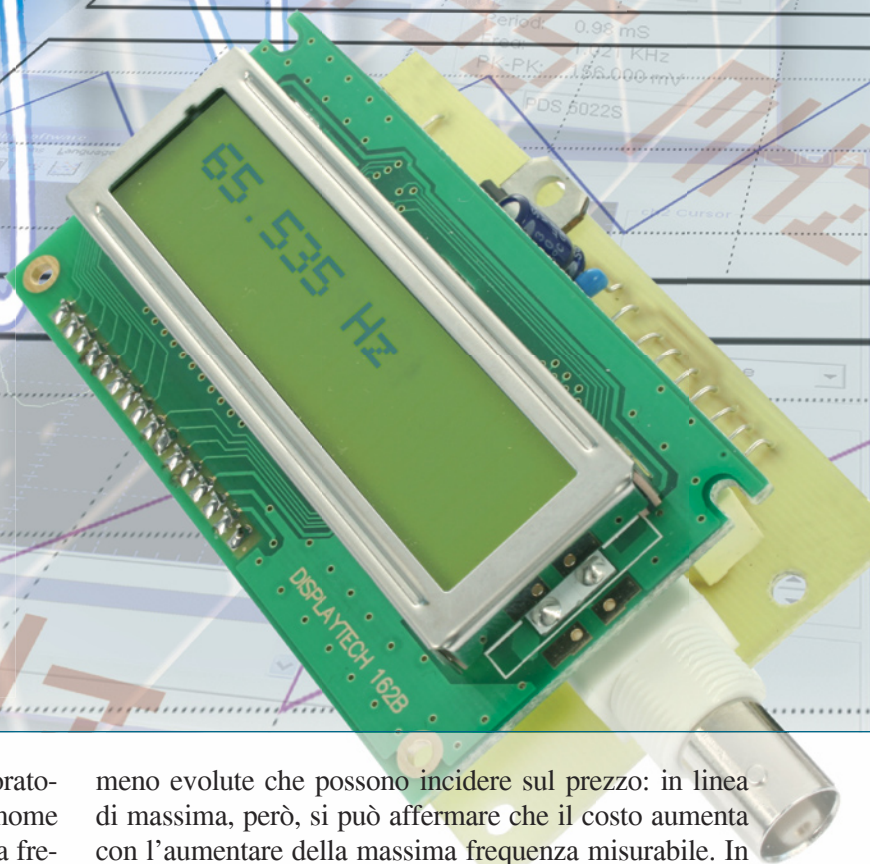
- Frequenzimetro 5 Hz ÷ 50 MHz con PIC**
- TX/RX per ascolto ambientale**
- Allarme gas per camper**
- Semaforo per fermodellismo**
- Tutto sul protocollo PS2**

+++ ESCLUSIVO +++
CORSO EM1000
Programmazione
modulo Tibbo

Frequenzimetro 5 Hz ÷ 50 MHz con PIC

dell' Ing. Mirco Segatello

Dotato di display LCD e di microcontrollore, è in grado di misurare frequenze comprese tra 5 Hz e 50 MHz. La selezione della scala di lettura è automatica, così da garantire sempre la massima risoluzione nella visualizzazione della frequenza. Presenta una sensibilità di 50 mV e può essere alimentato con un adattatore di rete da 9÷12 Vdc.



Uno strumento quasi sempre presente nei laboratori di elettronica è il frequenzimetro: come il nome stesso lascia intendere, esso consente di verificare la frequenza di un segnale elettrico variabile, ossia il numero di volte che tale segnale si ripete nell'unità di tempo (1 sec). La misura della frequenza è molto utile, se non indispensabile, in molti casi: quando si deve tarare l'oscillatore di un trasmettitore RF, oppure quando si deve ritoccare il compensatore di taratura di un oscillatore quarzato, e in tanti altri casi ancora. Esistono differenti modelli di frequenzimetro, ognuno dotato di caratteristiche più o

meno evolute che possono incidere sul prezzo: in linea di massima, però, si può affermare che il costo aumenta con l'aumentare della massima frequenza misurabile. In pratica, se per le aziende l'acquisto di uno strumento di misura professionale non rappresenta un problema, per gli hobbisti, a volte, può essere difficile trovare uno strumento di buona qualità ad un prezzo ragionevole. Bene: quello che proponiamo oggi è un semplice ed economico frequenzimetro digitale che, grazie alle soluzioni tecniche adottate, garantisce grande precisione nella lettura pur impiegando componenti facilmente reperibili ad un



Versione con AD8561 nello stadio di ingresso

- Range di frequenza: 5 Hz ÷ 50 MHz
- Selezione gamma di misura: automatica (autorange)
- Impedenza di ingresso: 10 kohm
- Sensibilità: 50 mVeff
- Risoluzione: 5 digit

Versione con LM311 come stadio di ingresso

- Range di frequenza: 5 Hz ÷ 5 MHz
- Selezione gamma misura: automatica (autorange)
- Impedenza di ingresso: 50 kohm
- Sensibilità: 100 mV ÷ 20 Veff
- Risoluzione: 5 digit

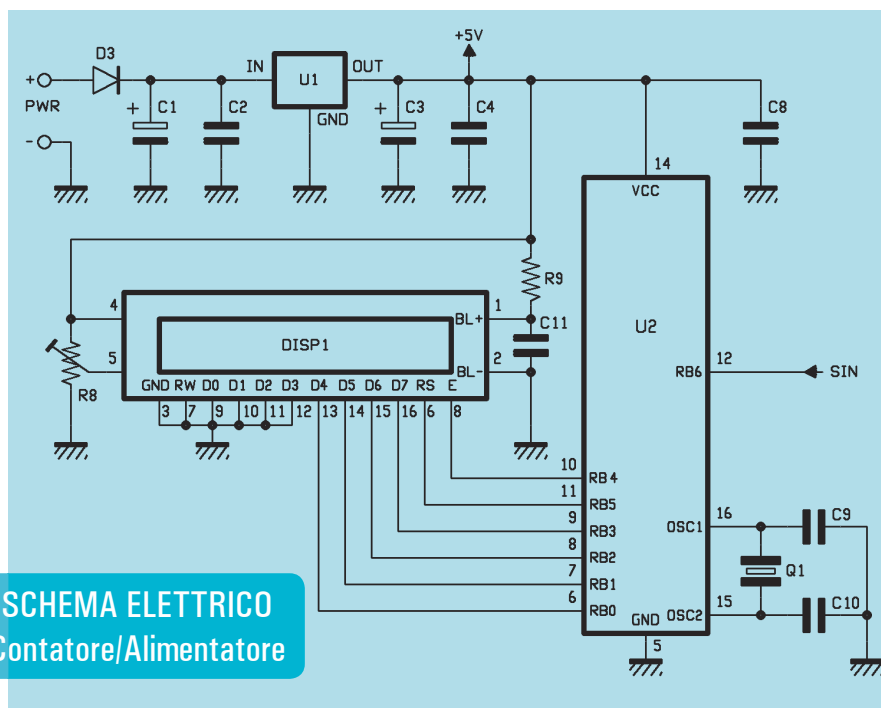
Schema elettrico

Il frequenzimetro è piuttosto semplice in entrambe le versioni: gli stadi di alimentazione, conteggio e visualizzazione sono identici mentre lo stadio di ingresso cambia in funzione del comparatore utilizzato. Analizziamo, ora, ogni singola parte dello schema.

Lo stadio di alimentazione, pensato per poter alimentare il circuito tramite un piccolo adattatore di rete non stabilizzato da 8÷12 volt, è costituito da un diodo di protezione contro le inversioni di polarità (D3) e da un regolatore di tensione modello LM7805 (U1). Sia all'ingresso che all'uscita sono presenti due reti di filtro (C1-C2 e C3-C4), il cui scopo è quello di stabilizzare e ripulire da eventuali disturbi la tensione proveniente dall'adattatore. Dato che il frequenzimetro assorbe solo pochi milliampere, può facilmente essere reso portatile alimentandolo a batteria (ad esempio con una semplice batteria a 9 volt oppure con 8 pile ricaricabili tipo Ni-Mh o Ni-Cd); in questo caso, però, è conveniente non montare la resistenza R9 da 100 ohm, che fornisce l'alimentazione alla retroilluminazione del display. Quest'ultima, infatti, assorbe più corrente dell'intero circuito.

Lo stadio di conteggio della frequenza fa capo al chip U2, un PIC16F628 programmato per contare gli impulsi provenienti dal comparatore di ingresso ed applicati al pin RB6/T1CKIN. Per la visualizzazione, invece, utilizziamo un display LCD da 16x2 caratteri, soluzione che può sembrare costosa, ma il cui prezzo, alla fine, è simile a quanto si spenderebbe facendo uso di otto display a LED "7 segmenti", con cui visualizzare la frequenza a cinque cifre e l'unità di misura.

Il trimmer R8 permette di regolare il contrasto del display, unica taratura richiesta per far funzionare correttamente il circuito. Essendo questi installato sotto il display LCD, la

SCHEMA ELETTRICO
Contatore/Alimentatore

costo contenuto. In particolare, l'utilizzo di un microcontrollore PIC, che effettua i conteggi e le elaborazioni numeriche necessarie, ha consentito di realizzare uno strumento in grado di scegliere automaticamente il range di misura ottimale e di operare con frequenze comprese tra 5 Hz e 50 MHz, garantendo risoluzioni di misura molto alte (fino a 0,01 Hz), specialmente in bassa frequenza.

In realtà, per misurare frequenze fino a 50 MHz, è necessario utilizzare come stadio di ingresso un comparatore veloce non molto economico e

difficile da reperire (AD8561); proprio per questa ragione, abbiamo realizzato due versioni del circuito, una delle quali "depotenziata": quest'ultima impiega come stadio di ingresso un comparatore *general purpose* modello LM311. Così facendo, la massima frequenza misurabile è limitata a 5 MHz (valore più che dignitoso per applicazioni hobbistiche), ma quale contropartita delle prestazioni inferiori, l'LM311 risulta essere tanto economico quanto facile da reperire. Vediamo ora come abbiamo realizzato "i nostri frequenzimetri".

regolazione deve essere effettuata facendo passare un piccolo cacciavite sotto lo stampato, attraverso un foro in corrispondenza del trimmer.

Per ottenere la massima precisione nella misura, abbiamo preferito non utilizzare il generatore di clock interno del PIC, optando invece per un quarzo esterno, di valore facilmente reperibile. La realizzazione a sandwich con il display sovrastante al circuito, inoltre, permette di ottenere un frequenzimetro molto compatto.

Come avrete certamente notato, non sono presenti pulsanti di alcun tipo in quanto tutte le funzioni sono automatiche e gestite direttamente dal microcontrollore. Il conteggio degli impulsi presenti sul pin RB6, dopo le dovute elaborazioni numeriche, viene visualizzato sul display nel formato "frequenza a cinque digit + unità di misura (Hz, kHz...)".

Abbiamo già anticipato che le due versioni differiscono solo nello stadio di ingresso: in particolare, la versione a 5 MHz, che impiega l'LM311, necessita della resistenza di pull-up R10, non presente nella versione a 50 MHz. Per il resto, anche i due stadi di ingresso sono praticamente identici (valori di alcuni componenti a parte): le resistenze R1 e R2 servono per polarizzare a metà tensione l'ingresso invertente del comparatore di tensione U3. La resistenza R7 e i diodi D1 e D2 proteggono l'ingresso del comparatore da possibili sovratensioni; a tal proposito, sappiate che la tensione applicata all'ingresso non deve superare il valore di 20 Vpp: tensioni superiori potrebbero danneggiare lo stadio d'ingresso. Il condensatore C7 blocca la componente continua del segnale lasciando passare solo le frequenze misurabili dallo strumento. La frequenza di taglio, di circa 15 Hz nella versione a 50 MHz, comporta un lieve calo della sensibilità verso le frequenze molto basse (inferiori a 20 Hz), condizione peraltro facilmente migliorabile aumentando il valore del condensatore C7. Nella versione con

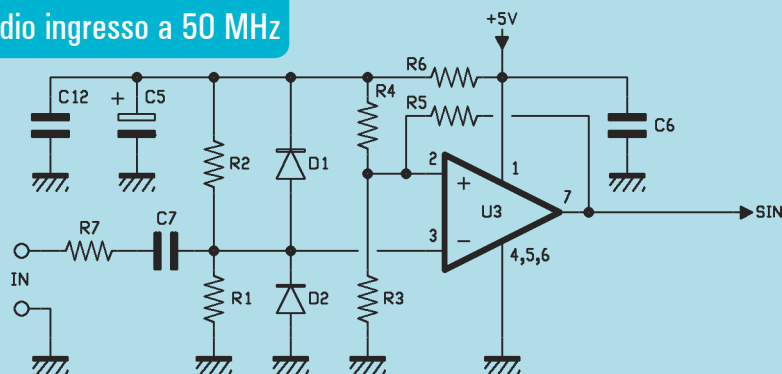
LM311 (5 MHz) invece, il problema non sussiste in quanto l'impedenza di ingresso è decisamente più alta e di conseguenza la banda passante alle basse frequenze non ne risente.

R3 e R4 sono le resistenze di polarizzazione dell'ingresso non invertente del comparatore, mentre R5

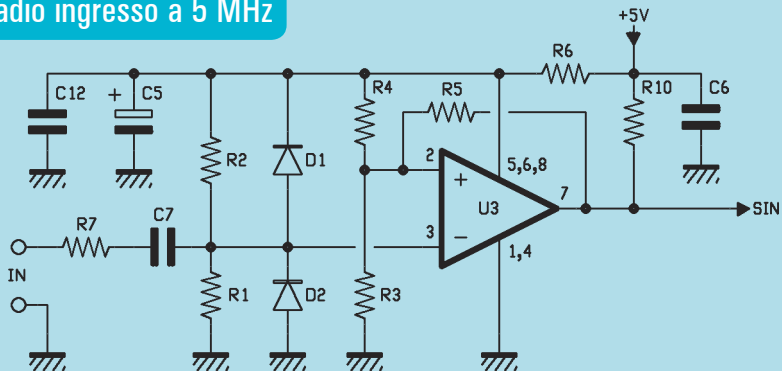
200 nsec riduce la massima frequenza di lavoro a circa 5 MHz, valore comunque perfettamente idoneo per essere utilizzato nel nostro circuito qualora non servano prestazioni in alta frequenza.

Quale che sia il comparatore scelto, il suo scopo è quello di confrontare la

SCHEMA ELETTRICO Stadio ingresso a 50 MHz



SCHEMA ELETTRICO Stadio ingresso a 5 MHz



in retroazione positiva con le due precedenti, crea una piccola isteresi, necessaria per ottenere una commutazione dell'uscita del comparatore più pulita e decisa. Relativamente ai comparatori scelti, va detto che l'integrato AD8561 è un comparatore di tensione ad alte prestazioni con un tempo di ritardo nella commutazione di soli 7 nsec e pertanto idoneo a lavorare con frequenze molto elevate (circa 50 MHz). L'LM311, invece, è un operazionale utilizzato come comparatore di tensione: il suo costo è modesto, ma il tempo di risposta di

tensione di riferimento presente sul pin non invertente con la tensione in ingresso che giunge al piedino invertente e - nel caso - di commutare la sua uscita in modo da fornire al pin RB6 del PIC un'onda quadra perfettamente squadrata e pulita, avente ampiezza pari a 5 V e adatta, pertanto, per essere conteggiata. Tenete presente che nella versione a 5 MHz, l'uscita dell'LM311 è di tipo open-collector e necessita pertanto di una resistenza di pull-up, rappresentata da R10. Entrambe le soluzioni per quanto operanti a frequenze diffe-

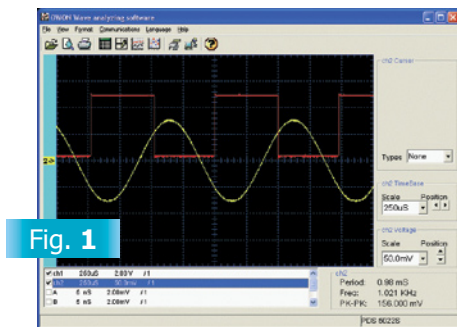


Fig. 1

renti, garantiscono prestazioni molto buone: prestate attenzione alle figure 1, 2 e 3, che mostrano lo stato dell'uscita del comparatore (visualizzata tramite un oscilloscopio) applicando all'ingresso una tensione sinusoidale di circa 150 mV di ampiezza con frequenza variabile. Nella versione a 5 MHz, alla frequenza di 1 kHz (Fig. 1) la tensione in uscita all'LM311 risulta perfettamente squadrata, mentre alla frequenza di 1 MHz pur mantenendo una forma accettabile, il tempo di salita del segnale (legato al tempo di commutazione del chip) comincia ad essere non trascurabile: introduce infatti una evidente distorsione sul fronte di salita che limita la massima frequenza misurabile a

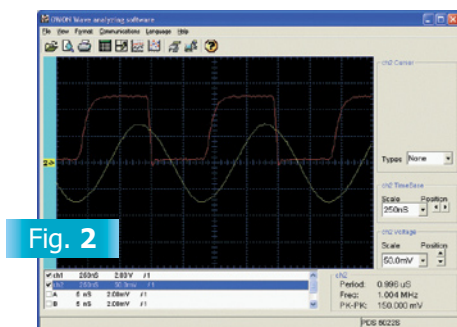


Fig. 2

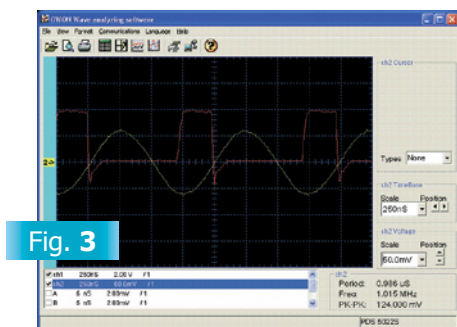


Fig. 3

circa 5 MHz (Fig. 2). Nella versione a 50 MHz, che utilizza il comparatore veloce modello AD8561, invece,

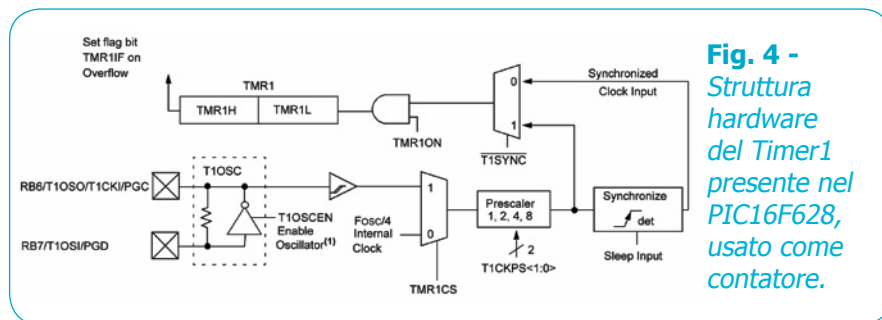


Fig. 4 - Struttura hardware del Timer1 presente nel PIC16F628, usato come contatore.

l'uscita a 1 MHz mantiene dei fronti molto ripidi, pur non presentando la simmetria del duty-cycle (Fig. 3).

Il firmware

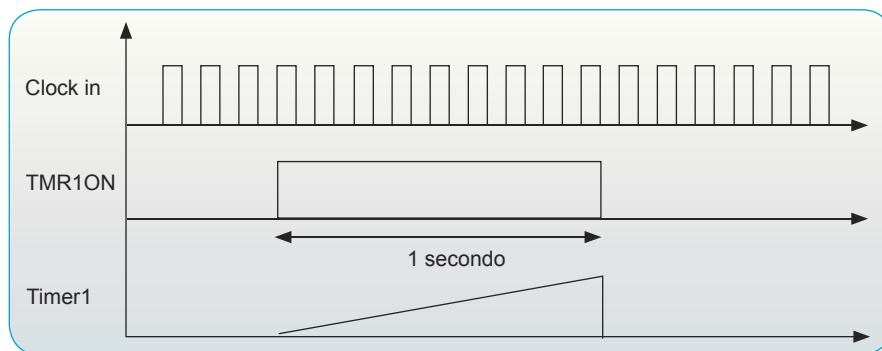
Dal punto di vista didattico, la parte più interessante del progetto è rappresentata dal firmware, insieme al tipo di microcontrollore utilizzato.

Il PIC16F628, micro di fascia bassa e pertanto molto economico, dispone di ben tre "Timer", tutti utilizzabili sia come Timer che come semplici contatori. Proprio uno di essi (per la precisione il Timer1) offre le caratteristiche da noi richieste: si tratta, infatti di un timer a 16 bit con la possibilità di conteggio di impulsi esterni (in modalità asincrona) oppure di

la possibilità di bloccare il conteggio agendo sul bit TMR1ON. Come in tutti i contatori interni dei PIC, anche in Timer1 è previsto un pin (denominato TMR1IF "Overflow flag bit") che viene portato a "1" se il conteggio supera il limite massimo dei 16 bit, ossia se supera il valore di $2^{16}-1 = 65.535$.

Un prescaler permette di dividere la frequenza in ingresso al contatore sino ad un massimo di otto volte. Il segnale applicato all'ingresso esterno, inoltre, viene squadrato da una porta logica a Trigger di Schmitt, così da fornire al contatore un segnale molto pulito e preciso. Detto questo, passiamo alla descrizione delle operazioni per determina-

Fig. 5 - Contatore di frequenza: Timer 1 viene abilitato per un periodo noto (TMR1ON); viene quindi letto il contenuto dei registri TMR1H e TMR1L.



impulsi interni derivati dal clock di sistema (in modalità sincrona). Dallo schema interno, disponibile sul datasheet del componente e riportato in Fig. 4, notate la doppia possibilità di conteggio interno o esterno (comandata dal bit TMR1CS) e l'opzione per il conteggio sincrono o asincrono (bit T1SYNC). Altra peculiarità di questo Timer è

re la frequenza del segnale in ingresso; innanzitutto occorre impostare il bit TMR1CS a 1 e il bit TMR1ON a 0: in questo modo, il contatore (che supponiamo essere a zero) è predisposto per il conteggio degli impulsi provenienti dal pin RB6. Le operazioni da eseguire sono le seguenti:

- TMR1ON=1 abilita conteggio,

- attesa di un secondo,
- TMR1ON=0 ferma conteggio,
- lettura del valore del contatore.

Con queste semplici operazioni (schematizzate in Fig. 4) abbiamo già ricavato la frequenza del segnale in ingresso come numero di impulsi al secondo, che è proprio la definizione di Hertz ($1 \text{ Hz} = 1/\text{sec} = 1 \text{ sec}^{-1}$). Ora è sufficiente visualizzare sul display il valore contenuto nei registri del contatore ed il gioco è fatto. Per calcolare nuovamente la frequenza è sufficiente azzerare il contatore e ripetere la procedura precedentemente descritta. Il sistema appena descritto permette di leggere tutte le frequenze da un minimo di 0 Hz ad un massimo di 65.535 Hz con la risoluzione di 1 Hz. Per misurare frequenze più alte è sufficiente ridurre il tempo di abilitazione del contatore, denominato *Tempo di Gate*. Portando tale tempo a 0,1 secondi, è possibile misurare frequenze tra 0 e 655.350 Hz con una risoluzione di 10 Hz.

Per poter salire ulteriormente in frequenza, è necessario abilitare il prescaler, facendo effettuare ad esso una

kHz. In pratica, però, non è possibile superare i 50 MHz per via dei tempi di commutazione del PIC e del comparatore. Per frequenze superiori a 50 MHz, quindi, abbiamo deciso di forzare sul display l'indicazione "Over Range". Teoricamente sarebbe possibile salire ulteriormente in frequenza, ma dovremmo realizzare uno stadio di ingresso ben più sofisticato e costoso, magari dotato di un ulteriore prescaler esterno. La frequenza visualizzata sul display, inoltre, dovrebbe essere opportunamente scalata e riportata con la giusta unità di misura. Tutto ciò aumenterebbe le prestazioni ma renderebbe vani i nostri sforzi per contenere il costo finale del dispositivo.

La procedura appena vista garantisce una buona risoluzione nella lettura se si misurano frequenze alte (centinaia/migliaia di Hz); ma cosa accadrebbe se dovessimo misurare frequenze più basse come ad esempio 10 Hz? Una frequenza così bassa verrebbe espressa con una risoluzione di solo 1 Hz, ossia di una parte su dieci, mentre una frequenza di 10 kHz ver-

troppo, però, dovremmo attendere ben 10 o 100 secondi tra una lettura e la successiva! La soluzione ottimale è a portata di mano e consiste, semplicemente, nell'utilizzare in modo differente il Timer1. Invece di usarlo per contare gli impulsi esterni, lo sfruttiamo per misurare il tempo che intercorre tra due fronti di salita del segnale in ingresso (Fig. 6). Il tempo viene ricavato facendo contare al timer gli impulsi del clock interno del microcontrollore, secondo la seguente sequenza:

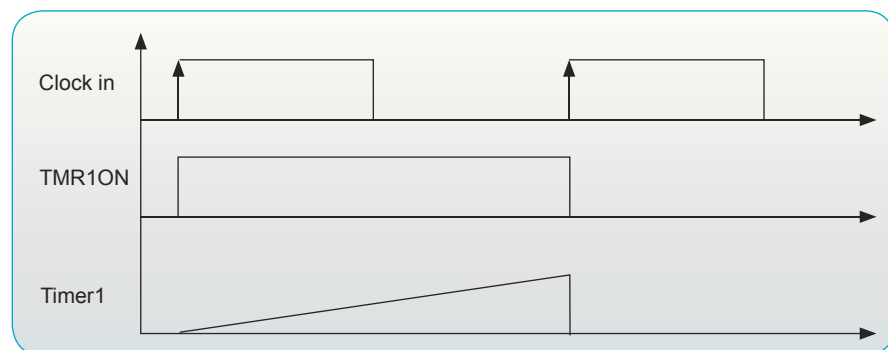
- Rileva fronte di salita del segnale in ingresso,
- TMR1ON=1, abilita conteggio,
- Rileva successivo fronte di salita del segnale in ingresso,
- TMR1ON=0, ferma conteggio,
- Lettura del valore contenuto nei registri del contatore

A questo punto, però, è necessario effettuare alcuni calcoli: la frequenza del quarzo (*osc*) è pari a 4 MHz, la frequenza di conteggio di Timer1 è di $\text{osc}/4$ cioè 1 MHz. Utilizziamo il prescaler per dividere per 8 tale frequenza ed otteniamo 125 kHz. Quest'ultima rappresenta la frequenza con la quale il Timer1 effettuerà il conteggio. Se chiamiamo *Conta* la variabile che contiene il valore del Timer1, possiamo scrivere

$$f = 125.000 / \text{Conta}$$

da cui si ricava che più la frequenza è bassa, maggiore è il suo periodo e, di conseguenza, maggiore è il valore della variabile *Conta*. Appare così evidente che, più è bassa la frequenza del segnale, maggiore è la risoluzione, situazione questa che è l'esatto contrario della situazione precedente. Per motivi pratici di calcolo, la variabile *Conta* assumerà tutti i valori tra 0 e 32.767 permettendo di misurare una frequenza minima di $125.000/32.767 = 3,815 \text{ Hz}$ da cui il limite minimo dichiarato di 5 Hz. Per frequenze inferiori, sul display comparirà la scritta "Under Range". Teoricamente è possibile scendere al di sotto di questo limite solo incremen-

Fig. 6 - Contatore di periodo: Timer1 conta gli impulsi (a frequenza costante e nota) compresa tra due fronti di salita del segnale in ingresso.



divisione per otto, ed impostando il tempo di Gate a 80 msec. così facendo si arriva a misurare frequenze da 0 Hz a 6.553.500 Hz (pari a 6,5535 MHz) a con risoluzione di 100 Hz.

La portata successiva, utilizzabile solo nella versione con AD8561, prevede un tempo di gate di 8 msec, tempo che permette misure sino a 65,535 MHz con precisione di 1

rebbe misurata con una risoluzione di 1/10000. Una bella differenza!

Per ovviare a questo inconveniente, la prima cosa che viene in mente è quella di aumentare il tempo di Gate, portandolo ad esempio a 10 e magari anche a 100 secondi: in questo modo si potrebbero visualizzare gli Hz con una risoluzione del decimo e del centesimo di Hz (10,00 Hz). Pur-

piano di MONTAGGIO versione 50 MHz

Le dimensioni del circuito stampato in versione da 50 MHz sono leggermente più piccole di quello da 5 MHz, in quanto il comparatore di ingresso non necessita della resistenza di pull-up sull'uscita (R10, qui non riportata). Durante il montaggio, fate attenzione al verso dei componenti polarizzati.

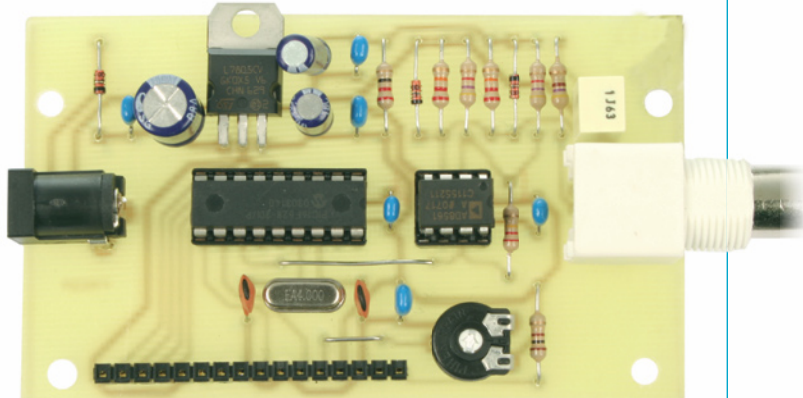
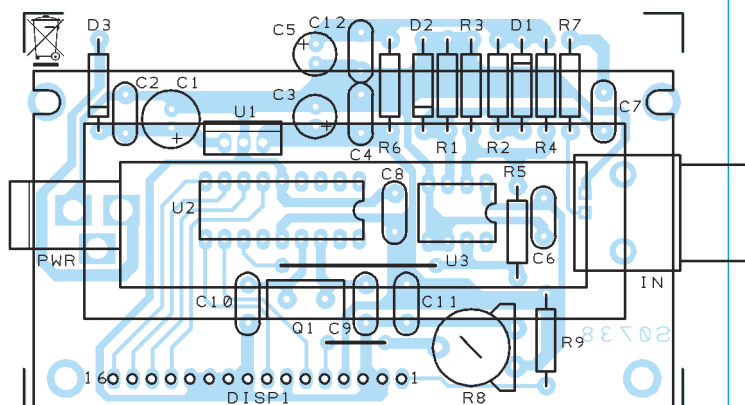
ELENCO COMPONENTI:

R1, R2: 22 kohm
R3, R4: 4,7 kohm
R5: 220 kohm
R6: 22 ohm
R7: 470 ohm
R8: Trimmer MO 10 kohm
R9: 100 ohm
C1: 220 μ F 25 VL elettrolitico
C2, C4, C8, C11: 100 nF multistrato
C3: 10 μ F 25 VL elettrolitico
C4, C8, C11: 100 nF multistrato
C5: 10 μ F 25 VL elettrolitico
C6, C12: 100 nF multistrato
C7: 1 μ F 50 VL poliestere

C9, C10: 22 pF ceramico
U1: 7805
U2: PIC16F628 (MF738)
U3: AD8561A
D1÷D3: 1N4148
Q1: Quarzo 4 MHz
basso profilo
DISP1: Display LCD 16x2

Varie:

- Zoccolo 9+9
- Zoccolo 4+4
- Presa BNC 90° da CS
- Plug alimentazione
- Circuito stampato
- Strip maschio 16 vie
- Strip femmina 16 vie



piano di MONTAGGIO versione 5 MHz

Attenzione: le illustrazioni qui a fianco sono ridotte al 90% circa delle dimensioni reali. Nonostante molti componenti siano comuni alle due versioni, abbiamo preferito riportare due elenchi componenti distinti in modo da ridurre il rischio di errori. Notate come la sezione digitale e quella di alimentazione dei due circuiti siano praticamente identiche.

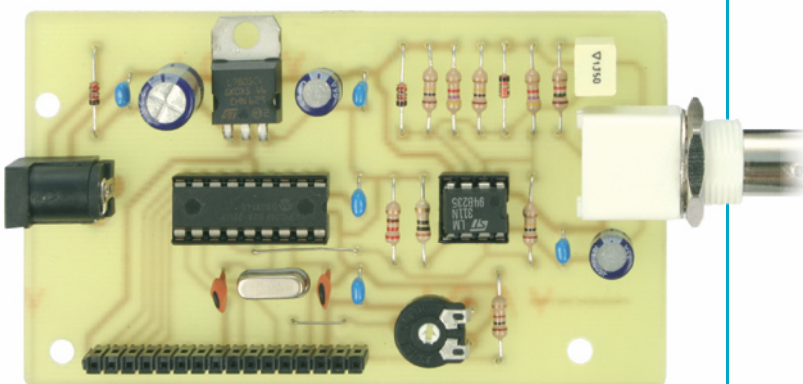
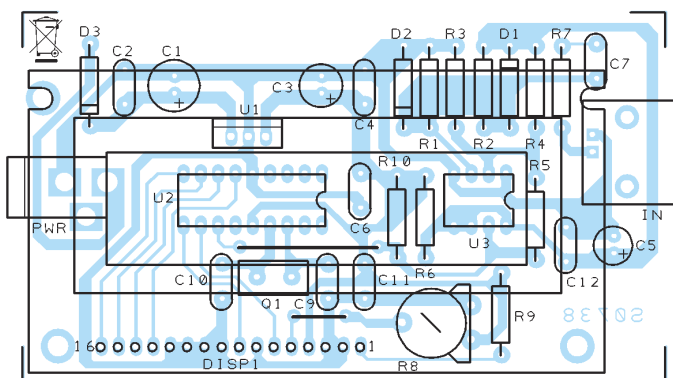
ELENCO COMPONENTI:

R1, R2, R5: 100 kohm
R3, R4: 4,7 kohm
R6: 10 ohm
R7: 1 kohm
R8: Trimmer MO 10 kohm
R9: 100 ohm
R10: 220 ohm
C1: 220 μ F 25 VL elettrolitico
C2, C4, C8, C11: 100 nF multistrato
C3: 10 μ F 25 VL elettrolitico
C4, C8, C11: 100 nF multistrato
C5: 10 μ F 25 VL elettrolitico
C6, C12: 100 nF multistrato
C7: 1 μ F 50 VL poliestere

C9, C10: 22 pF ceramico
U1: 7805
U2: PIC16F628 (MF738)
U3: LM311
D1÷D3: 1N4148
Q1: Quarzo 4 MHz basso profilo
DISP1: Display LCD 16x2

Varie:

- Zoccolo 9+9
- Zoccolo 4+4
- Presa BNC 90° da CS
- Plug alimentazione
- Circuito stampato
- Strip maschio 16 vie
- Strip femmina 16 vie



tando la possibilità di conteggio della variabile *Conta* oppure diminuendo il clock di sistema per il conteggio di Timer1. I cambi di portata sono facilmente gestibili, considerando che la condizione di overflow da parte di Timer1 è sempre verificabile controllando lo stato del bit TMR1F. Per completezza, abbiamo elencato in Tabella 1 le varie portate implementate in questo frequenzimetro con la relativa risoluzione. Non abbiamo, invece, riportato il codice sorgente in quanto è piuttosto lungo ma molto ripetitivo. A chi volesse approfondire l'argomento, diciamo che con le nozioni appena fornite e col codice sorgente modificare le portate e le risoluzioni per adattarle alle proprie esigenze non sarà difficile.

Realizzazione pratica

Osservando gli schemi elettrici degli stadi di ingresso possiamo notare che la pin-out dei due comparatori è leggermente differente; in fase di realizzazione dei prototipi potevamo adottare due differenti possibili opzioni: realizzare un circuito stampato di dimensioni maggiori, ma in grado di ospitare sia L'AD8651 che l'LM311, oppure realizzare due circuiti stampati distinti, ma più compatti e facili da realizzare. Abbiamo optato per i due circuiti differenti, dei quali potete, come sempre, scaricare i master monofaccia in scala 1:1 dal sito della rivista (www.elettronicaain.it).

A prescindere da quale dei due circuiti avrete realizzato con la tecnica

Tabella 1 - Scale di lettura della frequenza

Range	Frequenza minima	Frequenza massima	Unità di misura	Risoluzione
1	0	< 5	Under Range	-
2	3,99	99,99	Hz	0,01 Hz
3	100	999	Hz	1 Hz
4	1,000	9,999	KHz	1 Hz
5	10,000	65,535	KHz	1 Hz
6	65,53	655,35	KHz	10 Hz
7	0,6553	6,5535	MHz	100 Hz
8*	6,553	65,535	MHz	1000 Hz
9	-	> 65,535	Over Range	-

* Solo nella versione a 50 MHz, con stadio di ingresso AD8651.

della fotoincisione (oppure, vista la semplicità del circuito, anche con la tracciatura manuale), dopo aver forato le varie piazzole, montate per primi i componenti a basso profilo (diodi, resistenze, condensatori non polarizzati...) facendo attenzione al verso dei componenti polarizzati. Procedete quindi saldando i due zoccoli (uno da 8 pin DIL per il comparatore e l'altro da 18 pin DIL per il PIC) ed i condensatori elettrolitici. Ultimate il montaggio con il connettore jack per l'alimentazione, il BNC per l'ingresso del segnale, ed infine il pin strip femmina a 16 vie (passo 2,54 mm), in cui andrà inserito il display, dopo che vi avrete saldato il pin strip maschio (Fig. 7). A questo punto, siete pronti per accendere il frequenzimetro ma, prima, verificate attentamente che la realizzazione sia stata eseguita correttamente, aiutandovi con le illustrazioni riportate nel piano di montaggio. Senza inserire né i circuiti integrati né il display, alimentate il circuito (9 Vdc, 200 mA) e verificate

che la tensione sui pin di alimentazione dei chip sia di 5 V. Spegnete il circuito e inserite nelle rispettive sedi i due chip ed il display: il microcontrollore deve essere programmato con il codice esadecimale MF738.HEX, scaricabile anch'esso dal sito della rivista (insieme al codice sorgente in linguaggio PicBasic, riservato però agli abbonati). Riaccendete e, dopo aver regolato il contrasto del display agendo sul trimmer Rx (che si trova sotto il display e che, pertanto, va tarato attraverso un forellino sul circuito stampato), verificate che appaia la scritta "Under Range". Applicando all'ingresso un segnale variabile di ampiezza adeguata, vedrete apparire sul display l'indicazione esatta della frequenza.



Fig. 7

per il

MATERIALE

Il materiale utilizzato in questo progetto è facilmente reperibile presso qualsiasi negozio di elettronica. Il master del circuito stampato ed il firmware col quale programmare il microcontrollore possono essere scaricati dal sito della rivista (www.elettronicaain.it). Il microcontrollore già programmato (cod. MF738) può essere acquistato al costo di 12.00 Euro, IVA compresa.

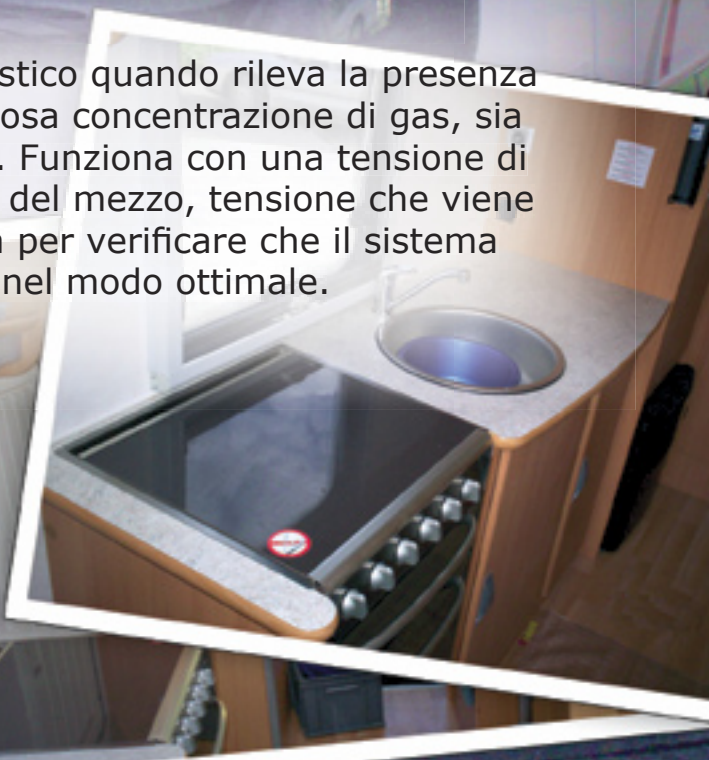
Il materiale va richiesto a: Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)

Tel: 0331-799775 ~ Fax: 0331-792287 ~ <http://www.futurashop.it>

Allarme gas per camper e roulotte

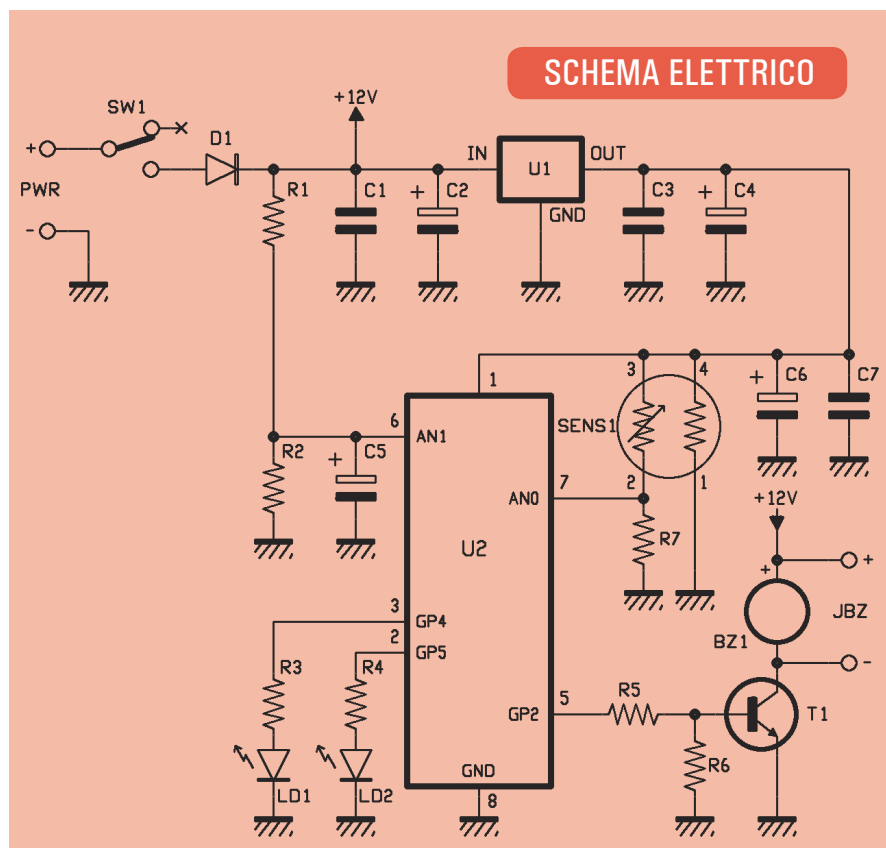
di *Francesco Doni*

Genera un forte segnale acustico quando rileva la presenza nell'ambiente di una pericolosa concentrazione di gas, sia esso metano, butano o GPL. Funziona con una tensione di 12 volt fornita dalla batteria del mezzo, tensione che viene costantemente monitorata per verificare che il sistema funzioni sempre nel modo ottimale.



