

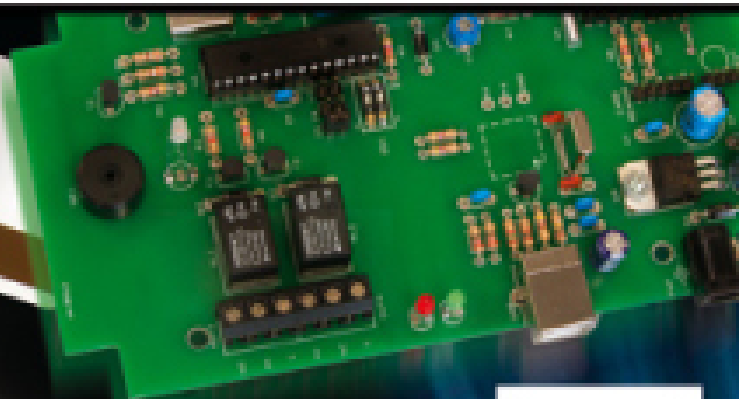
Electronica nnovativa In

€ 4,50

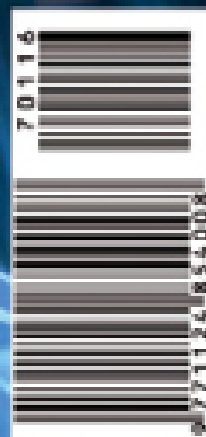
Mensile di elettronica applicata, attualità scientifica, novità tecnologiche.

116

Lettore di badge con porta USB



Braccio Robotizzato con controllo seriale



Induttanzimetro e capacimetro digitale

Nuove funzioni per l'impianto domestico Velbus

Serratura a combinazione con encoder e display

Mambo, localizzatore portatile programmabile

*** **ESCLUSIVO** ***
CORSO LCD
(Liquid Crystal Display)

Serratura a combinazione con encoder e display

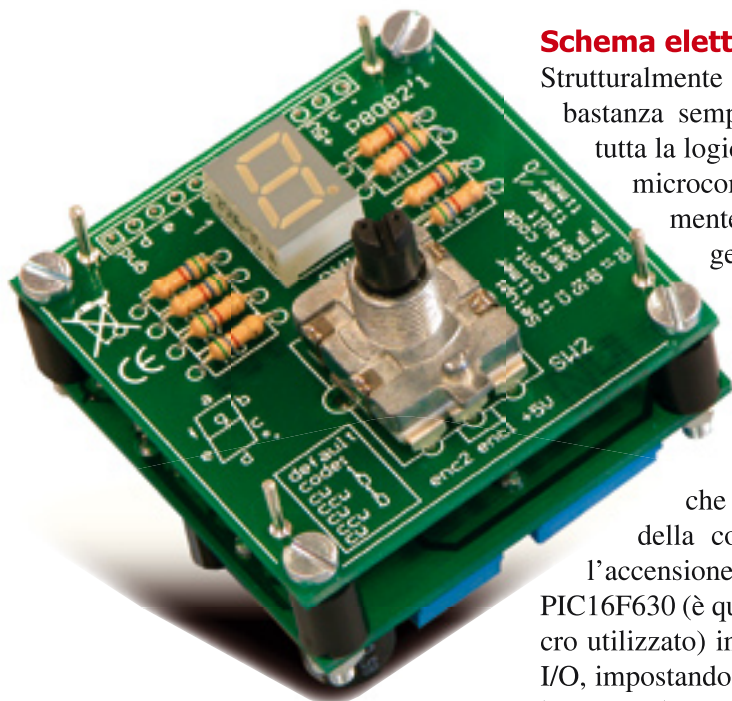
di Boris Landoni

Consente l'attivazione di qualsiasi elettroserratura o dispositivo di sicurezza con una procedura simile a quella delle casseforti. Le quattro cifre della combinazione vengono selezionate con un encoder rotativo e visualizzate da un display. Il relé di uscita si attiva quando la sequenza viene eseguita correttamente.



Tutti noi abbiamo visto almeno un film nel quale il ladro di turno apre la cassaforte della banca o della gioielleria ruotando abilmente la manopola della combinazione: lo stetoscopio all'orecchio, per sentire cosa accade all'interno, uno scatto a destra, uno a sinistra, e così via sino all'apertura della pesante porta blindata con il premio finale in gioielli o mazzette di banconote. Oggi, nell'era dell'elettronica, anche il cinema si è adeguato proponendoci fantomatici eroi che violano computer con password complicatissime, sensori biometrici ingannati da copie di impronte digitali realizzate con

pellicole applicate ai polpastrelli o da lenti a contatto con la struttura dell'iride del direttore della banca o del supercarcere di turno e quant'altro l'inafferrabile Arsénio Lupin mai si sarebbe immaginato. Anche le tecniche di protezione di locali e le casseforti si sono adeguate all'evoluzione tecnologica diventando sempre più zeppe di elettronica. Passateci, però, un pizzico di nostalgia: volete mettere il fascino della tradizionale serratura a combinazione con le fredde cifre di un display e i tasti insignificanti di una tastiera numerica oppure con il fascino di uno scanner che analizza l'impronta del dito? ➤



Schema elettrico

Strutturalmente il dispositivo è abbastanza semplice, anche perché tutta la logica è contenuta in un microcontrollore opportunamente programmato per gestire tutte le funzioni, controllare il visualizzatore a led a sette segmenti e leggere il commutatore (encoder) rotativo che emula la manopola della combinazione. Dopo l'accensione del circuito, il PIC16F630 (è questa la sigla del micro utilizzato) inizializza le linee di I/O, impostando RA1 ed RA5 come input per leggere SW2, RA3 ancora come ingresso per controllare lo stato del pulsante di programmazione (SW1) quindi RA0 da uscita riservata all'attivazione del relé ed RA2, RA4, RC0, RC1, RC2, RC3, RC4, RC5 ancora come output, destinati alla gestione del display. La linea RC0 si occupa dell'accensione del punto decimale mentre gli altri segmenti sono controllati dalle linee citate in precedenza; tutte operano in modalità "sink", nel senso che vengono attivate a zero logico ed assorbono corrente invece di erogarne; questo perché il display è del tipo ad anodo comune.

Completata l'inizializzazione degli I/O si entra nel main-program nel quale il PIC testa ciclicamente la condizione del pulsante SW1 e delle linee corrispondenti al commutatore rotativo; nel frattempo, comanda il display al fine di far apparire un'animazione rappresentata da due segmenti che scorrono lungo il bordo del visualizzatore, come una sorta di piccolo serpente che percorre i segmenti esterni (resta spento il solo tratto centrale). Quando viene rilevato un cambiamento, in funzione della natura di quest'ultimo, il programma principale lancia la routine relativa e sospende l'animazione,

proponendo sul display il messaggio corrispondente.

Analizziamo inizialmente il normale utilizzo, quello, cioè, in cui il circuito deve confrontare la combinazione introdotta con quella memorizzata; a tale riguardo dobbiamo precisare che dopo la prima accensione, la serratura è predisposta per riconoscere 1010, perché questa è la combinazione memorizzata in fase di programmazione del microcontrollore. Per comprendere il funzionamento del dispositivo nel suo complesso, compresa la configurazione (che verrà esposta più avanti) bisogna capire come funziona l'encoder o commutatore rotativo e considerare che, per rendere la nostra serratura elettronica il più possibile simile a quella tradizionale (meccanica), abbiamo realizzato il circuito in maniera che la combinazione corrisponda e venga definita da una serie di scatti in una direzione o nell'altra.

Ci spieghiamo meglio: nella serratura il **numero della combinazione è quello su cui ci si ferma prima di ruotare la manopola all'indietro**; quello successivo corrisponde quindi alla cifra sulla quale il cursore viene fermato dopo averlo nuovamente ruotato in senso orario; il numero viene rilevato nel momento in cui la manopola viene riportata indietro. Ad esempio, se i primi numeri della combinazione sono 1 e 5, dopo aver ruotato la manopola in senso orario fino alla comparsa del numero 1, per rilevare questa cifra e poi andare sul 5 bisogna tornare indietro (in quel momento viene rilevato il numero 1) di almeno uno scatto e poi ruotare la manopola in senso orario fino a quando il display non indica il 5. Bisogna poi tornare indietro: in quel preciso istante il sistema acquisisce anche il numero 5.

Chiarito questo concetto, ci siamo posti il problema di come far capire al microcontrollore il verso della rotazione e il numero sul quale fermarsi; abbiamo trovato la soluzione

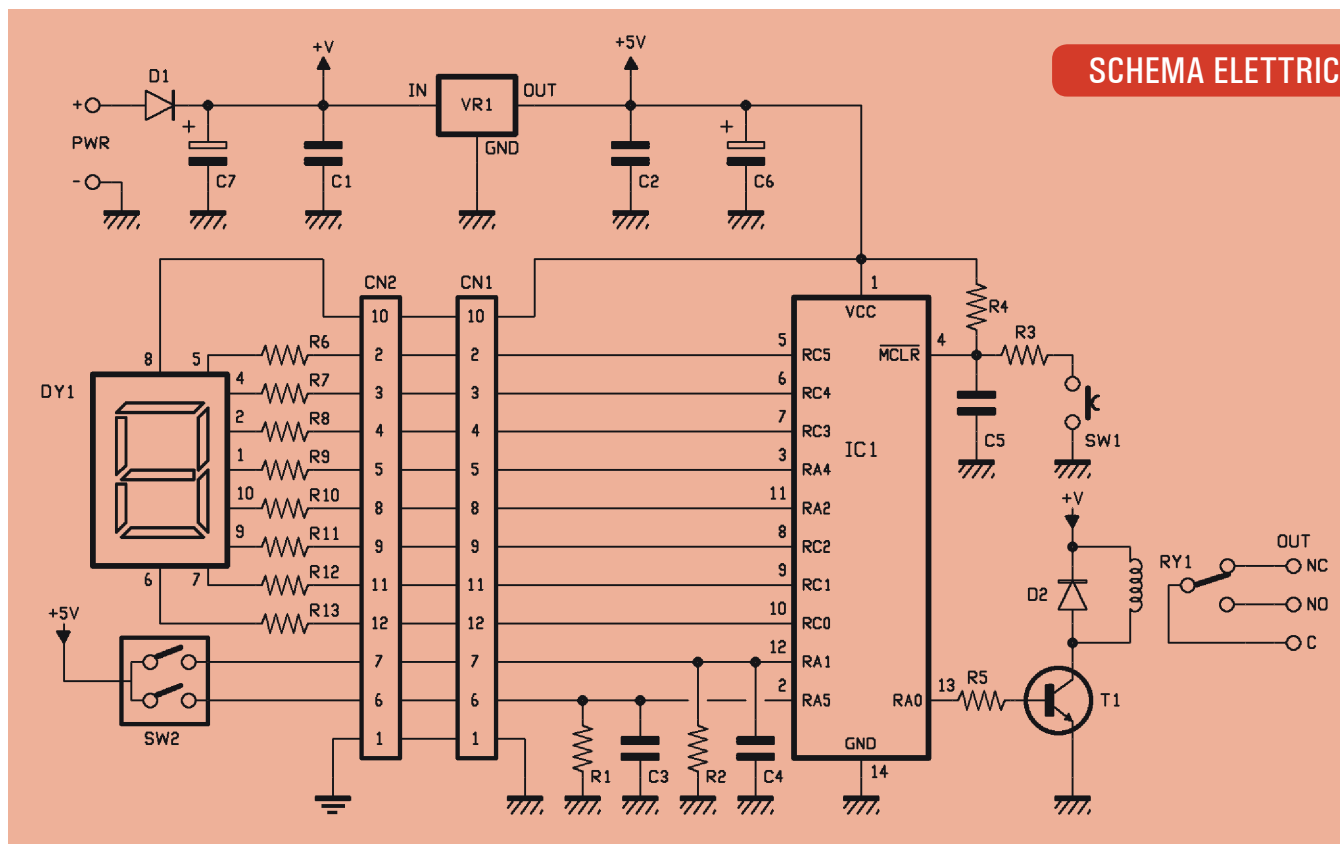
Certo anche la moderna tecnologia ha un suo fascino ma, almeno per un volta, abbiamo pensato di farvi vivere un momento di revival proponendo la costruzione di una serratura a combinazione da cassaforte, rivisitata in chiave moderna, con l'ausilio dell'elettronica.

Si tratta di un circuito elettronico che permette di comandare un relé e con esso delle elettroserrature o altri dispositivi, quando si compone, con la manopola di cui è provvisto, l'esatta combinazione numerica di quattro cifre.

La combinazione può essere impostata con estrema facilità, mediante una procedura rapida e assistita da un solo display.

Il relé del circuito può essere attivato in modo sia bistabile che impulsivo; in quest'ultimo caso, con un'apposita procedura di programmazione, è possibile definire il tempo di attivazione.

La serratura è quindi utilizzabile non solo per azionare elettroserrature, ma anche per produrre il comando di apertura di un cancello motorizzato, quello di disattivazione o attivazione di un impianto antifurto, ma anche per abilitare o disabilitare l'accesso a un Personal Computer.



nell'adozione di un encoder rotativo decimale, che è sostanzialmente un commutatore a due vie che utilizza una serie di interruttori che hanno un punto in comune, alimentato, nel nostro caso, a 5 volt. Il commutatore è costruito in modo che, ruotando il perno di comando, i contatti vengano chiusi e riaperti sequenzialmente, mai insieme; in altre parole, uno apre e chiude prima dell'altro. Il movimento dei rispettivi contatti si ottiene facendo alzare le lamelle corrispondenti mediante delle sfere o dei denti di sega, che esse incontrano durante la rotazione del perno centrale di comando; il sollevamento è però sfalsato in modo che l'ordine di apertura e chiusura dipenda dal verso di rotazione. Se enc-1 ed enc-2 sono i contatti di uscita dei rispettivi interruttori, possiamo affermare che ruotando in senso orario il perno, enc-1 commuta prima di enc-2, mentre in senso antiorario accade il contrario, cioè enc-1 commuta dopo enc-2. Fermando il perno in una posizione ambigua, ovvero a

cavallo tra due scatti, i due contatti sono chiusi contemporaneamente. Bene, a questo punto capite sicuramente come, leggendo enc-1 ed enc-2 con le linee RA1 ed RA5 (sulle quali sono inserite reti R/C anti-rimbalzo allo scopo di evitare false commutazioni) il microcontrollore possa determinare quale sia il verso di rotazione della manopola e identificare il numero relativo alla posizione in cui l'encoder si ferma; tale cifra viene memorizzata in RAM (e visualizzato dal display) in attesa che l'utente inverta il senso di rotazione e torni al punto di partenza. Nella sequenza successiva il programma verifica la successione degli scatti e memorizza il numero sul quale la manopola si è fermata. Insomma, la prima cifra corrisponde al numero di scatti contati da quando si ruota la manopola in un verso (il dato viene rilevato nel momento in cui si ruota indietro l'encoder). Quando si effettua la quarta inversione, il firmware confronta le quattro cifre acquisite e memorizzate in

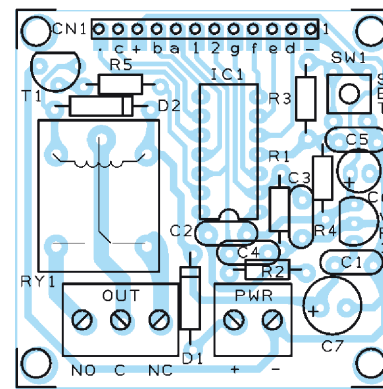
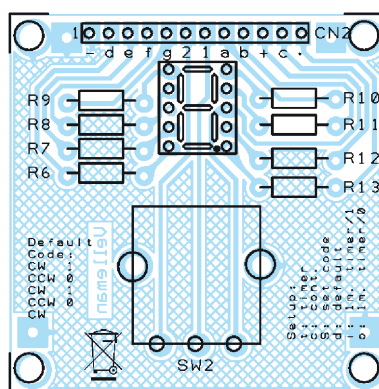
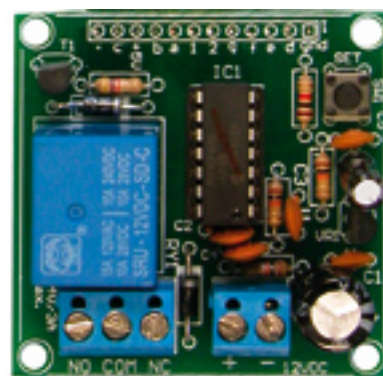
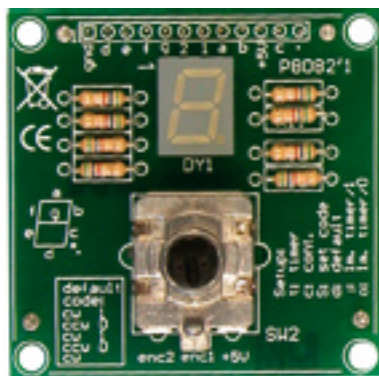
RAM con quelle, collocate in EEPROM durante la configurazione, corrispondenti alla combinazione valida: se i due numeri sono uguali, abilita il relé secondo la modalità impostata (ovvero quella predefinita in fabbrica, se è la prima volta che si usa la serratura o non è mai stata cambiata la configurazione dell'uscita). Questo fatto viene segnalato dal sistema con la visualizzazione sul display della lettera a (un 6 specchiato...) e con l'illuminazione del punto decimale. A riguardo, si noti che quest'ultimo indica lo stato del relé, quindi fin quando è illuminato, RY1 risulta eccitato; si osservi altresì che nel modo bistabile (a permanenza) il relé resta attivato fin quando non si dà al circuito il comando di sblocco, che corrisponde alla rotazione di un paio di scatti in un verso qualsiasi. In breve, dopo che il relé è scattato e il punto decimale è stato acceso, bisogna ruotare la manopola in un senso o nell'altro fino allo spegnimento del punto; fatto ciò, il micro segnala il ritorno in loop del main- ➤

ELENCO COMPONENTI:

R1: 10 kohm
 R2: 10 kohm
 R3: 1 kohm
 R4: 10 kohm
 R5: 4,7 kohm
 R6=R13: 560 ohm
 C1=C5: 100 nF multistrato
 C6: 10 µF 50 VL elettrolitico
 C7: 470 µF 25 VL elettrolitico
 D1, D2: 1N4007
 T1: BC547
 VR1: 78L05
 IC1: PIC16F630 (VK8082)
 RY1: Relè 12V singolo scambio (VR15M121C)
 SW1: Microswitch
 SW2: Encoder rotativo
 DY1: Display 7 segmenti anodo comune

Varie:

- Morsettiera 2 poli
- Morsettiera 3 poli
- Zoccolo 14+14
- Contenitore plastico
- Manopola per potenziometro
- Vite 3 MA 22 mm (4 pz.)
- Dado 3 MA (4 pz.)
- Circuito stampato



Tutti i componenti utilizzati sono ad inserzione, quindi il circuito non presenta particolari difficoltà durante la procedura di montaggio. Occorre prestare un po' di attenzione alla lunghezza dei fili con cui si collegano i due circuiti stampati.

program, in attesa di un comando con la solita animazione iniziale (segmenti luminosi che scorrono sul bordo del display).

Configurazione dell'uscita

L'impostazione predefinita del funzionamento del relé è bistabile; è possibile cambiarla entrando in caratterizzazione, fase che si avvia premendo SW1 fin quando il display non mostra illuminato il trattino centrale. Ricordiamo che questa operazione è possibile esclusivamente quando RY1 è a riposo (punto decimale del display spento).

Per fare in modo che, ad ogni confronto positivo, l'uscita lavori in maniera impulsiva, rilasciate il pulsante dopo circa un secondo dall'accensione del trattino: il display

mostra la cifra zero. A questo punto, ruotate la manopola della combinazione fino a far apparire il tempo di attivazione voluto per il relé, che può essere scelto fra 0 (corrispondente a 0,5 sec) e 9 secondi; ora, premendo e mantenendo premuto SW1, dovreste ruotare la manopola fino a veder comparire sul display la lettera *t*. Abbiamo così impostato il modo impulsivo; il display lampeggerà tre volte, insieme al punto decimale, indicando il tempo scelto, quindi tornerà a mostrare zero. Ciò segna la fine della fase di configurazione e dell'esecuzione della routine software corrispondente; la serratura torna nel funzionamento normale ed il microcontrollore, al solito, attende un evento (rotazione dell'encoder rotativo o nuova pressione di SW1).

Quando si effettua un confronto con esito positivo, il relé si attiva e il display mostra il tempo che manca al rilascio, sempre accompagnato dal punto decimale, che ne segnala l'attivazione; scaduto l'intervallo previsto, viene mostrato zero e il decimal point si spegne.

In ogni momento (a patto che RY1 sia a riposo) è possibile modificare il tempo di attivazione o tornare alla modalità bistabile; la procedura è analoga a quella già vista, nel senso che, ancora una volta, bisogna premere SW1 e attendere l'accensione del trattino. Ora il display mostra, lampeggiando tre volte, il tempo attualmente impostato; per modificarlo, attesa la comparsa della cifra zero fissa, bisogna ruotare il perno dell'encoder fino a far apparire il va-

lore voluto e quindi procedere come descritto in precedenza.

Per impostare la modalità di uscita bistabile, premete SW1 e attendete la comparsa del solito trattino luminoso, quindi rilasciate e verificate che avvenga il lampeggio indicante il tempo attualmente in vigore per il comando impulsivo; quando appare zero fisso, premete e mantenete premuto il pulsante, quindi ruotate il perno del commutatore fino a veder apparire sul display la lettera *c*, quindi rilasciate e verificate che appaia, lampeggiante per tre volte, il punto decimale. Si attiva, così, la modalità bistabile.

Esistono poi due funzionalità, da utilizzare in particolari applicazioni, che prevedono il mantenimento della condizione assunta dall'uscita dopo l'accensione del circuito solo se entro un minuto viene introdotta la combinazione esatta; in caso contrario, scaduto tale intervallo il relé inverte la propria condizione e la serratura si porta in allarme, fa lampeggiare la **A** maiuscola e non accetta più alcun comando. Bisogna togliere tensione e ricominciare da capo, oppure rialimentare il circuito e ripristinare le condizioni iniziali con la funzione *d*.

Andiamo con ordine e vediamo la modalità nella quale all'accensione del circuito il relé si attiva immediatamente; per impostare tale modalità, premete SW1 fin quando non appare il trattino luminoso, quindi attendete

il lampeggio del display, che mostra il tempo (se è attiva la funzione monostabile) o il solo puntino decimale (se è prevista la modalità bistabile). Quando appare lo zero, premete e mantenete premuto il tasto, ruotate il perno dell'encoder rotativo fino a far apparire sul display la lettera *i* (deve risultare illuminato il solo trattino in basso a destra) e rilasciate SW1; il display segnala che ora la serratura funziona nella modalità prevista. La segnalazione consiste in un gioco di luci nel quale un trattino luminoso si muove continuamente dall'alto verso il basso. Per rendere attiva la modalità *i* bisogna togliere e ripristinare alimentazione al circuito; il display effettuerà la stessa segnalazione di prima.

Ora il relé scatterà e resterà eccitato: a tempo indeterminato, se entro un minuto verrà composta l'esatta combinazione della serratura (condizione evidenziata della solita *a* minuscola e dal punto decimale); per un minuto, in caso contrario, ossia se i tentativi di introduzione del codice saranno condotti in maniera errata o non verrà introdotto nulla. In quest'ultimo caso RY1 ricadrà e il display segnerà la condizione di allarme (**A** maiuscola lampeggiante). Per ritentare l'introduzione del codice occorre nuovamente togliere e dare alimentazione. Per entrare in programmazione e disattivare la modalità *i*, bisogna sempre attendere che il relé torni a riposo. La fun-



zionalità in questione è utilizzabile, ad esempio, per locali o autoveicoli nei quali il sistema si attiva nel momento in cui si entra ma sospende temporaneamente l'allarme, dando all'utente un minuto di tempo per mantenere disattivato il sistema antifurto; se la persona riesce a inserire in tempo la combinazione esatta, l'allarme resta disattivato, altrimenti il relé ricade e fa partire l'antifurto. Una seconda modalità, opposta a quella appena descritta, prevede che all'accensione RY1 sia disattivato e lo resti a patto che entro il solito minuto venga inserita la combinazione valida; in tale evenienza, l'indicazione del display non cambia, mentre in caso contrario, allo scadere del tempo, il relé viene eccitato e appare la lettera *a* minuscola accompagnata dall'accensione del punto decimale, indicante l'eccitazione dell'RY1. Questa funzionalità si imposta premendo SW1 fin quando non appare il trattino luminoso e il lampeggio del display che indica il tempo (nel

>

Vendita componenti elettronici specializzati in forniture di enti pubblici e istituti professionali

Electronica s.a.s Via R. Grazzoli Lante, 22 - 00195 ROMA - Tel. 06 372 8112 - Fax. 06 3751 5380
www.electronicaroma.it - email: elyda@tiscali.net














Il dispositivo descritto in queste pagine può essere configurato per ottenere le diverse funzionalità elencate di seguito, ciascuna identificata dalla lettera corrispondente del menu di configurazione.

- **t** = ad ogni inserimento della combinazione esatta, il relé scatta e resta eccitato per un certo intervallo; il display mostra il tempo corrispondente, che decresce fino a zero, allorché RY1 torna a riposo.
- **c** = ad ogni inserimento della sequenza corretta, il relé scatta (il display mostra la lettera **A** minuscola e il punto decimale) e resta eccitato fin quando non si ruota da una parte o dall'altra la manopola.
- **i** = all'accensione del circuito, il relé si eccita e resta attivo solo se entro un minuto viene introdotta la combinazione esatta; in tale evenienza il display mostra la lettera **a** minuscola e il puntino decimale. Scaduto il minuto, il relé va a riposo e la serratura si inibisce, facendo lampeggiare la **A** maiuscola e rifiutando qualsiasi comando. Dopo aver spento e riacceso il circuito è possibile ricominciare da capo oppure, sempre dopo uno spegnimento momentaneo, ripristinare le condizioni iniziali con la funzione **d**.
- **o** = all'accensione del circuito, il relé si trova a riposo e vi resta solo se entro un minuto viene introdotta la combinazione esatta; in caso contrario, scaduto tale intervallo, si eccita (e resta attivo fin quando non si toglie tensione) e la serratura si inibisce facendo lampeggiare la **A** maiuscola e rifiutando qualsiasi comando. Bisogna togliere tensione e ricominciare da capo, oppure rialimentare il circuito e ripristinare le condizioni iniziali con la funzione **d** prima che passi un minuto (perché a relé attivato SW1 non ha più effetto).

Viste le funzionalità della serratura, vediamo qui di seguito, una per una,

come si configura il circuito per ottenerle; spieghiamo inoltre, sinteticamente, le procedure per operare tutte le altre impostazioni.

MODO IMPULSIVO (t).

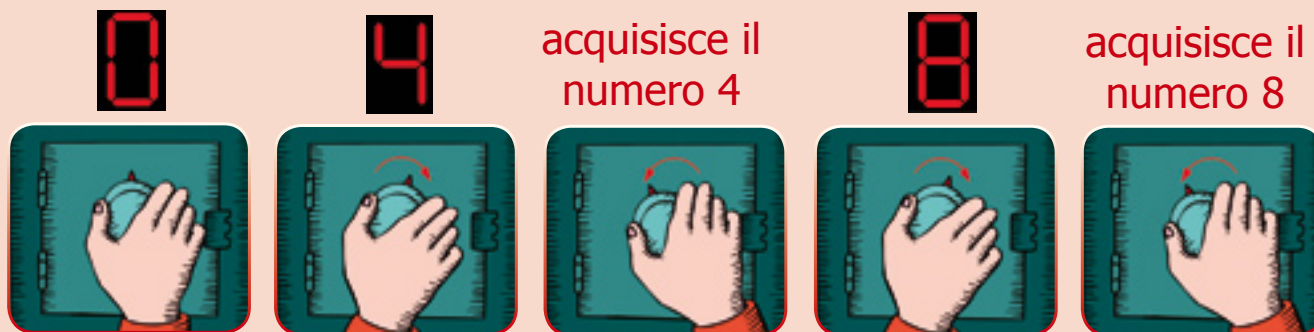
Per impostare la modalità impulsiva premete SW1 e attendete la comparsa del trattino luminoso; trascorso poco più di un secondo rilasciate il pulsante: il display deve visualizzare la cifra zero. A questo punto, ruotate la manopola della combinazione fino a far apparire il tempo di attivazione scelto per il relé, che può essere compreso fra 0 (corrispondente a 0,5 sec) e 9 secondi; ora si deve nuovamente premere SW1 e, mantenendolo premuto, ruotare la manopola fino a fare comparire sul visualizzatore la lettera **t**. Si è così impostato il modo impulsivo; il display lampeggerà tre volte, insieme al punto decimale, indicando il tempo scelto, quindi tornerà a mostrare zero. Ciò segna la fine della fase di configurazione.

MODO BISTABILE (c).

Per impostare la modalità di uscita bistabile (a livello) premete SW1 e attendete la comparsa del solito trattino luminoso, quindi rilasciate e verificate che avvenga il lampeggio indicante il tempo attualmente previsto per il comando impulsivo; quando appare zero fisso, premete e mantenete premuto il pulsante, quindi ruotate il perno del commutatore fino a veder apparire sul display la lettera **c**, quindi rilasciatelo e verificate che appaia, lampeggiante per tre volte, il punto decimale.

ATTIVAZIONE RELÉ ALL'ACCENSIONE (i)

Per impostare tale modalità, premere SW1 fin quando non appare il trattino luminoso, quindi attendete il lampeggio del display, che mostra il tempo se



caso di impostazione monostabile) o il solo puntino decimale (se ci troviamo in modalità bistabile). Quando appare lo zero, premete e mantenete premuto il tasto, per poi ruotare il perno dell'encoder rotativo fino a far apparire sul display la lettera **o**; dopo il rilascio di SW1 il display segnala (con il solito gioco di luci) che la serratura funziona nella modalità prevista. Per rendere attiva la modalità bisogna togliere e ridare alimentazione al circuito; il display mostrerà la solita segnalazione; il relé normalmente sarà a riposo e, all'occorrenza, sarà possibile intervenire su SW1 per modificare la

configurazione della serratura, cambiare combinazione o disattivare la modalità in questione. Introducendo la combinazione esatta entro un minuto, RY1 resterà a riposo e nulla cambierà nel circuito, altrimenti il relé verrà eccitato, cosa evidenziata dall'accensione lampeggiante della **A** maiuscola (non di quella minuscola e del punto decimale).

Cambiare la combinazione

Oltre alla modalità di funzionamento del relé, il software del micro consente altre funzioni di setup con le quali è possibile sostituire la combinazione (quella preimpostata

è 1010) o riportare il sistema alle impostazioni di fabbrica; nel primo caso, sempre quando RY1 è a riposo, premete SW1 e rilasciatelo solo dopo che si è illuminato il trattino orizzontale del display, quindi attendete i tre lampeggi (del punto decimale, del trattino o del numero corrispondente al tempo della modalità monostabile) e la comparsa dello zero fisso, ripremete e mantenete premuto il tasto, ruotando il perno del commutatore finché non viene visualizzata la lettera **s**: ciò indica l'avvio della procedura di sostituzione della combinazione. Ora, ruotando il solito perno bisogna im-

è prevista l'attivazione monostabile o il solo puntino decimale se, invece, è impostata la modalità bistabile. Quando appare lo zero, premete e mantenete premuto il tasto, ruotate il perno dell'encoder rotativo fino a far apparire sul display la lettera *i* (deve risultare illuminato il solo trattino in basso a destra) e rilasciate SW1; il display segnala con un gioco di luci che da questo momento la serratura funziona nella modalità prevista. In pratica il trattino luminoso si muove continuamente dall'alto verso il basso. Per rendere attiva la modalità *i* bisogna spegnere e riaccendere il circuito. Il circuito segnala la correttezza dell'operazione con la stessa segnalazione luminosa.

ATTIVAZIONE RELÉ AUTOMATICA (o)

Volendo che, dopo l'accensione, RY1 resti a riposo a patto che entro un minuto venga inserita la corretta combinazione, bisogna operare premendo (a relé disattivato) SW1 fino a fare apparire il trattino luminoso, quindi attendendo il lampeggio del display, che visualizza il tempo impostato (se è prevista l'attivazione monostabile) o il solo puntino decimale (se il circuito opera in modalità bistabile). Quando appare lo zero, bisogna premere e mantenere premuto il tasto, ruotando il perno dell'encoder fino a far apparire sul display la lettera *o*; rilasciate quindi il pulsante SW1.

CAMBIO COMBINAZIONE

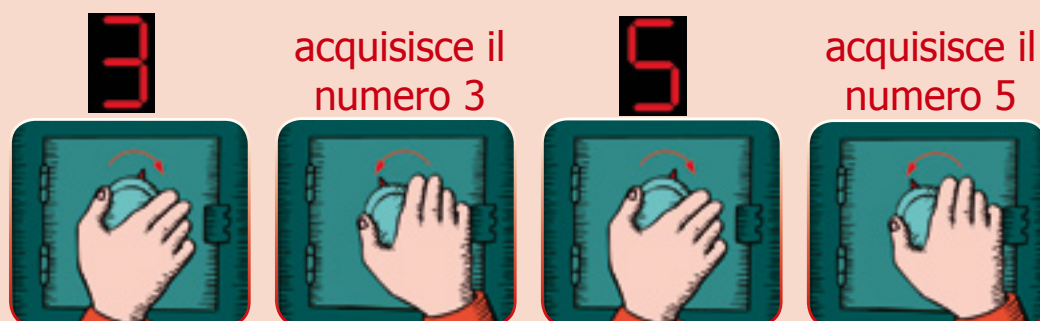
Operando sempre con il relé a riposo, premete SW1 e rilasciatelo solo dopo che si è illuminato il trattino orizzontale del display, quindi attendete i tre lampeggi (del punto decimale, del trattino o del numero corrispondente al tempo della modalità monostabile) e la comparsa dello zero fisso; ripremete e mantenete premuto il tasto ruotando la manopola del commutatore finché non viene visualizzata la lettera *s*. Ora, ruotando il solito perno, bisogna

impostare il nuovo codice. Ricordiamo che viene memorizzata la cifra corrispondente al numero visualizzato sul display ogni volta che si inverte il verso di rotazione rispetto a quello iniziale; per esempio, se si ruota in senso orario e ci si ferma a 6, poi si cambia verso, il microcontrollore memorizza 6, quindi se si torna a ruotare in senso orario e ci si ferma a 2, per poi invertire la rotazione, viene memorizzato 2, e così via di seguito.

Dopo la memorizzazione della quarta cifra che completa la sequenza della combinazione, il punto decimale del display lampeggia tre volte.

RIPRISTINO

Quanto alla funzione di ripristino della configurazione originaria, si realizza premendo il solito SW1 (a relé disattivato...) e rilasciandolo quando appare il trattino; ora, atteso che il display faccia lampeggiare tre volte l'impostazione attuale dell'uscita a relé e mostri fisso lo zero, ripremete e mantenete premuto il pulsante, ruotando poi il perno dell'encoder rotativo fino a quando il display non visualizza la lettera *d*. Rilasciate quindi SW1 e verificate che il punto decimale lampeggi tre volte; la serratura è ora nelle condizioni iniziali (uscita bistabile, combinazione 1010).



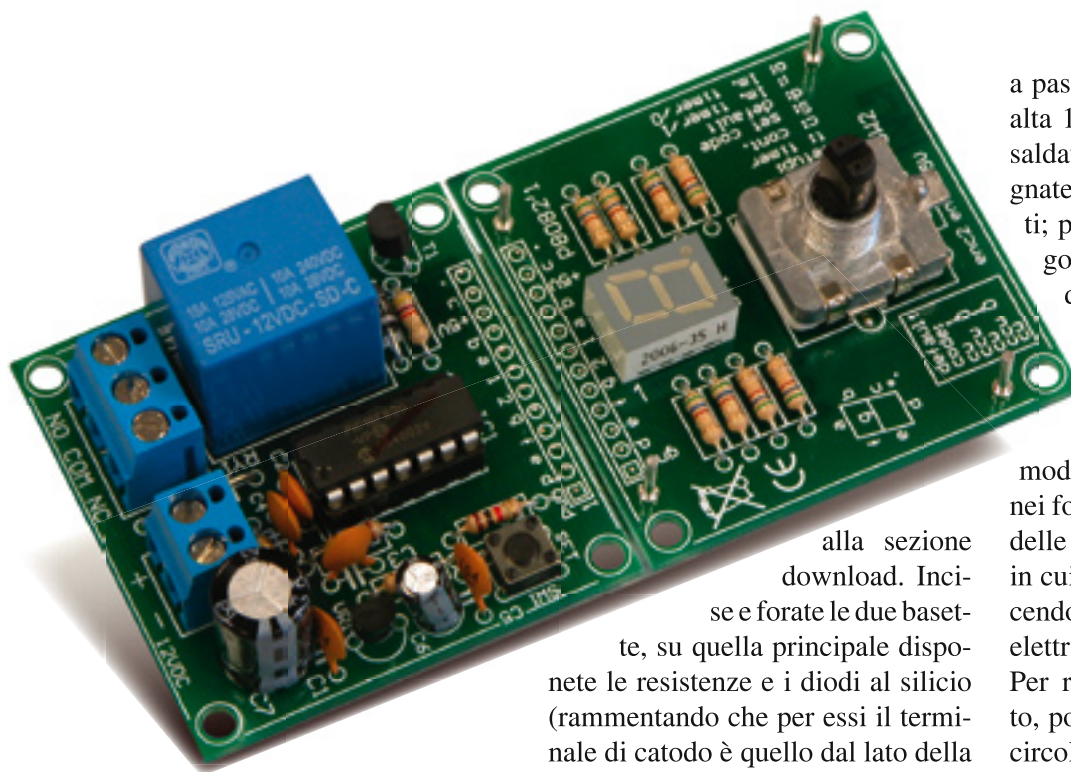
Le cifre di una combinazione vengono acquisite solo a fronte di una inversione del verso di rotazione.

Nell'esempio (4835) si seleziona la prima cifra (4) ruotando la manopola a destra. Invertendo il senso di rotazione il "4" viene acquisito dal sistema.

postare il nuovo codice. Ricordiamo che viene memorizzata la cifra corrispondente al numero visualizzato sul display ogni volta che si inverte il verso di rotazione rispetto a quello iniziale; per esempio, se si ruota in senso orario e ci si ferma a 6, poi si cambia verso, il microcontrollore memorizza 6, quindi se si torna a ruotare in senso orario e ci si ferma a 2, per poi invertire la rotazione, viene memorizzato 2, e via di seguito. Dopo la memorizzazione della quarta cifra che completa la sequenza della combinazione, il punto decimale del display lampeggia tre volte. La nuova combinazione è così

memorizzata. Quanto alla funzione di ripristino della configurazione originaria, si realizza premendo il solito SW1 (a relé disattivato...) e rilasciandolo quando appare il trattino; ora, atteso che il display faccia lampeggiare tre volte l'impostazione attuale dell'uscita a relé e mostri fisso lo zero, ripremete e mantenete premuto il pulsante, ruotando poi il perno dell'encoder rotativo fino a quando sul display non appare la lettera *d*. Rilasciate SW1 e verificate che il punto decimale lampeggi tre volte; la serratura è ora nelle condizioni iniziali, come la prima volta che l'avete alimentata (uscita bistabile,

combinazione 1010). Chiudiamo la descrizione del circuito con lo stadio di alimentazione, che fa capo al regolatore integrato VR1, al quale è affidato il compito di ricavare 5 volt ben stabilizzati con cui viene fatto funzionare il microcontrollore e con esso il display; il componente (un 78L05) prende tensione a valle del diodo di protezione (dall'inversione di polarità) D1 ed ha l'ingresso ben filtrato tramite i condensatori C1 e C7. La bobina del relé prende tensione a valle del diodo di protezione. L'intero apparato richiede un'alimentazione in continua compresa tra 12 e 15 V con una corrente >



di 100 mA. La chiave a combinazione può quindi essere inserita anche in sistemi funzionanti a batteria (impianti antifurto con accumulatore di emergenza).

Realizzazione pratica

Per il montaggio di questa particolare chiave a combinazione abbiamo utilizzato due circuiti stampati, uno dei quali ospita microcontrollore, regolatore integrato e quant'altro serve, ad eccezione del display a sette segmenti, delle relative resistenze di limitazione e del commutatore rotativo, che vanno sul secondo stampato; le tracce per realizzare entrambi sono scaricabili dal nostro sito www.elettronica.in.it,

alla sezione download. Incidete e forate le due basette, su quella principale dispo-
nete le resistenze e i diodi al silicio (rammentando che per essi il terminale di catodo è quello dal lato della fascetta colorata sul corpo) quindi lo zoccolo per il microcontrollore; proseguite sistemando il pulsante da c.s. e i condensatori (prestando attenzione alla polarità degli elettrolitici) quindi il relé a 12 V uno scambio. Infine, per l'alimentazione e i contatti dello scambio del RY1, montate due morsettiere da stampato a passo 5 mm, una bipolare e l'altra tripolare. Sulla seconda basetta collocate prima le resistenze, poi il display che va posizionato inserendolo a fondo nei rispettivi fori, come mostra l'apposito disegno (il punto decimale deve risultare vicino al commutatore rotativo). Per ultimo montate l'encoder. A questo punto procuratevi uno spezzone di piattina a dodici conduttori, oppure una strip

a passo 2,54 mm ad altrettanti poli, alta 15 mm circa, infilatela dal lato saldature di uno stampato e lì stagnatela nelle piazzole corrispondenti; prendete quattro colonnine esagonali lunghe 10 mm o altrettanti distanziali di pari dimensione e fissate l'altra basetta, montando le due tracce rame una di fronte all'altra (vedere foto del prototipo) e facendo in modo che le punte libere entrino nei fori corrispondenti. Strette le viti delle colonnine, stagnate le piazzole in cui sono entrate le punte; così facendo avete assemblato e collegato elettricamente i due stampati.

Per realizzare un insieme compatto, potete preparare una mascherina circolare, rettangolare o quadrata, di plastica o metallo, dello spessore che desiderate, con un occhio per rendere visibile il display e un foro dal quale far spuntare il perno dell'encoder rotativo, col cui dado potrete poi fissare la scheda.

