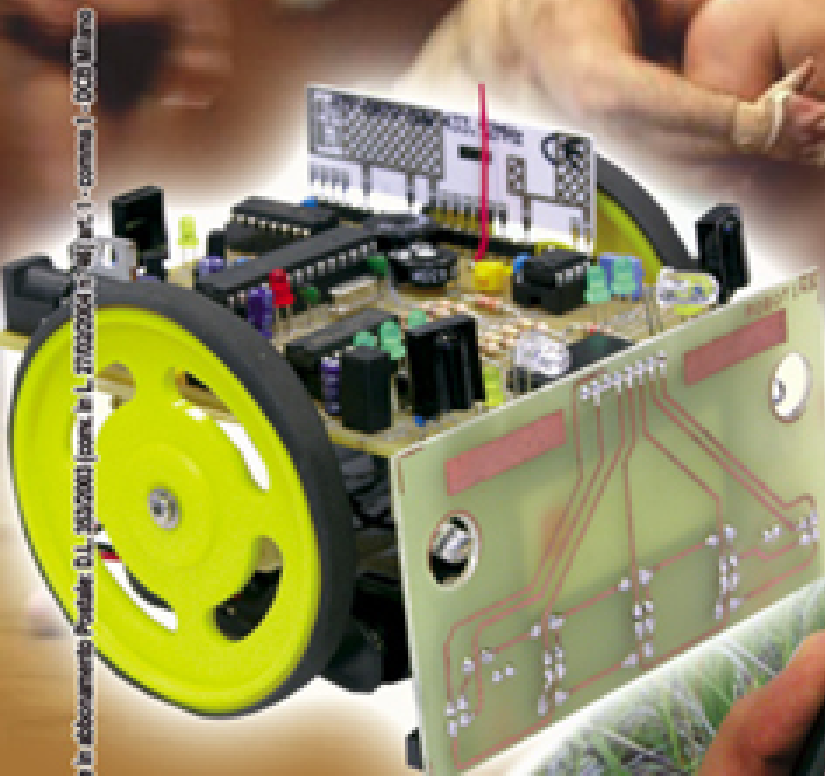
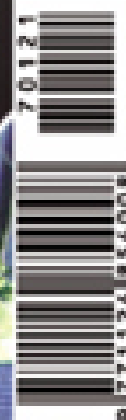


Elettronica In

Mensile di elettronica applicata, attualità scientifica, novità tecnologiche.

121

Robot In: il piccolo gigante



Localizzatore con bussola



Telecontrollo GSM per sistemi Velbus

Trasmittitore 16 canali programmabile

Come aggiornare il firmware del Mambo

Demoboard GPS/GSM: adattore per GPS

Controllo con joystick per braccio robotizzato

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

Trasmittitore 16 CH programmabile

di *Alessandro Sottocornola*

Può comandare fino a 16 ricevitori operanti in banda UHF a 433,92 MHz e codificati a base MM53200/UM3750. I comandi possono essere inviati sia mediante i pulsanti di cui dispone, sia tramite un computer, collegato via USB, dove è installato un applicativo specifico. L'applicativo, sviluppato in ambiente Windows, emula una tastiera virtuale e permette anche di configurare il dispositivo.

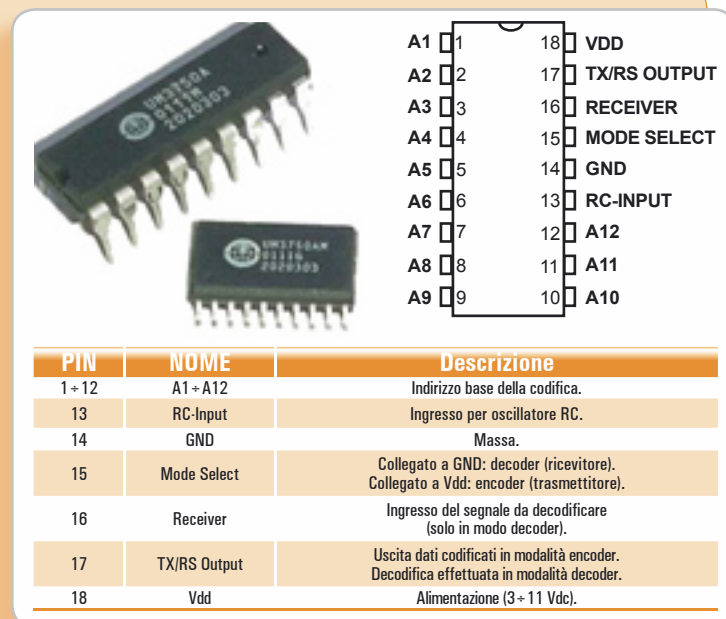
Sarà per praticità, sarà per pigrizia, ormai da tempo i costruttori di apparati elettronici e non solo ci hanno abituati a controllare praticamente qualsiasi cosa di casa senza fare un passo o alzarsi dal divano sul quale, bicchiere in una mano e telecomando nell'altra, ci si gode la propria musica o il proprio film preferito. È così che oramai, fra centraline radiocomandate per il controllo delle luci, portoni con elettroserratura, cancelli, tapparelle e tende motorizzati, siamo invasi da tanti telecomandi che teniamo in tasca, insieme al portachiavi, o sul piano di un mobile: dato il possibile impiego in

luoghi così diversi, è inevitabile che prima o poi qualche telecomando venga perso di vista: tante comodità che, alla fine, diventano scomode da gestire... Per non doversi preoccupare di cercare ogni volta il telecomando giusto per ogni apparato, bisognerebbe disporre di un unico dispositivo in grado di comandare tutto quello che viene gestito dai singoli trasmettitori; ovviamente ciò è possibile solamente se tutti i sistemi in questione operano con la stessa modalità (via infrarossi o via radio) sulla medesima frequenza e condividono la stessa codifica. Fortunatamente gran parte dei comandi a distanza usati

L'integrato MM53200/UM3750 è un circuito integrato che può funzionare sia da encoder che da decoder, progettato appositamente per realizzare telecomandi. Permette una codifica di 4.096 combinazioni (12 bit) che si imposta agendo sui piedini 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, ovvero assegnando ad essi uno stato logico che può essere 0 o 1. Tutti i dodici pin di codifica sono provvisti internamente di un resistore di pull-up, quindi per forzarli a livello alto è sufficiente lasciarli scollegati; per ottenere lo zero, invece, occorre collegarli a massa (piedino 14, GND) mediante un interruttore o un transistor. Ciò che definisce la modalità di funzionamento del chip è la condizione assegnata al pin 15 (MODE) che deve trovarsi ad "1 logico" (potenziale circa uguale a quello di alimentazione applicato al pin 16) se l'integrato deve operare come codifica (encoder) o a "0 logico" nel caso opposto (decoder). Affinché un chip impostato come decoder riconosca la trasmissione di uno operante come encoder, devono essere soddisfatte due condizioni:

- 1) entrambi i dispositivi devono essere programmati con la stessa combinazione di codifica (Pin A1÷A12),
- 2) la frequenza di oscillazione deve essere la medesima in entrambi i dispositivi, considerando le tolleranze accettate dal chip.

L'oscillatore stabilisce, nell'encoder, la temporizzazione delle stringhe di dati trasmesse; nel decoder, invece, genera il sincronismo per la lettura delle stringhe di dati ricevute. Per questo motivo, è indispensabile avere frequenze di lavoro uguali o il più possibile simili. Quanto alla codifica, va detto che è possibile sfruttare parte dei bit A1÷A12 per realizzare sistemi pluricanale; certo, ciò

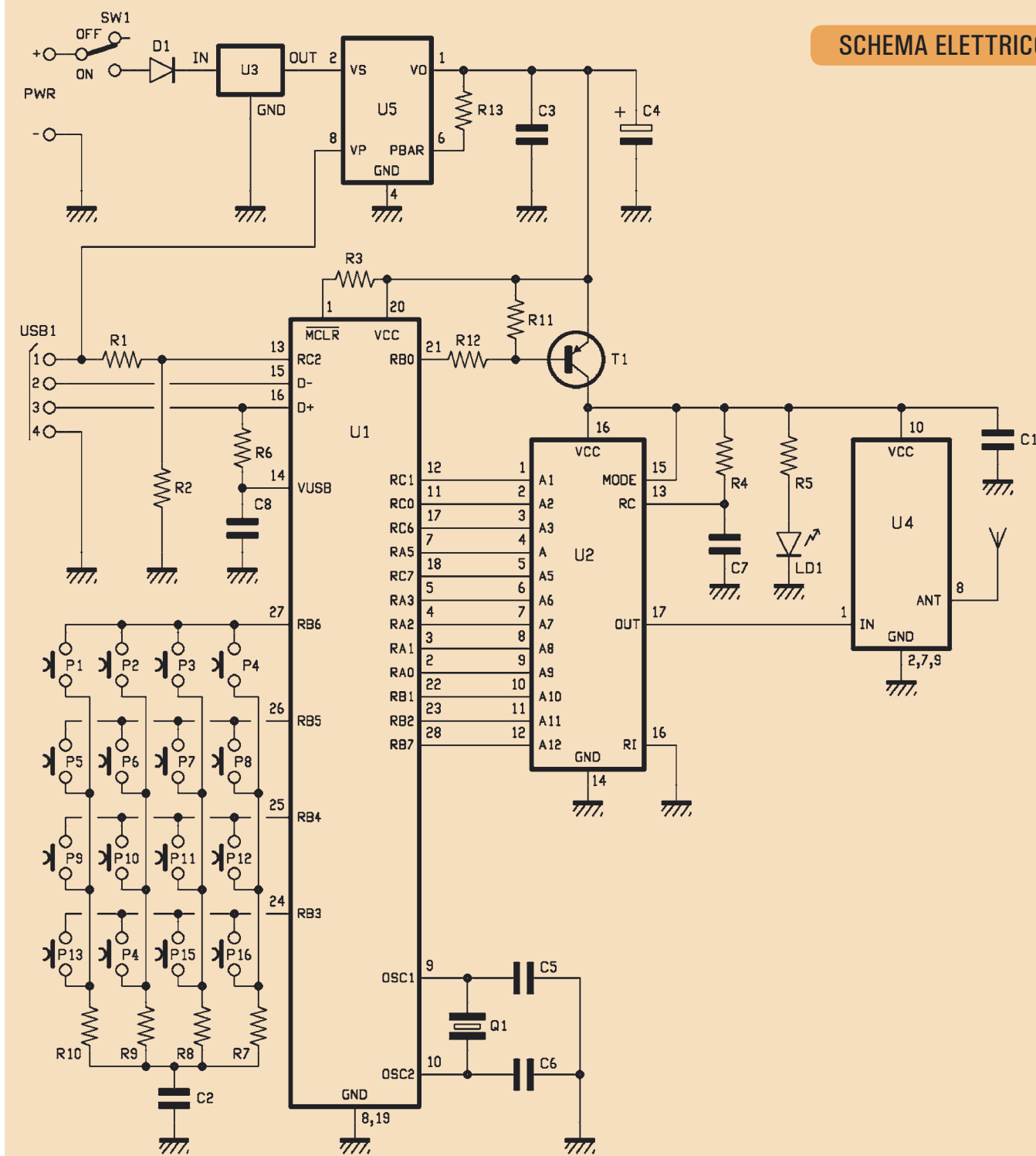


viene fatto a discapito della sicurezza, poiché il numero delle possibili combinazioni si abbassa di conseguenza. Per esempio, mantenendo inalterati i primi dieci bit (A1÷A10) e variando ad ogni trasmissione la combinazione degli ultimi due (A11, A12), si possono ottenere quattro diversi codici di comando. Se questi quattro codici di comando vengono inviati ad altrettanti integrati funzionanti da decoder, ognuno di essi può essere configurato per riconoscere una delle quattro possibili combinazioni. Ne risulterà un sistema a quattro canali, ma con 1024 possibilità di indirizzo.

in casa è via radio e opera intorno ai 434 MHz, banda in cui è ammesso l'uso libero di unità trasmettenti di potenza relativamente bassa (alcuni milliwatt). Ma non solo: tipicamente, la codifica usata è basata sull'encoder/decoder MM53200 National Semiconductors e sui suoi equivalenti UM3750 ed UM86409. Stando così le cose, abbiamo pensato di realizzare e proporvi in queste pagine un trasmettitore a 433,92 MHz codificato con l'integrato UM3750, capace di indirizzare fino a 16 canali, che potrete utilizzare per sostituire altrettanti TX singoli. Non è la prima volta che proponiamo un radiocomando a più canali, ma quello qui descritto è particolare perché, oltre a poter inviare i comandi autonomamente, può essere usato in abbinamento ad un Personal Computer come unità

di trasmissione pura e semplice, indirizzata dai comandi che possiamo dare con il mouse agendo su tasti virtuali del programma da noi preparato allo scopo. Ma non finisce qui: un'altra particolarità del trasmettitore è che possiamo programmarlo a piacimento, decidendo, per ogni pulsante, quale combinazione dei bit (leggasi "indirizzo") assegnare. Per comprendere quest'ultima funzionalità bisogna conoscere un po' meglio l'UM3750 (o MM53200, che poi è lo stesso) e sapere che la sua codifica è a 12 bit binari (che possono valere 1 o 0) i quali consentono 4.096 combinazioni; lo stato logico di ciascun bit si assegna collegando a massa (0) o lasciando scollegati (1) i piedini 1÷12, corrispondenti rispettivamente ai bit di indirizzo A1÷A12. Solitamente, nei trasmettitori e ri-

cevitore multicanale, il numero di canali si decide modificando la condizione logica degli ultimi bit e usando i piedini restanti per definire la codifica comune: ad esempio, in un sistema a due canali i primi 11 bit compongono la codifica (2.048 combinazioni) e il dodicesimo vale 0 per il primo canale e 1 per il secondo o viceversa. Ipotizzando di dover realizzare un sistema tradizionale a 16 canali, visto che 16 è pari a 2 elevato alla quarta, occorrerebbe variare gli ultimi quattro bit, lasciando i primi otto per l'impostazione della codifica. Invece, nel nostro caso, questo vincolo non c'è: programmando opportunamente il trasmettitore, infatti, si può fare in modo di assegnare ad un determinato pulsante un'impostazione dei 12 bit completamente differente da



quella degli altri, così da poter comandare tramite ciascun pulsante un qualsiasi canale utilizzando una qualsiasi delle 4096 combinazioni. In altre parole, non esiste il vincolo della separazione tra bit di indirizzo e di codifica, perché ogni singolo bit può essere programmato a piacere per operare in entrambi i modi.

Schema elettrico

Chiarito questo concetto, possiamo vedere di cosa si tratta, iniziando con l'analizzare lo schema elettrico dell'unità: l'intero trasmettitore è gestito da un microcontrollore PIC18F2550, da noi scelto perché, oltre ai tanti I/O, implementa anche un'interfaccia USB completa (compatibile USB

2.0) tramite la quale può dialogare con un computer.

Dopo il power-on-reset, il micro inizializza le proprie linee di I/O RA0, RA1, RA2, RA3, RA5, RB1, RB2, RB7, RC0, RC1, RC6 ed RC7 come uscite destinate all'impostazione della codifica dell'encoder U2, integrato che viene sempre mantenuto spento e ➤

Per impostare tutte le funzioni del trasmettitore ci avvaliamo di un software operante in ambiente Windows 2000/XP (o più recente) la cui finestra di dialogo principale, che appare all'avvio, è illustrata nella Figura A1; in essa possiamo distinguere quattro sezioni: quella in alto riguarda la gestione dei dispositivi (dei trasmettitori) collegati, la seconda contiene i comandi di impostazione dei codici per i singoli canali, la terza ripete la tastiera formata da P1+P16 e l'ultima contiene i comandi di gestione della memoria del microcontrollore che gestisce il trasmettitore. Vediamole una ad una, iniziando da quella in alto: il pulsante *Dispositivi* dà accesso alla finestra di dialogo illustrata dalla Figura A2 nella quale è possibile selezionare il trasmettitore con cui dialogare, qualora ne fosse collegato più di uno al computer. Il pulsante *Refresh* permette di aggiornare l'elenco dei trasmettitori mentre il pulsante *Ok* conferma la scelta: notate che, dopo aver selezionato il dispositivo, la finestra principale diventa quella di Figura A3, in cui le sezioni non attive (tastiera, configurazione e gestione memorie) ora sono abilitate. Il secondo pulsante (*Connetti/Disconnetti*) permette di chiudere una sessione di comunicazione aperta con un trasmettitore. In quest'ultimo caso, notate che per riconnettere il trasmettitore collegato all'USB bisogna fare nuovamente clic su *Dispositivi* e selezionare di nuovo il TX. Infine, il pulsante *Dettagli* permette di conoscere i dettagli della connessione.

La seconda sezione della finestra di dialogo contiene una serie di dip-switch virtuali, che servono per programmare indirizzo e codifica di ogni canale (pulsante); le operazioni da compiere sono intuitive: si seleziona il canale da configurare tramite il menu a tendina, quindi si fa clic su ogni levetta del dip switch virtuale per definire la codifica, ed infine si memorizza la combinazione scelta facendo clic sul pulsante *Memorizza*. Con *Test TX*, infine, si può subito verificare l'effetto del comando.

Quanto alla terza sezione, sarebbe quasi superfluo descriverla: appare piuttosto evidente che essa emula la tastiera quindi, ogni volta che si fa clic su uno dei pulsanti virtuali, il trasmettitore invia il codice associato al pulsante premuto.

E veniamo, infine, alla sezione inferiore, contenente due pulsanti: *Leggi Memoria Canali* permette di scaricare dal micro del trasmettitore i codici associati ai sedici canali; per vedere lo stato di ognuno di essi basta scegliere il numero desiderato nel menu a tendina della seconda sezione, dopodiché i dip virtuali cambieranno di stato automaticamente mostrando l'impostazione corrispondente. Il pulsante *Reset Memoria Canali*, infine, permette di cancellare tutti i codici memorizzati.

Fig. A1



Fig. A3

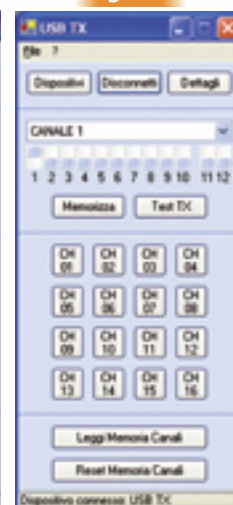
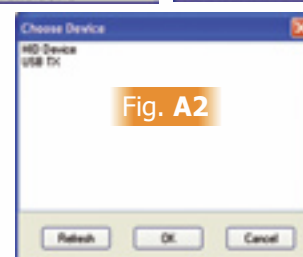


Fig. A2



che il PIC alimenta, mediante l'RB0 (output) e il transistor T1 (usato come interruttore statico) solo quando deve trasmettere; le linee RB3, RB4, RB5 ed RB6 vengono inizializzate come bidirezionali e servono alla lettura della tastiera, operazione per la quale il software ricorre a un particolare artificio.

Oltre alle predette linee di I/O, il programma che gira nel microcontrollore inizializza la porta USB, la cui connessione al computer viene rilevata mediante RC2 (I/O impostato come ingresso e usato per "sentire" la presenza dei 5 volt sul conduttore 1 del cavo USB). Sempre a proposito di alimentazione, l'intero circuito richiede una tensione di valore compreso tra 8 e 15 volt, dalla quale il regolatore

U3 (posto a valle del diodo di protezione dall'inversione di polarità D1) ricava 5 V ben stabilizzati, i quali, però, non alimentano direttamente il resto del trasmettitore; infatti abbiamo inserito fra U3 (un LM293-5) e la linea dei 5 volt un commutatore integrato ICL7673 (U5): si tratta di uno switch allo stato solido prodotto dalla Maxim e progettato esplicitamente per i circuiti funzionanti sia a rete che a batteria, nei quali effettua la commutazione automatica tra le due alimentazioni in base al valore di tensione rilevato. Nel nostro caso l'integrato serve a decidere se microcontrollore, encoder e trasmettitore ibrido debbano funzionare con la tensione principale, in arrivo dall'alimentatore tramite l'interruttore

SW1, oppure con i 5 V forniti dal bus USB del computer eventualmente collegato alla presa USB1.

L'ICL7673 funziona così: quando è presente la tensione sul piedino Vp, essa passa fino all'uscita (Vo) e lo fa indipendentemente dal fatto che vi sia o meno tensione su Vs; se la differenza di potenziale applicata tra i piedini Vp e GND viene a mancare o si abbassa al disotto di quella presente tra Vs e GND, lo switch commuta l'uscita su Vs, quindi il circuito viene alimentato da quest'ultima.

Nel nostro circuito, a Vp giungono i 5 volt portati dall'USB; in questo modo, se anche il tutto viene alimentato da un alimentatore esterno collegato ai punti + e - PWR, la tensione proveniente dal cavo USB collegato

ad un computer è sempre prioritaria e forza comunque l'esclusione dell'alimentatore esterno.

Prima di passare al funzionamento del microcontrollore e quindi all'attività del telecomando, soffermiamo la nostra attenzione su due dettagli circuitali, il primo dei quali è il trasmettitore ibrido U4: si tratta di un recente prodotto di casa Aurel, funzionante in UHF a 433,92 MHz e provvisto di un oscillatore modulabile in ampiezza (in modo ON/OFF) stabilizzato da un risuonatore SAW, alla cui uscita è stato posto uno speciale buffer, scopo del quale è smorzare le emissioni armoniche e compensare, nei limiti del possibile, le variazioni di impedenza dell'antenna. Il modulo dispone di un ingresso per la modulazione OOK (On-Off Keying), ingresso che accetta tensioni comprese tra 0 e 5 V e che presenta un'impedenza di 50 kohm. La potenza RF emessa dall'antenna è dell'ordine dei 10 dBm su 50 ohm.

Come funziona

Bene, analizzato l'hardware possiamo parlare del programma di gestione del micro, il quale, dopo l'inizializzazione degli I/O, gira in loop attendendo un evento che può essere la pressione di uno dei 16 pulsanti (funzionamento autonomo) o la connessione a un computer (funzionamento asservito); nel primo caso avvia la routine di trasmissione del codice richiesto, mentre nel secondo si predispone a ricevere gli ordini dal PC, ordini che possono riguardare la trasmissione di un codice o la memorizzazione dell'impostazione di codifica di uno o più canali.

Restiamo, per il momento, sul funzionamento autonomo (stand-alone) e vediamo come il microcontrollore legge la tastiera per valutare quale tasto è stato premuto; per risparmiare linee, la lettura della tastiera viene svolta sfruttandone l'organizzazione a matrice, ed impiegando quattro sole linee collegate in modo abba-

stanza originale: ciò è stato dettato sostanzialmente dai pochi piedini di I/O disponibili nel PIC, certamente insufficienti a gestire una vera matrice di 4 righe per 4 colonne. Con l'accorgimento da noi adottato, la connessione con la tastiera consta di sole quattro linee.

Per leggere lo stato di tutti i sedici pulsanti, si alimentano in sequenza, una per una e molto rapidamente, le linee RB6, RB5, RB4 ed RB3, quindi si verifica il tempo di scarica del condensatore C2; per ogni linea vi sono quattro possibilità, corrispondenti ai tempi dettati dalle resistenze R7, R8, R9 ed R10, quindi, in tutto, il microcontrollore può rilevare 16 condizioni (quattro tempi su quattro linee digitali), che corrispondono alla pressione di tutti i pulsanti della tastiera. Il software sa assegnare a ciascuna combinazione il suo significato, perché conosce i valori ammissibili per ciascun tasto; siccome le quattro resistenze sono di valore decrescente (R7 ha il valore più alto, R10 il più basso), il micro associa la colonna 1 e i rispettivi tasti (P1, P5, P9, P13) al tempo più breve e la colonna 4 (P4, P8, P12 e P16) al tempo più lungo. Le colonne 2 (P2, P6, P10, P14) e 3 (P3, P7, P11, P15) sono associate a tempi di durata intermedie. Ad esempio, se il micro esegue il ciclo di carica/scarica del condensatore C2 attraverso la linea RB6 (riga 1) e rileva il tempo più breve, vuol dire che l'operazione si sta svolgendo tramite la resistenza R10, quindi che è stato premuto il pulsante P1. Se invece il ciclo è svolto tramite la riga 2 (RB5) e si registra il tempo più breve, il tasto premuto è P5.

Quanto all'invio del segnale codificato, ogni volta che viene premuto un pulsante della tastiera, il microcontrollore provvede a svolgere le seguenti operazioni: innanzitutto va a leggere dalla EEPROM la codifica associata a quel pulsante, quindi aggiorna lo stato delle proprie linee RA0, RA1, RA2, RA3, RA5, RB1, RB2, RB7, RC0, RC1, RC6, RC7

(U2) in modo da comporre la stessa codifica sui pin A1÷A12 dell'encoder; poi porta a livello basso la linea RB0, mandando in saturazione il transistor PNP T1, il quale alimenta tramite il collettore l'integrato U2 e il trasmettitore ibrido U4, condizione evidenziata dall'illuminazione del led LD1. Ora l'UM3750 genera le stringhe contenenti la codifica impostata sui suoi pin di impostazione della codifica (A1÷A12) con una temporizzazione determinata dai valori della rete R/C formata da R4 e C7; la sequenza viene ripetuta fin quando il microcontrollore non riporta a livello logico alto la propria linea RB0. Quando sono presenti i dati sul pin 17 di U2, ogni impulso (stato alto) di qualsiasi durata, determina l'emissione di un segnale a radiofrequenza di pari durata. Tale segnale viene irradiato tramite l'antenna che, nel nostro caso, è costituita da uno spezzone di filo in rame lungo 17 cm collegato al pin 8. Terminato l'invio del codice, il micro ripone a zero logico il proprio piedino 21, lasciando andare in interdizione T1, così da scollegare l'ibrido e l'encoder dall'alimentazione.

L'alimentazione del decoder e dell'ibrido per il solo periodo necessario a trasmettere un comando è un accorgimento utile quando si alimenta il circuito a batteria: noi, infatti, abbiamo previsto l'utilizzo con una batteria a 9 V (meglio se alcalina) quando il trasmettitore lavora in stand-alone e pertanto, lo spegnimento di encoder e trasmettitore a telecomando inattivo consente di ridurre la corrente assor-

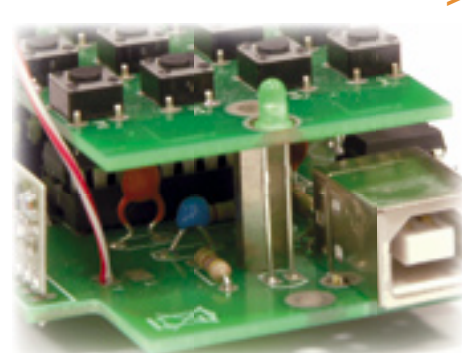
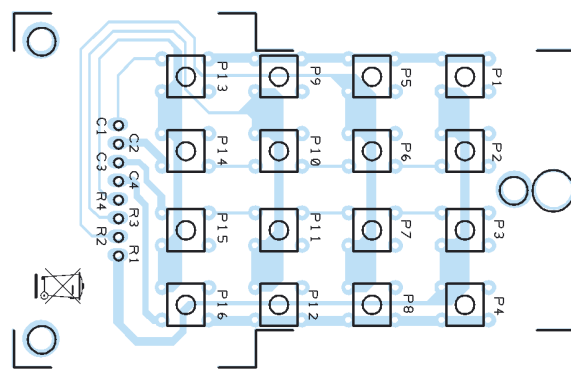
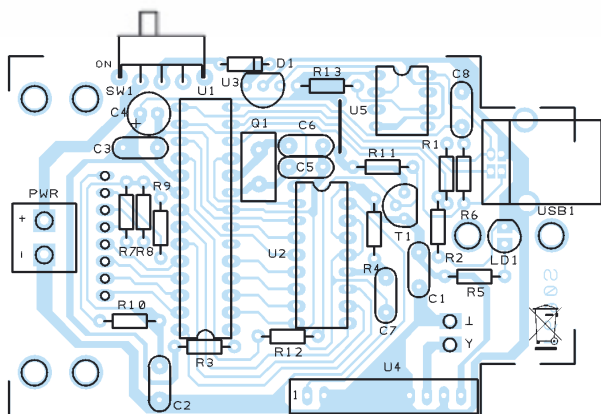
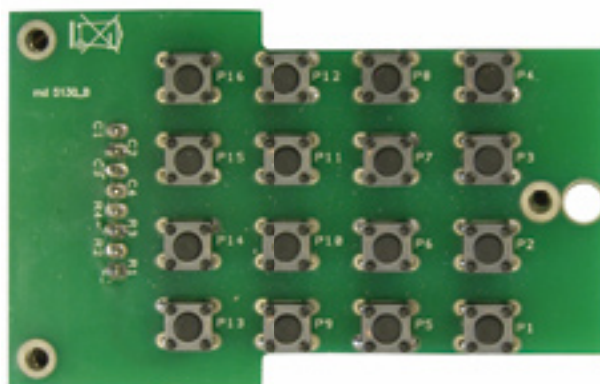
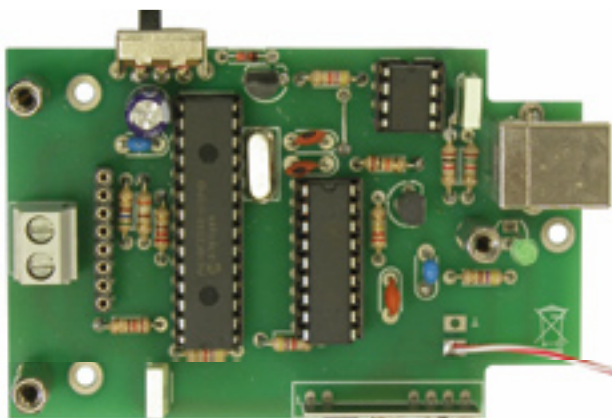


Fig. 1 - Particolare di montaggio del led LD1.

Il radiocomando a 16 canali è costituito da due circuiti stampati: il primo, a sinistra, ospita tutti i componenti attivi, ossia quelli che svolgono le operazioni di codifica e trasmissione dei dati ricevuti tramite la porta USB o la tastiera; quest'ultima, invece, è montata sul secondo circuito stampato, visibile a destra. Il montaggio è molto semplice in quanto entrambi i PCB sono monofaccia e realizzati con componenti ad inserzione.

Prestate attenzione al led verde LD1: deve essere saldato sulla scheda base in modo che la sua testa fuoriesca dal foro praticato sul circuito stampato della tastiera.



ELENCO COMPONENTI:

R1: 1 kohm
 R2: 10 kohm
 R3: 4,7 kohm
 R4: 220 kohm
 R5: 470 ohm
 R6: 1,5 kohm
 R7: 560 ohm
 R8: 390 ohm
 R9: 220 ohm
 R10: 120 ohm
 R11: 10 kohm
 R12: 4,7 kohm
 R13: 4,7 kohm
 C1: 100 nF multistrato
 C2: 220 nF 63 VL poliestere

C3: 100 pF ceramico SMD
 C4: 220 μ F 16 VL elettrolitico
 C5: 15 pF ceramico SMD
 C6: 15 pF ceramico SMD
 C7: 100 pF ceramico
 C8: 220 nF 63 VL poliestere
 Q1: quarzo 20 MHz basso profilo
 U1: PIC18F2550 (MF691)
 U2: UM3750
 U3: LM293-5
 U4: TX-SAW MID-5V
 U5: ICL7673
 D1: 1N4148
 T1: BC557
 LD1: led 3 mm verde

SW1: Deviatore slitta 90°
 P1÷P16: Microswitch

Varie:
 - Morsetti 2 poli
 - Zoccolo 4+4
 - Zoccolo 9+9
 - Zoccolo 14+14
 - Connettore USB-B
 - Torretta F/F 15 mm (4 pz.)
 - Vite 6 mm 3 MA (8 pz.)
 - Strip femmina 8 poli
 - Strip maschio 8 poli 12 mm
 - Circuito stampato base
 - Circuito stampato tastiera

bita. Va da sé che se quando il TX è collegato a un computer, riceve l'alimentazione dalla porta USB, anziché dalla batteria.

La configurazione tramite PC

Per impostazione predefinita del firmware, il trasmettitore funziona così come appena descritto e assegna ai canali dei codici ottenuti facendo variare gli ultimi quattro (A9÷A12) dei 12 bit di codifica; più esattamente, il canale 1 si ottiene con la combinazione 0000 e il sedicesimo con la 1111. I primi otto bit (A1÷A8) invece, sono predefiniti tutti a 0 logico. Tuttavia è possibile modificare a piacimento le combinazioni dei canali, assegnando a ciascuno una differente impostazione, non solo degli ultimi quattro ma anche dei primi otto bit; ciò perché può essere necessario comandare con il nostro TX più apparati che non condividono affatto la stessa impostazione dei primi bit, ma che hanno ciascuno codifiche totalmente differenti gli uni dagli altri.

Per poter personalizzare la codifica di ogni singolo canale bisogna avvalersi dell'aiuto di un PC e dell'apposito programma di gestione da noi scritto in Visual Basic, il quale, oltre a permettere la programmazione dei canali, consente di svolgere altre operazioni quali la lettura dell'attuale impostazione del trasmettitore collegato all'USB e la cancellazione totale della memoria. Per poter operare in abbinamento al PC, viene uti-

lizzata la connessione USB con cavo del tipo AB; fatto ciò, si deve prima installare il driver richiesto e poi avviare il programma di gestione, le cui schermate sono illustrate e descritte dettagliatamente a pagina 18. Dalla finestra principale si possono condurre tutte le operazioni del caso, ossia la lettura dell'attuale impostazione, la modifica dei bit dei singoli canali, la cancellazione della memoria del TX, eccetera. Inoltre, tramite tastiera virtuale posta nel mezzo della finestra di dialogo, è possibile simulare la pressione dei pulsanti della scheda e trasmettere i comandi come se si stesse agendo fisicamente sul trasmettitore.

Realizzazione pratica

Bene, riteniamo si possa lasciare da parte la teoria e passare alla pratica, vedendo i principali aspetti della costruzione del dispositivo; per prima cosa occorre preparare i due circuiti stampati, le cui tracce lato rame sono illustrate in queste pagine; scaricato il master in formato 1:1 dal sito della rivista (www.elettronica.in.it), potete realizzare i due PCB (uno per i pulsanti e l'altro la logica) con il procedimento della fotoincisione. Una volta incise e forate le basette, vi si possono inserire e saldare i componenti occorrenti partendo da quelli a basso profilo (diodi e resistenze) e procedendo con gli zoccoli, i condensatori, il transistor e quant'altro serve; non dimenticate il quarzo e il modulo ibrido. Per il montaggio seguire attentamente l'apposito dise-

gno, ma attenzione: per il momento il led LD1 non deve essere saldato. Non dimenticate il connettore USB di tipo B e la morsettiera bipolare a passo 5 mm per l'alimentazione. Per le connessioni con la basetta che funge da tastiera usate delle strip a passo 2,54 da otto elementi: la presa va disposta sullo stampato base, dietro la morsettiera di alimentazione, mentre la spina va inserita nei relativi fori della tastiera. Finite le saldature, collocate l'ICL7673, l'UM3750 e il microcontrollore (già programmato con l'apposito software...) al proprio posto, badando di far coincidere la loro tacca di riferimento con quella dello zoccolo sottostante, rispettando quanto riportato sul piano di montaggio. A questo punto montate a "sandwich" le due basette, fissandole con colonnine esagonali da 3MA x 15 mm e riprendete il led LD1: esso va saldato sulla scheda del micro, ma i suoi pin devono essere lasciati sufficientemente lunghi affinché la sua testa possa sporgere dal c.s. della tastiera (vedere la Figura 1). Per l'alimentazione della trasmittente è sufficiente una pila da 9 volt da collegare, mediante una presa volante a clip connessa alle piazzole siglate + (positivo, filo rosso) e - (negativo, filo nero) PWR; racchiudete dunque il tutto in un piccolo contenitore plastico sul coperchio del quale realizzerete i fori necessari ad accedere ai sedici pulsanti ed a far spuntare il led. Lateralmente, realizzerete una cava quadrata per la presa USB.

per il

MATERIALE

Il componenti utilizzati in questo progetto sono facilmente reperibili presso qualsiasi negozio di materiale elettronico. Il master in formato 1:1 dei circuiti stampati monofaccia ed il firmware necessario alla programmazione del PIC possono essere scaricati gratuitamente dal sito della rivista (www.elettronica.in.it). Il trasmettitore Aurel (cod. TX-SAW MID-5V) può essere acquistato al prezzo di 6,00 Euro, IVA inclusa.

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

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

Telecontrollo GSM per sistemi Velbus

di Corrado Rossi

Permette di gestire otto carichi collegati a dispositivi di attuazione Velbus, tramite la ricezione di semplici SMS di comando.

Gli SMS possono essere inviati utilizzando qualsiasi telefono cellulare e sono stati studiati per simulare il comportamento dei pulsanti collegati ad una scheda di espansione VMB8PB.

Il dispositivo comunica con la rete Velbus attraverso una scheda seriale VMB1RS.

A conclusione dell'articolo sul radiocontrollo Bluetooth pubblicato sul fascicolo precedente, avevamo anticipato che il progetto successivo per sistemi Velbus sarebbe stato quello di un telecontrollo GSM. Poiché ogni promessa è un debito (e a noi non piace avere debiti di alcun genere!) ecco l'ultimo dei tre progetti preannunciati e dedicati ai sistemi Velbus: si tratta di un sistema funzionalmente molto simile a quelli già visti fino ad ora (operanti a 433 MHz o con interfaccia BT) che però utilizza come mezzo di comunicazione un collegamento telefonico e dei comandi SMS, impartiti tramite

qualsiasi cellulare. Il modem GSM di cui il sistema dispone è il noto Telit GM862, più volte utilizzato in molti nostri progetti: esso comunica con il microcontrollore (un PIC18F458) che, a fronte di un SMS ricevuto, estrae il contenuto, ne verifica la validità e, nel caso l'SMS rappresenti una richiesta valida, genera una stringa di comando da inviare alla rete Velbus. Contrariamente ai due radiocontrolli già pubblicati, questo emula una sola scheda di espansione pulsanti VMB8PB e pertanto occupa un solo indirizzo sulla rete domotica (ricordate che sia il telecontrollo RF che quello Bluetooth occupano due ➤

indirizzi). La scelta di emulare una sola scheda è dettata da un'esigenza particolare che di seguito cerchiamo di spiegare.

I radiocontrolli Velbus BT e RF sono dei sistemi studiati per applicazioni locali in cui l'utente si trova sempre in casa (o molto vicino ad essa) ed è in grado di tenere sotto controllo visivo ciò che accade attorno a lui; pertanto la possibilità di telecomandare via radio o PC più carichi rappresenta una grande comodità.

Il telecomando Velbus GSM, invece, è nato per il controllo da remoto di pochi sistemi specifici, situazione questa in cui l'utente non ha riscontro visivo di quanto accade in casa. Tipicamente, infatti, non si usa un controllo remoto con SMS per accendere lampadine, bensì per comandare altri dispositivi come la caldaia, l'impianto di climatizzazione o di irrigazione, oppure per inviare un comando di spegnimento generale a tutti i sistemi domotici, qualora si abbia il dubbio di aver lasciato le tapparelle aperte o qualche lampada accesa. A questo proposito rammentiamo che nella seconda puntata relativa ai sistemi Velbus, pubblicata sul fascicolo 115, avevamo previsto che il nostro impianto domotico dimostrativo disponesse di un comando di spegnimento luci e chiusura tapparelle generale: ebbene questa situazione si sposa perfettamente con il concetto di controllo da remoto tramite cellulare. Da quanto

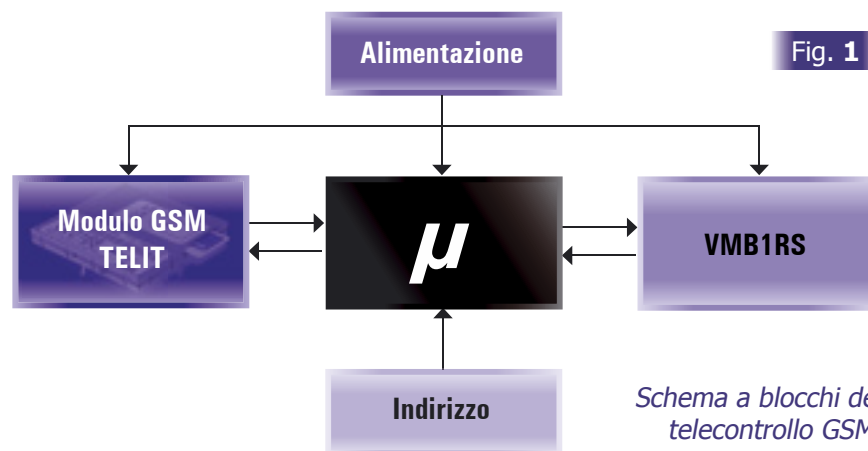


Fig. 1
Schema a blocchi del telecomando GSM.

appena esposto, appare evidente che un telecomando GSM come il nostro è utile per attivare o disattivare dispositivi specializzati (caldaia, climatizzatore...) o per sopperire a dubbi e dimenticanze piuttosto che per attivare carichi generici (lampade). Entriamo ora nel dettaglio del nostro progetto, codificato come FT704, analizzandolo in ogni sua parte.

Struttura del dispositivo

La struttura di questo progetto, in cui ancora una volta abbiamo sfruttato le capacità di conversione da seriale a protocollo Velbus (e viceversa) offerte dalla scheda VMB1RS di Velleman, è molto simile a quella dei due radiocontrolli presentati sui fascicoli precedenti. La principale differenza risiede nel mezzo di comunicazione, qui rappresentato dal modem GSM modello GM862 di Telit. Prestate ora attenzione allo schema a blocchi riportato in Figura 1: potete notare che il sistema è

concettualmente suddivisibile in tre blocchi funzionali: il primo (a sinistra) è il modem GSM; il secondo (a destra) è la scheda seriale VMB1RS. Il terzo blocco, invece (quello centrale) è il microcontrollore, a cui sono affidati i seguenti compiti:

- gestione dell'indirizzo di sistema;
- analisi degli SMS ricevuti e riconoscimento dei comandi di programmazione e controllo;
- configurazione dei parametri di funzionamento del GSM;
- traduzione dei comandi di controllo da "SMS in formato testo" a "stringhe secondo protocollo seriale Velbus".

Quanto appena spiegato relativamente al blocco "microcontrollore" vale per la parte "firmware" dell'apparato; ma per poter interagire fisicamente con la nostra rete domotica, il dispositivo deve essere collegato direttamente alla linea Velbus, composta da 4 conduttori (due per i dati e due per l'alimentazione): le linee

RM ELETTRONICA SNC

vendita componenti elettronici

rivenditore autorizzato:



**FUTURA
ELETTRONICA**

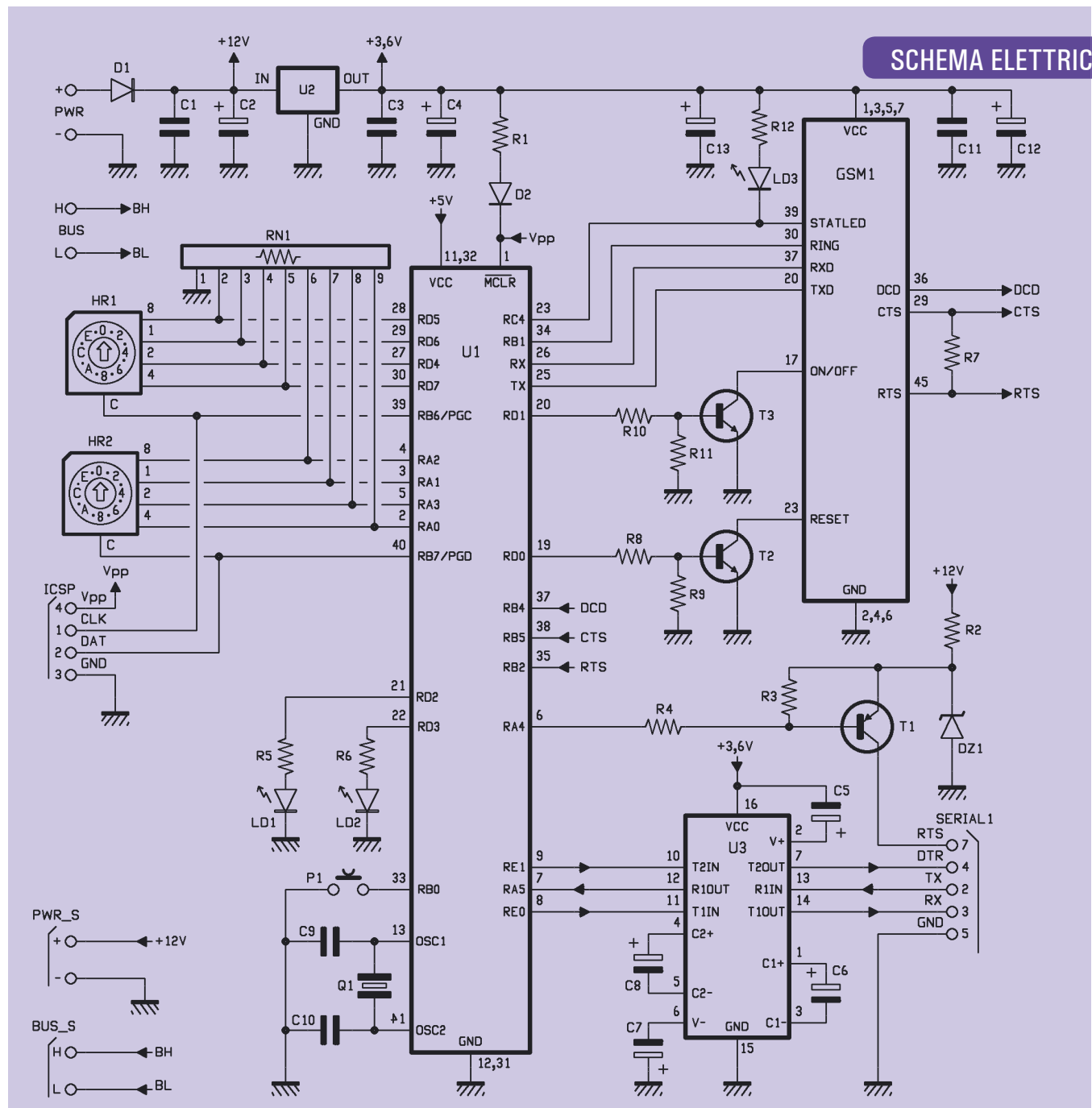
NUOVA

ELETTRONICA

G.P.E.



