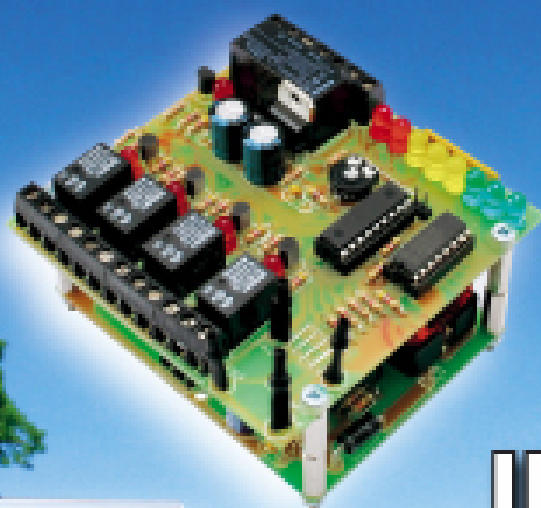
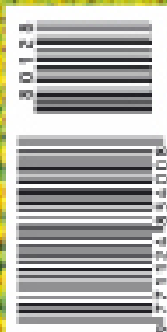


Mensile di elettronica applicata, attualità scientifica, novità tecnologiche.

128



INSEGUITORE per IMPIANTI SOLARI



- 
- Riconoscitore colori parlante
 - Registratore vocale digitale low - cost
 - Radiocontrollo via USB
 - Protezione finale di potenza
 - Piattaforma di sviluppo per PIC
 - Telecomando IR per sistemi VELBUS
 - Cartellino orario firmware e software

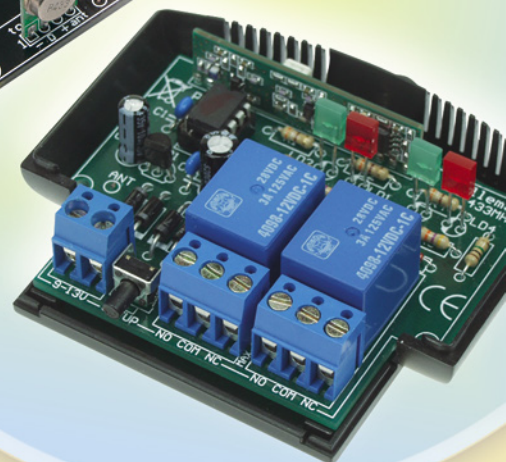
+++ ESCLUSIVO +++
CORSO EM1000
Programmazione
modulo Tibbo

Radiocontrollo gestibile da PC tramite USB

di *Cristiano Ruggeri*

Costituito da una unità trasmittente via radio, può controllare il funzionamento di ben 255 ricevitori con uscita a relé.

Il trasmettitore è collegato alla porta USB di un PC che, tramite un semplice programma di gestione, viene utilizzato sia per la configurazione che per il controllo del sistema. Del programma vengono forniti sia i codici sorgente che la libreria di controllo, quest'ultima utilizzabile per sviluppare applicazioni personalizzate.



Il PC viene ormai utilizzato praticamente dovunque: per configurare le centrali antifurto domestiche, per aggiornare il firmware di molti Set Top Box (i ricevitori per il digitale terrestre o per la TV via satellite), per attivare e disattivare da remoto dispositivi differenti come impianti di riscaldamento, climatizzazione, irrigazione, eccetera. Perché, quindi, non utilizzarlo anche per realizzare un telecontrollo via radio configurabile ed espandibile in base alle necessità di ognuno, con cui creare dei sistemi di attuazione distribuiti su piccole aree, quali la propria abitazione o la propria azienda?

Bene, quello che vi presentiamo in queste pagine è un telecontrollo USB via PC che, con un solo trasmettitore RF (codice progetto K8074), permette di comandare fino a 255 ricevitori a singolo canale (codice progetto K8070) oppure a 8 canali (K8056).

Il sistema viene gestito attraverso un programma versatile ed intuitivo, installato sul PC in uso, che consente sia di configurare l'indirizzo del trasmettitore che di inviare i comandi alle varie schede periferiche.

Qualora, però, il programma non dovesse rispondere alle vostre necessità, vi mettiamo a disposizione una libreria



ria software (K8074.DLL, completa di istruzioni) semplice e compatta, che potrete utilizzare per sviluppare la vostra applicazione pratica, semplicemente includendola in un progetto Visual Basic, DotNet, Delphi o qualsiasi altro ambiente di programmazione desideriate utilizzare. Vediamo ora come è composto il sistema.

Applicazione tipica

L'utilizzo combinato del trasmettitore e dei due ricevitori permette di realizzare delle reti di attuazione molto vaste: fate riferimento alla Fig. 1 in cui abbiamo rappresentato un ambiente di utilizzo che riassume tutte le varie casistiche.

La situazione tipica è quella in cui un trasmettitore K8074 opera in

unione ad un ricevitore, che può essere il K8070 (monocanale a due relé) oppure il K8056 (scheda a 8 canali a relé). Poiché, però, il trasmettitore K8074 dispone di un suo indirizzo, impostabile tra 0 e 255, è possibile realizzare fino a 255 reti differenti operanti nella stessa area, in cui possono essere presenti diversi moduli di ricezione e attuazione K8070 e K8056. Anche questi ultimi sono indirizzabili tra 0 e 255, il che consente di controllare singolarmente fino a 2040 relé differenti (utilizzando solo schede K8056).

Come potete intuire, il sistema è molto flessibile e consente di realizzare reti di controllo wireless vaste.

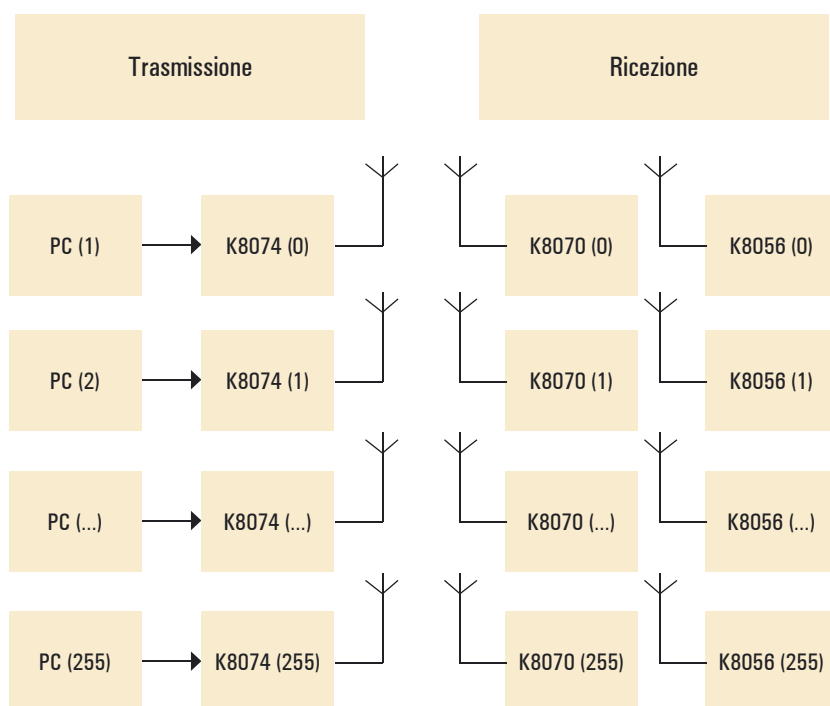
Il trasmettitore K8074

Si tratta di un dispositivo dalle dimensioni molto compatte, completamente configurabile e comandabile per mezzo di un software installato su di un PC, cui è collegato attraverso la porta USB. Lo schema elettrico è piuttosto semplice: il microcontrollore IC1, un PIC18F2450-I/SP, è programmato per gestire sia la porta USB di cui dispone (facente capo ai pin 15 e 16, rispettivamente RC4/D- e RC5/D+) che le altre periferiche. I tre led di segnalazione di attività (LD1, LD2 e LD3) vengono comandati dalle linee RA0, RA1 e RA2. L'ingresso DATA del modulatore RF viene attivato, attraverso una resistenza da 470 kohm (R9), dalla linea di trasmissione seriale del micro (pin 17, RC6). La linea RB0 è collegata al pulsante SW1, che svolge la funzione di test; l'alimentazione viene prelevata direttamente dalla porta USB del PC cui è collegato.

È presente il connettore ICSP tramite il quale programmare il microcontrollore. Notate, infine, la presenza di un jumper (MODE) collegato alla linea RB1 del micro: è stato previsto per eventuali sviluppi futuri, ma al momento non svolge alcuna funzione.

Fig. 1 - Schema a blocchi.

Nella stessa area possono coesistere differenti trasmettitori e ricevitori.



- **Interfaccia:** USB compatibile 1.1 e 2.0.
- **Portata:** fino a 30 metri in funzione dell'ambiente di utilizzo.
- **Compatibilità schede periferiche:**
 - K8056: scheda a relé a otto canali,
 - K8070: scheda monocanale a due relé indipendenti.
- **Indicazioni luminose:**
 - Led portante RF,
 - Led alimentazione,
 - Due led per il funzionamento della porta USB.
- **Indirizzo:** impostabile tra 0 e 255.
- **Controlli aggiuntivi:** pulsante per test di trasmissione.
- **Frequenza di trasmissione:** 433 MHz.
- **Omologazione:** conforme allo standard R&TTE.
- **Alimentazione:** tramite porta USB.

La libreria K8074.DLL

Dal punto di vista hardware, questo dispositivo è simile a molti altri; ciò che lo contraddistingue, però, va ricercato nel software, o meglio, nella libreria software specifica (scaricabile gratuitamente dal sito www.velleman.be) che consente di aprire un canale di comunicazione con il dispositivo attraverso la porta USB e di inviare comandi sia di configurazione del trasmettitore che di attuazione delle varie uscite presenti sui ricevitori periferici.

In un riquadro a parte, cui vi rimandiamo per gli approfondimenti tecnici, abbiamo riportato l'elenco dettagliato delle varie funzioni messe a disposizione dalla libreria. Qui, invece, vi descriviamo brevemente cosa essa consente di fare.

Essa, innanzitutto, permette di aprire e chiudere un canale di comunicazione con il dispositivo che, una volta installato sul PC, viene visto come una porta seriale virtuale.

L'apertura e la chiusura del canale si effettuano con le chiamate alle due funzioni *OpenDevice* e *CloseDevice*. Dopo aver aperto il canale, è possibile inviare due categorie di comandi: quelli indirizzati al dispositivo stesso e quelli destinati ai vari ricevitori.

Della prima categoria fanno parte *GetDLLVersion*, *SendCommand*, *DisplayAddress*, *SetAddress* e *ForceAddressReset* che consentono, rispettivamente, di verificare la versione FW della DLL, di inviare un comando e di mostrare, modificare o forzare a "1" l'indirizzo del trasmettitore.

Nella seconda categoria, invece, troviamo tutti i comandi di attuazione dei relé delle varie schede periferiche e precisamente *SendByte*, *EmergencyStop*, *SetRelay*, *ClearRelay* e *ToggleRelay*.

Il primo consente di attivare e disattivare a piacimento tutti i relé di una periferica specifica, attraverso l'invio di un byte in cui il bit 0 corri-

sponde al primo relé e il bit 7 all'ottavo. *EmergencyStop*, come dice il nome stesso, forza tutti i relé di tutte le schede periferiche a Off.

Set, *Clear* e *Toggle*, infine, permettono di attivare e disattivare un relé qualsiasi (*Set* e *Clear*) oppure di invertirne lo stato (*Toggle*).

Qualora si voglia utilizzare la libreria per realizzare una propria appli-

Realizzazione pratica

Per prima cosa dovete realizzare il circuito stampato monofaccia (il master può essere scaricato dal sito della rivista); iniziate quindi il montaggio partendo dai componenti a basso profilo, ossia lo zoccolo a 14+14 pin DIL, le resistenze, i diodi led (da montare leggermente ➤

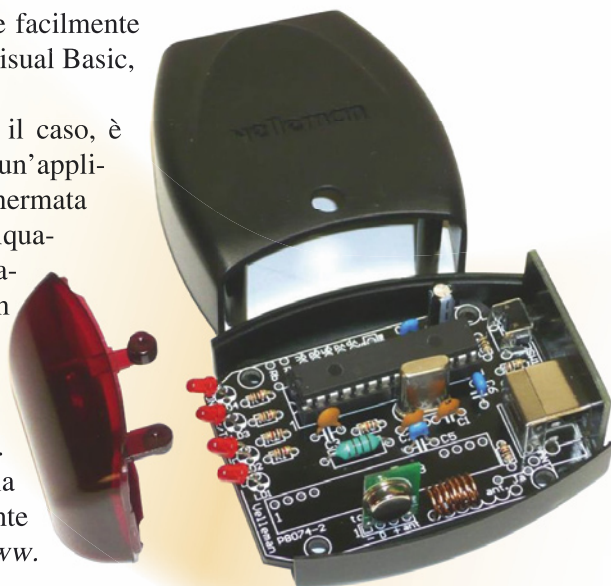
- Configurazione: autoapprendimento del trasmettitore; non occorre alcuna impostazione hardware.
- Canali in uscita: uno.
- Uscite: due, a relé con contatto in scambio, 3 A massimi, attivabili con singolo comando.
- Modalità delle uscite: configurabile singolarmente tra
 - bistabile
 - monostabile (0,5 s)
 - temporizzata programmabile tra 5 s, 30 s, 1 min, 5 min, 15 min, 30 min e 60 min.
- Comandi memorizzabili: fino a 31 trasmettitori differenti oppure fino a 31 pulsanti provenienti da trasmettitori differenti.
- Funzione di spegnimento di tutte le uscite.
- LED indicatori sia dello stato dell'uscita che delle operazioni di configurazione.
- Antenna interna o esterna, a scelta dell'utente.
- Alimentazione: 9 ÷ 12 V AC/DC, 100 mA massimi.
- Frequenza di ricezione: 433 MHz.
- Copertura: in funzione del trasmettitore; 30 metri circa in aria libera con K8074. Utilizzare un'antenna a filo per estendere la portata.
- Dimensioni: 80 x 70 x 25 mm.

Specifiche tecniche K8070

cazione, essa può essere facilmente inclusa in un progetto Visual Basic, DotNet, eccetera.

Se, però, non è questo il caso, è comunque disponibile un'applicazione generica (la schermata unica è visibile nel riquadro dedicato all'applicazione) sviluppata sia in DotNet che in Delphi, tramite la quale potrete controllare facilmente il trasmettitore K8074.

È possibile scaricare sia l'eseguibile che il sorgente di entrambe dal sito www.velleman.be.



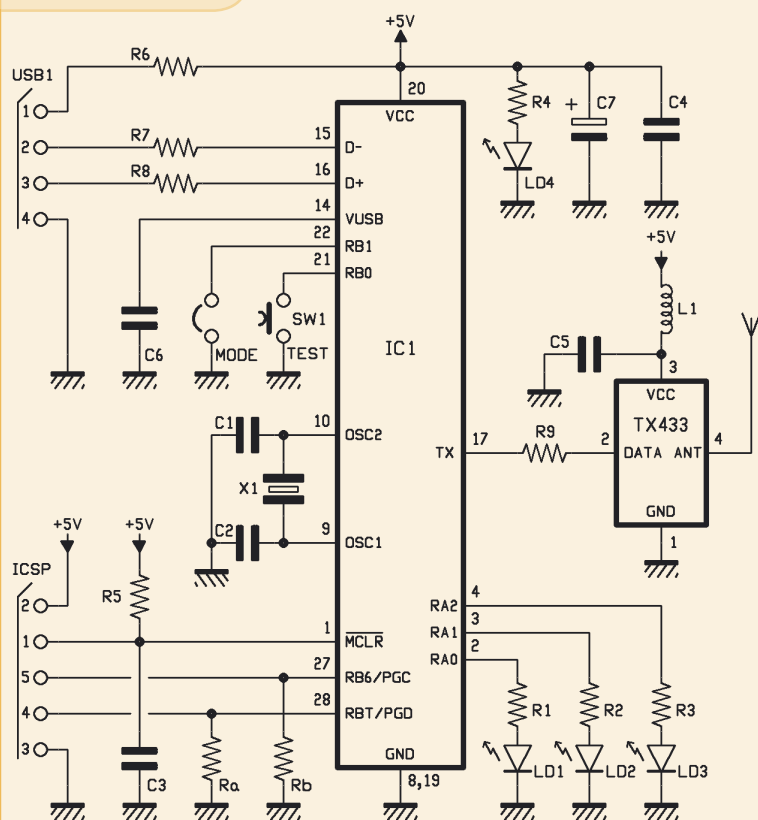
sollevati e piegati in avanti), l'induttanza e i condensatori non polarizzati. Montate, ora, il pulsante da stampato a 90°, il connettore USB, il quarzo, il condensatore elettrolitico ed il trasmettitore. Non rimane che realizzare l'antenna: prendete una punta da 5 mm e avvolgete su di essa sette spire con del filo di rame smaltato del diametro di 0,8 mm. Distanziate, infine, le spire in modo

gnò non aderirà al rame. Guardate attentamente il circuito, alla ricerca di eventuali errori e cortocircuiti, quindi inserite nell'apposito zoccolo il microcontrollore programmato (o da programmare tramite il connettore ICSP con il file esadecimale VK8074.hex) nel proprio zoccolo quindi racchiudete il tutto in un contenitore plastico, come quello che abbiamo previsto allo scopo.

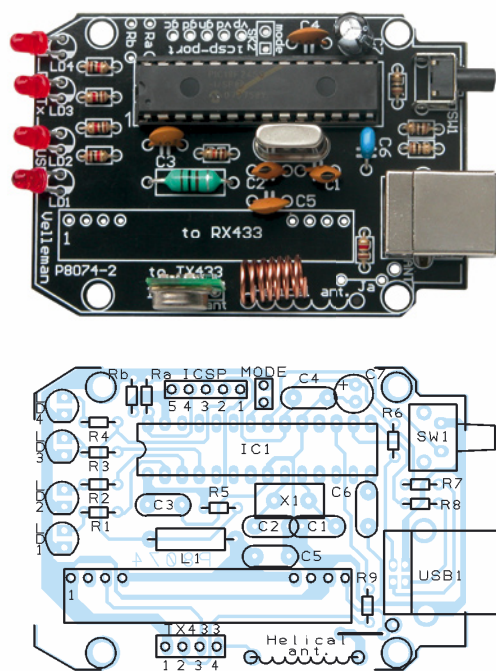
co, il circuito può essere idealmente suddiviso in tre blocchi; il primo è rappresentato dallo stadio di alimentazione.

La tensione d'ingresso, fornita da un adattatore di tensione da 9÷12 V AC/DC 100 mA, viene portata alla giusta polarità dal ponte di diodi D3÷D6, cui seguono la rete di filtro costituita da C5 e C6 (su quali è possibile prelevare una tensione

Trasmettitore



Il trasmettitore è molto semplice da realizzare. L'unico componente cui occorre prestare attenzione è l'antenna a solenoide, che va realizzata avvolgendo sette spire di filo smaltato da 0,8 mm di diametro su un diametro interno di 5 mm e distanziando le spire in modo da ottenere un solenoide lungo 10 mm. Prestate, inoltre, attenzione al verso di inserimento del trasmettitore ibrido siglato TX433.



ELENCO COMPONENTI:

R1÷R4, R9: 1 kohm	TX433: Modulo TX433N
R5÷R8: 10 kohm	SW1: Microswitch 90°
C1, C2: 22 pF ceramico	X1: Quarzo 4 MHz
C3÷C5: 10 nF multistrato	LD1÷LD4: led 3 mm rosso
C6: 220 nF multistrato	
C7: 4,7 µF	Varie:
16 VL elettrolitico	- Connettore USB-B
L1: Induttanza 100 µH	- Zoccolo 14+14
VR1: 78L05	- Contenitore plastico
IC1: PIC18F2450 (VK8074)	- Circuito stampato

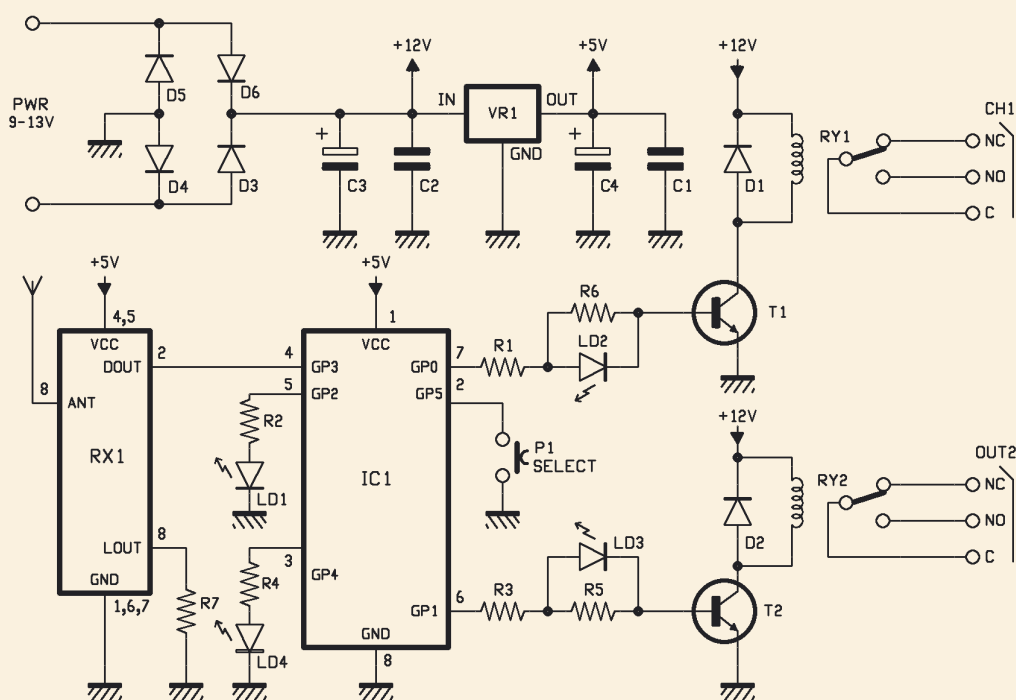
che la lunghezza dell'antenna sia di 10 mm (fate riferimento all'illustrazione sul piano di montaggio). Prima di saldare l'antenna allo stampato, rimuovete lo smalto da uno dei suoi capi, altrimenti lo sta-

Il ricevitore K8070

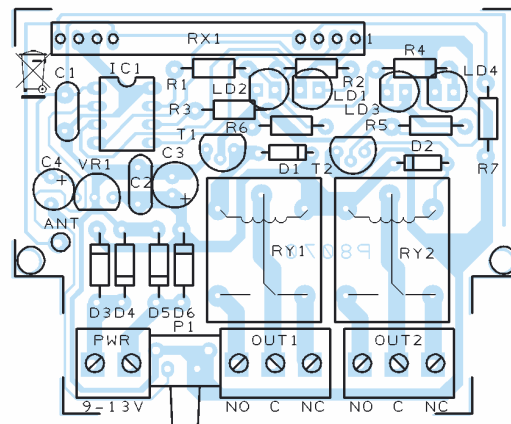
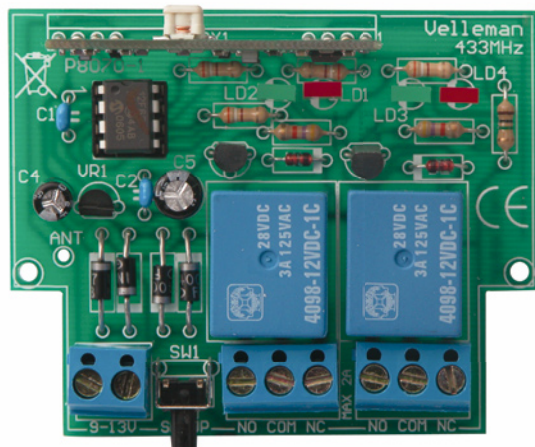
Così come il trasmettitore, anche il ricevitore è un dispositivo dalle dimensioni molto compatte e dal circuito semplice.

Come visibile dallo schema elettri-

continua di 11÷14 Vdc), il regolatore di tensione VR1 (un 78L05) ed un altro condensatore elettrolitico di filtro (C4) che ripulisce ulteriormente da eventuali disturbi residui la tensione di alimentazione a 5 V.



Come il trasmettitore, anche il ricevitore è molto semplice da montare: esso, infatti, è stato realizzato con soli componenti ad inserzione. Fate attenzione durante il montaggio dei led piatti: il loro lato superiore deve trovarsi allo stesso livello del lato superiore dei relé, così da sporgere dal contenitore senza impedirne la chiusura.



ELENCO COMPONENTI:

R1+R4: 390 Ohm
R5: 47 kohm
R6: 47 kohm
R7: 1 Mohm
C1, C2: 100 nF multistrato
C3: 100 μ F 25 VL elettrolitico
C4: 10 μ F 35 VL elettrolitico
T1: BC547

T2: BC547
D1: 1N4148
D2: 1N4148
D3÷D6: 1N4007
LD1, LD4: led rettangolare rosso
LD2, LD3: led rettangolare verde
P1: microswitch 90°
IC1: PIC12F629 (VK8070)
RX1: modulo ricevitore 433 MHz

VR1: 78L05
RY1: Relé 12V/3A - 1 scambio
RY2: Relé 12V/3A - 1 scambio

Varie:

- zoccolo 4+4
- morsettiera 2 poli
- morsettiera 3 poli (2 pz.)
- circuito stampato

Tale tensione alimenta sia il ricevitore RF a 433 MHz, il secondo dei tre blocchi, sia il microprocessore (l'ultimo blocco). Il ricevitore demodula il segnale RF proveniente dal trasmettitore e ne

ripropone il contenuto, in formato digitale, sulla sua uscita DOUT collegata direttamente all'ingresso GP3 del microcontrollore, un piccolo PIC12F629 in formato 4+4 pin DIL. Esso è stato programmato per

decodificare lo stream di dati proveniente dal ricevitore ed attuare i due relé in uscita, nonché per verificare lo stato del pulsante P1, utilizzato per configurare il dispositivo. I due relé, la cui bobina è alimenta-

ta direttamente a 11÷14 V, vengono comandati dalle linee GP0 e GP1 attraverso i transistor T1 e T2 (due NPN modello BC547). Le stesse linee comandano i led LD2 e LD3, utilizzati come indicatori dello stato di attivazione dei relé.

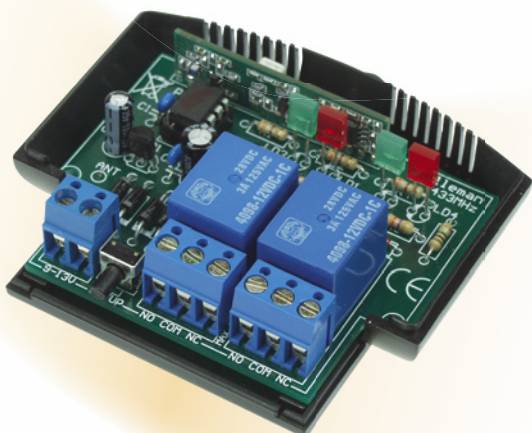
Prestate attenzione ad un particolare: nonostante i relé in uscita siano due, il dispositivo è monocanale; in altre parole, quando il ricevitore riceve un comando valido, esso attiva entrambe le uscite, ma con modalità di funzionamento che dipendono dalla configurazione effettuata.

Ad esempio, l'uscita 1 può operare in modalità toggle (inverte il proprio stato ad ogni comando ricevuto) mentre l'uscita 2 funziona come timer a 30 secondi.

Rimangono da descrivere le linee GP2, GP4 e GP5, utilizzate rispettivamente per gestire i led di segnalazione LD1 e LD4, e il pulsante di configurazione P1.

Realizzazione pratica

Realizzate il circuito stampato monofaccia ed iniziate il montaggio procedendo nel modo consueto: inserite, quindi, i componenti a basso profilo (lo zoccolo a 4+4 pin DIL, le resistenze, i diodi, eccetera). Montate, ora, il pulsante da stampato a 90°, i transistor, le morsettiere, i condensatori elettrolitici, i relé ed il ricevitore. Posizionate per ultimi il ricevitore e i quattro led piatti: il bordo superiore di questi ultimi deve risultare allo stesso livello della faccia superiore dei due relé ossia



Le API della libreria K8074.DLL

Il radiocontrollo K8074 può essere gestito tramite un programma in ambiente Windows. Tale programma, disponibile come eseguibile e come codice sorgente sia in linguaggio C per DOTNET che Delphi, è basato su una libreria, nominata K8074.DLL, che può essere utilizzata per realizzare facilmente degli applicativi di controllo personalizzati. Riportiamo di seguito tutte le chiamate a funzione e le definizioni per il corretto utilizzo della DLL.

HANDLE WINAPI OpenDevice(LPCSTR szPort);
Collega l'applicativo al radiocontrollo aprendo una porta specifica. La funzione restituisce il valore FALSE se il collegamento non può essere stabilito.

VOID WINAPI CloseDevice(HANDLE hDevice);
Chiude una connessione aperta utilizzando un particolare evento. Questa funzione deve essere sempre utilizzata per chiudere una connessione aperta tramite la chiamata alla OpenDevice.

DWORD WINAPI GetDllVersion(VOID);
Restituisce una variabile DWORD (32 bit) in cui la WORD più significativa (bit 16÷31) contiene la parte intera della versione FW della DLL, mentre i restanti bit definiscono la parte decimale.

BOOL WINAPI SendCommand(HANDLE hDevice, BYTE address, BYTE cmd, BYTE param);
Invia un comando al dispositivo K8074. Il valore del comando può essere uno qualsiasi dei parametri specificati con le seguenti definizioni:

```
#define K8074_SET_ADDRESS 0x41 // 'A'
#define K8074_SEND_BYTE 0x42 // 'B'
#define K8074_CLEAR_RELAY 0x43 // 'C'
#define K8074_DISPLAY_ADDRESS 0x44 // 'D'
#define K8074_EMERGENCY_STOP 0x45 // 'E'
#define K8074_FORCE_RESET 0x46 // 'F'
#define K8074_SET_RELAY 0x53 // 'S'
#define K8074_TOGGLE_RELAY 0x54 // 'T'
```

Param è un valore numerico compreso tra 1 e 8 che definisce il relé da comandare sulla scheda ricevente.

BOOL WINAPI SendByte(HANDLE hDevice, BYTE address, BYTE bMask);
Funzione che permette di attivare qualsiasi relé attraverso un valore binario (bitmask): inviando 3 (00000011) verranno attivati il primo e il secondo relé, inviando 6 (00000110) il secondo e il terzo, eccetera.

BOOL WINAPI EmergencyStop(HANDLE hDevice);
Funzione che permette di stop di emergenza al K8074. Tutti i relé di tutte le schede gestite dal trasmettitore vengono disattivati.

BOOL WINAPI DisplayAddress(HANDLE hDevice);
Mostra l'indirizzo della scheda K8074 in modo binario attivando il relé corrispondente.

BOOL WINAPI SetRelay(HANDLE hDevice, BYTE address, DWORD dwRelayId);
Attiva un relé specifico. L'identificativo del relé è compreso tra 1 e 8.

BOOL WINAPI ClearRelay(HANDLE hDevice, BYTE address, DWORD dwRelayId);
Disattiva un relé specifico. L'identificativo del relé è compreso tra 1 e 8.

BOOL WINAPI ToggleRelay(HANDLE hDevice, BYTE address, DWORD dwRelayId);
Cambia lo stato di un relé specifico. Se il relé è OFF, passa ad ON e viceversa. L'identificativo del relé è compreso tra 1 e 8.

BOOL WINAPI SetAddress(HANDLE hDevice, BYTE oldAddress, BYTE newAddress);
Cambia l'indirizzo del K8074. Per conoscere l'indirizzo impostato è necessario chiamare la funzione DisplayAddress; in alternativa, è possibile forzare l'indirizzo a 1 tramite la funzione ForceAddress.

BOOL WINAPI ForceAddressReset(HANDLE hDevice);
Imposta l'indirizzo a 1.

I driver per Windows

Il radiocontrollo K8074 viene gestito dal PC tramite l'interfaccia USB: il PC riconosce il dispositivo nell'esatto istante in cui questi viene collegato alla porta. A questo punto appare la prima schermata della classica installazione guidata di Windows, in cui viene richiesto di cercare il driver in rete. Scegliete l'opzione



che consente di specificare manualmente il percorso verso la cartella in cui avete salvato i driver del dispositivo, preventivamente scaricati dal sito www.velleman.be. Il driver è disponibile in due versioni, una per Windows Vista e una per Windows 2000/XP, in entrambi i casi nominato K8074.inf.

a 15 mm di altezza circa. Se non fate così, non potrete poi, racchiudere il circuito nel contenitore che abbiamo previsto per esso.

Non è necessario dotare il dispositivo di un'antenna esterna: la sensibilità del ricevitore è tale da garantire una copertura di almeno 30 metri in campo aperto con il trasmettitore K8074. Qualora, però, vogliate estendere tale portata, potrete saldare alla piazzola nominata ANT un cavetto plastificato lungo 30 cm, di sezione 0,5 mm².

Anche in questo caso, verificate in modo visivo che non vi siano errori, quindi inserite nel proprio zoccolo il micro programmato con il file VK8070.hex e racchiudete il dispositivo nel contenitore. Prima di fare ciò, però, ricordate di collegare alle rispettive morsettiere il cavo

di alimentazione e i cavi con cui i contatti dei relé commanderanno i due carichi.

Note per l'utilizzo

È giunto il momento di verificare che i due dispositivi appena realizzati funzionino correttamente. Iniziamo con l'installazione del trasmettitore sul PC.

Per prima cosa, scaricate dal sito <http://www.velleman.be/be/en/download/files/> il file *K8074_software_rev2.zip*, che include

- l'applicativo di controllo in formato eseguibile e sorgente sia per Delphi (K8074.exe) che per DOTNET (K8074.NET.exe),
- la libreria K8074.DLL ed un codice sorgente generico utilizzabile come esempio di utilizzo,
- il manuale della libreria,

• il driver USB in due versioni, una per Vista e una per Win2000/XP. Estraiete il contenuto del file in una cartella di servizio, quindi collegate il trasmettitore ad una porta USB libera. Il PC identifica immediatamente il dispositivo *K8074 USB to RF transmitter* ed avvia la procedura con la quale associa ad esso una comport virtuale. A questo punto appare la classica schermata di Windows con la quale il sistema operativo ricerca il driver in rete. Negate il consenso alla ricerca in rete e fate clic su **Avanti**: appare una nuova finestra in cui dovette scegliere l'opzione "Installa da un elenco o percorso specifico (per utenti esperti)" e fate nuovamente clic su **avanti**. Nella terza schermata, ora, dovette specificare (passando attraverso il pulsante **Sfoglia**) ➤

Tabella 1 - Configurazione del ricevitore

Azione	Sequenza operazioni	Note:
Accensione	1) LD1 tre lampeggi 2) LD1 spento. LD4 acceso a luce fissa	Nessun intervento da parte dell'utente.
Autoapprendimento	1) Mantenere premuto il pulsante setup del ricevitore. LD1 si accende a luce fissa. 2) Mantenere premuto un pulsante qualsiasi sull'interfaccia grafica di controllo. 3) Comando riconosciuto: LD1 e LD4 accesi a luce fissa. Pulsante/trasmettitore memorizzato. 4) Rilasciare tutti i pulsanti. LD4 acceso a luce fissa.	- Possono essere memorizzati fino a 31 pulsanti/trasmettitori. - A memoria piena i led LD1 e LD4 lampeggiano rapidamente.
Cancellazione di tutta la memoria per ripristinare la configurazione predefinita	1) Spegner il ricevitore. 2) Mantenere premuto il pulsante setup e accendere il ricevitore. 3) LD1 e LD4 iniziano a lampeggiare. 4) Rilasciare il pulsante setup non appena LD1 e LD4 smettono di lampeggiare.	- Questo processo necessita di circa 10 secondi. - Rilasciando il pulsante prima che i led smettano di lampeggiare, la memoria non viene cancellata. - L'unità, ora, non risponde ad alcun indirizzo.
Modo di funzionamento dei relé	1) Premere ripetutamente il pulsante setup per selezionare il relé da configurare. Il relé commuta ed il relativo led verde si accende. 2) Confermare la scelta del relé mantenendo premuto il tasto setup fino a quando LD4 non lampeggia tre volte. Il relé scelto resta eccitato. 3) Premere il pulsante setup il numero di volte necessario per scegliere il modo di funzionamento (una pressione corrisponde alla modalità toggle, due alla modalità monostabile e così via, vedere la tabella a lato). Ad ogni pressione, LD1 lampeggia un numero di volte pari al modo attualmente definito. 4) Premere e mantenere premuto il pulsante per memorizzare il modo di utilizzo. LD4 lampeggia tre volte per confermare che l'impostazione è stata effettuata.	L'unità ritorna al modo operativo normale se viene lasciata inattiva per dieci secondi.

Lampeggi LD1	Modo relé
1	Toggle (ON/OFF)
2	0,5 s (monostabile)
3	Timer 5 s
4	Timer 30 s
5	Timer 1 min
6	Timer 5 min
7	Timer 15 min
8	Timer 30 min
9	Timer 1 h

Il programma di controllo è molto intuitivo: all'avvio, sono abilitati solamente i controlli *combobox* per la selezione della comport e *pulsante* per aprire il collegamento con il trasmettitore, entrambi posizionati in alto a sinistra.

Cliccando su Connect, viene avviata la comunicazione con il dispositivo e contemporaneamente, tutti gli altri controlli vengono abilitati. Partendo dall'alto, troviamo la prima sezione *Address*, che contiene due *combobox* e tre pulsanti. I pulsanti permettono di verificare l'indirizzo corrente del trasmettitore (pulsante *show addr.*; il valore appare nella combo box di sinistra), di impostarlo (*set addr.*) al valore definito nella *combobox* di destra, o di forzarlo all'indirizzo predefinito (1, pulsante *Reset addr. To 1*). Seguono, quindi tre sezioni che contengono 18 pulsanti la prima e 8 pulsanti le altre due.

La prima sezione consente di attivare le varie uscite delle schede a relé in modalità ON/OFF. Sono previsti due pulsanti che permettono di attivare (*SET ALL*) o disattivare

(*CLEAR ALL*) tutti i relé contemporaneamente.

La seconda sezione, contiene otto pulsanti utilizzabili per comandare in modalità toggle i relé.

L'ultima sezione, infine, contiene altri otto pulsanti con cui comandare in modalità monostabile le varie uscite.

È, infine, previsto un unico grande pulsante che permette di effettuare lo stop di emergenza, ossia lo spegnimento contemporaneo di tutte le uscite. Notate che ogni pulsante esegue un'azione per la quale esiste una funzione di libreria specifica.



la cartella dove avete salvato tutti i file visti in precedenza. In particolare, dovreste identificare quella che contiene il driver adatto al vostro sistema operativo. Fate clic nuovamente su Avanti per terminare l'installazione: a questo punto, potrebbe apparire una schermata con cui il sistema operativo vi notifica che il driver non ha superato i test per l'utilizzo con il logo di Windows; fate comunque clic su Continua ed attendete la schermata finale, con la quale il sistema operativo termina l'installazione della periferica. Fate clic su Fine e verificate, tramite il gestore delle periferiche, che sul vostro PC sia ora presente una Communications Port.