Un'opportuna foratura permetterà di agganciare la mascherina e l'intera serratura al pannello della porta da aprire, della cassaforte da proteggere o dell'impianto di allarme da attivare. Per concludere, ricordiamo come la serratura a combinazione qui pubblicata rappresenti una fusione sinergica tra semplicità, efficienza e costo di realizzazione; le applicazioni da noi consigliate non sono che una parte di quelle che, siamo certi, una volta montata vi verranno in mente.

per il

MATERIALE

Il progetto descritto in queste pagine è disponibile in scatola di montaggio (cod. K8082) al prezzo di 24,00 Euro. Il kit comprende tutti i componenti, le basette forate e serigrafate, le minuterie meccaniche, il microcontrollore programmato, l'encoder e le morsettiere da circuito stampato. Del kit fa anche parte il pannellino frontale forato e serigrafato. Tutti i prezzi si intendono IVA compresa.

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

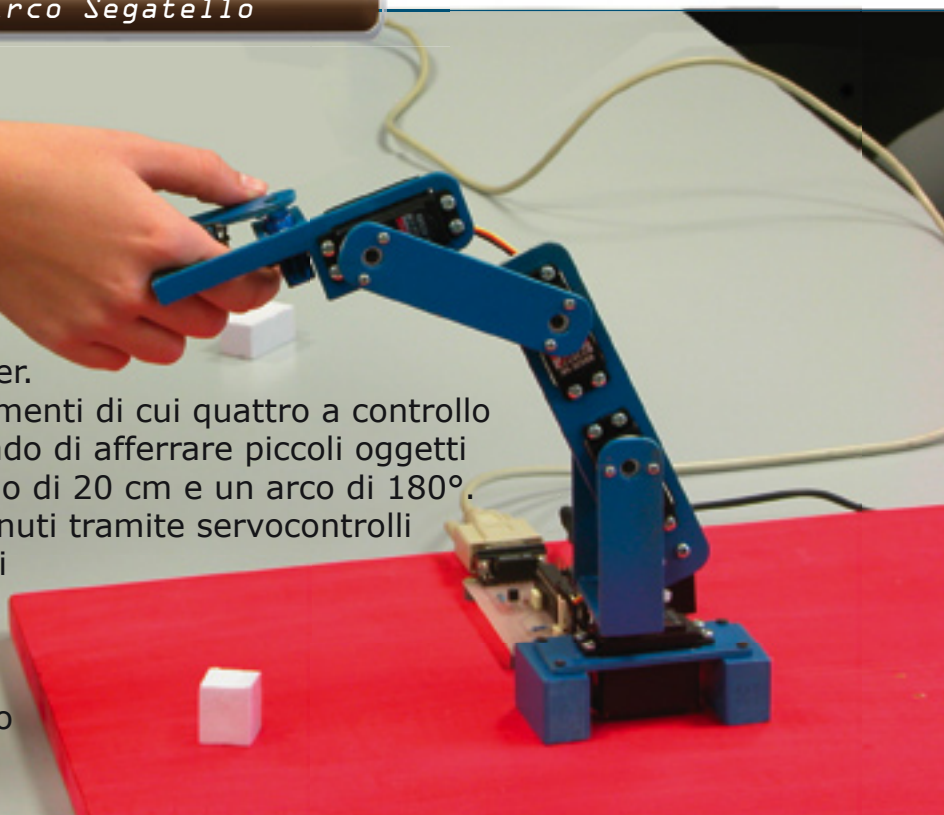
Tel: 0331-799775 ~ Fax: 0331-778112 ~ <http://www.futuranet.it>

Braccio robotizzato con controllo seriale

di *Mirco Segatello*

Braccio
elettromeccanico
controllato da computer.

Dotato di cinque movimenti di cui quattro a controllo proporzionale, è in grado di afferrare piccoli oggetti disposti entro un raggio di 20 cm e un arco di 180°. I movimenti sono ottenuti tramite servocontrolli per modellismo pilotati da una scheda a microprocessore che comunica con il programma di controllo installato nel PC.



Tra le varie applicazioni della robotica, sicuramente tutte quelle apparecchiature che cercano in qualche modo di riprodurre i movimenti del corpo umano presentano sicuramente un grande fascino. Una di queste è il braccio meccanico o robotizzato che dir si voglia. Si tratta di un complesso di meccanismi atti a simulare il movimento di un braccio umano e la presa della mano, che sicuramente è il migliore esempio di articolazione di movimenti. Al giorno d'oggi l'utilizzo di apparecchiature robotiche è fondamentale, soprattutto in campo industriale anche se ultimamente l'affidabilità

e l'alta tecnologia hanno permesso il suo utilizzo anche in campo medico (manipolatori per ambienti sterili o a rischio di contagio, telemedicina ed interventi chirurgici remotizzati). Lungi dal cercare di replicare un arto umano, in questo articolo spiegheremo come realizzare un semplice braccio meccanico pilotabile da Personal Computer e capace di cinque diverse articolazioni. Si tratta di un'apparecchiatura composta da una parte di supporto, una parte elettro-meccanica per eseguire i singoli movimenti, una scheda di interfacciamento al PC e di un software di controllo. Nel corso dell'artico-



lo descriveremo come realizzare le singole parti e come assemblarle nel modo più semplice possibile.

Gli attuatori elettromeccanici

Iniziamo la descrizione di questo progetto di robotica amatoriale partendo dai componenti che realizzeranno, in pratica, il movimento vero e proprio delle articolazioni. Per realizzare un sistema meccanico

in grado di eseguire dei movimenti sono disponibili più soluzioni: pistoni pneumatici, motoriduttori, motori passo-passo. Scartiamo a priori i pistoni pneumatici che, anche se permettono dei movimenti fluidi e potenti, sono difficilmente gestibili per due ragioni: la necessità di aria compressa ed i costi elevati. I motoriduttori, invece, sono gestibili tramite corrente elettrica, ma bisogna

tener conto, anche in questo caso, dei costi. Utilizzare dei semplici motoriduttori implica la realizzazione di un controllo di posizione fatto con encoder o trasduttori lineari con rilevante complessità ed elevati costi. Parte di questo problema può essere risolta con l'impiego di motori passo-passo magari recuperati da vecchie stampanti o acquistati in qualche fiera dell'elettronica. An- ➤

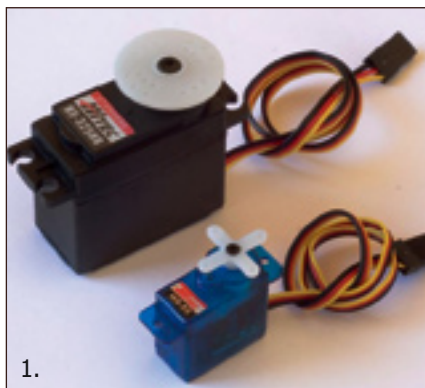


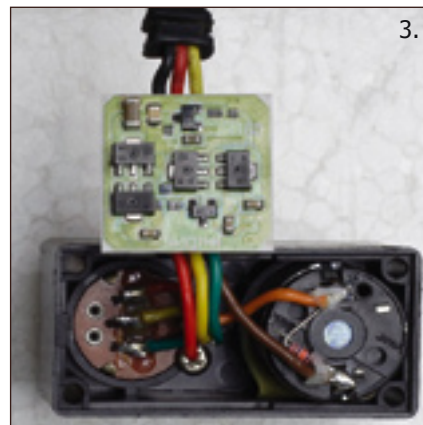
Fig. 1: Esempi di servo per modellismo.

che questi, comunque, necessitano di un controllo di potenza che necessariamente va ad incidere sulla complessità del progetto. Volendo accontentarci di realizzare un braccio robotizzato di dimensioni limitate, tanto da poter essere contenuto in una scatola da scarpe e che sia di facile realizzazione, la scelta per i movimenti è caduta sull'utilizzo di servocontrolli da modellismo. Si tratta di piccoli dispositivi elettromeccanici che comprendono un sofisticato sistema di posizionamento e che sono facilmente gestibili con i segnali provenienti da un microcontrollore. Questa soluzione offre la possibilità di semplificare moltissimo la parte meccanica e di rendere il progetto fruibile ad una più vasta cerchia di lettori; inoltre, il loro costo è decisamente contenuto e inferiore a tutte le altre soluzioni viste in precedenza. Per rendersi conto di come è fatto un servo da



Fig. 2: Gli ingranaggi interni del riduttore.

Fig. 3: Aspetto interno di un servo da modellismo. In basso a sinistra è visibile il potenziometro; a destra, il motore. Al centro, invece, il circuito di controllo.



modellismo osserviamo la foto di figura 1.

Per comprenderne meglio il funzionamento abbiamo provveduto ad aprirne uno e fare alcune foto. In figura 3 è visibile la parte interna costituita da una piccola scheda di controllo in SMD, un potenziometro (in basso a sinistra) ed il motore (in basso a destra). Nella figura 2, invece, si vedono gli ingranaggi interni ed il perno (in alto a sinistra) che spunta dal contenitore e rende disponibile il movimento.

Un servo è costituito quindi da un piccolo motore in corrente continua che grazie ad un sistema di ingranaggi fa ruotare un perno sul quale è calettato un piccolo potenziometro che, grazie al suo valore resistivo, fornisce la posizione esatta del perno sotto forma di tensione elettrica. Il circuito elettronico provvede al controllo bidirezionale del motore e al suo corretto posizionamento in

relazione al segnale elettrico che gli viene fornito. Il tutto funziona secondo lo schema riportato in figura 4a. Il posizionamento avviene confrontando il valore in tensione fornito dal potenziometro con il valore di tensione ricavato dal segnale di ingresso e ruotando di conseguenza il motore sino a quando questi due valori non coincidono.

In questo modo si ottiene un controllo di posizione molto veloce e preciso, controllabile con semplici segnali elettrici. Il cavetto di collegamento è composto da un filo di riferimento (GND), uno per l'alimentazione (da 4,8 a 6 volt) e uno per il segnale di comando. Ci preme precisare che non è prevista la rotazione continua del perno ma solo del valore di $\pm 60^\circ$ rispetto alla posizione iniziale, anche se è possibile espandere la rotazione sino a $\pm 90^\circ$. Il servo utilizzato nel nostro progetto è di tipo standard, tra i più comuni

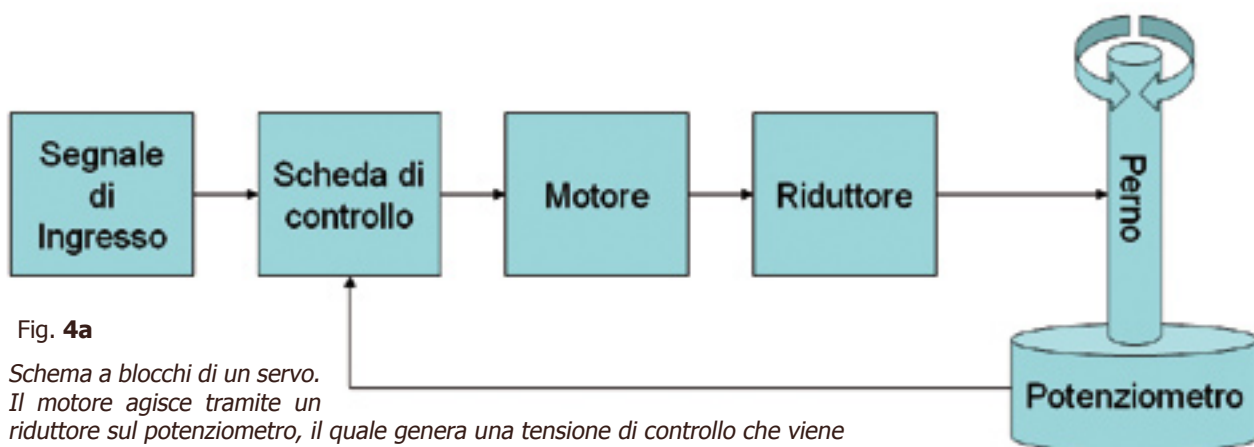
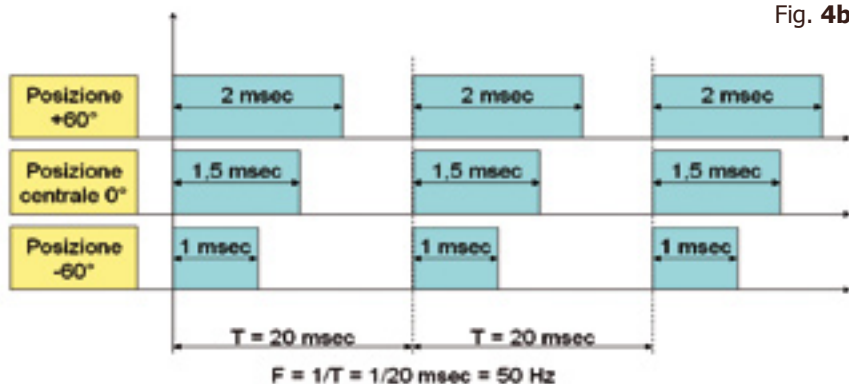


Fig. 4a

Schema a blocchi di un servo. Il motore agisce tramite un riduttore sul potenziometro, il quale genera una tensione di controllo che viene costantemente confrontata dal circuito elettronico con il segnale in ingresso.



ed economici reperibili in commercio; queste sono le caratteristiche più significative:

- Alimentazione da 4,8 a 6 volt;
- Corrente massima assorbita 300 mA;
- Massima rotazione $\pm 60^\circ$;
- Coppia 3 Kg/cm;
- Rotazione di 60° in un tempo di appena 0,19 sec;
- Peso 43 grammi;
- Dimensioni 40x20x37 mm;

In commercio si trovano numerosi servocontrolli con caratteristiche simili e costo contenuto. Esistono ovviamente versioni di potenza superiore e inferiore con alcuni modelli realizzati con ingranaggi in metallo e assemblati con cuscinetti a sfere in modo da garantire migliori prestazioni e durata superiore. Recentemente sono stati introdotti sul mercato alcuni servocontrolli digitali con controllo interno a microprocessore che garantiscono una funzionalità migliorata. Nel modello da noi utilizzato (ma anche in tutti gli altri: si tratta di uno standard) il segnale di controllo è di tipo PWM (Pulse Wide Modulation) formato da impulsi ad onda quadra, ripetuti ogni 20 msec, la cui "larghezza" permette di stabilire la posizione del perno. Con impulsi della durata di 1,5 msec il perno si porta in posizione centrale, come si può vedere in figura 4b. Questo tipo di segnale digitale si presta facilmente ad essere generato da una logica programmabile semplificando l'interfacciamento

di questi dispositivi con i microcontrollori.

Tester per servocontrolli

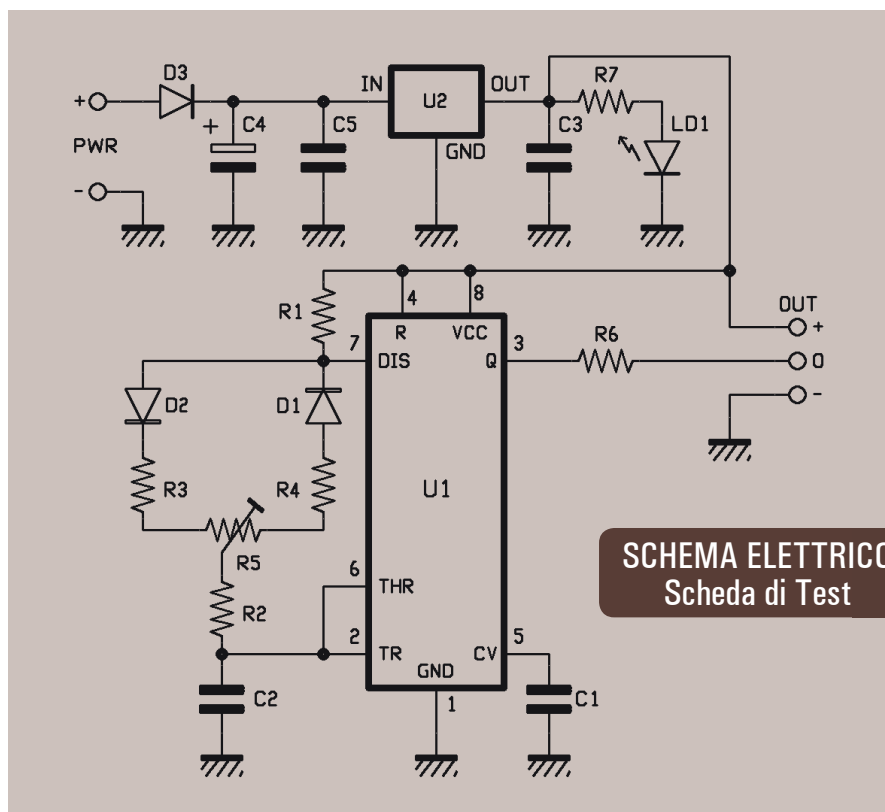
Durante la costruzione del nostro braccio robotizzato potrebbe essere necessario (in caso di dubbio o per fare dei test di velocità e potenza), controllare il funzionamento di un servo controllo. In questo caso è utilissimo un tester dedicato in grado di fornire il segnale di controllo ad onda quadra in modo variabile tramite un potenziometro. Esso può essere realizzato semplicemente con un integrato NE555 e con pochi al-

tri componenti montati secondo lo schema visibile qui sotto.

Come potete notare si tratta di un circuito molto semplice che può essere realizzato facilmente in pochissimo tempo. Il master dello stampato può essere scaricato dal sito della rivista. L'alimentazione può essere fornita da un piccolo adattatore non stabilizzato con tensione di uscita di 9-12 V. Per il collegamento del servo, si faccia attenzione alla polarità: il filo nero (o marrone) va collegato al negativo, il filo rosso al positivo e il filo giallo (o bianco) al segnale di controllo.

Potete comunque aiutarvi con le informazioni riportate nella pagina successiva, in cui abbiamo riportato, oltre al piano di montaggio e all'elenco componenti, una tabella che riassume i cablaggi dei servo più comuni e la relativa connessione sulla nostra scheda di test.

Ruotando il trimmer è possibile variare il valore del duty-cycle del PWM e quindi la posizione del servo. Se provate a ruotare l'alberino noterete che rimarrà bloccato nella ➤





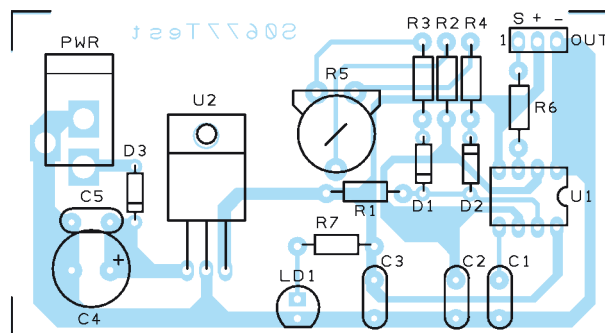
Colore	Pin di JP1	Funzione
Nero o marrone	3	Negativo di alimentazione (GND)
Rosso	2	Positivo di alimentazione (+Vcc)
Giallo o bianco	1	Segnale di ingresso

ELENCO COMPONENTI:

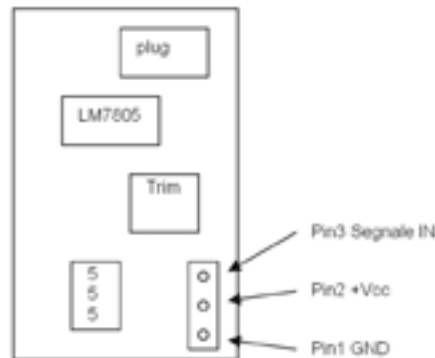
D1, D2: 1N4148	R6: 470 ohm	elettrolitico
D3: 1N4007	R7: 1 kohm	C5: 100 nF multistrato
R1, R2: 1 kohm	C1: 100 nF multistrato	U1: NE555
R3: 6,8 kohm	C2: 120 nF multistrato	U2: 7805
R4: 180 kohm	C3: 100 nF multistrato	LD1: led 5mm rosso
R5: trimmer 10 kohm	C4: 100 µF 25VL	

Varie:

- Plug alimentazione
- Strip maschio 3 poli
- Zoccolo 4+4
- Dado 3 MA
- Vite 10 mm 3 MA
- Circuito stampato



Piano di montaggio e specifiche dei fili di collegamento dei servocontrolli.



sua posizione anche se udirete un leggero rumore dovuto allo sforzo del motore interno.

La meccanica

Per realizzare la struttura meccanica abbiamo utilizzato quattro servo di tipo standard come quelli descritti precedentemente; in questo progetto è stato scelto il modello HS-325HB prodotto dalla Hitec. Per realizzare la chiusura e apertura della pinza, invece, abbiamo scelto un servo più piccolo, sempre prodotto dalla Hitec: il modello HS-55. Esso presenta le seguenti caratteristiche:

- Alimentazione da 4,8 a 6 volt;
- Corrente massima assorbita 200 mA;
- Massima rotazione $\pm 60^\circ$;
- Coppia 0,7Kg/cm;
- Tempo di 0,85 sec per una rotazione di 60° ;
- Peso 8 grammi;
- Dimensioni 21x13x22 mm;

In questo modo si ottengono quattro gradi di libertà, più un quinto movimento dedicato alla presa degli og-

getti. La struttura meccanica è stata pensata per poter essere facilmente realizzata da chiunque anche se è comunque necessario possedere un minimo di manualità nonché disporre degli strumenti adeguati. Il materiale più idoneo da usare è il plexiglass o la vetronite, entrambi dello spessore di 4-5 mm. Il plexiglass

è facilmente reperibile nel reparto bricolage dei Centri Commerciali, è lavorabile e si può incollare con semplicità. La vetronite, se di tipo ramato, può essere saldata ed è reperibile oltre che nei negozi di elettronica anche nelle fiere di settore. Per il taglio dei vari elementi si faccia riferimento alla figura 6. Lo stes-

Braccio Robotizzato
schema di montaggio

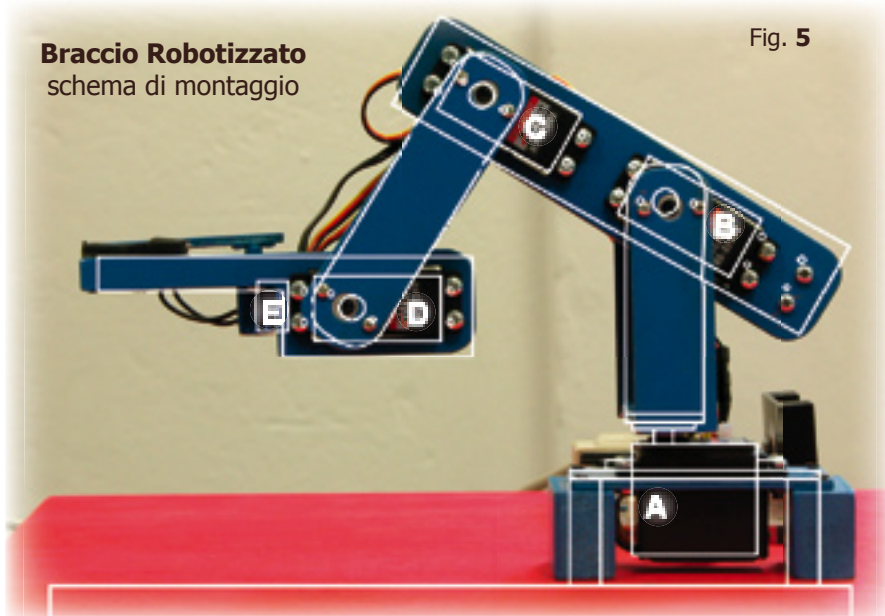


Fig. 5

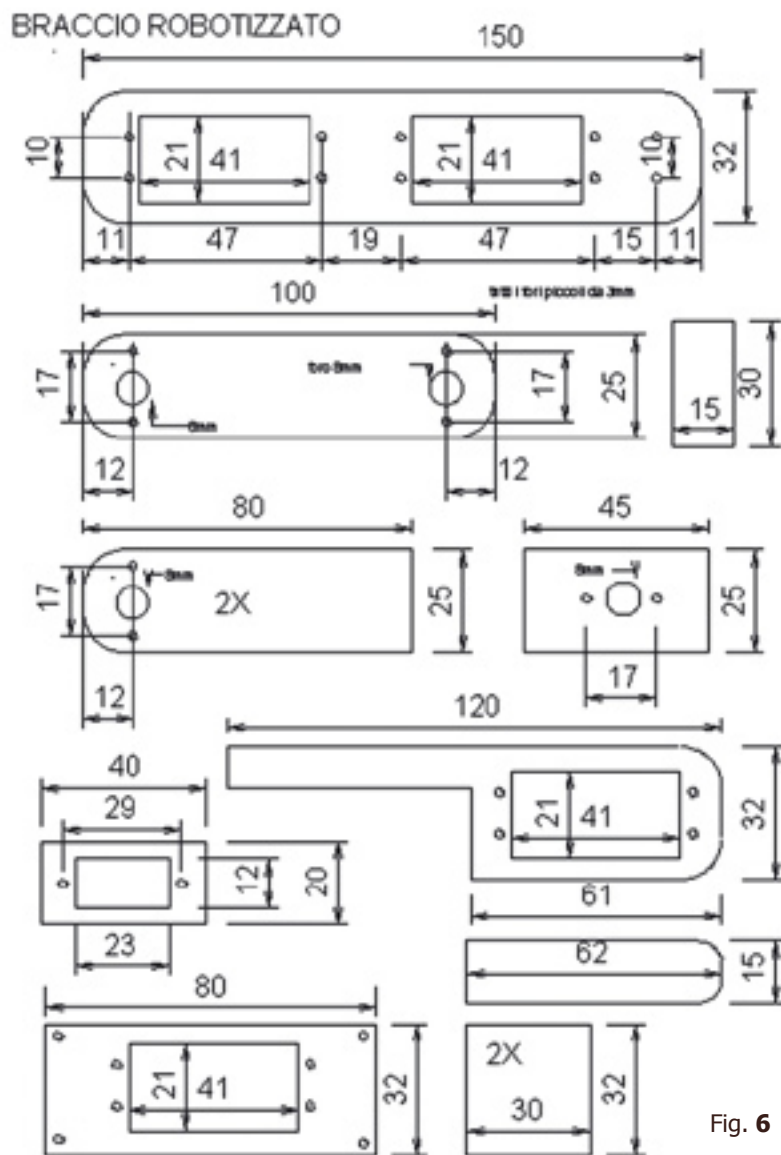


Fig. 6

so disegno in formato DXF può essere scaricato, insieme ai file relativi a questo progetto, direttamente dal sito della rivista.

Questo file permette la stampa in formato 1:1 che risulta compati-

le con un foglio formato A4. Per il montaggio delle varie parti si faccia riferimento alla figura 5 ed alle foto che trovate in queste pagine.

Ovviamente, durante le fasi di montaggio, potrete eseguire tutte le mo-

difiche che riterrete più opportune al fine di migliorare il progetto proposto. Durante il montaggio prestate particolare attenzione alle operazioni riguardanti la pinza; infatti, non essendo note a priori le dimensioni dell'oggetto da afferrare, è stato necessario dotarla di due sensori per rilevare la forza esercitata durante la presa dell'oggetto: superato un certo valore, i sensori fermano il servo evitando così che rimanga inutilmente e pericolosamente sotto sforzo.

Il tutto è stato realizzato con l'utilizzo di due semplici pulsanti da circuito stampato (collegati in parallelo) che commutano quando vengono premuti con una certa pressione.

Quando il PIC ne rileva lo stato di chiusura, ferma la chiusura della pinza per poi aprirla leggermente, lasciando comunque una sufficiente presa (figura 7).

Due gommini autoadesivi (del tipo di quelli utilizzati come paraspifferi) forniscono il giusto attrito anche su oggetti lisci o di forma irregolare (figura 8).

Il fissaggio del servo è stato effettuato utilizzando delle comunissime viti (con relativo bullone) da 3 mm facilmente reperibili al supermercato; per fissare le squadrette del servo sono state invece utilizzate delle viti autofilettanti da 2,3 mm.

Un piccolo peso realizzato in ferro o piombo provvede a bilanciare il braccio riducendo lo sforzo sul servo B (figura 9).

Al fine di distribuire meglio lo sfor- ➤

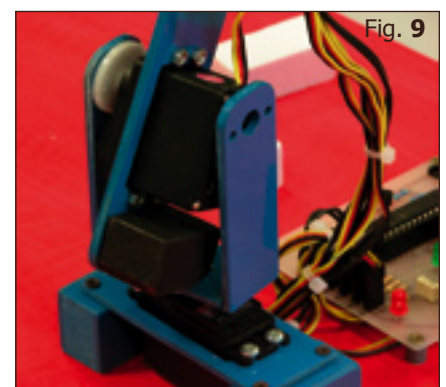
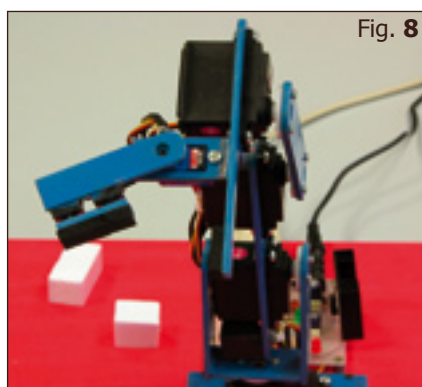
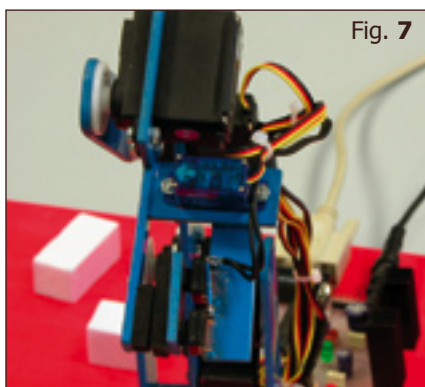


Fig. 10a

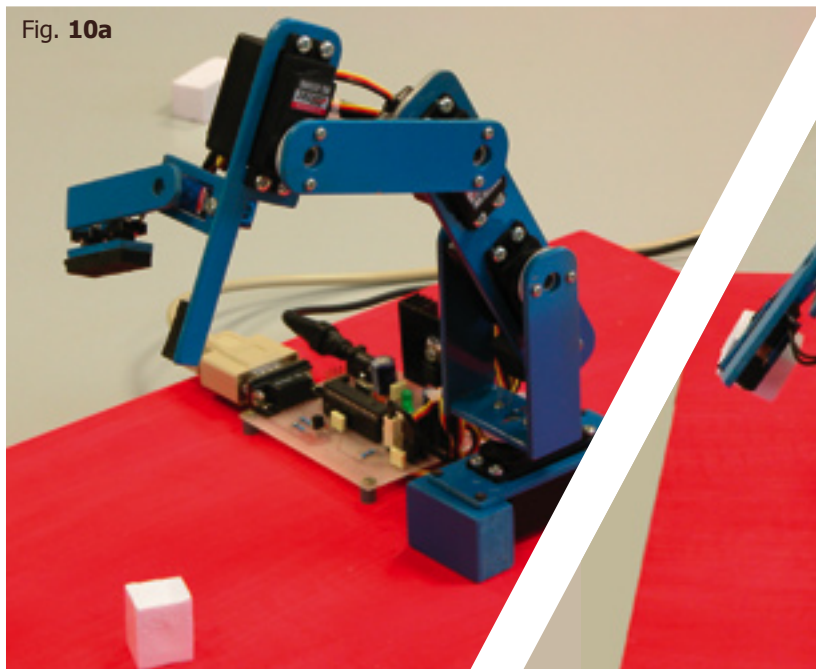
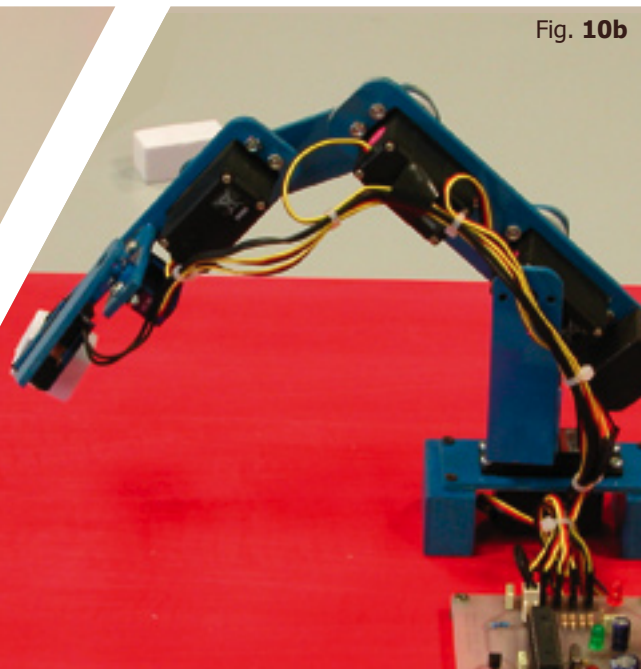


Fig. 10b



zo su questo servo è stato previsto di poter montare un piccolo perno aggiuntivo, come indicato dalla freccia nella figura 9.

In alcune posizioni, soprattutto con il braccio tutto esteso, i servocontrolli più vicini alla base sono sog-

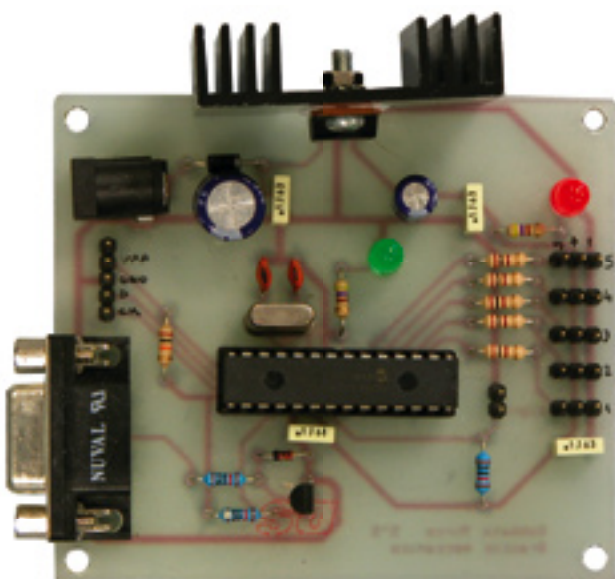
getti a notevole sforzo ed il piccolo motore interno potrebbe emettere un leggero trillo.

Una base in legno di dimensione adeguate permette di fissare il braccio e l'elettronica in modo ottimale. Tutti i cavi dei servo possono essere

raggruppati tramite delle semplici fascette da elettricista acquistabili anche al supermercato.

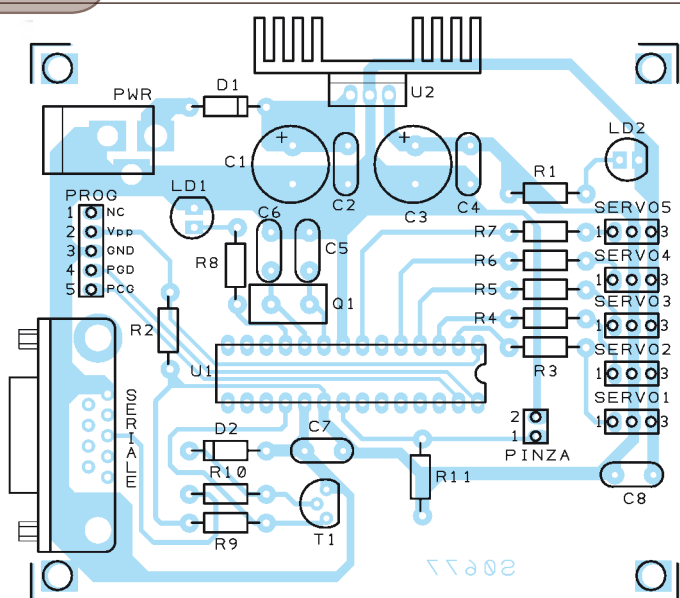
L'aspetto finale del prototipo del braccio robotizzato da noi realizzato è visibile in questa pagina, precisamente in figura 10a e 10b.

piano di MONTAGGIO della SCHEDA di CONTROLLO



ELENCO COMPONENTI:

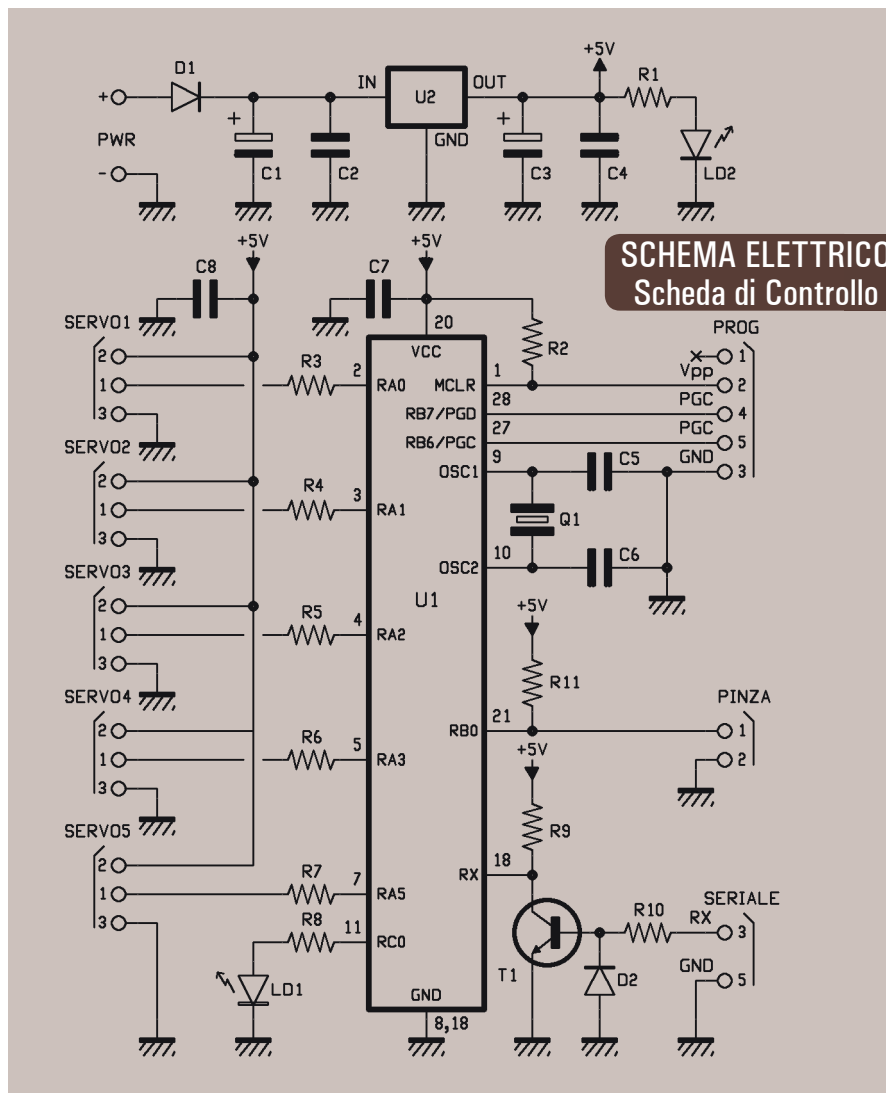
D1: 1N4007	C2, C4: 100 nF multistrato
D2: 1N4148	C3: 100 µF 25VL elettrolitico
R1, R8: 470 ohm	C7, C8: 100 nF multistrato
R2, R9, R10, R11: 10 kohm	C5, C6: 22 pF ceramico
R3+R7: 100 ohm	Q1: quarzo 4 MHz
C1: 470 µF 25VL elettrolitico	T1: BC547



U1: PIC16F876A (MF677)
U2: 7805
LD1: led 5mm verde
LD2: led 5 mm rosso

Varie:
- Connettore DB9 femmina

- Strip maschio 3 poli (5 pz.)
- Strip maschio 5 poli
- Strip maschio 2 poli
- Plug alimentazione
- Dissipatore (ML26)
- Zoccolo 14+14
- Circuito stampato



Il circuito elettrico

I cinque servo vengono gestiti da un microcontrollore della Microchip, un PIC 16F876, ed il circuito che ne risulta è molto semplice, come visibile nello schema elettrico.

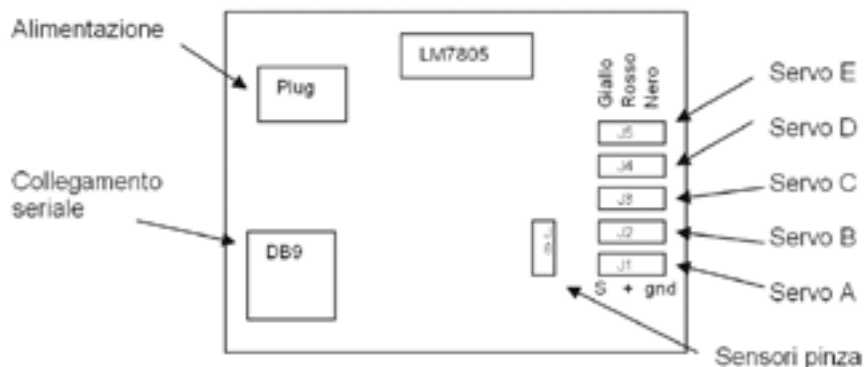
I disegni relativi al circuito stampato a singola faccia sono scaricabili dal sito della rivista insieme a tutti gli altri file di questo progetto. Per la realizzazione non ci sono particolari indicazioni e si procede come per tutti gli altri circuiti elettronici. L'unica accortezza è quella di utilizzare un piccolo dissipatore per l'integrato 7805, in quanto i servo assorbono una notevole quantità di corrente durante il funzionamento. Per migliorare la dissipazione di calore consigliamo anche l'utilizzo di un po' di pasta al silicone tra il

dissipatore e l'integrato. Farete inoltre in modo di far muovere un solo servo alla volta in modo da ridurre l'assorbimento di corrente ed ottimizzare il loro controllo. Il circuito, quindi, può essere alimentato con un semplice alimentatore non stabilizzato da 500 o 1000 mA in grado

di erogare una tensione di 9 volt. Il connettore J7 serve per la programmazione In-Circuit ed è utile per chi volesse sperimentare firmware personalizzati.