Ogni roulotte o camper che si rispetti è una piccola casa e come in ogni casa troviamo un letto, un tavolo, un bagno e una cucina, con lavello e piano di cottura che solitamente è a gas (seppure a bombola di GPL o butano, usato nelle cucine da campeggio); non è quasi mai elettrica perché (pur essendo più sicura) per funzionare dovrebbe utilizzare una tensione a 220 volt, disponibile solamente nelle piazzole attrezzate: prelevare per questo scopo l'energia dalla batteria del camper o della vettura che traina la roulotte, infatti, è impensabile, dato che un fornello elettrico consuma mediamente 1200 watt, per ri-

cavare i quali un inverter richiederebbe, alla batteria da 12V, oltre 100A di corrente. Questa è la ragione per cui si preferisce il gas, che pure comporta non pochi rischi. A migliorare e complicare nel contempo la vita di chi soggiorna in camper e roulotte si è aggiunto, negli ultimi anni, un accessorio che, se da un lato fa trascorrere notti non più da brivido (di freddo!) dall'altro aggiunge rischi a quelli già relativi al fornello: si tratta del riscaldamento a gas, che viene realizzato mediante una sorta di piccola caldaia capace anche di produrre l'acqua calda sanitaria. È chiaro che la presenza del gas in uno spazio così ri- ➤



dotto richiede ancora più cautela e la necessità di prendere le dovute precauzioni; con la sola cucina, sarebbe sufficiente chiudere la valvola della bombola e tutto filerebbe liscio, ma quando si va in giro in autunno e inverno e si desidera il riscaldamento? Qui sorgono i problemi, perché la bombola deve restare aperta e, sebbene in un camper o una roulotte ben costruiti la caldaia debba stare in un vano isolato, le fughe di gas non si possono escludere. Ecco quindi sorgere la necessità di qualcosa che possa rilevare eventuali perdite e dare l'allarme quando la concentrazione supera una soglia pericolosa sia per gli effetti dell'inalazione che per il rischio di deflagrazione alla prima scintilla. Quel qualcosa lo trovate in queste pagine.

Schema elettrico

Il circuito è basato su un sensore Figaro TGS2610 e un microcontrollore cui è affidata la gestione delle varie funzionalità. Il sensore di gas è composto da una membrana di allumina

rivestita di biossido metallico e semiconduttore, riscaldata mediante un filamento. L'elemento riscaldante è necessario perché, a temperatura più alta, il sensore è più sensibile ai gas che deve rilevare. Il componente ha quattro terminali, due per la membrana di biossido ed altrettanti per il filamento; per poterlo utilizzare occorre alimentare il riscaldatore, preferibilmente a tensione costante, in modo da prevenire fluttuazioni di temperatura. Ponendo un resistore in serie allo strato di biossido, si possono facilmente rilevare le variazioni di concentrazione dei gas combustibili perché in loro presenza la membrana sensibile varia la propria conducibilità in funzione della concentrazione della sostanza gassosa cui viene esposta: per l'esattezza, maggiore è la quantità di gas (in termini di ppm, ossia "parti per milione"), minore diviene la resistenza.

Nel circuito da noi realizzato, l'elemento sensibile è collegato in serie a una resistenza da 1 kohm (R7) con la quale fa partitore, in modo da tra-

sformare le variazioni resistive in un cambiamento della tensione letta dall'ingresso AN0 del microcontrollore U1; quest'ultimo è un PIC12F675 programmato in modo da leggere la differenza di potenziale del sensore secondo un particolare metodo che ignora la naturale deriva termica. La lettura sfrutta l'A/D converter che legge, alternativamente, lo stato del sensore e il livello (mediante il partitore R1/R2) della tensione di alimentazione. L'ADC del micro, però, alla fine legge solamente una tensione che dipende da quanto rilevato dal sensore ma che comunque viene influenzata da molto fattori esterni quali la deriva termica dell'ambiente il cui il sensore opera (non stiamo parlando della temperatura del filamento riscaldante!), l'umidità relativa nell'aria, la presenza di più gas a concentrazioni differenti, eccetera. Tenere conto di tutti i fattori appena citati migliorerebbe certamente l'efficienza del circuito, ma complicherebbe non poco il firmware. Abbiamo quindi ipotizzato una certa stabilità (in termini di temperatura ed umidità) dell'ambiente di utilizzo e, sulla base delle caratteristiche specificate dal costruttore, abbiamo provveduto a realizzare un firmware in grado di calcolare la resistenza del sensore in aria fresca e di ricavare quella di riferimento (R0, che rappresenta la resistenza del sensore in presenza di 1800 ppm di isobutano) che, secondo le caratteristiche, è circa 1/10 della resistenza in aria fresca. Contem-



Calcolare la concentrazione del gas

Il nostro sensore, un TGS2610 Figaro, è composto da una membrana di allumina rivestita di biossido metallico e semiconduttore, riscaldata da un filamento: questo perché la parte sensibile, se riscaldata, diventa più sensibile ai gas. Il componente ha due elettrodi per la membrana di biossido ed altrettanti per il filamento (Fig. a); il riscaldatore va alimentato a 5 Vcc. Ponendo un resistore in serie allo strato di biossido, si possono facilmente rilevare le variazioni di concentrazione dei gas combustibili perché in loro presenza la membrana sensibile varia la propria resistenza, riducendola in funzione della concentrazione della sostanza gassosa cui viene esposta. L'elemento sensore rileva bene la presenza nell'aria circostante di gas combustibili anche in basse concentrazioni (100 ppm) quindi è più che adatto a captare perdite di gas che, seppure non letali, richiedono attenzione.

Il circuito applicativo è illustrato in Fig. b e corrisponde a quello usato dal nostro progetto per permettere al microcontrollore di rilevare il gas; la tensione V_{RL} sale all'aumentare della concentrazione di gas e viceversa. Il software acquisisce il valore di riposo (che può essere molto diverso da sensore a sensore a causa della tolleranza nel processo produttivo) poi legge la V_{RL} ciclicamente per controllare la concentrazione di gas. La verifica dello stato di allarme viene effettuata via software, calcolando di volta in volta il rapporto R_S/R_0 , in cui R_S rappresenta la resistenza del sensore in funzione della concentrazione di gas, mentre R_0 rappresenta la resistenza presentata in presenza di 1800 PPM di isobutano, a 20 °C e col 65% di umidità relativa. Dal grafico di Fig. c si ricava che R_0 corrisponde circa a 1/10 della resistenza in aria fresca. Quando il rapporto R_S/R_0 scende meno di 2, si è in presenza di una concentrazione di butano/propano di circa 1000 ppm.

Fig. a

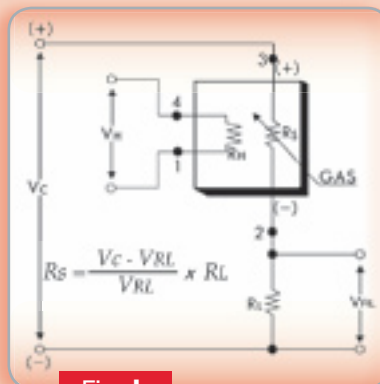


Fig. b

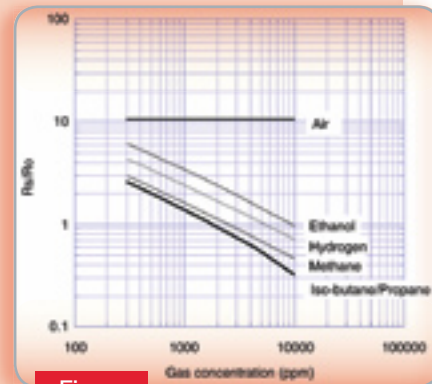


Fig. c

poraneamente il firmware controlla continuamente il sensore e ne calcola la resistenza media su 10 letture consecutive, effettuate a distanza di 500 msec l'una dall'altra, così da elaborare un valore medio che sia compensato dagli effetti della deriva termica e dell'umidità relativa ambientale. Ricalcola, infine, il rapporto R_S/R_0 e genera un allarme qualora questo valga meno di 2 (250 ppm di metano). Chiarito il meccanismo di lettura, proseguiamo nella nostra analisi dello schema elettrico: dicevamo che la gestione dell'insieme è affidata al

microcontrollore U1, che legge ciclicamente la differenza di potenziale ai capi della R7 e avvia le azioni del caso, gestendo le segnalazioni ottiche e acustiche prodotte da led e cicalino. L'alimentazione è in continua, e consiste in una tensione di 12 volt (sono richiesti 150 milliampère di corrente) applicata tra i punti + e - PWR; il diodo D1 protegge dall'inversione di polarità, e i condensatori C1 e C2 filtrano i disturbi. Il regolatore integrato U1 serve per ricavare i 5 volt stabilizzati necessari al micro e al filamento riscaldatore del sensore.

Come funziona

Il dispositivo va collocato all'interno del camper (nel vano di soggiorno) o roulotte, tipicamente in basso (basta un metro di altezza dal pavimento) dato che il gas usato nelle cucine trasportabili è più pesante dell'aria. Appena alimentato, il software del microcontrollore inizializza le linee di I/O e attende quaranta secondi affinché il sensore si porti alla temperatura di regime, ossia stabilizzi la sua resistenza tra i contatti 3 e 4; a questo punto si calibra, nel senso che con il proprio A/D converter acquisisce il ➤

per il

MATERIALE

Il materiale utilizzato in questo progetto è facilmente reperibile presso qualsiasi negozio di elettronica. Il master del circuito stampato ed il firmware col quale programmare il microcontrollore possono essere scaricati dal sito della rivista (www.elettronica.in.it). Il microcontrollore già programmato (cod. MF732) può essere acquistato al costo di 12.00 Euro, IVA compresa.

Il materiale va richiesto a: Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)

Tel: 0331-799775 ~ Fax: 0331-778112 ~ <http://www.futurashop.it>

Listato 1

Il programma esegue poche semplici operazioni. Per prima cosa inizializza lo stato degli elementi di segnalazione (buzzer e led); quindi attende 40 secondi, durante i quali il sensore TGS2610 viene preriscaldato. Calcola, quindi la resistenza del sensore in aria fresca e il valore R0. Avvia, poi, il ciclo principale che controlla lo stato della batteria e segnala tramite brevi pulsazioni del led rosso se questa è scarica. Procedo leggendo il sensore e verificando il valore medio (in 5 sec.) della concentrazione di gas: la situazione di allarme viene segnalata dal buzzer e dal led rosso, attivati e disattivati a cadenza di 200 msec ON e 200 msec OFF. Quando la concentrazione di gas scende al di sotto della soglia di allarme ($R_s/R_0 < 2$), il segnale acustico/visivo assume la cadenza di 500 msec e perdura per 30 secondi, trascorsi

```

***** INIZIALIZZAZIONE *****
CLEAR          ' Cancella la RAM.
LOW LEDV       ' Spegne led rosso.
HIGH LEDR      ' Accende led verde.
LOW BUZZ       ' Spegne il buzzer.
PAUSE 500      ' Pausa di 500 msec.
***** PRERISCALDAMENTO *****
FOR TMP=0 TO 200
  TOGGLE LEDV  ' Riscaldamento sensore 40 sec.
  TOGGLE LEDR  ' Fa lampeggiare il led verde.
  TOGGLE LEDR  ' Fa lampeggiare il led rosso.
  PAUSE 200    ' Pausa di 200 msec.
NEXT TMP       ' Ripete il ciclo.
LOW LEDV       ' Spegne led verde.
LOW LEDR      ' Spegne led rosso.
GOSUB LETTURA ' Legge il sensore.
GOSUB CALCRES  ' Calcola la resistenza.
RES0=RESS/10   ' Calcola R0
***** LOOP PRINCIPALE *****
MAIN:
  PULSOUT LEDV,500 ' Programma principale.
  ADCIN 1,BATT     ' Accende led verde 500 msec.
  GOSUB LETTURA   ' Legge tensione batteria
  GOSUB VERIFICA   ' Va a leggere il sensore.
  IF ALLARME>0 THEN ' Verifica quanto letto
    ALLARME=ALLARME-1 ' Esiste condizione allarme?
    TOGGLE LEDR      ' Si. Decrementa contatore 1sec.
    TOGGLE BUZZ      ' Accende/spegne il led rosso.
  ELSE              ' Accende/spegne il buzzer.
    PAUSA=5000      ' Non esiste allarme.
    LOW LEDR        ' Pausa di 5 secondi.
    LOW BUZZ        ' Spegne il led rosso.
  ENDIF            ' Spegne il buzzer.
                  ' Fine controllo allarme.

IF BATT<48 THEN   ' Batteria meno di 10,5 V?
  PULSOUT LEDR,500 ' Fa pulsare il led rosso.
ENDIF             ' Fine controllo batteria.
PAUSE PAUSA       ' Attende per il tempo "pausa".
GOTO MAIN         ' Riprende il ciclo principale.
***** VERIFICA CONCENTRAZIONE GAS *****
VERIFICA:
  GOSUB CALCRES   ' Calcola la resistenza.
  RAPP=(RESS*10)/RES0 ' Rs/R0, per 10 per evitare
                    ' risultati inferiori a 1.
  IF RAPP<2 THEN  ' RAPP=2 250 PPM METANO
    ALLARME=30    ' RAPP=1 1800 PPM isobutano
    PAUSA=200     ' Se meno di 250 ppm metano.
                  ' allarme ancora 30 sec.
  ELSE            ' Pausa di 200 msec
    PAUSA=500     ' Altrimenti
                  ' Pausa 500 msec.
  ENDIF           ' Fine verifica soglia.
  RETURN          ' Ritorna al main.
***** LETTURA E CALCOLO DELLA MEDIA *****
LETTURA:
  MEDIA=0
  FOR TMP=1 TO 10
    ADCIN 0,ADC
    MEDIA=MEDIA+ADC
    PAUSE 500
  NEXT TMP
  ADC=MEDIA/10
  RETURN
CALCRES:
  RESS=((255/ADC)*RES7)-RES7 ' RES7 è costante (R7=1k).
  RETURN                    ' Ritorna al loop principale.

```

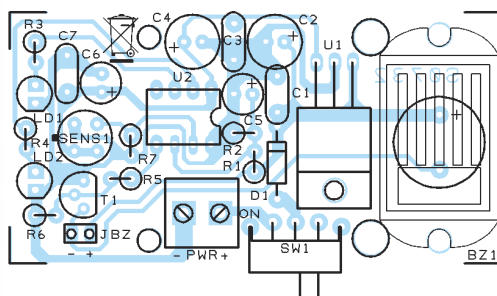
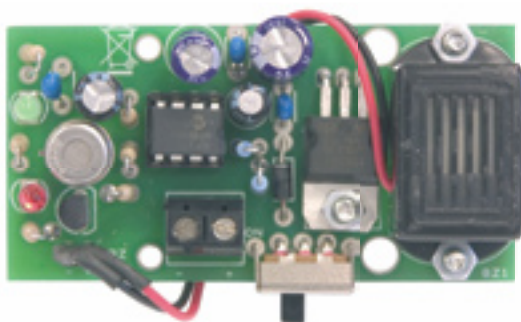
valore di tensione ai capi della resistenza di carico R7, che corrisponde alla condizione di riposo: durante questa fase è necessario che il sensore sia immerso in aria fresca, ossia priva, per quanto possibile, di tracce di gas a cui il sensore è sensibile. L'attesa e

l'esecuzione della calibrazione sono segnalate dal lampeggio alternato dei due led, ognuno dei quali pulsa al ritmo di 0,2/0,2 secondi, trascorsi i quali si entra nella fase operativa: fin quando non viene rilevata una condizione di allarme, il led verde

pulsa per circa mezzo secondo ogni cinque secondi. Quando il sensore rileva una concentrazione pericolosa nell'aria circostante, varia la sua resistenza interna e tale variazione si ripercuote sulla tensione letta dall'ADC del PIC che, pertanto, emette

piano di MONTAGGIO

Per contenere l'ingombro del PCB, abbiamo optato per il montaggio verticale di diodi e resistenze. Abbiamo anche previsto un interruttore di alimentazione.



ELENCO COMPONENTI:

R1: 10 kohm 1%
R2: 1 kohm 1%
R3: 470 ohm
R4: 470 ohm
R5: 4,7 kohm
R6: 10 kohm
R7: 6,8 kohm

C1: 100 nF multistrato
C2: 220 µF 25 VL elettrolitico
C3: 100 nF multistrato
C4: 220 µF 16 VL elettrolitico
C5: 1 µF 100 VL elettrolitico
C6: 100 µF 100 VL elettrolitico
C7: 100 nF multistrato
D1: 1N4007

T1: BC547
U1: 7805
U2: PIC12F675 (MF732)
LD1: led 3 mm verde
LD2: led 3 mm rosso
SW1: Deviatore a slitta 90°
SENS1: Sensore TGS2610
BZ1: Buzzer 12V (SVM12)

Varie:
- Morsettiere 2 poli
- Zoccolo 4+4
- Strip maschio 2 vie
- Strip femmina 2 vie
- Vite 8 mm 3 MA
- Dato 3 MA
- circuito stampato

una segnalazione sia acustica che luminosa: in pratica accende e spegne sia il led rosso che il buzzer a cadenza di 200 msec ON e 200 msec OFF. Tale cadenza, insieme al suono emesso dal buzzer, è adeguata per richiamare l'attenzione di chi è sveglio e sta guardando la TV o chiacchierando, ovvero di destare anche chi ha un sonno piuttosto pesante. Naturalmente non sono stati previsti relé o altri elementi che potrebbero causare scintille: l'avviso ottico e quello acustico sono sicuri, perché non innescano effetti che possano determinare esplosioni. Una particolarità molto interessante del nostro allarme è che mantiene l'avviso ottico e acustico sia durante la fase critica della concentrazione di gas, sia dopo che essa è scesa al di sotto della soglia di pericolosità: in questo secondo caso, la segnalazione viene mantenuta solamente per trenta secondi e la cadenza viene rallentata, in modo da segnalare una situazione di "rientrato allarme". Il protrarsi dell'avviso serve per consentire all'allarme di farsi sentire anche nel caso in cui ci sia un momentaneo superamento della soglia di pericolo, troppo breve perché una persona, magari distratta, possa accorgersene. Anche se sarebbe superfluo farlo, non possiamo esimerci dal ricordarvi che il sensore è in allarme quando rileva una concentrazione di gas considerata pericolosa, per cui, in questo caso, areate bene l'ambiente evitando di attivare qualsiasi dispositivo che possa generare scintille ed innescare pericolose esplosioni. Ulteriore dettaglio di pregio del nostro circuito è il controllo dinamico dello stato della batteria: il programma che gira nel microcontrollore e che si occupa di analizzare l'aria alla ricerca di gas, monitorizza la tensione di alimentazione e fornisce un allarme se si accorge che non è sufficiente al corretto funzionamento, ovvero se rileva a valle del D1 meno di 10,5 volt. Tale funzione serve sia quando il circuito viene alimentato con una

propria batteria, sia quando si prelevano i 12 V dall'impianto elettrico di un camper.

Realizzazione pratica

Per quel che riguarda la costruzione, la prima cosa da fare è preparare il circuito stampato previsto, del quale potete scaricare dal nostro sito Web (www.elettronica.in.it) la traccia lato rame in scala 1:1 per ricavare la pellicola necessaria alla fotoincisione. Una volta ottenuta la basetta, potete montare su di essa, in ordine, le resistenze, il diodo al silicio e lo zoccolo per il microcontrollore, quindi i condensatori (prestando particolare attenzione alla polarità di quelli elettrolitici) il transistor e il regolatore integrato 7805, da disporre sdraiato con il lato metallico appoggiato al piano dello stampato. Quanto ai due led, disponeteli in piedi ricordando che il catodo è il terminale che sta dal lato smussato del contenitore. Il sensore di gas va montato dal lato saldature, orientandolo in modo che la sua tacca di riferimento sia rivolta alla resistenza R4. Per le connessioni di alimentazione prevedete una morsettiera per circuito stampato a passo 5 millimetri. Una volta completato il montaggio e verificatane l'esattezza, si può dare tensione (occorrono da un minimo di 11,5 a un massimo di 15 Vcc) e attendere che smetta il lampeggio veloce 0,2/0,2 secondi dei due led; a questo punto il dispositivo è pronto ad operare. Come prova empirica, potete avvicinare al sensore un accendino e liberare del gas per qualche secondo. Entro cinque secondi viene avviata la segnalazione di allarme secondo la procedura che abbiamo già descritto. Date le dimensioni contenute del dispositivo, non sarà difficile inserirlo in qualsiasi contenitore di dimensioni adeguate. Sappiate, comunque, che noi lo abbiamo progettato per essere inserito in un contenitore plastico SC703, facilmente reperibile ovunque.

CD3+
Disturbatore Cellulare
All Bands
23W totali min.

Vendita
soggetta
a
limitazioni



NEW



PD-1
Rilevatore tascabile
Cellulare e
microspie
GSM, DCS, UMTS
1+3000 Mhz
Portata 5+10m
Vibra-beep-led
Batteria ricaricabile

DESPY

Ideale per
la ricerca
di micro spie
fino a 3 Ghz



Bias s.d. www.bias.sm

Strada del Lavoro 119-sub2
Zona Industriale Gualdicciolo
47894-Chiesanuova
Repubblica di San Marino
Tel. 0549-999408 - Fax. 0549-999431

Alla scoperta del Protocollo PS/2

dell' *Ing. Renzo Morelli*

Il Protocollo PS/2 viene utilizzato in tutto il mondo per far colloquiare un PC con la propria tastiera. Nonostante la grande diffusione, però, non tutti sono a conoscenza del modo in cui funziona. Leggendo questo articolo ne scoprirete tutti i segreti; verranno infatti analizzate le caratteristiche hardware, le funzioni, i comandi, i codici utilizzati nonché le procedure di avvio e di riprogrammazione della tastiera.

Alcuni anni fa era abbastanza difficile credere che si sarebbe arrivati, un giorno, ad avere uno o più personal computer per ogni famiglia. Ebbene oggi è proprio così: le persone che non hanno mai avuto a che fare con un personal computer sono sempre di meno, la diffusione dei PC è in continua crescita ed i segreti di questa prodigiosa macchina sono sempre più rari. Il PC, però, per poter essere utilizzato, necessita di una tastiera, periferica usata da tutti ma conosciuta veramente da pochi. Gli utenti di PC, infatti, sanno cos'è e come si usa, ma quanti sanno effettivamente come funziona? Segui-

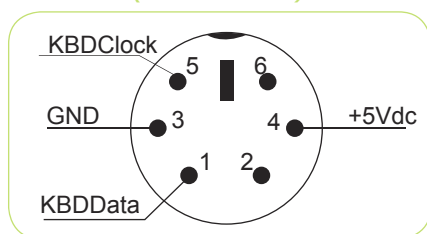
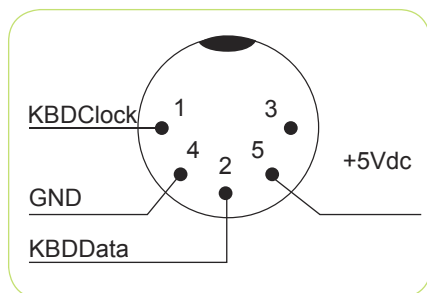
tecì in questo articolo e sarete tra quelli: stiamo infatti per svelarvi i segreti del protocollo con cui una tastiera standard PS/2 ed un PC si scambiano dati ogni volta che viene premuto un tasto. Poi, come nostra consuetudine, con le informazioni ottenute realizzeremo un progetto pratico che verrà pubblicato in un articolo successivo: vi anticipiamo che si tratta di una tastiera a 16 tasti con uscita PS/2, da collegare ad un PC (da sola o in parallelo ad una tastiera standard), i cui tasti sono liberamente programmabili dall'utente per riproporre una sequenza qualsiasi di tasti premuti. La programmazione delle se- ➤

Fig. 1a

Tastiera Microsoft con 114 tasti (standard + speciali).

**Fig. 1b**

Tastiera PS/2 con 114 Tasti e Scan Code Set 2.

**Fig. 2** Mini-DIN 6 Poli Maschio (vista frontale).**Fig. 3** DIN 5 Poli Maschio (vista frontale).

quenze avviene in autoapprendimento, identificando il tasto di interesse e poi premendo sulla tastiera del PC la sequenza di comandi che vogliamo associare a quel tasto. Per realizzare quanto sopra, però, è necessario conoscere perfettamente il modo in cui un PC e la sua tastiera dialogano tra loro: a ciò abbiamo dedicato le pagine seguenti.

La connessione Tastiera/PC-Host

La tastiera PS/2 viene collegata al PC (che da ora in poi chiameremo

Tabella 1 - Pin-out connettori Tastiera AT e PS/2

Segnale	Pin AT DIN 5 poli	Pin PS/2 Mini-DIN 6 poli
KBDDData	2	1
Non Connesso	3	2
GND	4	3
+5Vdc	5	4
KBDClock	1	5
Non Connesso		6

Tabella 2 - Stati Tastiera

CLOCK	DATA	Stato
Alto	Alto	Tastiera può trasmettere
Alto	Basso	Host vuole trasmettere
Basso	Alto	Host non pronto (Busy)
Basso	Basso	Stato indefinito

anche Host) tramite un cavo dotato di connettore circolare Mini-DIN maschio a 6 poli (Fig. 2) dei quali solo quattro sono utilizzati: alimentazione (+5 Vdc), massa (GND), Clock (KBDclk) e Data (KBDDa). La calza (schermo) è collegata alla carcassa del connettore e collega a terra la massa della tastiera. A titolo informativo, vi diciamo che una tastiera AT (antenna della PS/2 ed obsoleta da anni) dispone di un connettore circolare DIN maschio a 5 poli (Fig. 3), più grande di quello PS/2, ma con gli stessi segnali elet-

trici, anche se disposti su pin diversi (Tab. 1).

La comunicazione tra tastiera PS/2 e Host è seriale sincrona bidirezionale e fa uso di due linee di comunicazione (Clock e Data) bidirezionali open-collector in cui il Clock è sempre generato dalla tastiera. La scelta di linee open collector è stata fatta in modo da consentire sia alla tastiera che al PC di segnalare l'intenzione di trasmettere dei dati: ad esempio, l'Host può forzare bassa la linea Clock per interrompere una trasmissione da tastiera oppure può forzare bassa la linea Data per segnalare che vuol trasmettere un comando. La tastiera, dal canto suo, può trasmettere solo quando rileva che entrambi gli ingressi Clock e Data sono a livello alto (Tab. 2)

L'alimentazione (+5 Vdc, max. 300 mA) della tastiera viene sempre fornita dal PC.

Trasmissione di dati da Tastiera a Host

Quando una tastiera PS/2 deve trasmettere un byte al PC, per prima cosa disattiva le linee di Clock e Data (entrambe poste in Idle Mode) che si portano così a livello logico alto (grazie a delle resistenze di pull-up interne alla tastiera), a meno che l'Host non voglia trasmettere (Data Low) o non sia pronto a ricevere (Clock Low).

Quando la tastiera rileva che entrambe le linee sono a livello logico alto (condizione che, come abbiamo visto, consente la trasmissione dell'informazione relativa al tasto premuto) essa invia l'informazione usando il protocollo seriale sincrono a 11 bit visibile in Fig. 4 in cui lo Start Bit (primo della serie) è sempre a zero logico. Seguono quindi gli otto bit dati, il primo dei quali è sempre il bit meno significativo (LSB); la sequenza procede con il bit di parità dispari - odd - che vale 1 se il numero di livelli logici 1 contenuti nei bit dati è pari. La fine della

trasmissione viene segnalata con lo Stop bit, sempre a livello logico 1. La trasmissione dei dati è sincronizzata sul segnale di Clock generato dalla tastiera, la quale invia ogni bit in corrispondenza del fronte di salita. L'Host, invece, legge i bit in corrispondenza del fronte di discesa. La frequenza del Clock va da 20 kHz a 30 kHz (da 50 μ sec a 33 μ sec per ciclo). Dopo il bit di stop,

durante un periodo in cui la tastiera si aspetta di trovarla a livello alto, essa interrompe immediatamente la trasmissione ed entra nella modalità di ricezione da Host. Può anche accadere che l'Host non sia pronto a ricevere perché momentaneamente impegnato ad eseguire altri processi: durante la trasmissione di ogni bit a livello alto, la tastiera verifica se l'Host forza la linea Data a livello

ma cosa mantiene a livello basso il Clock per almeno 60 msec, evitando così il rischio che la tastiera inizi a trasmettere contemporaneamente ad esso; mantiene bassa, poi, la linea Data fino a quando riceve il primo colpo di Clock trasmesso dalla tastiera e quindi inizia a trasmettere usando lo stesso protocollo sincrono ad 11 bit del caso precedente (Fig. 5). La trasmissione è ancora sin-

Fig. 4 *Protocollo Trasmissione da Tastiera ad Host.*

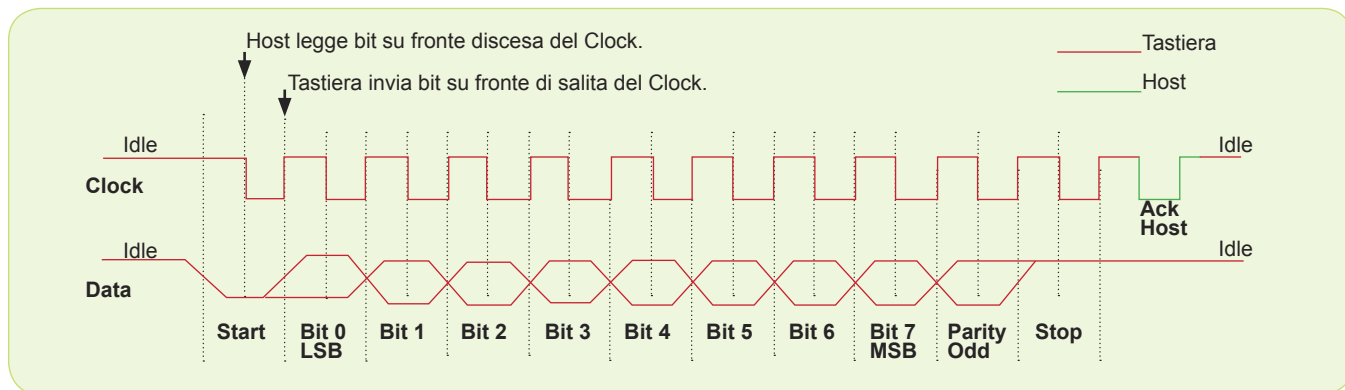
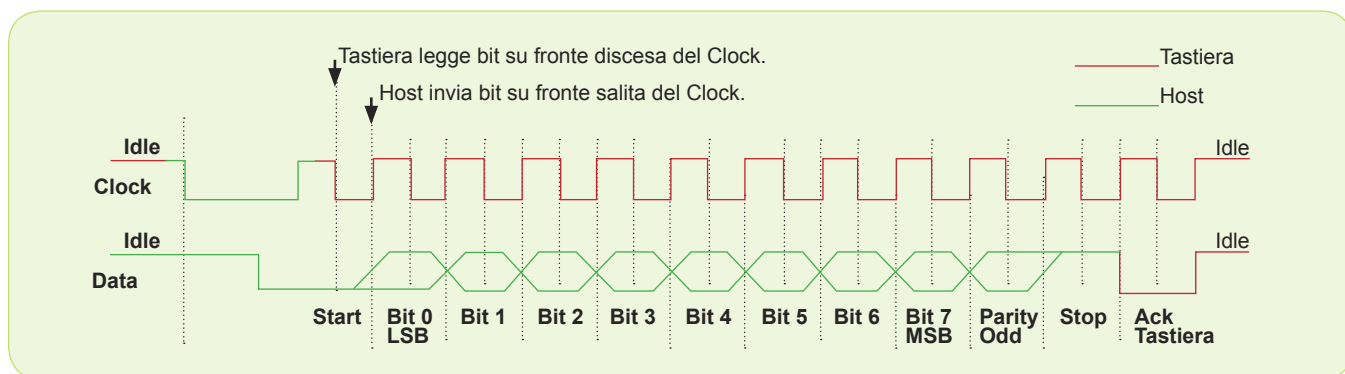


Fig. 5 *Protocollo Trasmissione da Host a Tastiera.*



la tastiera pone a livello logico alto sia il Clock che il dato, rimanendo in attesa che l'Host comunichi l'avvenuta ricezione (Ack) ponendo basso il Clock entro 50 μ sec dallo Stop bit.

Abbiamo detto che anche l'Host può richiedere di trasmettere dei dati alla tastiera, e ciò viene fatto forzando a livello logico basso il Clock: la tastiera, quindi, anche se si trova in trasmissione, deve verificare lo stato della linea di Clock durante ogni periodo a livello logico alto. Se viene trovata a livello basso

basso (Host non pronto); quando ciò accade, la tastiera interrompe la trasmissione e rimane in attesa di rivedere entrambe le linee a livello alto (Host pronto).

Qualora l'Host rilevi un errore di parità nel dato ricevuto, invia alla tastiera il comando RESEND, a cui la tastiera reagisce inviando nuovamente l'ultimo dato trasmesso.

Trasmissione di dati da Host a Tastiera

Quando l'Host deve trasmettere un comando alla tastiera, per pri-

cronizzata sul Clock generato dalla tastiera: l'Host invia i bit in corrispondenza del fronte di salita e la tastiera li legge in corrispondenza del fronte di discesa. Dopo il bit di Stop, l'Host pone alta (Idle) la linea Data e rimane in attesa che la tastiera comunichi l'avvenuta ricezione (Ack) inviando un bit a "0 logico" (un ciclo di Clock con dato=0). Se la linea Data non va in idle dopo lo Stop bit, la tastiera continuerà a generare cicli di Clock fino al verificarsi di tale condizione. Così come l'Host, anche la tastiera può rilevare ➤

un errore di parità nel dato ricevuto: in questo caso attende che la linea Data torni alta (Idle) e poi invia all'Host il comando RESEND, per chiedere la ritrasmissione del dato in errore.

Tabella 3 - Le tre famiglie di tastiere

Tipo Tastiera	Caratteristiche
XT (1981)	- 83 o 101 tasti. - Connettore 5 poli DIN. - Protocollo seriale unidirezionale. - Scan Code Set 1. - Nessun comando da Host a tastiera.
AT (1984)	- 101 o più tasti. - Connettore cinque poli DIN. - Protocollo seriale bidirezionale. - Scan Code Set 2. - Otto comandi da Host a tastiera. - Non compatibile con tastiera XT.
PS/2 (1987)	- 101 o più tasti. - Connettore 6 poli Mini DIN. - Protocollo seriale bidirezionale. - Scan Code Set 2 Standard. - Scan Code Set 3 Opzionale. - 17 comandi da Host a tastiera. - Compatibile con tastiera AT. - Non compatibile con tastiera XT.

Gli Scan Code

Ogni volta che viene premuto un tasto, la tastiera invia al PC (Host) una sequenza seriale di bit che rappresenta un codice (Scan Code) indicante la posizione di quel tasto sulla matrice della tastiera; se, poi, il tasto viene mantenuto premuto, la tastiera continua ad inviare lo stesso Scan Code (Autorepeat), con una frequenza di trasmissione detta Typematic Rate, dopo una pausa iniziale detta Typematic Delay.

Gli Scan Codes sono storicamente raggruppati in tre famiglie (Set

1, 2, 3, riportate in tabella 3) legate ai modelli di tastiera (XT, AT, PS/2) che via via si sono evoluti nel tempo.

Attualmente le tastiere più usate in ambito civile sono le PS/2, insieme alle AT ancora diffuse in ambito industriale. Esse differiscono solo per la dimensione ed il pin-out del connettore (DIN o mini-DIN) ed utilizzano entrambe lo Scan Code Set 2 (Tab. 4 e Tab. 5), il solo a cui d'ora in poi faremo riferimento.

L'invio dei Codici Tasto

Nelle tastiere attualmente in commercio, i tasti vengono codificati secondo lo Scan Code Set 2 in cui tutti i codici hanno dimensione di 1 byte, ad eccezione di un gruppo detto Extended Key, codificato con sequenze di due o più byte, in cui il primo vale sempre "0xE0".

Ogni volta che un tasto viene premuto, la tastiera invia al PC direttamente lo Scan Code di quel tasto; analogamente alla pressione, ogni volta che un tasto viene rilasciato, la tastiera invia il codice di tasto rilasciato "0xF0" (Break Code) seguito dallo Scan Code di quel tasto.

Se il tasto rilasciato è un Extended (codice che inizia con "0xE0") allora la tastiera invia prima "0xE0" poi "0xF0" (Break Code) seguito dal resto dei codici che seguono "0xE0" nello Scan Code di quel tasto. Nella Tabella 4 sono riportati gli Scan Code del Set 2 per tutti i 101 tasti (premuti e rilasciati) relativi alla classica tastiera PC.

Attualmente esistono molte tastiere

che contano più di 101 tasti: stiamo parlando dei modelli multimediali, dotati di tasti "particolari" per accedere rapidamente alle funzioni di cui i PC più recenti dispongono. Ad esempio il tasto "@" avvia il programma predefinito per la posta elettronica, mentre il tasto "Start" richiama il menu di avvio di Windows. Analogamente il tasto "Home" avvia il browser Internet.

Nella Tabella 5 sono riportati gli Scan Code dei 13 tasti di questo tipo, relativi alla tastiera 114 Tasti Microsoft riportata in Fig. 1a. Facciamo ora alcuni esempi di invio di codici da tastiera a PC, leggendo gli Scan Codes dalle Tabelle 4 e 5.

- ESC (Standard Key), Scan Code = 0x76. Quando è premuto, la tastiera invia "0x76"; quando è rilasciato, invece, invia "0xF0 0x76".
- LEFT (Extended Key), Scan Code = "0xE0 0x6B". Quando è premuto, la tastiera invia "0xE0 0x6B"; al rilascio viene inviato "0xE0 0xF0 0x6B".
- WINDOWS START (Extended Key), Scan Code = 0xE0 0x1F. Quando è premuto la tastiera invia "0xE0 0x1F"; il rilascio corrisponde a "0xE0 0xF0 0x1F".
- PRNT SCR (Print Screen), Scan Code = "0xE0 0x12 0xE0 0x7C". Alla pressione, la tastiera invia "0xE0 0x12 0xE0 0x7C"; al rilascio, invece, invia "0xE0 0xF0 0x7C 0xE0 0xF0 0x12".

