

VISUALIZZATORE LCD A 16 MESSAGGI SU EEPROM

Un utilissimo dispositivo per scrivere fino a 16 frasi di 32 caratteri ciascuna su un display a cristalli liquidi e per memorizzarle una memoria di tipo non volatile

di Andrea Sbrana

Dopo la pubblicazione del campanello elettronico apparsa nel lontano 1992, sono stati molti i lettori che ci hanno interpellato per avere un dispositivo simile, ma con caratteristiche decisamente migliori.

Uno dei limiti della versione precedente, infatti, era il non poter modificare la scritta da visualizzare una volta

programmata all'interno del chip. Altro svantaggio era l'impiego di un display a Led molto difficile da reperire sul mercato e con un numero di caratteri gestibili non superiore a otto.

Il campanello elettronico, però, è stato il primo progetto presentato nel nostro Paese con un microprocessore della famiglia dei PIC.

Da allora è passato molto tempo e in questi due anni abbiamo affinato le nostre tecniche progettuali, sviluppando via via circuiti sempre più complessi, ma allo stesso tempo facili da realizzare, sicuramente funzionanti e, non ultimo, con un costo limitato.

Proprio partendo dal nostro primo circuito con i PIC, abbiamo deciso di apportare delle modifiche che permettono di ampliare le possibilità di utilizzo, non limitandole a un semplice campanello elettronico.

Abbiamo cercato, quindi, di venire incontro alle esigenze di tutti, realizzando un circuito non molto più complesso del precedente a livello hardware, ma sicuramente più flessibile a livello software.

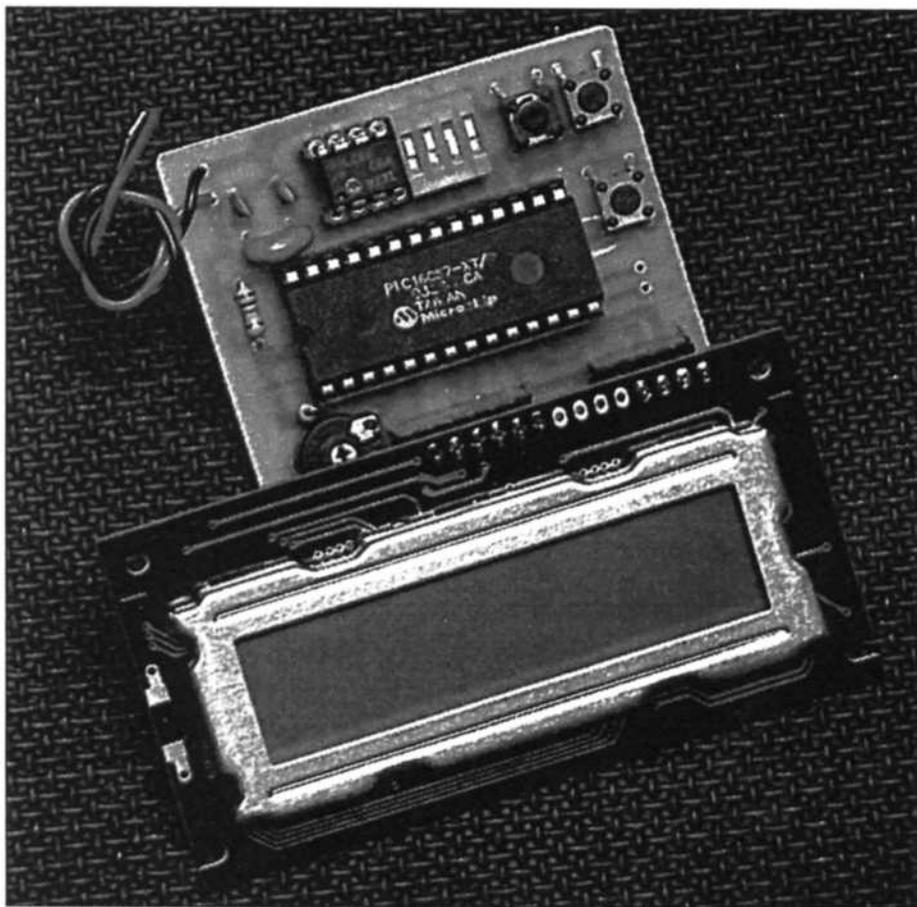
Dato che, come accennato, il display precedente creava alcuni problemi di reperibilità e, comunque aveva un costo che non si poteva sottovalutare, il nostro primo obiettivo è stato quello di reperire facilmente sul mercato tutti i componenti: abbiamo trovato un display a cristalli liquidi da due righe e sedici colonne, in grado di visualizzare contemporaneamente 32 caratteri, senza doverli far scorrere.