Il circuito è predisposto per il collegamento alla porta seriale del PC dalla quale riceve i comandi per lo spostamento dei servo: il PIC deve solo interpretare i comandi in standard RS-232 e generare i segnali elettrici secondo il formato previsto per i servocontrolli. Il posizionamento è di tipo assoluto, nel senso che una volta fornite le coordinate di arrivo, ogni servo si muoverà esattamente su di esse con un errore davvero trascurabile. A lavoro finito, il circuito si presenta come nella foto del piano di montaggio. Il collegamento del circuito al robot deve essere fatto seguendo le indicazioni della figura 11. Alla prima accensione il PIC è predisposto per portare tutti i servo in posizione centrale. È opportuno, dopo aver alimentato la scheda, inserire uno alla volta i servo cercando di limitare per quanto possibile il brusco movimento iniziale che potrebbe procurare danni alla struttura meccanica ed ai servo stessi. Il programma del PIC prevede che alle successive accensioni i servo vengano mantenuti nell'ultima posizione nota prima dello spegnimento. Sul connettore J8 (PINZA) vengono connessi in parallelo i contatti dei due pulsanti utilizzati come sensori di presa. In questo modo anche se uno solo dei pulsanti ve-

Fig. 11



nisse premuto (conferma di avvenuta presa) il PIC se ne accorgerebbe comunque. La sensibilità nella presa è essenzialmente determinata dalla sensibilità dei pulsanti utilizzati, quindi è conveniente utilizzare dei pulsanti piuttosto “morbidi” in quanto il piccolo servo della pinza non dispone di molta forza.

Il firmware

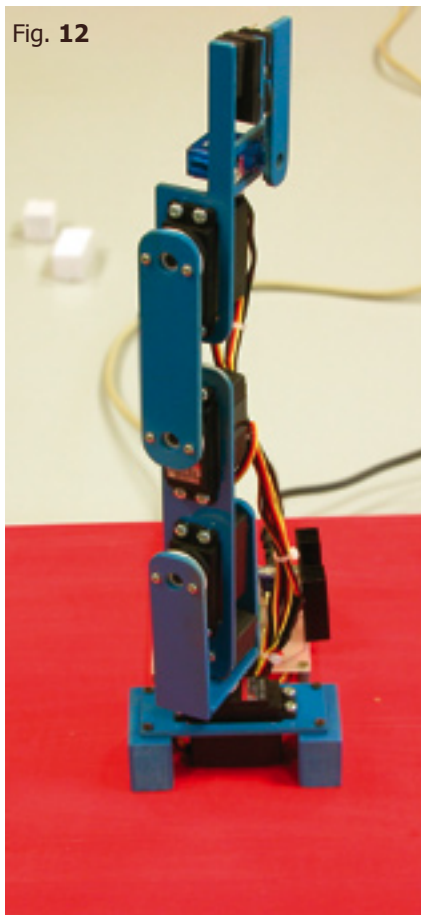
Il firmware del PIC è stato scritto in PicBasic sfruttando la funzione PULSOUT per generare il segnale di controllo dei servo. Questa funzione permette di generare un impulso di durata calcolata su uno dei pin specificati. Il cuore del firmware è rappresentato dal loop che gestisce

Listato 1

```
main
  pulsout sv1, pos1
  pulsout sv2, pos2
  pulsout sv3, pos3
  pulsout sv4, pos4
  pulsout sv5, pos5
  pause 15
  if RCIF == 1 then ricezione
  goto main
```

il controllo dei servo: le label SV1-SV5 rappresentano i rispettivi pin dai quali esce il segnale di comando dei servo, mentre le variabili pos1-pos5 contengono il valore relativo alla durata dell'impulso (posizione assoluta). Considerando una durata media di 1 msec per comando, è stata aggiunta l'istruzione pause 15 per fare in modo che i vari segnali vengano riprodotti ogni 20 msec, circa (figura 13). Il controllo del bit RCIF permette di verificare se sono giunti dei comandi dalla seriale al modulo USART del PIC. E' sufficiente modificare il valore delle variabili pos1÷pos5 affinché al successivo ciclo venga impostata la nuova posizione del servo altrimenti questo rimane immobile. La posizione iniziale di default del braccio è quella in cui tutti i servo si trovano in posizione centrale, considerata una possibile rotazione rispetto a quella iniziale di $\pm 90^\circ$. Dopo ogni comando la posizione dei servo viene me-

Fig. 12



danneggiare la parte meccanica. Se togliamo corrente ai servo, il peso del braccio stesso ne comporta la rotazione innaturale con possibile danneggiamento degli ingranaggi interni. La posizione più naturale è quella in cui i servo (per la pinza non ha importanza) sono in posizione centrale ed il braccio si trova in posizione verticale, come visibile in figura 12.

Come microprocessore è stato scelto il PIC 16F876 perché integra un modulo USART per la gestione delle comunicazioni seriali, caratteristica questa di fondamentale importanza perché dobbiamo eseguire la gestione dei servo in real-time e contemporaneamente gestire la ricezione seriale del PC. Per evitare movimenti involontari del braccio è importante generare con regolarità i cinque segnali PWM con la massima precisione e sincronizzare la lettura dei dati seriali dal PC. Per ottenere tutto ciò non è stata impiegata la funzione HSERIN del PicBasic ma una gestione basata sulla lettura dei flag dei registri associata al modulo USART (figura 13). La ricezione seriale dei dati avviene in modo asincrono rispetto al controllo dei servo, ma la lettura dei dati ricevuti dal modulo USART avviene in sincronismo. I dati ricevuti

morizzata nella memoria EEPROM perché in questo modo, se il braccio viene spento, alla successiva accensione i servo vengono riportati nella stessa posizione. E' importante che dopo lo spegnimento il braccio non venga mosso al fine di evitare bruschi movimenti che potrebbero

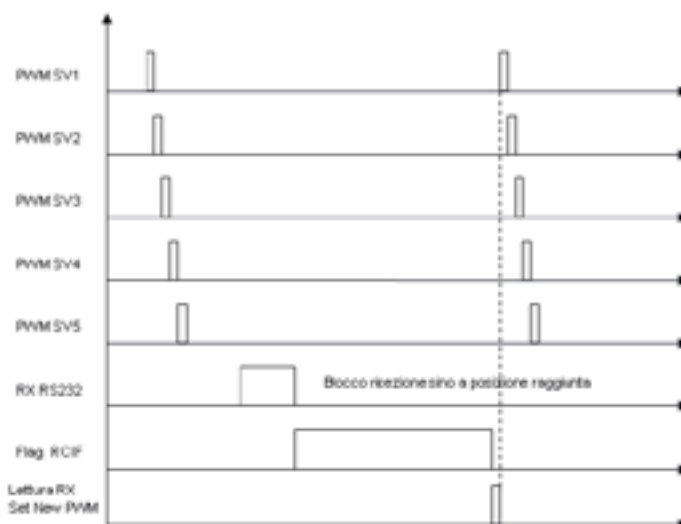


Fig. 13

Tabella 1

Carattere associato al 1° byte	Servo corrispondente nel firmware	Nome associato	Funzione
1	SV1	A	rotazione del braccio
2	SV2	B	movimento primo segmento
3	SV3	C	movimento secondo segmento
4	SV4	D	movimento terzo segmento
5	SV5	E	apre / chiude pinza

sono composti da due byte: il primo contiene il carattere ASCII relativo al numero associato al servo che si vuole muovere secondo la tabella 1; il secondo byte contiene invece un carattere il cui codice ASCII è il valore in centesimi di microsecondo relativi alla posizione da raggiungere. Se, ad esempio, inviamo la stringa "1d" indicheremo al PIC di muovere il servo N°1 (quello relativo alla rotazione) sino alla posizione corrispondente al segnale PWM di 1 msec ovvero una rotazione di 60° (circa) rispetto alla posizione centrale: infatti al carattere 'd' corrisponde il codice ASCII 100. Se, ad esempio, volessimo posizionare il servo C nella posizione centrale, dovremmo inviare la stringa "3û", in quanto il codice ASCII relativo alla U accentata è proprio 150. Vengono accettati tutti i valori esadecimali a partire da 70 e sino a 230 e corrispondenti ad una rotazione di $\pm 90^\circ$ con la posizione centrale corrispondente a 150. Solitamente la funzione di conversione da valore decimale a carattere ASCII viene fatta dal software di programmazione sia esso il Pascal, C, VisualBasic oppure Delphi: nel nostro caso questa funzione è svolta da un apposito software descritto nel capitolo successivo. La nuova posizione viene raggiunta portando gradualmente il valore del duty-cycle del PWM dal vecchio valore al nuovo con tutti i valori intermedi: così facendo si evitano movimenti bruschi che potrebbero essere pericolosi per la meccanica e

si rendono più fluidi gli azionamenti. La ricezione di nuovi dati seriali viene disabilitata sino a quando non viene raggiunta la nuova posizione. Il led verde posto sul circuito indica la corretta ricezione del comando che rimane acceso fino al completamento del movimento, periodo durante il quale non vengono accettati ulteriori comandi.

Il software

A lavoro terminato e quindi con la parte meccanica pronta ed il circuito elettronico montato e cablato, possiamo connettere il circuito alla porta seriale del PC tramite un normalissimo cavo seriale DB9 facilmente reperibile in qualsiasi negozio di elettronica o in un centro commerciale con reparto informatico. A questo punto lanciate il file BraRob.msi del programma denominato BRAROB,

scaricabile gratuitamente dal nostro sito insieme a tutti gli altri file necessari per la costruzione di questo circuito. Durante l'installazione verrà creata un'icona sul desktop per facilitare l'avvio del software la cui schermata iniziale si presenta come visibile in figura 14.

Per prima cosa impostate la porta di comunicazione utilizzata, COM1 oppure COM2. I comandi a disposizione prevedono lo spostamento assoluto di ogni singolo servo permettendo molteplici funzioni operative. Per inviare un comando di spostamento è sufficiente cliccare sopra un cursore e trascinarlo: il comando viene inviato quando si rilascia il pulsante del mouse nella posizione del cursore desiderata. Per piccoli spostamenti è possibile cliccare sulle freccette poste agli estremi degli slider. Il software fornisce anche l'indicazione della posizione impartita tramite una casella numerica posta sotto ogni slider ed indicante il tempo del duty-cycle del PWM in centesimi di millisecondo: al numero 150 corrisponde il valore di 1,5 msec (posizione centrale del servo). La chiusura e apertura della pinza avviene tramite la spunta del corrispondente comando. All'avvio del software tutti gli slider sono in posizione centrale anche se questo ➤

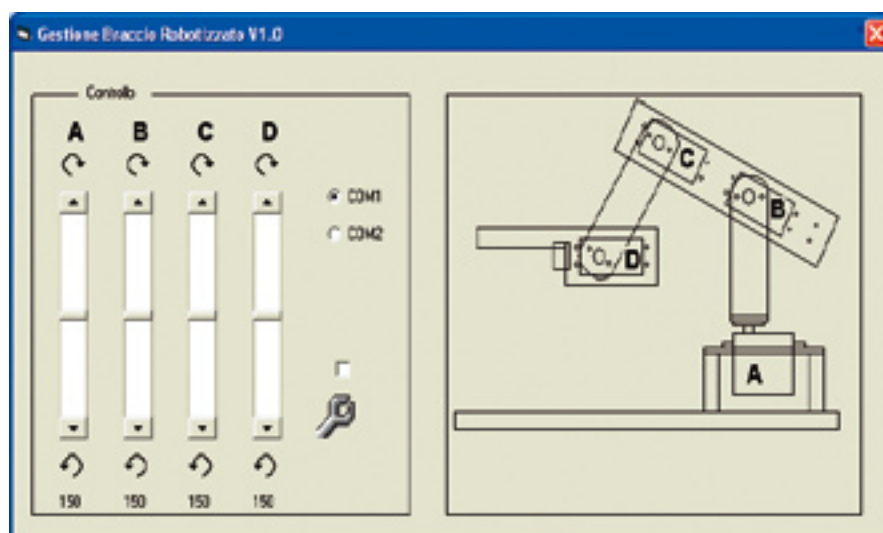


Fig. 14: Schermata principale del programma di controllo.

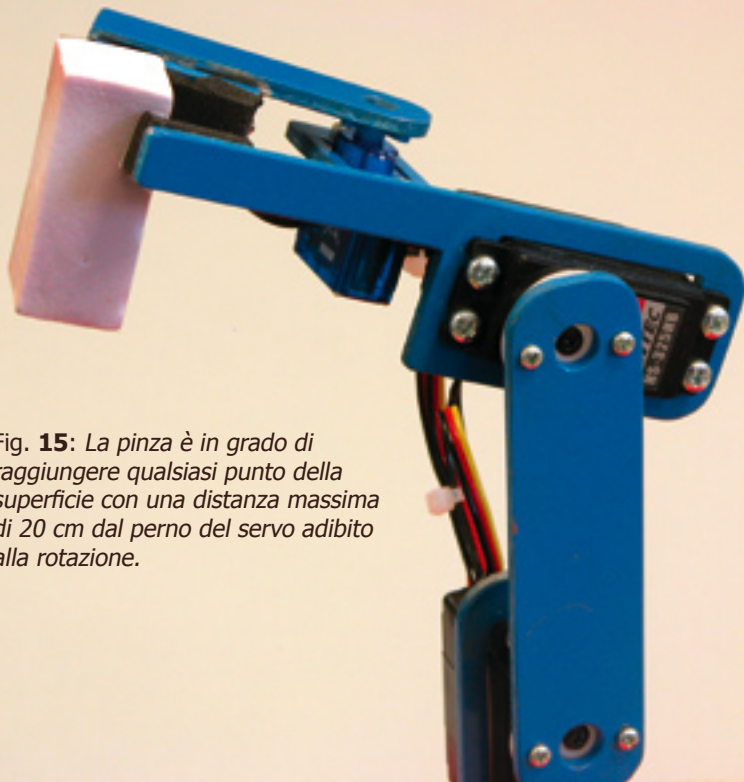


Fig. 15: La pinza è in grado di raggiungere qualsiasi punto della superficie con una distanza massima di 20 cm dal perno del servo adibito alla rotazione.

potrebbe non corrispondere alla posizione iniziale del braccio. Per fare le prime prove, consigliamo di realizzare dei piccoli oggetti di polistirolo o di cartone delle dimensioni di 2x2x4 cm che possano essere presi con facilità dalla pinza, come visibile in figura 15. I più esperti potranno realizzare un software specifico per funzioni più complesse, come ad esempio la possibilità di spostarsi in una determinata coordinata X, Y, Z. È anche possibile realizzare un programma per PC che possa eseguire alcuni movimenti prestabiliti secondo le proprie esigenze. Per fare questo è necessario l'invio di due byte tramite la porta seriale del PC, come descritto nel capitolo precedente. Il

protocollo di comunicazione è del tipo 9600 Baud, 8 bit di dati, 1 bit di stop, nessuna parità. Alcune prove possono essere eseguite anche tramite HyperTerminal di Windows che opportunamente configurato permette di gestire l'invio di stringhe tramite la porta COM1 o COM2.

Collaudo

Fissate il braccio ed il circuito su una base in legno di dimensioni adeguate: un multistrato di spessore 18 mm con dimensioni di 40x40 cm. Considerate che la pinza è in grado di raggiungere qualsiasi punto della superficie con una distanza massima di 20 cm dal perno del servo adibito alla rotazione. Il circuito stampato

deve essere fissato nelle vicinanze della base del braccio, tramite quattro piccoli distanziatori, con delle semplici viti da legno. Fate in modo che i cavi di collegamento tra i servo ed il circuito siano ben raccolti ma lasciateli liberi di piegarsi e di seguire senza sforzo i movimenti di ogni singola articolazione. Alla prima accensione la scheda è predisposta per inviare il segnale corrispondente alla posizione centrale dei servo. Questi potrebbero non trovarsi nella posizione centrale quindi collegateli uno alla volta sul rispettivo connettore del circuito di controllo già alimentato: questo permetterà di portare tutti i movimenti nella posizione centrale, corrispondente al braccio posto sulla verticale, come visibile in figura 12. A questo punto potete collegare la scheda alla seriale del PC e avviare il software per impartire i primi comandi di movimento. Quando decidete di spegnere il circuito, assicuratevi che il braccio possa rimanere in una posizione stabile in modo che alla prossima accensione si trovi nella stessa posizione in cui è stato spento. I servo sono molto robusti ma non indistruttibili, raccomandiamo quindi, di usare sempre molta cautela nell'azionare le varie articolazioni, cercando di evitare forzature eccessive che potrebbero danneggiare i delicati ingranaggi interni. Anche a circuito non alimentato evitate movimenti inutili o troppo bruschi dei servo. Buon divertimento!

per il

MATERIALE

Tutti i componenti utilizzati in questo progetto sono facilmente reperibili in commercio. I master dei circuiti stampati, i disegni dei vari elementi meccanici necessari alla costruzione del braccio nonché il software possono essere scaricati gratuitamente dal sito della rivista (www.elettronica.in.it). Il microprocessore programmato (cod. MF677) può essere acquistato al prezzo di 18,00 Euro, IVA compresa.

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

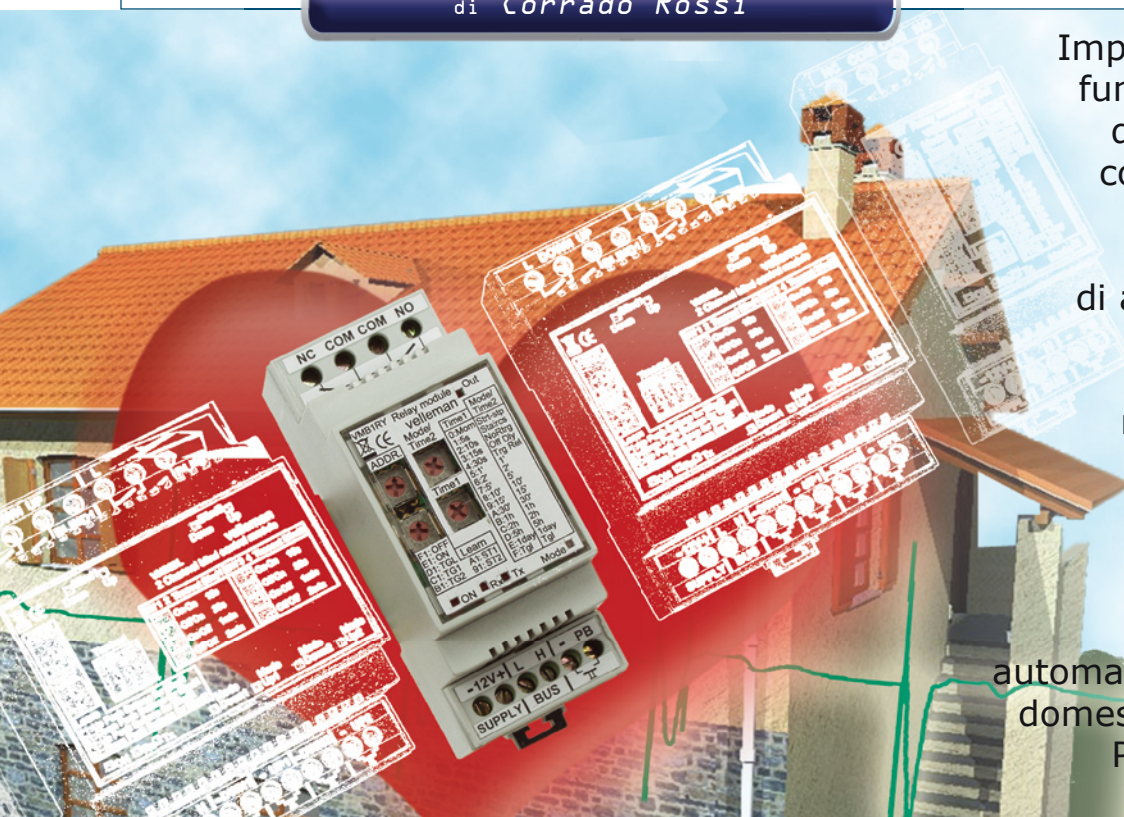
Tel: 0331-799775 ~ Fax: 0331-778112 ~ <http://www.futuranet.it>

Nuove funzioni per l'impianto domestico con Velbus

di Corrado Rossi

Implementiamo nuove funzioni nell'impianto domotico realizzato con i moduli Velbus.

L'aggiunta di un dimmer permette di acquisire maggiore confidenza con la programmazione manuale mentre le porte di comunicazione seriale e USB consentono di interagire con le automazioni dell'ambiente domestico utilizzando un Personal Computer.



Nel precedente articolo dedicato ai sistemi di automazione domestica Velbus abbiamo visto come può essere strutturato l'impianto elettrico di un ipotetico monolocale e quali funzioni evolute tali moduli consentono di ottenere. Partendo da un disegno topografico dell'appartamento, abbiamo posizionato nei vari punti di controllo e comando i dispositivi idonei ad aprire e chiudere le tapparelle, nonché per accendere e spegnere le luci, una alla volta o tutte insieme. I controlli evoluti (che possiamo anche definire "funzioni domotiche") sono rappresentati dalle seguenti azioni: accensione e

spegnimento contemporaneo di tutte le luci, apertura e chiusura simultanea delle tapparelle, spegnimento di tutte le luci e chiusura delle tapparelle con un solo comando. Ricordiamo che abbiamo utilizzato i moduli a relé VMB1RY e VMB4RY per la gestione delle luci, le unità di ingresso VMB6IN e VMB8PB per la gestione dei pulsanti, il modulo VMB2BL per il controllo tapparelle a due canali e, infine, l'alimentatore VMB3PS per fornire tensione a tutta la rete Velbus.

In questo articolo vediamo come aggiungere il dimmer VMB1DM per controllare un abat-jour o una qualsiasi



Fig. 1

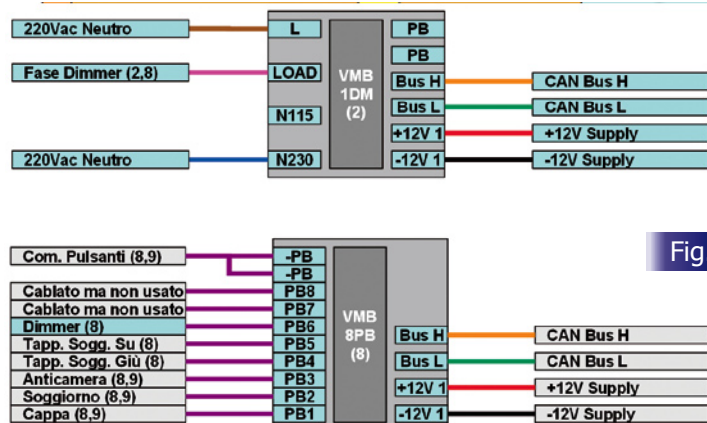
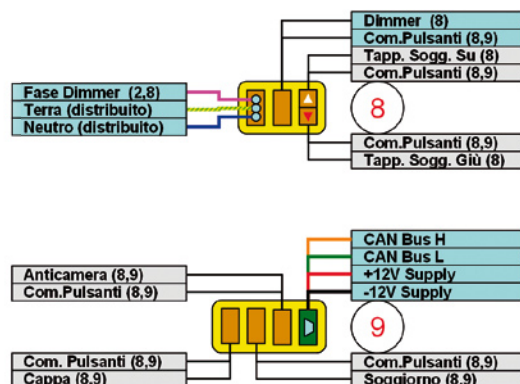


Fig. 2



Modifiche apportate all'impianto domotico analizzato nel precedente articolo.

altra lampada: gli assegneremo un indirizzo e lo assoceremo ai pulsanti che comandano le accensioni e gli spegnimenti in contemporanea di tutte le luci (funzioni domotiche), nonché al suo pulsante specifico, posizionato nel gruppo 8, eseguendo la procedura di memorizzazione manuale.

Successivamente collegheremo al bus l'interfaccia per PC; in par-

ticolare useremo la scheda USB VMB1USB, funzionalmente identica alla versione seriale VMB1RS. Installeremo quindi sul computer il programma di configurazione e controllo VelbusLink.

Aggiunte all'impianto

Si tratta di due aggiunte molto semplici dal punto di vista logistico. Sugeriamo comunque di recupera-

re il fascicolo precedente e di confrontare lo schema presentato allora con quello riportato nelle pagine seguenti: il raffronto permetterà di comprendere meglio il tipo di aggiunte che, ripetiamo, sono comunque piuttosto semplici.

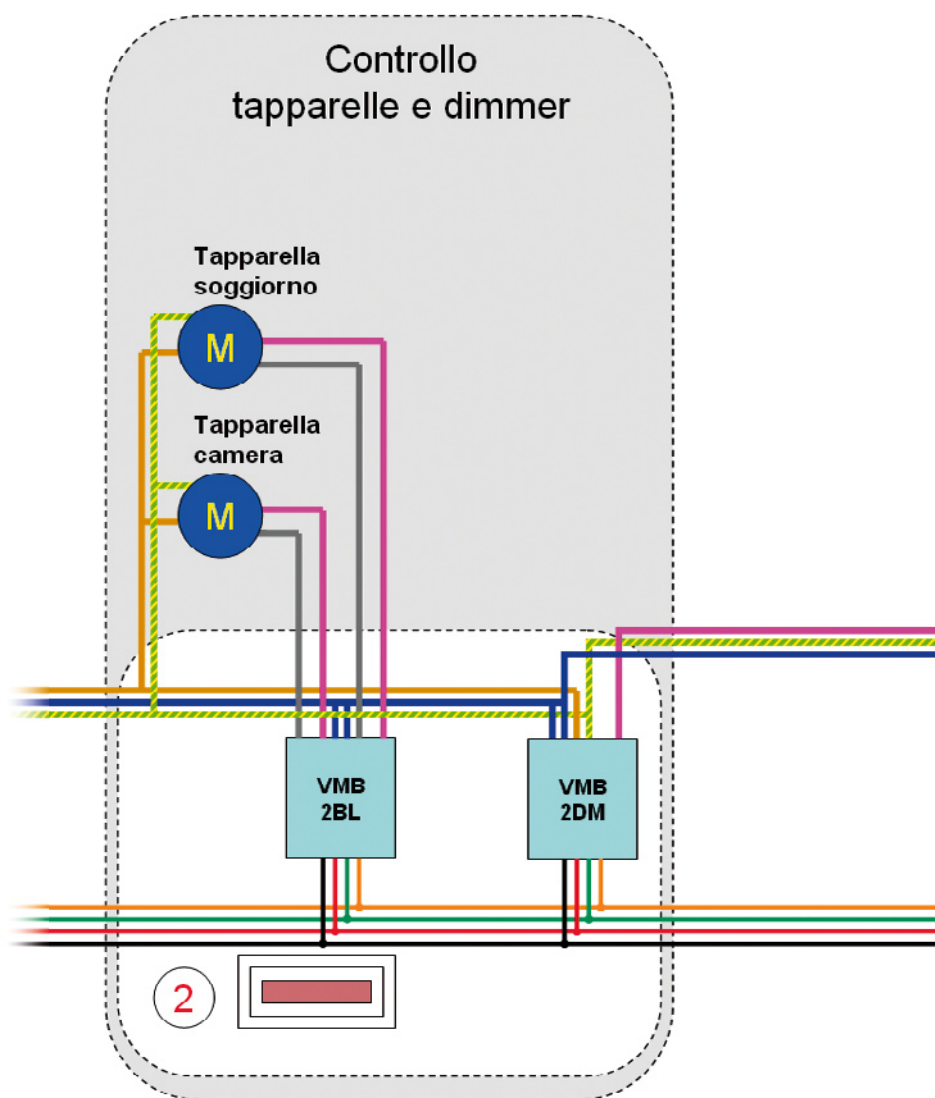
Osservando lo schema notiamo che il dimmer VMB1DM è stato inserito nel centralino secondario del gruppo 2 (vedere figura 1), dove è già ➤

montato il controllo tapparelle: non avremo quindi difficoltà a collegare i quattro fili del Velbus.

Il gruppo 8 (figura 2) ha invece subito cambiamenti più significativi: infatti, al controllo della tapparella in salotto sono stati affiancati il pulsante di attuazione del dimmer e la presa a cui si collega l'abat-jour; tale presa è anche stata collegata ai fili di neutro, terra e "fase controllata dal dimmer". A tale proposito ricordiamo che nel primo articolo avevamo detto che ci saremmo concentrati sulla parte comandi introdotta da Velbus, non preoccupandoci della parte di potenza (le prese), ed infatti non ne abbiamo parlato. In questo caso, però, avendo inserito un elemento di controllo per una lampada mobile (non fissata a muro), è stato necessario evidenziare anche l'aggiunta della presa a muro che abbiamo inserito nel gruppo 8. Ovviamente, in un impianto reale, tale presa si troverebbe in un'altra scatola portaprese murata, per tenere separate le linee di potenza dalle linee di comando (Velbus). Il pulsante di comando è stato collegato all'ingresso libero 5 della scheda di espansione ingressi VMB8PB inserita nello stesso gruppo 8 (vedere figura 2): ciò significa che il dimmer viene gestito solo attraverso comandi che viaggiano attraverso il bus.

Anche il gruppo 9 è stato leggermente modificato: notate che rispetto allo schema precedente, la posizione vuota è ora occupata dalla scheda di comunicazione VMB1USB, per cui è stato necessario portare i quattro fili più schermo del bus anche al gruppo 9 (figura 4, ellisse rosso dell'area giorno).

Riguardo a questa scheda, occorre precisare che essa è "nuda", ovvero priva di contenitore. Noi l'abbiamo rappresentata in verde e dimensionalmente equivalente ad un modulo pulsante o presa solo per renderla facilmente identificabile all'interno del gruppo 9. Ricordiamo che



la linea dati del CAN-Bus deve essere terminata con una resistenza: noi avevamo scelto di usare la terminazione presente sulla scheda d'espansione del gruppo 5 oppure nel controllo tapparelle del gruppo 2; essendo, dunque, già presente la resistenza di terminazione, dovremo lasciare aperto il ponticello TERM sia nel dimmer che nell'interfaccia VMB1USB. Se lo preferite, potete scegliere una terminazione differente, basta che sia chiuso un solo ponticello TERM. Giunti a questo punto, l'impianto domotico è stato modificato ed è pronto per essere configurato: per il dimmer eseguiremo una procedura manuale simile a quelle descritte il mese scorso.

Programmazione manuale del dimmer

Come primopasso, occorre assegnare al dimmer un indirizzo: poiché nella rete esistente l'ultimo usato era 06, con molta fantasia gli assegniamo il valore 07, da impostare sui suoi pre-selettori binari rotativi (esattamente come abbiamo fatto per tutti gli altri dispositivi Velbus). Ad impianto acceso il led di presenza tensione è illuminato e quelli di trasmissione e ricezione lampeggiano ogni volta che transitano dati sul bus. Il passo successivo consiste nel definire il modo operativo (dimmer, timer, dimmer con memoria...): anche se ai fini didattici un modo vale l'altro, scegliamo comunque la funzionalità

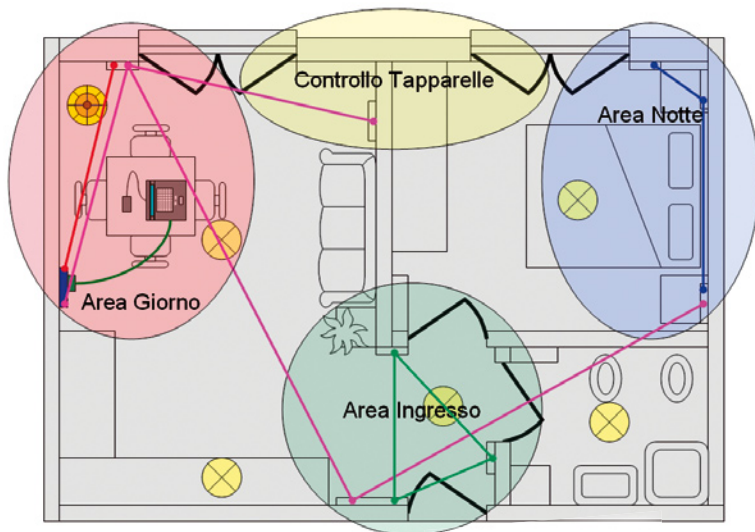


Fig. 4

riportata nel riquadro di pagina 41. Successivamente installate il programma VelbusLink lanciando il file eseguibile che avrete estratto dal file compresso presente nella cartella di servizio già usata.

La procedura è identica a quella che si segue per installare qualsiasi applicativo Windows, quindi diamo per scontato che abbiate portato a termine l'installazione seguendo tutte le richieste che il Sistema Operativo vi pone durante la sequenza. Giunti a questo punto, avete la rete Velbus programmata manualmente e sicuramente operativa, la scheda VMB1USB installata e riconosciuta, nonché il programma VelbusLink pronto per essere usato. Non resta che lanciare il programma e prendere confidenza con le varie finestre e con tutte le numerose opzioni che questo software prevede.

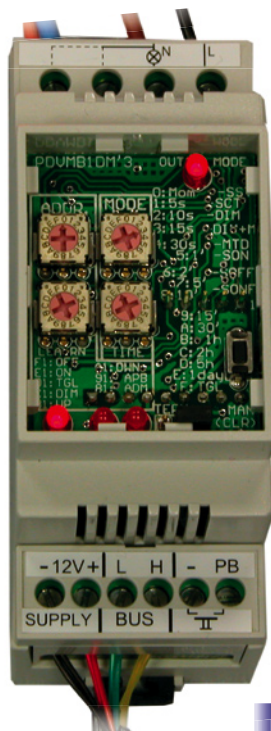


Fig. 5b

Usiamo il VelbusLink

Lanciamo il programma facendo doppio clic sull'icona che il file di installazione ha generato sul Desktop. Appare per qualche istante la finestra di presentazione visibile in Figura 6, quindi il programma apre la schermata principale, visibile in Figura 7.

Noterete che le opzioni di comando abilitate, sia dai menu a tendina che dai tasti di avvio rapido, sono veramente poche; tra queste è presente l'opzione *Connect*, disponibile sia da pulsante che da menu, nonché richiamabile da tastiera con la combi-

nazione Ctrl "O". Selezionate, dunque, *Connect* cliccando sul pulsante relativo: dalla finestra che appare ("RS232 Connection", Fig. 8) è possibile scegliere la COM da utilizzare per la connessione alla scheda (nel nostro caso si tratta della COM5, ma ovviamente potrebbe essere differente; fate sempre riferimento al riquadro di installazione del driver). Confermate la scelta facendo clic su *OK*.

Il programma entra ora in comunicazione con la rete Velbus e verifica innanzi tutto la presenza di dispositivi interrogando ogni singolo indirizzo: la scansione alla ricerca di dispositivi dura circa un minuto ed è accompagnata da un'indicazione grafica animata, posta alla destra dell'intestazione "Network".

Al termine della ricerca, sulla parte sinistra della schermata di avvio appare l'elenco dei dispositivi individuati, visualizzati nella classica disposizione ad albero, tipica dei sistemi Windows (Fig. 9).

Ogni elemento Velbus è identificato da un suo simbolo specifico, alla sinistra del quale è presente un quadratino contenente un "+".

La struttura ad albero del dispositivo viene espansa facendoci clic sopra: ora vengono mostrati i nomi assegnati ai suoi canali, sia in ingresso che in uscita; poiché fino ad ora i dispositivi sono stati programmati solo con procedura manuale, non possiamo vedere alcun nome associato a pulsanti e comandi e pertanto la struttura appare esattamente come visibile in Fig. 10.

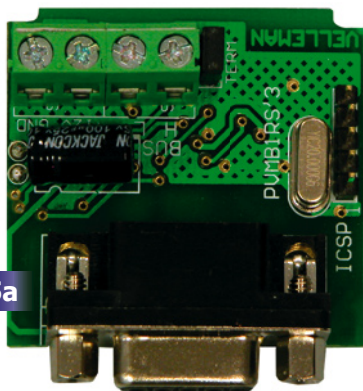


Fig. 5a

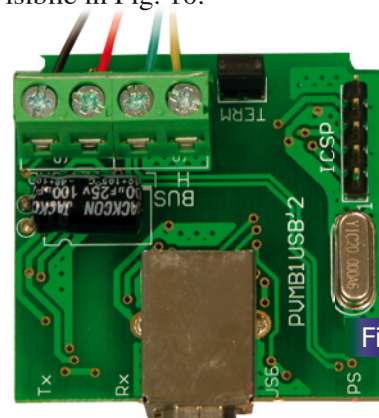
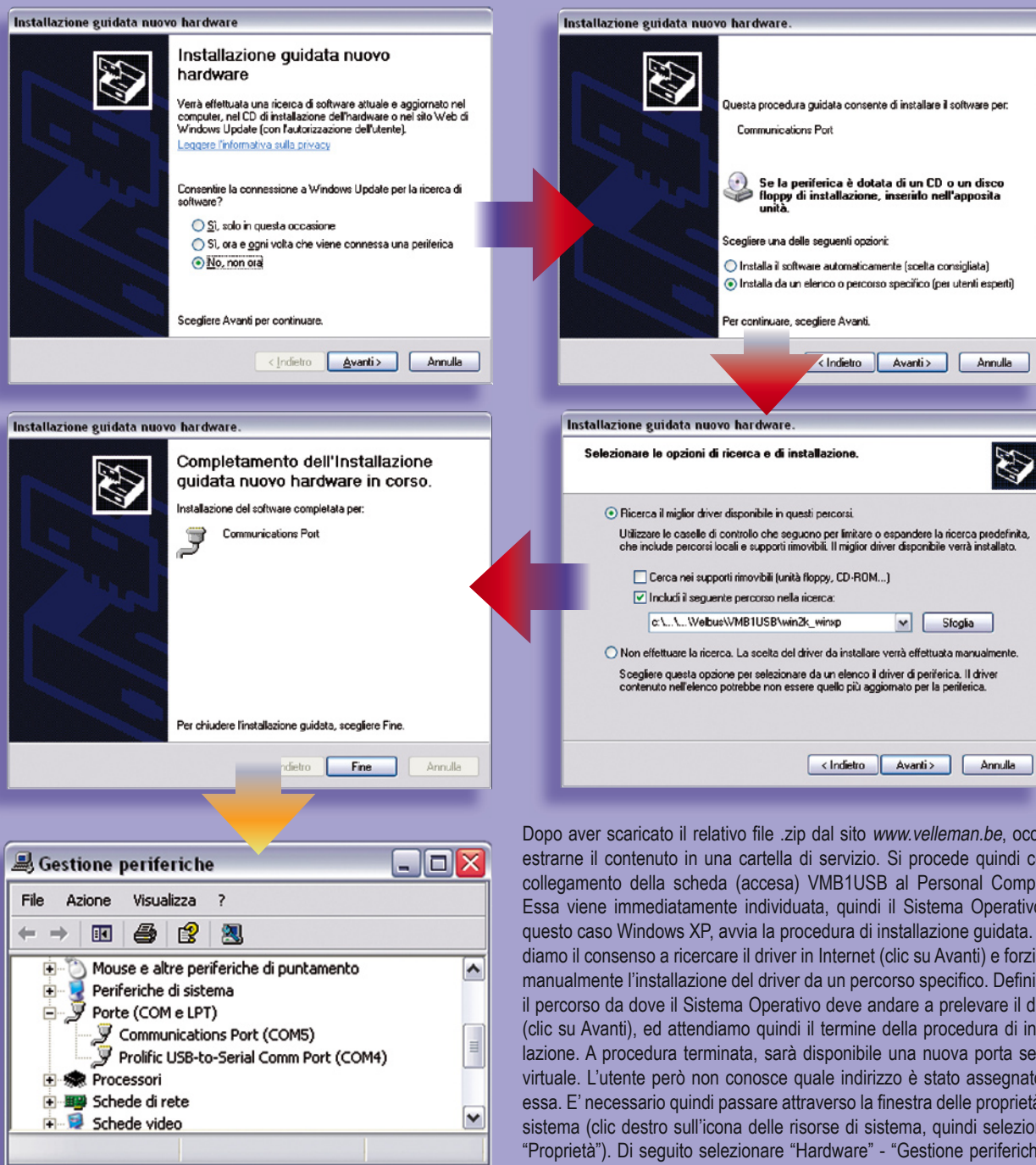


Fig. 5c

Installazione del Driver USB



Dopo aver scaricato il relativo file .zip dal sito www.velleman.be, occorre estrarne il contenuto in una cartella di servizio. Si procede quindi con il collegamento della scheda (accesa) VMB1USB al Personal Computer. Essa viene immediatamente individuata, quindi il Sistema Operativo, in questo caso Windows XP, avvia la procedura di installazione guidata. Non diamo il consenso a ricercare il driver in Internet (clic su Avanti) e forziamo manualmente l'installazione del driver da un percorso specifico. Definiamo il percorso da dove il Sistema Operativo deve andare a prelevare il driver (clic su Avanti), ed attendiamo quindi il termine della procedura di installazione. A procedura terminata, sarà disponibile una nuova porta seriale virtuale. L'utente però non conosce quale indirizzo è stato assegnato ad essa. E' necessario quindi passare attraverso la finestra delle proprietà del sistema (clic destro sull'icona delle risorse di sistema, quindi selezionare "Proprietà"). Di seguito selezionare "Hardware" - "Gestione periferiche" e "Porte" per identificare quale Com Port virtuale è stata creata per gestire la scheda VMB1USB: nel nostro caso è la COM5.

Col VelbusLink è possibile associare i nomi ai vari pulsanti e canali, cosa che andremo a fare subito; in questo modo gli accoppiamenti tra dispositivi saranno molto più comprensibili e immediati.