La ricezione dei Codici su PC

Ogni volta che un tasto è premuto o rilasciato, il Sistema Operativo del PC legge lo Scan Code di tastiera ed invia all'applicazione in esecuzione il carattere ASCII corrispondente al simbolo disegnato sul tasto stesso, tenendo conto se contemporaneamente vengono premuti (e non rilasciati) i tasti "Shift, Alt, AltGr e Ctrl". Ad esempio, quando il tasto "e" è premuto/rilasciato, la tastiera trasmette sempre lo stesso Scan Code 0x24 (0xF0 0x24), ma il Siste-

Oggi esistono tastiere specializzate per applicazioni grafiche, tecniche o ludiche (come la Logitech G15 qui a fianco). La mole di dati scambiati con il PC è superiore rispetto alle tastiere standard e pertanto necessitano di una connessione veloce (USB) e di un applicativo/driver specifico, ma il protocollo base rimane il PS/2.



ma Operativo lo interpreta in quattro modi differenti a seconda che siano premuti, contemporaneamente al tasto “e”, i tasti Shift (“E” maiuscola, 0x45), Ctrl (“ENQ”, 0x05), AltGr

(“€”, 0x0012) o nessuno di essi (“e” minuscola, 0x65). Il tasto “Alt”, in unione agli altri tasti, invece, genera dei codici di comando utilizzati dai vari programmi in esecuzione. Per

ogni tasto sono quindi associati fino a cinque caratteri differenti, che portano il totale “tasti” a $5 \times 114 = 570$, numero di gran lunga superiore al numero massimo di caratteri ASCII ➤

Tabella 4a - Scan Codes per Tastiera PS/2 101 tasti Set 2

TASTO	Premuto		Rilasciato			TASTO	Premuto		Rilasciato		
Descrizione	1°	2°	1°	2°	3°	Descrizione	1°	2°	1°	2°	3°
ESC	76		F0	76		HOME [Tastierino]	6C		F0	6C	
F1	05		F0	05		UP [Tastierino]	75		F0	75	
F2	06		F0	06		PGUP [Tastierino]	7D		F0	7D	
F3	04		F0	04		+ [Tastierino]	79		F0	79	
F4	0C		F0	0C		CAPS	58		F0	58	
F5	03		F0	03		A	1C		F0	1C	
F6	0B		F0	0B		S	1B		F0	1B	
F7	83		F0	83		D	23		F0	23	
F8	0A		F0	0A		F	2B		F0	2B	
F9	01		F0	01		G	34		F0	34	
F10	09		F0	09		H	33		F0	33	
F11	78		F0	78		J	3B		F0	3B	
F12	07		F0	07		K	42		F0	42	
PRT SCR	E0,12	E0,7C	E0,F0,7C	E0,F0,12		L	4B		F0	4B	
SCROLL LOCK	7E		F0	7E		ò ç	4D		F0	4D	
PAUSE	E1	14,77	E1	F0,14	F0,77	à °	52		F0	52	
	0E		F0	0E		ù \$	5D		F0	5D	
1 !	16		F0	16		ENTER	5A		F0	5A	
2 "	1E		F0	1E		CANC	E0	71	E0	F0	71
3 £	26		F0	26		PGDW	E0	7A	E0	F0	7A
4 \$	25		F0	25		LEFT [Tastierino]	6B		F0	6B	
5 %	2E		F0	2E		VUOTO [Tastierino]	73		F0	73	
6 &	36		F0	36		RIGHT [Tastierino]	74		F0	74	
7 /	3D		F0	3D		SHIFT SX	12		F0	12	
8 (	3E		F0	3E		< >	61		F0	61	
9)	46		F0	46		Z	1A		F0	1A	
0 =	45		F0	45		X	22		F0	22	
' ?	4E		F0	4E		C	21		F0	21	
ì ^	55		F0	55		V	2A		F0	2A	
BACKSPACE	66		F0	66		B	32		F0	32	
HOME	E0	6C	E0	F0	6C	N	31		F0	31	
END	E0	69	E0	F0	69	M	3A		F0	3A	
NUM [Tastierino]	77		F0	77		, ;	41		F0	41	
/ [Tastierino]	E0	4A	E0	F0	4A	. :	49		F0	49	
* [Tastierino]	7C		F0	7C		- _	4A		F0	4A	
- [Tastierino]	7B		F0	7B		SHIFT DX	59		F0	59	
TAB	0D		F0	0D		UP	E0	75	E0	F0	75
Q	15		F0	15		END [Tastierino]	69		F0	69	
W	1D		F0	1D		DW [Tastierino]	72		F0	72	
E	24		F0	24		PGDW [Tastierino]	7A		F0	7A	

Tabella 4b - Scan Codes per Tastiera PS/2 101 tasti Set 2

TASTO		Premuto		Rilasciato		TASTO		Premuto		Rilasciato	
Descrizione	1°	2°	1°	2°	3°	Descrizione	1°	2°	1°	2°	3°
R	2D		F0	2D		ENTER [Tastierino]	E0	5A	E0	F0	5A
T	2C		F0	2C		CTRL SX	14		F0	14	
Y	35		F0	35		ALT SX	11		F0	11	
U	3C		F0	3C		SPACE	29		F0	29	
I	43		F0	43		ALT GR	E0	11	E0	F0	11
O	44		F0	44		CTRL DX	E0	14	E0	F0	14
P	4D		F0	4D		LEFT	E0	6B	E0	F0	6B
è é	54		F0	54		DW	E0	72	E0	F0	72
+ *	5B		F0	5B		RIGHT	E0	74	E0	F0	74
INS	E0	70	E0	F0	70	INS [Tastierino]	70		F0	70	
PGUP	E0	7D	E0	F0	7D	CANC [Tastierino]	71		F0	71	

disponibili (256). Per poterli gestire è stato pertanto introdotto un gruppo di caratteri con codice a 2 byte il primo dei quali è sempre 0x00 che indica Tasto doppio o Esteso. La conversione da Scan Code a carattere del simbolo disegnato sul tasto, viene effettuata dal S.O., che conosce il layout (associazione carattere/

due layout differenti (Italia e USA), utilizzabili entrambi ma uno solo per volta. Nelle Tabelle da 7a a 7d sono riportati tutti i codici che il S.O. invia all'applicazione in esecuzione quando riceve uno Scan Code, nei vari casi di Shift, Ctrl, Alt, AltGr attivi, sia per tastiera italiana che per tastiera USA. Nelle tabelle abbiamo

“0x40, offset 0x17 e offset 0x18” (vedere Tabella 6), vengono utilizzati dal S.O. per scegliere il carattere fra i cinque possibili per ogni Scan Code. La lettura di queste particolari locazioni di memoria, normalmente delegata ad una libreria specifica per Compilatore, fornisce in uscita il nome del tasto, il suo stato (premuto

Tabella 5 - Scan Codes speciali per Tastiera PS/2 114 tasti Set 2

TASTO		Premuto		Rilasciato	
Descrizione	1°	2°	1°	2°	3°
Windows 1	E0	48	E0	F0	48
Windows 2	E0	3A	E0	F0	3A
Windows 3	E0	18	E0	F0	18
Windows 4	E0	34	E0	F0	34
Windows 5	E0	32	E0	F0	32
Windows 6	E0	21	E0	F0	21
Windows 7	E0	23	E0	F0	23
Windows 8	E0	50	E0	F0	50
Windows 9	E0	40	E0	F0	40
Windows 10	E0	2B	E0	F0	2B
Windows Start	E0	1F	E0	F0	1F
Windows Application	E0	2F	E0	F0	2F
@	4C		F0	4C	

Scan Code) della tastiera, installata dall'operatore selezionando il tipo (tastiera Inglese, Italiana, Tedesca, eccetera) tramite il Pannello di controllo di Windows all'opzione Tastiera/Lingua.

Le Fig. 6a e 6b, mostrano l'esempio di un PC in cui sono stati installati

indicato con Flag gli otto tasti che non producono caratteri, ma che vengono memorizzati nei bit dei due registri Indicatori di Status Tastiera Kb-Flag e Kb-Flag1. Tali registri, che possono essere letti dall'applicazione in esecuzione direttamente in memoria agli indirizzi segmento

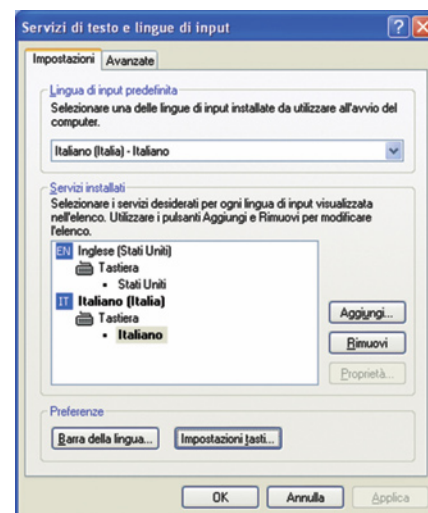


Fig. 6a - I sistemi operativi recenti possono gestire tastiere differenti.

o rilasciato), lo stato della tastiera, eccetera.

Nella Tabella 7 abbiamo inserito i nomi standard dei caratteri ASCII non stampabili (Caratteri di controllo), che per comodità riportiamo nella Tabella 8 insieme alle combinazioni di tasti che li generano.

Fig. 6b - È anche possibile definire delle impostazioni base differenti per ogni tastiera.

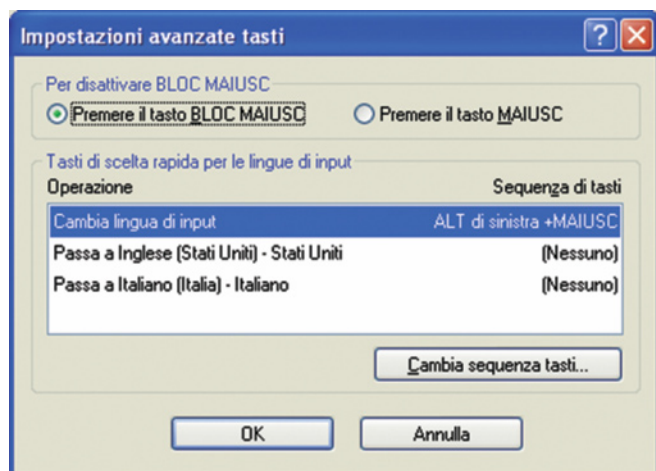


Tabella 6 - Registri KB-FLAG e KB-FLAG1

KB-FLAG (Indirizzo 40:17)		KB-FLAG1 (Indirizzo 40:18)	
Bit	Significato	Bit	Significato
7	INS Attivo.	7	INS Premuto.
6	CAPS Lock Attivo.	6	CAPS Lock Premuto.
5	NUM Lock Attivo.	5	NUM Lock Premuto.
4	SCROLL Lock Attivo.	4	SCROLL Lock Premuto.
3	ALT Premuto.	3	Non usato.
2	CTRL Premuto.	2	Non usato.
1	SHIFT Sx Premuto.	1	Non usato.
0	SHIFT Dx Premuto.	0	Non usato.

Comandi da Tastiera a Host

Oltre ai codici dei tasti, la tastiera invia al PC i seguenti comandi:

- 0x00 (OVERRUN): inviato dalla tastiera quando il buffer tasti è pieno.
- 0xAA (INIT OK): inviato al PC se il test automatico all'accensione della tastiera è positivo.
- 0xEE (ECHO): è la risposta della tastiera alla ricezione del comando Echo inviato dall'Host.
- 0xFA (ACK): è la risposta della tastiera dopo la ricezione corretta di un comando (diverso da ECHO e da RESEND) inviatole dall'Host. Se il comando ricevuto non è valido o si è verificato un errore di

parità, la tastiera invia RESEND invece che ACK.

- 0xFD (FAILURE): inviato all'Host quando la tastiera ha rilevato un guasto. Se questo avviene durante il test all'accensione, la tastiera invia FAILURE, poi interrompe la scansione dei tasti e rimane in attesa di ricevere un co- ➤

Tabella 7a - Conversione da Scan Codes a Caratteri ASCII per tastiera 114 tasti Italiana ed USA

KEY	Scan	Tasto		Tasto+ Shift		Tasto+ Ctrl		Tasto+ Alt		Tasto+ AltGr	
Descrizione	Code	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII	
Windows 1	E0,48										
Windows 2	E0,3A										
Windows 3	E0,18										
Windows 4	E0,34										
Windows 5	E0,32										
Windows 6	E0,21										
Windows 7	E0,23										
Windows 8	E0,50										
Windows 9	E0,40										
Windows 10	E0,2B										
ESC	76	ESC	1B	ESC	1B						
F1	05	F1	00,3B	Shift F1	00,54	Ctrl F1	00,5E	Alt F1	00,68		
F2	06	F2	00,3C	Shift F2	00,55	Ctrl F2	00,5F	Alt F2	00,69		
F3	04	F3	00,3D	Shift F3	00,56	Ctrl F3	00,60	Alt F3	00,6A		
F4	0C	F4	00,3E	Shift F4	00,57	Ctrl F4	00,61	Alt F4	00,6B		
F5	03	F5	00,3F	Shift F5	00,58	Ctrl F5	00,62	Alt F5	00,6C		
F6	0B	F6	00,40	Shift F6	00,59	Ctrl F6	00,63	Alt F6	00,6D		
F7	83	F7	00,41	Shift F7	00,5A	Ctrl F7	00,64	Alt F7	00,6E		
F8	0A	F8	00,42	Shift F8	00,5B	Ctrl F8	00,65	Alt F8	00,6F		
F9	01	F9	00,43	Shift F9	00,5C	Ctrl F9	00,66	Alt F9	00,70		
F10	09	F10	00,44	Shift F10	00,5D	Ctrl F10	00,67	Alt F10	00,71		

Tabella 7b - Conversione da Scan Codes a Caratteri ASCII per tastiera 114 tasti Italiana ed USA

KEY	Scan	Tasto		Tasto+Shift		Tasto+Ctrl		Tasto+Alt		Tasto+AltGr	
Descriz	Code	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII	Hex
F11	78	F11	00,85	Shift F11	00,87	Ctrl F11	00,89	Alt F11	00,8B		
F12	07	F12	00,86	Shift F12	00,88	Ctrl F12	00,8A	Alt F12	00,8C		
PRT SCR	E0,12, E0,7C	Copia Screen		Stampa Screen		Stampa Screen					
SCROLL	7E	Flag									
PAUSE	E1,14, 77	Pausa				Break					
(ITA) , ~ (USA)	0E	(ITA) , (USA)	5C 4E	(ITA) ~ (USA)	7C 7E						
1 !	16	1	31	!	21						
2 " (ITA) 2 @ (USA)	1E	2	32	" (ITA) @ (USA)	22 40						
3 £ (ITA) 3 # (USA)	26	3	33	£ (ITA) # (USA)	9C 23						
4 \$	25	4	34	\$	24						
5%	2E	5	35	%	25						
6 & (ITA) 6 ^ (USA)	36	6	36	& (ITA) ^ (USA)	26 5E	RS	1E				
7 / (ITA) 7 & (USA)	3D	7	37	/ (ITA) & (USA)	2F 26						
8 ((ITA) 8 * (USA)	3E	8	38	((ITA) * (USA)	28 2A						
9) (ITA) 9 ((USA)	46	9	39	) (ITA) ((USA)	29 28						
0 = (ITA) 0) (USA)	45	0	30	= (ITA)) (USA)	3D 29						
' ? (ITA) , _ (USA)	4E	' (ITA) , (USA)	60 2D	? (ITA) , (USA)	3F 5F	US	1F				
ì ^ (ITA) = + (USA)	55	ì (ITA) = (USA)	8D 3D	^ (ITA) + (USA)	5E 2B						
BACKSPACE	66	BS	08	BS	08	Ctrl Bk Sp	00,7F	Alt Bk Sp	00,0E		
HOME	E0,6C	Home	00,47	Home	00,47	Ctrl HOME	00,77	Alt HOME	00,97		
END	E0,69	END	00,4F	END	00,4F	Ctrl END	00,75	Alt END	00,9F		
NUM	77	Flag									
/	E0,4A	/	2F	/	2F	Ctrl /	00,95	Alt /	00,A4		
*	7C	*	2A	*	2A	Ctrl *	00,96				
.	7B	.	2D	.	2D	Ctrl .	00,8E	Alt .	00,4A		
TAB	0D	HT	09	Shift TAB	00,0F	Ctrl TAB	00,94	Alt TAB	00,A5		
Q	15	q	71	Q	51	DC1	11				
W	1D	w	77	W	57	ETB	17				
E € (ITA) E (USA)	24	e	65	E	45	ENQ	05	Alt E	00,12	€ (ITA)	
R	2D	r	72	R	52	DC2	12				
T	2C	t	74	T	54	DC4	14				
Y	35	y	79	Y	59	EM	19	Alt Y	00,15		
U	3C	u	75	U	55	NAK	15				
I	43	i	69	I	49	HT	09				
O	44	o	6F	O	4F	SI	0F				
P	4D	p	70	P	50	DLE	10				
è é [(ITA) [{ (USA)	54	è (ITA) [(USA)	8A 5B	é (ITA) { (USA)	82 7B	ESC (ITA)	1B			[(ITA)	5B

Tabella 7c - Conversione da Scan Codes a Caratteri ASCII per tastiera 114 tasti Italiana ed USA

KEY	Scan	Tasto		Tasto+Shift		Tasto+Ctrl		Tasto+Alt		Tasto+AltGr	
Descriz	Code	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII	Hex
+ *] (ITA) } (USA)	5B	+ (ITA)] (USA)	2B 5D	* (ITA) { (USA)	2A 7D	GS	1D			] (ITA)	5D
ENTER	5A	CR	0D	CR	0D	LF	0A	Alt ENTER	00,1C		
INS	E0,70	Ins	00,52	Ins	00,52	Ctrl INS	00,92	Alt INS	00,A2		
PGUP	E0,7D	Page Up	00,49	Page Up	00,49	Ctrl PGUP	00,84	Alt PGUP	00,99		
HOME 7	6C	Home	00,47	7	37	Ctrl HOME	00,77				
UP 8	75	Freccia Up	00,48	8	38	Ctrl UP	00,8D				
PGUP 9	7D	Page Up	00,49	9	39	Ctrl PGUP	00,84				
+	79	+	2B	+	2B	Ctrl +	00,90	Alt +	00,4E		
CAPS	58	Flag									
A	1C	a	61	A	41	SOH	01				
S	1B	s	73	S	53	DC3	13				
D	23	d	64	D	44	EOT	04				
F	2B	f	66	F	46	ACK	06				
G	34	g	67	G	47	BEL	07				
H	33	h	68	H	48	BS	08				
J	3B	j	6A	J	4A	LF	0A				
K	42	k	6B	K	4B	VT	0B				
L	4B	l	6C	L	4C	FF	0C				
ò ç @ (ITA) ; : (USA)	4C	ò (ITA) ; (USA)	95 3B	ç (ITA) : (USA)	80 3A					@ (ITA)	40
à ° # (ITA) , " (USA)	52	à (ITA) ' (USA)	85 27	° (ITA) " (USA)	A7 22					# (ITA)	23
ù \$	5D	ù	97	\$	15	FS	1C				
CANC	E0,71	Cancella	00,53	Cancella	00,53	Ctrl CANC	00,93	Alt CANC	00,A3		
PGDN	E0,7A	Page Dw	00,51	Page Down	00,51	Ctrl PGDN	00,76	Alt PGDN	00,A1		
LEFT 4	6B	Freccia Sx	00,4B	4	34	Ctrl LEFT	00,73				
CENTER 5	73	Center	00,4C	5	35	Ctrl CENTER	00,8F				
RIGHT 6	74	Freccia Dx	00,4D	6	36	Ctrl RIGHT	00,74				
SHIFT SX	12	Flag									
< > (ITA) (USA)	61	< (ITA) (USA)	3C 1B	> (ITA) (USA)	3E 7C						
Z	1A	z	7A	Z	5A	SUB	1A	Alt Z	00,2C		
X	22	x	78	X	58	CAN	18	Alt X	00,2D		
C	21	c	63	C	43	ETX	03				
V	2A	v	76	V	56	SYN	16				
B	32	b	62	B	42	STX	02				
N	31	n	6E	N	4E	SO	0E				
M	3A	m	6D	M	4D	CR	0D				
, ; (ITA) , < (USA)	41	,	2C	; (ITA) < (USA)	3B 3C						
. : (ITA) , > (USA)	49	.	2E	: (ITA) > (USA)	3A 3E						

>

mando o un nuovo restart.

- 0xFE (RESEND): è la risposta della tastiera alla ricezione di un comando non valido o con errore di parità.

per modificare lo stato dei Led, escludendo il test iniziale che li fa accendere contemporaneamente per circa un secondo, così da consentire all'operatore di verificarne

non modifica lo stato attuale dei Led, processa il nuovo comando e riprende la scansione dei tasti (se ancora abilitata).

- 0xEE (ECHO): la tastiera rispon-

Tabella 7d - Conversione da Scan Codes a Caratteri ASCII per tastiera 114 tasti Italiana ed USA

KEY	Scan	Tasto		Tasto+Shift		Tasto+Ctrl		Tasto+Alt		Tasto+AltGr	
Descriz	Code	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII	Hex
· (ITA) / ? (USA)	4A	· (ITA) / ? (USA)	2D 5C	· (ITA) ? (USA)	5F 3F			Alt · (ITA)	00,35		
SHIFT DX	59	Flag									
UP	E0,75	Freccia Up	00,48	Freccia Up	00,48	Ctrl UP	00,8D	Alt UP	00,98		
END 1	69	END	00,4F	1	31	Ctrl END	00,75				
DW 2	72	Freccia Dw	00,50	2	32	Ctrl DW	00,91				
PGDN 3	7A	Page Dw	00,51	3	33	Ctrl PGDN	00,76				
ENTER	E0,5A	CR	0D	CR	0D	LF	0A	Alt ENTER	00,A6		
CTRL SX	14	Flag									
Windows Start	E0,1F										
ALT SX	11	Flag									
SPACE	29	Spazio	20	Spazio	20	Spazio	20	Spazio	20		
ALT GR	E0,11	Flag									
Windows App.	E0,2F										
CTRL DX	E0,14	Flag									
LEFT	E0,6B	Freccia Sx	00,4B	Freccia Sx	00,4B	Ctrl LEFT	00,73	Alt LEFT	00,93		
DW	E0,72	Freccia Dw	00,50	Freccia Dw	00,50	Ctrl DW	00,91	Alt DW	00,A0		
RIGHT	E0,74	Freccia Dx	00,4D	Freccia Dx	00,4D	Ctrl RIGHT	00,74	Alt RIGHT	00,9D		
INS 0	70	Ins	00,52	0	30	Ctrl INS	00,92				
CANC .	71	Cancella	00,53	.	2E	Ctrl CANC	00,93				

Comandi da Host a Tastiera

In risposta a particolari eventi, o durante l'aggiornamento dei parametri durante il normale funzionamento, una tastiera PS/2 può ricevere dal PC i seguenti comandi:

- 0x00÷0xEC (ILLEGAL): tutti comandi non validi. La risposta della tastiera alla ricezione di uno di questi comandi è sempre RESEND.
- 0xED (STATUS): composto da due byte, è utilizzato per la gestione dei led *Num Lock*, *Caps Lock* e *Scroll Lock* presenti sulla tastiera. Questo comando è il solo mezzo

il corretto funzionamento. Dopo la ricezione del primo byte (Status), la tastiera trasmette ACK (0xFA), interrompe la scansione dei tasti e rimane in attesa del secondo, così utilizzato:

Bit 0 (LSB): Led Scroll Lock
Bit 1: Led Num Lock
Bit 2: Led Caps Lock
Bit 3..7: Riservati

Dopo la ricezione, risponde con ACK, aggiorna i Led in base al dato ricevuto e riprende la scansione. Se invece riceve un comando diverso (byte >= 0xED), interrompe la sequenza di Status,

de inviando ECHO.

- 0xF2 (ID): l'Host richiede l'identificazione della tastiera, che risponderà inviando ACK seguito da "0xAB 0x83".
- 0xF3 (TYPEMATIC): inviando questo comando (anch'esso costituito da due byte), l'Host impone alla tastiera il Typematic Repeat Rate, composto dal primo ritardo (Typematic Delay) e dalla frequenza (Typematic Rate) con cui la tastiera invia lo Scan Code quando un tasto viene mantenuto premuto. Dopo la ricezione del primo byte (Typematic), la ta-

stiera trasmette ACK e rimane in attesa del secondo, che definirà il Delay ed il Rate del Typematic. Il secondo byte è strutturato in questo modo:

$Bit\ 0\div2 = Rx$
 $Bit\ 3,4 = Ry$
 $Bit\ 5,6 = Dx$
 $Bit\ 7 = 0$

e i vari bit vengono utilizzati per il calcolo dei tempi tramite le formule seguenti:

$Rate[Car/sec]=240/[(8+Rx)*(2^{Ry})]$
 $Delay[msec]=250*(Dx+1)$

I valori di default tipici sono:

$Rx = 4, Ry = 1$
(Rate = 10 caratteri/secondo),
 $Dx=1$
(Delay= 500 msec).

Dopo aver ricevuto il secondo byte, la tastiera invia ACK, aggiorna frequenza e ritardo con i nuovi valori ricevuti e resta in attesa di ricevere il comando ENABLE prima di riprendere la scansione dei tasti.

Se, invece, riceve un comando diverso dal byte con il ritardo ed il delay, allora interrompe la sequenza di Typematic e processa il nuovo comando senza modificare i suoi valori di rate e delay.

Anche in questo caso, però, rimane in attesa di ricevere il comando ENABLE, che consente di riprendere la scansione dei tasti.

- 0xF4 (ENABLE): la tastiera svuota il buffer, inizia nuovamente la scansione dei tasti e risponde all'Host inviando ACK.
- 0xF5 (DISABLE): la tastiera si resetta, interrompe la scansione dei tasti e risponde all'Host inviando ACK.
- 0xF6 (DEFAULT): la tastiera imposta i suoi parametri (Typematic Rate, ecc.) ai valori di default, continua la scansione dei tasti (se abilitata) e risponde all'Host inviando ACK.
- 0xFE (RESEND): quando la tastiera riceve questo comando, risponde inviando nuovamente al-

Tabella 8 - Caratteri ASCII di Controllo

Tasto+ Ctrl	Hex	Nome	Descrizione
^@	00	NUL	Null
^A	01	SOH	Start of Header
^B	02	STX	Start of Text
^C	03	ETX	End of Text
^D	04	EOT	End Of Trasmission
^E	05	ENQ	Enquiry
^F	06	ACK	Acknowledge
^G	07	BEL	Bell
^H	08	BS	Backspace
^I	09	HT	Horizontal Tab
^J	0A	LF	Line Feed
^K	0B	VT	Vertical Tab
^L	0C	FF	Form Feed
^M	0D	CR	Carriage Return
^N	0E	SO	Shift Out
^O	0F	SI	Shift In
^P	10	DLE	Data Link Escape
^Q	11	DC1	Device Control 1
^R	12	DC2	Device Control 2
^S	13	DC3	Device Control 3
^T	14	DC4	Device Control 4
^U	15	NAK	Negative Acknowledge
^V	16	SYN	Synchronous Idle
^W	17	ETB	End of Trasmission Block
^X	18	CAN	Cancel
^Y	19	EM	End of Medium
^Z	1A	SUB	Substitute
^[	1B	ESC	Escape
^\ (^_)	1C	FS	File Separator
^] (^_)	1D	GS	Group Separator
^6 (^_)	1E	RS	Record Separator
^. (^_)	1F	US	Unit Separator

l'Host l'ultimo byte trasmesso. Se quest'ultimo fosse proprio il comando RESEND trasmesso dalla tastiera, allora ritrasmetterà al PC il byte inviato ancora prima.

- 0xFF (RESET): inviando questo comando, l'Host impone il riavvio della tastiera che, dopo averlo ricevuto, risponde inviando ACK, interrompendo la scansione e rimanendo in attesa dell'accettazione del suo ACK da parte del PC. L'Host segnala l'accettazione della risposta mantenendo alte sia la linea del Clock che quella dei Data per almeno 500 microsecon-

di. Qualsiasi altro comando inviato alla tastiera interromperà la sequenza di Reset e verrà gestito normalmente.

Una volta che ACK è stato accettato, la tastiera continua la sequenza di Reset svuotando il buffer, riprendendo i parametri di default e resettando i Led.

Power On Reset

All'accensione, la tastiera esegue il test BAT (*Basic Assurance Test*) per verificare che:

- nessun tasto sia guasto (premuto),
- la RAM funzioni,



• il checksum della ROM sia valido. Durante il BAT, la linea di Clock viene mantenuta bassa dalla tastiera per impedire all'Host di trasmettere; vengono inoltre tenuti accesi tutti i led.

Alla fine del test (della durata di 300÷500 msec), la tastiera spegne i Led, riporta a livello alto la linea di Clock ed invia all'Host (entro 2,5 secondi dal Reset) il comando INIT OK (0xAA) nel caso in cui il test abbia dato esito positivo, o FAILURE (0xFD) nel caso opposto. Terminato il BAT, la tastiera rimane disabilitata in attesa di ricevere comandi.

Una possibile sequenza di comandi scambiati fra Host e tastiera durante la fase di accensione è riportata in Tabella 9.

Note conclusive

Giunti a questo punto, siete in possesso delle nozioni fondamentali su cui è basato il funzionamento di una tastiera PS/2: come avrete sicuramente notato, si tratta di un protocollo piuttosto semplice che presenta, per certi aspetti, dei tratti simili all'PC Bus.

Se sviluppate applicazioni firmware su PIC o su qualsiasi altra piattaforma a micro, grazie alle informazioni acquisite in queste pagine, potrete prendere in considerazione la possibilità di creare una libreria firmware per integrare in ogni vostro sistema una tastiera PS/2 standard, piuttosto che realizzare la classica tastie-

Tabella 9 - Scambio dati Tastiera/Host in avvio

Sorgente	Codice	Comando	Significato
Tastiera	AA	INIT OK	Test iniziale positivo.
Host	ED	STATUS	Inizio Sequenza Led.
Tastiera	FA	ACK	Acknowledge.
Host	00		Tutti i led off.
Tastiera	FA	ACK	Acknowledge.
Host	F2	ID	Richiesta identificatore.
Tastiera	FA	ACK	Acknowledge.
Tastiera	AB		Primo byte identificatore.
Tastiera	83		Secondo byte identificatore.
Host	ED	STATUS	Inizio sequenza led.
Tastiera	FA	ACK	Acknowledge.
Host	02		Led NumLock ON, gli altri OFF.
Tastiera	FA	ACK	Acknowledge.
Host	F3	TYPOMATIC	Programmazione autorepeat.
Tastiera	FA	ACK	Acknowledge.
Host	20		Delay: 500 msec Rate: 30 caratteri per secondo.
Tastiera	FA	ACK	Acknowledge.
Host	F4	ENABLE	Abilita tastiera.
Tastiera	FA	ACK	Acknowledge

ra a matrice, specifica per quella sola applicazione.

Come vi accennavamo all'inizio dell'articolo, nei prossimi mesi pubblicheremo un progetto pratico: si tratta di una piccola tastiera programmabile in grado di apprendere automaticamente le sequenze di Scan Code che devono essere as-

sociate ai suoi tasti funzionali. Sarà certamente una buona occasione per verificare in pratica come è stato realizzato un firmware in ambito "non PC" in grado di interagire con una tastiera PS/2 in assenza di sistema operativo. Appuntamento quindi alle prossime uscite, ci raccomandiamo: non mancate.



**4° SALONE
dell'ELETTRONICA**

con spazi dedicati:

- allo scambio del disco e CD • a radio e grammofoni d'epoca
- all'elettronica in genere

5 - 6 aprile 2008
al CENTRO FIERE - VILLA POTENZA
MACERATA

orari: sabato 14,30 - 19,00
domenica 09,30 - 19,00

abbinato alla 21ª edizione



segreteria organizzativa:  0733.63.73.13 - 335.373.156 - www.servicepromotion.it • info@servicepromotion.it

Set TX-RX per ascolto ambientale

di Alessandro Orlando

Sistema semplice ed economico composto da due unità, una trasmittente ed una ricevente, in grado di operare entro un raggio d'azione di alcune decine di metri. Il trasmettitore, alimentato direttamente con tensione di rete tramite un adattatore specifico, dispone di un microfono con caratteristiche professionali. Il ricevitore è dotato di controllo di volume e di squelch.



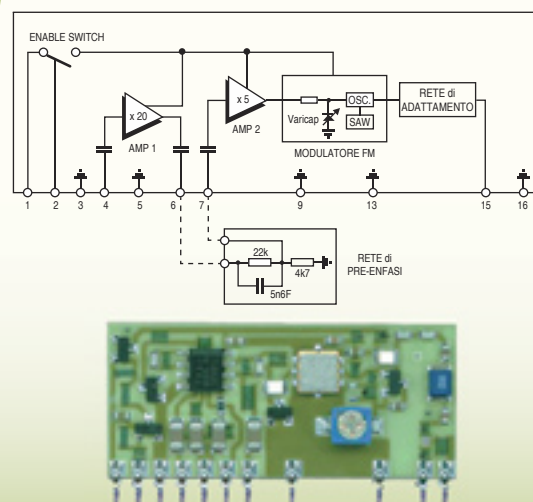
I nostri lettori ricorderanno certamente i numerosi progetti dedicati all'ascolto ambientale presentati negli anni passati sulla nostra rivista. Progetti di ogni genere, da quelli con caratteristiche professionali destinati a quanti sono autorizzati a svolgere questo genere di attività (Forze dell'Ordine su mandato della Magistratura), a quelli con intenti più didattici o che per le loro caratteristiche (leggi: portata) possono al massimo essere utilizzati come baby-monitor o come interfono all'interno di un'abitazione. In quest'ambito abbiamo anche presentato progetti destinati alla scoperta dei sistemi di

ascolto ambientale, sia con prestazioni professionali che con caratteristiche più ...casalinghe. L'ultimo progetto di questo genere è apparso sul fascicolo n. 120, un rivelatore di microspie tanto sensibile quanto economico e di facile realizzazione, operante nella gamma di frequenza comprese tra 100 MHz e 2,5 GHz. Questo genere di apparecchiature (sia i sistemi di trasmissione e ricezione che i rivelatori RF) hanno sempre interessato moltissimo i nostri lettori: la possibilità di trasmettere senza fili, anche a pochi metri di distanza, la voce (ma anche il video), esercita ancora un fascino irresistibile anche nel-

l'era dei cellulari. Per questo motivo la richiesta di circuiti a ciò destinati è una costante nelle lettere e nelle e-mail che giungono in redazione; ed è proprio per questo motivo che questo mese presentiamo il progetto di un sistema di ascolto ambientale via radio le cui caratteristiche si posizionano a metà strada tra quelle dei dispositivi professionali e quelle dei circuiti per impiego casalingo. In ogni caso questo progetto ha una valenza didattica non trascurabile: la costruzione del trasmettitore e del ricevitore consentono di prendere confidenza con circuiti e tecnologie (ah, l'alta frequenza!) poco usuali nell'era del digitale.

Questo progetto utilizza i moduli TX-FM-AUDIO e RX-FM-AUDIO prodotti da Aurel: si tratta di due dispositivi non recentissimi ma dalle ottime caratteristiche tecniche, facili da utilizzare e con un buon rapporto prezzo/prestazioni che li rendono adatti al nostro scopo. Nel trasmet-

Modulo TX-FM-Audio



PIN	Funzione
1	+ 12 volt
2	Abilitazione / Disabilitazione (Se messo a + VCC il modulo è in trasmissione.)
3	GND
4	Ingresso BF 1 stadio (Guadagno 20:1)
5	GND
6	Uscita 1 stadio BF
7	Ingresso BF 2 stadio (Guadagno 5:1)
9	GND
13	GND
15	Uscita antenna +10 dBm
16	GND

intensità emessi anche da sorgenti acustiche lontane dal trasmettitore.

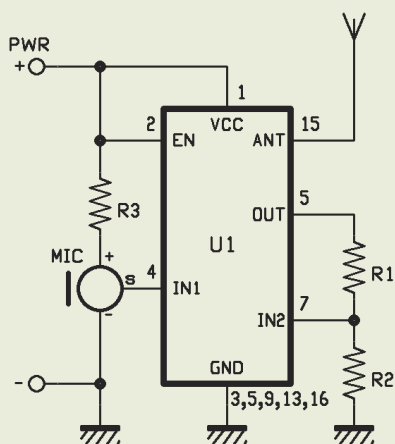
Il trasmettitore

Da una rapida occhiata allo schema elettrico, potete notare che il trasmettitore, oltre al modulo TX-FM, necessita di pochi altri componenti ossia del microfono, di tre resisten-

una portante avente frequenza centrale di 433 MHz e potenza di 10 dBm.

Nel riquadro in alto riportiamo lo schema a blocchi e la pin-out del modulo: il pin 2 (Enable) agisce su uno switch elettronico che attiva i due amplificatori di tensione (collegati in serie) che guadagnano 20 volte il

Trasmettitore



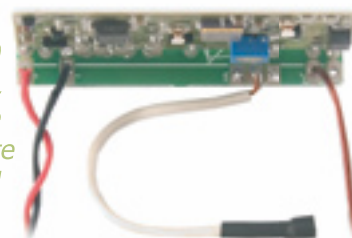
Saldare le resistenze SMD con un saldatore a punta fine da 30 W. Il microfono è molto piccolo, per cui saldate con cautela i tre fili sulle sue piazzole: proteggete sia i fili che il microfono con del termoretraibile.

ELENCO COMPONENTI:

R1, R2: 22 kohm (0805)
R3: 10 kohm (0805)
U1: TX-FM

Varie:

- Microfono Knowles
- Circuito stampato



titore, poi, abbiamo inserito una capsula microfonica Knowles ad elevate prestazioni e di dimensioni molto contenute, in modo da ottenere un'elevata gamma dinamica, che consente di captare suoni a bassa

ze e di un'antenna a filo. Descrivere il trasmettitore, pertanto, equivale a descrivere il modulo Aurel; si tratta un ibrido alimentato a 12 V, in grado di trasmettere un segnale audio (20÷20.000 Hz) modulando in FM

primo e 5 volte il secondo. Seguono il modulatore FM a diodo varicap e il risuonatore SAW che stabilizza la frequenza portante (controllati anch'essi dal pin di enable); per ultimo abbiamo lo stadio finale di potenza

Fig. 1-2 - Il microfono Knowles FG6163 ed il diagramma della caratteristica sensibilità/frequenza.

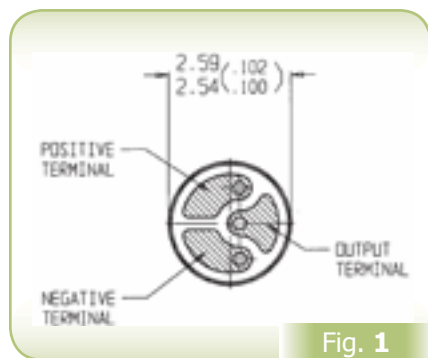


Fig. 1

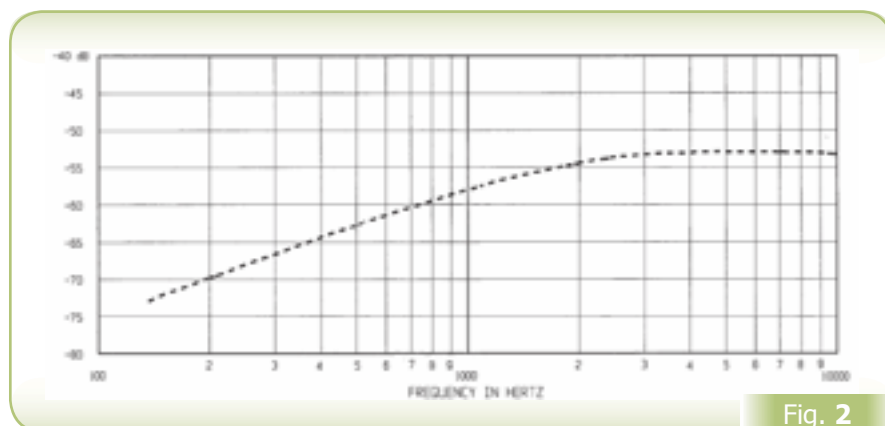


Fig. 2

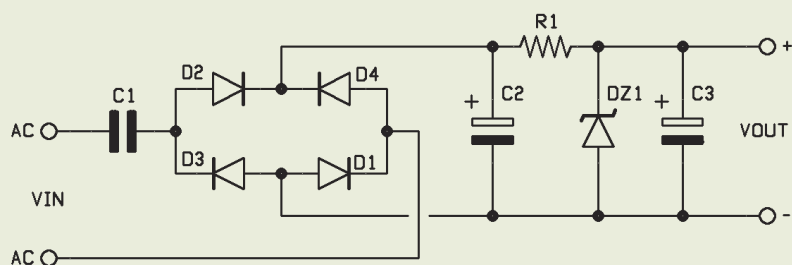
(10 dBm). I due amplificatori possono essere collegati sia direttamente in serie (cortocircuitando i piedini 5 e 7), sia tramite una rete di pre-enfasi esterna (formata da due resistenze ed un condensatore) il cui scopo è quello di migliorare il rapporto S/N e la dinamica del segnale modulato. È possibile, inoltre, utilizzare una rete costituita da due sole resistenze (nel nostro caso R1 e R2), in modo da aumentare o diminuire la sensibilità in ingresso al piedino 4 del modulo

(ingresso audio) considerando che il valore massimo accettato, necessario per pilotare correttamente il diodo varicap, è di 10 Vpp. Il microfono, cui abbiamo dedicato un paragrafo a sé stante vista la particolarità dell'oggetto, è collegato al pin 4 e viene alimentato tramite la resistenza di bias R3. L'alimentazione di 12 Vdc viene applicata ai pin "+" e "-".

Microfono Knowles

Come dicevamo poco fa, volevamo che il nostro trasmettitore avesse buone caratteristiche in termini di dinamica: il piccolo microfono della Knowles ci ha permesso di raggiungere l'obiettivo. Tale microfono misura appena 2,54 mm di diametro e capta il rumore ambientale attraverso un foro del diametro di 0,8 millimetri. Dispone di tre contatti a saldare (Fig. 1), rispettivamente +1,3 Vdc, massa e uscita audio: date le dimensioni ridotte, occorre fare molta attenzione in fase di saldatura per non rovinare i contatti. La sensibilità del microfono per la gamma audio che ci interessa (grafico di Fig. 2,) va da - 65 dB a 350 Hz a - 53 dB a 3.500 Hz, valori questi che garantiscono un'ottima dinamica.

Alimentatore



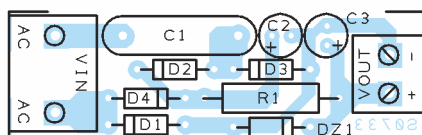
ELENCO COMPONENTI:

R1: 1 kohm 1W
C1: 470 nF 250 VL poliestere
C2: 22 µF 50 VL elettrolitico
C3: 47 µF 25 VL elettrolitico
D1÷D4: 1N4007
DZ1: Zener 12V 1W

Varie:

- Morsettiera 2 poli passo 10 mm
- Morsettiera 2 poli passo 5 mm
- Circuito stampato

Durante il collaudo, non toccate il circuito che, ricordiamo, è percorso dalla tensione di rete. C1 deve poter sopportare almeno 250 Vac.



