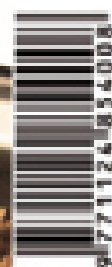


Electronica In

Mensile di elettronica applicata, attualità scientifica, novità tecnologiche.

119

Termostato con controllo GSM



Caricabatterie programmabile professionale

Trasmittitore FM per lettori CD e mp3

Allarme con scritte scorrevoli

Mambo, la prima applicazione pratica

Porta USB Host per sistemi embedded

Interfaccia Bluetooth per sistemi Velbus

Oscilloscopio con PC, DLL e software di esempio

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

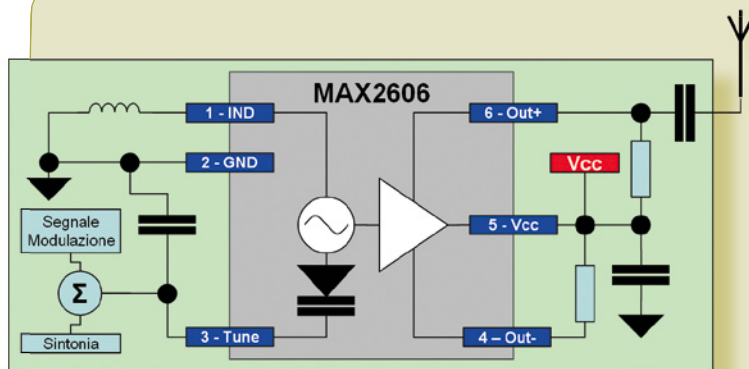
Trasmittitore FM per lettori CD e MP3

di Davide Scullino

Trasmette nella banda compresa tra 88 e 108 MHz il segnale di lettori CD ed MP3, registratori a cassette e simili con una potenza sufficiente a coprire una decina di metri. Particolarmente indicato per impieghi in auto, dove permette di ascoltare, ad esempio, la musica in MP3 su qualsiasi autoradio.

Probabilmente vi sarà già capitato di voler ascoltare in altoparlante il segnale musicale di un piccolo apparecchio BF progettato per l'uso con la cuffia, ma di non trovare, nell'impianto stereo, una presa di ingresso adatta dalla quale far entrare la componente audio: succede tipicamente con gli hi-fi compatti che oggi si trovano a poco prezzo anche nei centri commerciali, con i radioregistratori, ma anche con le autoradio tradizionali, prive di ingresso ausiliario (jack da 3,5 mm sul frontale). In un caso del genere, per risolvere il problema vi sono due alternative: manomettere l'apparecchio per

rintracciare il percorso del segnale verso l'amplificatore di potenza che pilota gli altoparlanti, quindi collegare una presa e un deviatore che permetta di scegliere la fonte BF esterna; escogitare qualcosa che permetta di introdurre l'audio senza modificare nulla. Nella prima eventualità, alle difficoltà oggettive, si aggiunge il problema che se l'impianto stereo è stato acquistato da meno di due anni si perde il diritto alla garanzia. Allora non resta che la seconda soluzione, per la quale vi proponiamo un dispositivo elettronico ad hoc: si tratta di un piccolo trasmettitore radio che opera nella gam- ➤



Si tratta di un trasmettitore integrato di piccolissima potenza (-10 dBm) operante fra 70 e 150 MHz in un campo di frequenze definito essenzialmente dai valori del condensatore di bypass (quello che collega a massa il pin 3) e dell'induttanza connessa al piedino 1. Internamente dispone di un oscillatore simile al Colpitts, nel quale il circuito accordato è composto dal predetto induttore e da un diodo varicap la cui polarizzazione si può modificare dall'esterno tramite il piedino 3 (TUNE); il varicap è un particolare diodo nel quale la capacità parassita di giunzione varia in maniera significativa in funzione del valore della tensione di polarizzazione inversa applicata ai suoi

capi. L'uscita dell'oscillatore è costituita da uno stadio differenziale progettato per pilotare ogni genere di antenna, a stilo o a dipolo aperto/ripiegato, di impedenza compresa fra 75 e 300 ohm; per l'uso, va ricordato che i due piedini di uscita (OUT+ e OUT-) devono essere polarizzati mediante due resistenze o impedenze da collegare al positivo di alimentazione, anche nel caso venga adottata la configurazione single-ended, ovvero anche nel caso l'antenna venga collegata solamente all'uscita OUT+.

Quanto alla scelta dei componenti del circuito accordato, la Maxim raccomanda di mantenere l'induttore fra 0,15 e 0,82 microhenry ed il condensatore di bypass sopra i 680 picofarad; al minimo valore di induttanza corrisponde la massima frequenza di oscillazione (150 MHz) e al massimo la frequenza minore (70 MHz). Va comunque detto che la frequenza di lavoro dipende in maniera proporzionale dalla polarizzazione del pin 3, al quale può essere applicata una tensione continua compresa tra 0 a 5 V. Per modulare la portante a radiofrequenza è sufficiente sovrapporre alla tensione continua il segnale audio: l'ampiezza di quest'ultimo non deve eccedere il 14,5 % della tensione continua, pena una sovramodulazione della portante RF con conseguente distorsione.

ma FM, con le stesse modalità delle emittenti radiofoniche per le quali sono progettati gli impianti hi-fi, le radio portatili, i radioregistratori e le autoradio; basta applicare al suo ingresso il segnale BF che si desidera ascoltare in altoparlante e il circuito lo trasmette entro un raggio di una decina di metri a qualsiasi radiorecettore FM (88÷108 MHz) che si trovi nei paraggi. Il tutto, senza manomettere alcunché e senza perdere il tempo che andrebbe dedicato all'eventuale modifica: basta alimentare il trasmettitore con una pila a 9 volt, collegarne con un cavetto audio l'ingresso all'uscita della fonte BF, accendere l'autoradio e sintonizzarla sul canale voluto; gli altoparlanti diffonderanno la musica della fonte BF collegata al TX esattamente come quella di una qualsiasi emittente FM.

Il circuito

Vediamo dunque lo schema elettrico del minitrasmittitore radio, per

comprendere come funziona e in che modo utilizzarlo al meglio; il tutto gravita attorno ad un circuito integrato SMD prodotto dalla Maxim e siglato MAX2606, contenente un oscillatore sinusoidale basato sulla struttura Colpitts, ma leggermente modificato, un modulatore di frequenza e uno stadio di uscita di tipo differenziale, al quale collegare l'antenna trasmittente. Il componente, nell'attuale configurazione, irradia nell'etere una potenza molto ridotta, dell'ordine di -10 dBm (meno di 1 milliwatt) cosa più che soddisfacente per il nostro scopo, dato che ci serve trasmettere nel raggio di una decina di metri e comunque entro una stanza. Il MAX2606 può lavorare a frequenze comprese entro la gamma fra 70 e 150 MHz e richiede una tensione d'alimentazione dell'ordine di 2,7÷5,5 V (la corrente assorbita è di poco superiore a 2 milliampere).

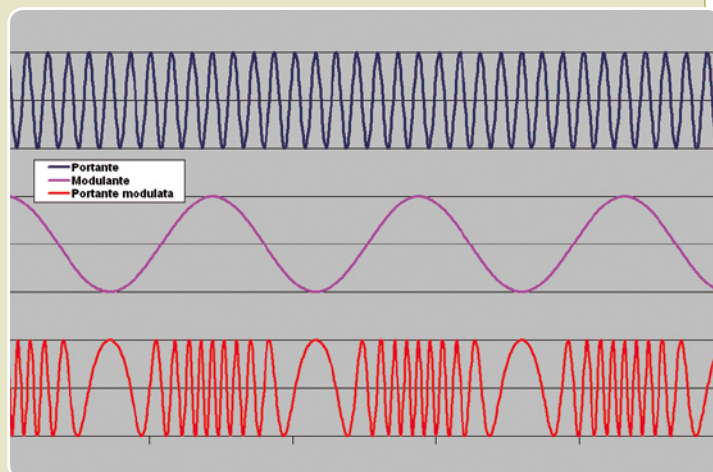
L'oscillatore ha una configurazione di tipo Colpitts, con una variante: il circuito di accordo (quello che de-

cide la frequenza di lavoro) viene chiuso attraverso il positivo di alimentazione e la massa. Proviamo a spiegarci meglio: i componenti che impostano la frequenza dell'oscillatore sono l'induttanza L1 e il diodo varicap interno al chip (un varicap è un particolare diodo a giunzione PN del quale si sfrutta la capacità parassita, che viene fatta variare modificando la tensione di polarizzazione inversa) il cui catodo è accessibile dal piedino 3 (TUNE); formano un circuito antirisonante che in continua è aperto, mentre viene chiuso in alta frequenza mediante il condensatore C2, che già a qualche MHz presenta un'impedenza trascurabile e mette a massa il capo della bobina collegato al pin 3.

Il condensatore serve perché non è possibile cortocircuitare verso massa la linea TUNE, in quanto il MAX2606 prevede che la frequenza di lavoro dell'oscillatore venga decisa giocando sul potenziale applicato al piedino 3, ovvero tra il pie-

La modulazione in frequenza

Nella radiodiffusione ad alta fedeltà, che avviene nella cosiddetta banda FM (estesa fra 87,5 e 108 MHz), il segnale audio viene trasmesso sfruttando la modulazione di frequenza; in breve, l'ampiezza della musica o della voce fa deviare la frequenza originaria del trasmettitore entro quella che viene detta larghezza del canale e che vale tipicamente ± 75 kHz (o ± 50 kHz nell'FM stretta). Siccome la deviazione è proporzionale alla variazione di ampiezza del segnale audio, è possibile trasportare quest'ultimo certi che un ricevitore adatto possa poi estrarlo, riconoscendo la corrispondente deviazione di frequenza. L'apparecchio ricevente consta di uno stadio di sintonia, composto da un circuito accordato, e da convertitore di frequenza alla cui uscita è presente un segnale di 10,7 MHz, quale che sia la frequenza del segnale radio sintonizzato. La presenza di uno stadio a frequenza intermedia consente di aumentare la sensibilità e la selettività dell'apparato radio. In questo modo si ottiene una banda passante poco più larga della massima deviazione (150 kHz). Segue quindi lo stadio demodulatore, un circuito accordato con la media frequenza, realizzato in diversi modi (il più usato è quello a quadratura) che



presenta tensione d'uscita nulla quando l'onda radio sintonizzata è stabile e fornisce, invece, un segnale variabile, se la radiofrequenza in antenna e quindi la media frequenza deviano rispetto al valore originario. All'uscita del demodulatore FM, l'ampiezza è tanto maggiore quanto più ampia è la deviazione: insomma, il ricevitore ricava un segnale audio analogo a quello utilizzato per modulare la portante.

dino 3 e la massa. Restando in tema, diciamo che nella nostra applicazione i componenti dell'oscillatore (essenzialmente L1 e C2) sono stati dimensionati affinché l'integrato lavori a una frequenza compresa tra 75 e 130 MHz; l'esatta frequenza di oscillazione, quindi quella trasmessa dall'antenna, dipende dalla tensione applicata al piedino TUNE, ovvero dalla posizione del cursore del trimmer R1. Quest'ultimo componente provvede alla sintonia del trasmettitore, nel senso che permette di scegliere su quale canale si deve trasmettere; abbiamo preferito evitare la soluzione a frequenza fissa perché a seconda della località e delle condizioni di impiego del dispositivo, ci può essere qualche difficoltà a trovare libera una determinata frequenza della gamma FM, quindi è meglio permettere un'ampia scelta che consente di adattarsi ad ogni situazione. Notate che è stato scelto un trimmer multigiri per consentire di regolare con estrema precisione la frequenza

di trasmissione e quindi di sintonizzare finemente il canale desiderato. Il segnale radio dell'oscillatore viene applicato allo stadio di uscita, che nel MAX2606 è di tipo differenziale ed è progettato per alimentare antenne a dipolo, eventualmente ripiegato; tuttavia nel nostro caso sfruttiamo la configurazione single-ended, nel senso che preleviamo il segnale RF da irradiare tramite l'antenna (un semplice spezzone di filo in rame) da una sola delle uscite: precisamente, dal piedino 6. L'altra uscita (piedino 4) è inutilizzata ma viene ugualmente mantenuta polarizzata mediante R5, in osservanza alle disposizioni del costruttore.

Come avviene la modulazione

Finora è stato spiegato come funziona il trasmettitore una volta scelti il campo di frequenze e il canale su cui trasmettere; resta, però, da vedere come si manda nell'etere il segnale audio. Dando uno sguardo

allo schema elettrico, a sinistra vediamo gli ingressi (IN) che sono due in previsione dell'abbinamento a dispositivi stereofonici; ad essi si applica la componente audio, che può avere un'ampiezza compresa tra poche decine di millivolt e qualche volt. L'ampia tolleranza è assicurata dalla presenza del trimmer R2, che serve a dosare a piacimento il livello dell'audio.

Prima di procedere, va fatta una precisazione: il trasmettitore funziona in mono, nel senso che può trasmettere una sola componente audio e non due, come farebbe un apparato stereo; i due ingressi (che poi fanno capo ad un'unico jack a due canali) servono solo per adattarsi ai dispositivi che si collegheranno, che potranno essere mono o stereofonici. Nel primo caso basta inviare il segnale su un contatto dello spinotto (rispetto a massa) mentre nel secondo i segnali dei due contatti L ed R vengono fatti confluire, opportunamente miscelati (R3 ed R4 formano

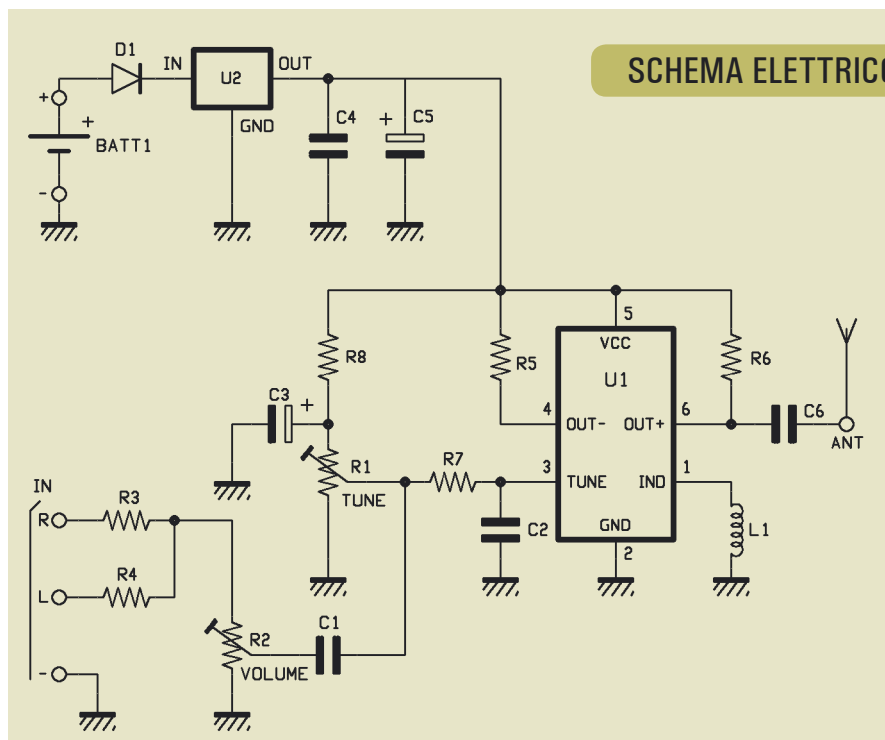
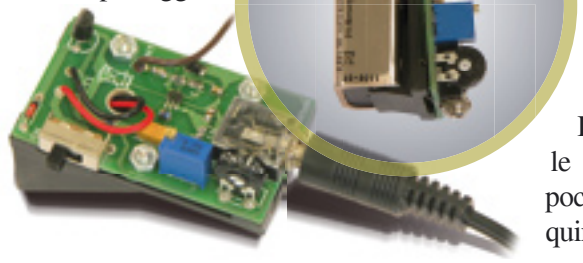
>

un semplice mixer) all'ingresso del controllo di volume R2. Per trasmettere un segnale stereofonico avremmo dovuto complicare il circuito, perché sarebbe stato necessario aggiungere un encoder.

La modulazione di frequenza viene operata giocando sul fatto che nel MAX2606 la frequenza di lavoro si imposta con la tensione applicata al piedino 3; ora, se è vero che applicando un potenziale costante si ottiene la trasmissione di una componente RF a frequenza stabile, sovrapponendo alla continua una tensione variabile quale quella del segnale, si realizza una vera e propria modulazione di frequenza: l'onda radio varia leggermente la propria frequenza e lo fa secondo un andamento che segue quello del segnale musicale.

Naturalmente la deviazione della frequenza deve restare entro certi limiti (in banda FM, tipicamente ± 75 kHz) altrimenti un canale può sconfinare in quello vicino e comunque il ricevitore riceve il segnale come distorto, in quanto il suo stadio d'antenna è tanto selettivo da accettare l'escursione di frequenza prestabilita per la radiodiffusione. In conseguenza di ciò, R2 va regolato opportunamente.

Concludiamo la descrizione del trasmettitore dicendo che il tutto si alimenta a 9 Vcc, prelevati da una pila o da qualsiasi alimentatore in grado di erogare una corrente di circa 10 mA; a ricavare la tensione stabilizzata necessaria al funzionamento del MAX2606, provvede un regolatore 78L05, preceduto dal diodo D1, che serve a protegger-



SCHEMA ELETTRICO

re il circuito nel caso, per errore, si applichi l'alimentazione con polarità invertita. Notate che è necessario il regolatore perché se la tensione di alimentazione del MX2606 non è sufficientemente stabile, la frequenza di trasmissione si sposta e quindi il ricevitore finisce per perdere il segnale del nostro piccolo trasmettitore.

Realizzazione pratica

Per quanto riguarda gli aspetti costruttivi, diciamo che realizzare il minitrasmettitore è abbastanza semplice nonostante alcuni componenti utilizzati siano in tecnologia SMD. Munitevi quindi di un saldatore a punta fine, di stagno di buona qualità, di flussante in gel e di tanta pazienza. Si parte preparando il piccolo circuito stampato, mediante fotoincisione, ricavando la pellicola da una buona fotocopia della traccia lato rame scaricabile dal nostro sito (www.elettronica.in.it).

Incisa la basetta, si forano le piazzole corrispondenti ai pochi componenti tradizionali, quindi si procede alla saldatura

degli elementi in SMD: il primo deve essere l'integrato della Maxim, che consigliamo di posizionare ben centrato nelle rispettive piazzole e stagnare partendo da un pin per lato. Si può quindi procedere con i restanti componenti SMD, quindi con il diodo D1, il regolatore 78L05 (da posizionare in modo che il suo lato piatto sia rivolto al vicino lato corto della basetta) la presa jack 3,5 mm stereo, il piccolo deviatore a slitta e i due trimmer.

È bene che tutte le saldature siano fatte con un saldatore di piccola potenza (25 W) a punta sottile, utilizzando, in special modo per la componentistica SMD, filo di lega saldante del diametro di 0,5 mm e una quantità di flussante adeguata.

Per rendere l'insieme più compatto, suggeriamo di fissare il circuito stampato al retro di un alloggiamento con presa per pile a 9 volt, stagnando poi i fili positivo (rosso) e negativo (nero) nelle piazzole corrispondenti e cercando di non scambiare il positivo con il negativo.

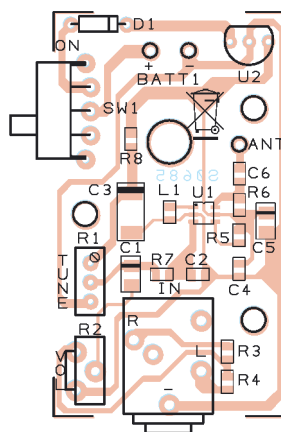
L'antenna del circuito può essere uno stilo telescopico o un semplice spezzone di filo di rame con guaina, lungo circa 1 metro.

ELENCO COMPONENTI:

R1: Trimmer multigiri 100 kohm
 R2: Trimmer MV 10 kohm
 R3: 22 kohm (0805)
 R4: 22 kohm (0805)
 R5: 1 kohm (0805)
 R6: 1 kohm (0805)
 R7: 270 ohm (0805)
 R8: 4,7 kohm (0805)
 C1: 470 nF ceramico (0805)
 C2: 2,2 nF ceramico (0805)
 C3: 10 μ F 25 VL tantalio (TAJ-C)
 C4: 100 nF multistrato (0805)
 C5: 1 μ F 25 VL tantalio (TAJ-A)
 C6: 1000 pF ceramico (0805)
 D1: 1N4148



U1: MAX2606 (SOT23-6)
 U2: 78L05 (TO92)
 L1: 390 nH (0805)
 SW1: Deviatore slitta 90°



Varie:
 - Presa jack stereo da cs
 - Porta batteria 9V
 - Circuito stampato

Il modulatore FM, pur utilizzando alcuni componenti a montaggio superficiale (tra cui l'integrato della Maxim), può essere realizzato senza particolari difficoltà. Consigliamo l'impiego di un saldatore con punta molto fine da utilizzare in abbinamento a stagno di buona qualità e fluxante in gel.

Come usarlo

Una volta completato e verificato che non sono presenti saldature fredde e baffi di stagno, il TX funziona subito, perché non richiede alcuna operazione preliminare di taratura: basta inserire una pila a secco da 9 volt nell'apposito alloggiamento (non preoccupatevi per la polarità: entra solo in un modo) e inserire nella presa jack lo spinotto del cavetto mono o stereo che usate per portare il segnale dall'apparecchio audio che intendete ascoltare attraverso la radio; fatto ciò, bisogna mettere provvisoriamente al minimo il controllo di volume (cursore dell'R2 tutto verso massa) in modo da disinserire l'audio. Ora non resta che

mettere in funzione l'autoradio o il sintonizzatore del caso, scegliere la banda FM, cercare un canale vuoto, ossia una zona nella quale non si sentano trasmissioni, ma solo il fruscio, poi ruotare, in un verso o nell'altro, il perno dell'R1, fino a trovare il punto in cui scompare il fruscio e l'altoparlante tace; quando ciò accade, significa che il ricevitore FM sente la portante della nostra trasmittente. A questo punto si può alzare gradualmente il volume (R2) fino ad ascoltare, ad un livello simile a quello delle altre emittenti, la musica dell'apparecchio collegato alla presa jack del trasmettitore, cercando di non esagerare, altrimenti l'ascolto risulterà distorto.

Se occorre, ritoccate leggermente la posizione del trimmer di sintonia (R1). Nell'effettuare le operazioni di ricerca della stazione FM, accertatevi che il ricevitore radio non abbia il muting e che se lo ha, che sia disinserito. Nel nostro paese, con tutte le stazioni radio che operano su questa banda, la ricerca di un "buco" è abbastanza problematica: munitevi perciò di una buona dose di pazienza e studiate con attenzione dove fare operare il microtrasmettitore. Per quanto riguarda l'antenna è sufficiente uno spezzone di filo della lunghezza di una trentina di centimetri: l'eventuale impiego di un'antenna a stilo accordata non garantisce prestazioni significativamente superiori.

per il

MATERIALE

Tutti i componenti utilizzati in questo progetto sono facilmente reperibili in commercio con l'eccezione del circuito stampato che va autocostruito partendo dal disegno del master in scala 1:1 scaricabile dal sito della rivista (www.elettronica.in.it). Il circuito integrato MAX2606 e l'induttanza L1 da 390 nH costano rispettivamente 3,00 e 0,50 Euro. I prezzi si intendono IVA compresa.

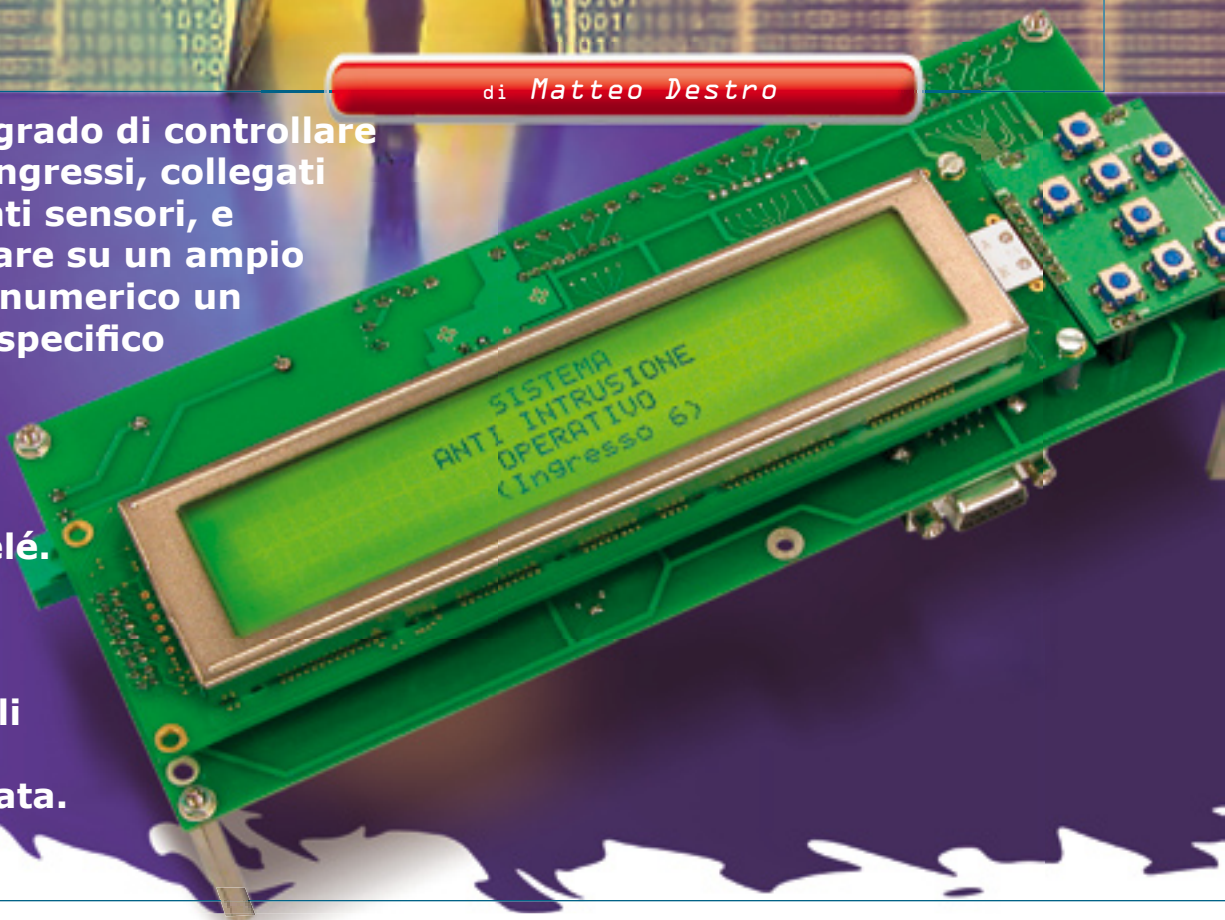
Il materiale va richiesto a: Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)

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

Allarme con scritte scorrevoli

di Matteo Destro

Sistema in grado di controllare fino a 512 ingressi, collegati ad altrettanti sensori, e di visualizzare su un ampio display alfanumerico un messaggio specifico per ogni ingresso. In caso di allarme si attiva un relé. Messaggi e ingressi facilmente configurabili tramite PC. Prima puntata.



Nell'ambito della gestione e del controllo di impianti con un elevato numero di sensori, soprattutto industriali ma anche civili, spesso è necessario associare una segnalazione visiva o un avviso ad uno specifico evento. Si pensi, ad esempio, ad un sistema elettronico in grado di rilevare il livello dell'acqua in un pozzo perdente: quando il livello supera la soglia di guardia, un sensore a ciò preposto si attiva mettendo in atto tutta una serie di azioni ma anche, se collegato ad uno degli ingressi del nostro circuito, determinando la visualizzazione di uno specifico messaggio sul display alfanume-

rico che, nel caso in questione, potrebbe essere del tipo: "Attenzione livello acqua nel pozzo 1 oltre la soglia! Chiamare pronto intervento". Se i pozzi sono più di uno, è possibile richiamare diversi messaggi, ciascuno associato alla relativa centralina di controllo e quindi ad uno specifico pozzo. Un altro esempio potrebbe riguardare una centralina di allarme anti-intrusione. Molte di queste sono provviste di contatti di uscita i quali potrebbero pilotare gli ingressi del nostro dispositivo e quindi richiamare messaggi inerenti ad alcune situazioni rilevate (ad esempio "Attenzione porta garage aperta", oppure >

“Attenzione cancello aperto” ecc). Da quanto appena esposto si può quindi comprendere come le possibili applicazioni siano molteplici. Il nostro visualizzatore di messaggi d’allarme è completamente personalizzabile dall’utente, sia tramite un semplice programma di configurazione per piattaforma Windows, sia direttamente da pannello. In questa prima parte, analizziamo l’hardware mentre nella seconda e ultima parte, che pubblicheremo sul prossimo numero della rivista, analizzeremo il firmware e l’utilizzo pratico.

Schema a blocchi

Come si può intuire osservando lo schema a blocchi di figura 1, il dispositivo è piuttosto complesso pur essendo riconducibile a sei blocchi funzionali che andiamo subito ad analizzare:

- una sezione di alimentazione dalla quale si ricavano i +5 Vdc, necessari per alimentare la logica di controllo, e i +24 Vdc necessari ad alimentare la bobina dell’unico relè presente sulla scheda;
- un’interfaccia seriale per la connessione del dispositivo al PC (MAX202). Essa è necessaria per le operazioni di lettura e scrittura della EEPROM I²C presente sulla scheda;
- un bus a 8 bit per collegare i 3 latch necessari a gestire gli ingressi presenti sulla scheda e il display LCD;
- una tastiera a 8 pulsanti, per la configurazione in locale;
- una memoria EEPROM, interfacciata al microcontrollore tramite il bus I²C.
- un PIC18F4620, che rappresenta il cuore dell’allarme a scritte scorrevoli.

Vediamo ora lo schema elettrico nel dettaglio.

Schema elettrico

Come si può osservare dallo schema, la scheda elettronica comprende

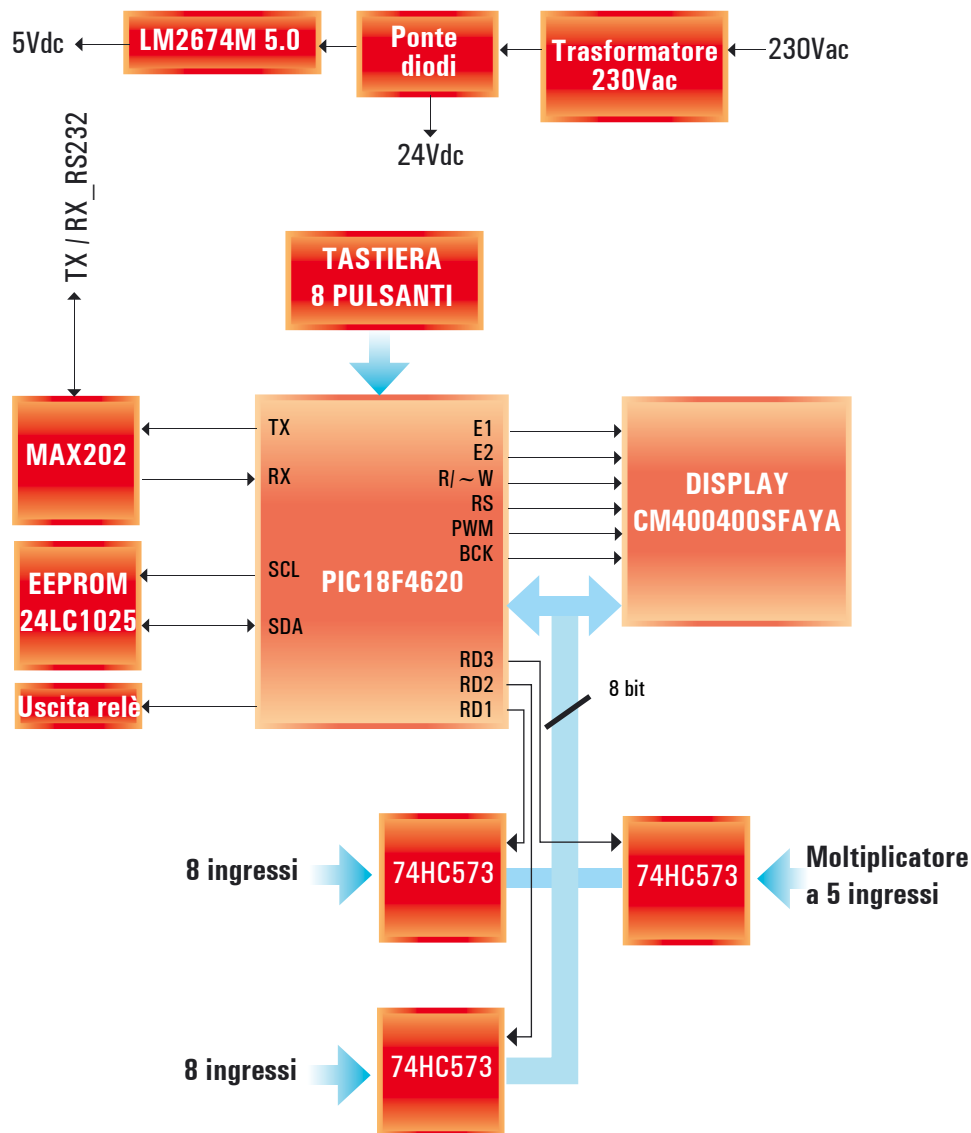


Fig. 1 Schema a blocchi del sistema di visualizzazione messaggi. Tutte le funzioni vengono gestite da un micro PIC18F4620.

l’alimentatore da rete che fa capo al trasformatore con doppio secondario a 9 Vac.

Utilizzando i due secondari è possibile ricavare sia la tensione di +24 Vdc necessaria per alimentare la bobina del relè (uscita del ponte a diodi PD1) sia i +5 Vdc generati dal regolatore di tensione switching LM2674M_5.0. Tra il secondario del trasformatore e l’ingresso del regolatore di tensione (pin 7) è stato interposto un circuito di protezione contro le sovratensioni di rete: esso è composto da un transzorb bidirezionale (D1), un termofusibile (F1) e una resistenza da 10 ohm. In caso di

picchi di tensione presenti sulla rete (che ovviamente vengono riproposti sul secondario del trasformatore di alimentazione anche se attenuati in base al rapporto primario/secondario) interviene il transzorb la cui soglia di innesco è di 33 Vdc. Se l’ampiezza del disturbo è superiore a questo livello, questo elemento entra in conduzione e si comporta come un cortocircuito facendo così intervenire il termofusibile il quale diventa un circuito aperto. In questo modo il dispositivo si spegne.

La sezione di alimentazione è realizzata con un regolatore di tensione switching step-down ad

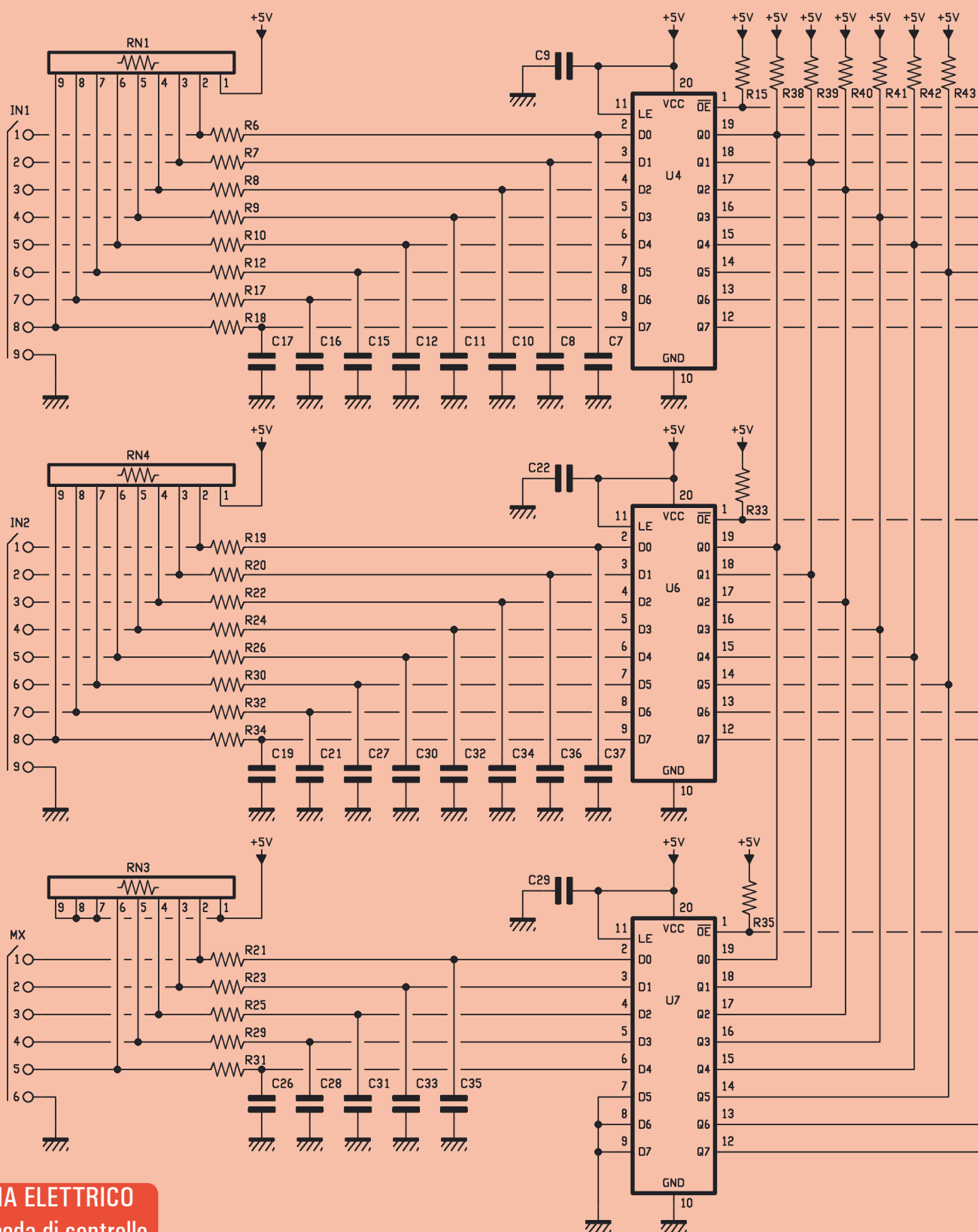
Nel caso di regolatori switching bisogna prestare particolare attenzione alla realizzazione del PCB. Ad esempio il condensatore elettrolitico di ingresso (C39) e il diodo schottky (D2) sono componenti critici e devono essere posizionati il più vicino possibile al pin 7 il primo e al pin 8 il secondo, ciò per ridurre al minimo gli effetti induttivi delle piste. È buona norma realizzare un piano di massa molto esteso (se possibile su entrambi i lati del PCB), il cui compito è duplice: schermare da eventuali disturbi generati dalle commutazioni dello switching e aiutare l'integrato a dissipare il calore prodotto. Il pin 5 (ON/OFF) va lasciato scollegato: se fosse collegato a massa, lo stabilizzatore LM2674M si spegnerebbe.

- Display alfanumerico a 40 caratteri per 4 righe. Totale 160 caratteri per ogni messaggio di allarme;
- Gestione retroilluminazione display LCD (ON/OFF);
- Possibilità di memorizzare fino a 512 messaggi distinti più un messaggio di default (il messaggio di default è visualizzato quando non sono presenti allarmi sugli ingressi). I 512 messaggi vengono memorizzati su una EEPROM esterna da 128kbyte, mentre il messaggio di default è memorizzato nella EEPROM interna del PIC;
- Sedici ingressi digitali diretti per acquisizione eventi dal campo;
- Cinque ingressi ausiliari utilizzati per moltiplicare gli ingressi diretti. Questi ingressi permettono 32 combinazioni binarie che, moltiplicate per i sedici ingressi diretti, danno un totale di 512 possibili allarmi;
- Interfaccia seriale RS232, velocità di trasferimento dati 115.200 Baud sia in lettura che scrittura;
- Alimentazione diretta da rete (230 Vca) attraverso il trasformatore montato direttamente sulla scheda;
- Regolatore di tensione con tecnologia switching;
- Una uscita a relè che viene attivata dopo la rilevazione di un allarme sugli ingressi;
- 8 pulsanti per la gestione dei menu di configurazione. Da pannello è possibile effettuare le seguenti operazioni:
 - Regolare il contrasto dell'LCD alfanumerico;
 - Selezionare i 16 ingressi diretti come normalmente aperti (NO) oppure normalmente chiusi (NC);
 - Stabilire l'intervallo tra un messaggio e il successivo (tempo massimo 10 secondi, minimo 1 secondo);
 - Modificare i 512 messaggi di allarme memorizzati nella EEPROM collegata al bus I²C;
 - Cancellare la coda degli allarmi letti dal campo (lunghezza massima coda 128 messaggi);
 - Visualizzare i messaggi di allarme presenti nella coda.

saggi. Gli otto pulsanti della tastiera sono collegati direttamente alla porta A e alla porta E e sono tutti provvisti di una resistenza di pull-up; il firmware considera premuto un pulsante quando sull'ingresso del PIC è presente un livello logico 0 per più di 50 millisecondi.

Abbiamo detto che il bus dati serve per connettere tra loro le varie periferiche, ma poiché al bus può accedere una sola periferica alla volta, è stato necessario utilizzare dei pin di abilitazione per ogni singolo Latch (CS1, CS2 e CS3, tutti attivi bassi). I latch utilizzati (74HC573D) hanno due pin di abilitazione chiamati



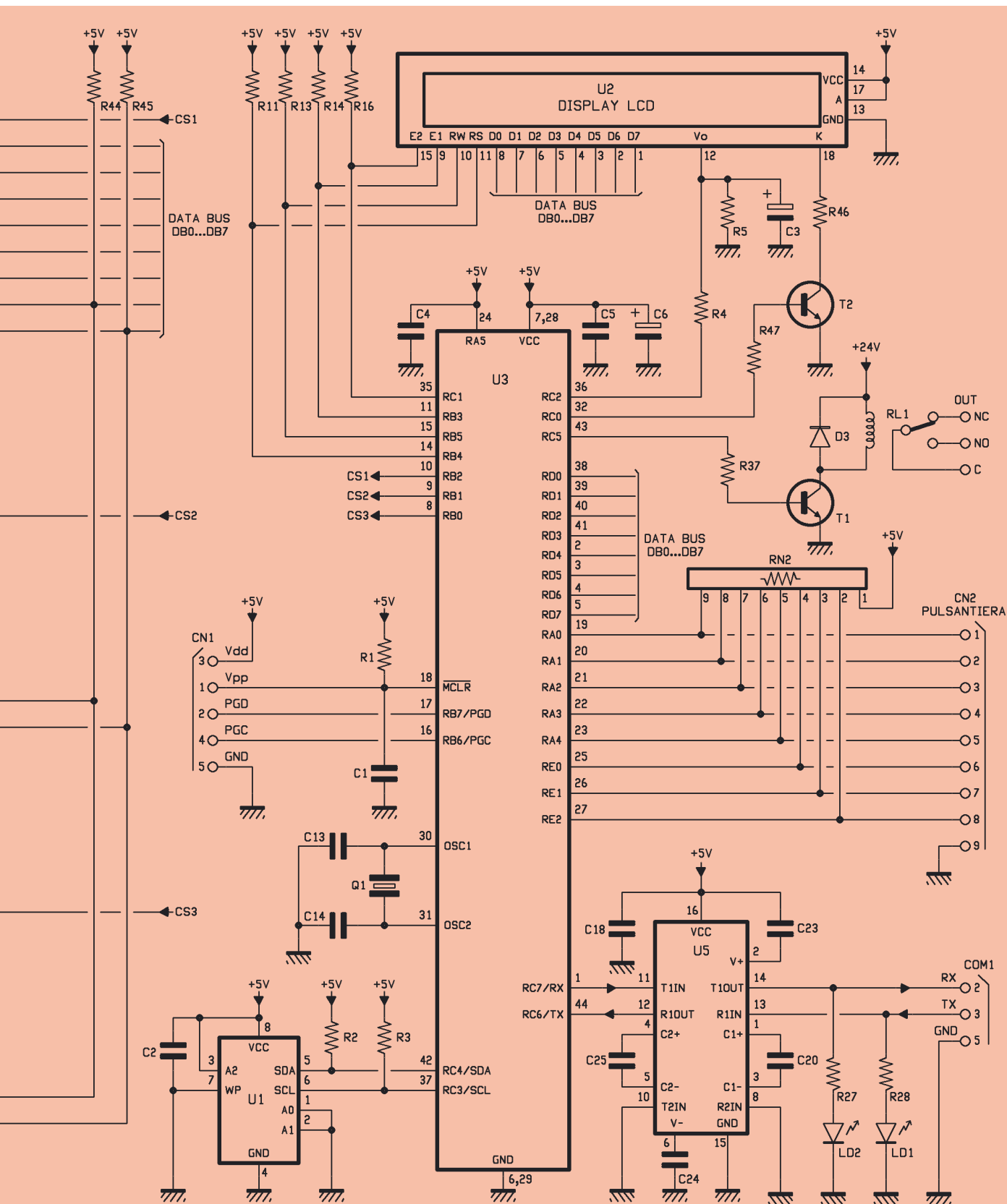


SCHEMA ELETTRICO
della Scheda di controllo

LE (pin 11) e OE (pin 1). Il primo, il Latch Enable (attivo alto), abilita l'acquisizione dei livelli logici presenti sugli ingressi (pin da 2 a 9)

ed è sempre attivo quindi collegato a +5 Vdc. Il secondo è l'Output Enable (attivo basso) e permette il trasferimento sul bus dati a 8 bit

dei livelli logici acquisiti dal latch. Questi ultimi sono controllati dal micro tramite le già menzionate linee CS1, CS2 e CS3 (rispetti-



vamente per i latch U4, U6 e U7). Per proteggere gli ingressi da eventuali disturbi, che potrebbero generare falsi allarmi, è stato inserito un

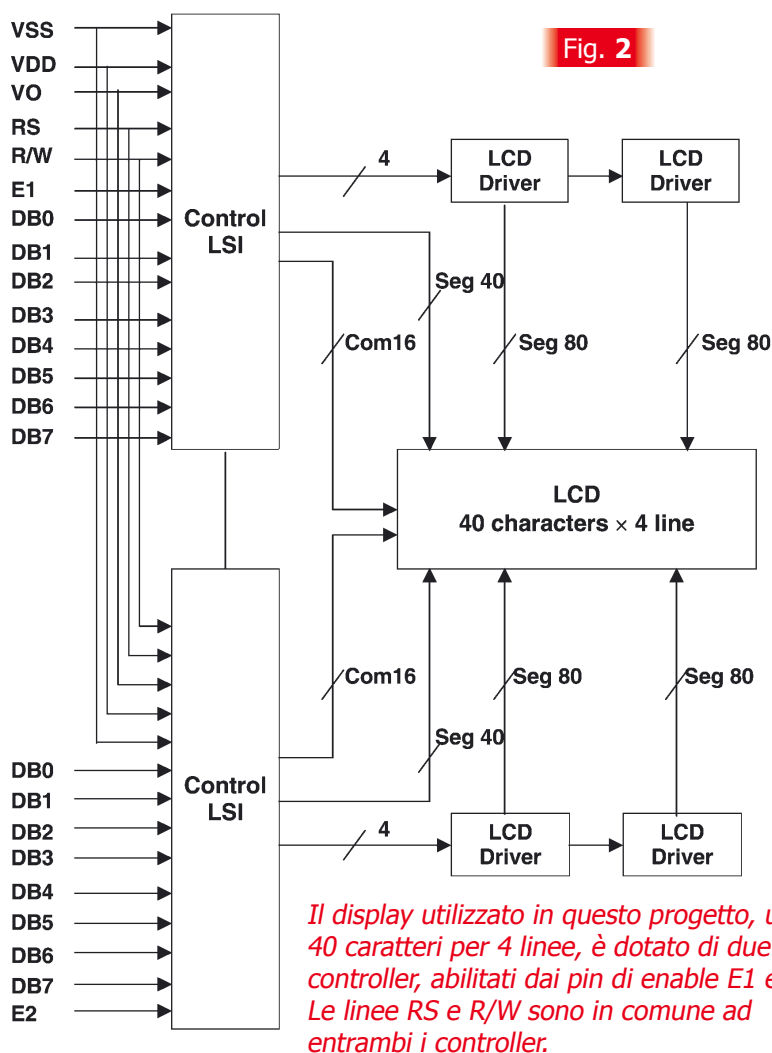
filtro RC per ogni ingresso dei latch (ad esempio R6 e C7 proteggono l'ingresso 1 del latch U4). Le resistenze di pull-up servono per forzare

un livello logico alto nel caso in cui gli ingressi delle morsettiere siano lasciati flottanti (reti resistive RN1, RN3, RN4). Per quanto riguarda il ➤

display LCD, la gestione risulta più complessa del solito a causa della presenza di due pin di Enable (E1 ed E2); in figura 2 riportiamo lo schema a blocchi interno che ci consente di capire meglio come funziona questo dispositivo. Le due linee di Enable, attive a livello logico alto, non devono mai essere attivate contemporaneamente ai segnali per la gestione dei latch (CS1, CS2 e CS3), pena un conflitto sul bus. Ricordiamo che i comandi di enable sono due perché un display da 40 caratteri per 4 righe dispone di due controller e pertanto si comporta come due display da 40 caratteri per 2 righe. I segnali RS e R/W necessari per stabilire l'invio di istruzioni o di dati e per discriminare le operazioni di scrittura da quelle di lettura, sono comuni perché hanno effetto solo sul controllore abilitato tramite l'apposito pin di Enable; E1 (collegato al pin RB3 del micro) e E2 (RC1) non possono essere attivi contemporaneamente.