Abbiamo inoltre "riarrangiato" il programma svincolando il software contenuto nel chip dall'area dei dati da visualizzare, impiegando una memoria di tipo EEPROM, che abbiamo già impiegato diverse volte in circuiti simili.

In sostanza, queste memorie possono riprogrammate praticamente all'infinito e in più mantengono i dati anche togliendo loro la tensione di alimentazione.

Lavorando con un display a cristalli liquidi, infine, siamo riusciti a ridurre l'assorbimento di tensione che è stato portato dai precedenti 150 mA agli attuali 5 mA.

Come se non bastasse, abbiamo portato il numero dei messaggi memorizzabili a 16, permettendo di richiamarli semplicemente inviando il codice binario di indirizzamento.



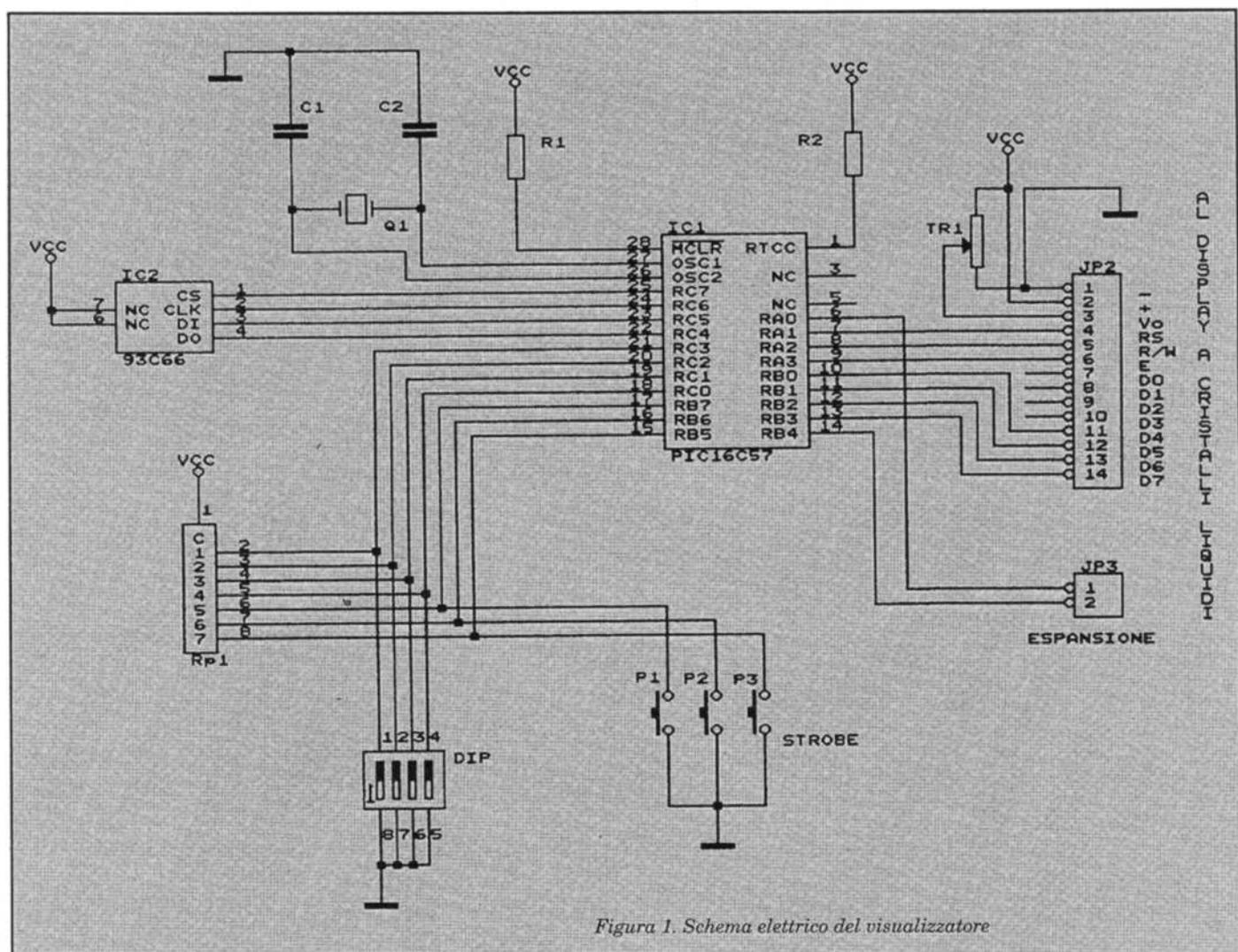


Figura 1. Schema elettrico del visualizzatore

Come funziona

In Figura 1 è visibile lo schema elettrico del circuito: il solo chip IC1 gestisce sia il display che la memoria seriale. Come si può facilmente intuire, esistono pochissimi altri componenti e di questi nessuno è attivo, a garanzia di un circuito che difficilmente potrà guastarsi.

Ma veniamo ora all'analisi della Figura 2 che evidenzia il diagramma di flusso del programma che è stato inserito nel microprocessore.

Appena viene fornita alimentazione, che deve necessariamente essere di 5 V continui, il controller si mette in attesa per mezzo secondo, per permettere al display di inicializzarsi, poi invia al display stesso un reset software che cancella tutta la finestra di scrittura in attesa di un primo messaggio.

A questo punto, il controller attende un segnale negativo sul pin 15, ovvero dal pulsante P3. Nel momento in cui questo segnale giunge, IC1 va a leggere