L'alimentazione

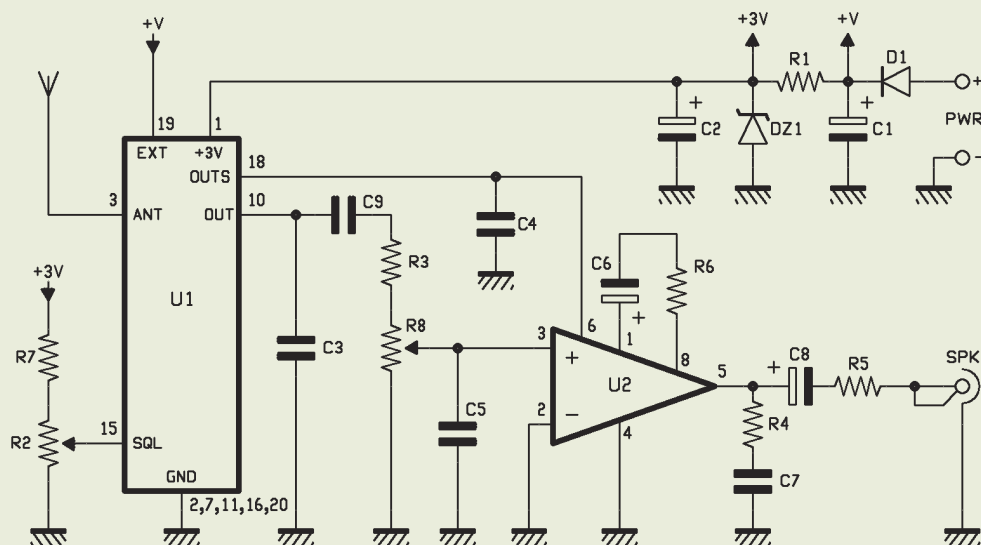
Per alimentare il trasmettitore, abbiamo realizzato un piccolo adattatore AC-DC a tensione di rete, sfruttando la proprietà della reattanza capacitiva del condensatore come resistenza di caduta per la tensione in eccesso. Tale proprietà rappresenta la resistenza che un condensatore oppone al passaggio della corrente alternata.

La reattanza viene calcolata con la formula:

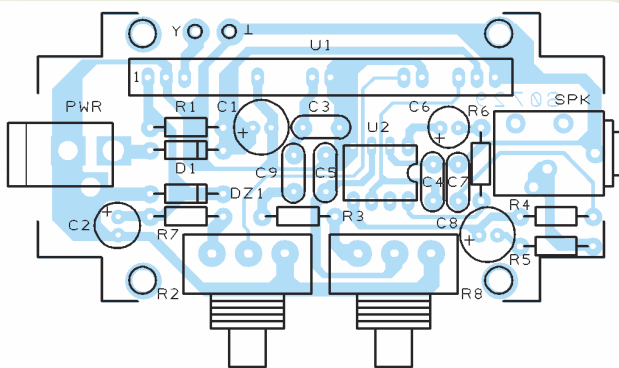
$$X_c = 1/(2\pi fC)$$

in cui X_c rappresenta la reattanza capacitiva espressa in ohm, f è la frequenza della tensione alternata in Hz e C è la capacità del condensatore, espressa in farad. Dalla formula si evince che la

>



Il ricevitore è dotato sia di soppressore di rumore (squelch, R2) che di controllo del volume (R8). L'amplificatore di BF (U2) viene alimentato dal pin 18 (controllato dallo squelch) del modulo RX-FM: se il livello del segnale RF captato è superiore alla soglia impostata con R2, il pin 18 fornisce tensione. Una volta montato, il dispositivo può essere racchiuso in un contenitore modello G404, acquistabile presso www.futurashop.it.



ELENCO COMPONENTI:

R1: 220 ohm
R2: potenziometro 10 kohm
R3: 270 ohm
R4: 10 ohm
R5: 4,7 ohm
R6: 100 ohm
R7: 10 kohm

R8: potenziometro 4,7 kohm
C1: 470 µF 16 VL elettrolitico
C2, C8: 100 µF 25 VL elettrolitico
C3: 2200 pF ceramico
C4, C7: 100 nF multistrato
C5: 1 nF multistrato
C6: 10 µF 100 VL elettrolitico
C9: 220 nF multistrato
D1: 1N4007

DZ1: Zener 3,6V 400 mW
U1: RX-FM
U2: LM386

Varie:
- Zoccolo 4+4
- Plug alimentazione
- Presa stereo 3,5 mm
- circuito stampato

reattanza è inversamente proporzionale sia alla frequenza che alla capacità: in altre parole, diminuisce con l'aumentare di "f" e "C". Nel nostro circuito, l'elemento reattivo è costituito dal condensatore C1, un poliestere da 470 nF 250 VL, che presenta una reattanza di 6.775 ohm, risultato, questo, ottenuto sostituendo i valori noti ai vari simboli della formula appena vista:
 $X_c = 1 / (6,28 * 50 * 470 * 10^{-9}) = 6775$.

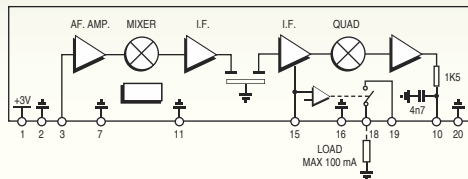
Dopo C1 troviamo il ponte a diodi (D1÷D4), il condensatore di filtro C2 (elettrolitico, 22 µF) e lo stabilizzatore di tensione, costituito da R1, DZ1 e C3. In condizioni di normale funzionamento, attraverso il condensatore C1 scorrono circa 27 mA, che determinano una caduta di tensione di circa 180÷190 V ai suoi capi. Sul condensatore C2, quindi sono presenti circa 40 V, che vengono poi abbassati a 12 V tramite R1

e DZ1, e stabilizzati ulteriormente grazie a C3.

Realizzazione pratica del trasmettitore

Visto l'esiguo numero di componenti impiegati sia nel trasmettitore che nell'alimentatore, siamo certi che non avrete problemi di alcun tipo, a patto che facciate attenzione durante il montaggio delle tre resistenze SMD ed il collegamento dei

Modulo RX-FM-Audio



*Eventuale tensione (3 – 25 V) da utilizzare per pilotare attraverso il piedino 18 eventuali carichi in base al livello impostato dallo squelch.

PIN	Funzione
1	+ 3 V
2	Massa
3	Ingresso Antenna
7	Massa
10	Uscita audio demodulato
11	Massa
15	Livello Squelch
16	Massa
18	Uscita Squelch
19	Comando*
20	Massa

fili al microfono: a tal proposito, munitevi di un saldatore con punta fine (30 Watt massimi), di fluxante in gel e di una pinzetta a becchi. Una volta realizzati i circuiti stampati per fotoincisione (i master sono come sempre scaricabili dal nostro sito www.elettronica.in.it), saldate le tre resistenze SMD sulla basetta di appoggio al trasmettitore.

Collegate quindi i tre fili necessari al microfono: potete usare del filo sottile, di quello utilizzato, per esempio, per avvolgere i trasformatori, oppure del cavetto tripolare schermato, magari recuperato da un auricolare per cellulari guasto. Saldare ora i due fili di alimentazione (rosso e nero), quindi inserite e fissate il modulo TX-FM, mantenendolo a 90° rispetto alla basetta.

Non rimane quindi che collegare l'antenna, costituita da uno spezzone di filo di circa 35 cm, pari a 1/2 onda, sulla piazzola posta in prossimità del pin 15. Se lo riterrete necessario, poi, potrete proteggere il trasmettitore

racchiudendolo in un pezzo di termorestringente da 2 cm di diametro, oppure in un piccolo contenitore, sul quale dovrete praticare una serie di forellini, in modo che il microfono possa captare il rumore ambiente. Per quanto riguarda l'alimentatore, montate prima i componenti a basso profilo, lasciando le morsettiere ed il condensatore C1 per ultimi. Prestate attenzione al verso dei componenti polarizzati e ricordatevi, durante il collaudo, che su di esso è presente la tensione di rete. Grande attenzione, dunque, una distrazione potrebbe essere molto pericolosa!

Il ricevitore

Per ricevere il segnale radio emesso dal trasmettitore, potete utilizzare uno scanner, ad esempio il modello IC-R5 della ICOM, sintonizzato sulla frequenza di 433,750 MHz, oppure realizzare il nostro ricevitore dedicato, su cui sono montati il modulo Aurel RX-FM-AUDIO, il circuito per l'inibizione dell'audio

in assenza di segnale RF (squelch), l'amplificatore di BF e, infine, il regolatore di tensione.

Fate ora riferimento allo schema elettrico: il circuito di squelch vero e proprio è inserito nel ricevitore Aurel e, per essere gestito, necessita di alcuni componenti esterni, ossia la resistenza R7 in serie al potenziometro R2, il cui cursore pilota il pin 15 del modulo RX-FM. L'uscita audio, posta sul pin 10, è collegata al finale di BF, un LM386 che, con l'ausilio di pochi componenti passivi esterni, può pilotare direttamente una cuffia oppure un piccolo altoparlante da 8 ohm. C3, C9 ed R3 realizzano una rete di equalizzazione, che rende più intelligibile la banda di frequenze in cui sono localizzate le voci. Il potenziometro R8, infine, permette di regolare il volume.

Di questo schema notiamo subito un particolare: il finale di BF viene alimentato tramite il pin 18 del modulo Aurel. Tale pin, che eroga un massimo di 100 mA, porta all'amplificatore la tensione di alimentazione +V, applicata sul pin 19, ma ciò solamente quando il segnale RF ricevuto è di livello superiore alla soglia impostata con lo squelch (R2). In questo modo, il finale viene acceso solo in corrispondenza di un segnale RF adeguato ed impostabile.

Il circuito può essere alimentato con una tensione di 9÷12 V applicata sul connettore PWR: il diodo D1 protegge il circuito da eventuali inversioni ➤

per il

MATERIALE

I componenti utilizzati in questo progetto sono facilmente reperibili in commercio. I master utilizzati per realizzare i circuiti stampati dei nostri prototipi possono essere scaricati gratuitamente dal sito www.elettronica.in.it. Il modulo trasmettente (cod. TX-FM-AUDIO) costa 16,00 Euro, quello ricevente (cod. RX-FM-AUDIO) 26,50 Euro mentre la capsula microfonica professionale Knowles FG6163 costa 34,00 Euro. Tutti i prezzi si intendono IVA compresa.

Il materiale va richiesto a: Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)

Tel: 0331-799775 ~ Fax: 0331-792287 ~ <http://www.futurashop.it>



radio è sempre presente e piuttosto forte.

Realizzazione pratica del ricevitore

Anche in questo caso, la realizzazione

del dispositivo è molto semplice ed alla portata di tutti.

Una volta in possesso del circuito stampato, che può essere realizzato per fotoincisione dopo aver scaricato il master in formato 1:1 dal sito www.elettronica.in.it, potete iniziare a montare i componenti a basso profilo, facendo attenzione al verso di quelli polarizzati. Montate quindi lo zoccolo (8 pin DIL) del finale audio, quindi inserite i componenti più ingombranti ossia i due connettori jack per l'alimentazione e per la cuffia, i potenziometri ed il modulo RX-FM. Per l'antenna, fate esattamente come spiegato nella realizzazione del trasmettitore.

di polarità, mentre il gruppo

R1-DZ1-C2 ricava i 3

Vdc necessari per alimentare il modulo RX-FM. Il ricevitore (di cui potete vedere una foto, lo schema a blocchi e la pin-out nell'apposito riquadro) è a singola conversione (media frequenza a 10,7 MHz), ha una sensibilità in ingresso di -100 dBm ed un range per lo squelch che va da -50 dBm a -100 dBm, questi ultimi corrispondenti alla massima sensibilità in ingresso. Per utilizzare il ricevitore con il trasmettitore ambientale alimentato a rete, lo squelch non è necessario perché il segnale

Note per l'utilizzo

I due dispositivi, una volta accesi, funzionano immediatamente: in particolare, il trasmettitore inizia a modulare la portante RF con il segnale audio captato dal microfono, mentre il ricevitore, sempre attivo, inizia a demodulare tale segnale. L'unica regolazione da effettuare è l'impostazione del volume. Lo squelch, in questo caso, non serve, in quanto il trasmettitore emette sempre il segnale RF, a meno che non sia guasto o non manchi la tensione di alimentazione; potete quindi lasciare il potenziometro R7 al minimo. Noi abbiamo previsto tale regolazione, così da poter utilizzare il ricevitore con altri tipi di trasmettitori, operanti sempre in FM a 433,75 MHz, ma dotati di sistemi di controllo della trasmissione, come ad esempio il VOX, che attiva la trasmissione solo in presenza di suoni, oppure il controllo della corrente di linea, utilizzato nei trasmettitori telefonici.

Tutti i prezzi si intendono IVA inclusa.

Sistemi di sviluppo per SMART CARD

SET DI SVILUPPO PER APPLICAZIONI SMART-CARD

SDK-ACR38
€119,00

Completato sistema di sviluppo contenente tutti i componenti necessari per realizzare applicazioni Smart-Card. È rivolto sia a programmatori esperti e sia a chi non possiede una specifica esperienza, consentendo di velocizzare qualsiasi applicazione. Il set può essere utilizzato anche a scopo didattico per approfondire le conoscenze sulle carte a microprocessore e sulle carte a memoria. SDK-ACR38 include esempi completi di codice sorgente nei principali linguaggi di programmazione: C#, Delphi, VB.NET, VB6, Visual C++, nonché la documentazione tecnica di riferimento per Smart-Card per il dispositivo ACR38 e tutte le informazioni per utilizzare al meglio il supporto PC/SC oppure, in alternativa, le librerie proprietarie. Caratteristica importante dell' SDK-ACR38 è di poter gestire le carte di tipo a memoria, non solo attraverso librerie proprietarie, ma anche all'interno di PC/SC (che come noto è stato ideato per supportare le carte a microprocessore). Uno specifico documento incluso nell' SDK-ACR38 descrive in dettaglio tale possibilità.

Contenuto della confezione:

- 1 ACR38 Smart Card Reader
- Interfaccia USB full speed;
- Legge e scrive le carte a microprocessore ISO 7816 1, 2, 3, 4 (protocolli T=0 e T=1);
- Supporta carte a 1.8 V, 3 V, 3.3 V MCU, con sistema interno di protezione da corto circuito e le più comuni carte memoria tra cui le IC2 bus e le secure memory;
- Supporto per sistemi operativi: Win 98, Me, 2000, XP, XP-SP1, XP-SP2;
- Alimentazione: 5 Vdc mediante interfaccia USB;
- 20 Smart-Card per test
- 1 ACR38 SIM-size plug-in Smart Card Reader
- Dispositivo di interfaccia di dimensioni molto contenute per Smart-Card in formato SIM plug-in con caratteristiche simili a quelle dell'ACR38;
- 1 ABR00LS Balance Reader
- 1 un visualizzatore con display ad otto digit per la lettura dei dati da Smart-Card;
- 1 CD-ROM con software e driver di supporto

LETTORE SMART-CARD

MINILECTOR
€34,00

Letto per Smart-Card con interfaccia USB progettato per rispondere nel miglior modo ai crescenti requisiti delle applicazioni più evolute con smartcard (firma digitale a validità legale, carta d'identità elettronica, carta nazionale dei servizi, carte bancarie). Consente una semplice e piena integrazione con i sistemi a chiave pubblica per l'autenticazione, la firma digitale e la cifratura di documenti. In questo ambito è di gran lunga il più diffuso ed apprezzato prodotto in Italia ed è qualificato dalle più importanti Autorità di Certificazione. È il prodotto ideale anche per applicazio-

SMART-CARD ACOS2 DA 8 K

ACOS2-8K
€3,90

5 codici segreti + codice di sblocco;
Codice PIN modificabile;
Coppia di codici per autenticazione reciproca;
Session Key basate su numeri random.

Smart-Card a memoria EEPROM con capacità di 8 K. È particolarmente indicata per applicazioni dove è richiesto un alto grado di sicurezza sulla transazione dei dati: controllo accessi, autenticazioni, royalty, network security, ecc. Conforme alle norme ISO 7816-1/2/3, protocollo T=0, crittografia MAC e DES.

viene fornita completamente bianca senza alcuna personalizzazione grafica.

SMART CARD SECURE ACCESS MODULE BK ACOS3SAM
€22,00

FUTURA ELETTRONICA

Via Adige, 11 - 21013 GALLARATE (VA)
Tel. 0331/799775 - Fax 0331/778112
www.futurashop.it

Disponibili presso i migliori negozi di elettronica o nel nostro punto vendita di Gallarate (VA).
Scegliete la tecnica e vendete on line www.futurashop.it

Tabellone segnapunti radiocomandato

di *Gabriele Daghetta*

Specificamente progettato per il gioco del tennis, dato che tiene conto di game, set e Tie-Break, può essere utilizzato anche per indicare il punteggio di altri sport. I led giganti ad alta efficienza consentono di vedere il punteggio a grande distanza. Alimentato con batterie sigillate al piombo-gel che possono essere ricaricate anche con un pannello solare. Prima puntata.



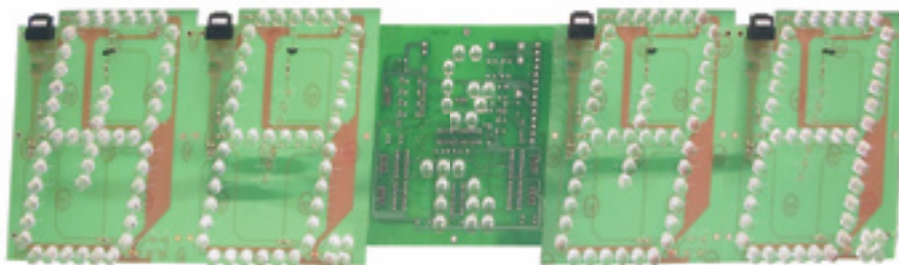
Da parecchi anni, per contrastare gli effetti della sedentarietà dovuta al lavoro che si svolge sempre meno in movimento e sempre più (causa l'inarrestabile avanzamento del terziario a discapito dell'industria, dell'artigianato e dell'agricoltura) dietro ad una scrivania, sempre più persone si danno ad uno sport, che praticano nel tempo libero grazie anche alla diffusione di impianti privati e pubblici; calcio a parte, tra gli sport più praticati ci sono tennis e pallavolo, che presentano il vantaggio di avere aree attrezzate al coperto e al caldo. Nelle aziende o tra gruppi di amici nascono così vere e proprie gare,

che si organizzano, magari, la sera usciti dal lavoro, tanto che nel sedile posteriore della macchina del collega con cui si va a pranzo non è improbabile trovare la borsa da cui spunta il manico della racchetta: c'è chi porta quella in fibra di carbonio, chi quella tradizionale in legno e chi quella in alluminio; infine, i perfezionisti caricano in macchina anche il tabellone segnapunti. Già, perché per quanto attrezzati, i campi hanno tutto quel che serve a giocare ma quasi mai dispongono di un segnapunti, indispensabile per tenere e mostrare il punteggio dei giocatori o delle squadre, come nei campionati ufficiali. Purtroppo, >

a meno di non voler destinare un giocatore a rotazione che segni su una lavagna il punteggio di quelli che stanno disputando una competizione, bisogna procurarsi un tabellone elettronico, oggetto che, nei negozi specializzati, costa un bel po' di soldi. Chiacchierando con amici che giocano a tennis, abbiamo percepito l'esigenza di molti che, come loro, praticano lo sport per passatempo e non per profitto, e che quindi vorrebbero evitare di limitare le già cospicue spese per l'attrezzatura di gioco; ci è quindi venuta l'idea di progettare e proporre un tabellone segnapunti professionale, studiato per funzionare in due diverse modalità: ad avanzamento illimitato oppure specifico per il tennis. In quest'ultima modalità l'apparato avanza e visualizza i punteggi secondo quanto previsto dal regolamento ufficiale della Federazione Italiana Tennis, nel senso che tiene conto dei giochi vinti e dei set, e che prevede il Tie-Break.

Il nostro prodotto ha quindi caratteristiche professionali e può essere utilizzato in spazi anche relativamente grandi: ad esempio in palestre dove si gioca a pallavolo; la visualizzazione dei punti è ottenuta mediante un display a due cifre (grandi ciascuna 10x17 centimetri) per lato, realizzato con led giganti e integrato dalle indicazioni utili per il tennis. La dimensione dei display e l'impiego di led ad alta efficienza, consente di vedere le cifre ad una distanza di qualche decina di metri. L'avanzamento del punteggio si comanda mediante un radiocomando a quattro canali, tramite cui l'addetto (che può anche essere un giocatore) può assegnare i punteggi ai contendenti e accedere a funzioni speciali che descriveremo nelle prossime pagine. In considerazione dell'uso che se ne deve fare e dei luoghi dove viene abitualmente impiegato, il nostro tabellone elettronico utilizza per l'alimentazione una batteria a 12 volt, che può essere ricaricata mediante un adattatore da

Fig. 2 - Le cinque schede collegate tra loro: date le dimensioni ragguardevoli, è consigliabile prevedere un adeguato supporto meccanico.



rete o, clima permettendo (visto che si cerca di giocare all'aperto), da un pannello solare opzionale. È comunque sempre possibile far funzionare il dispositivo con un adattatore di rete qualora sia disponibile una presa di corrente.

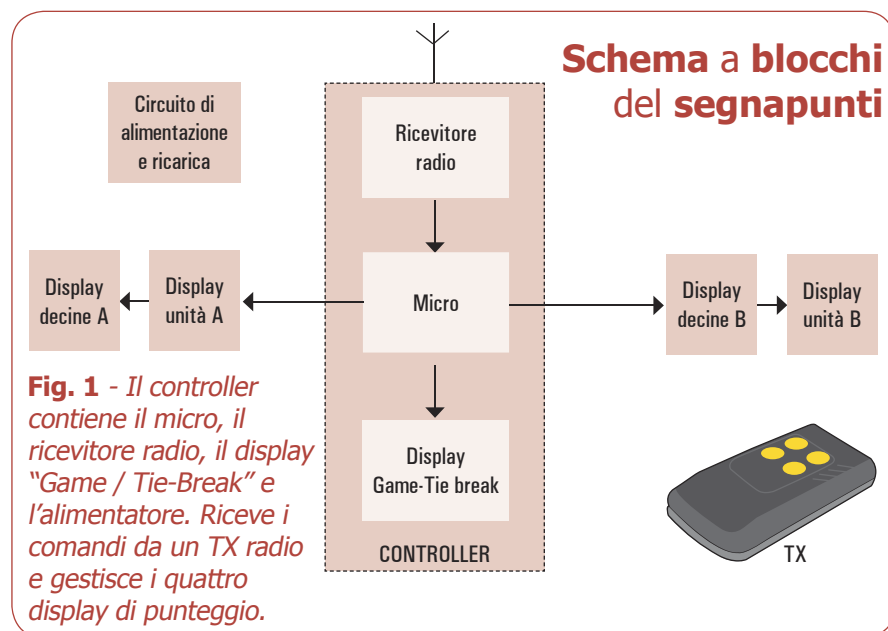
Il circuito

Il tabellone segnapunti, il cui schema a blocchi è visibile in Fig. 1, è composto da un'unità centrale a microcontrollore e da quattro schede display da una cifra più punto decimale ciascuna; completano l'insieme il trasmettitore a quattro canali rolling-code TX-4M-HCS (prodotto dall'Aurel) e lo stadio di alimentazione. L'unità centrale, chiamata anche controller, è una scheda che contiene la logica di controllo delle schede a display, oltre al ricevitore del telecomando e a due visualizzatori a led giganti che segnalano,

nella modalità tennis, il Tie-Break e il Game (la conclusione di un gioco); abbiamo previsto queste indicazioni sulla scheda base (invece che su ulteriori circuiti stampati), per rendere il più possibile compatto il tabellone. Oltre a quanto elencato, l'unità centrale ospita l'alimentatore (che nella Fig. 1 è un blocco a parte) e le connessioni per collegare la batteria, l'eventuale pannello solare o un caricabatteria, oppure un alimentatore da rete.

L'unità centrale (controller)

Il circuito che gestisce il visualizzatore è basato su un microcontrollore Microchip PIC16F876A, il quale dispone di risorse sufficienti per svolgere i compiti assegnatigli; il micro scrive i punteggi in una zona di RAM e li aggiorna in base ai comandi ricevuti dal telecomando. Mediante quattro driver da BCD a 7



Il tennis è un gioco che si svolge in un campo rettangolare diviso a metà da una rete, tra 2 o 4 giocatori che si tirano la palla a vicenda con una racchetta, da una parte all'altra della rete. Un incontro di tennis si divide in partite (set): in base al tipo di torneo, un incontro si gioca al meglio di 3 o 5 set; a sua volta un set è costituito da 6 o più giochi (game), nonché se i giocatori sono sul punteggio di 6-6, quando il regolamento del torneo lo prevede, si disputa un game decisivo, in inglese tie-break, per conseguire la vittoria sul 7-6, ad eccezione della quinta partita in alcuni importanti tornei.

Punteggio in un game (gioco):

1. Game (gioco) normale

a) Il punteggio in un game normale viene chiamato come di seguito indicato, indicando per primo il punteggio del battitore:

- nessun punto "zero"
- primo punto "15"
- secondo punto "30"
- terzo punto "40"
- quarto punto "gioco"

eccetto quando, se entrambi i giocatori/coppie vinto tre punti, il punteggio è "parità". Dopo "parità" il punteggio è "vantaggio" per il giocatore/coppia che vince il punto successivo. Se lo stesso giocatore/coppia vince anche il punto successivo, vince il game; se invece il punto successivo è vinto dal giocatore/coppia avversario, il punteggio è di nuovo "parità". Il giocatore/coppia deve vincere due punti consecutivi subito dopo "parità" per vincere il game.

2. Game (gioco) Tie-Break

a) Nel game Tie-Break i punti vengono chiamati "zero", "1", "2", "3", ecc. Il giocatore/coppia che vince per primo sette punti vince il game e il set, purché vi sia un margine di due punti sull'avversario/i. Se necessario, il Tie-Break

continua fino al raggiungimento di questo margine.

b) Il giocatore cui tocca il turno di battuta batte il primo punto del Tie-Break. I due punti successivi sono battuti dall'avversario/i (in doppio, dal giocatore della coppia avversaria cui spetti di battere dopo). Dopo ciò, ogni giocatore/coppia batte alternativamente per due punti consecutivi fino alla fine del Tie-Break (in doppio, la rotazione della battuta fra i due compagni continua nello stesso ordine tenuto in quella partita).

c) Il giocatore/coppia che ha battuto per primo nel Tie-Break è ribattitore nel primo game del set successivo.

Punteggio in un set (partita):

Vi sono diversi metodi di punteggio di un set; i due metodi principali sono "partita ai vantaggi" e "partita con tie-break". Possono essere usati entrambi, ma quello adottato deve essere comunicato prima della manifestazione. Se si usa il metodo di "partita con Tie-Break", si deve comunicare se anche la partita decisiva si gioca con questo metodo.

a) "Partita con vantaggio"

Il giocatore/coppia che per primo vince sei game, vince il set, purché abbia un vantaggio di due game sull'avversario/i. La partita continua, se necessario, sino a raggiungere questo vantaggio.

b) "Partita con Tie-Break" (Supportata dal segnapunti)

Il giocatore/coppia che per primo vince sei game, vince la partita, purché abbia un vantaggio di due giochi sull'avversario/i. Se il punteggio arriva a sei game pari, si disputa il Tie-Break.

Punteggio in un incontro:

L'incontro può essere giocato al meglio dei tre set (partite): il giocatore/coppia che vince due set vince l'incontro, o al meglio delle cinque partite: il giocatore/coppia che vince tre set vince l'incontro.

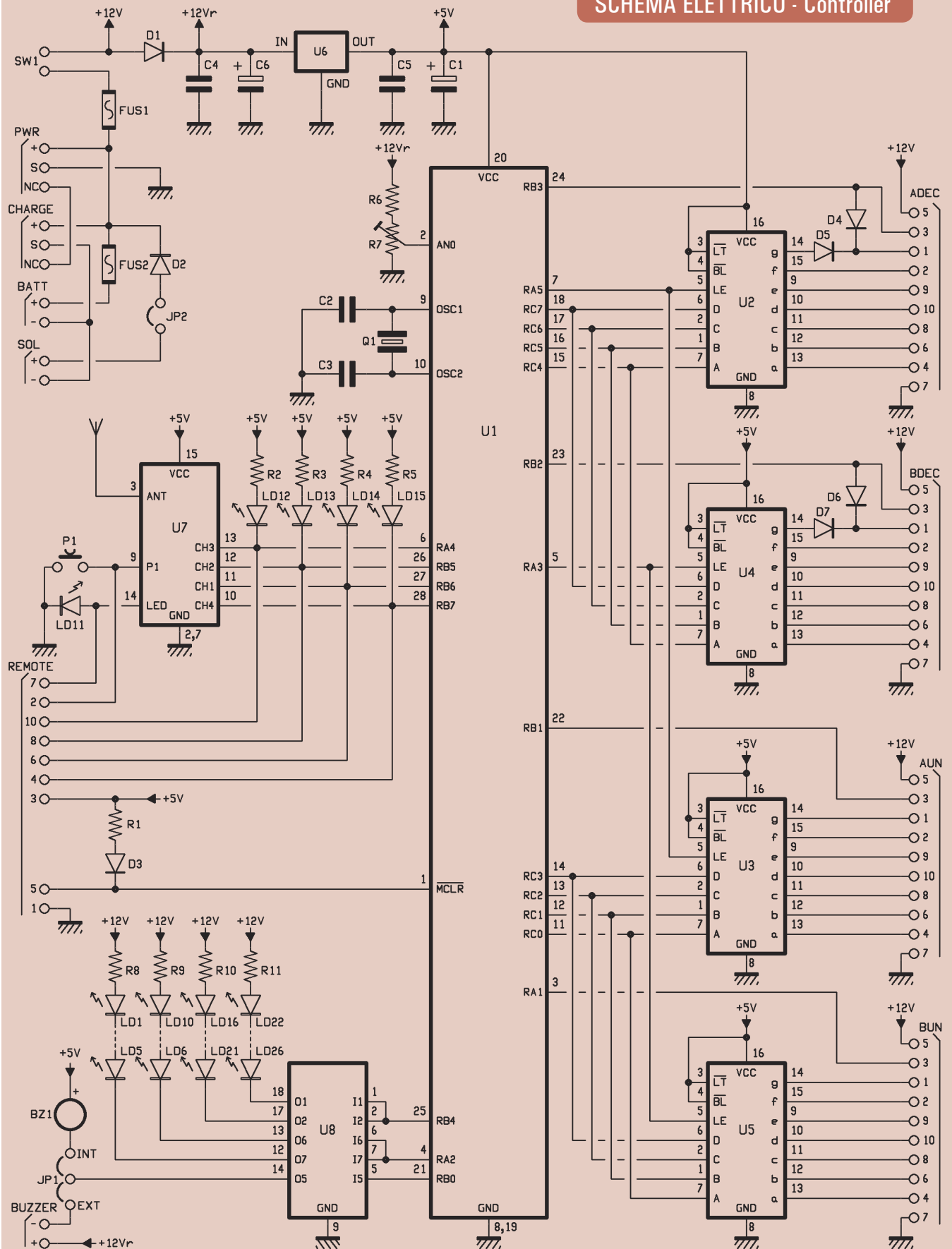
segmenti di tipo 4511, pilota altrettante schede display, al fine di visualizzare i punteggi attuali mantenuti in memoria; tramite il line-driver U8 si occupa di comandare l'illuminazione dei led usati per comporre la segnalazione del Game (LD1÷LD10) che è una G, oltre a quella del Tie-Break (LD16÷LD26) che corrisponde alle lettere Tb. Lo stesso ULN2803 comanda il segnalatore acustico impiegato per richiamare l'attenzione sui cambiamenti di punteggio, l'aggiudicazione di un gioco o di un set; in merito, va detto che è stata prevista la possibilità di usare un cicalino interno o un ronzatore esterno; ciò per dare la possibilità di scegliere il tipo

di avviso acustico più adatto alla situazione di impiego. Il cicalino (si inserisce chiudendo il jumper JP1 su INT) può servire quando si sta in spazi ristretti, mentre il buzzer (chiudere JP1 su EXT), che produce un suono più potente, è più indicato in ambienti ampi o all'aperto.

Oltre a gestire i display e il segnalatore acustico, il PIC16F876A legge lo stato di uscita del ricevitore del radiocomando, che fa capo all'U7: quest'ultimo è un modulo ibrido Aurel siglato RX4M-HCS che incorpora un decoder rolling-code a quattro canali per l'encoder HCS301. Abbiamo preferito tale sistema (a cui abbiamo già dedicato molti articoli,

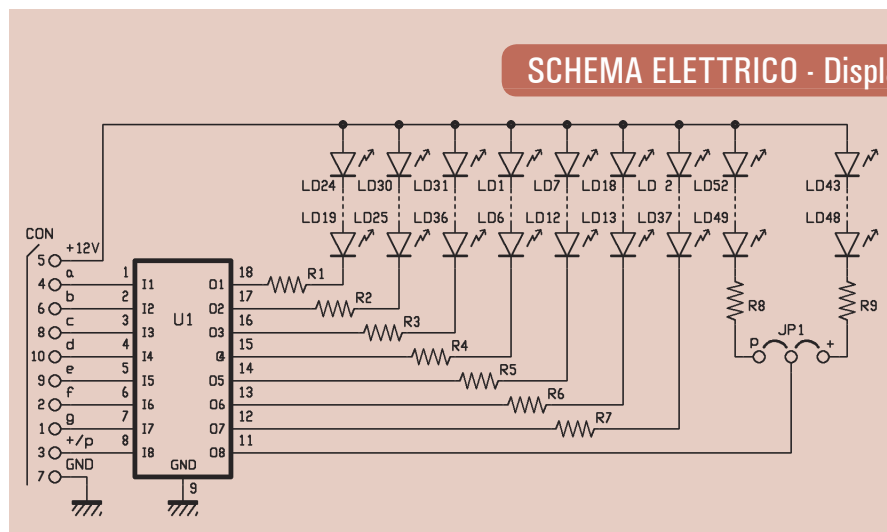
l'ultimo dei quali relativo ad un apricancello a quattro canali pubblicato sulla rivista N. 117) ad uno a codice fisso (per esempio MM53200/UM3750 o MC1450xx) per ottenere l'esclusività del comando (cioè evitare interferenze da parte di altri trasmettitori simili operanti nelle vicinanze: ad esempio quelli di apricancelli). Per l'uso che ne facciamo, il canale 1 comanda l'avanzamento dei punti del visualizzatore di sinistra, il 2 l'avanzamento del display di destra, mentre il 3 ed il 4 servono al comando delle funzioni accessorie. P1 effettua anche il reset e P2 cambia la modalità di esercizio. Il ricevitore, un RX4M-HCS (U7), è un ibrido ➤

SCHEMA ELETTRICO - Controller



che contiene un radioricevitore con stadio d'ingresso superrigenerativo (dotato di filtro ceramico per restringere la banda) accordato a 433,92 MHz e demodulatore d'ampiezza, seguito da un comparatore che squadrizza i fronti del segnale demodulato; all'uscita dello squadratore si trova un microcontrollore programmato per funzionare da decoder e gestire le quattro uscite pilotando opportunamente altrettanti transistor NPN configurati in modo open-collector. Il micro è programmato con il software necessario a decifrare i segnali inviati dal minitrasmittitore con HCS301 (TX-4M-HCS), e contiene perciò l'algoritmo KeeLoq. Trasmittitore palmare e ricevitore si acquistano sempre in coppie abbinate che abbiano lo stesso codice base di 28 bit e il medesimo Device Code di 64 bit. Le uscite dell'U7 sono localizzate ai piedini 10 (canale 4), 11 (canale 1), 12 (canale 2) e 13 (canale 3), e ognuna di esse, oltre a comunicare la propria condizione logica al rispettivo piedino del microcontrollore, quando è attivata fa illuminare un led di segnalazione; notate a riguardo che i led sono alimentati verso il positivo di alimentazione e, insieme ai loro resistori di limitazione della corrente, operano il pull-up delle uscite del modulo RX. Il pin siglato P1 è quello cui si collega il pulsante che serve ad effettuare l'apprendimento del trasmettitore mentre quello chiamato LED serve a collegare il diodo luminoso tramite il quale l'RX-4M-HCS segnala il proprio stato.

Le quattro uscite e questi ultimi due pin, insieme alla linea /MCLR del micro U1 e all'alimentazione a 5 volt, vengono portati a un connettore siglato REMOTE per permettere, a chi lo desiderasse, di comandare il visualizzatore tramite un telecomando a filo, ovvero una pulsantiera collocata sul banco dell'operatore addetto a tenere il punteggio. In questo caso il radio-comando, il pulsante P1, il led LD11 e l'ibrido U7 non sono necessari. I



pin 1-5-4-6 del connettore REMOTE possono anche essere utilizzati per effettuare la programmazione ICSP del PIC. La gestione dei quattro moduli display giganti avviene pilotando distintamente i segmenti e il punto decimale di ciascuno, mediante apposite linee facenti capo a quattro connettori (uno per modulo); ora va detto che il PIC16F876A non dispone di linee di I/O sufficienti a pilotare con ciascuna un segmento, abbiamo perciò dovuto fare ricorso al comando in binario e all'utilizzo di decoder da BCD a 7 segmenti, che sono appunto i 4511. In pratica per ogni digit il microcontrollore genera il valore binario corrispondente espresso in BCD (su 4 bit...) che poi il converter trasforma nei segnali elettrici necessari a far accendere i segmenti del relativo display al fine di visualizzarlo. Ad esempio, l'1 viene espresso come 0001 e provoca, all'uscita del 4511, l'attivazione delle linee corrispondenti ai segmenti b e c; l'8, espresso dal micro come 1000, determina l'attivazione di tutte le uscite (a, b, c, d, e, f, g). Si noti che per attivazione intendiamo che le uscite si portano a livello alto (12 V) dato che il 4511 è progettato per comandare display a catodo comune. Si noti altresì che, potendo visualizzare solo da 0 a 9, se riceve word di 4 bit costituenti valori da 10 a 15, l'integrato non visualizza alcunché, nel senso che manda a zero

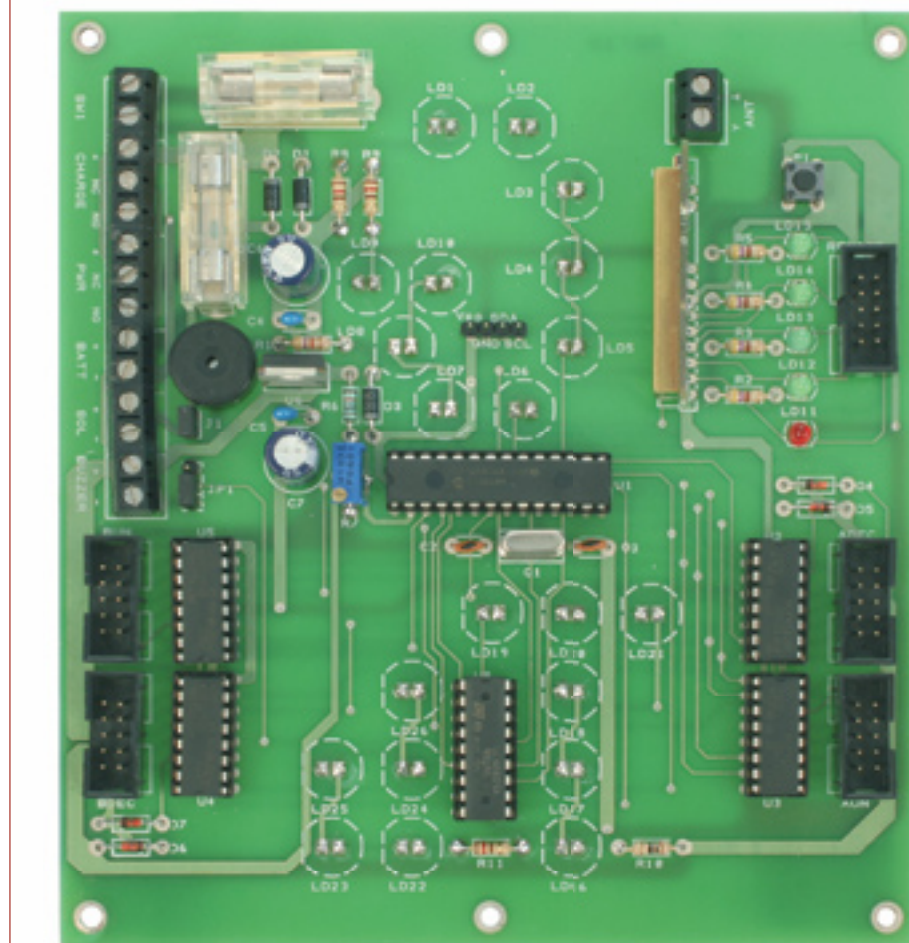
logico tutte le proprie uscite. Anche ricorrendo alla codifica BCD, resta il problema della quantità di linee di I/O: infatti, per gestire quattro cifre occorrerebbero 16 piedini, senza contare quelli occorrenti ad accendere i punti decimali; ecco quindi che ci siamo visti costretti a ricorrere al comando in multiplex. In altre parole, usiamo le linee RC0÷RC3 per pilotare le due decodifiche delle unità, e le linee RC4÷RC7 per le due decodifiche delle decine, selezionando di volta in volta la coppia su cui agire. Per riuscire nell'intento, sfruttiamo il fatto che il 4511 è stato progettato per funzionare con la tecnica del multiplex, quindi dispone di un latch nel quale memorizza i dati in ingresso anche se poi vengono a mancare o variano perché diretti ad altri dispositivi; quando bisogna far visua-



lizzare una certa cifra, i dati corrispondenti vanno caricati nel latch abilitando la linea LE (Latch Enable) e poi riportandola a riposo (zero logico). Quando LE è disattivata, il 4511 mantiene in memoria i dati e ignora qualsiasi variazione presente sul bus. La tecnica multiplex consiste, in pratica, nell'inviare alternativamente i dati diretti a più dispositivi collegati alle stesse linee, abilitando di volta in volta il dispositivo che si vuole li utilizzi effettivamente. Nella nostra applicazione, per mostrare i punti (unità o decine), il microcontrollore esegue le seguenti operazioni:

- aggiorna la porta RC con i codici BCD di decine ed unità;
- disattiva il Latch Enable delle decodifiche indirizzate (giocatore A o giocatore B), facendo così trasferire i due codici BCD sulle linee a 7 segmenti;
- riattiva il Latch enable per memorizzare il cambiamento.