Assegnazione dei nomi e programmazione

Per assegnare i nomi, potete parti-

re da qualsiasi dispositivo, dato che il programma lo permette. Noi abbiamo scelto di iniziare dai pulsanti ma nulla vieta di incominciare, ad esempio, dagli attuatori. Si tratta di una scelta soggettiva e, dal punto di vista pratico, una cosa vale l'altra. Volendo fare la nostra stessa scelta selezionate dunque da menu *Tools* e poi *Module Memory Edi-*

tor (Fig. 11): appare la finestra di Figura 12 nella quale selezionate il modulo TVMB6IN, corrispondente al gruppo pulsanti 1. Ora cliccate su *Retrieve* per fare in modo che il programma legga la configurazione direttamente dalla memoria del modulo in esame: dopo aver risposto con *OK* alla richiesta di lettura della memoria del modulo (Figura 13), ➤

Fig. 6

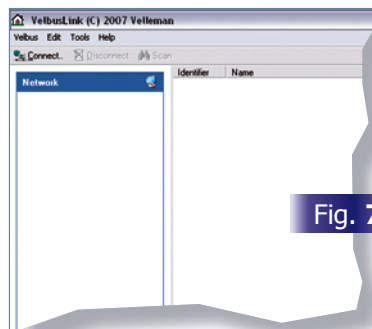


Fig. 7

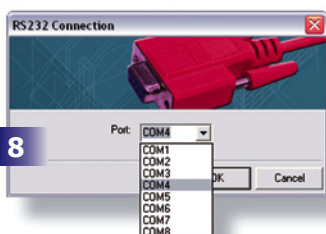


Fig. 8

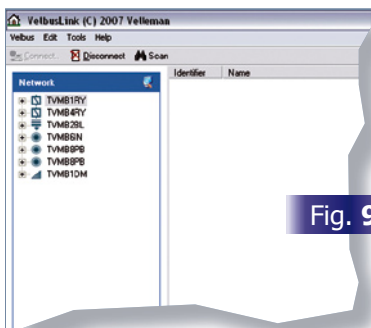


Fig. 9

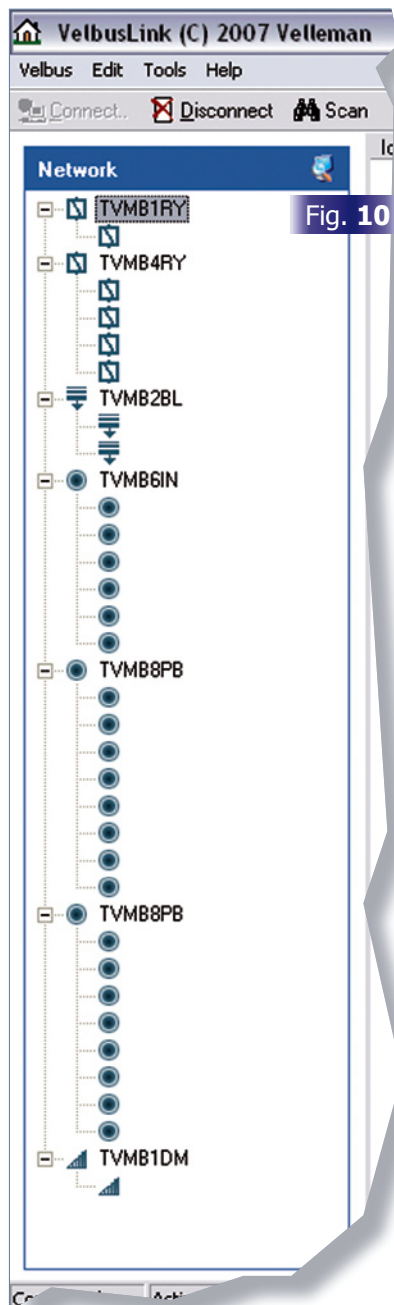


Fig. 10

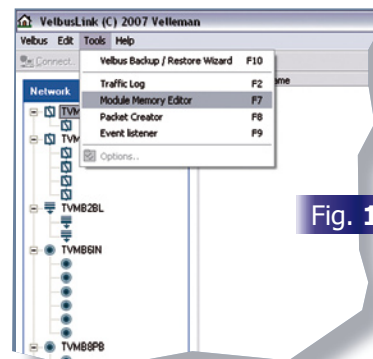


Fig. 11

di 16 caratteri e la procedura è molto semplice:

- 1) selezionate l'ingresso sulla parte sinistra della finestra di configurazione;
- 2) digitate nel campo a destra il nome da assegnare all'ingresso (figura 17);
- 3) fate clic su *Commit* e quindi scegliete *OK* sulla finestra di conferma (figura 18). Occorre scegliere l'opzione *Commit* ogni volta che si vuole memorizzare una modifica.

Terminata la configurazione, chiudete la finestra e fate clic su *Commit* nella finestra di Figura 12.

Ora il programma chiede conferma sulla programmazione del modulo in quanto, una volta avviata, non può essere annullata; dopo la conferma una barra di stato segnala il trasferimento dei dati e, successivamente, gli stessi dati vengono letti e scaricati dalla memoria del modulo per verificare che la programmazione sia andata a buon fine (figura 15).

Nel caso ci sia qualche problema appare una segnalazione di errore. Ovviamente il programma consente di modificare ogni singolo parametro di configurazione, sia che si tratti di un'impostazione molto semplice (come nel caso di un'espansione ingressi) che molto complessa (come nel caso del comando relé a quattro canali). La caratteristica fondamentale del programma VelbusLink consiste nella possibilità di configurare completamente qualsiasi modulo, compreso l'accoppiamento di pulsanti e funzioni. In Figura 20 riportiamo un ulteriore esempio delle

appare la schermata di Figura 14 che evidenzia la procedura di scarico dei dati; al termine della lettura del dispositivo, selezionate *Configure*. Appare ora la schermata visibile in Figura 16: da questa finestra è possibile accedere alla configurazione di tutto il gruppo pulsanti. Per quanto ogni finestra di configurazione sia specifica per il modulo da configurare, esistono comunque tre elementi comuni a tutte le finestre di configurazione: la "combo-box" in alto a sinistra, con cui si selezionano i parametri da modificare, l'area di

testo a sinistra, che contiene l'elenco dei parametri, e l'area di testo a destra in cui si definiscono le variazioni da effettuare. Poiché lo scopo di questa sezione è di far prendere confidenza con l'interfacciamento al Velbus tramite PC, non tratteremo nel dettaglio ogni singola finestra; ci occuperemo invece di chiarire nel modo più completo possibile le procedure da seguire.

A questo punto possiamo iniziare a definire i nomi da assegnare ai vari ingressi di comando associati al VMB6IN. La lunghezza massima è

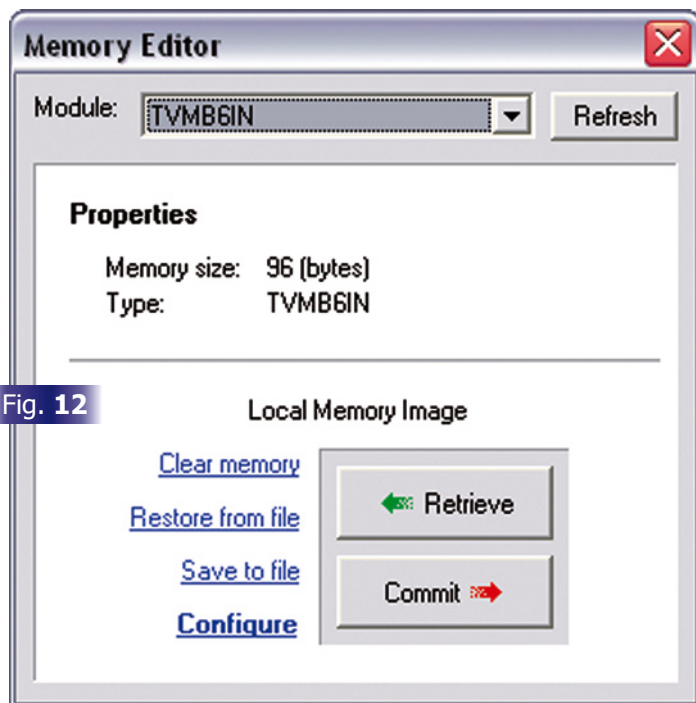


Fig. 12

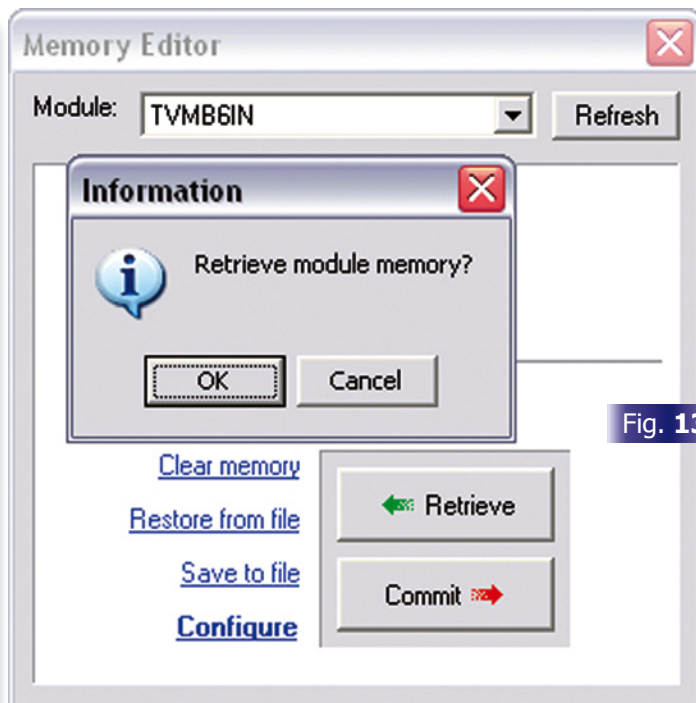


Fig. 13

possibilità offerte dal VelbusLink, presentando parte della configurazione del modulo a relé a singolo canale: alla funzione *Clear* (disattivazione del relé) sono associati i pulsanti di spegnimento di tutte le luci e di spegnimento e chiusura totale che avevamo impostato manualmente e, successivamente, nominato tramite PC. Volendo accoppiare un altro pulsante di comando, è sufficiente

selezionarlo dalla finestra *Available buttons* e trascinarlo con il tasto sinistro del mouse premuto nella finestra *Registered buttons*, quindi fare clic su *Commit* secondo la procedura già vista in precedenza. In figura 19 potete vedere la struttura ad albero del nostro impianto demo, così come appare dopo aver portato a termine l'assegnazione dei nomi sia degli ingressi che delle uscite.

Il Programma VelbusLink offre anche la possibilità di controllare il traffico dati in rete (figura 21), funzione utile soprattutto quando si vogliono sviluppare delle applicazioni personalizzate per interagire con i moduli Velbus; in questa sede ci limitiamo ad accennare a tale possibilità rinviando alla prossima puntata per un approfondimento dell'argomento. Il prossimo mese ci occupa-

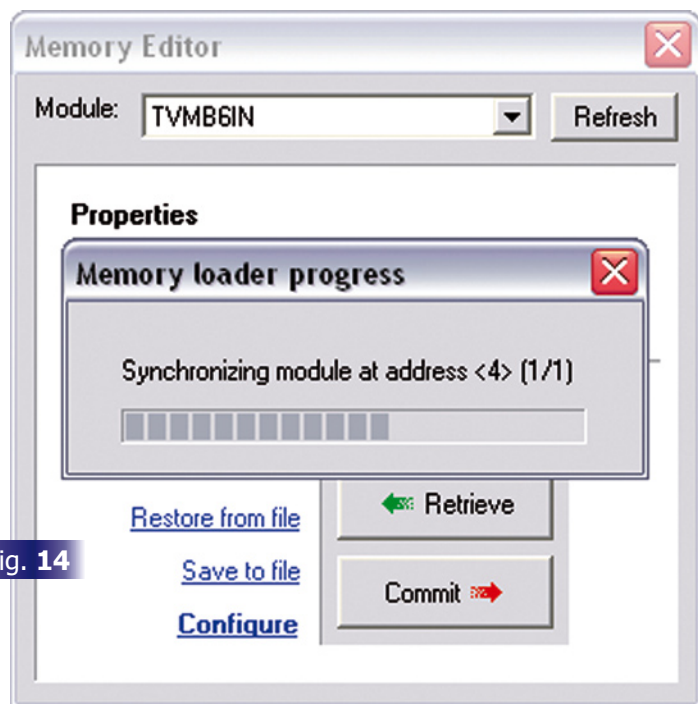


Fig. 14



Fig. 15

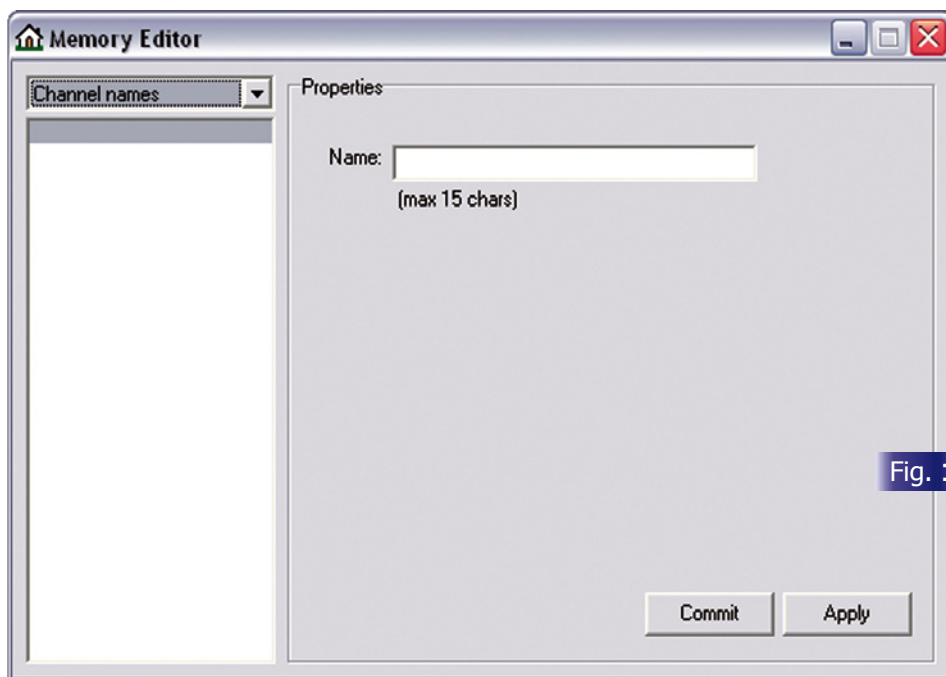


Fig. 16

porto allo sviluppo di applicazioni domotiche personalizzate offrendo differenti funzionalità, molto utili e potenti.

Il programma dispone di un efficiente “wizard di backup e restore” ovvero di una procedura guidata con la quale effettuare sia una copia su hard disk di tutta la configurazione della rete (backup) che una riprogrammazione simultanea di tutti i dispositivi (restore) partendo dalla copia di backup.

Abbiamo dedicato al wizard uno specifico riquadro, per non appesantire troppo la trattazione dell'argomento. In questa sede

remo in dettaglio del protocollo Velbus, premessa indispensabile per tentare di realizzare applicazioni personalizzate sia software che hardware.

Altre funzioni

VelbusLink nasce sia come programma di configurazione assistita da computer (essenzialmente per rendere più facile il compito agli installatori) sia come sup-

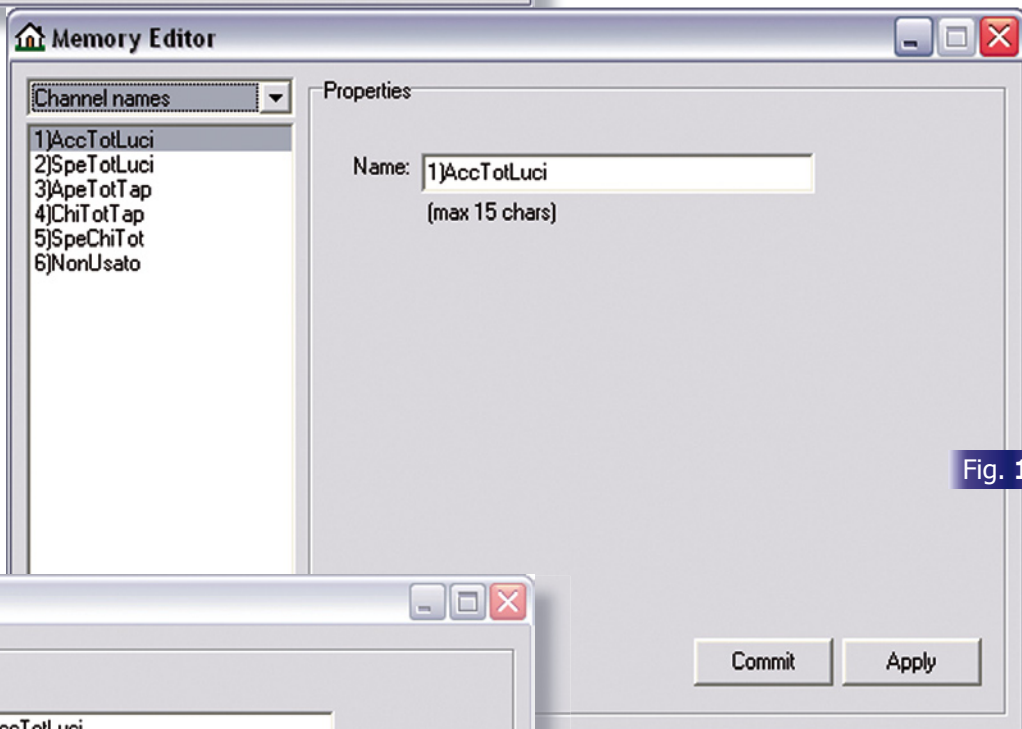


Fig. 17

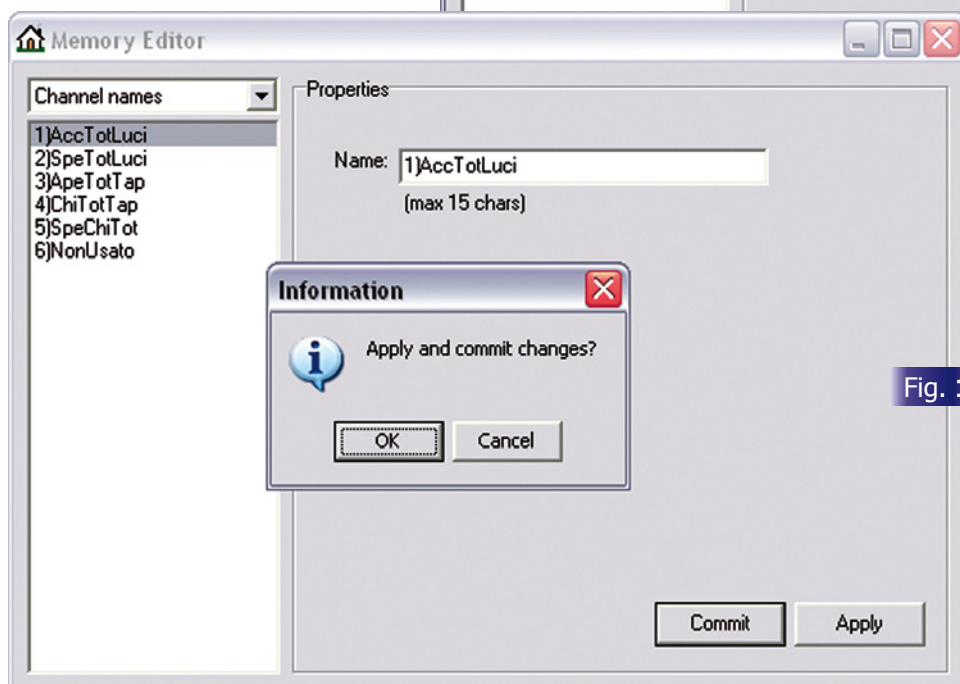


Fig. 18

vogliamo sottolineare l'utilità di questa procedura, specie in presenza di anomalie di sistema dovuti a fattori esterni quali, ad esempio, la mancanza di alimentazione durante la fase di programmazione. Anche la funzione *Packet Creator*, accessibile dal menu *Tools* oppure con il tasto di funzione F8, è particolarmente utile ed interessante: essa permette infatti di costruire pacchetti dati già formattati sia come lunghezza che come indiriz-

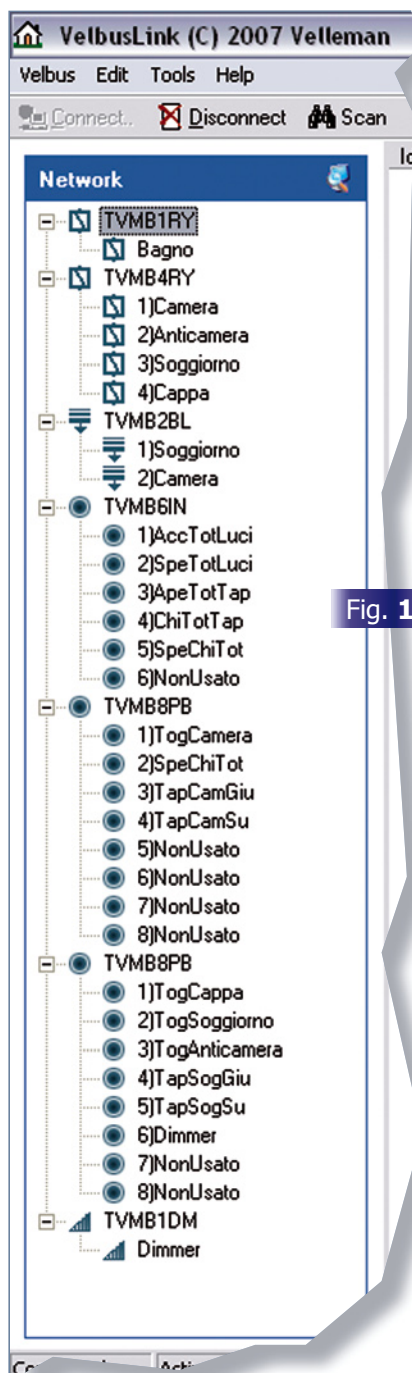


Fig. 19

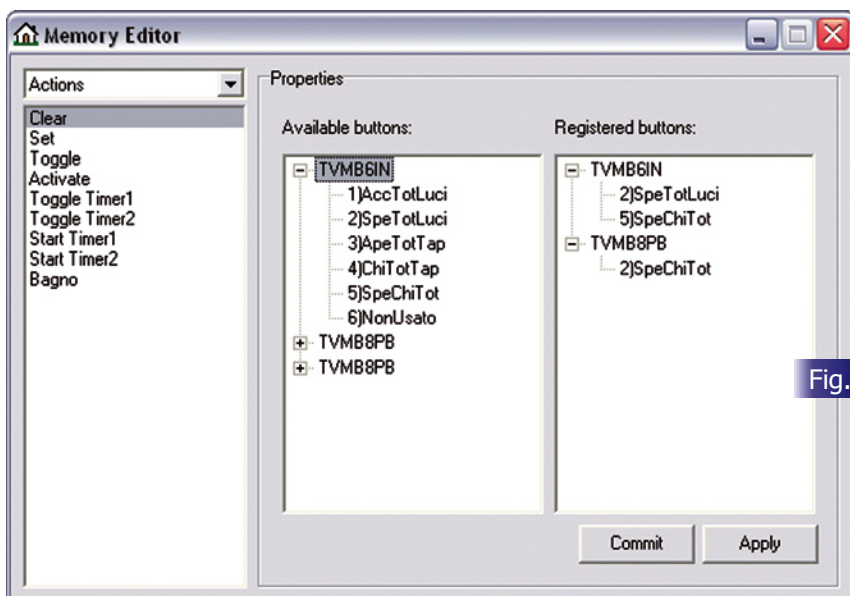


Fig. 20

zo e priorità: tramite questa utility lo sviluppatore che sta studiando il protocollo e realizzando un pro-

prio applicativo ha la possibilità di verificare, in tempo reale, se le stringhe di comando generate dal >

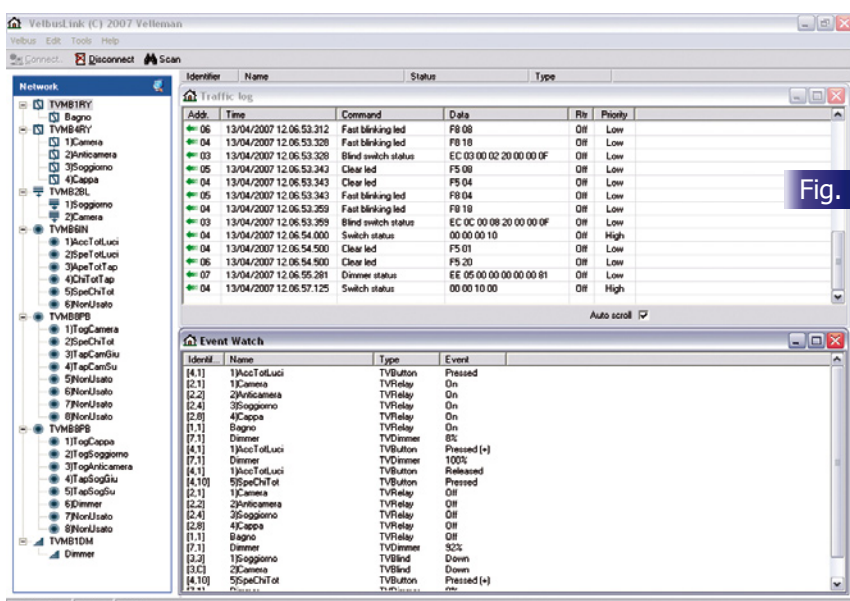


Fig. 21

RM ELETTRONICA SNC

vendita componenti elettronici

rivenditore autorizzato:

**FUTURA
ELETTRONICA**

ELETTRONICA

G.P.E.

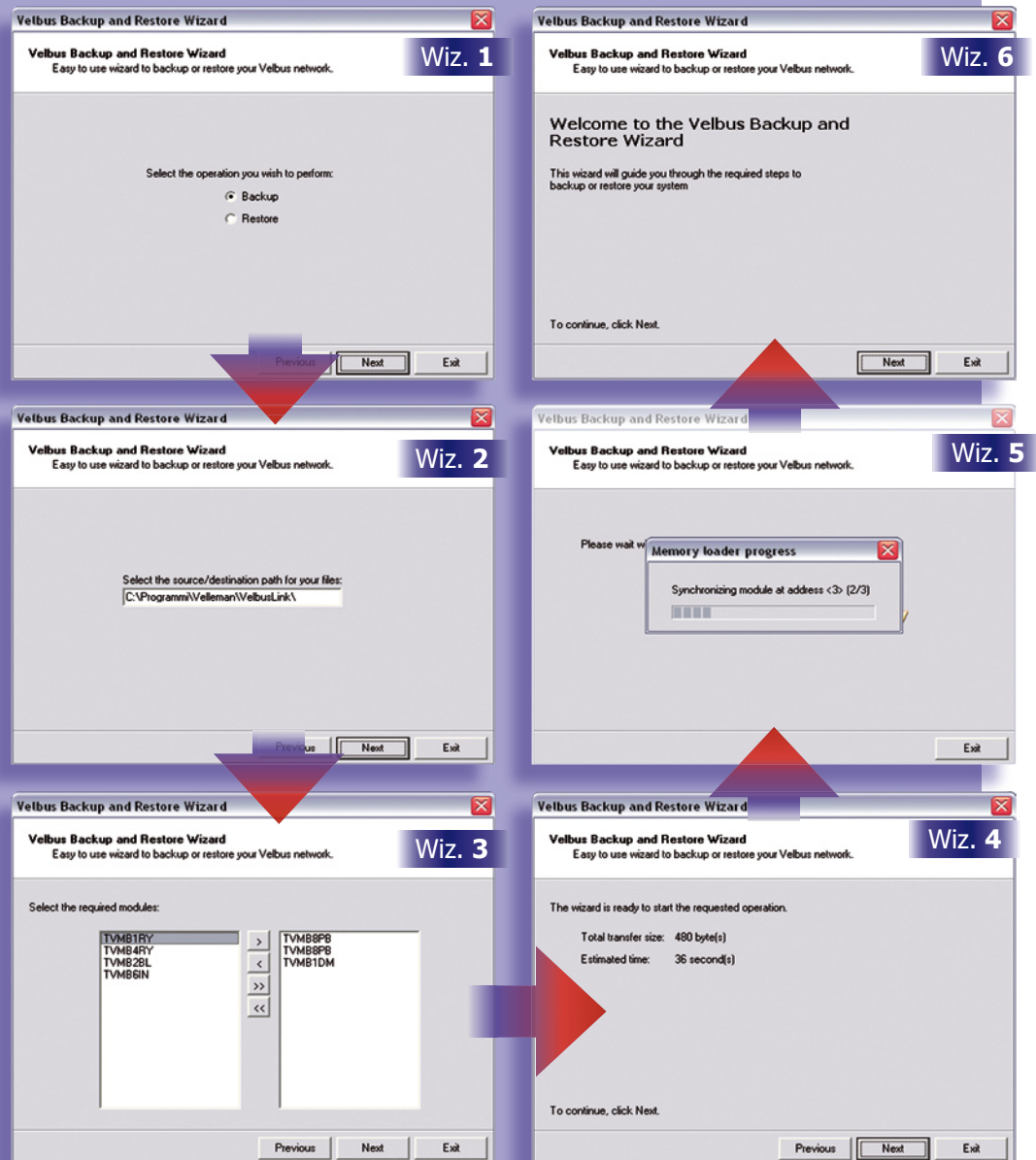
**ELSE
Kit**

Via Val Sillaro, 38 - 00141 ROMA - tel. 06/8104753

Questa procedura consente di salvare la configurazione del sistema Velbus nel disco rigido del PC per poi ripristinarla, a fronte di problemi di varia natura che abbiano fatto perdere la configurazione a uno o più moduli. Si avvia premendo F10 oppure selezionando "Tools/Velbus Backup-Restore Wizard". Appare la finestra visibile in Wiz.1, in cui viene chiesto se si intende procedere con un Backup o un Restore. Nel nostro caso abbiamo optato per un Backup. Facendo clic su "Next", appare la schermata Wiz.2, in cui il programma chiede conferma della directory dove salvare i file. Si procede ancora una volta facendo clic su "Next", passando quindi alla schermata Wiz.3 in cui si selezionano i moduli di cui si intende salvare la configurazione. Quelli non selezionati si trovano a sinistra, mentre quelli selezionati stanno a destra. È possibile selezionarli e deselectarli sia individualmente (tasti "<" e ">") che in gruppo (CNTRL o SHIFT clic sinistro di Windows) che tutti insieme (tasti "<<" e ">>"). Effettuata la selezione, si procede ancora con "Next" per entrare nella schermata Wiz.4, in cui il programma esegue il calcolo della dimensione totale del file transfer e si prepara ad effettuare lo scarico dei dati. Si preme ancora una volta

il tasto "Next" per avviare la lettura delle configurazioni (Wiz.5). Il messaggio visibile in Wiz.6 appare al termine della procedura che, a questo punto, può essere chiusa. Volendo effettuare un Restore, la procedura è molto simile: l'unica differenza significativa si ha nella finestra

Wiz.5, in cui la barra di stato viene mostrata due volte per ogni dispositivo: la prima volta si riferisce al trasferimento della configurazione, mentre la seconda volta è relativa alla verifica della programmazione che, se errata, genera un messaggio d'allarme.



programma in preparazione, una volta copiate nella finestra *Packet creator*, sortiscono l'effetto desiderato. Vogliamo infine segnalare che il programma VelbusLink è in continua evoluzione; per questo motivo quanto riportato in questo articolo (basato sulla versione 4.4 del pro-

gramma) potrebbe non corrispondere esattamente con le funzionalità offerte da versioni più recenti del software. In ogni caso non preoccupatevi più di tanto: potrà cambiare leggermente la presentazione grafica, potranno essere introdotte nuove funzionalità, e potranno anche

essere gestiti nuovi prodotti Velbus ma le funzionalità descritte di base - di cui ci siamo occupati in questo articolo - non cambieranno mai. Appuntamento, dunque, al prossimo numero nel quale analizzeremo il protocollo e tenteremo di realizzare la nostra prima applicazione.

Induttanzimetro e capacimetro digitale

dell'ing. Massimo Del Fedele

Strumento di misura dalle dimensioni molto contenute alimentato a batteria. Gestito completamente da un microprocessore, è in grado di compensare le capacità e le induttanze parassite tramite la procedura di autozero, permettendo così di misurare con maggiore precisione il valore dei componenti.

Una delle esigenze più sentite da chi, per hobby o per professione, deve metter mano ad un circuito elettronico per ripararlo, migliorarlo o anche solo studiarlo, consiste nel determinare il valore dei componenti utilizzati. Nel caso delle resistenze, risalire al valore è abbastanza semplice dato che esso viene indicato in modo univoco da un codice composto da barre colorate, valido anche in quelle di precisione che, a volte, riportano addirittura il valore stampato in chiaro sul corpo. Nel caso il codice non sia leggibile, basta un semplice tester per risalire al valore. Le cose si complicano un

po' con i condensatori, in particolare con quelli di basso valore, essendo il metodo di codifica più complesso. Nei condensatori ceramici, poi, spesso col tempo le scritte si cancellano, rendendo praticamente impossibile determinarne il valore. La problematica è simile anche nel caso delle induttanze. Risalire ai valori dei componenti in questi ultimi due casi è praticamente impossibile senza l'ausilio di uno strumento di misura adeguato: basti pensare alle bobine per alta frequenza, costituite da poche spire avvolte in aria, in cui il calcolo teorico dell'induttanza, già difficile così, diventa quasi impro- ➤

Misure di induttanza e capacità

La misura di un valore di induttanza o capacità può essere effettuata in svariati modi. Si potrebbe pensare, ad esempio, di sfruttare la reattanza del componente: dalle reminiscenze di elettrotecnica, essa rappresenta la "resistenza" equivalente del componente alla corrente alternata di una determinata frequenza. Si ha, per il condensatore

$$X_c = \frac{1}{2\pi FC}$$

in cui X_c è il valore della reattanza, F la frequenza della corrente alternata applicata e C la capacità.

Immaginiamo di inserire il condensatore in un partitore alimentato con una tensione alternata di frequenza e tensione noti, come nello schema in alto a destra.

Misurando V_c , si può risalire alla reattanza X_c secondo la seguente relazione

$$V_c = V_i \frac{X_c}{RX_c} \Rightarrow X_c = \frac{RX_c}{V_i - V_c}$$

e da questa alla capacità C incognita

$$C = \frac{1}{2\pi FX_c}$$

Un altro metodo utilizzato spesso nei capacitometri di precisione consiste nel misurare il tempo di carica e scarica di un circuito RC. Questo ha un andamento esponenziale come da figura in basso a sinistra.

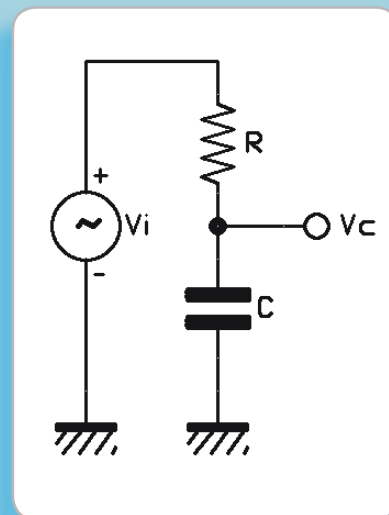
L'equazione matematica che descrive l'andamento della carica è la seguente

$$V(t) = V_i \left(1 - e^{-\frac{t}{RC}}\right)$$

in cui RC (nota anche come τ) è la costante di tempo del circuito costituito da una resi-

stenza di valore R in serie di un condensatore di valore C . Fissati i valori di V_i e $V(t)$, nonché il valore della resistenza in serie R , misurando il tempo " t " di carica è possibile risalire al valore di capacità C . Nella pratica la cosa si complica leggermente, essendo necessario, per ottenere una buona precisione, una misura sia del ciclo di carica che di scarica, in modo da eliminare il più possibile gli errori relativi alla misura delle tensioni.

Questo procedimento, matematicamente più complesso del precedente, si presta benissimo alla misura di condensatori di elevata capacità, garantendo ottima precisione; per quelli di piccola capacità è impreciso, anche se in questo caso le difficoltà di misura sono dovute ai tempi troppo brevi invece che alle tensioni troppo basse. Nel caso delle impedenze, se il valore è troppo elevato il circuito diventa sensibile ai disturbi esterni. Fin qui abbiamo elencato i procedimenti con i quali non misureremo i nostri componenti... Cosa rimane a disposizione? Semplicemente, la misura di una frequenza prodotta da un oscillatore LC. Il procedimento, già ad un primo



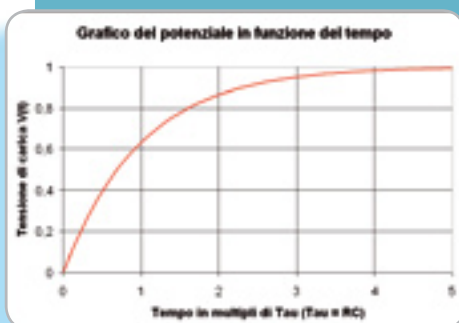
a limitare il campo di valori misurabili.

Il circuito richiede inoltre la misura di una tensione alternata V_c con una buona precisione, compito anch'esso non facile, sempre tenendo conto dell'utilizzo a cui è destinato lo strumento.

esame, presenta parecchi vantaggi che elenchiamo di seguito:

- La misura di una frequenza è un'operazione semplicissima e di notevole precisione, sfruttando i moderni microcontrollori, quali i PIC.
- In un oscillatore LC ben progettato è possibile rendere la frequenza in uscita scarsamente dipendente dalla tensione di alimentazione, e comunque questa è stabilizzabile in modo molto semplice.
- Un oscillatore LC è realizzabile con pochi ed economici componenti.

E gli svantaggi? In pratica, ne esiste uno solo: affinché l'oscillatore abbia una buona stabilità, il circuito oscillante deve avere un ottimo fattore di merito Q , il che limita il campo di impiego a componenti di valore medio/basso; ritorneremo comunque in seguito su questo argomento, dedicando ad esso un capitolo specifico.



ponibile in presenza di un nucleo di materiale sconosciuto. Se queste considerazioni sono vere per i componenti tradizionali, lo sono ancor di più per quelli a montaggio superficiale. Mentre le resistenze hanno il valore serigrafato sul corpo e ben

visibile (a patto di avere a disposizione una lente o una vista da falco), per quanto riguarda condensatori ed induttori... nulla. Essi semplicemente "non" sono marchiati, essendo la cosa molto complessa a causa delle ridotte dimensioni. In questo caso

uno strumento di misura potrebbe risultare molto comodo. Ma non è tutto: tale strumento sarebbe utile anche per selezionare componenti con valori quanto più possibile simili tra loro. Spesso, infatti, nei circuiti elettronici è importante un'elevata

simmetria, che può essere ottenuta solamente con componenti di valore molto simile, piuttosto che un'elevata precisione in assoluto.

Per tutte le ragioni sopra elencate, uno strumento in grado di misurare con buona precisione sia i condensatori che le induttanze e, perchè no, anche economico, sarebbe molto utile. Per questo motivo abbiamo deciso di proporre il progetto di un apparecchio ottimizzato per la misura di bassi valori di capacità (da 0,5 pF ad un massimo circa 0,5 - 1 µF) e di induttanza (da pochi nanoHenry a qualche centinaio di microHenry o pochi milliHenry). Questo strumento, perciò, non è adatto né per misurare le capacità di grossi condensatori elettrolitici, né le induttanze dei trasformatori o degli elementi induttivi usati nei filtri per casse acustiche. Si tratta dunque di uno strumento dedicato principalmente a chi lavora in media ed alta frequenza. In un prossimo articolo, invece, proporremo un apparecchio specifico per esigenze diametralmente opposte, cioè la misura di grandi valori di induttanza e capacità.

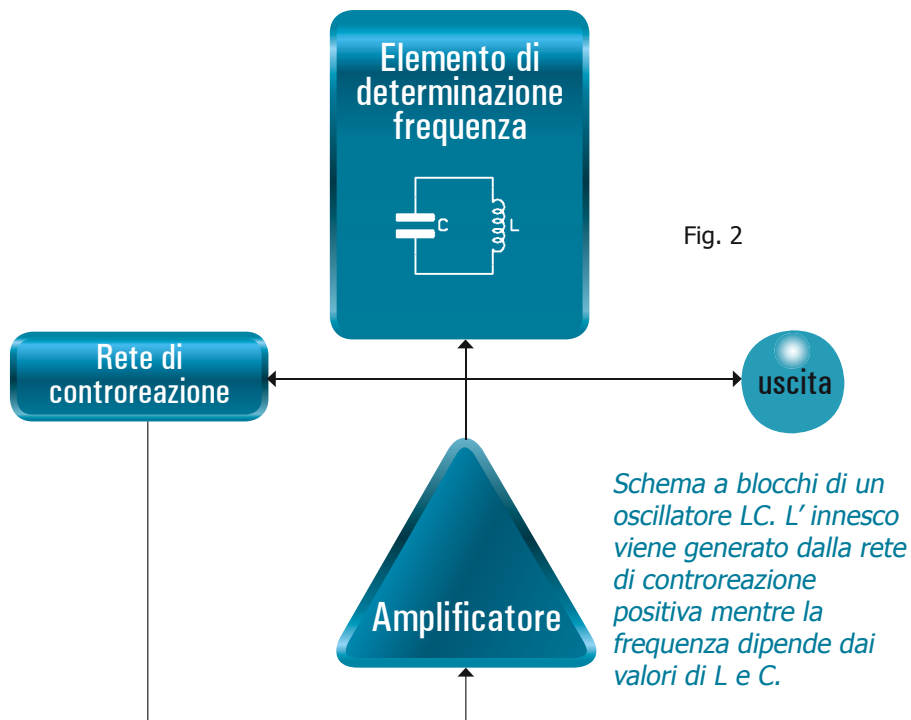
Scelta del tipo di oscillatore

Esistono innumerevoli tipologie di oscillatori LC, ognuno con specifici vantaggi e svantaggi; quelli più noti il Colpitts e l'Hartley, giusto per citarne un paio. Nelle figure 1a e 1b ne presentiamo lo schema semplificato. Nel primo schema i compo-

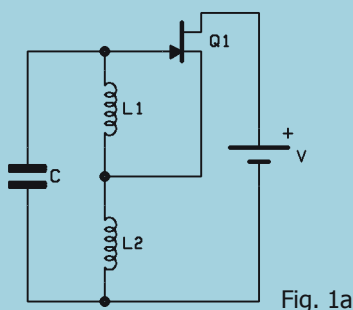
nenti che controllano la frequenza di oscillazione sono due induttori (o un induttore con presa intermedia) ed un condensatore mentre nel secondo vengono impiegati due condensatori ed un induttore. E' evidente che se dovessimo misurare solo condensatori il più adatto sarebbe l'oscillatore Hartley mentre il Colpitts sarebbe l'ideale per le bobine. L'uscita di entrambi questi oscillatori è di tipo sinusoidale: il segnale va pertanto squadrato (convertito in onda quadra) prima di essere inviato al frequenzimetro. Una soluzione potrebbe essere l'impiego di entrambi i circuiti ma, poiché gli oscillatori devono essere elettricamente separati tra loro, sarebbe necessario se-

parare anche le boccole di misura, cosa questa piuttosto scomoda.