Il contrasto del display LCD è ottenuto mediante un segnale PWM (ossia un'onda quadra a frequenza costante e duty cycle variabile) generato dal PIC (pin RC2). Questo segnale viene filtrato attraverso le resistenze R4 e R5 e mediante il condensatore C3. Sul pin 12 (Vo) dell'LCD arriva quindi una tensione continua, che può variare tra 0Vdc e +1,2Vdc. È stato inserito un controllo sull'accensione della retroilluminazione che fa capo al pulsante P3 (enter) della tastiera. Tale controllo è costituito dal transistor Q2 e dalle resistenze R46 e R47. Il transistor è comandato dal pin 32 del PIC appartenente alla porta C. Per accendere o spegnere la retroilluminazione, dunque, è necessario premere il pulsante P3.

L'interfaccia di comunicazione seriale utilizza un MAX202E (U5). Questo integrato, a differenza del MAX232, utilizza dei condensatori di 100 nF invece che da 1μF. È evidente che è molto più facile (oltre che più economico) utilizzare e montare

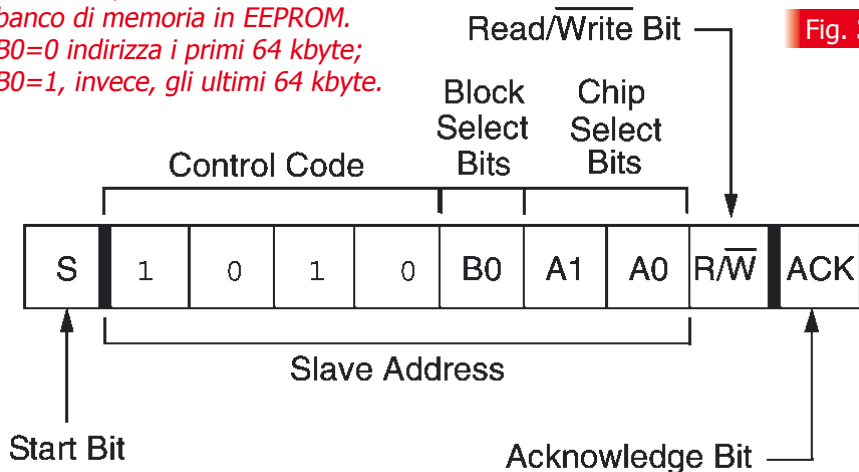


dei condensatori da 100 nF anziché da 1μF, a maggior ragione se, come in questo caso, il montaggio è di tipo SMD. Sulla linea di comunicazione seriale abbiamo previsto due led di segnalazione che ci indicano, con la

loro accensione, che è in corso un trasferimento di dati. I driver contenuti nel MAX202 funzionano come buffer invertenti.

Vediamo ora la memoria EEPROM collegata al bus I²C del PIC18F4620.

Comando per la selezione del banco di memoria in EEPROM. B0=0 indirizza i primi 64 kbyte; B0=1, invece, gli ultimi 64 kbyte.



Si tratta di una memoria da 1 Mbit che consente la memorizzazione di ben 128 kbyte di dati. A titolo informativo, diciamo che sarebbe possibile collegare al bus I²C fino a 4 memorie EEPROM, agendo opportunamente sui due pin di indirizzo A0 e A1 di cui esse dispongono. Il pin A2 va collegato obbligatoriamente a +5 Vdc altrimenti non è possibile scrivere sulla memoria. Per la stessa ragione occorre collegare il pin WP a massa. La figura 4 mostra una trasmissione seriale relativa alla scrittura su EEPROM tramite I²C Bus. La traccia gialla rappresenta il pacchetto dati inviato dal PC al PIC18F4620 e quella azzurra è la risposta del PIC al PC. Prima di rispondere, il micro deve scrivere i dati in memoria: la traccia viola mostra la linea dati SDA, mentre quella in verde è il segnale di clock SCL; la scrittura è rappresentata dal secondo pacchetto viola/verde. Quindi deve rileggere quello che ha appena scritto (terzo pacchetto) e confrontarlo con il contenuto del buffer per la trasmissione RS232: se tutto va a buon fine, risponde al PC con “trasmissione OK”, altrimenti in caso di errore comunica “trasmissione KO”. La risposta da parte del PIC è evidenziata dal segnale in azzurro.

Principio di funzionamento

Vediamo ora la logica di funzionamento del nostro circuito, mettendo in evidenza le scelte progettuali effettuate per arrivare al codice definitivo. Per prima cosa abbiamo stabilito come indirizzare 512 messaggi avendo solo sedici ingressi diretti a disposizione. Il trucco consiste nell'utilizzo di altri cinque ingressi ausiliari i quali vengono utilizzati come moltiplicatore dei sedici ingressi diretti, i cinque ingressi del moltiplicatore permettono 32 combinazioni logiche per un totale di 512 messaggi (32 x 16 = 512). Pertanto quando sui cinque ingressi del moltiplicatore si legge il valore logico 0

è possibile indirizzare i messaggi da 1 a 16, se il valore logico è uguale a 1 è possibile indirizzare i messaggi da 17 a 32 e così via fino ad arrivare al valore logico 32 per i messaggi da 497 a 512. Inoltre è possibile stabilire se i sedici ingressi, in condizione di riposo, debbano essere normalmente aperti (NO) oppure normalmente chiusi (NC). Questa impostazione può essere effettuata sia da PC che da tastiera. I cinque ingressi del moltiplicatore, invece, possono solamente essere dei contatti normalmente aperti.

I 512 messaggi di allarme (escluso il messaggio di default) risiedono nella EEPROM esterna, collegata tramite bus I²C, la quale è stata mappata in modo da memorizzare sia i messaggi che e i parametri di ognuno di essi. Ogni messaggio occupa 160 caratteri (160 byte), quindi per memorizzare 512 messaggi è necessario disporre di una memoria di almeno 81920 byte. Ogni messaggio, inoltre, necessita di altri 12 byte per i parametri (altri 6144 byte), per cui sono necessari 88064 byte in totale. Si è reso pertanto necessario scegliere una EEPROM da 1 Mbit (128 kbyte). In figura 5 è riportata la mappatura della memoria EEPROM. Per ogni messaggio è stato allocato uno spazio pari a 256 byte. In questo modo viene utilizzata tutta la memoria EEPROM a disposizione.

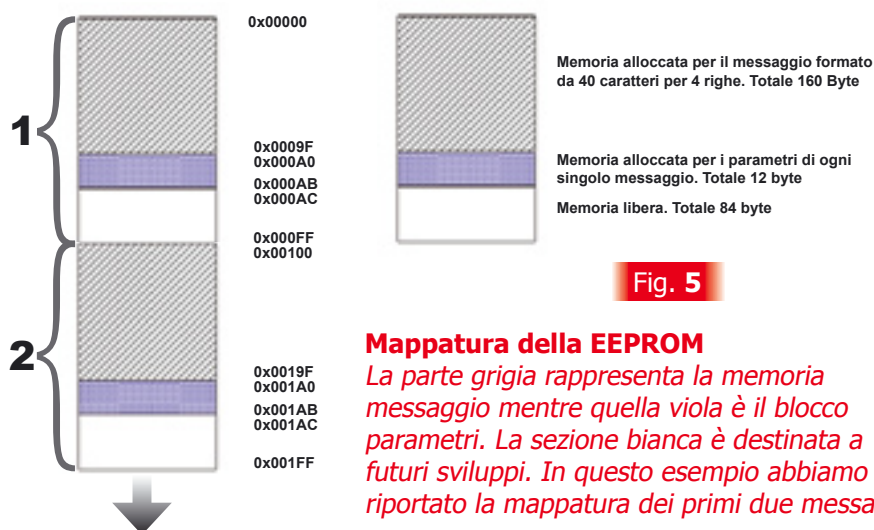
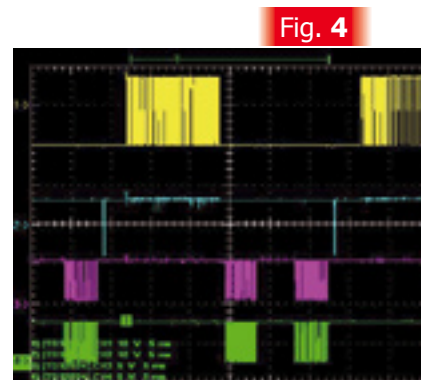


Fig. 5

Mappatura della EEPROM

La parte grigia rappresenta la memoria messaggio mentre quella viola è il blocco parametri. La sezione bianca è destinata a futuri sviluppi. In questo esempio abbiamo riportato la mappatura dei primi due messaggi.



Sequenza di scrittura in EEPROM.
Giallo: il PC invia i dati al PIC.
Viola/Verde: dopo aver ricevuto i dati, il PIC scrive e legge la EEPROM.
Azzurro: il PIC manda il messaggio di conferma o errore al PC.

Tra un messaggio e il successivo ci sono 84 byte liberi per eventuali sviluppi futuri. Per scrivere un byte in EEPROM alla locazione specifica è necessario un tempo di 5 msec; in teoria, dunque, occorrono circa 655 secondi per scrivere su tutti i 128 kbyte della EEPROM. Per ridurre drasticamente i tempi è possibile utilizzare la scrittura a pagine messa a disposizione dalla memoria stessa. Tale metodo di scrittura permette di scrivere 128 byte alla volta, sfruttando la capacità del buffer di ricezione interno alla EEPROM stessa. Quindi durante la trasmissione su bus I²C, oltre al comando e all'indirizzo della locazione da cui iniziare a scrivere, vengono inviati 128 byte di dati. Ricevuto il bit di stop, alla conclusione quindi della trasmissione dati, la ➤

EEPROM provvede a “travasare” tutto il contenuto del buffer nella EEPROM stessa impiegando “solo” 5 msec. Pertanto sfruttando la scrittura a pagine il tempo necessario a scrivere tutta la EEPROM diventa 5,12 secondi.

A questo è tuttavia necessario aggiungere i tempi di trasmissione sia sul bus I²C sia sull’interfaccia seriale, arrivando così ad un tempo complessivo di circa 30 secondi per tutta la memoria EEPROM.

Per indirizzare una memoria da 1 Mbit sono necessari 3 byte, ma il protocollo di trasmissione prevede solo 2 byte per l’indirizzo. Ne consegue che la memoria è suddivisa in due blocchi distinti da 65.536 byte ciascuno. Per indirizzare l’uno o l’altro è necessario impostare B0 nel byte di comando (figura 3): con B0=0 si indirizzano i primi 65536 byte con B0 = 1 i rimanenti byte. Il control code è fisso a “1010” 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.

Ogni allarme letto dagli ingressi diretti viene memorizzato in un buffer da 128 elementi chiamato “coda degli allarmi”. Esso serve per tenere traccia degli eventi esterni (codice allarme) e per evitare di memorizzare nuovamente un allarme che si ripresenta più volte. Se il numero degli allarmi intervenuti supera la lunghezza del buffer (ossia sono stati letti più di 128 allarmi) il sistema provvede in automatico a cancellare il primo elemento della coda (il più vecchio), ad eseguire uno shift verso il basso della stessa, e salvare l’ultimo allarme rilevato nell’ultima posizione, che si è liberata per via dello shift (posizione 128).

Grazie alla coda degli allarmi, è possibile visualizzare ciclicamente sul display LCD il messaggio di allarme relativo al codice memorizzato

nella coda. I parametri dei messaggi (memorizzati in EEPROM) indicano al PIC18F4620 in che modo deve eseguire il passaggio da “messaggio attuale” a “messaggio successivo”. Ad esempio, il passaggio dal messaggio “n” al messaggio “n+1” può avvenire semplicemente dopo che è trascorso un certo intervallo, oppure dopo che tutte le 4 righe del messaggio hanno subito uno shift verso destra o verso sinistra, ecc. Abbiamo previsto diversi casi, riportati tutti in tabella 1; poiché “shift” è un termine ricorrente, riportiamo in figura 6 un disegno che chiarisce cosa si intende per shift con e senza carry. Quando si esegue uno shift di un

messaggio senza carry i caratteri scompaiono alla destra o alla sinistra del display.

Eseguendo uno shift con carry, invece, i caratteri ricompaiono dalla parte opposta alla direzione dello shift; in altri termini se si sta eseguendo uno shift verso sinistra il carattere uscito dalla parte visibile del display ricompare a destra.

Uno dei parametri è il tempo di blink (espresso in msec) ed avente dimensione di due byte. Il tempo di blink serve per fare lampeggiare il messaggio di allarme durante tutta la sua permanenza sul display LCD. Questo parametro può essere modificato solo tramite il programma su

Tabella 1 - Opzioni di visualizzazione

Testo fisso, nessun blink. Transizione dal messaggio “n” al messaggio “n+1” dopo un tempo t di ciclo definito dall’utente e modificabile da pannello (Tempo minimo 1 secondo, tempo massimo 10 secondi).

Testo fisso con blink. Il messaggio blinka durante la sua visualizzazione. Transizione dal messaggio “n” al messaggio “n+1” dopo un tempo t di ciclo definito dall’utente e modificabile da pannello (Tempo minimo 1 secondo, tempo massimo 10 secondi).

La riga “m” del messaggio viene shiftata a destra durante la sua visualizzazione. Solo dopo il completo shift della riga viene eseguita la transizione al nuovo messaggio. Naturalmente anche le rimanenti righe stanno shiftando verso destra o sinistra a seconda dell’impostazione.

La riga “m” del messaggio viene shiftata a destra con carry durante la sua visualizzazione. Solo dopo il completo shift della riga viene eseguita la transizione al nuovo messaggio. Naturalmente anche le rimanenti righe stanno shiftando verso destra o sinistra a seconda dell’impostazione.

La riga “m” del messaggio viene shiftata a sinistra durante la sua visualizzazione. Solo dopo il completo shift della riga viene eseguita la transizione al nuovo messaggio. Naturalmente anche le rimanenti righe stanno shiftando verso destra o sinistra a seconda dell’impostazione.

La riga “m” del messaggio viene shiftata a sinistra con carry durante la sua visualizzazione. Solo dopo il completo shift della riga viene eseguita la transizione al nuovo messaggio. Naturalmente anche le rimanenti righe stanno shiftando verso destra o sinistra a seconda dell’impostazione.

L’intero display viene shiftato a destra. Solo dopo il completo shift del display viene eseguita la transizione al nuovo messaggio.

L’intero display viene shiftato a destra con carry. Solo dopo il completo shift del display viene eseguita la transizione al nuovo messaggio.

L’intero display viene shiftato a sinistra. Solo dopo il completo shift del display viene eseguita la transizione al nuovo messaggio.

L’intero display viene shiftato a sinistra con carry. Solo dopo il completo shift del display viene eseguita la transizione al nuovo messaggio.

PC. Esistono altri quattro parametri, non modificabili, riguardanti il tipo di allineamento dei messaggi (a destra, a sinistra o centrato). Con il software su PC è possibile scegliere tra allineamento a destra, a sinistra o al centro. La figura 8 mostra una schermata in cui le prime due righe sono allineate a sinistra, la terza al centro e l'ultima a destra. Nella tabella 3 viene mostrato l'elenco completo dei parametri descritti in precedenza. Essi sono undici ed occupano 12 byte in totale. Tutte le informazioni descritte fino ad ora risiedono nella EEPROM esterna, ma altri dati vengono memorizzati nella EEPROM del PIC18F4620.

In particolare:

- Mascheratura per determinare se i sedici ingressi diretti sono impostati con i contatti normalmente aperti o normalmente chiusi (2 byte). Ogni bit dei 16 disponibili identifica un contatto; se il bit è a "1" significa che il contatto è normalmente aperto altrimenti, se



- "0", è normalmente chiuso.
- Valore contrasto LCD (1 byte).
- Tempo di ciclo per visualizzazione messaggi di allarme presenti nella coda (2 byte).
- Allineamento messaggio 1 (1 byte).
- Allineamento messaggio 2 (1 byte).
- Allineamento messaggio 3 (1 byte).
- Allineamento messaggio 4 (1 byte).
- Messaggio di default (160 byte).

Tutti questi parametri di configurazione vengono inviati al modulo al-

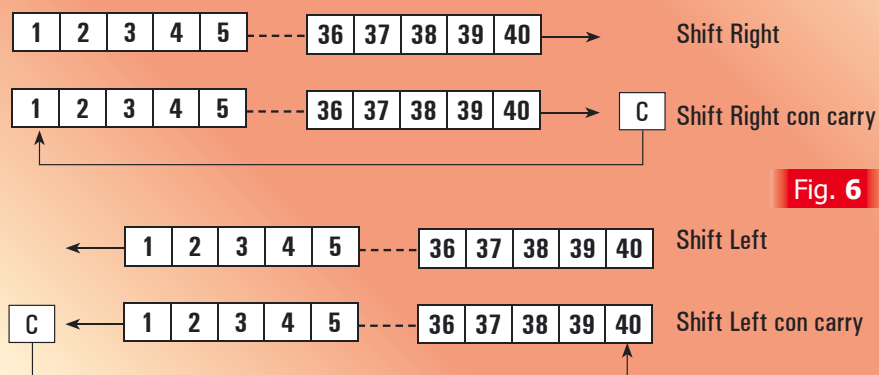


Fig. 6

Rappresentazione schematica di "shift con carry", sia a destra che a sinistra.

larmi attraverso l'interfaccia seriale ed il programma per PC. Per realizzare una comunicazione efficiente tra il PC e il PIC18F4620, è stato

checksum.

- Add_3, Add_2, Add_1, Add_0: indirizzo della locazione di memoria su cui si vuole scrivere. Add_3 è

Fig. 7

Struttura del protocollo di comunicazione seriale.



sviluppato il protocollo di comunicazione riportato in figura 7 e che contiene i seguenti parametri:

- CMD: comando da inviare al PIC.

il byte più significativo e Add_0 il meno significativo.

- Data n: primo byte dati. Sono previsti pacchetti dati da 32, 64, 128 byte. Dipende dall'applicazione.
- Data n+1: ultimo byte dati.
- CHK: checksum della stringa. Consente sia al PIC che al PC di verificare se la stringa di dati è valida. Il valore del checksum si ottiene eseguendo un AND logico della somma di tutti i dati trasmessi con 255, quindi negando ogni bit del risultato, ossia: $CHK = NOT [(Somma\ dati\ trasmessi) AND\ 255]$. Durante il calcolo del checksum non bisogna tenere conto del checksum stesso. ➤

Tabella 2 - Elenco comandi di accesso all'EEPROM

Codice Comando	Nome Comando	Descrizione
0x01	WRITE_EEPROM_PIC	Scriva nella EEPROM del PIC18F4620
0x07	ERASE_EEPROM_PIC	Cancella la EEPROM del PIC18F4620
0x81	READ_EEPROM_PIC	Legge la EEPROM del PIC18F4620
0x87	READ_PIC_FW_REV	Legge la versione FW presente sul PIC18F4620
0xFE	RESET_PIC	Resetta PIC18F4620
0x04	WRITE_EEPROM	Scriva sulla EEPROM esterna
0x0A	ERASE_EEPROM	Cancella la EEPROM esterna
0x84	READ_EEPROM	Legge la EEPROM esterna

L“AND” della somma serve per ottenere un risultato a 8 bit.

I comandi utilizzati per la trasmissione sono riassunti in tabella 2.

Durante la trasmissione seriale, il PC si comporta come MASTER, cioè definisce le operazioni di lettura e scrittura inviando appositi comandi al PIC18F4620, che è sempre SLAVE. Se durante la comunicazione il PIC non risponde, il PC ritenta la trasmissione per tre volte dopodiché considera conclusa la trasmissione seriale, avvisando l'utente con un messaggio di errore. Tenete presente che una trasmissione seriale potrebbe fallire anche per altre ragioni, ad esempio per il checksum errato oppure perché il PIC18F4620 non riesce a scrivere sulla EEPROM in modo corretto, eccetera. In tutte queste situazioni (sia chiaro, anomale) l'utente viene avvisato con uno specifico messaggio di errore.

Realizzazione pratica

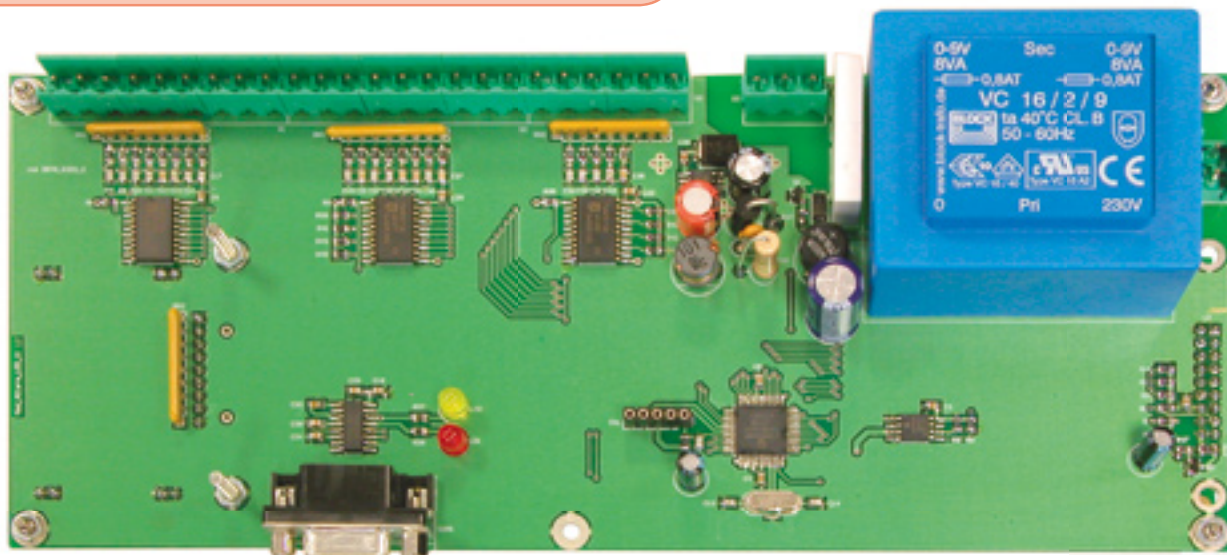
La scheda allarmi è stata realizzata in tecnologia SMD quindi munitevi del circuito stampato, di un saldatore a punta fine, di fluxante, di stagno di buona qualità ed infine di tanta pazienza. Consigliamo di iniziare a saldare tutti i componenti SMD iniziando dai passivi a basso profilo: saldate pertanto tutte le resistenze in formato 0805 e 1206. Successivamente potete montare i condensatori, in case 0805. I condensatori elettrolitici devono essere montati in un secondo momento. Conclusa la saldatura delle resistenze e dei condensatori è possibile iniziare a saldare i vari integrati presenti sulla scheda, partendo con i 3 latch e proseguendo con il MAX202, la memoria EEPROM, il regolatore di tensione switching ed infine il PIC. Saldare quindi il quarzo da 10 MHz, collegando a massa il suo contenitore (sul PCB è stata predisposta una piazzola per eseguire questa saldatura). Ora si può passare ai componenti tradizionali i quali comprendono

Tabella 3 - Parametri di visualizzazione dei messaggi

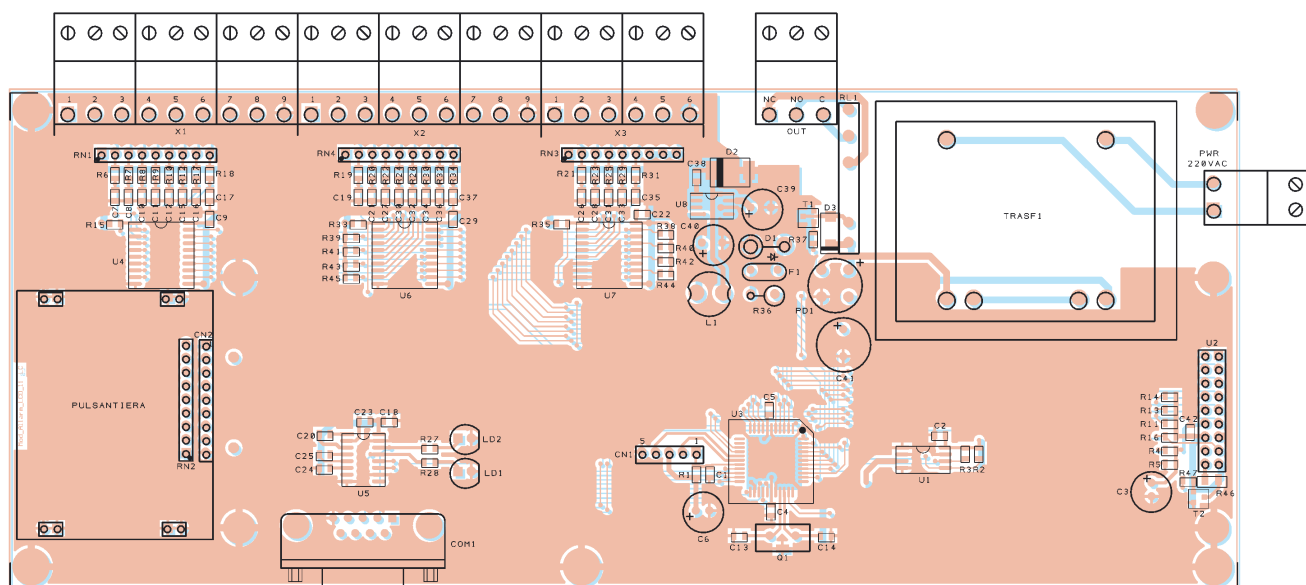
P	N° Byte	Valore	Descrizione
1	1	0	Allineamento testo riga 1 “Sinistra” .
		1	Allineamento testo riga 1 “Destra” .
		2	Allineamento testo riga 1 “Centrato” .
2	1	0	Allineamento testo riga 1 “Sinistra” .
		1	Allineamento testo riga 2 “Destra” .
		2	Allineamento testo riga 3 “Centrato” .
3	1	0	Allineamento testo riga 3 “Sinistra” .
		1	Allineamento testo riga 3 “Destra” .
		2	Allineamento testo riga 3 “Centrato” .
4	1	0	Allineamento testo riga 4 “Sinistra” .
		1	Allineamento testo riga 4 “Destra” .
		2	Allineamento testo riga 4 “Centrato” .
5	1	Max 1000	Tempo di Blink messaggio. Tempo in msec
6	1	0	Testo Blink (Nessuno Shift di riga o display)
		1	Testo NO Blink (Nessuno shift ri riga o display)
		2	Shift di riga
		3	Shift intero display
7	1	0	Nessun codice associato (Già impostato il comando di Blink)
		1	Shift SX riga 1 NO Carry
		2	Shift SX riga 1 NO Carry
		3	Shift DX riga 1 NO Carry
		4	Shift DX riga 1 con Carry
8	1	0	Nessun codice associato (Già impostato il comando di Blink)
		1	Shift SX riga 2 NO Carry
		2	Shift SX riga 2 NO Carry
		3	Shift DX riga 2 NO Carry
		4	Shift DX riga 2 con Carry
9	1	0	Nessun codice associato (Già impostato il comando di Blink)
		1	Shift SX riga 3 NO Carry
		2	Shift SX riga 3 NO Carry
		3	Shift DX riga 3 NO Carry
		4	Shift DX riga 3 con Carry
10	1	0	Nessun codice associato (Già impostato il comando di Blink)
		1	Shift SX riga 4 NO Carry
		2	Shift SX riga 4 NO Carry
		3	Shift DX riga 4 NO Carry
		4	Shift DX riga 4 con Carry
11	1	0	Nessun codice associato (Già impostato il comando di Blink)
		1	Shift SX riga 5 NO Carry
		2	Shift SX riga 5 NO Carry
		3	Shift DX riga 5 NO Carry
		4	Shift DX riga 5 con Carry

le morsettiere, il relè, i condensatori elettrolitici, il connettore RS232 a 9 poli, i vari connettori a passo 2,54 mm per la connessione dell’LCD e dello stampato su cui vanno montati i pulsanti. Per ultimo deve essere

saldato il trasformatore, che tra tutti i componenti è il più ingombrante. Il display LCD andrà collegato alla scheda solo dopo aver verificato che tutte le alimentazioni siano corrette. Montati tutti i componenti è neces- ➤



Piano di montaggio e prototipo della piastra base a montaggio ultimato. Dal sito della rivista (www.elettronica.in.it) è possibile scaricare gratuitamente i master dei CS in dimensioni reali.

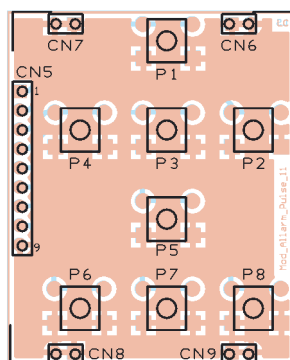
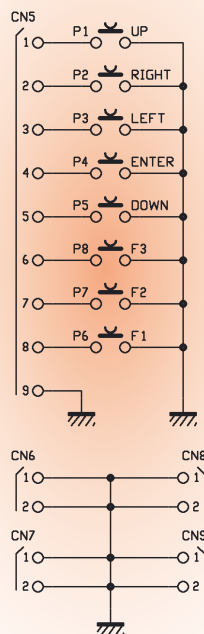


ELENCO COMPONENTI:

R1: 22 kohm (SMD)	RN1÷RN4: rete resistiva 8x4,7 kohm + comune	C29: 100 nF multistrato (SMD)	U8: LM2674M_5.0
R2, R3 4,7 kohm (SMD)	C1: 220 pF ceramico (SMD)	C30÷C37: 220 pF ceramico (SMD)	Q1: Quarzo 10 MHz
R4: 680 ohm (SMD)	C2, C4, C5: 100 nF multistrato (SMD)	C38: 10 nF multistrato (SMD)	T1, T2: BC846B
R5: 220 ohm (SMD)	C3, C6: 10 µF 25 VL elettrolitico	C39, C40: 100 µF 50 VL elettrolitico	RL1: Relè Finder 6A
R6÷R10: 100 kohm (SMD)	C7, C8: 220 pF ceramico (SMD)	C41: 470 µF 50 VL elettrolitico	COM1: Connettore DB9 femmina
R11: 4,7 kohm (SMD)	C9: 100 nF multistrato (SMD)	C42: 100 nF multistrato (SMD)	L1: Induttanza 100 µH
R12: 100 kohm (SMD)	C10÷C12: 220 pF ceramico (SMD)	D1: 1.5KE33CA	F1: Termofusibile 100 mA
R13÷R16: 4,7 kohm (SMD)	C13, C14: 22 pF ceramico (SMD)	D2: SS34	TRASF1: Trasformatore 16VA/230V/2x9Vac
R17÷R26, R34: 100 kohm (SMD)	C15÷C17: 220 pF ceramico (SMD)	D3: GF1M (DO214)	
R27, R28: 1,2 kohm (SMD)	C18: 100 nF multistrato (SMD)	PD1: W08M	
R29÷R32: 100 kohm (SMD)	C19: 220 pF ceramico (SMD)	LD1: led 5 mm rosso	
R33, R35: 4,7 kohm (SMD)	C20, C22÷C25: 100 nF multistrato (SMD)	LD2: led 5 mm giallo	
R36: 10 ohm 2W	C21, C26÷C28: 220 pF ceramico (SMD)	U1: 24FC1025-I/SM	
R37: 1,2 kohm (SMD)		U2: LCD DATA IMAGE CM400400	
R38÷R45: 4,7 kohm (SMD)		U3: PIC18F4620-I/PT (MF690)	
R46: 10 ohm (1206)		U4, U6, U7: 74HC573D	
R47: 1,2 kohm (SMD)		U5: MAX202ECUE	

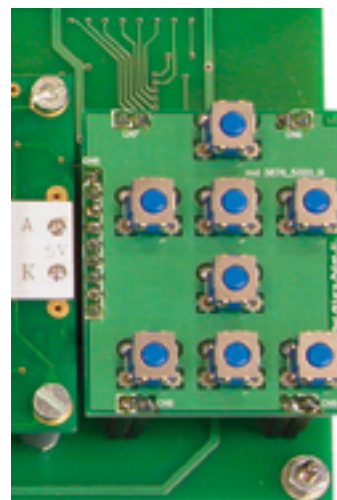
Varie:
 - Morsetteria 2 poli componibile
 - Morsetteria 3 poli componibile (6 pz.)
 - Strip maschio 5 poli
 - Circuito stampato

SCHEMA ELETTRICO



ELENCO COMPONENTI:

- P1: Microswitch
- P2: Microswitch
- P3: Microswitch
- P4: Microswitch
- P5: Microswitch
- P6: Microswitch
- P7: Microswitch
- P8: Microswitch



Varie:

- CN5: pin strip 9 poli
- CN6: pin strip 2 poli
- CN7: pin strip 2 poli
- CN8: pin strip 2 poli
- CN9: pin strip 2 poli

La tastiera è molto semplice essendo composta solamente da otto micropulsanti da CS e cinque connettori a passo 2,54 mm.

sario eseguire un'ispezione visiva della scheda alla ricerca di eventuali saldature mal fatte o eventuali cortocircuiti, soprattutto tra i pin dei componenti SMD, come ad esempio il PIC18F4620.

Eseguite tutte le verifiche del caso, è finalmente possibile dare alimentazione alla scheda collegandola alla rete elettrica dei 230 Vac. Controllate con un tester le alimentazioni su ogni singolo integrato e assicuratevi che siano presenti i +5Vdc. Concluso il test sulle alimentazioni

è possibile collegare il display LCD, naturalmente ad alimentazione scollegata. Non rimane che programmare il PIC18F4620 con l'apposito firmware, utilizzando un programmatore come l'ICD2 o l'FT386. Dovete anche realizzare la tastiera ma essa non rappresenta un ostacolo: su di essa, infatti, vanno montati solamente gli otto pulsanti da stampato (ad inserzione) ed i connettori a passo 2,54 mm con i quali la tastiera viene collegata (sia meccanicamente che elettricamente) alla scheda madre.

A questo punto il modulo allarmi è pronto per essere configurato (tramite PC o tastiera) e per visualizzare i vari messaggi. Nella prossima puntata descriveremo il firmware e le funzioni messe a disposizione dal nostro visualizzatore di messaggi: analizzeremo alcune porzioni di codice sorgente, e vedremo come configurare il sistema sia tramite tastiera che tramite il programma per ambiente Windows.

Appuntamento quindi al prossimo numero della rivista.

per il

MATERIALE

Il materiale utilizzato in questo progetto è facilmente reperibile in commercio. Il master del CS, il firmware con cui programmare il PIC ed il software di configurazione sono scaricabili gratuitamente dal sito della rivista (www.elettronica.in.it). Il microcontrollore in SMD utilizzato in questo circuito (PIC18F4620-I/PT) è disponibile al costo di 14,00 Euro, IVA compresa.

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

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

Porta USB Host per sistemi Embedded

di Matteo Bosio

Conosciamo e impariamo a utilizzare i nuovi dispositivi della famiglia Vinculum di FTDI, sistemi che consentono di aggiungere facilmente una porta USB di tipo Host ad apparecchiature differenti dal PC e sprovviste di tale interfaccia.



La porta di comunicazione USB, grazie alle sue doti di velocità e versatilità, ha sostituito ormai da parecchi anni le più datate seriali e parallele, specialmente in ambito Personal Computer dove sempre più spesso sono assenti entrambe. Rispetto allo sviluppo tecnologico, il mondo PC ha assorbito la maggior parte delle risorse per la ricerca, al punto che le periferiche (stampanti, scanner, modem ADSL...) dotate di porta USB sono sempre più numerose; è quindi comprensibile come il PC e tutto ciò che orbita attorno ad esso rappresentino una elevata fonte di guadagno. L'integrazione di tale interfaccia su sistemi

“non PC”, invece, non è stata considerata con la stessa attenzione dedicata al PC, forse perché non era altrettanto interessante economicamente. Ciò ha generato per lungo tempo difficoltà in fase di progettazione in quanto integrare il protocollo USB su un micro con risorse hardware limitate richiede una grossa mole di sviluppo firmware. Alcune aziende hanno quindi deciso di approfittare di questa situazione, realizzando dei circuiti integrati in grado di svolgere facilmente la conversione da seriale (presente anche nei microcontrollori più piccoli) a porta USB e fornendo i Driver per i sistemi operativi più diffusi. Una ➤



di queste aziende, FTDI (Future Technology Devices International LTD, www.ftdichip.com), ha realizzato alcuni chip con cui aggiungere facilmente la porta USB Device (Slave) a qualsiasi micro dotato di una seriale. Ora l'azienda propone una nuova soluzione per aggiungere altrettanto facilmente ad una seriale una porta USB Host: stiamo parlando della famiglia di prodotti Vinculum.

Classificazione porte USB

Prima di vedere nel dettaglio questi nuovi dispositivi, è necessario chiarire alcuni punti che riguardano i due tipi di dispositivi USB: quelli di tipo Device (o Slave) e quelli di tipo Host (impropriamente Master).

USB di tipo Device (Slave)

Con questa definizione si intende un'applicazione (leggasi anche "prodotto") che, se connessa ad un PC, viene riconosciuta come periferica USB e, una volta installati nel computer i relativi driver, è in grado di funzionare correttamente: stampante, macchina fotografica, mouse, tastie-

ra, altoparlanti, scanner, ecc. sono tutte periferiche Device. Tipicamente, ognuna di esse monta a bordo un connettore USB da circuito stampato di tipo B (più piccolo), in modo che possa essere collegata al PC per mezzo di un semplice cavo USB; in altri casi (esempio, memorie di archiviazione di massa USB, le cosiddette USB Flash Drive o Pen Drive), possono essere collegati direttamente nella presa USB del PC.

USB di tipo Host (Master)

Si intende un sistema in grado di riconoscere periferiche USB di tipo Slave e di gestire la comunicazione con esse per mezzo di Driver specifici. La porta Host per applicazioni da computer è costituita da un chip che svolge la funzione di adattatore di livello e da uno "stack" software, spesso di dimensioni ragguardevoli. Lo stack è installato nel PC che, normalmente, non è limitato nella potenza di calcolo. La porta Host è inoltre in grado di alimentare un dispositivo Device, se quest'ultimo non assorba più di 500 mA.

Soluzioni USB disponibili

Come già anticipato, alcune aziende hanno realizzato soluzioni per aggiungere la porta USB Device a processori di piccolo taglio.

Microchip è stata tra le prime ad affrontare il discorso sviluppando dei microcontrollori con l'USB incorporata (esempio, PIC18F2550 o PIC18F4550): dotati di due uscite specifiche, una serie di buffer interni e di registri dedicati, essi tuttavia richiedono una conoscenza approfondita sia del protocollo USB che dei metodi con cui vanno gestiti i processi di enumerazione, di configurazione, di riconoscimento. Su questo argomento ricordiamo il Corso proposto sui fascicoli dal 92 al 102.

FTDI, invece, ha sviluppato dei chip specializzati (e relativo driver) da associare al micro per realizzare applicazioni di tipo USB Slave: tra i tanti citiamo i recenti FT232R e FT245R, in grado di convertire rispettivamente da seriale TTL a USB e da parallela FIFO a USB. Questa soluzione è molto versatile dato che la maggior parte del lavoro è svolta dalla coppia chip/driver: al micro non rimane che dialogare con una porta seriale standard (FT232R) oppure con una porta parallela a 8 bit (FT245R) qualora non dovesse disporre della seriale.

Ora, però, FTDI propone una soluzione innovativa con la quale ha risolto brillantemente il problema di come dotare apparati non PC della porta USB Host. Il prodotto di cui parliamo, il VNC1L-1A, è un chip dotato di due porte USB Host (all'occorrenza possono essere configurate anche come Slave) con le quali interfacciarsi ad alcune delle periferiche USB più utilizzate tipo le Pen Drive, delle quali gestisce la struttura dei file FAT in modo totalmente trasparente. Il chip, primo della famiglia "Vinculum", comunica con il micro di sistema attraverso UART, SPI o FIFO parallela, senza richiedere grandi nozioni software, poiché le suddette interfacce sono disponibili anche sui

microcontrollori più semplici.

A complemento del chip, FTDI ha poi sviluppato una serie di moduli "finiti" programmati con firmware diversi a seconda dei casi, tutti basati sul VNC1L. Nei paragrafi successivi entreremo nel dettaglio sia del chip VNC1L sia di detti prodotti. Relativamente ai moduli, parleremo di:

- **VDIP1**, che presenta su zoccolo DIP 24 pin tutti i segnali del VNC1L;
- **DRIVE2**, pensato per dare a sistemi finiti la possibilità di leggere e scrivere su Pen Drive;
- **VF2F2**, un prodotto stand-alone per far comunicare direttamente due memorie USB senza passare attraverso il PC;
- **VMUSIC2**, per riprodurre file audio da memoria USB.

Iniziamo con il chip.

Vinculum VNC1L

Il chip VNC1L è il primo dispositivo FTDI della famiglia di circuiti integrati funzionanti come controller USB di tipo Host/Slave. Esso integra al suo interno un microprocessore con core da 8/32 bit, che gestisce l'intero protocollo USB, e una memoria di tipo Flash. Il firmware implementato permette di gestire i file della FAT (File Allocation Table, la descrizione di come i file sono memorizzati e suddivisi) in modo completamente trasparente quando viene interfaccia-

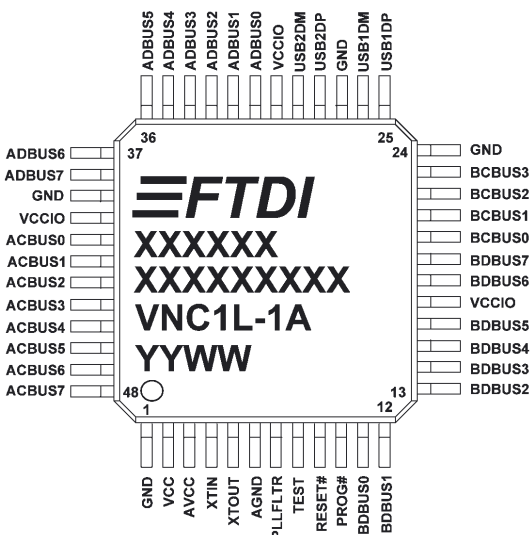
VNC1L-1A

- Doppio controller DMA per l'accelerazione hardware;
- Moltiplicatore di clock da 12 MHz a 48 MHz integrato;
- Alimentazione a 3,3 V con ingressi tolleranti i 5 V;
- Basso consumo (25 mA operativi, 2 mA in stand-by);
- Circuito di reset integrato con ingresso reset opzionale;
- Memoria Flash di programma da 64 kbyte;
- Memoria dati SRAM interna da 4 kbyte;
- Due porte USB 2.0 configurabili separatamente tramite firmware come Host o Slave e in modalità Low Speed (1,5 Mbps) o Full Speed (12 Mbps);
- Completamente compatibile con le specifiche USB 2.0;
- Interfaccia UART per I/O dati, programmazione del firmware e interfaccia di comando;
- Interfaccia FIFO parallela con bus a 8 linee bi-direzionale e semplice handshake a 4 linee per I/O dati e interfaccia di comando;
- Interfaccia SPI per I/O dati e interfaccia di comando;
- Porta PS2 per tastiera o mouse;
- Fino a 28 pin GPIO (General Purpose Input Output);
- Disponibile in formato 48-Lead LQFP conforme alle direttive RoHS

Specifiche tecniche



Fig. 1



to ad un dispositivo USB tipo Pen Drive. Il controllo avviene tramite un semplice set di istruzioni, utilizzando come canale di comunicazione l'interfaccia UART, la SPI o la FIFO parallela. Il chip VNC1L dispone di due porte USB che possono essere configurate separatamente come Host o Slave durante la programmazione; deve inoltre

essere programmato con l'apposito firmware, scaricabile gratuitamente dal sito della FTDI (www.ftdichip.com), per l'uso del quale non sono necessarie licenze particolari.

Questo dispositivo può essere utilizzato in una vasta gamma di applicazioni:

- collegamento di Flash Drive a microcontrollori o FPGA.
- Data Logging, per esempio in applicazioni in cui si deve acquisire una serie di misure analogiche tramite AD e salvarle su una qualsiasi USB Flash Drive, in alternativa a memoria di tipo SD Card o MMC.
- Interfaccia di microcontrollori verso dispositivi USB di tipo slave.
- Interfaccia diretta tra USB Flash Drive e dispositivi quali cellulari, PDA, MP3 Player o fotocamere digitali.
- Interfaccia diretta tra ricevitori GPS USB e cellulari.

Il VNC1L può comunicare con il microcontrollore attraverso tre interfacce distinte: seriale, parallela ed SPI.