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



GND e +12Vcc provenienti dal bus alimentano sia la scheda di controllo che il modulo VMB1RS; le due linee dati, invece, vengono collegate direttamente agli ingressi CAN Bus del modulo VMB1RS.

Schema elettrico

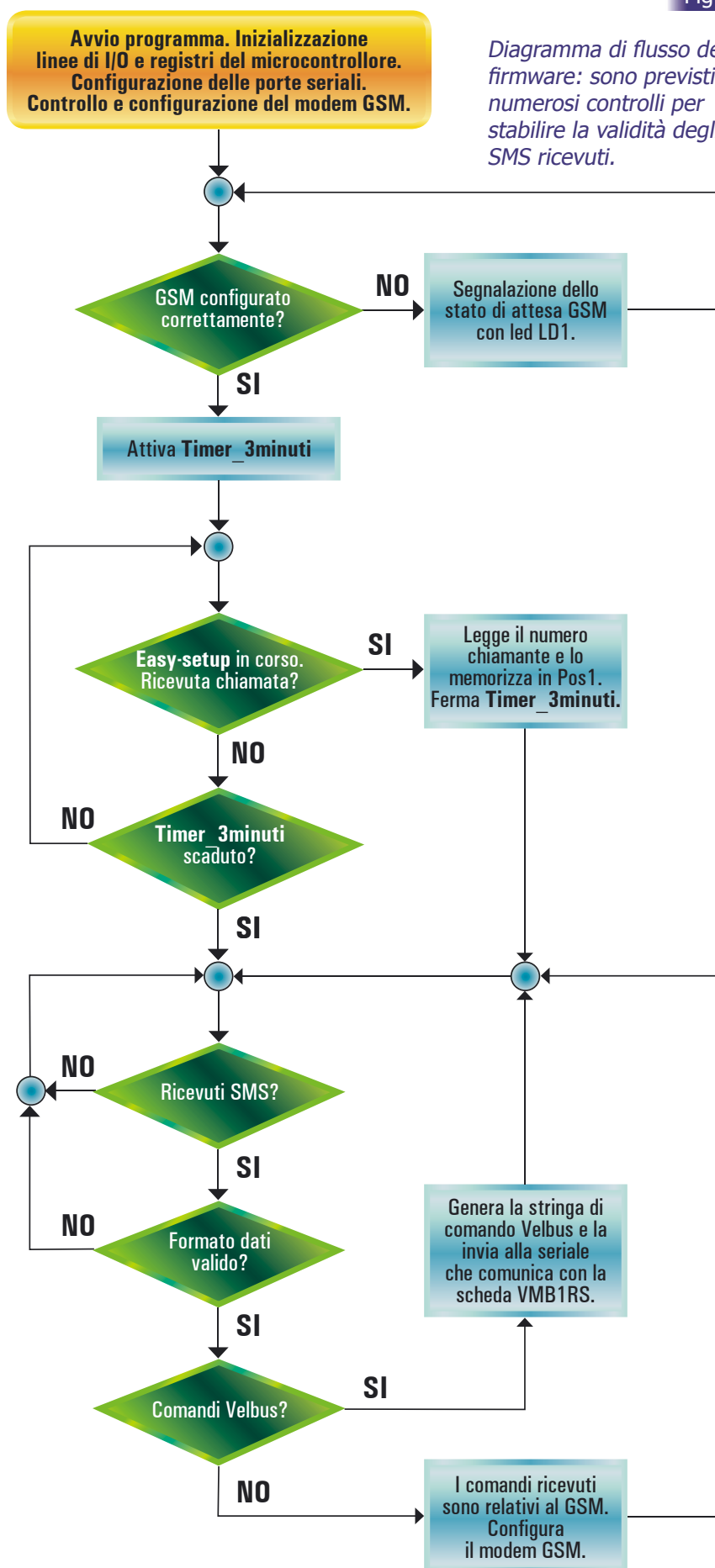
Il cuore del sistema è costituito dal microcontrollore U1, un PIC18F458: a questo punto, quanti di voi hanno letto gli articoli relativi ai progetti precedenti non potranno fare a meno

di notare che, mentre in precedenza avevamo utilizzato dei PIC della famiglia 16F87x, qui abbiamo adottato un 18Fxxx. La scelta è dipesa da due ragioni: la prima riguarda la mole di lavoro con cui il firmware carica il micro, molto più gravosa dovendo gestire un GSM piuttosto che un modem Bluetooth o un ricevitore RF generico; la seconda invece, dipende dalla tensione di alimentazione di 3,6 V del modem GM862, valore accettato dal micro scelto, ma

non dai chip della famiglia 16F87x. Tornando al nostro schema elettrico, il micro U1 comunica con il GSM (GSM1) attraverso la seriale hardware facente capo ai pin 25 e 26 (TX ed RX): tramite queste linee, esso impartisce al modem le istruzioni necessarie per verificare la presenza degli SMS ed eventualmente scaricarli, nonché per impostare i parametri di funzionamento relativi alla rete di fonìa mobile. Attraverso i transistor T3 e T2 (comandati dai

pin di I/O RD1 e RD0), il PIC è anche in grado di accendere e spegnere o semplicemente di effettuare il reset del modem. Con quest'ultimo, inoltre, U1 interagisce attraverso altre cinque linee digitali. La prima è la linea STATLED collegata, oltre che al micro, anche al led LD3: quando questo lampeggia alla frequenza di 1 Hz, il micro comprende che il GSM sta cercando la rete telefonica, mentre se il led si accende per 300 msec ogni due secondi, il micro sa che il GSM è registrato in rete. La seconda linea di controllo è rappresentata dal segnale di RING, con il quale il GM862 informa il PIC della presenza di una chiamata entrante. Tale linea viene utilizzata solo durante la fase di *easy-setup*, funzione questa di cui forniremo dettagli nei capitoli dedicati al firmware ed ai comandi. Le ultime tre linee (DCD, CTS e RTS) vengono utilizzate solamente come elementi di controllo dello stato del collegamento.

Come già anticipato, quanto ricevuto in forma di SMS deve essere tradotto dal micro ed eventualmente inviato alla rete Velbus sotto forma di comando seriale impartito alla scheda VMB1RS: il PIC U1 comunica con tale scheda per mezzo del convertitore di livello MAX3232 (U3) che, a differenza del più noto MAX232, può operare a 3,6 V. Il chip U3 è collegato alle linee RE0 e RA5 del PIC, usate dal firmware per emulare i segnali RX e TX di una porta seriale: pertanto sui pin 14 e 13 di U3 sono disponibili "trasmissione e ricezione" di una seriale a livello RS232. I pin RE1 e RA4 invece, collegati il primo al pin 10 di U3 e il secondo alla base di T1 (attraverso la resistenza R4), servono per generare i segnali seriali RTS (fornito dal collettore di T1) e DTR (presente sul piedino 7 di U3) necessari per alimentare la sezione RS232 della scheda VMB1RS: ricordiamo infatti che tale scheda è isolata galvanicamente dal resto della rete Velbus e



pertanto necessita di un'alimentazione separata. A questo punto abbiamo terminato la descrizione della sezione di circuito dedicata alla comunicazione con il mondo esterno, ma non dobbiamo dimenticare che il micro svolge altri compiti. Esso, infatti, deve gestire anche l'indirizzamento di sistema e verificare lo stato del pulsante di test.

L'indirizzo viene definito attraverso i preselettori binari rotativi HR1 e HR2: su HR1 si imposta il nibble più significativo dell'indirizzo, mentre su HR2 quello meno significativo (ad esempio dato l'indirizzo di scheda 8B, si imposta $HR1 = 8$ e $HR2 = B$). La modalità di test, invece, viene attivata premendo il pulsante P1, collegato alla linea RB0 (pin 33) di U1. Ritourneremo su questi due punti nel capitolo dedicato alle modalità di impiego.

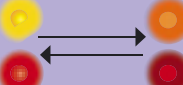
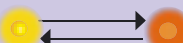
L'ultima sezione dello schema elettrico è quella relativa all'alimentazione. Il circuito viene alimentato a 12 Vdc, prelevati direttamente dal bus Velbus. Il diodo di protezione D1 serve per non danneggiare il telecontrollo qualora si dovesse inavvertitamente applicare la tensione di alimentazione con polarità invertita.

Il filtro costituito dai condensatori C1-C2 ripulisce la tensione in ingresso da eventuali disturbi captati dall'ambiente, mentre il filtro C3-C4 (insieme a C12 e C13), oltre a filtrare ulteriormente la tensione di alimentazione, garantisce una riserva di energia in presenza di picchi di assorbimento quando il GSM va in trasmissione. Lo stabilizzatore di tensione U2 (un LD1086-3.6) fornisce in uscita una tensione di 3,6 Vdc ben stabilizzati.

Diagramma di flusso del firmware

Come avevamo già anticipato in precedenza quando avevamo chiarito le ragioni per cui abbiamo adottato un micro di taglia superiore a quel-

Il telecontrollo comunica i vari stati operativi per mezzo di segnalazioni luminose effettuate tramite tre led, uno rosso, uno giallo e uno verde. Vediamo di seguito le varie situazioni che si possono presentare.

Segnalazione	Descrizione
	Led rosso e giallo accesi per un breve istante: si verifica solo all'accensione, quando il microcontrollore sta configurando tutte le linee digitali.
	Led rosso acceso a luce fissa: il microcontrollore è entrato in comunicazione con il modem GSM. Alla prima operazione di comando viene spento.
	Led verde che lampeggia a 1 Hz: il modem GSM sta cercando la rete telefonica oppure sta effettuando la registrazione in rete. Led verde che rimane acceso 300 msec ogni 2 sec: il modem GSM è registrato in rete e pronto a ricevere SMS o chiamate voce.
	Led rosso e giallo che effettuano cinque lampeggi veloci per poi spegnersi: il modem ha ricevuto un SMS oppure una chiamata voce durante la fase di easy-setup, attiva durante i tre minuti successivi all'accensione.
	Led giallo che effettua dei brevi lampeggi di durata variabile: il telecontrollo ha riconosciuto un comando Velbus e sta inviando i comandi sul CAN Bus.
	Led giallo e rosso che lampeggiano alternativamente: easy-setup in corso. Il GSM ha ricevuto una chiamata voce. Risponde con i cinque lampeggi già visti.

li usati per gli altri progetti Velbus, il firmware di questo telecontrollo è più complesso di quelli precedenti come del resto conferma il diagramma di flusso riportato in Figura 2.

All'accensione, il PIC inizializza le linee di I/O e i registri, quindi configura in maniera opportuna le due porte seriali, la prima realizzata dalla UART hardware (collegata al modem GSM) disponibile nel micro e la seconda emulata via firmware su alcuni pin di GPIO. La procedura di inizializzazione procede e termina con la configurazione del modem GSM. Durante questa fase, il GM862 ha il controllo della situazione: esso infatti deve verificare la SIM card e registrarsi in rete (procedura che necessita di alcuni secondi); pertanto il micro, prima di procedere con l'esecuzione del programma, deve attendere che il modem Telit confermi di essere operativo. Fino a che il GSM non è pronto, lo stato di attesa viene segnalato dal led rosso LD1, acceso a luce fissa.

A registrazione in rete effettuata, il programma attiva un timer, programmato per scadere dopo 3 minuti: durante questo intervallo il sistema entra nella fase di easy-setup, ossia

una situazione in cui il telecontrollo attende una chiamata voce da un cellulare. A chiamata ricevuta il programma riconosce il numero del chiamante e lo memorizza nella prima posizione della rubrica telefonica, definendo che tale numero ha pieno controllo sul sistema, anche se invia comandi SMS sprovvisti di password (salvo casi particolari). La fase di easy-setup termina allo scadere del timer oppure alla ricezione della chiamata voce.

Il programma, quindi, si pone in un ciclo di attesa in cui interroga regolarmente il GSM per verificare se esso ha ricevuto degli SMS. In caso affermativo, il firmware estrae il testo contenuto nell'SMS e ne verifica la validità tanto del testo quanto del mittente e della password. Se il messaggio non è valido viene scartato, sia dal telefono che dal buffer del programma, quindi il ciclo di attesa riparte dall'inizio. In caso contrario il programma deve comprendere se si tratta di un comando destinato alla rete Velbus oppure di una richiesta di configurazione verso il modem. Nel primo caso il firmware genera la stringa di comando seriale che viene successivamente inviata alla scheda ➤

Tabella 1 - Il **protocollo di comunicazione dell'interfaccia GSM**

Funzione	Comando	Valore di default	Pwd obbligatoria
Cambiare Password.	PWDxxxxx;yyyyy	12345	Sempre.
Memorizzare un numero (massimo 8 numeri, da 19 caratteri massimi ognuno).	NUMxxxxxxxxnnn;yyyyy	12345	Solo se la pos. x non è libera e il cellulare non è tra gli 8 noti.
Cancellare un numero.	NUMx;yyyyy	-	Sempre.
Verificare i numeri memorizzati.	NUM?;yyyyy	-	Sempre.
Reset completo.	RES;yyyyy	-	Sempre.
Pulsante 1÷8 attivo: simulazione pressione breve seguita dal rilascio dopo 1 secondo.	OUTx;yyyyy	-	Solo se proviene da un telefono non memorizzato.
Pulsante 1÷8 attivo: pressione breve seguita da pressione lunga e rilascio dopo 1 sec.	OUTLx;yyyyy	-	Solo se proviene da un telefono non memorizzato.
Posto alla fine dell'SMS disabilitare la risposta per quel comando.	RISP	-	-
Easy-setup.			
Tenendo premuto il pulsante P1 viene inviata sul CAN Bus (come test) la simulazione della pressione lunga dei canali da 1 a 8.			

VMB1RS; nel secondo, invece, fa sì che il micro invii al GSM i relativi comandi di configurazione, in funzione dell'SMS ricevuto. In entrambi i casi, infine, al termine delle due procedure il programma ritorna al ciclo di attesa dell'SMS.

Tenete presente una cosa molto importante: il diagramma di flusso che abbiamo commentato è stato semplificato, ma in realtà, il programma svolge molte altre funzioni, come ad esempio il controllo del mittente e della password contenuta negli SMS, la generazione dei messaggi di conferma o la loro disabilitazione, il controllo dei "multimessaggi" eccetera. Descrivere tutte le possibili opzioni del firmware esula dal nostro scopo, che è quello di farvi capire i concetti basilari con cui il sistema opera. Qualora siate interessati ad approfondire il discorso, potete scaricare l'intero firmware dal sito Internet della rivista (www.elettronica.in).

Il set di comandi

Esattamente come per il radiocontrollo Bluetooth (codice progetto FT703), anche in questo caso è stato necessario ideare un protocollo di comandi tramite il quale configurare il dispositivo ed interagire con la

rete Velbus. Tale set, da inviare al sistema sotto forma di messaggio SMS è riassunto in Tabella 1; esso comprende sia i comandi di configurazione (utilizzati per definire i parametri di funzionamento del modem GSM) che quelli di attuazione, ossia quelli che simulano la pressione di un pulsante sulla scheda VMB8PB emulata dal telecontrollo. Iniziamo la descrizione dei comandi partendo da quelli di configurazione, facendo una breve premessa: la sintassi del comando, tipicamente, è costituita da una stringa di testo divisa in due parti, il comando e la password, separati dal simbolo ";". Ricordate inoltre che la password di default è "12345".

Il comando per modificare la password è PWDxxxxx;yyyyy, in cui xxxxx rappresenta la nuova password e yyyyy quella attuale. A comando eseguito, la nuova password diventa immediatamente operativa. Per rendere il dispositivo quanto più possibile versatile, abbiamo previsto la possibilità di memorizzare otto numeri di telefono, ai quali è consentito inviare comandi sprovvisti di password, a meno che non si tratti di richieste particolari quali cancellazione di una posizione in rubrica

oppure ritorno alla configurazione di default (situazioni queste in cui la password è sempre obbligatoria indipendentemente dal numero che invia l'SMS), ed il comando preposto è NUMxxxxxxxxnnn;yyyyy, in cui x è la posizione in rubrica su cui si vuole memorizzare il numero nnnnnnnnnn (lungo al massimo 19 cifre), e yyyyy è la password attuale. La password è obbligatoria quando la posizione su cui si sta memorizzando il numero non è libera. Sempre relativamente ai numeri memorizzati, è possibile inviare sia un messaggio per cancellare un numero in particolare (NUMx;yyyyy, in cui x è la posizione in rubrica e yyyyy la password attuale) sia per richiedere l'elenco dei numeri memorizzati (NUM?;yyyyy). Il telecontrollo risponde ad entrambi i comandi inviando a sua volta un SMS di conferma al numero da cui ha ricevuto il comando. Qualora si voglia richiamare la configurazione di default del dispositivo, il comando da inviare è RES;yyyyy. Poiché la configurazione di fabbrica prevede che ad ogni comando valido ricevuto il telecontrollo risponda con un SMS di conferma, abbiamo previsto il comando RISP che, posto alla fine dell'SMS e separato dal comando

Riconoscimento SMS e trasmissione comando

Il protocollo Velbus è piuttosto complicato, ma grazie alla scheda seriale VMB1RS, che converte da seriale a Velbus e viceversa, scrivere del firmware per inviare dei comandi sul CAN Bus diventa molto facile. In questo riquadro riportiamo le tre routine più significative. ANALIZZASMS si occupa di estrarre il testo dall'SMS ricevuto, di identificare i comandi ed eseguirli, di capire se l'SMS è un multimessaggio e se il messaggio di risposta deve essere disabilitato. TRASMETTI, invece, è la routine che formatta la stringa di comando Velbus, stringa che viene poi inviata alla scheda VMB1RS tramite la terza routine TXVEL.

```

*****
ANALIZZASMS:      ' Inizio Routine lettura SMS.
  FOR TMP=0 TO LUNGMESS
    IF BUFF[TMP]="," THEN ' Controlla il messaggio.
      MULTIMESS=1 ' Conta le virgole presenti
      POSMULTIMESS=TMP ' per definire se il comando
      TMP=200 ' è un multimessaggio.
      PAUSE 1000
    ELSE
      MULTIMESS=0
    ENDIF
  NEXT TMP
  FRASE=0
*****
  SELECT CASE BUFF[0]
    CASE "N","n" ' Interpretare il messaggio ed
      ' agisce in funzione di esso
      ' tramite un ciclo di
      ' select case.
      .....
    CASE "P","p"
      .....
    CASE "R","r"
      .....
    CASE "O","o" ' Riconosce comando OUT o OUTL
      PRIORITA=$F8 ' Imposta la priorità.
      NBYTE=4 ' Crea un buffer d 4 byte.
      IF BUFF[3]="L" OR BUFF[3]="I" THEN ' Ric. OUTL
        CH=buff[4]-48 ' Riconosce numero ingresso.
        PRESSIONE=1 ' Imposta tasto premuto.
        CH=CH-1
        GOSUB TRASMETTI ' Invia stringa.
        PRESSIONE=2 ' Imposta "mantenuto premuto".
        PAUSE 100
        GOSUB TRASMETTI ' Invia stringa.
        PAUSE 1000
        PRESSIONE=0 ' Imposta "puls. rilasciato".
        GOSUB TRASMETTI ' Invia stringa.
      ELSE
        CH=buff[3]-48 ' Riconosce numero ingresso.
        PRESSIONE=1 ' Imposta "puls. premuto".
        CH=CH-1
        GOSUB TRASMETTI ' Invia stringa.
        PAUSE 1000
        PRESSIONE=0 ' Imposta "puls. rilasciato".
        GOSUB TRASMETTI ' Invia stringa
      ENDIF
      frase=8 ' Seleziona frase da inviare.
    END SELECT
  *****
  ' Termine esecuzione comando.
*****

*****
' Codice gestione risposta.
IF RISPOSTA=1 THEN ' Se risposta = 1, comando
  WRITE 170+FRASE,1 ' RISP non presente, invia SMS
  IF MULTIMESS=0 THEN ' di risposta ad ogni comando.
    FOR TMPW2=170 TO 190
      READ TMPW2,TMP
      IF TMP=1 THEN
        FRASE=TMPW2-170
        GOSUB INVIOSMS
      ENDIF
    NEXT TMPW2
    FOR TMPW2=170 TO 190
      WRITE TMPW2,0
    NEXT TMPW2
  ENDIF
  ' Termine funzione RISPOSTA
ENDIF
IF MULTIMESS=1 THEN ' Codice gest. multimessaggio.
  FOR TMP=0 TO LUNGMESS ' Se multimess=1 analizza
    BUFF[TMP]=BUFF[TMP]+POSMULTIMESS+1 ' comando
    NEXT TMP
  ENDIF
  ' successivo.
  ' Termine funzione MULTIMESS.
  ' Fine Routine lettura SMS.
RETURN
*****
TRASMETTI:      ' Routine formatta stringa.
  FOR TMP=0 TO 7 ' Azzerà buffer.
    BUFFV[TMP]=0
  NEXT TMP
  TMP=1
  TMP=TMP << CH
  PRIORITA=$F8
  NBYTE=4
  SELECT CASE PRESSIONE
    CASE 0
      BUFFV[2]=TMP
    CASE 1
      BUFFV[1]=TMP
    CASE 2
      BUFFV[3]=TMP
  END SELECT
  GOSUB TXVEL
  ' Chiama routine trasmissione.
  ' Fine routine format stringa.
  ' Routine trasm. seriale.
  ' Accende LED TX in corso.
  ' Calcola il checksum.
  RETURN
TXVEL:
  HIGH LD2
  GOSUB CALCOLACHK
  SEROUT2 TXV,6, [$OF,PRIORITA,ADDRESS,NBYTE] ' Invia.
  FOR TMP=0 TO NBYTE-1
    SEROUT2 TXV,6, [BUFFV[TMP]]
  NEXT TMP
  SEROUT2 TXV,6,[CHK,$04]
  LOW LD2
  ' Spegne LED TX in corso.
  ' Fine routine invio.
RETURN

```

tramite una virgola (,) ordina al dispositivo di non inviare alcun messaggio di risposta al telefono che ha inviato l'SMS.

Nel corso dell'articolo abbiamo già accennato alla funzione easy-setup che ora vi spieghiamo dettagliatamente: come avrete potuto notare, la configurazione del dispositivo prevede l'uso di comandi sempre corredati da password, a meno che non siano inviati da numeri già memorizzati (salvo alcuni casi). Ebbene con la funzione di easy-setup è possibile memorizzare il primo numero di telefono in posizione 1 semplicemente accendendo il dispositivo ed effet-

tuando verso di esso una chiamata voce entro tre minuti dall'accensione: in questo modo il telecomando memorizza il numero e lo considera sempre valido, quindi ogni comando inviato da esso può essere sprovvisto di password (vedere le eccezioni in Tabella 1).

Abbiamo così esaurito la spiegazione relativa alla configurazione della sezione telefonica; vediamo ora i comandi per interagire con la rete Velbus. A tal proposito, pur avendolo già detto nel corso dell'articolo, ricordiamo che il sistema emula una scheda di espansione a otto pulsanti modello VMB8PB: tale scheda, che

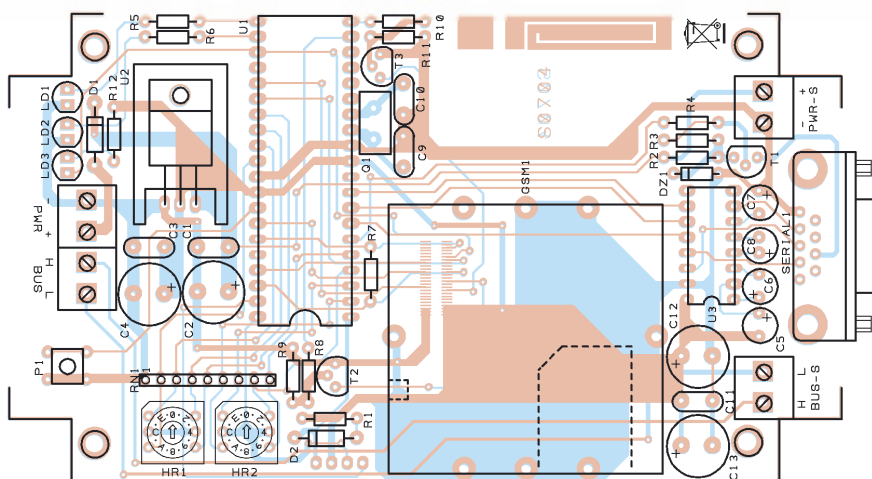
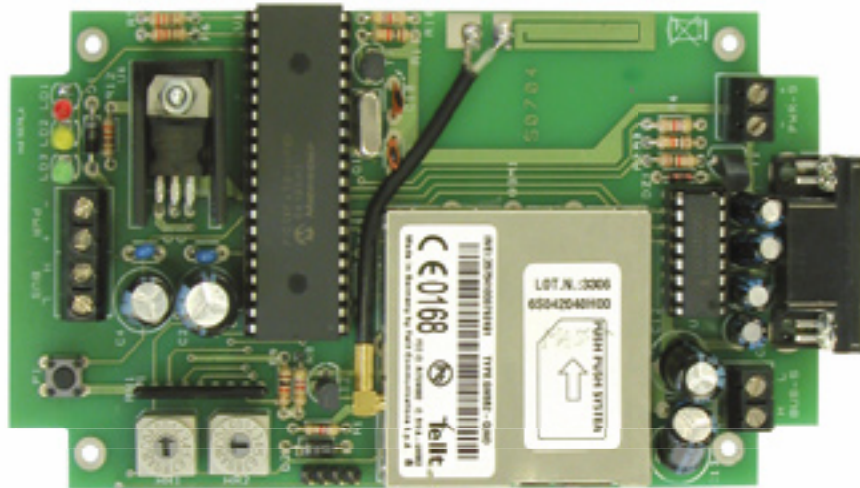
opera con i classici pulsanti meccanici, è in grado di rilevare e segnalare gli stati di "pulsante premuto (pressione breve)", "pulsante mantenuto premuto (pressione lunga)" e "pulsante rilasciato". Operando con gli SMS non è possibile gestire in tempo reale questi tre stati distinti, tipici di un contatto meccanico, quindi abbiamo dovuto definire il comando OUTx per simulare una pressione breve sull'ingresso x (da 1 a 8) e OUTLx per simulare una pressione lunga.

Entrambi i comandi necessitano di password se provengono da un telefono diverso dagli otto memorizzati ➤

Il telecontrollo GSM, una volta realizzato, si presenta pulito e ordinato. In linea di massima, realizzare il dispositivo non è particolarmente difficile: esso infatti utilizza esclusivamente componenti ad inserzione, fatta eccezione per il connettore GSM1 (per il modem GSM) che, essendo SMD, deve essere montato con la massima precisione e cautela.

ELENCO COMPONENTI:

R1, R4, R8, R10 : 4,7 kohm
 R2: 1 kohm
 R3: 10 kohm
 R5, R6, R12: 330 ohm
 R7: -
 R9, R11: 10 kohm
 RN1: Rete resistiva 8 x 10 kohm
 C1, C3, C11: 100 nF multistrato
 C2: 470 μ F 25 VL elettrolitico
 C4: 470 μ F 16 VL elettrolitico
 C5÷C8: 1 μ F 100 VL elettrolitico
 C9, C10: 10 pF ceramico
 C12, C13: 470 μ F 16 VL elettrolitico
 D1, D2: 1N4007
 DZ1: Zener 7,5 V 400mW
 T1: BC557
 T2, T3: BC547
 Q1: Quarzo 20 MHz
 U1: PIC18F458 (MF704)
 U2: LD1086-3.6
 U3: MAX3232
 GSM1: Modulo GSM GM862
 LD1: led 3 mm rosso
 LD2: led 3 mm giallo
 LD3: led 3 mm verde



HR1, HR2: Sel. rotativo esadecimale
 P1: Microswitch

Varie:

- Morsettiere 2 poli (4 pz.)
- Strip maschio 4 poli
- Connettore 50 poli per modulo GSM

- Connettore DB9 maschio
- Zoccolo 8+8
- Zoccolo 20+20
- Dissipatore (ML26)
- Vite 10 mm 3 MA
- Dado 3 MA
- circuito stampato codice S704

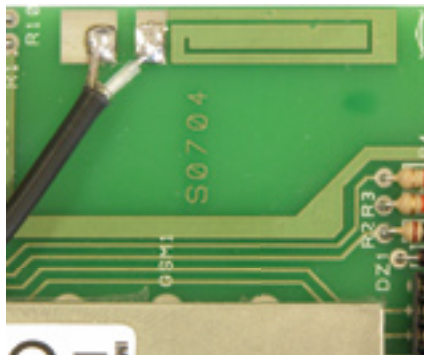
in rubrica. Quando il telecontrollo riceve l'istruzione OUTx, il programma invia sul bus RS485 due sentenze di comando, una dopo l'altra, contenenti l'informazione "pulsante numero x premuto" (la prima) e "pulsante numero x rilasciato" (la seconda). A fronte del comando OUTLx, invece, il sistema genera tre stringhe (pulsante numero x premuto, mantenuto premuto ed infine rilasciato). Manca ancora un ultimo punto: per quanto non si

tratti di un comando SMS, mantenendo premuto per qualche istante il pulsante P1, il sistema genera una serie di sentenze Velbus esattamente come se avesse ricevuto otto SMS OUTL (uno per ogni ingresso virtuale). Questa funzione non serve durante la normale attività però è utile per generare del traffico di test in fase di installazione della rete. Esaurita la descrizione dei comandi, passiamo ora alla parte pratica, sicuramente più divertente.

Realizzazione pratica

La realizzazione di questo dispositivo non è particolarmente complessa: esso infatti fa uso quasi esclusivamente di componenti a montaggio tradizionale (ad inserzione). Diciamo "quasi" perché un componente SMD è presente: si tratta del connettore per il modem GM862, connettore che va montato facendo uso di una buona dose di calma e di precisione. Per prima cosa bisogna disporre del circuito stampato, che potete realizzare con la

Figura 3 - L'antenna per il modem GSM è ricavata direttamente sul circuito stampato e il cavetto di adattamento deve essere saldato come visibile nella figura a destra. Se volete inserire il dispositivo in un contenitore, quest'ultimo deve essere in plastica, per non schermare il segnale RF. Con contenitori metallici è necessario utilizzare un'antenna esterna.



tecnica della fotoincisione (il master relativo - realizzato in doppia faccia - può essere scaricato gratuitamente dal nostro sito www.elettronica.in.it). A chi non se la sentisse di affrontare la realizzazione in proprio di un circuito stampato come questo (non impossibile, ma sicuramente impegnativo) suggeriamo di richiederlo direttamente all'azienda che realizza i CS dei nostri prototipi (www.mdsrl.it), indicando il codice del circuito che interessa (S0704). Una volta in possesso del circuito stampato, potete iniziare il montaggio proprio con il connettore SMD GSM1. Per fare ciò occorre un saldatore a punta fine con potenza di 30 W, del fluxante in gel, una lente di ingrandimento (meglio se montata su supporto o su una lampada) ed una pinzetta a becchi. Identificate il punto esatto in cui il connettore deve essere montato e, aiutandovi con la lente, fate fondere un po' di stagno su due piazzole diagonalmente opposte; depositate quindi del fluxante su tutta l'area del connettore ed appoggiateci sopra quest'ultimo che, aderendo al fluxante, rimane fermo sul PCB. Centrate perfettamente il connettore controllando gli allineamenti di ogni pin sulle rispettive piazzole, quindi fate fondere con il saldatore lo stagno depositato in precedenza sui due pad diagonalmente opposti. Verificate l'allineamento ancora una volta e procedete con molta calma e cautela saldando ogni singolo pin del connettore. Il rischio di fare un cortocircuito tra pin adiacenti è elevato: se dovesse accadere, non vi preoccupate, continuate a saldare tutti i pin

mancanti quindi, a montaggio ultimato, depositate una minima quantità di fluxante su ogni corto, pulite molto bene la punta del saldatore e ponetela a contatto dei pin interessati: lo stagno in eccesso tende a risalire in parte sui pin e in parte sulla punta. Ripulite ancora una volta quest'ultima e ripetete la procedura "fluxante-saldatore-pulizia" fino a che lo stagno in eccesso non sarà completamente risalito sulla punta del saldatore. Ricordate sempre che il fluxante è "il miglior amico di chi salda componenti SMD". Giunti a questo punto la parte più impegnativa è terminata e quindi potete proseguire con il montaggio dei componenti a basso profilo (resistenze, diodi, la rete resistiva RN1, i condensatori non polarizzati, il pulsante P1 e i preselettori binari HR1 e HR2). Saldare gli zoccoli del microcontrollore e del driver RS232: entrambi hanno la tacca di riferimento in basso, osservando il PCB come riportato nel piano di montaggio. Proseguite inserendo i tre transistor T1, T2 e T3 (disponendo il lato piatto come visibile nella pagina di fianco), le tre morsettiere a passo 5 mm, il quarzo Q1 (montato verticalmente), tutti i condensatori elettrolitici, il connettore pin strip a passo 2,54 mm per la programmazione "in circuit" (ICSP) ed il connettore seriale DB9 sul quale viene collegata la scheda VMB1RS. Successivamente montate lo stabilizzatore di tensione (U4) che, dovendo alimentare anche il GSM oltre che il micro e le periferiche, necessita di un piccolo dissipatore. Piegate pertanto i pin di U4 a 90°, come visibile nel piano di montaggio:

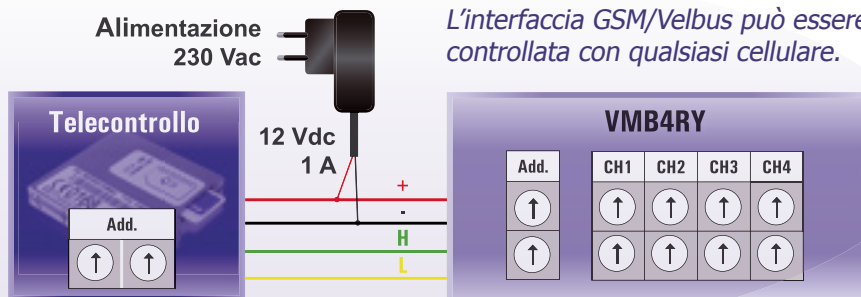
in questo modo è possibile appoggiare il lato metallico a ridosso del dissipatore, a sua volta a contatto del PCB, e fissare meccanicamente il tutto con una vite da 3 MA e relativo dado. Una volta serrata la vite, stabilizzatore e dissipatore non possono più muoversi quindi procedete saldando i tre pin di U4. Montate ora i led LD1, LD2 e LD3 tagliandone i contatti alla lunghezza di 28 mm e facendoli sporgere dal lato saldature di 2 mm circa. In questo modo le loro teste potranno fuoriuscire leggermente dal coperchio del contenitore plastico in cui eventualmente inserirete il telecomando seguendo la procedura spiegata un po' più avanti. Saldare infine il cavetto di adattamento per l'antenna GSM (ricavata direttamente sul PCB) come visibile sul piano di montaggio e in Figura 3. Per il momento non innestare i due chip (PIC e MAX3232) ed il modem nei rispettivi zoccoli. Il micro deve essere programmato con il file MF704.hex, scaricabile dal sito: per far ciò vi occorre un programmatore idoneo, in grado di effettuare la programmazione ICSP.

Collaudo

Giunti a questo punto, disponete del telecomando montato e pronto per il collaudo. La prima verifica da fare è quella delle tensioni di alimentazione: effettuate un'ultima analisi visiva del PCB, per essere certi di aver montato correttamente tutti i componenti e di non aver fatto cortocircuiti indesiderati. Il microcontrollore, il driver RS232, il modem GSM e la scheda VMB1RS non devono essere inseriti. Fornite al circuito una tensione di 12 Vdc, ottenuta da un alimentatore stabilizzato in grado di erogare almeno 500 mA, e misurate con l'ausilio di un multimetro digitale le varie tensioni sul circuito. La tensione di 11,4 Vdc circa deve essere presente su un capo della resistenza R2 (quello opposto al diodo DZ1), sul morsetto "+" della morsettiere PWR-S e sul pin 1 (IN) dello stabilizzatore U2. La tensione di ➤



Fig. 4



L'interfaccia GSM/Velbus può essere controllata con qualsiasi cellulare.

3,6 Vdc, invece, deve essere presente sui pin U1-1, U3-16 e GSM1-1,3,5,7. Infine sul catodo dello zener DZ1 dovete misurare una tensione di 7,5 V circa. Ora potete spegnere il circuito e inserire il MAX3232 nel proprio zoccolo. Riaccendete il circuito e verificate che sui suoi pin 2 e 6 siano presenti le tensioni di +7 e -7 Vdc circa. Spegnete nuovamente il circuito, quindi inserite il microcontrollore programmato, il modem GSM (saldando al PCB i pin di fissaggio meccanico di cui dispone) e la scheda seriale VMB1RS sui rispettivi connettori. Per quest'ultima, non dimenticate di collegare i quattro fili "+, -, Bus H e Bus L" alla morsettiera del circuito base (Figura 5). Assicuratevi inoltre di avere innestato nel modem il connettore del cavetto per l'antenna e inserito nel porta-SIM una tessera valida con il PIN disabilitato (altrimenti il GSM non può funzionare). Alimentate nuovamente il circuito e verificate le indicazioni dei led:

- i led LD1 (rosso) e LD2 (giallo) si accendono per un breve istante;
- il led rosso rimane acceso mentre tutti gli altri sono spenti;
- il led LD3 (verde) inizia a lampeggiare alla frequenza di 1 Hz, per indicare che il GSM sta effettuando la ricerca della rete e la successiva registrazione mentre il led LD1 resta acceso a luce fissa, indicando che il GSM non presenta anomalie;
- il led LD3 inizia a lampeggiare con cadenza di 300 msec ogni 2 secondi, indicando che il modem GSM è registrato in rete.
- la scheda VMB1RS presenta il led rosso dello stato di ON acceso.

A questo punto il sistema è pronto per poter essere configurato.

Note di utilizzo

E' arrivato ora il momento di utilizzare il nostro telecontrollo GSM: per procedere, è necessario che disponiate di un attuatore Velbus (noi abbiamo utilizzato durante le prove l'unità a quattro relé VMB4RY) e di un telefono cellulare. Lo schema funzionale, visibile in alto nel disegno di Figura 4, mostra un sistema costituito da un telecontrollo GSM, un dispositivo VMB4RY, un alimentatore stabilizzato a 12 Vdc in grado di fornire una corrente di almeno 1 A e un cellulare.

Un esempio di configurazione

Visto lo scopo prettamente didattico di questo paragrafo, riproponiamo l'esempio di configurazione già utilizzato nell'articolo relativo all'utilizzo del telecontrollo Bluetooth, pubblicato sulla rivista numero 120, esempio molto semplice, ma di grande efficacia, in cui ogni uscita a relé viene configurata per operare in modalità "toggle" (ossia bistabile: il primo comando ricevuto attiva il relé, il secondo lo disattiva) e controllata

da due pulsanti differenti, secondo la mappatura visibile in Tabella 2.

In pratica i relé sono comandati ognuno da due canali (SMS) distinti (1-5, 2-6, 3-7 e 4-8): questa configurazione non presenta utilità pratica, ma vi permette di prendere confidenza con tutti i comandi da utilizzare nelle applicazioni pratiche reali.

Scelta degli indirizzi

Ancora una volta, il buon senso suggerisce di utilizzare indirizzi in sequenza, più facili da ricordare, scelti tra quelli utilizzabili. Ricordiamo che quelli riservati sono riportati sui manuali di utilizzo di ciascun dispositivo Velbus. Noi abbiamo assegnato gli indirizzi 02 al telecontrollo (sui rotary switch impostiamo HR1 a 0 e HR2 a 2) e 01 al VMB4RY. La rete deve essere adattata in impedenza con una resistenza da 120 ohm, inseribile tramite il ponticello TERM di cui sono dotati sia il controllo a relé che la scheda seriale. Noi abbiamo lasciato aperto il ponticello sulla seriale e chiuso quello del VMB4RY. A sistema in funzione, sia la scheda seriale che quella a relé presentano un led rosso acceso.

Tabella 2 - Assegnazione pulsante virtuale / uscita

Uscita VMB4RY	Primo pulsante virtuale di comando (attivazione)	Secondo pulsante virtuale di comando (disattivazione)
Uscita Relé 1	OUT1	OUT5
Uscita Relé 2	OUT2	OUT6
Uscita Relé 3	OUT3	OUT7
Uscita Relé 4	OUT4	OUT8

Programmare la sezione telefonica

Prima di poter operare sulla rete Velbus, il telecomando deve essere configurato con i parametri di funzionamento corretti e, per fare ciò, dovete disporre del dispositivo acceso e con la SIM inserita (rispondente ad esempio al numero +393334445555) nonché del cellulare (+393337778888, sempre per esempio). Il primo SMS da inviare (non obbligatorio) è

RES;12345

con il quale si riporta il telecomando alla configurazione di default, operazione durante la quale viene anche cancellata la rubrica, che poteva contenere dati errati o semplicemente dimenticati. Il sistema risponde facendo lampeggiare i led LD1 ed LD2, nonché inviando al cellulare il seguente messaggio di conferma:

+393334445555 Reset eseguito

Spegnete e riaccendete il telecomando, attendete il tempo necessario affinché completi la registrazione in rete, quindi effettuate una chiamata voce verso di esso: il sistema fa lampeggiare alternativamente LD1 e LD2, quindi li accende a luce fissa e abbatte la chiamata. Infine fa lampeggiare cinque volte i suddetti led per confermare di avere memorizzato in posizione 1 della rubrica il numero del telefono con cui avete effettuato la chiamata voce. A questo punto la fase di easy-setup è terminata.

Per essere certi che il numero è stato memorizzato, potete inviare il comando

NUM?;12345

al quale il telecomando risponde con il messaggio

+393334445555

Numeri memorizzati:

Pos1 +393337778888

Pos2

Pos3

Pos4

Pos5

Pos6

Pos7

Pos8

Volendo aggiungere il numero +393992226666 in posizione 8 della rubrica, dovete inviare il comando

NUM8+393992226666

per il quale ottenete in risposta il messaggio

+393334445555

Numeri memorizzati:

Pos1 +393337778888

Pos2

Pos3

Pos4

Pos5

Pos6

Pos7

Pos8 +393992226666

Se lo desiderate, potete registrare altri numeri, ma quanto memorizzato fino ad ora vi permette già di interagire via cellulare con la rete Velbus.

Nota: anche se è abbastanza ovvio, voi dovete usare i vostri numeri di telefono reali; noi abbiamo usato dei numeri inventati, per non correre il rischio di violare la privacy di qualche ignaro utente di telefonia mobile.

Accoppiare i pulsanti virtuali alle uscite a relé

In base alle linee guida stabilite all'inizio del paragrafo, dal manuale del VMB4RY si ricavano le seguenti informazioni:

- per operare in modalità toggle, i rotary switch del modo operativo di ogni canale devono essere impostati a FF;
- l'assegnazione di un pulsante di controllo bistabile si effettua im-

stando sui rotary switch di indirizzo D1 per il canale 1, D2 per il canale 2 e così via;

- il pulsante da accoppiare va mantenuto premuto fino a quando il VMB4RY non risponde attivando il relativo relé: secondo la sintassi descritta in tabella 1, ciò corrisponde all'invio tramite SMS del comando OUTL, che simula la pressione lunga di un pulsante.

Procedete con il primo relé che, come da tabella, deve essere controllato dai pulsanti virtuali 1 e 5.

Impostate i rotary switch di indirizzo della scheda VMB4RY a D1: il led CH1 lampeggia.

Inviare ora l'SMS

OUTL1,OUTL5

al quale il telecomando reagisce generando due sentenze seriali Velbus che fanno commutare per due volte il relé del canale 1, confermando così che tale relé viene comandato in toggle dai comandi OUT1 e OUT5. A questo punto il telecomando risponde con due messaggi di conferma contenenti il testo

Commutazione avvenuta

Ripristinate l'indirizzo della scheda VMB4RY a 01 quindi inviate il comando

OUT1

Dopo qualche secondo, il relé CH1 commuta e voi ricevete ancora l'SMS a confermare l'avvenuta commutazione. La procedura per accoppiare gli altri pulsanti ai canali è identica: ciò che varia è solamente l'identificativo di canale. ➤



*Connessione
tra telecomando
e modulo
VMB1RS.*

Fig. 5

Il contenitore plastico

Il PCB del telecomando GSM è stato progettato per poter essere inserito facilmente in un contenitore plastico modello TEK0 660.5, lo stesso tipo di contenitore utilizzato per i progetti Velbus già pubblicati. In questo caso però, date le maggiori dimensioni del circuito (determinate soprattutto dal modem GSM), non è stato possibile racchiudere nel contenitore anche la scheda VMB1RS che, come visibile in Figura 5, fuoriesce completamente da un'apertura che dovete praticare

sul fianco destro. In Figura 6 trovate le indicazioni necessarie per poter praticare tutti i fori e le cave. Sul coperchio devono essere realizzati 3 fori da 3,5 mm di diametro, dai quali sporgono le teste dei tre led da 3 mm. Sul fianco sinistro va eseguito un foro del diametro di 7 mm dal quale far uscire (attraverso un passacavo in gomma morbida) il cavo quadripolare necessario per il collegamento del telecomando alla rete Velbus. Sul fianco destro, invece, devono essere praticati un foro di diametro 7 mm da cui esce (sempre attra-

verso un passacavo) il cavo quadripolare che collega la scheda VMB1RS al CAN Bus passando attraverso la scheda madre (vedere figura 5) nonché la finestra rettangolare, necessaria per portare all'esterno il connettore seriale DB9. Nel praticare i fori, usate punte di diametro progressivo e fate girare lentamente il mandrino del trapano. La finestra può essere ricavata con un taglierino (oppure con un seghetto a lama fine) e rifinita con una limetta a taglio dolce. Terminata la preparazione meccanica del contenitore,

posizionate la scheda come visibile in Figura 5 e fissatela con quattro viti autofilettanti da 3 mm x 6 mm di lunghezza. Notate che la scheda VMB1RS va innestata e collegata solo dopo aver racchiuso il circuito.