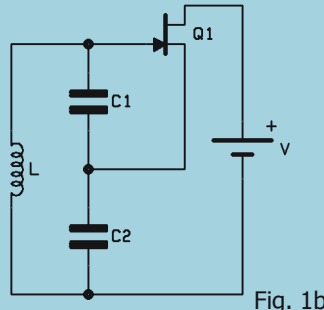
Andiamo ora per un istante allo schema a blocchi di un generico oscillatore LC (ma potrebbe essere un altro tipo di oscillatore) come quello di figura 2. Da esso si evince che un oscillatore generico è costituito da un amplificatore, una rete di retroazione *positiva* e da un elemento che determina la frequenza di oscillazione. In particolare, questo elemento presenta un comportamento tale da permettere che la frequenza voluta venga reinviata all'ingresso dell'amplificatore tramite la rete di retroazione, in modo da sostenere l'oscillazione, mentre tutte le altre frequenze vengono attenuate drasticamente. Normalmente, l'elemento di determinazione della frequenza e la rete di retroazione positiva sono uniti in un unico elemento circuitale, ma ciò non è un fattore indispensabile così come non lo è che l'amplificatore abbia un comportamento lineare; la linearità serve solo se si desidera in uscita un'onda sinusoidale. Veniamo quindi allo schema semplificato del nostro oscillatore. Il circuito è abbastanza insolito, meritevole quindi di una spiegazione ap- ➤



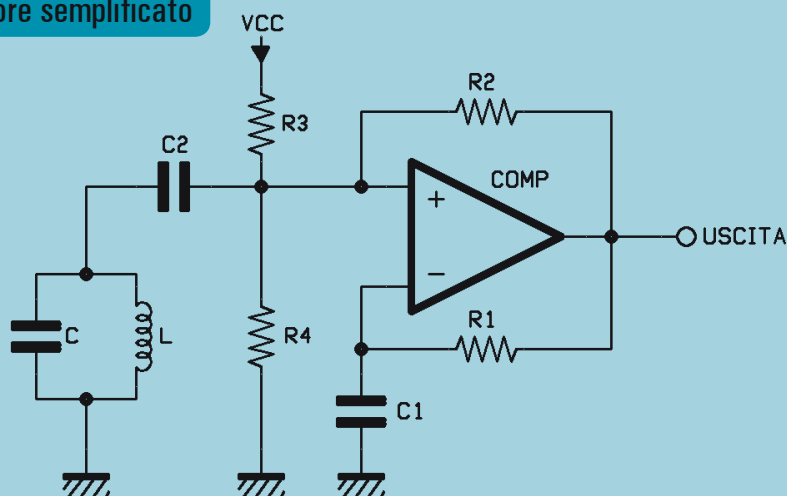
OSCILLATORE HARTLEY



OSCILLATORE COLPITTS



Oscillatore semplificato



profondità. Esso si differenzia dallo schema a blocchi precedente per due particolari: il primo è l'elemento usato per determinare la frequenza (prima costituito da un risonatore LC) e cioè il comparatore che sostituisce l'amplificatore. A prima vista sembra presente una retroazione positiva tramite R2 ed una negativa tramite R1. Prendiamo in esame per prima cosa il circuito costituito dal comparatore e le resistenze R3 e R4. Queste ultime, di uguale valore, costituiscono un partitore di tensione, cosicché all'ingresso non invertente del comparatore è presente, in assenza di altri segnali in ingresso, una tensione pari a metà di quella di alimentazione. L'uscita del comparatore è quindi positiva se all'ingresso invertente c'è una tensione inferiore a $V_{cc}/2$, altrimenti è negativa. La re-

sistenza R2 introduce una retroazione positiva che provoca un'isteresi nel funzionamento del comparatore, in unione alle già citate R3 e R4. Questo determina, una volta che la tensione in ingresso del comparatore ha raggiunto la soglia impostata, una commutazione attorno al punto di transizione senza oscillazioni spurie. In sintesi, l'isteresi migliora la "pulizia" dell'onda quadra in uscita anche in presenza di lievi disturbi in ingresso. Il circuito, con i soli elementi indicati ed in assenza di segnale di ingresso assume quindi un valore in uscita casuale (positivo o negativo) sul quale si ferma stabilmente. Supponiamo, per semplicità, che il valore sia positivo. Inseriamo ora i componenti R1 e C1 e vediamo cosa succede. La tensione in uscita (positiva, abbiamo detto) inizia a

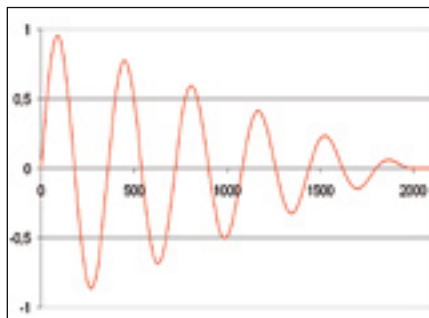
caricare il condensatore C1 tramite la R1. Non appena il valore ai capi di C1 raggiunge $V_{cc}/2$ (in realtà un valore leggermente superiore a causa dell'isteresi), il comparatore commuta rendendo l'uscita negativa; l'isteresi fornita da R2 rende la commutazione rapida e decisa. Ora, l'uscita è negativa e C1 inizia a scaricarsi; quando la tensione ai suoi capi scende sotto un valore leggermente inferiore a $V_{cc}/2$ (sempre in base all'isteresi del circuito), questi commuta nuovamente l'uscita al positivo, e via di seguito.

In sintesi, abbiamo ottenuto un oscillatore RC la cui frequenza dipende dall'isteresi applicata al comparatore tramite R2, nonché dalla costante di tempo RC.

Qualcuno starà pensando... un oscillatore RC? Che cosa ce ne facciamo? Non dovevamo creare un oscillatore LC? Vero, ma ora arriva la parte interessante della nostra soluzione: inseriamo il circuito LC, con il condensatore C2 che ha il solo scopo di bloccare la componente continua altrimenti scaricata a massa dall'induttanza L, e scegliamo dei valori per R1 e C1 abbastanza elevati, diciamo in modo che la frequenza di oscillazione del circuito RC sopra esaminato sia di almeno di un ordine di grandezza inferiore a quella di risonanza del circuito LC. Per ora trascuriamo la formula della frequenza di risonanza del "tank" LC, ci servirà solo in seguito; basti sapere che il circuito risona ad una frequenza f . Torniamo al funzionamento del circuito RC precedente che, diciamo, oscilla autonomamente ad una frequenza pari a $f/10$ circa. Al primo impulso una parte di questo viene riportata indietro tramite la R2 e il C2, e va ad eccitare, se pur in minima parte, il circuito LC appena inserito. Questo inizia ad oscillare alla sua frequenza di risonanza, f ; in assenza di stimoli ulteriori, l'oscillazione è una sinusoide smorzata. Questa sinusoide rientra però nel

- Misura di capacità: $\sim 0,5 \text{ pF} - 0,5 \text{ }\mu\text{F}$;
- Risoluzione capacità: $0,1 \text{ pF}$ o 4 cifre significative;
- Misura di induttanze: $\sim 10 \text{ nH} - 1 \text{ mH}$;
- Risoluzione induttanze: $0,1 \text{ nH}$ o 4 cifre significative;
- Precisione: 1% o migliore, dipendente dal condensatore di riferimento;
- Alimentazione: $9 V_{cc}$, 50 mA circa;
- Pulsante di cambio modalità;
- Pulsante di azzeramento capacità ed induttanza parassite;
- Display LCD a 2 righe x 16 caratteri;
- Possibilità di aggiornamento firmware tramite connettore compatibile con programmatore LCD2;
- Porta USB per future estensioni del firmware;
- Realizzazione completamente in SMD;
- Dimensioni: $120 \times 60 \times 20 \text{ mm}$ circa compreso il display.

comparatore attraverso C2, causando l'oscillazione di questo alla frequenza di risonanza del tank LC e non più a quella propria del circuito R1C1, a patto che quest'ultima sia, come abbiamo detto, di molto

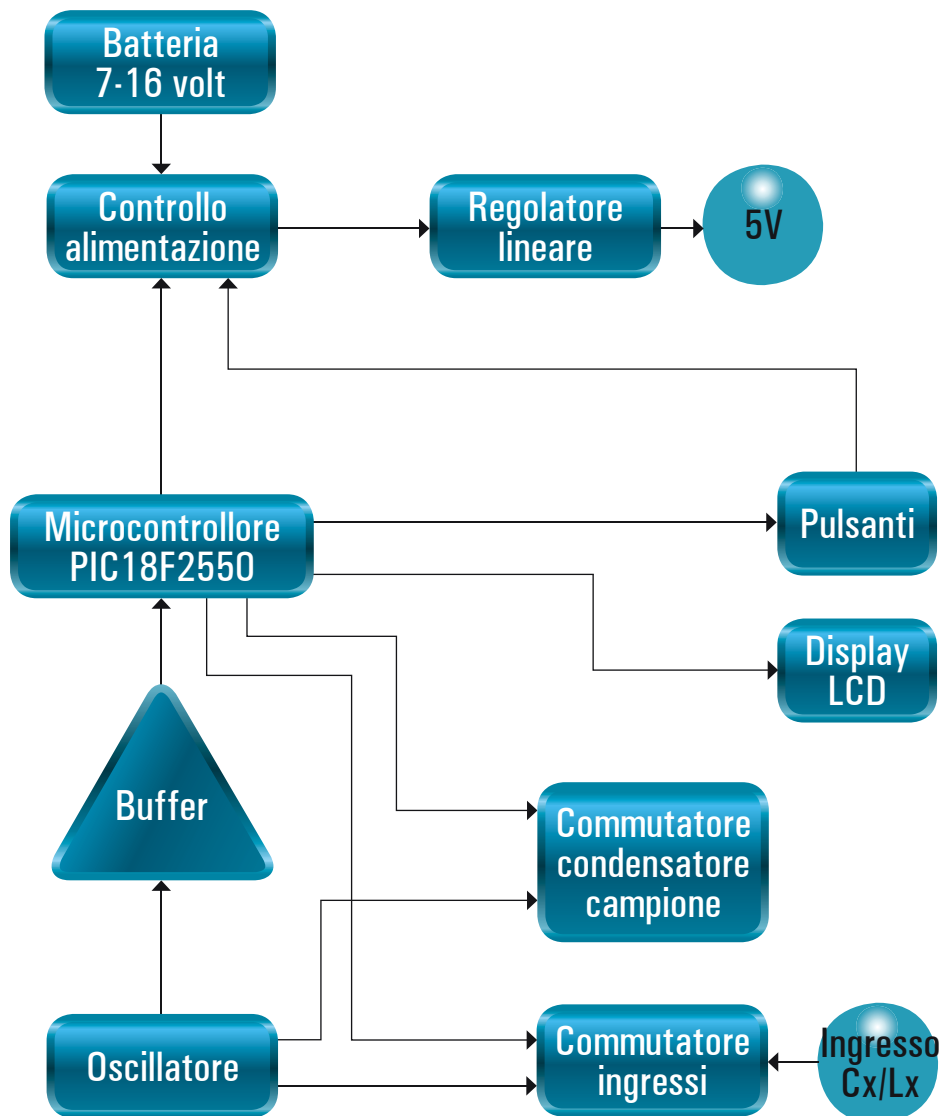


inferiore. A che serve quindi l'oscillatore RC di cui sopra? Semplicemente a dare "la prima spinta" (innesco) al circuito oscillante LC, nonché a dare spinte successive nel caso che l'oscillazione tendesse a smorzarsi.

Bisogna ricordare infatti che il nostro circuito LC, teoricamente perfetto, è tutt'altro che tale; in particolare l'induttanza *non* è pura ma ha in serie una resistenza, quella del filo metallico che costituisce l'avvolgimento. Maggiore è la resistenza, maggiore è l'attenuazione delle sinusoidi e minore è il fattore Q (di merito) del circuito oscillante. Se l'induttanza ha un Q troppo basso, e quindi una resistenza troppo alta, il circuito non sarà più in grado di oscillare alla frequenza di risonanza e inizierà a farlo alla frequenza

propria del circuito R1C1. Una trattazione completa di quanto esposto risulta matematicamente complessa e comunque inutile per lo scopo che ci siamo prefissati; basti sapere che

il fattore di merito Q diminuisce con l'incremento della resistenza serie R e, a parità di questa, con l'incremento della capacità C e la diminuzione dell'induttanza L. Questo spiega ➤



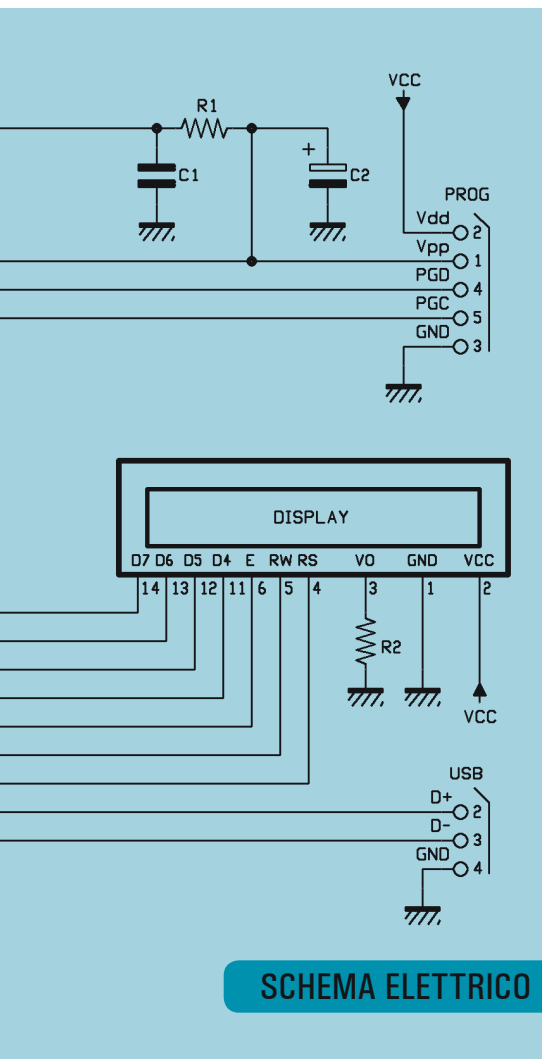
per il

MATERIALE

Tutti i componenti utilizzati sono facilmente reperibili in commercio. Dal sito della rivista (www.elettronica.in.it) è possibile scaricare il master del circuito stampato che, lo ricordiamo, è del tipo a doppia faccia a fori metallizzati. Anche il firmware necessario per la programmazione del micro (che avviene in-circuit) è disponibile sia in formato .hex che come codice sorgente. Il microcontrollore utilizzato è un PIC18F2550-E/SO a montaggio superficiale che costa 12,50 Euro IVA compresa.

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

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



SCHEMA ELETTRICO

era stato scritto in assembler su un PIC della serie 16, più che sufficiente per questa applicazione.

Successivamente, avendo constatato in altri progetti tanto la grande flessibilità della programmazione in linguaggio C, quanto il basso costo dei PIC della serie 18, il tutto è stato rivisitato utilizzando un microcontrollore PIC18F2550 che dispone, oltre a tutte le risorse necessarie, anche di una porta USB utilizzabile per eventuali future applicazioni.

Il clock è ottenuto tramite un quarzo da 16 MHz, moltiplicato per 4 dal PLL interno e portato a 48 MHz, frequenza necessaria per il funzionamento della porta USB. Il display LCD a due righe da sedici caratteri è gestito in modalità a 4 bit; vengono quindi utilizzate quattro linee di

I/O per i dati più tre linee per i bit di controllo.

Due linee di I/O sono poi dedicate ai pulsanti di controllo, una alla gestione dell'alimentazione, due per il selettore di modalità, una per l'inserimento della capacità di calibrazione ed infine due per la misura della frequenza dell'oscillatore LC, per un totale di quindici linee di I/O.

Stadio Oscillatore

Lo stadio oscillatore fa capo alla prima parte dell'integrato IC1, un operazionale veloce della Maxim, qui utilizzato come comparatore. La scelta è caduta su questo operazionale perchè, dopo aver provato nel circuito alcuni comparatori ed altri operazionali, abbiamo constatato che questo chip offriva i migliori risultati. Dallo schema sono facilmente identificabili i vari elementi già presi in esame durante la trattazione teorica; l'unico particolare degno di nota è rappresentato dal circuito LC.

Questo è realizzato tramite l'induttanza L1 da 68 μ H, il condensatore C5 da 1 nF e, come verrà spiegato nella descrizione degli stadi di commutazione d'ingresso, dall'induttanza incognita in serie alla L1 nel caso di misura di induttanze, oppure dalla capacità incognita in parallelo alla C5 nel caso di misura di capacità.

La frequenza di oscillazione, che a vuoto è pari a circa 615 kHz, diminuisce in proporzione al valore del componente da testare.

Dallo schema si può notare inoltre la strana sequenza di condensatori in parallelo C9-C10-C11-CCal.

Questi sono gli elementi di autocalibrazione dello strumento, e sono gli unici per cui è richiesta una elevata precisione. Il firmware prevede un valore complessivo pari a 1 nF, come indicato nello schema per CCal; gli altri condensatori sono stati inseriti solo per poter montare sul circuito stampato componenti di precisione con geometrie differenti.

Il valore complessivo da inserire è quindi pari a 1 nF, salvo eventuali modifiche al firmware per adattare altri valori disponibili.

Commutatori ingressi e capacità campione

Gli switch analogici SW1 e SW2 costituiscono il commutatore di ingresso in grado di inserire l'induttanza incognita in serie a L1 (SW2) oppure la capacità incognita in parallelo a C5 (SW1).

Questi componenti, degli switch CMOS della ONSEMI, non sono stati scelti per far perdere la vista a chi deve montare il circuito (purtroppo sono disponibili nel solo package SC-88 a 6 pin, spaziatura tra i pin di 0,65 mm), bensì per la loro resistenza ON estremamente bassa, tipicamente intorno ai 6,5 / 7 ohm.

Per avere un termine di paragone, un comune CD4066 ha una resistenza tipica di 270 ohm, mentre un più recente 74LVX4066 ha una resistenza standard di 25-30 ohm.

Questi valori possono sembrare bassi, ma già i soli 6,5 ohm dei componenti scelti hanno un effetto tale sul Q del sistema che lo strumento difficilmente riesce a raggiungere la misura di 0,5 μ F, mentre con un collegamento diretto potrebbe misurare condensatori ben oltre il μ F. Componenti con una resistenza di ON più elevata avrebbero reso il circuito talmente instabile da essere pressoché inutilizzabile.

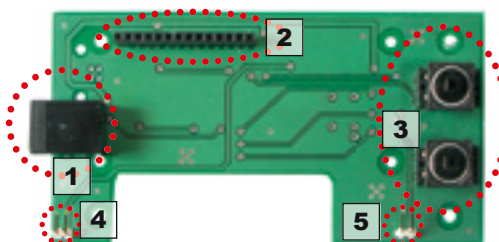
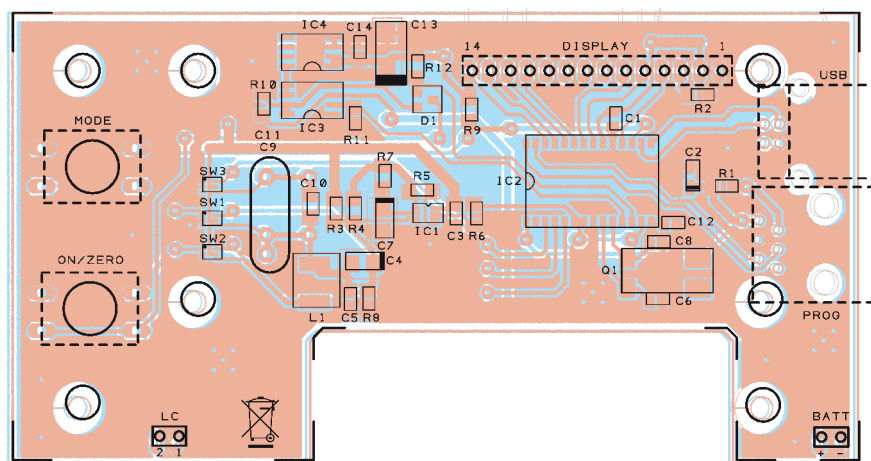
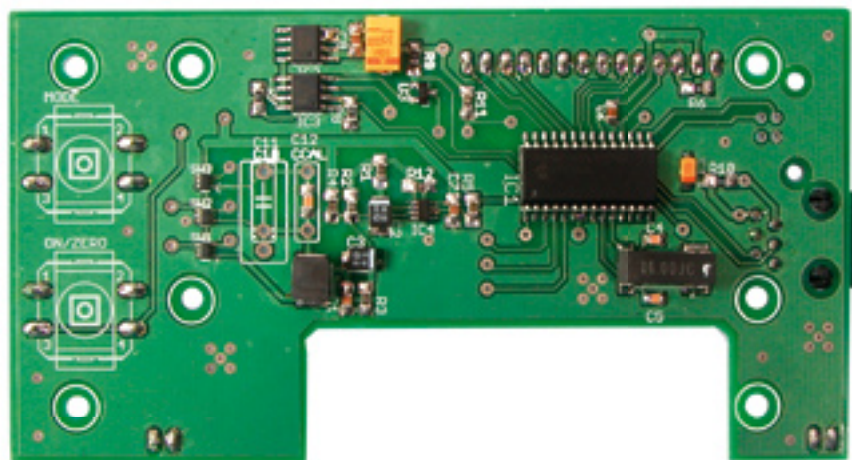
In un primo momento, al posto degli switch digitali, avevamo utilizzato dei relè, i quali - come noto - hanno una resistenza di contatto dell'ordine dei milliohm. A fronte dei vantaggi offerti da questa soluzione dobbiamo tuttavia considerare il maggior consumo, la minore compattezza meccanica ed il rumore continuo dovuto all'autocalibrazione. Abbiamo perciò deciso di utilizzare gli switch digitali. Il terzo di questi (SW3) viene utilizzato dal micro per inserire nel circuito di ingresso

ELENCO COMPONENTI:

C1: 100 nF multistrato (SMD)
 C2: 2,2 µF 10 VL tantalio (3216)
 C3: 100 nF multistrato (SMD)
 C4: 10 µF 10 VL tantalio (3528)
 C5: 1 nF multistrato (SMD)
 C6: 15 pF ceramico (SMD)
 C7: 10 µF 10 VL tantalio (3528)
 C8: 15 pF ceramico (SMD)
 C9: vedi testo
 C10: vedi testo
 C11: vedi testo
 C12: 100 nF multistrato (SMD)
 C13: 10 µF 35 VL tantalio (6032)
 C14: 100 nF multistrato (SMD)
 D1: BAV70 (SOT23)
 IC1: MAX4413 (SOT23-8)
 IC2: PIC18F2550-SO (SO-28W)
 IC3: MMDF2C03HD (SO08)
 IC4: CS801D (SO08)
 L1: 68 µH (L5650M)
 LCD1: Display LCD 16x2
 Q1: Quarzo 16 MHz (MG3A)
 R1: 39 kohm (SMD)
 R2: 1 kohm (SMD)
 R3÷R5: 22 kohm (SMD)
 R6: 4,7 kohm (SMD)
 R7: 22 kohm (SMD)
 R8: 22 kohm (SMD)
 R9: 10 kohm (SMD)
 R10: 1 Mohm (SMD)
 R11: 100 kohm (SMD)
 R12: 100 kohm (SMD)
 SW1÷SW3: NLASB3157 (SC70-6L)
 MODE: Switch
 ON/ZERO: Switch

Varie:

- Presa USB-A
- Connettore RJ12 6/6
- Strip maschio 90° 2 pin (2 pz.)
- Strip maschio 14 pin



1	Porta di programmazione
2	Connettore Display
3	Pulsanti di controllo
4	Connettore alimentazione
5	Connettore morsetti esterni

Il montaggio va condotto con grande attenzione e precisione essendo quasi tutti i componenti del tipo a montaggio superficiale. Notate nella foto qui sopra i connettori ed i pulsanti, unici elementi ad inserzione. La tabella a fianco ne descrive le funzioni.

il condensatore di calibrazione. Qui serve obbligatoriamente un po' di matematica per mostrare come, con un solo componente di precisione di facile reperibilità lo strumento sia in grado di autocalibrarsi e di compensare le tolleranze di tutti gli altri componenti del circuito LC. La formula per la frequenza di risonanza di un circuito LC è data da:

$$f = \frac{1}{2\pi \sqrt{LC}}$$

che, con i valori adottati nel circuito genera una frequenza pari a 610331 Hz circa. Questo valore non è casuale essendo stato scelto inizialmente per essere contenuto in 2 byte (massimo 65535 Hz), sfruttando quindi l'inte-

ro range di misura. Con la serie 18 del PIC, invece, la misura avviene su 3 byte, quindi il limite non sussiste più. Indicando quindi con f_0 la frequenza a riposo e con LC_0 il prodotto $L1 \cdot C5$, possiamo scrivere:

$$f_o = \frac{1}{2\pi \sqrt{LC_o}}$$

Inseriamo ora un condensatore di capacità nota e precisa in parallelo a C5; indichiamo quindi il prodotto $L1*(C5+C_{cal})$ con LC_c .

Si ha quindi:

$$f_c = \frac{1}{2\pi\sqrt{LC_c}}$$

Ora, pur non conoscendo esattamente i valori di $L1$ e $C5$, essendo questi componenti con elevata tolleranza, misurando le due frequenze f_0 e f_c , e conoscendo il valore della capacità di calibrazione C_{cal} si ottiene la relazione:

$$LC_c - LC_o = \left(\frac{1}{2\pi f_c}\right)^2 - \left(\frac{1}{2\pi f_o}\right)^2 = A$$

in cui la costante "A" semplifica le formule scritte in seguito. Sostituendo LC_c con $L1*(C5+C_{cal})$ e LC_o con $L1*C5$ si ottiene:

$$L1(C_5 + C_{cal}) - L1C_5 = A$$

$$L1C_5 + L1C_{cal} - L1C_5 = A$$

e semplificando:

$$L1C_{cal} = A \Rightarrow L1 = \frac{A}{C_{cal}}$$

Abbiamo quindi ottenuto il valore dell'induttanza $L1$, comprese le tolleranze, con una precisione dipendente solo dalla tolleranza di C_{cal} e dall'errore nella frequenza, quest'ultimo dell'ordine di una parte per milione circa. Dalla prima relazione (f_0) si può poi ricavare il valore di $C5$. Avendo ora a disposizione i valori reali di $L1$ e $C5$, è possibile determinare il valore di qualsiasi componente incognito inserito nel circuito misurando la frequenza.

Buffer

Il buffer, ottenuto con il secondo operativo (IC1b) collegato come

inseguitore di tensione a guadagno unitario, serve per aumentare la separazione tra il circuito oscillante e quello di misura (PIC), soluzione che permette di aumentare la stabilità del sistema in lettura.

Firmware

Come già anticipato, il firmware originale era scritto in assembler per PIC16. È stato però riscritto in C e trasportato su PIC18 per sfruttare i vantaggi offerti sia dal linguaggio che dal micro, entrambi più potenti. Come di consueto, anche in questo progetto abbiamo preferito cercare

di ottenere la massima modularità definendo una serie di librerie riutilizzabili in altri progetti, a discapito dell'efficienza del codice stesso. Abbiamo inoltre cercato di evitare, per quanto possibile, l'utilizzo di routine in assembler, presenti solo nel modulo DELAY che necessita di elevata precisione.

Tutte le nuove librerie sono inoltre configurabili all'occorrenza tramite file esterni, come avrete modo di vedere in dettaglio nei moduli LCD e DELAY. I vari moduli sono suddivisi in due cartelle separate: la prima, denominata `_PIC18LIB_`, contiene la maggior parte del codice (quello generico e riutilizzabile) mentre la seconda, `LC METER`, contiene il solo codice specifico per questo progetto.

L'organizzazione del firmware è descritta a pagina 58.

Realizzazione pratica

Abbiamo già discusso brevemente alcuni aspetti relativi al montaggio di circuiti in SMD in occasione del progetto DDS; per chi non avesse letto quell'articolo, rotorniamo sulle cose principali.

Montare a mano i circuiti SMD è possibile: con un po' di esperienza, di pratica e un minimo di attrezzatura, può essere persino più semplice

che montare i classici componenti ad inserzione.

Attrezzatura necessaria:

- Saldatore con punta molto fine. Il primo prototipo è stato realizzato con una punta da 1 mm però consigliamo una punta da 0,5 mm. Non servono attrezzi particolari controllati in temperatura o ad aria calda o simili; sarebbe invece opportuno, ma non indispensabile, un isolamento galvanico tra saldatore e rete elettrica, per evitare problemi di elettricità statica. Riteniamo assai improbabile che un componente SMD si possa guastare per effetto dell'elettricità statica, ma consigliamo comunque di prestare attenzione.

- Flussante. È un prodotto chimico reperibile nei negozi di elettronica e disponibile sia liquido che in gel. Ha lo scopo di ripulire i contatti dall'ossido superficiale e di favorire l'adesione dello stagno. Nelle saldature SMD è indispensabile poiché si una quantità di stagno residua e il poco disossidante in esso contenuto, bruciando immediatamente, non è sufficiente ad eliminare l'ossido. I flussanti in gel sono i migliori: essi aderiscono al PCB ed evaporano lentamente; hanno inoltre un ottimo comportamento in alta frequenza. Vengono venduti in tubetti e sono molto più costosi di quelli liquidi. I flussanti liquidi vanno applicati con un pennellino, evaporano abbastanza rapidamente e spesso lasciano residui da ripulire con trielina o simili, però sono in genere molto economici. Non è conveniente tentare un montaggio SMD senza disporre di questo prodotto.

- Una lente di ingrandimento, o meglio, una lampada da tavolo con lente incorporata.

- Una pinzetta piccola (anche del tipo reperibile nei negozi di cosmetici, in mancanza di meglio)

- Una mano decisamente ferma; questo è l'attrezzo più difficile da reperire...



Iniziamo con la descrizione della sottocartella **PIC18LIB**:

Modulo DELAY

Gestisce i ritardi in millisecondi in modo indipendente dal quarzo. Questo modulo può essere utilizzato in varie applicazioni: esso fornisce le seguenti funzioni:

```
// INIZIALIZZAZIONE DEL MODULO DELAY
void DELAY_Init(unsigned long XTal) ;

// RITARDO DI n MILLISECONDI (n <= 255)
void DELAY_NmSec(BYTE mSec) ;

// RITARDO DI n CENTINAIA DI MICROSECONDI
void DELAY_N100uSec(BYTE HuSec) ;

// RITARDO DI n MILLISECONDI (n <= 65535)
void DELAY_mSec(unsigned int mSec) ;
```

Come si può notare, tutte le funzioni utilizzano un tempo in millisecondi e non in numero di cicli di clock, molto più comprensibile. L'associazione alla reale velocità del processore avviene mediante la funzione `DELAY_Init`, richiamata all'inizio del programma, alla quale viene passata la frequenza di clock. Il modulo `DELAY` è l'unico contenente codice assembler.

Modulo LCD

Si tratta di un classico modulo di gestione di display LCD, impostato però per ottenere la massima flessibilità di utilizzo; in esso infatti tutti i collegamenti al display sono configurabili in maniera indipendente, e non devono neppure essere sequenziali; ad esempio è possibile connettere la linea dati D0 dell'LCD alla porta RA4 e la linea D1 alla porta RC5. Il modulo è in grado di gestire display a 2 e a 4 linee, sia con controller singolo che doppio, sia da 20 che 24 caratteri o anche altre tipologie. La configurazione del modulo avviene tramite il file di inclusione `LCD_PARAMS.H`, impostabile a partire dal file `LCD_PARAMS_SAMPLE.h` che costituisce un esempio di utilizzo.

Nel modulo `LCD`, oltre alle classiche funzioni di stampa caratteri, sono presenti differenti routine per il posizionamento del cursore, la cancellazione della linea intera o a partire dalla posizione corrente, lo scorrimento in alto del testo, eccetera. Per ulteriori dettagli su queste funzioni rimandiamo alla lettura del codice sorgente.

Modulo PRINTVAL

Questo modulo consente la stampa formattata sul display LCD di va-

lori numerici, scegliendo il numero massimo di cifre, le cifre decimali, il formato scientifico o ingegneristico, ecc.

Fornisce due funzioni; la prima, `PrintDbl`, consente la stampa del numero; la seconda, utilizzata nel formato ingegneristico, stampa un prefisso corrispondente all'esponente ritornato dalla prima; ad esempio, per un esponente -12 stampa il prefisso 'p' che sta per 'pico'.

Modulo TIMER0

Gestione completa delle funzioni del `TIMER0` del PIC.

Forniamo ora una breve descrizione della sottocartella **LC_METER**, contenente le funzioni specifiche per il progetto corrente.

Modulo XTAL.H

È un modulo di configurazione in cui viene fornita al programma la frequenza di clock interna della PIC. Questa viene utilizzata principalmente dal modulo `DELAY` per calcolare i cicli macchina necessari per i ritardi.

Modulo PIC.H

Si tratta di un modulo di configurazione contenente il tipo di PIC utilizzato e alcuni parametri necessari al suo funzionamento.

Modulo LC_METER.H

È il file di inclusione principale del progetto. In esso vengono elencati tutti i moduli necessari e definite le funzioni utilizzate. È il modulo principale e meriterebbe un capitolo a sé stante. Non potendo fornire ulteriori dettagli, si rimanda alla lettura del codice sorgente.

Modulo LCD_PARAMS.H

Modulo di configurazione del collegamento tra display LCD e PIC. Per meglio comprendere questa sezione consigliamo di riguardare il modulo `LCD` e studiare in dettaglio il codice sorgente.

Modulo LC_BUTTONS

Modulo di gestione dei due pulsanti dello strumento.

Gestisce il cambio di modalità di lettura, l'azzeramento e lo spegnimento dell'apparecchio.

Modulo LC_CALIBRATE

Consente la calibrazione dello strumento tramite il condensatore campione interno.

Modulo LC_DISPLAY

Gestisce la visualizzazione ad alto livello sul display LCD. Contiene inoltre il codice di stampa della modalità corrente e dei valori letti.

Montaggio

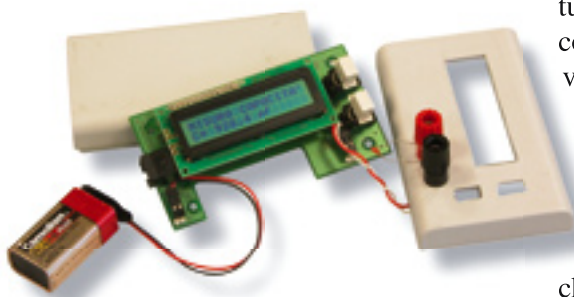
Contrariamente al solito suggeriamo di montare per primi i componenti attivi, ed in particolare quelli con i piedini più ravvicinati. Questo perché, se si partisse dagli altri, risulterebbe difficile in seguito effettuare

le saldature di precisione. I componenti che richiedono una notevole attenzione durante la saldatura sono i tre switch analogici `NLASB3157` e l'operazionale `MAX4413`.

In particolare, gli switch sono abbastanza ostici non tanto per la saldatura in sé quanto per la difficoltà nel controllare se i piedini sono effettivamente saldati bene.

Consigliamo quindi un'ottima lente o ancora meglio un "contafili" da filigrana per controllare le saldature di questi componenti. Può essere molto utile effettuare anche una verifica della continuità con

il tester. Tornando alla saldatura di elementi con piedini a passo molto ravvicinato, il procedimento è il seguente: per prima cosa stendete un velo di fluxante sul PCB; è meglio usare il gel perché, essendo piuttosto viscoso, fa aderire il circuito integrato al PCB. Appoggiate il chip con la pinzetta, controllatene bene la posizione e saldate leggermente un pin d'angolo. Ricontrollate la posizione ed aggiustatela se necessario. Nel caso non fosse abbastanza precisa, dissaldare il pin e ricominciate da capo. Quando il chip è perfettamente allineato (attenzione a tutti





flussante-saldatore finché lo stagno che metteva in corto i piedini non sparisce. Anche qui, non preoccupatevi troppo per il calore, sulla base della nostra esperienza i moderni componenti elettronici, anche da questo punto di vista, sono molto performanti.

Una volta montati i componenti sopra descritti, il resto è abbastanza semplice in quanto il procedimento è lo stesso e i componenti sono di dimensioni maggiori. Tra i componenti passivi, gli induttori sono i più delicati perché un calore eccessivo potrebbe far bruciare la vernice isolante del filo con cui sono realizzati. In generale, per resistenze, condensatori e induttori dovete applicare un po' di stagno su una delle due piazzole del PCB, posizionare il componente e saldarlo facendo fondere lo stagno appena depositato, saldare l'altro pin e, infine, ripassare la prima saldatura col flussante per un migliore risultato.

I componenti ingombranti (connettori, pulsanti, ecc.) devono essere montati per ultimi.

Terminata la basetta principale, montate il connettore sul display e quindi inseritelo sul PCB.

Programmazione firmware

Lo strumento di misura è dotato di un connettore RJ11 a cui il programmatore/emulatore ICD2 della Microchip può essere collegato tramite il cavo di adattamento fornito a corredo.

L'ICD2 potrebbe risultare molto utile soprattutto per eventuali modifiche al codice, permettendo il debug passo/passo, l'aggiunta di punti di controllo ecc.; però non è indispensabile se si deve solo programmare il chip.

Non disponendo di questo strumento, è possibile utilizzare qualsiasi programmatore di PIC; consigliamo di realizzare un cavo adattatore con connettore RJ11 seguendo la piedinatura riscontrabile nello schema ➤

i lati) saldate un altro pin, possibilmente in posizione diagonalmente opposta al primo, tenendo fermo il chip. Ricontrollate che l'allineamento sia corretto quindi proseguite. Se dovessero esserci delle imprecisioni, niente paura: dissaldare i due pin risulta piuttosto semplice. Visto che la prudenza non è mai troppa, saldate un terzo pin e ricontrollate nuovamente l'allineamento che, a questo punto, dovrebbe essere quello definitivo. Ora potete procedere alla saldatura finale: stendete una quantità adeguata di flussante su una fila di pin e procedete alla saldatura usan-

do pochissimo stagno e lasciando raffreddare il chip di tanto in tanto. Può capitare di esagerare con lo stagno e di fare un ponte tra due pin, cosa possibile in particolar modo coi quattro chip di cui sopra. Anche qui, niente paura, pulite bene il saldatore con una spugnetta, stendete un velo abbondante di flussante sui pin in cortocircuito e appoggiateci sopra il saldatore. Lo stagno in eccesso tenderà a risalire sia sui pin sia sulla punta del saldatore per effetto della tensione superficiale; se una volta sola non dovesse bastare, si può ripetere l'operazione pulizia-

elettrico e nel circuito stampato. Purtroppo, essendo il progetto realizzato con componenti SMD, è molto difficile programmare il PIC prima del montaggio, a meno che non si utilizzi uno specifico adattatore, che potrebbe essere o autocostruito o scelto tra quelli reperibili in commercio. Una nota sulla programmazione: poiché l'alimentazione del circuito è controllata dal PIC, occorre fare in modo che l'apparecchio rimanga alimentato durante tutto il tempo necessario al trasferimento del file; questo può avvenire solo mantenendo premuto il pulsante di accensione per tutta la durata dell'operazione di programmazione. In sintesi occorre :

- Connettere il programmatore al circuito;
- Collegare l'alimentazione all'LC-Meter;
- Mantenere premuto il pulsante di accensione;
- Eseguire la programmazione, senza rilasciare il pulsante di accensione;
- Rilasciare il pulsante quando appare il menu iniziale.

Utilizzo pratico

All'accensione lo strumento mostra un messaggio di benvenuto, quindi si predispone per la lettura delle capacità. Più in dettaglio, la prima riga mostra la scritta "MISURA CAPACITA'" e la seconda indica un valore di capacità parassita intorno ai 10 pF in assenza di puntali connessi: ad esempio "C = 15.14 pF". Questa capacità parassita dipende dai fili interni al contenitore, dalle boccole, dalle piste del PCB, ecc. Quando si inseriscono dei puntali di misura con cavi più o meno

lunghi, il valore letto aumenta rispetto al precedente.

L'azzeramento si effettua con una breve pressione del tasto ON/AZZERA; lo strumento si spegne se lo stesso tasto viene mantenuto premuto per più di 3 secondi.

Una volta azzerato, lo strumento indica una capacità residua solitamente inferiore a 0.1 pF: notate che le cifre decimali sono molto instabili, salvo la prima; ciò è normale.



La capacità misurata aumenta quando avvicinate le mani alle boccole di ingresso; nel prototipo, afferrando le boccole (isolate) con le dita, lo strumento indica una capacità di circa 1.5 pF. Se non avete ancora connesso i puntali (anche quelli del tester vanno bene) potrete farlo adesso; chiaramente la capacità parassita varia di nuovo ma può essere compensata ancora tramite il tasto AZZERA.