In Tabella 1 abbiamo riportato l'elen- ➤

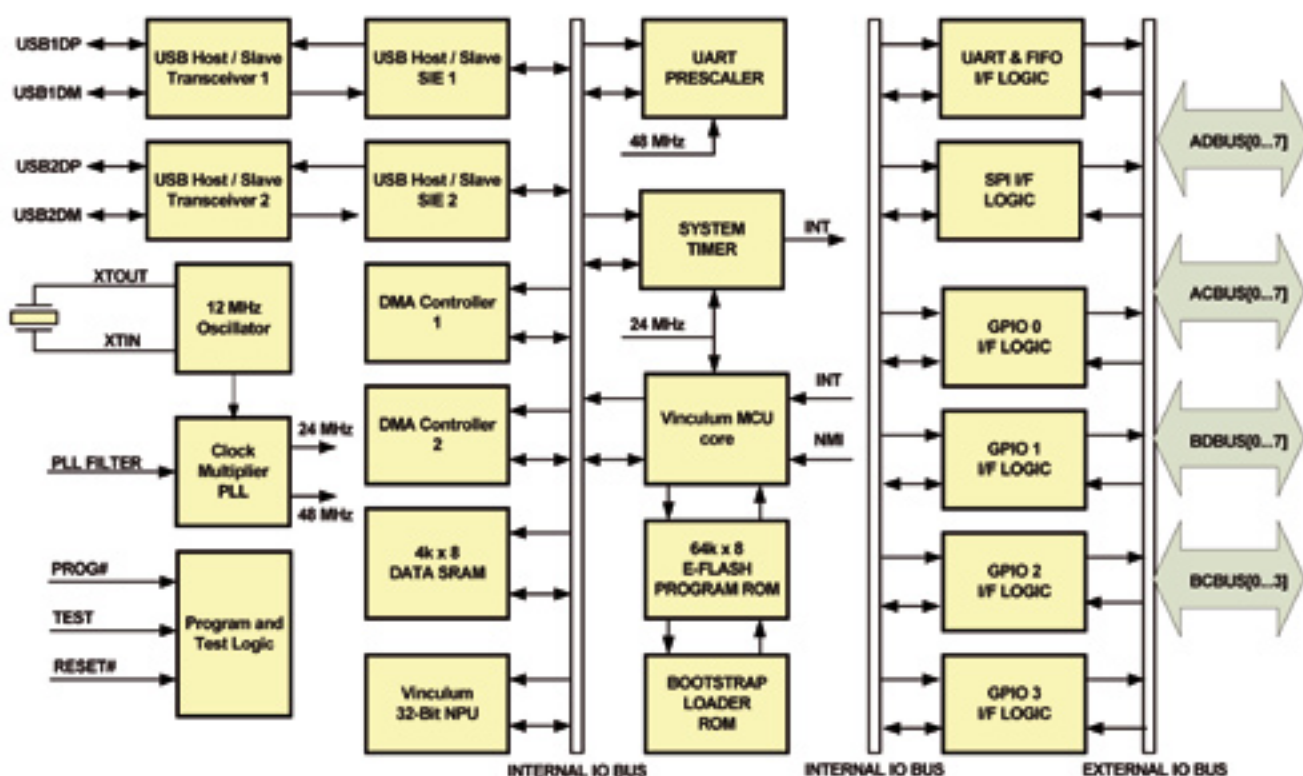
Schema a blocchi

Vediamo nel dettaglio i blocchi funzionali in cui è possibile schematizzare la struttura del VNC1L:

- **USB Host/Slave Transceivers 1 e 2:** forniscono l'interfaccia fisica USB 2.0.
- **USB Host/Slave Serial Interface Engine 1 e 2 (SIE):** gestiscono la conversione da seriale a parallela del livello fisico USB e viceversa, compresi il controllo degli errori, la generazione di frame USB, il CRC e il bit stuffing.
- **12 MHz Oscillator:** genera un clock di riferimento a 12 MHz partendo da un quarzo esterno.
- **Clock Multiplier PLL:** partendo dal clock a 12 MHz, genera i clock di riferimento a 24 MHz e 48 MHz, che vengono usati dal SIE, dal core MCU, dal timer di sistema e dal prescaler UART.
- **Program and Test Logic:** permette di programmare la Flash

parte delle operazioni avviene a 8 bit.

- **UART Prescaler:** fornisce al blocco **UART and FIFO Logic** il clock per la trasmissione e la ricezione dei dati. Può operare da 300 baud a 1 Mbaud.
- **System Timer:** crea un interrupt ogni 1 msec, usato dalla MCU per gestire i vari timeout e le temporizzazioni.
- **Vinculum MCU Core:** è il cuore del VNC1L. Si tratta di un microprocessore a 8 bit proprietario della FTDI, con una architettura Harvard (memoria dati di 64 kbyte divisa in pagine, memoria di codice di 64 kbyte e 256 byte di spazio IO). Utilizza la tecnologia "enhanced CISC", insieme ad un set di istruzioni VNCU (proprietarie) che rimpiazzano diverse linee di codice convenzionale garantendo prestazioni di tipo RISC, con una ottima compressione del codice in ROM.



interna. Quando l'ingresso PROG# viene tenuto a livello logico basso il dispositivo va in reset, la Flash interna viene aggirata dal **BootStrap Loader Rom**, blocco che contiene il codice necessario per programmare la Flash attraverso l'interfaccia UART. FTDI fornisce un software con cui programmare il VNC1L.

- **DMA Controller 1 e 2:** il doppio controller DMA garantisce elevate prestazioni consentendo ai dati provenienti dai SIE e dalle interfacce UART, FIFO e SPI di essere trasferiti alla SRAM minimizzando gli interventi della MCU.
- **Data SRAM:** questo blocco di 4 kbyte contiene le variabili usate dalla MCU, ma si può anche accedere ad esso in modo trasparente alla MCU attraverso il doppio controller DMA.
- **NPU (Numeric CoProcessor):** interviene nei casi in cui vengono effettuate operazioni aritmetiche a 32 bit, anche se la maggior

- **E-Flash Program:** il VNC1L integra una memoria Flash da 64 kbyte, programmabile via UART tenendo basso il PROG# e resettando il dispositivo o utilizzando la programmazione via USB Flash Drive, inclusa in alcuni firmware caricabili sul VNC1L.
- **BootStrap Loader ROM:** questo blocco di 512 Byte viene utilizzato quando si aggira la memoria Flash (PROG# tenuto basso). Permette di programmare la Flash attraverso l'interfaccia UART.
- **UART and FIFO Logic:** forniscono le interfacce opzionali seriale e parallela al VNC1L, equivalenti a quelle presenti sui chip FT232 e FT245.
- **SPI I/F Logic:** realizza la logica di controllo per la porta seriale SPI.
- **GPIO Blocks (0÷3):** ingressi e uscite generiche a livello TTL.

co dei segnali da utilizzare quando il chip è configurato per comunicare via UART. Potete notare che la seriale dispone di tutti i segnali tipici della RS232, nonché del comando di abilitazione alla trasmissione, tipico della RS485.

In Tabella 2, invece, è possibile vedere quali pin sono coinvolti nella comunicazione tramite porta parallela: notate che la maggior parte di essi coincide con quelli usati anche nel collegamento seriale. Le figure 3 e 4 mostrano il ciclo di lettura e quello di scrittura dell'interfaccia FIFO: notate che il dato deve essere stabile quando viene effettuata la lettura o la scrittura. Le tempistiche relative alle due figure sono riportate nelle tabelle dei diagrammi temporali. In Tabella 3, infine, sono riportati i segnali utilizzabili per comunicare con l'interfaccia SPI del chip. Nel ciclo di lettura, visibile in Figura 5, il segnale SPI-CS deve essere tenuto alto per tutto il periodo di lettura quindi deve essere tenuto basso per almeno un colpo di clock dopo la fine della lettura. Il primo bit su SDI (SPI Data In) è il R/W bit ("1" logico abilita la lettura). Il successivo è il bit di indirizzo, ADD, usato per indicare se

Tabella 1

Pin N°	Nome	Tipo	Descrizione
31	TXD	Output	Trasmissione asincrona dei dati
32	RXD	Input	Ricezione asincrona dei dati
33	RTS#	Output	Request To Send
34	CTS#	Input	Clear To Send
35	DTR#	Output	Data Terminal Ready
36	DSR#	Input	Data Set Ready
37	DCD#	Input	Data Carrier Detect
38	RI#	Input	Ring Indicator - Quando si usa l'opzione Wake-Up da remoto, tenendo questo ingresso a livello logico basso è possibile ripristinare l'Host USB dallo stato di sospensione
41	TXDEN	Output	Abilitazione alla trasmissione dati per RS485

Tabella 2

Pin N°	Nome	Tipo	Descrizione
31	D0	I/O	FIFO Bus Data Bit 0
32	D1	I/O	FIFO Bus Data Bit 1
33	D2	I/O	FIFO Bus Data Bit 2
34	D3	I/O	FIFO Bus Data Bit 3
35	D4	I/O	FIFO Bus Data Bit 4
36	D5	I/O	FIFO Bus Data Bit 5
37	D6	I/O	FIFO Bus Data Bit 6
38	D7	I/O	FIFO Bus Data Bit 7
41	RXF#	Output	Se alto, non legge dati dalla FIFO; se basso indica che ci sono dati disponibili nella FIFO, che possono essere letti abbassando e poi alzando RD#
42	TXE#	Output	Se alto, non scrive dati nella FIFO; se basso, i dati possono essere scritti nella FIFO alzando WR e poi abbassandolo
43	WR	Input	Segnale di write per il ciclo di scrittura (Figura 3)
44	RD#	Input	Segnale di read per il ciclo di lettura (Figura 4)

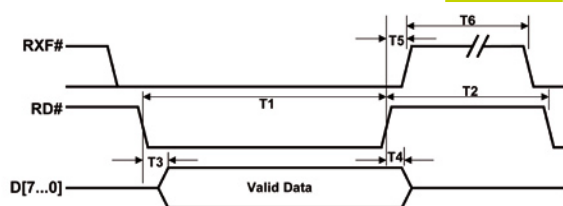


Fig. 3

Time	Description	Min	Max	Unit
T1	RD Active Pulse Width	50	-	ns
T2	RD to RD Pre-Charge Time	50 + T6	-	ns
T3	RD Active to Valid Data*	20	50	ns
T4	Valid Data Hold Time from RD Inactive*	0	-	ns
T5	RD Inactive to RXF#	0	25	ns
T6	RXF# Inactive After RD Cycle	80	-	ns

viene letto il data register ('0') o lo status register ('1'). Seguono poi gli 8 bit del dato, partendo dal bit più significativo (MSB).

Nel ciclo di scrittura, come si può osservare in Figura 6, il CS deve essere tenuto alto per tutto il periodo della scrittura quindi deve essere tenuto

basso per almeno un colpo di clock dopo la fine della scrittura. Il primo bit su SDI è il R/W bit ("0" logico abilita la scrittura). Il bit successivo è >

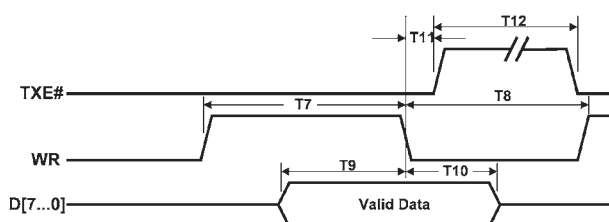


Fig. 4

Time	Description	Min	Max	Unit
T7	WR Active Pulse Width	50	-	ns
T8	WR to RD Pre-Charge Time	50	-	ns
T9	Data Setup Time before WR Inactive	20	-	ns
T10	Data Hold Time from WR Inactive	0	-	ns
T11	WR Inactive to TXE#	5	25	ns
T12	TXE# Inactive After WR Cycle	80	-	ns

ha realizzato dei moduli finiti che utilizzano il VNC1L come cuore del sistema e differiscono per forma e firmware integrato. Vediamoli uno ad uno.

Vinculum VDIPI

Il VDIPI1, visibile in figura 8, è un dispositivo completo realizzato su un piccolo circuito stampato: è dotato di un connettore USB tipo A per connessioni Host e di due pin strip

sere usato anche come modulo finito in progetti di basso e medio volume. Sul lato componenti sono presenti due jumper a tre vie (J3 e J4) che permettono di selezionare la modalità di comunicazione tra il modulo VDIPI1 e il microprocessore. In Tabella 4 sono riportate le combinazioni possibili.

Il firmware presente all'interno del VDIPI1 riconfigura i vari pin in base alla modalità di comunicazione se-

(5V e GND).

In questo dispositivo il chip VCN1L è programmato con il firmware specifico, chiamato VDAP e scaricabile dal sito del produttore, che permette di c o -



Fig. 8

VDIPI1

- Connettore USB tipo A;
- Seconda porta USB disponibile sui pin del modulo;
- Selezione attraverso i jumper J3 e J4 del tipo di interfaccia (UART, SPI o FIFO parallela);
- Alimentazione a 5V;
- Uscita 3,3V ausiliaria (fino a 200mA);
- Led di presenza alimentazione, e led indicatori di traffico USB.

Specifiche tecniche

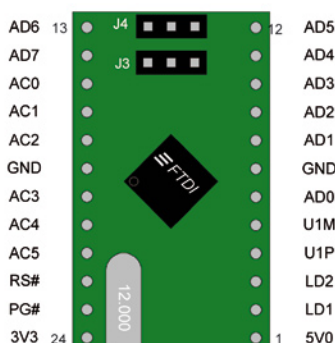


Fig. 9

Tabella 4

VDIP J3	VDIP J4	Modalità I/O
1-2 chiuso	1-2 chiuso	Seriale UART
1-2 chiuso	2-3 chiuso	SPI
2-3 chiuso	1-2 chiuso	FIFO parallela
2-3 chiuso	2-3 chiuso	Seriale UART

che simulano un connettore DIL 24 maschio, sul quale sono fisicamente riportati tutti i segnali indispensabili per comunicare con il chip. Questo dispositivo, le cui caratteristiche tecniche sono riportate in questa pagina, rappresenta la soluzione ideale per realizzare velocemente dei prototipi di sistemi basati sul chip, ma può es-

lezionata. A titolo informativo, in tabella 5 trovate le funzioni dei vari pin a seconda della modalità scelta.

Vinculum VDRIVE2

Il modulo VDRIVE2, riportato in figura 10, rappresenta la soluzione per collegare un dispositivo di archiviazione di massa, ad esempio un Flash Drive, a un prodotto già esistente. Ciò avviene semplicemente collegando il VDRIVE2 al sistema attraverso 4 linee di segnale e 2 di alimentazione

comunicare con il microcontrollore attraverso la porta SPI o la UART. Il VDRIVE2 è particolarmente adatto per effettuare prove di valutazione, prima di iniziare lo sviluppo di prodotti che usano il VNC1L; però, grazie al contenitore di cui è dotato, può anche essere incorporato in prodotti finiti quali elettrodomestici, data-logger industriali e prodotti in cui è possibile fare un upgrade del software attraverso un Flash Drive USB. In Figura 11 ed in tabella 6

Tabella 5

Pin	Nome Pin	UART	FIFO Parallela	SPI
6	AD0	TXD (output)	D0 (I/O)	SCLK (input)
8	AD1	RXD (input)	D1 (I/O)	SDI (input)
9	AD2	RTS (output)	D2 (I/O)	SDO (output)
10	AD3	CTS (input)	D3 (I/O)	CS (input)
11	AD4	DTR (output)	D4 (I/O)	
12	AD5	DSR (input)	D5 (I/O)	
13	AD6	DCD (input)	D6 (I/O)	
14	AD7	RI (input)	D7 (I/O)	
15	AC0	TXDEN (input)	RXF (output)	
16	AC1		TXE (output)	
17	AC2		WR (input)	
19	AC3		RD (input)	
20	AC4			



Fig. 10



Fig. 11

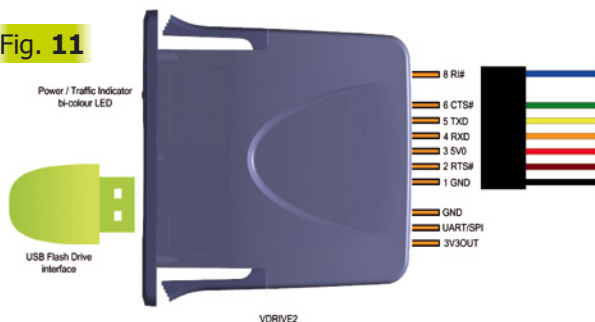


Tabella 6

Pin N°	Nome	Tipo	Descrizione
1	GND	Power	Massa
2	RTS#	Output	Request To Send
3	5V0	Power	Alimentazione 5V
4	RXD	Input	Ricezione asincrona dati
5	TXD	Output	Trasmissione asincrona dati
6	CTS#	Input	Clear To Send
7	NC	-	Non connesso
8	RI#	Input	Ring Indicator

Tabella 7

Pin N°	Nome	Tipo	Descrizione
5	SCLK	Input	SPI Clock Input, frequenza massima 12MHz
4	SD1	Input	SPI Serial Data Input
2	SD0	Output	SPI Serial Data Output
6	CS	Input	SPI Chip Select Input

il chip, vediamo un esempio in cui il VDRIVE2 è collegato ad un microcontrollore della famiglia PIC16F (Figura 12). Questo semplice circuito consente di realizzare un efficiente data logger, in cui il segnale analogico presente sul pin AC input viene convertito in digitale per mezzo dell'AD converter a 10 bit presente nel PIC, ed in seguito memorizzato su una Pen Drive, collegata al VDRIVE2.

Vinculum VF2F2

È l'unico dispositivo completamente standalone della famiglia Vinculum ed è l'unico che funziona in maniera completamente autonoma: l'unica cosa che necessita è l'alimentazione. La sua funzione principale è quella di interfacciare due memorie Flash USB, per effettuare il trasferimento di dati dalla porta 1 alla porta 2 (es. la copia di file dalla Flash di una fotocamera ad una Flash Drive USB). Alla pressione di un pulsante, tutti i file immagine della memoria sorgente vengono copiati in un'unica directory sulla memoria destinazione.

In questo dispositivo il chip VNC1L è programmato con il firmware

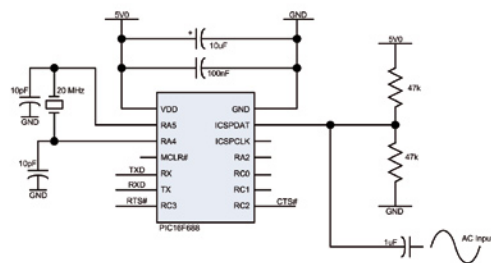


Fig. 12

VDRIVE2

- Utilizza l'integrato FTDI VNC1L-1A;
- Set comandi VDAC;
- Connettore USB tipo A;
- Alimentazione a 5 Vdc;
- Pin di selezione interfaccia UART o SPI;
- Pb-free, conforme alle direttive RoHS.

DF2FC, in grado di gestire tutti i file di tipo immagine.

Il VF2F2 è un accessorio ideale per gli utilizzatori di fotocamere digitali che vogliono fare un back-up delle proprie foto senza dover passare attraverso l'uso del PC.

Il dispositivo, le cui caratteristiche sono riportate nel riquadro alla pagina successiva, è alimentato da due batterie alcaline di tipo AAA e dispone di quattro pulsanti:

- **Button 1:** pulsante di Power ON; premerlo per 200ms per accendere il VF2F2. Il led giallo indica che la scheda è accesa. Dopo 30 secondi di inattività il dispositivo si spegne da solo per conservare le batterie.
 - **Button 2:** libero per eventuali applicazioni future.
 - **Button 3:** avvia il trasferimento dati dal device sorgente (USB Port 1) al device destinazione (USB Port 2). Il trasferimento dati in corso è segnalato da un'accensione in sequenza dei quattro led.
 - **Button 4:** pulsante di Power OFF; tenerlo premuto per 200ms per spegnere.
- Come già accennato, il dispositivo dispone di due porte USB:
- **USB Port 1:** è il la sorgente, a cui si

VF2F2

- Due connettori USB di tipo A;
- LED intelligenti che indicano l'enumerazione corretta del USB Flash Disk, funzione di copia file e condizioni di errore;
- DC-DC converter a bordo che alimenta il circuito a 3,3V e 5V Partendo da due celle AAA;
- VNC1L-1A programmabile attraverso il pin PROG# presente sul connettore J1.

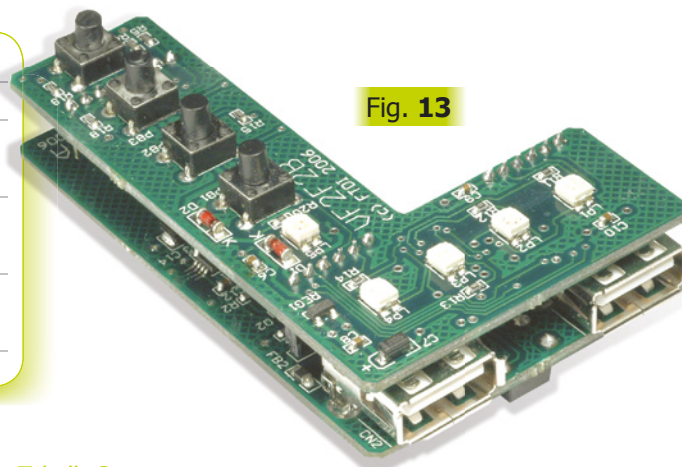
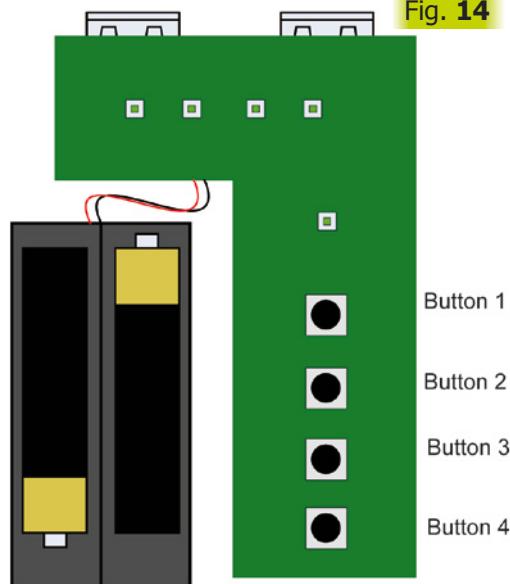


Fig. 13

Fig. 14



Il VF2F2 è inoltre dotato di un connettore (posto nel lato inferiore del circuito stampato su cui sono montati i pulsanti e identificato dalla sigla J3) sul quale sono state portate l'interfaccia UART e la porta per la riprogrammazione. Quest'ultima può essere utilizzata per riprogrammare il firmware del VNC1L-1A attraverso un altro dispositivo FTDI, il TTL-232R-3V3 (un convertitore da USB a seriale TTL).

Tabella 8

Pin N°	Nome	Tipo	Descrizione
1	RESET#	Input	Usato in combinazione col pin PROG# per riprogrammare via UART il VNC1L a bordo
2	PROG#	Input	Usato in combinazione col pin RESET# per riprogrammare via UART il VNC1L a bordo
3	GND	Power	Massa
4	RTS#	Output	Request To Send
5	TTL PWR	Power	3.3V
6	RXD	Input	Ricezione asincrona dati
7	TXD	Output	Trasmissione asincrona dati
8	CTS#	Input	Clear To Send

deve connettere la fotocamera digitale (o il Flash Drive USB sorgente).

• **USB Port 2:** è la destinazione, a cui si deve connettere il Flash Drive USB nel quale si vogliono copiare i file. I due led bicolore rosso / verde associati alle porte indicano lo stato dei due device USB connessi. Questi due led sono spenti quando nessun dispositivo è collegato alle porte USB, rossi durante la fase di riconoscimento dei dispositivi e verdi se i dispositivi sono stati riconosciuti e sono pronti per il trasferimento dati.

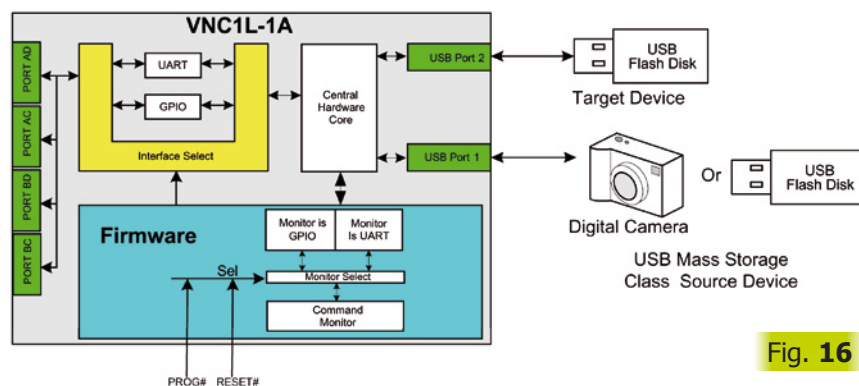


Fig. 16

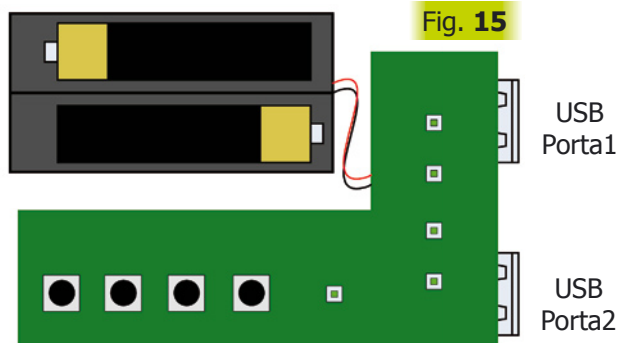


Fig. 15

Durante la fase di riprogrammazione, il VF2F2 è collegato al TTL-232R-3V3 e prende l'alimentazione da esso invece che dalle batterie AAA. È inoltre disponibile un software applicativo per PC (VPROG) con cui

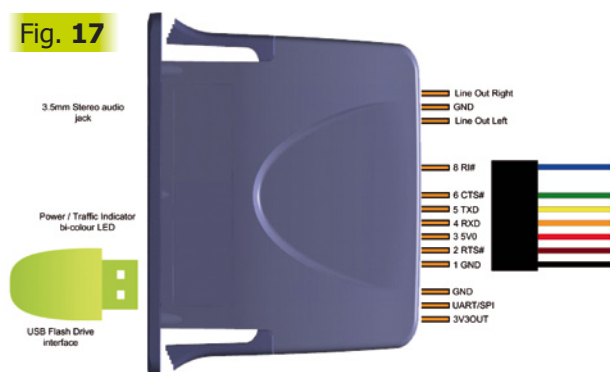
effettuare la riprogrammazione. Nella tabella 8 abbiamo riportato a titolo informativo le descrizioni dei segnali presenti sul connettore J3

mentre in figura 16 è stato schematizzato il funzionamento del software del VF2F2.

Vinculum VMUSIC2

Con questo prodotto concludiamo l'articolo sulla famiglia Vinculum. Lo abbiamo lasciato per ultimo non perché sia il meno importante, bensì perché è il più particolare: esso infatti è in grado sia di leggere e ➤

Fig. 17

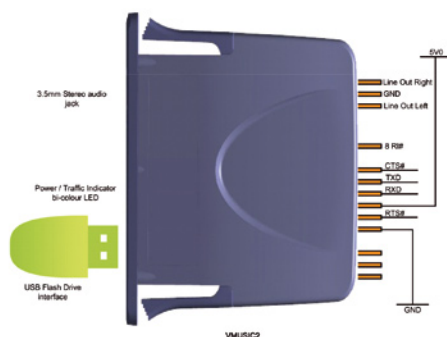


scrivere le Pen Drive (esattamente come il VDRIVE2, dal quale eredita anche l'aspetto) sia di riprodurre dei file audio digitali in formato MP3 (e negli altri più diffusi) direttamente dalla memoria della Pen Drive. Per funzionare, questo dispositivo necessita solamente di 4 linee di segnale. Un'estensione dei comandi del firmware VDAC (usato nel VDRIVE2) permette di riprodurre un file audio, controllandone il volume, il

Tabella 9

Pin N°	Nome	Tipo	Descrizione
1	GND	Power	Massa
2	RTS#	Output	Request To Send
3	5V0	Power	Alimentazione 5V
4	RXD	Input	Ricezione asincrona dati
5	TXD	Output	Trasmissione asincrona dati
6	CTS#	Input	Clear To Send
7	NC	-	Non connesso
8	RI#	Input	Ring Indicator

bilanciamento e il tempo trascorso. Il VMUSIC2, le cui caratteristiche sono riportate nel riquadro a destra, è il prodotto ideale per aggiungere facilmente un riproduttore MP3 ad un sistema sprovvisto di riprodu-



zione audio, ma dotato almeno di una porta seriale. In Figura 17 e in tabella 9 abbiamo riportato la disposizione dei piedini e la relativa funzione mentre in figura 18 abbiamo un'applicazione di esempio, in cui si utilizza un PIC16F per gestire quattro tasti, che svolgono le funzioni di play, pause, stop, skip forward, skip back, volume up e volume down. Il codice del PIC rileva la pressione dei tasti, pressione che viene convertita in un comando seriale per il VMUSIC2.

Giunti a questo punto, riteniamo di aver fornito informazioni sufficienti

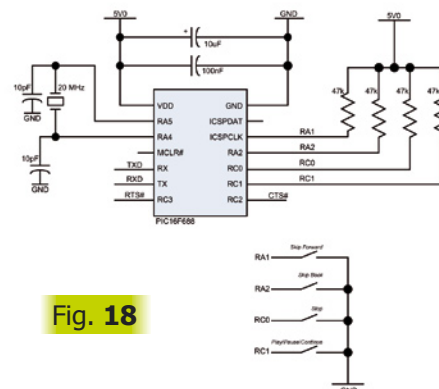


Fig. 18

per avere un completo quadro d'insieme di tutti i prodotti della famiglia Vinculum. Il chip VNC1L-1A rappresenta un sistema facile, veloce ed economico per aggiungere una porta USB Host ai sistemi che, per loro natura, ne sono sprovvisti, mentre i moduli dedicati (VDIP1, VDRIVE2, VF2F e VMUSIC2) sono stati realizzati per risolvere ognuno una problematica specifica. A coloro che vorranno cimentarsi nella sperimentazione, auguriamo buon lavoro, annunciando sin d'ora che abbiamo in cantiere una serie di progetti pratici che utilizzano questi dispositivi.

per il

MATERIALE

I prodotti Vinculum sono disponibili in commercio ai seguenti prezzi: Cod. VNC1L-1A (Chip USB Host Controller Device) Euro 18,00; Cod. VDIP1 (Modulo Host USB) Euro 46,00; Cod. VDRIVE2 (Modulo USB Host Flash Disk) Euro 46,00; Cod. VF2F2B (Modulo Back-up) Euro 72,00; Cod. VMUSIC2 (Modulo riproduttore audio) Euro 78,00. 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>

Termostato con controllo GSM

di Boris Landoni

Consente di gestire da remoto il riscaldamento in abitazioni di villeggiatura. Funziona in parallelo al termostato esistente e consente di stabilire la temperatura desiderata. Può inoltre avvisare, tramite SMS o chiamate, quando la temperatura rilevata si discosta dalla finestra impostata o l'ingresso di allarme è attivo.



Una delle applicazioni più note cui si deve la diffusione di dispositivi di telecomando telefonico, probabilmente quella più importante, è la gestione a distanza degli impianti di riscaldamento delle case in montagna: opportunamente collegato, un sistema del genere permette di avviare la caldaia, magari un paio d'ore prima di arrivare per trascorrere il week-end, così da trovare l'abitazione a una temperatura confortevole. L'utilità del telecomando si manifesta anche in situazioni lavorative: ad esempio può servire per avviare il riscaldamento di un ufficio o un laboratorio se ci si deve recare lì a compiere un lavoro impre-

visto durante un periodo di chiusura (fine settimana o ferie di fine d'anno). Quando ancora non esistevano i cellulari, il comando a distanza si effettuava con il telefono fisso, mediante interfacce, collocate sul luogo in cui si trovava l'apparato da comandare, alle quali si inviavano i bitoni corrispondenti ai comandi, mediante la tastiera del telefono. Con la massiccia diffusione della telefonia mobile sono nati telecomandi più sofisticati, gestibili semplicemente con una chiamata (alla quale non è richiesta risposta) o con messaggi di testo. Considerando proprio quella che è l'applicazione più significativa e pensando alle esigenze di ➤

quanti finora si sono accontentati di un generico telecontrollo adattato alla gestione del riscaldamento, abbiamo pensato di progettare un dispositivo GSM finalizzato specificatamente al monitoraggio ed alla impostazione della temperatura di un locale da remoto: opportunamente collegato alla caldaia, permette di controllarla al fine di mantenere la temperatura impostata dall'utente, ma anche di comunicare a distanza, in automatico o su richiesta, il valore attuale.

Tutti i comandi relativi al controllo ed alle impostazioni del dispositivo sfruttano gli SMS, sebbene localmente sia previsto un pulsante per scegliere manualmente la modalità di funzionamento tra le tre possibili: forzata (M), automatica (A) e asservita (T). Nel primo caso (M) l'impianto di riscaldamento **viene forzato a funzio-**

mento sono stati previsti per rendere il sistema il più versatile possibile e per integrarlo facilmente in un impianto esistente. Ribadiamo che per passare da una modalità all'altra è possibile agire localmente, mediante la pressione di un pulsante, oppure da remoto, tramite l'invio di un apposito SMS: ciò offre la massima flessibilità e semplicità di impiego.

Proprio per rendere ancora più completa l'integrazione con l'impianto esistente, l'unità che vi proponiamo realizza anche la funzione di teleallarme: dispone di un ingresso optoisolato a livello di tensione che può essere collegato alla spia di malfunzionamento della caldaia e configurato per adattarlo alle diverse condizioni di funzionamento (ciò per dare allarme se riceve tensione oltre la soglia o se è a zero volt); così facendo, nel

o telefonate. Il circuito è in grado di memorizzare al massimo otto numeri telefonici cui inviare gli allarmi sotto forma di messaggi di testo o semplici chiamate.

Schema elettrico

Prima di vedere come impiegare in pratica il telecontrollo, riteniamo utile dare uno sguardo al suo schema elettrico. Notiamo per prima cosa che l'insieme è gestito da un microcontrollore di casa Microchip, il quale controlla l'attività del modulo GSM/GPRS Telit GM862 (da noi già usato in svariate applicazioni), legge la temperatura ambiente mediante una sonda intelligente Dallas DS1820, controlla la condizione logica di due ingressi e comanda il relé cui è affidato il compito di aprire e chiudere il contatto di accensione del sistema di riscaldamento. Il circuito dispone di un ingresso optoisolato, la cui funzione è stata descritta poc'anzi, e di un altro ingresso (IN2) destinato, nella modalità asservita, a leggere il termostato esistente nell'impianto per ripeterne l'attività usando il relé come attuatore.

Detto ciò, diamo un'occhiata più approfondita allo schema elettrico: l'alimentazione è a tensione continua, anche non stabilizzata (applicata ai punti + e - PWR) di valore compreso tra 5 e 32 V, che viene filtrata a valle del diodo di protezione contro l'inversione di polarità (D1) mediante i condensatori C1 e C2; il regolatore switching U1 ricava 3,6 V stabilizzati con i quali alimenta tutto il resto del circuito. Il regolatore è basato sul chip MC34063, utilizzato nella classica configurazione di regolatore PWM serie a carica d'induttanza, la cui tensione di uscita dipende dall'energia immagazzinata in L1; viene stabilizzato mediante la componente retrocessa dal partitore resistivo R2/R3, che serve a fissare a 3,6 V la componente livellata ai capi di C4 e C5. A valle di C4 e C5, si trova un filtro L/C che pulisce quasi totalmente dai resi-

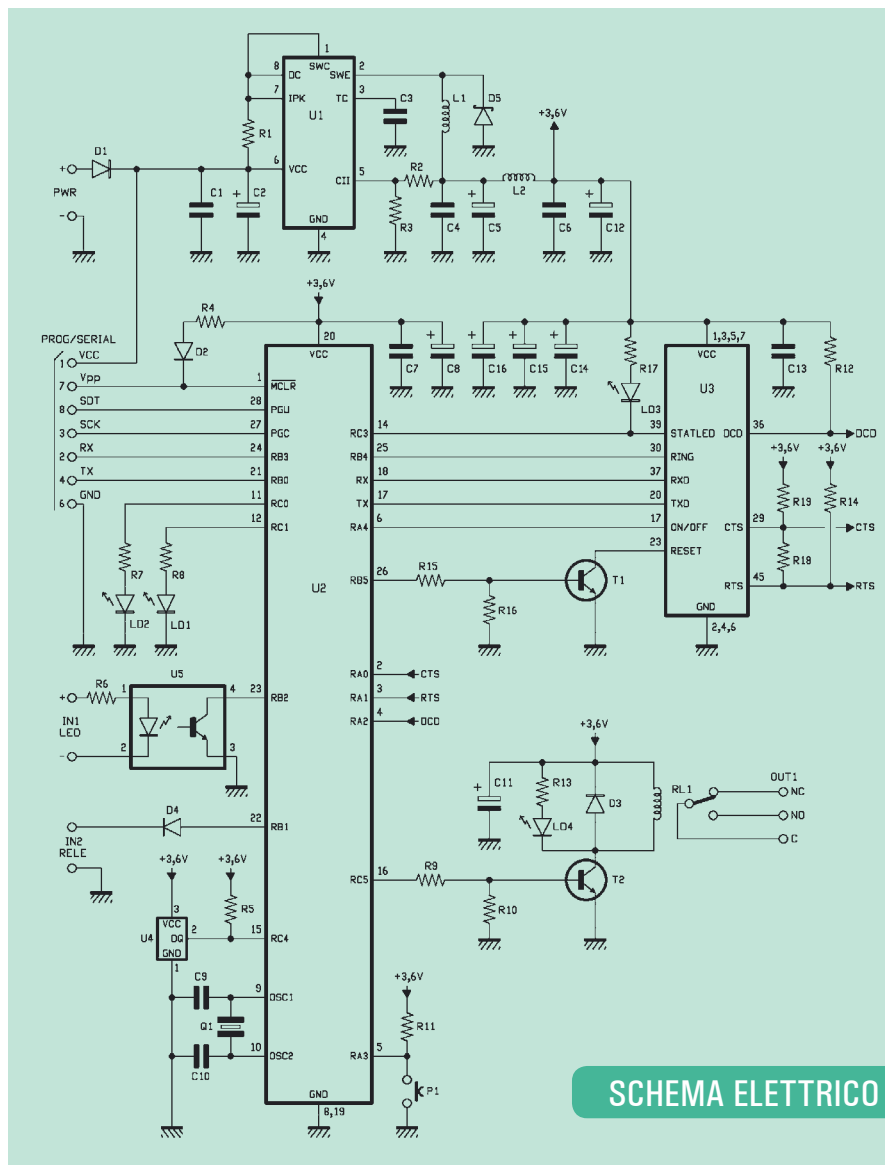
- Escursione di temperature: da - 55° a + 99°C.
- Ingressi, 2 a livello logico:
 - uno di allarme, optoisolato (allarme caldaia in blocco),
 - uno controllato dal contatto NO del termostato esterno.
- Uscite: 1 a relé, gestibile in:
 - automatico, manuale o asservita a termostato esterno.
- Programmazione funzionamento: tramite SMS, pulsante locale e PC.
- Alimentazione: da 5 a 32 Vdc, 300 mA nominali.

nare in continuazione, a prescindere dalle impostazioni di temperatura, e ciò fino a quando non si passa ad una differente modalità di funzionamento.

Nel secondo caso (A), **l'impostazione della temperatura avviene tramite un SMS** ed il circuito utilizza il sensore di cui è dotato per verificare la temperatura ambiente e fare sì che questa venga mantenuta al valore definito dall'utente con l'SMS.

Nel terzo caso (T), il dispositivo **ripristina il collegamento tra il termostato esistente e la caldaia** in modo da far funzionare l'impianto in maniera tradizionale, come se la scheda di controllo GSM non esistesse. Questi tre modi di funziona-

mento caso l'impianto vada in blocco (per un'ostruzione della canna fumaria, l'eccessivo abbassamento della pressione del liquido nei caloriferi, la mancanza di gas e via di seguito) il sistema comunica a distanza l'anomalia con un SMS di allarme o una telefonata. L'allarme sull'ingresso optoisolato vale solo quando il circuito lavora in modalità automatica (A). Ma non è tutto qui: è previsto anche un allarme termico, che scatta quando la temperatura rilevata nell'ambiente esce da una finestra impostata dall'utente, ovvero eccede la soglia massima o scende al disotto di quella minima; come per quello collegato alla spia della caldaia, l'allarme termico può determinare l'invio di SMS



SCHEMA ELETTRICO

dui di commutazione la componente continua usata per alimentare il resto del circuito: consta della bobina L2 e dei condensatori C6, C7, C8, C12, C14, C15 e C16, che sono in gran numero per poter fornire la corrente necessaria nei picchi di assorbimento che il modulo cellulare richiede quando è in trasmissione.

Il microcontrollore usato per gestire l'insieme è un PIC18F2620 Microchip che, inizializzate le linee di I/O, verifica la condizione logica dell'ingresso a livello di tensione (RB2) e quella delle linee RB4, RC3, RX, che servono a ricevere dal modulo cellulare le principali segnalazioni; precisamente, RB4 viene usata per rilevare l'arrivo delle chiamate entranti,

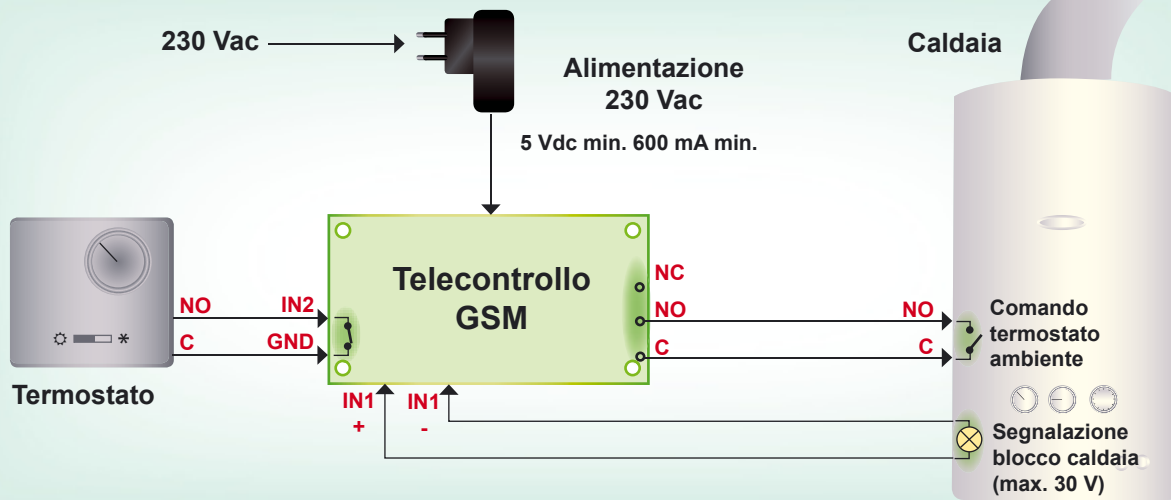
mentre RC3 controlla il "led di campo" (LD3). La relativa uscita (siglata STATLED) pulsa alla frequenza di 1 Hz quando il GM862 sta cercando la rete radiomobile, mentre fornisce un impulso a zero logico della durata di 0,5 sec. seguito da una pausa di 2 sec. quando il modulo ha agganciato il segnale. Dalla frequenza e dalla durata degli impulsi uscenti dal piedino 39 del modulo Telit, il PIC18F2620 comprende le condizioni della copertura radiomobile e si regola di conseguenza; ad esempio, se a seguito dell'entrata in allarme dell'ingresso optoisolato deve inviare gli SMS o effettuare le chiamate del caso, ma rileva che il modulo cellulare è fuori campo, attende che il modulo Te-

lit si riconnetta alla rete GSM/GPRS e solo allora effettua le chiamate. Il tentativo di chiamare o inviare l'SMS viene ripetuto solo tre volte, dopodiché il dispositivo rinuncia.

Il micro contiene una UART accessibile dai piedini 17 (trasmissione) e 18 (ricezione) che usa per dialogare con il cellulare; per l'esattezza, mediante il primo pin (TX) interroga ciclicamente il modulo Telit al fine di verificare l'arrivo di SMS, mentre TX ed RX insieme servono, durante le procedure di chiamata, ricezione e trasmissione messaggi, per la comunicazione tra il microcontrollore e il GM862. Sempre dell'UART, vengono usati i segnali di controllo CTS (Clear To Send) RTS (Request To Send) e DCD (Data Carrier Detect) che corrispondono a quelli del modulo Telit.

Completa la carrellata degli I/O destinati al cellulare RB5, che gestisce, mediante il transistor T1 (configurato come open-collector) la linea di reset. Il pulsante per la gestione locale della modalità di funzionamento del dispositivo viene letto mediante la linea RA3, configurata come ingresso e dotata di resistore di pull-up esterno (R11) e quindi attiva a livello basso. La lettura degli ingressi avviene mediante le linee RB1 ed RB2, entrambe configurate come input e provviste di pull-up interno: la prima viene protetta mediante il diodo D4, che serve ad evitare che applicando a IN2 tensioni che superano i 5 V il microcontrollore possa essere danneggiato; l'input corrispondente è attivo a zero logico. All'IN2 deve essere collegato il contatto normalmente aperto dell'eventuale termostato già esistente nell'impianto, ma solo se si intende attivare la modalità asservita (T): infatti in questa il relé RL1 segue le vicende del termostato.

Come accennato, il telecontrollo permette di monitorare lo stato di una linea elettrica, quale, ad esempio, quella corrispondente alla spia di allarme della caldaia; per leggerlo, il micro ➤



Una volta completata la costruzione del telecontrollo, dovete connetterlo all'impianto di riscaldamento: l'ingresso IN1 va collegato in parallelo alla lampadina spia della caldaia che ne segnala il blocco; nell'effettuare il cablaggio accertatevi che la spia sia in bassa tensione e che il positivo sia rivolto sul + (resistenza R6) dell'ingresso. È anche possibile collegarsi a spie alimentate in corrente alternata, purché funzionanti a non più di 100 V (picco-picco); in tal caso non serve rispettare alcuna polarità. Quanto al relé, il contatto normalmente aperto del suo scambio deve essere usato per chiudere il comando remoto della caldaia; se questa risulta collegata a un proprio termostato, interrompete

il collegamento ed effettuate le connessioni illustrati nel disegno: inviate i due fili del termostato all'ingresso IN2 della scheda e collegate le uscite NO e C del relé al comando della caldaia. Quanto al termostato esistente, se avete a che fare con un elemento del tipo estate/inverno bisogna considerare il solo contatto corrispondente al comando della caldaia; insomma, non usate il contatto riservato al condizionamento. Se il termostato ha un solo contatto ed è elettronico o comunque dispone del selettore estate/inverno, deve essere disposto su inverno (simbolo col fiocco di neve). Realizzato l'impianto, alimentate l'apparecchio con un alimentatore capace di fornire almeno 5V / 600mA.

sfrutta il fotoaccoppiatore U5, che gli permette di rilevare tensioni (di valore compreso fra 3 e 30 V) in circuiti elettrici di qualsiasi tipo restando galvanicamente isolato da essi. Il rilevamento funziona così: quando all'ingresso IN1 è applicata una tensione di almeno 3 volt, il fotoaccoppiatore ha il led acceso e il fototransistor d'uscita in conduzione, cosicché il collettore (piedino 5) è a circa zero volt, per effetto della caduta sul resistore di pull-up interno configurato, durante l'inizializzazione degli I/O, per la linea RB2. Se l'ingresso non è polarizzato, l'optoisolatore è interdetto e il suo pin 4 risulta a livello alto. Va notato che RB1 determina solo azioni locali ed esclusivamente nella modalità asservita al termostato esistente mentre RB2 produce l'esecuzione della sequenza di allarme, consistente nell'invio, in base all'impostazione fatta dall'utente, di SMS o chiamate dirette ai numeri memoriz-

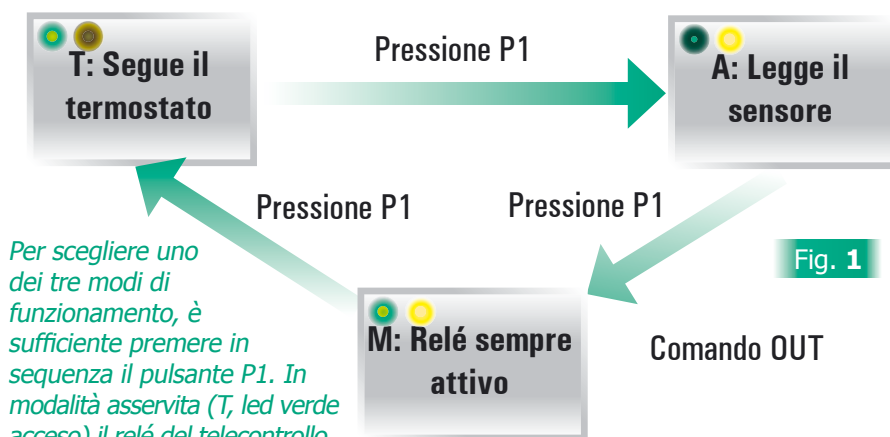
zati in lista. La sequenza parte quando l'ingresso IN1 LED viene sottoposto a una differenza di potenziale superiore a quella di soglia (circa 3 volt). Per il comando della caldaia, il microcontrollore si avvale del relé RL1, pilotato mediante la linea RC5 e il transistor NPN T2; più esattamente, quando il PIC pone RC5 a livello logico alto manda in saturazione il transistor, determinando lo scorrimento di corrente nella bobina del relé, condizione evidenziata dall'accensione del led LD4. Per proteggere la giunzione di collettore del T2 nei passaggi dalla saturazione all'interdizione, quando l'induttanza della bobina del relé genera picchi di tensione inversa, è stato collegato il diodo D3, che elimina, appunto, tali impulsi. Del RL1 viene reso disponibile l'intero scambio, così da poter gestire circuiti che richiedono un contatto normalmente chiuso o uno normal-

mente aperto. Sempre del relé si osservi come, sebbene sia del tipo con bobina a 5 volt, nel nostro circuito funzioni ad appena 3,6 V; ciò è possibile perché il modello che abbiamo scelto riesce a muovere lo scambio anche a meno di 3,5 volt. Finora si è visto come il micro gestisce tutto quello che gli sta attorno, ma non è stato spiegato in che modo rileva la temperatura ambiente, cosa che deve fare, nella modalità automatica per decidere quando accendere la caldaia e in tutti gli altri casi per comunicare, su richiesta remota, il valore attuale; il rilevamento della temperatura serve anche quando l'utente desidera sfruttare la funzione di teleallarme termico, che gli consente di essere avvisato del superamento della soglia massima o dell'abbassamento oltre quella minima impostata. La temperatura viene letta con l'ausilio di una precisissima sonda DS1820 di Dallas Semiconductor (vedere

apposito riquadro), capace di misurare temperature comprese nel range $-5 \div 150$ °C con un'accuratezza di appena $\pm 0,5$ °C (nel campo $-10 \div 85$ °C), di digitalizzarle e di renderle disponibili su una linea seriale unidirezionale, collegata al pin RC4 del micro.