Conclusioni

Con questo articolo abbiamo concluso la serie dei progetti pratici dedicati ai sistemi domotici Velbus di Velleman. Ciò, però, non significa che abbiamo esaurito il nostro compito: la domotica è un argomento in cui abbiamo sempre creduto e per il quale non vogliamo smettere di dedicare spazio e risorse. Continuate quindi a seguirci perché abbiamo in cantiere molte altre utili ed interessanti applicazioni con gli apparati Velbus.

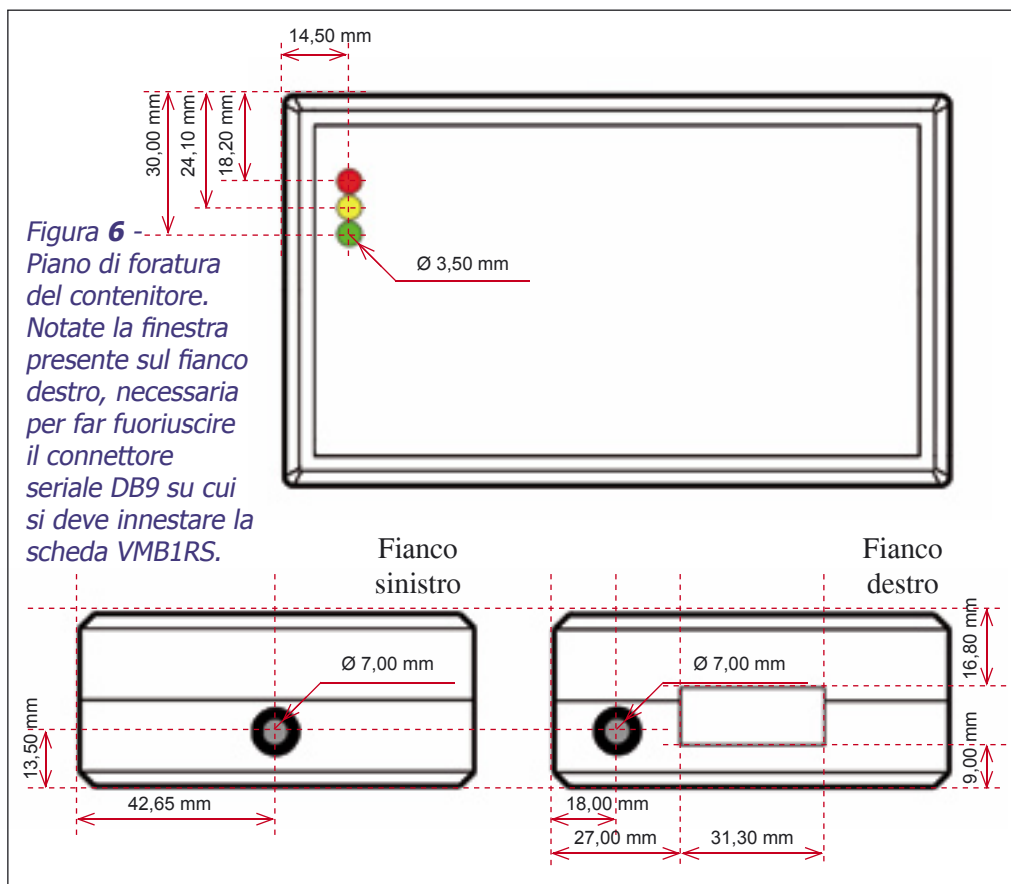


Figura 6 - Piano di foratura del contenitore. Note la finestra presente sul fianco destro, necessaria per far fuoriuscire il connettore seriale DB9 su cui si deve innestare la scheda VMB1RS.

per il

MATERIALE

Tutti i componenti utilizzati in questo progetto sono facilmente reperibili in commercio. Il master del circuito stampato può essere scaricato gratuitamente dal sito della rivista (www.elettronica-in.it) così come il firmware col quale programmare il micro (MF704.hex). Il microcontrollore è disponibile anche già programmato al costo di 18,00 Euro (cod. MF704). Tutti i prezzi si intendono IVA compresa.

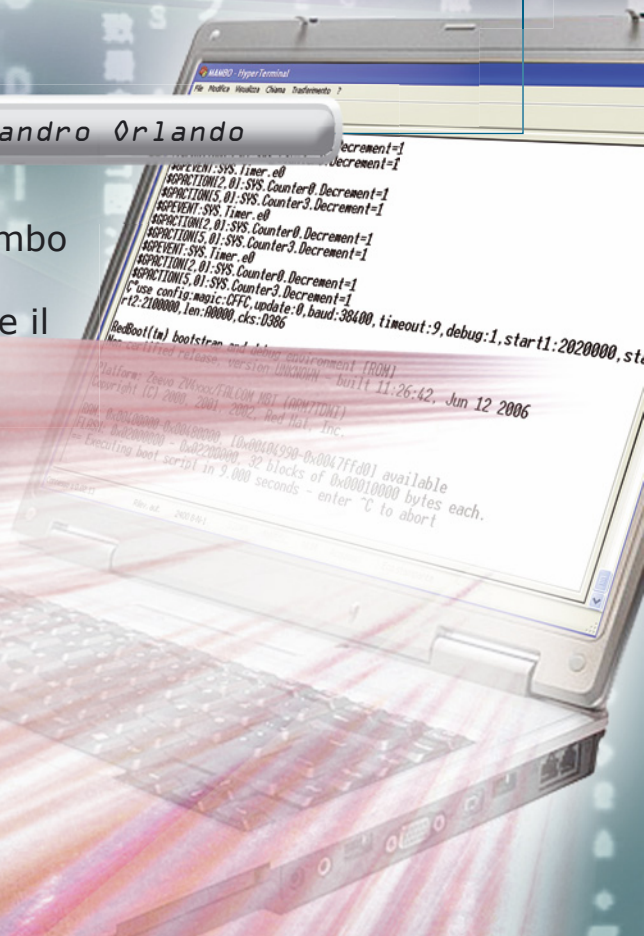
Il materiale va richiesto a: Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)

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

Aggiorniamo il firmware del Mambo

di Alessandro Orlando

Analizziamo in dettaglio la procedura per aggiornare il firmware del localizzatore Mambo che viene effettuata tramite qualsiasi applicativo di emulazione di terminale come il comune HyperTerminal di Windows.



Nelle note conclusive del precedente articolo, pubblicato sulla rivista 119, avevamo anticipato che Falcom, l'azienda che produce il localizzatore Mambo, aveva appena rilasciato una nuova versione del firmware. Nel corso di tale articolo non abbiamo volutamente commentato il nuovo firmware perché volevamo provarlo prima di pubblicare qualsiasi informazione a riguardo. Bene: dopo averlo studiato e provato adeguatamente, è giunto il momento di effettuare l'upgrade del localizzatore. Per aggiornare il dispositivo con il nuovo firmware, esistono teoricamente due vie: il Workbench e l'emulazio-

ne di terminale. La prima consiste nell'uso dell'opzione firmware Uploader, presente nel Workbench (a titolo puramente informativo, vi ricordiamo che tutti i software di gestione nonché i manuali ed i file di aggiornamento sono contenuti nel CD allegato al prodotto); la seconda invece, viene messa in pratica facendo uso del classico HyperTerminal, l'emulatore di terminale presente in ogni versione di Windows. Attingendo al nostro bagaglio di esperienze eravamo sicuri che non avremmo avuto problema alcuno nell'utilizzo del Workbench, mentre HyperTerminal ci avrebbe fatto tribolare, ma di fatto le cose sono andate >

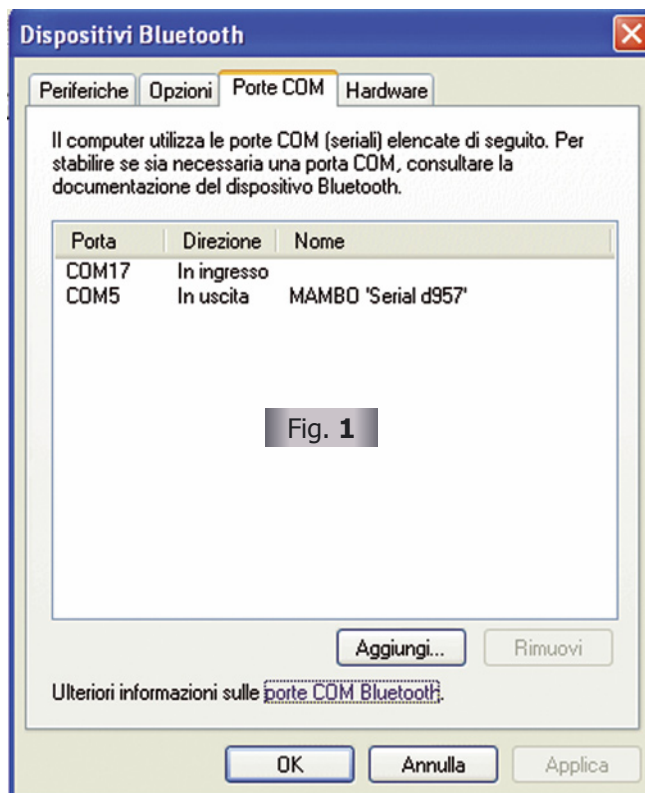


Fig. 1

diversamente da quanto pensavamo. Con il Workbench abbiamo effettuato molteplici tentativi di aggiornamento della flash, ma abbiamo riscontrato una certa difficoltà nel porre il Mambo in modalità di RedBoot (modalità necessaria per effettuare l'aggiornamento firmware) quindi, a fronte dei risultati ottenuti, consigliamo di usare la versione attuale di Workbench (sicuramente Falcom interverrà in seguito, apportando le

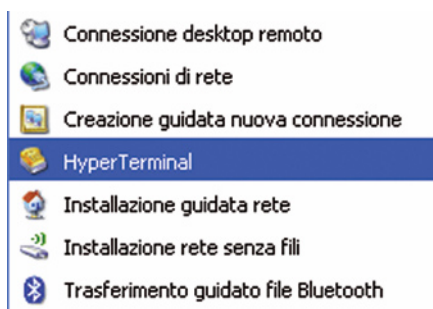
di Mambo è abbastanza laboriosa: ebbene anche l'aggiornamento del firmware tramite terminale non è da meno ma con il vantaggio, tuttavia, di funzionare sempre senza problemi se le istruzioni vengono seguite alla lettera.

Il firmware ufficiale

Le versioni rilasciate sino ad oggi da Falcom sono tre e precisamente la 1.0.1 (versione iniziale), la 1.0.2

dovute correzioni in una nuova versione) mentre consigliamo la seconda soluzione, che fa uso di HyperTerminal. Sappiate che qualsiasi programma di emulazione di terminale è adatto allo scopo: noi abbiamo scelto di utilizzare HyperTerminal perché usato anche da Falcom e perché presente su qualsiasi PC con sistema operativo Windows. Dagli articoli precedenti avete compreso che ogni procedura di configurazione e programmazione

Fig. 2



(che implementa la sostituzione degli indirizzi della Flash Memory) ed infine la 1.0.3 rilasciata in data 04/07/2007, per l'appunto durante la stesura dell'articolo precedente, dove sono state introdotte sostanziali modifiche al linguaggio di programmazione del Mambo. A tal proposito il set di istruzioni riconosciute dal localizzatore può essere definito di "basso livello" e "stand-alone": basso livello perché le istruzioni risultano stringate ed essenziali, e non contengono le classiche locuzioni logiche utilizzate, per esempio, dal BASIC; "stand-alone" perché si tratta di un linguaggio scritto appositamente e che pertanto non può essere compreso (leggasi "interpretato ed eseguito") da altri dispositivi presenti sul mercato.

Configurazione con HyperTerminal

Il punto di partenza della procedura è rappresentato dall'aggiunta del dispositivo all'elenco degli apparati Bluetooth noti al PC che siamo soliti utilizzare, aggiunta che viene effettuata eseguendo la ricerca del localizzatore e il relativo *pairing* (eventualmente fate riferimento a quanto già pubblicato sul fascicolo 117). Dopo aver aggiunto il nuovo dispositivo Bluetooth e identificato la porta che gli è stata assegnata da Windows (Figura 1), si avvia HyperTerminal (Figura 2): a questo punto appare la finestra con cui si procede alla generazione di una "Nuova connessione" (Figura 3), connes-

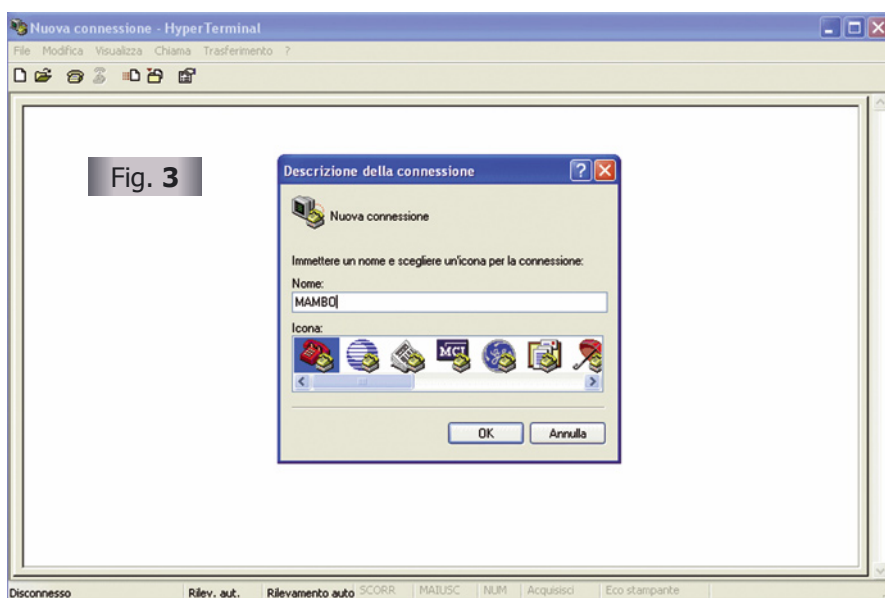


Fig. 3

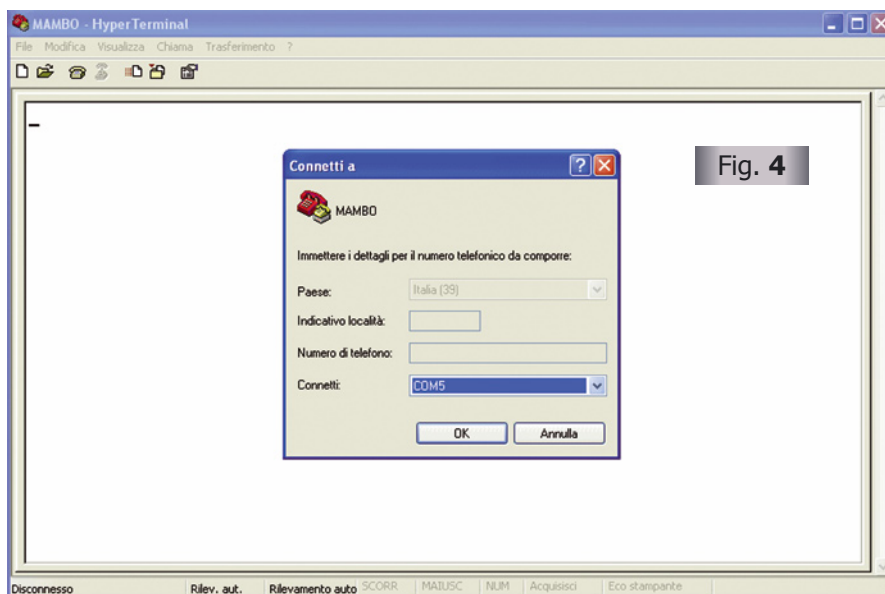


Fig. 4

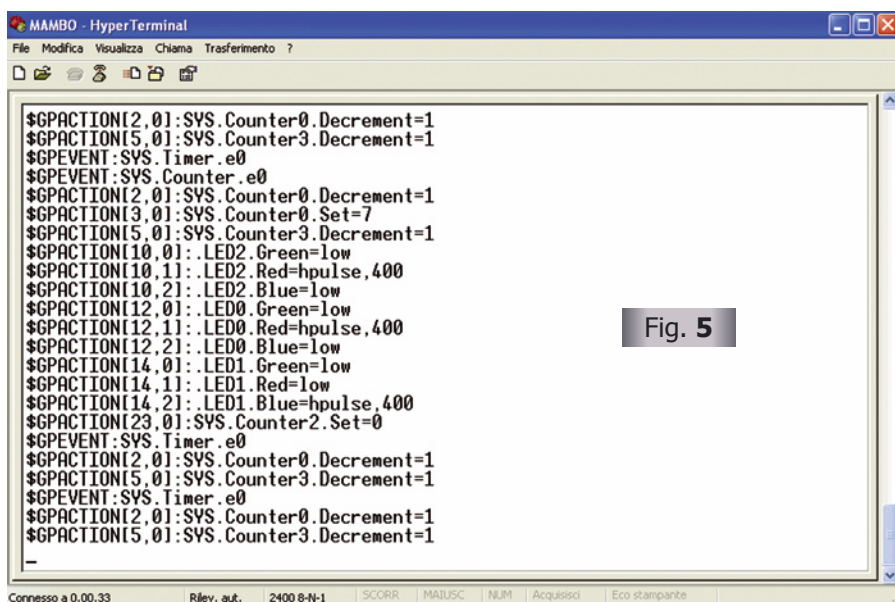


Fig. 5

ne a cui diamo per esempio il nome “Mambo” molto facile da ricordare anche per gli aggiornamenti futuri. Ora l'applicativo fa richiesta della porta COM su cui aprire la connessione (Figura 4): l'identificativo della seriale è quello visto nella Figura 1. Una volta inserita la ComPort e confermato il comando, si apre una nuova finestra in cui scorre il flusso di dati generato dall'esecuzione della “configurazione utente” (leggasi anche “programma”) presente attualmente in memoria (Figura 5).

Il passo successivo consiste nel configurare le opzioni ASCII del terminale: dal menu “File-Proprietà” di HyperTerminal si richiama la finestra delle “Impostazioni”, che va configurata come in Figura 6. Quindi si clicca sul pulsante “Impostazioni ASCII” per caricare la finestra visibile in Figura 7, sulla quale va selezionata la casella “Aggiungi avanzamento riga ad ogni ritorno a capo inviato” (prima casella di spunta).

Modalità RedBoot

Per effettuare l'aggiornamento, il dispositivo deve essere posto in modalità *RedBoot* (questo nome è stato scelto da chi ha scritto il firmware!) ossia quel particolare stato operativo che consente al Mambo di ricevere e ➤

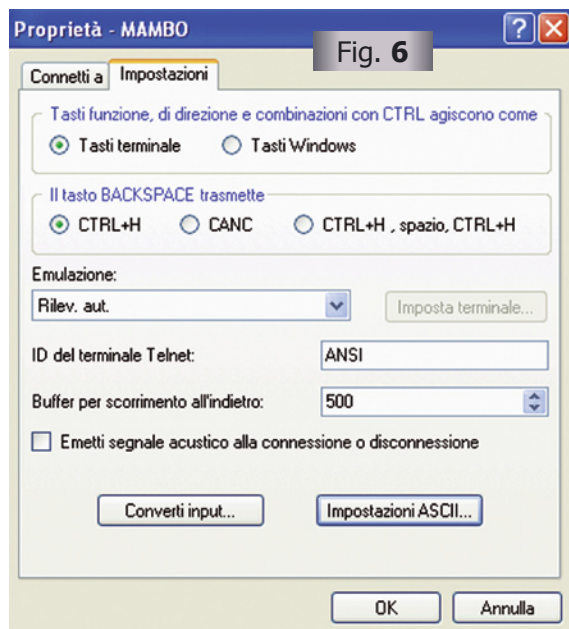


Fig. 6



Fig. 7

memorizzare un file Binario. Per prima cosa, si deve preparare e salvare un file di testo (ad esempio *reset.txt*) contenente l'istruzione

\$PFAL,Sys.ZV.Reset<CR-LF>

in cui i caratteri terminatori di stringa CR-LF sono ottenuti premendo il tasto di invio "ENTER" (Figura 8).

Ora è necessario inviare al Mambo il file *reset.txt* appena creato: per fare ciò si utilizza l'opzione di HyperTerminal "Trasferimento - Invia File di testo". Dalla finestra che appare selezionate il file *reset.txt* e confermate l'invio facendo clic sul tasto "Apri".

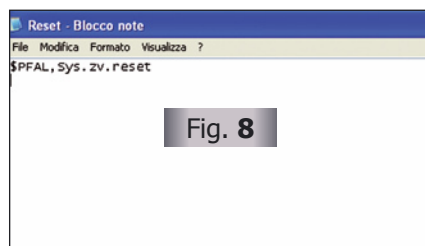


Fig. 8

Nella finestra principale di HyperTerminal appare una serie di dati a segnalare che si è entrati in modalità RedBoot. L'ultima riga comunica che si hanno a disposizione 9 secondi per premere i tasti CTRL+C e terminare (se si vuole) la procedura (Figura 9).

Aggiornamento del firmware

Quanto visto fino ad ora serve esclusivamente per predisporre il Mambo a ricevere l'aggiornamento.

Per inviare il file binario vero e proprio (il nuovo firmware) occorre innanzitutto stabilire l'indirizzo di partenza ossia l'indirizzo fisico di memoria da cui il dispositivo deve iniziare a salvare i byte ricevuti: per fare ciò si deve digitare il comando

fload -f 0X2020000 -r <ENTER>

sulla finestra principale di HyperTerminal (Figura 10); il Mambo risponde avvertendo di inviare il file facendo uso del protocollo XModem,

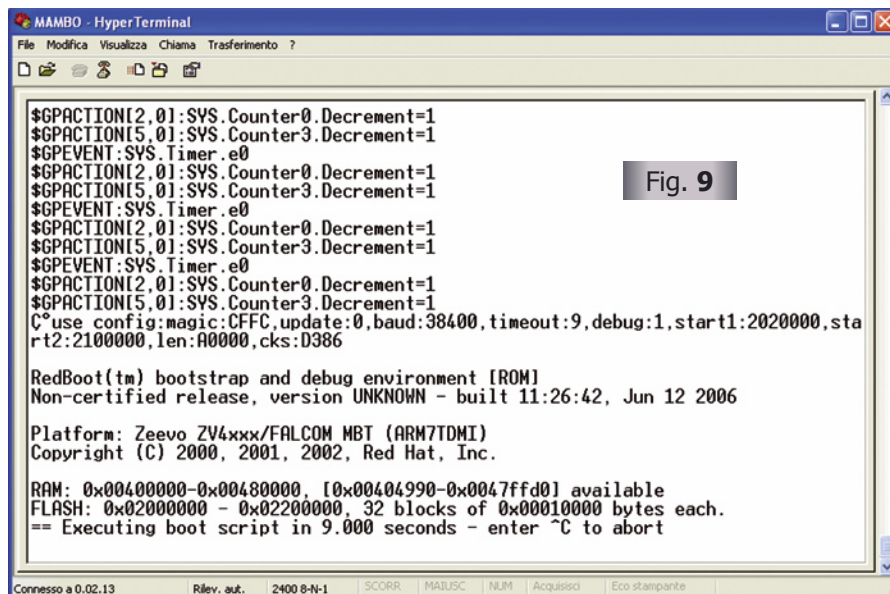


Fig. 9

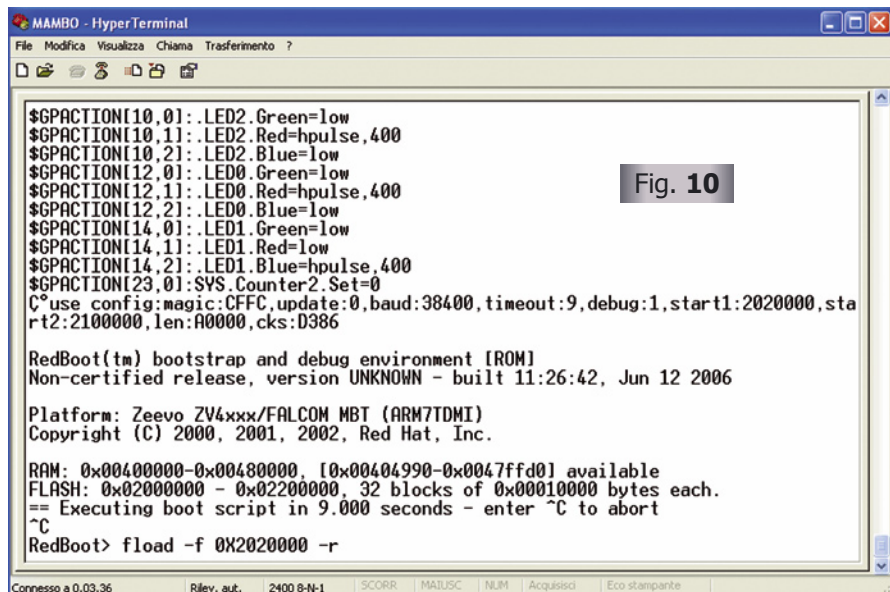


Fig. 10

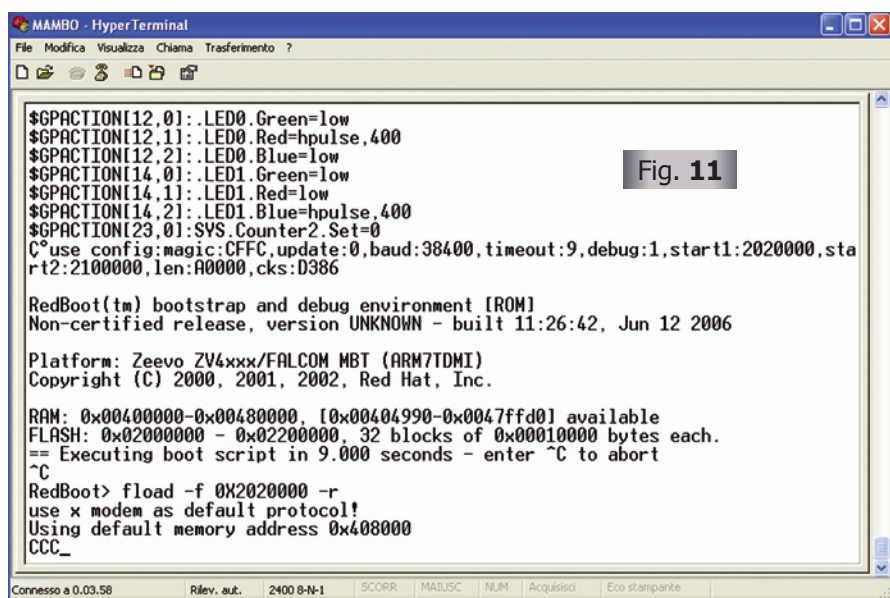


Fig. 11

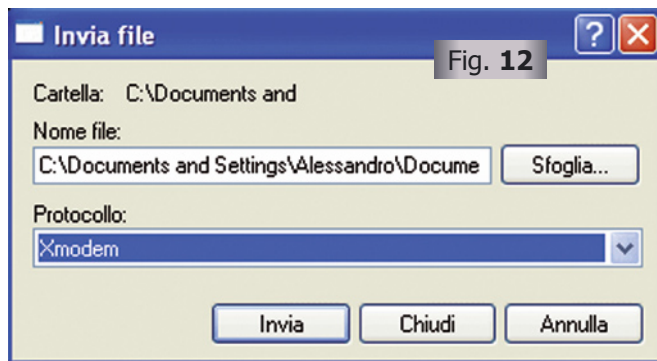


Fig. 12

come visibile in Figura 11. Ora, tramite l'opzione "Trasferimento - Invia File", si carica la finestra visibile in Figura 12, attraverso la quale potete scegliere il file da inviare (nel nostro caso, questi è nominato "Mambo_2.4.3_final") e il protocollo XModem, selezionabile tramite l'apposito menu a tendina. Ora il Mambo è in fase di aggiornamento (Figura 13) al termine della quale occorre digitare il comando

Reset <ENTER>

per effettuare un riavvio completo. In alternativa è possibile scollegare fisicamente la batteria per qualche istante, al fine di essere certi che il reset hardware abbia effetto. Una volta aggiornato il prodotto, le istruzioni usate fino ad ora non sono più valide in quanto il set di comandi utilizzato nel nuovo firmware è differente e quindi, se per qualsiasi motivo doveste effettuare un downgrade (ossia ritornare alla versione precedente) il file di testo *reset.txt* dovrà contenere l'istruzione "\$PFAL,Sys.

Device.Reset" mentre per iniziare l'aggiornamento si dovrà digitare (sempre entro 9 secondi) solo l'istruzione "flood <ENTER>": gli indirizzi verranno definiti automaticamente e si potrà

scegliere indifferentemente tra protocollo *1K XModem* (più veloce e pertanto consigliato dalla stessa Falcom) e XModem.

Il nuovo firmware

Rispetto alla revisione precedente, quella nuova presenta delle differenze sostanziali in quanto sono state aggiunte nuove importanti funzioni.

Per esempio sono stati inseriti il protocollo 3D per il GPS, la funzione "Spy-Call" (che consente di silenziare completamente il dispositivo durante un ascolto ambientale), le possibilità di scaricare i dati GPS scritti nel registro "storico" effettuando una sola lettura e di inviare delle e-mail attraverso il protocollo SMTP e molte altre ancora. Sfortunatamente, i file di configurazione preparati per il firmware precedente non sono più validi, quindi vanno riscritti seguendo le procedure definite nel manuale in Inglese. In un prossimo articolo vedremo come utilizzare le nuove funzioni, compresa quella che consente di inviare i dati GPS attraverso le e-mail, con conseguente notevole risparmio sui costi di gestione telefonica.

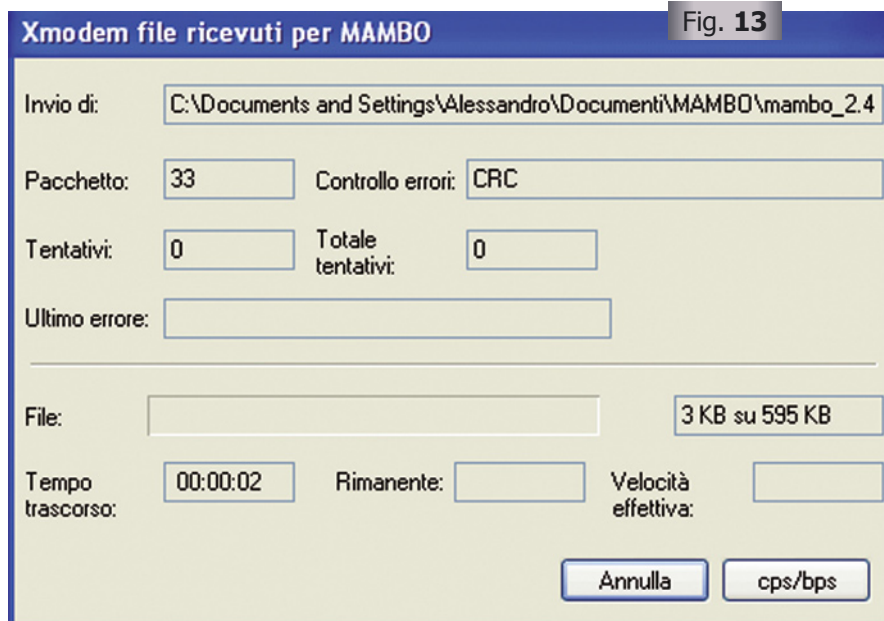


Fig. 13

per il

MATERIALE

Il sistema di localizzazione satellitare GPS/GSM/GPRS Bluetooth Mambo (cod. Mambo55) può essere acquistato al prezzo di 510,00 Euro, IVA inclusa. La confezione include, oltre al localizzatore, il pacco batterie, un auricolare, un caricabatterie con adattatore sia di rete che da accendisigari per auto, un laccio per il trasporto al collo ed il CD contenente i manuali in inglese e l'ultima versione di firmware disponibile al momento dell'acquisto.

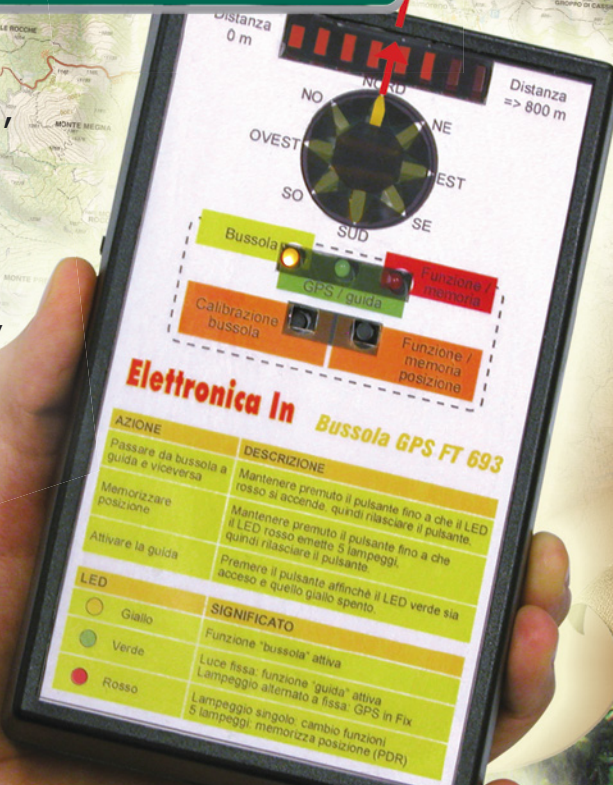
Il materiale va richiesto a: Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)

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

Localizzatore con bussola

di Giona Calzone

Pensato per gli amanti delle passeggiate nei boschi, permette di memorizzare il punto di partenza e di ritrovarlo facilmente, grazie a un display a led che, sulla strada del ritorno, indica distanza e direzione da seguire.



Quando ci si avventura in luoghi nei quali, per struttura o estensione, diventa difficile orientarsi e ritrovare il punto di partenza, non è affatto infrequente perdersi e mettersi nei guai; infatti, a meno di non avere una memoria fotografica e uno spiccato senso dell'orientamento, trovare la strada del ritorno è sovente impresa assai ardua. A parte il mitologico labirinto di Dedalo, i posti dove è più facile perdersi sono indubbiamente boschi e foreste, perché la densità e l'uniformità della vegetazione rende tutt'altro che facile la percezione di punti di riferimento o del luogo di partenza; perciò, per evitare di smarrire la strada del rientro

conviene dotarsi di una bussola e annotare, prima di partire, l'indicazione che fornisce. Ma la bussola non è l'unico strumento; sebbene per secoli lo sia stato e abbia segnato la rotta dei grandi esploratori, oggi abbiamo a disposizione una tecnologia capace di andare ben oltre le sue possibilità: quella del GPS. E che ce ne facciamo di una costellazione di satelliti (anzi due, con il GLONASS, quasi tre con il nascente Galileo...) se non la sfruttiamo per orientarci? Proprio pensando a chi si avventura in escursioni e passeggiate, ma anche a quelli che, in autunno, vanno in cerca di funghi, abbiamo deciso di realizzare un localizzatore, però



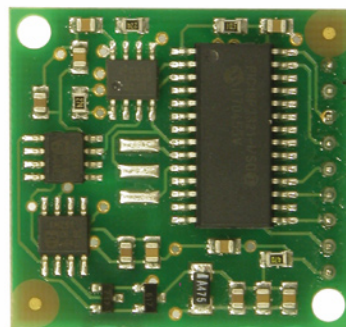
Il modulo bussola

La tradizionale bussola è uno strumento conosciuto fin dall'antichità per la sua proprietà di indicare la collocazione rispetto ai quattro punti cardinali; il suo funzionamento si basa sul fatto che un ago o asta la cui punta sia stata magnetizzata (ad esempio per sfregamento su una calamita artificiale) tende a disporsi verso il Polo Nord del nostro pianeta e ciò perché la Terra è effettivamente un grosso magnete, caratterizzato da un naturale magnetismo dovuto a un campo che ha il Polo Nord all'incirca in corrispondenza del Polo Nord geografico e il Sud in prossimità del Polo Sud. L'effetto del magnetismo si traduce in una leggera forza esercitata su corpi aventi una propria polarizzazione magnetica; un modo per vederlo è, appunto, magnetizzare la punta di un ago metallico appoggiato su un sughero galleggiante in una bacinella d'acqua: in tali condizioni la forza del campo terrestre è sufficiente a muovere l'ago fino a orientarne la punta verso il Nord. Sfruttando il fenomeno si possono costruire non solo le bussole meccaniche ma anche, la tecnica ce l'ha dimostrato, quelle elettroniche allo stato solido; sappiamo infatti che la corrente elettrica che attraversa, ad esempio, un semiconduttore immerso in un campo magnetico determina una differenza di potenziale in senso trasversale, la cui polarità e ampiezza sono funzione dell'intensità del campo stesso. Questo è noto come effetto di Hall.

I sensori Philips montati nel modulo CMPS03 (Figura A1) da noi usati sfruttano proprio tale effetto e un microcontrollore provvede a interpretarne le correnti e a ricavare un'informazione che viene resa disponibile sia in formato analogico che digitale. Nel primo caso, l'angolazione rispetto alla direzione del Nord viene indicata con un'uscita PWM, la quale genera un'onda rettangolare il cui duty-cycle viene fatto variare in maniera direttamente proporzionale all'angolo che i sensori della bussola rilevano rispetto alla linea che porta al Nord Terrestre; per l'esattezza, il segnale di uscita del piedino 4 è composto da impulsi intervallati da pause fisse di 65 millisecondi ciascuna e ciò che varia è la durata degli impulsi stessi. Il campo di variazione è compreso tra un minimo di 1 ms quando la bussola è puntata verso il Nord della Terra e un massimo di 36,99 millisecondi se è ruotata di un angolo pari a $359,9^\circ$. In pratica il coefficiente di variazione è di $100 \mu s/^\circ$ di rotazione e c'è un offset di 1 millisecondo, dato che quando la bussola punta esattamente il Nord gli impulsi sono comunque larghi 1 ms.

Quanto all'uscita digitale, è del tipo I²C-bus e, quando il modulo viene interrogato, fornisce numerose informazioni contenute, ciascuna, in un apposito registro; l'indirizzo del modulo è fisso e pari a 0.

Lato corrispondente al NORD



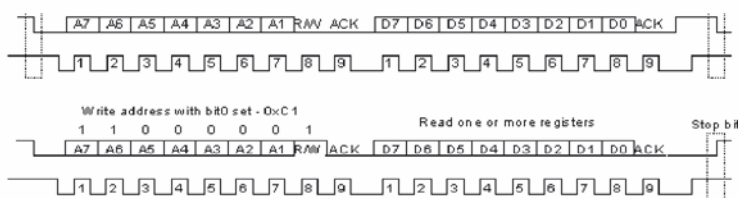
- Pin 9: massa (0 V).
- Pin 8: non collegato.
- Pin 7: 50/60 Hz.
- Pin 6: calibrazione.
- Pin 5: non collegato.
- Pin 4: uscita PWM.
- Pin 3: SDA.
- Pin 2: SCL.
- Pin 1: Vcc +5 V.

Fig. A1

Il protocollo di comunicazione (visibile in Fig. A2) è quello tipico di un dispositivo *Slave* dietro interrogazione del dispositivo *Master*. Quest'ultimo inizia la sessione con un bit di start, quindi con un byte contenente l'indirizzo I²C e un altro indicante il registro da leggere (0 contiene la versione del software del microcontrollore che governa la bussola, 1 la rotazione, espressa in forma binaria con un valore tra 0 e 255, 2 e 3 la rotazione espressa in forma binaria ma con valori da 0 a 3.599 ecc.); il modulo risponde inviando sul filo SDA lo stato del registro richiesto. Quando si utilizza la sola uscita PWM, per il buon funzionamento dell'insieme le linee SDA ed SCL dell'I²C-bus devono essere portate a livello alto mediante resistori di pull-up da 22÷100 kohm.

Una particolare funzione del CMPS03 è la compensazione dell'interferenza prodotta dalla rete di distribuzione elettrica, attivabile mediante il piedino 7: internamente, è previsto un circuito in grado di minimizzare l'effetto che il campo magnetico prodotto dalle linee elettriche induce nei due sensori Philips; per funzionare correttamente, esso deve conoscere la frequenza della rete. L'informazione del caso si fornisce, appunto, dal piedino 7, il quale deve stare a zero logico se si è in un Paese in cui le linee elettriche sono a 50 Hz (Europa, Australia ecc.) e ad 1 quando si opera dove la rete è a 60 Hz (USA, Giappone, Centro e Sud America, ad esempio). Nel nostro apparecchio non usiamo la funzione (pin 7 scollegato) perché si presume che venga usato all'aperto, dove non dovrebbe esserci interferenza della rete... a meno di non camminare sotto i cavi dell'alta tensione...

Fig. A2



un po' diverso dal solito: si tratta, infatti, di un apparecchio ideato per indicare la direzione verso casa, o, meglio, per tornare al punto di partenza, quale può essere quello dove si lascia l'automobile per proseguire a piedi. Tipicamente, quando si scende dall'auto per addentrarsi nel bosco, si memorizza la

propria posizione; nel momento in cui si decide di tornare indietro, si preme un apposito pulsante e l'apparato indica la direzione da seguire mediante un visualizzatore a led che raffigura il quadrante di una bussola e una barra indicante la distanza approssimativa. Sicuro e facile da usare, è una sorta di

navigatore satellitare fatto su misura di coloro che, per passione o professione, si avventurano in luoghi sconosciuti, come ad esempio i cercatori di funghi, che presi dalla foga di trovare porcini, prataioli e chiodini, spesso dimenticano di stabilire dei punti di riferimento per trovare la strada del ritorno.

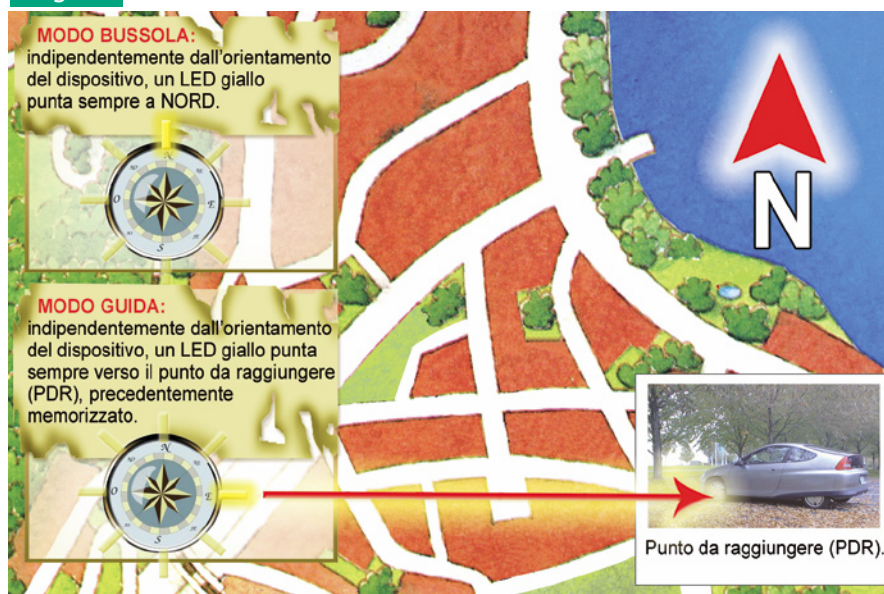
Come funziona

Il dispositivo permette in ogni momento di determinare la direzione del Nord (modo *bussola*), ma anche di memorizzare un punto, detto di *partenza* o *da raggiungere* (PDR) e, all'occorrenza, di calcolare e tracciare la via del ritorno ad esso (modo *guida*, vedere figura 1). Normalmente si usa così: prima di iniziare l'escursione, si memorizza la propria posizione (PDR) premendo un pulsante quindi si può andare in giro tranquilli a fare ciò che si deve, praticamente senza limiti. Quando si decide di rientrare, con un secondo comando si chiede all'apparecchio come fare a tornare dal punto in cui si trova (PR) a quello di partenza; mediante il visualizzatore a led, esso indica la direzione da seguire e la distanza residua. Ecco, questo è il funzionamento di massima.

A questo punto, tra i nostri lettori che un po' conoscono il GPS, qualcuno si sarà certo già chiesto che senso ha progettare il dispositivo descritto in queste pagine, quando ormai in tutti i negozi di elettronica di consumo si trovano navigatori satellitari portatili, palmari, a prezzi più che abbordabili; ebbene, un senso c'è, eccome: se è vero che un qualsiasi navigatore GPS può, decisa la meta, guidarci fino a raggiungerla, esistono problematiche (quasi sempre trascurabili in auto) che lo rendono praticamente inutilizzabile per guidare una persona a piedi.