Avviate il programma K8074.exe

ed aprite il collegamento con la comport assegnata al dispositivo USB (nel nostro caso è la com 2): non appena il canale di comunicazione viene attivato, tutti i controlli visibili nell'interfaccia grafica del programma vengono abilitati.

Guardate, ora, lo stato dei led del trasmettitore: il primo deve essere acceso a luce fissa (Power), il terzo e il quarto devono lampeggiare alternativamente, segnalando che la porta USB è attiva. Il secondo si deve illuminare per un breve istante ogni volta che cliccate un tasto sul programma di controllo oppure premete il pulsante di test. Accendete il ricevitore: il led LD1 effettua tre lampeggi quindi si spegne; ora LD4 si accende a luce fissa, a conferma

del corretto funzionamento del dispositivo.

L'ultima operazione da eseguire consiste nell'accoppiare il trasmettitore con il ricevitore: mantenete premuto il pulsante SELECT (P1) del ricevitore (LD1 acceso, LD4 spento) e premete uno qualsiasi dei pulsanti presenti sul programma di controllo (ad esempio il Momentary 8). LD1 si accende a luce fissa, segnalando di aver riconosciuto e memorizzato il telecomando. Ora potete rilasciare il pulsante per concludere la fase di autoapprendimento. Cliccate il pulsante Momentary 8 e verificate che i relé si attivino per tutta la durata della pressione del tasto.

Procedendo nello stesso modo potete accoppiare al ricevitore più pul-

per il

MATERIALE

I progetti descritti in queste pagine sono disponibili in scatola di montaggio (i kit vengono prodotti dalla società Velleman e distribuiti da Futura Elettronica). Il trasmettitore (cod. K8074) e il ricevitore (cod. K8070) costano rispettivamente 30,00 e 24,00 Euro, IVA compresa. Le scatole di montaggio comprendono tutti i componenti, la basetta forata e serigrafata, il micro programmato, i connettori, il contenitore plastico nonché tutte le minuterie meccaniche.

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

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

Ricevitore a 8 canali

Questa scheda a relé (K8056), ampiamente descritta sul fascicolo N. 100, si presta bene per differenti applicazioni. Gli otto relé di cui dispone, infatti, possono essere controllati da PC attraverso una connessione seriale RS232 o current-loop, oppure via radio tramite un telecomando come il K8074 presentato in questo articolo o il K8058 presentato sul fascicolo N. 100. Può, anche, essere impiegata come semplice interfaccia a relé con schede elettroniche che dispongano di sole uscite open collector. I relé possono funzionare in modalità bistabile oppure impulsiva.

Specifiche

- 8 uscite a relé, 5A/230Vac.
- Varistore di protezione su ogni contatto.
- LED indicatore di stato per ogni uscita.
- 8 ingressi open collector / interruttore.
- Ingresso di controllo RS232 per comandare la scheda attraverso un computer o un terminale: baud rate di 2.400 bps, nessuna parità, 8 bit di dati, 1 bit di stop.
- Possibilità di comandare la scheda attraverso segnali RF.
- Uscita in tensione: non regolata 12 V continui, 200 mA.
- Alimentazione: 12 Vac, 500 mA, tenendo conto dei 200 mA prelevabili sull'uscita a 12 V.
- Software di test scaricabile dal sito www.velleman.be.



santi e più trasmettitori (fino ad un massimo di 31).

Relativamente all'impostazione del modo di funzionamento delle uscite a relé, fate riferimento alla Tabella 1, che riporta le varie procedure nel

dettaglio. Qui vi anticipiamo che, nonostante entrambi i relé vengano comandati dallo stesso evento, il modo di funzionamento di ognuno di essi può essere impostato liberamente tra nove differenti. È pos-

sibile, ad esempio, decidere che il primo relé funzioni in modalità timer mentre il secondo possa essere comandato in toggle. Ciò conferisce al sistema una maggiore flessibilità d'impiego.

Tutti i prezzi si intendono IVA inclusa.

Sistemi di sviluppo per

SMART CARD

SET DI SVILUPPO PER APPLICAZIONI SMART-CARD



SDK-ACR38
€ 149,00

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

Contenuto della confezione:

- 1 ACR38 Smart Card Reader
- Interfaccia USB full speed;
- Legge e scrive le carte a microprocessore ISO7816 1, 2, 3, 4 (protocolli T=0 e T=1);
- Supporta carte a 1,8 V, 3 V, 5 V MCU, con sistema interno di protezione da corto circuito e le più comuni carte memoria tra cui le IC2 bus e le secure memory;
- Supporto per sistemi operativi: Win 98, Me, 2000, XP, XP-SP1, XP-SP2;
- Alimentazione: 5 Vdc mediante interfaccia USB.
- 20 Smart-Card per test
- 1 ACR38T SIM-size plug-in Smart Card Reader
- Dispositivo di interfaccia di dimensioni molto contenute per Smart-Card in formato SIM plug-in con caratteristiche simili a quelle dell'ACR38.
- 1 ABRO8LS Balance Reader
- È un visualizzatore con display ad otto digit per la lettura dei dati da Smart-Card.
- 1 CD-ROM con software e driver di supporto

LETTORE SMART-CARD

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



MINILECTOR
€ 36,00

applicazioni come la carta sanitaria, eCommerce, logon, identificazione, pagamento, borsellino e loyalty (carte per sconti e punti). La Confezione non include CD ed altri supporti per l'installazione essendo rivolta esclusivamente ad installatori ed integratori. I driver ed il software di supporto è disponibile on-line tramite download.

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

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



SMART-CARD
ACOS2 DA 8 K
ACOS2-8K
€ 3,90

viene fornita completamente bianca senza alcuna personalizzazione grafica.

SMART CARD SECURE ACCESS MODULE 8K
ACOS3SAM € 22,00

FUTURA ELETTRONICA idee in elettronica

Via Adige, 11 • 21013 GALLARATE (VA)
Tel. 0331/799775 • Fax 0331/792287
www.futurashop.it

Disponibili presso i migliori negozi di elettronica o nel nostro punto vendita di Gallarate (VA).
Caratteristiche tecniche e vendita on-line: www.futurashop.it



Cartellino orario, il software e il firmware

di Boris Landoni

Progettato per monitorare le entrate e le uscite dal luogo di lavoro, impiega per il riconoscimento delle persone un lettore di impronta digitale professionale che rende superfluo l'utilizzo di schede, tessere e transponder esterni. In questo articolo analizziamo la routine utilizzata per riconoscere i vari comandi inviati tramite PC; prendiamo, quindi, confidenza con il set di comandi e con il programma Windows per il download e la gestione del file timbrature. Ultima puntata.

Nel corso delle due puntate precedenti abbiamo analizzato le procedure per realizzare in pratica il cartellino orario, quelle relative alla configurazione e all'utilizzo in modalità stand-alone e il metodo da adottare per esportare manualmente il file timbrature su un foglio di calcolo Excel. In tali circostanze, abbiamo più volte avuto modo di dirvi che il dispositivo può essere completamente telegestito tramite una postazione PC remota, collegata al cartellino attraverso un link radio (radiomodem Aurel XTR7020A) oppure Ethernet (convertitore seriale-Ethernet Tibbo EM202). Bene, in

quest'articolo conclusivo vedremo come si effettua la telegestione da remoto, iniziando con la descrizione del set di comandi, e terminando con le note per l'utilizzo del programma Windows che abbiamo realizzato appositamente. Abbiamo, infine, dedicato un riquadro a parte alla sezione di firmware utilizzata dal microcontrollore per effettuare tanto il riconoscimento dei vari comandi inviati dal PC quanto la gestione di tutte le operazioni sul file timbrature. A questo punto non ci sono molte altre parole da aggiungere: vediamo subito come gestire il dispositivo dalla propria scrivania. ➤

Il set di comandi

Prima di iniziare a parlare dei vari comandi, è necessario fare una breve premessa: il collegamento tra PC e cartellino orario, a prescindere dal supporto per il collegamento fisico utilizzato (radiomodem o bridge Ethernet), emula una porta seriale con formato dati a 38.400,N,8,1 (38.400 baud, nessuna parità, 8 bit dati e uno stop bit) in quanto sia il convertitore Ethernet che il radiomodem comunicano con il microcontrollore del cartellino attraverso una seriale standard. Ciò significa che i comandi inviati al dispositivo devono essere generati da un applicativo software in grado di aprire una porta seriale virtuale verso il cartellino orario, e nel paragrafo successivo vedremo come.

Esaurita questa breve premessa, pas-

siamo subito ad analizzare i vari comandi (riportati nella Tabella 1) che il cartellino orario è grado di accettare. Immediatamente dopo l'accensione, il dispositivo avvia la normale attività che comprende, oltre la gestione delle timbrature, il controllo periodico della seriale collegata al radiomodem o al bridge Ethernet. Il controllo è necessario in quanto il cartellino deve riconoscere il comando di connessione da remoto “*/”, che può essere inviato dal PC in qualsiasi momento. Quando tale comando viene riconosciuto, il dispositivo conferma la ricezione del comando con il messaggio “OK” quindi si pone in uno stato di attesa dei vari comandi, inibendo tutte le funzioni di timbratura e comunicando lo stato di momentanea inattività con il messaggio a display visibile in Fig. 1.

Fig. 1 - Il display segnala quando il cartellino è collegato al PC.



L'uscita dalla modalità telegestione viene effettuata inviando il comando “*/X” cui il cartellino orario risponde con “EXIT”. Immediatamente dopo l'uscita da tale modalità, le normali funzioni di timbratura vengono riattivate. La telegestione consente di

- leggere e scrivere l'orologio,
- effettuare il download dell'intero file timbrature, delle timbrature di un giorno, di un mese, di un utente o di una sola timbratura specifica,
- effettuare il download delle sole timbrature odierne,
- leggere le dimensioni del file TIM-

Tabella 1 - Il set di comandi seriali

Funzione	Comando	Risposta	Note
Attivazione telegestione	*/	OK	Modo comandi avviato.
Attivazione gestione orologio	*/O	OROLOGIO (a comando ricevuto) ! (timeout ricezione richiesta lettura/scrittura)	Attende due secondi il comando lettura/scrittura. Se non arriva, termina la gestione dell'orologio.
Lettura data-ora	*R	AA/MM/GG hh:mm:ss	Entro due secondi dal comando */O
Scrittura data-ora	*W*[AA][MM][GG][hh][mm][ss]	AA/MM/GG hh:mm:ss	Entro due secondi dal comando */O. I parametri sono a due cifre.
Download file timbrature	*/D	DOWNLOAD	Inizia il download delle timbrature.
Download timbrature di un solo giorno	*/G	RICERCA GIORNO	Attende l'inserimento del giorno. I parametri sono a due cifre.
Inserimento giorno	*[AA][MM][GG]	RICERCA AA/MM/GG	Inizia il download. Può essere dato solo dopo il comando */G.
Download timbrature di un solo mese	*/M	RICERCA MESE	Attende l'inserimento del mese. I parametri sono a due cifre.
Inserimento mese	*[AA][MM]	RICERCA AA/MM	Inizia il download. Può essere dato solo dopo il comando */M.
Download timbrature di un solo utente	*/U	RICERCA UTENTE	Attende l'inserimento dell'utente. Parametro a tre cifre.
Inserimento utente	*[UTENTE]	RICERCA [UTENTE]	Inizia il download. Può essere dato solo dopo il comando */U.
Download di una singola timbratura	*/T	RICERCA TIMBRATURA	Attende l'inserimento dell'utente. Parametro ESADECIMALE a otto cifre.
Inserimento mese	*[TIMBRATURA]	RICERCA [TIMBRATURA]	Inizia il download. Può essere dato solo dopo il comando */T.
Cancellazione file timbrature	*/C	CANCELLA TIMBRA.TXT	Attenzione: non viene richiesta la conferma. A cancellazione eseguita, il file è perso.
Richiesta dimensioni file timbrature	*/S	DIMENSIONE TIMBRA.TXT SIZE [byte]	Non esegue alcuna operazione sul file.
Disattivazione telegestione	*/X	EXIT	Modo comandi disattivato.

Installazione del Radiomodem

	DIP-1	DIP-2	S5	Velocità/Modo
DS1	OFF	OFF	0	9.600 bps
	OFF	ON	0	19.200 bps
	ON	OFF	0	38.400 bps
	ON	ON	0	Test: pseudonoise
	OFF	OFF	1	9.600 bps
	OFF	ON	1	57.600 bps
	ON	OFF	1	115.200 bps
	ON	ON	1	Test: pacchetti dati

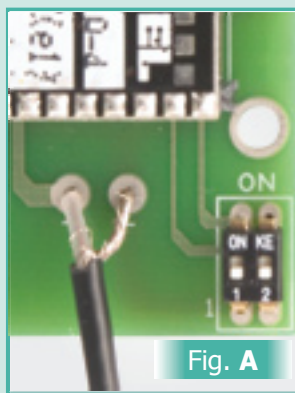


Fig. A

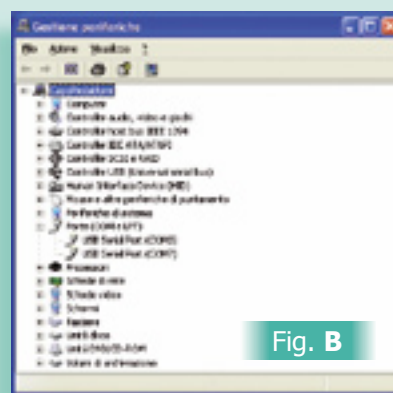


Fig. B

Prima di collegare il radiomodem al PC, è necessario scegliere la velocità di collegamento: poiché il programma di telegestione comunica con il cartellino a 38.400 bps, occorre impostare sul radiomodem il DIP1 a ON e il DIP2 a OFF (nota: i dip di Fig. A sono entrambi a OFF, corrispondenti a 9.600 bps). Sul cartellino orario la configurazione viene effettuata direttamente dal microcontrollore. A questo punto potete collegare il radiomodem al PC: se disponete di un sistema operativo molto recente (Win XP o Vista), questi identifica il convertitore seriale-USB modello FT232RL (contenuto nel radiomodem) ed installa automaticamente i driver. Assegna, infine, una COM virtuale al dispositivo,

che viene riportato nell'elenco delle periferiche installate (Fig. B). Con sistemi operativi meno recenti, è probabile che il dispositivo venga riconosciuto, ma non possa essere installato in quanto i driver non sono presenti tra quelli di sistema. Occorre, quindi scaricare i driver aggiornati (attualmente V 2.04.06) dal sito <http://www.ftdichip.com/Drivers/VCP.htm> e procedere seguendo le istruzioni impartite dal sistema operativo secondo la normale procedura di installazione di una periferica, avendo cura di specificare il percorso corretto verso la cartella dove avete salvato i driver. Anche in questo caso, la procedura termina con la creazione di una COM virtuale.

BRA.TXT e cancellarlo.

Il comando per avviare le procedure di lettura/scrittura dell'orologio è `"*/O"`, che genera il messaggio di conferma `"OROLOGIO"`. La successiva lettura o scrittura deve essere effettuata entro due secondi dalla ricezione del messaggio di conferma: `"*R"` è il comando di lettura, cui il cartellino risponde con una stringa formattata come segue

AA/MM/GG hh/mm/ss

(es. 08/03/03 11:24:40)

in cui il primo campo emesso è l'anno (AA).

Mettendo l'anno come primo campo, si ottiene una stringa di valore numerico assoluto incrementale, che rende facilmente gestibile l'ordinamento dei dati secondo data crescente oppure decrescente. Tutti i campi numerici sono visualizzati a due cifre. Il comando per la scrittura della data, invece, è `"*W*"` seguito da anno, mese, giorno, ora, minuti e secondi, ognuno espresso a due cifre, secondo la sintassi

`*W*[ANNO][MESE][GIORNO][O`

`RA][MINUTI][SECONDI]`.

Se la sintassi è stata rispettata, il cartellino orario riconosce le nuove impostazioni, le memorizza e conferma l'esecuzione inviando come risposta i nuovi parametri data/ora.

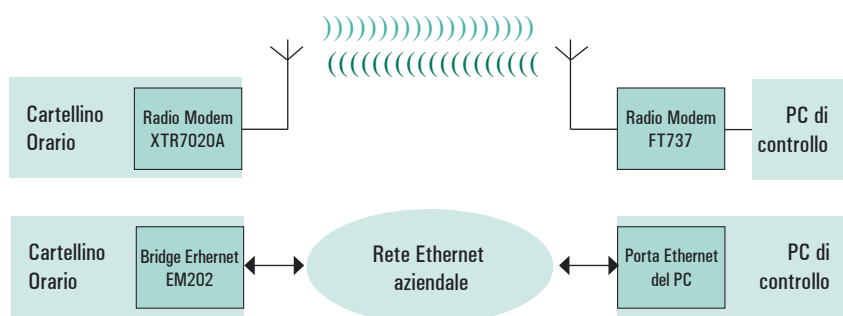
Terminata la descrizione dei comandi specifici per l'orologio, vediamo quelli relativi al download dei dati.

Il primo di essi, `"*/D"`, avvia il download immediato di tutto il file timbrature. Dopo la trasmissione del messaggio di conferma `"DOWNLOAD"`, inizia l'invio di tutti i record memorizzati nel file TIMBRA.TXT, a prescindere dalle sue dimensioni. Questo comando è particolarmente utile quando si sa a priori che le dimensioni del file sono abbastanza contenute (ciò avviene ad esempio nelle aziende con pochi dipendenti). Se, invece, il numero di utenti registrati è elevato, il rischio che le dimensioni del file aumentino sensibilmente di giorno in giorno è più marcato. In questo caso, prima di avviare il download completo, conviene utilizzare il comando `"*/S"` che

comunica le dimensioni del file TIMBRA.TXT. Così facendo, la persona incaricata per lo scarico del log potrà decidere, in base alle dimensioni del file, se proseguire da remoto oppure recarsi nel luogo dove è installato il dispositivo ed effettuare lo scarico del file rimuovendo la pen drive.

È comunque possibile effettuare dei download mirati: il comando `"*/G"`, infatti, consente di scaricare tutti gli eventi relativi ad un solo giorno. Il cartellino risponde con in messaggio di conferma `"RICERCA GIORNO"` e si pone in attesa della data, che deve essere inviata secondo la sintassi `*[ANNO][MESE][GIORNO]`, in cui i parametri vanno inseriti a due cifre senza le parentesi quadre, che noi utilizziamo (qui e in tutti gli altri comandi) solamente per rendere più evidenti i vari campi. Entro pochi istanti il dispositivo inizia ad inviare solamente il log degli eventi relativi a quella data. Analogamente, i comandi `"*/M"` e `"*/U"` consentono di accedere alle timbrature di un mese intero il primo (RICERCA MESE) e di ➤

Fig. 2 - Collegamento fisico tra cartellino orario e PC



un solo utente il secondo (RICERCA UTENTE). Con entrambi i comandi, il cartellino si pone in attesa dei relativi parametri di filtro e precisamente `*[ANNO][MESE]` nel primo caso e `*[UTENTE]` (parametro a tre cifre) nel secondo. Notate che per UTENTE si intende l'identificativo numerico a tre cifre, non il nome descritto nel file NOMI.TXT. Il comando `"/T"`, infine, permette di scaricare una timbratura specifica

noto il suo indirizzo esadecimale a otto cifre. La risposta di conferma al comando è "RICERCA TIMBRATURA", dopo la quale il cartellino resta in attesa del comando `*[TIMBRATURA]`. Dopo aver effettuato lo scarico del file, potrebbe capitare, specie all'inizio del mese successivo, di doverlo cancellare in quanto non più necessario. Ciò può essere effettuato molto facilmente utilizzando l'ultimo dei

comandi e precisamente `"/C"`, cui il cartellino risponde con il messaggio di conferma CANCELLA TIMBRA.TXT.

Il collegamento fisico

In Fig. 2 abbiamo schematizzato il principio del collegamento per entrambi i tipi di supporto fisico: notate che, dal punto di vista hardware, sono molto simili; le procedure per creare il canale dati, però, sono leggermente differenti in funzione del tipo di supporto scelto. Nel caso in cui abbiate dotato il cartellino orario di un radiomodem XTR7020A (a prescindere dalla frequenza di trasmissione), dovrete necessariamente dotare anche il PC di un modem analogo. A tal proposito, potete utilizzare il radiomodem USB presentato sulla rivista N. 125 (codice progetto FT737), dispositivo che abbiamo progettato appositamente per l'impiego con il nostro cartellino. Poiché il radiomodem e tutte le procedure che lo riguardano sono state ampiamente descritte nel corso di quell'articolo, qui ci limitiamo a dirvi che sia il cartellino orario che il radiomodem connesso al computer devono essere dotati dello stesso modulo RTX (XTR7020A4 a 433 MHz, oppure XTR7020A8, a 868 MHz). Per quanto riguarda i dettagli dell'installazione, vi rimandiamo alla rivista 125 oppure al riquadro specifico presente in queste pagine. Giunti a questo punto, diamo per scontato che il radiomodem sia installato ed assegnato ad una comport virtuale sul vostro PC.

Se, invece, avete optato per il bridge seriale-Ethernet, dovrete procedere con l'installazione di un driver particolare che consenta di creare una nuova comport virtuale verso il modulo Tibbo EM202 attraverso la rete Ethernet.

Innanzitutto, dovete scaricare dal sito www.tibbo.com (previa registrazione gratuita) la suite di programmi per la configurazione dei bridge seriali chiamata "Device Server Toolkit



Fig. 3

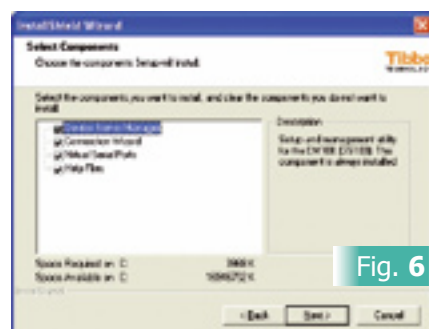


Fig. 6



Fig. 4



Fig. 7

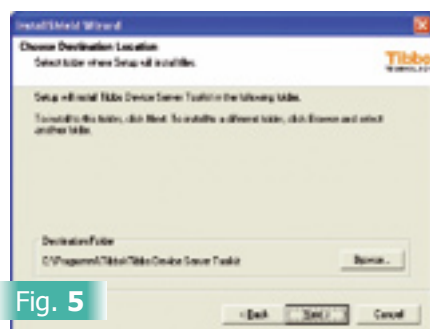


Fig. 5



Fig. 8

Fig. 9

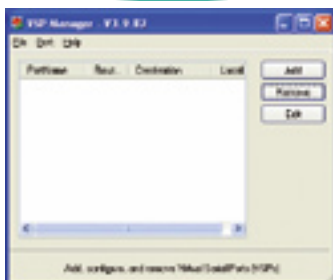


Fig. 10



(DST) for Windows”, raggiungibile attraverso l’hyperlink *Download*: la versione attuale è la 3.9.82. Il passo successivo consiste nell’installazione: decomprimete in una cartella di servizio il file Zip scaricato, quindi fate doppio clic sul programma di installazione SETUP.EXE; appare immediatamente la schermata visibile in Fig. 3. Fate clic su Next confermando le richieste effettuate nelle due schermate successive (Fig. 4 e 5). Nella quarta schermata potete decidere quali tool installare: noi suggeriamo di eseguire l’installazione completa (Fig. 6) in quanto l’ingombro di tutta la suite è inferiore ai 30 Mbyte. Se

volete, potete effettuare l’installazione in una cartella differente da quella predefinita (Fig. 7). Fate clic su Next per completare l’installazione quindi su Finish (Fig. 8) per terminare l’esecuzione del programma. Ora nel vostro PC sono presenti tutti i driver e tool software necessari per poter creare la comport virtuale in collegamento diretto con il modulo EM202 a bordo del cartellino orario.

Tramite il menu di avvio, mandate in esecuzione il programma VSP Manager che, secondo l’installazione predefinita, viene avviato attraverso Start/Programmi/Tibbo/VSP Manager. Questo programma, la cui schermata principale è visibile in Fig. 9, permette di identificare il dispositivo Tibbo EM202 (che deve essere collegato in rete ed acceso), di assegnare ad esso una comport virtuale e di configurarlo. Notate che, alla prima attivazione, il VSP Manager non mostra alcun dispositivo in quanto non è stata ancora effettuata la procedura per l’aggiunta di un nuovo bridge. Per aggiungere il nostro EM202, occorre fare clic su Add, in modo da far apparire la finestra di Fig. 10 che contiene tutti i campi impostabili relativi al modulo EM202, ma che non è ancora stata configurata in quanto non ancora associata ad un dispositivo. Fate clic sul tasto “Select Device Server from the list...”: appare ora la schermata di Fig. 11 che, in automatico, esegue la ricerca di tutti i Tibbo presenti in rete. La figura mostra che è stato individuato un dispositivo già configurato: lo si capisce dal fatto che



ad esso risulta assegnato un indirizzo IP; se, invece, fosse stato da configurare, l’indirizzo IP mostrato sarebbe stato quello predefinito (127.0.0.1). La configurazione viene effettuata cliccando il pulsante Settings, che richiama una finestra contenente differenti “cartelle” (in Visual Basic vengono chiamate Pannelli di controllo); quelle che ci interessano sono visibili in Fig. 12 e 13: esse consentono di modificare l’indirizzo IP del modulo Tibbo (Fig. 12) e di configurare il modo di funzionamento dei parametri seriale, come ad esempio il tipo di controllo di flusso (che nel nostro caso è disabilitato, Fig. 13). Confermate le impostazioni facendo clic su OK: la finestra viene chiusa. Fate clic sull'icona vicino all’indirizzo MAC (Fig. 11) per caricare la finestra di Fig. 14, in cui viene assegnato automaticamente l’indirizzo IP del modulo EM202 (appena impostato) alla comport virtuale installata sul PC. Ricordatevi di impostare il campo Connection Mode a Immediately (Fig. 14) e di impostare la velocità della seriale (Fig. 15), altrimenti il collegamento tra PC e cartellino non potrà essere stabilito. ➤

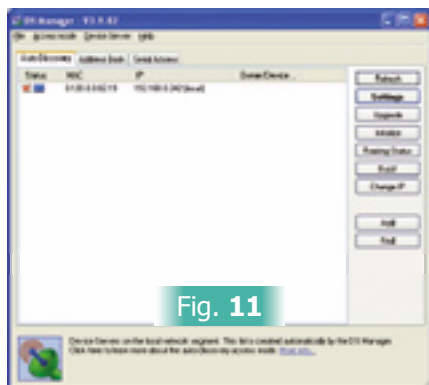


Fig. 11

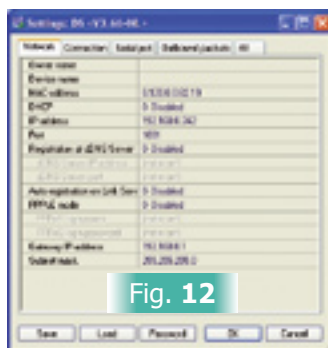


Fig. 12

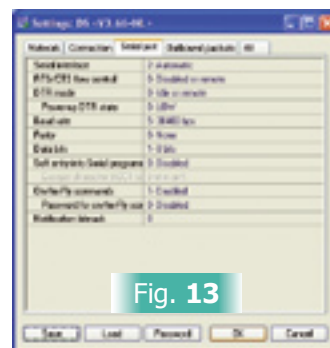


Fig. 13

Fig. 14



Fig. 15



Fig. 16

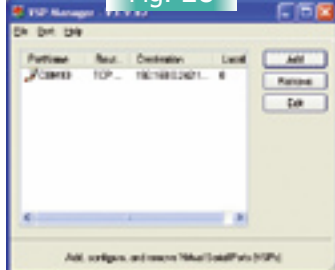


Fig. 17



Nel caso del nostro EM202, la configurazione da effettuare (eccetto il valore della COM, a vostra discrezione) è visibile nella schermata di Fig. 13, che mostra come configurare la comport e il controllo di flusso (di-

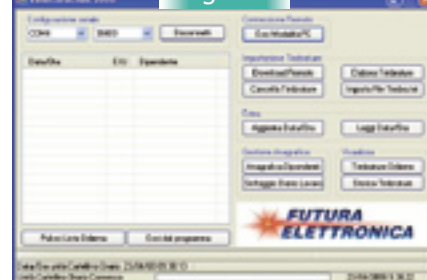
sattivato). Fate clic su OK per confermare la programmazione e tornare alla schermata principale (Fig. 16) che ora mostra il vostro nuovo server installato, associato ad una com virtuale e pronto all'uso. La fase di installazione è finita e pertanto potete chiudere il programma.

Giunti a questo punto, quale che sia il tipo di collegamento da voi scelto (Ethernet o radio), avete a disposizione una com virtuale che potete utilizzare per entrare in comunicazione diretta con il cartellino orario, usando il set di comandi già visti in precedenza.

Anche se la maggior parte di voi effettuerà la telegestione tramite il nostro programma specifico (cui è dedicato il prossimo paragrafo), è probabile che qualcuno vorrà realizzare un proprio applicativo: a titolo di esempio abbiamo riportato in Fig. 17 una schermata ottenibile tramite un qualsiasi programma di emulazione di terminale, tipo il nostro SerialControl TEST, già utilizzato nell'articolo relativo al radiomodem. L'EM202 del cartellino orario è stato associato alla com virtuale N. 1 ed ha ricevuto una serie di comandi inviati manualmente: dapprima è stata avviata la modalità telegestione (*/) quindi è stata attivata la gestione dell'orologio (*O), terminata in automatico dopo due secondi (!) perché in cartellino non ha ricevuto alcun comando di lettura o scrittura. È stata riattivata (*O) e quindi si è inviato un comando di lettura dell'orologio (*R) seguito da un aggiornamento di data e ora (*W*080423121500) e da una nuova lettura (*R). Dopo aver verificato la validità della nuova data, si è usciti dalla fase di telegestione (*X)

Quanto visto in Fig. 17 può essere ottenuto facilmente anche con Hyperterminal o ProcommPlus (e altri ancora) ma, di fatto, ciò avrebbe scopo puramente didattico: come già detto, infatti, il set di comandi viene utilizzato dal programma di telegestione specifico che, oltre a comunicare con

Fig. 18



il cartellino e scaricare i dati, può anche eseguire una serie di operazioni sui dati ricevuti come il calcolo delle ore complessive, l'estrapolazione degli straordinari, eccetera.

Il programma per la telegestione

Prima di iniziare a descrivere il programma di telegestione, chiamato TimeCardCheck, è necessario fare una premessa: non potendo prendere in considerazione tutte le possibili casistiche d'utilizzo, nella descrizione delle varie finestre, noi abbiamo sempre ipotizzato che sul cartellino orario fosse installato il modulo VDIP1, insieme alla pen drive dove vengono memorizzate le timbrature, situazione che, secondo noi, sarà la più comune. Il programma quindi, accede direttamente al file timbrature attraverso il set di comandi (visto in precedenza) solo in caso di necessità. Il cartellino orario, però, può operare anche in modalità assistita da remoto, ossia con salvataggio dei dati di timbratura su pen drive ed invio immediato dell'evento al PC dell'operatore: in questo caso non è necessario che il VDIP1 e la pen drive siano installati. Ciò non rappresenta alcuna situazione d'errore: il cartellino può continuare a funzionare correttamente a patto che venga utilizzato nelle modalità 1 o 3 che, a prescindere dal salvataggio dei dati su pen drive, garantiscono l'invio dell'evento di accesso alla postazione remota.

Bene, in questo caso, l'evento di timbratura inviato dal cartellino viene intercettato da un secondo programma, chiamato VirtualPenDrive, che

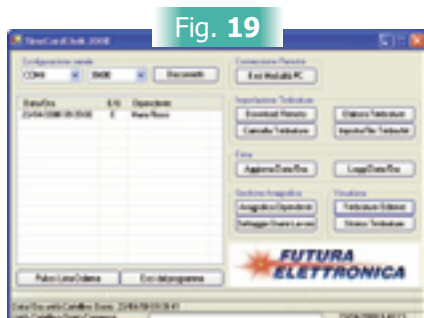


Fig. 19

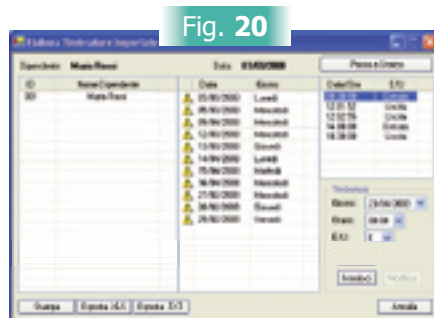


Fig. 20

memorizza tutti gli eventi in un file chiamato TIMBRABG.TXT (BG significa background), contenuto direttamente nel PC ed uguale per formato a quello che verrebbe generato nella pen drive. I nomi differenti servono in quanto questo file è residente in locale ed è gestito dal programma in background. Per comprendere meglio come funziona il VirtualPenDrive (cui abbiamo dedicato un paragrafo specifico), è però necessario conoscere il programma TimeCardCheck.

Il file eseguibile per l'installazione completa include sia il TimeCardCheck che il VirtualPenDrive e può essere scaricato gratuitamente, come sempre, dal sito della rivista (www.elettronica.in.it). Installate entrambi sulla postazione di telecontrollo seguendo la normale procedura di installazione di Windows quindi dimenticatevi temporaneamente del VirtualPenDrive; avviate, invece, il TimeCardCheck attraverso il menu di Start. Appare immediatamente la schermata principale visibile in Fig. 18, schermata che ora analizziamo in ogni particolare.

In alto a sinistra troviamo il riquadro relativo all'impostazione della porta seriale: nel caso della figura, il cartellino è collegato alla COM4, aperta a 38400 baud. Immediatamente al di sotto di questa sezione si trova l'elenco delle timbrature relative al giorno corrente (pulsante Timbrature odierne): esso mostra i tre dati fondamentali relativi alla timbratura ossia data, ora, entrata-uscita e identificativo del dipendente. Sotto alla lista sono presenti due tasti che permettono di pu-

lire l'elenco delle timbrature visualizzate (non agisce sul file TIMBRA.TXT) e di terminare l'esecuzione del programma.

Nella parte destra della schermata, invece, troviamo tutti i tasti associati ai vari comandi seriali visti in precedenza.

La prima sezione in alto abilita e disabilita, tramite un unico tasto, la connessione remota con il cartellino orario: essa, in pratica invia i comandi "*" e "*/X". Troviamo poi la sezione relativa alla gestione delle timbrature: il tasto Download Timbrature consente di effettuare la copia in locale (rinominata DOWNLOAD.TXT) del file TIMBRA.TXT, file che viene eliminato (direttamente sull'apparato) tramite il tasto Cancella Timbrature. I tasti Controlla Timbrature e Importa Timbrature permettono rispettivamente di analizzare (ossia calcolare le ore ordinarie, straordinarie, la mancanza di una o più timbrature, eccetera) il file e di importarlo direttamente dalla pen drive o da una directory.

La sezione Extra prevede la lettura e l'aggiornamento di data e ora di sistema, quest'ultima visualizzata (durante una connessione attiva) nella seconda linea alla base della schermata. La sezione Anagrafica permette di gestire ed aggiornare tanto i dati personali di ogni singolo utente registrato quanto gli orari di lavoro assegnati ad ogni giorno della settimana. La sezione Visualizza, infine, dispone di due tasti che permettono sia di scaricare le timbrature della data

odierna, sia di visualizzare lo storico delle timbrature; come ultima nota, ogni volta che il PC sta scambiando dati con il cartellino orario, una barra di stato alla base della schermata ne indica l'attività.