Delle linee di I/O usate dal microcontrollore abbiamo finora traslasciato /MCLR, PGU, PGC, RB0 ed RB3: costituiscono un bus usato sia per la programmazione in-circuit del PIC18F2620 che per il dialogo con il Personal Computer; più esattamente, PGU e PGC sono, rispettivamente, il Serial Data (SDA) e il Serial Clock (SCL) del bus I²C per la comunicazione con il programmatore, linee riferite alla massa comune GND. L'insieme delle linee è reso accessibile da un connettore femmina RJ45, siglato PROG.

La connessione al PC serve nel caso si intenda programmare le modalità e tutti i parametri di funzionamento dell'unità avvalendosi dell'apposita utility progettata per l'ambiente



Per scegliere uno dei tre modi di funzionamento, è sufficiente premere in sequenza il pulsante P1. In modalità asservita (T, led verde acceso) il relé del telecontrollo riproduce lo stato del termostato ambiente. In modalità automatica (A, led giallo acceso) lo stato del relé dipende dalla temperatura ambiente e dalle impostazioni effettuate tramite SMS; in modalità manuale (M, entrambi i led accesi), invece, il relé è sempre attivo, a prescindere da tutte le altre impostazioni. Il passaggio da un modo di funzionamento all'altro può anche essere comandato con opportuni SMS.

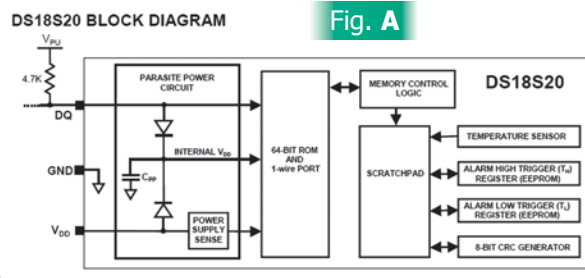
Windows: un semplice programma scaricabile gratuitamente dal nostro sito Web www.elettronica.in e che, installato e lanciato, dà accesso a un pannello di controllo dal quale l'utente può, facendo clic nelle apposite caselle, introdurre o modificare dati quali le soglie dell'allarme termico, la temperatura da mantenere, eccetera.

Il funzionamento

Descritto lo schema elettrico, vediamo ora come funziona il telecontrollo: all'accensione, dopo l'inizializzazione degli I/O, il PIC lancia l'Easy Setup, che è una modalità grazie alla quale si può configurare rapidamente il funzionamento del dispositivo; si attiva se, nei primi tre minuti di esercizio, >

Il sensore di temperatura DS1820

Il DS1820 di casa Dallas, il cui schema a blocchi è visibile in figura A, è un dispositivo a microcontrollore dotato di un sensore termico a semiconduttore, la cui corrente di fuga, legata alla temperatura, viene campionata, convertita in formato digitale e inserita in una stringa generata secondo un protocollo proprietario Dallas chiamato "1-wire". L'AD converter della sonda ha una risoluzione di 9 bit ed il bit meno significativo ha valore 2^{-1} (ossia $0,5$ °C). Durante il campionamento, il risultato della lettura viene collocato in due registri da 8 bit ciascuno: il primo contiene i primi 8 bit del valore campionato, mentre il secondo contiene l'ultimo bit restante e 7 bit di segno. Quando richiesto dal microcontrollore, la sonda trasmette serialmente tutti i bit contenuti nei registri, suddividendoli in due byte (uno per registro). Come visibile nella tabella A1,



Tab. A1 - Formato dei registri della temperatura

LSByte	bit 7 2^6	bit 6 2^5	bit 5 2^4	bit 4 2^3	bit 3 2^2	bit 2 2^1	bit 1 2^0	bit 0 2^{-1}
MSByte	bit 15 S	bit 14 S	bit 13 S	bit 12 S	bit 11 S	bit 10 S	bit 9 S	bit 8 2^7

Tab. A2 - Relazione tra temperatura e dati

Temperatura	Uscita digitale (binaria)	Uscita digitale (esadecimale)
+85,0°C	0000 0000 1010 1010	00AAh
+25,0°C	0000 0000 0011 0010	0032h
+0,5°C	0000 0000 0000 0001	0001h
0°C	0000 0000 0000 0000	0000h
-0,5°C	1111 1111 1111 1111	FFFFh
-25°C	1111 1111 1100 1110	FFCEh
-55°C	1111 1111 1001 0010	FF92h

il byte meno significativo contiene la parte bassa della conversione da analogico a digitale (bit0 ÷ bit7, aventi peso da 2^{-1} a 2^7), mentre il byte più significativo contiene il segno e il rimanente bit del campione (bit8, di peso 2^8). Notate che il valore esadecimale 0000 (Tab. A2) corrisponde ad una temperatura di 0 °C e che temperature negative sono segnalate dai primi 7 bit dell'MSByte a 1 logico. Questa sonda è in grado di rilevare temperature comprese tra -55 e $+150$ °C, garantendo una precisione di $0,5$ °C in gamma $-10 \div +85$ °C.

riceve una chiamata da un telefono che chiami senza nascondere l'ID. In tal caso, il telecontrollo memorizza nella prima posizione della propria lista il numero del telefono che ha effettuato la chiamata.

Che riceva o meno la chiamata, trascorsi tre minuti il circuito abbandona l'Easy Setup ed entra nel normale funzionamento, nel quale comanda la caldaia nella modalità richiesta dall'utente, verifica l'eventuale allarme in arrivo all'ingresso optoisolato e gestisce il relé.

Alla prima accensione, a meno che l'unità non sia stata collegata ad un Personal Computer e da esso configurata, per impostazione predefinita funziona in automatico (A) ossia attiva il relé quando la temperatura rilevata nell'ambiente è minore di quella impostata dall'utente, ovvero, se questi non l'ha mai modificata, di quella predefinita (22 °C). La condizione è evidenziata dall'accensione del led giallo (LD2).

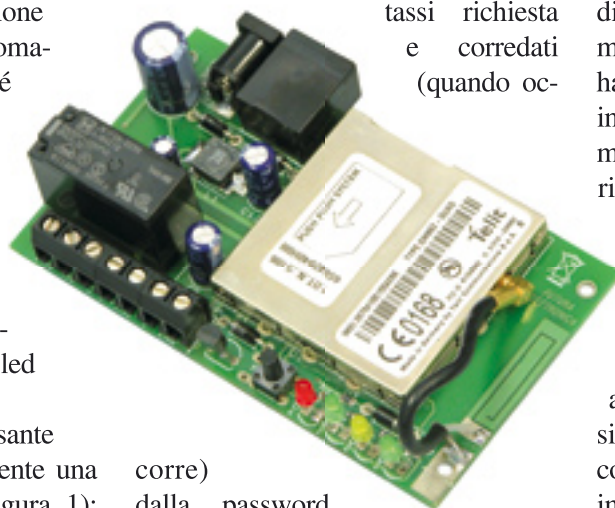
Localmente, agendo sul pulsante si può definire sequenzialmente una diversa modalità (vedere figura 1): premendo P1 si passa dall'automatica alla forzata (M) nella quale RL1 è sempre attivo fin quando non lo si disattiva da SMS o non si cambia modalità; la condizione è visualizzata dall'accensione dei led LD1 (verde) e LD2 (giallo). Premendo il pulsante un'altra volta, si passa nel modo asservito (T) che viene segnalato dall'accensione del solo LD1: qui, il circuito segue le vicende del termostato eventualmente collegato all'ingresso IN2, nel senso che il relé scatta (chiude C su NO) se l'input viene cortocircuitato e rimane a riposo nel caso contrario (ingresso aperto). Un'ulteriore pressione di P1 riporta l'unità nella modalità automatica.

Ora, se si preme nuovamente il tasto ricomincia la sequenza, nel senso che si riparte con il ciclo di tre passi appe-

na descritto. Resta inteso che se da remoto (quindi tramite SMS) si ordina al circuito di impostare una modalità differente da quella decisa localmente con P1, l'ordine è prioritario, nel senso che il comando da telefono (ma anche quello da computer) annulla quello da pulsante.

Impostazioni e comandi

Visto il funzionamento dell'apparato, bisogna spiegare come gestirlo da remoto via telefono; lo facciamo con una premessa e cioè che le maniere per comandarlo sono due: la prima consiste nell'inviare gli opportuni SMS secondo la sintassi richiesta e corredati (quando oc-



corre) dalla password, indispensabile ad evitare che la configurazione possa essere operata da estranei; la seconda consta nell'introdurgli, mediante Easy Setup il numero del telefono cellulare abilitato a fare le successive impostazioni tramite SMS: in tal caso buona parte dei messaggi potrà essere inviata senza la password, perché il circuito ammette i comandi provenienti da un cellulare il cui numero sia già in lista.

Gli SMS di impostazione possono essere inviati da qualsiasi telefonino purché i relativi messaggi comprendano la password; per rendere più semplici le procedure, alcuni comandi, se inviati da numeri già in lista, possono fare a meno della password. I numeri contenuti in tale lista sono gli stessi ai quali (se abilitati) ver-

ranno inviati i messaggi in caso di allarme. Il telecontrollo accetta SMS multipli, ossia contenenti più di un comando o comandi riguardanti uno o più numeri telefonici; i comandi devono essere separati ognuno da quello successivo con una virgola. Ciò permette di risparmiare tempo e denaro. Normalmente un comando di impostazione prevede una risposta che il dispositivo invia tramite SMS; di conseguenza un comando multiplo produrrà più messaggi di risposta; per evitarli, bisogna disattivare la risposta, cosa che si ottiene terminando l'SMS pluricomando con il testo RISP. Disattivare la risposta permette di risparmiare denaro, perché ogni messaggio inviato dal telecontrollo ha un costo; se si sfrutta un SMS per inviare più comandi lo si fa per risparmiare, quindi è evidente che se poi la risposta che ne deriva consta di molti SMS, il risparmio sfuma...

Dopo questa necessaria premessa, possiamo iniziare a descrivere i comandi: tenete presente che dedicheremo maggiore attenzione ai comandi più complessi, mentre lasceremo alla Tabella 1 il compito di descrivere la sintassi delle impostazioni più semplici.

Iniziamo con il comando per la modifica della password, che consiste in un SMS del tipo *PWDxxxxx;pwd*, dove al posto di xxxxx si deve scrivere la nuova password (numerica, di cinque cifre...) mentre *pwd* indica la password attuale; la password predefinita è 12345.

La memorizzazione di uno degli otto numeri si effettua inviando un SMS contenente il testo *NUMx+nnnnnnnnnnnnnnnn;pwd*, dove la x definisce la posizione in rubrica (1÷8) del numero che si sta memorizzando, *nnnnnnnn* il numero e *pwd* l'attuale password. Il tutto va scritto senza spazi in mezzo. Sono ammessi numeri di 19 cifre; notate che il + è quello che, nel prefisso internazionale composto dai cellulari, sostituisce lo 00. Per esempio, l'inserimento del

Tabella 1 - Elenco comandi

FUNZIONE	COMANDO SMS	VALORE DI DEFAULT	PWD OBBLIGATORIA
PASSWORD (PWD)		12345	
CAMBIARE LA PASSWORD	PWDxxxxx; 12345	12345	SEMPRE
MEMORIZZARE UN NUMERO IN POSIZIONE "x" (fino ad 8 numeri, al massimo di 19 caratteri ognuno)	NUMx + 392229876543;12345	-	SE LA POSIZIONE x NON È VUOTA
CANCELLARE UN NUMERO	NUMx;12345	-	SEMPRE
VERIFICARE I NUMERI MEMORIZZATI	NUM?;12345	-	SEMPRE
EFFETTUARE IL RESET COMPLETO DEL TELECONTROLLO	RES	-	SEMPRE
ATTIVARE IL RELÉ IN MODALITÀ BISTABILE	OUT:ON	Relé a riposo	
DISATTIVARE IL RELÉ IN MODALITÀ BISTABILE	OUT:OFF	Relé a riposo	
RIPRISTINARE LO STATO RELÉ DOPO UN BLACK OUT (x = 1: ripristino attivo. x=0: relé disattivato dopo il black-out)	RIPx	0	
RICHIEDERE LO STATO DELLA FUNZIONE "RIPRISTINO"	RIP?	-	
IMPOSTARE LA MODALITÀ DI FUNZIONAMENTO (T = Termostato, A = Automatico, M = Manuale)	FUN:X	A	
RICHIEDERE LA MODALITÀ DI FUNZIONAMENTO	FUN:?		
RICHIEDERE LA TEMPERATURA	TEMP		
AZZERARE LE TEMPERATURE MINIMA E MASSIMA	TRES		
IMPOSTARE LA TEMPERATURA DA MANTENERE (00 ÷ 99)	NOR:XX	22	
IMPOSTARE L'ALLARME PER TEMPERATURA MINIMA (y = "+" o "-"; xx = 00 ÷ 99)	ALLMIN:yxx		
IMPOSTARE L'ALLARME PER TEMPERATURA MASSIMA (y = "+" o "-"; xx = 00 ÷ 99)	ALLMAX:yxx		
RICHIEDERE LE TEMPERATURE DI ALLARME	ALL?		
TEMPO INIBIZIONE ALLARME TEMPERATURA (da 00 a 59 min.)	INT:mm	5	
TEMPO INIBIZIONE INGRESSO CALDAIA (da 00 a 59 min.)	INC:mm	5	
IMPOSTARE I NUMERI A CUI VENGONO INVIATI GLI SMS	SMSxxxxxxx: ON/OFF	Tutti attivi	SEMPRE
IMPOSTARE I NUMERI A CUI VIENE FATTO UNO SQUILLO	VOCxxxxxxx: ON/OFF	Tutti attivi	SEMPRE
INVIARE AUTOMATICAMENTE UN SMS ALL'ACCENSIONE (x = 1 per l'invio automatico; x = 0 per disattivare l'invio)	AVVx	0	
MODIFICARE IL TESTO DI: ALLARME TEMPERATURA SUPERIORE A SOGLIA ALLMAX (massimo 100 caratteri)	THI:xxxxxxxxxx	ALLARME!! SUPERATA SOGLIA TEMPERATURA	
MODIFICARE IL TESTO DI: ALLARME TEMPERATURA INFERIORE A SOGLIA ALLMIN (massimo 100 caratteri)	TLO:xxxxxxxxxx	ALLARME!! SUPERATA SOGLIA TEMPERATURA	
MODIFICARE IL TESTO DI: ALLARME INGRESSO CALDAIA QUANDO È PRESENTE TENSIONE (massimo 100 caratteri)	TIN:xxxxxxxxxx	ALLARME INGRESSO!!	
MODIFICARE IL TESTO DI: MESSAGGIO ACCENSIONE (massimo 100 caratteri; invio solo al primo numero in memoria)	TSU:xxxxxxxxxx	SYSTEM STARTUP	
RICHIEDERE LA TEMPERATURA (comando TEMP)	Chiamata telefonica al numero del telecontrollo		
DISABILITARE LA RISPOSTA PER UN MULTIMESSAGGIO	RISP (in coda al multimessaggio)		
EASY SET UP (memorizza in posizione 1 il primo numero che chiama entro 3 minuti dall'accensione)	Chiamata telefonica con identificativo chiamante visibile		

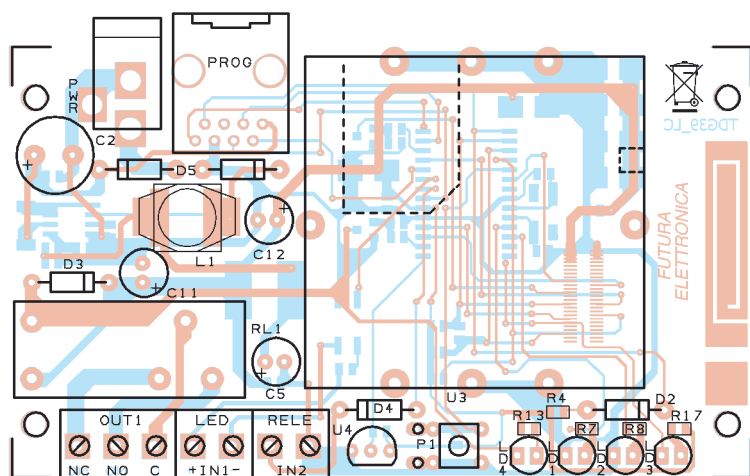
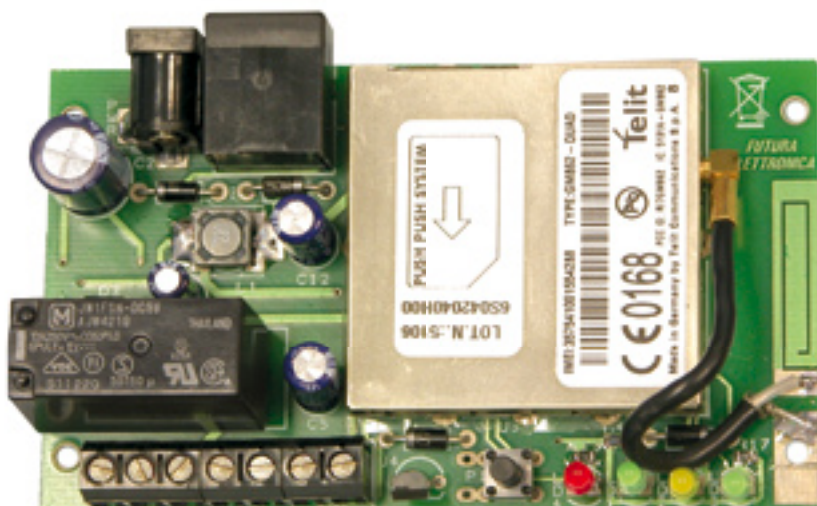
numero 0039335845676 in ottava posizione si opera con un comando del tipo: *NUM8+39335845676;pwd*.

Per memorizzare un numero inviando il comando da un cellulare già in lista è necessario introdurre la password solamente quando si tenta di sovrascrivere una posizione già occupata; se invece si deve inserire un numero in una posizione vuota basta inviare un SMS contenente *NUMx+nnnnnnnnnnnnnnnn*. Resta inteso che se si invia il comando da un

telefono estraneo, la password è sempre necessaria.

La rimozione di un numero si effettua con il comando *NUMx;pwd* dove al posto della x va scritto il numero della posizione che contiene il numero da cancellare; *pwd* è la solita password, necessaria anche quando il telefono è in lista. Per richiedere la lista dei numeri attualmente memorizzati nel telecontrollo occorre inviare l'SMS *NUM?;pwd*; questo comando vale con qualsiasi telefono. Vedia-

mo adesso i due comandi con cui si dice al telecontrollo quali numeri della lista devono, in caso di allarme (presenza di tensione) all'ingresso optoisolato, ricevere gli SMS o le chiamate (semplici squilli). Per definire i numeri cui inviare i messaggi, il comando è *SMSxxxxxxx:ON;pwd*, dove le x indicano le posizioni dei numeri telefonici memorizzati (da 1 a 8); ad esempio, *SMS1367:ON;pwd* significa che in caso di allarme il telecontrollo invierà messaggi ai nu- ➤



ELENCO COMPONENTI:

R1: 0,1 ohm 1W
R2: 2,2 kohm
R3: 1,2 kohm
R4+R6, R9, R11: 4,7 kohm
R7, R8: 330 ohm
R10: 10 kohm
R12, R14, R18, R19: *non montate*
R13: 1 kohm
R15: 4,7 kohm
R16: 10 kohm
R17: 330 ohm
C1, C4, C6, C7: 100 nF multistrato SMD
C2: 470 µF 35 VL elettrolitico
C3: 100 pF ceramico SMD
C5: 220 µF 16 VL elettrolitico
C8: 470 µF 6,3 VL tantalio SMD
C9, C10: 10 pF ceramico SMD
C11, C12: 100 µF 16 VL elettrolitico
C13: 100 nF multistrato SMD
C14+C16: 470 µF 6,3 VL tantalio SMD
C17, C18: (470 µF 6,3 VL tantalio SMD)
Q1: quarzo 20 MHz SMD
U1: MC34063
U2: PIC18F2620 (MF686)
U3: Modulo GSM GM862
U4: DS18B20
U5: TLP181
D1+D4: 1N4007
D5: 1N5819
T1, T2: BC817
LD1, LD3: led 3 mm verde
LD2: led 3 mm giallo

meri memorizzati nelle posizioni 1, 3, 6 e 7. Se al posto di ON fosse stato scritto "OFF" il comando avrebbe avuto l'effetto contrario.

Per impostazione predefinita, tutti i numeri nelle otto posizioni della lista (sempre che esistano) riceveranno SMS in caso di allarme.

Oltre agli SMS, il telecomando può anche effettuare brevi chiamate per far squillare il telefono delle persone che devono essere avvertite di un allarme all'ingresso optoisolato; la chiamata è solitamente più immediata dell'SMS, quindi usare la funzione può essere molto utile, senza contare che non costa nulla, perché non serve rispondere (il circuito non invia alcun segnale). Il comando

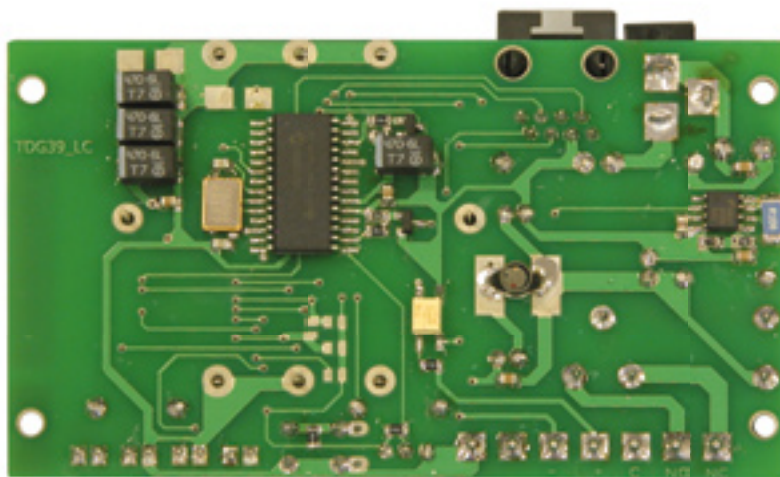
per configurare i numeri che ricevono chiamate vocali è **VOCxxxxxxx: ON;pwd**, al quale si applicano le stesse regole viste per il comando relativo agli SMS. Appurato come definire le azioni in caso di allarme all'ingresso optoisolato, va detto che il telecomando ci permette di stabilire, impostandolo con un apposito comando SMS, un eventuale tempo di inibizione (un intervallo durante il quale non viene considerato lo stato di IN1) per discriminare situazioni di disturbo da segnali veri e propri. L'intervallo si può scegliere tra 0 e 59 minuti primi (quello predefinito è 5 minuti). L'impostazione si effettua con il messaggio **INC:mm**, dove **mm** identifica il tempo in minuti.

Come accennato in precedenza, l'allarme può essere generato, oltre che dalla presenza di tensione all'ingresso optoisolato, anche quando la temperatura ambiente esce da una finestra preventivamente impostata tramite i comandi **ALLMIN:yxx** (soglia minima) e **ALLMAX:yxx** (soglia massima): in entrambi, al posto di y si deve mettere il segno + o - a seconda che la temperatura sia positiva o negativa e in xx si scrive il valore assoluto nel formato a due cifre, tenendo conto che è possibile impostare valori compresi tra -10 e 99 °C. Ad esempio i comandi **ALLMIN:-05** e **ALLMAX:+30** definiscono una finestra compresa tra -5 e +30 °C: il telecomando avvierà la sequenza di allarme per temperature in-

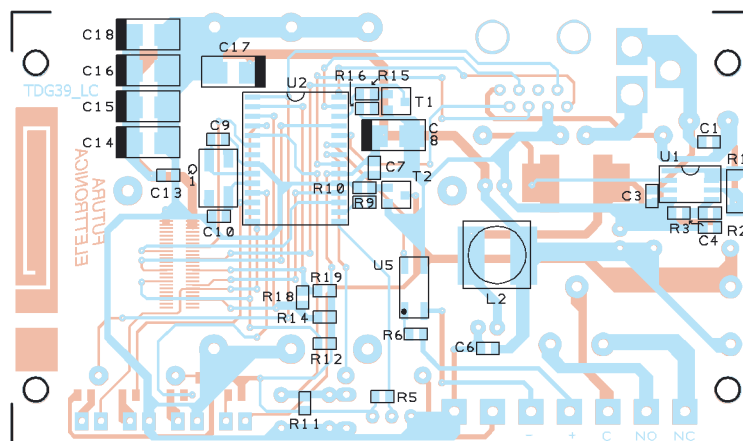
LD4: led 3 mm rosso
 L1: Bobina 22 μ H
 L2: Bobina 1 μ H
 P1: Microswitch
 RL1: Relé 3,6V 1 scambio

Varie:

- Morsettiera 2 poli (2 pz.)
- Morsettiera 3 poli
- Connettore 50 poli per modulo GSM
- plug alimentazione
- Connettore RJ45
- Circuito stampato codice S686



Il montaggio del termostato GSM è abbastanza impegnativo poiché vengono impiegati numerosi componenti molti dei quali a montaggio superficiale. Essi inoltre sono montati su entrambi i lati del PCB. È consigliabile iniziare il montaggio dal connettore del modulo GSM (il più critico); quindi si prosegue con i componenti SMD, situati sul lato inferiore del circuito. Si termina il montaggio con i componenti ad inserzione, sul lato superiore. Fate attenzione a non surriscaldare il connettore del telefono (che potrebbe deformarsi) e gli induttori (le cui spire potrebbero andare in corto).



feriori a -5°C e superiori a $+30^{\circ}\text{C}$. Si noti che, per impostazione predefinita (e comunque dopo un reset totale) non sono definiti valori e quindi l'allarme termico è disattivato; è possibile impostare anche un solo valore, ma in questo l'allarme scatta solo per il superamento di quella soglia. L'allarme termico può essere disinserito con il comando da **SMS ALL:OFF** ed eventualmente reinserito con **ALL:ON**. È possibile richiedere l'invio di un SMS contenente i valori delle temperature di allarme tramite il comando **ALL?**. Anche per l'allarme termico è stata prevista la possibilità di impostare un eventuale intervallo di inibizione (il valore predefinito è 5 minuti) utile quando con il circuito si deve

tenere sotto controllo un ambiente la cui escursione termica varia con lentezza; l'intervallo serve, ad esempio, quando l'utente, ricevendo l'allarme indicante che l'ambiente si è raffreddato eccessivamente, attiva forzosamente da remoto il riscaldamento. In tal caso, se non si imposta un intervallo di inibizione sufficiente a dare al riscaldamento il tempo di alzare la temperatura oltre la soglia di allarme, il telecontrollo continua a inviare SMS e telefonate. L'impostazione del caso si effettua inviando il comando **INT:mm**, nel quale al posto di mm va scritto il numero di minuti (tra 00 e 59) di inibizione dal rilevamento dell'allarme. Il telecontrollo dispone di altre fun-

zioni relative alle temperature: la prima di esse è la memorizzazione delle temperature più bassa e più alta registrate dal sensore Dallas dall'ultimo reset o dalla prima accensione; per richiedere da cellulare il contenuto di tale memoria eventi, si deve inviare al telecontrollo l'SMS **TEMP**. Il circuito risponde con un messaggio, diretto al numero del telefono che ha impartito il comando, contenente i valori minimo e massimo. È anche possibile ottenere le temperature senza inviare alcun SMS: basta chiamare, con un telefono il cui numero sia in lista, quindi riagganciare; l'unità invierà il solito messaggio. In ogni momento è possibile azzerare la memoria della temperatura con il comando **TRES**. Il ➤

comando con cui si definisce la temperatura ambiente che il telecomando deve mantenere quando funziona in modalità automatica, è *NOR:xx*, dove al posto delle *x* va inserito il valore desiderato (ad esempio 24) espresso in gradi centigradi.

Visto che abbiamo parlato di modalità di funzionamento, vediamo il comando da utilizzare per impostarla da remoto, via cellulare: è un SMS contenente il testo *FUN:x*, in cui *x* può assumere i seguenti valori:

- T, modalità asservita (RL1 segue il termostato esterno)
- A, modo automatico
- M, il relé è forzato attivo.

Se non ci si ricorda in che modalità sta lavorando il telecomando, è possibile chiederglielo inviando un SMS contenente il testo *FUN?* a seguito del quale si ottiene un SMS di risposta contenente la modalità attiva. La modalità vigente viene memorizzata e quindi eventuali black-out non la alterano. Passiamo adesso ai comandi riguardanti l'attività del relé, con la premessa che RL1 può essere gestito indipendentemente dalla modalità di funzionamento scelta; in particolare, in ogni momento è possibile attivarlo (se a riposo) o disaccenderlo (se è attivo) con i comandi *OUT:ON* e *OUT:OFF*. È possibile richiedere in qualsiasi momento lo stato del relé tramite il comando *OUT?*, al quale il telecomando risponde inviando un SMS. Il telecomando può memorizzare lo stato del relé e ripristinarlo dopo un eventuale black-out. Tale funzio-

ne (disabilitata di default) può essere attivata con i comandi *RIP1* e *RIP0*. Al comando *RIP?*, il telecomando restituisce un SMS indicante l'attuale condizione del ripristino espressa in forma numerica, ossia 1 se il ripristino è attivo e 0 se, invece, non lo è.

Bene, giunti a questo punto parliamo dell'ultima serie di comandi: quella che permette di definire i messaggi di notifica inviati dal telecomando. Il nostro sistema prevede l'invio di un SMS personalizzabile ogni volta che viene alimentato; per impostazione predefinita (e dopo ogni reset totale) la funzione è disattivata ma può essere controllata tramite il comando *AVVx* in cui *x* vale 1 se la si vuole attivare e 0 se la si vuole disattivare. Nel caso la funzione sia attiva, il testo del messaggio inviato all'accensione (*SYSTEM STARTUP* è il messaggio predefinito) si imposta con il comando *TSU:xxxxxxxxxxxx*, dove al posto delle *x* va scritto il testo, restando nel limite di 100 caratteri (compresi gli spazi). Il messaggio di notifica dell'allarme sull'ingresso optoisolato viene modificato dal comando *TIN1A:xxxxxxxxxxxx*, che segue le stesse regole viste per il comando *TSU*. Quanto ai messaggi in caso di allarme termico, è possibile definire individualmente quelli derivanti dal superamento della soglia superiore e dall'abbassamento della temperatura oltre la soglia inferiore con i comandi *THI:xxxxxxxx* e *TLO:xxxxxxxx* per quello dovuto alla soglia minima. In ogni momento l'utente in possesso della password può ripristi-

nare le impostazioni predefinite del sistema e cancellare tutti i numeri telefonici memorizzati tramite il comando *RES;pwd*, nel quale *pwd* è la password corrente.

La realizzazione

Siamo finalmente arrivati alle note costruttive; una volta procurata la basetta, inserite e saldate i componenti usando un saldatore a punta sottile (per integrati) da non più di 30 watt di potenza e del flussante (per i componenti a montaggio superficiale): disponete i componenti iniziando con il connettore miniatura per il modulo Telit, che è quello più critico, in quanto i suoi piedini sono a passo molto ridotto. Per il montaggio consigliamo di posizionarlo ben centrato rispetto alle proprie piazzole, quindi di stagnare due piedini esterni per lato, procedendo un pin per lato alla volta, così da non surriscaldare la struttura. Sistemate poi gli altri elementi, usando, per quelli SMD, la tecnica descritta per il connettore.

Per le connessioni con l'esterno, inserite e stagnate nelle piazzole siglate PROG un connettore femmina RJ45 da circuito stampato, poi una presa plug da c.s. che servirà all'alimentazione; quanto allo scambio del relé e ai due ingressi, usate delle morsettiere da stampato a passo 5 mm.

Finite le saldature, inserite il modulo cellulare a fondo e stagnatene le alette di ancoraggio; diversamente, se urtate la basetta o la rovesciate, il componente si staccherà e comunque non riuscirà a rimanere correttamente

per il

MATERIALE

Il progetto del termostato GSM descritto in questo articolo è disponibile in scatola di montaggio (cod. TDG39) al prezzo di 198,00 Euro. Il kit include tutti i componenti, il microcontrollore già programmato, il sensore di temperatura ed il modulo GSM (non è compresa la SIM telefonica). Il circuito è disponibile anche già montato e collaudato allo stesso prezzo della scatola di montaggio. I prezzi si intendono IVA inclusa.

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

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

Per il nostro telecontrollo abbiamo previsto un software che gira in ambiente Microsoft Windows dal cui pannello di controllo è possibile dialogare con il dispositivo per modificare facilmente i parametri e le modalità di funzionamento, ma anche e soprattutto per effettuare la configurazione iniziale prima di andare a installarlo nel locale in cui dovrà operare. L'utilizzo del PC permette di risparmiare tempo e denaro richiesti dall'esecuzione della configurazione completa via SMS, la quale, pur essendo effettuata con messaggi pluricomando, sarebbe lenta e costosa. Inoltre, torna utile anche quando si vogliono verificare tutte le impostazioni senza, per questo, richiederle da SMS.

Una volta installato e avviato, il programma presenta una finestra di dialogo che può essere suddivisa nelle seguenti cinque sezioni:

- **Informazioni porta seriale:** mostra la porta seriale in uso e le relative impostazioni (parità e velocità di connessione); quando il dispositivo si interfaccia mediante adattatore USB, il campo "Porta COM" indica la COM virtuale assegnata all'USB; facendo clic nelle caselle corrispondenti, è possibile variare le impostazioni (in Porta COM si apre un menu a tendina che riepiloga le porte disponibili).
- **Informazioni:** riepiloga la modalità impostata, l'eventuale presenza di allarmi e la temperatura che il circuito deve mantenere; se la modalità non è Automatica, la casella Soglia Temp. non è attiva.
- **Temperature Registrate:** mostra la massima e la minima temperatura registrata dalla sonda fino al momento della connessione tra PC e telecontrollo, oltre alla temperatura ambiente del momento.
- **Temperatura di allarme:** riepiloga le soglie massima e minima attualmente impostate.
- **Configurazioni:** raggruppa tutte le impostazioni che definiscono la configurazione del circuito, tra le quali la password, il modo di ripristino, gli intervalli di inibizione degli allarmi termico e dovuto all'ingresso optoisolato e via di seguito; per modificare le impostazioni basta selezionare e scrivere nelle relative caselle quelle nuove, ovvero accedere ai menu a tendina (caselle Ripristino relé, Modalità, SMS Startup, Inibizione IN1, Inibizione IN2).



Facendo clic sul pulsante Azzera Griglia, si effettua il reset delle informazioni (record temperature) e impostazioni (modalità, soglie ecc.) memorizzate nel circuito e occorre, perciò, provvedere a una nuova configurazione; l'azione corrisponde ad inviare via SMS al telecontrollo un comando di reset totale. Per effettuare la connessione al computer e lavorare con il programma qui descritto, il nostro apparato richiede un'interfaccia, che può essere la FT475 se si intende connettersi da seriale RS232 o la FT635 quando si desidera usare l'USB; entrambe le unità sono prodotte dalla ditta Futura Elettronica (www.futuranet.it) alla quale possono essere richieste.

La FT475 è una semplice interfaccia TTL/RS232 realizzata con un converter MAX232; chi volesse vederne lo schema, sappia che è stata pubblicata nel fascicolo n° 75 di Elettronica In. Quanto alla FT635, è un po' più complessa e impiega un convertitore TTL/USB implementato dall'integrato FT232BM della FTDI; lo schema relativo è stato descritto nel fascicolo n° 117.

inserito nel proprio connettore. Prima di inserire la SIM nel GM862, con un comune cellulare disattivatene il PIN, perché il software del PIC da noi usato per gestire il circuito non prevede il confronto del codice d'accesso alla card. Inserendo una SIM con impostato il PIN code il micro non riuscirebbe ad utilizzarla e il telecontrollo non funzionerebbe. Concludiamo con l'antenna: siccome è costituita da una

pista dello stampato opportunamente dimensionata, bisogna collegarla al modulo con un cavetto coassiale (terminante su un connettore MMCX) l'anima del quale va stagnata sull'esterno della pista.

Potete anche usare un'antenna esterna: basta montare un connettore sul c.s. e collegare ad esso il coassiale che giunge dal modulo, ovvero connettere l'antenna direttamente al mo-

dulo mediante un cavetto provvisto di connettore FME.

Qualora il dispositivo venga inserito in un contenitore, quest'ultimo dovrà essere necessariamente plastico, onde evitare l'attenuazione del segnale radio; utilizzando un contenitore metallico dovrete impiegare un cavo adattatore d'antenna con presa FME, in modo da poter usare un'antenna esterna.

Oscilloscopio con PC, DLL e esempi software

di Alessandro Sottocornola

Impariamo ad utilizzare la libreria PCS100D.DLL per creare applicazioni software personalizzate. Analizziamo anche due listati completi, uno in Delphi e l'altro in Visual Basic, che sfruttano tale DLL per realizzare una semplice interfaccia grafica per il nostro oscilloscopio digitale. Ultima puntata.



Nei due precedenti numeri della rivista abbiamo presentato l'hardware e l'interfaccia grafica di un dispositivo che permette di trasformare il PC in un ottimo ed economico oscilloscopio.

Avevamo anticipato che questo circuito dalle prestazioni molto interessanti può essere gestito sia tramite un software standard (disponibile gratuitamente), sia tramite applicazioni personalizzate la cui realizzazione è resa molto semplice dalla disponibilità di un'apposita DLL, anch'essa scaricabile gratuitamente.

Nel corso di quest'ultimo articolo dedicato all'oscil-

loscopio per PC, vediamo in dettaglio tutte le istruzioni (procedure e funzioni) messe a disposizione dell'utente dalla libreria, il cui nome è PCS100D.DLL. In particolare, di ogni procedura e funzione, analizziamo la sintassi, i relativi parametri e lo scopo per cui viene utilizzata, avvalendoci ogni volta di un semplice esempio di codice sorgente. In questo stesso articolo presentiamo anche il listato di due applicativi completi, uno in Visual Basic ed uno in Delphi, che potete utilizzare come punto di partenza per iniziare a sviluppare le vostre applicazioni software. ➤

Tabella 1

Istruzioni di carattere generale	
Start_PCS100 e Stop_PCS100	Apri e chiude il canale di comunicazione parallelo con il dispositivo
RunOn(Run:Boolean)	Avvia la modalità di acquisizione continua del segnale in ingresso
SingleOn(Single:Boolean)	Esegue una sola lettura del segnale in ingresso
SelectPrinterPort(Printer_no: Longint)	Seleziona la porta parallela
Show_PCS100(Visible: Boolean)	Visualizza o nasconde l'interfaccia grafica di configurazione
Istruzioni di configurazione	
Voltage(Volts:Longint):	Seleziona la scala di lettura (in Volt/Divisione)
SampleRate(Rate:Longint):	Programma la base dei tempi
TriggerLevel(TrgLevel:Longint):	Definisce il livello della tensione di trigger ossia il valore del segnale di ingresso che avvia l'acquisizione
TrgEdge(Positive_Negative:Logint);	Seleziona il fronte del trigger (salita o discesa)
TrgOn(trg_on: Boolean);:	Attiva e disattiva il trigger
YPosition(y_pos:Longint);	Serve per il posizionamento della traccia sull'asse verticale (Y)
Coupling(AC_DC_GND:Logint):	Seleziona il tipo di accoppiamento del segnale in ingresso (continua, alternata o massa).
Istruzioni di lettura	
ReadData(Buffer: Pointer; Count; Longint):	Legge i dati immagazzinati dall'hardware dell'oscilloscopio.

La libreria PCS100D.DLL

Per poter realizzare un'applicazione Windows con la quale gestire l'hardware dell'oscilloscopio, è necessario che il progetto del software includa la libreria PCS100D.DLL. Essa è stata sviluppata appositamente per poter controllare i modelli PCS100 e K8031 tramite la porta parallela del computer, e fornisce un set di Procedure e Funzioni completo e facile da utilizzare. Ciò non significa che non è necessario sviluppare del software, bensì che lo sviluppo è limitato alla sola interfaccia grafica, la quale deve interagire in maniera corretta con le varie chiamate alle funzioni della libreria.

Una caratteristica molto importante è rappresentata dalla compatibilità con quattro sistemi operativi Microsoft differenti e precisamente Windows 98Se, 2000, ME e XP: in questo modo qualsiasi applicazione può essere facilmente trasportata su computer facenti uso di differenti sistemi operativi (purché siano di casa Microsoft) a prescindere dalla libreria, che non cambia.

Le istruzioni (Procedure e Funzioni) gestibili tramite la DLL in oggetto possono essere classificate in tre macrogruppi (riportati in tabella

1): istruzioni di carattere generale, istruzioni di configurazione ed istruzioni di lettura.

Vediamo ora ognuna di esse in dettaglio, tenendo presente che sia le dichiarazioni che gli esempi sono stati scritti in Delphi

Start_PCS100

È la procedura più importante poiché apre il canale di comunicazione con la porta parallela a cui il circuito è collegato. Essa deve essere eseguita ogni volta che ci si appresta a comunicare con l'interfaccia. Non necessita di alcun parametro di configurazione.

Sintassi:

```
PROCEDURE Start_PCS100
```

Parametri:

Nessuno

Esempio:

```
procedure
  TForm1.FormCreate(Sender: TObject);
begin
  Start_PCS100;
end;
```

Stop_PCS100

Svolge la funzione opposta a quella appena descritta: essa infatti scarica la routine di comunicazione per il

dispositivo. È l'ultima azione compiuta dal programma applicativo e, come la precedente, non necessita di alcun parametro.

Sintassi:

```
PROCEDURE Stop_PCS100
```

Parametri:

Nessuno

Esempio:

```
procedure
  TForm1.FormCloseQuery
  (Sender: TObject; var CanClose: Boolean);
begin
  Stop_PCS100;
end;
```

Show_PCS100

Serve per visualizzare o nascondere l'interfaccia utente (che contiene la traccia e i comandi) dell'oscilloscopio, ossia la simulazione grafica del pannello di controllo dell'oscilloscopio. Come parametro, necessita la dichiarazione della variabile booleana "visible" ("true" o "false").

Se visible=true allora l'interfaccia grafica viene visualizzata, altrimenti viene nascosta.

L'esempio riportato di seguito mostra la dichiarazione di procedura che visualizza l'interfaccia utente.

Sintassi:

```
procedure Show_PCS100(Visible: Boolean);
```

Parametri:

Run: Boolean, se vero visualizza l'interfaccia; se falso la nasconde.

Esempio:

```
procedure
  TForm1.ShowPCS100Click
  (Sender: TObject);
begin
  Show_PCS100(true);
end;
procedure
  TForm1.HidePCS100Click
  (Sender: TObject);
begin
  Show_PCS100(false);
end;
```

RunOn

Come avviene negli oscilloscopi digitali a memoria, anche col nostro circuito è possibile avviare l'acquisizione continua di una traccia, eseguire una sola acquisizione oppure sospendere la misurazione, congelando la forma d'onda sullo schermo. Questo comando avvia o ferma l'esecuzione (Run mode) del programma corrispondente alla visualizzazione della traccia sull'interfaccia utente. Necessita della dichiarazione del parametro booleano "Run". Se nell'esempio qui di seguito si dichiarasse RunOn(Run=false) l'oscilloscopio non effettuerebbe alcuna acquisizione del segnale presente all'ingresso.

Sintassi:

```
PROCEDURE RunOn(Run: Boolean);
```

Parametro:

Run: Boolean, se vero avvia l'oscilloscopio; se falso ferma l'esecuzione dell'interfaccia.

Esempio:

```
procedure
  TForm1.RunClick
  (Sender: TObject);
begin
  RunOn(Run.down);
end;
```

SingleOn

Questa procedura è molto simile alla RunOn però permette di acquisire un segnale per una sola volta, ossia avvia l'acquisizione e, quando la traccia disegnata sul monitor ha raggiunto il limite destro dell'area di visualizzazione, l'acquisizione si ferma. È molto utile quando si deve effettuare un'analisi visiva su un segnale acquisito. Necessita della dichiarazione del parametro "Single", booleano anch'esso, e che pertanto può valere "true" o "false".

Sintassi:

```
PROCEDURE SingleOn(Single: Boolean);
```

Parametro:

Single: Se vero imposta la modalità Single; se falso arresta Single.

Esempio:

```
procedure
  TForm1.SingleClick
  (Sender: TObject);
begin
  SingleOn(true);
end;
```

SelectPrinterPort

Come il nome lascia intuire, serve per comunicare al programma ed alla libreria l'indirizzo della porta parallela a cui l'oscilloscopio è collegato. Necessita del parametro Printer_no, descritto da una variabile di tipo "Long Integer", che può assumere valore da 0 a 3 (0 corrisponde alla modalità dimostrativa).

Sintassi:

```
PROCEDURE
  SelectPrinterPort(Printer_no: Longint);
```

Parametro:

Printer_no: Indirizzo della porta (LPT) stampante cui è collegato l'oscilloscopio.
 0 = Demo mode
 1 = 378 hex
 2 = 278 hex
 3 = 3BC hex

Esempio:

```
procedure
  TForm1.Select_LPTClick
```

```
(Sender: TObject);
```

```
begin
```

```
  SelectPrinterPort
```

```
((sender as TRadioButton).tag);
```

```
end;
```

Voltage

Consente di impostare la sensibilità dell'ingresso ossia imposta i volt per divisione (V/div) sulla griglia di visualizzazione della traccia. Necessita del parametro "Volts" di tipo "Long Integer", che può assumere valore compreso tra 0 e 5.