Per comprendere ciò e capire come mai abbiamo ritenuto di realizzare il nostro dispositivo, pensate che tutti i navigatori sono progettati per gli autoveicoli, tipicamente veloci, mentre il nostro è stato ideato per chi si muove a piedi, quindi molto lentamente. Cerchiamo ora di chiarire perché un GPS non può essere utilizzato per stabilire una direzione quando lo spostamento è lento, facendo una premessa: il ricevitore può calcolare la posizione in cui si trova quando conosce la distanza da almeno tre satelliti della costellazione; le distanze sono calcolate eseguendo delle operazioni molto

Fig. 1



complesse sui segnali trasmessi dai satelliti, segnali che, pur iniziando nello stesso istante, raggiungono il ricevitore con ritardi differenti.

Bene: il GPS calcola la posizione e la velocità proprio misurando questi ritardi, che, purtroppo, sono influenzati da tanti parametri quali condi-

zioni atmosferiche, effetto delle macchie solari, precisione del segnale di clock del GPS, riflessioni e così via. Quindi se riportassimo le varie posizioni (una al secondo) calcolate da un ricevitore immobile su di un diagramma circolare del diametro di tre metri come quello di Figura 2, noteremmo ➤

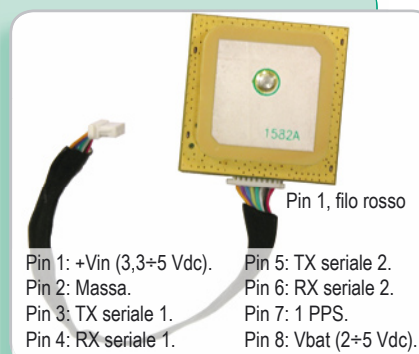
Il ricevitore GPS

Per realizzare il nostro sistema di navigazione GPS per "camminatori", ci siamo avvalsi di uno dei ricevitori GPS introdotti recentemente sul mercato: si tratta dell'FV-M8 della Sanav, un prestante modulo con antenna integrata amplificata, compatibile con i sistemi di correzione dell'errore di posizionamento più avanzati (DGPS, WAAS, EGNOS, MSAS) e capace, perciò, di garantire una precisione di addirittura $\pm 2,6$ metri. Le sue ottime performance sono dovute indubbiamente anche alla capacità di ricevere i segnali dei satelliti su ben 32 canali, quindi più dei satelliti della rete GPS stessa; merito anche dell'elevatissima sensibilità, che ammonta a -158 dBm, aiutati dall'elevato guadagno in ricezione (ben 21 dB) dell'antenna integrata.

Il modulo è gestito da una potente CPU che permette, a regime, acquisizioni di posizione anche cinque volte al secondo; il tempo necessario a determinare la posizione è inferiore al secondo a "caldo" (quando il ricevitore sta ricevendo i segnali della rete satellitare) e limitato in 41 secondi a "freddo" (ossia quando viene alimentato e deve cercarsi i satelliti).

L'interfaccia di comunicazione è una seriale UART programmabile, impostata in partenza a 4.800 bps con protocollo NMEA (lo standard compatibile con tutte le interfacce per ricevitori GPS) ma configurabile per raggiungere velocità di comunicazione fino a 115,2 kbps; i livelli sono TTL-compatibili.

Il protocollo di comunicazione è l'NMEA V 3.01 e le sentenze di uscita supportate sono: RMC, GGA, GSV, VTG, GSA.



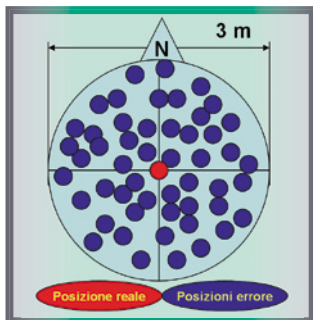


Fig. 2 - GPS fermo: le posizioni calcolate distano dal punto reale perché affette da un intrinseco errore di sistema.

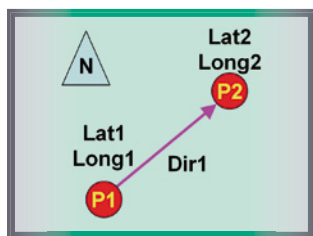


Fig. 3 - Noti il Nord e due punti, si può calcolare una direzione.

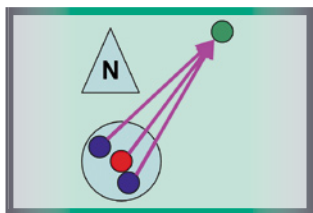


Fig. 4 - Se i due punti sono lontani tra loro, l'errore sulla direzione è minimo.

tre particolari molto importanti:

- la posizione calcolata è vicina a quella reale, ma difficilmente uguale;
- ogni posizione calcolata differisce anche di svariati metri dalla precedente;
- non esiste legame tra due calcoli contigui, cioè non è possibile predire la posizione successiva nota la precedente.

Tipicamente, un buon GPS in condizione statica (fermo in un posto preciso) restituisce a cadenza di un secondo delle posizioni che nel 90% dei casi distano non oltre 5 metri dal punto reale (centro figura), ma che ruotano intorno ad esso in modo casuale (in funzione degli inevitabili errori di calcolo).

Vediamo ora il passo successivo: per calcolare una direzione geografica,

oltre al Nord, occorrono un punto di partenza e uno di arrivo e, maggiore è la distanza tra i due punti, maggiore è la precisione della direzione calcolata (Figura 3). Ipotizziamo quindi di voler calcolare la direzione di un GPS che si sta muovendo a 90 km orari, corrispondenti a 25 m/s (Figura 4), e che presenta un errore tipico di 3 metri (si tratta di un ottimo GPS!). A 90 km/h, due letture di posizione intervallate di un secondo generano due posizioni distanti tra loro da 22 a 28 metri quindi, note le loro latitudini e longitudini, un sistema a microcontrollore esterno al GPS è in grado di ricavare un valore numerico che rappresenta la deviazione rispetto al Nord geografico, ossia la direzione in cui il GPS si muove, oltre che la velocità. Sicuramente, tale calcolo è impreciso (25 metri sono poca cosa nei confronti dei 20.000 km di quota dei satelliti), ma utilizzabile per generare una rappresentazione grafica: notate che nella Figura 4, le due direzioni calcolate sulla base di un punto di partenza affetto da errore (blu) non sono molto differenti dalla direzione calcolata sul punto di partenza reale (rosso).

Vediamo ora cosa succede quando il GPS non è in movimento, aiutandoci con la Figura 5. P1, P2, P3, Pn... sono posizioni restituite dal GPS, distanti un secondo l'una dall'altra, contenute in un'area circolare di 6 metri di diametro, che possono essere interpretate come spostamenti alla velocità massima di 3 m/s rispetto alla posizione reale: tutto ciò rispecchia al 100% l'errore del nostro GPS di prova. Gli inevitabili errori di calcolo generano dati che, quando usati da un micro, lo inducono a credere che il GPS si stia spostando di poco e lentamente. Calcoliamo ora qualche direzione, esattamente come farebbe il micro: P1>P2 corrisponde a EST, P2>P3 SUD-OVEST, P3>P4 NORD OVEST e così via: tre cambi di direzione in tre secondi! ...e il GPS è immobile! Da quanto appena spiegato, si deduce che la direzione calcolata tra due punti usando i soli dati forniti

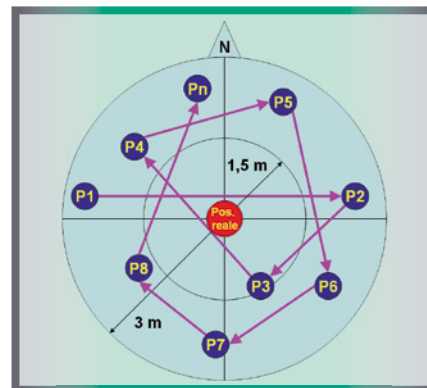


Fig. 5 - A bassa velocità, la direzione elaborata varia notevolmente in funzione del punto utilizzato per il calcolo.

dal GPS diventa affidabile quando le coordinate dei due punti sono distanti molto di più del massimo errore del GPS stesso. E ciò è assolutamente normale. Ora, una persona che cammina alla velocità di 1 m/s (3,6 km/h) si muove più lentamente rispetto allo stesso errore del GPS, che, come abbiamo già detto, può sbagliare di 3 metri ogni secondo: il risultato è che, pur camminando, la nuova posizione della persona è contenuta in quel cerchio di tre metri che deve essere considerato come errore. Per il GPS, quindi, possono accadere due fatti: la persona non si è spostata e il calcolo delle direzioni genera gli stessi risultati strani visibili nella Figura 5, oppure si è spostata di botto di qualche metro (valore ottenuto per somma di errori) con conseguente calcolo di una direzione non corretta. Di conseguenza se si volesse realizzare un indicatore di direzione in cui un micro legge i dati del GPS, calcola la direzione e aggiorna la posizione dell'ago, questo ago indicherebbe una

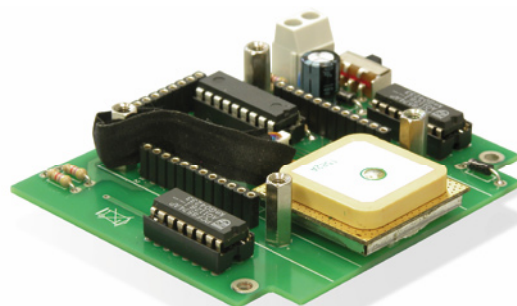
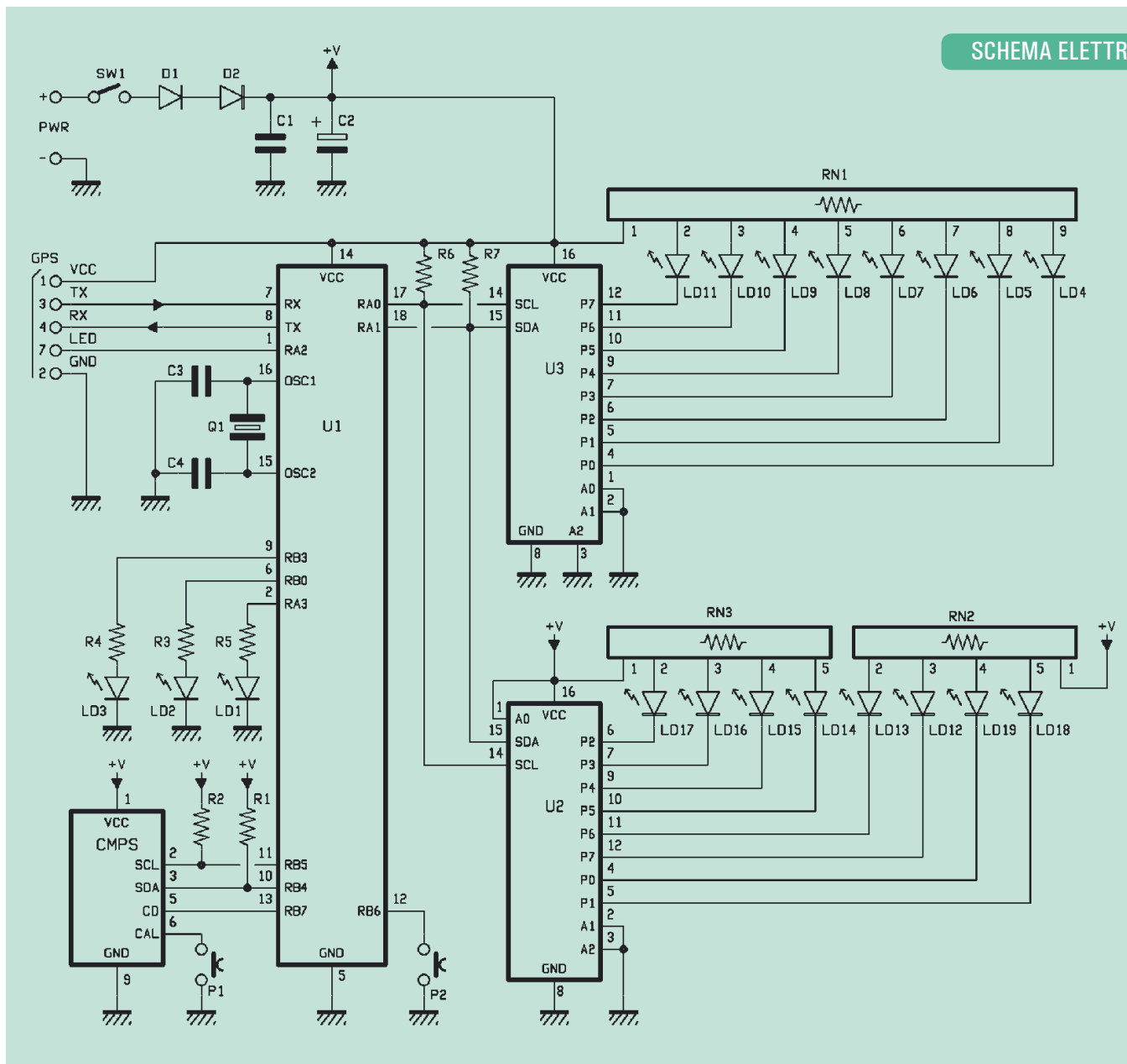


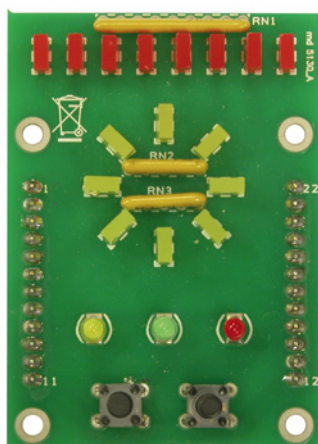
Fig. 6 - Il ricevitore GPS si trova sotto la scheda Led e viene fissato alla base con del nastro biadesivo.



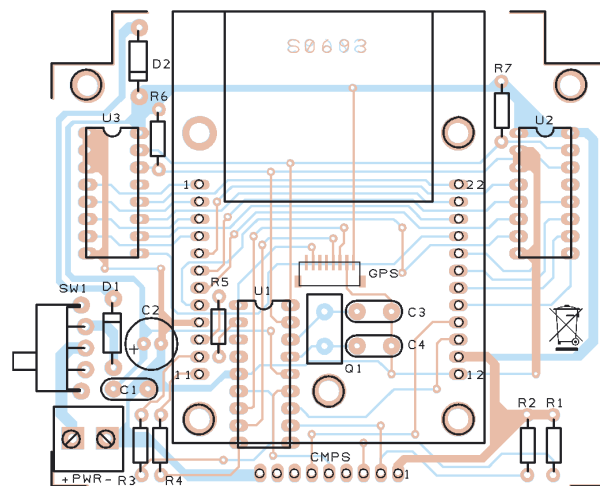
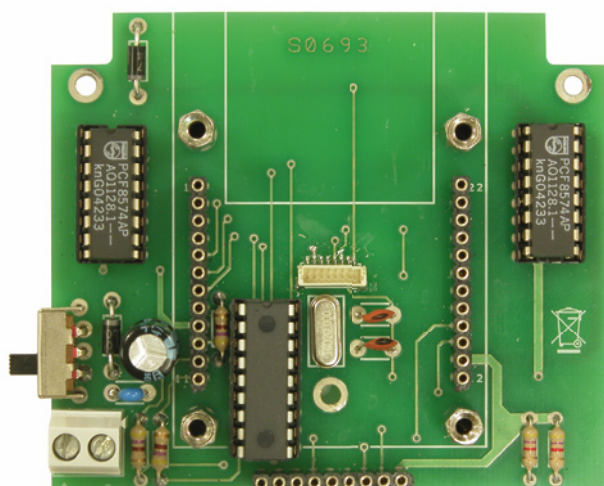
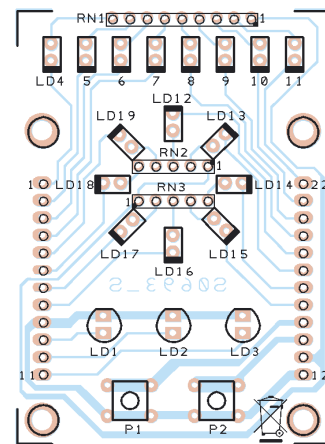
direzione affidabile solo con un movimento sufficientemente veloce, mentre stando fermo girerebbe all'impazzata. Il problema si risolve facilmente aiutando il GPS con una bussola elettronica. Nel nostro caso, il microcontrollore che gestisce l'insieme tiene conto delle informazioni ricavate dal ricevitore GPS per quel che concerne le posizioni iniziale (PDR) e di arresto (PR, quando gli chiediamo di aiutarci a tornare indietro) e il calcolo della distanza tra i due, ma per quanto riguarda la direzione da seguire si avvale dell'indicazione fornita da una speciale bussola elettronica, che, giustamente,

punta sempre a NORD. Ora, noti un punto cardinale e la direzione tra PDR e PR (parecchio lontani tra loro), calcolare e visualizzare una direzione da seguire diventa facile e il risultato è affidabile. Il nostro dispositivo è dotato di una barra composta da otto led rettangolari che indica la distanza tra il punto da raggiungere (PDR) e il punto rilevato (PR); ogni led rappresenta approssimativamente una distanza di 100 m, pertanto lo strumento è in grado di indicare distanze tra 0 e 800 m. Oltre, l'indicazione corrisponde sempre al massimo (tutti i diodi illuminati) perché supera il limite misurabile; quindi,

se ci si trova ad oltre 800 metri, quando ci si avvicina scendendo sotto tale distanza i led cominciano a spegnersi, a partire da quello corrispondente alla distanza maggiore. Altri otto led (sempre rettangolari, ma gialli) sono posti in cerchio e simulano il quadrante di una bussola; questo secondo display può essere usato per dare indicazioni nelle due modalità di funzionamento previste: *bussola* e *guida*. Nella prima, il led che si illumina indica il Nord e gli otto diodi corrispondono ai punti cardinali (*Nord*, *Sud*, *Est*, *Ovest*) e a quelli intermedi (*Nord-Est*, *Nord-Ovest*, *Sud-Est*, *Sud-Ovest*). Nella seconda ➤



Poiché i due dispositivi più delicati (ricevitore GPS e modulo bussola) vengono forniti già montati, il cablaggio dei due PCB non presenta particolari difficoltà. Occorre prestare però un minimo di attenzione durante il montaggio del connettore GPS, unico componente SMD e pertanto un po' delicato. Il ricevitore GPS FV-M8 non è saldato sul PCB ma viene collegato al connettore sulla base (GPS) tramite un cavetto di adattamento e fissato meccanicamente con del nastro biadesivo nello spazio centrale libero e identificato da un quadrato. Rispetto alle fotografie, l'antenna è posizionata verso l'alto.



ELENCO COMPONENTI:

R1,R2: 4,7 kohm
R3+R5: 470 ohm
R6, R7: 4,7 kohm
RN1: rete resistiva 8x470 ohm
RN2, RN3: rete resistiva 4x470 ohm
C1: 100 nF multistrato
C2: 470 µF 16 VL elettrolitico
C3, C4: 15 pF ceramico
Q1: quarzo 20 MHz basso profilo

U1: PIC16F648A (FT693)
U2, U3: PCF8574AP
CMPS: CMPS03
D1, D2: 1N4007
LD1: led 3 mm giallo
LD2: led 3 mm verde
LD3: led 3 mm rosso
LD4+LD11: led rettangolare rosso
LD12+LD19: led rettangolare giallo
P1, P2: Microswitch
SW1: Deviatore a slitta 90°

Varie:
- Morsettiere 2 poli
- Zoccolo 8+8 (2 pz)
- Zoccolo 9+9
- Strip femmina 11 poli (2 pz.)
- Strip femmina 9 poli
- Strip maschio 11 poli (2 pz.)
- Connettore maschio 8 poli SMT passo 1 mm
- Circuito stampato codice S693
- Circuito stampato codice S693-T

modalità, il led che si accende indica la direzione in cui si trova il punto di partenza, quindi la direzione da seguire, e viene aggiornata in tempo reale. Oltre al display distanza/direzione, il dispositivo monta tre led tondi (giallo, verde e rosso) che indicano la modalità impostata e servono da interfaccia utente: in modalità *bussola* viene acceso il led giallo (LD1) mentre in *ricerca punto di partenza* si accende il led verde (LD2). Per passare da una modalità all'altra è

sufficiente premere il pulsante P2 fino a quando si accende il led rosso LD3 e rilasciarlo subito dopo. Si noti che nella modalità *bussola* il dispositivo funziona esclusivamente come una tradizionale bussola ad ago in cui il display circolare indica la direzione del Nord. Invece, quando si attiva la modalità di ricerca del PDR, il led che si illumina indica la direzione in cui bisogna camminare per tornare al punto di partenza. Quest'ultimo può essere memorizzato in ogni momento; basta premere il

solito pulsante per almeno 3 secondi. Fatto ciò, il ricevitore GPS rimane sempre acceso e lo stato di *Fix* (localizzazione della posizione avvenuta correttamente) viene visualizzato dal led verde, che emette dei lampeggi quando il ricevitore stesso ha "agganciato" il segnale di più di tre satelliti. Dopo la memorizzazione della posizione, il dispositivo può essere portato in qualsiasi modalità, anche spento; tornando in ricerca, ricalcola la direzione e comincia a dare le pro-

prie indicazioni mediante il display distanza/direzione.

Schema elettrico

Visto come funziona il nostro localizzatore con bussola, diamo uno sguardo allo schema del circuito che lo realizza: l'insieme è abbastanza semplice, almeno a vedersi, perché le funzioni principali vengono svolte da due moduli: il ricevitore GPS e la bussola elettronica, che sono coordinati da un microcontrollore PIC16F648A, il quale provvede, tra l'altro, a gestire il visualizzatore a led mediante due I/O expander.

Il ricevitore GPS che abbiamo usato per questa applicazione, un FV-M8 prodotto dalla Sanav, si collega mediante un'interfaccia seriale a livello TTL e dispone di un'antenna integrata. È un prodotto qualitativamente molto valido, ad alta sensibilità, capace di ricevere ben 32 canali (quindi ben oltre i satelliti del sistema Navstar) oltre che di correggere autonomamente l'errore di posizionamento avvalendosi dei sistemi DGPS, WAAS, EGNOS, MSAS, con i quali è compatibile.

Il microcontrollore dialoga con esso mediante la propria UART, che fa capo alle linee RX (Piedino 7) e TX (pin 8); tramite l'I/O RA2, verifica lo stato dell'uscita LED, che gli consente di conoscere se ha realizzato o meno il *Fix* (ossia se ha agganciato un numero sufficiente di satelliti). Le altre linee del PIC utilizzate dal software che lo governa sono le seguenti: RB6 (inizializzata con resistore di pull-up interno attivo) legge lo stato del pulsante P2 che decide la modalità di funzionamento e permette la memorizzazione del punto da ritrovare (PDR); RA3, RB0 ed RB3, che comandano i led, rispettivamente, LD1, LD2 e LD3.

Per quanto riguarda l'accensione della barra del visualizzatore che segnala la distanza dal PDR e del cerchio di led gialli che dà la direzione, non disponendo di abbastanza I/O il microcontrollore si avvale di due I/O expander, siglati U2 e U3, con i quali

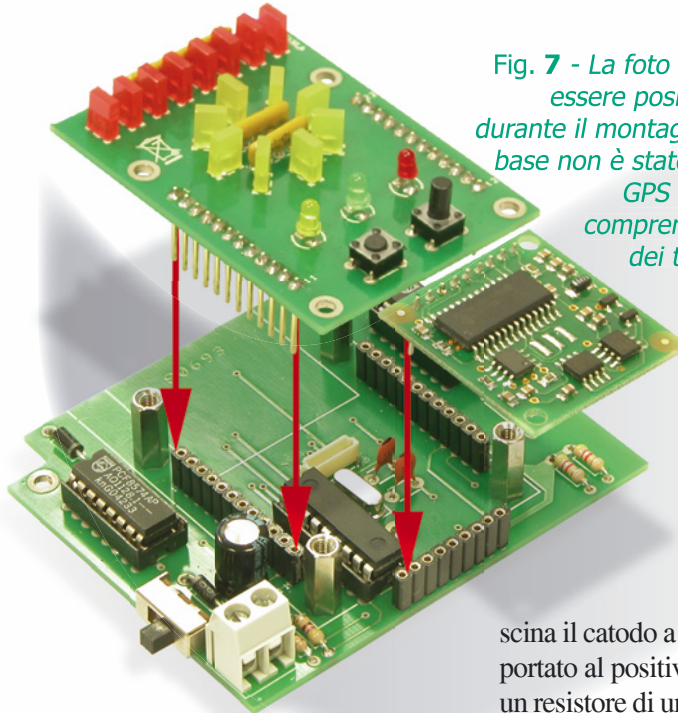


Fig. 7 - La foto mostra come devono essere posizionate le tre schede durante il montaggio. Notate che sulla base non è stato inserito il ricevitore GPS al fine di rendere più comprensibile l'orientamento dei tre PCB. È necessario posizionare il ricevitore GPS prima di innestare le due schede Led e bussola.

dialoga mediante un I²C-bus realizzato dalle proprie linee RA0 (SCL, ovvero il clock della comunicazione) ed RA1 (SDA, cioè il canale dati seriale). Un secondo bus I²C viene realizzato dal microcontrollore mediante RB4 (SDA) ed RB5 (SCL): serve per dialogare con la bussola elettronica (CMPS); torneremo tra breve su quest'ultimo, spiegando come funziona. Prima, diamo uno sguardo agli I/O expander usati per gestire il visualizzatore: si tratta dei PCF8574 prodotti dalla Philips, che possono funzionare in modo bidirezionale, nel senso che le otto linee (P0÷P7) disponibili funzionano, all'occorrenza, sia da ingressi che da uscite. In altre parole, il dispositivo con cui comunicano può sia leggere lo stato di esse (input) sia impostare la loro condizione (output). Ciò dipende dal fatto che il PCF8574 dispone, per ciascuna delle otto linee P0÷P7, di uno stadio driver logico e di un buffer di lettura. Il modo di funzionamento dipende dal comando ricevuto dal microcontrollore: nel nostro caso ognuno dei PCF8574 funziona solo in scrittura (output) e manda in uscita le condizioni logiche necessarie per far accendere i led secondo quanto stabilito dal microcontrollore. Ciascuna uscita opera in modo *sink*, ossia alimenta il rispettivo led quando ne tra-

scina il catodo a zero logico (l'anodo è portato al positivo di alimentazione da un resistore di una rete resistiva). Ogni linea di output dell'I/O expander può alimentare direttamente un led perché il proprio driver drena (a zero logico) fino a 25 mA.

Siccome lo standard I²C-bus prevede che i dispositivi siano collegati in parallelo sulle linee SDA ed SCL e che le istruzioni dirette dal dispositivo master (il nostro PIC16F648A...) alle periferiche contengano un indirizzo che permetta di farle riconoscere alla sola periferica destinataria, ogni appa-

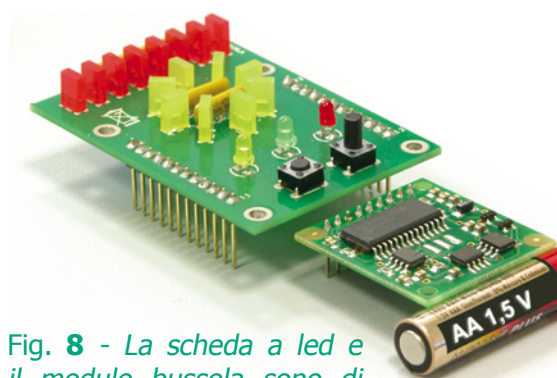


Fig. 8 - La scheda a led e il modulo bussola sono di dimensioni molto contenute.

rato dispone di tre linee che permettono di definire l'indirizzo. Nel caso del PCF8574, i pin corrispondenti sono 1, 2, 3 (A0, A1 e A2, rispettivamente bit di peso 1, 2, 4). Guardando lo schema elettrico, notate che abbiamo dato a U2 l'indirizzo 001 (1) mentre per U6 si è scelto 000 (0), in modo da evitare conflitti nel funzionamento.



Quando abbiamo progettato questo dispositivo, abbiamo pensato anche alla sua veste oltre che al suo circuito. Il localizzatore può essere facilmente inserito in un contenitore plastico modello 90-43-4A-R-BL, dotato di vano portatile in grado di ospitare 4 celle di formato AA, prodotto da BoxEnclosures (www.boxenclosures.com) e distribuito da Farnell (codice Farnell 4952870). Poiché il contenitore non viene fornito né forato né serigrafato, abbiamo preparato il materiale necessario sia per forare che per decorare il coperchio superiore, dotato di un'incassatura rettangolare profonda circa 1 mm. La, Figura B1, mostra il piano di foratura del coperchio, perfettamente simmetrico. Se praticate i fori rispettando le quote da noi riportate non avrete alcun problema nel far fuoriuscire da esso tanto i led quanto i pulsanti. La Figura B2, invece, rappresenta la serigrafia che deve essere stampata a colori su un foglio trasparente. Infine la Figura B3 propone lo sfondo della serigrafia, che deve essere interposto tra il coperchio e la serigrafia stampata su lucido. Il documento contenente le tre figure è scaricabile dal sito della rivista e include altre due serigrafie con sfondo nero, nel caso il bianco (editorialmente più indicato) non sia di vostro gradimento. Per incollare sia lo sfondo che la serigrafia, potete usare del nastro biadesivo o della colla a stick.

Dovete anche praticare un piccolo foro sul lato sinistro per far fuoriuscire l'interruttore di alimentazione. Il risultato finale è visibile in figura 5.

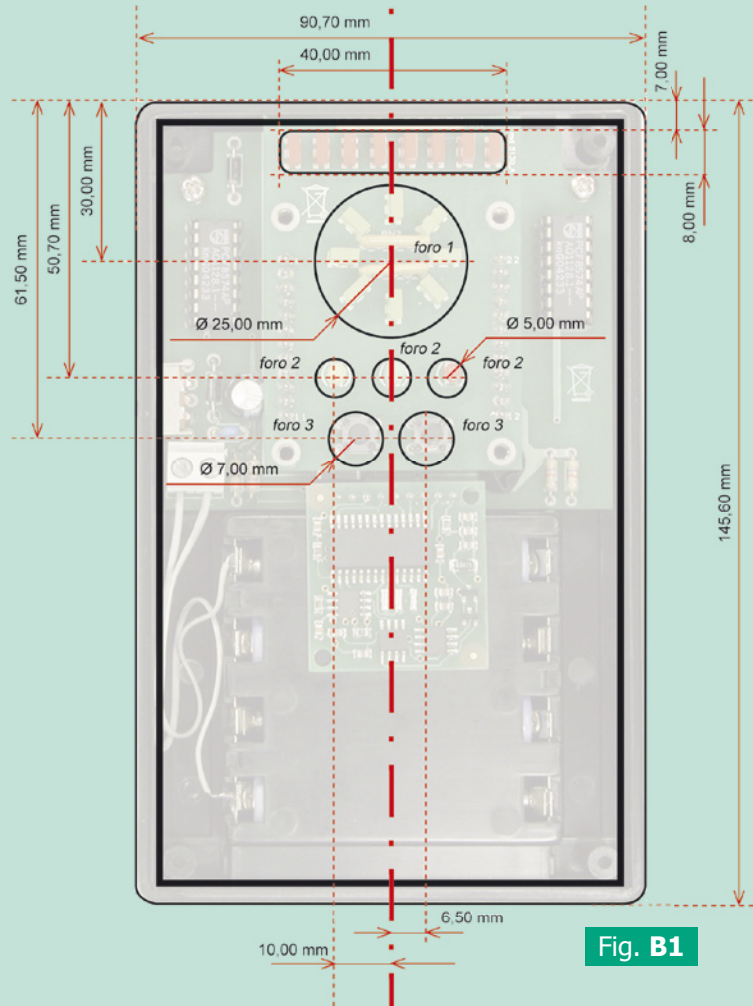


Fig. B1

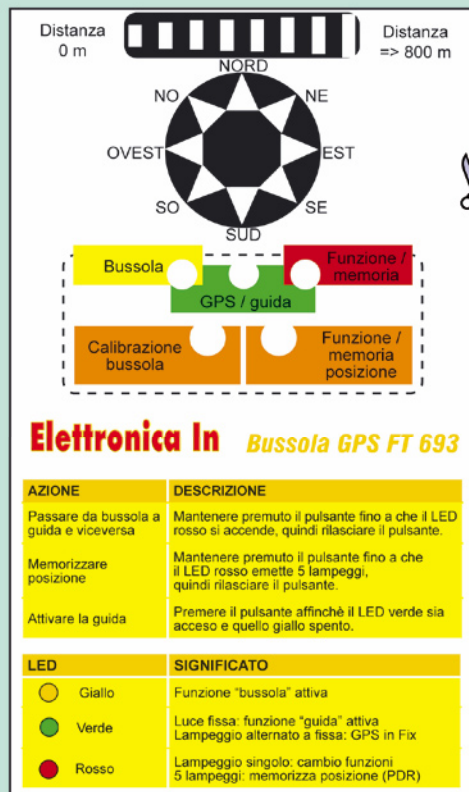


Fig. B2

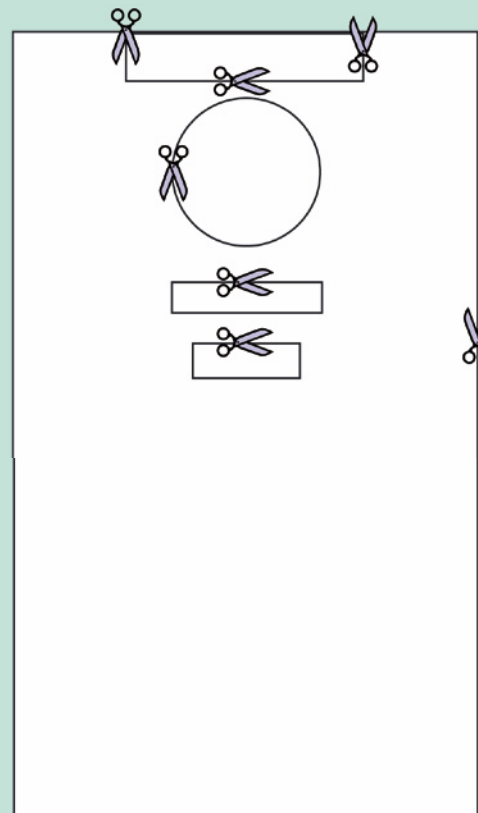


Fig. B3

Va inoltre detto che, essendo usati come esecutori dei comandi del microcontrollore e quindi soltanto in uscita, i due PCF8574 lavorano come unità *Slave* I²C-bus. Spiegato il funzionamento degli I/O-expander, concentriamoci sul modulo bussola: si tratta di un CMPS03, il cui funzionamento è basato su un sensore di campo magnetico terrestre KMZ51 di produzione Philips; anzi, a dire il vero i sensori sono due, collocati ortogonalmente. Il tutto viene gestito da un microcontrollore Microchip (PIC16F872 in versione SMD) programmato per leggere i segnali rilevati dai sensori e riuscire, con essi, a determinare la propria angolazione rispetto al Nord magnetico della Terra. Il micro trasmette all'esterno i dati elaborati e lo fa in due modi: mediante un segnale PWM e tramite un I²C-bus realizzato da due delle proprie linee di I/O, un po' come fa il nostro PIC16F648A. In quest'ultimo caso il modulo è una periferica *Slave*, nel senso che invia i dati sulla propria angolazione (è una bussola, alla fine...) solamente quando il dispositivo *Master*, che è poi l'U1 del nostro circuito, glielo chiede. L'uscita PWM può essere utilizzata in alternativa al bus I²C quando si intende realizzare una bussola analogica; diversamente dalla I²C-bus, fornisce sempre in uscita il proprio segnale, anche se non richiesto. Affinché possa funzionare correttamente, il modulo bussola deve stare perfettamente orizzontale, in piano; se non è a livello con il terreno, l'indicazione che fornisce è affetta da un certo errore, dovuto al fatto che il rilevamento viene influenzato da un'angolazione "errore", e quello indicato non è proprio l'angolo formato con la direttrice del Nord. Comunque, l'errore che ne deriva nella nostra applicazione non è significativo, quindi basta tenere il circuito in modo che sia abbastanza in piano e tutto va bene.

Inoltre, per dare indicazioni che siano le più corrette possibile, il CMPS03 deve essere calibrato; ciò si fa mediante una procedura che inizia con la

pressione del pulsante P1 collegato al suo contatto 6 (CAL) e che descriveremo quando affronteremo le note di utilizzo.

L'intero circuito è alimentato a batterie: per l'esattezza con un accumulatore da 4,8 V ottenuto con quattro elementi Ni-MH da 1,2 V ciascuno (da almeno 2.000 mA/h) posti in serie tra loro. La batteria va connessa ai punti + e - PWR; mediante l'interruttore SW1, possiamo accendere e spegnere l'apparecchio. I diodi D1 e D2 servono, oltre che per proteggere il circuito nel caso l'alimentazione venga applicata con polarità invertita, a ridurre la tensione da 4,8 a circa 3,6 V, con la quale vengono fatti funzionare il microcontrollore, il modulo bussola, il ricevitore GPS e gli I/O expander.

Realizzazione pratica

Giunti a questo punto possiamo vedere come realizzare l'apparecchio: fate riferimento al piano di montaggio (Pag. 48) ed aiutatevi anche con le Figure 6, 7 e 8, che mostrano alcuni particolari utili; la prima operazione da compiere è preparare i circuiti stampati, che sono due: uno è destinato ad ospitare tutti i 19 led e i pulsanti e va montato "a sandwich" sull'altro, sul quale vanno collocati i componenti attivi, ad eccezione del ricevitore GPS. Quest'ultimo andrà collegato all'apposito connettore.

Incise e forate le due basette, su quella di base disponete le resistenze, i diodi al silicio, gli zoccoli per il microcontrollore e i PCF8574, i condensatori e l'interruttore da c.s. SW1, che deve essere preferibilmente del tipo con terminali piegati a 90°, così da sporgere lateralmente dallo stampato; montate il connettore miniatura per il ricevitore GPS. Infine, inserite e stagnate (dal lato opposto) due file di strip femmina a passo 2,54 mm (alte 15 mm) da 11 vie ciascuna, che serviranno per inserire il circuito contenente il visualizzatore e i pulsanti (Fig. 7). Per l'alimentazione, montate una morsettiera a passo 5 mm in corrisponden-

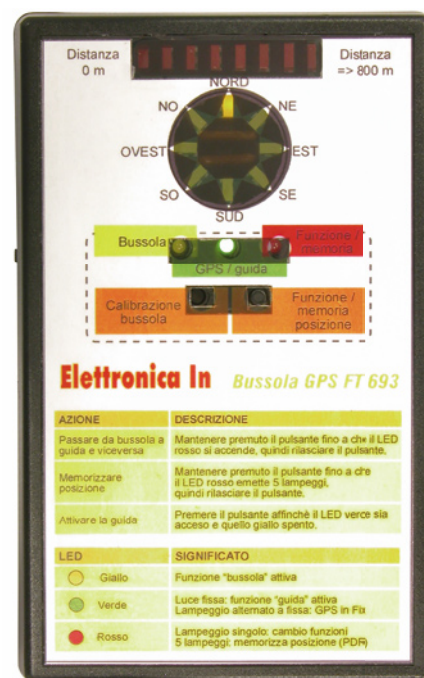


Fig. 9 - Ecco come si presenta il dispositivo una volta ultimato.

za delle piazzole + e - PWR. Fatto ciò, seguendo i disegni di montaggio illustrati in queste pagine inserite al loro posto i tre integrati.

Passate ora alla seconda basetta, sulla quale dovrete montare tutti i led, le reti resistive, i due micro-pulsanti da c.s., oltre, naturalmente, ai connettori per l'inserzione nella scheda base; essi sono delle semplici strip a passo 2,54 mm da 11 punte ciascuna, che avrete cura di collocare sotto, ovvero dal lato delle saldature (opposto a quello dove si trovano led e pulsanti).

Completati i due moduli, fissate il ricevitore GPS sul circuito più grande (base) usando del nastro biadesivo, quindi collegatelo allo stampato utilizzando il suo cavo.

Adesso, allo stesso stampato base, potete collegare il modulo CMPS03, usando uno spezzone di piattina a 9 conduttori. Affinché possa funzionare correttamente, il modulo bussola deve risultare perfettamente in piano; tuttavia, anche se non è parallelo al livello dell'acqua ma abbastanza orizzontale, l'errore che ne deriva, per la nostra applicazione non è significativo.

A questo punto, inserite il c.s. del vi- ➤

sualizzatore sul circuito base (Figura 7). L'alimentazione è ricavata da una batteria di quattro elementi Ni-MH da 1,2 volt posti in serie; bloccate con del silicone o del nastro isolante il pacco, così da renderlo compatto; poi collegate il positivo e il negativo, rimasti liberi, rispettivamente a +PWR e -PWR dello stampato base con due spezzoni di filo in rame isolato da 0,5 mm². Il contenitore deve essere in plastica o altro materiale non metallico, in primo luogo perché se il ricevitore GPS sta al suo interno le onde radio in arrivo dai satelliti devono attraversarne le pareti e poi perché parti metalliche troppo vicine al modulo bussola, specie se ferromagnetiche, potrebbero alterare il rilevamento della posizione.

Calibrare la bussola

La bussola prima di essere utilizzata deve essere calibrata; l'operazione si avvia premendo il pulsante P1, che suggeriamo di non rendere accessibile all'esterno, ma solo tramite un foro praticato sul piano superiore del contenitore che avrete scelto per racchiudere l'intero dispositivo; tale accorgimento permetterà di premere il tasto con la punta di una penna solo quando serve ed eviterà la pressione accidentale durante l'uso, pressione che potrebbe scalibrare la bussola. La procedura di calibrazione viene effettuata a sistema acceso, agendo sul pulsante P1: dopo aver dato tensione al circuito ed aver impostato la modalità bussola (ciò si ottiene premendo P2 fino a veder accendersi LD3, poi rilasciandolo e verificando che stia ac-

ceso LD1), aiutandosi con una bussola tradizionale si deve orientare il tutto in modo che il lato del modulo CMPS03 opposto a quello che ha i nove contatti sia rivolto a Nord (vedere Figura 6 che riassume tutta la sequenza di calibrazione), ovvero che sia esattamente perpendicolare alla linea immaginaria che porta al Nord Terrestre. Fatto ciò bisogna premere P1 e rilasciarlo; ora è stato calibrato il Nord. Nello stesso modo bisogna procedere con la calibrazione degli altri punti cardinali, rammentando che la sequenza è Nord, Est, Sud, Ovest. Perciò si deve orientare il dispositivo in maniera da disporre il solito lato del CMPS03 ad Est; a questo punto si preme e si rilascia P1, così da tarare il punto cardinale dell'Est. Dopo si passa alla calibrazione del Sud: si orienta il tutto in modo che il lato del CMPS03 opposto a quello dei contatti sia rivolto a Sud e si preme un'altra volta il solito pulsante; rilasciandolo, avviene la taratura del Sud. Infine, non resta che calibrare la bussola per l'Ovest: si orienta il dispositivo in modo che il solito lato del CMPS03 sia rivolto a Ovest, poi si preme e si rilascia P1. Fatto questo, la bussola è calibrata e l'intero apparato è pronto per l'uso.

Utilizzo in pratica

Accendete il dispositivo ed attendete che il GPS vada in Fix (ve lo segnala con dei lampeggi del led verde). Posizionatevi nel punto che considerate "da raggiungere" quando ritornerete dall'escursione (PDR, ad esempio dove avete parcheggiato l'autovet-

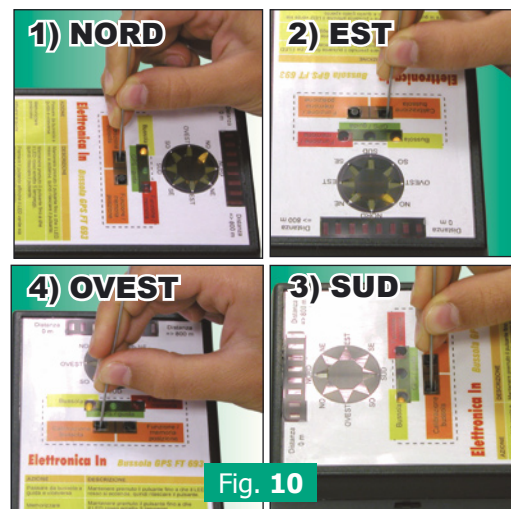


Fig. 10

Le quattro fasi della calibrazione.

tura) e premete per tre secondi il pulsante, al fine di memorizzare il PDR. Spegnete la bussola GPS e allontanatevi di qualche centinaio di metri, in direzione casuale, avendo sempre un contatto visivo del PDR. Riaccendete il dispositivo ed attendete che vada in Fix. La modalità predefinita all'avvio è sempre la bussola, per cui dovete vedere uno degli otto led gialli acceso ad indicare il Nord: ruotate su voi stessi per verificare che si accenda sempre e solo un led che punta a Nord. Premete ora il pulsante, per attivare la modalità "guida": si accende un led giallo che, indipendentemente dal vostro orientamento, punta sempre verso il PDR ed un led rosso ogni 100 metri di distanza dal PDR. Ruotate su voi stessi per verificare che il dispositivo punti sempre verso l'autovettura. Ricordate di mantenere la bussola parallela al suolo. Buon divertimento.

per il

MATERIALE

Il materiale utilizzato in questo progetto è reperibile presso qualsiasi negozio di elettronica. Il master dei CS, il firmware per programmare il micro ed il file con le quote meccaniche sono scaricabili gratuitamente dal sito della rivista. Il ricevitore GPS cod. FV-M8 (65,00 Euro), la bussola cod. CMPS03 (44,00 Euro) ed il PIC16F648A vergine (8,00 Euro) possono essere acquistati presso Futura Elettronica. Tutti i prezzi sono comprensivi di IVA.