Collegate ora il componente sconosciuto alle boccole: il suo valore viene indicato sul display con l'unità di misura corretta e con 4 cifre significative; ad esempio, un condensatore da 15 pF potrebbe dare 15.23 pF; uno da 12 nF, invece, 11.45 nF.

Un valore troppo grande (o un componente errato connesso alle boccole) comporta l'indicazione di "FUORI SCALA" sul display. Per la misura di induttanze, occorre premere il tasto MODE; il display ora mostra nella prima riga la scritta "MISURA INDUTTANZA" e "FUORI SCALA" sulla seconda

riga se non dovesse essere collegato alcun componente alle boccole.

Ciò dipende dal fatto che un valore



di induttanza prossimo a zero corrisponde ad un cortocircuito, contrariamente a quanto avviene per i condensatori in cui il valore zero è riconducibile ad un circuito aperto. Inserendo, ad esempio, un cavallotto di rame tra le boccole, lo strumento indica un'induttanza parassita di pochi nH (nanoHenry), azzerabile con il pulsante AZZERA.



Anche in questo caso, connettendo dei puntali alle boccole, l'induttanza parassita aumenta bruscamente e deve essere eliminata con la pressione del tasto AZZERA, avendo però l'accortezza di cortocircuitare prima i due puntali di misura.

Spegnimento dello strumento

Per spegnere l'apparecchio, è sufficiente mantenere premuto il tasto ON/AZZERA per circa 3 secondi, e comunque fino a che la scritta AZZERAMENTO verrà sostituita dalla scritta SPEGNIMENTO.

Per semplicità, nel firmware non è



stata prevista una funzione di spegnimento temporizzato; nulla vieta, tuttavia, di aggiungere le poche righe necessarie ad ottenere tale funzione.



Mambo, localizzatore portatile programmabile

di Davide Scullino

Nuovo protagonista nel mercato dei ricevitori GPS, MAMBO è un dispositivo di localizzazione completamente programmabile, funzionante a batteria e provvisto di cellulare GSM/GPRS integrato, interfaccia Bluetooth e sensore di movimento. Scopriamolo insieme. Prima puntata.



I sistemi di localizzazione e navigazione assistiti da satellite, sono stati e sono argomento che affrontiamo da sempre con vivo interesse, profondendo sforzi ed energie nella ricerca di nuovi prodotti o di applicazioni innovative; un esempio di ciò è stato il localizzatore portatile descritto nel fascicolo n° 100, vera punta di diamante di quella che è la nostra filosofia in materia di GPS: un dispositivo miniaturizzato, alimentabile a batteria e provvisto di cellulare GSM tramite il quale può sia essere interrogato a distanza per conoscere posizione e percorso di chi lo porta con sé,

sia inviare esso stesso, in caso di necessità, e-mail contenenti le coordinate del momento tramite il servizio SMS. Cercando qua e là, di recente ci siamo imbattuti in un prodotto che somiglia molto al nostro localizzatore GPS/GSM, ma che offre qualcosa in più e che certo interesserà chi opera professionalmente in questo settore: si chiama Mambo, è prodotto dalla Falcom ed è anch'esso un localizzatore GPS provvisto di cellulare integrato GSM/GPRS, il quale, anche in questo caso, può essere utilizzato per fare delle normali telefonate. Dispone di una batteria ricaricabile che gli permette di funziona-

>



Navigatore satellitare tramite PDA con programma di navigazione ed interfaccia Bluetooth.

re autonomamente e che lo rende adatto sia ad essere affidato a una persona (per esempio un escursionista, che può portarlo nello zaino così da essere rintracciato nel caso si perda o abbia un incidente) sia a venire collocato in un autoveicolo, nel quale può funzionare anche senza essere necessariamente collegato all'impianto elettrico: ciò permette di nascondere più facilmente e di impedirne la scoperta misurando la corrente assorbita a vuoto dalla

batteria dell'auto. Il Mambo può essere chiamato e quindi interrogato a distanza da una postazione di controllo, ma non solo: può inviare SMS contenenti la posizione o effettuare autonomamente chiamate dirette a chi tiene sotto controllo i movimenti della persona o del veicolo sul quale si trova. Particolarità molto interessante è la presenza di un sensore di movimento il quale può controllare importanti funzioni; sul coperchio sono presenti tre pul-

santi ai quali possono essere affidati compiti differenti che dipendono da come il dispositivo viene programmato. In tema di programmazione, va detto che il Mambo è provvisto di un'interfaccia Bluetooth (di classe 2 versione 1.1), la quale consente di dialogare con il modulo ed effettuare tutte le impostazioni e la programmazione, ma anche di sfruttare le funzioni tipiche di un cellulare Bluetooth. Il tutto viene controllato da una CPU ad architettura RISC.

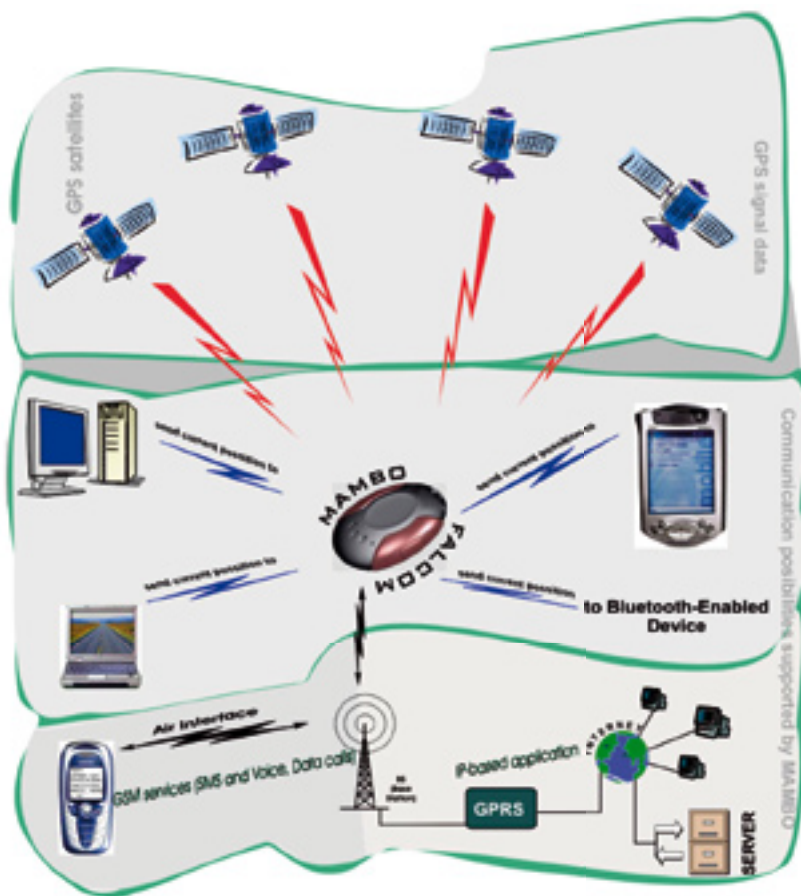
La programmabilità è proprio la caratteristica più importante di questo dispositivo nel senso che tutte le funzionalità disponibili possono essere gestite, correlate o escluse mediante la programmazione iniziale. In pratica possiamo definire, via software, come deve operare l'apparato, escludendo quelle funzionalità che, magari, per il nostro utilizzo, sono superflue e sfruttando al massimo quelle che più ci servono. E' come se potessimo realizzare l'hardware ed il firmware del nostro localizzatore ideale. Se quello che abbiamo detto ha stuzzicato la vostra curiosità, continuate a leggere e scoprirete quali e quante sono effettivamente le potenzialità del Mambo, il primo e forse l'unico localizzatore GPS che integra un vero e proprio telefono cellulare, un'interfaccia Bluetooth, una batteria e un sensore di movimento e che viene fornito completo di software per la programmazione e la gestione di tutte le funzioni.

Di seguito analizzeremo una ad una le varie sezioni del Mambo in modo da poterne sfruttare tutte le potenzialità.

Il localizzatore

La sezione GPS è basata su un ricevitore a 20 canali (ad antenna integrata) capace quindi di agganciare e seguire i segnali di 20 satelliti contemporaneamente.

E qui va fatta una precisazione: se è vero che per localizzare un veicolo



o una persona che porta indosso un dispositivo del genere bastano i segnali di tre satelliti, è anche vero che normalmente oggi non è più così difficile ricevere la portante di 8÷10 satelliti.

Inoltre alla rete tradizionale Navstar sono stati affiancati alcuni satelliti addizionali che hanno lo scopo di aiutare i ricevitori a terra per calcolare la posizione con maggiore precisione. Per cui, anche se oggi è piuttosto difficile riuscire ad utilizzare contemporaneamente tutti i 20 canali, esistono delle situazioni particolari che comunque teoricamente consentirebbero di tracciare anche 14 o 15 satelliti per periodi di tempo variabili nel corso della giornata ed anche dipendenti dalla posizione in cui si trova il GPS.

Non vogliamo ripetere quanto abbiamo già scritto nell'articolo "Nuovi GPS avanzano: l'era dei 32 e dei 50 canali" pubblicato sulla rivista N. 115, ma vi suggeriamo di rileggerlo in quanto gli argomenti tecnici trattati in quella sede sono pertinenti al 100% con questa presentazione. Qui invece vogliamo parlare delle capacità del chipset che governa la sezione ricevente GPS, ovvero il SiRFStarIII prodotto da SiRF Technology, uno dei più importanti produttori di chipset per ricevitori GPS. Il front-

end radio dell'SS3 (abbreviazione di SiRFstarIII) riceve i segnali in banda L1 (1575,42 MHz) con codifica C/A a 1023 MHz.

La parte digitale, invece, che integra Down Converter, Banda Base, DSP, Microprocessore ARM7 e memoria Flash / RAM, gestisce le sentenze GGA, GGL, GSV, RMC secondo il protocollo NMEA0183 (per quanto comunque esso sia in grado di fornire maggiori dettagli tramite il protocollo Binary SiRF, non usato però dal Mambo). Il chipset è molto sensibile: esso presenta infatti un valore di -142 dBm in acquisizione e di -158 dBm in tracking, con tempi di acquisizione di 35 secondi in cold start (effemeridi, almanacchi e ora UTC sono sconosciuti) e meno di 8 secondi in hot start (effemeridi, almanacchi e ora UTC non più vecchi di due ore).

Riguardo alla precisione, il ricevitore ha un'accuratezza migliore di 10 m CEP sulla posizione, di 0,1 m/s sulla velocità e 1 msec rispetto all'orario ricevuto dal sistema GPS.



Sezione cellulare

Il modulo telefonico radio contenuto nel Mambo è un Tri-Band di tipo GSM/GPRS ed è un vero e proprio modem-cellulare con antenna integrata, tant'è che il dispositivo prevede, come accessorio, un auricolare da usare a corredo, se si intende sfruttarne tutte le funzionalità telefoniche in auto. Non manca >



Hardware Setup



Sfilare il coperchio e rimuovere la batteria



Inserire la SIM



Chiudere il porta SIM



Inserire la batteria



Richiudere il coperchio

no microfono, altoparlante e la possibilità di conversare a viva-voce; per la connessione di un auricolare cablati, è previsto un jack da 2,5

mm di diametro a quattro poli. Naturalmente è previsto anche l'abbinamento con un auricolare wireless Bluetooth. Fatte salve le limitazioni imposte dalla Legge, nell'utilizzo del Mambo come dispositivo di localizzatore e ascolto ambientale all'insaputa del conducente, le funzioni vocali permettono di sfruttare l'apparecchio anche per ascoltare da remoto quanto viene detto all'interno del veicolo; infatti, opportunamente programmato, il Mambo può attivare il proprio microfono quando riceve una chiamata e permettere al chiamante di ascoltare quel che avviene nel veicolo sul quale è collocato. Oltre alle funzioni vocali, supporta gli SMS e la ricetrasmisione di dati; in modalità GPRS, comunica secondo i protocolli TCP/IP e PPP (GPRS classe 10, B).

Sul piano telefonico, va detto che il nostro localizzatore è disponibile in due versioni, adatte ciascuna ad uno specifico mercato: il Mambo 55 e il Mambo 56; il primo è progettato per funzionare a 900/1800/1900 MHz (frequenze utilizzate nel sistema cellulare europeo) mentre il secondo lavora a 850/1800/1900 MHz (sistema radiomobile americano).

Nelle applicazioni che descriveremo nei prossimi articoli faremo sempre riferimento alla versione del Mambo 55. Il connettore per la SIM-Card è accessibile sollevando il coperchio e rimuovendo la batteria.

Interfaccia Bluetooth

Come accennato, Mambo integra un dispositivo di comunicazione Bluetooth utilizzato per collegare un auricolare o altro apparato standard wireless (Bluetooth, ovviamente) ma anche per la comunicazione con il Personal Computer nel quale va installato il programma di configurazione; questo, anche perché, il dispositivo non ha altra porta per dialogare con l'esterno: il programma di funzionamento ed eventuali modifiche, vengono tutti caricati nella



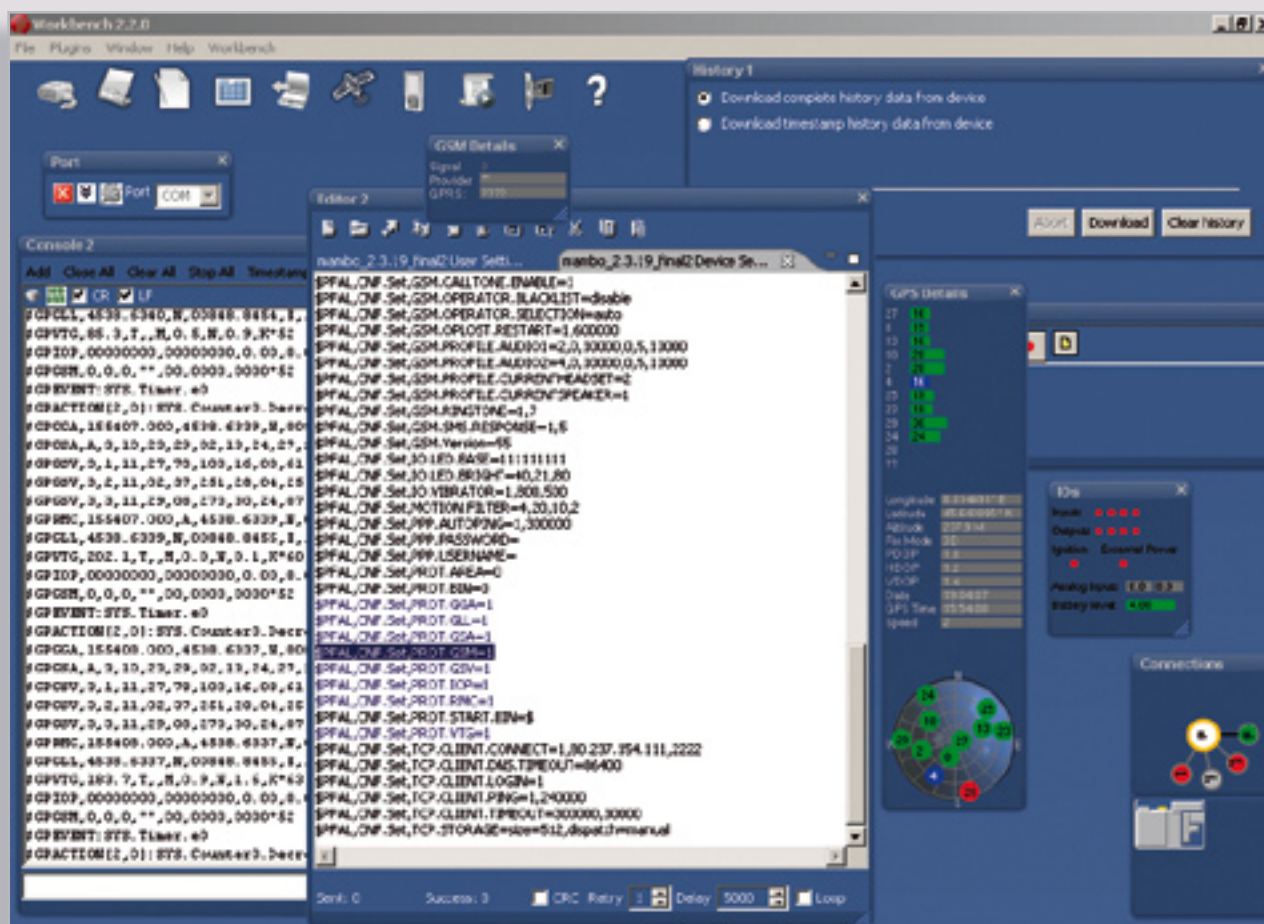
memoria del Mambo tramite l'interfaccia wireless Bluetooth.

Entrando nel dettaglio, l'interfaccia contiene una ricetrasmittente con potenza conforme alla classe 2, versione di protocollo BT 1.1 e profilo SPP (Serial Port Profile) integrato. L'antenna integrata garantisce la copertura entro una distanza di 10 metri circa.

Oltre a garantire la connessione col PC per la configurazione e consentire l'utilizzo di un auricolare wireless, il canale Bluetooth può servire per inviare i dati rilevati dal ricevitore GPS ad un Pocket PC o ad uno Smartphone equipaggiati con interfaccia analogica e dotati di un software di navigazione con la relativa cartografia. In tal modo il Mambo consente di realizzare un vero e proprio navigatore satellitare: basta entrare in auto e attendere che i due apparati si riconoscano, per poi aprire il canale dati col palmare e visualizzare sullo schermo di quest'ultimo la posizione o il percorso corrispondente.

Sensore di movimento

Il dispositivo integra un preciso sensore di movimento basato su un accelerometro capace di rilevare lo spostamento lungo i tre assi (verticale, orizzontale in avanti e di lato); il sensore è completamente configurabile da software, nel senso che nel programma da caricare nel Mambo



Definire semplicemente "programma" l'ambiente di sviluppo, programmazione ed utilizzo di MAMBO è piuttosto riduttivo: esso infatti dispone di tantissime opzioni per mezzo delle quali è possibile plasmare le funzionalità del sistema in relazione alle proprie necessità. Anche il nome scelto dalla stessa Falcom ne è la conferma: infatti la suite di programmi a corredo di MAMBO viene chiamata "Workbench", banco di lavoro.

Nel corso delle prossime puntate daremo ampio spazio a Workbench in quanto tutti gli esempi pratici che realizzeremo verranno progettati, trasferiti e verificati tramite questo tool. In questo riquadro - essendoci occupati a lungo nell'articolo della parte squisitamente hardware - vogliamo anticipare qualche informazione sull'ambiente di

sviluppo allo scopo di fornire una panoramica completa delle potenzialità di questo sistema. Naturalmente, in seguito, approfondiremo queste informazioni con esempi pratici e suggerimenti d'utilizzo.

Nell'immagine in alto abbiamo rappresentato alcune finestre particolarmente significative che andiamo a commentare.

Port: seleziona la COM port BT a cui MAMBO è virtualmente collegato.

GSM Details: fornisce indicazioni sul provider GSM al quale il ricevitore è connesso nonché sulla qualità del segnale.

Console 2: riporta la sequenza di dati e comandi scambiati tra PC e ricevitore. La visualizzazione è in formato testo e non c'è limite al numero di console apribili, sulle quali possono essere applicati filtri di visualizzazione.

Editor 2: permette di gestire la parte Firmware del MAMBO, ovvero leggere / scrive-

re il file di configurazione. Anche in questo caso si possono aprire più finestre di Editor.

History 1: permette di leggere il file che contiene le coordinate del percorso memorizzato quando il MAMBO viene usato come data logger.

GPS Details: rappresenta la visualizzazione grafica dell'immagine astrale della rete satellitare. Permette di ricavare informazioni sulla qualità del segnale, sul numero di satelliti visibili e sulla relativa posizione nella sfera celeste, nonché di indicare lo stato di navigazione in cui il GPS si trova.

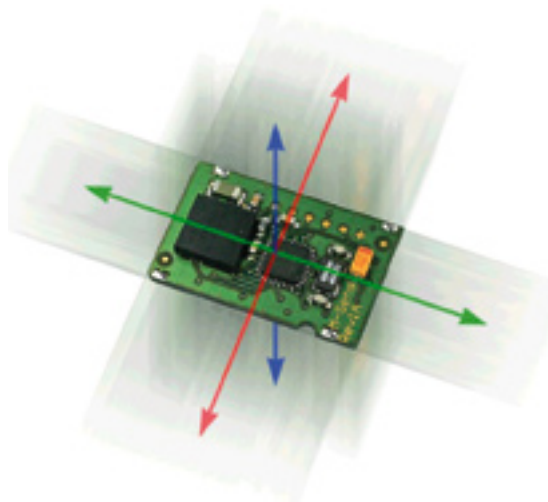
IOs: fornisce alcune indicazioni relative allo stato degli ingressi e delle uscite, oltre allo stato della batteria.

Connections: permette di collegare le finestre sia tra loro sia direttamente alla seriale del MAMBO.

si può definire cosa il dispositivo deve fare quando viene mosso. Ad esempio, si può decidere che l'unità sia normalmente in stand-by e

che, dopo un qualsiasi movimento, si risvegli ed inizi a funzionare normalmente; oppure che in caso di spostamento chiami una persona o

invii, tramite SMS, le coordinate in tempo reale o dopo un ritardo più o meno lungo. L'utilizzo del sensore per rilevare quando il dispositivo si ➤



L'accelerometro integrato a tre assi rileva gli spostamenti permettendo di attivare particolari funzioni come, ad esempio, l'invio di un SMS.

muove e la conseguente possibilità di abilitare molte funzioni solo in tal caso, è di grande aiuto per ottimizzare il consumo dell'energia immagazzinata nella batteria che, in pratica, si traduce nel miglioramento dell'autonomia d'esercizio che può andare ben oltre le ore dichiarate dal costruttore.

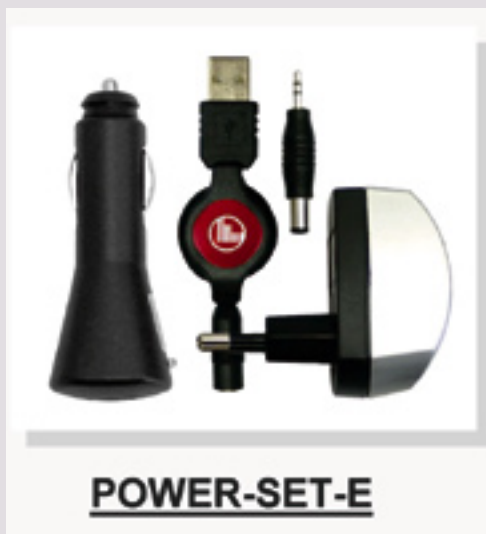
Alimentazione

Mambo è un dispositivo portatile e facilmente installabile a bordo di autoveicoli grazie all'impiego di una batteria ricaricabile a ioni di litio da 3,7 volt fornita di serie; il costruttore offre anche la possibilità di scegliere tra due capacità, ossia 850 mA/h (che garantisce un'autonomia di cinque ore) e 1700 mA/h (dieci



Grazie all'auricolare in dotazione (4), Mambo può essere usato per effettuare normali telefonate: si gestisce con i tasti 1, 2 e 3.

Gli accessori



MAMBO viene fornito con una nutrita dotazione di accessori. Infatti, oltre al ricevitore, disponibile sia in versione americana (la sezione GSM è soggetta a differenti regolamentazioni e l'adattatore di rete è dotato di presa bipolare a passo USA) che europea (nella foto in alto è visibile il POWER-SET-E in cui l'adattatore dispone di spina italiana), e al pacco batterie, la confezione include:

- auricolare con avvolgicavo a molla e pulsante di controllo;
- cordicella dotata di moschettoni, per trasportare al collo il MAMBO;
- cavo adattatore da USB a jack, da utilizzare per ricaricare la batteria;
- adattatore da jack stereo 2,5 mm a jack universale per ricaricare sia il cellulare che il ricevitore GPS/GSM;
- riduttore di tensione veicolare da collegare alla presa dell'accendisigari; fornisce una tensione di uscita di $5\text{ V} \pm 5\%$ con una corrente di 450 mA su un connettore USB e va collegato al MAMBO tramite l'apposito cavo adattatore;
- adattatore di tensione da parete disponibile sia con spina bipolare americana che bipolare europea, a seconda della confezione richiesta; anch'esso fornisce una tensione di uscita di $5\text{ V} \pm 5\%$ con una corrente di 450 mA con connettore di uscita tipo USB;
- CD di installazione della suite Workbench; la versione a corredo è sempre la più recente disponibile nel momento in cui viene preparata la confezione.

ore di funzionamento continuato). L'autonomia d'esercizio dichiarata vale per l'uso continuativo di tutte le funzioni, ma è chiaro che ottimizzando i consumi, ad esempio programmando lo stand-by quando non viene rilevato movimento dall'accelerometro, essa aumenta decisamente.

L'accumulatore è posizionato sotto il coperchio, sopra il lettore per la SIM e viene ricaricato attraverso un regolatore interno collegato al connettore d'alimentazione posto sull'involucro del dispositivo; di serie viene fornito un cavo USB per prelevare l'alimentazione necessaria alla ricarica dall'omonima porta di un computer, ma anche di un alimentatore da auto, provvisto di presa accendisigari, nonché di un alimentatore / caricabatteria da rete universale, funzionante a $100\div 240\text{ Vac}$, sia a 50 che a 60 Hz, utilizzabile quindi in tutto il mondo. Tanto l'adattatore da rete che quello da auto, rendono disponibile l'alimentazione per il Mambo mediante una presa USB della quale sono collegati i soli contatti + e -; ciò permette di introdurre direttamente il cavo in

dotazione al Mambo. Come accennato, la praticità dell'alimentazione a batterie si apprezza sia quando il dispositivo debba "viaggiare" nello zaino di un escursionista che si muove a piedi o in mountain-bike, sia nel caso si intenda tracciare il percorso di un autoveicolo; in una situazione del genere, la capacità di operare in piena autonomia (senza necessità di connessione all'impianto elettrico della vettura) si apprezza maggiormente quando è necessario occultare il localizzatore che così non può essere scoperto neanche andando a verificare il consumo a vuoto della batteria.

Questo metodo è uno dei più usati per scoprire se a bordo di un'auto è stato installato un localizzatore o un sistema di intercettazione ambientale, sebbene sia aggirabile usando dispositivi a batteria o in grado di spegnersi per alcune ore quando la batteria della vettura viene scollegata (operazione necessaria per collegare in serie all'impianto un amperometro per la misura).

Programmazione

Mambo è governato da un microprocessore RISC che esegue i pochi passi necessari a dialogare con il computer per poter caricare il programma di funzionamento prescelto; nella sua Flash EPROM è dun-

que collocato quello che possiamo considerare il suo BIOS, l'insieme delle istruzioni fondamentali per attivare una connessione Bluetooth e accettare le disposizioni che il programma di configurazione gli invie-

rà dal PC, che, evidentemente, dovrà a sua volta disporre di un'interfaccia Bluetooth. In altre parole, una volta acceso, il nostro apparecchio aspetta che gli si carichi il programma da eseguire: i tre pulsanti di cui dispone e i led bicolore posti sul coperchio non fanno inizialmente nulla, però possono essere personalizzati. Il software prevede la possibilità di assegnare a ciascun tasto una specifica funzione oppure un insieme di funzioni, ma inizialmente nessun tasto è attivo.

Più esattamente, va detto che il pulsante centrale serve all'accensione e allo spegnimento, ma per il resto, salvo diversa programmazione, non fa altro.

Oltre alle funzioni di localizzazione, invio delle coordinate da Bluetooth e tracciamento dei percorsi, il programma dà la possibilità di genera-

Sezione GPS

- Chipset SiRFStarIII di SiRF Technology (www.sirf.com);
- Frequenza: L1 a 1.575,42 MHz, C/A code a 1.023 MHz, con Front End ad alta sensibilità;
- Numero canali: 20 paralleli, spread spectrum;
- Sensibilità: -142 dBm in acquisizione; -158 dBm in tracking;
- Tempi di acquisizione: Cold Start 38 s, Warm Start 35 s, Hot Start 1 s;
- Precisione:
10 m CEP per la posizione, 0,1 m/s per la velocità, 1 μ s per il tempo;
- Correzione differenziale: WAAS.

Sezione Telefono

- Bande di frequenze GSM: 900 / 1800 / 1900 MHz
- Sezione GPRS: classe 10
- Modi operativi: Voce, Dati, SMS, TCP/IP, PPP

Sezione Bluetooth

- Versione protocollo: Bluetooth 1.2
- Classe di funzionamento: classe 2, 10 m in campo aperto.

Varie

- Batteria a ioni di litio, disponibile a scelta da 850 o 1600 mA/h
- Dimensioni: 85 x 59 x 25 mm, peso 70 g.
- Microfono ed altoparlante integrati.
- Jack da 2,5 mm a 4 poli per auricolare
- Tre pulsanti e tre led con funzionalità definibile da programma.
- Sensore di movimento (accelerometro a tre assi) integrato.





Per gli sviluppatori è disponibile un'elegante valigetta demo la quale contiene, oltre alla suite Workbench, entrambe le versioni di MAMBO (europea ed americana) nonché tutti gli accessori.

re un allarme se il Mambo esce da una zona definita sulla cartografia; una sorta di allarme che permette di avvertire chi deve sorvegliare una persona o un veicolo se questi si allontanano da un'area predefinita. Tale funzione è di grande importanza in molti ambiti: nel controllo veicolare permette di verificare se l'autista di un camion, di un autobus di linea o di un qualsiasi veicolo aziendale durante la giornata "divaga" ed esce dal proprio percorso; inoltre consente di sapere se l'auto in cui il Mambo è installato viene portata via e/o si allontana dalla zona nella quale dovrebbe trovarsi, ovvero se varca i confini di una regione o dello Stato. Nell'ambito del controllo delle persone, può essere usato per av-

vertire il Soccorso Alpino o chi di dovere se un escursionista o uno scalatore esce dalla zona considerata sicura e si va ad avventurare in luoghi che potrebbero presentare il rischio di frane, valanghe o costituire pericolo per la presenza di animali che possono aggredire l'uomo, e via di seguito. In tutti i casi esposti, il localizzatore, rilevato l'allontanamento dalla zona impostata mediante un apposito software di cui parleremo nella prossima puntata, può chiamare il personale ad-

detto alla sorveglianza del veicolo o della persona che lo porta con sé, ovvero inviare un SMS che avverta della situazione, cosicché si possano prendere i provvedimenti del caso (ad esempio una telefonata all'escursionista per avvertirlo di tornare indietro).

Per impostare i limiti della zona va utilizzato il solito programma di configurazione: questi dati vanno poi trasferiti (via Bluetooth) con l'apposita funzione nella memoria del Mambo.

Il dispositivo è molto versatile e lo dimostra il fatto che di esso è possibile personalizzare qualsiasi funzione, tasti e led di segnalazione compresi; tuttavia va detto che presenta una limitazione, se vogliamo, concettuale: i progettisti che l'hanno sviluppato, forse per un po' di presunzione, hanno realizzato istruzioni rigide; in pratica, i comandi per impostarne il funzionamento sono specifici per ogni condizione, quindi va da sé che il Mambo può essere programmato per eseguire determinate operazioni e solo quelle.

Non è possibile impostare combi-

nazioni tra le varie funzioni che non siano state previste da un apposito comando.

Diciamo presunzione perché in effetti chi ha progettato il Mambo ha probabilmente pensato di aver previsto tutte le possibili situazioni di impiego e in effetti, dando uno sguardo al set di comandi, sembra proprio così; sarebbe stato comunque auspicabile avere a disposizione delle istruzioni più flessibili.

Il discorso sulla programmazione del dispositivo si comprende facendo un paragone con i linguaggi di programmazione dei microprocessori: solitamente ogni istruzione è associata a dei parametri che permettono di usare un comando per fare diverse cose, a seconda dei parametri definiti di volta in volta; invece, nella programmazione del Mambo per ogni azione esiste un comando specifico, quindi due azioni simili non possono essere impostate impartendo la stessa istruzione con un diverso parametro.

Questo spiega perché la lista delle istruzioni è decisamente lunga.

Per fare un esempio, se con la pressione del tasto 1 si desidera far partire una telefonata diretta ad un certo numero, per abbinare al predetto pulsante un'altra funzione oltre alla telefonata (per esempio l'invio di un SMS) non si può utilizzare il medesimo comando ma bisogna impartirne uno a sé stante.

Nella prossima puntata analizzeremo il software di programmazione col quale si sviluppa il programma da caricare nel Mambo e con cui, a stesura ultimata, lo stesso programma viene caricato via Bluetooth; avremo così modo di vedere quanto accennato relativamente alla struttura del set dei comandi.

La programmazione si esegue su qualunque Personal Computer mediante il software fornito direttamente dalla casa costruttrice, che opera in ambiente Microsoft Windows.

Lettore di badge magnetici con porta USB

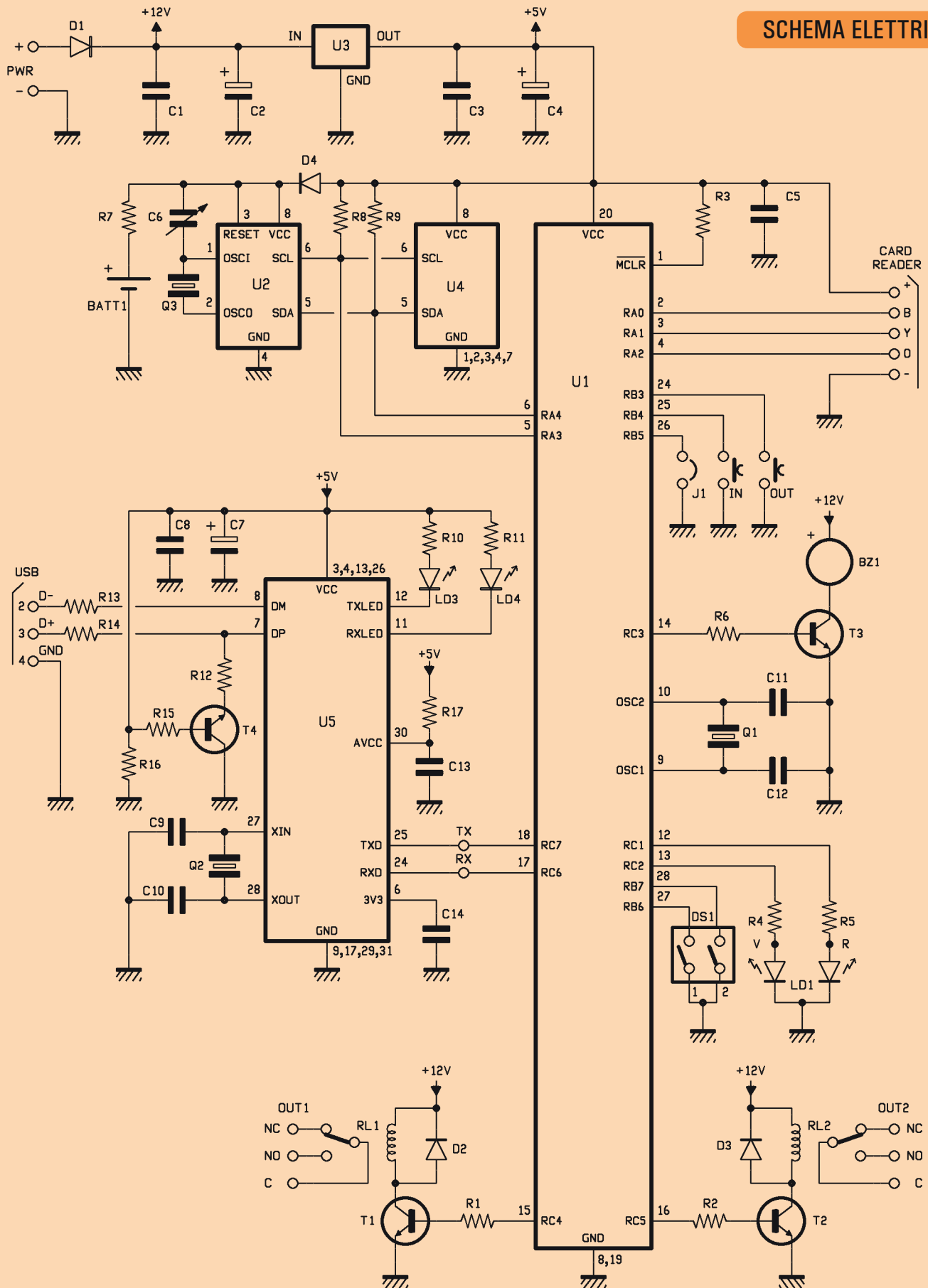
di Matteo Aldizio

Nuovo lettore di badge magnetici USB dotato di funzionalità evolute. Integra un registratore orario di entrata / uscita e memorizza fino a 5300 eventi di lettura nella EEPROM di cui è dotato. È in grado di riconoscere fino a 20 badge memorizzati. In base ai dati della tessera può decidere in modo autonomo di comandare due relé o attendere direttive dal computer. È dotato di orologio con batteria tampone.



I lettori di tessere magnetiche trovano impiego in molteplici applicazioni. Per questo motivo più di una volta abbiamo proposto progetti di questo tipo; l'ultimo in ordine di tempo risale al fascicolo dell'ottobre 2003. Il circuito presentato in quell'occasione (codice progetto FT500) era in grado di riconoscere i dati registrati nella seconda traccia dei badge e, se questi venivano riconosciuti come validi, poteva attivare o disattivare dei relé. Poiché in questo nuovo lettore abbiamo deciso di implementare molte funzioni che avevamo sviluppato per l'FT500, suggeriamo, a quanti sono in possesso

del fascicolo 85, di andare a rivedere questo progetto. Quello che proponiamo in queste pagine è una versione ancora più performante del progetto di quattro anni fa, in grado di offrire due nuove funzionalità: la memorizzazione dell'orario in cui la tessera è stata strisciata e l'aggiunta di funzionalità utili alla realizzazione di un controllo presenze. Questo risultato è stato raggiunto con delle modifiche sia hardware che firmware, la più importante delle quali è probabilmente l'implementazione di una porta di comunicazione USB, tanto funzionale quanto indispensabile dal momento che molti PC >



non dispongono più di una seriale. Per il collegamento al PC abbiamo sviluppato un protocollo molto semplice con dei comandi di testo intuitivi, simili a quelli AT dei modem; questi comandi possono essere facilmente implementati in software dedicati da realizzare in VB, Delphi o tool di sviluppo simili.

Le nuove funzionalità

La nuova versione del lettore di badge, come già anticipato, è dotata di una porta USB, in sostituzione della vecchia porta seriale, e strutturalmente si differenzia perchè sono stati aggiunti un orologio e una memoria che dialogano con il microcontrollore.

L'orologio, un PCF8593 in formato DIL8, integra un bus I2C attraverso il quale scambia le informazioni con il PIC: al suo interno sono presenti tutti i registri necessari per conteggiare anno, mese, giorno, ore, minuti, secondi e comunicarli al micro su sua richiesta. Abbiamo anche previsto l'impiego di una batteria che permette di mantenere attivo l'orologio in mancanza di tensione di rete anche per tempi molto lunghi. Sempre a proposito di questa sezione, ricordiamo che il chip RTC (real time clock) utilizzato non è in grado di passare automaticamente dall'ora legale a quella solare e viceversa.

Sul circuito stampato abbiamo previsto un condensatore variabile con cui eseguire la regolazione fine della frequenza di clock al fine di ridurre al minimo l'errore dell'orologio. Per regolare la frequenza generata conviene in un primo momento aggiornare l'ora tramite il computer e verificare, dopo almeno una settimana, che l'orario non si discosti troppo da quello reale. Nel caso fosse necessario ritoccare la frequenza, consigliamo di eseguire piccole correzioni ogni tanto cercando, per approssimazioni successive, di ri-

ducere al minimo lo scarto. Sicuramente la soluzione ottimale per la regolazione del clock consiste nell'impiego di un frequenzimetro con almeno cinque cifre (l'RTC utilizza un oscillatore a 32.768 Hz) ma non tutti dispongono di uno strumento del genere. Dobbiamo tuttavia considerare che se anche la frequenza fosse leggermente fuori centro, l'errore che ne deriverebbe sarebbe minimo (ad esempio, un errore di 0,5 Hz del clock comporta una differenza di 8 minuti in un anno, facilmente compensabile con un riallineamento settimanale o mensile effettuato tramite la porta USB del PC).

Così come per l'RTC, anche la memoria esterna, una 24LC256 in cui viene tenuta traccia dello storico degli accessi e delle presenze, è in formato DIL8 e integra l'interfaccia I2C. La sua capacità è decisamente superiore a quella della EEPROM integrata nel micro, disponendo infatti di 32 kbyte anziché di soli 256 byte con possibilità di memorizzare più di 5000 timbrature.

Nel circuito sono previsti anche due pulsanti che consentono di discri-

minare tra "entrata" e "uscita": premendo il primo pulsante e strisciando la tessera, il sistema memorizza l'ora dell'operazione, il codice della tessera e un flag che indica l'entrata (o l'uscita nel caso venga premuto il secondo pulsante). È anche possibile utilizzare la nuova scheda con le stesse funzioni della vecchia (chiave normale, PC-Link, ecc..) semplicemente aprendo un ponticello.