Dovendo aggiornare il punteggio di entrambi i giocatori, esegue due volte la stessa operazione: prima invia sul bus i dati dell'A e fornisce un impulso sugli LE di U2 e U3, poi manda i dati del B e comanda gli LE dei decoder U4 ed U5. Tenete presente un fatto: a parte casi particolari, il microcontrollore aggiorna sempre e solo le cifre corrispondenti al giocatore che ha incrementato il punteggio, lasciando inalterata la condizione dei 4511 corrispondenti al lato dell'altro giocatore. C'è da analizzare, infine, un ultimo dettaglio del blocco circuitale usato per comandare i display: i diodi D4 e D5 sono stati inseriti per poter comandare il segmento g (il tratto centrale) di ciascun visualizzatore, sia mediante l'uscita corrispondente del CD4511, sia direttamente dal microcontrollore; quest'ultimo agisce direttamente quando occorre visualizzare il segno "+", cosa che viene ottenuta alimentando (tramite il piedino 3 del connettore) il tratto verticale centrale e, appunto, la linea del segmento g. In questo caso



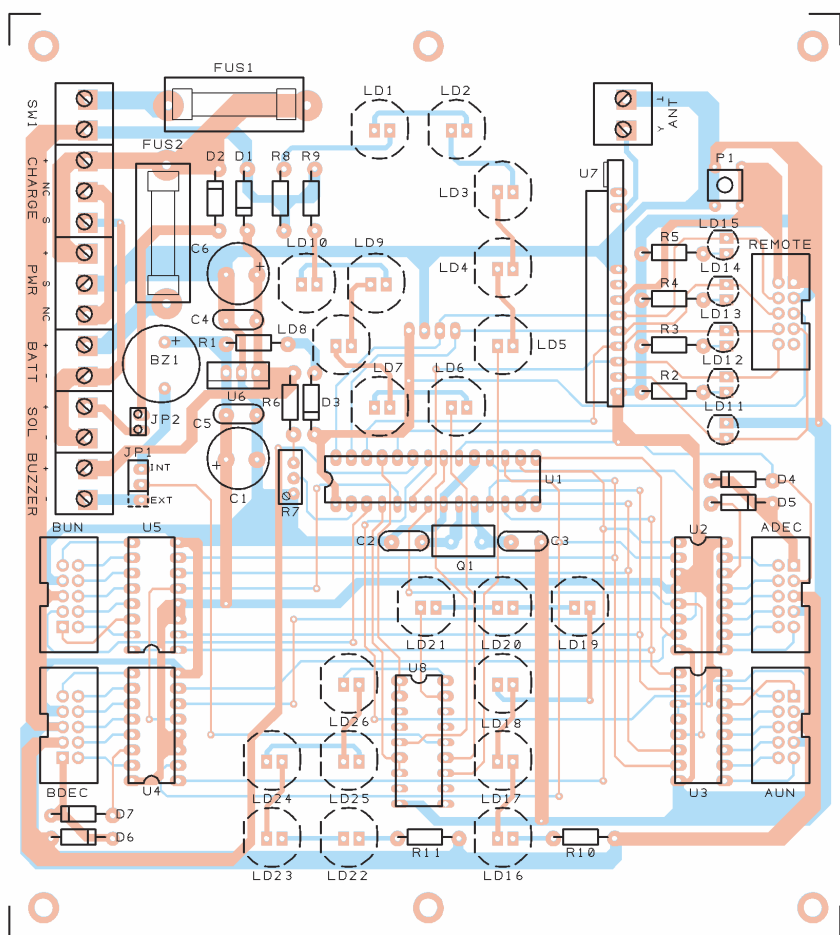
ELENCO COMPONENTI:

R1: 4,7 kohm
 R2÷R5: 470 ohm
 R6: 10 kohm 1%
 R7: trimmer multigiri 10 kohm
 R8, R9: 120 ohm
 R10: 10 ohm
 R11: 120 ohm
 C1: 470 µF 25 VL elettrolitico
 C2, C3: 10 pF ceramico
 C4, C5: 100 nF multistrato
 C6: 470 µF 35 VL elettrolitico
 D1÷D3: 1N4007
 D4÷D7: 1N4148
 Q1: Quarzo 20 MHz

U1: PIC16F876A (MF720)
 U2÷U5: 4511
 U6: 7805
 U7: RX4M-HCS
 U8: ULN2803
 BZ1: Buzzer con elettronica
 P1: Microswitch
 LD1÷LD10, LD16÷LD26:
 led 10 mm rosso alta luminosità
 LD11: led 3 mm rosso
 LD12÷LD15: led 3 mm verde
 Varie:
 - Zoccolo 8+8 (4 pz.)

lo stato logico del 4511 è irrilevante. Se non vi fossero i diodi, eventuali livelli logici opposti presenti sulla decodifica 4511 e sul micro potrebbero causare il danneggiamento di entrambi i chip. Bene, visti anche questi dettagli è il caso di approfondire il discorso sul-

l'alimentazione: l'intero tabellone segnapunti è alimentato con una batteria a 12 V (da connettere ai morsetti + e - BATT) per ricaricare la quale bisogna utilizzare un alimentatore/caricabatteria da collegare ai contatti + CHARGE e massa (-). Il circuito è stato progettato in modo da poter



Durante il montaggio prestate particolare attenzione ai led ad alta efficienza che vanno inseriti dal lato saldature ed ai cinque connettori flat che, invece, vanno montati sul lato componenti rispettando l'orientamento della chiave hardware, come visibile nel disegno.

- Zoccolo 9+9
- Zoccolo 14+14
- Strip maschio 2 vie
- Strip maschio 3 vie
- Strip maschio 4 vie
- Jumper (2 pz.)
- Antenna accordata 433 MHz
- Morsettiera 2 poli (5 pz.)
- Morsettiera 3 poli (2 pz.)
- Conn. POD10 maschio da CS (5 pz.)
- Conn. POD10 femmina a crimpare (8 pz.)
- Flat cable 10 vie - 200 mm
- Portafusibile orizzontale da CS (2 pz.)
- Prese DC da pannello (2 pz.)
- Fusibile 1A
- Fusibile 2A
- Interruttore a pulsate
- Carica Batteria (VL12LA)
- Spinda DC volante
- Alimentatore 12Vdc 1,2A (AL08-12)
- Batteria 12V/2A (NP21-12)
- circuito stampato

funzionare anche con la tensione di rete, mediante un alimentatore a 12 Vdc stabilizzati (modello AL08-12 o simile); in tal caso, però, la batteria risulta esclusa e non viene caricata. Per caricarla (ma durante la carica il visualizzatore non funziona) bisogna collegare il caricatore mediante una

presa DC: per questa ragione, nello schema elettrico appaiono tre contatti, siglati +, S ed NC: il primo (+) è lo spinotto interno o polo caldo, il secondo (S) è il contatto esterno o polo freddo; il terzo (NC), infine, è un contatto normalmente chiuso che rileva l'inserimento del plug. A plug

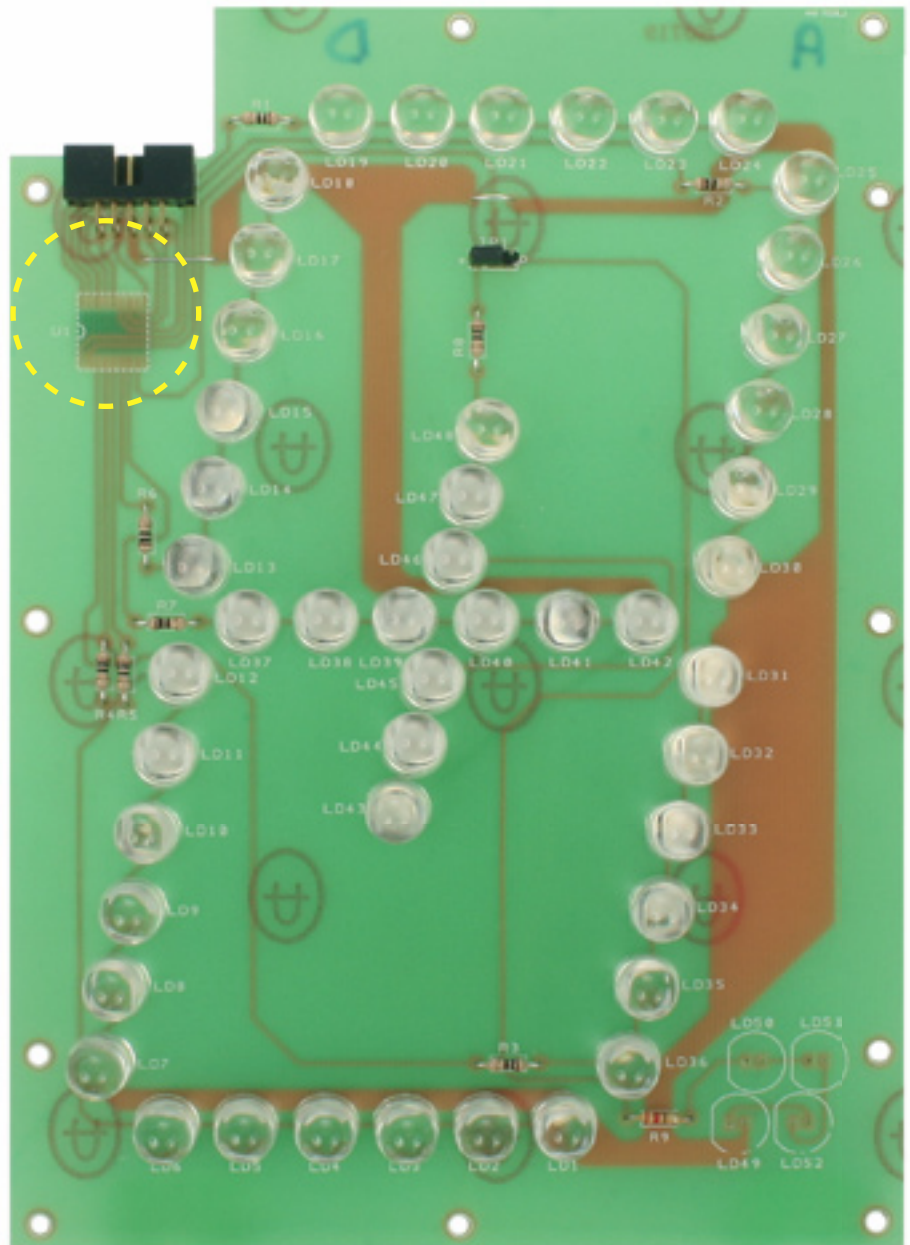
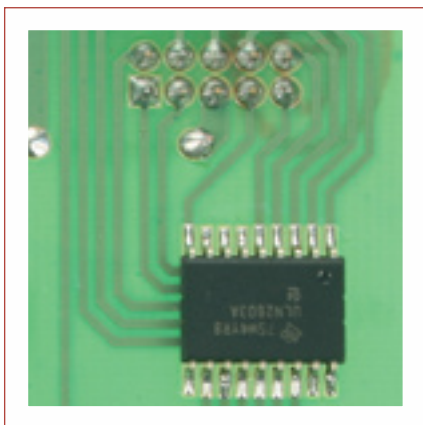
disinserito, S e NC sono in corto tra loro. La ricarica può anche avvenire mediante un pannello solare da 12 V, da collegare ai contatti + e - SOL; resta inteso che il pannello può servire solo per la ricarica e non per alimentare contemporaneamente il resto del circuito, a meno di non procurarne uno abbastanza potente. Che si alimenti a batteria o con un adattatore di rete collegato a + e - PWR, l'interruttore collegato alla morsettiera SW1 consente di dare e togliere tensione senza sconnettere alcuno dei due. A valle dell'interruttore è posto il diodo D1, che protegge il circuito dall'inversione di polarità; i condensatori C4 e C6 filtrano ulteriormente la tensione continua prima che venga applicata al regolatore integrato U6, la cui uscita fornisce i 5 volt che servono al microcontrollore e alla logica (eccetto i driver ULN2803 a 12 V). In tema di alimentazione, vogliamo far notare un dettaglio: oltre a svolgere le funzioni descritte finora, il software che "gira" nel microcontrollore provvede (tramite una specifica routine lanciata ciclicamente) a monitorare la tensione di ingresso presente a valle di D1 e a darne segnalazione mediante il punto decimale del display B quando il potenziale scende sotto un certo livello. La lettura viene effettuata usando l'A/D converter che fa capo alla linea AN0 (piedino) la quale rileva, tramite il partitore resistivo formato dalla R6 e dal trimmer R7, parte del potenziale all'ingresso del regolatore 7805.

Il modulo display

Analizzato il funzionamento dell'unità centrale, possiamo concentrarci sullo schema elettrico dei moduli visualizzatori che, sebbene siano tutti uguali, funzioneranno ciascuno in base alla posizione in cui si troveranno, ovvero a seconda di quale sarà il connettore tramite cui riceveranno i segnali. Ogni modulo riceve i livelli logici corrispondenti ai sette segmenti usati per comporre ciascun ➤

digit nonché quelli relativi all'accensione del punto decimale o del tratto verticale; in particolare, notate che quest'ultimo livello è lo stesso per entrambe le funzioni (+/p) e ciò perché siamo stati costretti a risparmiare linee di I/O. Abbiamo quindi dovuto inserire un ponticello da impostare a seconda che il circuito venga usato come visualizzatore delle unità o delle decine. Nel primo caso, JP1 va chiuso verso p (perché serve che si accenda, all'occorrenza, il punto decimale) e nel secondo il jumper deve andare su + (perché quando serve, il micro deve far illuminare il +, accendendo contemporaneamente il segmento g).

I segmenti di ciascun modulo non vengono alimentati direttamente dai CD4511 né, tantomeno, dal microcontrollore, perché le correnti richieste da essi sono maggiori di quelle che tali integrati possono sopportare; con i livelli logici corrispondenti ai segmenti pilotiamo dei line-driver che alimentano i led fornendo la corrente necessaria. Più precisamente, a, b, c, d, e, f, g e +/p pilotano ciascuno un ingresso dell'U1, che è un line-driver ULN2803 composto



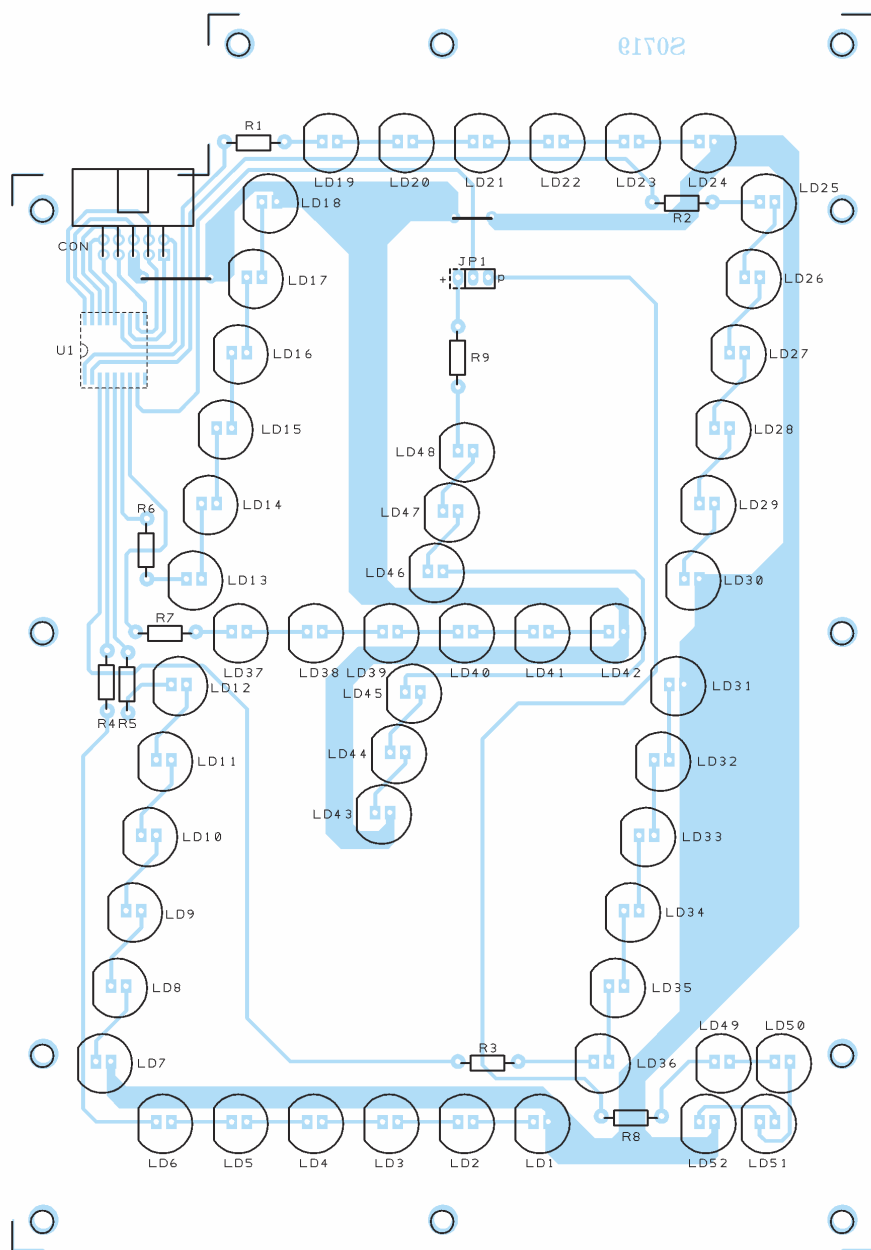
I led ad alta efficienza, le resistenze, il jumper, il connettore flat cable a 90° ed i due ponticelli vanno montati sul lato componenti. Sul lato saldature, invece, nella zona evidenziata dal cerchio giallo, va saldato il driver SMD ULN2803 (illustrazione qui a sinistra). L'immagine mostra un display "decine". Per realizzare un display "unità" vanno montati i quattro led del punto decimale (LD49÷LD52) mentre non si inseriscono i led centrali LD43÷LD48.

da otto Darlington NPN configurati in modo open-collector (e provvisti delle resistenze di polarizzazione di base) aventi gli emettitori in comune e connessi al pin 9 (GND) e con le basi ed i collettori accessibili separatamente dall'esterno. Sempre all'ingresso dell'integrato, tutti i collettori

sono provvisti di un diodo di protezione che in questo caso non viene usato, dato che serve quando con l'ULN2803 si alimentano carichi induttivi.

Ogni collettore alimenta in modo sink (ad assorbimento di corrente) un segmento del digit, mediante una re-

sistenza di limitazione della corrente; notate che, per contenere il consumo di energia dell'intero tabellone segnapunti, abbiamo preferito collegare in serie i led di ciascun segmento; così facendo, alimentiamo ben sei led con la stessa corrente necessaria per accenderne uno solo.



ELENCO COMPONENTI:

R1÷R7, R9: 10 ohm

R8: 220 ohm

U1: ULN2803ADWG4

LD1÷52: led 10 mm rosso alta luminosità

Varie:

- Strip maschio 3 vie
- Jumper
- Connettore POD10 90° maschio
- Circuito stampato

Come funziona

Finora abbiamo visto com'è fatto il nostro visualizzatore, senza però parlare approfonditamente di come funziona; adesso è venuto il momento di spiegare quali sono le sue funzionalità e in che modo le si ottiene mediante il radiocomando. Come accenna-

to, il dispositivo è essenzialmente un segnapunti, tuttavia lo abbiamo programmato in modo che possa funzionare in due maniere: ad incremento di un'unità (modo non predefinito) per ogni comando ricevuto sul canale corrispondente; secondo le regole del tennis (modalità predefinita all'ac-

censione) e quindi contando da solo i punti, i giochi e i set. Nella modalità ad incremento, i pulsanti di sinistra aggiornano di un'unità il punteggio del giocatore A (P1 incrementa e P3 decrementa). Analogamente i pulsanti di destra agiscono sul giocatore B (P2 incrementa e P4 decrementa). L'aggiornamento del punteggio determina sempre l'emissione di una nota acustica da parte del cicalino o ronzatore. Al raggiungimento del valore 99, il contatore si ferma ed è pertanto necessario effettuare un reset (P1 premuto a lungo).

Nella modalità tennis, il microcontrollore, cui abbiamo insegnato (per così dire...) le regole del tennis, conta come si farebbe proprio in un match di tennis: 15, 30, 40, poi game. Per comprendere ciò bisogna ricordare che la prima palla guadagnata dà al giocatore 15 punti, la seconda 30, la terza 40 e la quarta aggiudica il game al giocatore. Quindi, un primo comando fa apparire 15, un secondo 30, un terzo 40 e un quarto il numero di game vinti dal giocatore; ad esempio, se si è partiti da zero (sistema resettato) la sequenza è 15, 30, 40, 01. Contemporaneamente all'aggiudicazione di un game, nel visualizzatore del modulo di controllo si illumina la lettera G. Ora si procede: un'ulteriore pressione riavvia la visualizzazione appena descritta, ossia 15, 30, 40, poi il numero dei game, con l'accensione della G. Tutto questo avviene per sei volte: quando, all'aggiudicazione di un game, il display conta 6, siccome il giocatore si è aggiudicato un set, viene accesa la G a luce fissa e i game di A e B, vengono visualizzati alternativamente.

Se si verifica la situazione di parità sul 6 a 6, si passa al cosiddetto Tie-Break: il visualizzatore accende la G per indicare che un giocatore si è aggiudicato un game che lo ha portato in condizione di parità con l'avversario. Viene anche fatta lampeggiare la sigla Tb (sotto la G di game), accompagnata da un tono acustico ➤

(ammesso che non sia stato disattivato il buzzer); il segnapunti, che ora è passato in modalità avanzamento normale, visualizza alternativamente i game vinti e i punti del Tie-Break. Attende quindi il comando dato da P1 o P2 del trasmettitore per attribuire il punteggio (che ora è ad incremento unitario), quindi il giocatore del lato che raggiunge 7 (sempre con vantaggio di 2 sull'altro) si aggiudica il set: ora l'indicazione Tb viene accesa a luce fissa, accompagnata da un beep prolungato, quindi si torna all'indicazione dei game e dei set esattamente come abbiamo già visto in precedenza. Resta inteso che il conteggio dei punti Tb può proseguire ad oltranza finché non si verifica la condizione di "vantaggio di 2".

Terminata la segnalazione di Tb, il segnapunti torna alla normale modalità di visualizzazione/conteggio, a meno che uno dei contendenti non abbia vinto al meglio di 3 o al meglio di 5. Nella realtà, però, le gare sono abbastanza combattute e può accadere che si arrivi a 40 pari; in questo caso, secondo il regolamento del tennis, per aggiudicarsi il game un tennista deve acquisire un vantaggio di due punti, ovvero non basta superare 40. Il microcontrollore lo sa e nella modalità tennis, quando si va a 40 su ciascun lato opera in maniera diversa da quella descritta: se si preme il tasto del canale 1 (lato sinistro) il display corrispondente mostra "+", per indicare che il giocatore A è in vantaggio (il display B viene spento, lasciando solo il +); un secondo comando sullo stesso lato aggiudica il game, con le modalità già viste.

Ma, se nessun giocatore acquisisce il doppio vantaggio cosa accade? Si procede ad oltranza: se si dà un comando al lato A e appare + nel display di sinistra, quindi si comanda il lato B, la situazione è nuovamente pari e i due lati mostrano non il + ma 40 ciascuno. Quando si dà un nuovo comando, il lato corrispondente mostra di nuovo il vantaggio (+) e

così via, fin quando non vi sono due pressioni consecutive dello stesso pulsante, ovvero non c'è il doppio vantaggio.

Bene, detto questo va precisato che nel tennis gli incontri si vincono in base al numero di set aggiudicati, numero che non è fisso, ma può essere 3 o 5. Si dice che la partita si vince al meglio di 3 o di 5 set: ciò significa che le condizioni di vittoria possono essere 2-0 / 2-1 (al meglio di 3) oppure 3-0 / 3-1 / 3-2 (al meglio di 5). All'inizio di ogni set i display sono azzerati, perché deve ricominciare il conteggio dei punti/game/set; è comunque possibile vedere, in ogni momento i set aggiudicati a ciascun giocatore: basta premere per un breve istante P4.

L'indicazione apparirà per 3 secondi

Ricaricare dal sole

Nelle installazioni portatili, è possibile caricare la batteria del tabellone segnapunti mediante un pannello solare; allo scopo, bisogna collegare positivo e negativo del pannello rispettivamente a + e - SOL dello stampato dell'unità centrale, quindi chiudere il jumper JP2. Durante la ricarica, è indispensabile che si verifichino due condizioni: il segnapunti deve essere spento e posto in pieno sole. Se accendete il dispositivo, questi assorbe corrente sia dal pannello che dalla batteria, impedendo così la ricarica di quest'ultima.



circa, poi il visualizzatore tornerà a mostrare i punti.

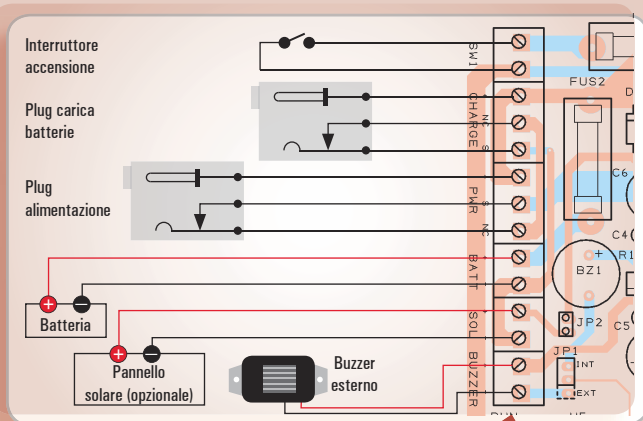
Relativamente all'assegnazione dei set, in caso di parità di game (6-6), il segnapunti procede con la disputa del Tie-Break, il cui conteggio può arrivare fino a 99, situazione improbabile (perché i giocatori esaurirebbero le forze molto prima), ma numericamente possibile dato che il contapunti assegna il set solo se si raggiunge il settimo punto con vantaggio di due, o in generale, quando (superando il 7) vede il vantaggio di 2.

Funzioni dei tasti

Dopo aver analizzato il modo in cui il segnapunti effettua i vari conteggi, vediamo dettagliatamente come utilizzare il telecomando, facendo riferimento alla Fig. 3. Posto che l'avanzamento del segnapunti si comandi usando i due pulsanti in alto P1 e P2 (ovvero ponendo a livello logico basso le linee corrispondenti alle uscite CH1, CH2 del modulo U7), vediamo come gli stessi possano essere usati anche per ottenere altre funzioni, insieme ai pulsanti P3 e P4 (CH3 e CH4 dell'U7) cominciando proprio da questi ultimi. P3, in modalità tennis, permette di annullare l'ultimo avanzamento effettuato, ossia di togliere l'ultimo punto assegnato; serve quando per errore l'addetto al tabellone luminoso assegna

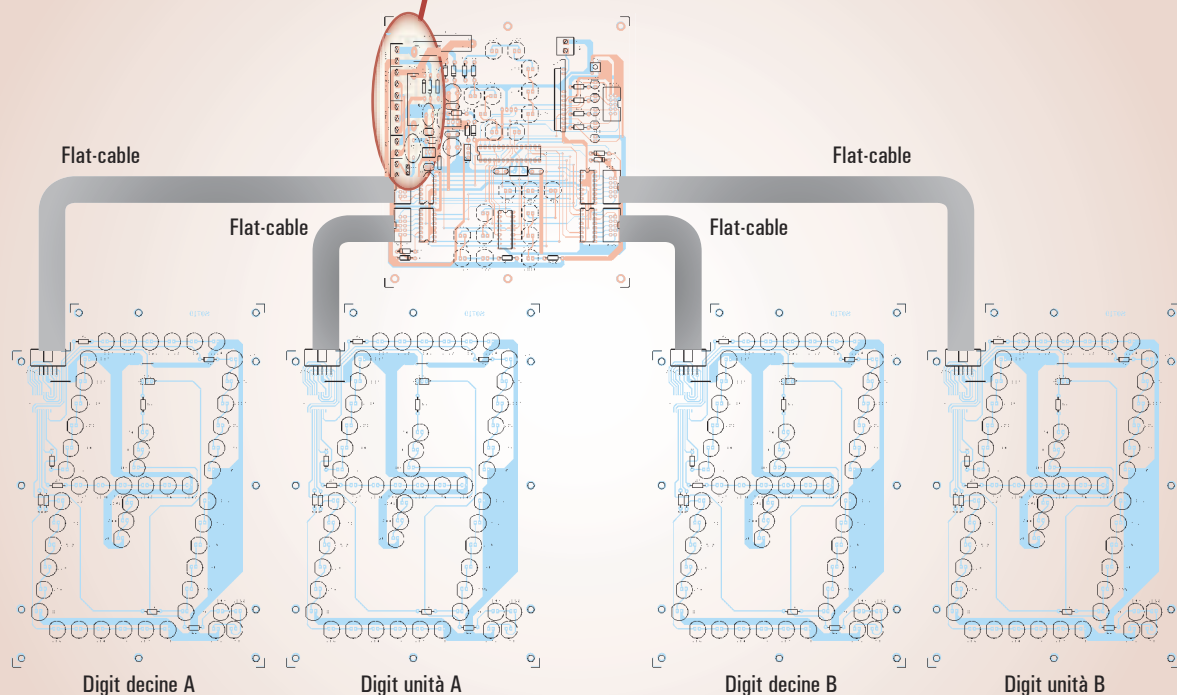
una "palla" al giocatore sbagliato. P4 serve, invece, ad attivare e disattivare il suono che il segnapunti genera quando esegue un aggiornamento del conteggio: ogni volta che viene premuto (a lungo: per circa 3 secondi) inverte l'impostazione attuale, quindi se il suono è attivo lo disattiva e viceversa. L'impostazione ordinata si intende confermata solo quando, a seguito del comando dato, il sistema risponde con una nota acustica.

E passiamo ai pulsanti normalmente usati per i comandi di avanzamen-



I quattro digit (due "decine" e due "unità") sono collegati al controller tramite dei flat cable dritti (il pin 1 di un lato corrisponde al pin 1 dell'altro lato) che trasportano sia l'alimentazione che i segnali di comando. Alla morsettiera fanno capo l'interruttore di alimentazione, un plug per l'alimentazione e uno per il caricabatterie, gli elettrodi della batteria, i fili del buzzer esterno e quelli del pannello solare (buzzer e pannello sono entrambi opzionali).

- Per alimentare il dispositivo a tensione di rete, collegare l'adattatore al plug di alimentazione (PWR).
- Per alimentare a batteria, lasciare scollegati entrambi i plug.
- Per ricaricare la batteria, collegare l'adattatore al plug "CHARGE": la batteria viene scollegata dal resto del circuito, che può comunque essere alimentato tramite i morsetti PWR.
- Per ricaricare tramite celle solari, collegare un pannello adeguato (14V, almeno 200mA di picco) ai morsetti SOL e porre al sole il dispositivo (che va tenuto spento).



to: P1, premuto a lungo (almeno 3 secondi) resetta il contatore e riporta a zero i display e tutti i punteggi. Questa funzione, attiva in entrambe le modalità (tennis o incremento), si usa al termine di ogni partita, quando se ne vuole iniziare una nuova. Quanto a P2, premuto a lungo (3 secondi) cambia la modalità del segnapunti: se si è in modo generico, si passa al tennis e viceversa. Quando è impostata la modalità ad incremento, entrambi i punti decimali dei display sono accesi.

Funzionalità nascoste

Alle funzioni tipiche del segnapunti abbiamo aggiunto qualcosa di più, ossia funzionalità che nel normale esercizio non appaiono e alle quali si può accedere solo mediante speciali comandi (riassunti ancora in Fig. 3). Eccole elencate: la prima riguarda la visualizzazione della versione firmware, utile per sapere, quando sono disponibili aggiornamenti (nel nostro sito Internet www.elettronica.in.it) se è il caso di aggiornare il sistema; vi si accede alimentando (con SW1)

il dispositivo e mantenendo premuto, contemporaneamente all'accensione, il pulsante P1. La visualizzazione della versione FW dura pochi secondi, poi il segnapunti passa nel suo normale modo di funzionamento, che corrisponde alla modalità tennis (predefinita).

La seconda, forse più interessante, prevede il controllo e la visualizzazione della tensione della batteria; per verificare quanto legge l'ADC del PIC16F876A, basta spegnere il tabellone segnapunti e accenderlo te-



nendo premuto P2 sul telecomando. In tal caso il display mostra la tensione letta, mettendo a sinistra unità e decine di volt e a destra i decimi di volt. Il controllo della tensione manifesta i suoi effetti anche automaticamente, dando un allarme durante il normale utilizzo del segnapunti: se durante una sessione di gioco il microcontrollore rileva che la batteria si sta scaricando, lo segnala illuminando il punto decimale più a destra (quello delle unità del giocatore B); batteria scarica significa che la tensione letta a valle del diodo D1 è minore o uguale a 11 volt.

Per uscire dalla modalità "visualizzazione tensione di batteria" occorre premere nuovamente P2: il tabellone, quindi, si avvia in modalità predefinita.

Per ultima, vediamo l'impostazione della modalità di aggiudicazione di una partita: all'accensione, il sistema è impostato per considerare una gara vinta quando viene raggiunto un punteggio pari "al meglio di 5", ovvero vince il giocatore che conquista tre set (3-0, 3-1, 3-2). Per passare alla modalità "al meglio di 3", bisogna dare tensione (chiudere SW1...) mentre si trasmette premendo il tasto 3 (assicurarsi sempre che il TX sia in trasmissione). Il segnapunti, quindi, entra automaticamente nel normale funzionamento da tennis e si predispone per chiudere una gara a 2 punti (2-0 o 2-1). Ricordatevi che l'impostazione del modo "al meglio di 3" non viene memorizzata e per-



tanto, alla successiva riaccensione, il sistema partirà nella modalità predefinita.

Realizzazione pratica

Ora che abbiamo spiegato tutto quel che riguarda lo schema elettrico e il funzionamento, possiamo dare qualche indicazione circa la costruzione dei circuiti che compongono il segnapunti; i master sono come sempre scaricabili gratuitamente dal sito della rivista (www.elettronica.in.it). Prima di procedere con la preparazione degli stampati, sappiate che dovete realizzare un c.s. dell'unità di controllo (centrale) e quattro dei display o digit; notate che il primo è a doppia ramatura mentre gli altri sono dei semplici monofaccia. Incise e forate le piastre, procedete con il montaggio della scheda di controllo, disponendovi i componenti in ordine di altezza e verificando, con il disegno di montaggio corrispondente, il corretto orientamento di diodi al silicio e led (in questi ultimi il catodo è il terminale che sta dal lato smussato) integrati e cicalino; non preoccupatevi per il modulo ibrido, perché entra

solo nel verso giusto.

Notate che i led grandi (in somma, tutti esclusi LD11, LD12, LD13, LD14 ed LD15...) devono essere posizionati sul lato delle saldature; ciò perché devono apparire dalla stessa parte di quelli delle schede display.

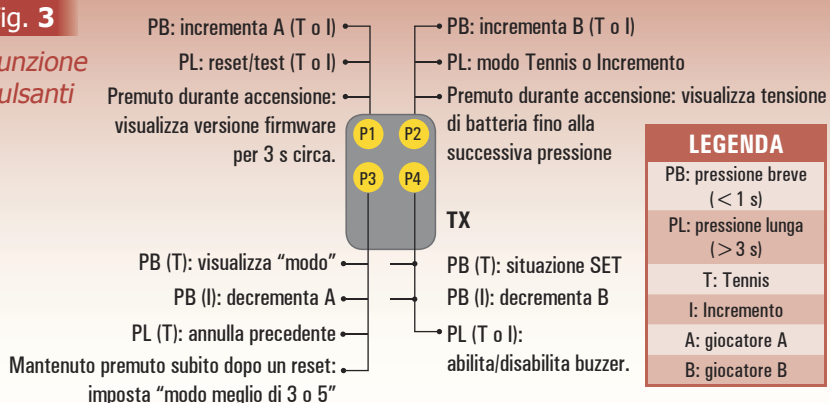
Per realizzare le connessioni con la batteria, l'eventuale pannello solare e le prese di alimentazione e caricabatteria, bisogna prevedere morsettiere da c.s. a passo 5 mm: due tripolari per PWR e CHARGE ed altrettante bipolari per BATT e SOL. Quanto alle connessioni con i circuiti display, nonché con l'eventuale pannello di telecomando a filo, sappiate che occorre montare sulla basetta cinque connettori maschio da c.s. (tipo AMP MODU II o simili) per flat-cable (passo 2,54 x 2,54 mm) da 10 poli (due file da 5 pin l'una) del tipo verticale (a terminali dritti).

Sistemata l'unità di controllo passate ai circuiti stampati dei display, montando su di essi resistenze, zoccoli per gli ULN2803 e led, sempre rammentando che in questi ultimi il terminale di catodo è riconoscibile dal fatto che sta dal lato smussato. Visto che i c.s. in questione sono monofaccia, sarà necessario realizzare i ponticelli (due per basetta) indicati nella disposizione componenti, usando, allo scopo, avanzi di reofori tagliati da resistenze o condensatori.

Per i collegamenti con la scheda base, bisogna montare su ciascuno un connettore maschio per flat-cable a passo 2,54 x 2,54 mm, da 10 poli (due file da 5 contatti ciascuna) stavolta non verticale, ma con ter-

Fig. 3

Funzione pulsanti



LEGENDA

PB: pressione breve (< 1 s)
PL: pressione lunga (> 3 s)
T: Tennis
I: Incremento
A: giocatore A
B: giocatore B

minali a 90°. Le connessioni tra la scheda di controllo e le unità display si realizzano mediante spezzoni di flat-cable terminanti con appositi connettori volanti femmina a 10 poli. Terminato il montaggio delle singole unità e verificato che tutto sia a posto, si possono inserire gli integrati nei rispettivi zoccoli; ricordate che il microcontrollore deve essere già stato programmato con l'apposito software. Per l'alimentazione, occorre realizzare l'apposito cablaggio (vedi disegni): notate che noi abbiamo previsto tutte le opzioni, ossia quella che consente tutte le possibilità di funzionamento (a batteria o tensione di rete, con ricarica della batteria da rete o da celle solari, più interruttore di alimentazione e buzzer esterno). Rammentate che per caricare la batteria serve un alimentatore da rete che fornisca 14÷15 Vdc e una corrente di almeno 300 mA; per far funzionare il segnapunti senza batteria e quindi direttamente con l'alimentatore da rete, servono da 13 a 15 Vdc e una corrente di circa 1 A.

Come si usa il radiocomando

Prima di poter comandare il tabellone segnapunti, bisogna abbinare il trasmettitore al ricevitore mediante un'operazione di "autoapprendimento", che può essere effettuata solo con TX caratterizzati da encoder Microchip HCS301 della stessa partita, cioè aventi il medesimo codice di fabbricazione (Device code)

di 64 bit. In ogni istante, salvo che non ci si trovi all'interno di un'altra procedura, premendo e rilasciando il pulsante di programmazione si entra nella fase di apprendimento, segnalata dal fatto che il led LD11 lampeggia rapidamente per 10 secondi; durante questo tempo, operando entro il campo di copertura radio del ricevitore, la pressione di un qualsiasi tasto del trasmettitore determinerà l'apprendimento del codice corrispondente al TX. L'avvenuta memorizzazione e il conseguente abbinamento trasmettente/ricevente vengono indicati dall'LD11 con l'immediata interruzione del rapido lampeggio e l'accensione a luce fissa per qualche istante; poi il led si spegne. L'apprendimento permette al ricevitore di riconoscere, ciascuno con la funzione corrispondente, tutti i quattro pulsanti disponibili nel TX. Con la stessa procedura appena descritta, è possibile fare apprendere al ricevitore fino a 10 trasmettitori. Dopo l'abbinamento a un trasmettitore, il ricevitore è predisposto per far funzionare tutte le sue uscite in modalità monostabile (l'uscita è attiva per tutto il tempo che viene premuto il relativo tasto nel trasmettitore; questo è il modo di funzionamento da noi utilizzato). Evitate di premere tasti del trasmettitore nei 10 secondi durante i quali LD11 lampeggia rapidamente a seguito dell'apprendimento: diversamente, il canale corrispondente verrà impostato in modo bistabile, il che impedirà il corretto uso delle funzio-

ni ad esso associate. Il cambiamento del modo di funzionamento delle uscite può essere fatto solo dopo che il ricevitore ha appreso il codice del trasmettitore usato per la configurazione; quindi, se per sbaglio avete impostato in bistabile un canale, ritegnete l'apprendimento e, durante il lampeggio veloce, premete di nuovo lo stesso pulsante della trasmittente. Così facendo, il led passa da lampeggiante a luce fissa; nei successivi 10 secondi è possibile premere il tasto relativo all'uscita che si vuole far tornare impulsiva; l'avvenuto ripristino della modalità impulsiva per l'uscita in questione viene segnalata con due lampeggi del led.