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

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

Robot In il piccolo gigante

di *Mirco Segatello*

Piccolo robot in grado di muoversi autonomamente su di un percorso o all'interno di un'area prestabilita. Dispone di numerose funzioni quali l'inseguimento di una linea disegnata sul piano d'appoggio, la ricerca della sorgente luminosa più intensa, il movimento libero evitando gli ostacoli e il programma per incontri di "minisumo". È inoltre telecomandabile tramite un programma per PC. Prima puntata.

La robotica rappresenta un settore di grande importanza a livello internazionale, sia in campo scientifico che culturale, e attualmente sta continuando ad avere un enorme sviluppo tecnologico; per quanto si tratti di una scienza relativamente recente, i grandi benefici che essa può recare hanno fatto sì che si sviluppasse molto più velocemente di tante altre tecnologie, al punto da venire impiegata nei più svariati settori quali l'informatica, la meccanica, l'elettronica, l'intelligenza artificiale, l'automazione, la fisica, ecc... Gli innumerevoli campi di impiego spaziano dalla medicina alle catene

di montaggio, dall'intrattenimento alla manipolazione di materiali e sostanze pericolose e l'interesse non è destinato a diminuire, anzi... tutt'altro: si pensi ad esempio ai bracci manipolatori utilizzati negli ambienti sterili, ai robot antimina degli artificieri, ai sistemi di assemblaggio sia di apparecchiature di grandi dimensioni (catene di montaggio in automobilistiche) che di componenti microscopici (linee di produzione "pick-and-place per il montaggio dei componenti elettronici sui circuiti stampati), per concludere con i "robot chirurgo", che sono giunti ad un livello di perfezione e sofisticazione tale da ➤

essere introdotti ed utilizzati nelle sale operatorie.

Nel suo piccolo, anche il settore dell'intrattenimento ha una sua ragione di esistere: viene infatti utilizzato come mezzo di divulgazione delle nuove tecnologie (a volte anche molto complesse e troppo spesso presentate con estrema pomposità o freddezza) in modo da far crescere l'interesse specifico facendo leva su attività divertenti, come la realizzazione di robot puramente ludici. Sia che si tratti di impiego tecnologico, sia che si tratti di puro e semplice divertimento, questi robot hanno raccolto il favore del pubblico, al punto che sono sorte molte iniziative fieristiche e competizioni tra "robottini" a livello mondiale: tant'è che sono anche state ideate le più svariate categorie di appartenenza, un po' come nel calcio in cui, partendo dalla serie D si arriva alla serie A, alla Champion's League e via dicendo. Ritorniamo comunque alla robotica che più ci interessa... Questa particolare attività è di grande interesse didattico proprio per la sua naturale propensione a raccogliere gli insegnamenti da molte discipline (complesse!!!) e di riproporli con veste più reale e pratica (se vogliamo "giocosa"), adatta ad un approccio diretto e ad un apprendimento veloce.

Di fronte ad oggetti concreti i giovani (futuri fruitori delle nuove tecnologie) sviluppano un nuovo atteggiamento di interesse più attivo, atteggiamento normalmente non riscontrabile nell'insegnamento puramente teorico. Questo perché l'attività pratica facilita ed alleggerisce l'apprendimento, stimola il pensiero creativo e migliora quello critico, sviluppa la capacità di analizzare e risolvere problemi concreti e molto ancora... La possibilità di lavoro di gruppo, inoltre, favorisce le interazioni tra persone, incrementa la capacità di comunicazione e permette di migliorare la capacità di organizza-

re il lavoro. Lo studio e l'applicazione in questo settore favoriscono anche un atteggiamento di interesse verso le discipline più classiche quali la matematica, la fisica e il disegno, che rappresentano le basi e il punto di partenza per lo sviluppo di applicazioni in campo robotico.

La possibilità di avviare delle piccole competizioni, magari a livello scolastico (ciò si sta verificando sempre più spesso), favorisce un coinvolgimento ancora più spinto dei giovani che, se opportunamente supportati, possono poi sviluppare in modo autonomo, al di fuori delle normali attività didattiche, le proprie idee e dare forma alle proprie realizzazioni. Non deve mancare un momento di analisi dei problemi incontrati durante l'attività: forse questo punto è più importante addirittura dei risultati raggiunti perché è attraverso la risoluzione dei problemi che avviene la maggior parte dell'apprendimento; è proprio quando si ha un problema da risolvere che si usa maggiormente la logica. La consapevolezza di scelte errate e l'analisi di soluzioni alternative

ti- colo presentiamo il progetto di un piccolo robot di categoria "minisumo". La presentazione del progetto, divisa in due puntate, è stata fatta puntando al massimo in termini di semplicità e chiarezza in modo da essere fruibile dalla maggior parte dei lettori, garantendo contemporaneamente la certezza di un sicuro funzionamento. Siamo convinti che questa è l'occasione giusta per schiere di appassionati di approfondire e allargare le conoscenze nella robotica, settore strettamente legato all'elettronica. Abbiamo anche posto attenzione alla parte meno tecnica ma più divertente: il nostro "Robot In" (questo è il nome che abbiamo scelto) è stato realizzato rispettando i regolamenti internazionali così da poterlo utilizzare anche in raduni ed eventi agonistici, ormai molto diffusi anche in Italia. Scopriamolo insieme.

La struttura di Robot In

Il nostro intento è quello di favorire una cultura robotica anche a livello hobbistico e didattico favorendo la massima diffusione di questo settore

Schema a blocchi di Robot In.

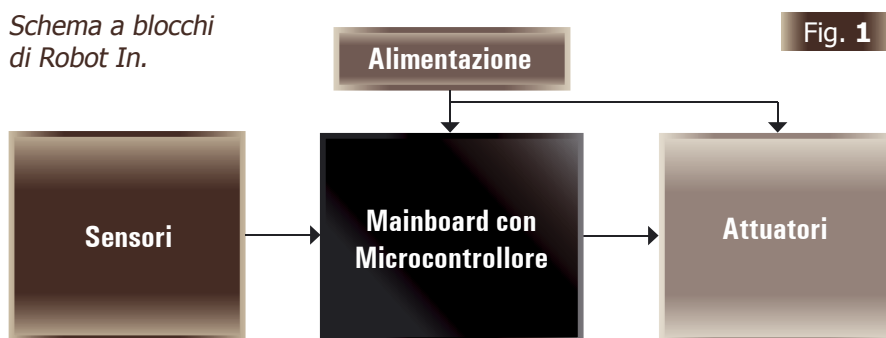


Fig. 1

implica un'attività di pensiero e di elaborazione che è uno degli scopi "guida" delle attività robotiche a scuola. Al di là del valore tecnico e sociale, l'argomento è di sicuro interesse per quanti, anche hobbisti, sono incuriositi da questa nuova forma di divertimento/apprendimento e desiderano realizzare in autonomia il proprio robot.

Anche *Elettronica In* ha più volte trattato l'argomento e in questo ar-

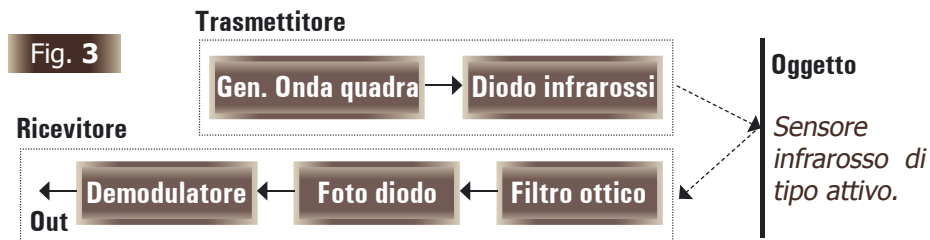
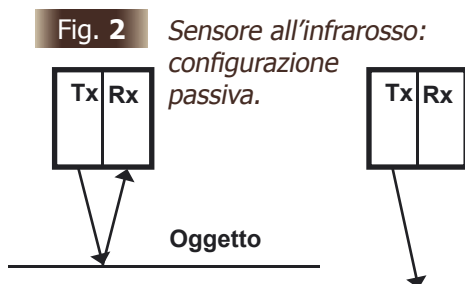
della tecnologia: l'obiettivo è quello di realizzare un piccolo robot di facile costruzione (per quanto possibile) ma nello stesso tempo completo ed evoluto dal punto di vista elettronico, tenendo conto che i lettori possono essere sia coloro che vogliono realizzare il proprio robot per le più svariate ragioni (studio, lavoro, curiosità...) sia coloro che, pur non volendo realizzare il progetto, intendono conoscere qualcosa

di più relativamente a questo nuovo filone tecnologico.

Essendo però questa una rivista di elettronica, ci sembra logico pensare ad una realizzazione in cui la parte meccanica sia ridotta al minimo. Dopo molte valutazioni, abbiamo optato per un robot a movimentazione autonoma su ruote, forse classico, ma sicuramente adatto come primo lavoro. Avremo comunque l'occasione per sviluppare alcuni concetti comuni a tutti i robot e per approfondire alcuni argomenti di interesse generale, ma veniamo al dunque...

L'idea di base è quella di realizzare una parte meccanica ed elettronica completa sulla quale poi di volta in volta andremo a selezionare le varie funzioni. In linea di principio Robot In può essere schematizzato come visibile in Figura 1: il blocco "sensori" rappresenta tutti quei dispositivi in grado di eseguire le rilevazioni relative all'ambiente (presenza di ostacolo, fonte di calore, suono, ecc.) e di comunicarle al blocco "microcontrollore" che agisce di conseguenza attivando opportunamente il blocco degli attuatori, ossia quei dispositivi a cui è affidato il movimento del robot.

Il blocco di alimentazione è comune a tutti gli elementi. Francamente, avremmo potuto limitarci a descrivere solo i dispositivi utilizzati in questo progetto, ma abbiamo preferito aggiungere qualche nota riguardante i più comuni sensori e attuatori che potete trovare applicati perché, come dicevamo, vogliamo favorire l'interesse e la comprensione di nuove tecnologie: questo è lo



spirito di questa rivista fin dal lontano fascicolo numero 1.

Sensori

Senza entrare eccessivamente nel dettaglio ed evitando inutili tecnicismi, vediamo ora i sensori più comunemente utilizzati nei robot come il nostro.

Finecorsa meccanici per il rilevamento di oggetti

Si tratta di interruttori e microswitch. Serve il contatto fisico con l'oggetto pertanto sono scarsamente utilizzati, ma sono semplici ed economici e quindi vengono solitamente impiegati nelle prime realizzazioni.

Sensori ad ultrasuoni per il rilevamento di oggetti

Si basano sull'effetto fisico dell'eco: viene generato un ultrasuono e si misura il tempo impiegato per ricevere il suo eco (Madre Natura docet: i pipistrelli usano da sempre l'eco per catturare gli insetti ed evitare gli ostacoli al buio). Quando applicato ai robot, questo metodo consente di rilevare oggetti, anche trasparenti, a grande distanza.

Sensori all'infrarosso passivi

Schematizzati in Figura 2, essi sono costituiti da un diodo led all'infrarosso (TX) e da un fototransistor (RX) posti nello stesso contenitore e rivolti sullo stesso lato. Il trasmettitore (led all'infrarosso) invia un fascio di luce che si riflette sulla superficie posta di fronte e colpisce il ricevitore (RX) mandandone in conduzione il fototransistor che in questo modo indica la presenza di un oggetto nelle vicinanze del sen-

sore. Quando invece non incontra alcun oggetto, il fascio di luce del trasmettitore non viene riflesso e al ricevitore non giunge alcun segnale. Hanno un raggio d'azione di pochi millimetri e sono facilmente influenzabili da fonti luminose esterne; per questo motivo vengono utilizzati esclusivamente per riconoscere le fasce colorate che definiscono l'area o il percorso dove si muove il robot.

Sensori all'infrarosso attivi

Sono costituiti da un trasmettitore e un ricevitore separati e sfruttano anch'essi la riflessione del raggio di luce all'infrarosso. Il trasmettitore genera degli impulsi luminosi modulati con un segnale a onda quadra (di solito a 38 kHz); il ricevitore è costituito da un filtro ottico per raggi infrarossi e un fotodiodo e dispone di un demodulatore centrato sulla frequenza del trasmettitore. Lo schema di principio di questa tipologia di sensori è visibile in Figura 3.

I due grandi vantaggi offerti dai sensori IR attivi sono la portata maggiore e l'alta immunità a fonti luminose esterne; esistono particolari versioni più sofisticate che rilevano sia la presenza che la distanza di un ostacolo.

Sensori di luce e calore

I primi sono sensori costituiti da fotoresistenze e fotodiodi in grado di captare le emissioni di luce visibile. Gli altri, invece, sono sensori di tipo PIR, particolari dispositivi sensibili ai raggi infrarossi localizzati nella gamma di frequenze che gli esseri viventi percepiscono come calore. ➤

Sensori di suoni

Sono essenzialmente dei microfoni opportunamente amplificati e a volte filtrati per riconoscere solo determinate frequenze.

Altri sensori

Esistono molti altri sensori idonei a captare ad esempio la direzione del campo terrestre (funzione bussola) oppure atti a rilevare l'inclinazione del robot (funzione inclinometro) oppure sensori di accelerazione utilizzati per rilevare eventuali urti. Robot In, in particolare, è dotato di un sensore di ribaltamento (Fig. 8).

Attuatori

Gli attuatori, sono tutti quei dispositivi che convertono l'energia elettrica in un movimento meccanico.

Motori

Nella maggior parte dei casi si tratta di motori in corrente continua (più raramente motori passo-passo) perché l'alimentazione del robot è a batterie e quindi in corrente continua. In realtà sarebbe meglio parlare di motoriduttori in quanto un normale motore in continua dispone di un elevato numero di giri (da 1000 a 20000 giri/min) e una bassa coppia, cioè l'esatto contrario di ciò che serve a una ruota di movimentazione (basso numero di giri e alta coppia). Si preferisce pertanto abbinare il motore ad un riduttore con rapporti di riduzione che variano da 1:100 a 1:500 così da ridurre il numero di giri ed aumentare la coppia. Indicativamente, considerando una marcia con velocità di circa 0,5 m/sec e una ruota di diametro 7 cm, il motoriduttore non deve girare oltre i 135 giri/min.

Servocontrolli da modellismo

Si tratta di motori fortemente ridotti con abbinata una piccola logica di controllo in grado di far girare un perno per circa $\pm 60^\circ$ o $\pm 90^\circ$ con una coppia che può arrivare anche a

qualche chilogrammo per centimetro. Sono impiegati per la movimentazione di robot umanoidi oppure per bracci meccanici e il loro controllo avviene con segnali digitali.

Caratteristiche e classificazione dei minirobot

Introducendo Robot In, vi avevamo già anticipato che lo abbiamo progettato rispettando alcune regole riconosciute a livello internazionale: vediamole insieme.

Per quanto riguarda i robot a movimentazione autonoma su ruote, in base alle dimensioni e al peso, sono state identificate le categorie elencate di seguito.

- Sumo: dimensioni 20 x 20 cm, peso massimo 3 kg;
- Minisumo: dimensioni 10 x 10 cm, peso massimo 500 g;
- Microsumo: dimensioni 5 x 5 x 5 cm, peso massimo 100 g;
- Pesi leggeri: dimensioni 20 x 20 cm, peso massimo 1 kg;
- Pesi medi: dimensioni 20 x 20 cm, peso massimo 1,5 kg.

Dal nome stesso si capisce che ci stiamo riferendo a categorie da combattimento ovvero "Sumo". Volendo realizzare un robot che non sia troppo impegnativo dal punto di vista elettro-meccanico abbiamo optato per la categoria Minisumo, che è anche la più diffusa nel nostro paese.

Come avete potuto intuire, Robot In è un robottino di piccole dimensioni (10 x 10 cm) senza limitazioni in altezza, ma di peso massimo consentito non superiore a 500 g. È studiato per le competizioni di Sumo su di un ring di 77 cm di diametro e quindi, tutto sommato, facilmente gestibile per chiunque. Non ci siamo limitati, però, alla sola funzione di "robot lottatore": abbiamo infatti voluto aggiungere altre funzioni particolarmente interessanti che tra poco andremo a descrivere.

Da quanto appena esposto è facile

comprendere che la realizzazione elettro-meccanica rappresenta solo uno degli aspetti di questo tipo di robot; lo sviluppo del programma che gestisce i movimenti è altrettanto importante poiché, in definitiva, è il firmware contenuto all'interno del microcontrollore a decidere il comportamento e i movimenti che il robot dovrà eseguire nelle varie situazioni. Tutto ciò sarà descritto nei prossimi capitoli; ora vediamo le varie funzioni che abbiamo voluto dare a Robot In.

Movimentazione

Il robot è posizionato semplicemente al di sopra di una superficie disseminata di oggetti e deve muoversi evitando gli ostacoli; se posizionato sopra un tavolo deve fare in modo di non cadere.

Labirinto

Il robot è posizionato all'interno di un vero e proprio labirinto in una posizione prestabilita. Sarà suo compito, discriminando tra ostacoli e percorsi possibili, raggiungere l'uscita nel più breve tempo possibile.

Inseguitore di linea

In questo caso il robot è posto su di una superficie piana di colore bianco sulla quale è stata disegnata una linea di colore nero (di solito del nastro isolante da elettricisti) e al robot è affidato il compito di avanzare seguendo la linea. Questa funzione è largamente utilizzata in campo industriale per far spostare dei carrelli automatizzati tra i vari reparti.

Sumo

Si tratta di una competizione che si svolge tra due robot della stessa categoria sopra un ring di dimensioni prefissate. Lo scopo è quello di fare in modo che il proprio robot butti l'avversario fuori dal ring. Per fare questo, deve essere in grado di riconoscere la posizione dell'avversario e quindi elaborare delle manovre per

spingerlo fuori. Ovviamente deve anche riconoscere il bordo del ring per evitare egli stesso di uscire.

Inseguire di luce

In questo caso il robot posizionato sopra un pavimento, una volta acceso, dovrà cercare la fonte luminosa primaria dirigendosi verso di essa ed inseguendola nel caso dovesse spostarsi.

Gestione radio

Anche se non fondamentale, in quanto i robot devono essere autonomi, riteniamo interessante poter proporre questo tipo di applicazione spesso presente in robot più evoluti. Le applicazioni sono svariate, dalla semplice telelettura dei sensori del robot al telecontrollo totale della movimentazione.

Schema elettrico

Come preannunciato, abbiamo fornito al nostro robot svariate funzioni fruibili in modo diverso a seconda di come il firmware del PIC viene istruito durante la fase di accensione. Abbiamo inoltre semplificato al massimo la parte meccanica facendo in modo che i circuiti stampati stessi diventassero anche la struttura di supporto, eliminando in questo modo la necessità di particolari metallici come viti e squadrette. In ragione di ciò, è stato necessario pensare a due diversi circuiti che, opportunamente connessi, permettessero una facile installazione delle batterie e dei motori. Poiché abbiamo due PCB, abbiamo anche due schemi, uno relativo alla scheda madre e uno per i sensori. Iniziamo a descrivere la scheda madre.

Il robot viene alimentato tramite sei celle AA (indifferentemente di tipo Ni-Cd o Ni-Mh) con una tensione pari a $6 \times 1,2 = 7,2$ volt. È stato previsto l'utilizzo di batterie ricaricabili in quanto a lungo andare rappresentano la soluzione più economica; tuttavia si possono usare senza pro-

blemi le normali pile a secco.

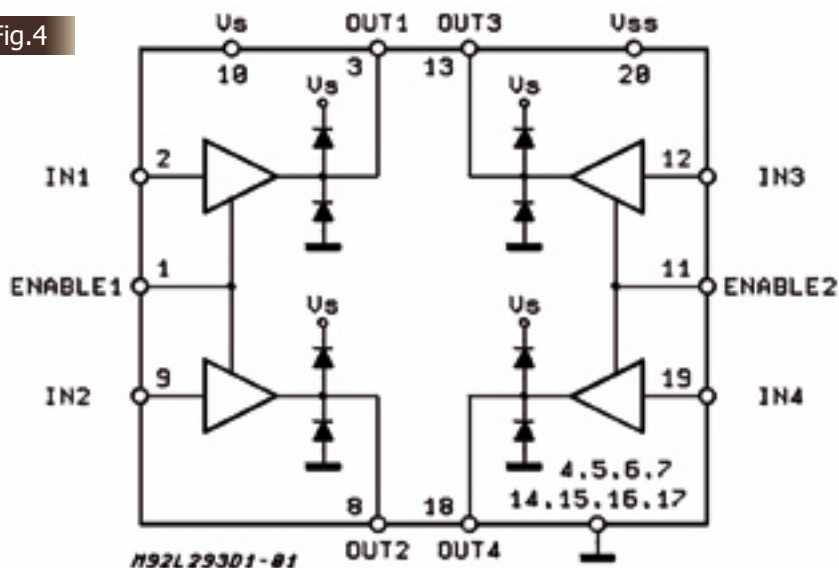
Il pacco batterie viene collegato al connettore J2 e il deviatore SW1 funge sia da interruttore di accensione che da commutatore per la ricarica: quando si trova nella posizione di ON esso collega le batterie al regolatore di tensione che alimenta tutto il circuito, mentre quando è a OFF collega le batterie al connettore J1, al quale si può innestare facilmente il caricabatterie esterno. Questo sistema permette la ricarica delle batterie senza doverle smontare dal robot, a patto di possedere un caricabatterie idoneo (come quello pubblicato sulle riviste 119 e 120). Segue uno stadio di stabilizzazione formato dal componente U1 (LM2937) che permette di ottenere 5 volt perfettamente stabilizzati indipendentemente dal valore di carica delle batterie. Il 7805 non poteva essere utilizzato perché necessita di almeno 8 V in ingresso per stabilizzare a 5 V in uscita: il nostro, invece, è un "low-drop-out" ossia un circuito in grado di generare 5 V stabilizzati con minimo 5,5 V in ingresso, permettendo così di sfruttare al massimo le batterie.

Per la movimentazione, sono stati usati due motoriduttori della Pololu modello GM2 con rapporto di riduzione di 224:1 e capaci di una

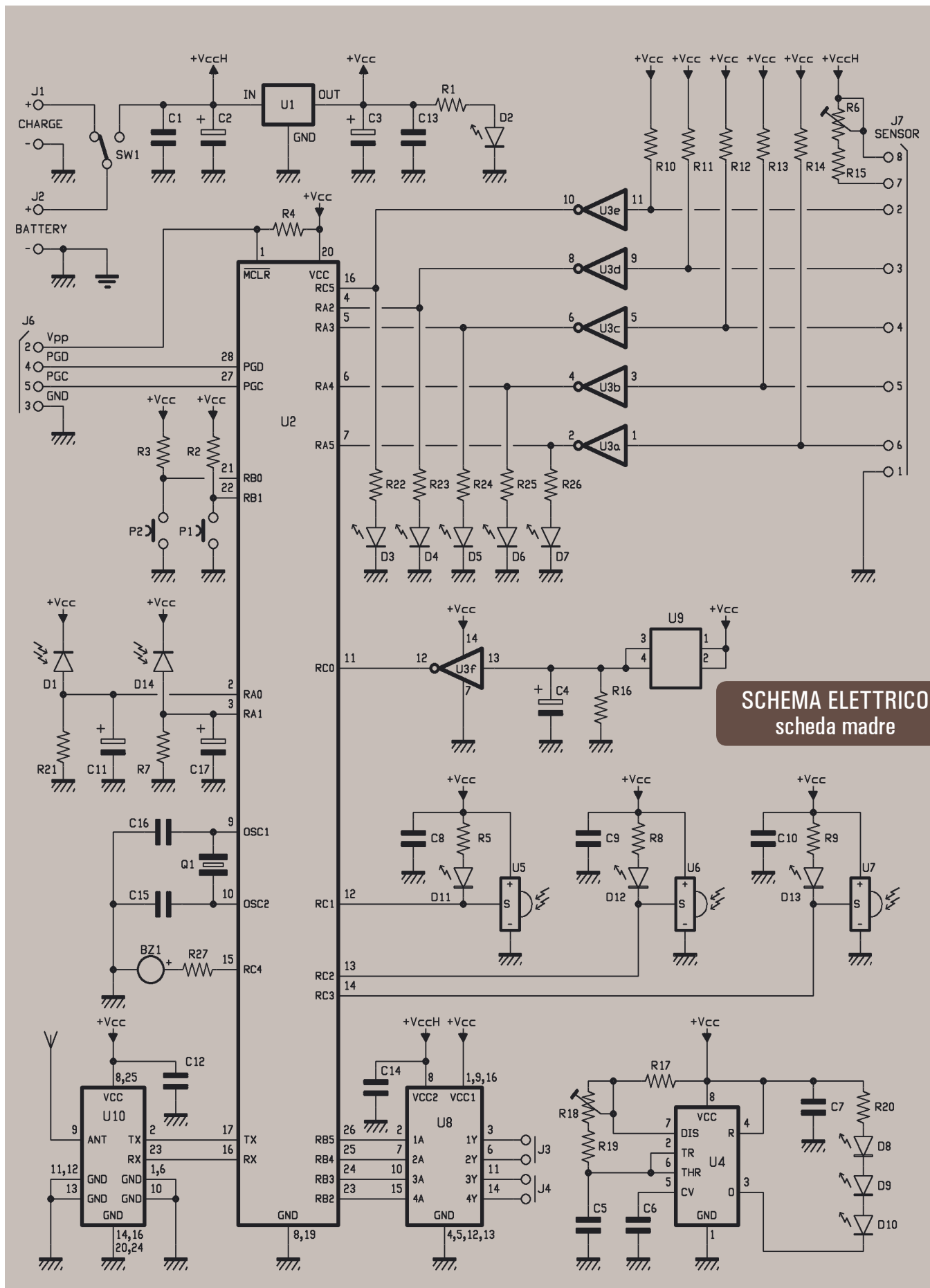
coppia di 3,5 kg/cm alla tensione nominale di 5 V. La corrente assorbita a vuoto è di 52 mA, mentre la corrente a ruota bloccata è di 600 mA, corrispondenti alla corrente di spunto dei motori al momento della partenza del robot. Le ruote abbinabili a questo motoriduttore hanno un diametro di 67 mm e sono predisposte per l'inserimento di un elastico sulla circonferenza per aumentarne l'attrito al suolo. Chi volesse utilizzare motori diversi da quelli indicati deve fare molta attenzione, pena il danneggiamento dei componenti sulla scheda di controllo. Diciamo subito che i motori per mini 4WD non vanno bene in quanto previsti per funzionare a tensioni molto basse e, se alimentati a 5 V, l'assorbimento in corrente sarebbe così elevato da bruciare subito il circuito di controllo. Tutti i motori in presa diretta (cioè senza motoriduttore) presentano un elevato numero di giri (da 1.000 sino a 20.000 giri/min) con una coppia molto bassa e sono praticamente inutilizzabili. Si possono eventualmente recuperare i motoriduttori funzionanti in corrente continua da qualche vecchia automobilina-giocattolo assicurandosi che funzionino con tensioni di 5-8 volt e che non assorbano più di 0,5 A. Quale che sia la scelta dei mo-

>

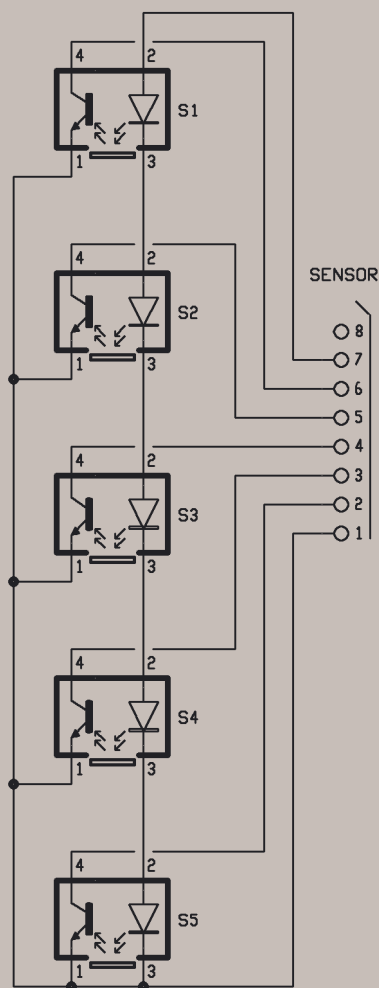
Fig.4



Struttura interna del "bridge ad H" modello L293D.



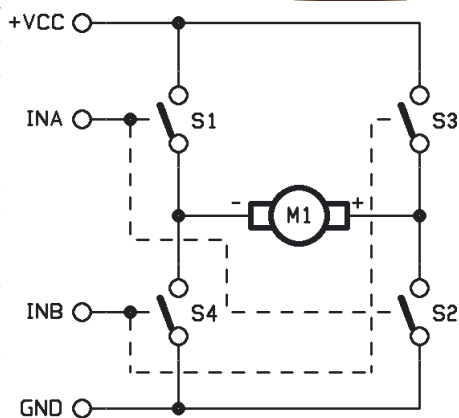
SCHEMA ELETTRICO sensori



tori, il circuito di controllo ad essi associato deve essere del tipo bidirezionale per permettere al robot di muoversi in tutte le direzioni; inoltre deve poter gestire la massima corrente assorbita dai motori: pertanto la scelta è caduta sull'integrato L293D (U8), che contiene quattro driver (amplificatori di corrente per motori) in configurazione push-pull in grado di pilotare carichi che assorbono fino a 1,2 ampere; inoltre è completo di diodi di protezione interni. Combinando opportunamente i quattro stadi interni all'L293D è possibile ottenere la classica configurazione denominata "ponte H" per il controllo bidirezionale dei motori. Per meglio comprendere questa

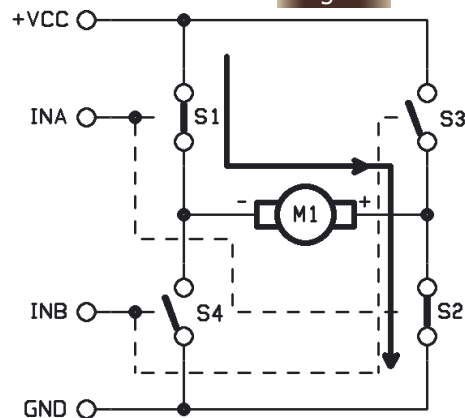
funzione, facciamo riferimento alla Figura 5 che schematizza il ponte H: notate che l'ingresso A chiude gli interruttori S1 e S2 mentre l'ingresso B comanda gli interruttori S3 e S4. Ovviamente questi interruttori all'interno dell'integrato L293 sono ottenuti con dei transistor. Se nessuno degli ingressi è attivo, non vi è circolazione di corrente pertanto il motore non gira; se invece attiviamo l'ingresso A, si chiudono gli interruttori S1 e S2 permettendo la circolazione della corrente come in Figura 6 e la rotazione del motore avviene in un senso. Quando invece è attivo il solo ingresso B, sono chiusi gli interruttori S3 e S4 e la circolazione di corrente avviene come in fig. 7, ovvero in direzione opposta al caso precedente e quindi anche la rotazione avviene nel verso opposto. Lo schema interno di questo integrato è stato ricavato direttamente dai data sheets di SGS Thompson ed è visibile in Figura 4. Gli ingressi Enable 1 e 2 abilitano i vari buffer di uscita: se opportunamente gestiti (ad esempio in PWM) essi permettono di variare la velocità di rotazione dei motori. I diodi interni, due per ogni uscita, servono a smorzare i picchi di tensione generati dalla componente induttiva dei motori durante le fasi di avvio, arresto e sforzo immediato. Non viene implementata in questo caso la funzione di frenatura, ottenibile attivando gli interruttori S2/S4 oppure S1/S3; al verificarsi di queste due situazioni, il motore sarebbe in cortocircuito e la corrente autogenerata verrebbe utilizzata come freno: quando viene tolta la tensione di alimentazione, infatti, il motore continua a girare a causa dell'inerzia meccanica, diventando di fatto una piccola dinamo che lavora su un carico tendente a 0 ohm e sforza molto. Viste però le esigue masse in gioco e la forte riduzione del numero di giri ottenuta con gli ingranaggi, la frenatura è inutile. Il cuore della mainboard è il >

Fig. 5



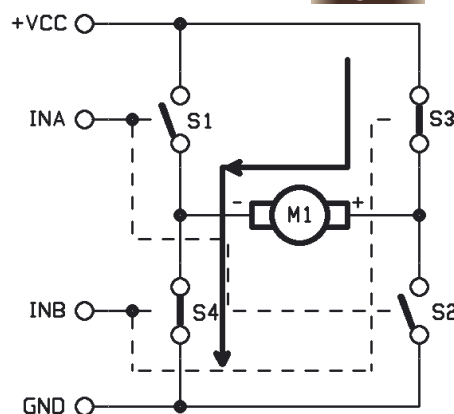
Accensione di un motore con ponte ad H, costituito da quattro interruttori comandati a due a due. Se nessun ingresso è attivo, non vi è circolazione di corrente e il motore non gira.

Fig. 6



Attivando l'ingresso A vengono chiusi gli interruttori S1 ed S2: il motore gira in un senso.

Fig. 7



Attivando l'ingresso B vengono chiusi gli interruttori S3 ed S4: il motore gira in senso opposto.

microcontrollore PIC16F876 (U2) che provvede a gestire tutti i segnali di controllo nonché i sensori, numerosi e tutti importanti: sensori di linea, di ribaltamento, di luce e di presenza ostacolo.

I sensori di linea permettono al robot di riconoscere tanto il bordo del tavolo quanto delle linee presenti su di esso: ne usiamo cinque, tutti montati sul secondo circuito stampato (fare riferimento allo schema dei sensori) e abbiamo scelto il modello CNY70 a riflessione (spiegato nel capitolo relativo ai tipi di sensore); quando il robot è assemblato, essi si trovano a pochi millimetri dalla superficie di appoggio. I due sensori laterali più esterni (S1 e S5) sono adibiti al riconoscimento del bordo esterno dell'area di manovra, permettendo al robot di muoversi in una zona ben definita. I tre sensori centrali (S2, S3 e S4) servono per rilevare una linea quando viene eseguito il programma "inseguitore di linea" e possono funzionare sia insieme agli altri che autonomamente. Affinché i sensori possano funzionare correttamente, è necessario un differente indice di rifrazione tra la zona di movimento e le linee e questo può essere ottenuto con colori differenziati. Possiamo quindi fare muovere il robot su di un tavolo di colore bianco e usare come linee del nastro isolante da elettricisti. Il tavolo deve essere molto luminoso (ad esempio bianco lucido) mentre il nastro isolante di colore nero, possibilmente opaco. Potremmo anche fare il contrario e cioè far muovere il robot sopra

una superficie nera poco riflettente ed usare come linee del nastro isolante bianco, il più lucido possibile. Questa modalità è impiegata solo nel caso di utilizzo del robot per il combattimento tipo Sumo.

Vediamo ora il sensore di ribaltamento U9: esso è utile per segnalare al micro che il robot non si trova più in posizione orizzontale sopra la superficie (è caduto oppure è stato preso in mano); in tal modo il micro può compiere azioni particolari come ad esempio forzare lo spegnimento dei motori. Questo sensore contiene una piccola sfera di metallo, libera di muoversi (Figura 8), che provvede al cortocircuito dei contatti interni quando il sensore è posto in verticale, mentre mantiene il circuito aperto se il sensore è posizionato in orizzontale: se lo scuotete delicatamente, potete ascoltare il rumore della sferetta che si muove all'interno. È un componente utile, ma non indispensabile pertanto può essere omesso a patto di cortocircuitare i relativi contatti sulla scheda; ovviamente il ribaltamento del robot non sarà riconosciuto. A titolo informativo, il controllo del ribaltamento è attivo solo con alcuni dei programmi di Robot In.

Sulla mainboard sono presenti anche due sensori di luce realizzati tramite due fotodiodi di tipo BPW40 (D1 e D14), che forniscono in uscita una tensione proporzionale alla quantità di luce che il robot incontra sulla parte frontale. Essendo due, è possibile identificare se la sorgente di luce si trova di più verso la destra o verso la sinistra rispetto alla marcia del robot. Questo permette di realizzare la funzione di rilevamento e inseguimento di fonti luminose.

Per quanto riguarda il rilevamento di oggetti sulla parte anteriore e posteriore, utilizziamo dei sensori all'infrarosso attivi.

Il fascio di raggi infrarossi viene generato dall'integrato NE555 (U4) in configurazione astabile che permette

di alimentare tre fotodiodi all'infrarosso (D8, D9, D10) alla frequenza di 38 kHz. Quando questo segnale luminoso (invisibile all'occhio umano) incontra un ostacolo, rimbalza e ritorna verso la scheda madre, dove viene rilevato da tre ricevitori modello TSOP1738 (U5, U6, U7) che provvedono ad inviare un segnale digitale al microcontrollore. Il funzionamento del sensore è ottimale quando l'oggetto rilevato offre delle superfici riflettenti gli infrarossi e presenta dei problemi con superfici che riflettono poco, come ad esempio oggetti di colore nero opaco, stoffe di colore scuro o materiali trasparenti.

Tra le funzioni descritte all'inizio, il robot è anche telecomandabile: a tal proposito sul connettore J5 a 25 poli è possibile inserire il modulo radio RTF-SAW dell'Aurel che permette al robot di inviare e ricevere messaggi e comandi; nella prossima puntata vedremo come telecomandare Robot In; per ora torniamo allo schema elettrico. La mainboard dispone anche di un piccolo buzzer (con cui invia delle segnalazioni acustiche) e del circuito di clock, realizzato con un quarzo da 4 MHz. Il connettore J6 permette la programmazione In-Circuit, a patto di disporre di un adeguato programmatore ICSP.

Realizzazione dell'elettronica

I circuiti stampati possono essere realizzati utilizzando i master scaricabili dal sito della rivista. La scheda sensori è monofaccia e non particolarmente ingombra di componenti, pertanto non presenta grosse difficoltà. Il PCB della motherboard, invece, è un doppia faccia con veramente tanti elementi e, di conseguenza, molto denso di piste sottili e piazzole piccole; si tratta pertanto di un circuito stampato piuttosto impegnativo, per quanto non impossibile. Durante il montaggio, un piccolo aiuto arriva dal fatto che ab-

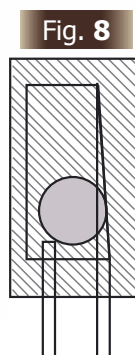
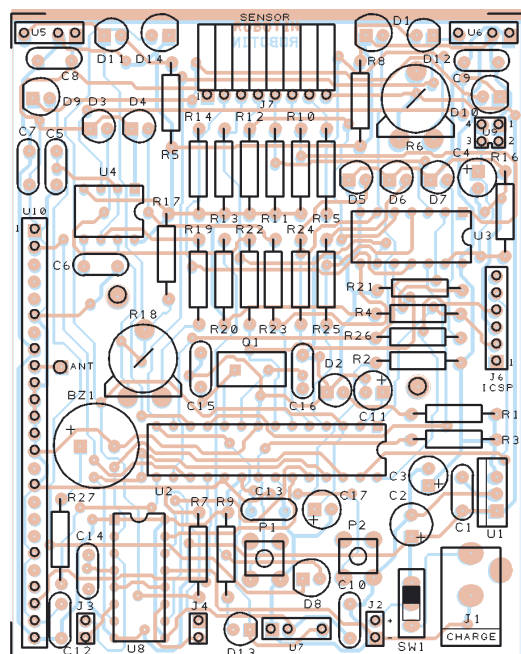


Fig. 8
Sensore di ribaltamento. Quando è in posizione verticale, la sfera metallica scende e chiude il contatto elettrico. Se posto sul fianco, la sfera apre il contatto elettrico.



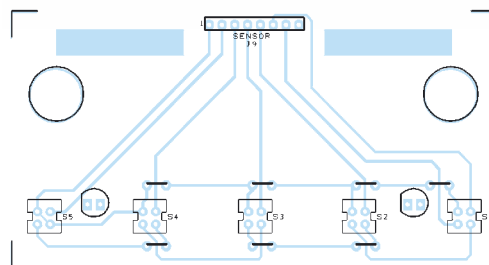
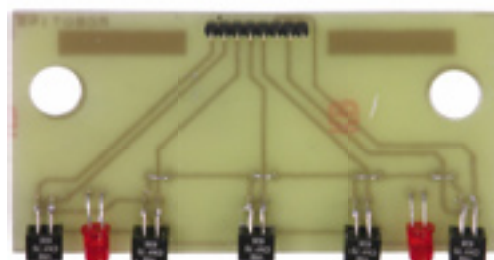
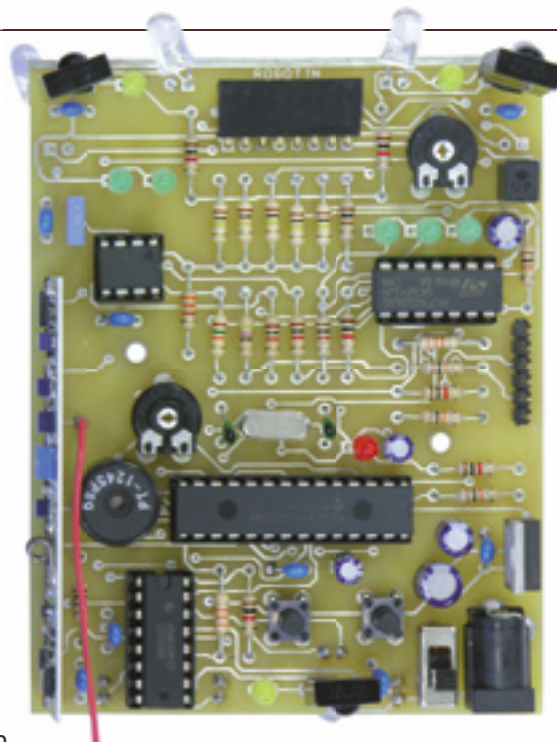
ELENCO COMPONENTI:

R1, R5, R8, R9, R22÷R26: 1 kohm
 R2÷R4, R16: 10 kohm
 R6: Trimmer MO 470 ohm
 R7, R21: 33 kohm
 R10÷R15, R27: 100 ohm
 R17: 3,3 kohm
 R18: Trimmer MO 4,7 kohm
 R19: 12 kohm
 R20: 47 ohm
 C1, C6÷C10: 100 nF multistrato
 C2: 100 µF 16 VL elettrolitico
 C3, C4: 10 µF 16 VL elettrolitico
 C11, C17: 10 µF 16 VL elettrolitico
 C5: 1 nF 63 VL poliestere
 C12÷C14: 100 nF multistrato
 C15, C16: 22 pF ceramico
 Q1: quarzo 4 MHz
 U1: LM2937
 U2: PIC16F876A (MF708)
 U3: 40106
 U4: NE555
 U5÷U7: TSOP1738
 U8: L293DNE
 U9: RBS040100
 U10: RTFSAW
 D1: Fotodiodo BPW40
 D2: led 3 mm rosso
 D3÷D7: led 3 mm verde

I circuiti che compongono Robot In non sono particolarmente complessi da montare. Occorre prestare attenzione alle saldature e al corretto posizionamento meccanico dei vari sensori, montati rialzati ed inclinati come illustrato nelle varie immagini.

D8÷D10: led IR 5 mm (L-53F3C)
 D11÷D13: led 3 mm giallo
 D14: Fotodiodo BPW40
 D15, D16: led 5 mm (vedi testo)
 BZ1: Buzzer con elettronica
 P1, P2: Microswitch
 SW1: Deviatore a slitta
 S1÷S5: CNY70

Varie:
 - Plug alimentazione
 - Strip femmina 8 poli 90°
 - Strip maschio 8 poli
 - Strip maschio 6 poli
 - Strip maschio 2 poli (3 pz.)
 - Zoccolo 4+4



- Zoccolo 7+7
- Zoccolo 8+8
- Zoccolo 14+14
- Portabatteria 4xAA
- Portabatteria 2xAA
- Ruote in plastica Pololu 0184 (2 pz.)
- Motoriduttore GM2 Pololu 0180 (2 pz.)
- Circuito stampato

biamo adottato solo componenti ad inserzione, ma occorre comunque fare molta attenzione alle saldature. Montando la scheda madre, noterete che alcuni componenti sono saldati sul lato piste e altri sul lato rame,

pertanto vi suggeriamo di affrontare la parte pratica con molta calma e con la dovuta cautela. Comunque abbiamo inserito nell'articolo molte illustrazioni, per aiutarvi anche con un supporto visivo, oltre che de-

scrittivo. Iniziate pure il montaggio partendo dalla scheda sensori, che è piuttosto semplice. I componenti sono montati tutti sul lato superiore e necessitano solo che ne rispettiate la polarità. Saldate anche i pon-

ticelli, ricavati da spezzoni di pin di resistenze e condensatori. Per la corretta saldatura dei CNY70 si faccia riferimento alla Figura 9, in cui è possibile notare che i quattro pin dei sensori devono essere allungati e piegati a 90°. Nella scheda sensori sono previsti due led non connessi elettricamente al circuito stampato. Essi servono per far scivolare il robot durante la marcia con il minimo attrito possibile: di fatto le loro teste appoggiano sulla superficie e devono pertanto sporgere dal circuito di circa 2-3 mm, come visibile in Figura 10. Ora potete dedicarvi alla scheda madre che, una volta ultimata, si presenta come visibile nel piano di montaggio alla pagina precedente. Abbiamo già detto che necessita di un po' di attenzione per quanto riguarda le saldature, ma impiega solo componenti a basso profilo (eccezion fatta per il modulo RF, che comunque va inserito su un connettore specifico). Potete iniziare a montare le resistenze, i condensatori, i trimmer, i led, gli zoccoli dei circuiti integrati, per terminare con i componenti più ingombranti quali il deviatore, il buzzer, il connettore per la ricarica delle batterie eccetera. Lasciate per ultimi i diodi e i sensori all'infrarosso, che devono essere tutti montati in modo abbastanza inusuale. In Figura 11 (frontale lato sinistro) potete vedere la giusta inclinazione di U5 e D14, ottenuta ruotando leggermente i componenti dopo averli saldati (nota: deve essere fatta la stessa cosa con U6 e D1, posti sul lato destro). La Figura 12, invece, mostra il diodo all'infrarosso D9 (D10 sul lato destro), montato anch'esso inclinato, ma sotto il PCB.