Utilizzo del programma

Per prima cosa, è necessario aprire la porta di comunicazione verso il dispositivo: per fare ciò, si seleziona nella sezione in alto a sinistra la comport voluta (essa corrisponde alla comport virtuale assegnata al bridge Ethernet a bordo del cartellino o al radiomodem connesso al PC), quindi si imposta la velocità di comunicazione (la predefinita è 38.400 baud) e si clicca sul pulsante Connetti, che cambia nome (Disconnetti) e avvia l'apertura della porta. La fase successiva consiste nell'attivare la comunicazione tra cartellino e programma, cliccando sul pulsante Entra Modalità PC (anch'esso cambia nome e diventa Esci Modalità PC); il programma entra in comunicazione con il cartellino e ciò viene mostrato nelle due linee di testo alla base della schermata. A questo punto, la prima operazione che, per logica, potrebbe essere eseguita è lo scarico delle timbrature odierne, avviata cliccando il pulsante Timbrature Odierne. Lo scambio dati ha ini-

zio e, dopo qualche istante, nell'elenco a sinistra vengono visualizzati tutti gli eventi di entrata ed uscita: la

Fig. 19 mostra una schermata



Listato 1

Riportiamo in questa sezione il codice sorgente relativo all'identificazione del comando ricevuto dal PC e all'esecuzione delle procedure di download. Il loop principale è rappresentato dalla routine GESTPC, che controlla costantemente i caratteri ricevuti dalla seriale di controllo remoto. Le routine DOWNLOAD, FINEFILE, DOWNLOADT, EXITSEARCH e EXITDOWNLOAD vengono chiamate automaticamente in funzione dello stato del download del file o della sezione di file.

```

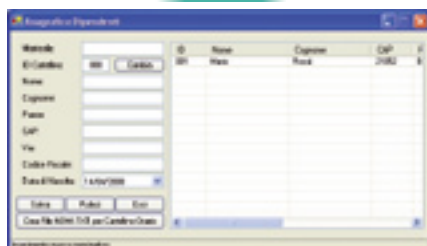
*****
GESTPC:      ' Routine gestione da PC.
IF P1=0 AND P2=0 THEN ' Se entrambi i pulsanti a 0:
  COMPUTER=0 ' il PC non è attivo
  GOTO EXITPC ' quindi esce dalla routine.
ENDIF ' fine routine.
SERIN2 RXRF,6,1000,GESTPC,[WAIT("*"),CMD]
' Attende il prompt seguito dal comando (CMD).
SELECT CASE CMD ' Controlla il comando.
CASE "O" ' CMD = O: Conf. orologio.
  SEROUT2 TXRF,6,["OROLOGIO",10,13]
  ' Output OROLOGIO come conferma ricezione comando.
  SERIN2 RXRF,6,2000,EXITPC,[WAIT("*"),tmp]
  ' Attende * che identifica nuovo comando.
  IF TMP="W" THEN ' W: scrittura orologio.
    SERIN2 RXRF,6,2000,EXITPC,[WAIT("*"),NOME[0],
      NOME[1],NOME[3],NOME[4],NOME[6],NOME[7],
      NOME[9],NOME[10],NOME[12],NOME[13],NOME[15],
      NOME[16]] ' Attende i vari campi data.
    ' Configura modulo RTC
    SEROUT TXCLOCK,2,["*W"],PAUSE 500:SEROUT
      TXCLOCK,2,[NOME[15],NOME[16]]
    PAUSE 500
    SEROUT TXCLOCK,2,["*W1"],PAUSE 500:SEROUT
      TXCLOCK,2,[NOME[12],NOME[13]]
    PAUSE 500
    SEROUT TXCLOCK,2,["*W2"],PAUSE 500:SEROUT
      TXCLOCK,2,[NOME[9],NOME[10]]
    PAUSE 500
    SEROUT TXCLOCK,2,["*W4"],PAUSE 500:SEROUT
      TXCLOCK,2,[NOME[6],NOME[7]]
    PAUSE 500
    SEROUT TXCLOCK,2,["*W5"],PAUSE 500:SEROUT
      TXCLOCK,2,[NOME[3],NOME[4]]
    PAUSE 500
    SEROUT TXCLOCK,2,["*W6"],PAUSE 500:SEROUT
      TXCLOCK,2,[NOME[0],NOME[1]]
    PAUSE 500
  ENDIF
  GOSUB READORA ' Non ha ricevuto W, quindi va
    ' a leggere l'ora e la invia.
  SEROUT2 TXRF,6,[DEC2 ANN,"/",DEC2 MES,"/",DEC2
    GIO,32,DEC2 HOR,":",DEC2 MI,":",DEC2 SEC,10,13]
  PAUSE 100 ' Pausa.
  SEROUT2 TXRF,6,["*"],10,13 ' Terminato invio ora.
CASE "D" ' CMD = D: download.
  SEROUT2 TXRF,6,["DOWNLOAD",10,13]
  ' Download avviato.
  SEARCH=0 ' Azzerare variabile di servizio.

GOSUB DOWNLOAD ' Chiama routine download.
CASE "G" ' CMD = G: ricerca giorno.
  SEROUT2 TXRF,6,["RICERCA GIORNO",10,13]
  ' Manda messaggio di conferma e attende data.
  SERIN2 RXRF,6,2000,EXITPC,[WAIT("*"),NOME[0],
    NOME[1],NOME[3],NOME[4],NOME[6],NOME[7]]
  ' Attende comando
  NOME[2]=47 ' Set variabili di servizio.
  NOME[5]=47 ' Set variabile di servizio.
  SEROUT2 TXRF,6,["RICERCA ",STR NOME\8,10,13]
  ' Messaggio conferma ricerca in corso.
  SEARCH=1 ' Set variabile di servizio:
    ' modo di ricerca.
  GOSUB DOWNLOAD ' Chiama routine download.
CASE "M" ' CMD = M: ricerca mese.
  SEROUT2 TXRF,6,["RICERCA MESE",10,13]
  ' Manda messaggio di conferma e attende mese.
  SERIN2 RXRF,6,2000,EXITPC,[WAIT("*"),NOME[0],
    NOME[1],NOME[3],NOME[4]] ' Attende una data.
  NOME[2]=47 ' Set variabile di servizio.
  SEROUT2 TXRF,6,["RICERCA ",STR NOME\5,10,13]
  ' Messaggio conferma ricerca in corso.
  SEARCH=3 ' Imposta modo di ricerca.
  GOSUB DOWNLOAD ' Chiama routine download.
CASE "U" ' CMD = U: ricerca utente.
  SEROUT2 TXRF,6,["RICERCA UTENTE",10,13]
  ' Manda messaggio di conferma e attende utente.
  SERIN2 RXRF,6,2000,EXITPC,[WAIT("*"),NOME[0],
    NOME[1],NOME[2]]
  SEROUT2 TXRF,6,["RICERCA ",STR NOME\3,10,13]
  ' Manda messaggio ricerca in corso.
  SEARCH=2 ' Imposta modo di ricerca.
  GOSUB DOWNLOAD ' Chiama routine download.
CASE "T" ' CMD = T: timbratura.
  SEROUT2 TXRF,6,["DOWNLOAD TIMBRATURA",10,13]
  ' Manda messaggio di conferma.
  SERIN2 RXRF,6,2000,EXITPC,[WAIT("*"),
    HEX8 CONTA2LONG]
  ' Attende parametro a 8 cifre (nome persona).
  SEROUT2 TXRF,6,["RICERCA ",HEX8 CONTA2LONG,10,13]
  ' Conferma ricerca in corso.
  SEARCH=10 ' Imposta modo di ricerca.
  CONTA2LONG=CONTA2LONG*25
  ' Set variabile di servizio.
  GOSUB DOWNLOADT ' Chiama routine download.
CASE "C" ' CMD = C: cancella TIMBRA.TXT
  SEROUT2 TXRF,6,["CANCELLA TIMBRA.TXT",10,13]
  ' Messaggio di conferma.
  SEROUT2 TXUSB,84, ["DLF TIMBRA.TXT",13]

```

ta d'esempio in cui l'unica timbratura presente è quella dell'utente Mario Rossi, che ha timbrato alle 9.39 del giorno 23/04/2008. Dopo aver scaricato le timbrature odierne, potrebbe essere necessario scaricare ed elaborare tutto il file timbrature, operazione questa, che viene eseguita in due fasi: si fa clic sul pulsante Download Timbrature e, a download terminato,

Fig. 21



si fa clic su Controlla Timbrature. A questo punto appare la schermata visibile in Fig. 20, che consente di verificare ed inserire eventuali modifiche sulle timbrature di ogni singolo utente. L'utente in corso di verifica e analisi viene sempre indicato nella parte superiore della schermata. Partendo da sinistra, troviamo un primo elenco che mostra sia l'identificativo numerico sia il nome per esteso di ogni dipendente: all'apertura del programma, l'utente predefinito è sempre il primo della lista ma è possibile selezionare qualsiasi utente semplicemente facendo clic sulla linea che lo riguarda.

Nella parte centrale, troviamo un secondo elenco che riepiloga tutti gli

eventi di timbratura relativi all'utente selezionato: i campi visualizzati sono tre e precisamente una segnalazione d'errore, la data e il giorno della settimana. La segnalazione d'errore (triangolo giallo con punto esclamativo) avviene ogni volta che il programma, durante l'elaborazione delle timbrature, trova un'incongruenza nel conteggio: potrebbe ad esempio capitare di timbrare due volte in entrata invece che una in entrata e una in uscita, oppure ci si potrebbe dimenticare di timbrare, eccetera. Facendo clic sulla linea che evidenzia l'errore, l'elenco a destra viene aggiornato con gli eventi di entrata ed uscita della linea: l'esempio della figura mostra che l'utente Mario

```

' Invia a VDIP1 comando cancellazione file.
PAUSE 1000 ' Pausa.
SEROUT2 TXRF,6,["*",10,13] ' Invia a PC un *.
CASE "S" ' CMD =S: dimensione TIMBRA.TXT.
SEROUT2 TXRF,6,["DIMENSIONE TIMBRA.TXT",10,13]
' Messaggio di conferma.
SEROUT2 TXUSB,84, ["DIR TIMBRA.TXT",13]
' Invia a VDIP1 comando richiesta dimensioni file.
SERIN2 RXUSB,84,5000,EXITPC,[WAIT("$"),
HEX2 CONTAB0,TMP,TMP,HEX2 CONTAB1,TMP,TMP,
HEX2 CONTAB2,TMP,TMP,HEX2 CONTAB3,TMP,TMP]
' Attende risposta da VDIP1.
SEROUT2 TXRF,6, ["SIZE ",#CONTALONG,13,10]
' Invia a PC le dimensioni del file.
SEROUT2 TXRF,6,["*",10,13] ' seguito da *.
CASE "X" ' CMD = X: fine modalità PC.
SEROUT2 TXRF,6,["EXIT",10,13]
' Messaggio di conferma.
COMPUTER=0 ' Computer non presente.
GOTO EXITPC ' Uscita.
SEROUT2 TXRF,6,["*",10,13]
' Comando non eseguito.
CASE ELSE ' CMD = non valido.
SEROUT2 TXRF,6,["COMANDO NON VALIDO",10,13]
' Manda messaggio d'errore.
SEROUT2 TXRF,6,["*",10,13] ' seguito da *.
END SELECT ' Fine scansione comandi.
GOTO GESTPC ' Ripete ciclo.
EXITPC: ' Uscita dalla subroutine.
SEROUT2 TXRF,6,["!",10,13] ' Timeout set orario.
RETURN ' Ripete scansione comandi.
' *****

DOWNLOAD: ' Routine di download.
FINE=25 ' Imposta variabile di servizio.
SEROUT2 TXUSB,84, ["OPR TIMBRA.TXT",13]
' Apertura file VDIP1 in lettura.
PAUSE 3000 ' Pausa.
SEROUT2 TXUSB,84, ["DIR TIMBRA.TXT",13]
' Ricerca file TIMBRA.TXT e attende risposta.
SERIN2 RXUSB,84,5000,EXITDOWNLOAD,[WAIT("$"),
HEX2 CONTAB0,TMP,TMP,HEX2 CONTAB1,TMP,TMP,
HEX2 CONTAB2,TMP,TMP,HEX2 CONTAB3,TMP,TMP]
PAUSE 1000 ' Pausa.
FOR CONTA2LONG=0 TO (CONTALONG-25) STEP 25
' Legge i vari record da 25 byte ognuno.
DOWNLOADT: ' Download timbrature.
SEROUT2 TXUSB,84, ["SEK ",#CONTA2LONG,13]
' Si posiziona alla riga giusta nel file.
PAUSE 10 ' Pausa.
SEROUT2 TXUSB,84, ["RDF 25",13]
' Legge 25 byte dalla riga dove è posizionato.
SERIN2 RXUSB,84,3000,FINEFILE,[STR d23]
' Attende fine file.
' *****

FINEFILE:
IF SEARCH=1 THEN ' Ricerca solo per data.
IF (CONTA2LONG//1250)=0 THEN ' Invia OK di segnale
SEROUT2 TXRF,6, ["OK",13,10] ' ricerca in corso
ENDIF ' durante ricerche lunghe.
TMP1=1 ' Imposta variabile di servizio.
FOR TMP=0 TO 7
IF D[TMP]<>NOME[TMP] THEN ' Verifica se data letta
TMP1=0 ' è quella d cercare.
GOTO EXITSEARCH ' Se non è quella, esce.
ENDIF
NEXT TMP
ENDIF
IF SEARCH=2 THEN ' Ricerca solo per un utente.
IF (CONTA2LONG//1250)=0 THEN ' Invia OK di segnale
SEROUT2 TXRF,6, ["OK",13,10] ' ricerca in corso
ENDIF ' durante ricerche lunghe.
TMP1=1 ' Imposta variabile di servizio.
FOR TMP=0 TO 2
IF D[TMP+18]<>NOME[TMP] THEN ' Verifica se utente
TMP1=0 ' letto è quello da cercare.
GOTO EXITSEARCH ' Se non è quello, esce.
ENDIF
NEXT TMP
ENDIF
IF SEARCH=3 THEN ' Ricerca solo per mese.
IF (CONTA2LONG//1250)=0 THEN ' Invia OK di segnale
SEROUT2 TXRF,6, ["OK",13,10] ' ricerca in corso
ENDIF ' durante ricerche lunghe.
TMP1=1 ' Imposta variabile di servizio.
FOR TMP=0 TO 4
IF D[TMP]<>NOME[TMP] THEN ' Verifica se mese letto
TMP1=0 ' è quello da cercare.
GOTO EXITSEARCH ' Se non è quello, esce.
ENDIF
NEXT TMP
ENDIF
FINE=23 ' Imposta Lunghezza stringa
GOSUB CALCOLACHECK ' su cui calcolare il checksum.
SEROUT2 TXRF,6, [HEX8 (CONTA2LONG/25),32,STR D,"*",
HEX2 CKC,13,10]
' Invio di tutta la stringa, checksum incluso.
IF SEARCH=10 THEN EXITDOWNLOAD ' Se download di una
' sola timbratura, esci.
' *****
EXITSEARCH: ' Altrimenti continua ricerca.
NEXT CONTA2LONG ' Incrementa variabile servizio.
' *****
EXITDOWNLOAD: ' Fine loop download.
SEROUT2 TXUSB,84, ["CLF TIMBRA.TXT",13]
' Chiudi file su pen drive.
PAUSE 100
SEROUT2 TXRF,6,["*",10,13] ' Invia carattere segnale
' fine download.
RETURN ' Fine procedura di download.

```

Rossi, il giorno 03/03/2008, ha timbrato due volte in uscita e pertanto, il programma non è in grado di effettuare il conteggio delle ore. I comandi controllati contenuti nella sezione Timbratura permettono di modificare (dopo averla selezionata facendo clic su di essa) e salvare la linea che presenta l'incongruenza. I tre pulsanti in basso, infine, permettono di stampare il resoconto delle timbrature di ogni utente, oppure di esportarle in formato Excel o testo. Per poter gestire le timbrature dei dipendenti, è necessario che esista un'anagrafica dei dipendenti: facendo clic sul pulsante Anagrafica Dipendenti presente nella schermata principale, appare la finestra di Fig. 21. Essa è molto intuitiva:

a sinistra troviamo i campi specifici (e modificabili) di ogni dipendente (che possiamo definire anche record) quali nome e cognome, data di nascita, indirizzo di residenza eccetera. Nella parte destra, invece, troviamo l'elenco di tutti i dipendenti presenti in anagrafica. Quando si crea un nuovo record, esso viene memorizzato facendo clic su Salva. Pulisci permette di cancellare tutti i dati specifici di un record ed Esci chiude la finestra. Per modificare un qualsiasi record, è sufficiente selezionarlo dall'elenco a destra quindi cliccare con il tasto destro e scegliere l'opzione Modifica: i campi editabili vengono aggiornati immediatamente con i dati specifici per il dipendente selezionato; dopo

aver effettuato le varie correzioni, si conferma il cambiamento facendo clic su Salva. Il tasto "Crea File NOMI.TXT per cartellino orario" permette di creare automaticamente il file nomi da trasferire sulla pen drive USB connessa al cartellino. Dopo aver gestito le varie timbrature, l'anagrafica, le potenziali incongruenze sul calcolo degli orari ➤

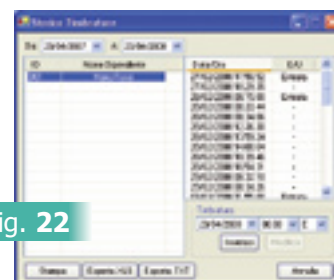
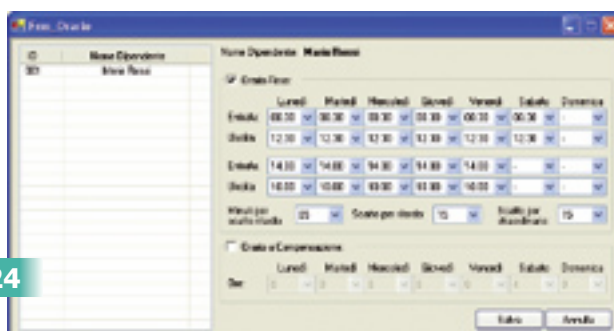
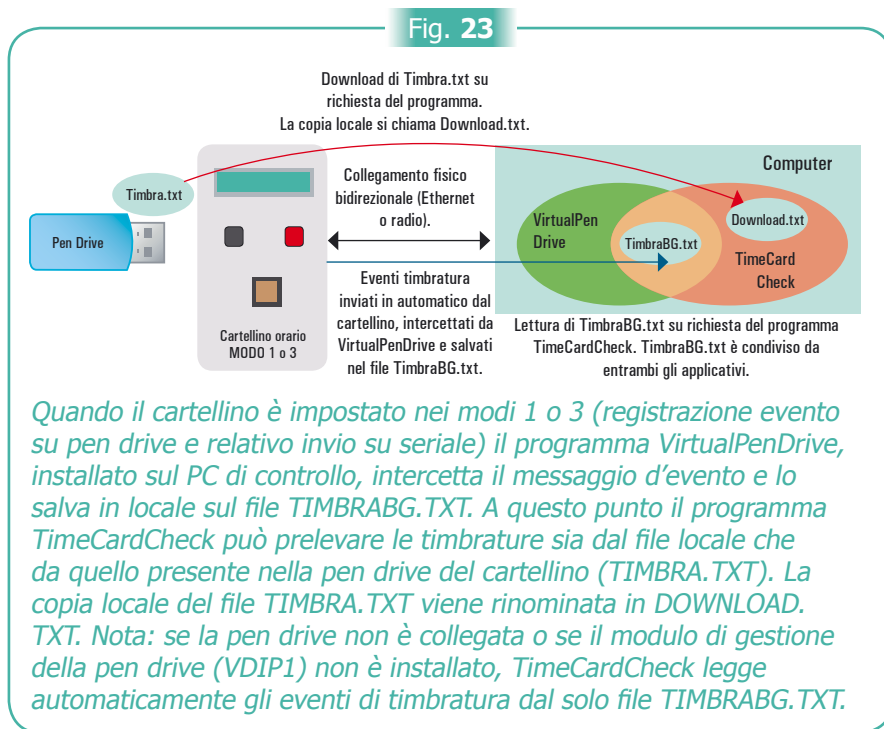


Fig. 22

di lavoro, è necessario archiviare le modifiche tramite il pulsante Passa a Storico in alto a destra (fig. 20). Lo storico delle timbrature viene visualizzato nella finestra di Fig. 22, che appare dopo aver fatto clic sul pulsante Storico Timbrature dalla finestra principale: da qui è possibile estrarre l'archivio completo delle timbrature specifiche per dipendente, comprese tra due date a scelta. Come già visto in precedenza per la gestione delle timbrature, anche in questo caso è possibile apportare delle variazioni relativamente ai vari eventi di entrata o uscita, nonché di stampare o esportare (in formato testo o Excel) la consultazione dello storico in corso. L'elenco a sinistra visualizza tutti i dipendenti presenti in anagrafica mentre l'elenco di destra mostra lo storico timbrature relativo al dipendente selezionato nell'elenco a sinistra.

Non rimane che definire gli orari di lavoro, necessari per poter gestire le timbrature: facendo clic sul pulsante Settaggio orario di lavoro (nella sezione anagrafica della finestra principale) viene aperta la finestra di Fig. 24. L'orario di lavoro è specifico per ogni dipendente e può essere scelto come "a Compensazione" (il conteggio a fine giornata deve sempre essere uguale o maggiore di otto ore lavorative) o "Fisso" (le timbrature per le varie entrate ed uscite devono essere effettuate nei tempi impostati). Le due modalità vengono selezionate facendo clic sulle Check Box Orario



Fisso / Orario a compensazione, che sono mutuamente esclusive (l'abilitazione di una disabilita l'altra). La sezione di sinistra mostra l'elenco dei dipendenti in anagrafica mentre la sezione di destra permette di impostare e verificare l'orario di lavoro applicato al dipendente selezionato dall'elenco.

Il programma VirtualPenDrive

Riprendiamo ora il programma che avevamo "dimenticato" temporaneamente. Da quanto già spiegato in precedenza, sappiamo che il cartellino orario può funzionare anche senza la pen drive o il modulo VDIP1: in questo caso, però, deve essere im-

per il

MATERIALE

Tutti i componenti utilizzati in questo progetto sono facilmente reperibili in commercio. Il master del circuito stampato, il firmware col quale programmare il microcontrollore ed il software descritto in queste pagine possono essere scaricati gratuitamente dal sito della rivista (www.elettronica.in.it). Il microcontrollore che sovrintende a tutte le funzioni del dispositivo è disponibile anche già programmato (cod. MF736) al prezzo di 25,00 Euro, IVA inclusa.

**Il materiale va richiesto a: Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)
Tel: 0331-799775 ~ Fax: 0331-792287 ~ <http://www.futurashop.it>**

postato in modalità 1 o 3, modalità che prevedono entrambe l'invio di un pacchetto di dati, contenente l'evento di timbratura, attraverso la porta di comunicazione con il PC remoto (via radiomodem o via EM202). Il pacchetto dati relativo all'evento viene intercettato dal programma VirtualPenDrive.exe che, come dice il nome stesso emula una pen drive virtuale, residente sul PC remoto, utilizzata per salvare gli eventi di timbratura sul file TIMBRABG.TXT. La Fig. 23 mostra il principio di funzionamento del programma VirtualPenDrive, che fa parte del pacchetto di installazione del TimeCardCheck. Una volta installato VirtualPenDrive, occorre lanciarne l'esecuzione, attraverso il menu d'avvio oppure riavviando il PC. Il programma, infatti, riparte automaticamente ogni volta che il PC viene riacceso. A programma attivo, nella parte in basso a destra del desktop appare un'icona che ricorda un hard disk (Fig. 25). Facendo clic con il tasto destro sull'icona appare un menu (Fig. 26) che permette di

- avviare o arrestare il servizio, ossia di aprire/chiudere il canale di

comunicazione con il cartellino orario;

- aprire il Pannello di controllo, cioè la schermata principale del programma TimeCardCheck;
- eliminare il file timbratura TIMBRABG.TXT;
- configurare la comport;
- terminare l'esecuzione del programma.

Selezionando l'opzione "Configura COM" viene caricata la finestra di Fig. 27 dove possono essere impostati tre parametri: la comport, la velocità e la connessione automatica al cartellino quando viene avviato il PC. A programma attivo, ogni evento di timbratura viene inserito nel file TIMBRABG.TXT che rispetta la sintassi impiegata nel TIMBRA.TXT presente nella pen drive e che, pertanto, si presenta in così

```
...
08/05/12 16:36:33 001 X
08/05/12 16:37:02 001 U
08/05/13 16:38:08 111 X
08/05/13 16:39:56 001 X
08/05/12 17:08:36 001 X
...
```

A questo punto è possibile avviare la schermata principale del programma TimeCardCheck (Pannello di Controllo, Fig. 26) quindi fate clic sul pulsante Timbrature odierne: il programma chiede all'utente se si vuole procedere con il download delle timbrature odierne direttamente dal file TIMBRA.TXT sulla pen drive a bordo del cartellino (nel caso sia presente) oppure dal file locale TIMBRABG.TXT (Fig. 28). Per scaricare le timbrature odierne dal TIMBRABG.TXT, occorre fare clic su NO: la sezione di sinistra del pannello di controllo, quindi, viene aggiornata con le timbrature odierne desunte dal file locale (Fig. 29). Il TimeCardCheck, però, consente anche di importare (ossia di prelevare un file noto da una cartella qualsiasi) il file timbrature facendo clic sul pulsante Importa File Timbrature. In questo caso, possono essere importati solamente tre file:



TIMBRA.TXT, prelevabile direttamente dalla pen drive (rimossa dal cartellino e collegata al PC); DOWNLOAD.TXT che, come abbiamo visto, è la copia in locale del TIMBRA.TXT, eseguita tramite la seriale; TIMBRABG.TXT, generato in locale dal VirtualPenDrive. È anche possibile annullare l'operazione di importazione. La Fig. 30 mostra la schermata per la scelta del file da importare. Da questo momento in poi, è possibile eseguire tutte le operazioni di verifica e controllo esattamente come già visto nel paragrafo precedente. Tenete presente un particolare: il VirtualPenDrive può eliminare solamente il file TIMBRABG.TXT in locale mentre il TimeCardCheck può eliminare il TIMBRA.TXT.

Fig. 25

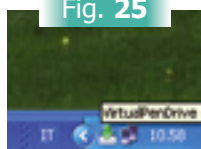


Fig. 26

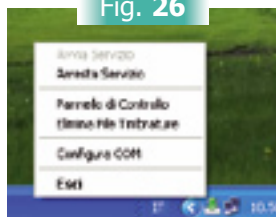


Fig. 27



Piattaforma di sviluppo PIC, firmware e software

dell'Ing. Vincenzo Ligorio

Analizziamo un semplice programma di test che legge lo stato dei pulsanti e ne visualizza su display l'identificativo numerico, insieme all'emulatore di terminale attraverso il quale è possibile eseguire il debug delle applicazioni firmware in fase di sviluppo. Ultima puntata.

Ora che avete a disposizione la piattaforma hardware, descritta sul fascicolo N. 126, e tutte le librerie con cui sviluppare le vostre applicazioni (le più importanti delle quali sono state descritte sul fascicolo N. 127), potete iniziare a cimentarvi con semplici programmi come quello che vi proponiamo in queste pagine, che rappresenta un valido esempio didattico. Si tratta di un programma molto semplice che esegue ciclicamente quattro operazioni ossia: controlla costantemente la UART per identificare eventuali comandi provenienti dall'emulatore di terminale; verifica e gestisce

il comando ricevuto. Controlla, infine, la tastiera ed aggiorna il display. Per quanto questo applicativo sia fine a sé stesso, le procedure in esso impiegate sono quelle più comunemente impiegate nelle applicazioni pratiche e pertanto, pur non costituendo alcuna applicazione reale vera e propria, il programma in esame (chiamato Demo-02.asm e scaricabile come sempre dal sito della rivista) offre molti spunti dai quali potrete partire per realizzare le vostre applicazioni firmware. Vediamo ora più in dettaglio sia il programma che l'emulatore di terminale.



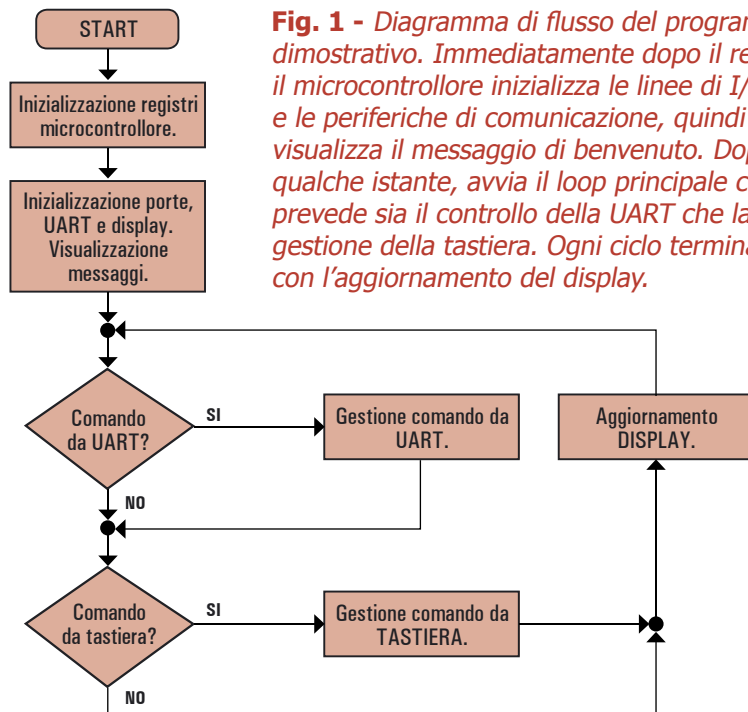


Fig. 1 - Diagramma di flusso del programma dimostrativo. Immediatamente dopo il reset, il microcontrollore inizializza le linee di I/O e le periferiche di comunicazione, quindi visualizza il messaggio di benvenuto. Dopo qualche istante, avvia il loop principale che prevede sia il controllo della UART che la gestione della tastiera. Ogni ciclo termina con l'aggiornamento del display.

Il programma DEMO-02.ASM

In Fig. 1 potete vedere il diagramma di flusso del loop principale, il quale fa uso di alcune routine preconfezionate e di altre preparate appositamente (esattamente come avverrà quando sarete voi a scrivere le vostre applicazioni).

Innanzitutto, il programma inizializza i registri speciali del microcontrollore, ossia il tipo di oscillatore, la programmazione della Flash in low voltage, il circuito di watch-dog, eccetera; tutto ciò viene eseguito con le istruzioni riportate nel Listato 1. Procede, quindi, con l'inizializzazione delle varie periferiche, secondo la sequenza riportata nel Listato 2. Notate un particolare: fatta eccezione per i dati da visualizzare a display, che sono stati definiti manualmente, il resto del listato si riduce a delle semplici chiamate (*call portset*, *call display*, eccetera) alle routine preconfezionate che abbiamo analizzato nella puntata precedente: ciò consente di rimanere concentrati sulla sequenza logica del programma, senza doversi preoccupare della scrittura di routine specializzate.

Ottemperata la fase di inizializzazione delle periferiche, viene avviato il main loop vero e proprio (visibile nel Listato 3) che, secondo la logica del diagramma di flusso di Fig. 1, controlla ed eventualmente processa i comandi ricevuti tramite UART dall'emulatore di terminale, fa la stessa cosa con la tastiera, quindi aggiorna il display.

Prestate attenzione a come anche in questa sezione di programma non si sia fatto altro che definire una sequenza logica di operazioni ripetute ciclicamente e legate, nella maggior parte dei casi, sempre alle solite routine preconfezionate.

È stato comunque necessario creare due nuove routine specifiche per questa applicazione: stiamo parlando della *keyboard* (Listato 4) e della *gestkeys* (Listato 5); la prima veri-

Listato 1 – Inizializzazione dei registri speciali

```

; Inclusioni degli Header File.
#include DEMO-02.inc
#include pl8f8722.inc
; Configurazione dei registri speciali.
__CONFIG _CONFIG1H, _OSC_HS_1H
__CONFIG _CONFIG2L, _BOREN_SBOARDS_2L & _BORV_43_2L & _PWRT_ON_2L
__CONFIG _CONFIG2H, _WDT_OFF_2H
__CONFIG _CONFIG3L, _MODE_MC_3L
__CONFIG _CONFIG4L, _LVP_OFF_4L & _DEBUG_OFF_4L & _XINST_OFF_4L
__CONFIG _CONFIG4L, 0x8B
__CONFIG _CONFIG5H, 0xC0
__CONFIG _CONFIG6H, 0xE0

org 0x0000      ; Vettore di reset:
goto start     ; punta a etichetta start.
org 0x0008      ; Vettore di interrupt 1:
nop            ; non usato.
retfie
org 0x0018      ; Vettore di interrupt 2:
nop            ; non usato.
retfie

```

Listato 2 – Inizializzazione delle periferiche

```

; Configurazione delle periferiche.
start          ; Il vettore di reset punta
nop            ; a start.
call portset   ; Inizializzazione porte.
call inizUART1 ; Inizializzazione UART1.
call inizdisplay ; Inizializzazione display.
call sec1      ; Attesa di 1 secondo
movlw 0xF0     ; Visualizza la scritta
movwf TBLPTRH  ; addr. 1F000.
movlw 0x00     ;
movwf TBLPTRL  ;
call copiadisplay ; Copia in RAM i dati estratti dal programma.
call display     ; Trasferisce da RAM a display.
call sec1        ;
call sec1        ;
movlw 0xF0       ; Visualizza la scritta
movwf TBLPTRH    ; addr. 1F050.
movlw 0x50       ;
movwf TBLPTRL    ;
call copiadisplay ; Copia in RAM i dati estratti dal programma.
call display     ; Trasferisce da RAM a display.

```

fica che un tasto sia premuto e, nel caso lo fosse, identifica il tasto. La seconda, invece, esegue le istruzioni associate a quel tasto.

In pratica, il programma vero e proprio è rappresentato da queste ultime due routine, cui è delegato il compito di gestire i pulsanti della tastiera in funzione delle varie necessità.

Il firmware, però, non è ancora ultimato: dobbiamo, infatti, definire ancora le scritte da trasferire a display e le operazioni da eseguire in funzione del comando ricevuto tramite seriale.

Il primo caso è abbastanza semplice: come visibile nel Listato 6, è sufficiente definire l'indirizzo di partenza della scritta (*org 0x1F0A0*), indirizzo che dipende dalle dimensioni della memoria del PIC in uso, e far seguire ad essa una serie di dichiarazioni *double-word* (dw) seguite da due caratteri del testo che si vogliono scrivere in memoria, contenuti tra apici.

Dal Listato 6 è facile comprendere come all'indirizzo 1F0A0 sia contenuto uno spazio, a 1F0A1 un altro spazio, a 1F0A2 una P, a 1F0A3 una r e così via fino ad esaurimento della memoria.

Nel secondo caso, invece (operazioni da eseguire in funzione del comando ricevuto), occorre lavorare un po' di più: la routine preconfezionata *incmdUART*, infatti, è in grado di riconoscere ogni singolo comando ricevuto tra quelli sia predefiniti che aggiunti da voi, ma può eseguire solo le procedure per le quali esiste del codice. Fate riferimento al listato 7: la *incmdUART*, dopo aver verificato che i caratteri ricevuti dalla seriale non siano affetti da errori o timeout, va a chiamare la subroutine *gestcmdUART*, il cui compito è quello di identificare il comando ed eseguire le istruzioni ad esso associate.

Ad esempio, la ricezione dei comandi A e a determina la chiamata alle subroutine predefinite *util1a* e *util1b*:

Listato 3 – Loop principale

```
*** Programma principale ***
main                               ; Inizio del main.
nop                                ;
call    indUART                    ; Controlla se UART ha ricevuto
movlw   0x00                       ; un comando.
cpfseq  RA                         ; Se sì, allora
call    incmdUART                  ; gestisce il comando
nop                                ;
call    keyboard                   ; Verifica se c'è un tasto premuto
movlw   0x00                       ;
cpfseq  KEY                        ; se sì (KEY <> 00), allora
call    gestkeys                   ; gestisce il tasto.
nop                                ;
call    display                    ; Aggiorna il display.
goto    main                       ; Ripete main loop.
*** Fine programma principale ***
```

Listato 4 – Controllo tasto premuto

```
*** Routine controllo tasti ***
; La routine restituisce nella variabile KEY
; 00 se nessun tasto premuto;
; 01-08 in funzione del tasto.
keyboard    ; Nome della routine.
clr         KEY                                ; Default KEY=00: nessun tasto premuto
testkeys1   ; Test su primo tasto.
btfsc      PORTB,0                            ; Tasto 1 premuto?
goto       testkeys2                          ; No. Va a verificare tasto 2.
movlw      0x01                                ; Sì. Scrive 01
movwf      KEY                                ; nella variabile KEY
goto       testkeysend                        ; ed esce dalla subroutine.
testkeys2   ; Test su secondo tasto.
btfsc      PORTB,1                            ; Tasto 2 premuto?
goto       testkeys3                          ; No. Va a verificare tasto 3.
movlw      0x02                                ; Sì. Scrive 02
movwf      KEY                                ; nella variabile KEY
goto       testkeysend                        ; ed esce dalla subroutine.
testkeys3   ; Test su terzo tasto.
...         ; Codice relativo al test
...         ; degli altri 5 tasti omesso
...         ; per questioni di spazio.
testkeysend ; Test ultimato.
return      ; Esce dalla subroutine.
*** Fine Routine controllo tasti ***
```

Listato 5 – Gestione tasto premuto

```
*** Routine gestione comandi tasti ***
gestkeys    ; Inizio Routine.
gestkeys1   ; Istruzioni relative al tasto 1.
movlw      0x01                                ;
cpfseq     KEY                                ; Se KEY=01 gestisce il tasto 1
goto       gestkeys2                          ; altrimenti verifica se deve gestire il tasto 2.
movlw      0xF0                                ; Seleziona la scritta contenuta in memoria flash
movwf      TBLPTRH                            ; all'indirizzo 1F0A0.
movlw      0xA0                                ;
movwf      TBLPTRL                            ;
call       copiadisplay                      ; Copia a display la scritta appena selezionata.
movff     KEY, RB                            ; COSA SIGNIFICA?
bsf       RB, 4                              ; per ASCII
bsf       RB, 5                              ; COSA SIGNIFICA?
movff     RB, 0x0CC ; COSA SIGNIFICA?
goto       gestkeysend                       ; Termine della procedura.
gestkeys2   ; Istruzioni relative al tasto 2.
movlw      0x02                                ;
cpfseq     KEY                                ; Se KEY=02 gestisce il tasto 2
goto       gestkeys3                          ; altrimenti verifica se deve gestire il tasto 3.
movlw      0xF1                                ; Seleziona la scritta contenuta in memoria flash
movwf      TBLPTRH                            ; all'indirizzo 1F100.
movlw      0x00                                ;
movwf      TBLPTRL                            ;
call       copiadisplay                      ; Copia a display la scritta appena selezionata.
movff     KEY, RB                            ; COSA SIGNIFICA?
bsf       RB, 4                              ; per ASCII
bsf       RB, 5                              ; COSA SIGNIFICA?
movff     RB, 0x0CC ; COSA SIGNIFICA?
goto       gestkeysend                       ; Termine della procedura.
gestkeys3   ; Istruzioni relative al tasto 3
...         ; e a tutti gli altri tasti.
...         ;
...         ;
gestkeysend ; fine routine
return      ;
*** Fine Routine gestione comandi tasti ***
```

la prima legge 16 byte dalla RAM e li manda su seriale; la seconda calcola l'indirizzo di una locazione di

memoria. Esistono molti comandi predefiniti, ma alla maggior parte di essi non è associata alcuna rou-

tine, come avviene per i comandi *E* e *e*, che richiamano le subroutine *miocomando_E* e *miocomando_e*, le quali devono essere scritte dallo sviluppatore del firmware. In ogni caso, inserire del codice per il riconoscimento di un comando è piuttosto semplice: basta aggiungere dopo la linea di confronto con esito negativo (che fa saltare alla verifica del comando successivo, *goto gestcmd_a*, *goto gestcmd_B*, *goto gestcmd_b*, *goto gestcmd_c...*) la chiamata alla propria subroutine (*call miocomando_e*, *call miocomando_E...*).

A questo punto, l'analisi del programma Demo-02.asm è completa. Potete aprirlo con l'ambiente di sviluppo MPLAB, compilarlo per creare il file eseguibile da scrivere sul microcontrollore tramite la porta ICSP, quindi accendere la scheda. Il programma parte immediatamente, e la scheda mostra a display due messaggi di benvenuto distanti qualche secondo l'uno dall'altro. Pigiare uno qualsiasi degli otto pulsanti e verificate che ogni volta venga visualizzato a display l'identificativo numerico del pulsante. Premendo i pulsanti uno e due (in basso a sinistra sulla scheda pulsanti), verranno visualizzati a display i messaggi "oggi c'è il sole" e "Mettere codice".

Collegate, ora, la seriale del PC su cui avete installato il programma di emulazione di terminale alla COM1 della demoboard (tenendo i connettori DB9 a sinistra, la COM1 è quella in basso). Siete pronti per iniziare ad interagire con la scheda.