Un'ultima nota riguarda la memoria del modulo RX4M-HCS: volendo rimuovere tutti i dati in essa, bisogna premere il pulsante P1 e rilasciarlo dopo che LD11 inizia a lampeggiare. Ora si deve ripremere nuovamente il pulsante tenendolo premuto per circa cinque secondi fino a quando non si spegnerà nuovamente. Al rilascio del pulsante il predetto led lampeggerà cinque volte, ad indicare che la memoria è stata cancellata. E' sempre consigliabile effettuare questa operazione alla prima installazione del ricevitore.

A questo punto non resta che darvi appuntamento al fascicolo di marzo nel quale approfondiremo il funzionamento del firmware e presenteremo i piani costruttivi del contenitore metallico nel quale alloggiare il nostro segnapunti.

per il

MATERIALE

Il materiale utilizzato in questo progetto è facilmente reperibile presso qualsiasi negozio di elettronica. Il master del circuito stampato ed il firmware col quale programmare il microcontrollore possono essere scaricati dal sito della rivista (www.elettronica.in.it). Il microcontrollore già programmato (cod. MF720) può essere acquistato al costo di 18.00 Euro, IVA compresa.

Il materiale va richiesto a: Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)

Tel: 0331-799775 ~ Fax: 0331-792287 ~ <http://www.futurashop.it>

Chiave DTMF bidirezionale un canale

dell'Ing. *Mirco Segatello*

Collegata alla linea telefonica, è in grado di ricevere dei comandi, impartiti tramite bitoni DTMF, e di controllare un carico con un relè di potenza. Risponde automaticamente all'arrivo di una chiamata ed identifica il chiamante attraverso una password.

A cosa serve una chiave DTMF? Questa domanda, apparentemente banale, ha una sua ragione di essere: non è sempre facile, infatti, conoscere tutti i circuiti in campo elettronico, soprattutto se questi hanno funzioni molto specifiche e sono difficilmente reperibili in commercio. Bene, una chiave DTMF è un circuito elettronico da collegare alla linea telefonica (ad esempio quella di casa vostra) che può essere comandato a distanza tramite una semplice chiamata, in modo che possa eseguire le operazioni per cui è stato progettato. Il tutto ha una certa utilità se pensate che, tramite una

chiave DTMF, si possono controllare dei carichi, anche alimentati a tensione di rete e di potenza elevata (climatizzatori, caldaie, stufe elettriche, impianti di irrigazione, eccetera) oppure leggere gli stati di alcuni ingressi, collegati a sensori di vario tipo. Le applicazioni possibili le potete immaginare voi stessi: collegato alla linea telefonica della casa di vacanze permette di attivare il riscaldamento o il condizionamento qualche ora prima di arrivare; oppure, collegato all'impianto di illuminazione, permette di accendere le luci a distanza, simulando la presenza di qualcuno all'interno dell'abitazione. Ma non ➤

- Risposta automatica con interfaccia telefonica inserita;
- Numero di squilli alla risposta impostabile da remoto;
- Comandi tramite toni DTMF standard;
- Conferma ricezione comando tramite invio di un segnalazione acustica;
- Gestione relè:
 - modalità monostabile (con tempo regolabile);
 - bistabile;
- Lettura di un ingresso digitale optoisolato (comando da $5 \div 12$ Vdc);
- Funzione PTT per apparato radio;
- Accesso tramite password a cinque cifre (modificabile in remoto);
- Possibilità di blocco hardware della configurazione;
- Funzione di autoripristino per mancanza di tensione di alimentazione (disinseribile in remoto);
- Impostazioni predefinite:
 - numero di squilli prima della risposta automatica = 3;
 - password = 11111;
 - ripristino = OFF.

solo. Disponendo di un ingresso digitale, potremo verificare a distanza il funzionamento di qualsiasi apparecchiatura, controllare se ci siamo ricordati di inserire l'impianto antifurto o di spegnere le luci, verificare lo stato delle tapparelle elettriche e così via. Siamo certi che ognuno di voi saprà sfruttare al meglio le caratteristiche di questo circuito.

Volendo, il circuito proposto può essere connesso anche ad un apparato radio ricetrasmittente, permettendo così una gestione a distanza svincolata dalla linea telefonica; anche in questo caso si potrebbero pensare molte possibili applicazioni di sicuro interesse e l'utilità di un simile dispositivo è confermata dal fatto che i sistemi anti-intrusione più moderni offrono, tra le tante funzioni, anche

la possibilità di gestire dei carichi e controllare degli ingressi esattamente come il nostro circuito: seguitemi nelle prossime pagine e scoprirete tutti i segreti della nostra chiave DTMF.

Schema elettrico: la sezione digitale

Analizzando lo schema elettrico, ciò che appare evidente è la semplicità del dispositivo, per il quale sono stati previsti due stampati, uno dedicato alla sezione digitale di controllo e decodifica (il cuore del sistema) ed uno specifico per l'interfaccia per la linea telefonica, nota anche come DAA, acronimo che significa "Data Access Arrangement" e che utilizzeremo nel corso di queste pagine.

Iniziamo la descrizione del circuito dalla sezione digitale: il cuore del sistema è rappresentato dal microcontrollore, un PIC16F88 che svolge tutte le operazioni di gestione dei vari segnali di ingresso e uscita. Il segnale DTMF, proveniente dalla linea telefonica, viene applicato sull'ingresso I e raggiunge il decoder DTMF (U1, il noto ed affidabile 8870, più volte utilizzato in passato) tramite la rete passiva costituita da

C1-C2-R1-R2-R3. In particolare, R1 è un trimmer che permette di dosare opportunamente l'ampiezza del segnale in ingresso al decoder. Quando un bitono DTMF viene presentato all'ingresso del decoder (pin 2 di U1), questi lo decodifica e lo converte in un codice binario a cinque bit (quattro bit dati più un bit di sincronismo) che viene passato in modalità parallela al PIC. Il collegamento tra decoder e microcontrollore (U2) è parallelo e fa uso delle linee di I/O RB0÷RB3 e RB6, configurate dal firmware come ingressi. L'impiego del decoder DTMF semplifica notevolmente il compito del PIC che può riconoscere quale tono arriva dalla linea telefonica (ossia quale tasto è stato premuto sulla tastiera del telefono remoto) interpretando in modo opportuno un codice binario. La linea STD del decoder (collegata al pin RB6 del microcontrollore) è il segnale con cui il decoder conferma al micro di avere:

- ricevuto e decodificato correttamente un tono DTMF;
- variato lo stato delle uscite Q1÷Q4 è presente il codice binario relativo al tono DTMF codificato.

Il PIC, pertanto, può acquisire con certezza lo stato delle linee Q1÷Q4 ogni volta che la linea STD cambia di stato. Il sistema, però, è bidirezionale: ciò significa che il PIC può, oltre che ricevere dei dati ed interpretare dei comandi esterni, inviare in linea un tono, con il quale informare l'operatore (che si trova all'altro capo della linea telefonica) di essere pronto a ricevere un comando o di averne ricevuto correttamente uno. Può anche inviare dei toni, differenti in base allo stato dell'ingresso digitale. La nota viene generata tramite la linea RA3 che comanda la base del transistor T2 attraverso la resistenza R11. La rete R12-C8 serve per "ammorbidire" il segnale digitale generato dal PIC: in pratica, limita il contenuto armonico dell'onda quadra, così da non iniettare sulla

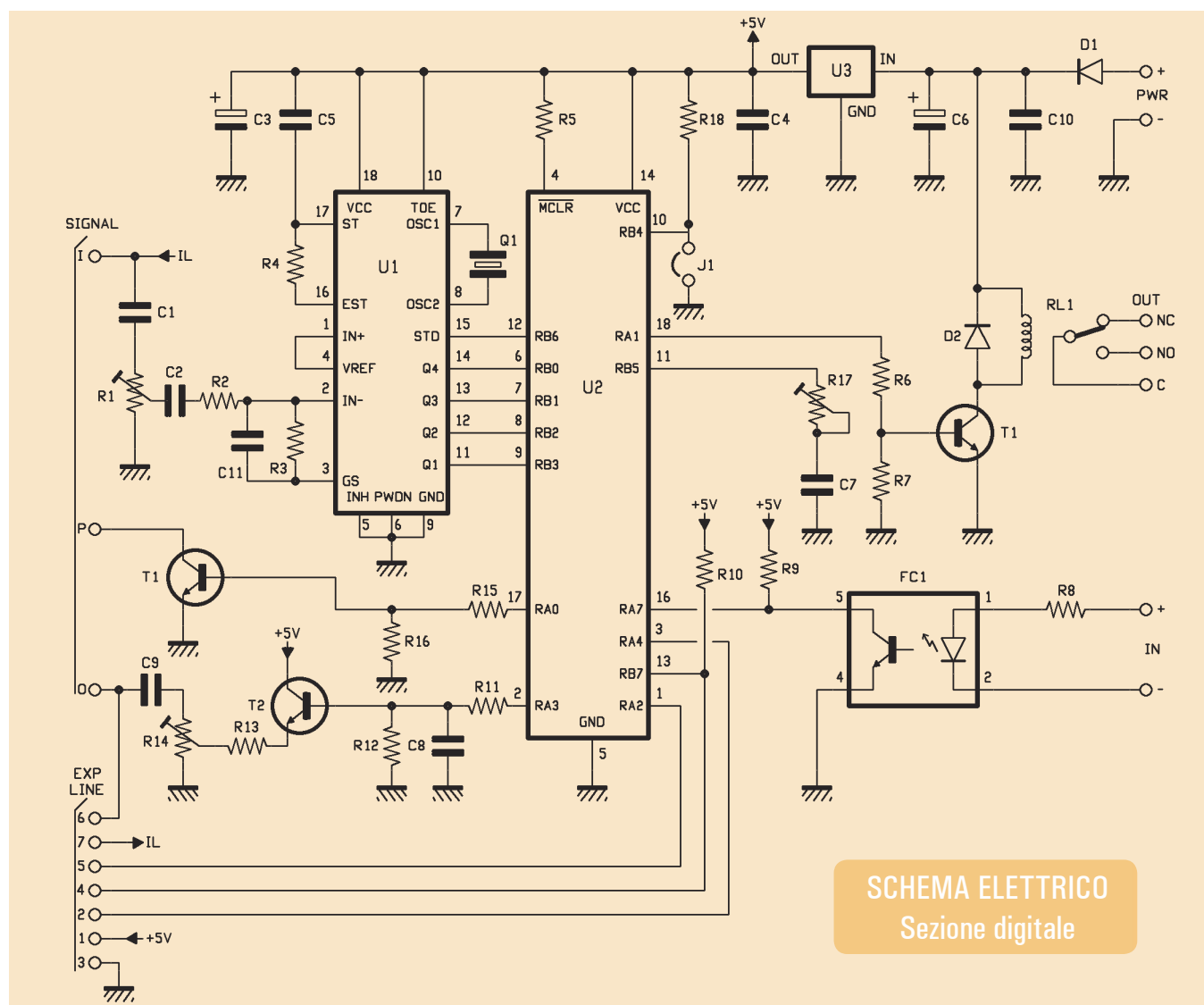


linea telefonica dei segnali analogici a frequenza superiore a 4 kHz e di livello troppo alto. L'ampiezza di questo segnale può essere regolata attraverso il trimmer R14.

L'uscita RA1, opportunamente amplificata in corrente tramite il transistor T1, comanda il relé RL1, il cui contatto è utilizzato per controllare dei dispositivi esterni (alimentati anche a tensione di rete di 230 Vac). Come abbiamo già anticipato, è anche possibile verificare lo stato di un ingresso: fornendo un segnale in continua (di valore compreso tra 5 e 12 Vdc) all'ingresso dell'optoisolatore FC1, è possibile far commutare la linea RA7, utilizzata dal PIC come ingresso digitale. La rete formata da R17 e C7 viene usata per impostare

il tempo di eccitazione del relé, nel caso lo si voglia utilizzare in modalità monostabile. Per l'alimentazione della chiave DTMF può essere usato un comune adattatore non stabilizzato da 12 Vdc, 500 mA, dai quali vengono ricavati i 5 V necessari al funzionamento della sezione digitale tramite il regolatore di tensione U3 (un 7805) che, dato l'esiguo consumo in corrente del circuito, non necessita di alcun dissipatore. Il diodo D1 protegge il circuito da inversioni accidentali della polarità, mentre la rete capacitiva C6-C10 filtra l'alimentazione da eventuali disturbi provenienti tanto dalla linea elettrica quanto dalla commutazione dei contatti del relé, soprattutto quando si attivano grossi carichi alimentati a

tensione di rete. Relativamente alla sezione digitale, non abbiamo ancora detto tutto: l'ingresso RB4, gestibile tramite il jumper J1, blocca la configurazione interna del PIC quando viene posto a zero logico. La linea RA0 comanda l'uscita P tramite il transistor T1: tale uscita viene utilizzata per comandare il PTT (Push To Talk, schiaccia per parlare) ossia il tasto che, negli apparati radioricetrasmittenti, commuta dalla ricezione alla trasmissione e viceversa. Il circuito così proposto è perfettamente in grado di operare con qualsiasi apparato radio purché dotato di un ingresso e di un'uscita audio nonché di un ingresso PTT. Il connettore denominato "EXP LINE", infine, serve per collegare alla sezione digi- ➤



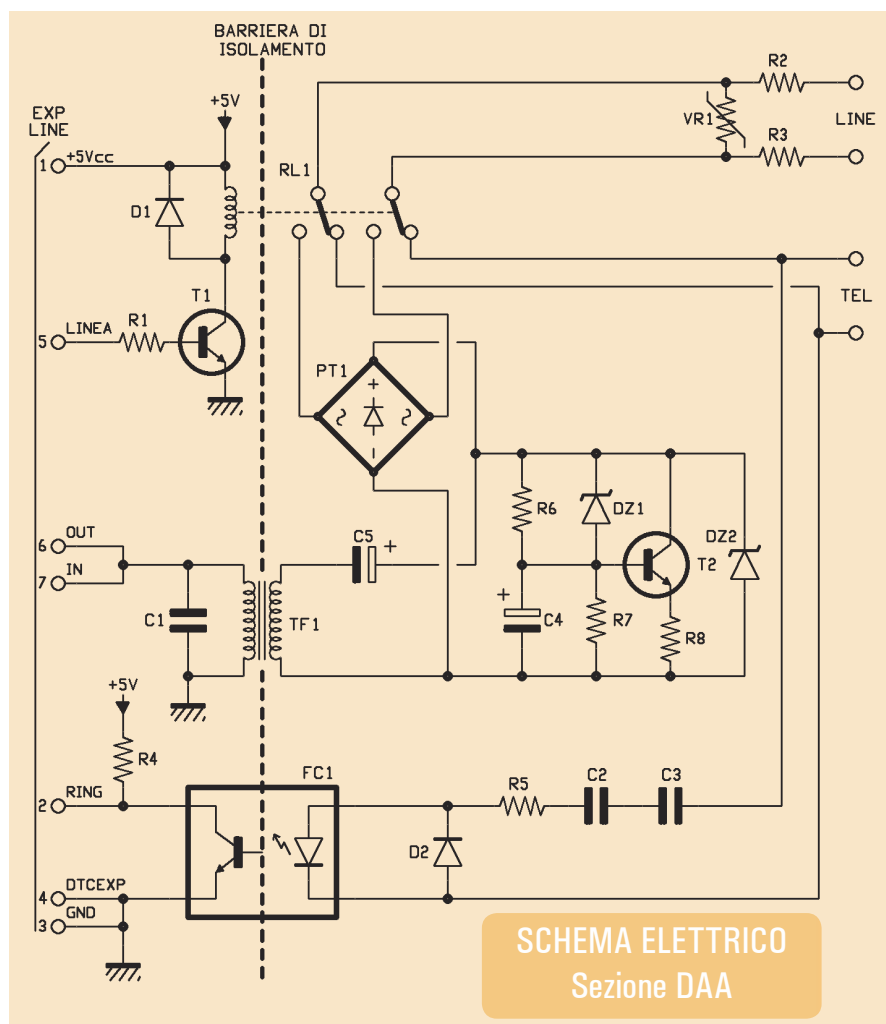
tale il circuito del DAA; fanno capo ad esso, oltre ai due terminali di alimentazione (+ e massa), anche la linea di ingresso del decoder DTMF (IL), il circuito di emissione del tono (O) e tre linee digitali provenienti dal microcontrollore: RA4 rileva il segnale di chiamata (RING), RA2 comanda il relé del DAA ed RB7 rileva la presenza o meno della scheda telefonica.

Schema elettrico: il DAA

Questa scheda, montata fisicamente sopra il circuito digitale, è collegata in serie alla linea telefonica: l'ingresso "LINE" va collegato alla borchia Telecom tramite un cavetto piatto con connettore RJ11, facilmente acquistabile in qualsiasi negozi di telefonia o supermercato, mentre l'uscita "TEL" va al telefono. Il circuito, molto semplice, può essere concettualmente suddiviso in tre sezioni: il circuito di commutazione, la bobina di tenuta ed il rilevatore di chiamata (che descriviamo per primo). Il rilevamento degli squilli telefonici avviene tramite l'eccitazione del fotoaccoppiatore FC1 (pilotato dal segnale di ring che giunge attraverso R5-C2-C3) che fa commutare di conseguenza la linea "RING". Tale commutazione, come già visto, arriva all'ingresso RA2 del PIC. Il segnale di ring è formato da una segnale a 25/50 Hz della durata di 1 secondo che si ripete ogni quattro: in origine veniva usato per far vibrare il ronzatore interno dei vecchi telefoni elettromeccanici. Oggi, invece, la suoneria telefonica è generata elettronicamente, ma è sempre agganciata al segnale di ring. Notate che il rilevatore è collegato fisicamente in parallelo al telefono, ma solo quando il DAA si trova in condizione di riposo (relé RL1 diseccitato). Quando rileva la chiamata entrante, il micro inizia a contare gli squilli e, al raggiungimento del valore impostato, attiva la linea RA2 che, a sua volta, provvede a far commutare il relé RL1 tramite

il transistor T1. A relé attivo, la linea telefonica viene deviata (escludendo così sia il telefono che il rilevatore di chiamata) verso il ponte raddriz-

quella adottata da noi), ma la definizione è stata mantenuta. Torniamo ora alla nostra linea telefonica che, a riposo, fornisce circa 48 volt negativi



zatore PT1, il cui scopo è fornire alla "bobina di tenuta" una corrente con la giusta polarità, indipendentemente dalla polarità in arrivo dalla borchia. Ma cos'è la bobina di tenuta? In origine, quando i modem erano molto lenti ed i telefoni prevalentemente elettromeccanici, il circuito di linea era rappresentato da un unico trasformatore che, con il primario, assorbiva la corrente necessaria ad attivare i commutatori di centrale; con il secondario, invece trasportava il segnale fonico. Ebbene, questo trasformatore era definito bobina di tenuta. Nelle apparecchiature moderne, il trasformatore è stato sostituito da soluzioni integrate o ibride (come

limitati in corrente. Dopo che il relé RL1 ha commutato, sui pin + e - del ponte PT1 sono presenti poco meno di 48 V (per un breve istante), che mandano in conduzione la bobina di tenuta, composta dai componenti T2, R6, C4, R7, R8, DZ1, DZ2. Il condensatore C4, inizialmente scarico, è equivalente ad un cortocircuito; il diodo Zener DZ1, pertanto, va immediatamente in conduzione e carica C4 in un tempo molto breve tramite un picco di corrente, mandando contemporaneamente in conduzione T2 che fa abbassare la tensione tra i pin + e - di PT1 con l'aumentare della sua polarizzazione. Lo Zener rimane in conduzione fino a che la tensione

su C4 non raggiunge i 4 volt circa. A questo punto la bobina entra in una condizione di lavoro lineare, in cui la polarizzazione del transistor T2 viene garantita dalle resistenze R6 e R7: la linea telefonica rimane pertanto occupata. Il filtro passa basso rappresentato da R6 e C4 impedisce che la base di T2 venga modulata dal segnale in banda fonica (300÷3400 Hz), segnale che, invece, può raggiungere il trasformatore di linea TF1, il cui secondario è collegato sia all'ingresso del DTMF decoder sia al transistor da cui si preleva il tono di uscita. Si può pertanto affermare che la bobina di tenuta presenta un comportamento resistivo nei confronti della componente continua necessaria a mantenere la linea occupata, mentre si comporta come un'alta impedenza nei confronti del segnale fonico.

A questo punto, è possibile inviare e ricevere segnali audio sulla linea telefonica tramite le linee IN e OUT. Sembrerà strano che i due segnali "IN" e "OUT" siano in cortocircuito tra di loro ma la cosa è possibile in quando l'invio e la ricezione di segnali sulla linea non avviene in contemporanea (modalità full-duplex) ma alternativamente (modalità half-duplex). Rimangono R2, R3 e VR1, che proteggono il DAA da eventuali scariche elettriche ad alta tensione, non rare in presenza di temporali. VR1 è un varistore da 220 V che, in presenza di sovratensioni in linea, ne limita il livello. R2 e R3 (entrambe da 10 ohm) invece, limitano la corrente che scorre nel varistore, così da evitarne, per quanto possibile, la rottura in caso di fulmine. Sappiate, comunque, che la protezione assoluta contro i fulmini non esiste: sarebbe buona norma, in caso di temporale, scollegare sempre, sia dalla borchia che dalla rete, gli apparati telefonici. Come ultimo particolare, notate che tra la parte di circuito collegata alla linea telefonica (TNV, Telephone Network Voltage) e quella collegata

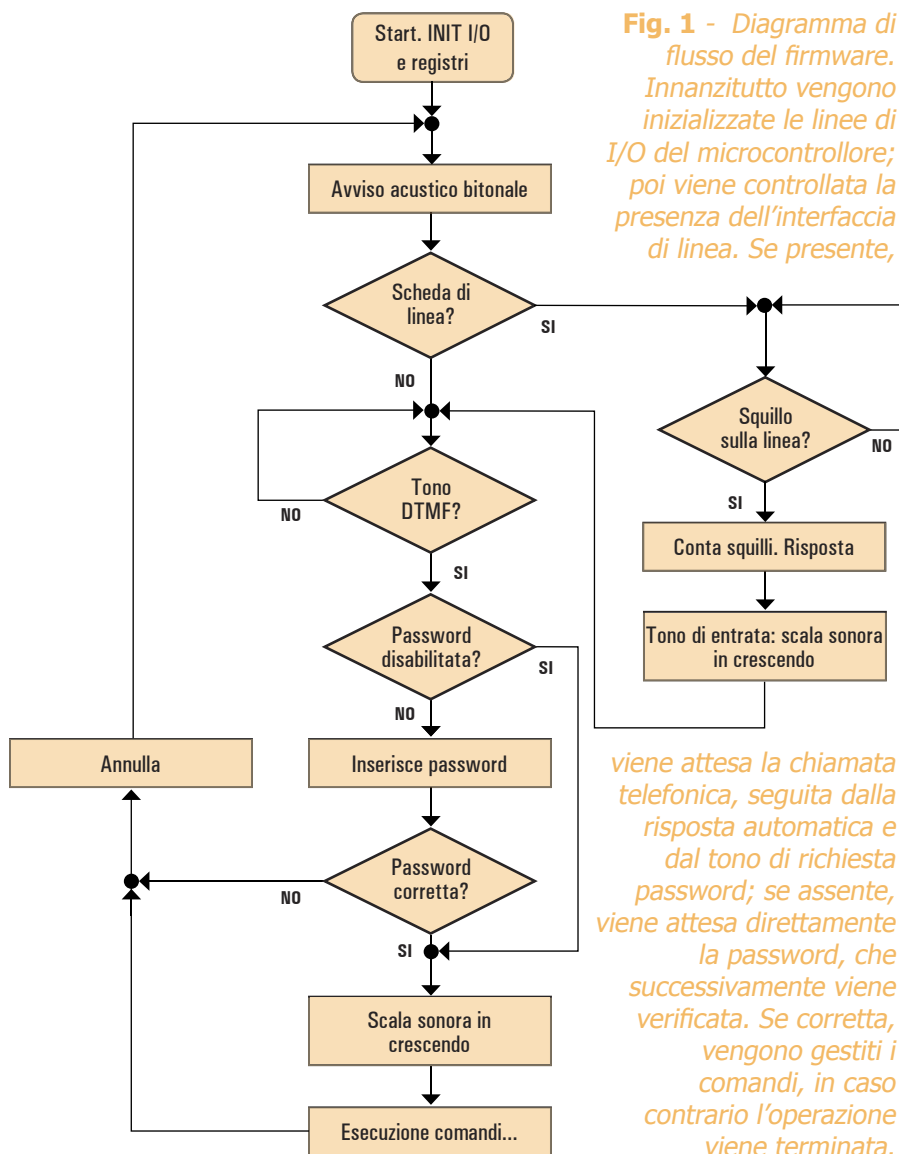


Fig. 1 - Diagramma di flusso del firmware. Innanzitutto vengono inizializzate le linee di I/O del microcontrollore; poi viene controllata la presenza dell'interfaccia di linea. Se presente,

viene attesa la chiamata telefonica, seguita dalla risposta automatica e dal tono di richiesta password; se assente, viene attesa direttamente la password, che successivamente viene verificata. Se corretta, vengono gestiti i comandi, in caso contrario l'operazione viene terminata.

alla sezione digitale (SELV, Separated Extra Low Voltage) non vi è collegamento elettrico, grazie alla separazione galvanica operata dal trasformatore, dall'optoisolatore e dal relé: in questo modo la linea telefonica viene protetta da eventuali scariche provenienti dalla rete elettrica e viceversa.

Descrizione funzionamento

Il firmware di questo dispositivo, nonostante l'impiego di un micro di fascia medio-bassa, è piuttosto complesso; abbiamo pertanto deciso di descriverne il funzionamento facendo riferimento al diagramma di flusso riportato in Fig. 1. All'avvio, il micro inizializza le linee di I/O e,

terminata la configurazione, cerca l'interfaccia di linea. La presenza della scheda DAA abilita l'accesso alla linea telefonica, pertanto il circuito si predispone alla ricezione e al conteggio degli squilli telefonici. Qualora il DAA non venga utilizzato, il software passa direttamente alla fase di attesa della password (ricordate che il dispositivo può anche essere collegato a dei ricetrasmittitori, purché questi dispongano di ingresso e uscita analogici, oltre al comando PTT). La ricezione di una password corretta viene confermata con un tono, cui fa seguito l'attesa e l'esecuzione dei comandi ricevuti. Una password errata o un time-out (che vedremo in seguito) terminano ➤

Tabella 1 - Elenco comandi

Tasto premuto	Funzione svolta
1	Attivazione relè
2	Lettura stato ingresso
4	Modifica funzione di ripristino*
5	Cambio password*
6	Modifica numero di squilli*
#	Uscita-sgancio linea telefonica

* Funzioni possibili sono se il ponticello J1 è aperto

le operazioni in esecuzione. Vediamo ora come si deve operare tramite linea telefonica. Per prima cosa, effettuate una normale chiamata voce (con qualsiasi telefono, cellulare e non) al numero di telefono della linea cui la chiave è connessa. All'ingresso dl nostro circuito giungono dunque gli squilli, tipici del segnale di ring: quando il numero di squilli rilevato coincide con il numero di squilli impostato, viene agganciata la linea telefonica. A questo punto la scheda si predispone alla ricezione dei cinque toni DTMF della password, toni che devono essere inviati digitando la password sulla tastiera del telefono chiamante; l'accesso alle funzioni avviene solo se la password inserita corrisponde a quella in memoria. Una volta ottenuto l'accesso, la chiave DTMF può essere controllata agendo sulla tastiera del telefono che ha eseguito la chiamata. Vediamo ora le funzioni assegnate a ciascun tasto.

Tasto 1: attivazione relé

Quando il cursore del trimmer R17 è completamente girato in senso orario, il relé viene attivato in modalità bistabile e inverte lo stato del suo contatto ad ogni comando ricevuto. La commutazione viene segnalata da una nota acuta se il relé si eccita e da due note gravi se si diseccita. Se, invece, il trimmer si trova in qualsiasi altra posizione, il relé si eccita in modalità monostabile, ovvero il relé si eccita (una nota acuta lo conferma) e rimane eccitato per il tempo

impostato tramite il trimmer (1÷10 secondi), quindi si diseccita (due toni gravi di conferma).

Tasto 2: lettura ingresso

Premendo questo tasto il microcontrollore rileva lo stato dell'ingresso e risponde con una scala sonora in crescendo di tre note se l'ingresso è alto (presenza di tensione all'ingresso IN); una scala sonora di tre note decrescenti indica, invece, che l'ingresso è a riposo (nessuna tensione all'ingresso IN).

Tasto 4: impostazione ripristino

La funzione di ripristino è molto utile nel caso di mancanza della tensione di rete: se è abilitata, la condizione del relé viene reimpostata allo stato che aveva prima che si verificasse la mancanza di tensione. Dopo aver premuto il tasto e ricevuto la conferma con un beep, è possibile abilitare questa funzione premendo il tasto 1 oppure disabilitarla premendo lo 0. Un beep finale conferma l'avvenuta operazione.

Tasto 5: cambio password

Quando il circuito riceve il tono corrispondente a questo tasto, genera due beep di conferma ed il firmware si predispone per ricevere la nuova password. A questo punto sarà possibile digitare i cinque tasti relativi alle nuove cinque cifre della password. La corretta ricezione dei toni viene confermata da un altro beep. Ricordiamo che la password di default è 11111, mentre se viene impostata una password composta da cinque zeri, non sarà necessario, nei successivi collegamenti, digitare la password.

Tasto 6: cambio numero di squilli

La ricezione del tono 6 viene confermata da tre beep. A questo punto si potrà digitare il numero sulla tastiera corrispondente al numero di squilli che si vuole impostare (tra 1 e 9). Un beep conferma l'avvenuta modifica.

Tasto “#”. Uscita

Termina la procedura di telecontrollo. Nel caso di una connessione alla linea telefonica, questo comando determina il rilascio immediato dalla linea. Un segnale acustico bitonale conferma questa funzione. L'accesso alle funzioni può avvenire solo reinserendo nuovamente la password.

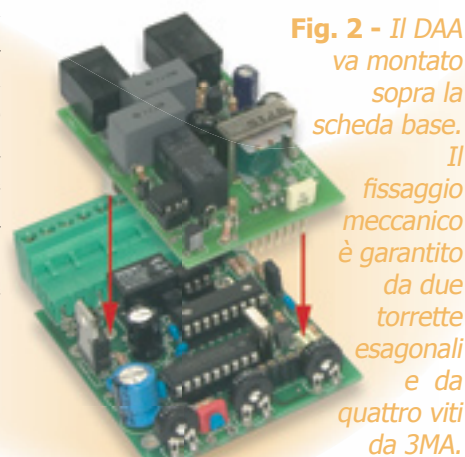
Tasti differenti da quelli previsti

Il riconoscimento di un tono DTMF (tasto premuto) non assegnato ad alcuna funzione verrà segnalato da una beep lungo.

Time-Out e note varie

È previsto un time-out che sospende qualsiasi procedura in corso (avviata tramite i tasti) in caso di inattività per più di 10 secondi. Se connessa alla linea telefonica, il time-out la libera. La nostra chiave DTMF è in grado di funzionare anche tramite telefono cellulare in quanto, durante la conversazione, è sempre possibile inviare un tono DTMF premendo un tasto numerico.

Alcuni cellulari di ultima generazione permettono, inoltre, di inviare durante la conversazione delle stringhe di toni DTMF digitate all'istante oppure precedentemente salvate, facilitando ulteriormente l'utilizzo di questo dispositivo già semplice.



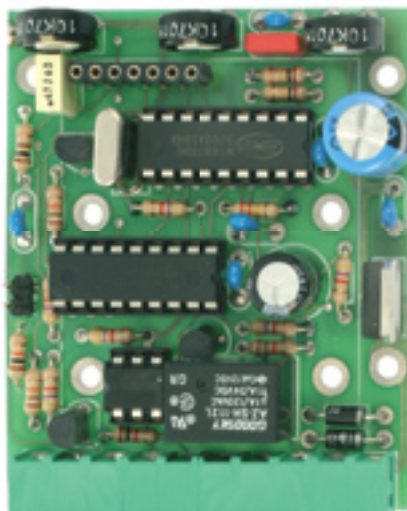
Come ultima informazione, tenete presente che l'interfaccia telefonica è stata progettata per escludere il telefono all'impegno della linea.

piano di MONTAGGIO - sezione digitale

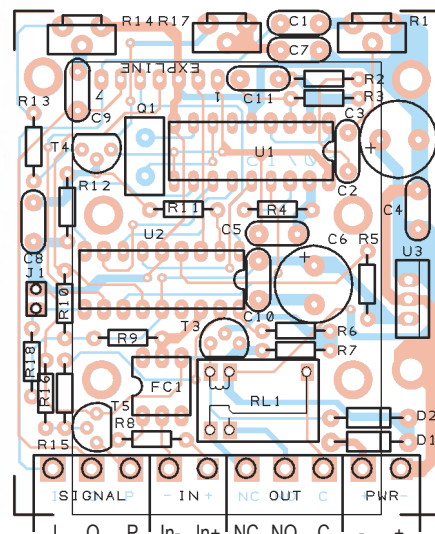
La realizzazione di questa scheda è abbastanza semplice. Notate il connettore pin strip, posto vicino al trimmer, su cui va inserita la scheda DAA. Notate, inoltre, il jumper a due poli, a sinistra di U2: la chiusura di questo ponticello inibisce la modifica da remoto dei parametri operativi tra cui la password d'accesso.

ELENCO COMPONENTI:

R1: Trimmer MV 10 kohm
R2, R7, R16, R18: 10 kohm
R3: 68 kohm
R4, R12: 330 kohm
R5, R6, R9+R12, R15: 4,7 kohm
R8: 1 kohm
R14, R17: Trimmer MV 10 kohm
C1, C2, C4, C5, C8, C10: 100 nF multistrato
C3, C6: 470 µF 16 VL elettrolitico
C7: 100 nF 63 VL poliestere
C9: 470 nF 63 VL poliestere



C11: 47 pF ceramico
D1, D2: 1N4007
T1+T3: BC547
FC1: 4N25
Q1: Quarzo 3,58 MHz
U1: 8870
U2: PIC16F88 (MF715)
U3: 7805
RL1: Relè 12V singolo scambio



Varie:

- Zoccolo 3+3
- Zoccolo 9+9 (2 pz.)
- Strip femmina 7 poli
- Torrette F/F 15 mm (4 pz.)
- Vite 3 MA (4 pz.)
- Morsetti 2 poli componibile (2 pz.)
- Morsetti 3 poli componibile (2 pz.)
- circuito stampato codice S715

piano di MONTAGGIO - sezione DAA

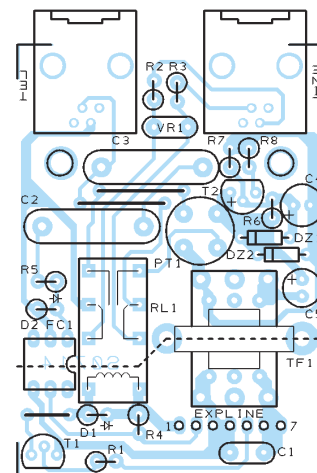
ELENCO COMPONENTI:

R1: 4,7 kohm
R2: 10 ohm
R3: 10 ohm
R4: 4,7 kohm
R5: 22 kohm
R6: 1,8 kohm
R7: 2,2 kohm
R8: 39 ohm
C1: 1 nF multistrato
C2: 1 µF 250 VL poliestere
C3: 1 µF 250 VL poliestere
C4: 100 µF 25 VL elettrolitico
C5: 10 µF 25 VL elettrolitico
PT1: Ponte diodi (W04M)
DZ1: Zener 5,1V 400mW
DZ2: Zener 27V 400mW
D1, D2: 1N4007
T1: BC547



T2: MPSA42
FC1: 4N25
TF1: Trasformatore di linea 1:1
VR1: Varistore 220V
RL1: Relè 5V doppio scambio

Le illustrazioni del DAA sono ridotte all'85%. Il varistore VR1 è molto importante: protegge il circuito da scariche elettriche ad alta tensione che possono provenire dalla linea telefonica durante i temporali.



Varie:

- Zoccolo 3+3
- Strip maschio 7 poli
- Connettore RJ12 da CS (2 pz.)
- circuito stampato codice S714

Realizzazione pratica

La realizzazione di questo dispositivo non è affatto complessa: la chiave DTMF, infatti, è realizzata con

componenti ad inserzione montati su due circuiti stampati; i relativi master possono essere scaricati, come sempre, dal sito www.elettronica.in.it.

Dopo aver realizzato i due c.s. con la tecnica della fotoincisione e forato le piazzole, potete iniziare il montaggio di entrambi partendo dai componenti ➤

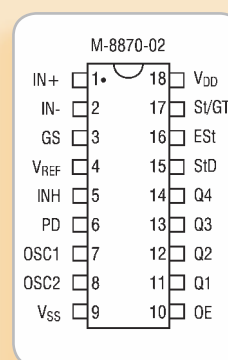
I toni DTMF

La codifica DTMF (Dual Tone Multi Frequency) viene utilizzata nei telefoni (fissi e cellulari) per inviare in linea un tono composto da due frequenze, tono che contiene l'informazione relativa al tasto numerico premuto. Questa codifica presenta due vantaggi: necessita di un numero ridotto di frequenze (8 invece che 16) e aumenta la sicurezza del comando, codificato a due frequenze invece che una sola.

	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1 (0001)	2 (0010)	3 (0011)	A (1101)
770 Hz	4 (0100)	5 (0101)	6 (0110)	B (1110)
852 Hz	7 (0111)	8 (1000)	9 (1001)	C (1111)
941 Hz	* (1011)	0 (1010)	# (1100)	D (0000)

La tabella mostra la corrispondenza tra il tasto numerico del telefono e le frequenze associate ad esso. Vicino a ogni riferimento numerico, abbiamo riportato tra parentesi il nibble binario che il DTMF decoder (M8870) genera in funzione del tono ricevuto. Ad esempio, notate che il tasto 1 è codificato con le frequenze a 697 e 1209 Hz e decodificato col codice binario 0001; tale codice è presente sulle uscite Q1+Q4, insieme al segnale di sincronismo StD.

Riportiamo anche la pin-out del decoder, che noi abbiamo scelto nella versione DIL a 18 pin, la più facile da reperire nei negozi di elettronica.



a basso profilo (resistenze, condensatori, diodi, eccetera), prestando attenzione al verso dei componenti polarizzati.

Procedete, quindi con gli zoccoli dei circuiti integrati della sezione digitale e del fotoaccoppiatore sull'interfaccia di linea; successivamente montate i transistor, i condensatori elettrolitici, i trimmer verticali, lasciando per ultimi i componenti più ingombranti quali i connettori, i relé, lo stabilizzatore di tensione (montato verticalmente), le morsettiere ed i condensatori di dimensioni maggiori, come gli elettrolitici di grossa capacità (presenti sulla sezione digitale) ed i poliestere ad alta tensione di lavoro (utilizzati nel DAA). Ultimate il montaggio inserendo i vari circuiti integrati. Il microcontrollore deve essere programmato con il file esadecimale MF715.HEX, anch'esso (come i master) sca-

ricabile gratuitamente dal sito della rivista. Le due schede devono essere connesse tra loro (sia elettricamente che meccanicamente) tramite una pin strip a passo 2,54 mm (maschio sul DAA e femmina nella scheda digitale) e due torrette esagonali alte 15 mm, come visibile in Fig. 2. Una volta acceso, il dispositivo deve funzionare immediatamente, per cui non dovete far altro che interporlo tra il telefono (collegato al connettore TEL) e la borchia telefonica (collegato a LINE) ed effettuare una chiamata al numero della linea.

Va detto che, comunque, sono necessarie alcune tarature, dipendenti dalla situazione di utilizzo e dalla vostra linea telefonica. In particolare, dovete agire sui trimmer come segue:

- il trimmer R17 va regolato in funzione del modo in cui desiderate che il relé commuti (come spiegato nel paragrafo "Tasto 1: attivazione

relé"); può essere regolato a prescindere che il dispositivo impegni la linea telefonica.

- R1 serve per regolare l'ampiezza del tono DTMF ricevuto dal decoder. Poiché va regolato sperimentalmente potete iniziare con il cursore in posizione centrale.

- R14 regola il livello del tono di risposta emesso dal dispositivo; va regolato anch'esso sperimentalmente, partendo con il cursore posto in posizione centrale.

R1 e R14 vanno regolati obbligatoriamente durante la connessione.

Qualora desideriate utilizzare la chiave DTMF insieme ad un ricetrasmittente radio, l'interfaccia di linea non deve essere montata. Dovete invece collegare il dispositivo all'unità radio utilizzando i morsetti I, O, P e Massa, seguendo quanto riportato negli schemi di collegamento a corredo dell'apparato ricetrasmittente.

per il

MATERIALE

La chiave DTMF è disponibile in scatola di montaggio. Il kit dell'interfaccia telefonica (cod. FT714K) costa 17,00 Euro mentre il kit della sezione digitale (cod. FT715K) costa 23,00 Euro. Entrambe le scatole di montaggio comprendono tutti i componenti, le basette forate e serigrafate, le minuterie ed il microcontrollore già programmato. I prezzi si intendono IVA compresa.