In Figura 13 è riportato il particolare della saldatura dei sensori posteriori e dei connettori J2, J3, J4. Il connettore J2 di alimentazione è del tipo polarizzato: in questo modo la spina di alimentazione non può essere invertita, impedendo di alimenta-

re la scheda in modo errato; dovrete fare però molta attenzione a saldarlo nella giusta posizione (quella indicata in figura). I connettori per i motori permettono invece di modificare la polarità di inserimento della spina. Connettori e diodo all'infrarosso (D8) sono montati sul lato saldature; il sensore U7 è invece montato sul lato componenti, come da Figura 13. Sempre dalla stessa figura, l'antenna per il modulo RF (visibile in rosso) può essere realizzata con un semplice spezzone di filo rigido della lunghezza di 17 cm.

Collaudo e taratura

Prima di assemblare tutte le parti meccaniche, è opportuno eseguire alcuni collaudi perché, se malauguratamente qualche componente fosse stato inserito in modo non corretto, ci sarebbe ancora la possibilità di intervenire.

Per prima cosa, se non fosse già stato fatto, eseguiamo un controllo visivo dello stato delle saldature e del corretto posizionamento dei componenti; quando sarete sicuri che il montaggio non presenta alcun errore, collegate la scheda sensori alla scheda madre tramite il connettore J7, come mostrato in Figura 14, dove potete anche notare la corretta posizione del modulo RF Aurel.

Preparate i cavi di collegamento per i due pacchi portabatterie ponendo in serie le due clips per batteria a

Fig. 13 - I connettori (J2, J3 e J4) e il diodo D8 sono montati sul lato saldature. J2 (bianco) va montato col riferimento a destra. Il sensore IR U7 è montato sul lato compo-

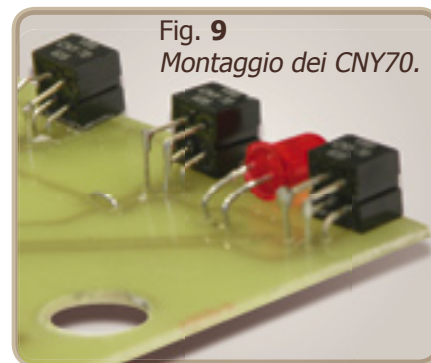


Fig. 9
Montaggio dei CNY70.

Fig. 10 - La testa dei led rossi funge da piedino a basso attrito e sporge di 3 mm dal bordo del PCB.

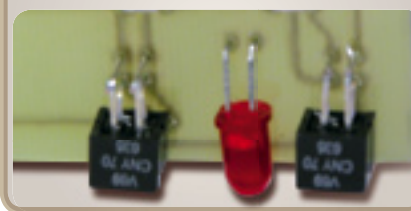
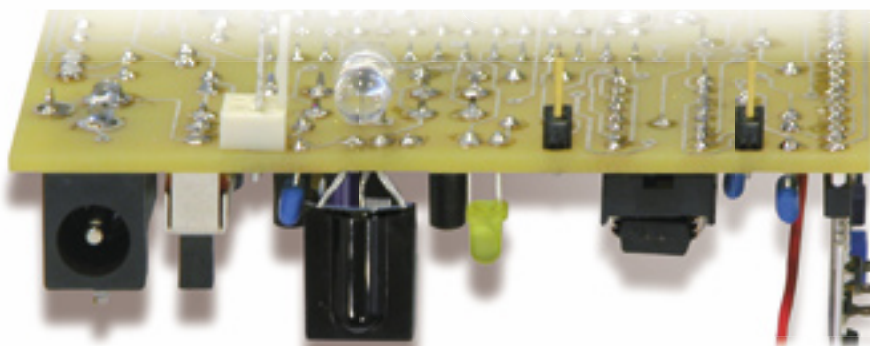


Fig. 11 - U5, U6, D1 e D14 vanno montati leggermente inclinati, per ampliare il raggio di rilevamento degli ostacoli.



Fig. 12
D9 e D10 vanno montati inclinati, ma sotto il PCB della scheda madre.



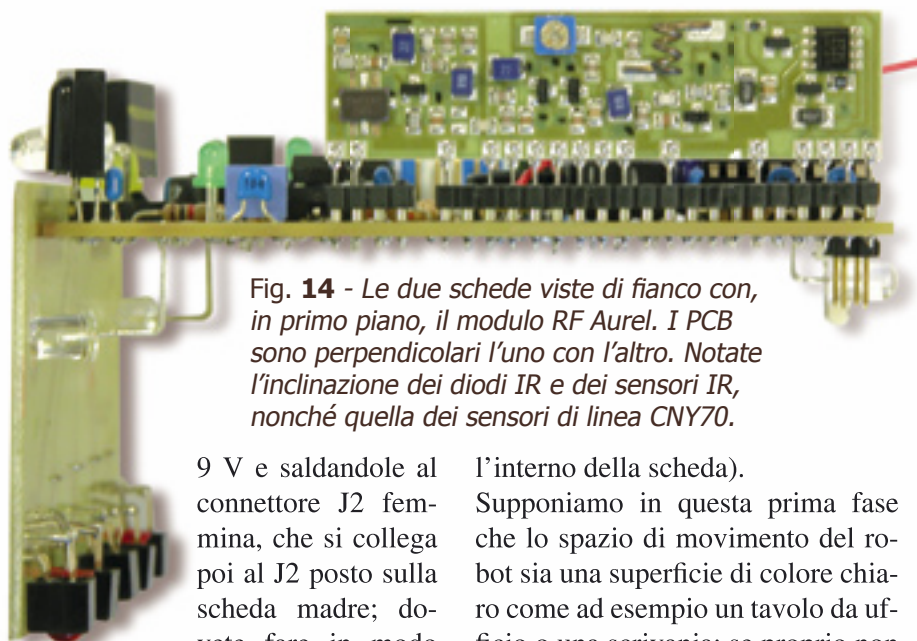


Fig. 14 - Le due schede viste di fianco con, in primo piano, il modulo RF Aurel. I PCB sono perpendicolari l'uno con l'altro. Notate l'inclinazione dei diodi IR e dei sensori IR, nonché quella dei sensori di linea CNY70.

9 V e saldandole al connettore J2 femmina, che si collega poi al J2 posto sulla scheda madre; dovete fare in modo che il terminale negativo (filo nero) faccia contatto sul pin rivolto verso l'esterno del connettore J2 della mainboard. Innestate quindi una delle clips sul pacco portabatteria a quattro posti: il risultato ottenuto è visibile in Figura 15.

Connettete anche il secondo portabatterie, inserite le sei pile di alimentazione e innestate il connettore sullo stampato (J2) della scheda madre avendo prima controllato che l'interruttore sia in posizione OFF (levetta rivolta verso l'esterno della scheda). Per il momento non inserite né il PIC, né l'integrato L293D e neppure il modulo radio: a questo punto accendete il circuito portando l'interruttore a slitta su ON (verso

l'interno della scheda).

Supponiamo in questa prima fase che lo spazio di movimento del robot sia una superficie di colore chiaro come ad esempio un tavolo da ufficio o una scrivania; se proprio non fossero disponibili, usate un foglio di cartone di colore bianco reperibile in cartoleria.

Per prima cosa controllate D2, che dovrebbe essere acceso ad indicare presenza tensione; se disponete di un tester potete misurare la tensione tra i pin di alimentazione degli integrati che deve essere di 5 V. Tenendolo in mano, posizionate il robot sopra la superficie di lavoro e verificate i sensori di linea (i CNY70 di Figura 9 e 10) controllando lo stato dei diodi led D3-D7: in questa condizione (superficie riflettente) devono essere accesi tutti. Ponete uno alla volta i sensori sopra del nastro isolante da elettricista di colore nero oppure su una strisciolina di cartone

nero e controllate che il relativo led si spenga. Potete eventualmente regolare la sensibilità dei sensori agendo sul trimmer R6 e trovando la posizione in cui i led sono accesi con superficie riflettente e spenti con superficie non riflettente.

La verifica dei sensori per gli oggetti è molto semplice e consiste nell'avvicinare alternativamente ad ogni

sensore U5, U6, U7 un oggetto, delle dimensioni del robot, con superficie chiara, dalla distanza 30 cm fino ai 5 cm: dovreste trovare un punto in cui si accendono i led D11, D12, D13. Regolate ora il trimmer R18 affinché sul pin 3 dell'integrato U4 ci siano 38 kHz. Se non disponete di un frequenzimetro, regolate (per tentativi) il trimmer R18 sino a quando troverete una situazione soddisfacente in cui i led si accendono quando l'oggetto rilevato dal sensore si trova ad una distanza inferiore a 10 cm. La distanza di riconoscimento dipende dalle caratteristiche riflettenti dell'oggetto e dalle sue dimensioni; è comunque possibile regolare leggermente la portata modificando il valore della resistenza R20: aumentando questo valore si diminuisce la portata. Vi consigliamo di lasciare il valore da noi scelto (47 ohm), ma, qualora vogliate modificarlo, non scendete al di sotto dei 33 ohm altrimenti correte il rischio di bruciare tutti i diodi all'infrarosso. Per ridurre facilmente la portata senza variare alcuna resistenza, potete agire sul trimmer R18, mandando l'oscillatore a lavorare leggermente fuori frequenza: attenzione però a non esagerare altrimenti il sensore attivo smette di funzionare.

A questo punto spegnete il circuito, inserite il PIC, l'integrato L293D e la scheda radio. Quest'ultima dovrà essere inserita nello zoccolo facendo in modo che i componenti siano rivolti verso l'esterno del robot (fate riferimento alla Figura 14).

Relativamente al PIC, indipendentemente dal fatto che lo abbiate programmato voi con il firmware ROBOTIN.HEX (scaricabile dal sito della rivista) o che lo abbiate acquistato già programmato, qui noi diamo per scontato che sia pronto per l'uso. Inserite negli appositi connettori anche i cavi di alimentazione dei due motoriduttori, non curandovi troppo della polarità.

Dopo aver alimentato il circuito ➤

Fig. 15 - Clips 9 V in serie e pacco portabatteria.

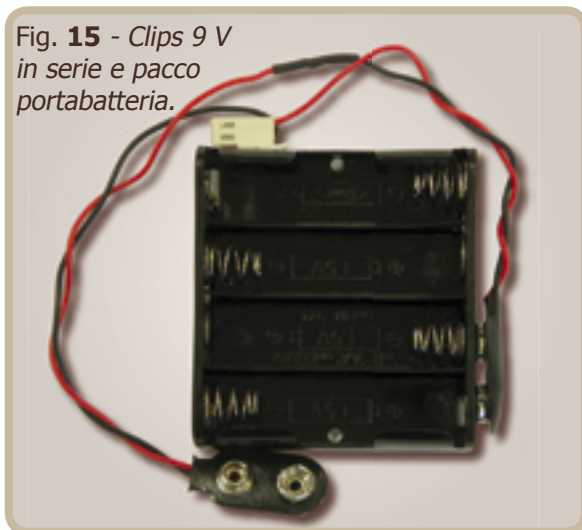


Fig. 17 - Posizionare del biadesivo sulla scheda sensori.

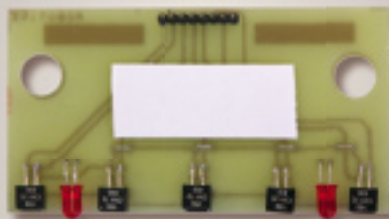


Fig. 16

Il disegno chiarisce come vanno montati i vari elementi del robot.

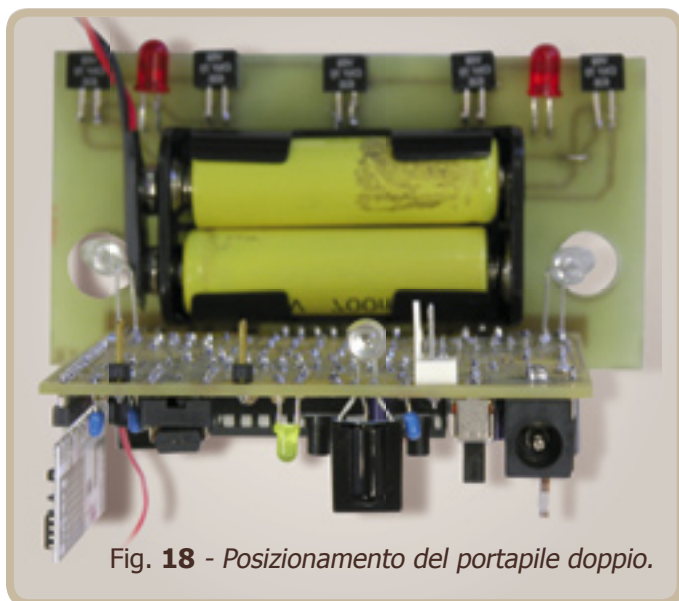
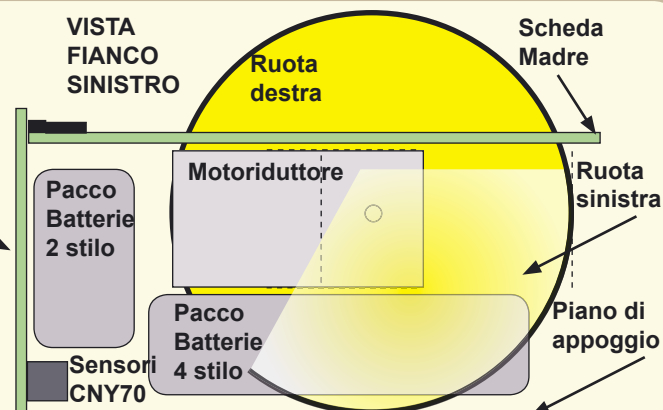
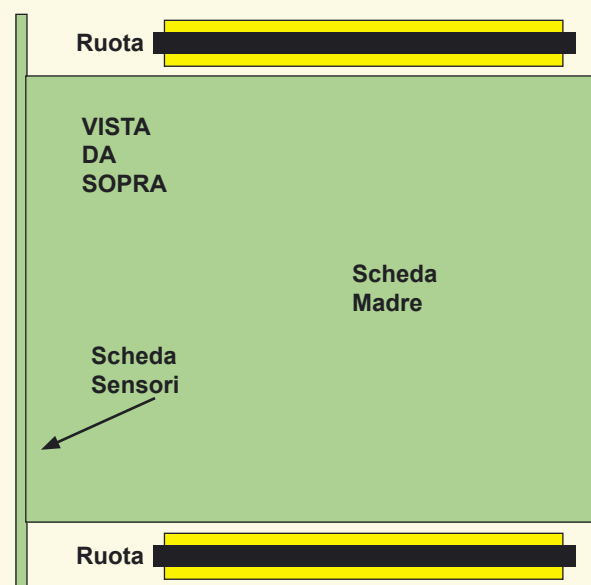


Fig. 18 - Posizionamento del portatile doppio.



VISTA DI FRONTE

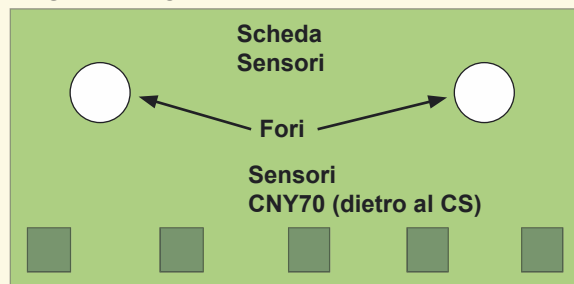


Fig. 20 - Posizionare il biadesivo sui motoriduttori.

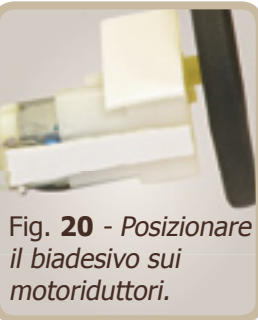


Fig. 19 - Condensatori antistatico.



Fig. 21 - I cavi dei motori hanno lunghezze differenti.



portando il deviatore nella posizione ON, udite alcune note acustiche ad indicare che il PIC e il buzzer sono funzionanti. Premendo il pulsante P1 (quello posto a destra sulla mainboard vista da dietro) si attiva il motore di destra con verso di rotazione in avanzamento, per cui potete verificare contemporaneamente tanto il corretto funzionamento del circuito di controllo motore quanto la polarità del cavetto: se il motore non gira in avanzamento, invertite la posizione del connettore. Premendo

il pulsante P2 (quello a sinistra) si deve attivare il motore di sinistra con gli stessi criteri di quello di destra. Se i due motori sono invertiti di posizione, scambiate la spina collegata su J3 con quella collegata su J4. Ribaltate ora il robot e ponetelo su di un fianco sopra il tavolo: il buzzer deve emettere una nota acustica per indicare l'avvenuto rovesciamento. Assicuratevi che la sferetta all'interno del sensore sia libera di muoversi: è facile sentirla scuotendo leggermente il robot.

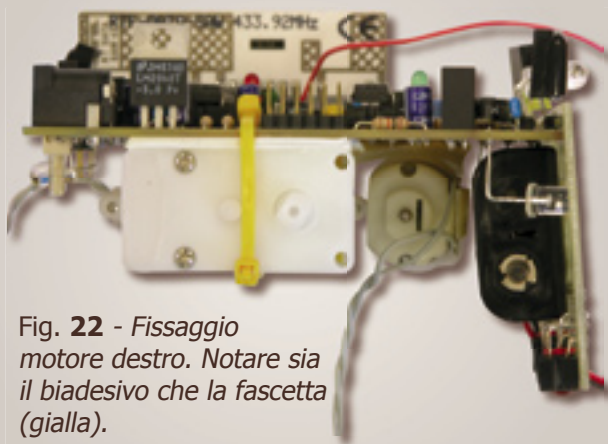


Fig. 22 - Fissaggio motore destro. Notare sia il biadesivo che la fascetta (gialla).

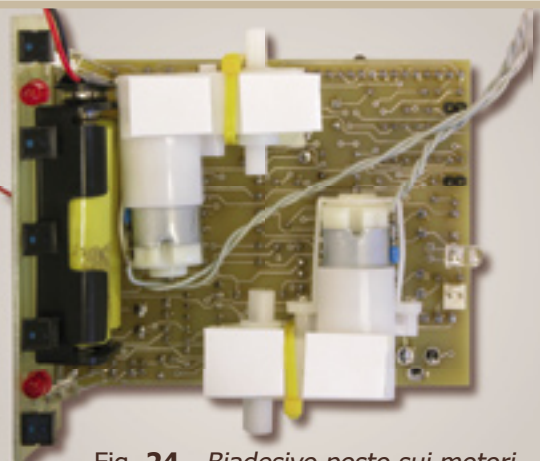


Fig. 24 - Biadesivo posto sui motori.

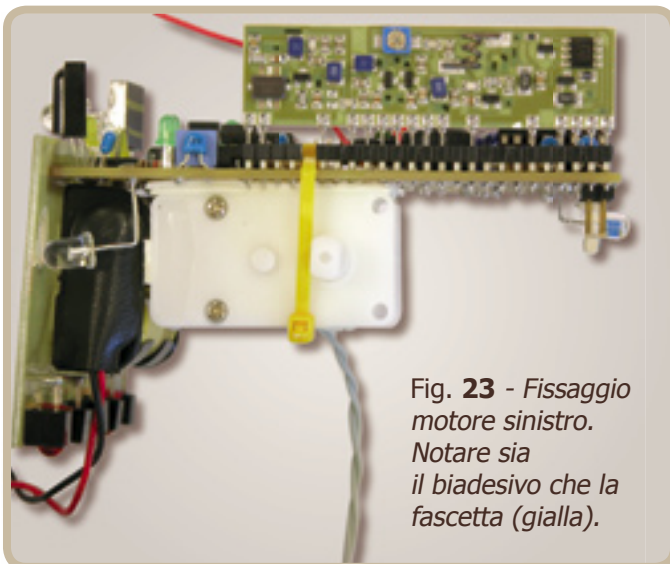


Fig. 23 - Fissaggio motore sinistro. Notare sia il biadesivo che la fascetta (gialla).

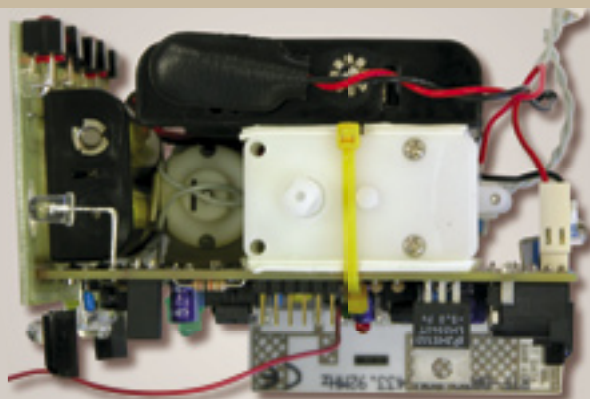


Fig. 25 - Montaggio del pacco batterie principale.

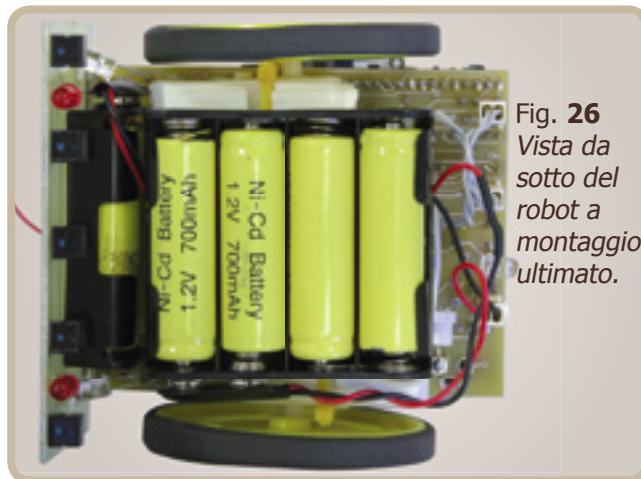


Fig. 26
Vista da sotto del robot a montaggio ultimato.

A questo punto il collaudo elettrico è terminato e potete procedere con il fissaggio dei motori e delle batterie.

Montaggio meccanico

In Figura 16 è riportata in modo schematico la disposizione dei vari elementi meccanici a montaggio ultimato; tenete in considerazione che in questo capitolo i circuiti stampati vengono considerati come parti meccaniche.

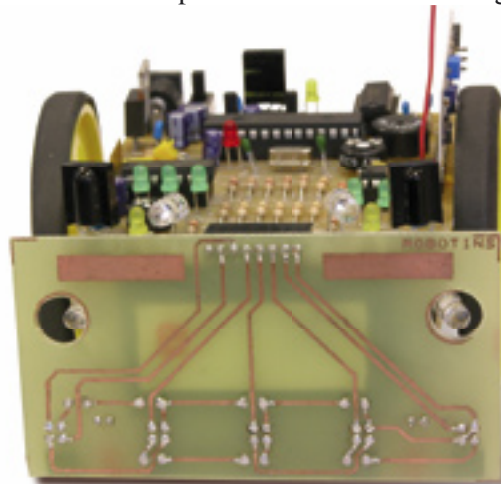
Il portabatterie per due stilo funge da blocco di connessione e fissaggio tra i motori e la scheda sensori, e conferisce solidità all'insieme che, per essere assemblato, necessita solamente di nastro biadesivo spesso e fascette da elettricisti (del tipo 2,4 mm x 100 mm) reperibile anche nei supermercati. Descriviamo ora la

sequenza di montaggio dei vari elementi che compongono il robot. Cominciate col mettere del nastro biadesivo sul circuito dei sensori come indicato in Figura 17.

Fissate la mainboard alla scheda sensori e posizionate la esattamente a 90°. Nel punto in cui il circuito stampato dei sensori tocca il circuito stampato della mainboard ci sono delle piazzole che permettono la saldatura tra di esse. Questa operazione è facoltativa in quanto l'insieme risulta comunque sufficientemente solido, ma così facendo

il robot sarà più robusto nel caso di urti o sollecitazioni; eventualmente potete saldare tra loro le piazzole in un secondo momento a seconda delle necessità. In ogni caso assicuratevi che i due circuiti stampati siano posizionati (per quanto possibile) a 90° tra di loro. Fissate il pacco portabatterie per due stilo sul circuito ➤

dei sensori come in Figura 18 quindi passate ai motori. Questi dispongono ognuno di un cavetto di alimentazione (Figura 21) e devono essere dotati di due condensatori da 100 nF. Ogni condensatore va saldato tra il terminale di alimentazione e la carcassa metallica del motore. Questo permette di attenuare i disturbi generati dal contatto delle spazzole durante la rotazione. Per facilitare l'operazione rimuovete la calotta che fissa il motore al riduttore svitando le due piccole viti autofilettanti: il risultato è visibile in Figura 19. Applicate sul motore un pezzo di biadesivo ed



eventualmente rimuovete la piccola sporgenza in plastica che potrebbe fare spessore sul portabatteria dei sensori (Figura 20). A questo punto fissate il motore di destra alla scheda madre come visibile nella Figura 22, utilizzando sia il nastro biadesivo che la fascetta; quest'ultima deve essere fatta passare nell'apposito foro del circuito stampato, per fis-

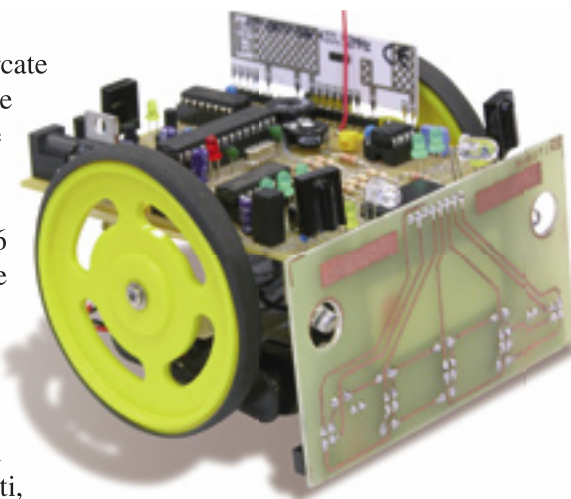
sare saldamente il motore: cercate di allineare al meglio il motore alla mainboard. La fascetta deve passare esattamente sopra i pin 3,4,5 del connettore J5 senza interferire con il pin 2 e il pin 6 occupati dalla scheda radio. Fate la stessa cosa con il motore di sinistra (Figura 23). Dopo aver posizionato dell'altro biadesivo sui due motori (Figura 24), fissate, a questi ultimi il portabatteria a quattro posti, appoggiandolo al portabatteria della scheda sensori, come illustrato in

Figura 25; se lo ritenete opportuno, potete utilizzare una fascetta molto lunga per fissare l'intero portabatterie al robot. Dalla Figura 26, infine, notate come e dove devono essere collegati i vari fili dei motori e delle batterie. A lavoro ultimato, il risultato è visibile nelle tre foto riportate in quest'ultima pagina.

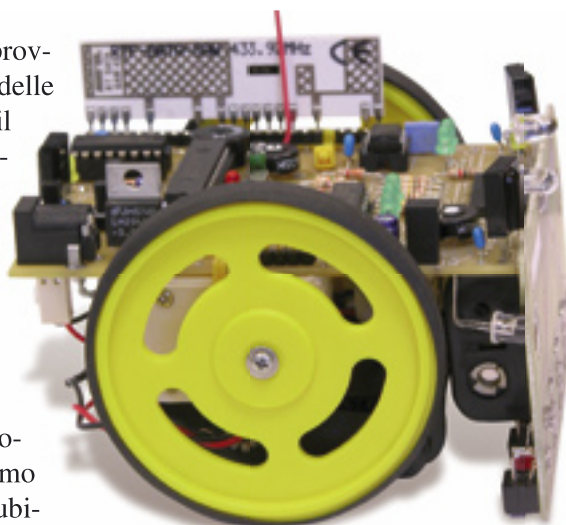
Ora, se necessario, provvedete alla ricarica delle batterie utilizzando il connettore J1 e lasciando il robot spento. Se non disponete di un apposito caricabatteria, potete realizzare quelli presentati più in passato su *Elettronica In*.

Note conclusive

Ora che avete a disposizione il robot montato, siamo certi vorreste giocarci subi-



to: beh... non fatevi prendere dal "giovanil entusiasmo", abbiate un po' di pazienza e attendete l'uscita del prossimo fascicolo, in cui vi spiegheremo come utilizzare questo gioiellino robotico casalingo. L'appuntamento è quindi tra un mese in edicola: ci raccomandiamo, non mancate.



per il

MATERIALE

Il materiale utilizzato in questo progetto è facilmente reperibile in qualsiasi negozio di elettronica. I master dei circuiti stampati ed il firmware in formato esadecimale (MF708.hex) col quale programmare il micro sono scaricabili gratuitamente dal sito della rivista (www.elettronica.in.it). I motoriduttori, le ruote, i pacchi portabatteria, il modulo radio ed i sensori all'infrarosso possono essere acquistati on-line sul sito www.futurashop.it.

**Il materiale va richiesto a: Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)
Tel: 0331-799775 ~ Fax: 0331-778112 ~ <http://www.futurashop.it>**

Demoboard GPS/GSM: adattatore per GPS

di Boris Landoni

Permette di collegare alla demoboard descritta il mese scorso i moduli GPS di nuova generazione ET312, ET333, FV-M8 ed FV-M11. Dispone di due connettori aggiuntivi a cui è possibile collegare altri modelli di ricevitori satellitari. Prevede, inoltre, la possibilità di collegare un'antenna esterna e l'eventuale batteria di back-up.

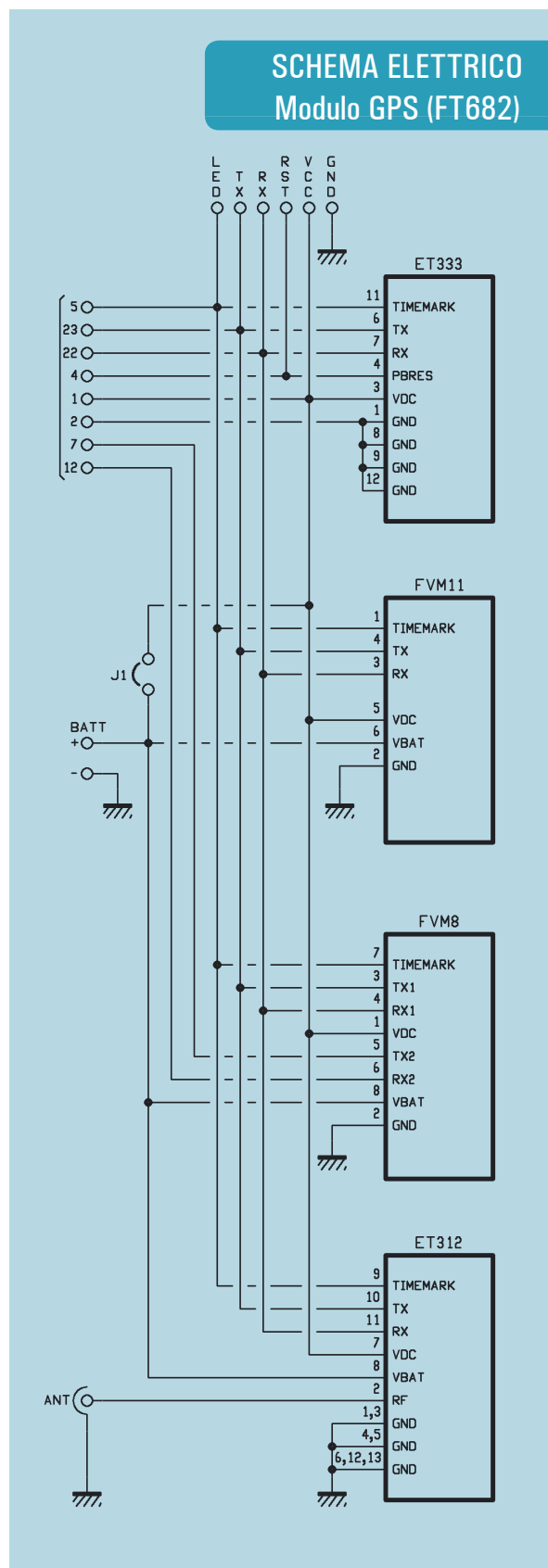
Nel fascicolo precedente abbiamo presentato una demoboard sviluppata appositamente per testare applicazioni con moduli GSM, GPS o combo (GPS+GSM) e progettata in modo da poter essere adattata a molti apparati, anche abbastanza diversi l'uno dall'altro; ciò è stato possibile perché la scheda dispone di periferiche comuni a tutti e che, sebbene in modo e in quantità differente, ognuno di essi può utilizzare. Quindi è stato sufficiente portare tutte le linee disponibili ad un connettore universale, formato da due strip, nel quale ogni modulo potrà essere inserito previo montaggio

su una scheda adattatrice ad esso dedicata: tale scheda, completamente passiva, si comporta da "convertitore di pin-out" e collega i vari pin disponibili sul connettore del dispositivo in prova ai pin di funzione equivalente posti sui due connettori strip a passo 2,54 mm. Nel fascicolo 120 abbiamo già descritto l'hardware e il funzionamento della demoboard e degli adattatori per i moduli GM862 Telit (cellulare GSM/GPRS) Wavecom Q2501 e Q2686 (combo); adesso introduciamo un nuovo adattatore, particolarmente importante perché pensato per una classe di ricevitori GPS di recente produzione e che, in virtù delle ➤

La parte probabilmente più interessante di tutta la demoboard è rappresentata dalle interfacce di comunicazione seriali, perché tutti i moduli, siano essi cellulari, combo o semplici GPS, ne hanno almeno una: la scheda è equipaggiata con ben tre converter TTL/RS232 che permettono di interfacciare i moduli, che comunicano tipicamente mediante porte seriali a livello TTL, con le comuni COM dei PC. Scendendo più in dettaglio troviamo tre seriali, due delle quali “ridotte”. Le porte differiscono essenzialmente per il fatto che quella chiamata SERIAL1 è completa, nel

La porta seriale completa è stata prevista perché solitamente i moduli che integrano un cellulare GSM/GPRS hanno al loro interno un modem e questi, normalmente, rende accessibili e utilizza tutte le linee della porta RS232.

Bene, adesso passiamo a vedere in cosa consiste l'adattatore per moduli GPS: si tratta di una basetta provvista, sui lati lunghi, di due strip di tipo maschio da 16 poli ciascuna, che può essere innestata sulla demoboard; l'adattatore dispone dei quattro connettori a cui possono essere collegati semplici ricevitori GPS quali l'ET333, l'ET312,



l'FV-M8 e l'FV-M11. Siamo riusciti a utilizzare una sola scheda perché, essendo questi quattro ricevitori abbastanza simili per quanto riguarda le

I GPS supportati dall'adattatore

In questo riquadro abbiamo voluto proporre tutte le possibili situazioni per cui l'adattatore è stato pensato.

Figura A1. Mostra l'adattatore su cui non è stato montato alcun ricevitore satellitare: in questo modo, è possibile collegare qualsiasi modulo GPS per il quale non abbiamo previsto un connettore specifico, attraverso le pin strip a tre pin e passo 2,54 mm visibili a destra dei connettori bianchi.

Figura A2. Il GPS modello ET312 può essere montato solo ed esclusivamente in una posizione, in quanto la disposizione dei suoi pin sui due lati non è simmetrica.

Figura A3. Il modulo FV-M8 di SANAV deve essere collegato all'apposito connettore tramite un cavetto di adattamento. Anche in questo caso il rischio di inserzioni errate non esiste perché il connettore presenta un solo verso di inserimento.

Figura A4. Il modulo GPS FV-M11 è molto simile all'FVM8: anch'esso va collegato tramite lo stesso cavetto di adattamento dotato di chiave meccanica e usato per il GPS FVM8.

Figura A5. L'ultimo dei ricevitori previsti sull'adattatore è l'ET333, che deve essere collegato attraverso un connettore a pin strip a due linee, con passo 1 mm. Questo connettore, a differenza degli altri, non è "polarizzato" pertanto il GPS deve essere innestato sulla demoboard esattamente come visibile in figura.

Fig. A1

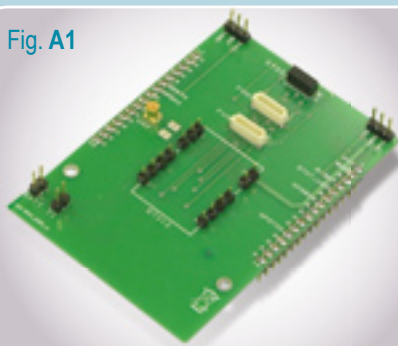


Fig. A2

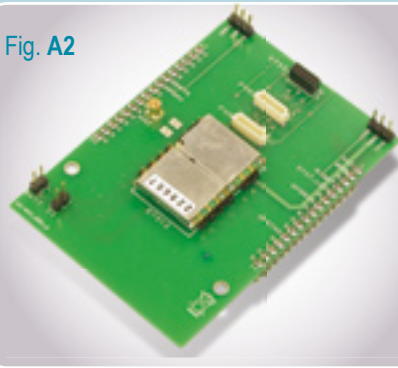
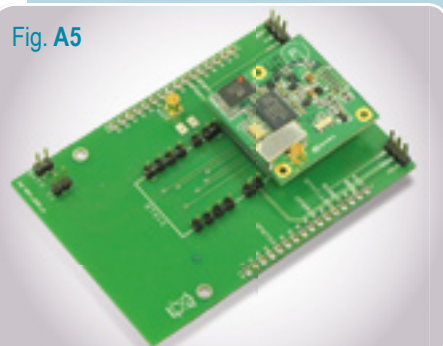
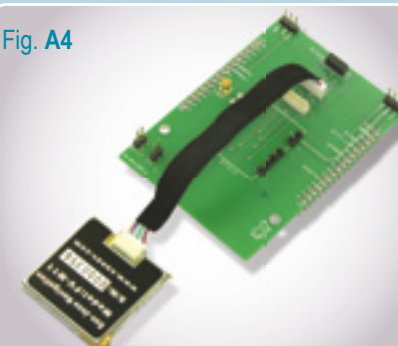


Fig. A3



Fig. A4



Lo schema elettrico visibile in queste pagine ci mostra tutte le connessioni del caso: come vedete, i moduli compatibili hanno in comune almeno le linee TIME-MARK (si collega normalmente al led di segnalazione dell'attività del ricevitore) TX, RX, Vdc e GND; esse vengono trasportate in parallelo ai corrispondenti contatti delle strip.

Vi sono moduli che hanno due porte di comunicazione seriale: è il caso dell'FM-V8 (prodotto dalla SANAV); in questo caso, TX1 ed RX1 sono in parallelo alle TX ed RX degli altri connettori, mentre TX2 ed RX2 vanno per conto proprio ai due pin delle strip collegati alla seconda seriale.

Un discorso analogo si può fare per la linea PBRES, che esiste solo nell'ET333, del quale rappresenta il reset esterno; oppure per Vbat, che costituisce un'alimentazione supplementare, ossia quella della batteria tampone esterna richiesta da moduli quali l'FM-V11, l'FM-V8 e l'ET312 per mantenere il clock. Nel caso specifico, notate che la scheda adattatrice prevede un connettore bipolare (due pin a passo 2,54 mm...) al quale collegare l'eventuale batteria.

Per estendere le possibilità d'impiego, ossia per poter sfruttare l'adattatore anche per provare altri moduli, è stato previsto un connettore strip che raggruppa le linee principali: è quello in alto nello schema e contiene LED (diodo luminoso di segnalazione pilotato da TIMEMARK) TX, RX, RST, Vcc e GND. Tale connettore ha la funzione di porta d'accesso alla demoboard. Certo, moduli con connessioni diverse da quelle previste per i quattro elencati non si possono fissare sulla basetta, ma almeno è possibile collegarli facilmente.

I ricevitori previsti hanno tutti l'antenna GPS interna ad eccezione del-

linee di comunicazione, è stato possibile realizzare facilmente un unico adattatore provvisto dei connettori specifici per ogni modello, connetto-

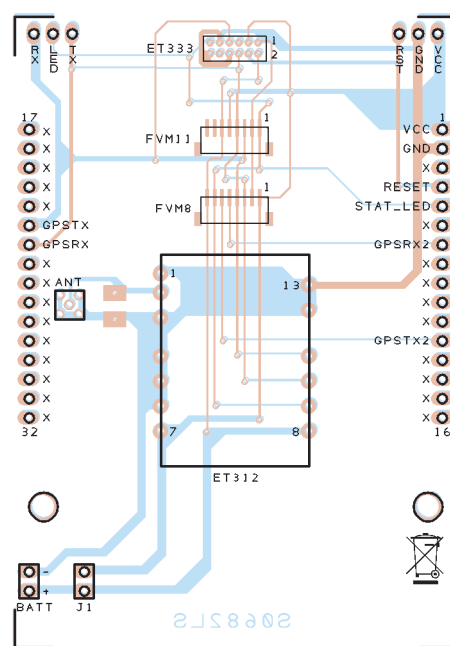
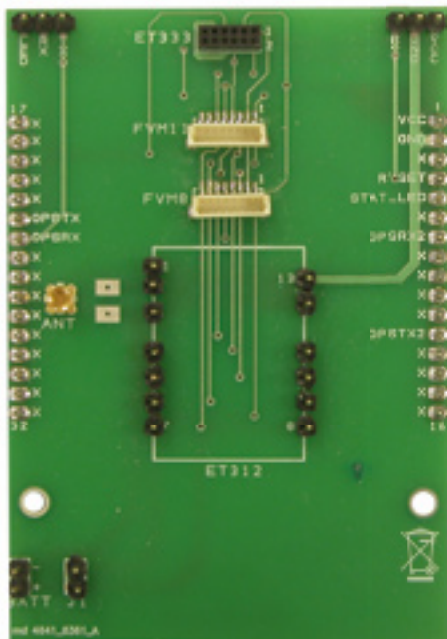
ri che condividono i segnali elettrici comuni. Al solito, le linee che servono all'emulazione sulla demoboard vengono portate alle strip a 16 poli.

La realizzazione di questo adattatore è impegnativa solo per quanto concerne il montaggio dei connettori FV-M8 e FV-M11, piuttosto delicati perché in formato SMD. Nota: il connettore ET333 viene fornito insieme al GPS.

ELENCO COMPONENTI:

Varie:

- Strip maschio 3 pin (2 pz.)
- Strip maschio 16 pin (2 pz.)
- Strip maschio 2 pin (3 pz.)
- Strip maschio 1 pin (11 pz.)
- Connettore SMD passo 1 mm (2 pz.)
- circuito stampato



l'ET312, per il quale è stata prevista una connessione terminante su una piazzola (ANT) alla quale dovrà essere saldato il cavetto coassiale dell'antenna esterna; naturalmente si parla dell'antenna GPS (che, all'occorrenza, dovete procurare) e non di quella GSM integrata nella demoboard. Nelle piazzole ANT è possibile stagnare direttamente una presa coassiale miniatura di tipo MMCX. Concludiamo la descrizione dello schema elettrico con il ponticello J1 che è stato previsto per far funzionare correttamente i moduli FV-M8, FV-M11 ed ET312 quando si intende effettuare la simulazione sulla demoboard senza collegare la batteria del clock ai punti BATT; in

tal caso, chiudendo il jumper la linea corrispondente (Vbat) viene alimentata dalla stessa Vcc.

Realizzazione pratica

A questo punto spendiamo qualche parola anche per quanto concerne il montaggio: la basetta è a doppia faccia con fori metallizzati, ma può essere preparata anche artigianalmente avendo cura, durante il montaggio, di collegare le vie (piazzole che fanno cambiare il lato ad una pista e che normalmente sono rappresentate dai fori metallizzati) saldando degli spezzi di reoforo o i reofori dei componenti stessi. Per montare i connettori riservati ai ricevitori GPS (sono quattro: tre a 12 pin per

ET333, FV-M8 ed FV-M11, più uno da 14 pin per l'ET312) usate un saldatore (al massimo da 30 W) a punta molto sottile e della lega saldante del diametro di 0,5 mm; posizionate il componente in modo da centrare i terminali nelle piazzole corrispondenti, poi stagnate un contatto per lato e proseguite saldando sempre un pin per lato, alternativamente, così da dare il tempo al materiale di raffreddarsi. Per la batteria esterna, l'espansione e il GPS modello ET312 montate delle file di strip a passo 2,54 mm seguendo quanto riportato sul piano di montaggio. Non dimenticate le due strip a 16 pin, montate sul lato piste inferiore, necessarie per la connessione con la demoboard.

per il

MATERIALE

Tutto il materiale utilizzato sia nella demoboard che nell'adattatore presentato in queste pagine è facilmente reperibile in qualsiasi negozio di componenti elettronici. I master delle schede di adattamento possono essere scaricati gratuitamente dal sito della rivista (www.elettronica.in.it).

I moduli GPS compatibili costano rispettivamente Euro 56,00 (ET312), Euro 56,00 (ET333), Euro 65,00 (FV-M8), ed Euro 68,00 (FV-M11). Tutti i prezzi sono comprensivi di IVA.

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

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

Controllo con joystick per braccio robotizzato

di Francesco Doni

Un programma in ambiente Windows interpreta i movimenti ed i comandi provenienti da un comune joystick; quindi li traduce in istruzioni seriali che invia alla scheda di controllo del braccio robotizzato. Quest'ultimo, pertanto, si sposta seguendo lo stato della leva del joystick.