Il programma Terminal 2008

Si tratta di un programma essenziale, privo di fronzoli e mirato alla praticità. L'unica schermata che appare quando viene mandato in esecuzione (Fig. 2) è divisa in tre sezioni: la parte in alto (insieme ai menu a tendina) permette di scegliere la comport collegata alla demoboard e

Listato 6 – Gestione tasto premuto

```

; 0000000001111111112 dw "t" ;
; 12345678901234567890 dw "as" ;
; xxxxxxxxxxxxxxxxxxxx dw "to" ;
org 0x1FOA0 ;1 Programma demo 2 dw " " ;
;2 Premere un tasto dw "og" ; Linea 3
;3 Oggi c'è il sole dw "gi" ;
;4 Tasto premuto: - dw "c" ;
dw " " ;
dw "e" ;
dw "Pr" ; Linea 1
dw " " ;
dw "og" ;
dw "ra" ;
dw "mm" ;
dw "a" ;
dw "de" ;
dw "mo" ;
dw "2" ;
dw " " ;
dw " " ; Linea 2
dw "Pr" ;
dw "em" ;
dw "er" ;
dw "e" ;
dw "un" ;
dw "Ta" ; Linea 4
dw "st" ;
dw "o" ;
dw "pr" ;
dw "em" ;
dw "ut" ;
dw "o" ;
dw " " ;
dw " " ;
dw " " ;

```

Listato 7 – Gestione comando ricevuto

```

;*** Routine di ricezione del comando dalla UART.
incmdUART ; Inizio routine.
call carechoUART ; Trasmette il "I" come conferma.
call rxUART ; Preleva la stringa comandi.
movlw 0x00 ; Controlla se CARRX vale 00.
cpfseq CARRX ; ossia errore o timeout di ricezione.
call gestcmdUART ; CARRX<>00: gestisce il comando.
return ; CARRX=00: fine routine.
;*** Fine Routine di ricezione del comando dalla UART.

;*** GESTIONE COMANDI UART ***
; Routine di gestione dei vari comandi UART.
; Questa routine si attiva quando e' ricevuto l'indirizzo della scheda.
; In funzione dei vari comandi attiva le routine idonee.
gestcmdUART ; Routine chiamata da incmdUART.
movff 0x100,RA ; Copia il codice del comando nel registro RA.
gestcmd_A ; Verifica comando "A".
movlw 0x41 ;
cpfseq RA ; Comando=41 (A)?
goto gestcmd_a ; No. Verifica comando successivo.
call utilla ; Si. Chiama Routine che invia 16 byte
; della memoria RAM.
goto gestcmd_fine ; Termina gestione comando.
gestcmd_a ; Verifica comando "a".
movlw 0x61 ;
cpfseq RA ; Comando=61 (a)?
goto gestcmd_B ; No. Verifica comando successivo.
call utillb ; Si. Chiama Routine calcolo indirizzo dai 3
; caratteri ASCII, trasforma il dato da ASCII
; a Hex e lo scrive in RAM.
goto gestcmd_fine ; Termina gestione comando.
... ; Codice di riconoscimento di altri comandi.
... ;
gestcmd_E ; Verifica comando "E".
movlw 0x45 ;
cpfseq RA ; Comando=45 (E)?
goto gestcmd_e ; No. Verifica comando successivo.
call miocomando_E ; Chiamata a routine proprietaria di gestione
; del comando.
goto gestcmd_fine ; Termina gestione comando.
gestcmd_e ; Verifica comando "e".
movlw 0x65 ;
cpfseq RA ; Comando=65 (e)?
goto gestcmd_F ; No. Verifica comando successivo.
call miocomando_e ; Chiamata a routine proprietaria di gestione
; del comando.
goto gestcmd_fine ; Termina gestione comando.
... ; Codice di riconoscimento di altri comandi.
... ;
gestcmd_fine ; Fine gestione comandi.
return ; Uscita dalla subroutine.

```

la velocità di comunicazione (la predefinita è 9600 baud).
Tutta la sezione sinistra della scher-

ta è presente il pulsante Clear, che permette di cancellare il contenuto delle caselle RX e TX.

Fig. 2 - La schermata principale dell'emulatore di terminale è molto intuitiva: in alto troviamo i menu per l'impostazione della comport, a sinistra le sezioni relative ai dati ricevuti o trasmessi. A destra, infine, i pulsanti e i campi comando, tramite i quali si può effettuare il debug.

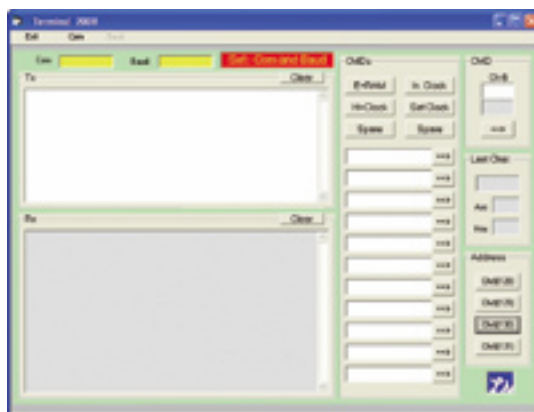
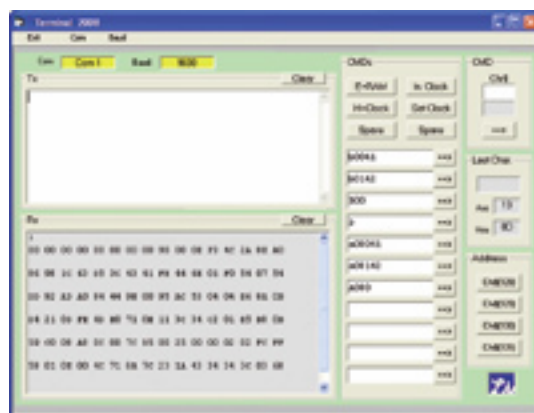


Fig. 3 - Debug: sono stati inseriti dei comandi predefiniti (ad esempio, b0041 scrive una A nella locazione EEPROM 00) nelle caselle di testo. Sono quindi stati premuti il pulsante di Address 128 (Chr\$(128)) per allertare la scheda e il pulsante associato al comando D (dump memoria EEPROM).



mata è dedicata occupata da due caselle di testo in cui vengono visualizzati sia comandi inviati alla scheda (TX) che quelli ricevuti da essa (RX). La casella TX è editabile quindi, dopo aver preso confidenza con i comandi, potrete inserire direttamente da tastiera i comandi da inviare alla scheda. Ogni comando inviato genera una risposta da parte della scheda, risposta che appare nella casella di sola lettura sottostante (RX, in grigio); se il comando è valido, la demo board risponde con il prompt "punto esclamativo", a conferma della validità del comando e dell'esecuzione delle istruzioni. La mancata risposta è sintomo di malfunzionamento (un "baco" nel software) o di comando errato. Nella parte centrale della schermata

La sezione destra del programma è quella più popolata di controlli, necessari per poter interagire (e quindi effettuare il debug) con la scheda. La sezione CMDs prevede sei pulsanti predefiniti tramite i quali è possibile interagire con la periferica "orologio" (qualora essa sia utilizzata dall'applicativo, manonè il nostro

caso), insieme a undici caselle di testo (associate ognuna ad un pulsante di invio identificato da ==>) in cui potrete inserire i comandi di debug che utilizzerete più frequentemente. La parte tutta a destra, infine prevede tre sezioni e precisamente:

- **CMD**, che consente l'inserimento e l'invio di un comandi specifico,
- **Last Char**, che mostra l'ultimo carattere inviato dalla scheda,
- **Address**, tramite la quale si seleziona l'indirizzo della scheda da debuggare.

L'invio di un comando è semplice: si fa clic sul pulsante di indirizzo (nel nostro caso è il 128: la scheda risponde con "!") quindi si clicca il pulsante relativo al comando da inviare. Da quanto esposto si deduce che, per debuggare il programma che sta girando sulla demoboard, è sufficiente prevedere delle strutture di codici che possano interagire con i vari comandi.

I comandi attualmente supportati sono visibili in Tabella 1: tramite essi è già possibile effettuare il debug reale della applicazione, poiché, interagendo con la routine *incmdUART* linkata nell'applicativo Demo-02.asm, essi consentono di leggere e modificare qualsiasi locazione di memoria RAM e EEPROM nonché fare il DUMP (scarico) della sezione di RAM che contiene tutte le variabili di sistema. Come già visto nel paragrafo ➤

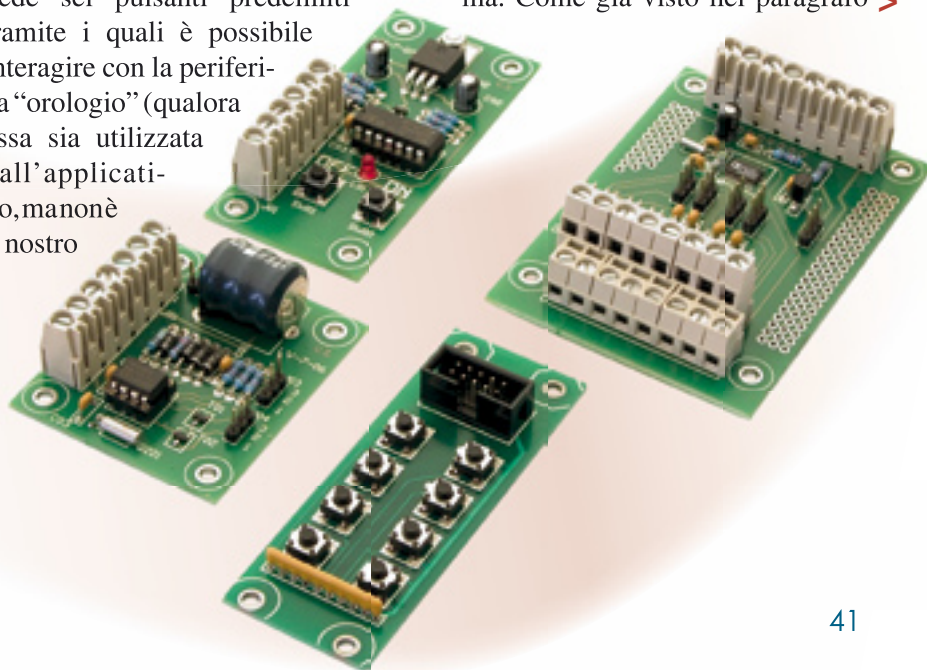


Tabella 1 - Elenco comandi supportati

Comando	Sintassi	Ricezione	Funzione
A	Axxx	8A BC ...24	Legge sedici byte dalla RAM, a partire dall'indirizzo xxx.
a	axxyy	-	Scrive il byte yy alla locazione RAM di indirizzo xxx.
B	Bxx	8A BC ...24	Legge sedici byte dalla EEPROM interna, a partire dall'indirizzo xx.
b	bxyy	-	Scrive il byte yy alla locazione EEPROM interna di indirizzo xx.
D	D	8A BC ...24	Preleva il contenuto della RAM dall'indirizzo 000 all'indirizzo 05F: in questa sezione di memoria sono contenute tutte le variabili di sistema usate dalle routine preconfezionate.

precedente, è possibile implementare altri comandi specifici, con cui modificare, per esempio, i messaggi a display, gli identificativi dei tasti, eccetera. Come già detto, l'in-

vio dei comandi è molto semplice:

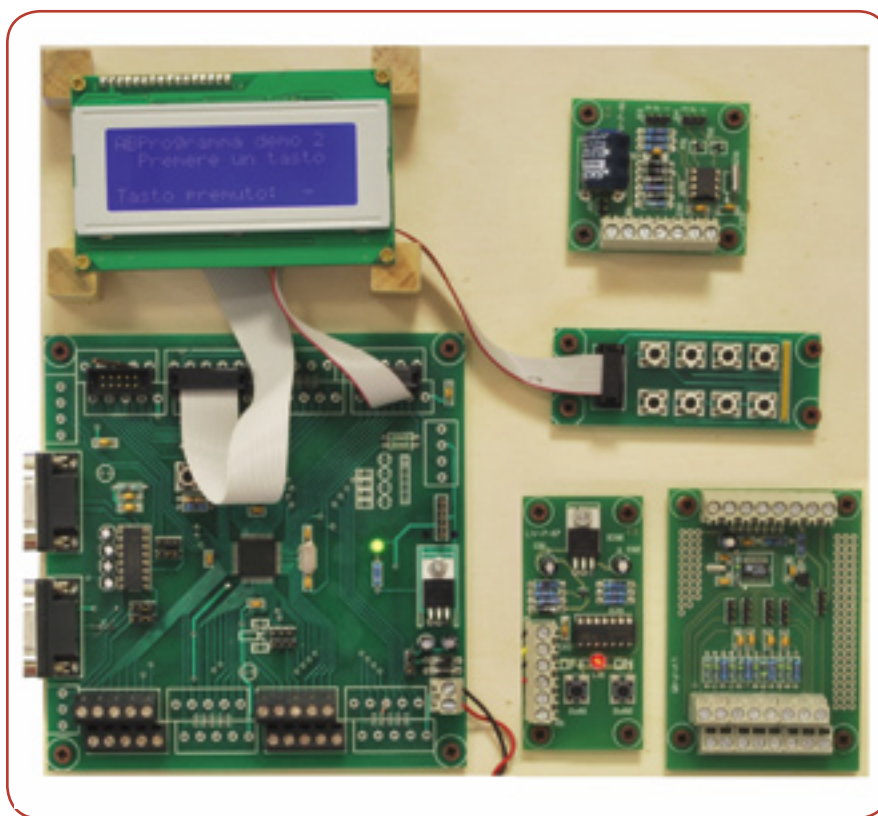
- si manda l'indirizzo del board di sviluppo (in questo caso 128);
- la board di sviluppo risponde con il prompt "!";

- si invia il comando desiderato;
- si visualizza, se prevista, la risposta al comando.

Semplici esempi pratici

Ipotizziamo che stiate sviluppando un programma che, facendo uso delle routine matematiche, legge e scrive dei dati sulla RAM e che, per qualche strana ragione, vi siano degli errori durante la scrittura in RAM. Bene, un'operazione certamente utile è rappresentata dal DUMP (lettura completa) delle locazioni di memoria utilizzate dalle variabili matematiche. Quest'operazione è visibile in Fig. 3: prima è stato cliccato il pulsante di indirizzo (Chr\$(128)) per alertare la scheda che ha confermato la ricezione del comando con l'invio del prompt "!". È stato, quindi, cliccato il pulsante associato alla casella di testo in cui abbiamo inserito il comando D (DUMP): la scheda ha risposto inviando sulla seriale il contenuto di tutte le variabili matematiche (pari a 96 byte).

Così come è stato letto il contenuto della RAM, è possibile modificarlo



per il

MATERIALE

Il materiale per la realizzazione della nostra piattaforma di sviluppo (il cui hardware è stato descritto sul fascicolo 126) è facilmente reperibile presso qualsiasi rivenditore di componenti elettronici. I master della scheda base e delle periferiche, il firmware DEMO-02.zip ed il software per Windows possono essere scaricati gratuitamente dal sito della rivista (www.elettronica.in.it). Il microcontrollore PIC18F8723-I/PT ed il real time clock PCF8563P possono essere acquistati rispettivamente al costo di 17,00 e 4,50 Euro inclusa IVA.

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

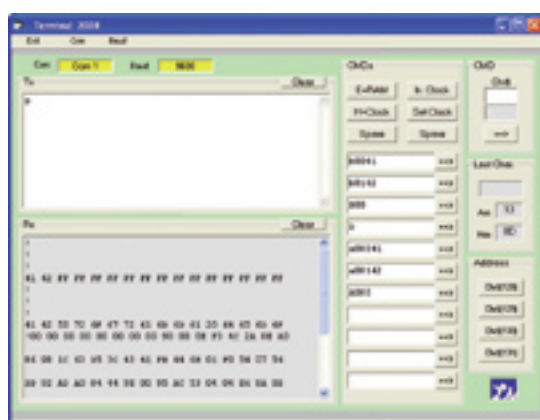
Tel: 0331-799775 ~ Fax: 0331-792287 ~ <http://www.futurashop.it>

(anche quello della EEPROM). Fate riferimento alla Fig. 4, in cui la casella RX mostra risposte differenti da quelle riportate in Fig. 3. Per prima cosa sono stati inviati i due comandi b0041 e b0142 (preceduti entrambi dal comando di indirizzo) con i quali sono stati scritti i caratteri A e B nelle locazioni EEPROM 00 e 01. La demoboard ha risposto ad entrambi i comandi con il prompt "!". Successivamente è stato inviato il comando B00, con il quale sono stati letti sedici caratteri dalla EEPROM, iniziando dalla locazione 00: notate che i primi due caratteri

contiene i messaggi visualizzati a display. Analogamente potremmo inviare A0090, A0A0, eccetera, per leggere, a pacchetti di 16 byte, tutto il contenuto della RAM display.

Come ultima operazione, abbiamo fatto nuovamente il DUMP delle variabili matematiche immettendo, però, il comando nella casella di testo TX. Ripetiamo che prima dell'invio di un comando (qualsiasi esso sia, già implementato o implementato da voi), il microcontrollore della scheda deve sempre essere allertato con il comando di indirizzo. Se non farete così, il comando successi-

Fig. 4 - Altro esempio di debug. I comandi b0041 e b0142 hanno scritto rispettivamente A e B nelle locazioni EEPROM 00 e 01. Analogamente, a08041 e a08142 hanno scritto A e B a display. B00 e A080 hanno letto 16 caratteri dalla EEPROM e dalla RAM. D ha letto tutta la EEPROM.



corrispondono a 41 e 42, ossia A e B, come da comandi precedenti. Successivamente abbiamo inviato i comandi a08041 e a08142, con i quali abbiamo scritto gli stessi caratteri A e B nelle locazioni RAM 080 e 081 (quelle che contengono l'immagine dei messaggi visualizzati a display). Questa azione ha determinato la scrittura di AB sulle prime due celle della prima riga del display. A questo punto facciamo un'altra ipotesi: l'applicazione in fase di sviluppo, che fa uso del display per visualizzare informazioni e messaggi di servizio, ad un certo punto mostra delle scritte affette da caratteri strani che, secondo quanto da voi definito nel sorgente, non dovrebbero esserci: bene, con il comando A080 preleviamo (e verifichiamo) 16 byte per volta dalla sezione di RAM che

vo non verrà gestito dalla routine principale.

Note conclusive

Bene, con ciò vi abbiamo detto le cose più importanti relative alla nostra demo board per macchine PIC della famiglia 18F. Se deciderete di realizzare questo mini sistema di sviluppo, vi renderete conto di quanto sia versatile. Se non lo realizzerete, beh, potrete sempre utilizzare le routine contenute in entrambi i file Demo-01.asm e Demo-02.asm: esse rappresentano comunque un vasto magazzino di idee e di soluzioni già pronte, a disposizione dello sviluppatore che non deve far alto che includerle nella propria applicazione ed utilizzarle.

A noi non rimane che augurarvi buon lavoro.

CD3+

Disturbatore Cellulari
All Bands
23W totali min.

Vendita
soggetta
a
limitazioni



NEW

CK1+

Disturbatore
Cellulari
Tascabile
All Bands
300 mW
Portata 5+20m
Batteria ricaricabile
al litio



NEW

DESPY-PRO

Strumento professionale
per bonifiche



NEW

Ideale per
la ricerca
di microspie
fino a 8 Ghz

BIAS

s.a.

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

www.bias.sm

Riconoscitore di colori con speech processor

di *Fabio Riscica*

È in grado riconoscere il colore di un oggetto per mezzo di un convertitore colore-frequenza gestito direttamente da un microcontrollore. Dopo aver acquisito il colore, il micro effettua l'elaborazione e programma un sintetizzatore vocale affinché descriva il colore rilevato.



Cio che vi stiamo per presentare in queste pagine è qualcosa di veramente particolare, qualcosa che non abbiamo mai pubblicato prima d'ora; si tratta di un ausilio tiflotecnico, ossia un dispositivo che può essere utilizzato per offrire maggiore autonomia ai non vedenti nei settori più diversi: in casa, nell'educazione, nella formazione professionale e quindi nel lavoro, nello sport e nel tempo libero. È noto, infatti, che troppo spesso (per quanto ultimamente vi sia un accenno di inversione di tendenza) i non vedenti e, più in generale, le persone colpite da menomazioni di carattere fisico sono costrette a

vivere ruoli secondari rispetto ai cosiddetti normodotati. Ebbene, quello che vi presenteremo nelle prossime pagine è un riconoscitore di colori parlante, realizzato a fini didattici in collaborazione con il dipartimento di strumentazione biomedica dell'università degli studi di Trieste. In linea di massima, questo tipo di dispositivo è molto utile per accrescere l'autonomia dei non vedenti nel vissuto quotidiano: i modelli in commercio hanno dimensioni simili a quelle di un telefonino e permettono di sapere di che colore è un certo oggetto (i vestiti, le suppellettili di casa e, in linea di massima, qualsiasi altra cosa) gra-

>

zie ad un occhio artificiale in grado di rilevare l'intero spettro dei colori visibili; l'utilizzo, tipicamente, è molto semplice: si appoggia l'occhio dello strumento all'oggetto di cui si vuole conoscere il colore e si preme un pulsante per far sì che una voce sintetizzata enunci il colore e la sfumatura relativa.

Per un non vedente, il poter riconoscere il colore di qualcosa è certamente d'aiuto: ad esempio, conoscere i colori dei vestiti permette alla persona di scegliere autonomamente l'abbigliamento desiderato; la scelta di confezioni di prodotti alimentari, contraddistinte da contenitori di differente colore è un'altra possibile applicazione; si potrebbe quasi affermare che ogni acquisto effettuato da un non vedente verrebbe agevolato grazie al riconoscimento di un colore e di una sfumatura. Purtroppo, però, il costo dei dispositivi commerciali è abbastanza alto, vuoi per il contenuto tecnologico, vuoi anche per mere questioni commerciali e pertanto questi oggetti rimangono talvolta invenduti piuttosto che venire destinati a chi ne potrebbe fare buon uso. Il nostro riconoscitore parlante, certamente, non ha le prestazioni in termini di numero di colori e gradazioni distinte, dei dispositivi commerciali; esso, però, funziona secondo gli stessi principi impiegati dai prodotti commerciali e rappresenta un'eccellente applicazione didattica per capire la teoria alla base del funzionamento di questi apparecchi, garantendo contemporaneamente la possibilità di qualche utilizzo pratico. Elemento chiave del progetto è il convertitore colore/frequenza TCS230, prodotto da TAOS (Texas Advanced Optoelectronics Solutions), che genera un'onda quadra di frequenza proporzionale all'intensità luminosa delle tre componenti RGB del colore rilevato. Un microcontrollore acquisisce ed elabora il segnale in frequenza in modo che possa essere riconosciuto e classificato secon-

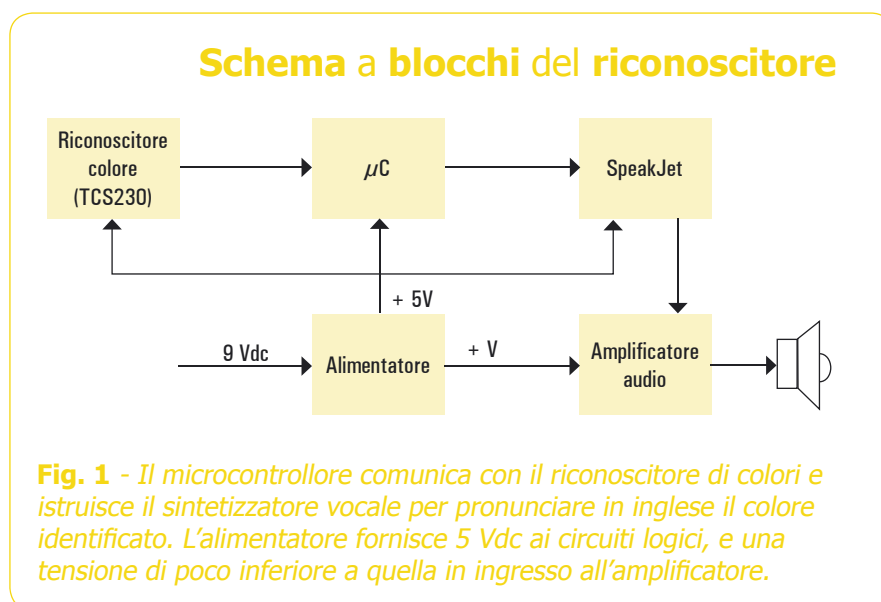


Fig. 1 - Il microcontrollore comunica con il riconoscitore di colori e istruisce il sintetizzatore vocale per pronunciare in inglese il colore identificato. L'alimentatore fornisce 5 Vdc ai circuiti logici, e una tensione di poco inferiore a quella in ingresso all'amplificatore.

do gli standard umani, quindi pilota il chip di sintesi vocale SpeakJet che, collegato ad un amplificatore e ad un altoparlante, riproduce la sequenza di *allofoni* (in fonetica, un allofono o *variante* è una realizzazione fonica di un fonema - suono particolare - che in una determinata lingua non ha carattere distintivo) che compone la frase relativa al colore rilevato (in lingua inglese). Vediamo insieme come è stato realizzato il dispositivo.

Schema a blocchi

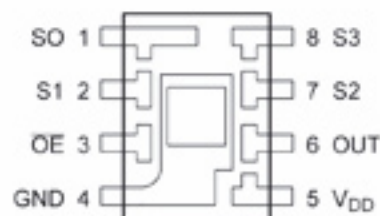
Lo schema a blocchi è riportato in Fig. 1: come potete notare, il riconoscitore può essere idealmente suddiviso in cinque blocchi, il primo dei quali è il sensore dei colori, ossia un circuito elettronico basato sul convertitore colore frequenza modello TAOS TCS230, cui abbiamo dedicato un paragrafo. Questo sensore comunica in modo bidirezionale con il blocco microcontrollore: in pratica, il TCS230 invia le informazioni sul colore "visto" su richiesta del microcontrollore. Questi, dopo aver elaborato le informazioni ricevute, genera una sequenza di comandi da inviare al terzo blocco, ossia il sintetizzatore vocale (lo SpeakJet, cui abbiamo dedicato un altro paragrafo), in modo che quest'ultimo pronunci il colo-

re individuato dal sensore. Poiché l'uscita dello SpeakJet non è abbastanza potente, occorre amplificarla tramite un piccolo amplificatore audio, idoneo a pilotare un altoparlante da 1 W massimo su 8 ohm: tale funzione è svolta dal quarto blocco. Non rimane che lo stadio di alimentazione: esso preleva i 9 Vdc in ingresso e genera le due tensioni V+ e 5 Vdc, necessarie per alimentare l'amplificatore audio la prima e tutto il resto del circuito la seconda.

Il convertitore colore-frequenza TCS230

Il dispositivo TCS230 integra un sistema configurabile di fotodiodi ed un convertitore corrente-frequenza, in un singolo circuito integrato CMOS secondo la pinout riportata in Fig. 2. La tensione di alimentazione è unipolare e può essere compresa fra 2,7

Fig. 2 - Pinout del chip riconoscitore di colori TCS230.



e 5,5 Vdc. Il fondo scala della frequenza di uscita può essere scalato di uno fra tre valori selezionabili mediante i due ingressi di controllo S0 e S1. Se tali ingressi sono entrambi a livello logico basso, il chip è in stato di risparmio energetico. Se S0 ed S1 sono entrambi a livello logico alto, la frequenza in uscita è a pieno range (100%). Se S0 è a livello logico alto ed S1 a livello logico basso, la frequenza di uscita è il 20% di quella a pieno range. Se, infine, S0 è a livello logico basso ed S1 a livello logico alto, la frequenza di uscita è il 2% di quella a pieno range. Le linee di ingresso e uscita digitali

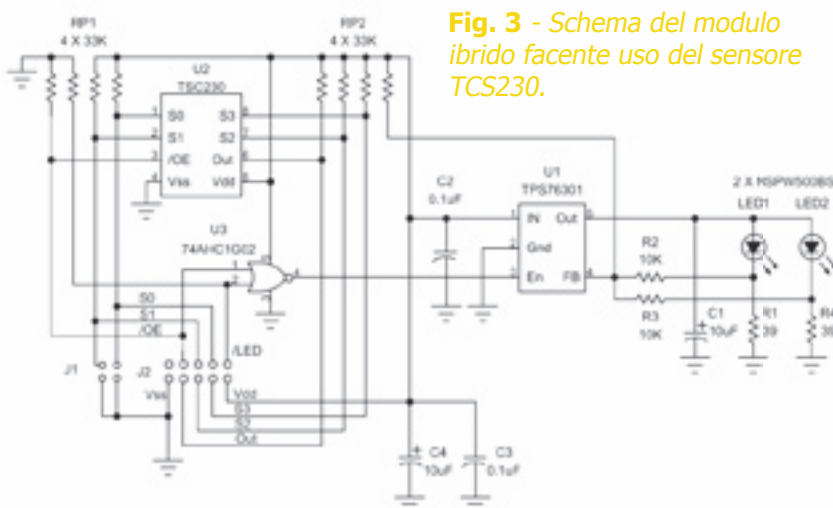
del colore). Tutti i sedici fotodiodi dello stesso colore sono connessi in parallelo ed il tipo di fotodiodo utilizzabile dal dispositivo è selezionabile mediante gli ingressi di controllo S2, S3. Se essi sono a livello logico basso viene selezionato il gruppo di fotodiodi rossi, se S2 è a livello logico basso e S3 a livello logico alto viene selezionato il gruppo blu. Se S2 è a livello logico alto e S3 a livello logico basso viene selezionato il gruppo senza filtro; se, infine, S2 e S3 sono entrambi a livello logico alto viene selezionato quello verde. Noi non abbiamo utilizzato direttamente il chip, bensì un modulo (vi-

tamente proporzionali all'intensità di rosso, verde o blu misurato sull'oggetto.

Fig. 4 - Ecco come si presenta il modulo ibrido usato in questo progetto.



Fig. 3 - Schema del modulo ibrido facente uso del sensore TCS230.



Se si abilitano i fotodiodi senza filtri, si acquisisce tutto lo spettro della radiazione riflessa ed in tal modo si può effettuare ad esempio la calibrazione del bianco iniziale.

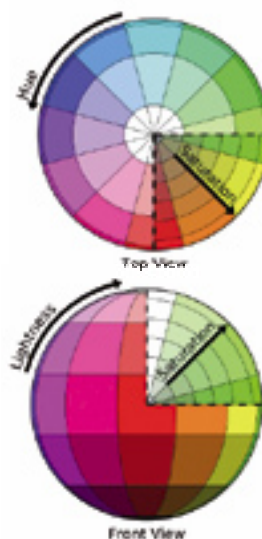
Classificazione dei colori

Mentre leggere un colore in funzione di una frequenza può essere facile, non è altrettanto facile descrivere un colore a chi non lo può vedere. Il sistema di rilevazione dei colori del sensore TCS230 è di tipo RGB e non si presta direttamente per una classificazione percettiva, basata sul sistema HSL (Hue, Saturation, Lightness). In ➤

possono essere interfacciate direttamente ad un microcontrollore o ad un altro circuito. Il terminale di OUTPUT ENABLE (/OE) pone l'uscita in frequenza in uno stato di alta impedenza, condizione necessaria per la condivisione di più unità in parallelo da parte di un microcontrollore. Il convertitore colore-frequenza dispone di un array di 8x8 fotodiodi. Sedici fotodiodi hanno dei filtri blu, sedici hanno dei filtri verdi, sedici dei filtri rossi e gli ultimi sedici non hanno filtri. I quattro tipi di fotodiodi sono intersecati per minimizzare l'effetto di non uniformità della radiazione incidente (in altre parole, per aumentare l'efficienza nel riconoscimento

sibile in Fig. 4; la Fig. 3 ne riporta lo schema elettrico) che integra due diodi led ad alta luminosità usati come illuminatori. Il pin /LED, se attivo (basso), fa accendere i due led. L'uscita in frequenza deve essere abilitata attraverso il pin /OE (attivo basso). In pratica, il funzionamento del sensore può essere così riassunto: una volta selezionato l'opportuno fondo scala della frequenza di uscita mediante i pin S0 ed S1, è sufficiente attivare l'illuminatore ed abilitare alternativamente i fotodiodi con i filtri rossi, verdi o blu per acquisire la radiazione riflessa da un oggetto "guardato"; ne conseguiranno delle frequenze del segnale di uscita diret-

Fig. 5 - La sfera dei colori.



letteratura è, infatti, nota la classificazione dei colori basata sulla relazione tra emozioni umane e colori, proposta da Johannes Itten e legata al sistema HSL che (basandosi a sua volta sulla teoria di Philipp Otto Runge) selezionò dodici tinte fondamentali e impose cinque livelli di luminanza e tre livelli di saturazione per ogni colore, ottenendo un totale di 180 colori fondamentali, rappresentati sulla superficie di una sfera (Fig. 5). I 12 colori puri sono posti lungo il cerchio equatoriale, la luminanza varia lungo i meridiani, e la saturazione aumenta con l'aumentare del raggio; i colori tra loro opposti sono situati alle posizioni diametralmente opposte. Questa disposizione permette un'oggettiva identificazione delle peculiarità dei colori, identificando cinque differenti tipi di contrasto: contrasto di tinta, contrasto chiaro/scuro, contrasto caldo/freddo, contrasto complementare, contrasto di saturazione. I descrittori di contrasto sopra citati possono essere utilizzati per definire una rappresentazione descrittiva del contenuto di un'immagine; tale rappresentazione, però, risulta fortemente condizionata dalla cultura e dalle esperienze individuali.

utilizzato nel digitale per trasmettere immagini a colori. Qualsiasi immagine, infatti, può essere scomposta, attraverso filtri, in questi tre colori base che, miscelati tra loro, danno tutto lo spettro del visibile. L'RGB è un modello additivo in cui unendo i tre colori con la loro intensità massima si ottiene il bianco (tutta la luce viene



Fig. 8
I colori secondo il cono HSL.

riflessa). La combinazione delle coppie di colori dà il ciano, il magenta e il giallo (vedere Fig. 6).

Lo spazio RGB può essere rappresentato con un cubo (Fig. 7), i cui spigoli sono: rosso, giallo, verde, blu, ciano, magenta, bianco e nero. Ciascuno dei tre valori RGB è compreso tra 0 e 255. La scala RGB, però, non permette di classificare il colore secondo un modello percettivo (cosa, invece, possibile con il modello HSL), essendo orientata all'hardware.

ta (Hue), saturazione (Saturation), luminosità (Lightness). La tinta definisce la componente cromatica, cioè la frequenza dell'onda luminosa, e rappresenta un angolo che può spaziare tra 0° e 360°: tipicamente 0° è rosso, 60° giallo, 120° verde, 180° blu, 240° magenta. Gli altri colori sono distribuiti linearmente tra questi angoli. La saturazione definisce quanto il colore sia vicino al grigio: 0 indica grigio mentre 1 rappresenta il colore primario puro. La scala è divisa in 100 passi (può in maniera equivalente essere espressa in valori percentuali).

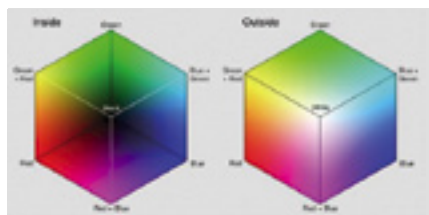
La luminosità rappresenta la luce media del colore. Come per la saturazione, il valore è compreso tra 0 e 1 in una scala di 100 valori. Diminuendo la luminosità si riduce il valore dei colori primari mantenendoli però nello stesso rapporto.

Lo spazio HSL è rappresentabile graficamente con un doppio cono (Fig. 8), in cui le due punte corrispondono al nero e al bianco. Il parametro angolare corrisponde alla tinta, la distanza dall'asse equivale la saturazione e la distanza lungo l'asse nero-bianco definisce la luminosità. Se lo spazio dei colori è rappresentato da



Fig. 6 - I colori nello spazio RGB.

Fig. 7 - Il cubo RGB.



RGB, invece, è il nome di un modello di colori additivo che si basa sui tre colori primari: rosso (Red), verde (Green) e blu (Blue). È un modello

Listato 1

```
#define MAX(a,b) (((a)>(b)) ? (a):(b))
#define MIN(a,b) (((a)<(b)) ? (a):(b))

// Converte il formato RGB
// nel range da 0 a 1 in formato
// HSL, sempre nel range da 0 a 1.

void ToHSL(const float red, const float green, const float blue, float& hue, float& saturation, float& luminance)
{
    float fmax, fmin;
    fmax=MAX(MAX(red, green), blue);
    fmin=MIN(MIN(red, green), blue);
    luminance = fmax;

    if (fmax > 0)
        saturation = (fmax-fmin)/fmax;
    else
        saturation = 0;
    if (saturation == 0)
        hue = 0;
    else
    {
        if (fmax == red)
            hue = (green - blue) / (fmax - fmin);
        else if (fmax == green)
            hue = 2 + (blue - red) / (fmax - fmin);
        else
            hue = 4 + (red - green) / (fmax - fmin);
        hue = hue / 6;
        if (hue < 0) hue += 1;
    }
}
```

Il modello HSL utilizza uno spazio dei colori che è una deformazione non lineare del cubo dei colori del sistema RGB. Ha tre coordinate: tin-

dischi di diversa luminosità, allora la tinta e la saturazione sono equivalenti alle coordinate polari di qualunque punto del piano.

Tabella 1 - Alcuni dei fonemi dello speakJet

Vowels			Voiced Fictive		
Long A - "Gate" & "Ate"	EYIY		V - "Vest" & "Even"	VV	
Long E - "See" & "Even"	IY		Z - "Zoo" & "Zap"	ZZ	
Long I - "Sky" & "Five"	OHIH	Also see IE	ZH - "Azure" & "Treasure"	ZH	
Long O - "Comb" & "Over"	OW	Also see OA	TH - "There" & "That"	DH	
Vowels with R			Voiceless Fricative		
R - "Ray" & "Brain"	RR		H - "Help" & "Hand"	HE	Front
AIR - "Hair" & "Stair"	EYRR		H - "Hoe" & "Hot"	HO	Back
AR - "Part" & "Farm"	AWRR	Also see IE	WH - "Who" & "Whale"	WH	
EAR - "Clear" & "Hear"	IYRR	Also see OA	F - "Food" & "Effort"	FF	

Tabella 2 - Alcuni comandi

Dec.	SpeakJet Use
000	Pause 0
001	Pause 1
002	Pause 2
003	Pause 3
004	Pause 4
005	Pause 5
006	Pause 6
007	Next Sound Fast
008	Play Next Sound Slow
014	Play Next Sound High Tone
015	Play Next Sound Low Tone
016	Wait
020	Volume, X

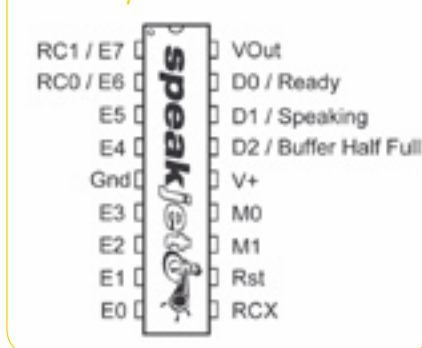
Tabella 3 - Elenco parziale degli allofoni

Code	Phoneme	Sample Words	Msec.	Phoneme Type
128	IY	See, Even, Feed	70	Voiced Long Vowel
129	IH	Sit, Fix, Pin	70	Voiced Short Vowel
130	EY	Hair, Gate, Beige	70	Voiced Long Vowel
131	EH	Met, Check, Red	70	Voiced Short Vowel
132	AY	Hat, Fast, Fan	70	Voiced Short Vowel
133	AX	Cotten	70	Voiced Short Vowel
134	UX	Luck, Up, Uncle	70	Voiced Short Vowel
135	OH	Hot, Clock, Fox	70	Voiced Short Vowel
136	AW	Father, Fall	70	Voiced Short Vowel
137	OW	Comb, Over, Hold	70	Voiced Long Vowel
138	UH	Book, Could, Should	70	Voiced Short Vowel
139	UW	Food, June	70	Voiced Long Vowel
140	MM	Milk, Famous,	70	Voiced Nasal

Utilizzando uno spazio dei colori HSL è possibile effettuare una classificazione degli stessi in termini percettivi introducendo dei test sui valori della tinta per definire il colore, sul valore della saturazione per definire quanto la tinta sia viva o spenta, sul valore della luminosità per definire quanto la tinta sia chiara o scura. Ad esempio, il rosso vivo chiaro è caratterizzato da una tinta compresa tra 0° e 12°, da una saturazione maggiore di 0,5 e da una luminosità compresa tra 0,6 e 0,8.