Il materiale va richiesto a: Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)

Tel: 0331-799775 ~ Fax: 0331-778112 ~ <http://www.futurashop.it>

Semaforo stradale per fermodellismo

di Davide Scullino

Doppio controller per semafori in miniatura. Può alimentare due terne di led così da simulare le lampade semaforiche usate negli incroci tra due strade; la durata del ciclo è regolabile tra 20 e 60 secondi.

Nel fascicolo precedente abbiamo presentato il primo di due progetti dedicati al modellismo ferroviario: un circuito elettronico pensato per azionare un passaggio a livello, automaticamente, al passaggio dei convogli. È giunto ora il momento di occuparci del secondo progetto: un semaforo elettronico a led, progettato per essere inserito nei plastici ferroviari; più esattamente, un circuito di controllo con il quale potete, a seconda delle vostre esigenze, controllare un semaforo miniatura di quelli per fermodellismo, ovvero creare qualcosa d'altro, destinato ad applicazioni differenti.

Allo scopo abbiamo previsto di non montare i led sulla basetta ma bensì all'esterno, rendendo disponibili su apposita morsettiera le tensioni di comando. Sebbene molto semplice, il nostro dispositivo viene gestito da un microcontrollore al fine di realizzare facilmente tutte le modalità di funzionamento tipiche dei semafori stradali veri e propri, cioè di comandare le colonnine semaforiche poste ai due lati del passaggio; a tale fine il circuito dispone di due distinte serie di uscite, destinate ciascuna ad accendere le luci rossa, gialla e verde in due diverse direzioni, come previsto negli incroci ➤

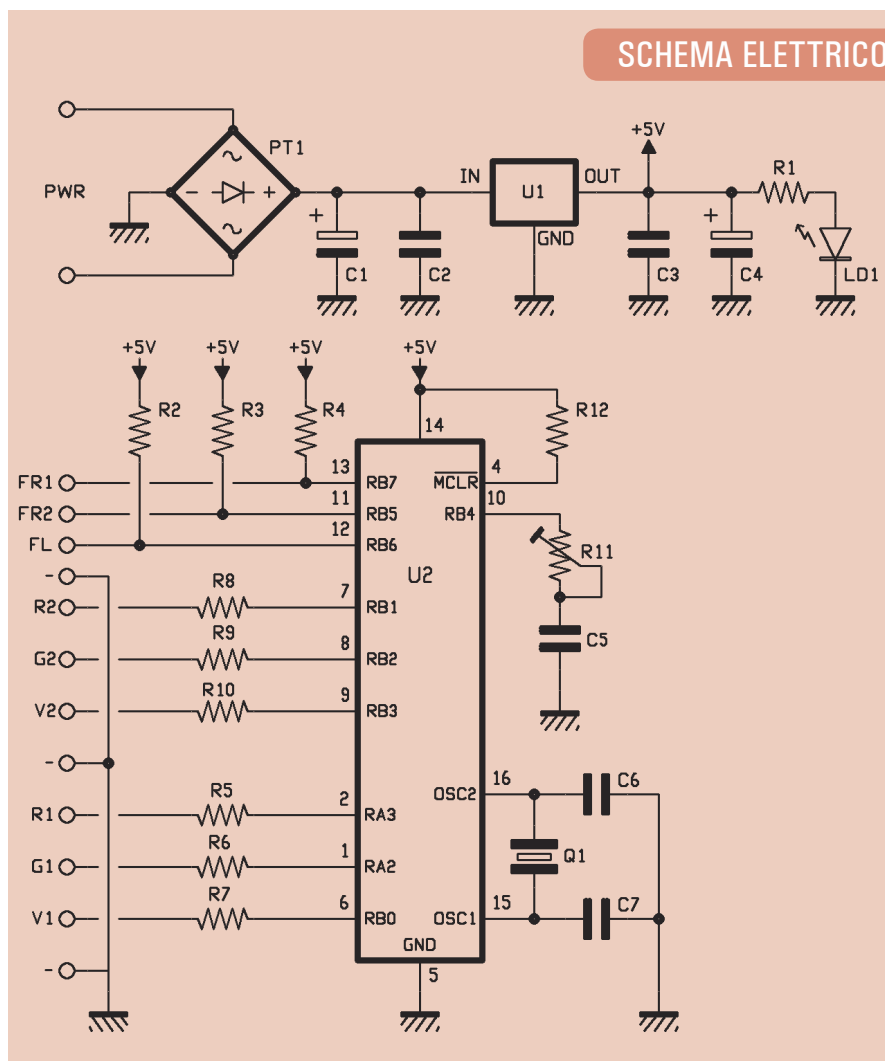
(le lampade di ciascuna colonnina funzionano sempre al contrario rispetto all'altra).

Il circuito

Da un primo sguardo si vede come il semaforo sia l'essenza della semplicità, essendo costituito praticamente dal solo microcontrollore e dall'alimentatore stabilizzato che serve a far funzionare l'insieme.

Il semaforo è bidirezionale, nel senso che può realmente gestire un incrocio: dispone, infatti, di due terne di uscite, così da eseguire su una serie di led (verde, giallo e rosso) la sequenza opposta a quella in corso nell'altra serie. Più esattamente, quando una strada dell'incrocio ha il verde, l'altra ha il rosso; quando il verde passa al giallo, la strada che incrocia rimane al rosso. Infine, non appena il giallo si spegne e compare il rosso, dopo un breve istante in cui rimangono accesi entrambi i rossi (come nei semafori veri) il lato che incrocia diventa verde. Il circuito si presta quindi a realizzare ogni possibile configurazione: lato singolo, incrocio sincronizzato a due lati e incrocio sincronizzato a quattro lati.

Stabilito il principio di funzionamento, passiamo alla descrizione dello schema: il microcontrollore è un PIC16F84A, uno dei più diffusi di casa Microchip; si tratta di un semplicissimo 8 bit ad architettura RISC con flash-EPROM, che ormai ha lasciato il posto ai più performanti PIC16F87x. Però, nel nostro caso, va benissimo, dato che gestire un semaforo richiede poche risorse. Dopo l'accensione, il micro inizializza i propri I/O impostando come uscite le linee RB0, RA2 ed RA3 e destinandole al controllo dei led, rispettivamente verde, giallo e rosso, del primo lato del semaforo; diventano uscite anche RB1, RB2 ed RB3, che comandano, nell'ordine, i led rosso, giallo e verde del secondo lato del semaforo.



Particolare attenzione la merita la linea RB4, che nel programma di gestione del PIC viene abbinata all'istruzione POT: quest'ultima permette la lettura di valori resistivi tramite la verifica del tempo di carica e scarica di un condensatore posto in serie al resistore in oggetto. Nello specifico, nella nostra applicazione ne facciamo uso per leggere il valore assunto dal trimmer (R11) col quale si imposta il tempo di esecuzione di un ciclo semaforico, tempo che può essere scelto fra un minimo di 20 e un massimo di 60 secondi.

Usiamo l'istruzione POT perché nel microcontrollore non disponiamo né di un A/D converter per leggere direttamente lo stato del potenziometro né, tantomeno, di linee di I/O sufficienti per impostare il tem-

po mediante dip-switch; in questo modo, dunque, ce la caviamo usando una sola linea. Ecco come avviene questo processo: per determinare il valore assunto da R11, mettiamo in serie a quest'ultimo C5 e usiamo il PIC per leggere la costante di tempo di scarica del bipolo R/C così formato; in pratica, il microcontrollore pone ciclicamente la linea RB4 a livello logico alto, poi la collega internamente a un timer tramite il quale conta il tempo impiegato a raggiungere il livello basso. Oltre alle linee già descritte, il PIC ne sfrutta altre tre, configurate come input e provviste ciascuna di un proprio resistore di pull-up, per impostare le modalità di funzionamento previste; tutte sono considerate, dal software di gestione, attive a zero logico. FR1,

Listato

Il programma che gira nel microcontrollore è molto semplice e presenta una routine che viene eseguita ciclicamente (con l'ausilio di un timer appositamente inizializzato) che può essere interrotta al verificarsi di un evento corrispondente al livello logico basso su una delle linee RB5, RB6, RB7. Pubblichiamo in questo riquadro le routine più significative, la prima delle quali è parte del *main program*, che gira ciclicamente fin quando la variabile "blocco" non presenta valore 1; tale variabile viene modificata attivando il comando che forza il lampeggiante. In condizioni normali, ad ogni ciclo completo il *main* azzerava le uscite ed incrementa/verifica il contatore *timer*, il cui valore determina lo stato delle uscite. Per *timer* di valore 0÷40, il lato 1 mostra il verde e il 2 il rosso. Con valore 41÷50, il lato 1 accende il giallo. Se il *timer* vale 51÷52, viene acceso il rosso in entrambe le direzioni. Con 53÷92 si inverte il ciclo di segnalazione (lato 1 rosso e lato 2 verde). Da 93 a 102 viene acceso il giallo sul lato 2, ed infine, da 103 a 104 vengono accesi i rossi in entrambe le direzioni. La seconda porzione di programma riguarda l'esecuzione della routine di forzatura del rosso su uno dei due o su entrambi: come si vede, la forzatura non fa altro che portare a termine il ciclo naturale, forzando manualmente la variabile *timer* ad un valore, legato allo stato degli ingressi FR1 e FR2, che obbliga il programma a rimanere fermo in un ciclo intermedio preciso (1° o 4° ciclo).

```
*****
IF BLOCCO=0 THEN          ' Lampeggiante disabilitato.
PAUSE PAUSA               ' Pausa (imp. trimmer).
TIMER=TIMER+1             ' Incrementa Timer
IF TIMER=105 THEN         ' Se timer supera 105
    TIMER=1               ' viene rimesso a 1.
ENDIF
LOW LR1                   ' Spegne led rosso 1.
LOW LG1                   ' Spegne led giallo 1.
LOW LV1                   ' Spegne led verde 1.
LOW LR2                   ' Spegne led rosso 2.
LOW LG2                   ' Spegne led giallo 2.
LOW LV2                   ' Spegne led verde 2.

IF TIMER >0 AND TIMER<=40 THEN ' 1° ciclo.
    HIGH LV1               ' Accende verde 1
    HIGH LR2               ' Accende rosso 2
ENDIF
IF TIMER >40 AND TIMER <=50 THEN ' 2° ciclo.
    HIGH LG1               ' Accende giallo 1.
    HIGH LR2               ' Accende rosso 2.
ENDIF
IF TIMER >50 AND TIMER <=52 THEN ' 3° ciclo.
    HIGH LR1               ' Accende rosso 1.
    HIGH LR2               ' Accende rosso 2.
ENDIF
IF TIMER >52 AND TIMER <=92 THEN ' 4° ciclo.
    HIGH LR1               ' Accende rosso 1.
    HIGH LV2               ' Accende verde 2.
ENDIF

IF TIMER >92 AND TIMER <=102 THEN ' 5° ciclo.
    HIGH LR1               ' Accende rosso 1.
    HIGH LG2               ' Accende giallo 2.
ENDIF
IF TIMER >102 AND TIMER <=104 THEN ' 6° ciclo.
    HIGH LR1               ' Accende rosso 1.
    HIGH LR2               ' Accende rosso 2.
ENDIF

*****
IF FR1=0 THEN              ' Se "Forza Rosso 1" attivo
IF TIMER >0 AND TIMER<=40 THEN ' e Timer vale 0÷40
    TIMER=40               ' sospende 1° ciclo
ELSE                        ' altrimenti
    IF TIMER=65 THEN        ' forza la permanenza
        TIMER=63            ' nel 4° ciclo (rosso 1
    ENDIF                  ' acceso e verde 2 acceso).
ENDIF
ENDIF

IF FR2=0 THEN              ' Se "Forza Rosso 2" attivo
IF TIMER >52 AND TIMER <=92 THEN ' e timer da 52÷92
    TIMER=92               ' sospende il 4° ciclo
ELSE                        ' altrimenti
    IF TIMER=35 THEN        ' forza la permanenza
        TIMER=33            ' nel primo ciclo (verde 1
    ENDIF                  ' acceso e rosso 2 acceso).
ENDIF
ENDIF
```

FR2 ed FL rappresentano gli ingressi di comando delle modalità, rispettivamente, rosso forzato sul lato 1, rosso forzato sul lato 2 e normale/lampeggiante. Alle tre linee possono essere applicati degli interruttori terminanti a massa, ma anche livelli logici prelevati da altre unità elettroniche come (nel caso dei plastici) gli stessi dispositivi che comandano i passaggi a livello. Proprio in questa evenienza, si può mandare alle linee FR il livello logico zero prelevato dal servocomando del passaggio a livello, così da far diventare rosso il semaforo quando le sbarre si stanno abbassando. Concludiamo la descrizione dello schema con il blocco di alimentazione, che inizia dal ponte a diodi PT1, grazie al quale il circuito può essere alimentato a tensione

sia continua che alternata; infatti in quest'ultimo caso PT1 raddrizza la componente bidirezionale restituendo ai capi dei condensatori di filtro C1 e C2 una forma d'onda impulsiva e unidirezionale. In continua, il ponte permette di collegare il circuito all'alimentazione senza curarsi di verificare la polarità: infatti fornisce a C1 e C2 una tensione che è sempre positiva sul + e negativa sul -, indipendentemente dalla polarità presentata ai morsetti PWR. Il regolatore integrato U1 genera i 5 volt stabilizzati (la cui presenza è segnalata da led LD1) con cui viene fatto funzionare tutto il resto del circuito.

Come funziona

Il firmware con cui è programmato il PIC consente di ottenere tre

modalità di funzionamento, che possono essere così riassunte: quella normale, nella quale si verifica l'avvicendamento delle luci verde, gialla e rossa (come previsto dalle direttive europee in materia di circolazione stradale) e poi il ritorno dalla rossa alla verde; il ciclo si svolge in un tempo che può essere regolato tra 20 e 60 secondi mediante l'apposito trimmer.

Esistono poi le modalità forzate, la prima delle quali riguarda la forzatura del rosso su un lato o sull'altro; entrambe si svolgono alla stessa maniera, solo che riguardano una il lato 1 (per il quale è stato previsto l'ingresso di comando FR1) e l'altra il lato 2 (l'input di comando fa capo ad FR2). Ponendo a livello logico basso FR1, il microcontrollore esce dal main-program (funzionamento ➤

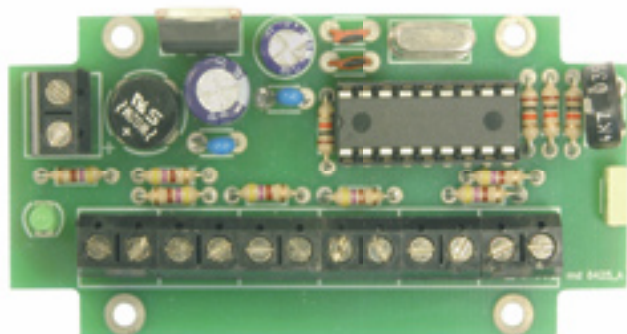
normale) e avvia la routine che gestisce l'accensione forzata del rosso sul lato 1; in osservanza di quelle che sono le modalità in uso nei semafori stradali, il passaggio non è brusco, ma prevede l'accensione del giallo, come se fosse esaurito il tempo del verde. Per l'esattezza, quando si pone FR1 a zero, sul primo lato del semaforo si spegne il verde e si accende il giallo, mentre sul lato opposto, che è rosso, rimane il rosso; finito l'intervallo del giallo, il lato 1 diventa rosso e resta in tale condizione fin quando non si riporta FR1 ad uno logico, mentre dal lato 2 si spegne il led rosso e si accende quello verde. Riportando a riposo FR1, la sequenza riprende come previsto dal funzionamento normale, quindi il lato dove è stato forzato il rosso mostra il verde e ri-

nulla cambia, perché il primo lato rimane rosso e il secondo verde. Se viene attivato FR1 quando il lato 2 è sul giallo, viene terminato il normale ciclo (che prevede il passaggio al rosso) e poi forzato il verde sul lato 2. Analogo è il discorso che riguarda l'ingresso FR2 con cui si forza il rosso sul lato 2: se questo è verde, si accende il giallo e poi si va sul rosso, altrimenti, se si compie la manovra mentre la linea R2 (rosso lato 2) del circuito è a livello alto (led rosso acceso) il semaforo resta in tali condizioni. Esattamente come visto per FR1, se FR2 viene posta a zero logico quando il lato 1 è sul giallo, da questa parte viene completata la sequenza, passando al rosso e quindi al verde. Quanto alla forzatura del giallo lampeggiante, si ottiene ponendo a livel-



G2 pulsano alla frequenza di circa 1 Hz). Riportando FL a livello logico alto, si interrompe il lampeggio, ma passa un certo tempo prima che il semaforo riprenda il normale funzionamento: circa cinque secondi durante i quali i led gialli restano accesi a luce fissa per segnalare che sta per riprendere il normale esercizio. Della procedura di lampeggio giallo forzato va notato un particolare: è prioritaria, nel senso che se anche è attiva una delle linee FR1,

piano di MONTAGGIO

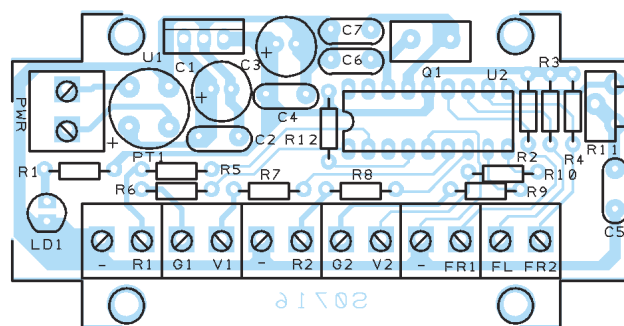


ELENCO COMPONENTI:

R1, R5+R10: 470 ohm
R2+R4, R12: 10 kohm
R3: 10 kohm
R11: trimmer MV 10 kohm
C1: 220 µF 25 VL elettrolitico
C2, C3: 100 nF multistrato
C4: 220 µF 16 VL elettrolitico
C5: 100 nF 63 VL poliestere
C6, C7: 15 pF ceramico

PT1: Ponte diodi W02M
U1: 7805
U2: PIC16F84A (MF716)
Q1: quarzo 8 MHz
LD1: led 3 mm verde

Varie:
- Morsettiere 2 poli (7 pz.)
- Zoccolo 9+9
- circuito stampato codice S716



Come si può osservare nelle illustrazioni, il circuito è veramente semplice da realizzare.

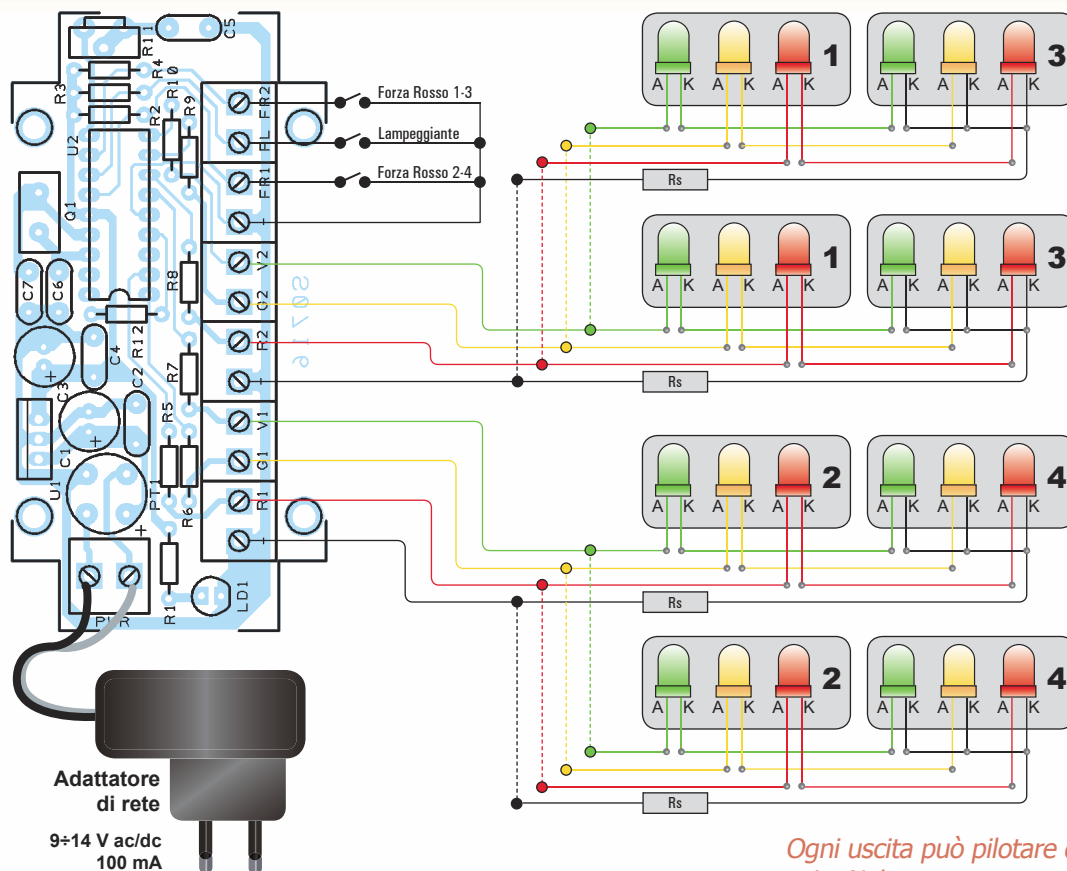
Le resistenze R5+R10 da 470 ohm sono state calcolate in modo da poter collegare un led per ogni uscita. Qualora dovesse essere necessario pilotare più di un led, potete fare riferimento al piano di cablaggio (riportato nella pagina seguente) ed alla relativa tabella, che indica i valori dei componenti da variare in funzione di tre differenti scenari di utilizzo.

parte il ciclo temporizzato secondo l'impostazione della R11. Va notato che questo è il funzionamento che si verifica se quando si attiva FR1 il lato 1 è sul verde; se è sul rosso, agli occhi dell'osservatore

lo basso la linea FL, allorché, indipendentemente dalla condizione della sequenza e dallo stato degli ingressi FR1 e FR2, entrambi i lati spengono i led verdi e rossi e fanno lampeggiare i gialli (le uscite G1 e

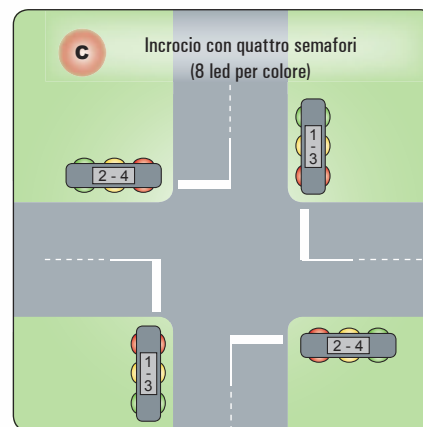
FR2 prevale sulle procedure corrispondenti; dunque, quando si attiva FL il microcontrollore sospende qualsiasi attività ordinaria o forzata e avvia la routine che realizza il lampeggio.

Piano di cablaggio



Configurazione HW in funzione dello scenario di utilizzo	Led Rossi in serie	Led Gialli in serie	Led Verdi in serie	R5 ÷ R10 (ohm)	Rs (ohm)
Scenario A	1	1	1	470	0
Scenario B	2	2	2	180	0
Scenario C " " = "in parallelo"	2 2	2 2	2 2	82	27

Ogni uscita può pilotare due led in serie. Nel caso ne occorran di più, vanno effettuare dei collegamenti serie e parallelo secondo il piano di cablaggio qui riportato. Le resistenze Rs esterne (necessarie) compensano le tolleranze delle cadute di tensione sui led.



Realizzazione pratica

Il montaggio prevede l'impiego di una piccola basetta stampata da preparare per fotoincisione (potete scaricare la traccia lato rame dal nostro sito www.elettronica.in.

it) che ospita tutti i componenti ad eccezione dei diodi luminosi (salvo LD1); questo perché i led vanno collocati nella sagoma del semaforo o perché vengono utilizzati i semafori miniatura per modellismo

con i led già presenti all'interno di questi. Inciso e forato lo stampato inserite e saldate per primi resistenze e zoccolo per il microcontrollore, poi il trimmer ed i condensatori (prima quelli non polarizzati e poi ➤

Se il semaforo è già fatto

Nella gamma degli oggetti per diorami ferroviari esistono anche i semafori stradali, che sono già completi di led miniatura i quali, dovendo restare in scala, sono SMD da circa 1,6 mm.

Le connessioni di ciascuno sono portate all'esterno mediante sottili fili di rame (uno per ogni anodo, mentre il catodo è in comune) così da poter alimentare ogni luce distintamente. Volendo pilotare con il nostro circuito uno di tali semafori, non bisogna fare altro che collegare le uscite R1, G1, V1 alle lampade di un lato dell'incrocio e le R2, G2, V2 a quelle del lato opposto. Dovendo realizzare un semaforo a quattro facce o l'intero impianto di un incrocio stradale, ogni uscita del circuito deve pilotare almeno due led, che conviene alimentare in parallelo, ciascuno con in serie una propria resistenza; ciò perché i catodi dei led dei semafori da fermodellismo sono in comune, quindi la connessione serie di due diodi non è fattibile. L'assorbimento non è problematico, dato che si tratta di led a



bassissimo consumo. Nella connessione in parallelo consigliamo di sostituire le attuali resistenze di uscita da 470 ohm con altre da 180 ohm, quindi di porre in serie all'anodo di ciascun led del parallelo un proprio resistore di limitazione da 220 ohm (ipotizzando 5 mA per ogni led). Con più di due semafori (led), l'esatto valore delle resistenze va trovato sperimentalmente, valutando la luminosità di ciascun led. Il motivo per cui conviene dare a ogni diodo una sua resistenza è che, sebbene dello stesso tipo, due componenti presentano sempre leggere differenze nella tensione di soglia. Ciò si traduce, a parità di tensione applicata, in differenze nella corrente che attraversa i led, le quali, seppure lievi, in componenti di dimensioni ridotte possono produrre apprezzabili differenze di luminosità. Usare per ciascun diodo una propria resistenza, permette di scegliere il valore adatto ad ottenere la giusta luminosità da ciascuno di essi.

gli elettrolitici), il quarzo, il led LD1, il regolatore integrato e il ponte raddrizzatore.

Per l'orientamento dei componenti polarizzati riferitevi al disegno di montaggio. Allo scopo di agevolare le connessioni con gli ingressi di comando FR1, FR2 ed FL2, nonché quelle dell'alimentazione e delle linee di led delle lanterne semaforiche (R1, G1, V1, R2, G2, V2), inserite e saldate nelle piazzole corrispondenti delle morsettiere da stampato a passo 5 mm.

Inserito il PIC16F84 (già programmato) non resta che collegare i led esterni; a riguardo si noti che le resistenze di uscita (R5, R6, R7, R8 ed R9) sono state dimensionate considerando di utilizzare un led per ciascuna linea.

Volendo realizzare un semaforo a quattro lati, bisogna sdoppiare i led, ossia disporne due per ognuna delle linee R1, G1, V1, R2, G2 e V2, così da accendere simultaneamente le lampade dei due lati opposti di ciascuna direzione. In questa evenienza raccomandiamo di connettere i diodi luminosi di ogni uscita non in parallelo ma in serie tra loro, per non sovraccaricare gli I/O del microcontrollore; infatti due led in parallelo richiederebbero il doppio della corrente mentre nei led posti in serie fluisce sempre la stessa corrente, a prescindere dai diodi impiegati. Nella connessione serie, dato che la caduta sui diodi è doppia rispetto all'uso di un singolo led, la resistenza di uscita va ridotta a 150 o 180 ohm. Un'ultima nota riguar-

da gli ingressi FL, FR1 ed FR2: se li comandate con tensioni prelevate da altri circuiti, ricordate che accettano livelli TTL (0/5 V); avendo a che fare con potenziali più elevati, conviene interporre in serie a ciascuna linea un diodo 1N4148 girato con l'anodo verso il rispettivo pin del microcontrollore.

*Semaforo in kit (cod. 161656)
prodotto da Faller (www.faller.de).*



per il

MATERIALE

Il materiale utilizzato in questo progetto è facilmente reperibile presso qualsiasi negozio di elettronica. Il master del circuito stampato ed il firmware col quale programmare il microcontrollore possono essere scaricati dal sito della rivista (www.elettronica.in.it). Il microcontrollore già programmato (cod. MF716) può essere acquistato al costo di 12.00 Euro, IVA compresa.

Il materiale va richiesto a: Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)

Tel: 0331-799775 ~ Fax: 0331-792287 ~ <http://www.futurashop.it>



3

Corso di programmazione modulo Tibbo EM1000

A CURA DI **MARCO BLANCATO**

Corso di programmazione e utilizzo del modulo EM1000 di casa Tibbo, un potente Ethernet Server completamente programmabile con funzioni di Web Server, Mail Server e molto altro ancora. Vediamo in dettaglio la piattaforma hardware e il software per PC, entrambi inclusi nel sistema di sviluppo. Terza puntata.

Nel corso delle due puntate precedenti abbiamo fatto conoscenza con l'Ethernet Server EM1000 di casa Tibbo: in particolare, nella prima puntata abbiamo visto le caratteristiche principali del dispositivo nonché alcuni esempi di potenziali applicazioni pratiche; nella seconda puntata, dedicata interamente alla struttura hardware del modulo, l'attenzione è stata focalizzata non tanto sui singoli pin quanto sulle varie fun-

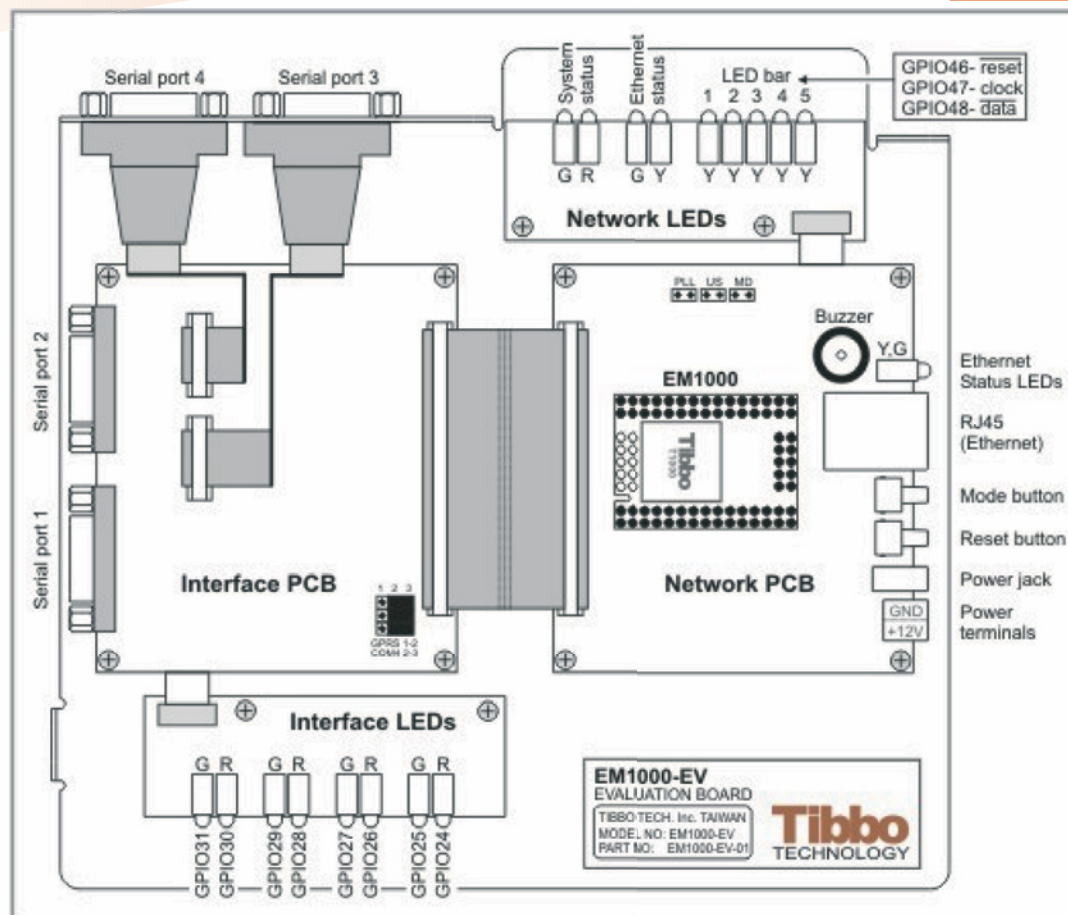
zioni (comuni a pin differenti o specifiche per pin) che ognuno di essi può supportare.

In questa puntata, invece, prendiamo confidenza con il sistema di sviluppo, sia dal punto di vista hardware che software, analizzando in dettaglio i due elementi che lo compongono ossia:

- l'Evaluation-Board EM1000-EV, che da ora in poi verrà utilizzata per i vari esperimenti,
- l'ambiente di sviluppo Taiko.



Fig. 1



L'Evaluation Board

Nonostante il nome “Evaluation Board” (che in Inglese significa “scheda di valutazione”) possa indurre a credere che si tratti di una scheda unica, il dispositivo, schematizzato in Fig. 1, è in realtà un “sistema” composto da quattro schede (cinque, considerando l’Ethernet server) connesse tra loro tramite dei “flat cable” e fissate meccanicamente ad una base metallica che conferisce robustezza e praticità all’apparato.

Scheda Network PCB

Questa scheda è il cuore del sistema: su di essa sono montati il modulo EM1000, il connettore di interfaccia Ethernet con i relativi led di stato (cavo connesso e velocità della porta), l’alimentatore switching, un buzzer, due pulsanti e tre jumper (le funzioni dei pulsanti e dei jumper sono descritte in Tabella 1). Relativamente ai led (Ethernet-Status), essi sono connessi alle linee EY (giallo) ed EG (verde) del modulo: quello verde si accende

Tabella 1 - Jumper e pulsanti scheda Network

Controllo Hardware	Funzione
Pulsante di Reset	Connesso al pin RST del modulo, permette di riavviare l’applicazione residente.
Pulsante M/D	Libero. Viene utilizzato come pulsante generico per i programmi dimostrativi.
Jumper JP1 (M/D)	Se inserito durante un reset hardware, il sistema entra in “serial upgrade mode”, stato in cui è possibile aggiornare il firmware attraverso la porta seriale. Poiché, durante il corso, gli aggiornamenti verranno effettuati tramite la porta Ethernet, questo jumper rimarrà aperto.
Jumper JP2 (U.S.)	Riservato a Tibbo; lasciare aperto.
Jumper JP3 (PLL)	Permette di abilitare il PLL e di selezionare la frequenza di clock del modulo. Se lasciato aperto, il modulo funziona alla frequenza massima di 88,4736MHz (maggiore velocità di calcolo e maggiori consumi). Se chiuso la frequenza scende a 11,0592MHz.

quando è attivo un collegamento Ethernet (cavo connesso) e lampeggia durante il transito dei dati. Quello giallo, invece, è acceso se la velocità della connessione è di 100 Mbit/s (100BaseT) mentre è spento nel caso di un collegamento a 10 Mbit/s (10BaseT). Sono presenti, infine, un jack per il collegamento dell'adattatore di rete (fornito nel kit) e due connettori di tipo dati: il primo di essi (CN1) è un flat-cable maschio che collega la scheda *Network* alla scheda *Interface*; il secondo (CN2), invece, è di tipo ZIF (Zero Insertion Force) e serve per collegare la scheda *LED*. CN1 e CN2 sono facilmente accessibili, in modo da poter collegare la scheda *Network* a circuiti differenti che dipendono dalle differenti esigenze di progetto.

Scheda Interface PCB

La scheda ospita i convertitori di livello RS232 (ADM3307, a cui abbiamo dedicato un piccolo riquadro a parte) delle quattro porte seriali e i connettori a vaschetta a nove poli (DB9) per il collegamento verso PC o periferiche seriali esterne. Due DB9 sono montati direttamente sulla scheda, mentre gli altri due sono fissati meccanicamente al basamento metallico e collegati alla scheda tramite due flat-cable. Sono presenti, poi, tre jumper (JP1, JP2 e JP3) da tenere in posizione 1-2. La posizione 2-3 serve a isolare il connettore della quarta porta seriale dell'EM1000, qualora questa dovesse venire collegata ad un modem GSM, previsto come opzione e, nel caso, collocato proprio su questa scheda.

Scheda Interface Led

È collegata fisicamente alla scheda *Interface* attraverso un flat-cable ZIF. Su di essa sono montati due led per ciascuna porta seriale (per un totale di otto

led), che mostrano il traffico in ingresso (linea RX) e in uscita (linea TX) ad ogni porta. Essi, comunque, non vengono pilotati direttamente alle linee seriali a livello RS232: sono invece collegati a otto linee di I/O del modulo (da GPIO24 a GPIO31) e possono pertanto essere usati come led generici di segnalazione per i vari esperimenti.

Scheda Network Leds

Collegata direttamente alla scheda *Network* tramite un cavetto ZIF uguale a quello visto in precedenza, su questa scheda si trovano altri nove led. Due di essi (giallo-verde) mostrano lo stato della rete Ethernet (Ethernet-Status) e replicano quelli già presenti sulla scheda *Network* PCB.

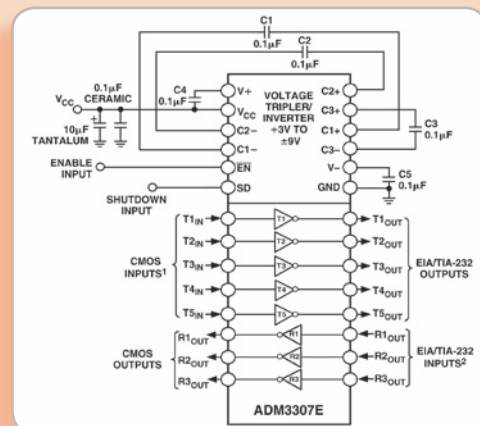
Altri due (rosso-verde) svolgono la funzione di "System Status" e forniscono informazioni utili durante le operazioni di verifica e di caricamento del programma applicativo sul modulo. L'interpretazione è, a dire il vero, un po' macchinosa e dipende dal "pattern" di lampeggio (modo in cui i led lampeggiano), quest'ultimo legato, a sua volta, al tratto di software in esecuzione. Chi volesse approfondire l'argomento prima delle prossime puntate, può consultare il manuale "Tibbo docs.pdf" a pagina 98; in questa sede ci limitiamo a indicare che

- finché i led sono spenti o lampeggia solo quello verde non sono presenti anomalie di sistema;
- un lampeggio del led rosso indica un problema di comunicazione durante il caricamento del programma, un tentativo di caricare un programma troppo grande o un malfunzionamento della memoria flash interna del modulo.

Questi due led sono connessi alle linee SG e SR del modulo e possono essere gestiti dalla nostra applicazione. Troviamo infine una barra di cin-

Il driver ADM3307

L'ADM3307 di Analog Device è un convertitore di livello "da CMOS a RS232" in grado di funzionare con una sola fonte di alimentazione compresa tra 2,7 e 3,3 Vdc. Concettualmente, è costituito da due blocchi: il triplicatore/invertitore di tensione e il gruppo driver/receiver. Il primo blocco è un convertitore DC-DC a trasferimento di carica (charge pump) in grado di ricavare le tensioni di ± 9 Vdc (necessarie per alimentare la sezione RS232) partendo dalla tensione singola di 3 V. Il secondo, invece, è costituito da otto buffer invertenti, cinque di trasmissione (T1÷T5) e tre di ricezione (R1÷R3), con i quali è possibile realizzare una porta seriale standard completa. Sono inoltre disponibili due pin con cui si abilita il trasferimento degli ingressi sulle rispettive uscite (Enable) e si spegne il chip (Shutdown).



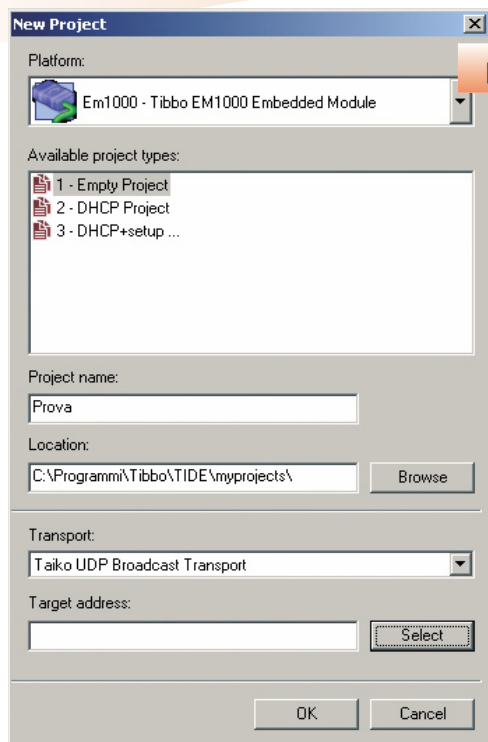


Fig. 3a

re solamente dell'hardware. Per prima cosa occorre scaricare la documentazione e il software che ci servirà: il materiale è disponibile al collegamento http://www.tibbo.com/taiko_dl.php; segnaliamo

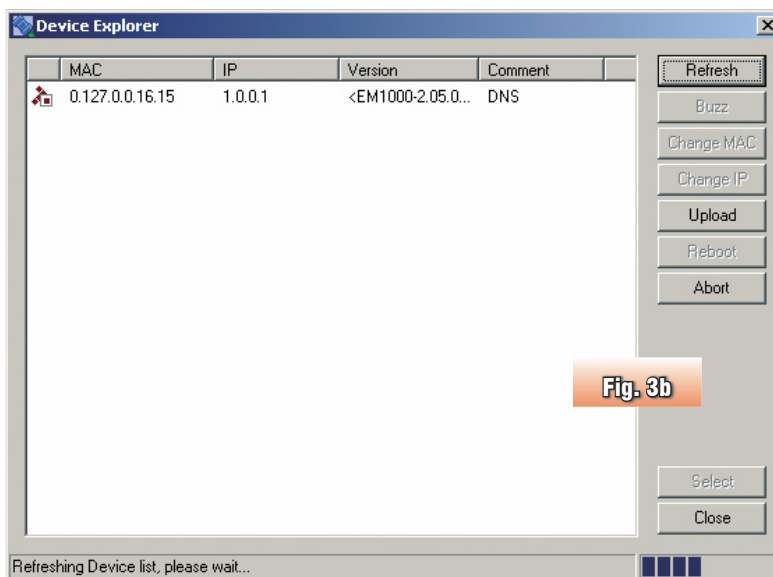


Fig. 3b

que led gialli, comandata attraverso 3 linee di I/O (GPIO46, 47 e 48) ed utilizzabile come indicatore luminoso in funzione delle esigenze di progetto. Bene, dopo aver visto come è strutturata l'Evaluation Board, siamo finalmente giunti al momento di preparare la postazione di lavoro e di “dare fuoco alle polveri”.