Sintassi:

```
PROCEDURE Voltage(Volts: Longint);
```

Parametro:

Volts: indica il campo dei V/div
 0 = 3 V/div
 1 = 1 V/div
 2 = 0.3 V/div
 3 = 0.1 V/div

**Requisiti di sistema**

Il nostro oscilloscopio può funzionare praticamente su qualsiasi Personal Computer, anche se datato; basta che abbia un processore Intel Pentium (o AMD K6, Cyrix 6x86 ecc.) o superiore, disponga di mouse e monitor SVGA (risoluzione di 800x600) una porta parallela e un lettore CD-ROM; se il PC è molto recente, è possibile che non disponga della porta parallela (ciò si può verificare con i notebook) quindi bisogna procurarsi una di quelle interfacce USB/parallela facilmente reperibili nei negozi di materiale informatico, ad un prezzo abbastanza contenuto. In questo caso, è necessario che il driver dell'adattatore sia in grado di emulare virtualmente una porta parallela: alcuni convertitori, infatti, supportano solamente il servizio di "porta periferica di stampa" e non permettono al PC di vedere l'indirizzo della porta, indispensabile per il funzionamento della DLL. Sul piano della compatibilità software, è sufficiente un sistema operativo Microsoft Windows 95, 98, Me, NT, 2000 o XP.

4 = 30 mV/div

5 = 10 mV/div

Esempio:

procedure

TForm1.VoltageRangeClick

(Sender: TObject);

begin

Voltage((sender as TSpeedButton).tag);

end;

SampleRate

Imposta la base dei tempi dell'oscilloscopio: in questo caso agisce sulla frequenza di campionamento del convertitore analogico digitale del nostro circuito. Necessita della dichiarazione del parametro Long Integer "Rate", che può assumere valore da 0 a 6.

Sintassi:

PROCEDURE SampleRate(Rate:Longint);

Parametro:

Rate: Indica la frequenza di campionamento

0 = 800 Samples/sec

1 = 8000 Samples/sec

2 = 80 k Samples/sec

3 = 800 k Samples/sec

4 = 8 M Samples/sec

5 = 16 M Samples/sec

6 = 32 M Samples/sec

Esempio:

procedure

TForm1.TimeRangeClick

(Sender: TObject);

begin

SampleRate((sender as TSpeedButton).tag);

end;

Coupling

Imposta l'accoppiamento dell'ingresso dell'oscilloscopio (accoppiamento in continua, in alternata o a massa). Il parametro AC_DC_GND (Long Integer) assume valore 0, 1 o 2.

Sintassi:

PROCEDURE Coupling(AC_DC_GND:Longint);

Parametro:

AC_DC_GND: indica il tipo di accoppiamento.

0 = AC

1 = DC

2 = GND

Usare la PCS100D.DLL in Delphi

In questo esempio applicativo, scritto per Delphi, si trovano tre informazioni molto importanti: all'inizio vedete come i controlli (pulsanti, radio-button, combo-box...) devono essere dichiarati; quindi, a seguire, potete vedere come devono essere effettuate le dichiarazioni di procedure e funzioni pertinenti alla PCS100D.DLL (parte centrale del listato) e, nell'ultima parte del programma, sono riportati una serie di esempi di come procedure e funzioni devono essere chiamate durante l'esecuzione del programma, nonché di come esse interagiscono con vari controlli. Le figure 1a e 1b mostrano la finestra di acquisizione dei dati in formato grafico (1a) ed in formato testo (1b). La libreria contiene il progetto completo comprensivo dei due form (ossia le due finestre) già impaginati. Dovete semplicemente scaricare la libreria compressa, decomprimerla rispettando la gerarchia delle cartelle quando è stata compressa, quindi eseguire il programma eseguibile nella cartella PCS100_DLL_Delphi.

```
{ Definizione controlli e variabili di sistema }
unit PCS_Runx;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, ComCtrls;
type
  TForm1 = class(TForm)
  VoltageRange: TSpeedButton;
  SpeedButton2: TSpeedButton;
  SpeedButton3: TSpeedButton;
  SpeedButton4: TSpeedButton;
  SpeedButton5: TSpeedButton;
  SpeedButton6: TSpeedButton;
  GroupBox1: TGroupBox;
  Select_LPT: TRadioButton;
  RadioButton2: TRadioButton;
  RadioButton3: TRadioButton;
  RadioButton4: TRadioButton;
  TimeRange: TSpeedButton;
  SpeedButton8: TSpeedButton;
  SpeedButton9: TSpeedButton;
  SpeedButton10: TSpeedButton;
  SpeedButton11: TSpeedButton;
  SpeedButton12: TSpeedButton;
  SpeedButton13: TSpeedButton;
  Memo1: TMemo;
  Run: TSpeedButton;
  Single: TButton;
  ShowPCS100: TButton;
  HidePCS100: TButton;
  TrgLevel: TTrackBar;
  TriggerOn: TSpeedButton;
  TriggerOff: TSpeedButton;
  Label1: TLabel;
  Position: TTrackBar;
  Label3: TLabel;
  SpeedButton17: TSpeedButton;
  TriggerEdge: TSpeedButton;
  Label4: TLabel;
  SelectCoupling: TSpeedButton;
  SpeedButton20: TSpeedButton;
  SpeedButton21: TSpeedButton;
  DataRead: TButton;
  { Dichiarazione procedure di programma }
  procedure VoltageRangeClick(Sender: TObject);
  procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
  procedure FormCreate(Sender: TObject);
  procedure Select_LPTClick(Sender: TObject);
  procedure TimeRangeClick(Sender: TObject);
  procedure DataReadClick(Sender: TObject);
  procedure RunClick(Sender: TObject);
  procedure SingleClick(Sender: TObject);
  procedure ShowPCS100Click(Sender: TObject);
  procedure HidePCS100Click(Sender: TObject);
  procedure TrgLevelChange(Sender: TObject);
  procedure PositionChange(Sender: TObject);
  procedure TriggerOnClick(Sender: TObject);
  procedure TriggerOffClick(Sender: TObject);
  procedure TriggerEdgeClick(Sender: TObject);
  procedure SelectCouplingClick(Sender: TObject);
  private
  { Dichiarazioni "Private" }
  public
  { Dichiarazioni "Public" }
  end;
var
  Form1: TForm1;
  data: array[0..4096] of longint;
```

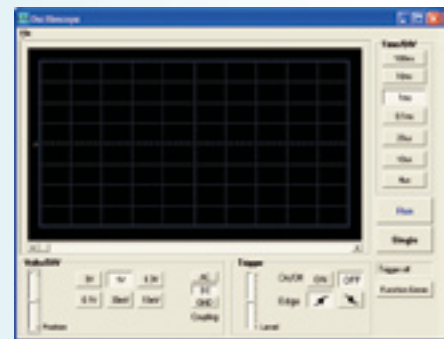


Fig. 1a



Fig. 1b

continuazione del listato in Delphi

```
s:string;
implementation
{$R *.DFM}
{Dichiarazione procedure e funzioni appartenenti alla DLL PCS100D.DLL}
PROCEDURE SelectPrinterPort(Printer_no: Longint); stdcall; external 'PCS100D.dll';
PROCEDURE RunOn(Run: Boolean); stdcall; external 'PCS100D.dll';
PROCEDURE SingleOn(Single: Boolean); stdcall; external 'PCS100D.dll';
PROCEDURE Voltage(Volts: Longint); stdcall; external 'PCS100D.dll';
PROCEDURE SampleRate(Rate: Longint); stdcall; external 'PCS100D.dll';
PROCEDURE TriggerLevel(TrgLevel: Longint); stdcall; external 'PCS100D.dll';
PROCEDURE TrgEdge(Positive_Negative: Longint); stdcall; external 'PCS100D.dll';
PROCEDURE Coupling(AC_DC_GND: Longint); stdcall; external 'PCS100D.dll';
PROCEDURE YPosition(y_pos: Longint); stdcall; external 'PCS100D.dll';
PROCEDURE TrgOn(trg_on: Boolean); stdcall; external 'PCS100D.dll';
PROCEDURE Start_PCS100; stdcall; external 'PCS100D.dll';
PROCEDURE Stop_PCS100; stdcall; external 'PCS100D.dll';
procedure Show_PCS100(Visible: Boolean); stdcall; external 'PCS100D.dll';
function ReadData(Buffer: Pointer; Count: Longint): Boolean; stdcall; external 'PCS100D.dll';
{Chiamate alle procedure e funzioni appartenenti alla DLL PCS100D.DLL}
procedure TForm1.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
begin
  Stop_PCS100; end;
procedure TForm1.FormCreate(Sender: TObject);
begin
  Start_PCS100;
end;
procedure TForm1.VoltageRangeClick(Sender: TObject);
begin
  Voltage((sender as TSpeedButton).tag);
end;
procedure TForm1.Select_LPTClick(Sender: TObject);
begin
  SelectPrinterPort((sender as TRadioButton).tag);
end;
procedure TForm1.TimeRangeClick(Sender: TObject);
begin
  SampleRate((sender as TSpeedButton).tag);
end;
procedure TForm1.DataReadClick(Sender: TObject);
var i: longint;
p:pointer;
begin
  p:= @data[0];
  if ReadData(p, 100) then
  begin
    memol.clear;
    for i:=0 to 99 do
      memol.lines.add(inttostr(data[i]));
    end;
  end;
  procedure TForm1.RunClick(Sender: TObject);
  begin
    RunOn(Run.down);
  end;
  procedure TForm1.SingleClick(Sender: TObject);
  begin
    SingleOn(true);
  end;
  procedure TForm1.ShowPCS100Click(Sender: TObject);
  begin
    Show_PCS100(true);
  end;
  procedure TForm1.HidePCS100Click(Sender: TObject);
  begin
    Show_PCS100(false);
  end;
  procedure TForm1.TrigLevelChange(Sender: TObject);
  begin
    TriggerLevel(TrgLevel.position);
  end;
  procedure TForm1.PositionChange(Sender: TObject);
  begin
    YPosition(Position.position);
  end;
  procedure TForm1.TriggerOnClick(Sender: TObject);
  begin
    TrgOn(true);
  end;
  procedure TForm1.TriggerOffClick(Sender: TObject);
  begin
    TrgOn(false);
  end;
  procedure TForm1.TriggerEdgeClick(Sender: TObject);
  begin
    TrgEdge((sender as TSpeedButton).tag);
  end;
  procedure TForm1.SelectCouplingClick(Sender: TObject);
  begin
    Coupling((sender as TSpeedButton).tag);
  end;
end.
```

Esempio:

```
procedure TForm1.SelectCouplingClick(Sender: TObject);
begin
  Coupling((sender as TSpeedButton).tag);
end;
```

TriggerLevel

Questa procedura imposta il livello di trigger ossia il livello al quale l'oscilloscopio inizia la visualizzazione del segnale in ingresso: ciò fa sì che ogni scansione dell'onda inizi sempre dallo stesso punto, rimanendo pertanto ferma sul display. Il livello varia a passi di mezza divisione (mezzo quadretto) dello schermo virtuale e viene definito dal parametro Long Integer "TrgLevel", il cui valore può essere compreso tra 1 e 15.

Sintassi:

PROCEDURE TriggerLevel(TrgLevel: Longint);

Parametro:

TrgLevel: Valore di trigger (compreso fra 1 e 15).

Esempio:

```
procedure
  TForm1.TrigLevelChange
    (Sender: TObject);
begin
  TriggerLevel(TrgLevel.position);
end;
```

YPosition

Aggiunge un offset al segnale d'ingresso ossia trasla verticalmente la traccia. Svolge la stessa funzione della manopola "Posizione Y" presente negli oscilloscopi tradizionali. Necessita del parametro "Position" che può assumere valore da -128 a +127.

Sintassi:

PROCEDURE YPosition(Position: position);

Parametro:

Position: Posizione verticale (Y-position) della traccia, rispetto a massa; valore tra -128 e 127.

Esempio:

```
procedure
```



Usare la PCS100D.DLL in Visual Basic

In questo esempio applicativo, scritto per gli utenti Visual Basic, si trovano le dichiarazioni di procedure e funzioni della PCS100D.DLL ed un esempio di come effettuare le chiamate ad esse. Per poter utilizzare il listato bisogna prima assicurarsi che i seguenti file siano stati copiati nella cartella System32 di Windows:

PCS100D.DLL, PCS100E.EXE, DLPORTIO.DLL
DLPORTIO.SYS, FASTTime32.DLL

Il listato in Visual Basic, riportato qui di seguito, è più compatto rispetto all'esempio in Delphi pur producendo esattamente lo stesso risultato, come visibile dalle figure 2a (acquisizione dati con rappresentazione grafica) e 2b (acquisizione dati in modo testo). Anche in questo caso le funzioni e procedure fornite dalla libreria sono state dichiarate all'inizio del programma.

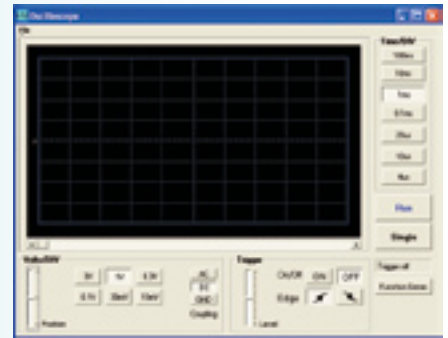


Fig. 2a

```
' Dichiarazione di variabili, procedure e funzioni pertinenti alla libreria PCS100D.DLL
Option Explicit
Dim s As String
Dim data(0 To 4096) As Long
Private Declare Sub SelectPrinterPort Lib "PCS100D.dll" (ByVal Printer_no As Long)
Private Declare Sub RunOn Lib "PCS100D.dll" (ByVal Run_On As Long)
Private Declare Sub SingleOn Lib "PCS100D.dll" (ByVal Single_On As Long)
Private Declare Sub Voltage Lib "PCS100D.dll" (ByVal Volts As Long)
Private Declare Sub SampleRate Lib "PCS100D.dll" (ByVal Rate As Long)
Private Declare Sub TriggerLevel Lib "PCS100D.dll" (ByVal TrgLevel As Long)
Private Declare Sub TrgEdge Lib "PCS100D.dll" (ByVal Positive_Negative As Long)
Private Declare Sub Coupling Lib "PCS100D.dll" (ByVal AC_DC_GND As Long)
Private Declare Sub YPosition Lib "PCS100D.dll" (ByVal y_pos As Long)
Private Declare Sub TrgOn Lib "PCS100D.dll" (ByVal trg_On As Long)
Private Declare Sub Start_PCS100 Lib "PCS100D.dll" ()
Private Declare Sub Stop_PCS100 Lib "PCS100D.dll" ()
Private Declare Sub Show_PCS100 Lib "PCS100D.dll" (ByVal Visible As Long)
Private Declare Function ReadData Lib "PCS100D.dll" (Buffer As Long, ByVal Count As Long) As Boolean
```

' Listato per delle subroutine associate ai vari controlli.
' E necessario creare due form contenenti i controlli necessari,
' nominati esattamente come definito nelle subroutine seguenti.

```
Private Sub Form_Load()
    Start_PCS100
End Sub
Private Sub Form_Terminate()
    Stop_PCS100
End Sub
Private Sub Hide_Click()
    Show_PCS100 0
End Sub
Private Sub Option1_Click(Index As Integer)
    SelectPrinterPort (Index)
End Sub
Private Sub Option2_Click(Index As Integer)
    TrgOn Index
End Sub
Private Sub Option3_Click(Index As Integer)
    TrgEdge Index
End Sub
Private Sub Option4_Click(Index As Integer)
    Coupling Index
End Sub
Private Sub Option5_Click(Index As Integer)
    SampleRate Index
End Sub
Private Sub Read_data_Click()
    Dim i As Long
    List1.Clear
    If ReadData(data(0), 50) = 1 Then
        For i = 0 To 49
            List1.AddItem data(i)
        Next
    Else: List1.AddItem "No new data"
    End If
End Sub
Private Sub Run_On_Click()
    RunOn Run_on.Value
End Sub
Private Sub Show_Click()
    Show_PCS100 1
End Sub
Private Sub Single_On_Click()
    SingleOn Single_on.Value
End Sub
Private Sub Voltage_Select_Click(Index As Integer)
    Voltage Index
End Sub
Private Sub VScroll1_Change()
    TriggerLevel VScroll1.Value
End Sub
Private Sub VScroll2_Change()
    YPosition (VScroll2.Value)
End Sub
```

' Apertura finestra principale con avvio della DLL
' Chiusura finestra principale. LA DLL viene scaricata
' Preme pulsante e nasconde finestra di visualizzazione
' Seleziona porta parallela su "radio button"
' Attiva/disattiva trigger su "radio button"
' Seleziona fronte trigger su "radio button"
' Seleziona accoppiamento su "radio button"
' Seleziona base tempi su "radio button"
' Acquisisce i dati tramite la parallela
' Avvia acquisizione continua dei dati
' Preme pulsante e mostra finestra di visualizzazione
' Avvia acquisizione di una singola serie di dati
' Seleziona i Volt/divisione
' Modifica il livello del trigger
' Cambia la posizione verticale della traccia visualizzata



Fig. 2b

```
TForm1.SetPositionChange
(Sender: TObject);
begin
  YPosition(SetPosition.position);
end;
```

TrgOn

Con questa procedura si attiva o disattiva il controllo del trigger. Necessita del parametro booleano “trg_on” che può assumere valore “vero” o “falso”.

Sintassi:

```
PROCEDURE TrgOn(trg_on: Boolean);
```

Parametro:

trg_on: Se vero attiva il trigger; se falso disattiva il trigger.

Esempio:

```
procedure
  TForm1.TriggerOnClick
  (Sender: TObject);
begin
  TrgOn(true);
end;
procedure
  TForm1.TriggerOffClick
  (Sender: TObject);
begin
  TrgOn(false);
end;
```

TrgEdge

Quando il trigger è attivo, esso può intervenire sul fronte di salita o sul fronte di discesa del segnale.

La procedura in oggetto, completata dal parametro “Positive_Negative”, permette di selezionare il fronte di aggancio.

Sintassi:

```
PROCEDURE TrgEdge(Positive_Negative:
Longint)
```

Parametro:

Positive_Negative: Indica il fronte di aggancio del trigger.

0 = Negative

1 = Positive

Esempio:

```
procedure TForm1.
  TriggerEdgeClick(Sender: TObject);
begin
  TrgEdge((sender as TSpeedButton).tag);
end;
```

ReadData

Questa funzione permette al programma di acquisire i dati relativi alla forma d’onda letta dall’interfaccia hardware.

Essa necessita di due parametri: “buffer” che punta alla memoria hardware dove sono contenuti i dati campionati e “Count”, che specifica quanti valori memorizzati devono essere letti.

Al termine dell’esecuzione di tale funzione, la libreria restituisce anche una risposta booleana attraverso il parametro “ReadData” che, se vero, conferma che sono presenti dati nuovi da leggere, mentre se falso, comunica che i dati in memoria non sono cambiati rispetto alla lettura precedente.

Sintassi:

```
FUNCTION ReadData(Buffer: Pointer;
Count: Longint): Boolean;
```

Parametri:

Buffer: Puntatore ad un array (32bit) dal quale dovranno essere letti i dati.

Count: Numero di dati corrispondenti ai valori, da leggere.

Risultato:

Boolean: TRUE aggiorna i nuovi dati letti dall’oscilloscopio; FALSE indica che non c’è stata una nuova lettura.

Esempio:

```
procedure TForm1.
  DataReadClick(Sender: TObject);
var i: longint;
data: array[0..4096] of longint;
p:pointer;
begin
  p:= @data[0];
  if ReadData(p, 100) then
    begin
      memo1.clear;
      for i:=0 to 99 do
        memo1.lines.add(inttostr(data[i]));
    end;
end;
```

Abbiamo così concluso la spiegazione di tutte le funzioni e procedure fornite dalla libreria PCS100D.DLL

Conclusione

Giunti a questo punto, non esistono ulteriori segreti da svelare: sapete come funziona l’hardware, come si controlla tramite il PC e come utilizzare l’hardware insieme ad un programma di vostra realizzazione. Il nostro compito finisce qui: non ci rimane che suggerirvi di dare ampio spazio alla fantasia per realizzare la vostra applicazione e, dicendo ciò, non manchiamo di augurarvi buon lavoro.

per il

MATERIALE

La DLL descritta nell’articolo riguarda l’oscilloscopio per PC presentato nelle puntate precedenti e disponibile in scatola di montaggio (cod. K8031) al prezzo di 144,00 Euro. L’oscilloscopio è anche disponibile già montato e collaudato (cod. PCS100A) al prezzo di 170,00 Euro. In questo caso la fornitura comprende anche l’alimentatore e una sonda, non compresi nel kit. 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>

Interfaccia Bluetooth per sistemi Velbus

di Corrado Rossi

Permette di attivare via Bluetooth 16 carichi collegati a dispositivi Velbus. I comandi possono essere impartiti sia tramite uno smartphone che mediante un PC sfruttando la loro interfaccia Bluetooth. Entrambi questi apparati debbono essere muniti dell'apposito software da noi messo a punto. Prima parte.



Il mese scorso vi avevamo anticipato che stavamo lavorando a due nuove applicazioni pratiche per interagire con i dispositivi Velbus e precisamente un controllo remoto Bluetooth ed uno GSM; in questo articolo vi presentiamo la versione Bluetooth.

La filosofia di questo progetto, identificato dal codice FT703, non si scosta molto da quella del progetto FT698 pubblicato sul fascicolo precedente: anche questo, infatti, permette di comandare in modalità wireless una serie di moduli di potenza Velbus, ma in questo caso il “telecomando” è costituito da un computer o da uno smartphone,

entrambi collegati all'interfaccia Velbus tramite collegamento Bluetooth (o semplicemente BT) ed entrambi dotati di un apposito programma sviluppato per l'occasione. Questa soluzione è molto versatile in quanto permette di rimpiazzare tutti i possibili telecomandi (il più delle volte distribuiti nei vari locali) con dei telefoni cellulari (che supportano applicazioni Java), normalmente a portata di mano. Non è inoltre cosa strana che a casa si passino ore di fronte al computer: in questo caso, la possibilità di interagire con la rete Velbus direttamente da PC permette di continuare nei nostri lavori senza abbandonare la postazio- ➤

ne. Questo controllo è più complesso di quello presentato il mese scorso; anche per questo motivo abbiamo previsto due puntate: in queste pagine ci occuperemo dell'hardware e della realizzazione pratica, mentre il mese prossimo presenteremo il firmware e i due programmi di controllo per PC e smartphone (quest'ultimo sviluppato in Java) oltre alle note pratiche di utilizzo. Dal punto di vista elettrico, questo dispositivo è in grado di controllare fino a 16 carichi differenti e, di fatto, si comporta esattamente come due schede di ingresso a 8 pulsanti del tipo VMB8PB quindi, una volta inserito in rete, esso occupa due indirizzi, il primo (indirizzo base) programmato manualmente sui due rotary switch, ed il secondo calcolato automaticamente incrementando l'indirizzo base di una unità. I canali 1÷8 sono associati all'indirizzo base, mentre i canali 9÷16 sono pertinenti all'indirizzo "base+1".

Struttura del dispositivo

La struttura di questo progetto è molto simile a quella già vista nel radiocontrollo a 16 canali: la principale differenza è rappresentata dall'interfaccia RF. Anche in questo caso abbiamo sfruttato le capacità di conversione da seriale a protocollo Velbus, e viceversa, offerte dalla scheda VMB1RS di Velleman. A quanti non conoscono tale scheda, diciamo che lo stream di dati che viaggia sull'apposita linea è piuttosto

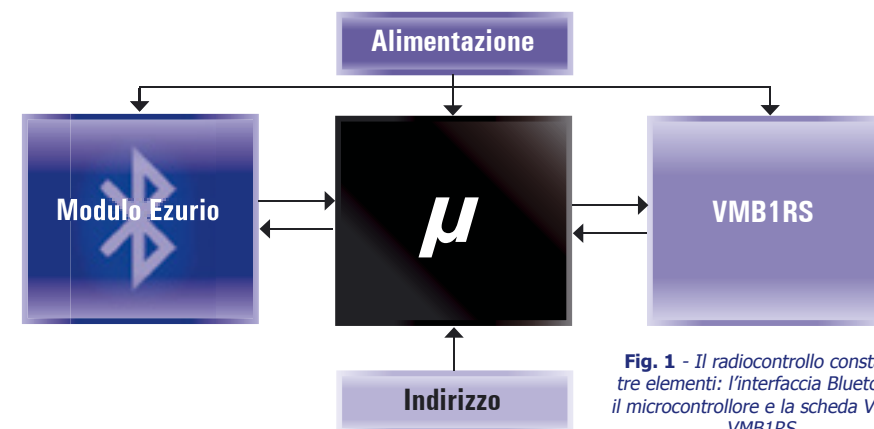


Fig. 1 - Il radiocontrollo consta di tre elementi: l'interfaccia Bluetooth, il microcontrollore e la scheda Velbus VMB1RS.

sto complesso: la scheda VMB1RS è in grado di effettuare una traduzione da Velbus a "seriale 38.400,8,N,1" e viceversa, utilizzando un set di istruzioni meno complesso di quello in transito sul bus. Pertanto qualsiasi microcontrollore dotato di porta seriale può comunicare in maniera relativamente semplice con i dispositivi Velbus passando attraverso l'interprete VMB1RS. Per avere maggiori informazioni, vi suggeriamo di recuperare i fascicoli precedenti nei quali abbiamo trattato in maniera approfondita l'argomento. Esaurita la premessa, vediamo come è costituita la scheda in oggetto che, per poter interagire con la nostra rete domotica, deve essere collegata direttamente alla linea Velbus che, come sappiamo, è composta da 4 conduttori (due per i dati e due di alimentazione). Esattamente come avviene nel radiocontrollo FT698, le linee GND e +12Vcc provenienti dal bus alimentano sia la scheda di

controllo che il modulo VMB1RS; le due linee dati, invece, vengono collegate direttamente alla linea seriale di ingresso/uscita del modulo VMB1RS. All'interno della scheda base possiamo distinguere tre elementi fondamentali, schematizzati in figura 1: il primo è un ricevitore Bluetooth (BISM2), che mette in collegamento il dispositivo al PC o allo smartphone (sui quali operano i programmi di gestione). Il secondo è un microcontrollore, che legge e interpreta i dati in arrivo dal BISM2 e successivamente genera lo stream di dati da inviare alla scheda VMB1RS, terzo e ultimo elemento del sistema.

Come già anticipato, questo circuito necessita di un indirizzo, programmabile tramite due preselettori rotativi esattamente come lo si programma in qualsiasi modulo Velbus.

Principio di funzionamento

Il principio di funzionamento è ab-

RM ELETTRONICA SNC

vendita componenti elettronici

rivenditore autorizzato:



**FUTURA
ELETTRONICA**

NUOVA

ELETTRONICA

G.P.E.



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

Tabella 1 - Il protocollo di comunicazione dell'interfaccia Bluetooth

Funzione	Comando	Parametro	Risposta
Invia comando "Pressione breve del pulsante".	*/BRxx	xx corrisponde al pulsante (canale) da gestire tramite un comando Velbus. (se xx=1÷8 il comando è inviato dall'indirizzo base impostato dal rotary switch, se xx=9÷16 il comando è inviato dall'indirizzo base+1, calcolato dal firmware).	*/BREVE < canale > #
Invia comando "Pressione lunga del pulsante".	*/LUxx	Come il precedente.	*/LUNGA < canale > #
Invia comando "Rilascio del pulsante".	*/RIxx	Come il precedente.	*/RILASCIO < canale > #
Richiede il nome del modulo Ezurio.	*/NAM?	Nessuno.	*/NAM:Ezurio
Cambia il nome del modulo Ezurio.	*/NAM:nnnnnnnnnn	nnnnnnnnnn corrisponde al nuovo nome.	*/NAM:Ezurio

bastanza semplice: sullo smartphone e sul PC sono installati i programmi di gestione che emulano sedici pulsanti virtuali per mezzo dei quali l'utente può controllare lo stato di sedici uscite. Sia il PC che lo smartphone, quando accesi e posizionati nell'area di copertura della rete Bluetooth, entrano in comunicazione con il BISM2 che, pertanto, deve essere stato configurato per riconoscere sia il telefono che il computer. Una volta creata la connessione Bluetooth, lo smartphone o il PC comunicano direttamente con il PIC che elabora delle stringhe di comando da inviare alla scheda VMB1RS in funzione dei comandi ricevuti. Il protocollo di comunicazione che il dispositivo remoto deve utilizzare è visibile in tabella 1: si tratta di pochi e semplici comandi, che potranno essere utilizzati per creare applicazioni personalizzate. I primi tre (*/BR, */LU e */RI) servono per informare il micro che un "pulsante vir-

tuale" è stato premuto brevemente, mantenuto premuto oppure rilasciato. I pulsanti da 1 a 8 sono associati all'indirizzo base, mentre quelli da 9 a 16 corrispondono all'indirizzo "base+1". Il circuito conferma sempre la ricezione di un comando con una risposta associata al canale attivato. Gli ultimi due comandi (*/NAM...), invece, permettono di leggere e modificare il nome dell'interfaccia Bluetooth: in tal modo è possibile identificare e personalizzare il nome del dispositivo (il nome predefinito è "Ezurio"). Il sistema è in grado di memorizzare fino a 10 dispositivi tra PC e telefoni, ma può comunicare solamente con uno di essi alla volta. Una volta stabilita la connessione, il dispositivo (sia esso un telefono o un PC) si comporta come un telecomando evoluto dotato di interfaccia grafica.

Schema elettrico

Poiché la caratteristica principale di

questo progetto è l'interfaccia Bluetooth, iniziamo la descrizione dello schema proprio da essa.

Il ricetrasmittitore da noi adottato è il noto BISM2 (TRBLU20, indicato nello schema con U3) della Ezurio del quale riassumiamo brevemente le caratteristiche:

- modem Bluetooth in classe 1 con antenna integrata;
- connessione seriale RS232 (a livello TTL 3,3 Vdc) con cui può essere collegato a qualsiasi dispositivo esterno;
- protocollo di comunicazione tramite un set di comandi AT molto potente e versatile.

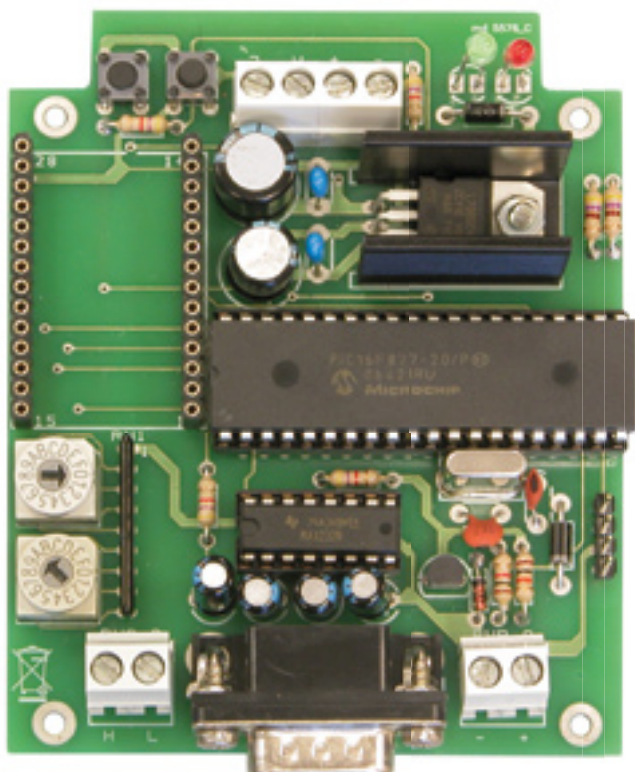
Tramite il BISM2, qualsiasi micro può disporre in maniera semplice, efficace e veloce di un accesso Bluetooth: ciò è particolarmente apprezzato dagli sviluppatori di software che, in questo modo, non devono affrontare complessi sviluppi partendo dal set di comandi HCI tipici di questo standard.

Il BISM2 comunica con il microcon- ➤

Tabella 2 - Segnalazioni visive

Il circuito comunica all'utente i vari stati operativi per mezzo di segnalazioni luminose effettuate tramite due led, uno verde ed uno rosso. Vediamo di seguito le varie situazioni che si possono presentare.

Segnalazione	Descrizione
	Led verde lampeggiante velocemente. Il microcontrollore è acceso, il firmware sta eseguendo il controllo del sistema. Se non sono presenti problemi hardware, il led verde si spegne.
	Led rosso acceso a luce fissa. Il microcontrollore sta inizializzando l'interfaccia Bluetooth BISM2. Se tutto procede correttamente, il led rosso resta acceso per 10 secondi quindi si spegne.
	Led rosso lampeggiante a cadenza di 1 secondo. Il modulo BISM2 non è presente, è guasto oppure non comunica con il micro. Il programma si ferma qui ed il led continua a lampeggiare.
	Led verde che effettua brevi lampeggi. Il dispositivo (PC o smartphone) su cui risiede il programma di gestione è in comunicazione (led verde del BISM2 acceso): ogni lampeggio corrisponde ad un comando ricevuto.
	Led rosso che effettua brevi lampeggi. Il microcontrollore ha ricevuto istruzioni dal programma di controllo ed ha inviato stringhe di comando alla scheda VMB1RS secondo il protocollo seriale Velbus.

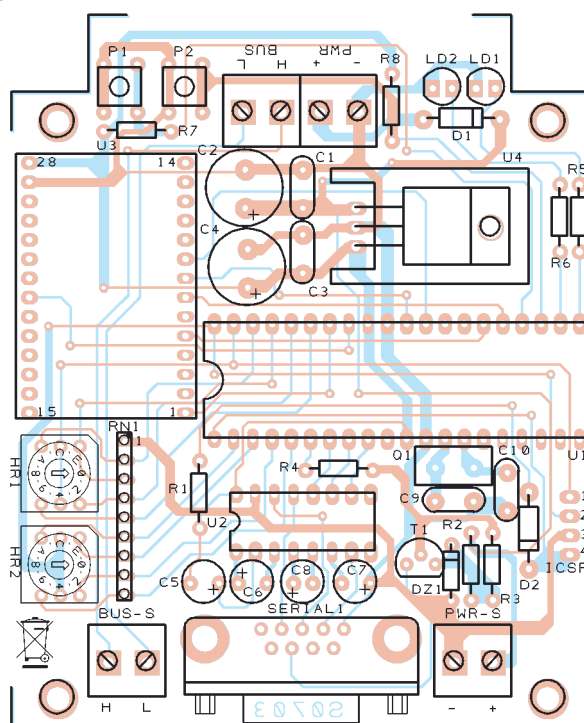


ELENCO COMPONENTI:

R1, R4, R7, R8: 4,7 kohm
 R2: 1 kohm
 R3: 10 kohm
 R5, R6: 470 ohm
 RN1: Rete resistiva 8 x 10 kohm + C
 C1: 100 nF multistrato
 C2, C4: 470 µF 25 VL elettrolitico
 C3: 100 nF multistrato
 C5÷C8: 1 µF 100 VL elettrolitico
 C9, C10: 10 pF ceramico
 D1: 1N4007

DZ1: Zener 7,5 V 400mW
 T1: BC557
 Q1: Quarzo 20 MHz
 U1: PIC16F877 (MF703)
 U2: MAX232
 U3: Modulo Bluetooth con scheda di supporto (FT622M)
 U4: 7805
 LD1: led 3 mm rosso
 LD2: led 3 mm verde
 HR1, HR2: Selettore rotativo esadecimale

P1, P2: Microswitch
 Varie:
 - Morsettiere 2 poli (4 pz.)
 - Connettore DB9 maschio
 - Zoccolo 8+8
 - Zoccolo 20+20
 - Dissipatore (ML26)
 - Vite 10 mm 3 MA
 - Dado 3 MA
 - Strip femmina 14 pin (2 pz.)
 - circuito stampato codice S703



La realizzazione di questo circuito, che impiega solo componenti ad inserzione, è piuttosto semplice.

imposta il nibble più significativo e HR2 quello meno significativo: ad esempio, volendo assegnare al dispositivo l'indirizzo 2A, dovremo portare HR1 sul valore 2 e HR2 sul valore A. Anche se lo abbiamo già detto, talvolta "repetita juvant": poiché questo dispositivo controlla 16 canali esso emula due schede VMB8PB a otto canali, quindi, occupa due indirizzi Velbus (quello impostato sui rotary switch e quello immediatamente dopo, calcolato automaticamente dal firmware). Come ultima funzione, il PIC gestisce anche lo stato di due led e di due pulsanti. I led, LD2 (verde) e LD1 (rosso), comunicano differenti

stati operativi a seconda dei lampeggi, come riportato in tabella 2. I pulsanti, invece, servono solo per test: la pressione di P1 simula l'attivazione del canale 1, associato all'indirizzo Velbus base; analogamente, il P2 simula l'attivazione del canale 9, associato al secondo indirizzo Velbus. Per analogia con le due schede VMB8PB, P1 attiva il canale 1 della prima scheda e P2 il canale 1 della seconda. L'alimentatore è realizzato con il classico 7805: il diodo D1 protegge il circuito da accidentali inversioni di polarità mentre i condensatori di ingresso C1 e C2 realizzano un filtro sulla linea a 12 V, mentre i condensatori C3 e C4 filtra-

no e ripuliscono da eventuali disturbi provenienti dal bus la linea a 5 V con cui viene alimentata tutta l'elettronica ➤



Fig. 2

Montaggio del modem Bluetooth.

ca. Anche questo dispositivo è stato progettato per essere inserito all'interno di un contenitore TEK0 modello Tenclos 660.5.

Realizzazione pratica

L'interfaccia impiega solo componenti tradizionali con montaggio ad inserzione, pertanto la sua realizzazione non presenta particolari difficoltà. Per prima cosa bisogna disporre del circuito stampato che potrete autocostituire con la tecnica della fotoincisione: il master relativo, come sempre, può essere scaricato gratuitamente dal nostro sito (www.elettronica.in.it); ricordiamo che si tratta di un PCB a doppia faccia con fori metallizzati quindi la realizzazione casalinga, anche se possibile, potrebbe presentare qualche piccola difficoltà. Se non ve la sentite di affrontare una realizzazione "do-it-yourself", allora potete richiedere il circuito stampato direttamente all'azienda che ci realizza i prototipi (www.mdsrl.it), indicando il codice del circuito di interesse (S0703). Una volta in possesso del circuito stampato, potete iniziare il montaggio partendo come sempre dai componenti a basso profilo ossia le resistenze, i diodi (fate attenzione alla polarità), la

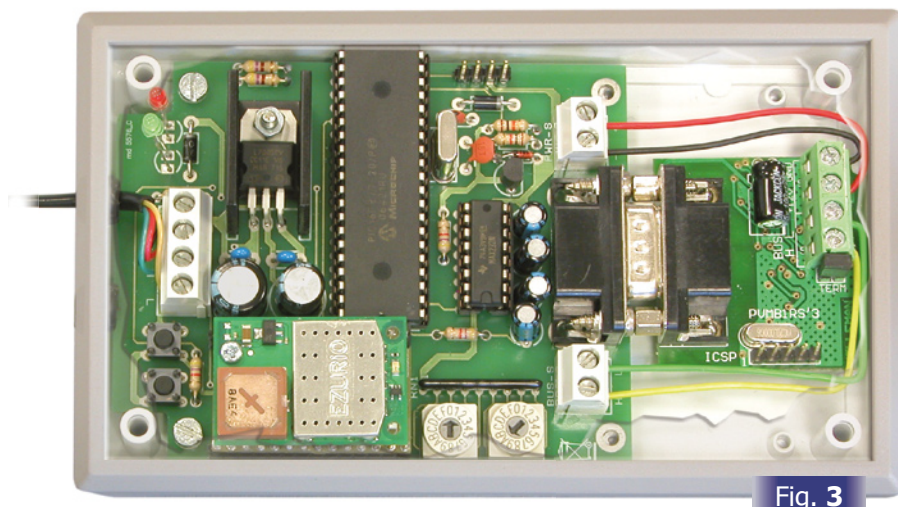


Fig. 3

Fig. 2 - L'interfaccia Bluetooth per reti Velbus è stata inserita all'interno di un contenitore Teko mod. 660.

rete resistiva RN1, i condensatori non polarizzati, i pulsanti P1 e P2 nonché i preselettori binari HR1 e HR2. Proseguite montando gli zoccoli del microcontrollore e del driver RS232: entrambi hanno la tacca di riferimento a sinistra, osservando il PCB come indicato nel piano di montaggio. Proseguite inserendo il transistor T1 (il lato piatto deve essere rivolto all'esterno del circuito stampato), le tre morsettiere a passo 5,08 mm, il quarzo Q1 (montato verticalmente), tutti i condensatori elettrolitici, le due linee di pin strip a passo 2,54 mm (sul-

le quali viene poi montato il circuito di adattamento per il ricetrasmittitore Bluetooth), la pin strip per la programmazione "in circuit" (ICSP) e il connettore seriale DB9 sul quale viene collegata la scheda VMB1RS. Resta da montare lo stabilizzatore di tensione che, alimentando anche il BISM2 oltre al micro e alle periferiche, necessita di un piccolo dissipatore. Piegate pertanto i pin dello stabilizzatore U4 a 90°, come visibile nel piano di montaggio: in questo modo è possibile appoggiare il lato metallico direttamente 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 potete saldare i tre pin del 7805. Montate ora gli ultimi due componenti, i led LD1 e LD2, tagliandone i contatti alla lunghezza di 28 mm e facendoli sporgere dal lato saldature di 2 mm circa. Ricordate infatti che, esattamente come per il radiocollaboro a 16 canali, le teste dei led devono sporgere dal coperchio del contenitore. Per il momento non innestare i due chip (PIC e MAX232) ed il modem Bluetooth nei rispettivi zoccoli, mentre potete inserire il circuito di adattamento del BISM2 sui relativi pin strip, rispettando il verso indicato nel piano di montaggio. Il micro deve essere

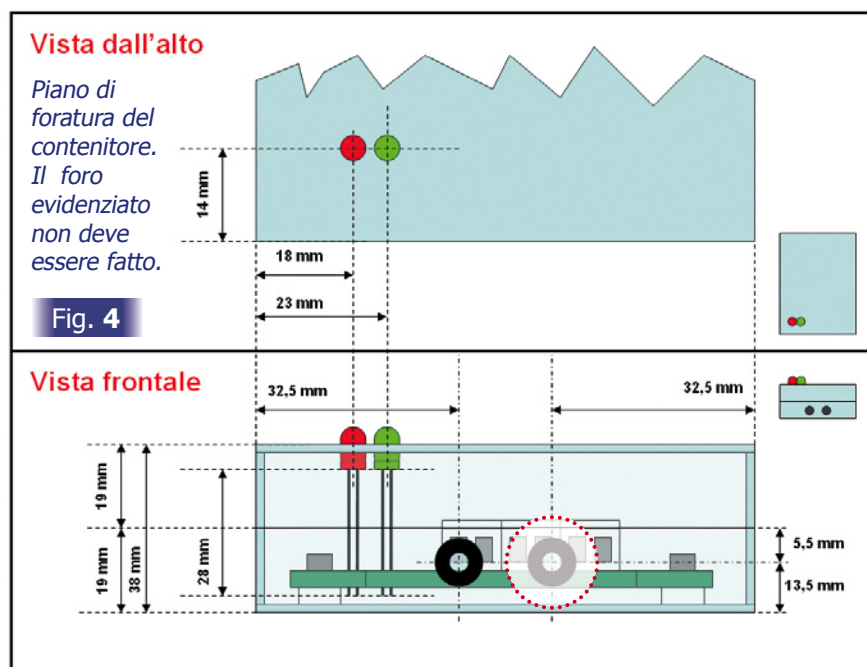


Fig. 4

I software di gestione

Il dispositivo descritto in queste pagine può essere accoppiato con un PC o con uno Smartphone (dotati di interfaccia Bluetooth) sui quali andranno caricati due differenti programmi da noi messi a punto. Il primo, la cui schermata principale è visibile in figura A, funziona in ambiente Windows ed è stato sviluppato con VBNet 2003. Il secondo, invece, è stato sviluppato in linguaggio Java: la figura B mostra la schermata principale. Per quanto differenti di aspetto, i due programmi svolgono la stessa funzione: essi infatti emulano tramite interfaccia grafica la pressione di 16 pulsanti e generano le stringhe di comando (già descritte in tabella 1) che il dispositivo riceve e converte in sentenze di comando secondo il protocollo Velbus. Entrambi i programmi segnalano la pressione breve, la pressione lunga ed il rilascio del pulsante virtuale, rispettando quanto avviene con le schede VMB8PB.



Fig. A



Fig. B

programmato successivamente con il file MF703.hex, scaricabile dal sito: per far ciò vi occorre un idoneo programmatore.

Collaudo e montaggio nel contenitore

Prima di procedere con l'accensione del sistema, è necessario verificare almeno la tensione di alimentazione: attivate perciò il circuito fornendo una tensione continua di circa 12 V ai morsetti +/- PWR; verificate quindi con un multimetro che la tensione di 5 V sia presente sul pin 28 del circuito di adattamento del BISM, sui pin 1 e 32 del PIC, e sul pin 5 del MAX232. Spegnete il circuito, attendete un paio di minuti per dar modo a tutti i condensatori elettrolitici di scaricarsi, quindi innestate il PIC ed il MAX232 rispettando il verso della tacca sugli

zoccoli, quindi accendete il circuito. Se il PIC non è programmato, dovete ovviamente procedere alla sua programmazione; noi qui lo consideriamo già programmato. Immediatamente dopo l'accensione ed il reset, il led verde lampeggia per segnalare che il programma funziona (fate riferimento alla tabella 2 per le segnalazioni visive). Dopo qualche secondo il led verde si spegne e quello rosso si accende a luce fissa (ricerca del BISM2), quindi inizia a lampeggiare a cadenza di un secondo per segnalare di non avere trovato l'interfaccia Bluetooth. Spegnete nuovamente il circuito, montate il BISM2 sul PCB di adattamento come visibile nel piano di montaggio e riaccendete: il led verde lampeggia ancora poi si spegne mentre il led rosso rimane acceso per 10 secondi, ad indicare che il PIC ha

trovato il BISM e lo ha configurato correttamente. Ora potete spegnere l'interfaccia ed inserirla nel contenitore plastico. Quest'ultimo deve essere preparato seguendo lo stesso piano di foratura utilizzato per il radiocontrollo a 16 canali descritto sul fascicolo precedente. Qui lo riportiamo solo a titolo informativo (vedere figura 4; il foro di passaggio per il cavo dell'antenna non serve); per le note meccaniche, fate riferimento a quanto visto sul fascicolo 118. Collegate la scheda VMB1RS al connettore DB9 e portate alla sua morsettiera i quattro fili (due di alimentazione e due per il bus) come visibile in figura 3, che mostra il risultato finale. Posizionate ora la scheda sulla base del contenitore e fissatela con quattro viti autofilettanti da 2,5 mm lunghe 6 mm circa.

Conclusioni

Giunti a questo punto, avete a disposizione la nostra scheda d'interfaccia Bluetooth montata e collaudata, pronta per essere utilizzata insieme ai programmi per PC e Smartphone e adatta ad essere inserita in qualsiasi punto di una rete Velbus esistente. Nella prossima puntata, interamente dedicata al software, verranno approfonditi i seguenti punti:

- il firmware;
- le procedure per accoppiare i dispositivi BT al radiocontrollo;
- i due programmi di gestione;
- l'utilizzo in pratica.

Appuntamento quindi al mese prossimo in edicola.

per il

MATERIALE

I componenti utilizzati in questo progetto sono facilmente reperibili in commercio. Il master del circuito stampato, i programmi di controllo per PC e smartphone ed il firmware per programmare il micro sono scaricabili dal sito della rivista (www.elettronica.in.it). Il micro è disponibile anche già programmato al prezzo di 18,00 Euro (cod. MF703). Il ricetrasmittitore BISM2 con relativo circuito stampato di adattamento (cod. FT622M) può essere acquistato al prezzo di 75,00 Euro. Tutti i prezzi sono comprensivi di IVA.