Affinché sia possibile effettuare tale classificazione, è necessario convertire i livelli RGB acquisiti nello spazio HSL; a tal fine sono disponibili svariati algoritmi fra i quali quello in linguaggio C riportato nel Listato 1,

Fig. 9 - Pinout del sintetizzatore vocale SpeakJet



che abbiamo implementato nel nostro firmware.

Il chip SpeakJet

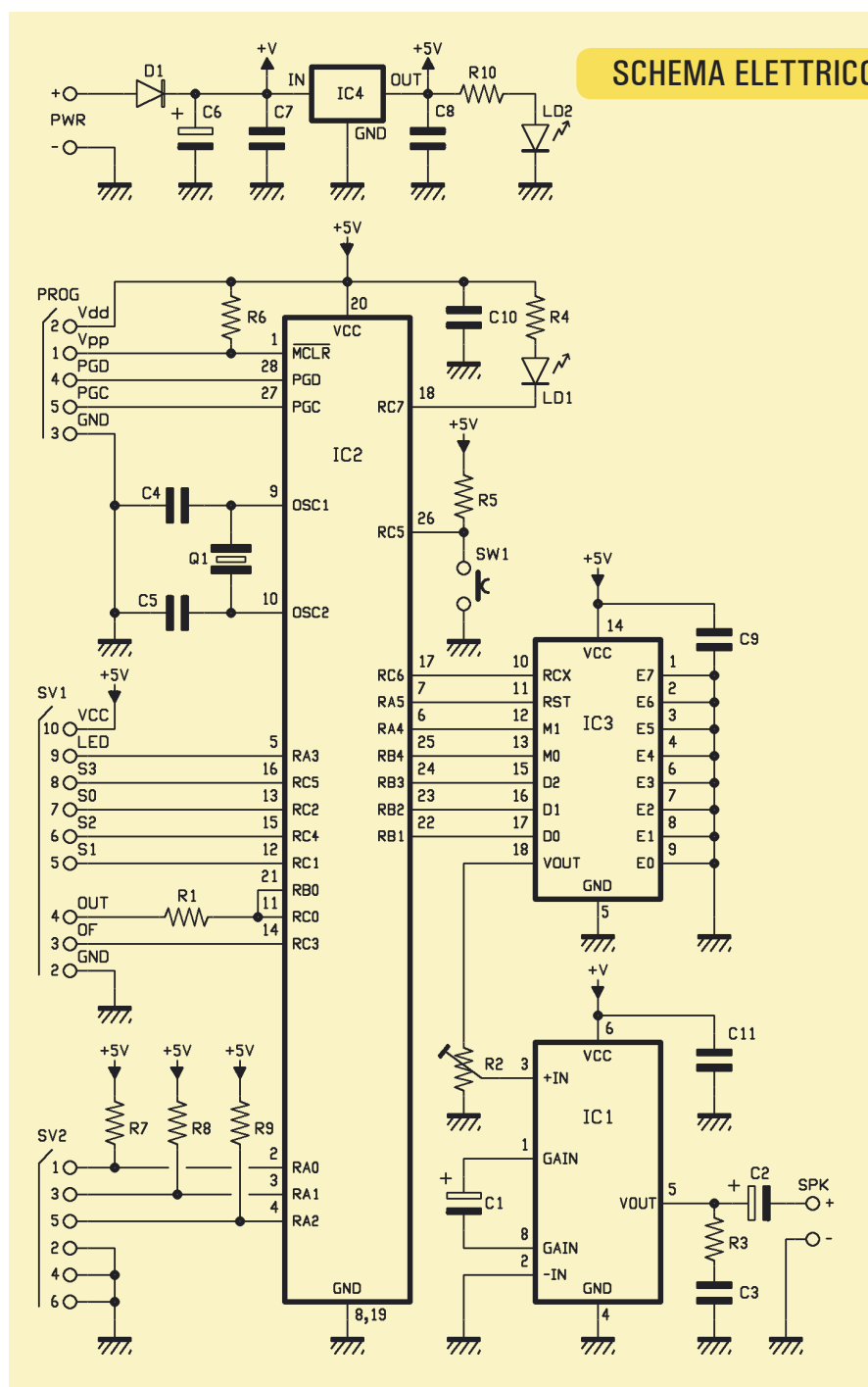
Dopo aver visto il sensore colori e i principi secondo cui i colori vengono classificati, vediamo ora il secondo

elemento chiave del progetto, ossia il sintetizzatore vocale. SpeakJet è un sintetizzatore di voce e di suoni complessi a singolo chip che utilizza la tecnologia MSA (Mathematical Sound Architecture) per controllare i suoi cinque canali interni usati per generare un vocabolario pressoché illimitato di sintesi vocale e di suoni complessi.

Il chip (la cui pinout è riportata in Fig. 9) è preconfigurato con 72 allofoni, 43 effetti sonori e 12 toni DTMF. Attraverso la selezione di questi componenti MSA in combinazione con il controllo del pitch (variazione di tono), del rate (velocità di riproduzione), del bend (variazione forzata della frequenza) e del volume, è possibile produrre frasi illimitate ed effetti

sonori con migliaia di variazioni in ogni momento. Non si tratta di forme d'onda registrate o di frammenti di suono, ma di vero suono sintetizzato partendo da formule matematiche. SpeakJet può essere controllato simultaneamente sia attraverso transizioni logiche su di una delle sue otto linee relative agli eventi sia mediante dati seriali trasmessi da una CPU, consentendo così operazioni controllate da micro oppure stand-alone. Altre caratteristiche sono la presenza di un buffer d'ingresso da 64 byte, una EEPROM interamente utilizzabile, tre uscite programmabili e la possibilità di accedere direttamente ai cinque canali interni di sintesi del suono. L'alimentazione del chip può essere compresa tra 2 e 5,5 V. Il pin RCX costituisce l'ingresso seriale mediante cui le periferiche esterne comunicano con SpeakJet. I livelli di segnale sono TTL e la configurazione seriale è predefinita a 9.600 baud, 8 bit dati, nessuna parità, 1 bit di stop. È possibile modificare il baud rate in qualsiasi momento commutando il chip nel modo di configurazione (ponendo dopo il reset il pin M1 a livello logico basso).

Il pin M0, se attivo dopo il reset, seleziona la modalità dimostrativa del chip. Nel modo di funzionamento normale il pin M0 deve essere posto a livello logico basso e il pin M1 a livello logico alto attraverso una resistenza di pull-up. I tre pin di uscita D0, D1, D2, segnalano rispettivamente che lo SpeakJet è pronto per accettare comandi, che sta riproducendo suoni e che il suo buffer è mezzo pieno. Il pin di master reset RST permette di resettare il chip alla condizione di power-up; nel funzionamento normale deve essere tenuto a livello logico alto. Il pin Vout fornisce l'uscita analogica che, per pilotare un altoparlante, deve essere amplificata. Per quanto riguarda il set di comandi MSA e la tabella dei codici relativi agli allofoni, si faccia riferimento alle Tabelle 1, 2 e 3, che



riportano parte di tali comandi. Quelle complete sono riportate nel datasheet del dispositivo. L'utilizzo del chip è estremamente semplice: basta inviare in modo seriale (con formato 9600,N,8,1) sul pin RCX la sequenza dei caratteri relativi agli allofoni che si vogliono riprodurre. Qui di seguito sono riportati, a titolo di esempio, allofoni e codici di controllo utilizzati per la pronuncia di alcune parole inglesi.

...
 hello = \HE \FAST \EHLE \LO \OWWW
 mail= \MM \EYIY \LO
 dog = \DO \OH \OH \OG
 memory = \MM \EH \MM \FAST \AXRR \Y
 goodbye=\Slow \GO \UW \OD \FAST \P4 \SOFT \BO \OHIH
 welcome = \WW \EHLE \KE \UX \MM
 ...

Schema elettrico

Il circuito, in verità piuttosto semplice, è basato su un microcontrollore PIC16F876 (IC2), che dispone di 22 pin di I/O liberamente configurabili come ingressi ed uscite, molti dei quali sono stati utilizzati per comunicare con il modulo TCS230 (RA3, RB0 e RC0÷RC5), e con il chip

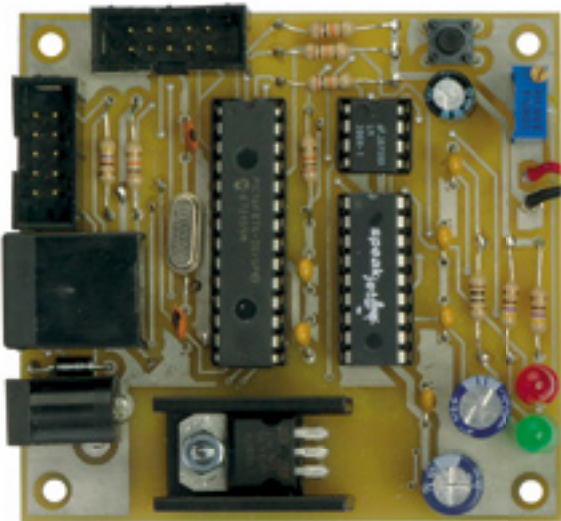
integrati digitali. Fa eccezione l'integrato LM386 che amplifica l'uscita dello SpeakJet in modo da pilotare l'altoparlante, alimentato direttamente dalla tensione di ingresso, che deve essere anch'essa preventivamente stabilizzata. Sono stati previsti degli ingressi aggiuntivi, accessibili tramite il connettore SV2, per il collega-

all'utente ulteriori eventi, come ad esempio che la misura è in corso o che si sono verificati errori.

Realizzazione pratica

Come potete vedere, il circuito non è assolutamente complesso da montare: esso, infatti, impiega solamente componenti ad inserzione montati

piano di MONTAGGIO

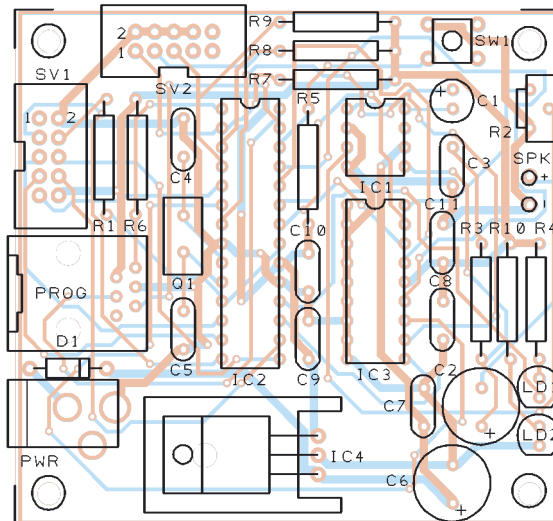


ELENCO COMPONENTI:

R1, R4, R10: 470 ohm
R2: trimmer multigiri 10 kohm
R3: 10 ohm
R5÷R9: 10 kohm
470 µF 25 VL elettrolitico
C1: 10 µF 100 VL elettrolitico
C2, C6: 220 µF 25 VL elettrolitico
C3: 100 nF multistrato
C4, C5: 15 pF ceramico
C7÷C11: 100 nF multistrato
IC1: LM386N
IC2: PIC16F876A (MF771)

IC3: SpeakJet
IC4: 7805
D1: 1N4007
LD1: led 5 mm rosso
LD2: led 3 mm verde
Q1: Quarzo 20 MHz
SW1: Microswitch
Varie:
- Sensore TCS230LM
(www.parallax.com o www.artek.it)
- Plug alimentazione
- Dissipatore (ML26)

- Vite 10 mm 3 MA
- Dado 3 MA
- Connettore RJ11 da cs
- Zoccolo 14+14
- Zoccolo 9+9
- Zoccolo 4+4
- Connettore POD 10 poli maschio da cs
- Connettore POD 10 poli femmina (2 pz.)
- Altoparlante 0,5W/8 ohm (MLS2)
- Flat cable 10 vie
- Circuito stampato



SpeakJet (RA4, RA5, RB1÷RB4 e RC6). Per il conteggio degli impulsi, il micro dispone di un contatore hardware (facente capo al pin RC0) collegato direttamente all'uscita del modulo TCS230. È inoltre disponibile un'interfaccia seriale RS232 hardware (usata per comunicare con lo SpeakJet). La tensione di alimentazione viene stabilizzata a 5V DC dal regolatore integrato 7805, che fornisce l'alimentazione ai circuiti

mento di eventuali fotodiodi per altre opzioni future come il rilevamento della luminosità dell'ambiente o di spie luminose. Il connettore PROG permette la programmazione del micro ed il debug del firmware tramite il debugger ICD2. Il pulsante SW1 è stato previsto per segnalare eventi al micro, come ad esempio la richiesta di riconoscimento del colore. Il led LED1, gestito tramite la linea RC7, viene utilizzato per segnalare

su un circuito stampato doppia faccia che può essere realizzato con i master scaricabili gratuitamente dal nostro sito (www.elettronica.in.it). Aiutandovi con le illustrazioni riportate nel piano di montaggio, saldate per primi i componenti a basso profilo quali resistenze, condensatori non polarizzati, il pulsante, eccetera. Montate quindi gli zoccoli per i tre chip, i condensatori elettrolitici (attenzione alla polarità) e i connet-

Listato 1

Riportiamo di seguito il codice sorgente completo, scritto in linguaggio ANSI C. Il file può essere scaricato gratuitamente dal sito della rivista (www.elettronica.in.it). Il principio di funzionamento è molto semplice: il micro attende la pressione di un pulsante, quindi acquisisce il colore in formato RGB dal sensore, lo traduce nel formato HSL, quindi, analizza il risultato della conversione per scegliere il set di comandi da inviare al sintetizzatore vocale, che pronuncerà, così, il colore identificato.

```

/*****
/*
/*      TAOS INTERFACE
/*
/* Autore       :      Fabio Riscica
/* Data        :      09/05/2007
/* Versione    :      1.0
/* Compilatore :      CCS C COMPILER
/*
/* Revisioni   :
/* 1.0 09/05/07 Versione iniziale.
/*
*****/

#include <16F876.h>
#define HS, NOWDT, PUT, NOPROTECT, BROWNOUT, NOLVP
#define delay(clock=2000000)
#define rs232(baud=9600, parity=N, BITS=8, STOP=1, xmit=PIN_C6)

#define AIO PIN_A0 /* input esterno 1 (I) */
#define AI1 PIN_A1 /* input esterno 2 (I) */
#define AI2 PIN_A2 /* input esterno 3 (I) */
#define LED PIN_A3 /* TAOS led (attivo basso) (O) */
#define M1 PIN_A4 /* Speakjet mode select 1 (O) */
#define RST PIN_A5 /* Speakjet reset attivo basso (O)*/
#define FIN PIN_B0 /* TAOS uscita in frequenza (I) */
#define D0 PIN_B1 /* Speakjet ready (I) */
#define D1 PIN_B2 /* Speakjet speaking (I) */
#define D2 PIN_B3 /* Speakjet buffer half full (I)*/
#define M0 PIN_B4 /* Speakjet mode select 0 (O) */
#define PUSH PIN_B5 /* pulsante attivo basso (I) */

#define FOUT PIN_C0 /* TAOS uscita in frequenza (I) */

#define S1 PIN_C1 /* TAOS frequency scaling (O) */
#define S0 PIN_C2 /* TAOS frequency scaling (O) */
#define OE PIN_C3 /* TAOS enable output frequency
/* attivo basso (O) */
#define S2 PIN_C4 /* TAOS photodiode selection (O)*/
#define S3 PIN_C5 /* TAOS photodiode selection (O)*/
#define LED1 PIN_C7 /* led attivo basso (O) */

float hue,saturation,lightness;

#define MAX1(a,b) (((a) > (b)) ? (a) : (b))
#define MAX2(a,b) (((a) > (b)) ? (a) : (b))
#define MIN1(a,b) (((a) < (b)) ? (a) : (b))
#define MIN2(a,b) (((a) < (b)) ? (a) : (b))

void RGBtoHSL(int r,int g,int b)
{
    float fmax,fmin;
    float var_R,var_G,var_B;
    extern float hue,saturation,lightness;

    var_R = r/255.0;
    var_G = g/255.0;
    var_B = b/255.0;

    fmax = MAX1(MAX2(var_R,var_G),var_B);
    fmin = MIN1(MIN2(var_R,var_G),var_B);

    lightness = fmax;

    if (fmax > 0)
        saturation = (fmax - fmin)/fmax;
    else
        saturation = 0;

    if (saturation == 0)
        hue = 0;
    else
    {
        if (fmax == var_R)
            hue = (var_G - var_B)/(fmax - fmin);
        else if (fmax == var_G)
            hue = 2 + (var_B - var_R)/(fmax - fmin);
        else hue = 4 + (var_R - var_G)/(fmax - fmin);
        hue = hue/6;
        if (hue < 0) hue += 1;
        hue*=360;
    }
}

void main()
{
    long red_value,green_value,blue_value;
    int red,green,blue;
    char buffer[16];

    extern float hue,saturation,lightness;

    output_high(RST);
    output_low(M0);
    output_high(M1);
    output_high(OE);
    output_high(LED);
    output_high(S0); /* TAOS 100% frequency scaling */
    output_high(S1);
    output_low(S2);
    output_low(S3);
    output_low(LED1);

```

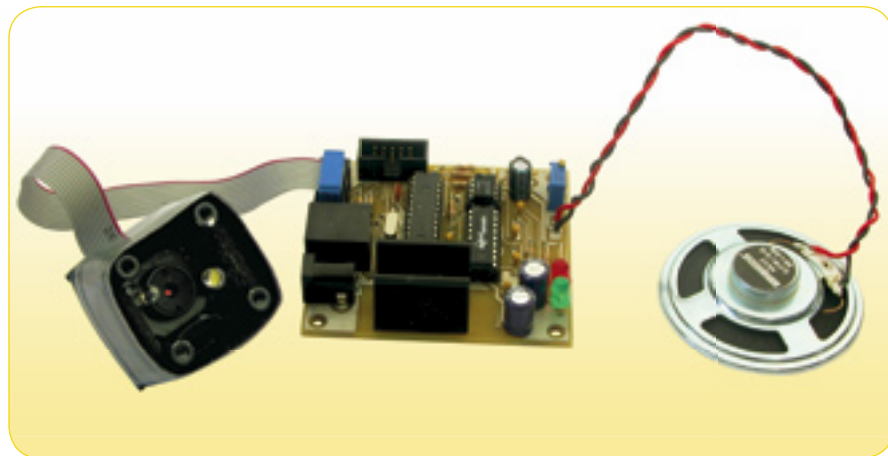
tori. Terminate il montaggio con lo stabilizzatore di tensione, che va montato a 90° a ridosso di un piccolo dissipatore a U. Collegate, infine,

il sensore di colore al connettore J2 per mezzo di un flat cable a 10 poli dritto () corrispondenza dei pin 1-1, 2-2 eccetera) di lunghezza adeguata.

Un'ultima occhiata alla ricerca di errori quindi posizionate i tre chip; il microcontrollore deve essere programmato con il file esadecimale MF771.hex, scaricabile anch'esso dal sito, attraverso il connettore PROG.

Il software e l'utilizzo in pratica

Il software di gestione del microcontrollore, cui abbiamo dedicato un riquadro, è stato implementato in linguaggio ANSI C utilizzando il compilatore PIC-C midrange prodotto da CCS. Le operazioni che il software esegue sono sostanzialmente; innanzitutto attende la



```

setup_timer_1(T1_EXTERNAL_SYNC|T1_DIV_BY_1);
while(1)
{
    while(input(PUSH)); /* Pulsante premuto */
    buffer[0]=195; /* pronuncia "color check" */
    buffer[1]=8;
    buffer[2]=134;
    buffer[3]=146;
    buffer[4]=7;
    buffer[5]=151;
    buffer[6]=5;
    buffer[7]=182;
    buffer[8]=131;
    buffer[9]=131;
    buffer[10]=196;
    buffer[11]=0;
    printf("%s",buffer);
    while(input(D1));
    output_low(LED1);
    output_low(OE); /* abilita sensore */
    output_low(LED); /* accende illuminatore */
    delay_ms(500);
    output_low(S2); /* seleziona fotodiodi rossi */
    output_low(S3);
    delay_ms(10);
    set_timer1(0);
    delay_ms(1000);
    red_value=get_timer1(); /* legge fotodiodi */
    output_high(S2); /* seleziona fotodiodi verdi */
    output_high(S3);
    delay_ms(10);
    set_timer1(0);
    delay_ms(1000);
    green_value=get_timer1(); /* legge fotodiodi */
    output_low(S2); /* seleziona fotodiodi blu */
    output_high(S3);
    delay_ms(10);
    set_timer1(0);
    delay_ms(1000);
    blue_value=get_timer1(); /* legge fotodiodi */
    output_high(LED);

    red = red_value*255.0/31261;
    green = green_value*255.0/46385;
    blue = blue_value*255.0/63866;

    RGBtoHSL(red,green,blue); /* Converte da RGB a HSL */

    output_high(LED1);

    if (lightness >= 0.90) /* Se il colore è bianco */
    {
        buffer[0]=185; /* Comandi allofoni */
        buffer[1]=155; /* pronuncia "white" */
        buffer[2]=191;
        buffer[3]=0;
        printf("%s",buffer);
        while(input(D1));
    }
}

}
else if (hue<30) /* Se il colore è rosso */
{
    buffer[0]=148; /* Comandi allofoni */
    buffer[1]=8; /* pronuncia "red" */
    buffer[2]=134;
    buffer[3]=131;
    buffer[4]=176;
    buffer[5]=0;
    printf("%s",buffer);
    while(input(D1));
}
else if (hue<73) /* Se il colore è giallo */
{
    buffer[0]=128; /* Comandi allofoni */
    buffer[1]=131; /* pronuncia "yellow" */
    buffer[2]=146;
    buffer[3]=137;
    buffer[4]=0;
    printf("%s",buffer);
    while(input(D1));
}
else if (hue<160) /* Se il colore è verde */
{
    buffer[0]=8; /* Comandi allofoni */
    buffer[1]=179; /* pronuncia "green" */
    buffer[2]=7;
    buffer[3]=148;
    buffer[4]=8;
    buffer[5]=128;
    buffer[6]=141;
    buffer[7]=0;
    printf("%s",buffer);
    while(input(D1));
}
else if (hue<252) /* Se il colore è blu */
{
    buffer[0]=171; /* Comandi allofoni */
    buffer[1]=7; /* pronuncia "blue" */
    buffer[2]=146;
    buffer[3]=162;
    buffer[4]=0;
    printf("%s",buffer);
    while(input(D1));
}
else if (hue<360) /* Se il colore è porpora */
{
    buffer[0]=199; /* Comandi allofoni */
    buffer[1]=7; /* pronuncia "purple" */
    buffer[2]=151;
    buffer[3]=199;
    buffer[4]=8;
    buffer[5]=146;
    buffer[6]=0;
    printf("%s",buffer);
    while(input(D1));
}
}
}

```

pressione del pulsante, cui seguono l'abilitazione del modulo TCS230 ed il conteggio degli impulsi generati in corrispondenza dei tre diversi

gruppi di fotodiodi: tali frequenze vengono memorizzate ed il livello RGB viene convertito nel sistema HSL (Hue, Saturation, Lightness)

secondo l'algoritmo illustrato in precedenza. Viene infine analizzato il formato HSL del colore e selezionato il set di comandi con cui ➤



Alcuni "color-teller" commerciali

Il mercato offre differenti modelli di riconoscitori di colori. Essi possono essere acquistati sia in negozi specializzati, sia su siti Internet consigliati dalle varie associazioni di non vedenti. Noi segnaliamo i modelli ColorTest2000 (Fig. A) e Colorino (Fig. B), in vendita su <http://www.cambratech.it/test.htm> rispettivamente a 728,00 e 208,00 Euro, e il Color Teller (Fig. C), in vendita su <http://www.brytech.com/colorteller/> a 155 dollari.



Fig. A



Fig. B



Fig. C

deve essere pilotato via seriale il chip SpeakJet, set che serve per far pronunciare il colore al dispositivo. In questa release firmware è stata prevista la gestione di un numero limitato di colori base, non viene presa in considerazione la luminosità e non è stata prevista la procedura di calibrazione del bianco: vengono correttamente rilevati sei colori, cioè bianco (white), rosso (red), giallo (yellow), verde (green), blu (blue), viola (purple). La normalizzazione dei livelli RGB è stata effettuata in modo empirico mediante misure di laboratorio in corrispondenza a valori standard di rosso, verde, blu. Dopo aver montato il circuito e programmato il microcontrollore con il firmware relativo, possiamo passare al test del nostro dispositivo. Per una prima verifica è opportuno munirsi

di alcuni fogli con i colori base rosso, verde e blu, che possono essere prodotti da una stampante a colori. Il sensore di colore deve essere opportunamente schermato dalla luce lungo i suoi bordi, per evitare che la luce ambiente disturbi la misura. Notate, infatti, che nelle foto pubblicate in questo articolo abbiamo avvolto un cartoncino scuro ricoperto di nastro isolante nero sul lato del c.s. del sensore. Appoggiando il modulo sul foglio ed agendo sul pulsantino di test, viene riprodotta la frase "color check" e quindi noteremo l'attivazione dell'illuminatore. Dopo circa tre secondi il dispositivo enuncerà il nome del colore rilevato. A questo punto è possibile verificare anche oggetti di altri colori, facendo attenzione che anche il colore di questi venga rilevato correttamente.

Come avrete notato, il riconoscimento di un colore viene effettuato semplicemente con una serie di IF e ELSE IF che analizzano i vari valori delle variabili *lightness* e *hue*. Non vi sarà difficile aggiungere altri controlli IF (fino ad esaurimento della memoria) che selezionino differenti comandi da inviare allo SpeakJet in modo da pronunciare colori differenti. Si osservi che la tecnica di sintesi vocale scelta ha dei limiti in termini qualità di riproduzione, dato che la pronuncia risulta essere metallica. Appare inoltre problematica l'implementazione in lingua italiana; non è, infatti, facilmente realizzabile la conversione diretta degli allofoni inglesi con i corrispondenti in lingua italiana. Per un utilizzo nel nostro paese, si può sostituire il chip SpeakJet con un dispositivo per la pronuncia di frasi preregistrate, memorizzate in EEPROM.

Conclusioni e ringraziamenti

Ci sembra doveroso, prima della conclusione, ringraziare la dottoressa Monica Cusenza, che ci è stata di aiuto con il suo studio sulla rappresentazione dei colori e la prima implementazione del dispositivo effettuata presso il laboratorio di strumentazione biomedica dell'università degli studi di Trieste. Desideriamo ringraziare particolarmente il prof. Agostino Accardo per la disponibilità ed i suggerimenti fornitici durante la realizzazione del progetto.

per il

MATERIALE

Tutti i componenti utilizzati in questo progetto sono facilmente reperibili in commercio. Il master del circuito stampato ed il firmware sono scaricabili dal sito della rivista (www.elettronica.in). Il microcontrollore è anche disponibile già programmato (Cod. MF771) al prezzo di 20,00 Euro; il sintetizzatore audio (Cod. SpeakJet) costa invece 38,00 Euro, IVA compresa.

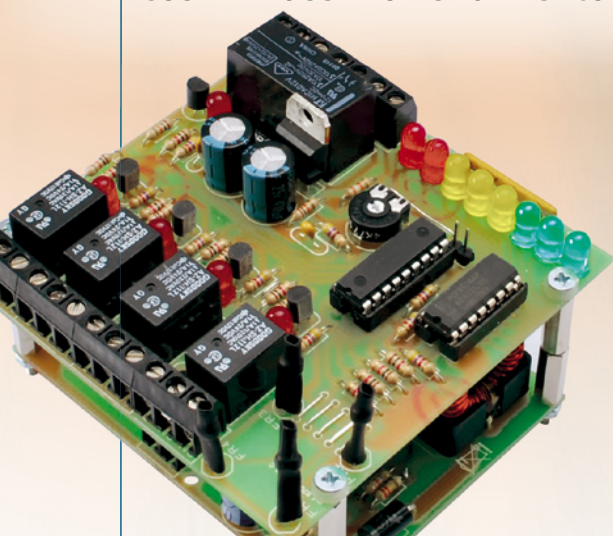
Il materiale va richiesto a: Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)

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

Inseguitore motorizzato per impianti solari

di Davide Scullino

Progettato per impianti di piccola potenza, orienta il pannello fotovoltaico in modo da fargli ricevere, in qualsiasi momento della giornata, la luce del sole in maniera ottimale garantendo così il massimo rendimento possibile.



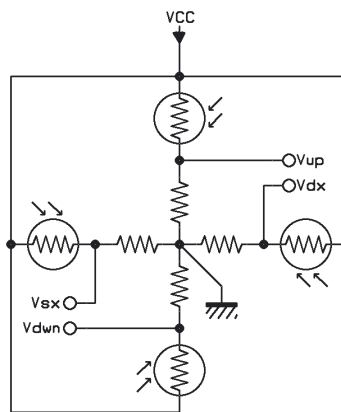
Tra le fonti energetiche rinnovabili, quella solare è certamente la più interessante e utilizzabile in pratica attraverso sistemi termodinamici e fotovoltaici. I pannelli fotovoltaici sono generatori di tensione e corrente capaci di ricavare elettricità dalla luce, in particolare da quella del sole. Ormai ne vediamo montati sui tetti delle case, su telai appoggiati al terreno, anche su piccole imbarcazioni, sempre orientati in modo da prendere la luce del sole durante tutta la giornata. Proprio riguardo al posizionamento, va detto che secondo la Legge di Lambert (una delle regole fondamentali dell'ottica...) l'ener-

gia ceduta da un raggio di luce alla superficie che investe è direttamente proporzionale al coseno dell'angolo formato tra la direzione della luce e la normale alla superficie stessa; quindi perché un pannello riceva la massima energia, ovvero lavori con la migliore efficienza possibile, bisogna che sia perpendicolare alla direzione dei raggi solari. Ecco perché per ottenere il massimo da un pannello o sistema di pannelli fotovoltaici bisogna che gli stessi seguano il sole, cioè che, nel corso della giornata, gli si rivolgano sempre in modo da risultare perpendicolari o quasi. Ciò viene ottenuto con quegli apparati elettronici

>

Fig. 1 - Rilevare la posizione del sole.

Quattro fotoresistenze sono disposte sui vertici di un rombo e si muovono insieme al pannello solare. Quelle in verticale rilevano l'elevazione del sole rispetto all'orizzonte mentre quelle in orizzontale ne rilevano lo spostamento destra-sinistra. Ogni fotoresistenza lavora come resistenza variabile di un partitore di tensione e produce una tensione che dipende dall'irraggiamento luminoso. Quando le quattro tensioni sono di uguale valore (o molto prossimo), il pannello solare è perpendicolare al sole. Quando una tensione varia (ad esempio $V_{up} > V_{dw}$), significa che il sole si è spostato rispetto al pannello, quindi il microcontrollore che controlla le tensioni corregge la posizione nella direzione della fotoresistenza a valore di tensione più alto.



ci noti come inseguitori solari, ossia dispositivi capaci di rilevare la direzione dalla quale il sole si percepisce con la massima intensità e comandare di conseguenza un attuatore con brandeggio, che permette di orientare i pannelli in quella direzione.

In queste pagine descriviamo proprio un inseguitore solare completo, seppure dimensionato per muovere pannelli leggeri; nello specifico, il brandeggio usato può orientare pannelli fotovoltaici di potenza fino a una cinquantina di watt e comunque pesanti non più di 5÷7 chilogrammi.

Come funziona

Per rilevare qual è la direzione privilegiata verso la quale orientare il pannello, il nostro inseguitore impiega un circuito a microcontrollore che esamina i segnali di quattro sensori ottici disposti come visibile in Fig. 1 e assemblati in maniera da rilevare la luce praticamente solo di fronte; il micro legge periodicamente i segnali forniti e considera direzione privilegiata quella corrispondente al sensore che trova più illuminato. Per essere certo di orientare il pannello dalla parte giusta, verifica sempre anche lo stato del sensore che si trova, nel quadrilatero, opposto a quello maggiormente illuminato.

Il pannello solare è montato su un brandeggio, cioè un movimento Pantilt in grado di farlo spostare in tutte le direzioni combinando i movimenti orizzontale e verticale, entro un'angolazione più che sufficiente a coprire lo spostamento del sole dall'alba al tramonto; le tensioni di comando vengono fornite dal circuito a microcontrollore sulla base delle risultanze dei sensori di luce.

Dato che il brandeggio impiegato funziona a corrente alternata, ma tutto l'inseguitore trae l'energia che gli occorre dalla batteria che il pannello fotovoltaico carica, per alimentarlo viene impiegato uno speciale convertitore DC-AC gestito direttamente dallo stesso microcontrollore.

Come è composto

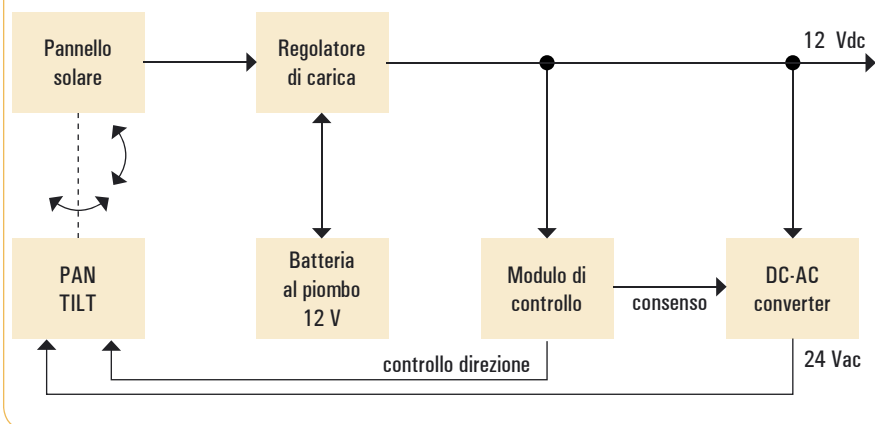
Lo schema a blocchi dell'inseguitore solare è riportato in Fig. 2: gli elementi fondamentali sono il modulo di controllo, che è quello contenente il microcontrollore e i sensori di luce e il convertitore DC-AC in PWM, contenente anch'esso un proprio microcontrollore e interfacciato direttamente con il modulo di controllo. Abbiamo, infine, il gruppo del pannello, costituito dal pannello stesso, dal regolatore di carica (di tipo shunt) con batteria e dal brandeggio.

Quest'ultimo si connette mediante cinque fili (quattro per il comando del movimento in una direzione e un comune) al modulo di controllo, dal quale prende l'alimentazione generata dal convertitore DC-AC. Modulo di controllo e inverter DC-AC possono essere montati a bordo del pannello solare (solidali con esso) ma, dal punto di vista pratico, è più conveniente fissare al pannello le sole fotoresistenze, lasciando tutti i circuiti elettronici e la batteria al riparo dalle intemperie (ad esempio nel sottotetto). Qui di seguito analizziamo dettagliatamente ogni unità, vedendone gli aspetti salienti.

Unità di controllo

Riportata in Fig. 3, essa è la parte più importante del sistema, perché presiede a tutte le funzioni; è interamente gestita da un microcontrollore Microchip PIC16F88, basato su un'architettura ad 8 bit, provvisto di Flash-EPROM organizzata in word da 14 bit e funzionante, nella nostra applicazione, con clock generato internamente. Il micro dispone, inoltre, di un A/D converter assegnabile a un massimo di otto linee di I/O. Durante l'inizializzazione, il programma imposta la linea RB3 come ingresso per la lettura del jumper J1 ed RA6, RA7, RB4, RB5, RB6 come uscite, destinate al comando dei relé cui è affidato il compito di comandare fisicamente il brandeggio; più esattamente, l'attivazione si ottiene quando la linea corrispondente viene portata a livello logico alto, allorché il relativo transistor NPN conduce e alimenta la bobina del relé che si trova collegato tra il suo collettore e il positivo di alimentazione dell'unità di controllo. I circuiti di gestione dei relé sono tutti uguali e ciascuno prevede un led che indica quale dei cinque è al momento attivo. Per quel che riguarda il significato delle uscite e quindi dei cinque relé, sappiate che solo quattro gestiscono effettivamente i comandi del brandeggio, ➤

Fig. 2 - Schema a blocchi. La tensione proveniente dal pannello viene utilizzata dal regolatore di carica sia per ricaricare la batteria che per alimentare l'inseguitore solare, il DC-AC converter e il carico applicato all'uscita 12Vdc. L'inseguitore controlla i movimenti pan-tilt del pannello e attiva il convertitore, che alimenta il comune del pan-tilt.



mentre uno viene usato per dare tensione al converter DC-AC, anche questo alimentato solamente quando occorre far compiere un movimento; per l'esattezza, il relé che comanda l'accensione del modulo DC-AC è RL5, gestito dalla linea RB4 del micro U1. Quanto agli altri quattro, RL2 è il comando dello spostamento orizzontale a sinistra, RL4 quello dello spostamento orizzontale a destra, RL1 comanda lo spostamento in alto e RL3 quello in basso; ogni relé chiude il circuito dell'alimentazione in alternata su un motore del brandeggio.

Proseguendo con l'analisi delle linee, vediamo che l'A/D viene assegnato ai piedini 1, 2, 17, 18 (rispettivamente AN2, AN3, AN0, AN1) per leggere lo stato delle fotoresistenze (come da Fig. 1) e a 3 (AN4) e 12 (AN5) per rilevare la tensione, rispettivamente, del pannello solare e del circuito di taratura delle soglie di rilevamento delle fotoresistenze.

Restano da esaminare ancora due I/O, che sono RB0 ed RB1: insieme costituiscono un I²C-bus impiegato dal PIC16F88 per comandare un visualizzatore a barra di led con il quale l'unità mostra, seppure approssimativamente, il livello della tensione erogata dal pannello fotovoltaico; si

tratta, evidentemente, della tensione letta dall'A/D converter attraverso i punti + e - SOLARE (linea AN4). La visualizzazione viene ottenuta usando un I/O expander siglato PCF8574, prodotto dalla Philips e da noi già utilizzato in molte altre applicazioni; non avremmo potuto fare altrimenti perché il microcontrollore non ha abbastanza linee di I/O per gestire direttamente anche la barra di led.