Qualche numero addietro (fascicolo 116) abbiamo presentato un braccio robotizzato realizzato con servocontrolli da modellismo e capace di cinque livelli di movimento. Il controllo, all'epoca un po' spartano, era stato affidato al personal computer tramite un programma specifico che faceva uso del mouse. In questo numero, invece, vogliamo proporre un altro tipo di controllo per la movimentazione in cui il mouse è stato rimpiazzato dal joystick: siamo così riusciti ad ottenere un controllo decisamente più preciso di quello che potevamo ottenere con dei semplici clic. Il movimento della

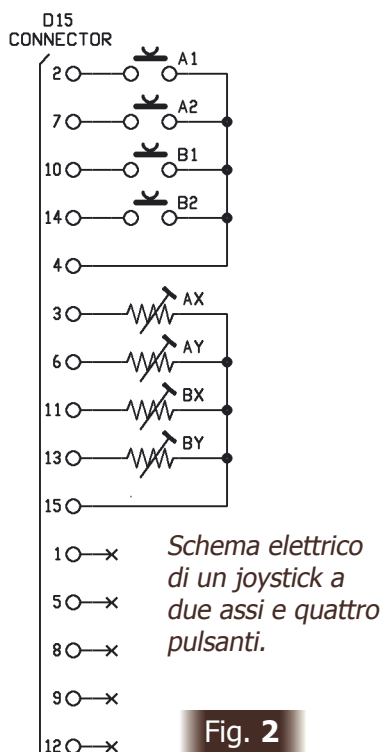
leva del joystick, infatti, viene replicato sul braccio robotizzato permettendo una maggiore flessibilità dei movimenti. A titolo informativo, sappiate che il joystick è il sistema di controllo preferito per i bracci robotizzati ad uso professionale come ad esempio quelli che si vedono sui batiscafi oppure sulle navette spaziali, situazioni queste dove l'uomo non è in grado di operare senza l'ausilio di supporti meccanici e in cui è indispensabile il massimo livello di controllo e precisione.

Probabilmente, l'utilizzo di un sistema di controllo dato come questo farà sorridere i giovanissimi, abituati ➤

ai moderni ed ergonomici “game pad”. Tuttavia riteniamo che questo articolo rechi in sé un elevato valore didattico, in quanto insegna come utilizzare una periferica di gioco per un’applicazione di “più nobile” livello.

Il joystick

Questa periferica è sicuramente la più utilizzata nei videogiochi: attualmente ne esistono di svariate forme e dimensioni, con comandi spesso dedicati a particolari funzioni di gioco. Alcune hanno la forma di una pedaliera, altre hanno la forma di un volante, ma tutte sono accomunate dal fatto di utilizzare gli stessi segnali elettrici e la stessa porta di comunicazione con il PC. Ogni joystick integra alcuni contatti meccanici, che si presentano sotto forma di pulsanti, funzionanti nello stato ON/OFF e di solito utilizzati per funzioni di sparo (comando) di varia natura. Il controllo vero e proprio, però, è effettuato dai “comandi analogici” ricavati da due (o più) potenziometri disposti l’uno perpendicolarmente rispetto all’altro: essi forniscono un segnale elettrico (riconosciuto dal PC) direttamente proporzionale allo spostamento sui due assi (X e Y) di una leva. Per meglio comprendere questo concetto, immaginiamo che il nostro joystick debba controllare la velocità di una navetta spaziale in un videogioco. Se usassimo un pulsante



te sarebbero possibili due soli casi: a pulsante rilasciato la navicella risulterebbe ferma, mentre premendo il pulsante la navicella avanzerebbe ad una specifica velocità che non potremmo modificare. La stessa cosa accadrebbe per le virate a destra e a sinistra. Con il controllo analogico, invece, è possibile definire la velocità della navetta in funzione della posizione della leva di comando; infatti, ad ogni sua posizione, corrisponde un determinato valore della resistenza (quindi della tensione sul cursore) del potenziometro: tale valore viene opportunamente interpretato dal circuito di controllo del PC che, in questo modo, può conoscere l’effettiva posizione della leva ed assegnare di conseguenza la velocità alla navetta del videogioco.

Dal semplice esempio appena esposto, si deduce che la nostra periferica genera sia dei segnali di tipo ON/OFF che analogici, questi ultimi utilizzabili per identificare una coordinata X-Y. Di fatto, il PC deve disporre di una sua porta joystick: essa ha il connettore simile a quello della seriale RS-232 ma è composto

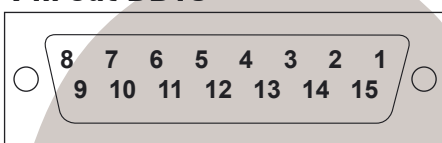
da 15 pin (DB15). In Figura 1 potete vedere l’assegnazione “numero pin / funzione” mentre la Figura 2 mostra i collegamenti elettrici di una tipica periferica che, come potete notare, è veramente semplice in quanto costituita da quattro pulsanti e quattro potenziometri del valore di 500 kohm. In alcuni casi, in serie a questi potenziometri se ne pone uno da 100 kohm necessario in fase di gioco alla funzione di trimmer per calibrare la posizione neutra. Notate che il connettore DB15 è in grado di accettare due joystick (A1-A2-AX-AY e B1-B2-BX-BY) che quindi possono operare come JOY-1 o JOY-2. La disposizione meccanica degli elementi elettrici dipende dall’utilizzo che se ne vuole fare: ad esempio, un potenziometro potrebbe essere associato ad un piccolo volante da usare per lo sterzo di un veicolo mentre un altro potrebbe fungere da acceleratore di una pedaliera.

Nel caso del controllo di un velivolo, si utilizza una disposizione a cloche sfruttando due potenziometri disposti a 90° e azionati da un’unica leva posta in verticale (questa è la nostra esatta situazione).

Il costo ridotto di queste periferiche di gioco non giustifica la realizzazione DIY della parte meccanica, ma nulla vieta di realizzare in casa



Pin-out DB15



- | | |
|--------------|--------------|
| 1. XY1 (+5V) | 9. XY2 (+5V) |
| 2. Switch 1 | 10. Switch 3 |
| 3. X1 | 11. X2 |
| 4. Ground | 12. Ground |
| 5. Ground | 13. Y2 |
| 6. Y1 | 14. Switch 4 |
| 7. Switch 2 | 15. C.C. |
| 8. N.C. | |

Porta joystick standard.

Fig. 1



Fig. 4

un controllo che disponga di specifiche caratteristiche. Per il nostro progetto, utilizziamo un semplice controllo a cloche con due assi e due pulsanti, visibile in Figura 3.

Per poter utilizzare questo tipo di periferica in ambiente Windows, è necessario effettuare una configurazione iniziale. Qui noi facciamo riferimento al sistema operativo Windows XP, ma la procedura è molto simile anche per le altre ver-

sioni di Windows; rammentiamo che il PC deve essere dotato della porta joystick, come visibile in figura 4. Talvolta può capitare che questa porta sia presente sulla scheda audio, in cui i pin 1, 5, 8, 9 e 12 servono come ingresso MIDI (sistema usato da vecchie tastiere musicali per comunicare alla scheda audio del PC la combinazione di tasti premuta). Per la configurazione della periferica dovete aprire il “Pannello di controllo” delle periferiche di Windows tramite il percorso “Avvio-Impostazioni-Pannello di controllo”. Selezionando “Stampanti ed altro hardware” e successivamente “Periferiche di gioco”: viene visualizzata la schermata riportata in Figura 5. Premete sul pulsante “Aggiungi” e selezionate come periferica “Joystick a 2 assi e 2 pulsanti”. Se invece utilizzerete un altro tipo di hardware non dovrete far altro che selezionare il tipo corrispondente. Sap-

piate che è possibile ridefinire a piacere i comandi disponibili in modo da personalizzare la periferica secondo le proprie esigenze. In ogni caso, se la procedura è andata a buon fine otterrete la finestra riportata in Figura 6. A questo punto la periferica è funzionante ma deve essere calibrata: per fare ciò, cliccate su “Proprietà” per visualizzare la fine-

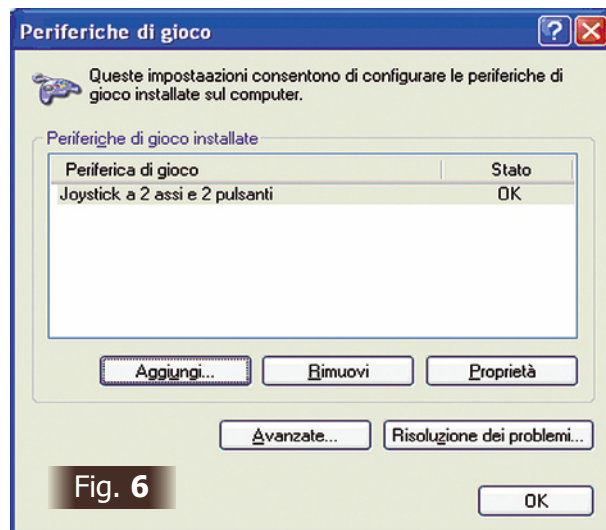


Fig. 6

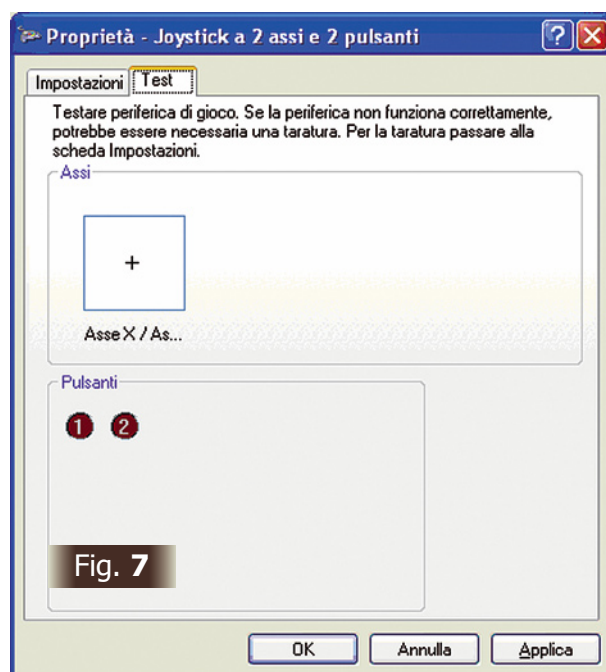


Fig. 7

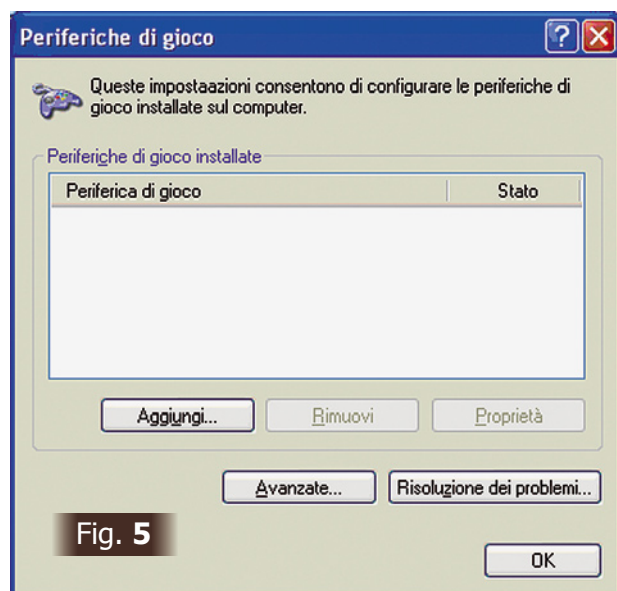


Fig. 5

stra di Figura 7. Selezionate prima “Impostazioni”, cliccate su “Taratura” e seguite le istruzioni che fornisce il sistema operativo. Questa procedura permette di impostare l’escursione massima e la centratura dei potenziometri. Selezionate ora l’opzione “Test” (sempre in figura 7) per muovere il puntatore a forma di croce e verificare che la calibrazione sia corretta. Se la periferica è stata realizzata da voi o state semplicemente compiendo degli esperimenti, potrebbe essere necessario invertire un potenziometro qualora il movimento della cloche non corrispondesse al movimento del punta- ➤

Listato 1

In questo riquadro riportiamo parte del sorgente dell'applicativo Visual Basic BrarobJoy, che controlla il braccio in funzione dello stato del joystick. Listato 1 contiene le dichiarazioni (*BrarobJoy.bas*); Listato 2, invece, contiene il codice vero e proprio (*BrarobJoy.frm*).

```
Attribute VB_Name = "Module1"

' Dichiarazione funzioni API di Windows.
Declare Function joyGetPosEx Lib "winmm.dll" _
    (ByVal uJoyID As Long, pji As JOYINFOEX) As Long
Declare Function joyReleaseCapture Lib "winmm.dll" _
    (ByVal id As Long) As Long
Declare Function joySetCapture Lib "winmm.dll" _
    (ByVal hWnd As Long, ByVal uID As Long, _
    ByVal uPeriod As Long, ByVal bChanged As Long) _
    As Long
Declare Function ShowCursor Lib "user32" _
    (ByVal bShow As Long) As Long

' Dichiarazione costanti API
Public Const JOYSTICKID1 = 0
Public Const JOYSTICKID2 = 1
Public Const JOY_POVCENTERED = -1
Public Const JOY_POVFORWARD = 0
Public Const JOY_POVRIGHT = 9000
Public Const JOY_POVLEFT = 27000
Public Const JOY_RETURNX = &H1&
Public Const JOY_RETURNX = &H2&
Public Const JOY_RETURNZ = &H4&
Public Const JOY_RETURNR = &H8&
Public Const JOY_RETURNU = &H10&
Public Const JOY_RETURNV = &H20&
Public Const JOY_RETURNPOV = &H40&
Public Const JOY_RETURNBUTTONS = &H80&
Public Const JOY_RETURNRAWDATA = &H100&
Public Const JOY_RETURNPOVCTS = &H200&
Public Const JOY_RETURNCENTERED = &H400&
Public Const JOY_USEDDEADZONE = &H800&

Public Const JOY_RETURNALL = (JOY_RETURNX Or _
    JOY_RETURNX Or JOY_RETURNZ Or JOY_RETURNR Or _
    JOY_RETURNU Or JOY_RETURNV Or JOY_RETURNPOV Or _
    JOY_RETURNBUTTONS)
Public Const JOY_CAL_READALWAYS = &H10000
Public Const JOY_CAL_READONLY = &H2000000
Public Const JOY_CAL_READ3 = &H40000
Public Const JOY_CAL_READ4 = &H80000
Public Const JOY_CAL_READXONLY = &H100000
Public Const JOY_CAL_READYONLY = &H200000
Public Const JOY_CAL_READ5 = &H400000
Public Const JOY_CAL_READ6 = &H800000
Public Const JOY_CAL_READZONLY = &H1000000
Public Const JOY_CAL_READUONLY = &H4000000
Public Const JOY_CAL_READVONLY = &H8000000
' Fine dichiarazione costanti API

' Dichiarazione della struttura API.
Type JOYINFOEX
    dwSize As Long ' Dimensione struttura.
    dwFlags As Long ' Flag che indica il ritorno.
    dwXpos As Long ' Posizione X.
    dwYpos As Long ' Posizione Y.
    dwZpos As Long ' Posizione Z.
    dwRpos As Long ' Posizione V.
    dwUpos As Long ' Posizione W.
    dwVpos As Long ' Posizione U.
    dwButtons As Long ' Stato dei pulsanti.
    dwButtonNumber As Long ' Numero pulsante premuto.
    dwPOV As Long ' Stato di funzionamento.
    dwReserved1 As Long ' Riservata.
    dwReserved2 As Long ' Riservata.
End Type
```

tore sullo schermo. Verificate infine il funzionamento dei due pulsanti, tramite il riquadro "Pulsanti".

A questo punto la vostra periferica è pronta per essere utilizzata col programma di controllo del braccio robot, programma a cui dedicheremo tutto il capitolo successivo.

Il software

Per semplificare al massimo la comprensione del programma (ampiamente descritto nei riquadri Listato 1 e 2), riportiamo di seguito alcune informazioni relative all'utilizzo della periferica joystick in ambiente di programmazione Visual Basic; tenete comunque presente che le indicazioni sono utilizzabili anche con altri sistemi di sviluppo, come ad esempio Delphi.

La lettura delle coordinate avviene tramite l'interrogazione di una particolare funzione API (Application Program Interface) di Windows denominata "joyGetPos" alla quale si accede tramite la libreria "WINMM.DLL" presente nella cartella System32. Non è necessaria alcuna installazione in quanto questo file è

già presente dopo l'installazione di Windows. A questo punto fate riferimento al Listato 1: esso mostra le dichiarazioni nell'applicativo BrarobJoy.exe, ossia il programma di controllo del braccio robot. Le dichiarazioni per l'utilizzo della funzione "joyGetPos" sono visibili in rosso nella parte iniziale del listato mentre la sua struttura (in parole povere, il sommario dei dati che la libreria può restituire a richiesta), sempre in rosso, è delimitata dai tag

Type JOYINFOEX

...

End Type

in cui le variabili che iniziano con "dwX..." e "dwY..." rappresentano il valore analogico delle coordinate dei potenziometri disponibili, mentre le

variabili "dwButton..." contengono le informazioni relative al pulsante premuto.

A questo punto occupiamoci del progetto vero e proprio, ossia del controllo del braccio robotizzato tramite joystick, il cui sorgente è visibile nel Listato 2.

Innanzitutto occorre aggiornare il firmware del PIC della scheda di controllo con la versione brarob2.hex, versione che differisce dalla precedente in quanto il comando seriale inviato da PC viene utilizzato immediatamente dal microcontrollore per posizionare i vari servi nella posizione voluta. Nella versione brarob.hex i servi vengono azionati portandoli lentamente nella posizione voluta evitando movimenti troppo bruschi.

Tabella 1 - Corrispondenza numero servo / segmento del braccio

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	Apri / chiude pinza.

Listato 2

```

* Dichiarazione delle variabili di programma.
Dim myJoy As JOYINFOEX
Dim X_pos, Y_pos, X_ini, Y_ini, Y1_pos, Y2_pos, Y3_pos, _
    Y4_pos, Y5_pos, X1_pos As Byte
Dim pinza, i, button, seleziona, premi, apre As Byte
* *****
Private Sub Command1_Click()
    ' Subroutine clic pulsante Command1
    ' funzione azzeramento joystick.
    myJoy.dwSize = 64
    myJoy.dwFlags = JOY_RETURNALL
    r = joyGetPosEx(JOYSTICKID1, myJoy) ' legge coord./puls. joystick
    X_ini = Int(myJoy.dwXpos / 500) ' mostra le coordinate
    Y_ini = Int(myJoy.dwYpos / 500) ' X e Y nelle caselle di testo.
    Y1_pos = 150 ' Configura variabili di programma.
    Y2_pos = 150
    Y3_pos = 150
    Y4_pos = 150
    Y5_pos = 150
    X1_pos = 150
    pinza = 0
    i = 0
    seleziona = 1
    apre = 0
    Timer1.Enabled = True
End Sub
* *****
Private Sub Form_Load()
    ' All'apertura del programma
    ' se il controllo MSCOMM è chiuso
    ' viene configurato come COM1
    ' e la seriale viene aperta.
    If MSCOMM1.PortOpen = False Then
        MSCOMM1.CommPort = 1
        MSCOMM1.PortOpen = True
    Else
        MsgBox "Porta già aperta!"
        MSCOMM1.PortOpen = False
    End If
End Sub
* *****
Private Sub Form_Unload(Cancel As Integer)
    ' Termine programma.
    ' La seriale viene comunque chiusa.
    MSCOMM1.PortOpen = False
End Sub
* *****
Private Sub Timer1_Timer()
    ' Subroutine timer, che scandisce
    ' le letture al joystick.
    myJoy.dwSize = 64
    myJoy.dwFlags = JOY_RETURNALL
    Dim data As String
    Dim delta As Integer
    r = joyGetPosEx(JOYSTICKID1, myJoy) ' Legge coord./puls.joystick.
    button = myJoy.dwButtons ' mostra i tasti premuti.
    Text3.Text = button
    If premi = 0 And button = 1 Then
        seleziona = seleziona + 1
        If seleziona > 3 Then seleziona = 1
        premi = 1
    ElseIf premi = 1 And button = 0 Then
        premi = 0
    End If
    Label7.Caption = Chr(seleziona + 65)
    X_pos = (myJoy.dwXpos / 500)
    Y_pos = (myJoy.dwYpos / 500)
    Text1.Text = Int(X_pos)
    Text2.Text = Int(Y_pos)
    delta = ((myJoy.dwYpos / 500) - Y_ini) / 40
    If pinza = 0 And button = 2 Then
        ' Spostamento dal centro.
        ' Funzione toggle pinza.
        ' Se pinza aperta
        ' allora chiude.
        If apre = 0 Then
            data = "5" + Chr(70)
            apre = 1
        Else
            data = "5" + Chr(200)
            apre = 0
        End If
    ElseIf pinza = 1 And button = 0 Then
        ' Puls. a riposo. Tutto fermo.
        pinza = 0
    Else
        ' Definisce pos. segmento B.
        If seleziona = 1 Then
            Y2_pos = Y2_pos - delta
            If Y2_pos < 70 Then Y2_pos = 70
            If Y2_pos > 230 Then Y2_pos = 230
            data = "2" + Chr(Y2_pos)
        End If
        ' Definisce pos. segmento C.
        If seleziona = 2 Then
            Y3_pos = Y3_pos + delta
            If Y3_pos < 70 Then Y3_pos = 70
            If Y3_pos > 230 Then Y3_pos = 230
            data = "3" + Chr(Y3_pos)
        End If
        ' Definisce pos. segmento D.
        If seleziona = 3 Then
            Y4_pos = Y4_pos - delta
            If Y4_pos < 70 Then Y4_pos = 70
            If Y4_pos > 230 Then Y4_pos = 230
            data = "4" + Chr(Y4_pos)
        End If
        ' Prepara alternativamente dato
        ' segmento A e dato di uno dei
        ' segmenti B, C, D.
        If i = 1 Then
            i = i + 1
        Else
            i = 2
        End If
        delta = ((myJoy.dwXpos / 500) - X_ini) / 20
        If X1_pos > 70 And delta < 0 Then X1_pos = X1_pos + delta
        If X1_pos < 230 And delta > 0 Then X1_pos = X1_pos + delta
        data = "1" + Chr(X1_pos)
    End If
    ' Mostra sul monitor le posizioni dei quattro servi.
    Label12.Caption = "A=" & X1_pos & " " & X1_pos - 150 & "°"
    Label14.Caption = "B=" & Y2_pos & " " & Y2_pos - 150 & "°"
    Label15.Caption = "C=" & Y3_pos & " " & Y3_pos - 150 & "°"
    Label16.Caption = "D=" & Y4_pos & " " & Y4_pos - 150 & "°"
    MSCOMM1.Output = data
End Sub
* *****

```

Questo purtroppo implica l'attesa del completamento del comando prima di inviare una nuova coordinata e quindi permette il movimento di un solo servo per volta. Nella nuova versione di firmware, invece, è il PC che decide con quale velocità si deve muovere il servo e, nota più importante, possiamo azionare più servo contemporaneamente: è sufficiente che l'applicativo su PC invii in successione ai singoli servocontrolli la nuova posizione richiesta.

I dati inviati via seriale sono composti da due byte: il primo contiene il carattere ASCII che identifica il servo da comandare, secondo l'ordine visibile in Tabella 1. Il secondo byte, invece, contiene un carattere il cui equivalente in ASCII rappresenta la durata degli impulsi di controllo del servo (espressa in centesimi di microsecondo) ed il cui valore stabilisce l'angolo di rotazione dell'alberino. Se, ad esempio, viene inviata la stringa "1d", indichiamo al PIC di muovere il servo n° 1 (quello relativo alla rotazione sinistra-destra) sino alla posizione corrispondente al segnale PWM di 1 msec ovvero 60° (o 90°: ciò dipende dal servo) a sinistra 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 "û" è proprio 150. Vengono accettati tutti i valori ASCII a partire da 70 sino a 230 e corrispondenti ad una rotazione di ±60° (o ±90°) con la posizione centrale corrispondente al valore 150.

Per rendere fluido il movimento, è necessario inviare una sequenza temporizzata di brevi spostamenti. Il programma di gestione deve quindi interpretare i dati provenienti dal joystick e renderli disponibili al microcontrollore che controllerà il successivo posizionamento.

Il software BrarobJoy può essere ➤

CD3+

Disturbatore Cellulari
All Bands
23W totali min.



NEW



PD-1

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

DESPY

Ideale per
la ricerca
di micro spie
fino a 3 Ghz



Bias s.a.

www.bias.sm

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

scaricato gratuitamente dal sito della rivista (www.elettronica.in, nome file: BrarobJoy.msi) ed installato seguendo la normale procedura di installazione di applicativi Windows. Per questa applicazione è necessario un joystick a due assi con due pulsanti correttamente configurato secondo la procedura già descritta. Si tratta del dispositivo più classico, che probabilmente è già in vostro possesso.

Joystick e porta seriale: sempre più una rarità

Prima di avviare il programma, è necessario aprire una piccola parentesi riguardante i PC e i joystick moderni: abbiamo già detto in molte altre occasioni che la porta seriale, considerata obsoleta, sta via via scomparendo perché rimpiazzata dalla più veloce porta USB. Ebbene, anche la porta joystick analogica sta diventando sempre più una rarità, dato che i moderni controlli gioco (dotati di molte funzioni in più rispetto al nostro modello a due assi e due pulsanti) sono stati progettati anche loro per utilizzare la porta USB. In ragione di quanto appena detto, il PC su cui installerete il programma BrarobJoy deve disporre tanto di porta joystick classica (connettore Canon DB15) quanto di una seriale per comunicare con la scheda di controllo del braccio.

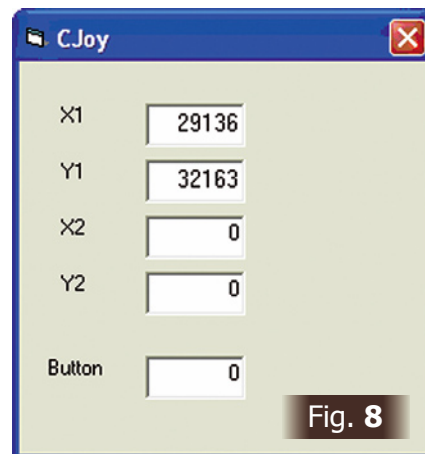


Fig. 8

Se così non fosse, dovete acquistare le periferiche necessarie, installarle e, dopo aver seguito tutte le istruzioni del sistema operativo e caricato i driver richiesti, verificare che tutto funzioni correttamente. Per fare ciò, potete utilizzare un altro semplice applicativo che abbiamo preparato per l'occasione (chiamato CJoy e scaricabile ancora una volta dal sito della rivista), tramite il quale potete verificare se un applicativo Visual Basic (nel nostro caso il CJoy stesso) dialoga correttamente sia con la periferica che con la libreria joystick. La schermata unica di CJoy è visibile in figura 8. Se muovendo il joystick vedrete cambiare i dati visualizzati, allora avrete la certezza che il PC è configurato correttamente. Pertanto potrete avviare ed utilizzare il programma BrarobJoy.

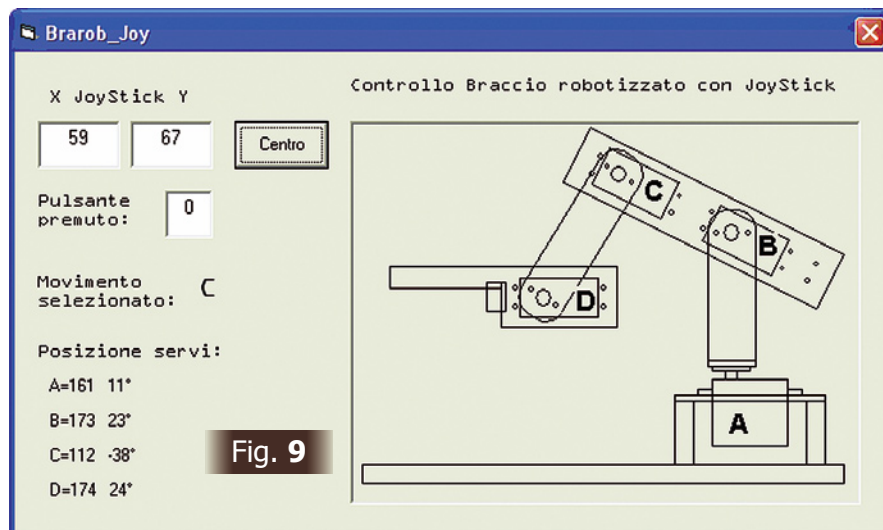
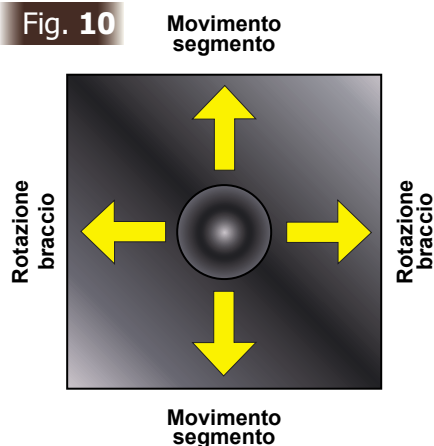


Fig. 9

Fig. 10

Utilizzo in pratica

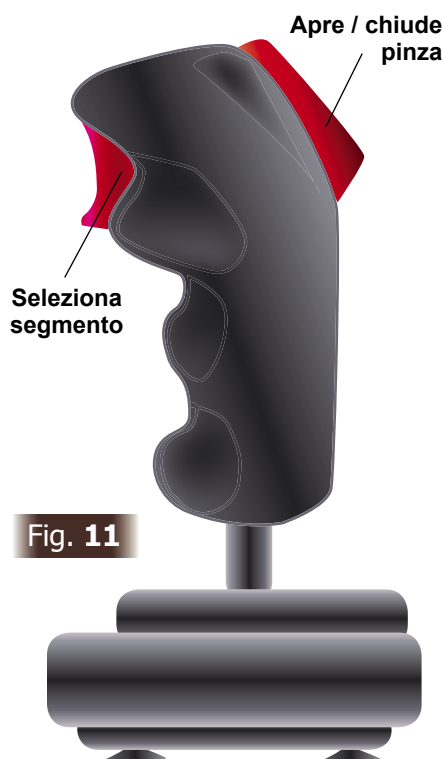
Avviate il programma BrarobJoy tenendo presente che non è dotato di controlli per modificare l'indirizzo della COM da usare (COM1, COM2, COM12...): esso infatti, come visibile dal Listato 2, utilizza sempre e solo la COM1 quindi, fate in modo che tale porta sia presente e libera sul vostro PC. Lasciando il joystick in posizione neutra (leva a riposo), cliccate con il mouse sul pulsante "Centro": ciò permette al programma di centrare correttamente la posizione di neutro del controllo. La schermata si presenta come visibile in Figura 9.

Collegate ora (senza cambiare la posizione della leva) il cavo seriale del braccio ed accendete quest'ultimo, così come descritto nell'articolo pubblicato sul fascicolo 116. A questo punto siete pronti per azionare il braccio. Come vedete, il software non comporta alcuna difficoltà di impiego e riporta a video i valori relativi a posizione del joystick, pulsante premuto e posizione dei servi del braccio. Essendo disponibili solo i due assi X e Y, per poter gestire tutti i movimenti abbiamo fatto in modo che al movimento della variabile X corrisponda la rotazione del braccio e al movimento della variabile Y corrisponda il movimento di uno dei segmenti B, C e D.

In Figura 9 potete identificare facilmente i quattro segmenti (A, B, C e D) mentre la Figura 10 mostra le va-

rie azioni associate alla leva del joystick: il movimento destra-sinistra (X) è sempre assegnato al segmento A (rotazione), mentre il movimento avanti-indietro (Y) è assegnato di volta in volta ad uno dei segmenti B, C, D.

Prestate ora attenzione alla Figura 11: la selezione del segmento viene effettuata tramite la pressione del pulsante frontale (ad ogni comando il programma seleziona il segmento successivo seguendo il ciclo B-C-D a rotazione) mentre il pulsante

**Fig. 11**

superiore permette di aprire e chiudere la pinza in modalità bistabile (ipotizzando che la pinza sia inizialmente aperta): una prima pressione la fa chiudere la successiva la fa riaprire. Il movimento del braccio tramite la cloche avviene nello stesso modo delle leve di comando delle macchine per movimento terra che si vedono nei cantieri ed è molto simile al funzionamento dei bracci robotizzati installati sui batiscafi per esplorazioni marine. Spostando la leva si genera il movimento e questo è tanto più rapido quando maggiore e veloce è lo spostamento della clo-

che.

Per fermare il movimento è sufficiente riportare la cloche nella posizione neutra, mentre il movimento opposto lo si ottiene portandola nella direzione opposta. Combinando i movimenti X-Y con i vari segmenti è possibile sia far assumere al braccio qualsiasi posizione nel raggio di 20 cm che afferrare con la pinza piccoli oggetti della grandezza di qualche centimetro cubo. Lasciando la cloche in posizione centrale il braccio mantiene la posizione raggiunta.

Come ultimo appunto, prima di spegnere il tutto, è consigliabile riportare il braccio nella posizione neutra con tutti i segmenti in verticale in quanto alla mancanza della tensione di alimentazione tutti i servo perderanno forza, rischiando di rompersi per effetto di movimenti innaturali. Sappiate che i servo da modellismo sono molto robusti, ma non indistruttibili: raccomandiamo quindi di evitare, per quanto possibile, movimenti bruschi o sforzi eccessivi, che potrebbero danneggiare le delicate parti meccaniche.

Note conclusive

I nostri lettori sanno che la robotica ci appassiona e questa divagazione prettamente software, in leggera controtendenza rispetto agli argomenti trattati da una rivista di elettronica, conferma che il nostro interesse per tale tecnologia spazia in tutte le possibili direzioni.

Sappiate che abbiamo altri interessanti progetti, alcuni nel cassetto, altri sulla scrivania e altri già pronti per la pubblicazione: in questo stesso numero presentiamo Robot In, un robot lottatore di minisumo mentre, sulla stessa linea, stiamo lavorando su una versione specializzata in "area di rigore" (chiamata Ronald In). Continuate a seguirci e vedrete insieme a noi quanto un argomento complesso come la robotica possa diventare facile e divertente.



Liquid Crystal Display dalla teoria alla pratica

A CURA DI **Matteo Destro**

Iniziamo questo mese ad occuparci dei display in tecnologia OLED analizzando il principio di funzionamento di questi componenti e proponendo la costruzione di una specifica demoboard.

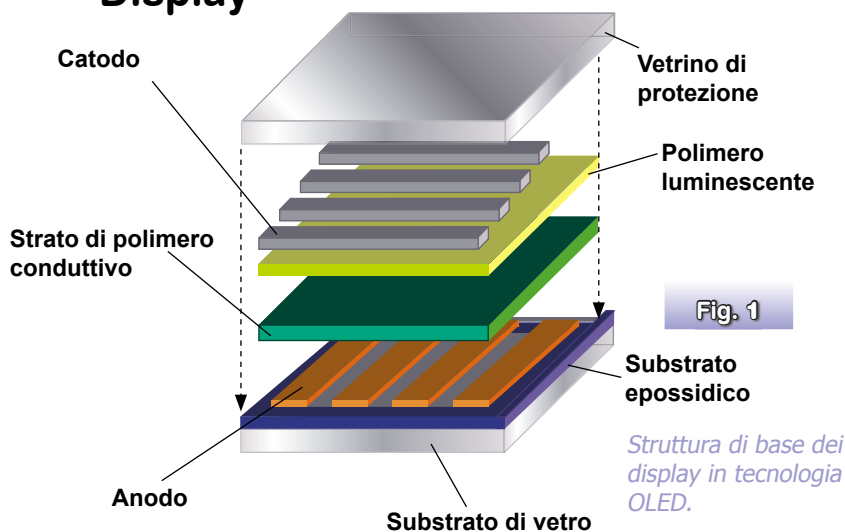
9

Con la precedente puntata abbiamo concluso la trattazione dei display grafici in tecnologia STN. È quindi giunto il momento di iniziare a parlare, come avevamo promesso all'inizio del corso, dei display in tecnologia OLED. Esattamente come è avvenuto per i display alfanumerici e grafici STN, abbiamo preparato una demoboard specifica per i componenti che utilizzano questa tecnologia. Abbiamo inoltre sviluppato una libreria firmware per la gestione del modello di display da noi utilizzato (che descriveremo nelle successive puntate del Corso). Poiché è nostra intenzione consentire l'impiego della demoboard con la maggior parte dei display OLED disponibili in commercio, abbiamo previsto alcune espansioni hardware che consentono di utilizzare la demoboard con numerosi modelli. Entriamo ora nel merito dell'argomento.

Tecnologia OLED

La sigla "OLED" è l'acronimo di "Organic Light Emitting Diode" che tradotto in italiano significa "diodo organico ad emissione di luce". Quando questa tecnologia viene applicata ai sistemi di visualizzazione, essa permette di realizzare sia display monocromatici che a colori, entrambi capaci di emettere luce propria; pertanto, a differenza dei display a cristalli liquidi, gli OLED non richiedono componenti aggiuntivi per essere illuminati. Ciò significa che non è necessario alcun tipo di "back-panel" che li illumini in trasparenza, permettendo così di realizzare display molto sottili, pieghevoli e addirittura, in un futuro non molto lontano, anche arrotolabili. Un ulteriore vantaggio dei display OLED è rappresentato dal fatto che, per funzionare correttamente, necessitano di meno energia rispetto ai display tradizionali. ➤

Display

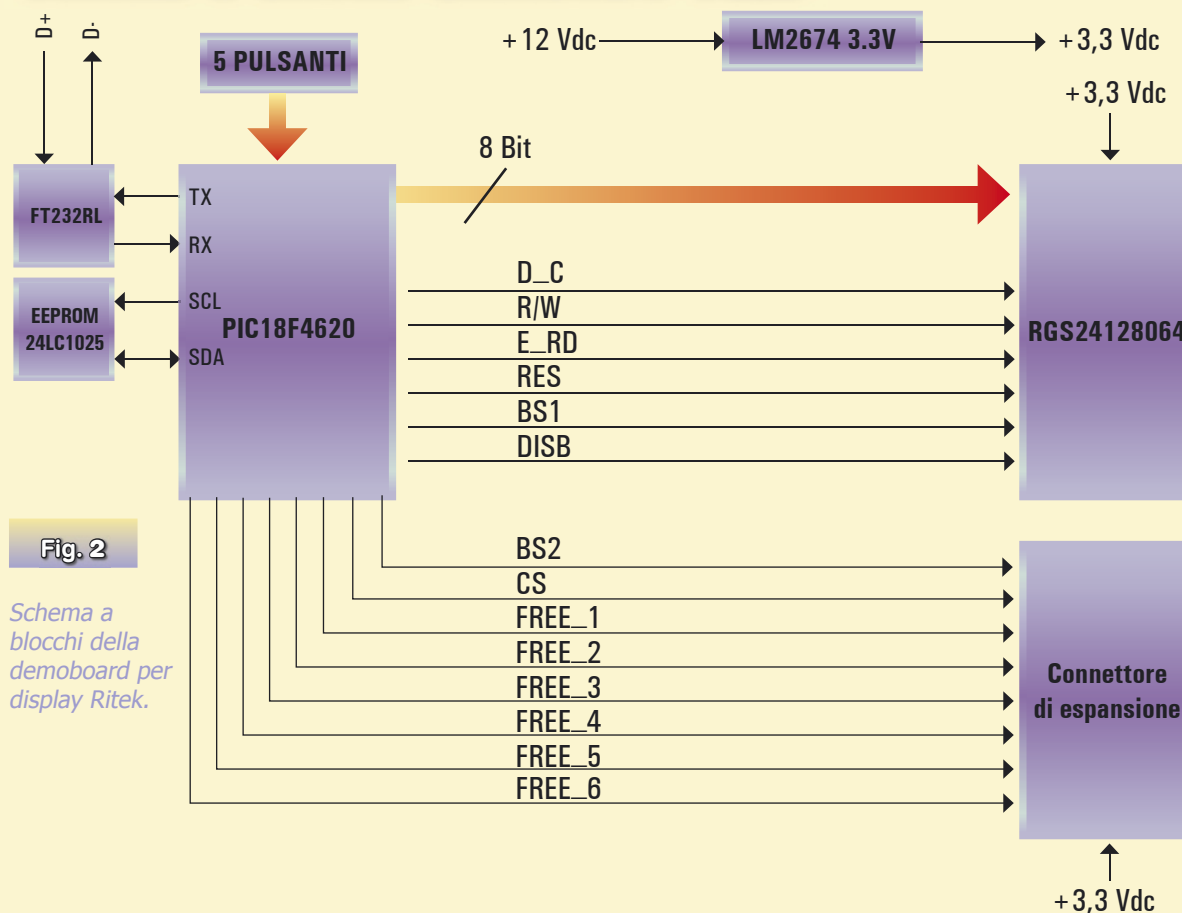


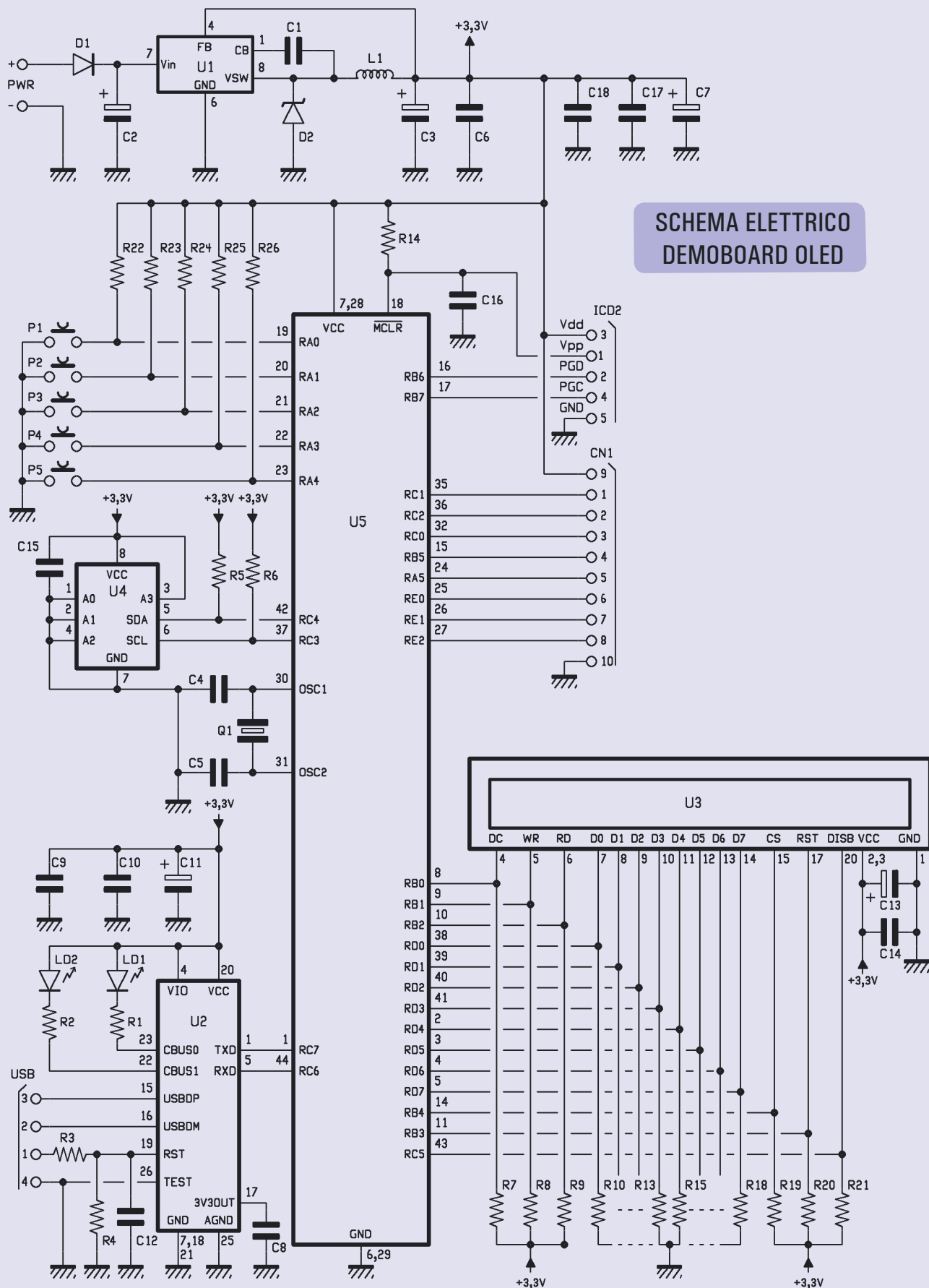
dell'elettroluminescenza posseduta da alcuni elementi organici sia conosciuta da lungo tempo, i primi sviluppi di display OLED non andarono mai oltre lo stadio di prototipo, in quanto richiedevano tensioni di alimentazione troppo alte per risultare utili nelle applicazioni pratiche. Successivamente, furono sviluppate con successo pellicole di materiale organico elettroluminescente molto sottili che funzionavano anche con tensioni di alimentazione più basse. I primi modelli presentavano una struttura molto semplice: erano infatti composti solamente da una pellicola di

Il materiale organico (ovvero contenente strutture di carbonio) con cui si realizzano questi dispositivi può essere un polimero conduttivo simile alla plastica oppure un materiale organico non polimerico con peso molecolare relativamente basso, entrambi elettroluminescenti. Benché la proprietà

sostanza organica posta tra due elettrodi (anodo e catodo); applicando tensione ai due elettrodi, si generava una corrente che, attraversando il polimero, determinava l'emissione di luce. Tuttavia questa tecnologia era poco pratica, in quanto richiedeva una precisione estrema in fase di produzione, al >

Schema a blocchi demoboard OLED





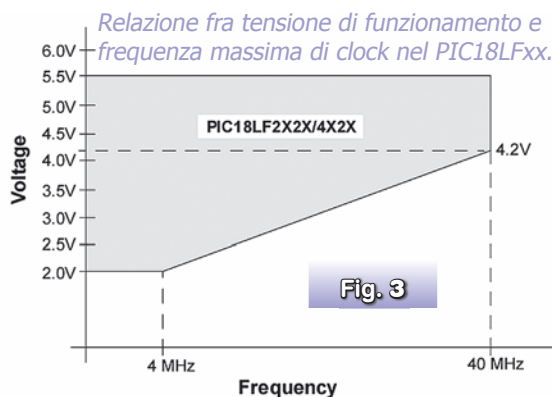
punto che un allineamento non perfetto causava grandi perdite di energia con conseguente scarso rendimento in termini di resa luminosa. I primi display efficienti funzionanti a bassa tensione furono presentati nel 1987 (da Ching Tang e Steve Van Slyke): essi facevano uso di due strati organici, uno predisposto per ricevere lacune, l'altro per ricevere elettroni. Con questa tecnica fu possibile costruire vetrini LCD ad alta luminosità alimentati da basse tensioni (circa 10 volt): era l'inizio dell'evoluzione vera e propria.