l'indirizzo impostato con i dip-switch, cancella nuovamente il display e va a leggere i 32 caratteri memorizzati nella EEPROM a partire dall'indirizzo specificato. La fase successiva è poi quella della visualizzazione sul display stesso.

Fatto ciò, il chip testa nuovamente lo stato del pulsante, che potrebbe ancora essere premuto, ed eventualmente attende il suo rilascio.

Non appena ciò avviene il ciclo adesso mostrato si ripete a una nuova pressione del pulsante P3.

In Figura 3 possiamo vedere il diagramma a blocchi per quel che riguarda la fase di lettura della EEPROM: inizialmente vengono settati due contatti. Poiché queste memorie sono di tipo "intelligente", accettano diversi comandi che ne semplificano l'utilizzo al progettista di firmware: infatti, per leggere una locazione di memoria, è sufficiente inviare un comando di lettura, seguito dall'indirizzo della locazione di memoria da leggere.

Quest'ultima potrà essere lunga 8 oppure 16 bit, a seconda di come viene posizionato il pin 6 della memoria. Infatti, se il pin 6 viene collegato al positivo, la locazione sarà di 16 bit, viceversa di 8.

Per motivi che analizzeremo nel proseguo dell'articolo, ma che potremmo indicare semplicemente per questioni di velocità, abbiamo optato per la soluzione dei 16 bit. Dato che la memoria impiegata è di 4096 bit, avremo un indirizzo di partenza per le 356 locazioni da 16 bit ($356 \cdot 16 = 4096$).

Questo indirizzo, ovviamente composto da 8 bit (2 elevato all'ottava dà proprio 256 combinazioni), viene ricavato sia dal posizionamento dei dip-switch che dal contatore software interno al programma.

Per rendere chiari questi concetti analizziamo alcuni esempi: è palese che per ogni carattere da visualizzare avremo bisogno di un byte della EEPROM.