Il PCF8574 è bidirezionale, nel senso che dispone, per ciascuna delle otto linee P0÷P7, di uno stadio driver logico e di un buffer di lettura; può, quindi, inviare alle linee P0÷P7 le condizioni logiche previste dai dati in arrivo dalla linea SDA del bus I²C, ma anche trasmettere, sulla medesima SDA, gli stati letti dagli stessi piedini 4, 5, 6, 7, 9, 10, 11, 12. Il modo di funzionamento dipende dal comando ricevuto dall'elemento che impartisce le istruzioni, ossia, nel nostro caso, il microcontrollore PIC16F88. Siccome lo standard I²C-bus prevede che i dispositivi siano collegati in parallelo sulle linee SDA ed SCL e che le istruzioni dirette dal dispositivo di comando alle periferiche contengano un indirizzo che permetta di farle riconoscere alla sola periferica destina-

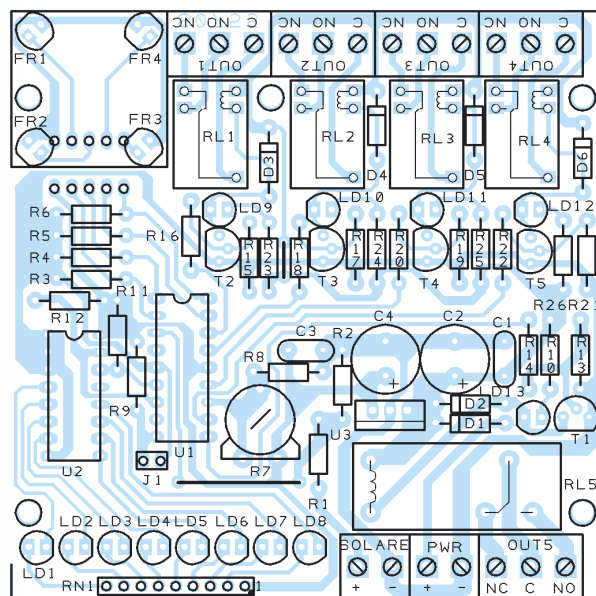
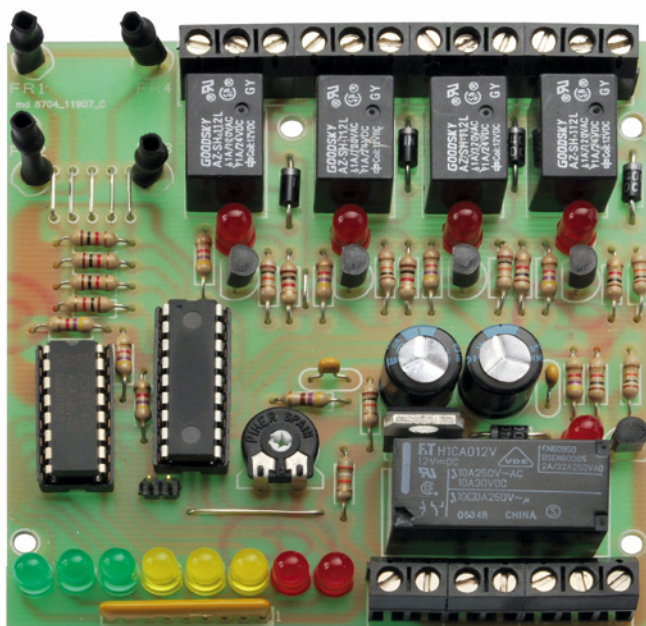
taria, ogni apparato I²-bus dispone di tre linee che permettono di definire l'address voluto, scegliendo tra otto possibili combinazioni. Nel caso del PCF8574, i pin corrispondenti sono 1, 2, 3 (rispettivamente bit di peso 1, 2, 4); guardando lo schema elettrico, notate che per U2 è stato imposto l'indirizzo 000 (0) ponendo tutti i pin di indirizzo a massa. Le uscite di U2 pilotano direttamente i led dato che la corrente assorbita da ciascuna sua uscita (modo sink) a livello basso può raggiungere i 25 mA.

Inizializzate le linee di I/O, il microcontrollore inizia a far girare il proprio main-program, che prevede il test ciclico dello stato delle fotoresistenze e delle linee RB3 e AN5; particolarmente interessante è la routine che riguarda la gestione dei dati ricavati dall'A/D converter riguardo, proprio, alle FR1, FR2, FR3, FR4. Come già detto, tali componenti sono disposti a quadrilatero e assemblati in modo da rilevare la luce solo frontalmente; opportunamente orientato verso l'alto e inclinato di circa 45°, ognuno dei fotoresistori rileva l'illuminazione in una precisa direzione. Elettricamente, FR1, FR2, FR3 ed FR4 risultano connesse come resistenze poste in cima ad altrettanti partitori resistivi, quindi all'aumentare della luce che le investe cresce la tensione (perché cala il valore resistivo assunto) applicata agli ingres-

Fig. 3 - L'unità di controllo montata. Essa rileva la posizione del sole e comanda di conseguenza sia il convertitore DC-AC che il brandeggio del pannello solare.



L'inseguire solare è di facile realizzazione. Tutti i componenti impiegati sono ad inserzione e piuttosto comuni. L'angolo di circuito destinato alle fotoresistenze può essere tagliato e collegato alla base tramite un cavetto a cinque conduttori, in modo da fissare sul pannello solare le sole fotoresistenze.



ELENCO COMPONENTI:

R1: 39 kohm
R2, R13, R15, R17, R19, R21: 10 kohm
R3÷R6, R10, R23÷R26: 1 kohm
R7: trimmer MO 10 Mohm
R8: 470 ohm
R9, R11, R12, R14, R16, R18, R20, R22: 4,7 kohm
RN1: rete resistiva 8x470 ohm + C
C1, C3: 100 nF multistrato
C2: 470 µF 25 VL elettrolitico

C4: 470 µF 16 VL elettrolitico
U1: PIC16F88 (MF755)
U2: PCF8574
U3: 7805
D1÷D6: 1N4007
LD1÷LD8: led 5 mm verde
LD9÷LD13: led 5 mm rosso
T1÷T5: BC547
FR1÷FR4: Fotoresistenza 2-20 kohm
RL1÷RL4: relé 12Vdc 1 scambio

RL5: relé 12Vdc 1 scambio 10A/250Vac

Varie:

- Zoccolo 8+8
- Zoccolo 9+9
- Morsettiera 2 poli (2 pz.)
- Morsettiera 3 poli (5 pz.)
- Strip maschio 2 vie
- Jumper
- Circuito stampato

si corrispondenti (AN3, AN2, AN0, AN1) del micro U1; viceversa, al buio (quando la resistenza aumenta) la tensione ottenuta dai singoli partitori è praticamente nulla. Per sapere dove va il sole e regolarsi su come orientare il pannello solare, il microcontrollore legge ciclicamente le quattro fotoresistenze ed estrae il valore più alto, il quale corrisponde, al componente più illuminato; noto qual è il fotoresistore che riceve più luce, il PIC ritiene che la direzione verso la quale girare il brandeggio e quindi il pannello sia quella corrispondente ad esso, quindi avvia la

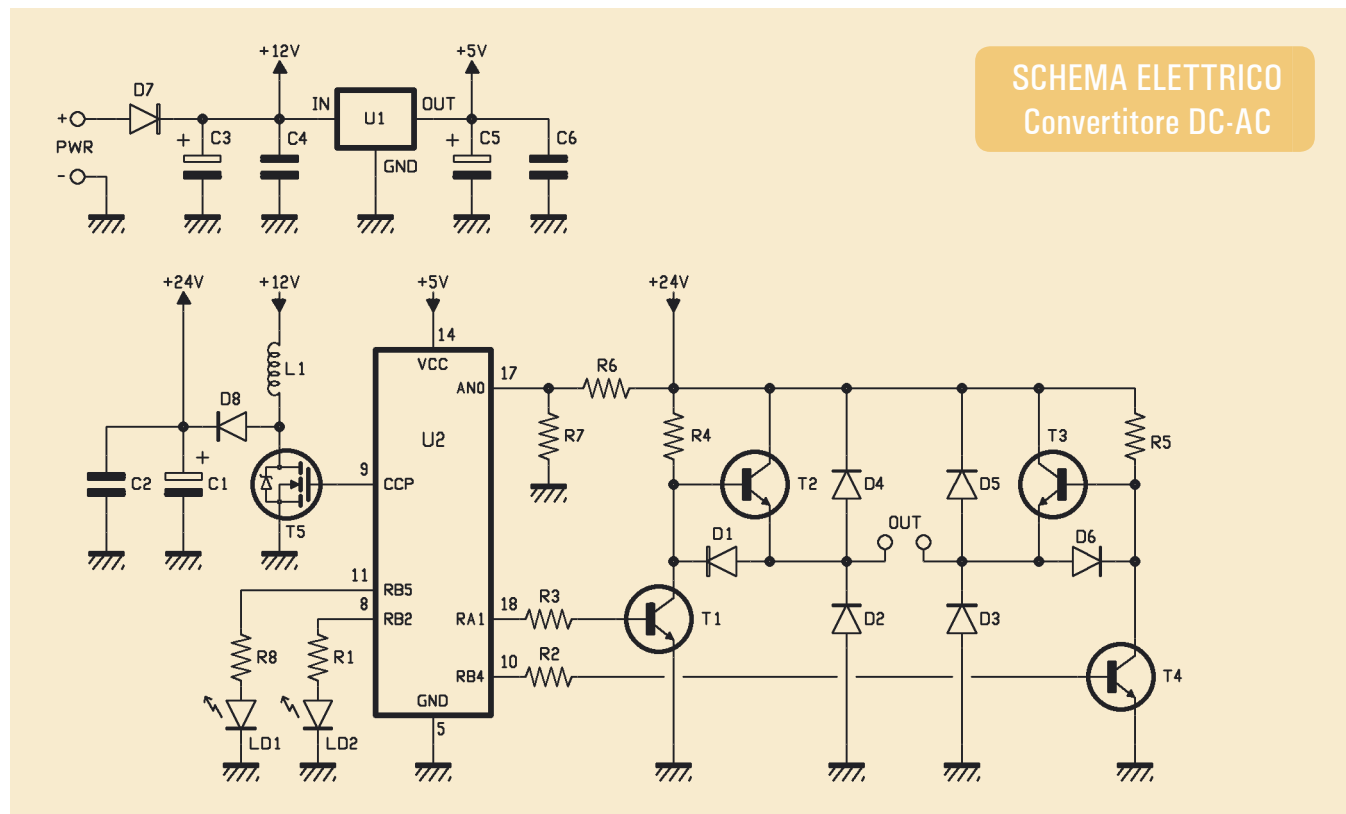
routine di movimento. Quest'ultima prevede il livello logico alto sulla linea RB4 e la conseguente eccitazione del relé RL5 per tutta la durata del movimento del brandeggio; attivato RL5 e alimentato quindi il DC-AC converter, il micro fa eccitare il relé corrispondente al comando di rotazione del pannello nella direzione voluta. Ma quando si ferma la rotazione? È chiaro che se il movimento tenesse conto solamente dell'illuminazione ricevuta, a un certo punto il pan-tilt continuerebbe ad oscillare per trovare la posizione di equilibrio, perché

qualsiasi minima variazione di illuminazione dell'ambiente o il rapido passaggio di nuvole porterebbe la fotoresistenza più illuminata a non esserlo più o a ritornare ad esserlo dopo pochissimo. Ma non solo: stando a bordo del pannello, le fotoresistenze si spostano nella direzione del sole, con il risultato che durante il movimento diminuisce la differenza di illuminamento tra quella rivolta nella direzione di spostamento del brandeggio e quella opposta, la quale viene illuminata sempre più. Allora il brandeggio tenderebbe a correggere l'errore riportando indie-

tro il pannello, poi muovendolo nuovamente nella direzione originaria, senza mai arrestarsi. Sorge quindi il problema di evitare questo “pendolarismo”, che non solo è praticamente inutile perché non porta apprezzabili incrementi di efficienza del sistema

messo in movimento quando la differenza di illuminazione supera una certa soglia (che si imposta mediante il trimmer R7) e si ferma solo se la differenza misurata tra le due fotoreistenze in diagonale scende sotto una soglia minore di quella che ha

5.500 lux ed FR4 appena 4.000 lux. Ora, il microcontrollore rileva che la fotoreistenza più illuminata è FR2 e va a guardare qual è la differenza rispetto all’opposta FR4; ma aziona il brandeggio e quindi ruota il pannello nella direzione in cui “guarda” FR2



fotovoltaico, ma spinge peraltro a consumare una parte non trascurabile dell’energia elettrica ricavata dal sole e accumulata nella batteria; infatti il brandeggio dovrebbe muoversi per brevi periodi e non continuamente.

Abbiamo risolto il problema imponendo una differenza minima fra i livelli di illuminazione (e quindi di tensione) tra due fotoreistenze poste in diagonale (FR1-FR3, FR2-FR4) sotto la quale il brandeggio non viene attivato. Ma non solo; affinché il sistema risulti stabile, il programma di gestione prevede che vi sia una certa distanza nella differenza di illuminazione fra fotoreistenze opposte che porta al movimento del brandeggio e quella che ne determina l’arresto. Nella pratica, il pannello viene

portato al movimento. Così si ottiene uno spostamento discontinuo ma stabile.

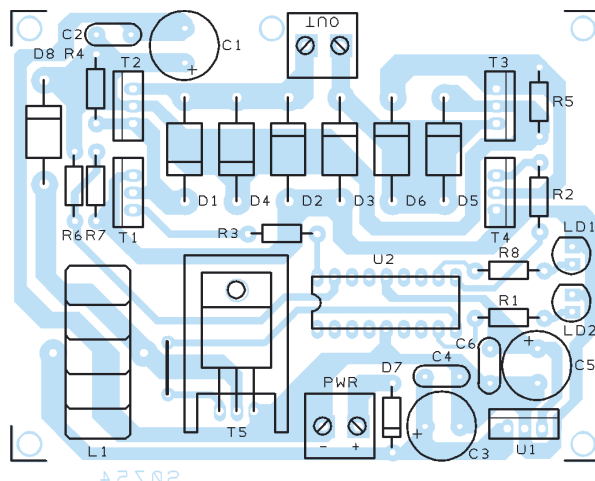
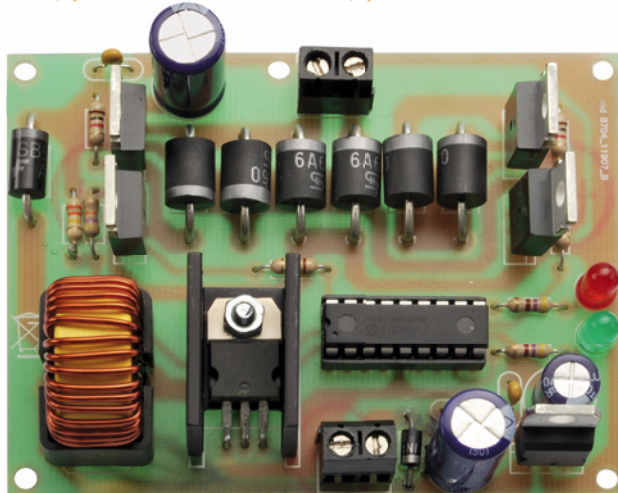
Facciamo un esempio pratico, supponendo di avere il sistema a riposo, ovvero in condizioni di equilibrio, con tutte le fotoreistenze esposte a un’illuminazione di 5.000 lux; a un certo punto il sole si sposta: FR1 ed FR3 (che supponiamo essere quelle che si trovano sulla diagonale che fa da asse al movimento del sole...) calano di luminosità in maniera abbastanza simile, mentre quella che guarda verso la direzione del sole (imponiamo sia FR2) cresce sensibilmente rispetto alla quarta (FR4) la quale, essendo opposta ad essa, risente molto più delle altre del calo di illuminazione. Diciamo pure che FR1 ed FR3 ricevono 4.500 lux, FR2

solamente se la differenza letta supera il valore definito con R7. Quindi, se la regolazione è tale che la distanza minima tra i valori di illuminamento risulti minore di 1.500 lux, il microcontrollore comanda il brandeggio, altrimenti nulla si muove.

Se la soglia è minore di 1.500 lux, il pan-tilt comincia a orientare il pannello fotovoltaico; il microcontrollore lo ferma, però, solamente quando si registra una differenza più bassa: ad esempio 1.200 lux. Così facendo si è certi di fermare il brandeggio per un tempo ragionevole ed evitare che il pannello continui a muoversi nell’intorno di una posizione.

A riguardo bisogna osservare che più alta si imposta la distanza tra i valori di illuminamento delle fotoreistenze (opposte in diagonale) che determina

Come l'inseguitore, anche il convertitore fa uso di componenti ad inserzione di uso comune ed è molto facile da realizzare. L'unico componente cui occorre prestare un minimo di attenzione è la bobina L1 che, per via delle dimensioni, potrebbe dover essere fissata allo stampato con della colla a caldo.



ELENCO COMPONENTI:

R1, R8: 470 ohm
R2+R5: 1 kohm
R6: 330 kohm
R7: 47 kohm
C1, C3: 470 µF 35 VL elettrolitico
C2: 100 nF multistrato
C4: 100 nF multistrato
C5: 470 µF 16 VL elettrolitico
C6: 100 nF multistrato

U1: 7805
U2: PIC16F88
LD1: led 5 mm rosso
LD2: led 5 mm verde
T1+T4: TIP122
T5: BUZ11
D1+D6: 6A60
D7: 1N4007
D8: SB540

L1: Bobina 68 µH

Varie:

- Zoccolo 9+9
- Morsettiera 2 poli
- Dissipatore (ML26)
- Vite 10 mm 3 MA
- Dado 3 MA
- Circuito stampato

il movimento e quella che fa fermare il tutto, più stabile è il movimento del brandeggio; per contro, una soglia elevata comporta una perdita di efficienza, dovuta al fatto che il pannello fotovoltaico viene orientato con maggiore ritardo rispetto a quanto accadrebbe se la distanza fosse più ridotta.

Notate che il programma che gira nel PIC16F88 prevede che la frequenza di lettura dei valori di tensione dati dalle fotoresistenze sia legata all'entità della distanza tra i valori di illuminamento di fotoresistenze opposte (soglia). Questo perché altrimenti è inutile imporre basse soglie di variazione dell'illuminamento se poi non si permette al sistema di seguire più rapidamente le variazioni.

Chiarito questo, riprendiamo il di-

scorso sul movimento del brandeggio: come già detto, quando il microcontrollore rileva il superamento della distanza prevista tra due soglie attiva RL5 e il relé corrispondente alla direzione nella quale far orientare il pannello solare; il movimento viene sospeso quando la differenza di illuminazione scende sotto il valore minimo.

Ed ora bisogna soffermarsi su un particolare: sebbene scelte di uguale modello, le fotoresistenze, come tutti i componenti elettronici, presentano differenze dipendenti dalla loro tolleranza e dalla deriva termica; questo vuol dire che, a parità di illuminamento, ognuna presenterà un valore resistivo diverso da quello delle altre e determinerà, all'ingresso dell'A/D converter, un differente livello di ten-

sione. Ciò non giova alla precisione del sistema, ragion per cui abbiamo pensato a come compensare questo errore; lo abbiamo fatto prevedendo una procedura di calibrazione, che permette al microcontrollore di definire quali sono le letture di ciascun fotoreistore a parità di livello di illuminazione.

La procedura si avvia chiudendo il ponticello J1, allorché il programma principale si arresta e parte l'apposita routine durante la quale l'A/D converter legge in sequenza i valori di tensione dovuti alle quattro fotoresistenze e ne memorizza i dati. Affinché vada a buon fine, FR1, FR2, FR3 e FR4 devono risultare illuminate alla stessa maniera.

Prima di concludere la descrizione dell'unità di controllo, un appunto ➤

Fig. 4 - Il convertitore DC-AC montato. Trasforma i 12 Vdc provenienti dal regolatore di carica in 24 Vac (a onda quadra) necessari per il funzionamento del brandeggio.



sull'indicatore a barra di led; il pannello fotovoltaico va collegato sia all'ingresso del regolatore di carica che all'input della scheda di controllo. Essendo il regolatore di tipo shunt, quando la batteria è carica, esso cortocircuita il pannello, con il risultato di far apparire nulla la tensione che questo fornisce. Può quindi capitare che il voltmetro a barra di led segni zero anche se la batteria è carica; non ve ne dovete preoccupare.

Converter DC-AC

Come accennato, impiegando sia un pannello fotovoltaico che una batteria a 12 volt e avendo previsto un brandeggio a 24 Vac, bisogna che il nostro sistema ricavi quest'ultima tensione. Per riuscire nell'intento si avvale di un inverter (Fig. 4) composto da un elevatore di tensione stabilizzato che ottiene i 24 V partendo dai 12 volt dell'accumulatore e da un driver a ponte; il tutto è realizzato mediante il microcontrollore U2 (un altro PIC16F88) opportunamente programmato. Dopo l'inizializzazione degli I/O, il micro genera una tensione rettangolare con la quale pilota il mosfet T5; quest'ultimo realizza, insieme alla bobina L1 e al diodo D8, un DC-DC converter a carica d'induttanza, dal quale si ottengono 24 volt. Il funzionamento del DC-DC è

il seguente: quando il mosfet riceve dal microcontrollore gli impulsi a livello logico alto va in stato di On e chiude praticamente a massa l'induttanza L1; nelle pause, T5 si interdice e L1 deve scaricarsi in qualche modo. Cerca di farlo generando una tensione di polarità inversa e ampiezza più grande rispetto a quella con la quale il mosfet l'aveva alimentata; ne deriva un impulso che attraversa D8 e carica C1 e C2, cedendo l'energia immagazzinata nel periodo di conduzione del T5.

Dunque, in corrispondenza dell'1 logico sul gate, il mosfet conduce e carica l'induttore; nelle pause L1 carica i condensatori C1 e C2, i quali durante la conduzione del transistor non possono scaricarsi su di esso perché D1 lo impedisce. Ai loro capi si crea così una tensione continua che il microcontrollore stabilizza a 24 volt mediante una routine che gestisce la larghezza degli impulsi di comando del mosfet sulla base di quanto l'A/D converter legge dalla linea AN0 (pin 17); più esattamente, all'inizio il microcontrollore dà sul piedino 9 una componente rettangolare con un certo duty-cycle, quindi rileva qual è la tensione presente ai capi di C1 e C2. Se il valore eccede i 24 volt, riduce il duty-cycle, mentre se (a causa del carico determinato dallo sforzo com-

Fig. 5 - Il brandeggio (pan-tilt) si presenta come in figura. Va scelto di potenza adeguata alle dimensioni del pannello solare.



piuto dal brandeggio) rileva meno di 24 V, aumenta la larghezza degli impulsi fino ad ottenere quanto voluto. Il software del micro realizza quindi un converter DC-DC step-up in PWM, stabilizzato nei limiti del possibile e protetto dal sovraccarico; la protezione è ottenuta imponendo un limite alla larghezza degli impulsi dati al mosfet. Senza, nel caso di un eccessivo assorbimento o un cortocircuito nel brandeggio non si raggiungessero i 24 volt, il micro arriverebbe, teoricamente, ad alzare la larghezza degli impulsi fino al punto che l'assorbimento e il riscaldamento del mosfet diverrebbero eccessivi e porterebbero al danneggiamento. Fin qui abbiamo visto come si ricavano i 24 Vdc; ma ai motori del

Fig. 6 - Il circuito di controllo e il convertitore DC-AC sono stati realizzati in modo tale da poter essere fissati uno sopra l'altro per mezzo di quattro torrette esagonali lunghe 30 mm. In questo modo è possibile inserire il circuito completo ad esempio in un contenitore a tenuta stagna. Il circuito di controllo, che dispone di un trimmer di taratura, nonché del voltmetro, va posto sopra il convertitore.

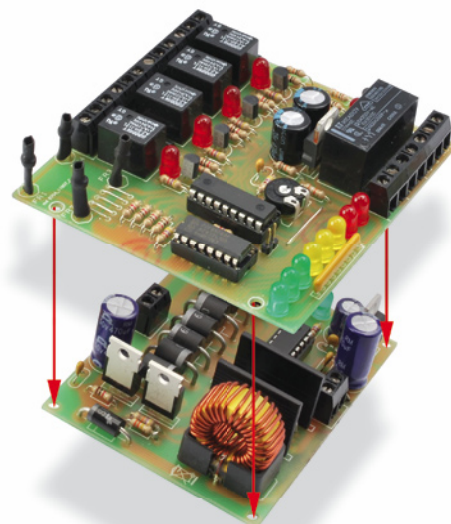


Fig. 7 - Il nostro prototipo è stato realizzato con un pannello da 24 watt massimi (17,4 Vp, 1,38 A massimi).



brandeggio serve corrente alternata. Dunque, questa da dove nasce? La risposta è semplice: la otteniamo mediante quattro transistor bipolari di potenza connessi a bridge, alimentati dalla tensione ricavata dal DC-DC e pilotati da una coppia di onde quadre (a bassa frequenza) in opposizione di fase tra loro generate dallo stesso PIC16F88. Quando la linea RA1 è a livello alto (ed RB4 è, di conseguenza, a zero logico) T1 è in saturazione e mantiene interdetto T2, quindi D1 trascina a livello basso il lato sinistro dell'uscita (OUT) del ponte; contemporaneamente T4 risulta interdetto ed R5 polarizza la base del T3 determinando la saturazione del T3, il cui emettitore porta a livello alto il lato destro dell'uscita del bridge.

Nel semiperiodo seguente, gli stati logici si invertono: RA1 si porta a livello basso cosicché T1 va all'interdizione, mentre T2 satura (perché ora R4 ne può polarizzare la base) e pone a livello alto il lato sinistro dell'uscita del ponte; RB4 si trova a 5 volt e T4 viene portato in saturazione, tanto che il suo collettore trascina a circa zero V la base del T3 (che s'interdice) e il catodo del diodo D6, il cui anodo pone a zero il lato destro dell'uscita del ponte. Il ciclo si ripete fin quando il microcontrollore riceve l'alimentazione dal regolatore lineare U1 (un comune 7805).

Ciò si verifica quando l'unità di controllo dà i 12 volt della batteria tramite RL5, ovvero nei periodi in cui il brandeggio deve essere azionato.

I led pilotati dalle linee RB2 ed RB5 indicano la presenza delle tensioni di lavoro. La componente alternata ricavata dal ponte alimenta direttamente il comune del Pan-Tilt tramite la morsettiera OUT, o meglio con un punto va su COM del brandeggio e con l'altro si fa i comuni dei relé RL1÷RL4 del modulo di controllo.

Pan-Tilt

Il brandeggio VISP2 scelto per il nostro inseguitore solare è del tipo da esterno (come quello di Fig. 5), funzionante a 24 Vac e comanda-

bile mediante cinque fili, che sono poi il comune dell'alimentazione e i terminali di azionamento dei motori per gli spostamenti in su, giù, destra e sinistra. Può reggere e muovere un oggetto del peso di alcuni chili e avente una superficie non superiore a 0,5 mq; oltre, un'eventuale raffica di vento potrebbe impedirgli di muovere correttamente il pannello solare. Va collegato come mostrato nello schema di cablaggio, con il comune ad un capo dell'uscita dell'inverter DC-AC e con gli altri quattro, ordinatamente, ai contatti normalmente aperti (NO) dei relé RL1÷RL4 dell'unità di controllo; l'altro capo della morsettiera OUT dell'inverter va a collegarsi ai C dei predetti relé, così da chiudere il circuito di alimentazione.

Pannello solare e regolatore

Per la nostra applicazione abbiamo impiegato un pannello fotovoltaico da 24 watt, delle dimensioni di circa 57x41 centimetri (Fig. 7), capace di dare in pieno sole una tensione dell'ordine dei 17,4 volt e una massima corrente di circa 1,38 ampere; lo colleghiamo direttamente alla morsettiera SOLARE del modulo di controllo, così da permettere al microcontrollore di quest'ultimo di leggerne e monitorarne la tensione erogata. In parallelo connettiamo il regolatore di carica (SOL4UCN2, cui abbiamo dedicato un riquadro specifico) della batteria al piombo a 12 volt (prevediamo di usarne una da 30 A/h, anche se nei test di laboratorio ne abbiamo usata una più piccola, da 4 A/h). Si tratta di un regolatore di tipo shunt che, quando l'accumulatore è totalmente scarico, lascia che tutta la corrente del pannello lo raggiunga, mentre a carica raggiunta comincia a pulsare cortocircuitando periodicamente e con frequenza sempre maggiore i morsetti + e - del pannello stesso. Insomma, il regolatore prende su di sé la corrente data

Il regolatore SOL4UCN2

Quello utilizzato nel nostro sistema per tenere in carica la batteria, è un regolatore shunt commercializzato dalla ditta Futura Elettronica e pensato per essere abbinato a pannelli solari che danno fino a 24 volt e accumulatori da 12 V di capacità compresa tra 4 e 40 A/h; è inserito in un apposito contenitore provvisto di morsettiera di entrata (pannello) e uscita (batteria e carico). Dispone, inoltre, di un interruttore di esclusione del carico e di tre uscite (sempre attive) in tensione stabilizzata. Le caratteristiche tecniche sono le seguenti:

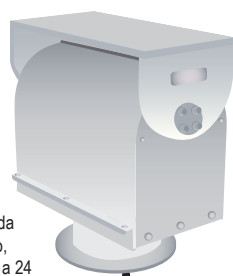
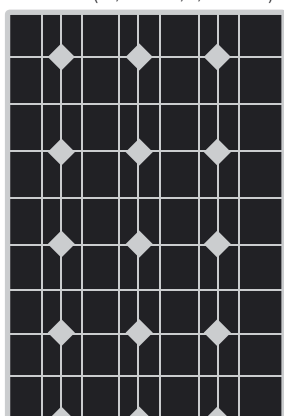


Tensione di ingresso (pannello)	max. 24 Vdc
Tensione e corrente in uscita al carico	12 V, max. 4 A
Tensioni di uscita ausiliarie	3, 6, 12 Vdc
Tensione di carica (batteria)	12÷14 V
Temperatura di esercizio	-10÷+42 °C

Piano di cablaggio

Materiale posto all'esterno

Pannello solare: quello del prototipo è un 24 W (17,4 V max., 1,38 A max.).



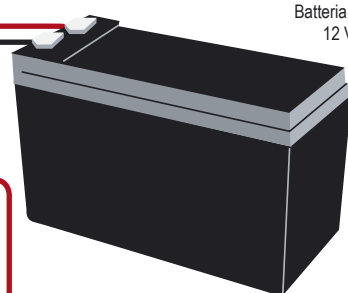
Pan-tilt da esterno, alimentato a 24 Vac.

Connettore del cavo di controllo. Il nostro pan-tilt dispone di un Din a sei poli + schermo. Attenzione: le dimensioni del connettore sono fuori scala per rendere più comprensibile la posizione dei pin.

- 1 Comune
- 5 Destra
- 3 Giù
- 6 Sinistra
- 4 Su
- 2 Nc.

Circuito di controllo, alimentato con l'uscita stabilizzata a 12 V, proveniente dal regolatore.

Materiale posto in un luogo riparato dalle intemperie



Batteria al piombo, 12 V 4 A/h

Positivo

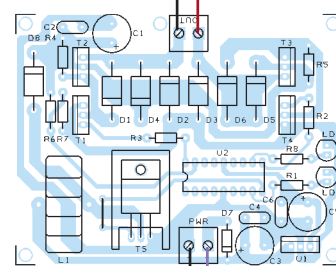
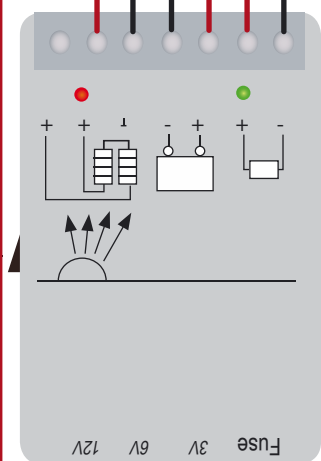
Negativo

Connettori per il collegamento del carico esterno, alimentato a 12 V 4 A. Carichi che necessitano di corrente maggiore vanno collegati in parallelo alla batteria.

ON = Sistema acceso

OFF = Solo ricarica batterie

Regolatore di carica per pannelli solari, modello SOL4UCN2. Dispone di un'uscita diretta al carico (stabilizzata a 12 V, 4 A massimi), uscita che viene interrotta quando la tensione di batteria scende al di sotto dei 10,5 V. Sono presenti, inoltre, tre uscite stabilizzate a 3, 6 e 12 Vdc (quest'ultima alimenta il circuito elettronico) dove la tensione è sempre presente. È presente un interruttore d'accensione.



Convertitore DC-AC: condivide l'alimentazione stabilizzata del circuito di controllo.

dal pannello fotovoltaico che eccede il fabbisogno della batteria. Il SOL4UCN2 è un regolatore in grado di accettare all'ingresso tensioni fino a 24 V e caricare batterie da 12 volt.

Realizzazione pratica

Bene, giunti a questo punto possiamo vedere come costruire in pratica l'inseguitore solare; procurati tutti gli elementi acquistabili già pronti (pannello fotovoltaico, regolatore di

carica, batteria e brandeggio) non resta che realizzare il modulo di controllo e il converter DC-AC, per i quali è previsto un circuito stampato ciascuno, realizzabile per fotoincisione dopo aver scaricato dal nostro sito Web (www.elettronica.in.it) le tracce lato rame corrispondenti. Incise e forate le basette, disponetevi i componenti in ordine di altezza, dando la precedenza alle resistenze e ai diodi e prestando attenzione a ri-

spettare l'orientamento di quelli polarizzati indicato negli appositi piani di montaggio. I microcontrollori e il PCF8574 è bene siano sistemati su zoccoli, solo dopo aver fatto tutte le saldature.

Particolare attenzione va posta nel montare le fotoresistenze, ciascuna delle quali va introdotta in un tubetto di guaina termorestringente nera lungo circa 2 cm, in modo che la superficie sensibile rientri grosso-

modo un centimetro rispetto al bordo superiore; se possibile, ogni componente deve poi essere chiuso posteriormente con dello stucco o silicone bianco. Questa realizzazione serve a sensibilizzare le fotoresistenze esclusivamente alla luce che le colpisce frontalmente, ovvero a renderle direttive; così ognuna rileverà lo stato di illuminazione in una sola direzione.

scopo prendete un alimentatore (meglio se stabilizzato) capace di dare 12 Vcc (oppure una batteria carica da 12 volt...) e collegatene + e - rispettivamente a + e - della morsettiera PWR. Fatto ciò, orientate provvisoriamente tutte le fotoresistenze (già chiuse nei loro tubetti termorestringenti) dalla stessa parte e illuminatele con un faro; chiudete J1 e attendere che tutti

fare) consiste nel lasciare all'esterno il pannello, il brandeggio ed il solo gruppo di fotoresistenze (la parte di c.s. su cui queste sono montate può essere facilmente rimossa dal resto del circuito e collegata via cavo schermato), mentre i circuiti elettronici, il regolatore di carica e la batteria verranno posti al riparo (ad esempio nel sottotetto) dalle intemperie ed

Il supporto del pannello

Il modo in cui il pannello solare viene fissato al brandeggio dipende sia dalla forma che dai fori di fissaggio di entrambi. Non potendo sapere a priori che tipo di pannello e brandeggio utilizzerete, nemmeno ci è possibile fornire istruzioni dettagliate su come effettuare il fissaggio meccanico. Possiamo, però, dirvi come abbiamo realizzato noi la staffa di fissaggio: abbiamo acquistato presso un negozio di "fai da te" un angolare in alluminio da 40x20 mm e abbiamo tagliato tre pezzi di lunghezza adeguata per poter assemblare (con delle viti e relativo dado) la struttura a "T" visibile nella Fig. B. Un quarto pezzo è stato utilizzato per fissare il contenitore dei circuiti elettronici (sia il solo gruppo fotoresistenze sia l'insieme inseguitore e inverter) a 45° sul pannello, in modo da poter effettuare le varie calibrazioni. A lavoro ultimato, solo le fotoresistenze devono essere fissate meccanicamente (opportunamente protette dalle intemperie) al pannello: i circuiti elettronici vanno posizionati in un luogo riparato come un contenitore a tenuta stagna oppure il sottotetto. In linea di massima, qualsiasi pannello solare può essere fissato al brandeggio per mezzo di una struttura ad "H" di dimensioni adeguate, come quella schematizzata nella Fig. A. Scegliendo gli angolari in modo opportuno, si riesce ad ottenere una staffa spessa come il pannello solare o poco più (Fig. C).

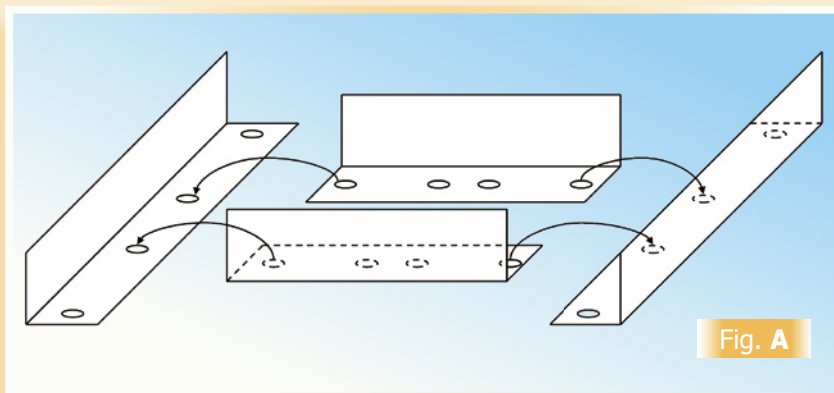


Fig. A

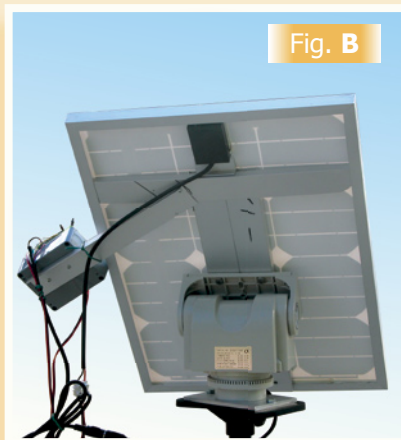


Fig. B



Fig. C

ne e l'inseguitore risulterà molto preciso. Per le connessioni degli scambi dei relé, l'alimentazione principale e l'ingresso SOLARE dell'unità di controllo, prevedete morsettiera a passo 5 mm da stampato; idem per PWR e OUT del modulo inverter DC-AC.

Completati i circuiti, prima di effettuare il cablaggio conviene tarare, ossia calibrare il gruppo di fotoresistenze dell'unità di controllo; allo

i led della barra si illuminino per due secondi. A questo punto il microcontrollore si è calibrato; potete rimuovere J1 e la lampada, quindi inclinare a 45° verso l'esterno le fotoresistenze come previsto (Fig. 8).

Per quanto riguarda il collegamento elettrico delle varie unità, potete fare riferimento al piano di cablaggio riportato in queste pagine. Relativamente all'assemblaggio meccanico, la cosa più conveniente (e facile da

inseriti in un contenitore adeguato.

Innanzitutto procuratevi un contenitore plastico con coperchio trasparente per impianti elettrici e inseritevi l'inverter e il modulo di controllo, montati come visibile in Fig. 6; per il fissaggio dei due circuiti al contenitore usate delle colonnine esagonali di lunghezza adeguata.

Praticate dei fori sui fianchi del contenitore in modo da far fuoriuscire, tramite dei passacavi, >

Non ce ne voglia Galileo...