Gli strati organici con cui sono realizzati i display OLED di base possono emettere solo luce bianca, ma con opportuni drogaggi (cioè utilizzando composti elettroluminescenti aggiuntivi aventi strutture chimico-fisiche particolari) essi possono generare anche luce rossa, verde o blu (RGB). Da ciò consegue che, combinando i tre colori primari, è possibile ottenere tutti i colori dello spettro visibile in modo analogo a quanto avviene nei normali display a colori. Ne consegue che ogni pixel dell'immagine a colori viene realizzato utilizzando tre "micro-display mono-pixel" affiancati e pertanto il colore del pixel cambia a seconda dell'intensità luminosa emessa da ogni singolo micro-display. Concludiamo questa breve introduzione dicendo che un moderno display OLED è composto da vari strati sovrapposti per uno spessore totale di 300 nanometri (vedere Figura 1). Il primo strato trasparente serve come protezione e sopra di esso viene depositato uno strato conduttivo, anch'esso trasparente, che ha la funzione di anodo. Successivamente vengono aggiunti tre strati organici, il primo per l'iniezione delle lacune, il secondo per il trasporto degli elettroni, e infine il terzo, uno strato riflettente che ha la funzione di catodo. Tra i tre strati vengono poi inseriti i tre materiali elettroluminescenti (rosso, verde e blu) disposti a formare un unico strato.

Demoboard e display supportati

La nostra demoboard prevede l'impiego di un display Ritek RGS24128064YW003 il quale, come suggerisce la sigla (128064), ha un'area visibile di 128x64 pixel; il controller presente sul display, un SSD1303T10 della Solomon Systech, gestisce il vetrino dell'OLED e la comunicazione con il microcontrollore.

La demoboard, come accennato nell'introduzione, è stata progettata per potere ospitare, oltre al display, delle schede di espansione con a bordo display differenti dal Ritek. Il firmware e le relative routine grafiche sono realizzate in linguaggio



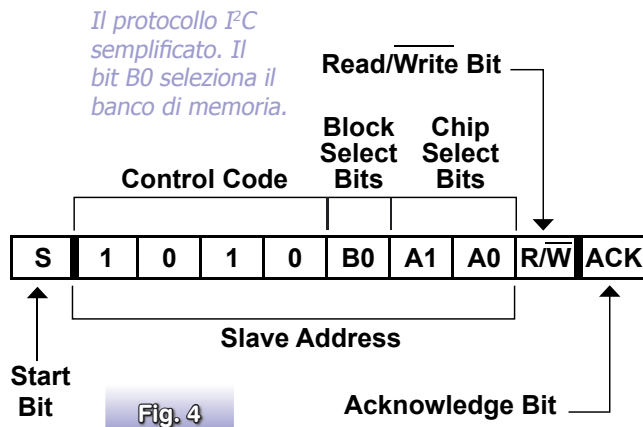
$$F_{MAX} = (16.36 \text{ MHz/V}) (V_{DDAPP\text{MIN}} - 2.0\text{V}) + 4 \text{ MHz}$$

Note: $V_{DDAPP\text{MIN}}$ is the minimum voltage of the PIC® device in the application.

ANSI C, linguaggio che permette di semplificare la stesura del codice sorgente. Naturalmente, per chi volesse cimentarsi nell'impresa, è possibile tradurre il codice da ANSI C a linguaggio Assembler per microcontrollori serie 18Fxxxx.

Schema elettrico demoboard

Durante la descrizione della demoboard fate riferimento allo schema a blocchi visibile in Figura 2 ed allo schema elettrico riportato a pagina 85. Notate che il display OLED è connesso al microcontrollore tramite un bus dati a 8 bit ed alcuni pin di controllo. Il Ritek è configurato per funzionare con interfacciamento in modalità "Motorola 8080", la quale prevede due pin di controllo per la gestione delle operazioni di lettura e scrittura (RD e WR). Sono inoltre necessari un pin per comunicare al controllore grafico se quanto presente sul bus è un dato o un comando (D/C), un pin di abilitazione del bus di comunicazione (BS1, che fa uscire il controller dalla condizione di alta impedenza), un pin di reset (RES) e un pin per attivare e disattivare il DC/DC converter presente a bordo del display stesso (DISB). Relativamente al nostro Ritek, esso prevede due alimentazioni, una per la logica di controllo e l'altra per la gestione del DC/DC converter: la prima non deve superare i +3,5 Vdc mentre la seconda può arrivare fino a +5 Vdc. Abbiamo quindi dotato la demoboard di un unico circuito di alimentazione a +3,3 Vdc (realizzato con il regolatore switching U1, un LM2674M-3.3) che garantisce la corretta alimentazione sia alla logica di controllo che al DC/DC converter (quest'ultimo, comunque, accetta alimentazioni fino a un minimo di +2,8 Vdc). Da quanto appena esposto consegue che anche tutti gli altri componenti attivi della demoboard sono alimentati +3,3 Vdc. Il microcontrollore utilizzato è un PIC18LF4620, identico a quello presente sul-



la demoboard dei display grafici STN, ma con la particolarità di poter essere alimentato a +3,3 Vdc in quanto facente parte della famiglia dei PIC con alimentazione estesa. La dicitura “LF” nella sigla indica che il micro supporta l’alimentazione estesa. Quando si utilizza questo tipo di dispositivi è necessario prestare attenzione alla massima frequenza di lavoro che possono accettare. Sui datasheet è sempre presente un grafico che evidenzia la relazione tra la tensione di alimentazione applicata e la massima frequenza di lavoro raggiungibile (Figura 3). È comunque sempre possibile fare riferimento all’equazione matematica:

$$F_{max} = 16,36 \frac{\text{MHz}}{\text{V}} * (V_{ddAppmin} - 2,0\text{V}) + 4\text{MHz}$$

che permette di calcolare la massima frequenza di lavoro nota la tensione di alimentazione. Rapportata all’applicazione in esame, quindi a un’alimentazione di +3,3 Vdc, si ottiene una frequenza massima di lavoro di circa 25 MHz. Questo significa che il quarzo da 5 MHz che abbiamo abbinato al PLL va più che bene: quest’ultimo, infatti, moltiplica per 4 la frequenza del quarzo, generando pertanto un clock a 20 MHz, compatibile con il valore massimo ottenuto tramite la formula matematica. Ai più attenti non sarà sfuggito che questa demoboard lavora ad una frequenza pari alla metà di quella utilizzata sulla demo precedente. Ciò comporta che il tempo di esecuzione di un’istruzione a singolo ciclo è di 200 nsec contro i 100 nsec della demoboard per LCD.

Al microcontrollore fanno capo cinque pulsanti (P1÷P5), necessari per la gestione delle varie funzioni disponibili nel controller SSD1303T: essi risultano tutti collegati alla porta A e ognuno di essi dispone di una resistenza di pull-up pertanto un pulsante risulta premuto quando sul relativo pin

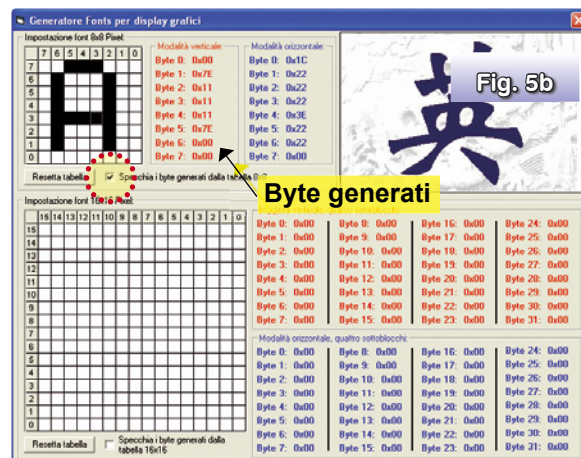
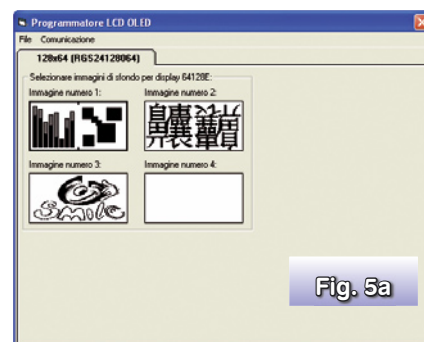
di ingresso alla porta A è presente una tensione di 0 Vdc (GND); a riposo, invece, sono presenti +3,3 Vdc. La parte dimostrativa della scheda è rappresentata da quattro immagini che possono essere visualizzate in successione. La memoria EEPROM, una 24FC1025 da 1 Mbit (U4) collegata al PIC attraverso il bus I²C, serve appunto per memorizzare tali immagini; essa viene programmata utilizzando un apposito applicativo su piattaforma Windows (già utilizzato per i display grafici standard e visibile in Figura 5a) che, grazie alla porta USB, permette di realizzare un collegamento fisico tra il PC e il microcontrollore: questi, a sua volta, si fa carico di trasferire i dati immagine alla EEPROM. Internamente, la memoria è stata suddivisa in blocchi logici da 32 kbyte ciascuno; attualmente in questa applicazione vengono sfruttati solo i primi 4 kbyte per memorizzare le immagini dimostrative, quindi rimane molta memoria libera per eventuali espansioni future, come ad esempio la gestione di nuove immagini per display diversi da quello in esame. Relativamente all’accesso in EEPROM via bus I²C, ricordiamo che per indirizzare una memoria da 1 Mbit sono necessari tre byte di indirizzo, ma in realtà il protocollo di trasmissione su bus ne presenta solo due, pertanto la memoria totale viene suddivisa in due blocchi ➤

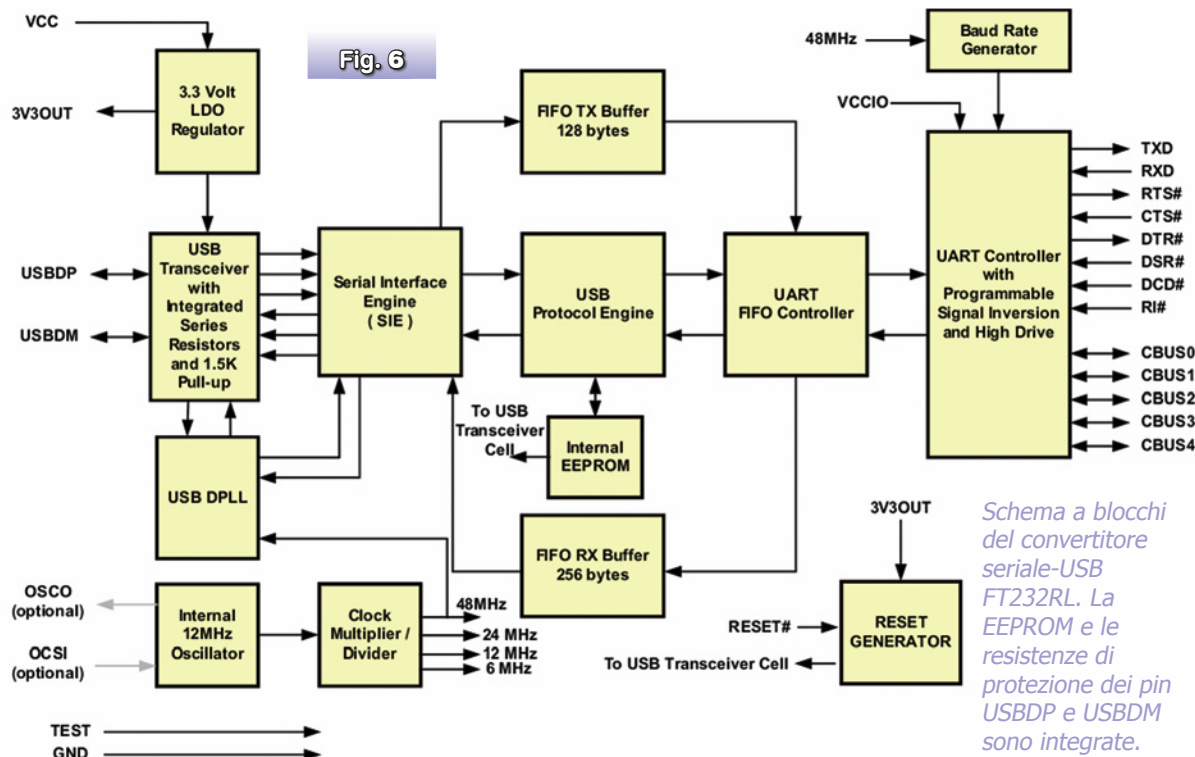
A destra, programma per trasferire le immagini alla memoria della demoboard.

In basso, il generatore di font.

Occorre spuntare

l’opzione cerchiata in rosso per generare i byte in modalità speculare rispetto all’asse X della tabella.



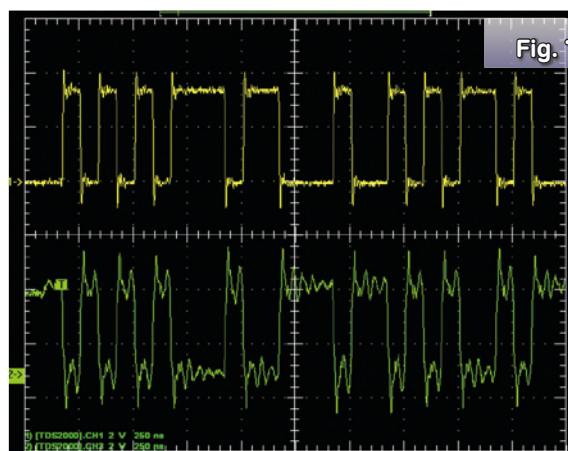


distinti da 65.536 byte ciascuno. Per indirizzare il primo o secondo blocco è necessario impostare il bit **B0** presente nel byte di comando. Fate ora riferimento alla Figura 4: con **B0** = 0 si indirizzano i primi 65.536 byte e con **B0** = 1 i rimanenti. Il “control code” è fisso mentre **A0** e **A1** vengono impostati a seconda di come sono stati collegati i pin **A0** e **A1** sulla scheda (nel nostro caso entrambi a zero). Il bit R/W serve per discriminare le operazioni di lettura da quelle di scrittura.

Ritorniamo ora all’argomento principale, ossia il display: dato che il controller SSD1303T non possiede una tabella interna dei caratteri (nessuna CGROM) abbiamo sfruttato una porzione della memoria FLASH del PIC18LF4620 per realizzare una mini tabella del font, generata utilizzando un programma su piattaforma Windows (vedere figura 5b). Considerato lo scopo didattico dell’applicazione, abbiamo realizzato una tabella caratteri ridotta (tutti di dimensioni di 8x8 pixel) in cui sono presenti solo i simboli dalla “A” alla “Z” e il simbolo di spaziatura: se necessario, è possibile estendere tale tabella esattamente come abbiamo già fatto in passato per generare i caratteri speciali nelle CGROM dei display sia grafici che alfanumerici, ma in questo caso occorre modificare il codice sorgente. Come si può osservare dalla figura, è necessario spuntare l’opzione di “mirroring”: se ciò non venisse fatto, i caratteri visualizzati a display sarebbero ribaltati (specchiati)

rispetto all’asse delle “X”. I byte da inserire in FLASH sono quelli evidenziati in rosso nella prima colonna (modalità verticale; fate riferimento alla precedente puntata per i dettagli sulle modalità verticale e orizzontale, le quali dipendono da come il produttore del display decide di gestire la memoria grafica e di conseguenza il vetrino).

Come accennato poco fa, per trasferire i dati dal PC al PIC18LF4620 si utilizza una connessione USB per cui abbiamo dotato la demoboard di un integrato che si occupa di eseguire la conversione da USB a RS232 e viceversa: come per la precedente versione, il baud-rate raggiunto è di 115200 bit per secondo. L’integrato che si occupa della conversione è un FT232RL (U2) della FTDI, chip che per funzionare necessita di pochissimi com-



Trasmissione USB in corso.

Tabella 1 - I pin di segnalazione del chip FT232RL

CBUSx	Descrizione
CBUS0	Funzione Led TX. Va basso quando trasmette un dato all'interfaccia USB.
CBUS1	Funzione Led RX. Va basso quando riceve un dato dall'interfaccia USB.
CBUS2	Funzione TXDEN. Abilita trasmissione per interfaccia 485.
CBUS3	Funzione PWREN. Va alto quando l'integrato entra in "suspend mode".
CBUS4	Funzione SLEEP. Va basso quando l'integrato entra in "suspend mode". Può essere utilizzato per attivare/disattivare circuiti TTL esterni all'integrato FT232RL.

ponenti aggiuntivi. Di seguito riportiamo le sue caratteristiche più significative:

- non richiede sviluppo di firmware;
- velocità di trasferimento fino a 1 Mbaud per interfaccia RS232;
- esente da royalty;
- EEPROM da 1024 byte integrata per la memorizzazione di VID e PID (codici di identificazione prodotto e fornitore), numero seriale e descrizione prodotto...
- resistenze serie poste sui pin D+ e D- e resistenze di pull-up sul segnale D+ integrate;
- filtro RC per la gestione del pin AVCC integrato;
- oscillatore integrato nel dispositivo.

Anche se non è la sede adatta, vogliamo spendere qualche parola per l'FT232RL perché è la prima volta che usiamo questo chip, la cui struttura interna è schematizzata in Figura 6. Osservando il disegno notiamo l'oscillatore interno a 12 MHz, l'interfaccia USB con le resistenze serie e di pull-up integrate, la memoria EEPROM, l'interfaccia RS232 con la gestione dei pin CBUSx, un blocco di gestione del reset e una sezione di alimentazione a +3,3 Vdc di tipo LDO (Low Drop Out).

Come visibile nello schema elettrico della nostra applicazione, non tutti i pin dell'FT232RL sono stati utilizzati: ad esempio, del blocco inerente l'interfaccia RS232 abbiamo collegato solo i pin "TX" e "RX".

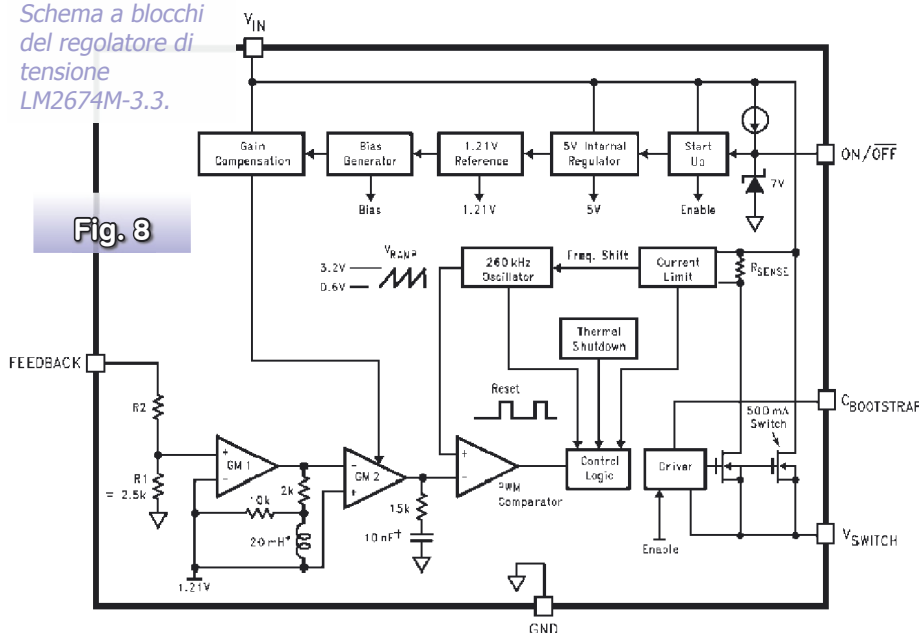
Il pin di "Reset#", attivo basso, è stato collegato attraverso un partitore di tensione al pin 1 del connettore USB, sul quale è presente l'alimentazione fornita dal PC per alimentare le periferiche connesse al bus USB; quindi tale pin viene utiliz-

zato per far "capire" al chip se esso è fisicamente collegato al computer. A cavo scollegato, infatti, sul pin di "Reset#" dell'integrato si misura una tensione di 0 Vdc, livello logico che pone l'FT232RL in condizione di reset. Viceversa, collegando il cavo alla scheda e al PC, si abilita l'integrato che, in un tempo molto breve, è

pronto a ricevere e trasmettere dati. In questo caso sul pin di "Reset#" sono presenti +3,3 Vdc (livello logico alto) ottenuti grazie al partitore di tensione formato dalle resistenze R3 e R4.

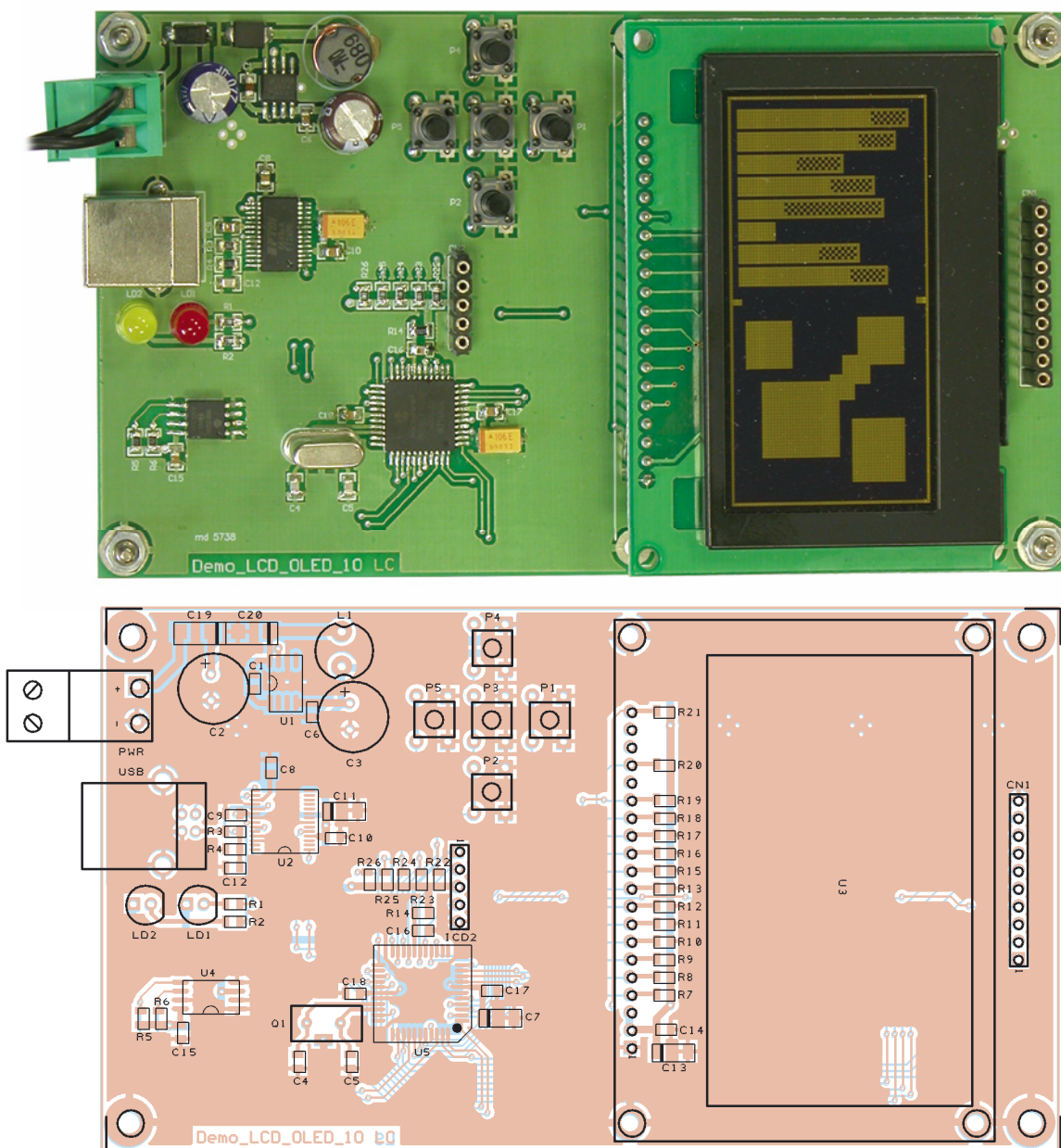
Il pin di "Test" deve essere collegato a massa: se così non fosse l'integrato entrerebbe in "test mode". I pin "USBDM" e "USBDP" identificano il segnale D- e D+ dell'interfaccia USB e devono essere rispettivamente collegati ai pin 2 e 3 del connettore USB. La Figura 7 mostra una trasmissione dati lato USB, in particolare i segnali D+

Schema a blocchi del regolatore di tensione LM2674M-3.3.



(giallo) e D- (verde).

I pin "CBUS0" e "CBUS1" vengono utilizzati per gestire due LED che servono per evidenziare la trasmissione di dati sui pin TX e RX lato RS232: se lampeggiano significa che è in corso una trasmissione dati. Tutti i pin "CBUSx" sono configurabili a seconda delle nostre necessità; le informazioni relative alle funzioni svolte vengono salvate in EEPROM. Per modificare i parametri di tali pin occorre l'utility "MPROG", scaricabile dal sito della FTDI: sappiate che quanto appena detto vale come informazione di carattere generale, dato che nella nostra applicazione abbiamo utilizzato tali pin così come sono stati configurati dal costruttore ➤



La realizzazione della demoboard non è difficoltosa: i componenti SMD utilizzati, infatti, sono molto spaziosi tra loro.

ELENCO COMPONENTI:

R1, R2: 270 ohm SMD
 R3: 10 kohm SMD
 R4: 22 kohm SMD
 R5+R6: 1,2 kohm SMD
 R7+R13: 3,3 kohm SMD
 R14: 22 kohm SMD
 R15+R26: 3,3 kohm SMD
 C1: 10 nF multistrato SMD
 C2: 220 µF 50 VL elettrolitico
 C3: 100 µF 25 VL elettrolitico
 C4, C5: 22 pF ceramico SMD
 C6, C7, C9, C10: 100 nF multistrato SMD
 C7: 10 µF 25 VL elettrolitico

C9: 100 nF multistrato SMD
 C10: 100 nF multistrato SMD
 C11: 10 µF 25 VL elettrolitico
 C12: 220 pF ceramico SMD
 C13: 10 µF 25 VL elettrolitico
 C14, C15: 100 nF multistrato SMD
 C16: 220 pF ceramico SMD
 C17, C18: 100 nF multistrato SMD
 D1: GF1M
 D2: 20BQ030
 U1: LM2674M-3.3
 U2: FT232RL
 U3: LCD Ritek RGS24128064YW003

U4: 24FC1025-I/SM
 U5: PIC18LF4620-TQFP
 Q1: Quarzo 5 MHz
 LD1: led 5 mm giallo
 LD2: led 5 mm rosso
 L1: Induttanza 68 µH
 P1+P5: Microswitch

Varie:

- Morsettiera 2 poli componibile
- Strip maschio 5 poli
- Connettore USB-B
- circuito stampato

re. Sempre a livello informativo, abbiamo riportato in Tabella 1 le funzioni di default svolte dai pin “**CBUSx**”; per tutte le altre caratteristiche e peculiarità vi suggeriamo di far riferimento al datasheet dell’integrato.

Come già anticipato, l’alimentazione della scheda è stata forzata a +3,3 Vdc perché il display della Ritek impone come limiti massimi +3,5 Vdc per l’alimentazione del controller SSD1303T10 e +5 Vdc per il convertitore DC/DC. Inoltre, alimentando tutto a +3,3 Vdc, non è necessario utilizzare integrati per la conversione dei livelli logici, ottenendo quindi una scheda più semplice e più facile da gestire. L’alimentazione viene ricavata con un regolatore di tensione switching modello LM2674M-3.3 (il suo schema a blocchi è riportato in figura 8), uno step-down ad alta efficienza che, con l’aggiunta di pochi componenti esterni, è in grado di fornire una corrente di 500 mA. In linea di massima, quando si utilizzano dei regolatori switching è necessario prestare particolare attenzione alla realizzazione del PCB. Ad esempio il condensatore elettrolitico di ingresso (C2) e il diodo Schottky (D2, vedere schema elettrico) sono piuttosto critici e devono essere posizionati il più vicino possibile al pin 7 per il primo e al pin 8 per il secondo, eliminando così la possibilità di effetti induttivi (che generano disturbi a radiofrequenza) dovuti all’eccessiva lunghezza delle piste. È inoltre buona norma realizzare sempre un piano di massa molto esteso (su entrambi i lati del PCB, se possibile) che, oltre a schermare i disturbi, funge anche da dissipatore di calore per l’integrato. Come ultimo punto, evidenziamo il connettore di espansione CN1 che, abbinato al connettore per il collegamento del display della Ritek, realizza un sistema di espansione per eventuali schede di gestione di nuovi display OLED. In questo modo si hanno a disposizione molti segnali con cui pilotare eventuali driver grafici più complessi, nonché tre linee di alimentazione a +3,3 Vdc e due punti per il collegamento della massa (GND).

Durante la realizzazione delle schede di espansione è necessario prestare attenzione alle resistenze di pull-up o di pull-down già presenti sulla scheda base: se fossero erroneamente presenti entrambe su una linea logica, tale linea non funzionerebbe. Comunque sono a di-

sposizione otto pin del micro (anch’essi collegati a CN1) sprovvisti di qualsiasi resistenza e quindi utilizzabili a piacimento, in funzione delle esigenze di progettazione.

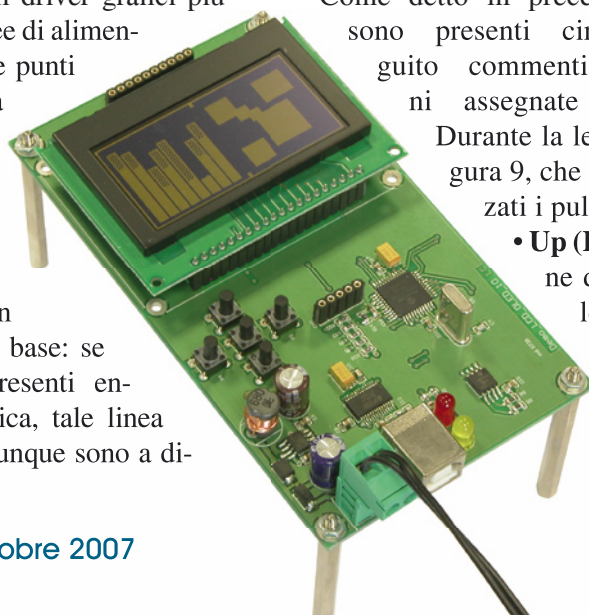
Realizzazione pratica

Come potete vedere dal piano di montaggio, la demoboard fa uso di molti componenti SMD e pochi componenti ad inserzione, pertanto è conveniente iniziare il montaggio proprio dai componenti SMD, per i quali si deve operare con calma e cautela, facendo uso di un saldatore a punta molto fine e di flussante in gel. Avendo a disposizione il PCB, potete iniziare il montaggio partendo dal microcontrollore, per proseguire con la EEPROM, il regolatore di tensione ed infine l’interfaccia USB (FT232RL della FTDI). Mentre EEPROM e regolatore switching hanno i pin abbastanza distanti e pertanto non difficili da saldare, il convertitore RS232 e il PIC hanno una piedinatura molto “densa” quindi è necessario che li saldiatelo con dello stagno fine (meno di 0,5 mm) di eccellente qualità utilizzando una punta molto sottile (l’optimum sarebbe disporre di una stazione di saldatura da 30/40 W con punte intercambiabili, sulla quale montarne una da 0,2 o 0,3 mm). Completato il montaggio degli integrati, procedete con i connettori, i pulsanti, i condensatori elettrolitici e l’induttanza. Terminate con i due diodi LED, il connettore USB e quello di alimentazione. Prima di inserire il display, applicate ai morsetti PWR una tensione continua stabilizzata di circa 8÷10 V e assicuratevi che la tensione in uscita dallo step-down (piedino positivo di C3) sia stabile a +3,3 Vdc. Programmate quindi il microcontrollore (tramite un programmatore ICSP) con il firmware FT707.hex, scaricabile dal sito della rivista.

Funzioni dei pulsanti

Come detto in precedenza, sulla demoboard sono presenti cinque pulsanti: di seguito commentiamo le varie funzioni assegnate ad ognuno di essi. Durante la lettura, aiutatevi con la Figura 9, che mostra dove sono localizzati i pulsanti.

- **Up (P4):** pulsante per la gestione dello scorrimento verticale dell’immagine. Ad ogni pressione l’immagine si sposta verso l’alto. Questo comando è attivo soltanto durante la



visualizzazione di una immagine. Serve anche per aumentare “**Contrasto**” e “**Luminosità**” quando viene aperta la pagina relativa.

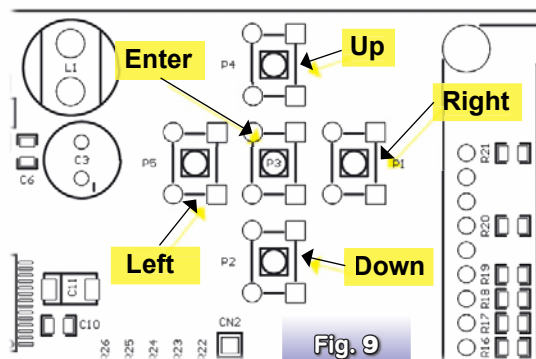
- **Down (P2)**: pulsante per la gestione dello scorrimento verticale dell’immagine. In modo simile al precedente, ad ogni pressione sposta l’immagine verso il basso. Riduce inoltre il valore di luminosità e contrasto.
- **Right (P1)**: pulsante per la navigazione tra le cinque pagine a disposizione. Ad ogni pressione viene richiamata la pagina successiva: le prime quattro sono immagini con dimensione da 128x64 pixel, mentre la quinta consente di regolare il contrasto e la luminosità, nonché di visualizzare i font realizzati per il display della Ritek. Infatti compaiono le scritte “**Luminosità**” e “**Contrasto**”. La sesta pressione richiama la prima pagina (Figura 10).



- **Left (P5)**: esattamente come Right, con la differenza che ad ogni pressione viene richiamata la pagina precedente. Dopo la prima immagine viene caricata la schermata per le regolazioni.
- **Enter (P3)**: pulsante per la gestione di comandi aggiuntivi applicabili alle immagini. Ad ogni pressione si esegue una rotazione dell’immagine di 90° mentre alla quinta pressione si attiva lo scorrimento orizzontale, funzione quest’ultima che rimane attiva finché non viene nuovamente premuto il pulsante “**Enter**” oppure non si esegue un cambio pagina. Quando è attiva la pagina delle regolazioni, questo pulsante permette di alternare la regolazione di luminosità e contrasto.



Fig. 10



Disposizione dei pulsanti di controllo.

Il driver SSD1303T

Nella prossima puntata tratteremo in modo dettagliato ed esaustivo le funzioni messe a disposizione dal driver grafico SSD1303T (presente sul display della Ritek) e della libreria firmware che abbiamo realizzato per gestirlo; prima è tuttavia necessario conoscere, almeno sommariamente, la struttura del chip: pertanto in questo paragrafo ci occuperemo dell’hardware presente in questo driver.

L’SSD1303T è un integrato in tecnologia CMOS per la gestione dei display di tipo OLED/PLED (Organic/Polymer Light Emitting Diode) ed è in grado di controllare display che abbiano una risoluzione massima di 132x64 pixel. Esso permette di regolare sia il contrasto che la luminosità (vedremo nella prossima puntata che per fare ciò sono disponibili due istruzioni specifiche); il chip dispone di una memoria RAM integrata, necessaria per memorizzare e visualizzare le immagini sull’area grafica.

Tale RAM è sufficientemente grande per poter gestire un unico livello grafico, ma, di contro (come avevamo già anticipato), non abbiamo a disposizione una memoria CGROM (ricordate che si tratta della memoria che contiene i font caratteri?) e quindi la tabella caratteri deve essere costruita esternamente, tipicamente nel micro che controlla il display.

Le caratteristiche salienti del controller grafico sono state riassunte nel riquadro riportato a pagina 93 mentre lo schema a blocchi di Figura 11 mostra la struttura interna del chip SSD1303T: appare subito evidente (nella parte in alto a sinistra della figura) il blocco che si occupa dell’**interfaccia** con un microcontrollore esterno, nel nostro caso il PIC18LF4620.

Oltre all’interfaccia parallela a 8 bit, sono disponibili i seguenti pin di controllo:

- **RES#**, pin di reset - è attivo basso quindi per uscire dalla condizione di reset è

- Massima risoluzione grafica supportata: 132 x 64 pixel;
- Possibilità di gestire un'area colore con un massimo di 4 colori selezionabili. 64 livelli di regolazione per colore;
- Alimentazione logica di controllo: 2,4 ÷ 3,5 Vdc;
- Alimentazione analogica per la gestione del vetrino: 7,0 ÷ 16,0 Vdc;
- Dimensione memoria RAM grafica: 132 x 64 bit, per un totale di 1056 byte;
- Regolazione di contrasto e luminosità: hardware, con controllo software a 256 livelli;
- Interfaccia micro: parallela a 8 bit sia in modalità 8080 che 6800; seriale unidirezionale;
- Possibilità di eseguire la rimappatura delle colonne e delle righe;
- Gestione scorrimento verticale e orizzontale;
- Gamma di temperatura di esercizio: -40°C ÷ +90°C.

Specifiche tecniche
SSD1303T

necessario applicare un livello logico alto (+3,3 Vdc).

- **CS#**, pin di abilitazione dell'interfaccia di comunicazione parallela - è attivo basso e quindi, per attivare l'interfaccia di comunicazione e uscire dalla condizione di alta impedenza del bus dati, è necessario applicare ad esso un livello logico basso.

La condizione di alta impedenza è utile quando il bus è condiviso con altri dispositivi quali ad esempio dei latch di lettura o scrittura, interfacce di vario tipo e periferiche in genere. Si ricordi, infatti, che quando un bus è condiviso da più dispositivi elettronici, il micro può accedere ad uno solo alla volta e, per farlo, occorre forzare tutti gli altri in modalità "alta impedenza", condizione che simula la sconnessione fisica del dispositivo dal bus.

- **D/C**, pin di discriminazione tra comando o dato - se si trova a livello logico alto, il driver attende un dato da scrivere nella memoria RAM, viceversa si aspetta di ricevere un comando da inviare poi alla relativa sezione di decodifica.
- **E (RD#)**, pin di abilitazione scrittura - se il driver è configurato per operare in modalità parallela 6800 (standard Motorola) questo pin è utilizzato come segnale di Enable, ossia tutte le volte che esso passa da livello logico basso a livello logico alto, obbliga il controller grafico ad acquisire il dato presente sul bus, dato che verrà poi usato per una eventuale

operazione di lettura o scrittura a seconda dello stato del pin R/W. Nel caso in cui l'interfaccia sia di tipo 8080, il segnale E (RD#) viene utilizzato solo per le operazioni di lettura della memoria RAM. La lettura del dato avviene durante la transizione da livello logico basso a livello logico alto mentre il pin R/W (WR#) si trova a livello logico alto.

- **R/W (WR#)**, pin di controllo Read/Write o solo Write (con l'interfaccia impostata in standard Motorola -6800): questo pin è utilizzato per discriminare tra le operazioni di lettura e quelle di scrittura. In modalità Intel (8080), invece, è utilizzato per le sole operazioni di scrittura. La scrittura del dato avviene durante la transizione ➤

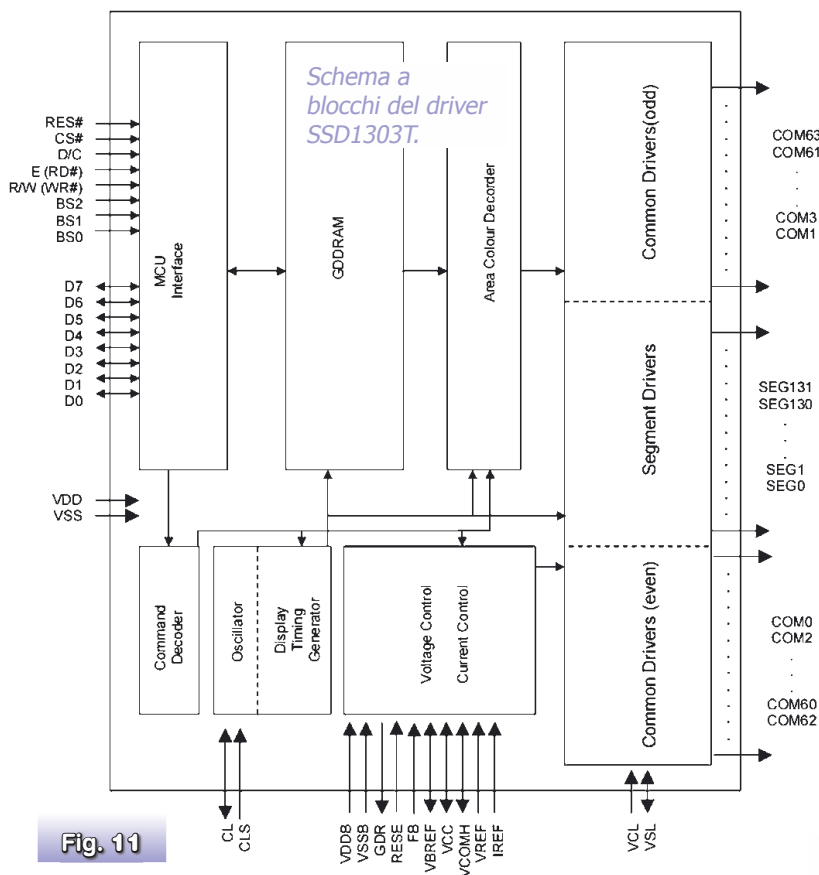


Fig. 11

Tabella 2 - Selezione interfaccia SSD1303T

	Interfaccia parallela 6800	Interfaccia parallela 8080	Interfaccia seriale
BS0	0	0	0
BS1	0	1	0
BS2	1	1	0

da livello logico basso a livello logico alto del pin, mentre il segnale presente sul pin E (RD#) è stabile a livello logico alto.

• **BS0, BS1, BS2** - questi pin servono per selezionare il tipo di interfaccia da utilizzare per comunicare con un microcontrollore esterno. Nel display da noi utilizzato, il driver è impostato per funzionare con l'interfaccia 8080 senza possibilità di modifica perché la casa costruttrice non ha reso disponibili sul connettore tali pin. A titolo informativo, la Tabella 2 mostra come impostare tali pin per ottenere l'interfaccia desiderata (6800, 8080 oppure seriale).

In Figura 12 abbiamo riportato il diagramma temporale da rispettare affinché il driver (come tutti quelli analizzati fino ad oggi) possa capire i comandi e i dati che gli vengono inviati dal microcontrollore. È fondamentale rispettare tutti i tempi indicati sul datasheet: se si utilizzano microcontrollori molto veloci, cioè con tempo di esecuzione di una istruzione inferiore a 100 nsec, è necessario introdurre delle istruzioni di "Nop" per allungare i tempi e rispettare il diagramma temporale. Attenzione a un particolare: qui non abbiamo riportato la tabella dei tempi massimi e minimi (presente comunque sul datasheet) poiché nella nostra applicazione, il microcontrollore ha un tempo di esecuzione delle istruzioni di 200 nsec e quindi ben al di sopra del limite minimo del driver. Non è stato quindi necessario aggiungere delle istruzioni "Nop".

Il blocco "**Command Decoder**", come suggerisce il nome stesso, si occupa di interpretare i comandi ricevuti dall'interfaccia, sempre che il pin D/C sia a livello logico basso.

La sezione "**Oscillator & Display Time Generator**" è il gruppo che ge-

nera il segnale di clock necessario al funzionamento del controller. L'oscillatore è un semplice RC.

Il blocco "**Voltage Control & Current Control**" si occupa di gestire le alimentazioni necessarie al corretto funzionamento di tutti le sezioni funzionali del driver. Inoltre contiene l'elettronica del convertitore DC/DC che, abbinata a pochi componenti esterni, permette di ottenere i +12 Vdc necessari ad alimentare il driver del vetrino del display. Nel nostro caso, tutti i componenti esterni necessari per realizzare il DC/DC converter sono già a bordo del PCB del display Ritek.

La "**GDDRAM**" è la memoria su cui viene caricata l'immagine che si vuole visualizzare: è una matrice composta da 132x64 bit per un totale di 1056 Byte.

I rimanenti blocchi "**Common Drivers & Segment Drivers**" servono per controllare il livello di accensione dei pixel del vetrino OLED.

Note finali

Giunti a questo punto, riteniamo di aver fornito le informazioni necessarie per poter operare correttamente. Possiamo pertanto considerare conclusa la descrizione di tutto ciò che riguarda l'hard-ware dei display OLED. Nella prossima puntata, spiegheremo in dettaglio il set di comandi specifici per controller SSD1303T10, mentre nella successiva (conclusiva del corso LCD), analizzeremo la libreria firmware sviluppata "ad hoc" per l'SSD1303T.