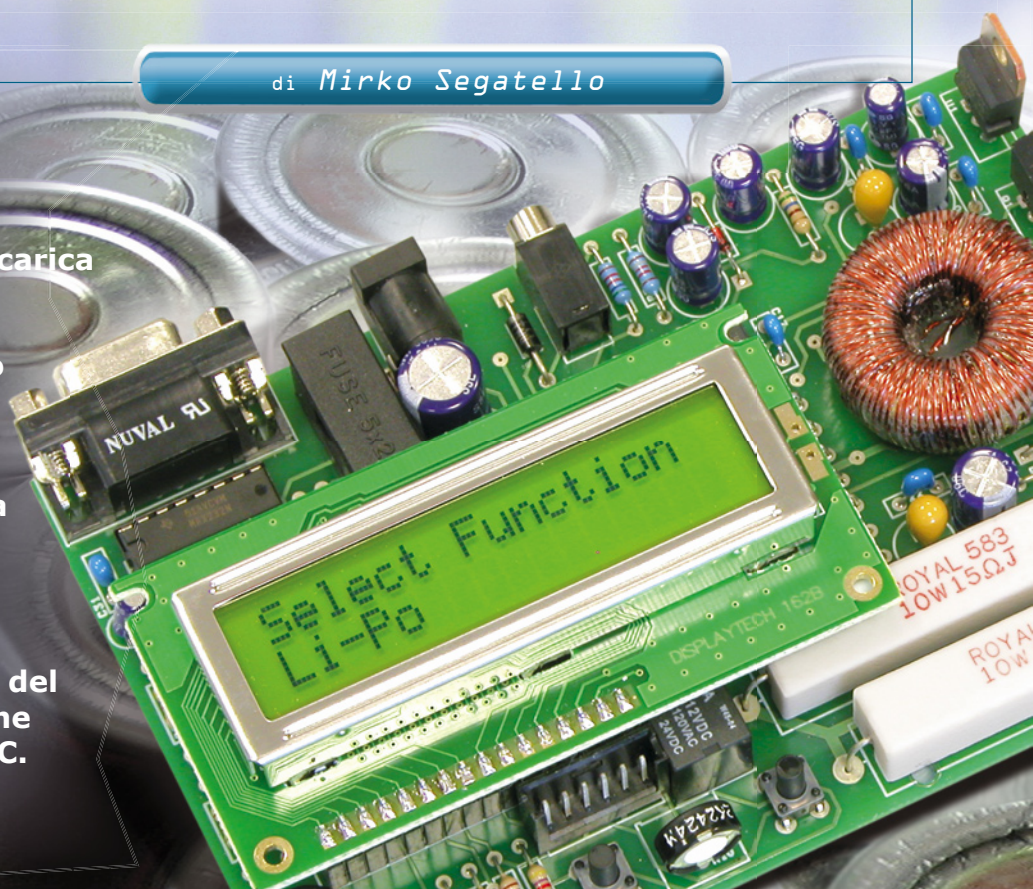
Il materiale va richiesto a: Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)

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

Caricabatterie programmabile per impieghi professionali

di Mirko Segatello

Può ricaricare differenti tipologie di batterie, incluse le più recenti ai polimeri di litio. I parametri elettrici di ricarica sono configurabili in funzione della batteria, ottimizzando in tal modo l'efficienza della ricarica stessa. Totalmente gestito da microcontrollore, utilizza solamente tre pulsanti di controllo. Ai fini della sicurezza, dispone di un efficace sistema di controllo della polarità e del tipo di batteria. Può anche essere collegato ad un PC. Prima parte.



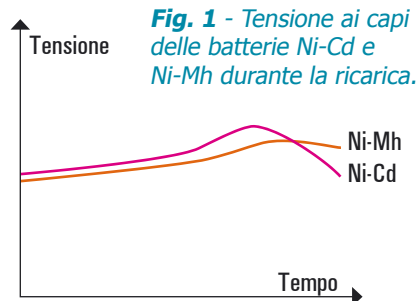
Con la disponibilità sul mercato di svariate tipologie di batterie ricaricabili, ognuna caratterizzata da parametri elettrici specifici, un unico sistema di ricarica programmabile rappresenta la miglior soluzione possibile per ottimizzare i costi e velocizzare i tempi di ricarica. Spesso, infatti, si acquistano delle batterie con relativo caricabatteria, ma all'uscita di una nuova versione delle batterie (magari di capacità superiore), quest'ultimo diventa inutilizzabile. L'FT705 (questo è il codice che abbiamo assegnato al progetto) nasce proprio per sopperire al rischio di dover sostituire il circuito di ricarica

ogni volta che si sostituiscono le batterie. Si tratta di un caricabatteria professionale programmabile per impiego modellistico ed hobbistico che, grazie alla grande flessibilità operativa e alla completa programmabilità delle funzioni, può adattarsi alla maggior parte delle necessità di ricarica in ambito domestico. La possibilità del funzionamento in automatico, insieme al controllo della polarità della batteria, inoltre, permette anche ai meno esperti di ottenere delle ricariche rapide e precise senza correre rischi di esplosioni, eventi questi non improbabili in caso di parametri di ricarica non adatti alla batteria. ➤

- Idoneo alla carica di batterie tipo Ni-Cd, Ni-Mh, Li-Po, Pb.
- Modalità di carica sia a corrente costante che a tensione costante.
- Corrente di carica impostabile da 0,1 A a 2 A a passi di 100 mA.
- Numero di celle ammesse: 1 ÷ 10 celle Ni-Cd, Ni-Mh oppure 1 ÷ 3 celle Litio oppure 1 ÷ 6 celle Piombo.
- Scarica a corrente costante impostabile da 0,1 A a 0,5 A a passi di 100 mA.
- Possibilità di carica a tempo, con durata massima di 25 ore e regolazione a passi di 30 minuti.
- Possibilità di spegnimento per sovratemperatura tramite sensore esterno.
- Carica rapida con spegnimento di tipo "delta-peak" per celle Ni-Cd e Ni-Mh.
- Carica rapida CC/CV per celle al Litio.
- Funzionamento automatico o manuale.
- Protezione contro l'inversione di polarità e le sovratensioni sia in ingresso che in uscita.
- Protezione contro i cortocircuiti e la sconnessione accidentale delle batterie.
- Tensione di alimentazione in corrente continua da 11 a 15 volt.
- Interfacciabile tramite porta seriale ad un PC, con possibilità di data-logger in tempo reale tramite l'apposito software.

Oltre alla modalità automatica, esso dispone anche di un set-up manuale per chi volesse personalizzare i parametri di funzionamento a seconda delle batterie utilizzate. Potete utilizzare questo dispositivo per ricaricare agevolmente le batterie del vostro avvitatore, della scopa elettrica, oppure dei giocattoli dei vostri figli. La realizzazione è particolarmente compatta con dimensioni di soli 130 x 90 mm che consentono l'utilizzo di un elegante contenitore che conferisce al caricabatteria un aspetto professionale, aumentando nel contempo la praticità di utilizzo. L'attenta scelta del circuito e dei componenti elettronici ha permesso di contenere i costi di realizzazione e, allo stesso tempo, di ottenere delle caratteristiche confrontabili con quelle di apparecchiature commerciali professionali. Il circuito viene alimentato con una tensione continua di 12 volt, facilmente ottenibile da un alimentatore non stabilizzato oppure da una batteria da auto, peculiarità questa sicuramente gradita ai numerosi appassionati di modellismo che hanno la necessità di alimentare il carica-batterie tramite la presa accendisi-

gari della propria vettura. Particolare attenzione è stata posta alla sicurezza durante il funzionamento: abbiamo infatti previsto una serie di protezioni per la salvaguardia delle batterie, della sorgente di alimentazione, del caricabatteria stesso e dell'utente. Potete stare quindi tranquilli: anche se inserite le batterie con il verso sbagliato non correte alcun rischio perché il circuito avvisa immediatamente dell'anomalia con un segnale sonoro ed un messaggio di testo, senza avviare la ricarica. Il display LCD da 16 caratteri su 2 righe permette di gestire con la massima facilità la programmazione e consente di seguire con estrema precisione lo stato di carica o di scarica. Visto lo "spessore" del progetto, abbiamo deciso di pubblicarlo in due puntate, una dedicata all'hardware ed una specifica



per il software e le note di utilizzo. In questa prima puntata scopriamo tutti i segreti dell'hardware.

Le batterie

Prima di presentare il circuito vero e proprio, vediamo brevemente quali sono le caratteristiche principali delle più comuni batterie ricaricabili.

Consideriamo innanzitutto le batterie al Nichel-Cadmio (abbreviato Ni-Cd), che sono state le prime ricaricabili introdotte sul mercato in larga scala e sono ancora oggi disponibili nei classici formati stilo, mini-stilo ecc. Il loro più grosso limite è "l'effetto memoria" che si manifesta quando si ricarica una batteria non completamente scarica: secondo tale effetto, la batteria, pur essendo completamente carica, non riesce a fornire tutta l'energia immagazzinata perché "ricorda" in un certo modo che non vi è stata una scarica completa. Per ovviare a questo inconveniente è necessario utilizzare la batteria fino a totale esaurimento, prima di ricaricarla. Una batteria affetta da "effetto memoria" può comunque essere recuperata con una serie di cicli di scarica e carica a bassa corrente. Con "scarica" si intende l'utilizzo della batteria sino a quando la sua tensione non scende a 0,8-0,9 volt. Se per qualche motivo la tensione dovesse scendere sotto questo valore, anche per corti accidentali, si rischia il danneggiamento della batteria stessa. Le Ni-Cd sono disponibili in tecnologia tradizionale, idonee a scariche a bassa corrente e facilmente ricaricabili (se completamente scariche) con una corrente costante ad un decimo della capacità nominale (1/10 C, ad esempio 80 mA per una capacità di 800 mA/h) per un tempo di 14 ore. Esiste anche una versione fast-charge, in grado di fornire maggiore corrente in uscita e di accettare ricariche più veloci con correnti più elevate. Per la ricarica, in questo caso, si sfrutta una proprietà legata alla tensione ai capi della batteria durante la ricarica: al raggiungi-

mento della piena carica, la tensione (sino a quel momento cresciuta lentamente) sale più velocemente per poi iniziare a decrescere. Questo picco (Delta-Peak) identifica il raggiungimento della massima carica ed è accompagnato da un incremento della temperatura interna della batteria (Figura 1). Ricaricando ulteriormente la batteria si rischia il suo danneggiamento. Rilevando il picco di tensione è possibile fermare la carica al punto giusto indipendentemente dal valore iniziale di energia prima della ricarica.

Le batterie Nichel-Metal-Hidrato (abbreviato Ni-Mh) presentano, rispetto alle Ni-Cd, una maggiore densità di carica e un minor effetto memoria con una corrente di scarica più limitata. Per la ricarica si applica quanto detto a proposito delle batterie Ni-Cd, tranne per il fatto che, al raggiungimento della massima carica, nelle Ni-Mh la tensione rimane prima stabile (non presenta il Delta-Peak) per poi scendere di pochissimo. È quindi necessario aumentare la

sensibilità nella lettura della tensione. Anche in questo caso, prossimi alla massima carica, la temperatura interna aumenta.

Le batterie al piombo sono decisamente ingombranti e pesanti rispetto all'energia immagazzinata, ma presentano due vantaggi: non soffrono dell'effetto memoria e non vengono danneggiate da una ricarica prolungata a basse correnti. Sono quindi idonee per tutti gli impieghi in apparecchiature non portatili dove la ricarica può avvenire per tempi prolungati, come ad esempio in autoveicoli, motociclette, sistemi di illuminazione di emergenza, impianti fotovoltaici. La carica non presenta particolari problemi, basta non eccedere con la corrente ed avere l'accortezza di interromperla se la tensione raggiunge la soglia massima (2,3 V circa per elemento).

Vediamo ora le batterie al litio, intro-

dotte sul mercato per la necessità di fornire ai telefoni cellulari una sempre maggiore energia pur mantenendo pesi e dimensioni contenute. Le prime ad essere introdotte sono state quelle agli ioni di litio (abbreviato Li-Ion) e successivamente quelle (migliorate) ai polimeri di litio (abbreviato Li-Poly). Non presentano effetto memo-

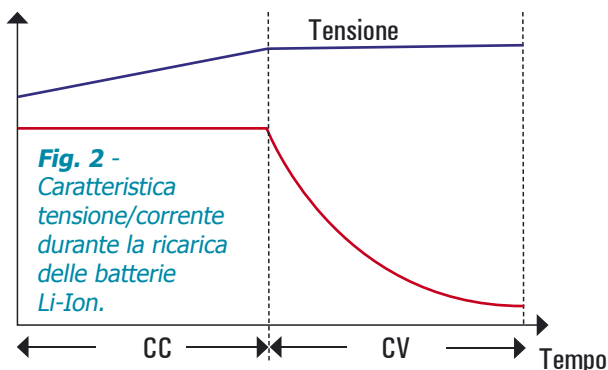


Fig. 2 -
Caratteristica
tensione/corrente
durante la ricarica
delle batterie
Li-Ion.

ria, sono in grado di fornire elevate correnti di scarica e possono essere ricaricate velocemente. Sono più pericolose delle batterie descritte in precedenza in quanto una sovratensione e/o una sovratemperatura durante la carica procurano immediatamente il loro danneggiamento con possibile pericolo di esplosione. La ricarica avviene prima a corrente costante e poi a tensione costante (abbreviato CC-CV). La fase a corrente costante prosegue sino al raggiungimento di un valore di soglia preimpostato, e successivamente si procede con la diminuzione della corrente erogata mantenendo il valore di tensione costante. Quando la corrente scende al di sotto di un decimo della corrente iniziale la batteria è carica. Il sistema anche se più complesso assicura cariche rapide e in tutta sicurezza. In figura 2 è visibile il grafico relativo alla tensione ai capi della batteria (azzurro) e alla corrente (rosso) durante la carica. La tensione massima da mantenere durante la fase a tensione-costante è di 4,1 volt per le Li-Ion e 4,2 volt per le Li-Po.

Quasi tutte le batterie al litio per impiego commerciale sui cellulari ➤

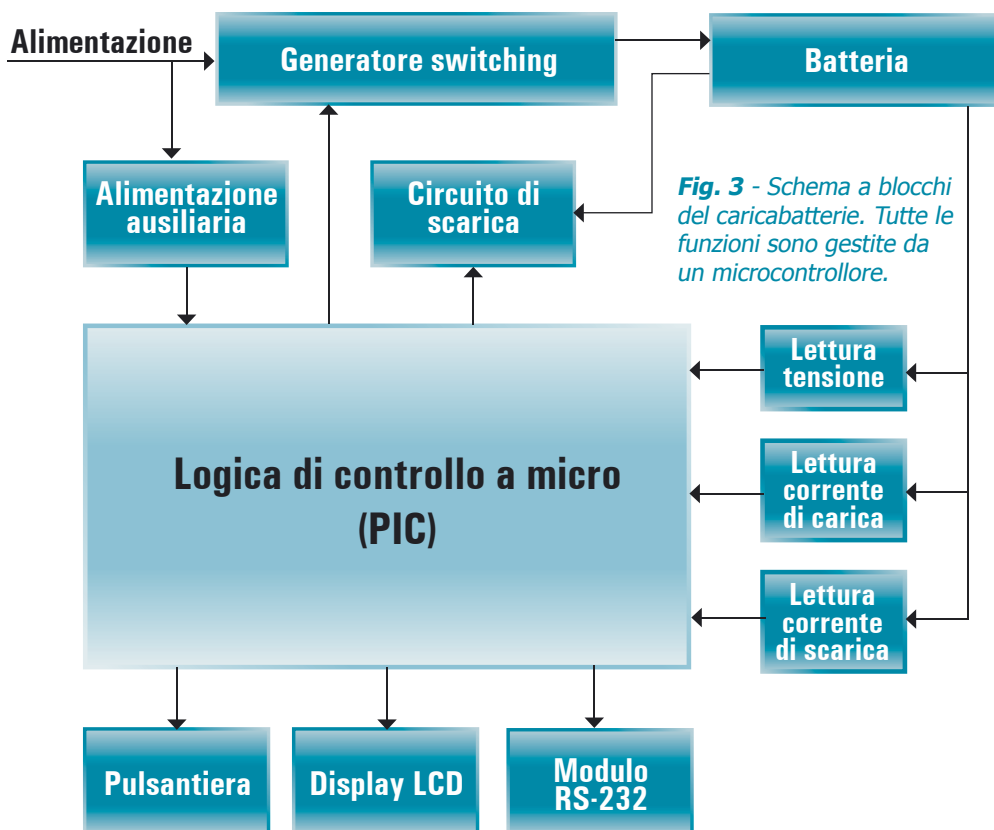
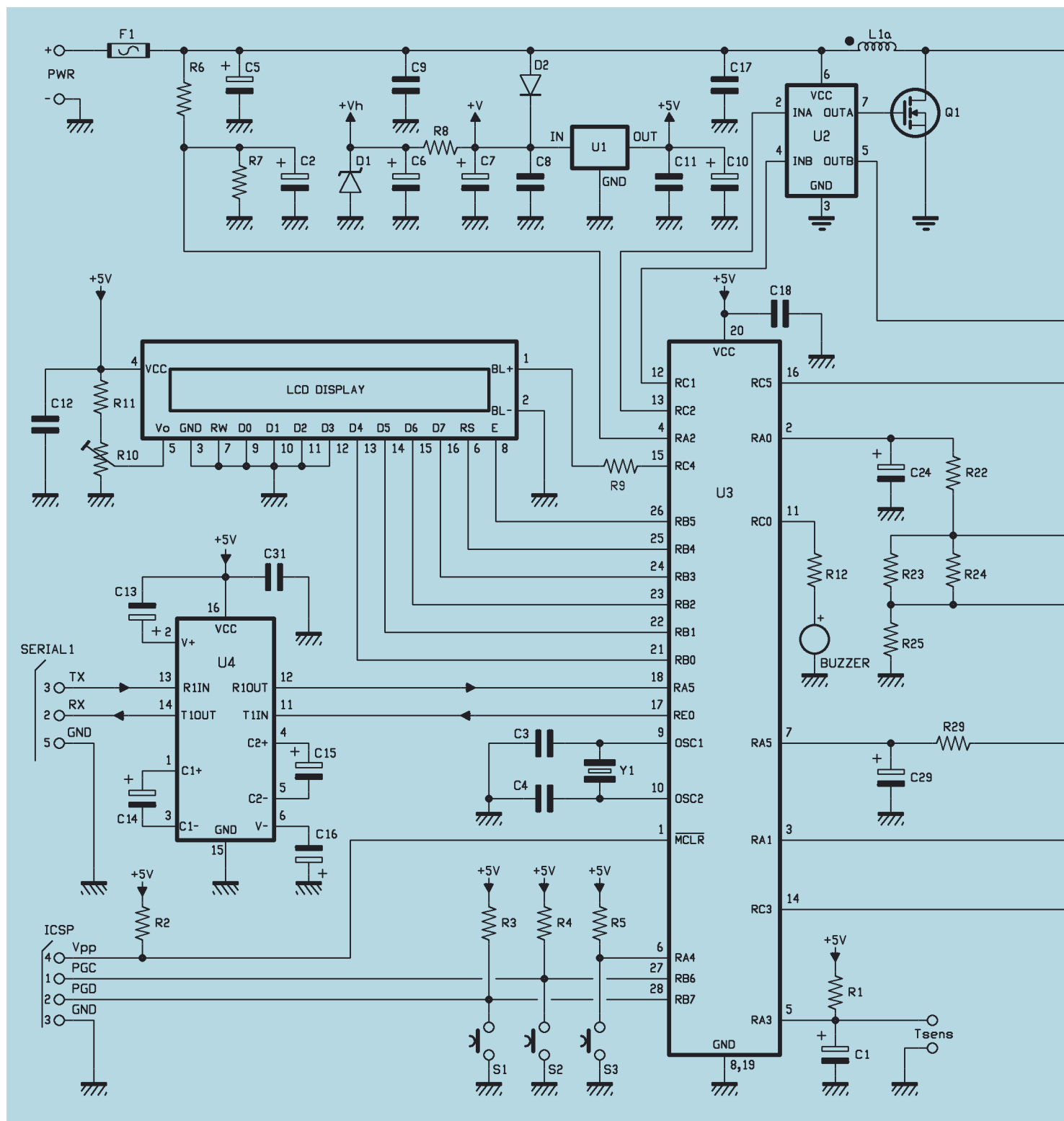


Fig. 3 - Schema a blocchi
del caricabatterie. Tutte le
funzioni sono gestite da
un microcontrollore.



dispongono di un sensore di temperatura, utilizzato tipicamente dal circuito di ricarica a protezione della batteria. In commercio sono disponibili batterie al litio con svariate capacità e tensioni, utilizzabili in tutte le applicazioni in cui sono richieste elevate capacità e correnti di scarica

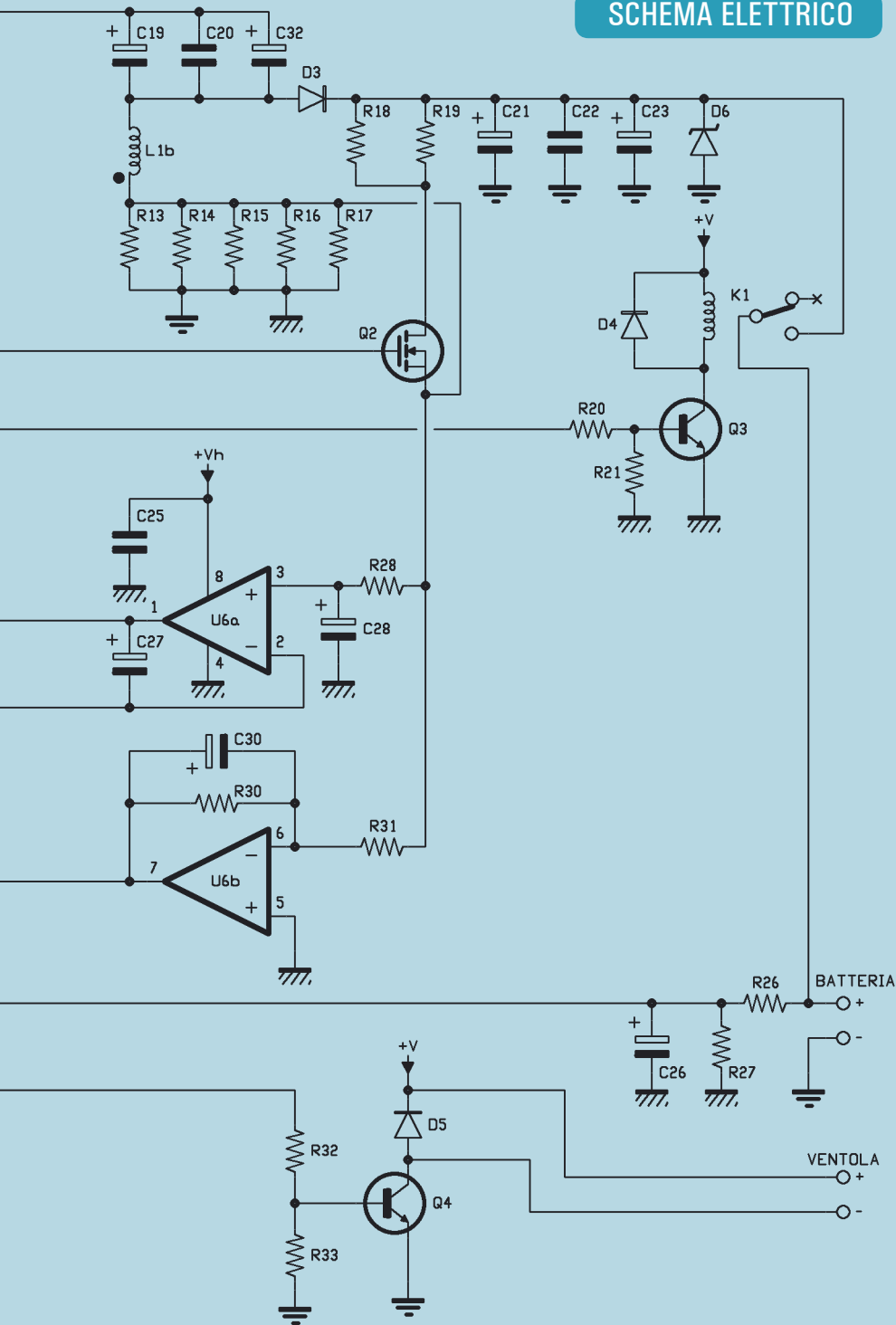
in volumi e pesi contenuti, come ad esempio nel modellismo. Il nostro caricabatteria, dovendo adattarsi a quanto il mercato può offrire, può operare con tensioni e correnti costanti in modo programmato e preciso, monitorando nel contempo tensione, corrente e temperatura.

Questa grande flessibilità è resa possibile dal microprocessore utilizzato nel progetto.

Principio di funzionamento

Un caricabatteria è, essenzialmente, un circuito in grado di generare una corrente o una tensione costante (a

SCHEMA ELETTRICO



un circuito switching gestito da un microcontrollore che disponga di moduli PWM: in questo modo si genera una corrente pulsante ad alta frequenza che successivamente viene convertita in tensione, livellata e filtrata. Il vantaggio di questa soluzione risiede nel poter utilizzare i componenti attivi (mosfet) come degli interruttori e quindi solo nel funzionamento in interdizione e saturazione, dove le perdite di potenza in calore sono minime. Per comprendere il funzionamento di questo circuito facciamo riferimento allo schema a blocchi riportato in figura 3.

A seconda del tipo di batteria ed in funzione del suo stato di carica, il generatore switching (controllato dal PIC, blocco centrale) può fornire una tensione oppure una corrente costante (blocco "Generatore switching"), indipendentemente dal numero di celle utilizzato e dal livello di carica presente. Nel nostro caso abbiamo utilizzato un convertitore DC-DC tipo Sepic che, pur essendo poco conosciuto, è il più idoneo: abbiamo dedicato ad esso un riquadro specifico a pagina 81. Il blocco del circuito di scarica è formato semplicemente da resistenze che hanno lo scopo di disperdere nell'ambiente sotto forma di calore l'energia immagazzinata nelle batterie. Entrambi i blocchi di carica e scarica sono controllati con un segnale PWM. Il caricabatteria è dotato di un display LCD 16x2 caratteri e di una pulsantiera, per la programmazione e la visualizzazione delle informazioni, nonché di una interfaccia standard RS-232 per il collegamento al PC.

Schema elettrico

Iniziamo la spiegazione partendo dall'alimentazione, che giunge alla scheda tramite un connettore plug ed un fusibile: quest'ultimo serve come protezione contro accidentali inversioni di polarità nell'alimentazione o sovracorrenti dovute a guasti nel circuito. L'alimentazione ausiliaria ➤

seconda delle batterie utilizzate) e di monitorare in tempo reale tutti i parametri elettrici più importanti come tensione, corrente e temperatura delle batterie in carica. Poiché esso deve funzionare per ore, è importante che il rendimento complessivo sia alto: in questo caso le ridotte perdite di

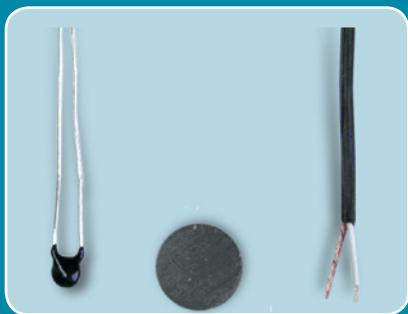
potenza consentono di mantenere il circuito a temperature basse senza la necessità di particolari e costosi sistemi di raffreddamento, garantendo inoltre un'efficienza maggiore nel trasferimento di energia alle batterie. Tutto ciò può essere ottenuto facilmente mediante l'adozione di

La sonda di temperatura

Per realizzare la sonda di temperatura occorrono una termoresistenza, un piccolo magnete e 30 cm di cavo coassiale. Saldate i due fili del cavetto alla termoresistenza, isolando le saldature con della guaina termorestringente. Utilizzate ulteriore termorestringente (di diametro maggiore) per proteggere e irrobustire le saldature. A questo punto fate aderire bene la termoresistenza al magnete e fissate il tutto con



del collante o della colla a caldo. Dall'altra parte del cavo saldate il connettore tipo Jack mono da 3,5 mm. A lavoro ultimato assicuratevi tramite un tester che la resistenza misurata sul connettore sia circa 10 kohm. Il magnete permette di collocare facilmente il sensore sul pacco batteria. Ovviamente è anche possibile assemblare la sonda in modo differente, purché la soluzione adottata consenta un efficiente scambio termico con la batteria.



è ottenuta tramite uno stabilizzatore 7805 protetto contro le inversioni di polarità da un diodo 1N4007 (D2). Tramite il partitore formato da R6 e R7 il microcontrollore monitorizza la tensione di alimentazione interrompendo qualsiasi attività e segnalando l'avvenuta anomalia se questa non è compresa tra i valori ottimali. Può infatti capitare che l'alimentazione non disponga di potenza sufficiente e la tensione possa scendere a valori troppo bassi pregiudicando il corretto funzionamento del circuito; il microcontrollore utilizzato è un PIC16F876 (U3). Il connettore Tsens permette il collegamento del sensore di temperatura formato da una semplice termoresistenza NTC (vedere riquadro in alto). Il sensore non è indispensabile in quanto il circuito, che riconosce in automatico la sua presenza, avvia comunque la carica anche in sua assenza. Se presente, permette di interrompere la carica qualora la temperatura delle batterie superi la soglia preimpostata, rappresentando quindi una sicurezza aggiuntiva per la salvaguardia di batterie, persone ed ambiente. Per la realizzazione del sensore, oltre

alla termoresistenza, occorre anche un piccolo magnete, acquistabile in ferramenta o in cartoleria, ed uno spezzone di cavo schermato (quello per audio va bene) della lunghezza di circa 30 cm.

Nella parte di schema in alto a destra si riconoscono i componenti del Sepic-converter, in cui le capacità di filtro sono costituite da differenti tipologie di condensatori in parallelo. Il motivo è presto detto: per garantire la corrente necessaria, occorre una capacità complessiva elevata, tipica del condensatore elettrolitico che, però, presenta una resistenza serie troppo elevata (soprattutto alle alte frequenze), resistenza che riduce il rendimento. Per ovviare a questo problema, si inseriscono in parallelo agli elettrolitici altri due condensatori di valore inferiore ma con resistenza interna molto bassa, come il condensatore al tantalio e quello ceramico.

Il parallelo di resistenze R13÷R17 crea un carico (shunt) necessario per la lettura della corrente. Utilizzando cinque resistenze in parallelo abbiamo ottenuto il valore più adatto ai nostri scopi, sia per quanto riguarda

la necessaria capacità di dissipazione di potenza, sia per ciò che concerne una maggiore precisione rispetto ad ogni singola resistenza. Inoltre la selezione dei componenti permette di eliminare le calibrizioni in quanto il circuito, una volta montato, presenta già una buona precisione. Nel nostro circuito, le resistenze necessarie alla misura di tensione e corrente sono tutte con precisione del 1%.

Il partitore composto da R26 e R27 permette al microcontrollore di leggere la tensione ai capi della batteria. Il relé K1 ha lo scopo di connettere la batteria al circuito solo dopo la verifica da parte del PIC della correttezza della polarità e della tensione di batteria, evitando così il rischio di danni dovuti a distrazioni o ad un uso incauto del caricabatteria.

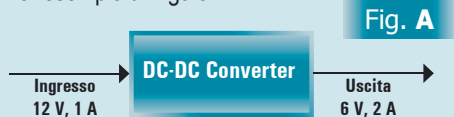
L'operazionale U6 dispone di due amplificatori necessari alla misura della corrente rilevata tramite la piccola caduta di tensione sullo shunt R13÷R17. Essendo possibile sia la carica che la scarica, la corrente può percorrere lo shunt sia in un senso che nell'altro: durante la fase di carica viene usato l'amplificatore invertente U6B e durante la fase di scarica l'amplificatore U6A in configurazione non invertente. La scarica è affidata alle resistenze di grossa potenza R18 e R19, capaci in totale di dissipare 20 W massimi, limitati dal microcontrollore per non eccedere con la temperatura.

Il connettore "VENTOLA" permette di collegare al circuito una piccola ventola di raffreddamento, qualora si decidesse di inserirlo in un contenitore. La ventola entra in funzione solo quando viene impostata una corrente di carica di almeno 500 mA oppure durante la fase di scarica. Il circuito usato in aria libera non necessita di dissipatori, ma si raccomanda comunque una buona ventilazione. La disposizione dei componenti TO220 sullo stampato è tale da consentire facilmente l'inserimento di un dissipatore qualora lo



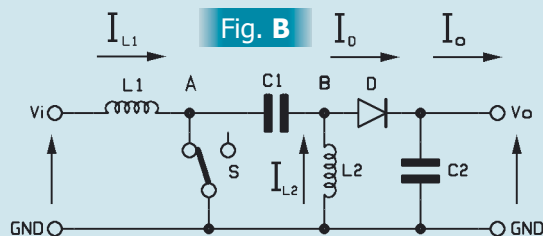
Il convertitore Sepic

Il convertitore Sepic fa parte della famiglia dei "DC-DC converter", cioè dispositivi elettronici che accettano in ingresso una tensione continua e forniscono in uscita una tensione sempre continua, ma con parametri elettrici diversi. Si comportano insomma come un trasformatore solo che, invece di funzionare in alternata, funziona in continua come nell'esempio di Figura A:



In questo esempio il convertitore DC-DC accetta in ingresso una tensione continua di 12 volt con una corrente di 1 ampere e quindi assorbe una potenza di 12 watt. In uscita fornisce una tensione di 6 volt con una corrente di 2 ampere e quindi sempre una potenza di 12 watt. Pur essendo cambiati i parametri elettrici di tensione e corrente, la potenza si è trasferita tutta dall'ingresso all'uscita. Ovviamente questo avviene nel caso ideale; nel caso reale invece, data la non idealità dei componenti utilizzati, sono presenti alcune perdite e pertanto la potenza in uscita è sempre leggermente più bassa della potenza assorbita in ingresso. Per definire queste perdite, si parla di rendimento, ovvero di quel valore calcolabile come il rapporto tra la potenza in uscita e la potenza in ingresso, che di solito raggiunge valori dell'80-90%. Esistono diverse tipologie di schemi a seconda delle funzioni svolte: ad esempio nella configurazione classica del convertitore "Buck" (o "step-down") la tensione di uscita è sempre più bassa della tensione di ingresso, mentre nella configurazione "Boost" (o "step-up") la tensione di uscita è sempre più elevata. Per la nostra applicazione è richiesto che la tensione di uscita possa essere sia maggiore che minore di quella in ingresso dovendo ricaricare agevolmente tanto una quanto dieci celle (in questo caso Ni-Cd o Ni-Mh). Dal momento che ogni singola cella ha una tensione nominale di 1,2 volt che diventano 1,6 a piena carica e quindi con 10 celle sono necessari ben 16 volt. Con una cella sono invece sufficienti 1,6 volt. Per questo motivo si è scelto un convertitore in configurazione Sepic il cui schema di principio è riportato in Figura B.

Nello schema sono evidenziati i segnali di maggiore im-



portanza, come la tensione di ingresso V_i e la tensione di uscita V_o . Il comando in PWM serve a controllare l'interruttore S che nel nostro circuito è un Mosfet di potenza. Sono presenti due condensatori ($C1$ e $C2$), un diodo (D) e le due induttanze $L1$ e $L2$. Nel nostro schema queste due induttanze sono realizzate sullo stesso nucleo magnetico e particolare attenzione deve essere posta nella scelta dei

vari componenti, dovendo lavorare a frequenze di decine di kHz. Tralasciando lo studio matematico dello schema, che esula dai propositi dell'articolo, possiamo riportare in queste righe le considerazioni finali. La tensione di uscita dipende dalla tensione di ingresso e dal valore del Duty-Cycle (δ) del PWM relativo alla chiusura e apertura dell'interruttore S secondo l'equazione qui a fianco ed il cui grafico è rappresentato in Figura C.

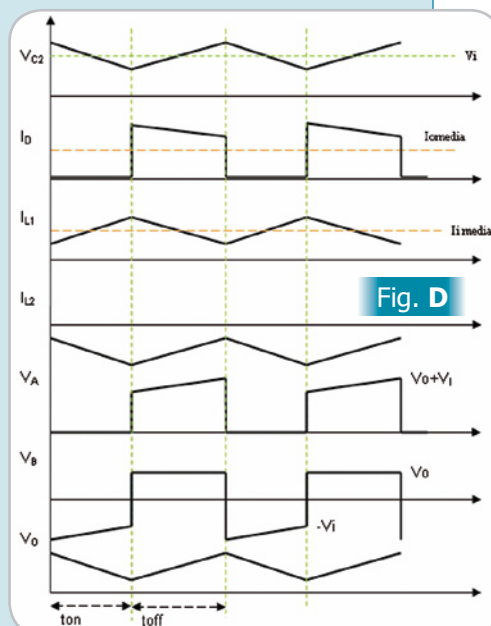
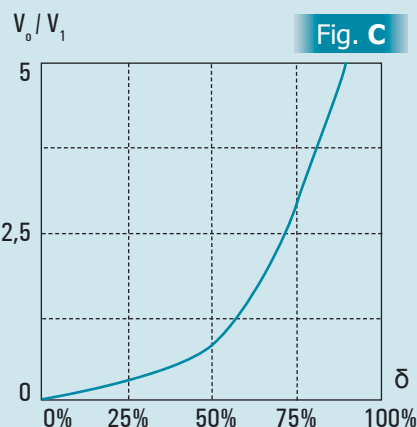
$$V_o = \frac{\delta}{1-\delta} \cdot V_i$$

Risulta evidente che per valori del duty-cycle inferiori al 50% la tensione di uscita è inferiore della tensione di ingresso, viceversa per valori maggiori al 50% la tensione di uscita risulta superiore tendendo all'infinito (teoricamente) per duty-cycle prossimi al 100%.

Notate una cosa molto importante: questo convertitore necessita di un solo interruttore e quindi di una sola linea di comando; ciò permette di semplificare i collegamenti con la logica di controllo.

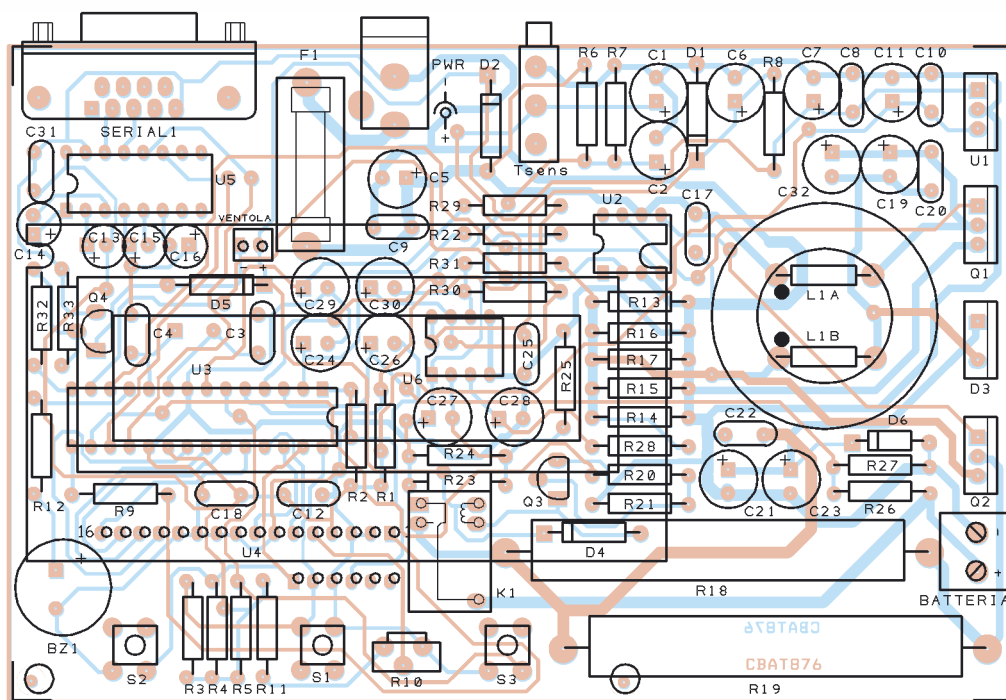
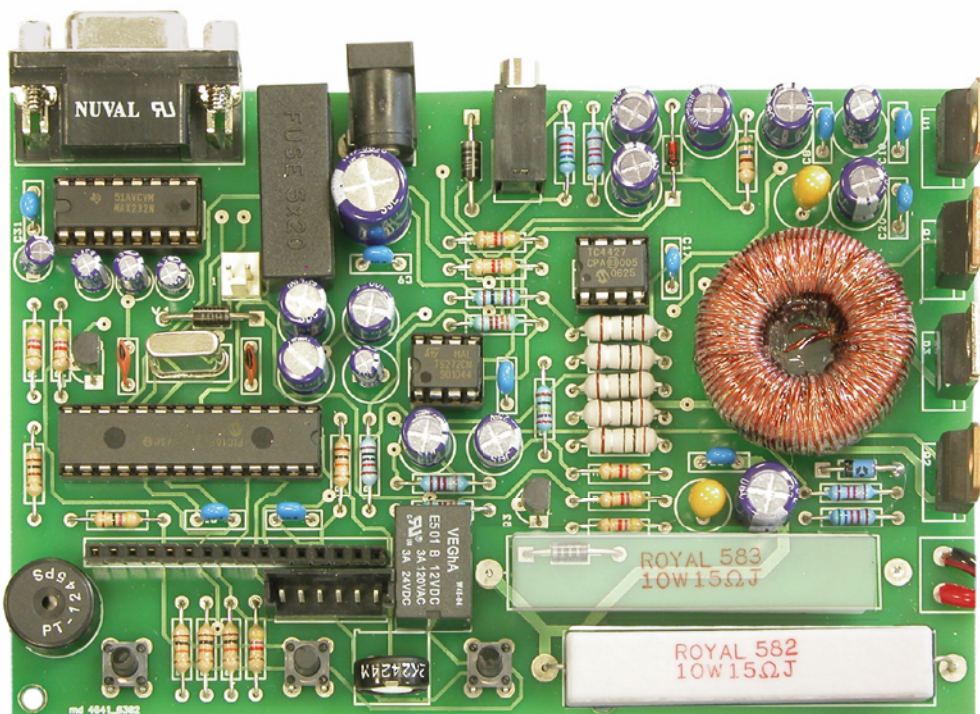
Il convertitore Sepic, però, offre altri vantaggi: ad esempio la sua tensione di uscita ha la stessa polarità della tensione di ingresso e mantiene lo stesso riferimento di massa. Possiamo anche citare il fatto che la corrente assorbita in ingresso presenta un'ondulazione assai ridotta, e ciò a vantaggio della sorgente di alimentazione, non sottoposta a stressanti picchi di assorbimento.

A titolo puramente informativo, riportiamo in Figura D l'andamento temporale dei segnali dello schema di principio. Il rendimento complessivo di questo convertitore è molto buono: le perdite maggiori sono presenti nell'interruttore S (nel nostro caso un MOSFET, con relativa resistenza $R_{DS(ON)}$), nelle induttanze (a causa della resistenza dell'avvolgimento) e nel diodo in uscita (per via della sua caduta di tensione). Generalmente la frequenza del segnale PWM di controllo è molto elevata e quindi occorre assicurare una velocità di commutazione dell'interruttore S molto rapida: il diodo è del tipo Schottky (Fast recovery time) ed i condensatori devono avere una bassa resistenza interna (ESR): ideali sono gli elementi al tantalio.



ELENCO COMPONENTI:

R1: 10 kohm 1%
 R2+R5: 10 kohm
 R6: 56 kohm 1%
 R7: 18 kohm 1%
 R8: 560 ohm
 R9: 47 ohm
 R10: Trimmer 4,7 kohm
 R11: 4,7 kohm
 R12: 100 ohm
 R13+R17: 1 ohm 1/2W
 R18, R19: 15 ohm 10W
 R20+R22: 4,7 kohm
 R23: 3,3 kohm 1%
 R24: 39 kohm 1%
 R25: 270 ohm 1%
 R26: 56 kohm 1%
 R27: 18 kohm 1%
 R28, R29: 4,7 kohm
 R30: 3,3 kohm 1%
 R31: 270 ohm 1%
 R32, R33: 4,7 kohm
 C1, C2: 100 µF 25 VL
 elettrolitico
 C3, C4: 22 pF ceramico
 C5: 470 µF 25 VL
 elettrolitico
 C6, C7: 100 µF 25 VL
 elettrolitico
 C8+ C10: 100 nF multistrato
 C11: 10 µF 25 VL elettrolitico
 C12: 100 nF multistrato
 C13+ C16: 1 µF 25 VL
 elettrolitico
 C17, C18: 100 nF multistrato
 C19: 100 µF 25 VL tantalio
 C20, C22: 100 nF multistrato
 C21: 100 µF 25 VL tantalio
 C23: 220 µF 25 VL elettrolitico
 C24: 100 µF 25 VL elettrolitico
 C25, C31: 100 nF multistrato
 C26: 1 µF 25 VL elettrolitico
 C27+ C30, C32: 100 µF 25 VL
 elettrolitico
 D1: Zener 8,2V 1/2W
 D2, D4, D5: 1N4007
 D3: PBYR745
 D6: Zener 24V 1/2W
 Q1, Q2: IRF540
 Q3, Q4: BC547
 U1: 7805
 U2: TC4427CPA
 U3: PIC16F876A (MF705)
 U4: Display LCD CDL4162

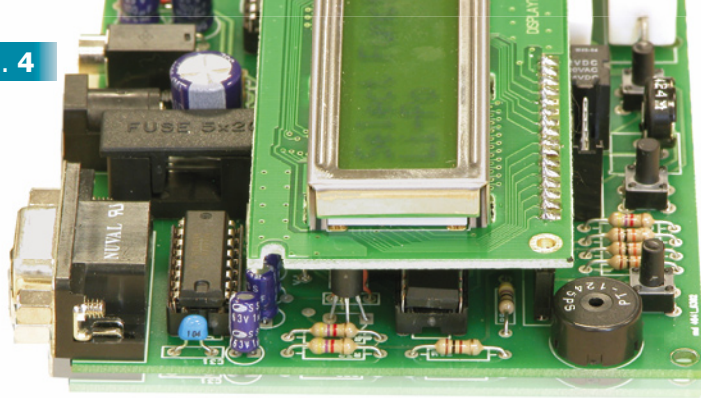


U5: MAX232
 U6: TS272CN
 K1: Relé 12V 2A MZ-12HS,
 JH4101F
 BZ1: Buzzer con elettronica
 S1+S3: Microswitch
 F1: Portafusibile 5x20 da cs
 + fusibile 3,15A
 Y1: Quarzo 20MHz
 L1: Nucleo toroidale

27 x 15 x 12mm (vedi testo)
 Tsens: NTC 10K
 Varie:
 - plug alimentazione
 - Presa Jack 3,5 mm da CS
 - Spina jack 3,5 mm volante
 - Strip maschio 2 poli
 - Strip maschio 6 poli
 - Strip femmina 16 poli

- Connettore DB9 femmina CS
 - Boccia 4mm nera
 - Boccia 4mm rossa
 - Zoccolo 4+4 (2 pz.)
 - Zoccolo 8+8
 - Zoccolo 14+14
 - Ventola 40x40x10 12Vdc
 - 10 metri filo di rame smaltato
 da 0,8mm
 - Circuito stampato

Fig. 4



si ritenga opportuno. L'interfaccia seriale, realizzata con un convertitore di livello MAX232, permette lo scambio dati bidirezionale con un personal computer.