...se in queste pagine abbiamo più volte detto che dall'alba al tramonto il sole si sposta: sappiamo benissimo (e speriamo lo sappiate anche voi...dato che a scuola lo insegnano...) che a spostarsi è la Terra, la quale ruotando su sè stessa ci fa vedere il sole in diverse posizioni dell'emisfero celeste. Abbiamo però preferito parlare di spostamento del sole perché più semplice, sia da esplicitare che da comprendere, dato che, meccanica celeste a parte, a noi comuni mortali appare che il sole si sposti nel cielo.

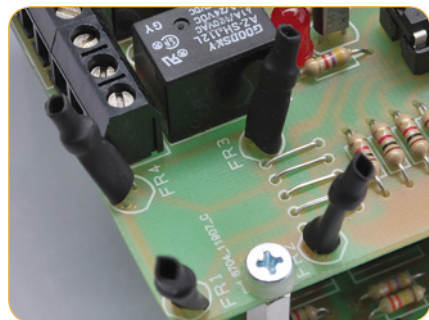
Astronomi e fisici si tranquillizzino, dunque: non siamo cultori della teoria tolemaica; come tutti gli studiosi moderni, riteniamo che Galileo Galilei abbia già subito abbastanza oltraggio dall'abiura cui il Sant'Uffizio lo costrinse nel lontano 1633...

come dovrete procedere per l'assemblaggio. In un riquadro specifico, però, abbiamo riportato alcune linee guida (seguite da noi durante la realizzazione del nostro prototipo) che vi saranno certamente utili. Scegliete, infine, il supporto del brandeggio, che può essere un palo, un pilastro o un braccio metallico, quindi fissatevi la base.

Collegato il tutto, vedrete il pannello solare spostarsi verso il sole; ricordatevi la posizione e andate a controllare dopo una decina o una ventina di minuti se è ancora orientato allo stesso modo. Se vi sembra che si adegui troppo lentamente, nel senso che rimane troppo tempo fermo in una posizione, agite sul cursore dell'R7 fino ad ottenere la periodicità di movimento che vi sembra più appropriata.

- i cavi di controllo del brandeggio,
 - il cavo bipolare del pannello,
 - il cavetto delle fotoresistenze.
- Queste ultime vanno montate in modo

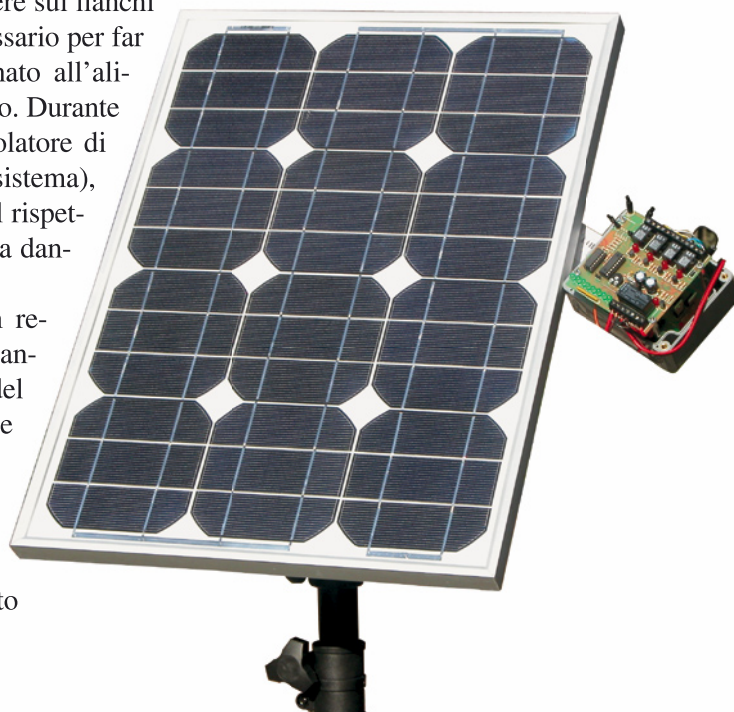
Fig. 8 - Le fotoresistenze sono state rese direttive con del termoretraibile nero e montate inclinate di circa $45^\circ \div 60^\circ$ rispetto allo stampato.



solidale al pannello solare e protette a loro volta dagli agenti atmosferici tramite un secondo contenitore tra-

sparente a tenuta stagna (IP68). Se lo desiderate (e se avete abbastanza spazio) all'interno del primo contenitore potete sistemare anche il regolatore di carica e la batteria, ma in questo caso dovrete prevedere sui fianchi un quarto foro, necessario per far uscire il cavo destinato all'alimentazione del carico. Durante il cablaggio del regolatore di carica (e di tutto il sistema), prestate attenzione al rispetto della polarità, pena danneggiamenti vari.

A questo punto non resta che fissare il pannello alla piastra del brandeggio: dato che il montaggio dipende sia dal pannello che dal brandeggio, non possiamo dirvi in modo assoluto



per il

MATERIALE

Tutti i componenti utilizzati in questo progetto sono facilmente reperibili in commercio. Il master dei circuiti stampati ed il firmware sono scaricabili dal sito della rivista (www.elettronica.in.it). Il microcontrollore che sovraintende tutte le funzioni è disponibile anche già programmato (Cod. MF755, 15,00 Euro). Il pannello solare monocristallino da 24 watt (cod. STP24M) costa 178,00 Euro, IVA inclusa.

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

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

Telecomando IR per sistema Velbus

di Arsenio Spadoni



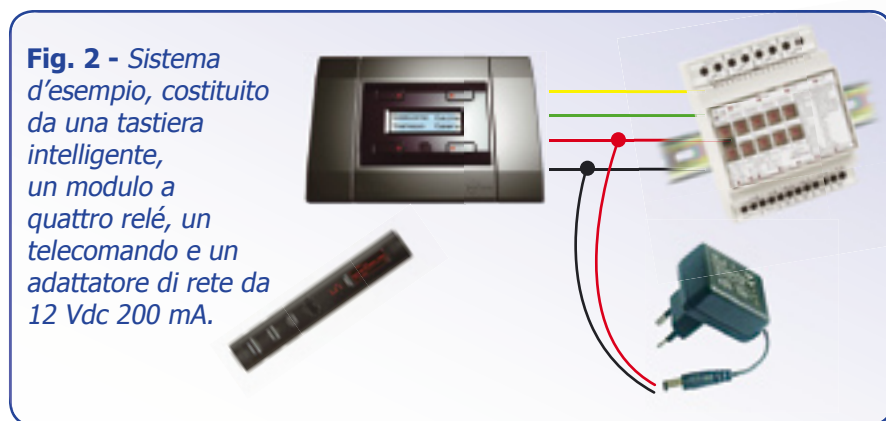
Progettato per operare con la tastiera intelligente VMB4PD, comanda fino ad otto carichi differenti, collegati ad altrettanti canali, per mezzo di un solo pulsante. Il telecomando

dispone di un sensore di inclinazione che, a dispositivo posto in verticale, permette di selezionare il canale. In orizzontale e orientato verso la tastiera, la pressione del pulsante genera l'invio del comando al canale selezionato, quest'ultimo mostrato su un display a led a sette segmenti.

La domotica sta gradualmente prendendo piede nella vita quotidiana; basti pensare ai vari automatismi, proposti da aziende specializzate nel settore e pubblicizzati su riviste e network televisivi, progettati per comandare l'apertura e la chiusura di tende e tapparelle, di cancelli, di porte scorrevoli e molto altro ancora. Tutti questi sistemi permettono di gestire più facilmente gli ambienti in cui si vive o si lavora, nonché di interagire in modo più efficiente ed immediato con essi. Di fatto, le applicazioni domotiche consentono di migliorare la qualità della vita e, sulla scia dell'interesse che questa disciplina reca con

sé, le più importanti aziende nel settore illuminotecnico, quali Bticino, Gewiss, Vimar e altre ancora, hanno realizzato e commercializzato le proprie soluzioni. Anche *Elettronica In*, nel corso delle proprie pubblicazioni, ha dedicato ampio spazio a questo argomento, focalizzando l'attenzione sul sistema domotico Velbus di Velleman e dedicando ad esso vari articoli e progetti.

Bene, in queste pagine vi presentiamo il VMB1RTS, un telecomando a otto canali progettato per impartire comandi ad una tastiera intelligente modello VMB4PD, cui abbiamo dedicato sia un riquadro a parte in queste pagine, ➤



sia l'articolo pubblicato sul fascicolo numero 123.

Proseguendo nella lettura, vi mostriamo come utilizzare il controllo remoto, applicato in un sistema molto semplice in cui un modulo a relé a quattro canali VMB4RY viene comandato in modalità bistabile (nota anche come "toggle" oppure On/Off) sia con la tastiera che con il telecomando, oggetto, quest'ultimo, che sarà argomento del prossimo paragrafo.

Il telecomando

Come visibile in Fig. 1b, il dispositivo è simile a un grosso pennarello: esso dispone di un'astina (che, all'occorrenza, può essere rimossa) per l'inserimento nel taschino di una camicia, di un unico pulsante da usare sia per selezionare il canale che per inviare il comando e, infine di

un display a led a sette segmenti, che segnala il canale selezionato, l'invio del comando in corso, più altre informazioni relative all'impostazione dei canali e dell'indirizzo; ogni dispositivo Velbus (fatta eccezione per le interfacce seriale e USB) infatti, per poter funzionare correttamente, necessita di un proprio indirizzo che, nel caso del telecomando, deve corrispondere a quello della tastiera cui è assegnato. Sul lato posteriore sono presenti due viti: una volta rimosse, è possibile aprire il dispositivo, che si presenta pertanto come visibile in Fig. 1a.

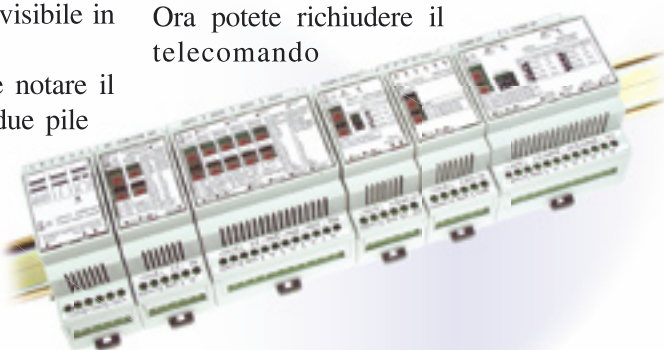
Partendo dal basso, potete notare il vano batterie, che ospita due pile ministilo (AAA). La parte superiore del contenitore, invece, è occupata dal circuito stampato su cui, sempre partendo dal bas-

so, trovate il pulsante di comando, il display a led, il microcontrollore programmato e, raggruppati nella parte superiore del c.s., il pulsante di programmazione dell'indirizzo (in alto a sinistra), il sensore di inclinazione e movimento (in mezzo) e il diodo all'infrarosso (a destra). Inserendo le batterie, il display visualizza in sequenza i numeri da 0 a 8, segnalando così il corretto funzionamento.

Prima di richiudere il contenitore, è necessario impostare l'indirizzo del telecomando, che corrisponde alla cifra meno significativa di quella della tastiera che dovrà comandare: se, ad esempio, l'indirizzo della tastiera fosse 25 (espresso in formato esadecimale), l'indirizzo del telecomando sarebbe 5. Analogamente il telecomando che comunica con la tastiera avente indirizzo 1C deve avere indirizzo C.

Poiché nel sistema d'esempio descritto in queste pagine facciamo uso di due soli moduli Velbus, abbiamo assegnato gli indirizzi 01 alla tastiera e 02 al controllo a relé, pertanto il telecomando deve avere indirizzo 1. Per impostare quest'ultimo, mantenete premuto il pulsante in alto (SW2-Setup): il microcontrollore accende il punto decimale del display (comunicando che la procedura per l'impostazione dell'indirizzo è attiva) ed inizia a visualizzare in sequenza i numeri da 0 a 9 e le lettere da A ad F. Rilasciate il pulsante non appena compare la cifra corrispondente all'indirizzo da impostare (in questo caso 1). Cifra e punto decimale lampeggiano tre volte, confermando la memorizzazione dell'indirizzo e l'uscita dalla procedura.

Ora potete richiudere il telecomando



e verificare che, a telecomando posto in verticale, la pressione del pulsante plastico determini il cambio del canale (visualizzato dai numeri 1, 2, 3 e 4). Ponete il telecomando in posizione orizzontale e verificate che, alla pressione del pulsante, si accendano

stiera che sul modulo a relé) sia innestato: tale jumper, infatti, inserisce una resistenza di terminazione da 120 ohm in parallelo al bus di sistema, un RS485 che, come da specifiche, deve presentare un'impedenza di 120 ohm tra i suoi due fili. Alimentate quindi il

due cifre. Usate i due pulsanti in alto per selezionare la cifra dell'indirizzo che dovete modificare, identificata dal cursore lampeggiante. Mantenete premuto, ora, il pulsante in basso a sinistra e agite ripetutamente su quello in basso a destra per modificare il

La tastiera VMB4PD

La tastiera intelligente VMB4PD è, in pratica, un pannello di controllo (con cinque pulsanti e un display) programmabile per il sistema domotico Velbus. La funzione di ogni singolo tasto è visualizzata sul display. Abbinato ai moduli relé (VMB1RY, VMB4RY), ai moduli dimmer (VMB1DM) e ai controlli tapparelle (VMB1BL, VMB2BL), permette di realizzare un sistema di automazione domestica facilmente espandibile in funzione delle necessità crescenti. La placca e la scatola da incasso sono forniti a parte.

Caratteristiche:

- Modulo pulsanti con display LCD (2 linee, 16 caratteri).
- Etichetta personalizzabile per ciascuno degli 8 pulsanti.
- Accesso immediato ai primi 4 canali (gli altri 4 sono disponibili nella seconda pagina).
- Collegamento al sistema Velbus tramite cavetto a 4 conduttori.
- LED d'indicazione per ciascun canale.
- Retroilluminazione a LED per ciascun tasto.
- Ricevitore IR integrato.
- Le impostazioni e le etichette di ciascun tasto possono anche essere effettuate tramite PC utilizzando il modulo con interfaccia USB (VMB1USB) o con interfaccia RS232 (VMB1RS).
- Il software di gestione velbuslink_rev5_6, è scaricabile gratuitamente dal sito www.velleman.be.

tamente dal sito
www.velleman.be.

Specifiche:

- Retroilluminazione: LED bianchi.
- Protocollo di comunicazione seriale: CAN Bus su RS485.
- Alimentazione: 12-18 Vdc / 30 mA.
- Numero canali: 8.
- Numero indirizzi Velbus: 255.
- Dimensioni: 118 (L) x 80 (H) x 30 (P) mm (placca inclusa).
- Dimensioni minime del foro nel muro: 70 (L) x 50 (H) x 20 (P) mm.



in sequenza i tre segmenti orizzontali del display, simulando in questo modo l'emissione del segnale ad infrarossi.

Il sistema d'esempio

Per poter utilizzare il telecomando, è necessario disporre di una tastiera intelligente (VMB4PD) collegata ad uno o più attuatori: dato che il telecomando, nella configurazione predefinita, gestisce quattro canali, noi abbiamo scelto di comandare un modulo d'uscita a relé a quattro canali modello VMB4RY: lo schema di massima è visibile in Fig. 2. Data la semplicità dell'ambiente di test, effettueremo la configurazione manuale di tutto il sistema. Per prima cosa, cablate tastiera e modulo a relé secondo quanto riportato sempre in Fig. 2; verificate quindi che un solo jumper TERM (presente sia sulla ta-

tutto con una tensione stabilizzata di 12 Vdc 200 mA minimo.

All'accensione, il display della tastiera deve essere illuminato e riportare le scritte Button1, Button2, Button3 e Button4 in prossimità dei quattro pulsanti di comando. Sul modulo a relé, invece, deve essere acceso il solo led ON.

Iniziamo la configurazione del sistema partendo dalla tastiera, cui dobbiamo solamente assegnare l'indirizzo di rete: mantenete premuto il pulsante menu (posto sotto il display, al centro) fino a che non compaiano sulla riga in alto del display le segnalazioni Clock e Display. Premete brevemente il pulsante menu per visualizzare le scritte Mode, Address, Reaction e Labels quindi agite sul pulsante posto vicino ad Address: a questo punto compare l'indirizzo della tastiera, espresso in formato esadecimale a

valore della cifra selezionata. Come dicevamo, la tastiera ha indirizzo 01. Premete ripetutamente il pulsante menu per confermare il cambio dell'indirizzo e tornare alla schermata principale, che mostra le etichette da Button1 a Button4.

Continuiamo con la configurazione del modulo a relé; con l'aiuto di un piccolo cacciavite a taglio, rimuovete il frontalino trasparente ed impostate sui preselettori ADDR il valore D1 (acquisizione dei pulsanti che controllano in modo bistabile il canale 1): il led CH1 inizia a lampeggiare. Premete il pulsante Button 1 sulla tastiera fino a che non udite il relé commutare: rilasciate il pulsante per diseccitare il relé. Notate che adesso lampeggia anche il led rosso del pulsante, segnalandovi che questi è stato associato al canale 1 del modulo a relé. Una nuova pressione lunga di ➤

Abilitazione canali

Fig. 3

Per cambiare il canale, agire sul pulsante tenendo il telecomando in verticale.



questo pulsante fa sì che esso venga rimosso dalla lista dei pulsanti associati al relé 1. Impostate ora D2 sui preselettori di indirizzo del modulo a relé: lampeggia il led CH2. Premete il pulsante Button2 sulla tastiera fino a che non udite anche il relé 2 commutare. Ripetete la stessa procedura con i due canali rimanenti (D3 e D4) quindi posizionate i preselettori di indirizzo a 02.

Impostate, infine, tutti gli altri preselettori a FF (modalità toggle su tutti i canali). La procedura di configurazione è ultimata.

Fig. 4

Per controllare il canale scelto, orientare orizzontalmente il dispositivo verso la tastiera ed agire sul pulsante.



Utilizziamo il telecomando

Verificate, per prima cosa, che premendo i pulsanti presenti sulla tastiera i relé commutino ad ogni pressione. Ora potete verificare il funzionamento del telecomando: selezionate il canale da comandare

Il telecomando è in grado di controllare tutti gli otto canali gestibili dalla tastiera intelligente VMB4PD. Questa, infatti, pur disponendo di quattro soli pulsanti di attuazione, permette di accedere tramite il pulsante menu ad una seconda pagina del display, dove vengono mostrate le indicazioni predefinite Button5, Button6, Button7 e Button8 relative ai canali 5÷8.

Poiché non necessariamente tutti i canali devono essere gestiti, è stata prevista una procedura con cui includere o escludere dal telecomando uno qualsiasi di essi. Tale procedura è molto semplice: occorre aprire il telecomando, premere brevemente e ripetutamente il pulsante SW2 (quello piccolo in alto) per visualizzare a display il canale desiderato, quindi premere brevemente il pulsante SW1 (quello grosso) che, in questo caso, accende o spegne il punto decimale. Quando il punto decimale è acceso, il canale è attivato. Ad esempio, l'indicazione "1." significa che il canale 1 è stato attivato mentre "5" segnala che il quinto canale è stato disattivato. Una pressione del pulsante SW2 quando è visualizzato a display il numero 8 fa apparire la segnalazione "E", che significa END. Premete il pulsante SW1 per confermare l'uscita dalla procedura di abilitazione dei canali: ripetendo la procedura, potete abilitare ogni canale individualmente, in funzione delle vostre esigenze. Una volta richiuso il telecomando, la pressione ripetuta del pulsante plastico con dispositivo in posizione verticale determina la visualizzazione dei soli canali attivi. Tenete presente che, al momento dell'inserimento delle batterie, risultano attivi i canali da 1 a 4.



premendo ripetutamente il pulsante con il telecomando in posizione verticale (Fig. 3): fate apparire, ad esempio la cifra 3, corrispondente al terzo canale. Orientate ora il telecomando verso la tastiera (Fig. 4) e premete il pulsante ripetutamente: il relé del terzo canale cambia di stato ad ogni comando impartito.

La stessa cosa avviene selezionando sul trasmettitore tutti gli altri canali, sia in modalità toggle che in una qualsiasi delle modalità (timer, luce scala, Set, Clear, lampeggiatore, eccetera) supportate dal modulo a relé.

A questo punto, potreste sperimentare diverse configurazioni, impostando ad esempio, il canale 2 come timer luci scale, il canale 3 come "spegnimento ritardato" ed il quarto come lampeggiatore. Il sistema funzionerà sempre e il telecomando si comporterà allo stesso modo dei pulsanti presenti sulla tastiera. Potete anche provare ad abilitare sul telecomando i canali da 5 a 8 (disabilitati nella modalità predefinita), eseguendo la semplice procedura cui abbiamo dedicato un riquadro a parte.

per il

MATERIALE

Il telecomando VMB1RTS, la tastiera con display VMB4PD e il modulo d'uscita a quattro relé VMB4RY costano rispettivamente 48,00, 90,00 e 149,00 Euro. I prezzi si intendono IVA compresa. Ricordiamo che questi moduli, così come tutti gli altri dispositivi della serie Velbus, sono distribuiti nel nostro paese da Futura Elettronica.

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

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

Protezione per finali di potenza

di Marco Landoni

Progettato per l'impiego con amplificatori alimentati a tensione duale, controlla costantemente sia le tensioni di alimentazione che la temperatura sul dissipatore di calore ed interviene scollegando gli altoparlanti qualora un ramo dell'alimentazione venisse a mancare o la temperatura dovesse superare i 65 °C. Dispone, inoltre, della funzione anti-bump sia all'accensione che allo spegnimento.

Sul fascicolo N. 127 abbiamo presentato il progetto di un amplificatore audio Hi-Fi in grado di erogare fino a 200 W r.m.s.; esso, però, non integra alcuna protezione né per l'amplificatore stesso, né per le casse acustiche ad esso collegate. Tale scelta è dettata dalla necessità di non complicare eccessivamente l'architettura: nelle premesse di progetto, infatti, avevamo enfatizzato come il nostro obiettivo fosse quello di minimizzare al massimo il numero di componenti presenti sullo stampato del finale, in modo da non inserire possibili elementi elettronici, non inerenti la catena dell'amplifi-

cazione, che avrebbero potuto, in qualche modo, avere effetti deleteri sul segnale audio durante il suo percorso dall'ingresso all'uscita.

Se avessimo inserito tutti i circuiti di protezione direttamente sul PCB dell'ampli, invece, avremmo sicuramente dovuto realizzare, per ogni canale, oltre al circuito di protezione e controllo vero e proprio (con il proprio relé), anche un alimentatore a bassa tensione per alimentare tale circuito.

Poiché è buona norma (per il bene del segnale analogico) che il circuito stampato di un amplificatore Hi-Fi ➤

Fig. 1 - A sinistra è riportata la situazione che si viene a creare in mancanza di un sistema di controllo della tensione di rete: il relé d'uscita potrebbe rimanere eccitato per qualche secondo anche dopo che la tensione di rete viene a mancare, a causa della carica immagazzinata nei condensatori di alimentazione del circuito anti-bump. A destra, è rappresentato il funzionamento del circuito qui presentato che, nella medesima situazione scollega tempestivamente il carico dall'uscita dell'amplificatore.

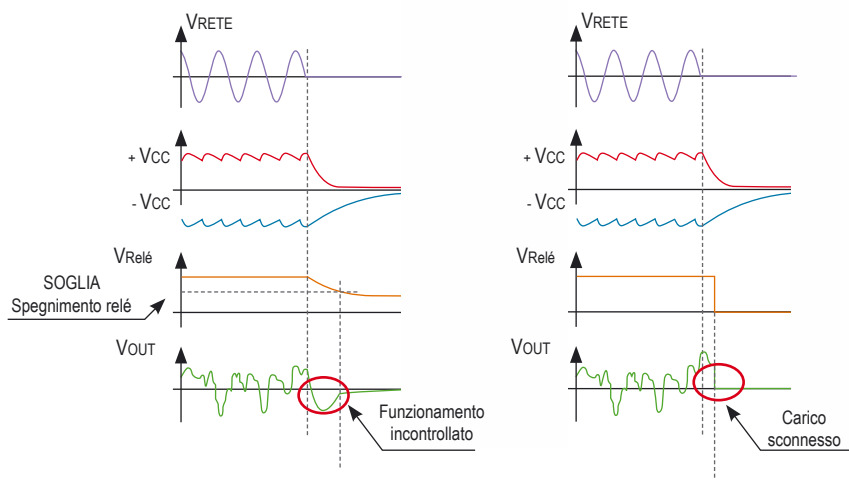
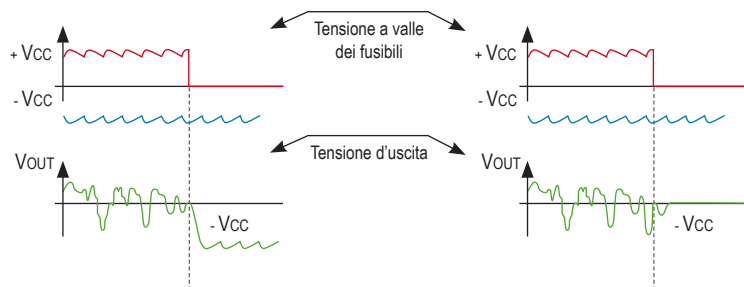


Fig. 2 - Intervento della protezione contro la rottura dei fusibili d'alimentazione. A sinistra il fusibile d'alimentazione positiva si interrompe e la tensione al carico assume un valore incontrollato tendente alla tensione negativa. A destra, utilizzando la protezione qui presentata, in presenza dello stesso guasto, il carico viene rapidamente scollegato dall'uscita dell'amplificatore.



contenga i soli componenti relativi "all'amplificatore", abbiamo ritenuto conveniente realizzare un circuito di protezione esterno, completamente autonomo e indipendente dal finale, alimentato attraverso un piccolo trasformatore a sé stante, in grado di proteggere sia gli altoparlanti sia i finali di un amplificatore montato in versione stereo. Questa protezione controlla costantemente sia la temperatura sui dissipato-

ri che le tensioni di alimentazione (positiva e negativa) ed interviene prontamente nel caso uno qualsiasi dei parametri controllati fuoriesca dai livelli di guardia.

Il circuito, inoltre, può essere facilmente utilizzato per proteggere altri amplificatori (anche con caratteristiche differenti da quello per il quale è stato concepito) dato che, a prescindere dalla potenza erogata in uscita, qualsiasi alimentatore si può

danneggiare per sovratemperatura e per problemi di alimentazione, così come ogni altoparlante può rimanere danneggiato quando viene alimentato (inavvertitamente, o per effetto di guasti) con una corrente continua generata da un finale di potenza guasto.

Vediamo insieme le caratteristiche di questo dispositivo.

Principio di funzionamento

Il circuito permette di effettuare molteplici controlli sullo stato dell'amplificatore da proteggere, che si suppone alimentato con una tensione duale (situazione peraltro, più frequente). In particolare, il circuito in questione implementa il classico ritardo dell'eccitazione del relé d'uscita rispetto l'accensione del finale, per evitare il fastidioso e dannoso "bump" che altrimenti si sentirebbe negli altoparlanti. Allo stesso modo, protegge anche dal "bump" che si verifica quando, ad amplificatore acceso, viene a mancare inaspettatamente la tensione di rete. Fate riferimento alla Fig. 1, che mostra l'andamento delle tensioni sui condensatori di filtro in funzione della corrente alternata dell'alimentazione. Nel momento in cui viene a mancare la tensione di rete, può capitare (in realtà è il caso più frequente) che i condensatori del finale si scarichino con tempi differenti: ciò può causare la fluttuazione incontrollata della tensione d'uscita dell'amplificatore, dato che esso è ancora alimentato con una tensione via via decrescente in modo non prevedibile. Tale situazione è certamente nociva per gli altoparlanti poiché questi ultimi sono potenzialmente pilotati da un segnale alternato sovrapposto ad una continua di valore imprevedibile. Il nostro circuito (che consuma poca corrente) tiene costantemente sotto controllo la tensione di rete e riesce reagire, qualora questa venisse a mancare, aprendo il relé d'uscita in meno di 50 ms (parte destra della

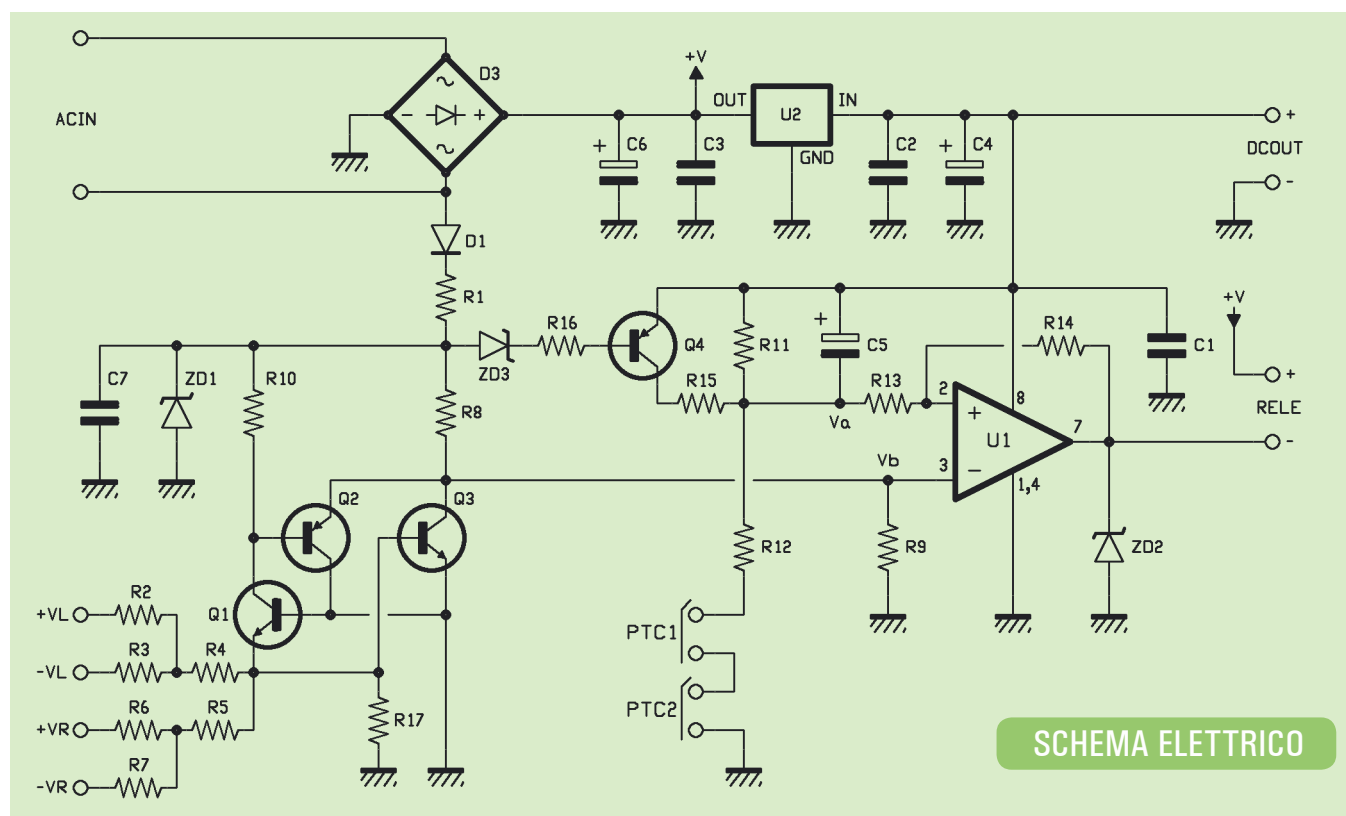
Fig. 1): grazie al consumo ridotto, infatti, esso rimane alimentato con la sola carica immagazzinata dal suo condensatore di filtro per un tempo di gran lunga maggiore di quello impiegato dai condensatori di alimentazione al finale per scaricarsi.

Unitamente a questa caratteristica, avete a disposizione una protezione nei confronti della rottura di eventuali fusibili inseriti in serie all'alimentazione dei finali (Fig. 2): alcuni amplificatori, come quello per il quale questa protezione è stata progettata, non dispongono di circuiti per la limitazione della corrente

cuni casi può assorbire una corrente istantanea ben superiore a quella che assorbirebbe un carico puramente resistivo utilizzato nella medesima situazione. Per questo motivo, in serie all'alimentazione di taluni finali, si possono trovare inseriti dei fusibili: questi ultimi possono essere attraversati da brevi correnti di picco ben superiori al loro valore nominale senza fondere mentre, se stimolati in modo continuo con la corrente nominale, si aprono entro qualche centinaio di ms. Ciò assicura una protezione nei confronti dei corto-circuiti sull'uscita ed allo stesso

facendo bruciare per sovratemperatura la vernice isolante di cui il suo filo è ricoperto. Anche in questo caso il circuito qui descritto può intervenire: esso monitorizza costantemente le tensioni d'alimentazione del finale e, nel caso una delle due venisse a mancare, provvede a scollegare l'altoparlante dai morsetti d'uscita dell'amplificatore.

Si è anche voluto inserire un'ulteriore protezione: il controllo della sovratemperatura dei transistor finali. Due termistori PTC vengono montati a contatto diretto con le alette di raffreddamento e ne con-



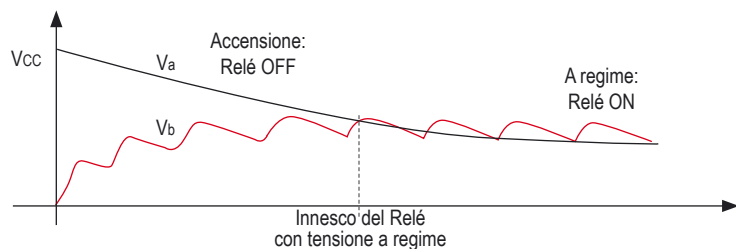
massima che il carico può assorbire. Ciò è dovuto essenzialmente a due ragioni: la semplificazione del circuito audio (intesa come l'assenza dalla basetta dell'amplificatore di circuiti che non hanno a che vedere con la parte di amplificazione) e la garanzia che l'altoparlante possa sempre ricevere tutta la corrente di picco di cui necessita. Ricordiamo, infatti, che un altoparlante, essendo un carico fortemente reattivo, in al-

tempo consente di sfruttare tutta la corrente di picco che l'amplificatore è in grado di erogare. Per contro, alla rottura accidentale di uno dei due fusibili, l'uscita dell'amplificatore non è più controllata e tipicamente assume un valore di tensione prossimo a quello del ramo d'alimentazione rimasto funzionante; tale tensione, che viene a trovarsi direttamente ai capi dell'altoparlante, potrebbe danneggiarne la bobina

trollano la temperatura: nel caso in cui quest'ultima dovesse superare e il valore di soglia di 65 °C circa, il carico verrebbe scollegato immediatamente.

Lo stesso circuito, infine, dispone di una tensione d'uscita regolata a 12 V 500 mA, pensata per alimentare una coppia di strumenti VUmeter a barra di LED quali possono essere per esempio i kit K4304, K4306 o K4305.

Fig. 3 - Evoluzione temporale delle tensioni agli ingressi del trigger di Schmitt della protezione, dal momento dell'accensione del circuito fino al regime, situazione in cui risulta $V_b > V_a$ (vedere schema elettrico). Il segnale V_b presenta del ripple residuo di circa 500 mV.



Schema elettrico

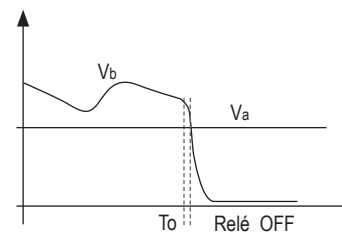
Lo schema del circuito è molto compatto poiché per realizzare tutte le funzioni descritte vengono impiegati solamente quattro transistor, un comparatore, un relé ed una manciata di componenti passivi. Iniziamo a descrivere lo schema partendo dal comparatore U1, un comune LM311: esso è in grado di pilotare direttamente un relé visto che la sua uscita open-collector può erogare fino a 50 mA. Il comparatore viene utilizzato in configurazione a trigger di Schmitt (comparatore la cui uscita commuta solamente quanto la tensione in ingresso eccede uno dei due valori di tensione predefiniti) la cui finestra di isteresi è definita dal rapporto tra R13 e R14 unitamente alla tensione d'alimentazione. Tale isteresi assicura che il ripple presente sulla tensione V_b al pin invertente, ottenuta raddrizzando la tensione d'uscita del trasformatore d'alimentazione tramite i diodi D1, ZD1 la resistenza R1 e il condensatore C7, non crei commutazioni spurie dell'uscita. A regime, infatti, la tensione sul pin invertente è di poco superiore a quella presente ai capi della serie "R12 con i due PTC" (V_a), pertanto il transistor d'uscita interno a U1 è in saturazione ed il relé risulta eccitato (Fig. 3).

In condizioni normali, inoltre, i transistor Q1, Q2 e Q3 sono interdetti: le due coppie di resistenze R2, R3 e R6, R7, infatti, formano dei partitori resistivi tra le tensioni d'alimen-

tazione dei finali prese a valle dei fusibili e, poiché tali tensioni sono di tipo duale, i punti centrali dei partitori si trovano, normalmente, a 0 V. Ne deriva che anche l'emettitore di Q1 e la base di Q3 sono a 0 V e i due transistor sono, quindi, interdetti (anche Q2 è interdetto perché forzato in tale stato da R10).

Nel caso in cui uno dei fusibili d'alimentazione dei finali si aprisse (Fig. 2), uno dei due ponti verrebbe sbilanciato e come conseguenza anche la tensione al nodo tra R4 ed R5 assumerebbe un valore diverso da zero. Per esempio, se bruciasse un fusibile sul ramo d'alimentazione negativa allora il punto comune tra R4 e R5 diventerebbe positivo e porterebbe in conduzione Q3; così facendo, la tensione sul pin invertente dell'operazionale sarebbe minore di quella presente sull'ingresso non in-

Fig. 4 - Tensioni all'ingresso del trigger in corrispondenza dell'apertura di uno dei fusibili d'alimentazione.



vertente e l'uscita di U1 cambierebbe di stato logico diseccitando il relé (Fig. 4). Nel caso contrario (brucia un fusibile sul positivo), la tensione sull'emettitore di Q1 diventerebbe negativa portando lo stesso Q1 in conduzione; contemporaneamente anche Q2 andrebbe in saturazione portando, come nel caso precedente, la tensione V_b all'ingresso invertente di U1 al disotto di quella presente sull'altro ingresso: anche in questo caso, pertanto, il relé verrebbe diseccitato. Supponiamo ora che si spenga l'amplificatore o venga a mancare la tensione di rete (Fig. 5): immediatamente manca la tensione al secondario del trasformatore d'alimentazione quindi C7, non più caricato, può solo scaricarsi attraverso la serie di resistenze R8-R9. Non appena la tensione al morsetto non invertente di U1 scende al

Fig. 5 - Evoluzione temporale delle tensioni d'ingresso del trigger al mancare della tensione di rete: il condensatore di filtro della tensione di trigger (C_7) si scarica molto rapidamente e determina lo sgancio immediato del relé, che scollega il carico.