Per visualizzare due righe da 16 caratteri ciascuna, abbiamo bisogno quindi di 32 byte, ovvero di 16 WORD, dove per WORD si intende una coppia di byte.

Quindi, le 256 WORD disponibili nella EEPROM verranno divise per 16, offrendo i 16 indirizzi necessari per la particolare applicazione.

Questi 16 indirizzi sono selezionabili attraverso i dip-switch. Matematicamente parlando, avremo dall'indirizzo 00h all'indirizzo 0Fh le prime 16 locazioni da 16 bit, dall'indirizzo 10h le seconde sedici locazioni e così via fino alle ultime che partiranno dalla F0h per arrivare a FFh.

Prendiamo allora ad esempio la quarta locazione, che andrà da 30h a 3Fh.

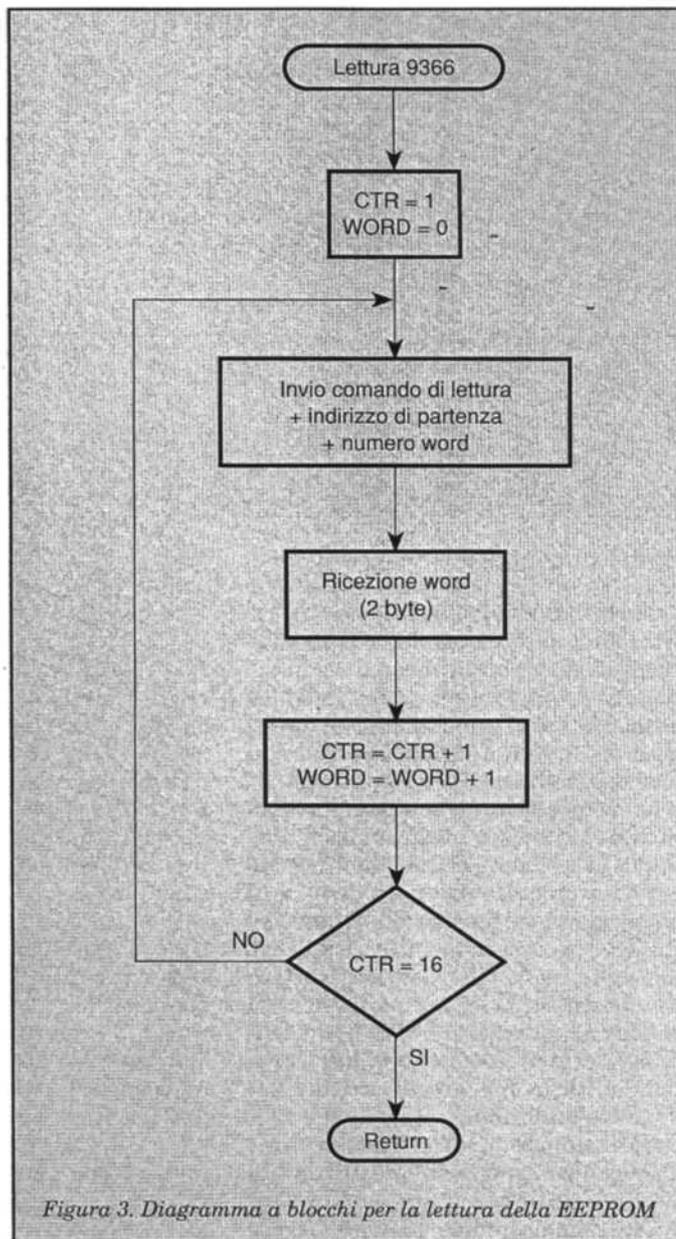
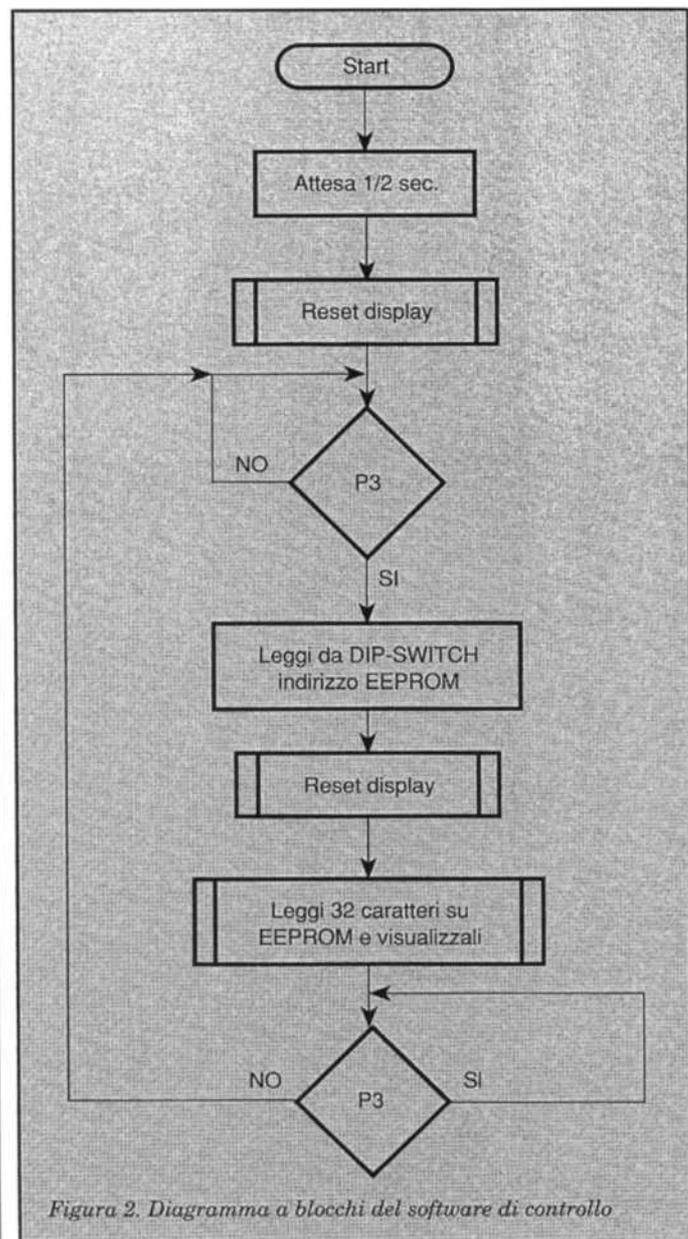
La parte più significativa dell'indirizzo è quella settata con i dip-switch, quindi, in questo caso il "3". Per vederla in binario, l'indirizzo 30h sarà 00110000, mentre l'indirizzo finale 3Fh sarà 00111111.