Il connettore ICSP, infine, consente la connessione di un programmatore per l'aggiornamento diretto del firmware del PIC.

Realizzazione pratica

Vista la complessità circuitale, spiegheremo in dettaglio le fasi relative alla realizzazione del circuito. Per prima cosa è necessario realizzare il circuito stampato doppia faccia partendo dal master il cui file è scaricabile gratuitamente dal sito della rivista www.elettronica.in.it. Dovendo ottenere un circuito compatto, la densità dei componenti è notevole e quindi lo stampato deve essere realizzato con la massima precisione in modo che anche le piste più sottili (le più ostiche) risultino ben definite. Il materiale utilizzato è di facile reperibilità, resistenze di precisione a parte; se non fossero disponibili le resistenze con tolleranza dell'1%, è possibile selezionare quelle al 5% più precise, in quanto abbiamo usato solo valori standard (E12). Assicuratevi che tutti i componenti siano compatibili con lo stampato e con lo spazio a disposizione. Fate molta attenzione al giusto valore delle resistenze, pena il malfunzionamento del circuito che in fase di carica potrebbe subire danneggiamenti o rovinare le batterie. Il connettore per l'alimentazione e il sensore di temperatura potrebbero necessitare di qualche modifica al diametro del foro nello stampato a seconda della casa produttrice del componente. Il montaggio inizia dai fori passanti che collegano le piazzole del lato superiore con quelle del lato inferiore e prosegue con la saldatura delle resistenze di bassa potenza. Le resistenze R13÷R17 vanno saldate distanti qualche millimetro dal circuito stampato, per favorire la circolazione dell'aria. Anche le due

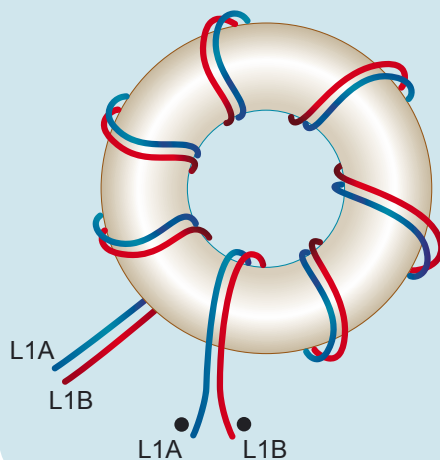
resistenze di potenza (R18 ed R19) devono essere saldate a qualche millimetro dallo stampato. Sotto la resistenza R18 è posizionato il diodo D4, che pertanto deve essere saldato prima della resistenza. Proseguite saldando i condensatori, i connettori e gli zoccoli degli integrati; infine i componenti in formato TO220, avendo l'accortezza di inserirli completamente sino alla tacca di riferimento presente sui reofori. Il diodo di potenza può essere sostituito da un diodo equivalente purché del tipo Schottky con bassa caduta di tensione in diretta; anche i mosfet di potenza possono essere sostituiti con modelli equivalenti purché a bassa resistenza diretta (RDS-ON), quali i modelli STP80NE e IRL2203. Quelli adottati in questo progetto offrono un buon rapporto qualità/prezzo e sono di facile reperibilità.

Per la saldatura del connettore del display, che deve stare ad una altezza di 17-18 millimetri dallo stampato, fate riferimento alla figura 4.

Particolare attenzione deve essere posta nella realizzazione dell'induttanza. Procuratevi un nucleo toroidale di misure 27x15x12 mm ed uno spezzone di filo di rame smaltato del diametro di 0,8 mm della lunghezza di circa 10 metri. Tagliate a metà il filo di rame ed iniziate ad avvolgere il toroide mantenendo accoppiate le due metà del filo di rame. I nuclei migliori sono quelli in MolliPermalloy in quanto permettono di raggiungere il valore necessario di induttanza con un minore numero di spire. Comun-

La bobina L1

La bobina L1 è composta da due avvolgimenti realizzati sullo stesso nucleo. Le dimensioni di quest'ultimo sono di 27x15x12 mm. Per realizzare la bobina è necessario procurarsi due spezzoni di filo di rame smaltato del diametro di 0,8 mm e della lunghezza di circa 5 metri ciascuno. Mantenendo accoppiati i due fili, iniziate a realizzare la doppia bobina così come indicato nel disegno. Cercate di avvol-



gere le spire nella maniera più ordinata possibile. A lavoro ultimato l'induttanza si presenta come nella fotografia. Per evitare vibrazioni, è consigliabile incollare le spire dell'induttanza con della colla a caldo. Occorre identificare in modo chiaro i due avvolgimenti nonchè l'inizio e la fine di ciascuna bobina (l'inizio viene contraddistinto da un pallino in tutti gli schemi). In fase di cablaggio è necessario inserire i terminali nelle piazzole corrette, pena il mancato funzionamento dello switching.

que è essenziale realizzare un'induttanza di almeno 300 μH (meglio se maggiore) con la più bassa componente resistiva. Nel riquadro di pagina 81 sono disponibili tutti i dettagli costruttivi. La saldatura sul circuito stampato deve avvenire rispettando scrupolosamente il giusto verso come indicato nella figura 5 in cui i due pallini neri indicano i punti dove devono essere inseriti i due terminali iniziali degli avvolgimenti. Fissate l'induttanza colando un po' di colla a caldo al suo interno, facendo in modo che giunga fino al circuito stampato. A lavoro ultimato controllate con la massima attenzione tutte le saldature e la disposizione dei componenti, per avere la certezza che tutto sia in ordine. Inserite il PIC programmato (oppure vergine: in questo caso lo potete programmare tramite il connettore ICSP) e sopra di esso il display LCD al quale avrete saldato l'apposito connettore. Inserite il fusibile sul portafusibile, magari di valore inferiore, diciamo 500 mA in modo da salvaguardare il circuito nel caso in cui qualche errore sia sfuggito all'esame visivo. Successivamente, dopo che avrete verificato il corretto funzionamento, lo sostituirete con quello di valore nominale. A questo punto alimentate il circuito prelevando la tensione di 12 volt da un alimentatore anche non stabilizzato oppure da un semplice alimentatore universale da almeno 1 ampere regolato per una tensione di 12 V. Il funzionamento alla massima potenza si ottiene ca-

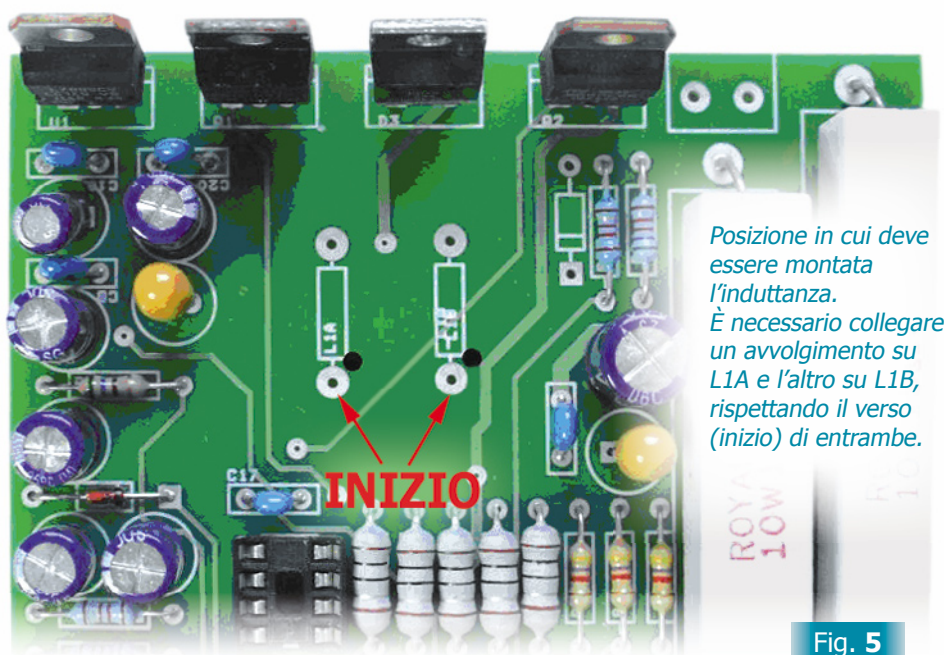


Fig. 5

ricando il massimo numero di celle con la massima corrente, situazione in cui l'assorbimento è di circa 3,5 ampere. In questo caso è necessario prevedere un'adeguata fonte di alimentazione. Consigliamo vivamente di non alimentare il circuito senza tutti i componenti installati ed il PIC debitamente programmato. Se si vuole verificare lo stato delle alimentazioni prima dell'inserimento dei circuiti integrati è fondamentale che non siano presenti i due MOSFET di potenza. L'unica regolazione presente riguarda il contrasto del display LCD che può essere effettuata agendo sul trimmer R10. Le batterie da ricaricare vanno collegate alla morsettiera "BATTERIA". È possibile utilizzare il circuito così com'è a patto di inserire un dissipatore sui componenti con "case" TO220 (U1, Q1,

Q2, D3) nel caso in cui lo si utilizzi per correnti superiori a 0,5 ampere. Come già detto, la disposizione di questi componenti è stata studiata per permettere di utilizzare facilmente un dissipatore al quale vanno fissati (opportunamente isolati con foglietti di mica e distanziali plastici) i componenti TO220. A lavoro terminato il circuito si presenta come visibile nel riquadro del piano di montaggio.

Conclusione

Ora che avete ultimato il montaggio, siete sicuramente ansiosi di utilizzare il vostro nuovo caricabatterie. Per intanto vi chiediamo di non farlo e di pazientare fino al prossimo fascicolo nel quale vi spiegheremo come utilizzarlo e come inserirlo in un elegante contenitore plastico. Al prossimo mese!

per il

MATERIALE

I componenti utilizzati in questo progetto sono facilmente reperibili in commercio. Il master del circuito stampato, il programma di controllo per PC (Cbat876.exe) ed il firmware per programmare il micro sono scaricabili gratuitamente dal sito della rivista (www.elettronica.in.it). Il microcontrollore è disponibile anche già programmato al prezzo di 18,00 Euro (cod. MF705) 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>

Mambo, la prima applicazione pratica

di Alessandro Orlando



Ecco il primo esempio di programmazione ed utilizzo: un semplice AVL (Automotive Vehicle Locator) in grado di informare via SMS, ad intervalli prestabiliti, la posizione del veicolo, se questo è fermo oppure in movimento, l'eventuale velocità e se manca il segnale GPS.

Nella precedente puntata abbiamo familiarizzato con la sintassi dei comandi utilizzati dal sistema di localizzazione MAMBO per eseguire specifiche operazioni: durante la spiegazione dei vari comandi, abbiamo anche visto qualche semplice esempio.

In questa puntata entriamo più nel dettaglio, realizzando ed analizzando un eseguibile scritto sulla base di precise specifiche: in altre parole, definiamo le specifiche, scriviamo il programma di configurazione per il Mambo e commentiamo dettagliatamente ogni singola istruzione. Per partire con il piede

giusto, la prima cosa da fare, dopo avere letto il manuale del Mambo in tutta le sue parti ed aver acquisito un minimo di confidenza con i vari esempi di script di cui il manuale stesso è pieno, è quella di stabilire con precisione tanto la modalità di funzionamento del dispositivo quanto come questi si debba comportare in determinate situazioni quali, ad esempio, l'assenza della copertura GSM, la mancanza del segnale GPS, la batteria quasi scarica, le connessioni da utilizzare in funzione dei comandi ricevuti sia in locale che da remoto, eccetera. Più semplicemente, occorre definire "le specifiche di >

progetto” che debbono prendere in considerazione tutte le possibili situazioni che si possono presentare in quella applicazione.

Dopo avere stabilito come il sistema deve funzionare (ossia cosa ci si aspetta dal Mambo durante il suo funzionamento) si deve disegnare, esattamente come si fa, ad esempio, per gli eseguibili scritti per il PC o per i microcontrollori, un diagramma di flusso che sintetizza le istruzioni del programma e determina per l'appunto le azioni da compiere in base alle nostre specifiche ed agli eventi.

A questo punto riprendiamo l'esempio della precedente puntata la cui dichiarazione (listato) - visibile in tabella 1 - svolge la funzione di registro di posizione: in caso di se-

(. **GPS.History.Write,0,""**) quando il Timer ha finito di contare (**Timer.e0**), ovvero ogni 10 secondi.

Il diagramma di flusso del programma appena descritto è visibile in figura 1: anche se abbiamo avuto modo più volte di parlarne, ripetiamo che il listato eseguito dal Mambo viene scritto nel file di configurazione rispettando la sintassi dei comandi; una volta caricato il file di configurazione, l'interprete del localizzatore si occupa di tradurre le istruzioni in azioni. Dal semplice esempio appena esposto, appare evidente che la struttura dell'eseguibile (leggasi “del file di configurazione”) memorizzato all'interno del Mambo deve necessariamente rispondere a delle precise stringhe, la cui sequenza logica tiene conto di tutte le possibili

condizioni di allarme (ricordate che tutti gli eventi controllati dal Mambo sono

gestiti come allarmi) al fine di non incorrere nel blocco improvviso del programma in cicli di loop infiniti, che attendono il verificarsi di un allarme che non si può di fatto verificare in quanto non previsto. Notate che quanto appena esposto rappresenta il punto di partenza per la stesura dell'esempio di

programma vero e proprio, oggetto di questo articolo.

Il programma completo

Il miglior esempio applicativo è, normalmente, quello con il maggior senso logico, quindi abbiamo ipotizzato di voler controllare la posizione di un veicolo ad intervalli regolari e di visualizzarla su un programma cartografico, come ad esempio il Fugawi o altri.

Poiché la stringa NMEA0183 RMC è quella normalmente letta sia dai software cartografici che dai siti che permettono di visualizzare un target all'interno di una mappa (www.mapquest.com), ai fini dell'esempio pratico è necessario capirne la struttura e la composizione in maniera tale da poter estrapolare i dati necessari al programma. La tabella 2 riassume tutti i campi dati della sentenza RMC: essa contiene tutte le informazioni che definiscono la navigazione del sistema, ossia latitudine e longitudine, data e ora, direzione e velocità, validità della posizione calcolata, più altri parametri quali elevazione, variazione magnetica e checksum. Si tratta di una serie di informazioni molto completa anche se, nel nostro caso, i dati utilizzati sono la data, l'ora, la velocità, la latitudine e la longitudine. Di tutti questi dati, quelli da utilizzare per la visualizzazione della posizione

Tabella 1 - Log di posizioni valide

```
$PFAL,Cnf.Set,AL0=GPS.Nav.eFix=valid:Sys.Timer0.Start=ciclic,10000
$PFAL,Cnf.Set,AL1=GPS.Nav.eFix=invalid:Sys.Timer0.stop
$PFAL,Cnf.Set,AL2=Timer.e0:GPS.History.Write,0,""
```

gnale GPS valido viene avviato un contatore da 10 secondi allo scadere del quale la posizione rilevata viene scritta in memoria. A questo punto il programma esegue un nuovo ciclo e così via, all'infinito. Se il segnale GPS non è valido, il timer non viene avviato e di conseguenza la posizione non viene memorizzata.

Con la prima dichiarazione (**AL0**) si comunica al localizzatore che esso deve avviare un timer ciclico della durata di 10 sec. (**Timer0.Start=ciclic,10000**), nel caso in cui il segnale sia valido (**GPS.Nav.eFix=valid**).

Con la seconda (**AL1**) viene forzato lo spegnimento del timer ciclico a 10 secondi nel caso in cui il segnale non sia valido (**GPS.Nav.eFix=invalid:Sys.Timer0.stop**).

Con la terza ed ultima dichiarazione, invece, il localizzatore viene forzato a salvare la posizione GPS nello storico della memoria

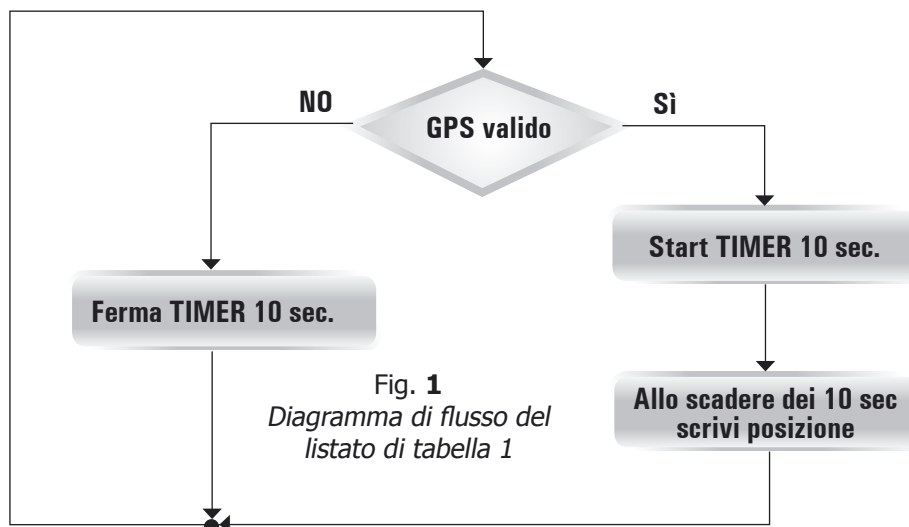


Fig. 1
Diagramma di flusso del
listato di tabella 1

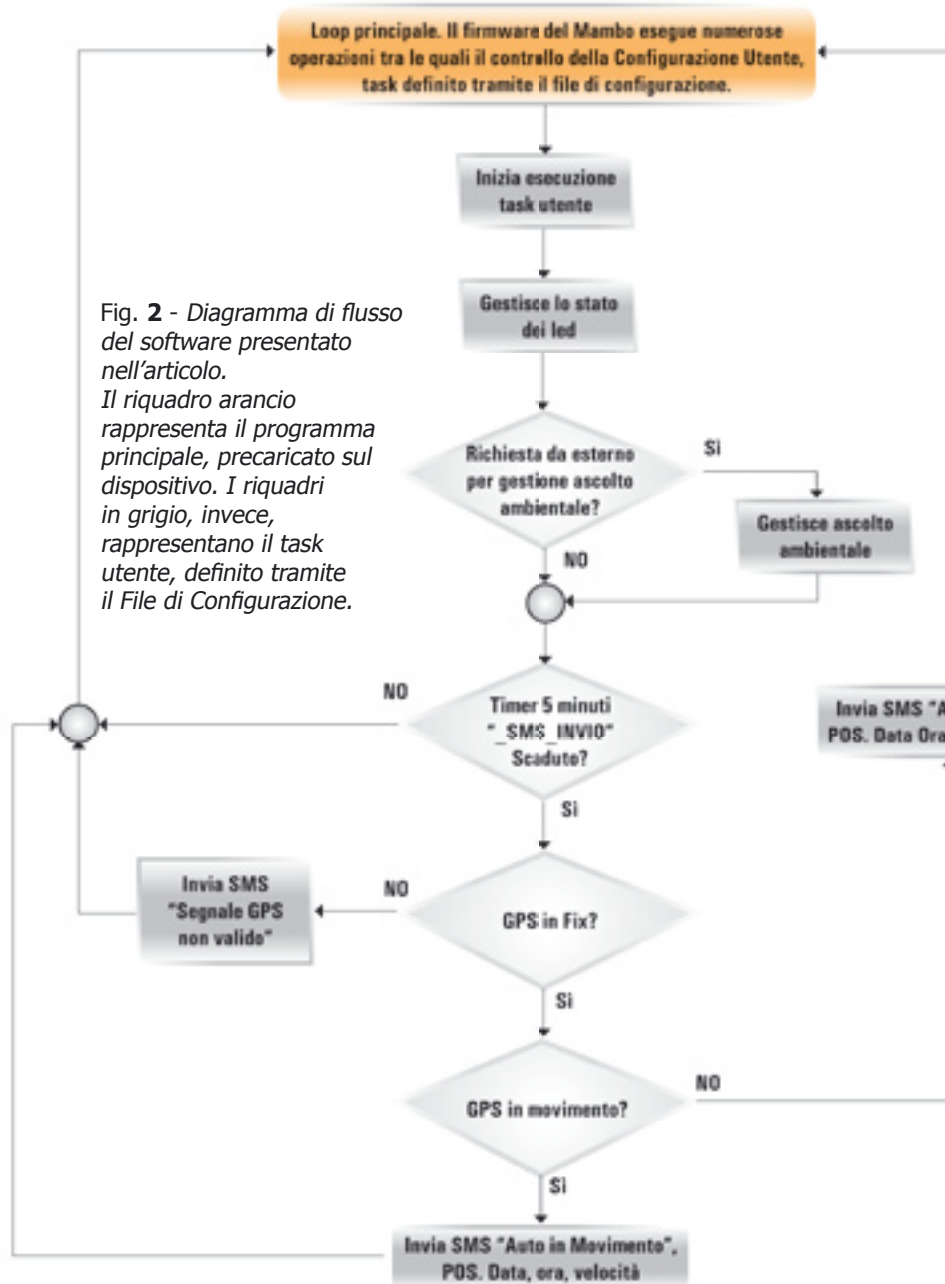


Fig. 2 - Diagramma di flusso del software presentato nell'articolo.

Il riquadro arancione rappresenta il programma principale, precaricato sul dispositivo. I riquadri in grigio, invece, rappresentano il task utente, definito tramite il File di Configurazione.

loop principale del programma ossia quella parte (la maggior parte!) di software che controlla e gestisce lo stato ed il funzionamento di tutte le periferiche (GPS, GSM, LED, pulsanti, interfaccia Bluetooth, codec audio, microfono, altoparlante ecc.) di cui Mambo dispone. Tra le tante cose, il microprocessore dedica un certo tempo alla gestione della Configurazione Utente, legata al file di configurazione ed argomento principale di questo articolo. A tempo debito, il microprocessore avvia il "task utente (file di configurazione)" in cui

la prima operazione effettuata è quella relativa allo stato dei led. Immediatamente dopo, il localizzatore verifica se è in corso una chiamata per l'attivazione ambientale: in caso affermativo, essa viene gestita. In ogni caso, il programma prosegue verificando se il timer che scandisce l'invio degli SMS (`_SMS_invio`, del valore di 5 minuti), è scaduto. Se non lo è, il task utente termina ed il programma rilascia il controllo al loop principale. Se lo è, invece, il programma controlla lo stato del segnale GPS, che può essere valido oppure no. Nel caso di un segnale non valido, il programma invia un SMS contenente tale

informazione. In presenza di segnale valido, invece, il programma esegue un ulteriore controllo, ossia verifica la velocità del localizzatore (o del veicolo in cui esso è installato). In funzione della velocità, il programma decide di informare l'utente remoto del fatto che il veicolo è in sosta oppure in movimento. In ogni caso l'invio di un SMS determina la fine dell'esecuzione del task utente ed il ripristino del controllo da parte del loop principale.

Bene: partendo dalle specifiche di progetto, abbiamo ricavato il dia-

con MapQuest o Fugawi sono solo la latitudine e la longitudine. Vista la sentenza NMEA utilizzata per il programma, stabiliamo ora le specifiche del progetto. Si vuole che il sistema svolga le seguenti funzioni:

1. Invio di un SMS ogni 5 minuti, nel caso di segnale GPS valido; l'SMS contiene data, ora, latitudine, longitudine, velocità.
2. In caso di segnale GPS non valido, invio di un SMS con l'indicazione di segnale non valido.
3. Nel caso in cui il Mambo non sia in movimento, invio di un SMS

contenente data, ora, latitudine, longitudine; la velocità, in questo caso, è inutile.

4. Il Mambo non deve emettere alcun tipo di tono, ma deve segnalare lo stato di funzionamento sia del GPS che del GSM.
5. In caso di chiamata dall'esterno deve rispondere e porsi in modalità di ascolto ambientale.

Avendo definito le specifiche di progetto, a questo punto è possibile realizzare il diagramma di flusso del software, visibile in figura 2. Il primo riquadro in alto rappresenta il

Tabella 2 - Descrizione dei campi della sentenza RMC

Campo	Formato	Esempio	Descrizione
1	\$GPRMC	\$GPRMC	Stringa iniziale
2	hhmmss.ss (Numerico)	133725.569	Ora UTC
3	a (Carattere)	Posizione valida	(A: valida, V: non valida)
4	ddmm.mmmm (Numerico)	5040.4365	Latitudine
5	a (Carattere)	N	Direzione Latitudine (N/S)
6	dddmm.mmmm (Numerico)	01058.5650	Longitudine
7	a (Carattere)	E	Direzione Longitudine (W/E)
8	x.x (Numerico)	0.05	Velocità (Nodi)
9	x.x (Numerico)	302.98	Heading (Gradi)
10	ddmmyy (Numerico)	251004	Data
11	x.x (Numerico)	Non utilizzato	Variazione Magnetica (Gradi)
12	a (Carattere)	Non utilizzato	Variazione direzione magnetica (W/E)
13	*hh (Esadecimale)	*00	Checksum
14	< CR > < LF >		Stringa finale del messaggio

gramma di flusso necessario allo scopo; vediamo ora di scrivere il programma.

Partendo dalla configurazione di base, precaricata nel dispositivo e ricavabile dopo il comando *\$PFAL,Sys.Device.FactoryReset*, abbiamo introdotto delle sostanziali modifiche, fino ad ottenere il programma di configurazione elencato nella tabella 3.

Innanzitutto è stato definito un contatore denominato *_SMS_invio*, il

quale viene decrementato di una unità ogni volta che il *Timer.e_1_sec* va a zero. Una volta raggiunto il valore "0", il contatore *_SMS_invio* viene riprogrammato a 300. Serve per definire l'intervallo di 5 minuti per l'invio degli SMS (per l'appunto 300/60 secondi equivale a 5 minuti). Abbiamo detto che il contatore *_SMS_invio* deve essere riprogrammato a 300 dopo che ha raggiunto il valore 0: questa operazione è ottenuta con la linea di comando de-

scritta in tabella 3 alla linea 1. Il decremento di una unità a cadenza di 1 secondo, invece, viene effettuato con il comando visibile nella linea 2: tramite esso, il Mambo decrementa *_SMS_invio* ogni volta che scade *TIMER_1SEC*.

Nel comando visibile alla linea 3, invece, è stato definito il contatore Counter3 come Alias di *_SMS_INVIO* (vedere il riquadro relativo agli Alias). Quando il contatore *_SMS_INVIO* va a zero, esso genera un evento software riconoscibile dal sistema operativo del localizzatore: questo evento, unitamente al controllo del Segnale GPS ed al fatto che l'autovettura sia in sosta oppure in movimento, fa sì che il sistema generi ulteriori allarmi, dai quali dipendono azioni specifiche verso il mondo esterno. Vediamo ora in dettaglio gli allarmi inseriti per il controllo degli SMS: il primo di essi è riportato nella linea 4.

Quando il contatore *SMS_INVIO* è a zero (significa che sono passati 5 minuti dall'evento precedente) ed il segnale GPS è valido, il localizzatore invia un SMS al numero predefinito, SMS che contiene informazioni su data, ora, latitudine, longitudine e velocità espressa in metri al secondo. Quando invece il segnale GPS non è valido (linea 5), viene inviato un messaggio con la relativa informazione "Segnale GPS non Valido". Il terzo ed ultimo allarme di questa serie, serve per inviare un messaggio per informare che l'autovettura è in sosta, pertanto invia le coordinate, la data e l'ora ma non la velocità, informazione inutile quando il veicolo è fermo. A questo punto, inserendo ad esempio le coordinate nel sito www.mapquest.com (oppure nel software Fugawi), è possibile verificare in tempo reale la posizione del veicolo. Visti i comandi utilizzati per gestire la messaggistica, vediamo ora l'uso degli allarmi nelle segnalazioni luminose tramite i led, segnalazioni effettuate secondo un

Tabella 3
Elenco comandi principali usati nel File di Configurazione di test

1	\$PFAL,CNF.Set,AL4=SYS.COUNTER.e_sms_invio:sys.counter_sms_invio.set=300
2	\$PFAL,CNF.Set,AL5=SYS.TIMER.e_1SEC:sys.counter_sms_invio.decrement=1
3	\$PFAL,CNF.Set,ALIAS.SYS.COUNTER3=_SMS_INVIO
4	\$PFAL,CNF.Set,AL6=SYS.COUNTER.e_SMS_INVIO&GPS.Nav.sFix=valid:GSM.SMS.SEND,"+399991234567",0,"Auto in Movimento: Data &(Date) Ora &(Time) Latitudine &(Lat) Longitudine &(Lon) Velocità &(Speed)"
5	\$PFAL,CNF.Set,AL7=SYS.COUNTER.e_SMS_INVIO&GPS.Nav.sFix=invalid:GSM.SMS.SEND,"+399991234567",0,"Segnale GPS non Valido"
6	\$PFAL,CNF.Set,AL16=SYS.COUNTER.e_SMS_INVIO&GPS.Nav.sFix=valid&IO.MOTION.sStanding:GSM.SMS.SEND,"+399991234567",0,"Auto in Sosta: Data &(Date) Ora &(Time) Latitudine &(Lat) Longitudine &(Lon)"
7	\$PFAL,CNF.Set,AL9=SYS.COUNTER.e_TIMING_MODE&GPS.Nav.sFix=valid:IO.LED_GPS.green=hpulse,400&IO.LED_GPS.red=low&IO.LED_GPS.blue=low
8	\$PFAL,CNF.Set,AL10=SYS.COUNTER.e_TIMING_MODE&GPS.Nav.sFix=invalid:IO.LED_GPS.green=low&IO.LED_GPS.red=hpulse,400&IO.LED_GPS.blue=low
9	\$PFAL,CNF.Set,AL11=SYS.COUNTER.e_TIMING_MODE&GSM.sOpValid:IO.LED_GSM.green=hpulse,400&IO.LED_GSM.red=low&IO.LED_GSM.blue=low
10	\$PFAL,CNF.Set,AL12=SYS.COUNTER.e_TIMING_MODE&GSM.sOpInvalid:IO.LED_GSM.green=low&IO.LED_GSM.red=hpulse,400&IO.LED_GSM.blue=low
11	\$PFAL,CNF.Set,AL14=SYS.COUNTER.e_TIMING_MODE&SYS.BT.sConnected:IO.LED_STATUS.green=low&IO.LED_STATUS.red=low&IO.LED_STATUS.blue=hpulse,400
12	\$PFAL,CNF.Set,AL13=GSM.voiceCall.eIncoming:GSM.VoiceCall.Accept

Aggiornare il file di configurazione

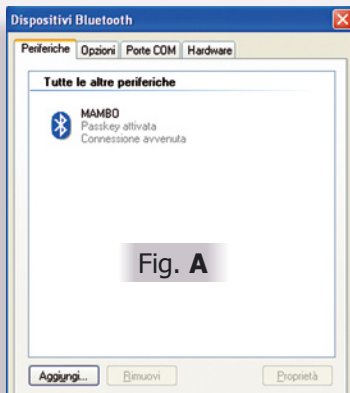


Fig. A

L'aggiornamento del file di configurazione prevede l'uso del software allegato al prodotto, denominato Workbench (tradotto alla lettera: il tavolo da lavoro). Una volta aggiunto al PC il nuovo dispositivo Bluetooth (nel nostro caso il MAMBO) e verificata la porta seriale virtuale che il sistema ha assegnato ad esso (Figura A), è necessario avviare il Workbench e connettere il sistema al Mambo facendo clic sulla prima icona in alto a sinistra (Figura B) denominata PORT (porta della periferica).

Per verificare se il Mambo

è connesso, si apre la finestra "Console" (dal menu Plugins) quindi la si collega facendo clic sul pallino "Console" della finestra "Connections": il funzionamento è confermato dal flusso di stringhe che appaiono sulla Console (Figura B). A questo punto occorre aprire e connettere la finestra "Editor" di Figura C (tramite il menu Plugins), sulla quale dobbiamo digitare ed inviare (facendo clic sull'icona "freccia in diagonale verso l'alto") la seguente stringa: **\$PFAL,Sys.device.FactoryReset**. Tramite questa dichiarazione, forziamo il MAMBO a ricaricare al suo interno la configurazione di default. Avendo azzerato la periferica, essa riparte dallo stato di reset, pertanto si potrebbe avere qualche problema di riconnessione automatica: conviene quindi abbattere la connessione PORT, togliere la batteria del MAMBO, reinserirla, riavviare il sistema e ristabilire la connessione tramite la procedura già descritta.

Una volta riconnesso il tutto, occorre visualizzare e collegare la finestra di Editor, nella quale apriamo il nostro file di configurazione (tale file è scaricabile dal sito della rivista): inviamo il file al Mambo facendo clic sull'icona già usata in fase di "default reset" (vedere figura D): notate che durante il trasferimento la "freccia" si trasforma in una "X". Alla fine dell'operazione conviene sempre scollegare il programma e rimuovere temporaneamente la batteria del Mambo, in modo da riavviare il sistema con un "Hard Reset". Riconnettendo il localizzatore ed aprendo la finestra Console (Figura E) vediamo fluire la serie di eventi che confermano il funzionamento della nuova configurazione. La procedura può sembrare un po' complessa, ma alla fine scoprirete che basta un po' di pratica.

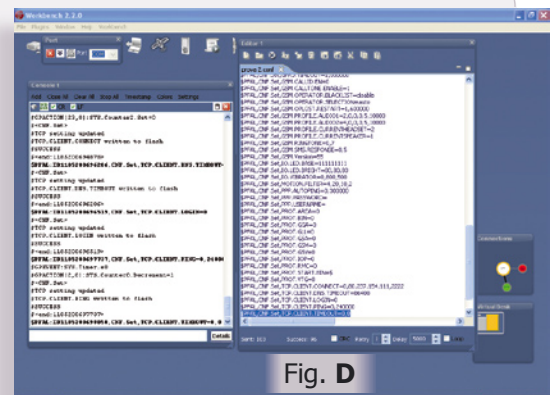


Fig. D

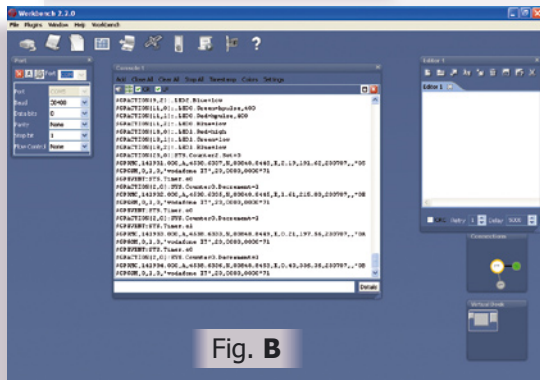


Fig. B

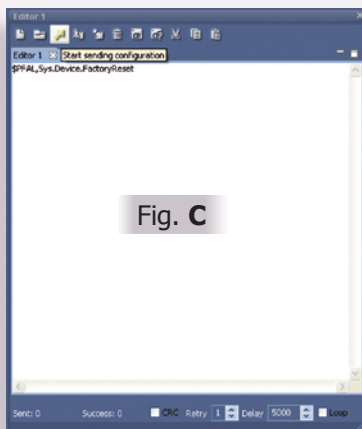


Fig. C

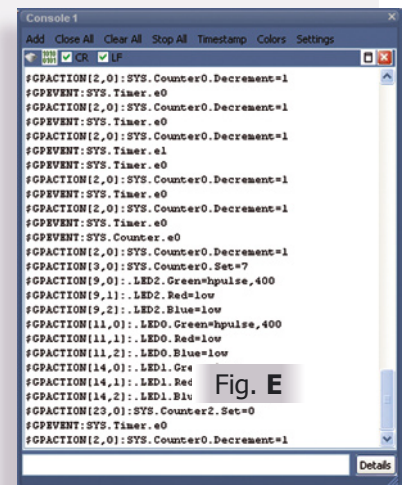


Fig. E

criterio logico e facilmente intuibile ossia facendo lampeggiare il led del GPS e quello del GSM a luce verde nel caso di segnale GPS o copertura GSM validi, e a luce rossa nel caso opposto. La connessione Bluetooth, invece, è indicata dal lampeggio del led blu. Di seguito riportiamo i riferimenti alle linee di comando in tutti i casi previsti:

- segnale GPS valido = led verde a luce pulsante (linea 7);
- segnale GPS non valido = led rosso acceso a luce pulsante (linea 8);

- copertura del servizio GSM = led verde acceso a luce pulsante (linea 9);
- mancanza di copertura del servizio GSM = led rosso acceso a luce pulsante (linea 10);
- Bluetooth connesso = led blu lampeggiante (linea 11).

Come potete notare, le segnalazioni in questo caso sono molto intuitive, al punto che basta il colpo d'occhio per conoscere lo stato del Mambo. Un ulteriore allarme è stato inserito per gestire le chiamate in ingresso, alle quali il Mambo risponde auto-

maticamente avviando la funzione di ascolto ambientale, ossia facendo ascoltare (tramite il cellulare che ha effettuato la chiamata) quanto avviene nell'ambiente in cui il localizzatore è stato posizionato. Il comando con cui si avvia l'ambientale è visibile in linea 12: se nel comando venisse inserito il numero del chiamante, il Mambo accetterebbe solamente le chiamate provenienti da tale numero, ignorando tutte le altre. Prestate attenzione ad un particolare molto importante: il listato che ab-

Cos'è un Alias

Col termine "Alias" si intende un nome scelto dal programmatore ed assegnato ad un particolare contatore, timer, pulsante, Led, eccetera; l'alias permette di ricordare più facilmente ed in modo intuitivo una certa "funzione" software avente un nome generico predefinito (Counter1, Timer2, LED1...). Parlando ad esempio del pulsante centrale del Mambo, si ricorda più facilmente "_MIDDLE" come alias piuttosto che BTN0 come nome predefinito. Pertanto, dichiarando in configurazione **\$PFAL,CNF.Set.Alias.IO.BTN0=_Middle**, il pulsante può essere richiamato tramite _MIDDLE piuttosto che BTN0. Di seguito trovate gli alias utilizzati nel File di Configurazione di questo articolo.

Estratto dal "File di Configurazione"

```
$PFAL,CNF.Set,ALIAS.CNF=Config
$PFAL,CNF.Set,ALIAS.IO.BTN0=_MIDDLE
$PFAL,CNF.Set,ALIAS.IO.BTN1=_UPPER
$PFAL,CNF.Set,ALIAS.IO.BTN2=_LOWER
$PFAL,CNF.Set,ALIAS.IO.BTN3=_HEADSET
$PFAL,CNF.Set,ALIAS.IO.LED0=_GSM
$PFAL,CNF.Set,ALIAS.IO.LED1=_STATUS
$PFAL,CNF.Set,ALIAS.IO.LED2=_GPS
$PFAL,CNF.Set,ALIAS.SYS.COUNTER0=_TIMING_MODE
$PFAL,CNF.Set,ALIAS.SYS.COUNTER1=_SLEEP_MODE
$PFAL,CNF.Set,ALIAS.SYS.COUNTER2=_BAT_MODE
$PFAL,CNF.Set,ALIAS.SYS.COUNTER3=_SMS_INVIO
$PFAL,CNF.Set,ALIAS.SYS.TIMER0=_1SEC
$PFAL,CNF.Set,ALIAS.SYS.TIMER1=_20SEC
$PFAL,CNF.Set,ALIAS.SYS.TIMER2=_SLEEP
$PFAL,CNF.Set,ALIAS.SYS.TIMER3=_DOUBLE
$PFAL,CNF.Set,ALIAS.SYS=System
```

biamo commentato in queste pagine non è quello completo; noi abbiamo voluto focalizzare la vostra attenzione sui comandi più importanti, ossia quelli che hanno un ruolo fondamentale nell'esecuzione delle procedure e che, proprio per tale ragione, presentano una sintassi lunga e complessa. Quello completo (che per motivi di spazio non è stato allegato in toto), contiene anche le istruzioni relative alla cancellazione e modifica di alcuni allarmi contenuti nella configurazione predefinita; esso può essere scaricato direttamente dal sito www.elettronica.in.it.

Conclusione

Con questo articolo abbiamo concluso la presentazione del primo vero programma per il localizzatore Mambo. Vi anticipiamo che ne abbiamo altri in cantiere, e tutti molto interessanti, quindi continuate a seguirci. A titolo puramente informativo, informiamo che la Falcom (casa produttrice del Mambo) ha di recente rilasciato una nuova release del firmware, proprio mentre noi stavamo scrivendo questo articolo. In questa sede, non abbiamo tenuto conto delle eventuali variazioni della nuova release, attualmente in fase di studio da parte nostra, la quale verrà discussa in seguito, in una puntata dedicata esclusivamente alle nuove funzioni introdotte.





comune di scandiano



27/28
OTTOBRE
2007
SCANDIANO
(REGGIO EMILIA)
CENTRO FIERISTICO

4^a FIERA REGIONALE DELL'ELETTRONICA

Orari:
sabato
9,00-18,30
domenica
9,00-18,00

- ◆ TELEFONIA
- ◆ COMPONENTISTICA
- ◆ COMPUTER
- ◆ HI-FI CAR
- ◆ RADIANTISMO
CB e OM
- ◆ VIDEOREGISTRAZIONE
- ◆ MERCATINO
DELLE PULCI
RADIOAMATORIALI

con il patrocinio
di A.R.I. sez. Scandiano

INFO:
0522/857436
www.fierascandiano.it

Biglietti:
Intero euro 7,00
Ridotto euro 4,00
Ragazzi fino a 12 anni gratuito



Liquid Crystal Display dalla teoria alla pratica

Impariamo a conoscere e utilizzare uno dei componenti elettronici più interessanti: il display LCD. In questa puntata analizziamo il controller grafico T6963C di Toshiba, un chip flessibile, potente e configurabile in base alle necessità. Settima puntata.

A CURA DI *Matteo Destro*

7

In questa puntata parliamo dei display 64240B e 128240A e del controller grafico che li governa: il T6963C, prodotto da Toshiba. Questo driver, apprezzato per la sua flessibilità, è largamente utilizzato sia da produttori di display che di apparati specifici dotati di visualizzazione grafica, ma non è integrato, come ad esempio il KS0713 visto nella puntata precedente: esso, infatti, necessita sia di una memoria RAM esterna che di un chip addizionale, al quale viene affidata la gestione dei pixel dell'LCD. In questo capitolo mettiamo in evidenza le sue potenzialità ed i comandi messi a disposizione per la gestione dei due display. Tenete presente che, per evitare di complicare la spiegazione, faremo riferimento solo al 64240B, ma quanto detto è perfettamente applicabile anche al modello 128240A: la sola diffe-

renza riguarda l'inizializzazione del controller il quale deve essere "adattato" alle differenti dimensioni del display, utilizzando la compilazione condizionata del software del PIC. Ritorneremo comunque su questo argomento tra qualche pagina.

Controller grafico T6963C

Il controller T6963C gestisce la comunicazione tra il microcontrollore presente sulla scheda madre e l'elettronica che controlla i pixel dell'LCD. In particolare esso interpreta i comandi del controllore e legge/scrive la memoria RAM, al fine di costruire le immagini sul vetrino dell'LCD, inviando dati e comandi specifici al "controllore grafico aggiuntivo". Il T6963C è in grado di indirizzare una memoria esterna di 64 kbyte. Tale memoria viene utilizzata sia per ➤

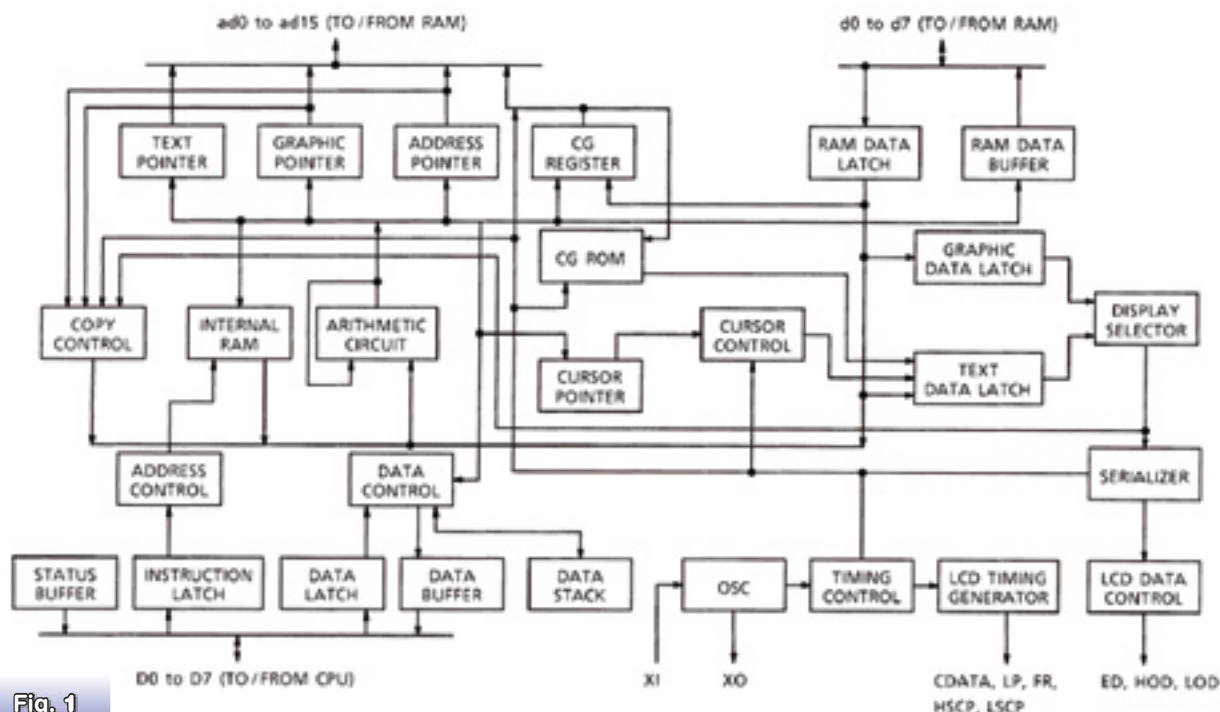


Fig. 1

memorizzare le informazioni del livello grafico sia quelle del livello testo. Infatti, a differenza dei chip trattati in precedenza, il T6963C permette di gestire sia la grafica che il testo: a seconda delle necessità esso miscela i due livelli, attraverso un'operazione booleana, ottenendo l'immagine desiderata. La memoria montata sul PCB del 64240 è di 32 kbyte, dimensione sufficiente per controllare i due livelli grafici e memorizzare i caratteri speciali non presenti nella CGROM del controller stesso. Questo concetto è identico a quello illustrato durante la trattazione dei display alfanumerici: una parte della memoria RAM viene utilizzata come CGRAM ovvero come memoria per la generazione dei caratteri speciali, mentre un'altra (CGROM) contiene i caratteri predefiniti. Dal punto di vista del testo, le caratteristiche del T6963C sono

le seguenti:

- numero di colonne gestibili: 32, 40, 64, 80;
- numero di linee gestibili: 2, 4, 6, 8, 10, 12, 14, 16, 20, 24, 28, 32;
- dimensione dei fonts: selezionabile tramite pin di controllo;
- numero di font: 128 font pre-memorizzati nella CGROM;
- dimensione RAM esterna: fino a 64 kbyte.