Step 1: installare l'ambiente di sviluppo

Il sistema di sviluppo dell'EM1000 è corredato di un CD che contiene tutto il software necessario, aggiornato al momento in cui è stato confezionato il kit. Dato che, comunque, ogni sistema di sviluppo è soggetto ad aggiornamenti e migliorie, potrebbe essere necessario dover scaricare dal sito di Tibbo e reinstallare delle nuove release firmware, per cui, in questo paragrafo, ipotizziamo di dispor-

che quando si accede alla pagina per la prima volta, per poter procedere è necessario effettuare la registrazione, procedura gratuita e molto rapida. Dopo aver effettuato la registrazione, si passa ad una nuova pagina dalla quale possono essere scaricati tre file differenti:

- **TIDE (Tibbo Integrated Development Environment)** è l'ambiente di sviluppo (simile al Basic) che permette di scrivere le nostre applicazioni e trasferirle sul modulo.
 - **EM 1000 Platform Firmware** è il firmware del modulo, cioè il sistema operativo che permette di accogliere ed eseguire i nostri programmi. Il modulo viene fornito con il firmware precaricato, tuttavia è possibile che non si tratti della versione più recente quindi, dovendo preparare o aggiornare la postazione di lavoro, conviene scaricare anche questo file.
 - **TAIKO Manual in pdf format** è il manuale di riferimento per la programmazione in Tibbo-Basic. Nella sezione <http://www.tibbo.com/downloads.php>, sottopagina “Technical Documentation, Docs in Adobe Pdf Format”, troviamo, invece, il manuale hardware di tutta la famiglia Tibbo e quindi anche la sezione riguardante l'EM1000.
- A questo punto disponiamo di tutto il necessario: possiamo quindi avviare l'applicazione *Tide-xxxx.exe* per installare l'ambiente di sviluppo, installazione che esegue le procedure tipiche di un sistema operativo Windows. Manteniamo le opzioni di installazione predefinite (Fig. 2) per eseguire l'instal-

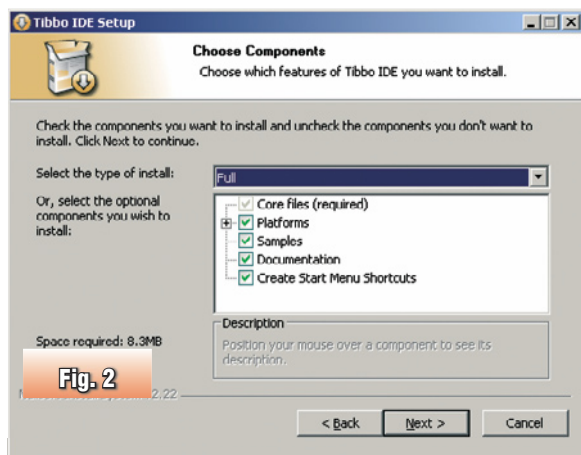


Fig. 2

lazione completa. La cartella di lavoro predefinita è identificata dal path "Programmi\Tibbo\Tibbo IDE", accessibile tramite il menu di avvio.

Step 2: i collegamenti hardware

Collegiamo l'Evaluation Board alla porta Ethernet del nostro computer (o all'Hub o Switch di rete) mediante il cavo in dotazione: si possono usare indifferentemente il cavo dritto (blu) o incrociato (verde) poiché, grazie alla funzionalità "Auto_MDIX", il modulo saprà adattarsi auto-

nella finestra che compare, selezionate "Invia a" e poi "Desktop (crea collegamento)": sarà molto più pratico, da ora in avanti, avviare l'ambiente di sviluppo direttamente dall'icona sul desktop.

Dal menu "File-New project" selezioniamo la piattaforma EM1000 (Fig. 3), evidenziamo la riga "1-Empty project", diamo un nome al progetto (ad esempio "prova"), lasciamo selezionato "Taiko UDP Broadcast transport" e clicchiamo poi su "Select" per avviare la ricerca del nostro dispositivo sulla rete. Se è stato eseguito tutto corret-

Utilizzando Windows Vista

Utilizzando un EM1000 con versione firmware antecedente alla V. 2.02.01 insieme ad un PC con sistema operativo Vista, il modulo non viene rilevato (non compare nella finestra Device Explorer) dall'ambiente di sviluppo poiché l'indirizzo IP predefinito è incompatibile con Vista. Esiste una procedura per effettuare



l'aggiornamento attraverso la porta seriale, ma è più immediato installare provvisoriamente l'ambiente di sviluppo (il file di installazione occupa solamente 5 MByte) su un computer Windows XP, effettuare l'aggiornamento del firmware attraverso la rete Ethernet quindi tornare sulla postazione Windows Vista.

maticamente al cavo. Collegiamo a questo punto l'alimentatore: probabilmente sarà visibile qualche led che lampeggia; ciò dipende da quale programma si trova eventualmente già installato sul modulo EM1000. In linea di massima, se state utilizzando un sistema di sviluppo nuovo, non dovrete vedere lampeggiare altri led oltre a quelli giallo e verde che indicano lo stato della rete. In caso contrario, non vi preoccupate: verificate invece che siano accesi i led di stato di rete secondo le modalità già viste.

Step 3: avvio dell'ambiente di sviluppo

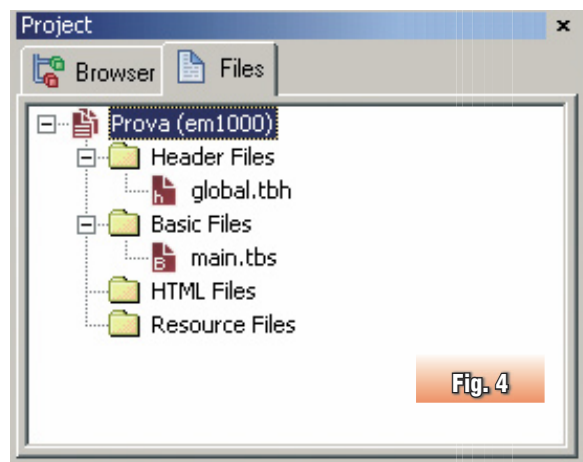
Tramite il menu di avvio, lanciate l'esecuzione dell'ambiente di sviluppo Tibbo IDE. Eventualmente, prima di eseguire l'applicazione "Tibbo IDE" cliccate su di essa con il tasto destro del mouse e,

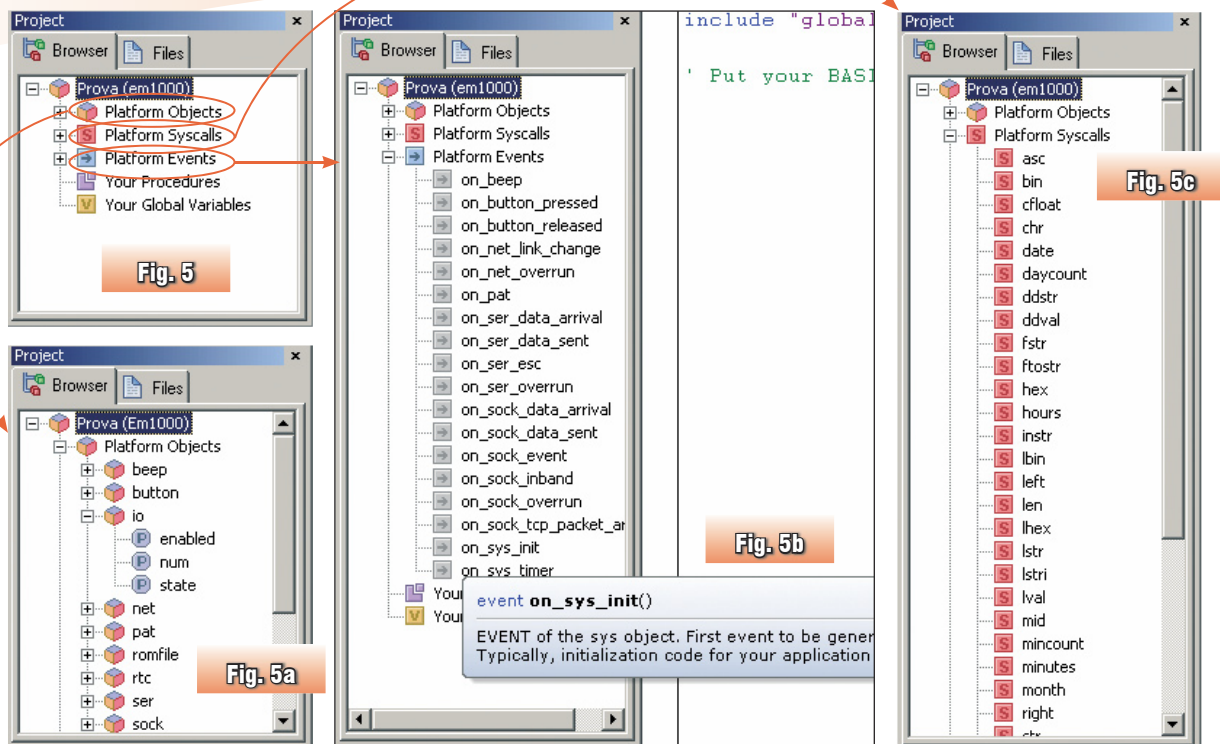
tamente, nella finestra "device explorer", compare una riga nuova che identifica la demoboard, il suo indirizzo di rete IP e la versione del firmware a bordo del modulo.

Step 4: l'aggiornamento del firmware e l'ambiente di lavoro

Anche se non è indispensabile, a titolo di esercizio ora effettuiamo l'aggiornamento firmware del modulo: dalla finestra "device explorer" si seleziona il dispositivo e poi si clicca sul pulsante "Upload". A questo punto appare una nuova finestra dove occorre impostare l'opzione "Load Firmware through the Network". Selezioniamo il file con estensione ".BIN" che abbiamo precedentemente scaricato dal sito e attendiamo i pochi secondi necessari per il completamento dell'operazione. Evidenziato nuovamente il dispositivo, clicchiamo su "Select" per tornare alla finestra precedente. Clicchiamo infine su "OK" per entrare finalmente nell'ambiente di sviluppo principale che mostra un progetto vuoto in attesa di nostre istruzioni. Prima di iniziare a scrivere il nostro primo codice sorgente, è opportuno dare un'occhiata ai vari riquadri dell'ambiente di "lavoro".

Sulla sinistra troviamo la sezione "project" (Fig. 4), suddivisa in due schede. La scheda "Files" contiene l'elenco di tutti i files che compongono il nostro progetto. In questo momento vediamo un file "global.tbh" (contenuto nella cartella "Header Files") che è aperto anche nella sezione destra per





l'editing, destinato a contenere tutte le dichiarazioni globali del nostro progetto (variabili, costanti, eccetera). Vi è poi il file "main.tbs" (cartella "Basic files"), anch'esso aperto sulla destra, nel quale viene scritto il programma in Tibbo Basic. Le cartelle "HTML" e "Resource" possono contenere file supplementari inclusi nel progetto, come ad esempio sezioni di codice HTML usate per descrivere graficamente le pagine WEB che il modulo presenta sulla rete, piuttosto che immagini, file dati e altro, richiamati all'occorrenza dal nostro codice.

Vediamo ora la scheda "Browser" (Fig. 5), che è veramente interessante e ben strutturata, perché ci permette di prendere una buona confidenza con il linguaggio quasi senza aprire una pagina del manuale. Anch'essa è strutturata ad albero come di consueto. Facciamo doppio clic sulla cartella "Platform Objects": il menu ad albero si espande ed elenca tutti gli "oggetti" di sistema (Fig. 5a) che possono essere configurati o interrogati, come ad esempio un I/O (per impostarne la funzione e conoscerne o modificarne lo stato) piuttosto che il timer di sistema (che può essere interrogato per ottenere data e ora). È sufficiente passare con il cursore del mouse sopra la voce di interesse per fare apparire

un riquadro che spiega dettagliatamente il significato dell'oggetto.

Troviamo poi i "Platform Events" (Fig. 5b) che sono simili agli "interrupt" dei microcontrollori, cioè quegli eventi che possono accadere spontaneamente (e che noi, poi, possiamo gestire opportunamente) durante l'esecuzione del programma. Sono relativi, principalmente, alla comunicazione seriale, alla comunicazione Ethernet e al timer di sistema.

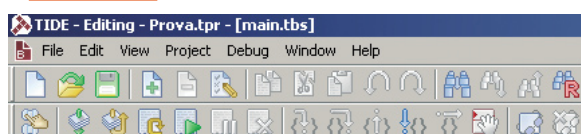
Procedendo, troviamo le "Platform Syscalls" (Fig. 5c), che sono delle funzioni già incorporate nel linguaggio per effettuare principalmente manipolazioni di stringhe e date.

Le cartelle "Your Procedures" e "Your Global Variables", infine, conterranno (quando ne avremo scritte) rispettivamente tutte le funzioni e subroutine e tutte le variabili dichiarate ed utilizzate nel nostro programma.

Nella parte alta dello schermo troviamo diverse icone che richiamano poi i comandi contenuti nei menu (Fig. 6): posizioniamo il cursore su ognuna di esse per far comparire l'etichetta di aiuto (una piccola finestra che spiega come si utilizza e quale funzione attiva tale icona) e familiarizzare con il loro significato. Il primo rigo contiene i classici comandi comuni a tutti i programmi (Nuovo, Apri, Salva, Taglia, Copia, Incolla, ecc.).

Nel secondo rigo, invece, troviamo i comandi specifici dell'ambiente di sviluppo: prima di descriverli, però, è necessario aprire una parentesi relativa

Fig. 6



alla funzione “debug”, dato che essi interagiscono strettamente con tale funzione.

Utilizzando un ambiente di sviluppo completo per una qualsiasi piattaforma a microprocessore, ci aspettiamo che tale sistema permetta di scrivere un programma, di trasferirlo all'interno della memoria del micro (in questo caso il modulo EM1000), di farlo funzionare e di analizzarlo durante le varie fasi di progetto e funzionamento. È noto, infatti, che se durante il progetto del firmware si commettono degli errori (cosa questa praticamente inevitabile), spesso è difficile identificare il punto esatto dove è presente il “baco” (errore). Attivando la funzionalità “debug”, invece, l'ambiente di sviluppo diventa un emulatore vero e proprio, ossia un sistema che consente di eseguire il programma passo per passo, di metterlo in pausa, di inserire dei punti di interruzione e controllo, nonché di osservare il contenuto di tutte le variabili (sia di sistema che “private”, cioè definite da noi) durante l'esecuzione.

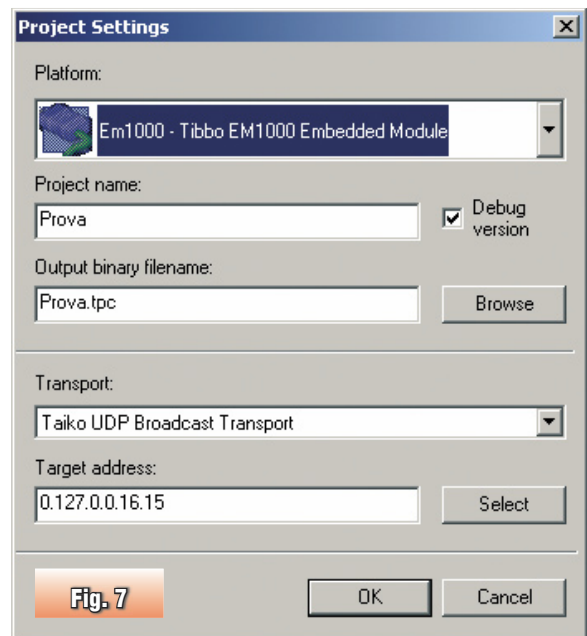
La modalità “debug” si attiva dal menu “Project – Settings”, mettendo il segno di spunta sull'opzione “Debug version” (Fig. 7): così facendo, quando il programma viene compilato ed assemblato, il sistema di sviluppo inserisce nel file “oggetto” (il file che viene fisicamente caricato nella memoria del micro) del codice aggiuntivo: quest'ultimo non è sotto il nostro diretto controllo, ma ci permette di ricavare le informazioni e di eseguire le operazioni citate poco fa. Una volta “debugato” (leggasi “eseguito il debug”; si pronuncia “debagato” e non è una parola molto elegante ma è entrata a far parte del linguaggio corrente del programmatore, un po' come “hardwarista” e “softwarista”) e completato il programma, disabilitiamo questa opzione, rifaremo la compilazione e l'upload del programma nel modulo EM1000, che potrà quindi funzionare autonomamente, cioè scollegato dall'ambiente di sviluppo sul PC.

A questo punto possiamo chiudere la parentesi sulla funzione “debug” e tornare a descrivere le icone del secondo rigo del quadro comandi (Fig. 6), che sono nell'ordine:

- **Select/Manage target:** richiama la finestra di selezione del dispositivo (Device Explorer) che abbiamo già visto in Fig. 3.
- **Build:** compila il programma evidenziando eventuali errori.
- **Upload:** trasferisce il programma compilato dal PC alla memoria Flash del modulo EM1000.
- **Restart:** invia un comando di reset al modulo.
- **Run, Pause e Stop:** attivi solo in modalità “debug”; rispettivamente “avvia, sospende e termi-

na” l'esecuzione del programma.

- **Step Into:** attiva solo in modalità “debug”; consente di inserire nel programma dei breakpoint (punti di interruzione). Il codice viene eseguito fino alla riga associata al breakpoint, per fermarsi alla riga successiva, che non viene eseguita.
- **Step Over:** come Step Into, ma se il codice del rigo associato al breakpoint è una chiamata a una procedura, quest'ultima viene eseguita per intero.
- **Step Out:** attiva solo in modalità “debug”. Se il breakpoint è posto all'interno di una procedura, questa viene completata ed il programma si ferma al rigo successivo a quello dove è avvenuta la chiamata.
- **Run to cursor:** è attivo solo in modalità “debug” quando il programma è in pausa oppure fermo su un breakpoint. In questo caso è possibile posizionare il cursore del mouse su un punto qualsiasi del programma, fare clic su quel punto e quindi agire sul comando *Run to cursor* per eseguire tut-



ta la porzione di codice compresa tra la linea in cui il programma si è fermato e il cursore.

- **Jump to cursor:** quando siamo in modalità “debug” e fermi su un breakpoint, posizionando il cursore su un altro punto del programma e poi cliccando su questo comando, l'esecuzione salta direttamente a quel punto. Tutte le istruzioni precedenti ad esso non vengono eseguite.
- **Toggle Breakpoint:** inserisce o disinserisce un breakpoint sulla linea di codice in cui si trova il cursore: si ottiene lo stesso risultato facendo doppio clic all'inizio del rigo desiderato.

- **Add to Watchlist:** permette di aggiungere a una tabella il nome di una variabile (di sistema o creata da noi). Quando siamo fermi su un breakpoint, possiamo osservare il contenuto di tutte le variabili della Watchlist nella finestra “Watch”, attivabile dal menu “View”.
- **Remove from Watchlist:** permette di rimuovere una variabile dalla finestra di Watch. Prima si

Lo troviamo nella scheda “Browser” del riquadro “Project” sulla sinistra. Espandiamo il gruppo “Platform Events” e andiamo a cercare l’evento in questione (Fig. 8). Notiamo come basti posizionare il puntatore del mouse sopra il nome dell’evento per far apparire un riquadro “pop-up” che lo descrive in dettaglio. Facciamo doppio clic sull’evento “On_Sys_Init” in modo da far comparire automa-

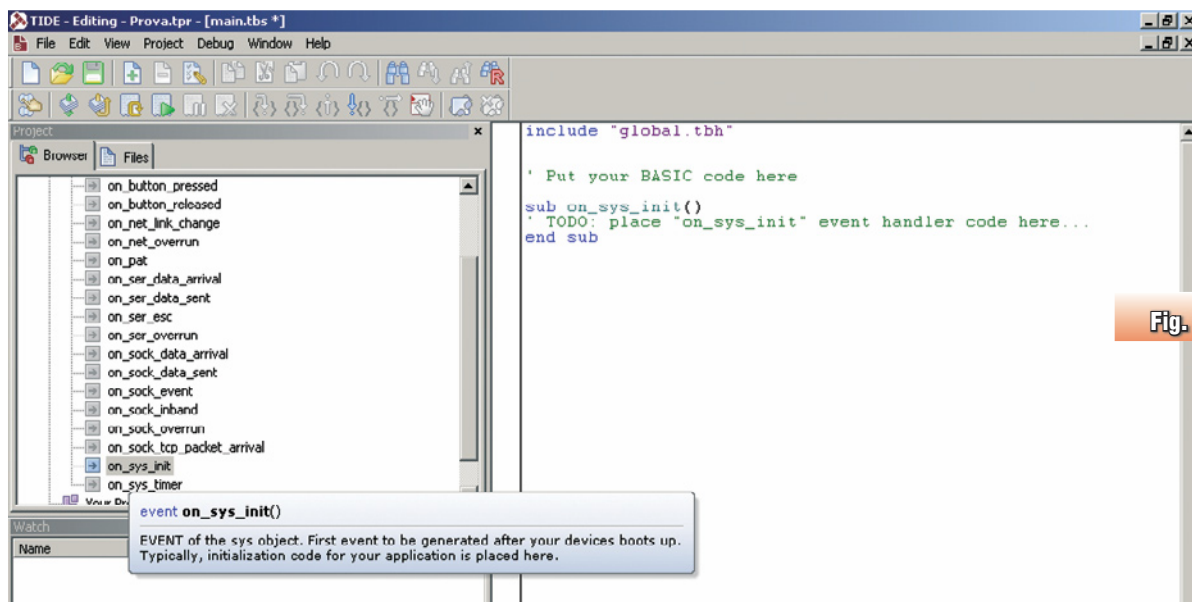


Fig. 8

seleziona la variabile da rimuovere e poi si clicca su questo comando.

Step 5: il nostro primo programma

Proviamo ora a scrivere un semplice programma che accende in sequenza i led della scheda “Interface Leds”. La prima cosa da dichiarare è l’evento di avvio, ossia il punto esatto dove il modulo EM1000 deve andare a leggere la prima riga di codice quando viene alimentato o riavviato: si tratta dell’evento di sistema “On_Sys_Init”.

Listato 1

```
main:
  dim pausa as word
  dim led as byte

  for led=0 to 7
    io.num=24+led
    io.enabled=YES
    io.state=low

    pausa=sys.timercount+1
    do
      loop until sys.timercount>pausa

    io.state=HIGH

    pausa=sys.timercount+1
    do
      loop until sys.timercount>pausa
    next led

  goto main
```

ticamente l’intestazione giusta (Nome_Evento / Fine_SUB) all’interno della finestra dove andremo a scrivere il codice (Fig. 8, area di “editing” a destra). Quanto appena visto vale per tutti gli eventi e agevola la scrittura del codice, soprattutto perché consente di evitare gli errori di sintassi. A questo punto occorre scrivere nella finestra di editing il codice sorgente (che poi analizzeremo in dettaglio) riportato in Listato 1: esso deve essere compreso tra le linee di comando “sub on_sys_init()” e “end sub”, esattamente dove è presente il commento in verde “TODO:...”.

Per prima cosa è necessario dichiarare tutte le variabili che useremo nel codice, assegnando ad esse un nome adeguato e specificando il tipo di variabile. Nell’esempio, usiamo la variabile “Led”, che avrà un valore compreso tra 0 e 7 e che possiamo quindi dichiarare come “BYTE”, e la variabile “Pausa”, che conterrà numeri a 16 bit (timer di sistema) e va dichiarata come “WORD”.

Immediatamente dopo la dichiarazione delle variabili, troviamo il ciclo “For Led <codice> Next Led” che ripete otto volte quanto scritto nel campo <codice>, assegnando ogni volta un valore diverso (da 0 a 7 appunto) alla variabile “Led”.

Ciò che vogliamo fare nel ciclo è piuttosto sem-

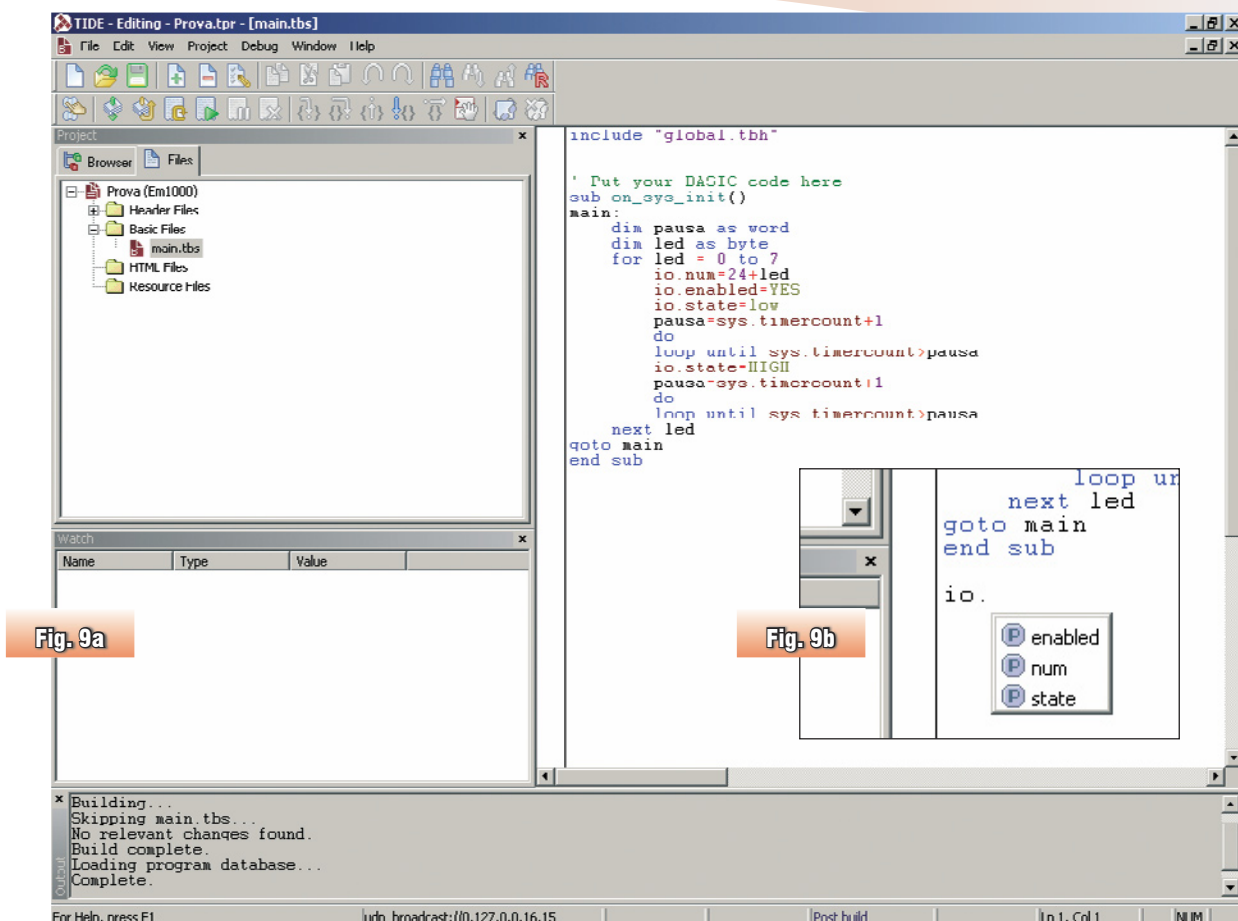


Fig. 9a

Fig. 9b

plíce: accendiamo un Led, attendiamo un secondo, spegniamo il led ed accendiamo il successivo dopo aver atteso un altro secondo. I led della scheda "Interface Leds" sono collegati direttamente al modulo EM1000 e precisamente agli I/O aventi indirizzo base 24 quindi da 24 a 31. Per poter utilizzare gli I/O, però, dobbiamo prima dichiarare se devono essere configurati come ingressi o uscite, quindi abilitarli e infine portarli ad uno stato a noi noto ("Low" o "High").

Quanto appena descritto viene effettuato con le istruzioni "io.num", "io.enabled" e "io.state".

Proviamo ora a scrivere in un punto qualsiasi della finestra di editor "io" seguito dal punto: vediamo che anche in questo caso appare un riquadro pop-up (Fig. 9b) ad elencare tutte le proprietà che possiamo assegnare all'oggetto "input-output". Utilizziamo i tasti cursore "freccia su" e "freccia giù" per vedere ancora comparire in pop-up la descrizione dettagliata della proprietà.

Portiamoci per esempio su "enabled" e premiamo il tasto "Invio". Scriviamo poi "=" e ancora abbiamo il pop-up che ci aiuta nella scelta di quello che possiamo scrivere dopo, cioè "yes" oppure "no" (Fig. 10): come potete vedere, il software aiuta molto durante la scrittura del sorgente, ma ora

cancelliamo quanto scritto per prova e torniamo al nostro codice. Il led deve rimanere acceso per un secondo, scaduto il quale si deve spegnere quindi, trascorso un altro secondo, si deve accendere il led successivo. La pausa viene generata sfruttando il timer di sistema (*sys.timercount*) che conta autonomamente intervalli di mezzo secondo e può essere interrogato in qualunque momento per conoscere lo stato del conteggio. Leggiamo il valore corrente, aggiungiamo 1 e salviamo il risultato nella variabile *pausa*, definendo così l'elemento di confronto che determina lo scadere dell'intervallo di un secondo. Dopo aver acceso il primo led con il comando "io.state=low", viene eseguito un ciclo "Do-Loop" in cui il programma continua a leggere lo stato del timer: quando il valore del timer supera quello salvato nella variabile *pausa*, il ciclo Do-Loop termina ed il programma esegue l'istruzione "io.state=HIGH", che spegne il led. Un nuovo ciclo Do-Loop determina una pausa di un secondo, allo scadere della quale il programma incrementa la variabile *led* e poi riprende il ciclo "for-next" dall'inizio.

Notate un altro particolare: il codice sorgente viene visualizzato con colori differenti a seconda del significato delle varie parole chiave che lo com-

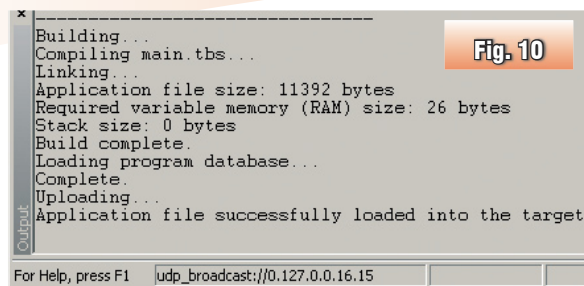


Fig. 10

pongono. Le variabili (pausa, led) e le etichette (main:) vengono visualizzate in nero, gli operatori logici e matematici (=, +, -...) in rosso, le istruzioni Tibbo Basic (dim, for... next, do... loop, eccetera) in blu, gli eventi di sistema (io.num, sys.timercount...) in bordeaux, le costanti numeriche in viola e i valori delle proprietà (low, high, yes...) in verde, colore quest'ultimo utilizzato anche per i commenti. Da quanto appena esposto si deduce che, con un po' di pratica, è sufficiente un colpo d'occhio sia per distinguere i vari elementi che costituiscono un programma sia per comprendere

se è presente un errore in una linea che presenta colori differenti da quelli previsti. Ma torniamo al nostro programma. Clicchiamo sul tasto "Build" e osserviamo il risultato della compilazione (Fig. 9a, area grigia in basso): il compilatore ci comunica di avere compilato e assemblato correttamente i file del firmware. Clicchiamo sul tasto "Build and Upload" e osserviamo il risultato in Fig. 10: come già visto in precedenza, la sezione grigia in basso della finestra di lavoro principale comunica che il codice oggetto è stato trasferito correttamente sulla scheda target (EM1000). Ora possiamo passare all'esecuzione del programma semplicemente cliccando sul tasto "Run": osserviamo gli otto led che si accendono e spengono in sequenza. In questo momento il programma è attivo e, insieme ad esso, è attivo anche l'emulatore di sistema, illustrato in Fig. 11: possiamo pertanto iniziare ad interagire con tutti i comandi già descritti ed eseguire porzioni di codice prestabilite. In una sola parola, possiamo iniziare a "debugare" il programma. Per prima

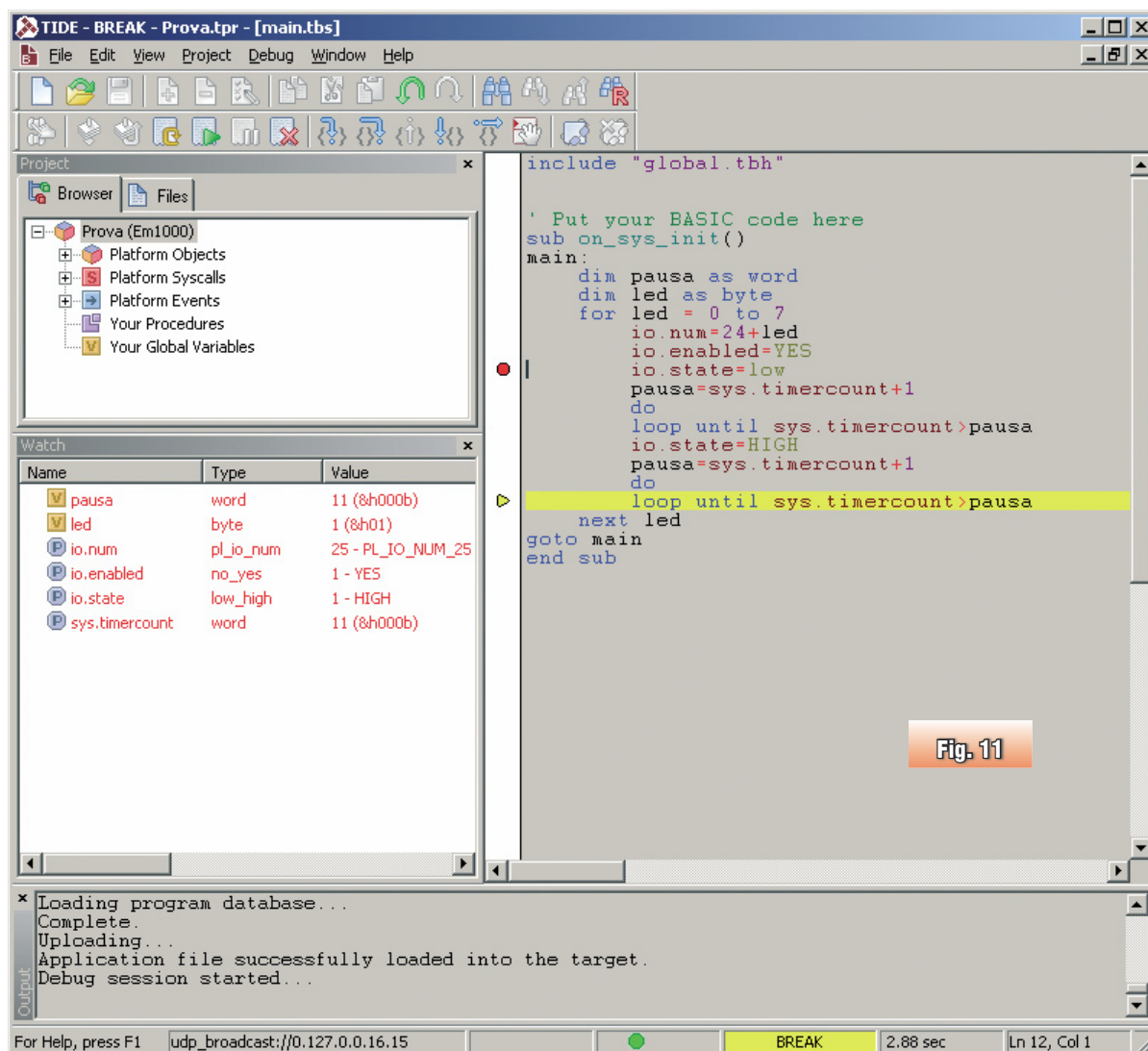
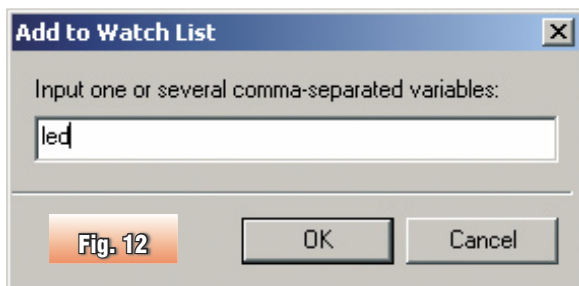


Fig. 11



cosa sospendiamo temporaneamente l'esecuzione del programma mettendolo in pausa tramite l'apposita icona: nella finestra principale appare un rigo di codice evidenziato in giallo, a indicare che il programma si è fermato in quel punto; il riavvio riprenderà con l'esecuzione di quella stringa di codice. La posizione è chiaramente casuale perché dipende dall'istante in cui abbiamo cliccato sul tasto "Pause". Proviamo ora a cliccare più volte sul tasto "Step Into": osserviamo che l'esecuzione del programma avviene istruzione per istruzione e che le uscite (quindi lo stato dei led) cambiano in funzione delle linee di programma. A questo punto clicchiamo sul tasto "Add to Watch List" e aggiungiamo alla lista (per ora vuota) la variabile *Led*, della quale vogliamo osservare il contenuto (Fig. 12). Dopo avere scritto il testo "led" nella linea di input e cliccato su "OK", la finestra "Watch" (a sinistra in basso, nella finestra di lavoro principale di Fig. 11) viene aggiornata: ora, infatti, contiene una riga di testo che indica l'elemento sotto controllo (la variabile *led*), il tipo di elemento (byte) ed il suo valore (in decimale, esadecimale oppure testo). Come potete notare dalla Fig. 11, è possibile aggiungere alla Watch List sia delle variabili che degli oggetti, degli eventi o delle proprietà. Continuiamo l'esecuzione passo-passo con il tasto "Step Into", osservando come cambia il contenuto della Watch-List: in questo caso il programma, che utilizza due sole variabili, è molto semplice per cui non si riesce ad apprezzare immediatamente la comodità offerta dallo strumento di "debug". Sviluppando programmi via via sempre più complessi, però, vi accorgete presto di come lo strumento "Watch" sia essenziale per scoprire gli inevitabili errori. Un altro strumento estremamente utile è il "Breakpoint" o punto di interruzione. Portiamo ad esempio il cursore sulla sinistra di un rigo qualunque del programma, come ad esempio su "*io.state=low*" e facciamo un clic: vediamo comparire un pallino rosso in corrispondenza di tale linea. Ciò significa che abbiamo istruito l'emulatore per sospendere temporaneamente l'esecuzione del programma quando questi giunge alla riga di codice

evidenziata dal pallino rosso. Clicchiamo su "Run" e vediamo che il programma continua l'esecuzione per fermarsi poi proprio su quel breakpoint: in quel rigo chiediamo alla porta di I/O corrente di portarsi nello stato basso (low) per accendere il led. Quindi, secondo la logica, prima di quell'istruzione l'uscita dovrebbe essere alta (high): verificiamolo aggiungendo alla Watch-List la proprietà "*io.state*" (Fig. 11). A questo punto abbiamo la conferma che l'uscita, prima del breakpoint, è a livello logico alto (led spento). Avanziamo di uno step l'esecuzione e verificiamo i cambiamenti sul led e sulla Watch-List. Bene, ora che tutto funziona come desideriamo, interrompiamo l'esecuzione (tasto "Stop") e disabilitiamo la modalità "debug" (menu "Project-Setting"). Clicchiamo su "Build and Upload" e ritrasferiamo il codice nel modulo EM1000 secondo le modalità già viste; da questo momento il nostro Ethernet Server è in grado di funzionare autonomamente e pertanto può essere scollegato dall'ambiente di sviluppo.

Note conclusive

Giunti a questo punto, abbiamo visto, mediante un esempio molto semplice, come si utilizzano oggetti ed eventi del linguaggio Tibbo Basic e abbiamo preso coscienza delle potenzialità dell'ambiente di sviluppo. Nelle puntate successive analizzeremo dettagliatamente alcuni programmi più complessi, basati sugli esempi applicativi che la stessa Tibbo mette a disposizione sul sito Internet.

In particolare vedremo

- come controllare via Ethernet delle linee di I/O,
 - come veicolare del traffico dati da seriale a Ethernet e viceversa,
 - come realizzare un Web Server e un Mail Server.
- A conclusione del corso, poi, realizzeremo un progetto pratico che farà uso di quanto spiegato nei vari esempi pratici.