Modi operativi

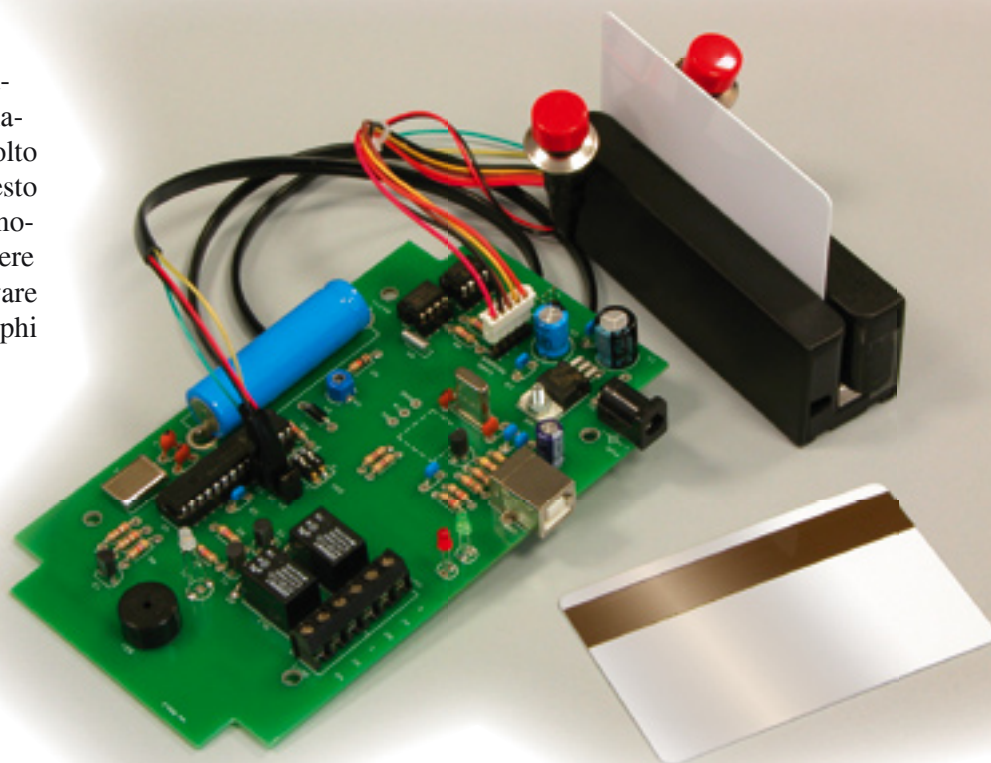
Come già anticipato, questo nuovo lettore può operare sia in modo autonomo, esattamente come faceva la versione precedente della quale peraltro eredita tutte le funzioni, sia in modalità assistita per le funzioni di controllo accessi, aggiunte in questa nuova versione del progetto.

Analizziamo ora le caratteristiche di entrambi le modalità operative.

La compatibilità con il progetto FT500 è ottenuta avviando il dispositivo con il ponticello J1 aperto.

Modalità FT500

Il lettore di badge può funzionare sia autonomamente per realizzare un ➤

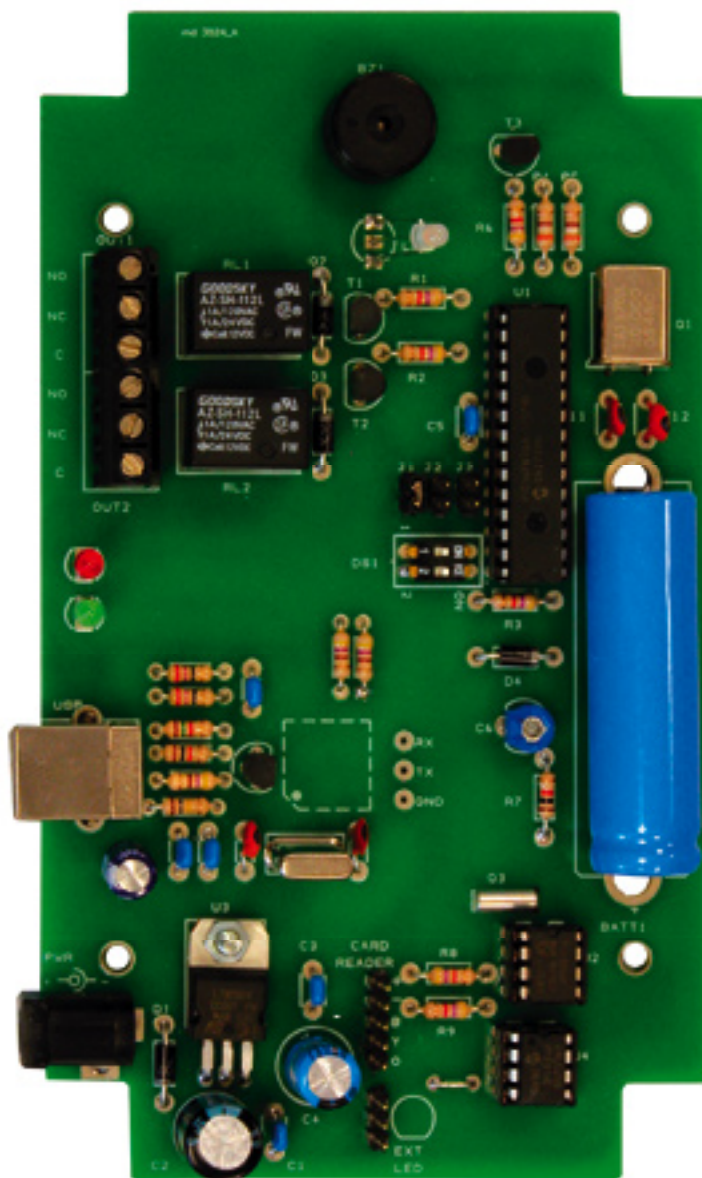


ELENCO COMPONENTI:

R1÷R3: 4,7 kohm
 R4, R5: 390 ohm
 R6, R8, R9: 4,7 kohm
 R7: 1 kohm
 R10, R11: 470 ohm
 R12: 1,5 kohm
 R13, R14: 27 ohm
 R15: 10 kohm
 R16: 100 kohm
 R17: 470 ohm
 C1, C3, C5, C8: 100 nF multistrato
 C2: 470 µF 25VL elettrolitico
 C4: 470 µF 25VL elettrolitico
 C6: compensatore 2-6 pF
 C7: 220 µF 16VL elettrolitico
 C9: 15 pF ceramico
 C10: 15 pF ceramico
 C11, C12: 10 pF ceramico
 C13, C14: 100 nF multistrato
 LD1: led 3mm bicolore
 LD2: led 3mm rosso
 LD3: led 3mm verde
 D1÷D4: 1N4007
 Q1: quarzo 20 MHz
 Q2: quarzo 6 MHz
 Q3: quarzo 32,768 kHz
 U1: PIC16F876 (MF675)
 U2: PCF8593
 U3: 7805
 U4: 24LC256
 U5: FT232BM
 T1÷T4: BC547
 DS1: dip switch 2 vie
 BZ1: buzzer con elettronica
 RL1, RL2: rele miniatura 12V
 BATT1: batteria ricaricabile
 NiCd 1,2V-700 mAh
 Varie:
 - plug alimentazione

- Strip maschio 5 vie
 - Strip maschio 2 vie (3 pz.)
 - Jumper (3 pz.)

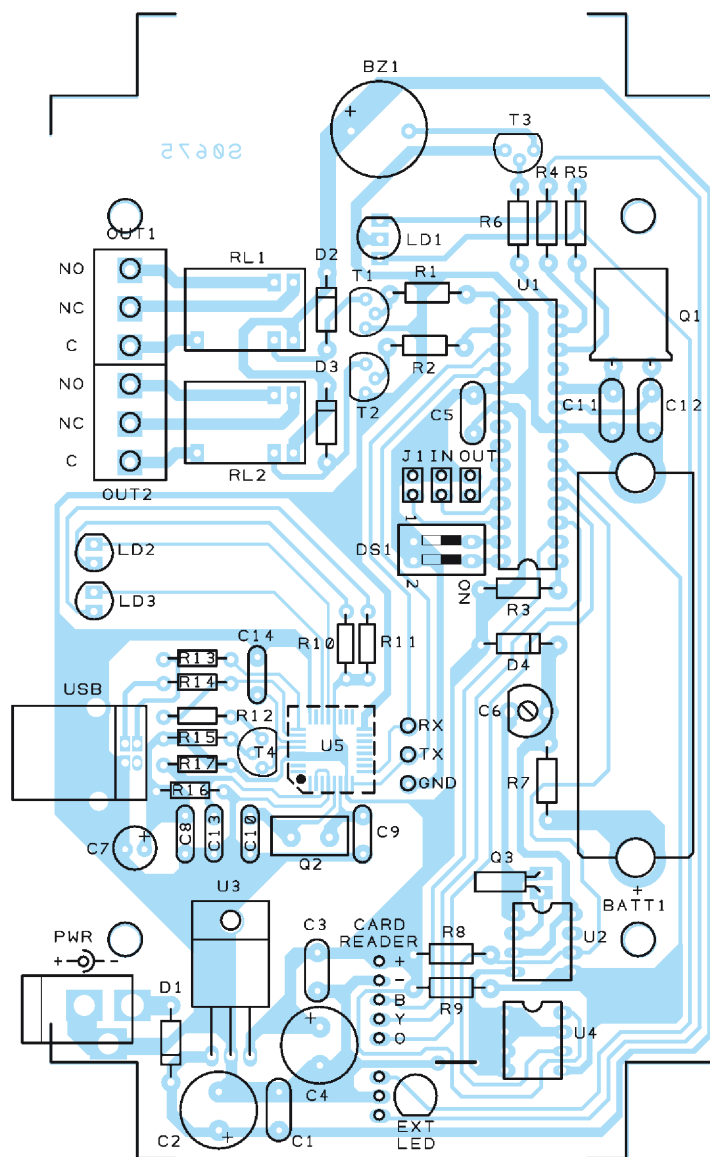
- connettore USB-B
 - morsetteria 3 poli (2 pz.)
 - lettore di tessere magnetiche LSB12



sistema di controllo degli accessi, quando è collegato, ad esempio, ad elettroserrature o sistemi di allarme (modalità Chiave), sia collegato ad un PC a cui demanda la gestione degli eventi (modalità *PC Link diretto* oppure *PC On Demand*). La selezione della modalità di funzionamento è identica a quella della versione FT500 (da cui la compatibilità) essendo controllata mediante

due dip-switch. Vediamo tutte le funzionalità offerte da questa modalità aiutandoci con la tabella di pagina 78. Mentre DIP1 definisce la funzione (*Chiave* o *PC Link*), DIP2 stabilisce delle sotto-modalità operative. Con DIP1 su ON si seleziona la modalità *Chiave* che consente di attivare l'uscita di controllo dell'elettroserratura se il codice della tessera

viene riconosciuto (in caso contrario viene attivata l'uscita di allarme). Stante questa condizione, agendo sul secondo dip switch (DIP2) è possibile porre il sistema in modalità *Riconoscimento* (DIP2 a ON) oppure *Apprendimento* (DIP2 a OFF). In quest'ultimo caso, strisciando la tessera nel lettore, il PIC ne memorizza il codice e lo considera come valido e abilitato. La memoria interna del



- filo 3 poli 10 Cm
- Zoccolo 14+14
- Zoccolo 4+4 (2 pz.)

- vite 3 MA 8 mm
- dado 3 MA
- circuito stampato cod. S0675

micro consente di memorizzare al massimo venti tessere. Per cancellare tutti i dati presenti nel PIC è sufficiente impostare DIP1=ON e DIP2=OFF, quindi spegnere e riaccendere il lettore.

Per selezionare la modalità *PC Link*, occorre invece impostare DIP1 a OFF. In questo caso se DIP2 è a OFF si seleziona l'opzione *PC Link diretto*, in caso contrario (DIP2

a ON) il sistema funziona in modalità *PC Link On Demand*.

Il modo *PC Link diretto* prevede la trasmissione immediata di tutti i codici delle tessere lette al PC tramite porta USB; sarà quindi compito del programma di gestione decidere che azione intraprendere.

Nel riquadro di pagina 79 riportiamo il set di comandi utilizzato dal PC per interagire con la scheda. La

variante *PC Link On Demand* permette invece di memorizzare gli eventi e di inviarli al PC solamente a fronte di un'interrogazione operata dal programma gestionale. Rimane sempre la possibilità per il PC di controllare alcune periferiche del lettore quali i relè o il buzzer.

Modalità FT675

Si tratta sicuramente di una modalità adatta ad utenti esperti, in quanto, in questo caso, il circuito si comporta da semplice lettore demandando al programma che "gira" su PC tutte le decisioni. Il circuito legge i dati contenuti nella scheda appena strisciata e li memorizza in EEPROM, associando ad essi data ed ora nonché l'informazione relativa all'entrata o all'uscita.

I dati memorizzati vengono trasferiti al PC quando quest'ultimo ne fa richiesta inviando il relativo comando. Con un altro comando è possibile cancellare completamente la memoria dell'EEPROM. I dati ricevuti dal PC possono successivamente essere analizzati, elaborati e salvati dal programma di gestione.

Appare evidente che in questa modalità il circuito non si comporta più come sistema autonomo in quanto non è più suo compito stabilire la validità del badge letto e attivare le uscite. L'elaborazione dei dati è affidata al programma di gestione utilizzato, programma che l'utente deve realizzare in relazione alle proprie esigenze. Nella tabella di pagina 80 riportiamo l'elenco di tutti i comandi disponibili per realizzare un software personalizzato.

Un semplice programma potrebbe, ad esempio, interrogare ciclicamente il lettore, memorizzare i dati in una tabella e, a fine giornata, segnalare le persone entrate in ritardo o che hanno fatto degli straordinari. Un programma più complesso potrebbe rilevare le presenze e le assenze di tutto un mese, le ore straordinarie, e tutti gli altri elementi temporali

Modalità FT500 compatibile

DIP 1	DIP 2	Modalità	Note
ON (Chiave)	ON (Riconoscimento)	Chiave, Riconoscimento	Attiva / Disattiva i Relé 1 e 2 a seconda che il codice della tessera sia già stato memorizzato o meno.
ON (Chiave)	OFF (Apprendimento)	Chiave, Apprendimento	Utilizzato per memorizzare nuove tessere o per cancellare completamente la memoria del micro. In questo caso la scheda deve essere accesa con i dip in questa configurazione.
OFF (PC link)	ON (On demand)	PC Link, On Demand	Il lettore riconosce la tessera e la memorizza nella EEPROM, informando comunque il computer ed attendendo istruzioni da esso.
OFF (PC link)	OFF (Diretto)	PC Link, Diretto	Il lettore legge la tessera (anche se sconosciuta), invia informazioni al PC ed attende istruzioni da eseguire.

Quando configurato per operare in emulazione FT500, il lettore di badge magnetici presenta due differenti modalità di funzionamento:

- Chiave o Riconoscitore, in cui può essere utilizzato autonomamente come controllore di porte, elettroserrature, cancelli, ecc.;

- PC Link, in cui il lettore viene collegato attraverso la porta USB ad un PC. L'eventuale memorizzazione degli accessi e la gestione di altre periferiche viene quindi demandata ad un opportuno software presente nel computer.

La selezione tra le due modalità di funzionamento avviene tramite il dip switch DIP1 secondo la logica riassunta in tabella:

DIP1 = ON, funzionamento Chiave o Riconoscitore

DIP1 = OFF, funzionamento PC Link.

Per ognuna delle due modalità, è possibile selezionare tramite DIP2 due ulteriori metodi di funzionamento. Iniziamo con le sottomodalità della funzione Chiave (DIP1=ON).

DIP2 = ON, sottomodalità "Normale". Il circuito legge le informazioni scritte nelle tessere che vengono strisciate e, se il codice della tessera è presente nella memoria del microcontrollore, per un certo periodo di tempo viene acceso il led verde e attivato il Relé 1; se, invece, il codice della tessera non è presente in memoria, viene acceso il led rosso e viene attivato il Relé 2. Occorre precisare che le durate di attivazione dei relé e dei led possono essere modificate via software, collegando il circuito tramite la porta USB ad un PC; i tempi specificati vengono memorizzati all'interno della EEPROM del PIC e utilizzati per i successivi accessi; i due periodi predefiniti sono di 1 secondo per entrambi i relé.

DIP2 = OFF, il lettore lavora secondo la sottomodalità "Apprendimento"; questa è utilizzata per memorizzare nel PIC i codici

delle tessere che devono essere abilitate. Attivando questa procedura, quando si passa una tessera nel lettore, il relativo codice viene memorizzato nel PIC e quindi considerato valido; come indicazione per l'utente, il sistema attiva per circa un secondo sia il buzzer che il led arancione. Naturalmente, se una tessera è già presente, questa non viene nuovamente memorizzata (situazione indicata da 5 lampeggi rossi del led). Il PIC dispone di una capacità massima di 20 tessere; se la memoria è già piena non sarà ovviamente possibile aggiungere nuovi codici (situazione indicata da 10 lampeggi arancioni del led). Infine è possibile cancellare completamente la memoria del microcontrollore; ciò avviene se si accende il dispositivo con DIP1 = ON e DIP2 = OFF.

Questa condizione viene indicata da alcuni veloci lampeggi del led rosso.

Vediamo ora la modalità PC Link (DIP1 = OFF).

DIP2 = OFF attiva il modo "Diretto" in cui il dispositivo funziona come lettore "trasparente" di tessere: qualsiasi codice rilevato dal lettore, viene inviato via porta seriale al PC; sarà poi il programma presente sul PC a decidere come gestire i dati ricevuti.

DIP2 = ON seleziona il funzionamento "On Demand". Con questa sottomodalità è possibile memorizzare nella EEPROM del microcontrollore un certo numero di eventi e scaricarli successivamente, mediante porta USB, nel PC. Inoltre, anche in questa modalità operativa, è possibile ordinare via software al circuito l'esecuzione di alcuni comandi; il dispositivo invia continuamente al PC la stringa "INSERISCI COMANDO" ed attende per circa 500 msec l'invio dello stesso. I comandi possono essere utilizzati per attivare i due relé, per leggere una o tutte le celle di memoria, eccetera.

necessari per la compilazione della busta paga.

Ovviamente, anche in questa modalità, esiste sempre la possibilità per il programma di attivare alcune periferiche della scheda, ad esempio, per far suonare una sirena che scandisca l'inizio e la fine dei turni di lavoro.

Schema elettrico

Il circuito del nuovo lettore di badge è molto semplice grazie all'impiego di un potente microcontrollore al quale sono affidate la maggior parte delle funzioni. Osservando lo schema possiamo identificare facilmente cinque blocchi funzionali: il

microcontrollore (U1), il gruppo periferiche costituito da orologio (U2) e memoria (U4), la porta USB (U5), la sezione di attuazione e interfaccia utente (relé, buzzer, pulsanti, LED e lettore di badge) ed infine lo stadio di alimentazione (U3).

Iniziamo la descrizione dal micro-

Modalità FT500 compatibile: elenco comandi

Il lettore di badge magnetici può funzionare sia in modalità PC Link diretto in cui legge e immediatamente trasmette i codici presenti nelle tessere, sia in modalità PC Link On Demand in cui memorizza localmente i dati letti ed attende la richiesta del computer prima di trasmetterli.

In questa seconda modalità il PC può richiedere al lettore anche l'esecuzione di alcuni comandi. Per ognuna delle due opzioni è quindi necessario scrivere un software che permetta al PC di colloquiare col dispositivo. In questo riquadro illustriamo il protocollo adottato dal sistema per comunicare: ciò consente di realizzare facilmente un programma di gestione (in VB, Delphi, ecc.) che risponda alle proprie esigenze.

L'invio e la ricezione dei dati avviene via porta USB utilizzando un baud rate di 19.200 bit/sec con codifica 8-N-1 senza alcun controllo di flusso. Inoltre, per ricevere i comandi e per inviare le proprie risposte, il sistema utilizza sempre dei caratteri

MAIUSCOLI in formato ASCII e alla fine delle risposte aggiunge i caratteri speciali "A CAPO" e "INVIO" (valore numerico decimale 13 e 10) per indicare la fine delle stringhe.

All'accensione il sistema invia la stringa SYSTEM STARTUP ad indicare che è in corso l'inizializzazione. Nella modalità PC Link diretto quando viene letta una tessera il circuito invia la stringa CODICE LETTO ->x<- in cui al posto della x vi sono i byte del codice; i caratteri ">" e "<" sono stati inseriti per facilitare al software il compito di riconoscere il codice all'interno della risposta. Questa stessa tecnica è stata utilizzata anche negli altri casi in cui si rende necessario individuare la posizione del codice all'interno di una stringa.

Nella modalità PC Link On Demand il dispositivo continua ad inviare la stringa INSERISCI COMANDO ed attende per 500 msec l'invio dello stesso. Nella seguente tabella riportiamo tutti i dettagli del protocollo di comunicazione.

Comando	Funzione	Note	Formato Risposta
* /L c	Accende/Spegne il LED	"c" indica il colore del led. Può valere V (verde), R (rosso) o A (arancione). Il funzionamento segue la logica Toggle.	
* /R n t1 t2	Attivazione dei relè	"n" indica il Relè (1 o 2) mentre "t1" e "t2" indicano il tempo di attivazione (in secondi). Se si specifica t1=0, t2=0 il relè funziona in modalità bistabile. Facciamo notare che ad ogni invio del comando, il tempo specificato viene memorizzato dal PIC ed utilizzato anche nella modalità Chiave-Normale.	
* /B t	Attivazione del buzzer	"t" indica il tempo di attivazione (da 1 a 9) del buzzer dove ciascuna unità vale 500 msec.	
* /C	Cancellazione della memoria	Esegue la cancellazione della memoria.	CANCELLAZIONE MEMORIA
* /N	Numero codici memorizzati	Il lettore comunica al PC quante tessere sono memorizzate nel PIC.	NUMERO TESSERE MEMO seguito da TESSERE MEMORIZZATE N.
* /P d1 d2	Scarica un singolo codice	Il lettore comunica al PC il codice della tessera memorizzata nella posizione "d1" e "d2" dove d1 indica la prima cifra e d2 la seconda.	SCARICO LA POSIZIONE d1 d2 seguito da CODICE LETTO -> x <-.
* /T	Scarica i codici memorizzati	Il lettore comunica al PC i codici di tutte le tessere memorizzate.	SCARICA TUTTO seguito dai codici nel formato CODICE LETTO -> x <-.

controllore PIC16F876, il quale può funzionare in due differenti modalità a seconda che il ponticello J1 sia aperto (modalità FT500) oppure chiuso (modalità FT675); in quest'ultimo caso vengono attivate le nuove funzioni di controllo presenze. Oltre ai dati della tessera

(acquisita dal lettore collegato alle linee RA0, RA1 e RA2), il micro verifica lo stato dei due pulsanti con i quali l'utente comunica al sistema che la lettura del badge è relativa ad una entrata o una uscita dal luogo di lavoro. A questo punto il processore deve memorizzare l'evento che si è

appena verificato, per cui, attraverso il bus I2C, effettua una serie di operazioni sul gruppo periferiche. Innanzitutto recupera data e ora dall'RTC U2, quindi genera una stringa di dati che si riferisce alla tessera, e all'ora dell'evento; infine, sempre attraverso lo stesso bus, memorizza



Modalità FT675: elenco comandi

Come già specificato nell'articolo, in modalità FT675 è disponibile un nuovo set di comandi evoluti con i quali il lettore di badge diventa a tutti gli effetti una periferica del PC in grado di memorizzare eventi che poi possono essere scaricati tramite la porta USB. Per semplicità, il nuovo set di comandi è stato strutturato per essere sintatticamente simile al set di comandi già visto per la modalità FT500 pur essendo più articolato.

La ragione di ciò è molto semplice: il set di comandi è ottimizzato per operare congiuntamente ad un software, eventualmente sviluppato, ad esempio, in ambiente Windows.

Un software dedicato consentirebbe di interagire facilmente e

velocemente con il lettore di badge, sia per ciò che concerne lo scarico del log di accessi, sia relativamente alla configurazione del sistema e/o all'attuazione dei relè. Di conseguenza, per consentire una notevole flessibilità di funzionamento, abbiamo dovuto espandere il set comandi rendendo meno specializzate le funzioni e delegando al PC un maggior "potere decisionale" nei confronti del lettore di badge. Comunque nessuna paura. Alla fine, il sistema è un lettore di badge per cui, anche se abbiamo aumentato il numero di comandi, essi rimangono molto intuitivi.

Analizziamoli dettagliatamente qui di seguito.

Comando	Funzione	Note	Formato Risposta
*/V	Richiesta Versione FW		Versione 1.00
*/O	Richiesta Data/Ora		23/09 14.47
*/P gg/mm hh.mm	Impostazione ora	Attenzione a rispettare la sintassi: dopo il comando va messo uno spazio.	
*/D	Richiesta numero accessi		55 (ad esempio)
*/N	Richiesta Data/Ora	Relativa all'ultima interrogazione mediante comando */O	23/09 14.47
*/C	Azzeramento contatore strisciate	Azzera la memoria di conteggio	Reset memoria completato
*/Q	Lettura memoria Entrate e Uscite	Riporta le sole strisciate relative ad una entrata o una uscita	NUMERO DI STRISCIAE = 2 23/04 12.13 CODICE IN = 22050 23/04 18.10 CODICE OUT = 22050
*/Rn	Attivazione relé	n=0, Relé1 OFF, Relé2 OFF n=1, Relé1 ON, Relé2 OFF n=2, Relé1 OFF, Relé2 ON n=3, Relé1 ON, Relé2 ON	
*/r	Richiesta stato relé		n (un numero che vale da 0 a 3)
*/i	Lettura completa della memoria	Riporta tutta la memoria, locazione per locazione, in formato decimale	1 6 132131133134135133 5331 31988 255255255255255255 5332 31994 255255255255255255

la stringa sulla EEPROM esterna U4. Termina qui il processo di memorizzazione dell'evento: il sistema è quindi pronto per una successiva registrazione o per rispondere ai comandi inviati dal PC (scarico log, cancellazione, attivazione periferiche, ecc.).

L'ultimo blocco da descrivere è la sezione di alimentazione: il circuito deve essere alimentato a 12 V in continua. Tale tensione va applica-

ta al morsetto power e, attraverso il diodo di protezione da accidentali inversioni di polarità (D1), raggiunge prima il gruppo di filtro costituito da C1 e C2, quindi lo stabilizzatore lineare di tensione (U3, un comune 7805). La tensione viene ulteriormente filtrata dal filtro composto da C3-C4 per poi essere distribuita a tutti gli elementi attivi del circuito. Prestate attenzione al circuito di alimentazione dell'orologio: la

tensione a 5 V giunge sia all'RTC che alla batteria di backup tramite il diodo D4. In questo modo, a circuito alimentato, la batteria viene ricaricata attraverso la resistenza R7 da 1 kohm mentre, in assenza della tensione di rete, l'RTC viene alimentato dalla corrente erogata dalla batteria tramite la resistenza R7 la cui presenza non determina alcuna caduta di tensione grazie al bassissimo assorbimento del chip. In questo

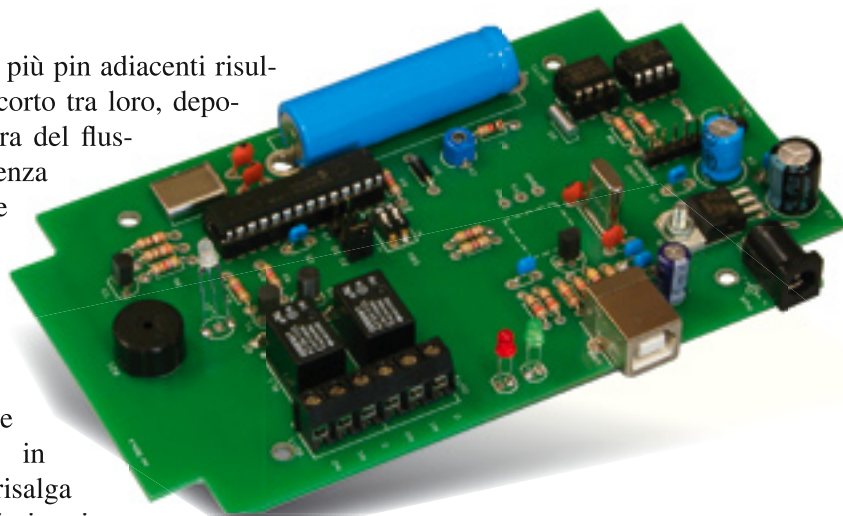
caso, la presenza di D4 polarizzato inversamente, impedisce che il resto del circuito venga alimentato.

Il montaggio

Fatta eccezione per il convertitore USB/seriale (FT232BM, U5, unico componente in SMD), il montaggio del circuito non presenta particolari difficoltà; consigliamo perciò di iniziare proprio con tale chip, il quale, tra l'altro, va montato sul lato saldature. Munitevi quindi di una pinzetta a becchi, di una lente di ingrandimento, di un saldatore a punta molto sottile (0,5 mm sarebbe ottimale) e di fluxante in gel.

Identificate sullo stampato il punto dove va posizionato il chip, depositate un leggero strato di stagno su due piazzole dello stampato diagonalmente opposte quindi, usando la pinzetta, posizionate il chip U5, sciogliete lo stagno sulle due piazzole staginate in precedenza e fissate i relativi pin di U5 controllando l'allineamento dei piedini. Durante questa fase, verificate che i pin coincidano perfettamente e, se necessario, prima della saldatura, correggete attentamente l'allineamento. Lasciate quindi raffreddare il chip per qualche secondo. Depositare ora una quantità di fluxante sufficiente a coprire tutta la fila di piedini che state per saldare. Scaldate con il saldatore i pin e fate sciogliere su di essi pochissimo stagno. Per effetto del fluxante esso aderirà sia ai pin del chip che alle piazzole. Se per

caso due o più pin adiacenti risultassero in corto tra loro, depositate ancora del fluxante e, senza aggiungere stagno, scaldate i pin in corto controllando che lo stagno in eccesso risalga in parte sui pin e in parte sulla punta del saldatore. Se necessario ripetere più volte la procedura, avendo cura di non fare troppa forza con la punta e di non scaldare troppo a lungo i pin e le piazzole. A questo punto, saldato il chip SMD, il resto è una passeggiata in pineta. Montate per primi tutti gli elementi a basso profilo: resistenze, diodi, condensatori non elettrolitici, condensatore variabile e dip-switch. Prestate attenzione ai diodi: essi sono polarizzati e quindi la fascia stampigliata sul corpo (catodo) deve coincidere con quella riportata nel piano di montaggio. Potete ora procedere con gli zoccoli dei circuiti integrati facendo attenzione alla tacca di riferimento riportata sia sullo zoccolo che sul disegno. Continuate con i transistor (posizionate il lato piatto come indicato), i quarzi, i led (rispettando ancora una volta la polarità) e lo stabilizzatore di tensione. Per quest'ultimo, piegate a 90° i suoi pin in modo che possa



essere fissato meccanicamente allo stampato anche con la vite da 3 MA oltre che con la saldatura dei terminali. Mancano ora solo i componenti più ingombranti e precisamente il buzzer, i connettori, i condensatori elettrolitici (polarizzati, quindi attenzione ancora una volta al verso), i relé e la batteria.

Prima di inserire i chip sui rispettivi zoccoli date un'occhiata finale alla ricerca di eventuali errori; successivamente innestate il connettore del lettore di badge sul pin strip "CARD READER".

Ora potete fornire alimentazione al circuito: impostate i dip-switch per la funzionalità desiderata e, passando un badge nel lettore, verificate che il circuito si comporti come descritto.

Ricordiamo che, per alcune applicazioni, la traccia 2 del badge va precedentemente incisa con un codice univoco utilizzando un apposito programmatore.

per il

MATERIALE

Il progetto descritto in queste pagine è disponibile in scatola di montaggio (cod. FT675K) al prezzo di 76,00 Euro. Il kit comprende tutti i componenti, la basetta forata e serigrafata, il lettore di badge, il microcontrollore programmato, i relé, la batteria e le morsettiere da circuito stampato. Il solo micro programmato (cod. MF675) costa 18,00 Euro. Separatamente sono disponibili sia le tessere vergini che quelle programmate. Tutti i prezzi si intendono IVA compresa.

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

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



Liquid Crystal Display dalla teoria alla pratica

Impariamo a conoscere e utilizzare uno dei componenti elettronici più interessanti: il display LCD. Approfondiamo l'argomento presentando i display grafici, la demoboard con relativo firmware e un applicativo di test che funziona in ambiente Windows.

A CURA DI **Matteo Destro**

4

In questo articolo iniziamo ad occuparci dei display LCD grafici per i quali, esattamente come per gli alfanumerici, è stata realizzata una demoboard di valutazione in grado di supportare quattro differenti tipi di display, naturalmente uno alla volta. Partendo dai concetti teorici si proseguirà con l'analisi di situazioni pratiche, per poi giungere alla realizzazione di quattro semplici librerie con cui gestire i quattro display utilizzati durante il Corso.

Display supportati

La demo board è stata progettata per poter operare con i quattro display grafici della **"Displaytech Ltd"** elencati di seguito:

- **64128A**: 128x64 pixel monocromatico.
Controller grafico KS0108B
- **64128E**: 128x64 pixel monocromatico.

Controller grafico KS0713

- **64240B**: 240x64 pixel monocromatico.
Controller grafico T6963C
- **128240A**: 240x128 pixel monocromatico.
Controller grafico T6963C

Anche per i display grafici esiste un codice di fabbrica che permette di identificarli in modo univoco. A titolo informativo in Figura 1 è riportata la sintassi con cui viene creato il codice di prodotto (part number) dei display grafici della Displaytech. Ogni carattere utilizzato ha un suo preciso significato e le opzioni possibili sono svariate, quindi, come esercizio, analizzate il codice del seguente prodotto:

- **64128EFCBC-3LP**

La parte in blu identifica il nome del display ➤

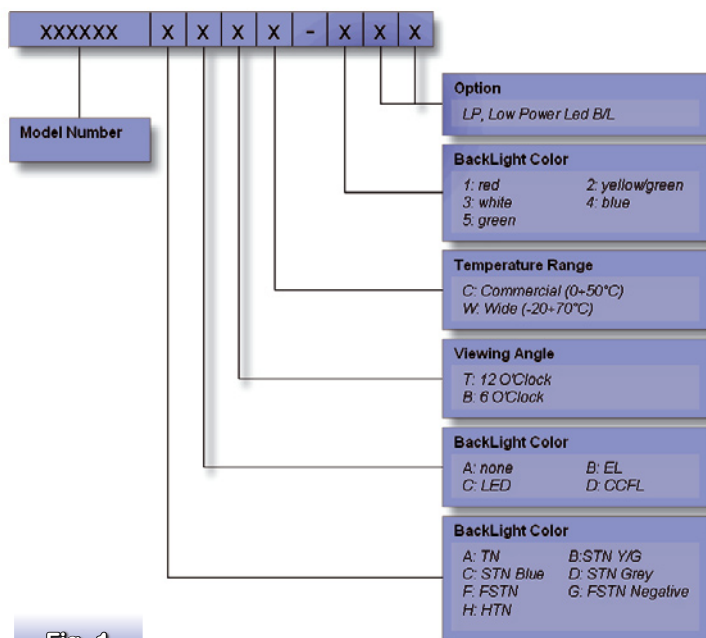


Fig. 1

grafico, la parte in rosso identifica le sue specifiche tecniche mentre la parte finale in blu chiaro fornisce alcune indicazioni relative alla retroilluminazione. A volte quest'ultima parte del codice è omessa. Con l'ausilio delle informazioni presenti in Figura 1 si ricava che:

- **“F”** Identifica la tecnologia con cui è stato realizzato il display, in questo caso **“Fast Super Twist Nematic”**, evoluzione della tecnologia STN;
- **“C”** Identifica la retroilluminazione utilizzata, in questo caso a LED;
- **“B”** Identifica l'angolo di visione;
- **“C”** Identifica la gamma di temperature in cui è consigliato utilizzare il display. Nel nostro caso il “range” è commerciale cioè compreso tra 0° C e +50° C;
- **“3”** Identifica il colore della retroilluminazione, nel nostro caso bianca;
- **“LP”** Identifica che la retroilluminazione è di tipo low power.

La demoboard può gestire un solo display grafico per volta ed è stata progettata con quattro connettori differenti, ognuno specifico per un display: ciò dipende dal fatto che i quattro modelli scelti non sono pin-to-pin compatibili dal momento che risoluzione e controller sono differenti.

Al contrario di quanto fatto per i display alfanumerici, in questo caso iniziamo subito commentandone lo schema elettrico, per poi entrare nel dettaglio dei singoli display grafici e del relativo controller. Seguirà quindi il codice sorgente delle routine di gestione. Sia il firmware che le re-

lative routine grafiche sono realizzate in linguaggio ANSI C, di più facile scrittura rispetto all'Assembler. Naturalmente è possibile riscrivere in linguaggio assembler per microcontrollori della serie 18Fxxxx il codice presentato, qualora qualcuno volesse provare.

Descrizione demoboard

Come si può osservare dallo schema a blocchi di Figura 2, i quattro display grafici hanno in comune un bus dati a 8 bit ed un set di linee di controllo. Sul bus dati transitano sia i comandi sia i dati da inviare ai controller, mentre le linee di controllo servono per gestire le operazioni di lettura/scrittura e la discriminazione dell'invio dei dati/comandi. Nel caso specifico del display

64128EFCBC-3LP è necessario utilizzare un traslatore di livello poiché va alimentato a + 3,3 Vdc, mentre il resto dell'elettronica funziona a + 5 Vdc. Il traslatore di livello è un MAX3002 del quale ci occuperemo dettagliatamente in seguito. La demoboard va collegata collegata alla porta seriale di un computer, che comunica con il microcontrollore per mezzo di un MAX202. Sulla scheda è prevista anche una memoria EEPROM 24LC1025 della Microchip, utilizzata per memorizzare le immagini e i caratteri da visualizzare sui display. Naturalmente la EEPROM può essere impiegata anche per scopi differenti dal nostro: le funzionalità che si possono aggiungere sono limitate solo dalla fantasia e dalla creatività, ma ovviamente il codice necessario va scritto appositamente. Abbiamo previsto cinque pulsanti per gestire alcune funzioni base controllate dal programma che gira su un PIC18F4620.

Iniziamo la spiegazione dello schema elettrico partendo dalle pin-out dei vari display grafici.

64128ABCBC

Esso necessita, oltre al bus dati a 8 bit, dei seguenti pin di controllo:

- **CS1:** Segnale di abilitazione per il controller 1 presente sul display: l'abilitazione si ottiene con un livello logico alto;
- **CS2:** Segnale di abilitazione per il controller 2 presente sul display: l'abilitazione si ottiene con un livello logico alto;
- **RSTB:** Segnale di reset. Se è presente un livello logico basso il controller è in RESET e quindi non né comandi né dati;

- **RS**: Segnale di discriminazione tra dati e comandi. Per i dati deve essere presente un livello logico alto, per i comandi un livello logico basso;

- **R/W**: Segnale di discriminazione per lettura/scrittura. Se presente livello logico alto si effettua una lettura, altrimenti si effettua una scrittura;

- **E**: Segnale di abilitazione per scrittura e lettura. Serve per obbligare il controller grafico a leggere il dato o il comando presente sul bus dati durante le operazioni di scrittura. La lettura viene eseguita quando il segnale di Enable passa da livello logico alto a livello logico basso, per cui è legata al fronte di discesa del segnale E. Se invece si sta eseguendo una operazione di lettura del controller grafico, il dato da leggere viene trasferito sul bus quando il segnale di Enable è a livello logico alto (lettura a livello);

- **Vout**: Tensione di uscita utilizzata per variare il contrasto. Il segnale Vout viene riportato in ingresso al pin Vo attraverso un trimmer multi-giro;

- **Vo**: Segnale di ingresso regolazione contrasto display.