Lo schema a blocchi interno del controller è visibile in figura 1. Nella parte in alto troviamo i circuiti necessari per dialogare con la memoria esterna: le sedici linee di indirizzamento (A0÷A15) e il bus dati a 8 bit (D0÷D7). In basso a sinistra è visibile il blocco di interfaccia tra il T6963C e il microcontrollore della scheda madre. Anche in questo caso il bus dati è a 8

bit (D0÷D7) e, come già visto nelle puntate precedenti, necessita dei canonici segnali di controllo aggiuntivi (WR, RD, RS, E e FS) non visibili nello schema a blocchi. Infine, in basso a destra, trovate le linee di collegamento verso

il chip del vetrino LCD, chip che non è integrato nel T6963C. Oltre a quelli menzionati, ci sono anche altri pin di controllo di particolare

Tabella 1a - Programmazione hardware

DUAL	H	H	H	H	H	H	H	H	L	L	L	L	L	L	L
MDS	L	L	L	L	H	H	H	H	L	L	L	L	H	H	H
MD1	H	H	L	L	H	H	L	L	H	H	L	L	H	H	L
MD0	H	L	H	L	H	L	H	L	H	L	H	L	H	L	L
LINES	2	4	6	8	10	12	14	16	4	8	12	16	20	24	32
V-DOTS	16	32	48	64	80	96	112	128	32	64	96	128	160	192	256
1 SCREEN									2 SCREENS						

Tabella 1b

MD2	H	L	H	L
MD3	H	H	L	L
Columns	32	40	64	80

importanza i quali però non sono accessibili perché configurati via hardware in fabbrica dal costruttore del display stesso. Stiamo parlando dei pin MD0, MD1, MD2, MD3 e MDS, che servono per impostare il numero di colonne e il numero di linee necessarie per pilotare in modo corretto il vetrino LCD. Per capire meglio il concetto utilizziamo le tabelle **1a** e **1b** tenendo presente che sia una linea che una colonna corrispondono a 8 pixel (pixel = colonne/linee x 8; colonne/linee = pixel / 8). Da tali tabelle è possibile ricavare il numero di linee e di colonne necessarie per programmare il controllore in base alla risoluzione del vetrino.

Ad esempio il display da 240x64 pixel necessita di 30 colonne x 8 linee, quindi, dalla tabella **1a** si ricavano le linee (8) e dalla tabella **1b** si ricavano le colonne (30 non è presente quindi si sceglie 32), pertanto la programmazione hardware diventa:

MD0=L, MD1=L, MDS=L per avere 8 linee;
MD2=H, MD3=H per avere 32 colonne.

Analogamente, il display da 240x128 pixel necessita sempre di 30 colonne per 16 linee quindi la programmazione diventa:

MD0=L, MD1=L, MDS=H per le 16 linee;
MD2=H, MD3=H per le 32 colonne.

Possiamo concludere che all'aumentare delle colonne cresce la risoluzione orizzontale dell'LCD fino a un massimo di 640 pixel (80 colonne) mentre aumentando il numero di linee aumenta la risoluzione verticale fino a un massimo di 128 pixel per la configurazione a singolo schermo e fino a 256 pixel per la configurazione a doppio schermo. Comunque questi pin di controllo non sono accessibili e quindi è compito del costruttore di display configurarli nel modo corretto.

Parliamo ora della mappatura della memoria RAM esterna di 64 kbyte. La tabella 2 evidenzia la differenza di mappatura tra una gestione a singola scansione e una a doppia scansione. L'esempio è riferito alla massima memoria gestibile (64 kbyte).

Nella gestione a singola scansione la memoria viene suddivisa in tre aree distinte mentre la doppia scansione suddivide la memoria in due blocchi da 32 kbyte ognuno dei quali è suddiviso in tre aree distinte come nella modalità a singola scansione; una parte della memoria vie-

Tabella 2 - Mappa della RAM

Single Scan		Dual Scan	
Address	Area	Address	Area
0x0000h	TEXT AREA	0x0000h	TEXT AREA
		0x3FFFh	
0x7FFFh		0x4000h	GRAPHIC AREA
		0x77FFh	
0x8000h	GRAPHIC AREA	0x7800h	CGRAM AREA
		0x7FFFh	
		0x8000h	TEXT AREA
0xF7FFh		0x8FFFh	
0xF800h	CGRAM AREA	0x8000h	GRAPHIC AREA
		0x8FFFh	

ne utilizzata per immagazzinare le informazioni inerenti il testo da visualizzare sul display, la seconda per memorizzare le informazioni grafiche e infine l'ultima parte di RAM serve per i caratteri speciali non presenti nella CGROM interna al controller. Questa mappatura, però, non è fissa, ossia la memoria può essere mappata anche in modo diverso a seconda delle esigenze imposte dall'applicazione da realizzare. Inoltre non è obbligatorio seguire l'ordine della tabel- ➤

Tabella 3a - Elenco comandi

Comando	Codice	D1	D2	Funzione
Register Setting	00100001	X Addr.	Y Addr.	Set Cursor Pointer
	00100010	Data	0x00	Set Offset Register
	00100100	Low Addr.	High Addr.	Set Address Pointer
Set Control Word	01000000	Low Addr.	High Addr.	Set Text Home Address
	01000001	Col.	0x00	Set Text Area
	01000010	Low Addr.	High Addr.	Set Graphic Home Address
	01000011	Col.	0x00	Set Graphic Area
Mode Set	1000X000	/	/	OR Mode
	1000x001	/	/	EXOR Mode
	1000x011	/	/	AND Mode
	1000x100	/	/	Text Attribute Mode
	10000XXX	/	/	Internal CGROM Mode
	10001XXX	/	/	External CGRAM Mode
Display Mode	10010000	/	/	Display OFF
	1001XX10	/	/	Cursor ON, Blink OFF
	1001XX11	/	/	Cursor ON, Blink ON
	100101XX	/	/	Text ON, Graphic OFF
	100110XX	/	/	Text OFF, Graphic ON
	100111XX	/	/	Text ON, Graphic ON

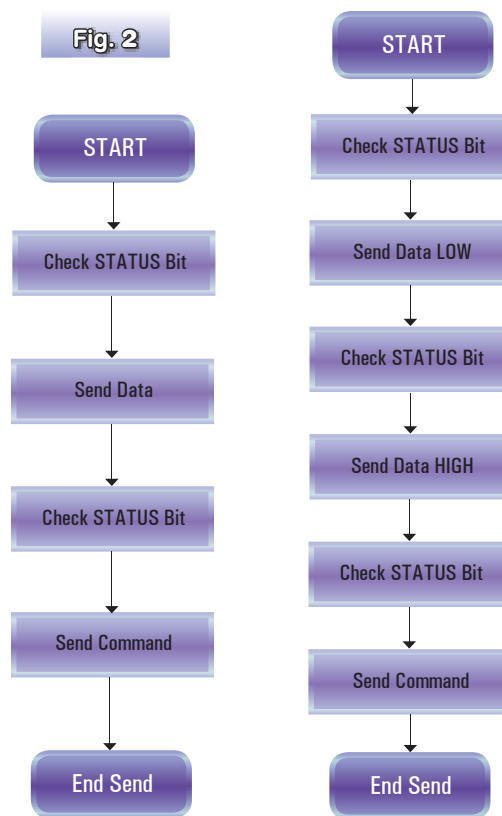
Tabella 3b - Elenco comandi

Comando	Codice	D1	D2	Funzione
Cursor Pat- tern Select	10100000	/	/	1 Line Cursor
	10100001	/	/	2 Line Cursor
	10100010	/	/	3 Line Cursor
	10100011	/	/	4 Line Cursor
	10100100	/	/	5 Line Cursor
	10100101	/	/	6 Line Cursor
	10100110	/	/	7 Line Cursor
	10100111	/	/	8 Line Cursor
Data Auto Read/Write	10110000	/	/	Set Data Auto Write
	10110001	/	/	Set Data Auto Read
	10110010	/	/	Auto Reset
Data Read/ Write	11000000	Data	/	Data Write and Increment ADP
	11000001	/	/	Data Read and Increment ADP
	11000010	Data	/	Data Write and Decrement ADP
	11000011	/	/	Data Read and Decrement ADP
	11000100	Data	/	Data Write
	11000101	/	/	Data Read
Screen Peek	11100000	/	/	Screen Peek
Screen Copy	11101000	/	/	Screen Copy
Bit Set/Reset	11110XXX	/	/	Bit Reset
	11111XXX	/	/	Bit Set
	1111X000	/	/	Bit 0 Set/Reset
	1111X001	/	/	Bit 1 Set/Reset
	1111X010	/	/	Bit 2 Set/Reset
	1111X011	/	/	Bit 3 Set/Reset
	1111X100	/	/	Bit 4 Set/Reset
	1111X101	/	/	Bit 5 Set/Reset
	1111X110	/	/	Bit 6 Set/Reset
	1111X111	/	/	Bit 7 (MSB) Set/Reset

la, infatti è possibile invertire l'area grafica con quella testo; l'importante è configurare in modo corretto il controller T6963C. Ad esempio, per la gestione del display 64240B (240x64 pixel) la RAM è così suddivisa: 0x0000÷0x077F per la memoria grafica, 0x0780÷0x0EFF per la memoria di testo ed infine da 0x1400 parte la memoria CGRAM. Quanto appena esposto conferma la dote di flessibilità del chip.

Analizziamo ora le istruzioni che il controller T6963C può accettare, riassunte nella tabella 3. Per alcuni comandi sono previsti dei parametri i quali devono essere inviati al controller prima del comando stesso seguendo un'apposita pro-

Fig. 2



cedura, riassunta nei diagrammi di flusso di figura 2. Il primo riguarda l'invio di un comando al quale è associato un solo parametro, mentre il secondo riguarda un comando a cui sono associati due parametri. In entrambi i casi prima bisogna inviare il parametro e poi il comando. Nel caso di comandi con due parametri

Tabella 4 - Status Register

Bit	Descrizione	Stato
STA0	Check command execution capability	0: Disable
		1: Enable
STA1	Check data read/write capability	0: Disable
		1: Enable
STA2	Check auto mode data read capability	0: Disable
		1: Enable
STA3	Check auto mode data write capability	0: Disable
		1: Enable
STA4	Non usato	
STA5	Check controller operation capability	0: Disable
		1: Enable
STA6	Error Flag used for Screen Peek and Screen Copy commands	0: Disable
		1: Enable
STA7	Check the blink conditions	0: Disable
		1: Enable

è importante inviare prima il parametro meno significativo e poi quello più significativo. Osservando la tabella 3 (che per esigenze di spazio è stata spezzata in due parti, 3a e 3b) il parametro meno significativo è “D1” mentre il più significativo è “D2”. In entrambi i casi prima di inviare un dato o un parametro è necessario testare se il controller è occupato, leggendo contemporaneamente i due bit STA0 e STA1 del registro di stato (Tabella 4) e verificando che siano entrambi a “1” logico. Nel caso in cui non fosse libero è necessario attendere che si liberi per poi inviare il dato o il comando.

Per eseguire la lettura del registro di stato è necessario attivare in modo corretto i seguenti pin di controllo RD (“0” logico), WR (“1” logico) CE (“0” logico) e C/D (“1” logico) rispettando il diagramma temporale riportato più avanti.

Esaurite le premesse, analizziamo i comandi dei vari gruppi, rispettando l’ordine di tabella 3.

Gruppo Register Setting

Prevede tre comandi distinti, ognuno dei quali necessita di due parametri aggiuntivi.

Set Cursor Pointer

Serve per impostare la posizione del cursore specificata rispetto agli assi X e Y: questi ultimi vengono definiti dai due parametri aggiuntivi “D1” e “D2”. La posizione del cursore può essere variata solo utilizzando questo comando mentre non viene alterata da normali operazioni di lettura o scrittura della memoria RAM.

Per l’asse X è possibile definire un valore tra 0 e

Tabella 5 - Set Cursor Pointer

C/D	RD	WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
0	1	0	0	X	X	X	X	X	X	X	D1
0	1	0	0	0	0	X	X	X	X	X	D2
1	1	0	0	0	1	0	0	0	0	1	COMM

79 in quanto il controller può gestire fino ad 80 colonne. Per quanto riguarda l’asse Y, invece, i valori consentiti vanno da un minimo di 0 a un massimo di 31. Se il controller viene fatto lavorare nella modalità dual scan, la coordinata Y può assumere valore compreso tra 0 e 15 e tra 16 e 31 in quanto il display LCD è suddiviso in due zone distinte. Per eseguire una scrittura è

necessario impostare il pin RD a livello logico alto e il pin WR a livello logico basso. Il pin C/D è a livello logico basso per la scrittura di un dato e a livello logico alto per la scrittura di un comando (vedere tabella 5).

Set Offset Register

Serve per impostare l’area di RAM dedicata alla CGRAM, cioè quella porzione di memoria utilizzata per memorizzare i caratteri speciali non presenti nella CGROM del controller stesso. Per impostare correttamente l’“Offset Register” è necessario tenere conto della memoria RAM già occupata dalla grafica e dal testo più la memoria

Tabella 6 - Set Offset Register

Offset Register Data					Character Code								Line Scan			
A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	

RAM rimasta disponibile. Nell’“Offset Register” è necessario impostare i campi “Offset Register Data” e “Character Code”. Il parametro “Line Scan” viene lasciato a 0 logico (vedi tabella 6).

Tabella 7 - Relazione indirizzo RAM/Offset

Offset Register Data	CGRAM Address
00000	0x0000 → 0x07FF
00001	0x0800 → 0x0FFF
00010	0x1000 → 0x17FF
00011	0x1800 → 0x1FFF
.....	
....	
11110	0xF000 → 0xF7FF
11111	0xF800 → 0xFFFF

La relazione tra la memoria RAM e l’Offset register è riassunta nella tabella 7. Da essa si evince che il codice assegnato all’“Offset Register Data” identifica una porzione di memoria RAM. Ad esempio la memoria compresa tra 0x0800 e 0x0FFF è identificata quando “Offset Register Data” ha valore 0x01. Quindi per capire quale valore assegnare all’“Offset Register Data” è necessario sapere quanta memoria RAM è già stata mappata per gestire le informazioni di grafica e testo. Nel caso del display 64240B, la memoria utilizzata sia per grafica che per testo occupa complessivamente 3840 byte (0x0F00) e quindi non eccede il secondo blocco di memoria come indicato in ➤

Tabella 8 - Esempio di Set Offset Register

Offset Register Data					Character Code								Line Scan
0	0	0	1	0	1	0	0	0	0	0	0	0	0

tabella 7. In questo caso il codice da assegnare all'“Offset Register Data” è 0x02, che significa partire dall'indirizzo 0x1000.

Definito il codice da assegnare all'“Offset Register Data” è necessario impostare il “Character Code”. Siccome i caratteri da 0x00 a 0x7F identificano i caratteri presenti nella CGROM del controller si decide che il codice di partenza per i caratteri presenti nella CGRAM debba iniziare da 0x80. Quindi i due parametri da inviare insieme al comando “Set Offset Register” sono “D1=0x14, D2=0x00”, come visibile in tabella 8; la sequenza completa da inviare al controllore è visibile in tabella 9.

Set Address Pointer

Riassunto in tabella 10, esso identifica l'indirizzo dal quale iniziare a scrivere o leggere la memoria RAM. Al posto delle “X” è necessario inserire il corretto indirizzo di memoria.

Tabella 9 - Sequenza di init

C/D	RD	WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
0	1	0	0	0	0	1	0	1	0	0	D1
0	1	0	0	0	0	0	0	0	0	0	D2
1	1	0	0	0	1	0	0	0	1	0	COMM

Tabella 10 - Set Address Pointer

C/D	RD	WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
0	1	0	X	X	X	X	X	X	X	X	D1
0	1	0	X	X	X	X	X	X	X	X	D2
1	1	0	0	0	1	0	0	1	0	0	COMM

Gruppo Set Control Word

Esso contiene le istruzioni per impostare il numero di colonne necessarie alla gestione del display LCD e gli indirizzi di partenza sia per la memoria testo che la memoria grafica. Si ricorda che il numero di colonne viene impostato tramite hardware, però è necessario affinare tale impostazione anche via software in quanto il numero di colonne impostato via hardware potrebbe essere maggiore del numero di colonne effettive del display LCD. Come abbiamo già visto in precedenza, il display 64240B è impostato a

32 colonne, però ne sono necessarie “solo” 30 (240 pixel / 8 bit a colonna = 30 colonne).

Set Text Home Address

Definisce l'indirizzo di inizio del blocco di RAM che contiene il testo, cioè i caratteri da visualizzare sul display LCD. Questi vengono prelevati dalla CGROM o dalla CGRAM utilizzando l'appropriato codice. I codici per identificare il testo non hanno corrispondenza con la codifica ASCII. Il comando è riportato in tabella 11.

Set Text Area

Visibile in tabella 12, esso identifica il numero di colonne necessarie per la corretta gestione del display LCD. Questo parametro va impostato sia per la memoria testo che per la memoria grafica.

Tabella 11 - Set Text Home Address

C/D	RD	WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
0	1	0	X	X	X	X	X	X	X	X	D1
0	1	0	X	X	X	X	X	X	X	X	D2
1	1	0	0	1	0	0	0	0	0	0	COMM

Tabella 12 - Set Text Area

C/D	RD	WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
0	1	0	X	X	X	X	X	X	X	X	D1
0	1	0	X	X	X	X	X	X	X	X	D2
1	1	0	0	1	0	0	0	0	0	0	COMM

Set Graphic Home Address e Set Graphic Area

Svolgono esattamente le stesse funzioni descritte dai due comandi precedenti, ma si applicano al blocco di memoria grafica. Sono inoltre molto simili, come potete verificare nelle tabelle 13 e 14.

Tabella 13 - Set Graphic Home Address

C/D	RD	WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
0	1	0	X	X	X	X	X	X	X	X	D1
0	1	0	X	X	X	X	X	X	X	X	D2
1	1	0	0	1	0	0	0	0	1	0	COMM

Tabella 14 - Set Graphic Area

C/D	RD	WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
0	1	0	0	X	X	X	X	X	X	X	D1
0	1	0	0	0	0	0	0	0	0	0	D2
1	1	0	0	1	0	0	0	0	1	1	COMM

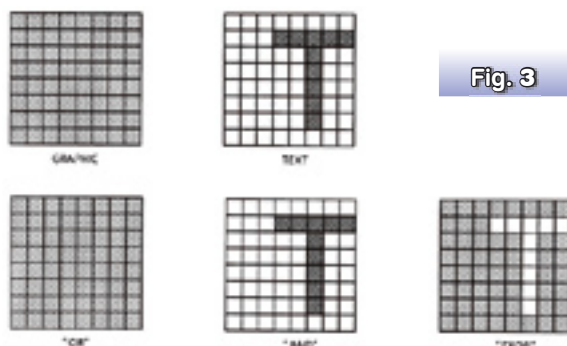


Fig. 3

Tabella 15 - OR MODE

C/D	RD	WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
1	1	0	1	0	0	0	X	0	0	0	COMM

Tabella 16 - EXOR MODE

C/D	RD	WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
1	1	0	1	0	0	0	X	0	0	1	COMM

Tabella 17 - AND MODE

C/D	RD	WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
1	1	0	1	0	0	0	X	0	1	1	COMM

Gruppo Mode Set

Questo gruppo serve per impostare la modalità di interazione tra il livello grafico e il livello testo. Il controller T6963C gestisce in modo separato i due livelli (testo e grafica) e, attraverso una operazione booleana, miscela l'uno con l'altro ottenendo così l'immagine finale. Le possibili operazioni che si possono fare tra i due livelli sono OR mode, EXOR mode e AND mode. È inoltre possibile decidere se eseguire tali operazioni booleane rispetto ai caratteri presenti nella CGROM o nella CGRAM. Questa serie di comandi non prevede alcun parametro aggiuntivo. Le immagini di figura 3 mostrano la differenza tra le tre operazioni booleane.

Nelle tabelle 15, 16 e 17 abbiamo riportato i comandi necessari per definire le tre operazioni.

Tabella 18 - Text Attribute Mode

D7	D6	D5	D4	D3	D2	D1	D0	Function
X	X	X	X	0	0	0	0	Normal Display
X	X	X	X	0	1	0	1	Reverse Display
X	X	X	X	0	0	1	1	Inhibit Display
X	X	X	X	1	0	0	0	Blink of normal Display
X	X	X	X	1	1	0	1	Blink of reverse Display
X	X	X	X	1	0	1	1	Blink of inhibit Display

Al posto della "X" occorre scrivere "0" se si utilizza la CGROM interna e quindi i caratteri pre-memorizzati nel controller T6963C; occorre, invece, scrivere "1" se si utilizzano i caratteri memorizzati nella CGRAM.

Nella modalità "Text Attribute Mode" è possibile agire solo sul livello testo in quanto il livello grafico viene utilizzato per la memorizzazione dei parametri inerenti le varie modalità di gestione del "Text Attribute Mode". Per il corretto funzionamento è comunque necessario accendere sia il livello grafico che il livello testo utilizzando l'apposito comando da inviare al controller. Se uno dei due livelli non è acceso le impostazioni effettuate in "Text Attribute Mode" non funzioneranno in modo corretto.

Le possibilità di configurazione messe a disposizione della modalità "Text Attribute Mode" sono riportate in tabella 18.

Gruppo Display Mode

Il gruppo di comandi serve per impostare la modalità di funzionamento dei due livelli (grafico e testo) e del relativo cursore.

Si ricorda che quando si devono combinare insieme elementi di grafica e di testo, entrambi i livelli devono essere accesi. Il dettaglio di ogni comando, bit per bit, è riportato nelle tabelle da 19 a 24.

È possibile combinare tra loro più comandi: ad esempio è possibile ottenere Text ON, Graphic ON, Cursore ON e blink ON inviando il comando ➤

Tabella 19 - Display OFF

C/D	RD	WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
1	1	0	1	0	0	1	0	0	0	0	COMM

Tabella 20 - Cursore ON, blink OFF

C/D	RD	WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
1	1	0	1	0	0	1	X	X	1	0	COMM

Tabella 21 - Cursore ON, blink ON

C/D	RD	WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
1	1	0	1	0	0	1	0	0	1	1	COMM

Tabella 22 - Text ON, Graphic OFF

C/D	RD	WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
1	1	0	1	0	0	1	0	1	X	X	COMM

Tabella 23 - Text OFF, Graphic ON

C/D	RD	WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
1	1	0	1	0	0	1	1	0	X	X	COMM

Tabella 24 - Text ON, Graphic ON

C/D	RD	WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
1	1	0	1	0	0	1	1	1	X	X	COMM

do visibile in tabella 25. Risulta evidente che è possibile ottenere tutte le combinazioni possibili messe a disposizione del controller semplicemente variando un bit per ognuna di esse.

Tabella 25

C/D	RD	WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
1	1	0	1	0	0	1	1	1	1	1	COMM

Gruppo Cursor Pattern Select

Tale gruppo di comandi serve per impostare le dimensioni del cursore, ossia il numero di linee di quest'ultimo, da un minimo di una linea a un massimo di 8 linee (Cursore pieno); esso può essere inviato solo se il cursore è stato precedentemente acceso con i comandi descritti al punto precedente. Il comando da inviare al controller per impostare le dimensioni del cursore è visibile in tabella 26.

Al posto delle X va inserito il codice che identifica la dimensione del cursore. Al valore logico "000" corrisponde una dimensione del cursore pari a una linea, mentre al valore logico "111" corrisponde il cursore pieno (8 linee).

Tabella 26 - Gruppo Cursor Pattern Select

C/D	RD	WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
1	1	0	1	0	1	0	0	X	X	X	COMM

Gruppo Auto Read/Write

Questo gruppo di comandi risulta utile quando è necessario trasferire le informazioni grafiche da visualizzare da una memoria esterna alla RAM del controller. Infatti, impostata ad esempio la

Tabella 27 - Auto Write

C/D	RD	WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
1	1	0	1	0	1	1	0	0	0	0	COMM

Tabella 28 - Auto Read

C/D	RD	WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
1	1	0	1	0	1	1	0	0	0	1	COMM

Tabella 29 - Auto Reset

C/D	RD	WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
1	1	0	1	0	1	1	0	0	1	0	COMM

modalità "Auto Write", è necessario inviare al controller solo il dato da scrivere nella RAM senza la necessità di specificare alcun ulteriore comando.

Conclusa la trasmissione dei dati inerenti l'immagine da visualizzare, è necessario inviare il comando di "Auto Reset" per uscire da questa particolare modalità e abilitare così il controller T6963C a ricevere nuovi comandi. Prima di inviare il comando di "Auto Write" o di "Auto Read" è necessario impostare "Address Pointer" per indicare al controller da quale locazione di memoria iniziare a scrivere o leggere.

Auto Write, Auto Read e Auto Reset sono visibili rispettivamente nelle tabelle 27, 28 e 29.

Gruppo Data Read/Write

Il seguente gruppo di comandi serve per leggere o scrivere dati nella memoria RAM del controller. Prima di iniziare un'operazione di lettura o scrittura della RAM è necessario impostare l'indirizzo di partenza da cui leggere o scrivere, ovvero l'"Address Pointer".

Impostato l'indirizzo di partenza è possibile iniziare con la lettura o scrittura. Durante ogni scrittura il comando "Data Write" deve essere preceduto dal dato stesso che si vuole scrivere in RAM, inoltre ad ogni lettura o scrittura è possibile decidere se il puntatore alla successiva locazione di memoria RAM debba essere incrementato o decrementato.

Tabella 30

C/D	RD	WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
0	1	0	0	X	X	X	X	X	X	X	D1
1	1	0	1	1	0	0	0	0	0	0	COMM

Tabella 31

C/D	RD	WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
0	1	0	0	X	X	X	X	X	X	X	D1
1	1	0	1	1	0	0	0	0	1	0	COMM

Tabella 32

C/D	RD	WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
0	1	0	0	X	X	X	X	X	X	X	D1
1	1	0	1	1	0	0	0	1	0	0	COMM

Tabella 33

C/D	RD	WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
1	1	0	1	1	0	0	0	0	0	1	COMM

Tabella 34

C/D	RD	WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
1	1	0	1	1	0	0	0	0	1	1	COMM

Tabella 35

C/D	RD	WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
1	1	0	1	1	0	0	0	1	0	1	COMM

Di seguito riportiamo i comandi messi a disposizione del controller:

- scrittura e incremento del puntatore in RAM (tabella 30);
- scrittura e decremento del puntatore in RAM (tabella 31);
- scrittura senza variazione del puntatore in RAM (tabella 32);
- lettura e incremento del puntatore in RAM (tabella 33);
- lettura e decremento del puntatore in RAM (tabella 34);
- lettura senza variazione del puntatore in RAM (tabella 35).

Gruppo Bit Set/Reset

Questo gruppo di comandi serve per impostare un singolo bit di un singolo byte presente nella memoria RAM del controller. Per eseguire correttamente questo comando è necessario impostare in precedenza l'indirizzo di memoria RAM su cui si vuole agire ("Address Pointer") e poi inviare il comando al controller T6963C.

L'unica differenza tra i due comandi, visibili nelle tabelle 36 e 37, è rappresentata dal bit DB3.

Al posto delle X va inserito il codice che identifica quale bit si vuole modificare. Al valore logico "000" corrisponde il bit meno significativo, mentre al valore logico "111" corrisponde il bit più significativo.

Firmware per controller T6963C

Come in tutti i controller grafici trattati fino ad ora, anche il T6963C necessita della corretta

Tabella 36 - Comando di Set

C/D	RD	WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
1	1	0	1	1	1	1	1	X	X	X	COMM

Tabella 37 - Comando di Reset

C/D	RD	WR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
1	1	0	1	1	1	1	0	X	X	X	COMM

inizializzazione. Il diagramma di flusso di figura 4 mostra la sequenza di operazioni da eseguire per programmare correttamente il T6963C. Per prima cosa è necessario impostare la memoria RAM allocando lo spazio necessario per memorizzare i dati inerenti la grafica ed il testo. Ad esempio il display 64240B è stato impostato affinché la memoria grafica parta dall'indirizzo 0x0000 e la memoria testo dall'indirizzo 0x0780. Essendo un display da 240x64 pixel sono necessari 1920 Byte ($240 \times 64 / 8 = 1920$) sia per memorizzare le informazioni grafiche che le informazioni testo. Se il display LCD fosse più grande, allora sarebbe necessario allocare più spazio per entrambi i blocchi di memoria. Durante l'allocazione dello spazio, è possibile assegnare indifferentemente prima la memoria grafica rispetto alla memoria testo e viceversa.

Il passo successivo consiste nell'impostare il numero di colonne necessario al funzionamento del display LCD: nel nostro caso si deve definire un numero di colonne pari a 30, in quanto la risoluzione orizzontale del nostro display è di 240 pixel (sia per il 64240B che per il 128240A).

Conclusa l'inizializzazione dello spazio di memoria e del numero di colonne, occorre impostare l'"Offset Register" necessario per gestire in modo corretto la CGRAM. Lo spazio dedicato alla CGRAM viene ricavato dallo spazio disponibile in RAM. Per i dettagli sulla configurazione dell'"Offset Register" consultate il paragrafo "Gruppo Register Setting" già visto in precedenza.

Successivamente va configurata la modalità di interazione tra il livello grafico e il livello testo ovvero l'operazione booleana che il controller

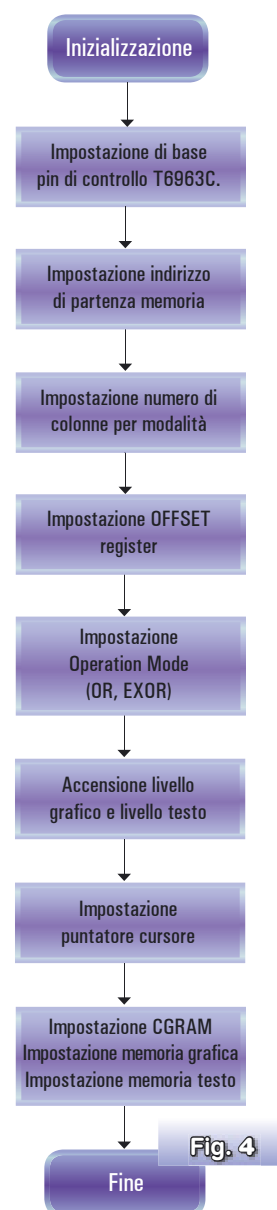


Fig. 4

deve eseguire tra i due livelli. Nel caso in esame è stata impostata l'operazione booleana OR.

Subito dopo si devono attivare i due livelli (grafico e testo) che, nell'applicazione in esame, sono entrambi utilizzati. Esistono comunque applicazioni in cui non è necessario utilizzare entrambi i livelli. Per ultimo si inizializzano il cursore e le sue dimensioni.

Conclusa la procedura iniziale, è necessario (ma non obbligatorio) impostare la CGRAM caricando in essa i caratteri speciali da utilizzare con l'applicazione. È inoltre consigliato inizializ-

zare con dei valori noti la memoria grafica e la memoria testo. Riportiamo nel riquadro Listato 1 il codice di inizializzazione del controller T6963C. Le funzioni "Write_Procedure_Data" e "Write_Procedure_Comm" si occupano di inviare un dato o un comando al controller: per funzionare correttamente, esse devono ricevere un parametro a 8 bit che identifica il dato o il codice del comando da inviare (vedere Listato 2).

Le due funzioni sono molto semplici, infatti contengono entrambe due ulteriori chiamate a funzioni. La prima verifica se il controller è libero, mentre la seconda serve per inviare un comando oppure un dato a seconda dei casi.

La funzione "Write_Procedure_Data" contiene una chiamata alla funzione "Write_T6963C_Data"

mentre la funzione "Write_Procedure_Comm" alla funzione "Write_T6963C_Comm". Come

Listato 1

```
void Init_T6963C(void)
{
  Set_Pin_T6963C();
  Write_Procedure_Data(SET_GRAPH_HOME_ADD_LOW);
  Write_Procedure_Data(SET_GRAPH_HOME_ADD_HIGH);
  Write_Procedure_Comm(SET_GRAPH_HOME_ADD);
  Write_Procedure_Data(SET_TXT_HOME_ADD_LOW);
  Write_Procedure_Data(SET_TXT_HOME_ADD_HIGH);
  Write_Procedure_Comm(SET_TXT_HOME_ADD);
  Write_Procedure_Data(SET_GRAPH_AREA_LOW);
  Write_Procedure_Data(SET_GRAPH_AREA_HIGH);
  Write_Procedure_Comm(SET_GRAPH_AREA);
  Write_Procedure_Data(SET_TXT_AREA_LOW);
  Write_Procedure_Data(SET_TXT_AREA_HIGH);
  Write_Procedure_Comm(SET_TXT_AREA);
  Write_Procedure_Data(SET_CGRAM_CODE_LOW);
  Write_Procedure_Data(SET_CGRAM_CODE_HIGH);
  Write_Procedure_Comm(SET_OFFSET_REG);
  Write_Procedure_Comm(SET_OR_MODE_CGRAM);
  Write_Procedure_Comm(TXT_ON_GRAPH_ON);
  Write_Procedure_Data(0x00);
  Write_Procedure_Data(0x00);
  Write_Procedure_Comm(SET_CURSOR_POINTER);
  Write_Procedure_Comm(LINE_CURSOR_1);
  Set_CGRAM_RAM_8x8Pixel();
  Set_CGRAM_RAM_16x16Pixel();
  Clear_Grap_RAM();
  Clear_Text_RAM();
}
```

Inizializzazione pin di controllo T6963C

Impostazione indirizzo di partenza memoria grafica

Impostazione indirizzo di partenza memoria testo

Impostazione numero di colonne area grafica

Impostazione numero di colonne area testo

Impostazione Offset register per gestione CGRAM

Impostazione interazione tra livelli Accensione livello grafica e livello testo

Impostazione coordinate cursore

Impostazione dimensione cursore

Inizializzazione memoria CGRAM, precaricamento caratteri speciali. Inizializzazione memoria grafica e memoria testo

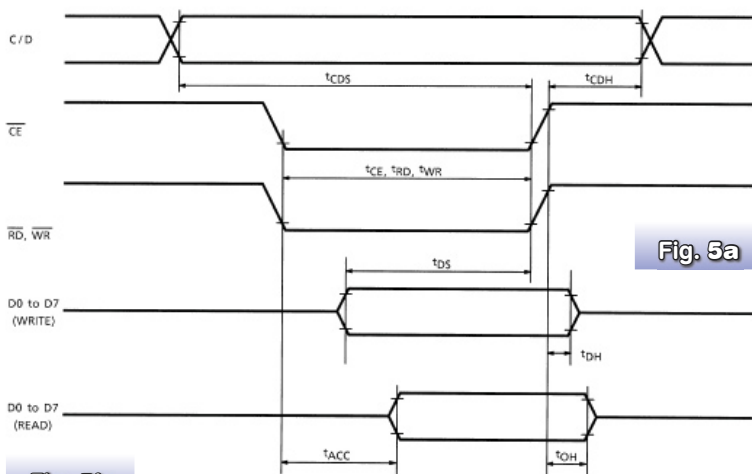


Fig. 5b

Item	Symbol	Test Conditions	Min	Max	Unit
C / D Set-up Time	t_{CDS}	—	100	—	ns
C / D Hold Time	t_{CDH}	—	10	—	ns
$\overline{CE}$, $\overline{RD}$, $\overline{WR}$ Pulse Width	t_{CE} , t_{RD} , t_{WR}	—	80	—	ns
Data Set-up Time	t_{DS}	—	80	—	ns
Data Hold Time	t_{DH}	—	40	—	ns
Access Time	t_{ACC}	—	—	150	ns
Output Hold Time	t_{OH}	—	10	50	ns

Listato 2

```

void Write_Procedure_Comm(uchar Data)
{
    Test_T6963C_Busy();
    Write_T6963C_Comm(Data);
}

void Write_Procedure_Data(uchar Data)
{
    Test_T6963C_Busy();
    Write_T6963C_Data(Data);
}

void Test_T6963C_Busy(void)
{
    TRISD = 0b11111111;

Status_Read:
    RS_LCD_M(HIGH);
    WR_LCD_M(HIGH);
    RD_LCD_M(LOW);
    E_LCD_M(LOW);
    Nop();
    Nop();
    E_LCD_M(HIGH);
    T6963C_Status = PORTD;

    if ((T6963C_Status & 0x03) != 0x03)
    {
        goto Status_Read;
    }
    TRISD = 0b00000000;
}

void Write_T6963C_Comm(uchar Data)
{
    LATD = Data;
    RS_LCD_M(HIGH);
    RD_LCD_M(HIGH);
    WR_LCD_M(LOW);
    E_LCD_M(LOW);
    Nop();

    E_LCD_M(HIGH);
}

void Write_T6963C_Data(uchar Data)
{
    LATD = Data;
    RS_LCD_M(LOW);
    RD_LCD_M(HIGH);
    WR_LCD_M(LOW);
    E_LCD_M(LOW);
    Nop();
    E_LCD_M(HIGH);
}

```

già anticipato, prima di inviare un comando o un dato è necessario testare se il controller è in condizione di busy. Nel caso in cui esso è occupato, il firmware deve rimanere in un ciclo di attesa fino a quando il controller torna nuovamente disponibile a ricevere dati e/o comandi. Come sempre, per accedere in lettura o scrittura al controller è necessario rispettare un diagramma temporale: quello del T6963C è visibile in figura 5a.

La funzione “Write_T6963C_Data” invia un dato (solitamente un parametro o un byte da scrivere in RAM) al controller. La routine si occupa, oltre che di caricare il dato sulla porta D, anche di gestire in modo corretto i pin di controllo del T6963C rispettando tutte le tempistiche richieste. I tempi riportati nella tabella di figura 5b si riferiscono a un controller alimentato a +5 Vdc. Nel caso in cui il controller

sia alimentato a +3,3Vdc, i tempi necessari per la gestione del controller aumentano. La funzione “Write_T6963C_Comm” è equivalente alla precedente, ma invece di inviare un dato invia un comando.

Passiamo ora ad analizzare la routine che invia i caratteri speciali alla memoria CGRAM del controller. A seconda dei parametri passati, essa permette di memorizzare i caratteri da 8x8 pixel oppure i caratteri da 16x16 pixel. Un carattere da 16x16 pixel non è altro che la somma di quattro caratteri da 8x8 pixel. La funzione

Listato 3

```

void Set_CGRAM_RAM_Pixel(uchar Font_Number,
                          uchar Counter,
                          uchar CGRAM_Add_Low,
                          uchar CGRAM_Add_High,
                          uint PIC_CGRAM_Table)
{
    Write_Procedure_Data(CGRAM_Add_Low);
    Write_Procedure_Data(CGRAM_Add_High);
    Write_Procedure_Comm(SET_ADDRESS_POINTER);

    TBLPTR = PIC_CGRAM_Table;
    do
    {
        T6963C_Counter = Counter;
        do
        {
            Write_Procedure_Data(Read_Data_Flash());
            Write_Procedure_Comm(DATA_WR_AUTOINC);
        } while (--T6963C_Counter != 0);
    } while (--Font_Number != 0);
}

```

per la programmazione della CGRAM deve ricevere i seguenti parametri:

- numero di font da caricare in CGRAM;
- numero di byte per font;
- indirizzo di partenza in CGRAM;
- indirizzo di partenza della tabella caratteri nella FLASH PIC.

I byte da inviare alla CGRAM sono prememorizzati nella FLASH del PIC18F4620 e ognuno di essi è stato ricavato utilizzando un semplice programma per piattaforma Windows realizzato per l'occasione.

La routine di scrittura in CGRAM, per prima cosa, imposta l'indirizzo da cui iniziare a scrivere.

Per il display 64240B l'indirizzo di partenza della CGRAM è 0x1400 il quale è identificato dalle due costanti “SET_CGRAM_8x8_ADD_LOW” e “SET_CGRAM_8x8_ADD_HIGH”, memorizzate nel file T6963C.h.

Successivamente trasferisce i byte memorizzati nel PIC dalla sua FLASH alla memoria RAM del controller. Sono stati previsti 32 caratteri speciali da 8x8 pixel per un totale di 256 byte. ➤

Listato 4

```
#pragma romdata _ADD_CGRAM_TABLE = 0x300
rom const char CGRAM_Table[256] = { 0x30, 0x48, 0x84, 0x84, 0x49, 0x32, 0x04, 0x08,
                                     0x00, 0x66, 0x00, 0x18, 0x99, 0x42, 0x3C, 0x00,
                                     0x00, 0x3C, 0x52, 0x10, 0x10, 0x12, 0x0C, 0x00,
                                     0x84, 0x84, 0xFF, 0x08, 0x08, 0x38, 0x44, 0x83,
                                     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                     .....
                                     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};
```

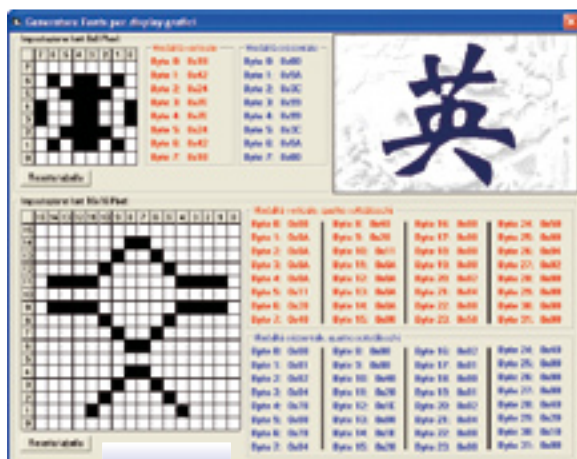


Fig. 6

Per richiamare i caratteri speciali bisogna utilizzare i codici da 0x80 in poi i quali puntano ai caratteri speciali. Definito il carattere sarà preoccupazione del controller andare a recuperare gli otto byte necessari per visualizzare il carattere sul display. La tabella di caratteri speciali da 8x8 pixel prememorizzata in FLASH è allocata a partire dall'indirizzo 0x300. Nel listato 4 è visibile una parte di questa tabella caratteri.

Per facilitare la creazione dei caratteri speciali, e quindi ricavare in modo semplice e intuitivo i byte da scrivere in CGRAM, si può utilizzare il semplice programma visibile in figura 6 tramite il quale si possono generare sia i caratteri da 8x8 pixel che i caratteri da 16x16 pixel. I byte generati possono essere uti-

lizzati con qualsiasi display che abbia la possibilità di gestire una memoria CGRAM. In rosso sono evidenziati i byte per i display che gestiscono la memoria RAM in modalità verticale mentre in blu sono evidenziati i byte per i display che gestiscono la memoria RAM in modalità orizzontale come nel caso dei due display 64240B e 128240A. Si osservi che per i caratteri da 16x16 pixel si ottengono come risultato 4 colonne da 8 byte, ognuna delle quali identifica ¼ del carattere in esame. In pratica, come detto precedentemente, un carattere da 16x16 pixel è formato da 4 caratteri da 8x8 pixel. I caratteri da 16x16 pixel sono pre-caricati nella FLASH del PIC18F4620 a partire dall'indirizzo 0x400 occupando anch'essi 256 byte per un totale di 8 caratteri.

Vediamo ora le due funzioni necessarie all'impostazione della memoria RAM dedicata alla grafica e al testo.

Come già spiegato in precedenza, è buona norma inizializzare la memoria del controller gra-

Listato 5

```
void Clear_Grap_RAM(void)
{
    T6963C_Counter = BYTE_FOR_GRAPH_AREA;
    Write_Procedure_Data(0x00);
    Write_Procedure_Data(0x00);
    Write_Procedure_Comm(SET_ADDRESS_POINTER);
    do
    {
        Write_Procedure_Data(0x00);
        Write_Procedure_Comm(DATA_WR_AUTOINC);
    } while (--T6963C_Counter != 0 );
}

void Clear_Text_RAM(void)
{
    T6963C_Counter = BYTE_FOR_TXT_AREA >> 3;
    Write_Procedure_Data(SET_TXT_HOME_ADD_LOW);
    Write_Procedure_Data(SET_TXT_HOME_ADD_HIGH);
    Write_Procedure_Comm(SET_ADDRESS_POINTER);
    do
    {
        Write_Procedure_Data(0x00);
        Write_Procedure_Comm(DATA_WR_AUTOINC);
    } while (--T6963C_Counter != 0 );
}
```

Numero di byte da inizializzare

Impostazione indirizzo di partenza da cui iniziare a inizializzare la memoria RAM dedicata alla grafica

Scrittura in memoria RAM all'indirizzo specificato. Usato comando con autoincremento puntatore in memoria RAM

Numero di byte da inizializzare

Impostazione indirizzo di partenza da cui iniziare a inizializzare la memoria RAM dedicata al testo

Scrittura in memoria RAM all'indirizzo specificato. Usato comando con autoincremento puntatore in memoria RAM

fico. Per l'applicazione in esame impostiamo la memoria al valore 0x00 che equivale ad avere uno schermo bianco per la grafica e un carattere di spaziatura per il testo.

Ultime, ma non meno importanti, trattiamo le funzioni di scrittura sul vetrino.

La funzione che si occupa di visualizzare una immagine sul display ha bisogno del parametro "indirizzo di partenza in EEPROM" con

Funzione dei pulsanti sulla demoboard

Diamo ora una breve descrizione delle funzioni assegnate ai cinque pulsanti presenti sulla demo-board.

- Pulsante "RIGHT": richiama l'immagine successiva. Nella memoria EEPROM sono memorizzate fino a 4 immagini richiamabili utilizzando il pulsante "RIGHT".

Listato 6

```
void Write_Grap_RAM(uint Address_EEPROM)
{
    T6963C_Counter = BYTE_FOR_GRAPH_AREA;
    I2C_EEPROM_Add.u1 = Address_EEPROM;
    Write_Procedure_Data(0x00);
    Write_Procedure_Data(0x00);
    Write_Procedure_Comm(SET_ADDRESS_POINTER);
    do
    {
        I2C_Process();
        Write_Procedure_Data(I2C_EEPROM_Data[0]);
        Write_Procedure_Comm(DATA_WR_AUTOINC);
        I2C_EEPROM_Add.u1 += 1;
    } while (--T6963C_Counter != 0 );
}

void Write_Char(uchar Char_Code, uchar Address_Low, uchar Address_High)
{
    Write_Procedure_Data(Address_Low);
    Write_Procedure_Data(Address_High);
    Write_Procedure_Comm(SET_ADDRESS_POINTER);

    Write_Procedure_Data(Char_Code);
    Write_Procedure_Comm(DATA_WR_AUTOINC);
}
```

Numero di byte necessari per visualizzare l'immagine

Indirizzo di partenza in EEPROM per "recupero" immagine

Indirizzo di partenza memoria RAM controller T6963C

Lettura EEPROM e scrittura in RAM

Impostazione coordinate carattere

Scrittura carattere sul display LCD

il quale "recuperare" l'immagine che si vuole visualizzare. Per i dettagli sulla gestione della memoria EEPROM consultate la quinta puntata del Corso, inerente i display grafici, in cui abbiamo parlato dell'hardware della demoboard. La funzione che si occupa di scrivere un carattere sul display, invece, necessita di più parametri.

In particolare ha bisogno di:

- il codice del carattere da visualizzare (Codici speciali a partire da 0x80);
- le coordinate in cui posizionare il carattere.

Per ogni riga è possibile inserire fino a un massimo di 30 caratteri da 8x8 pixel (nel caso in cui la risoluzione orizzontale del display sia di 240 pixel), mentre il numero di righe dipende dalla risoluzione verticale del display e dalla dimensione del carattere.

Con il display 64240B è possibile ottenere 8 righe con i caratteri di dimensione 8x8 pixel, mentre il display 128240A è in grado di gestire fino a 16 righe con caratteri della stessa dimensione (8x8 pixel).

Dopo la quarta immagine vengono mostrati i due font memorizzati nella memoria CGRAM.

- Pulsante "LEFT": richiama l'immagine precedente. Si comporta in maniera identica al pulsante "RIGHT", ma in direzione opposta. Al raggiungimento della prima, vengono poi richiamati i due font contenuti nella CGRAM. I pulsanti rimanenti non svolgono alcuna funzione.

Conclusione

Giunti a questo punto possiamo considerare conclusa la presentazione dei controller utilizzati nei vari display grafici supportati dalla nostra demoboard.

Nella prossima puntata, ultima dedicata a questo tipo di display, studieremo alcune funzioni per disegnare delle forme geometriche sui display grafici. Le routine che vedremo faranno riferimento al controller T6963C, ma i concetti trattati sono utilizzabili anche con altri chip, a condizione che venga apportato il necessario adattamento a seconda dei casi. Appuntamento quindi alla prossima uscita!