Notiamo, quindi che i primi quattro bit (più significativi) non cambiano e sono sempre a 0011. Variando, quindi, solo i quattro bit meno significativi, siamo in grado di leggere tutte e sedici le WORD che poi comporranno una frase di 32 caratteri sul display.

Tornando alla Figura 3, notiamo la presenza di due contatori, uno per incrementare l'indirizzo e l'altro per controllare il numero di cicli (sedici nel nostro caso). Questi due contatori vengono inizializzati entrambi a 1.

Successivamente si invia alla EEPROM un comando composto da uno START BIT, da un codice di LETTURA EEPROM e dall'indirizzo completo della locazione da leggere, indirizzo composto per i primi quattro bit dal settaggio sui dip-switch, per gli altri quattro bit dal numero inserito nel contatore WORD.

Immediatamente dopo il comando di lettura, la EEPROM "risponde" inviando un treno di sedici bit, ovvero la WORD richiesta. Vengono così incrementati i due contatori e si controlla se siamo arrivati alla lettura dell'ultima WORD oppure no. In caso affermativo, si esce dalla routine, altrimenti si torna ad inviare il comando di lettura per la WORD successiva.



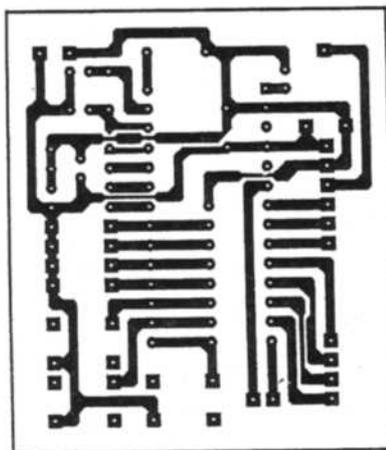


Figura 4. Circuito stampato scala 1:1

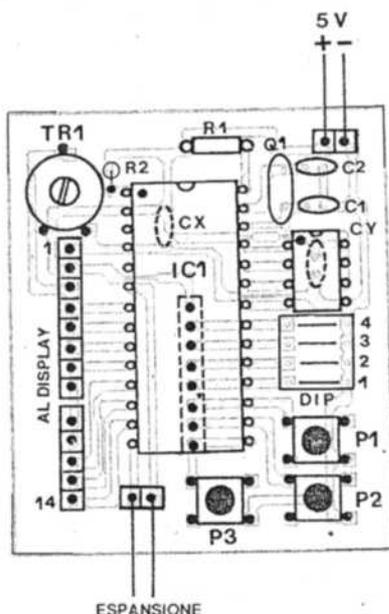


Figura 5. Disposizione dei componenti

Montaggio

La realizzazione di questo circuito non presenta particolari difficoltà, soprattutto se si tiene conto della traccia del circuito stampato di Figura 4 e della disposizione dei componenti proposta in Figura 5.

Proprio da questo schema, si evince che sono stati utilizzati due condensatori CX e CY, posti entrambi sotto i due integrati, che mancano nello schema elettrico. Questi due componenti, infatti, rappresentano due filtri per l'alimentazione che, come abbiamo avuto modo di dire, deve essere necessariamente di 5 V continui. Non abbiamo volutamente voluto inserire un regolatore di tensione, dato che conosciamo le esigenze degli sperimentatori.

Infatti, la tensione di 5 V è la più classica ed è sicuramente disponibile in qualsiasi tipo di circuito.

Nel caso contrario, potrete utilizzare il classico regolatore 7805 che supporta brillantemente la corrente richiesta dal circuito.

L'unica avvertenza da seguire in questa applicazione è il cablaggio dei fili dalla basetta al display, dato che i chip di controllo dello stesso si sono dimostrati particolarmente sensibili alle scariche elettrostatiche.

Inizializzazione

Le operazioni iniziali di inizializzazione dei messaggi avvengono tramite un programmatore di EEPROM come

ELENCO COMPONENTI

Semiconduttori

IC1: PIC16C57 programmato

(c/o 0337/259730)

IC2: 93C66 o 93LC66 Microchip

Resistori

R1, R2: 100 kΩ

RP1: Rete resistiva 10 kΩ, 1+7

TR1: Trimmer 4,7 kΩ

Condensatori

C1, C2: 56 pF

CX, CY: 100 nF (vedi testo)

Varie

Q1: Oscillatore ceramico:

3,58 MHz

1 Display a cristalli liquidi, 2 righe, 16 colonne

quello proposto sui numeri di febbraio e marzo del '94. È anche possibile richiedere direttamente all'autore, oltre al PIC già programmato, una EEPROM con memorizzati i messaggi, telefonando allo 0337/259730.

Anticipiamo fin d'ora che sul prossimo numero di Progetto presenteremo un programmatore dedicato, sviluppato nei nostri laboratori per sfruttare appieno le caratteristiche del nostro progetto, da collegare direttamente al computer.

Per quanto riguarda le applicazioni, invece, non c'è che l'imbarazzo della scelta: segnalazioni di pericolo, avvisi per la clientela, rubrica telefonica, check-panel per auto o per impianti anti-furto e tante altre, limitate soltanto dalla vostra fantasia. ■

Sono in edicola a sole L. 20.000 cad. i 2 preziosi volumi contenenti indirizzi, telefoni e frequenze di tutte le migliaia di emittenti radiotelevisive che operano in Italia.

Complessivamente si tratta di 600 pagine di dati e utilissime informazioni indispensabili per conoscere in modo capillare e completo questo affascinante settore.