Avrete già notato dalla descrizione dei segnali che questo display è equipaggiato con due distinti controller grafici (KS0108B della Samsung) ai quali è assegnata la gestione di metà display ognuno. Ciò significa che il display è suddiviso in due unità da 64x64 pixel che unite danno la dimensione effettiva di 128x64 pixel. I due controller

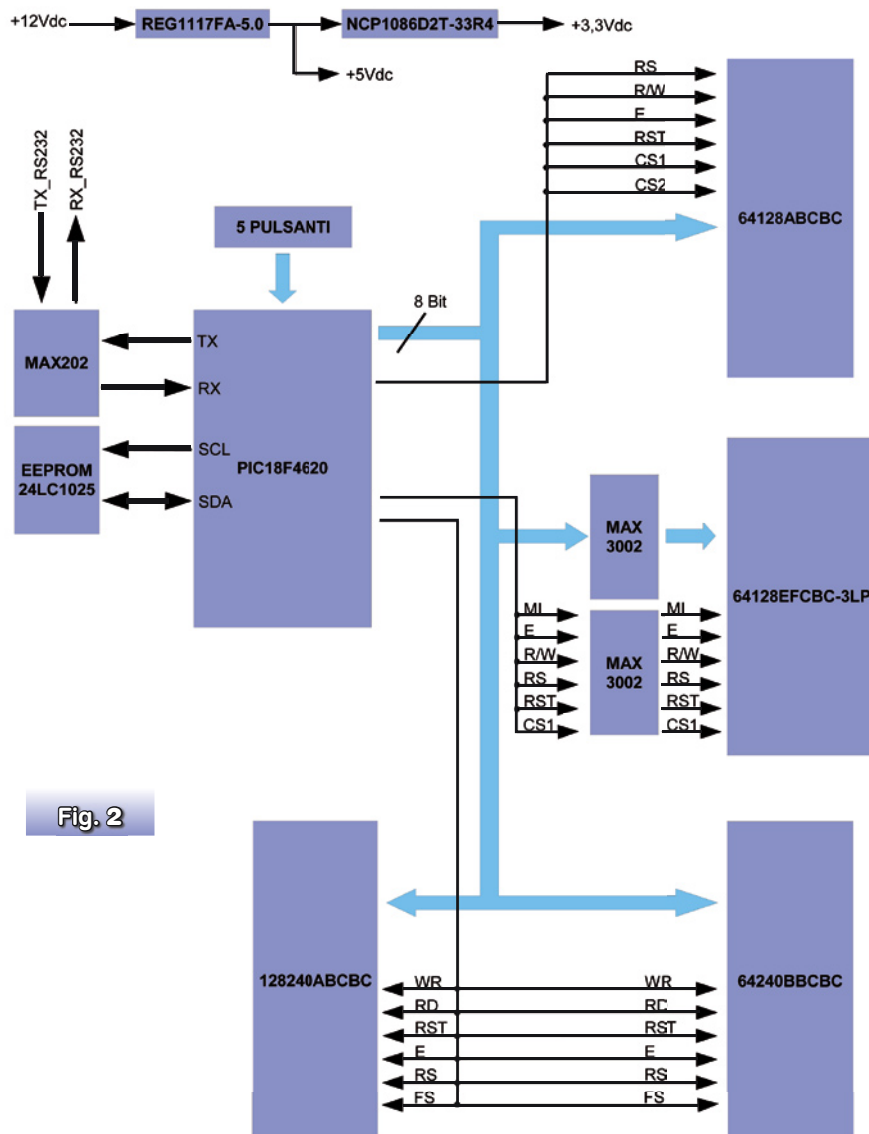


Fig. 2

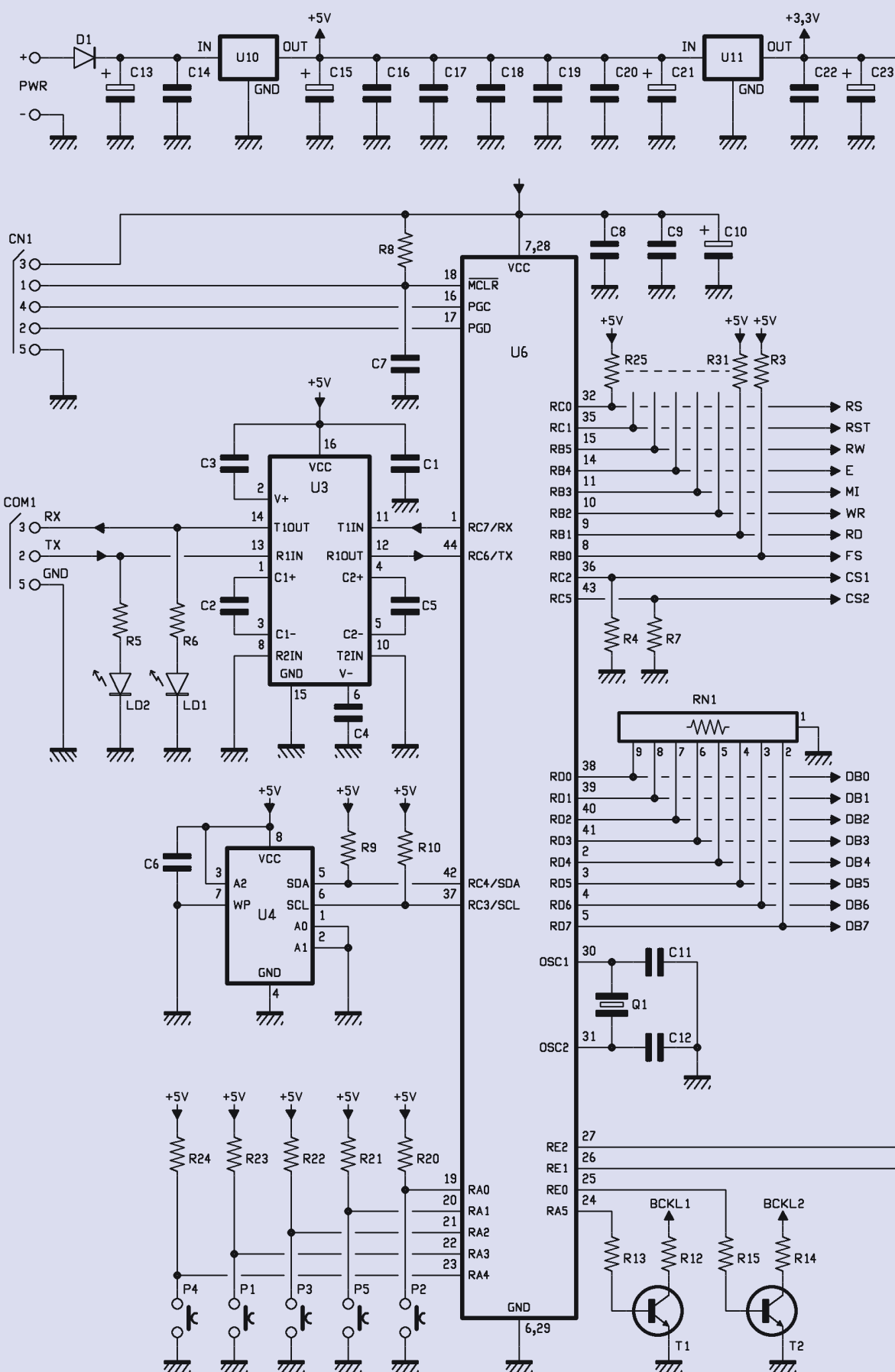
devono essere gestiti in modo indipendente.

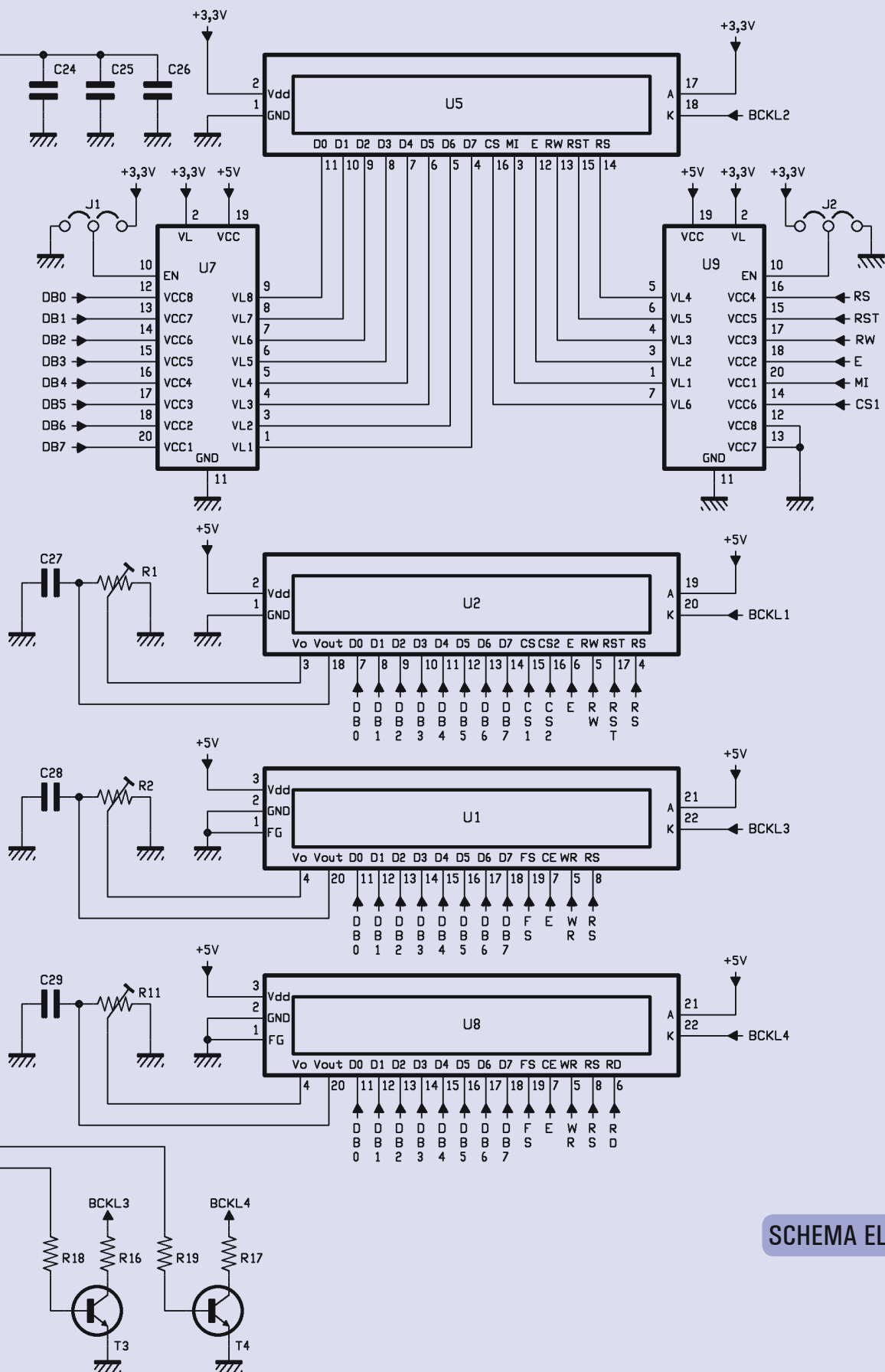
I dettagli tecnici verranno discussi in modo più approfondito insieme alle rispettive routine di controllo.

64128EFCBC-3LP

Anch'esso necessita, oltre al bus dati a 8 bit, di alcune linee di controllo:

- **CS1B**: Segnale di abilitazione per il controller presente sul display. Per abilitare il controller è necessario utilizzare un livello logico basso, esattamente il contrario del precedente display;
- **RST**: Segnale di reset. Se presente un livello logico basso il controller è in RESET e quindi non accetta né dati né comandi;
- **MI**: Segnale di selezione della modalità di interfacciamento con il microcontrollore. Se si trova a livello logico alto l'interfaccia è di tipo





SCHEMA ELETTRICO

6800, altrimenti l'interfaccia è di tipo 8080. Ricordiamo che l'interfacciamento 6800 è di tipo Motorola mentre l'interfacciamento 8080 è in stile Zilog Z80. Nel nostro caso utilizziamo l'interfaccia di tipo 6800;

- **RS:** Segnale di discriminazione tra dati e comandi. Per i dati deve essere presente un livello logico alto (basso per i comandi);
- **R/W:** Segnale di discriminazione per lettura/scrittura. Quando si trova a livello logico alto si effettua una lettura, altrimenti si effettua una scrittura;
- **E:** Segnale di abilitazione per la scrittura/lettura. Si comporta esattamente come descritto per il precedente display.

Questo display utilizza un unico controller KS0713 della Samsung in grado di gestire da solo tutti i pixel dell'intera matrice (128x64 pixel). Come si può osservare dallo schema elettrico, e come avevamo già accennato, dato che questo modulo LCD funziona a +3,3Vdc, è necessario utilizzare dei traslatori di livello per adattare i segnali di controllo provenienti dal PIC18F4620: in questo modo non si rischia di danneggiare l'elettronica del display. Il MAX3002 è un integrato che ben si presta a questa funzione, inoltre è bidirezionale e non necessita di alcuna linea di controllo aggiuntiva per discriminare la direzione in cui il segnale viene trasmesso. È però necessario alimentarlo con due tensioni distinte, una per i segnali a +5Vdc (VCC) e una per i segnali a +3,3Vdc (VL). Per abilitare l'integrato e attivare il canale di comunicazione tra il PIC e il controller del display grafico, il segnale di enable (EN) deve essere collegato a +3,3Vdc, altrimenti i pin di I/O VCC1 fino a I/O VCC 8 si trovano nella condizione di alta impedenza. Sulla scheda è stato previsto un jumper per sopperire a questa funzione: inserendolo verso massa l'integrato MAX3002 viene disabilitato. Per una corretta

traslazione del livello, l'alimentazione collegata al pin VL deve essere minore dell'alimentazione collegata al pin VCC.

Lo schema a blocchi del MAX3002 è riportato in Figura 3.

La bidirezionalità dell'integrato è garantita dal doppio circuito di traslazione. Uno si occupa di traslare il segnale da VL verso VCC mentre l'altro da VCC verso VL.

Il MAX3002 garantisce un transfer rate prossimo ai 20Mbps, quindi più che adatto per la nostra applicazione.

64240BBCBC

Anche il terzo display scelto per la nostra demoboard, oltre al solito bus dati a 8 bit, necessita di alcuni pin di controllo:

- **CE:** Segnale di abilitazione per il controller presente sul display. Il controller è abilitato con un livello logico basso;
- **RST:** Segnale di reset. Se forzato a livello logico 0, il controller è in RESET e quindi non accetta alcun comando o dato. Durante la fase di "power-on" è buona norma tenere il segnale di RESET a livello logico basso per almeno cinque impulsi di clock;
- **FS:** Segnale utilizzato per selezionare la dimensione del font utilizzato. Se è a livello logico alto si sceglie un font con dimensione di 6x8 pixel, altrimenti viene selezionato un font con dimensione da 8x8 pixel;
- **C/D:** Segnale per discriminare l'invio su bus da parte del microcontrollore di un dato oppure di un comando. Se si trova a livello logico alto il controller si aspetta l'invio di un comando; viceversa si aspetta l'invio di un dato
- **WR:** Segnale di scrittura. Se a livello logico basso significa che si sta eseguendo una scrittura. Di conseguenza il segnale di RD deve essere a livello logico alto
- **RD:** Segnale di lettura. Se a livello logico basso significa che si sta eseguendo una lettura. Di conseguenza il segnale di WR deve essere a livello logico alto
- **Vout:** Tensione di uscita utilizzata per la gestione del contrasto del display grafico. Il segnale Vout viene riportato in ingresso al pin Vo attraverso un trimmer multigiro.
- **Vo:** Segnale di ingresso per la regolazione del contrasto del display.

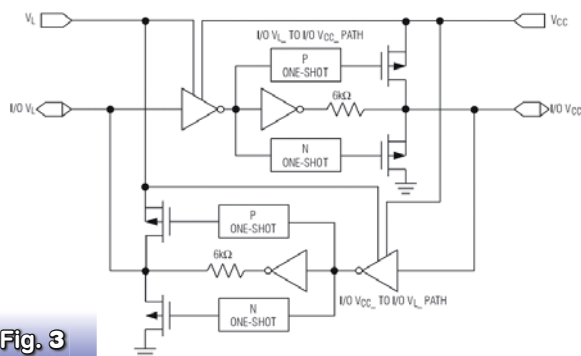


Fig. 3

Il controller utilizzato su questo display è il T6963C della Toshiba e, a seconda della confi-

gurazione, può gestire matrici grafiche fino a 640x256 pixel. In questo caso specifico il display ha una risoluzione di 64x240 pixel.

Inoltre, a differenza degli altri due controller, ha due possibili modalità di funzionamento:

- **Modalità grafica**
- **Modalità testo**

Naturalmente le due modalità possono essere combinate assieme ottenendo grafica miscelata a testo. Tutti i dettagli di funzionamento del controller verranno trattati in seguito.

128240ABCBC

Questo display ha una risoluzione di 128x240 pixel e utilizza lo stesso controller usato nel precedente modello, ovvero un T6963C.

Di conseguenza vengono utilizzati gli stessi segnali di controllo più il bus dati a 8 bit.

L'unica differenza sostanziale consiste nella possibilità di essere pilotato anche in modalità seriale sfruttando alcuni pin aggiuntivi che però non sono utilizzati nel nostro progetto in quanto si è preferito utilizzare la consueta modalità di trasferimento dati (parallela).

Sulla demoboard, oltre ai display grafici, trova

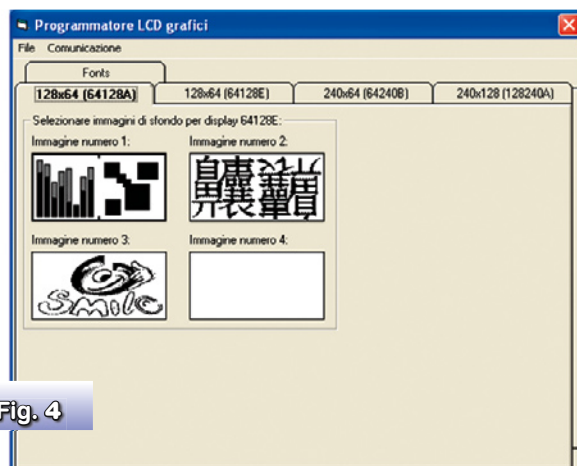


Fig. 4

posto una EEPROM da 1Mbit modello 24FC1025 di casa Microchip (a cui il micro accede tramite bus I2C) utilizzata per memorizzare le immagini e gli eventuali caratteri da visualizzare sui display grafici. La memoria è stata suddivisa in blocchi logici da 32 kbyte ciascuno. Nel primo blocco vengono memorizzate le immagini da visualizzare, nel successivo sono memorizzati i caratteri in formato grafico, più eventuali future informazioni. Gli ultimi due banchi (64 kbyte) rimangono a disposizione per sviluppi futuri o per uso di-

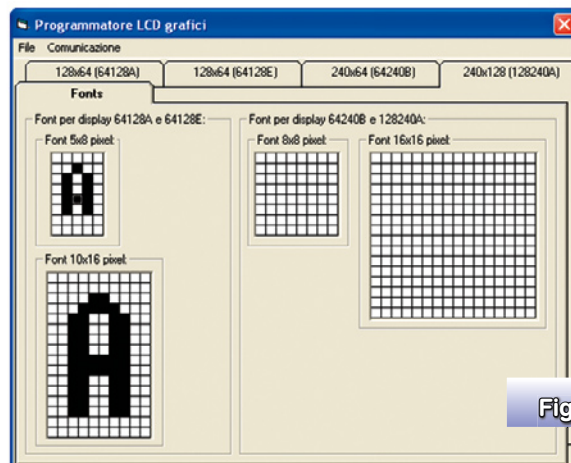


Fig. 5

dattico da parte del lettore. Per programmare la EEPROM si utilizza un programma per PC sviluppato per questo specifico scopo.

Come si può osservare dall'immagine di Figura 4, il programma è suddiviso in cartelle e ognuna di esse riguarda un particolare display, tranne l'ultima, utilizzata per creare una serie di font comuni a tutti gli LCD. Per ogni display si è pensato di dare la possibilità di selezionare e salvare fino a quattro immagini grafiche le quali verranno visualizzate in modo ciclico sul display grafico.

Per spostarsi da un'immagine alla successiva è necessario premere, sulla demoboard, il pulsante "Right" o "Left": i dati vengono prelevati dalla memoria e la relativa immagine viene visualizzata sullo schermo. Il firmware presente sulla scheda si occupa anche di gestire alcune funzioni aggiuntive del controller in uso al momento: tali funzioni possono essere richiamate utilizzando il pulsante "Enter" presente sulla scheda.

L'immagine di Figura 5 mostra invece la cartella dei font. Il limitato spazio di memoria consente di creare solamente due font per i display 64128A e 64128E e due font per i display 64240B e 128240A; risulta perciò evidente che in questo modo non è possibile creare una libreria di font da utilizzare nei propri progetti e pertanto possiamo affermare che questa gestione dei font ha solamente una valenza didattica.

Il font di dimensioni minori, 5x8 pixel, occupa in memoria 5 byte; se si volesse estendere la libreria di font sarebbe necessario prevedere un utilizzo di memoria pari a 5 byte x n (dove n rappresenta il numero di caratteri che si vuole realizzare). Ad esempio, per realizzare una libreria di 32 fonts sarebbe necessario uno spazio in memoria pari a 160 byte.

Vediamo ora come è stata organizzata la EEPROM. La Figura 6 evidenzia come viene utiliz-

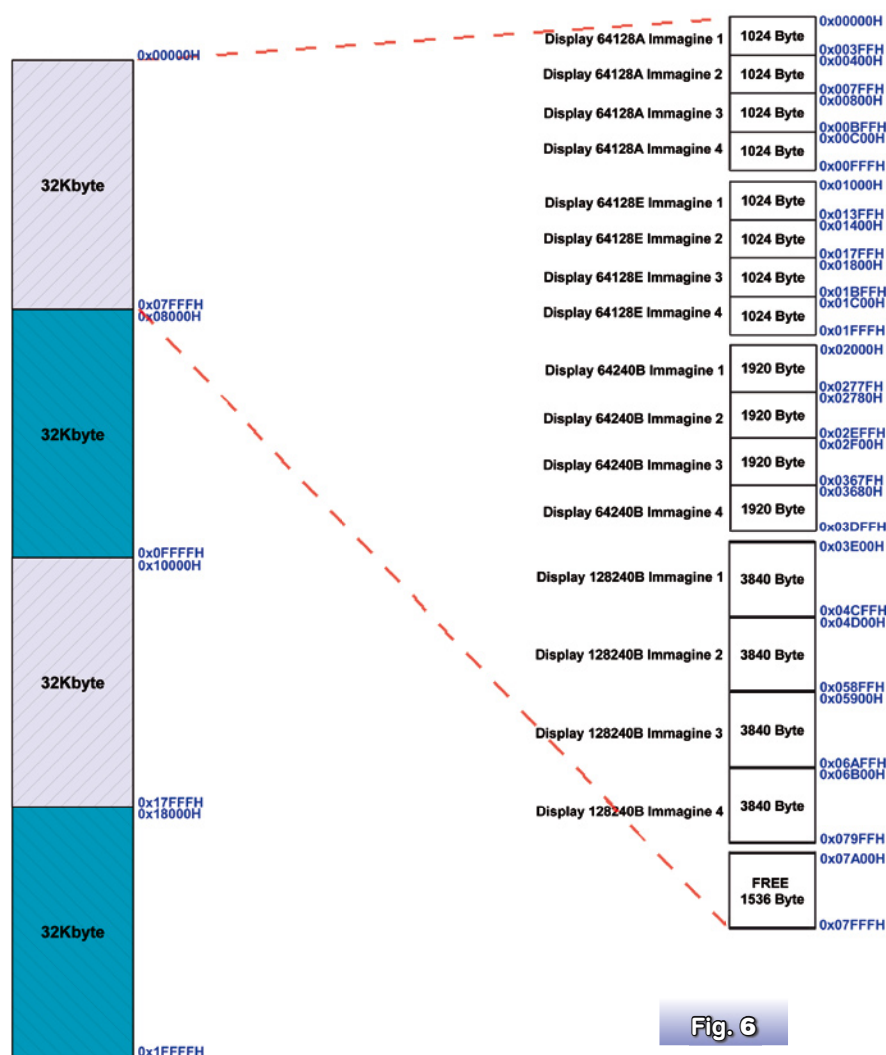


Fig. 6

zato lo spazio del primo banco di memoria da 32 kbyte. Notate che la quantità di memoria utilizzata cresce all'aumentare della risoluzione del display in uso. Ad esempio, i display da 128x64 byte richiedono 1024 byte per ogni immagine mentre i 128x240 ne richiedono 3840.

Quando si deve visualizzare una nuova immagine è necessario leggerne i byte dalla EEPROM e trasferirli nella memoria del controller grafico il quale, in modo automatico, genera i segnali di controllo per l'LCD rendendo così visibile l'immagine o il simbolo.

Ricordiamo che per indirizzare una memoria da 1 Mbit sono necessari 3 byte mentre il protocollo di trasmissione su bus I2C prevede solo 2 byte per l'indirizzamento. Ne consegue che la memoria deve essere suddivisa in due blocchi distinti da 65536 byte ciascuno. Per selezionare il primo o secondo blocco è necessario impostare il bit B0 nel byte di comando. Con B0 = 0 si indirizzano i primi 65536 byte con B0 = 1 i rimanenti byte.

zione di uno dei quattro display che la scheda è in grado di gestire. Ma di questo parleremo più avanti quando tratteremo il firmware memorizzato sul PIC.

Sulla scheda sono previsti anche i controlli per attivare la retroilluminazione dei quattro display: il circuito è molto semplice e utilizza solo un transistor come interruttore d'accensione.

Per ultimo parliamo dell'alimentazione, la quale è stata realizzata con dei regolatori di tensione lineari, in particolare è stato utilizzato un

Il control code è fisso mentre A0 e A1 vengono impostati a seconda di come sono stati collegati i pin A0 e A1 sulla scheda. Nel nostro caso entrambi sono a zero. Il bit R/W serve per discriminare le operazioni di lettura da quelle di scrittura.

I dati relativi a font ed immagini vengono trasferiti dal PC alla EEPROM passando attraverso il PIC18F4620, la cui interfaccia seriale è collegata ad un convertitore di livello RS232 (in questo caso si tratta di un MAX202E). Il transfer rate predefinito è 115200 Baud e permette di programmare la EEPROM molto velocemente. Per la trasmissione seriale vengono utilizzati solo i segnali TX e RX.

La demoboard integra anche 5 pulsanti, necessari per impartire semplici comandi al PIC18F4620. Ad esempio allo start up la pressione di una combinazione di essi permetterà di eseguire l'inizializzazione

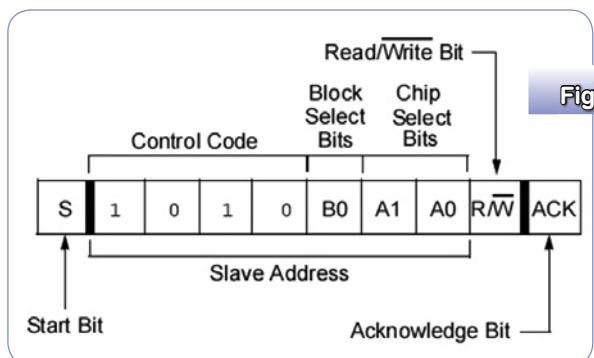
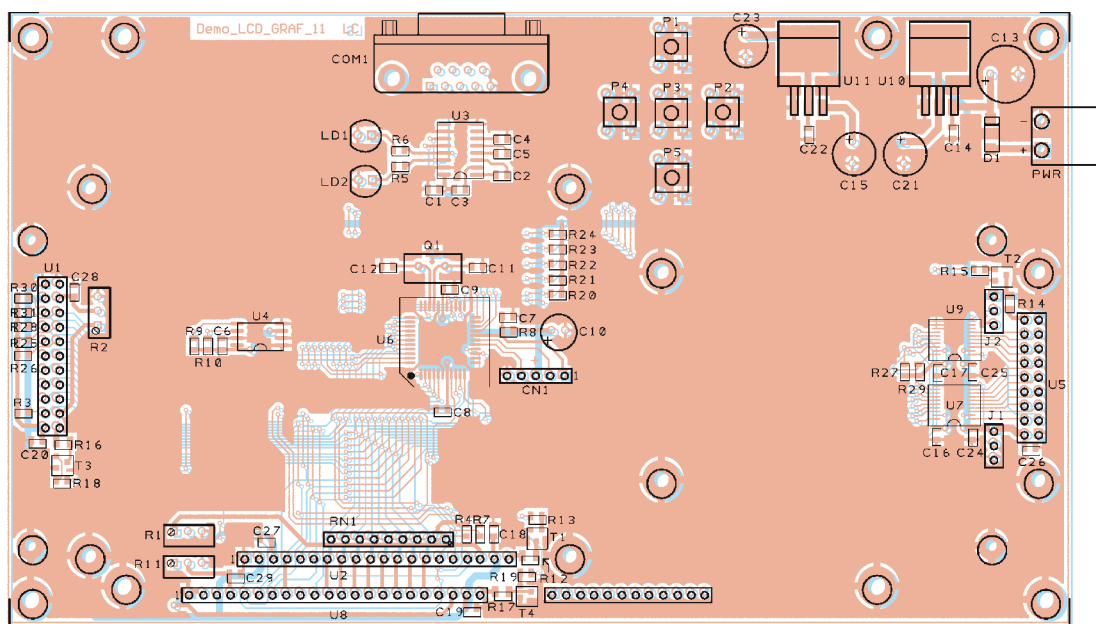
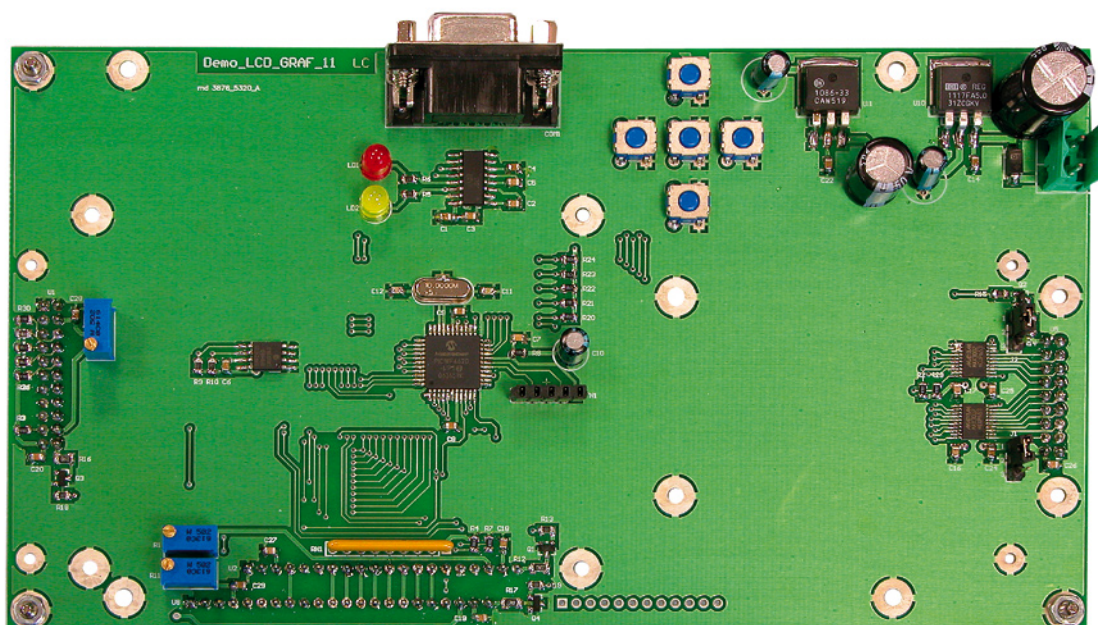


Fig. 7



Disegni e foto della demoboard in dimensioni ridotte (75%). I master in scala reale sono scaricabili dal sito della rivista (www.elettronica.in.it). Tutti i componenti utilizzati sono a montaggio superficiale.

ELENCO COMPONENTI:

R1, R2: trimmer multigiro 4,7 kohm
 R3, R4: 4,7 kohm SMD
 R5, R6: 1,2 kohm SMD
 R7: 4,7 kohm SMD
 R8: 22 kohm SMD
 R9, R10: 1,2 kohm SMD
 R11: trimmer multigiro 4,7 kohm
 R12, R14: 22 ohm SMD
 R13, R15: 1,2 kohm SMD
 R16, R17: 22 ohm SMD
 R18, R19: 1,2 kohm SMD
 R20+R31: 4,7 kohm SMD
 RN1: rete resistiva 8x4,7 kohm +

comune
 C1+C6: 100 nF multistrato SMD
 C7: 220 pF ceramico SMD
 C8, C9: 100 nF multistrato SMD
 C10: 10 µF 25 VL elettrolitico
 C11, C12: 22 pF ceramico SMD
 C13: 470 µF 50 VL elettrolitico
 C14: 100 nF multistrato SMD
 C15: 100 µF 50 VL elettrolitico
 C16+C20: 100 nF multistrato SMD
 C21: 10 µF 25 VL elettrolitico
 C22: 100 nF multistrato SMD
 C23: 10 µF 25 VL elettrolitico

C24+C29: 100 nF multistrato SMD
 D1: GF1M (DO214)
 LD1: led 5 mm rosso
 LD2: led 5 mm giallo
 U1: Display LCD 64240B
 U2: Display LCD 64128A
 U3: MAX202E
 U4: 24FC1025-I/SM
 U5: Display LCD 64128E
 U6: PIC18F4620-IPT (MF684)
 U7: MAX3002
 U8: Display LCD 128240A
 U9: MAX3002

U10: REG1117FA-5.0
 U11: NCP1086D2T-33R4
 Q1: Quarzo 10 MHz
 P1+P5: microswitch
 T1+T4: BC846B
 COM1: DB9 femmina da cs

Varie:
 - Morsetteria 2 poli componibile
 - Strip maschio 5 poli
 - Strip maschio 3 poli (2 pz.)
 - Jumper (2 pz.)
 - Circuito stampato

REG1117FA (Burr-Brown) per generare i +5 Vdc e un NCP1086D2T (ON Semiconductor) per generare i +3,3 Vdc necessari al display 64128E e ai due MAX3002. I due regolatori sono in SMD, come la maggior parte degli altri componenti, nello specifico in package DDPak. Solo i connettori e i condensatori elettrolitici sono di tipo tradizionale.

Per l'assemblaggio della scheda consigliamo di iniziare dalle resistenze e dai condensatori SMD, passando successivamente ai due MAX3002: prestate molta attenzione in quanto i package di questi due integrati hanno un passo veramente ridotto ed è perciò necessario utilizzare un saldatore a punta super fine, stagno con diametro inferiore a 0,5 mm, del flussante (meglio se in gel) e una lente di ingrandimento. Così facendo realizzerete delle ottime saldature. Conclusa la saldatura dei MAX3002 potrete passare al PIC18F4620 che richiede anch'esso moltissima attenzione. Successivamente dovrete saldare i rimanenti integrati SMD e, per ultimi, i due regolatori di tensione, il quarzo, i pulsanti e i rimanenti componenti ad inserzione through hole.

Gestione di un immagine bitmap

I display utilizzati con la demoboard sono monocromatici quindi il software di programmazione della EEPROM prevede l'utilizzo di immagini con profondità di colore a 1 bit (Bianco e Nero), in particolare delle bitmap (estensione .BMP) realizzate con le dimensioni adeguate per i display utilizzati.

Durante l'apertura di un file bitmap il programma lo analizza verificando che sia compatibile con il display a cui verrà associato (per caricare un file

.BMP è necessario cliccare con il tasto destro del mouse sul riquadro a cui si vuole associare l'immagine richiamando così la finestra di dialogo di Windows, come mostrato nella Figura 8).

A questo punto non rimane che selezionare l'immagine desiderata e fare clic sul pulsante "Apri". Se l'immagine selezionata è conforme al tipo di display e alla profondità di colore vedremo apparire la stessa nel riquadro bianco ricevendo così conferma della sua validità.

Ad esempio durante la selezione delle immagini per il display 64128ABCBC dovranno essere soddisfatte le seguenti condizioni affinché l'immagine sia considerata compatibile:

- Larghezza immagine 128 pixel
- Altezza immagine 64 pixel
- Profondità colore 1 bit

Se solo una di queste condizioni non è soddisfatta, il programma rifiuta l'immagine avvisando l'utente dell'incompatibilità riscontrata.

Per capire come il programma sia in grado di discriminare le immagini è fondamentale esaminare un file bitmap con un editor di testo in modo da visualizzare ogni singolo byte in formato esadecimale (Vedere più avanti nel testo).

Possiamo quindi suddividere il file in due parti distinte; la prima contiene gli Headers che altro non sono che delle informazioni riguardanti il file stesso e l'immagine in esso memorizzata.

La seconda parte contiene l'immagine. Gli Headers si trovano in testa al file e solitamente occupano i primi 62 byte; essi sono suddivisi in due gruppi il primo dei quali identifica Headers

del file .BMP (i primi 14 byte) mentre il secondo identifica gli Headers dell'immagine memorizzata nel file .BMP. La Tabella 1 descrive tutti gli Headers presenti in un file bitmap.

Oltre a quelli elencati nella tabella ci sono altri 4 byte i quali identificano la tabella dei colori; blu, verde e rosso mentre l'ultimo byte non è usato.

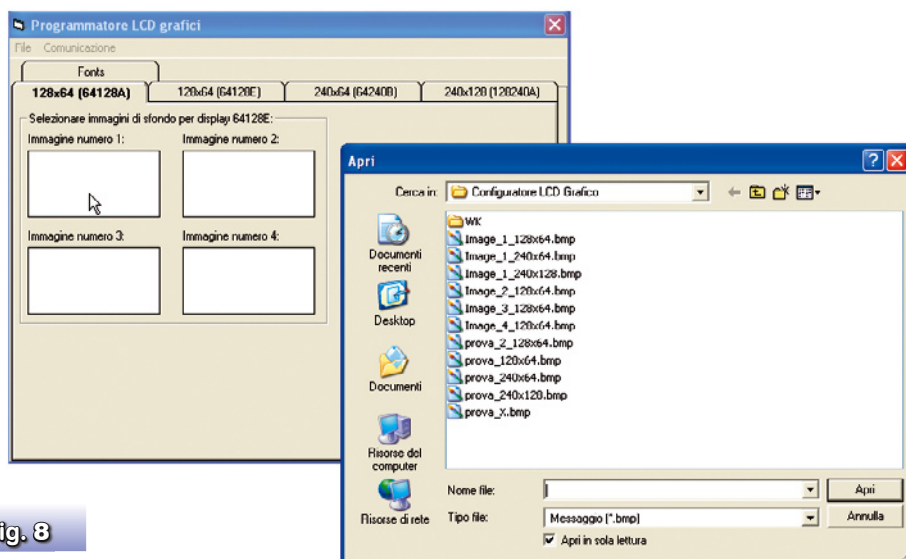


Fig. 8

Nelle immagini Bitmap di nostro interesse gli ultimi 4 byte sono sempre 0xFF, 0xFF, 0xFF, 0x00.

Adesso possiamo illustrare come viene memorizzata l'immagine nel file bitmap e per fare ciò utilizzeremo l'esempio di Figura 9a.

All'interno del file l'immagine è salvata specularmente rispetto all'asse X (Figura 9b), inoltre i pixel bianchi vengono salvati come "1" logico mentre i pixel neri vengono salvati come "0" logico, esattamente il contrario di quanto avviene sui nostri display grafici dove a "1" logico corrisponde nero e a "0" logico corrisponde bianco. Ogni byte salvato contiene i dati di 8 pixel dell'immagine; se l'immagine fosse stata a colori, ad esempio 256, era necessario un byte per ogni singolo pixel dell'immagine salvata.

Un'altra caratteristica del file bitmap è che i byte vengono salvati a gruppi di 4, questo vuol dire che un'immagine larga 128 pixel è formata da quattro gruppi di 4 byte per riga. Se invece l'immagine fosse stata larga 144 pixel sarebbero stati necessari cinque gruppi da 4 byte di cui gli ultimi due byte del quinto gruppo posti a 0x00 perché non usati ($144 / 8 = 18$ byte, quindi rispettando la regola dei 4 byte abbiamo 5 gruppi come detto sopra).

Le routine di estrapolazione dell'immagine dal file bitmap, realizzate per questo progetto, leggono il file dal basso verso l'alto, partendo dall'ultimo byte presente nel file, memorizzando i dati in una matrice la quale verrà poi resa speculare rispetto all'asse Y, ottenendo l'immagine di partenza.

La matrice, in questo caso, è composta da 64 righe per 16 colonne (matrice valida per l'esempio in corso; è scontato che se la dimensione in pixel del display fosse differente avremmo anche una matrice differente).

Essa viene poi ulteriormente elaborata per rendere compatibile il bitmap con la memoria grafica del display collegato al PIC18F4620.

Il tipo di elaborazione dipende da come viene gestita la memoria grafica: nel-

Tabella 1

Nome Campo	Dim. in byte	Descrizione
bftype	2 Byte	Identifica i due caratteri "B" e "M". Questi due caratteri ASCII danno la certezza che il file che si sta esaminando è un file bitmap
bfsSize	4 Byte	Identifica la dimensione del file .BMP in byte, la dimensione è data dagli Headers più l'immagine memorizzata nel file. Il byte meno significativo è il primo byte a sinistra dopo il campo "bftype"
bfReserved1	2 Byte	Non usato deve essere a 0x0000
bfReserved2	2 Byte	Non usato deve essere a 0x0000
bfOffBits	4 Byte	Identifica il punto di inizio dell'immagine all'interno del file, tecnicamente è uno spaziamento (offset) rispetto al primo byte del file. Solitamente per le immagini in Bianco e Nero è 0x3E (62 Byte). Il byte meno significativo è il primo a sinistra dopo il campo "bfReserved2"
biSize	4 Byte	Identifica la dimensione degli Headers. Tale parametro deve essere almeno 40 o superiore. Il byte meno significativo è il primo byte a sinistra dopo il campo "bfOffBits"
biWidth	4 Byte	Larghezza immagine in pixel. Il byte meno significativo è il primo a sinistra dopo il campo "biSize"
biHeight	4 Byte	Altezza immagine in pixel. Il byte meno significativo è il primo a sinistra dopo il campo "biWidth"
biPlanes	2 Byte	Deve essere impostato a 0x0001. Il byte meno significativo è il primo byte a sinistra dopo il campo "biHeight"
biBitCount	2 Byte	Identifica la profondità colore dell'immagine. I valori possibili sono: 1, 2, 4, 8, 16, 24, o 32. Nel nostro caso la profondità colore è impostata a 1 cioè Bianco e Nero. Il byte meno significativo è il primo byte a sinistra dopo il campo "biPlanes"
biCompression	4 Byte	Definisce il tipo di compressione dell'immagine. Nel caso di immagini non compresse il valore memorizzato è 0x0000. Il byte meno significativo è il primo byte a sinistra dopo il campo "biBitCount"
biSizelImage	4 Byte	Definisce la dimensione dell'immagine contenuta nel file .BMP. Ad esempio, se l'immagine fosse da 128x64 pixel, la dimensione in byte della stessa sarebbe 1024 byte. Mentre la dimensione totale del file .BMP sarebbe 1086 byte (1024 Byte + 62 Byte di "Headers"). Il byte meno significativo è il primo byte a sinistra dopo il campo "biCompression"
biXPelsPerMeter	4 Byte	Risoluzione in pixel per metro coordinata X. Il byte meno significativo è il primo byte a sinistra dopo il campo "biSizelImage"
biYPelsPerMeter	4 Byte	Risoluzione in pixel per metro coordinata Y. Il byte meno significativo è il primo byte a sinistra dopo il campo "biXPelsPerMeter"
biClrUsed	4 Byte	Definisce il numero della mappa dei colori che sono attualmente usati. Il byte meno significativo è il primo byte a sinistra dopo il campo "biYPelsPerMeter"
biClrImportant	4 Byte	Definisce il numero di colori significativi. Il byte meno significativo è il primo byte a sinistra dopo il campo "biClrUsed"

Fig. 9a



Fig. 9b



le prossime puntate ci occuperemo in maniera dettagliata di questo argomento analizzando il firmware relativo ai programmi di gestione di ogni singolo display.

A conclusione di questa puntata pubblichiamo (vedi Figura 10) una parte del contenuto del file bitmap in formato esadecimale.

In giallo vengono evidenziati i primi 62 byte i quali identificano gli Headers descritti nella tabella, mentre la prima riga e l'ultima riga dell'immagine sono identificate dal colore azzurro.

Nel file sono presenti 64 righe poiché l'immagine presenta un formato di 128x64 pixel.

Ogni riga è composta da quattro gruppi da 4 byte ciascuno. Nella prossima puntata ci occuperemo singolarmente dei vari controllori grafici utilizzati nei quattro display utilizzati nella demoboard. Per ognuno di essi analizzeremo le funzioni disponibili a cui fanno riferimento le nostre routine di gestione.

Appuntamento, dunque, alla prossima puntata del Corso.