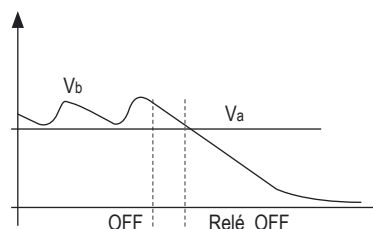
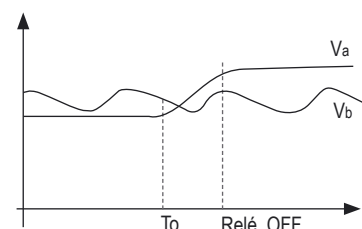
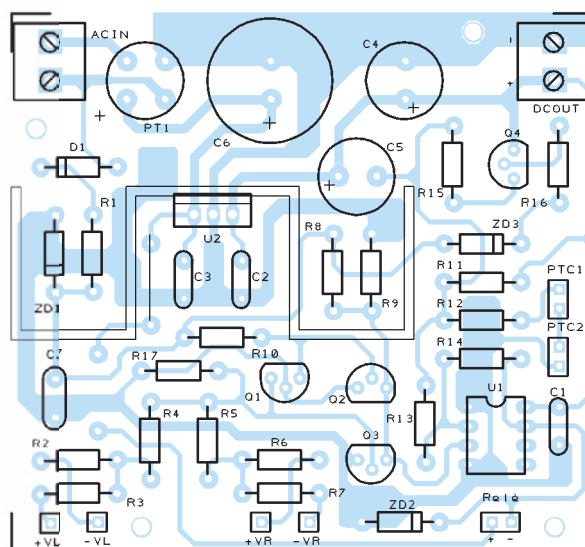
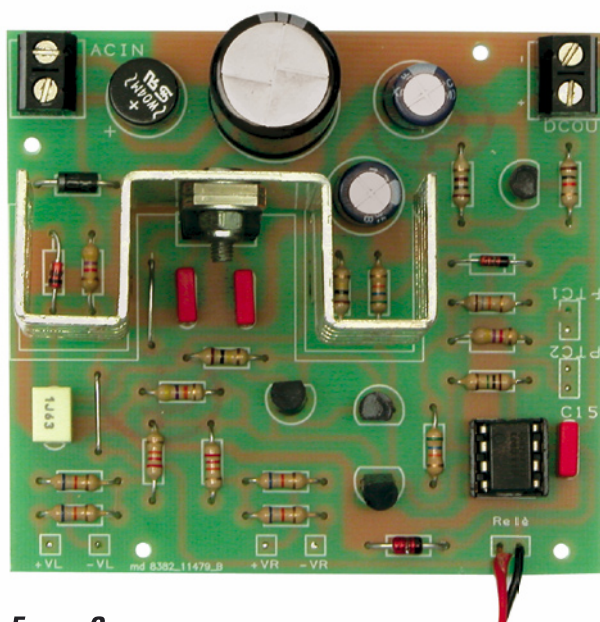


Fig. 6 - Al crescere della temperatura su una delle alette di raffreddamento, cresce anche la tensione V_a . Se quest'ultima raggiunge un valore sufficiente (ottenuto a circa 65÷70 °C) per innescare il trigger di Schmitt il relé viene diseccitato.



La protezione è stata realizzata con componenti ad inserzione facili da reperire in commercio. Notate il dissipatore di dimensioni abbondanti (resistenza termica inferiore a 10 °C/W), necessario solo qualora decidiate di utilizzare la protezione per alimentare un VUmeter a led (10 mA circa per ogni led acceso). In caso contrario, può essere più piccolo (resistenza termica di 18 °C/W).



ELENCO COMPONENTI:

R1, R12: 4,7 kohm
R2, R3, R6, R7: 68 kohm
R4, R5: 22 kohm
R8, R10: 100 kohm
R9, R13: 56 kohm
R11: 18 kohm
R14: 1 Mohm
R15: 100 ohm
R16: 10 kohm
R17: 47 kohm
C1+C3: 100 nF 63 VL poliestere
C4, C5: 470 µF 16 VL elettrolitico

C6: 2200 µF 25 VL elettrolitico
C7: 1 µF 63 VL poliestere
D1: 1N4007
D2: 1N4007
D3: ponte diodi 1.5A 100V
ZD1: Zener 12V 500mW
ZD2: Zener 24V 500mW
ZD3: Zener 5,1V 500mW
Q1, Q3: BC547
Q2, Q4: BC557
U1: LM311
U2: 7812

PTC1, PTC2: PTC temperature limit, EPCOS D901, 70°C

Varie:

- Morsettieria 2 poli (2 pz.)
- Zoccolo 4+4
- Relé 12V DC, 2 contatti x 6A
- Trasformatore prim. 230V, sec. 12V 5W
- Dissipatore TO220 max 10°C/W
- Vite 8 mm 3 MA
- Dato 3 MA
- circuito stampato

disotto della tensione di soglia del comparatore, l'uscita dello stesso cambia di stato logico diseccitando di conseguenza il relé in poche decine di ms.

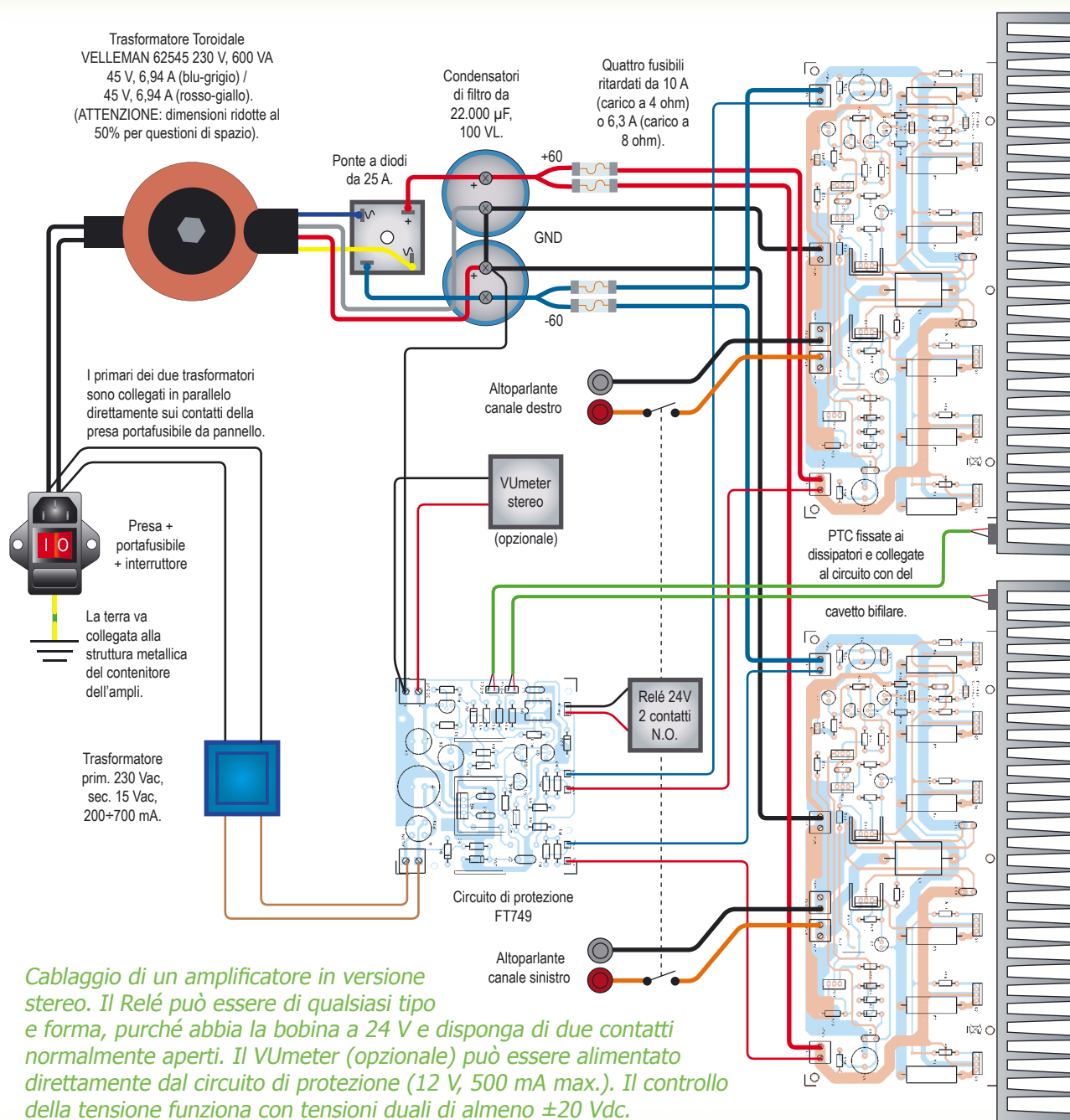
In serie alla resistenza R12 sono connessi due termistori PTC (Positive Temperature Coefficient), usati come sensori di temperatura per le alette di raffreddamento dei finali. Tali componenti hanno la capacità di aumentare la propria resistenza interna all'aumentare della temperatura e il tipo scelto è studiato per essere impiegato come discriminatore di temperatura: ha, infatti, il pregio

di riuscire a variare la propria resistenza di parecchie migliaia di ohm in un intervallo di poche decine di gradi. Esso passa dal centinaio di ohm a 50 °C ai 20 kohm alla temperatura di 80 °C. Praticamente, presenta una soglia di temperatura attorno i 65 °C oltre la quale da circuito "quasi" chiuso diventa un circuito "quasi" aperto. Questo dovrebbe già farvi intuire che all'aumentare della temperatura dell'aletta di raffreddamento anche la tensione all'ingresso non invertente del trigger di schmitt aumenta proporzionalmente (Fig. 6): al superamento della soglia fis-

sata sul morsetto invertente, l'uscita del comparatore cambierà stato diseccitando nuovamente il relé. Con i valori dei componenti inseriti e con il tipo di PTC scelto si ottiene un intervento della protezione attorno ai 65÷70 °C misurati sull'aletta di raffreddamento.

La funzione anti-bump d'accensione è ottenuta invece tramite il condensatore elettrolitico C5: esso, inizialmente scarico, comincia a caricarsi non appena viene alimentato il circuito. La tensione ai suoi capi cresce secondo il ben noto andamento esponenziale (Fig. 3): il relé si eccita ➤

Piano di cablaggio



Cablaggio di un amplificatore in versione stereo. Il Relé può essere di qualsiasi tipo e forma, purché abbia la bobina a 24 V e disponga di due contatti normalmente aperti. Il VUmeter (opzionale) può essere alimentato direttamente dal circuito di protezione (12 V, 500 mA max.). Il controllo della tensione funziona con tensioni duali di almeno ± 20 Vdc.

non appena la tensione sul morsetto non invertente del trigger scende al disotto della tensione presente sul morsetto invertente. Le costanti di tempo sono state scelte in modo tale per cui la carica di C5 sia più lenta di quella di C7, consentendo così di implementare la funzione anti-bump d'accensione. Il transistor PNP Q4 ed il diodo zener ZD3 servono per scaricare istantaneamente la capacità C5 al momento dello spegnimento

del circuito; se così non fosse, essa si scaricherebbe lentamente e impedirebbe alla funzione anti-bump di operare in modo corretto nel caso in cui il circuito venisse nuovamente alimentato entro di pochi secondi. Sono stati, infine, previsti dei morsetti di collegamento al regolatore U2 per poter alimentare eventuali altri circuiti esterni (come può essere una coppia di VUmeter) fino ad un massimo di 500 mA.

Realizzazione pratica

Come potete vedere nel piano di montaggio, il circuito è molto piccolo e piuttosto semplice da assemblare. Realizzate il circuito stampato (il master è come sempre scaricabile gratuitamente dal nostro sito www.elettronica.in.it) ed iniziate il montaggio partendo dai componenti a basso profilo quali i due ponticelli a filo, diodi, resistenze, condensatori non polarizzati e lo zoccolo

per il circuito integrato. Montate, quindi, i transistor, i condensatori elettrolitici, le due morsettiere a due poli, passo 5,08 mm e i due jumper (TC1 e TC2, ma se preferite, potete saldare sulle rispettive piazzole dei cavetti bifilari) cui collegherete, in fase di inserimento del circuito nell'amplificatore, le due resistenze PTC. Concludete il montaggio con lo stabilizzatore di tensione, che va saldato verticalmente e dotato di un robusto dissipatore (a U, con resistenza termica inferiore ai 10 °C/W) qualora pensiate di utilizzare il circuito per alimentare anche un VUmeter stereo a led. Senza inserire il comparatore nello zoccolo, applicate alla morsettiera ACIN una tensione alternata di 15 V e verificate che sulla morsettiera DCOUT siano presenti 12 V stabilizzati. Verificate infine che ai capi dello zener ZD1 siano presenti circa 12 V. Attenzione: questa tensione non è la stessa proveniente dall'uscita dello stabilizzatore. Ora spegnete il circuito, attendete che tutti i condensatori si siano scaricati, ed innestate il com-

Fig. 7 - Le resistenze PTC utilizzate sono delle EPCOS D901, ma potete usarne altre di pari caratteristiche.

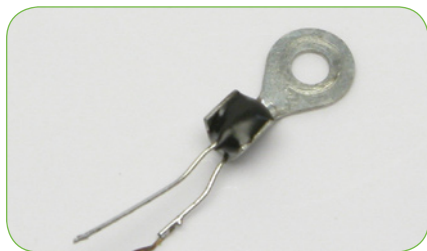
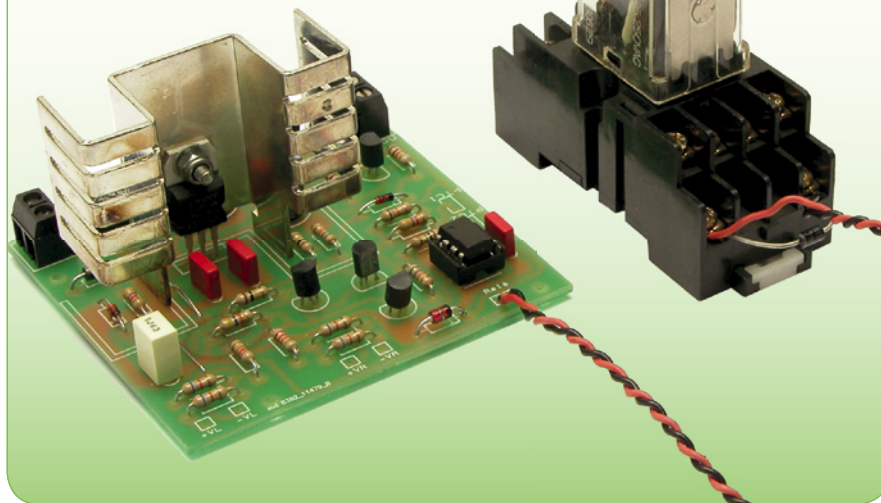


Fig. 8 - Il relé è esterno al dispositivo, così da poter scegliere il modello più appropriato. Va inserito un diodo 1N4004 in parallelo alla bobina: il catodo deve essere rivolto sul filo rosso (+21 Vdc).



paratore sul proprio zoccolo: siete pronti per collegare la protezione all'amplificatore. Per fare ciò, fate riferimento al piano di cablaggio riportato qui a sinistra, realizzato in previsione dell'inserimento in un amplificatore stereo.

Note particolari

Anche se le abbiamo già riportate nel piano di cablaggio, riteniamo opportuno aggiungere alcune note importanti, relative all'utilizzo in pratica. Innanzitutto, il circuito non è dotato di alcun relé: potete usarne uno di qualsiasi tipo, purché disponga di due contatti normalmente aperti e la sua bobina funzioni con una tensione minima di 17 Vdc (è il caso di quasi

tutti i relé a 24 V).

Viste le caratteristiche dell'amplificatore per cui è stato progettato, noi abbiamo optato per un robusto relé montato su zoccolo "octal" (Fig. 8). Relativamente alle resistenze PTC, potete acquistare lo stesso modello (o modelli simili) che abbiamo usato noi (EPCOS D901, Fig. 7) oppure realizzarle voi stessi: se utilizzate un capocorda giallo a occhio (fino a 6 mm² di sezione del conduttore), potete prima saldare il fili ai due reofori del PTC, quindi isolarli con del tubetto termoretraibile ed infine inserire il corpo della PTC nel capocorda, che nel frattempo avete riempito con del collante tenace, come le resine epossidiche a doppio componente.

per il

MATERIALE

Tutti i componenti utilizzati in questo progetto sono facilmente reperibili in commercio. Il master del circuito stampato e tutti i disegni di progetto sono scaricabili dal sito della rivista (www.elettronica.in.it). Il trasformatore toroidale adatto ad alimentare questo circuito (Futura Elettronica, cod. 62545) può essere acquistato al costo di 115,00 Euro, IVA compresa.

**Il materiale va richiesto a: Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)
Tel: 0331-799775 ~ Fax: 0331-792287 ~ <http://www.futurashop.it>**

Registratore vocale 10-35 secondi

di *Paolo Gaspari*

Memorizza e riproduce fino a 35 secondi di parlato o musica captati da un microfono preamplificato. Il tempo di registrazione e la velocità di riproduzione possono essere variati tramite un trimmer. Può essere attivato anche mediante un segnale esterno.



Uno dei settori in cui la tecnologia dei semiconduttori ha determinato drastici cambiamenti è quello della registrazione audio: il passeggero nel sedile di fronte a noi in treno, il podista o ciclista lungo i viali cittadini, il ragazzo che va o torna da scuola con la cuffia nelle orecchie, non ha in tasca un walkman, “gioiellino” degli anni '80, ma un minuscolo registratore e lettore MP3. Niente più cassette... oggi la musica si registra nei chip. Tra i primi a dare un assaggio della tecnologia digitale fu (una quindicina di anni or sono) la Information Storage Devices, con i suoi DAST, che erano registratori e lettori digi-

tali single-chip completi. Oggi la Winbond, che l'ha rilevata, continua a proporre integrati del genere, tra i quali spiccano i Chipcorder, veri e propri registratori allo stato solido comandabili con semplici pulsanti e capaci di registrare anche diversi minuti di parlato. Il dispositivo qui descritto è stato realizzato proprio con un Chipcorder: si tratta di un registratore e lettore digitale utilizzabile per realizzare dei promobox, ossia i ripetitori di messaggi promozionali che troviamo inseriti in sagome varie sagome (pupazzi fermi o in movimento, statue ecc.) dentro e fuori dai negozi e negli esercizi commerciali in genere, >

nelle stazioni ferroviarie e dovunque passi molta gente. Ma anche nelle farmacie, dietro le bilance o le macchinette per la misura della pressione arteriosa. L'utilizzo di messaggi promozionali riprodotti quando una persona passa o sosta davanti a una vetrina o un oggetto è ormai una consolidata tecnica di richiamo. Bene, identificato il possibile impiego, vediamo insieme il progetto.

Come funziona

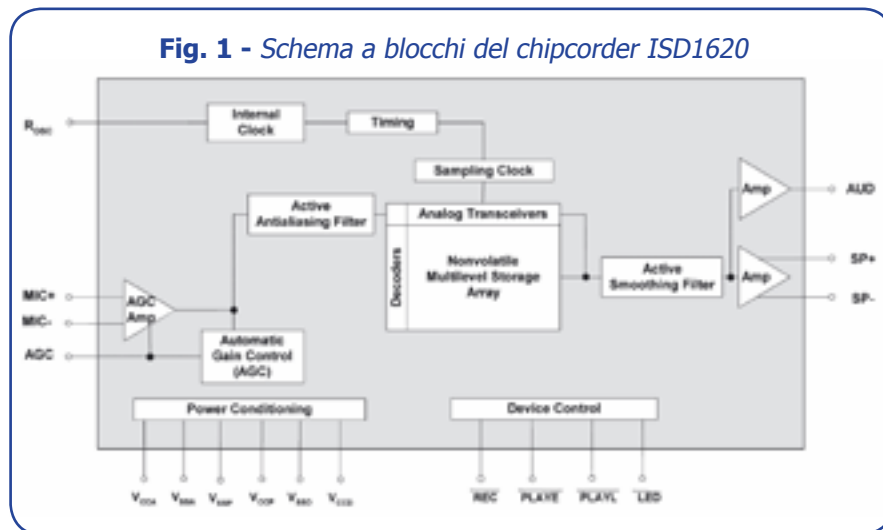
Il nostro circuito impiega un Chipcorder ISD1620 (il cui schema a blocchi è riportato in Fig. 1), ossia un chip a sintesi vocale in grado di memorizzare fino a 35 secondi di suoni o parlato, quindi riprodurli; il tutto mediante appositi comandi dati alle linee PLAY e REC. In registrazione campiona il segnale audio prendendone dei pezzi periodicamente e con una cadenza prestabilita, quindi ne colloca i livelli in una speciale memoria multilivello brevettata dalla Winbond; la memorizzazione è analogica, quindi non viene utilizzato alcun A/D converter. Ciò che viene campionato è il segnale di bassa frequenza applicato all'amplificatore differenziale i cui ingressi sono localizzati ai piedini 6 (MIC+) e 7 (MIC-); i dati risultanti dal campionamento vengono, infine, scritti nella memoria multilivello. In riproduzione, un'apposita logica provvede a estrarre sequenzialmente i livelli memorizzati nelle singole celle di memoria, nella stessa sequenza e con la medesima velocità (salvo che non venga modificata la

frequenza di lavoro dell'oscillatore) con la quale sono stati registrati; si ricostruisce, quindi l'audio originale (quello registrato in precedenza) che un piccolo amplificatore con uscita a ponte può rendere udibile mediante un altoparlante da 16 o 32 ohm collegato tra i piedini 14 e 15. Per registrare occorre portare a zero logico il piedino 2 (/REC) che normalmente deve stare a livello alto; facendo ciò, il campionatore inizia la digitalizzazione e memorizza la componente BF fin quando lo stesso

al secondo (/PLAYL, pin 4) pone in riproduzione il chip da quando viene portato a livello basso fino a che non torna ad 1. In entrambi i casi la riproduzione del messaggio termina automaticamente quando finisce il messaggio stesso.

Durante la riproduzione, la linea /LED pulsa con cadenza di 0,5/0,5 secondi, facendo lampeggiare il led; a fine riproduzione, indipendentemente da come è stato attivato il chip (ad impulso, con la linea /PLAYE, o a livello, mediante /PLAYL) il pin

Fig. 1 - Schema a blocchi del chipcorder ISD1620

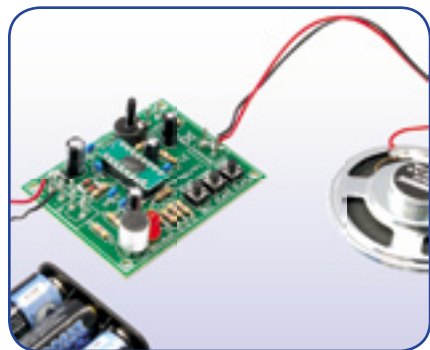


so pin non viene rilasciato. Per tutta la durata della registrazione, l'uscita /LED (piedino 16) va a zero logico per tornare a livello alto quando il pin 2 viene riportato ad uno logico, oppure quando la registrazione termina automaticamente perché si è esaurita la memoria a disposizione. Ogni volta che si verifica una transizione 1/0 sulla linea /REC (pin 2), l'integrato registra il nuovo segnale e cancella eventuali registrazioni preesistenti.

Vediamo adesso come avviene la riproduzione del messaggio registrato: il componente dispone di due piedini, che consentono di comandarlo sia con un impulso (un pulsante) sia a livello; il primo è siglato /PLAYE (pin 3) e attiva l'ISD1620 se semplicemente riceve un impulso a zero e poi torna a riposo (1 logico). Quanto

16 torna a livello alto. In altre parole, in registrazione /LED è a livello basso (led acceso) fin quando non si rilascia SW1 o termina il tempo disponibile, mentre in lettura pulsa fino al termine del messaggio se la lettura è stata avviata con /PLAYE. Avviando la riproduzione con il pulsante collegato a /PLAYL, il led lampeggia fino a che il pulsante rimane premuto e, in ogni caso, fino a fine messaggio se il pulsante viene mantenuto premuto.

Durante la riproduzione, il convertitore del Chipcorder trasforma i dati letti dalla memoria in segnale audio, che invia sia al piedino 13 (AUD) che ad un amplificatore interno a ponte in classe D, le cui uscite sono localizzate ai piedini 11 (SP+) ed 8 (SP-); a questi ultimi è possibile collegare un altoparlante di impedenza

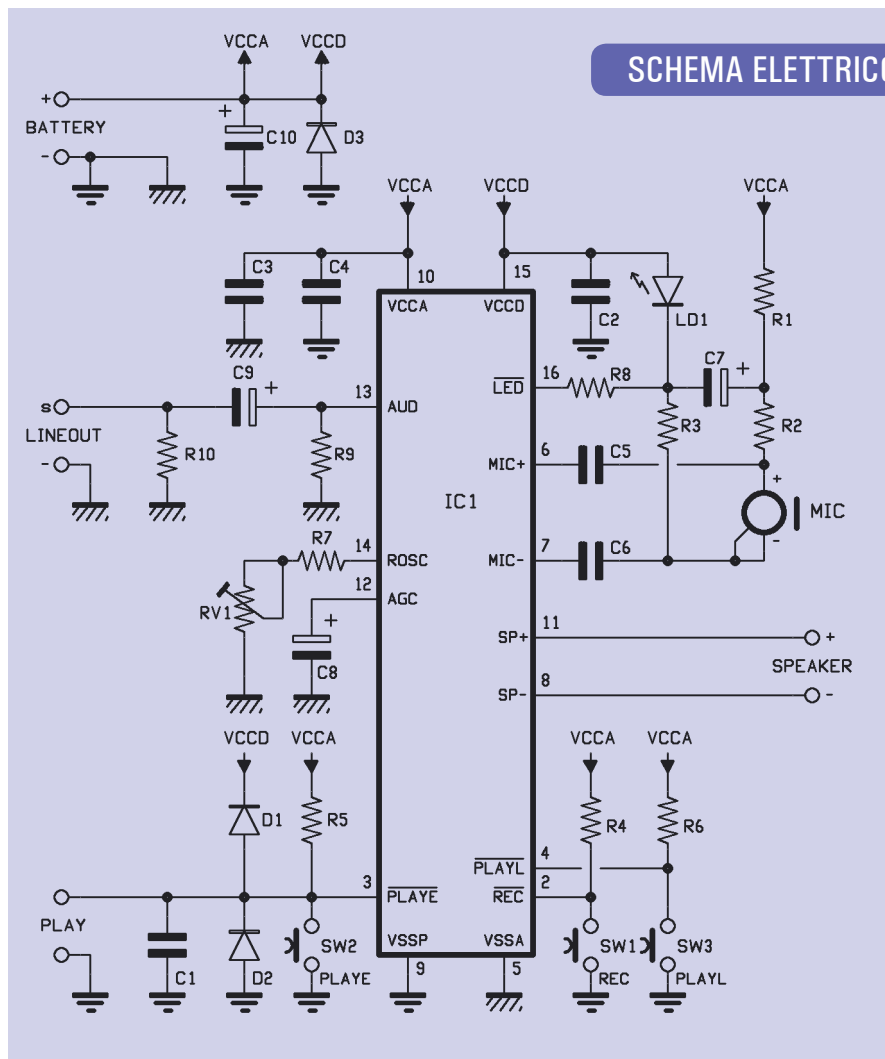


non minore di 8 ohm o una cuffia (sempre da non meno di 8 ohm). La potenza erogata (670 mW su 8 ohm a 5,5 V) è sufficiente a rendere udibile l'audio a qualche metro di distanza. Chi necessita di un livello sonoro più spinto può prelevare la BF dal piedino AUD (13, che è un'uscita ad alta impedenza, capace di erogare 3 mA) oppure, indifferentemente dai piedini 8 o 11 (rispetto a massa) e inviarla a un amplificatore di maggior potenza.

Entrambe le uscite audio vengono disattivate automaticamente durante le fasi di spegnimento (ritorno a riposo) e registrazione, così da evitare che l'altoparlante riproduca suoni casuali.

Sia il campionatore che il convertitore lavorano usando come clock la frequenza generata dall'oscillatore interno al chip; quest'ultimo decide, quindi, la durata effettiva della registrazione o lettura e, di riflesso, la qualità della registrazione. Dunque, più si alza la frequenza di campionamento, più estesa è la risposta in frequenza, ma, per contro, si accorcia il tempo a disposizione (infatti a parità di durata dell'audio viene consumato maggiore spazio in memoria, per avere una migliore definizione); viceversa, abbassando la frequenza del clock è vero che si allunga il tempo disponibile, tuttavia lo si fa a discapito della qualità della registrazione e dell'ascolto.

Per ogni Chipcorder il tempo nominale a disposizione è riferito a una ben determinata frequenza di clock, ottenuta con i valori suggeriti della resistenza di temporizzazione esterna dell'oscillatore, che è quella connessa tra il piedino 14 (ROSC) e massa. Il costruttore prevede una frequenza di clock di valore compreso tra 4 e 12 kHz, limiti cui, per l'ISD1620, corrispondono durate di registrazione (e lettura) rispettivamente di 40 e 13,3 secondi. Il clock consigliato è 8 kHz e permette la corretta riproduzione della voce umana. Volendo



apportare delle modifiche, la minima frequenza di clock si ottiene con una resistenza di 160 kohm, mentre la massima la si ha con 53 kohm; 120 kohm danno 5,3 kHz (30 secondi) 100 kohm determinano 6,4 kHz (durata di 25 s) mentre 80 kohm danno gli 8 kHz consigliati (20 secondi).

Schema elettrico

Il nostro circuito è sostanzialmente l'ISD1620 nella sua canonica configurazione: troviamo, infatti, i due pulsanti per l'attivazione della riproduzione parziale (/PLAYL, SW3) o dell'intero messaggio (/PLAYE, one-shot, SW2), oltre al comando per la registrazione (/REC, SW1); al piedino 16 è connesso il led di attività, mentre dal 13 possiamo prelevare l'audio per inviarlo a un amplificatore di potenza o un registratore. L'al-

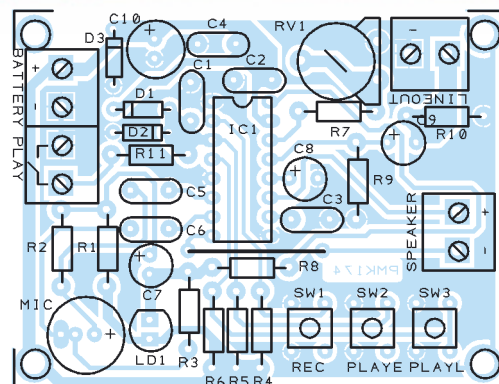
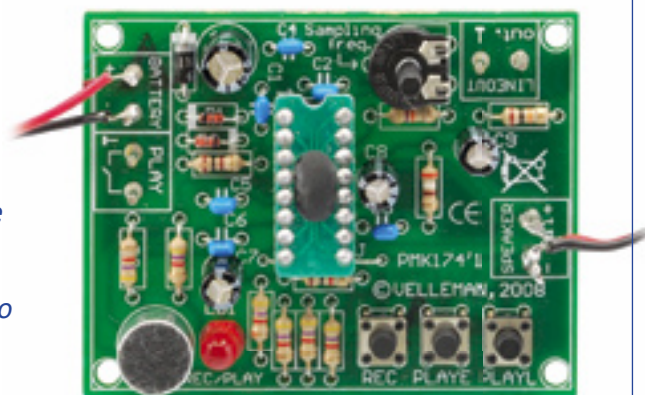
toparlante utilizzato per riprodurre il messaggio si connette tra i piedini 8 e 11, mentre il microfono (una capsula preamplificata a due fili) con la sua rete di polarizzazione è collegato ai pin 6 e 7 mediante i condensatori di disaccoppiamento indispensabili per separare in continua l'ingresso differenziale da massa e dal positivo di alimentazione. L'uscita BF supplementare (AUD) è collegata, mediante il condensatore di disaccoppiamento C9 alla resistenza R10 (che funge da carico fittizio) e ai contatti LINE OUT (s e -) dai quali può raggiungere l'ingresso di un finale di potenza mono.

La frequenza di lavoro dell'oscillatore di clock viene definita dalla rete composta dal trimmer RV1 e dal resistore R7: più il cursore del trimmer viene portato verso massa, più alta ➤

Il registratore vocale fa uso di componenti ad inserzione e pertanto non è difficile da montare. Prestate attenzione ad un particolare: quando abbiamo realizzato il prototipo, il chipcorder ISD1620B da noi utilizzato (normalmente disponibile sia in versione DIL 8+8 che SOIC) era disponibile solo in versione COB (Chip On Board), in cui il chip di silicio vero e proprio è stato montato direttamente su un c.s. a forma di DIL 8+8 e protetto con una goccia di resina scura. Questa tecnica è spesso utilizzata nei display, ma anche in molti altri casi dove gli ingombri sono una problematica da non trascurare.

ELENCO COMPONENTI:

R1+R6: 4,7 kohm	D3: 1N4007
R7: 47 kohm	IC1: ISD1620B
R8: 1 kohm	LD1: led 5 mm rosso
R9: 390 ohm	MIC: Capsula microfonica
R10: 10 kohm	SW1+SW3: Microswitch
R11: 100 ohm	
RV1: Trimmer MO 10 kohm	Varie:
C1+C6: 100 nF multistrato	- Zoccolo 8+8
C7+C9: 4,7 µF 50 VL elettrolitico	- Portabatteria 3xAA
C10: 100 µF 25 VL elettrolitico	- Altoparlante 8 ohm
D1, D2: 1N4148	- Circuito stampato



diviene la frequenza e viceversa. Va da sé che, per ottenere una riproduzione fedele alla registrazione, non bisogna cambiare la posizione del perno dell'RV1 dopo aver effettuato la registrazione; altrimenti si otterrebbe un ascolto a rallentatore o accelerato. Il motivo per cui abbiamo previsto la possibilità di variare la frequenza di clock è presto detto: in questo modo, l'utente può scegliere il miglior compromesso tra tempo di registrazione (da 10 a 35 secondi circa) e qualità di registrazione/riproduzione in base alle proprie necessità.

Il circuito è completato da un ingresso che permette di comandare la riproduzione dall'esterno mediante un segnale elettrico: collegando l'input siglato PLAY a massa, ovvero applicando ad esso un livello logico basso (non più di 0,7 V di tensione positiva) l'ISD1620 inizia a riprodurre e va avanti fino alla fine del messaggio in esso contenuto, anche se l'input viene ripristinato. Infatti PLAY è collegato in parallelo alla linea PLAYE e per avviare la riproduzione basta semplicemente che riceva un impulso a zero logico. Sono stati previsti

due diodi per evitare che l'ISD1620 venga danneggiato nel caso all'input di comando sia applicata una tensione negativa o di valore più elevato di quello dell'alimentazione. L'ingresso di comando è stato previsto per poter utilizzare il circuito come avvisatore acustico vocale attivato da un evento: ad esempio per fargli dare istruzioni a chi si avvicina a un tornello o a un cancello; in questo caso PLAY va attivato con un relé o un transistor comandato da un rilevatore di prossimità o un sensore a infrarossi passivi o una pe-

Applicazione pratica

Il registratore vocale presentato in queste pagine si presta a numerosi impieghi pratici: può essere, ad esempio, utilizzato come ripetitore di messaggi di benvenuto attivato da un sensore di movimento, comportandosi contemporaneamente come avvisatore di entrata / campanello. La cosa può essere facilmente ottenuta collegando l'ingresso PLAY al contatto normalmente aperto di un sensore PIR alimentato a tensione di rete (come il PIR1200C, acquistabile su www.futurashop.it) e posto in prossimità della porta d'ingresso del negozio o della propria abitazione.



Comandi e regolazioni

Il registratore/lettore digitale qui proposto dispone di tre pulsanti e un ingresso che permettono di comandare direttamente o tramite livelli di tensione le varie funzionalità; dispone altresì di un trimmer grazie al quale si può giocare sulla velocità di registrazione e lettura, ottenendo sia la regolazione (tra 10 e 35 secondi) del tempo disponibile, sia la riproduzione accelerata o rallentata del messaggio vocale registrato.

La tabella seguente riepiloga le modalità di funzionamento previste, i comandi tramite cui ottenerle e le segnalazioni date dal led.

Modalità	SW1	SW2	SW3	Input PLAY	LD1
Registrazione	Premuto finché si parla nel microfono.	A riposo.	A riposo.	1 logico (5V).	Acceso finché SW1 è premuto o termina il tempo a disposizione.
Riproduzione totale	A riposo.	Premuto per un istante, fa riprodurre l'intero messaggio.	A riposo.	1 logico (5V).	Lampeggiante fino alla fine del messaggio.
Riproduzione parziale	A riposo.	A riposo.	Premuto fin quando si vuole ascoltare il messaggio o una sua parte.	1 logico (5V).	Lampeggiante fin quando SW3 resta premuto o fino a fine messaggio.
Riproduzione da comando esterno	A riposo.	A riposo.	A riposo.	0 logico (0V).	Lampeggiante fino alla fine del messaggio.

dana. Un'altra applicazione vede il dispositivo usato come riproduttore di messaggi promozionali o di benvenuto negli esercizi commerciali: in quest'ultimo caso PLAY può essere attivato da un sensore ad infrarossi passivi, che dà lo zero logico (o chiude un contatto) ogniqualvolta rileva la prossimità di una persona. Concludiamo la descrizione dello schema elettrico con l'alimentazione: il tutto funziona con una tensione continua, di valore compreso tra 3 e 5 volt, applicata ai morsetti + e - PWR: con il potenziale a valle del

catodo del diodo di protezione D3 (quest'ultimo svolge un'efficace protezione nei riguardi dell'inversione accidentale di polarità) viene alimentato il circuito integrato. Notate che non è previsto alcun interruttore di accensione perché l'ISD1620 a riposo (ciò quando non viene premuto alcun tasto e non è in registrazione o riproduzione) consuma appena 1 μ A! Quindi può sempre restare alimentato, perché la logica che legge i pulsanti praticamente non chiede corrente; ciò rende il circuito ideale per l'alimentazione a pile, cosa che

abbiamo fatto noi nella realizzazione del nostro prototipo.

Realizzazione pratica

Il registratore/lettore digitale va realizzato montando i pochi componenti occorrenti su di un circuito stampato che autocostruirete ricorrendo alla fotoincisione (trovate la traccia del lato rame nel nostro sito Web www.elettronica.in.it); dopo l'incisione e la foratura, disponete i pochi componenti occorrenti in ordine di altezza, seguendo, per l'orientamento di quelli polarizzati, il disegno di montaggio visibile in queste pagine. La capsula microfonica impiegata è del tipo a due fili, sebbene quella che vedete nello schema e nel prototipo abbia tre contatti; in realtà un piedino è lo schermo, perché in questo componente l'involucro è, diversamente dalla media delle capsule electret, collegabile a parte. Se impiegate una, più comune, capsula a due soli fili, rammentate che il negativo è l'elettrodo connesso fisicamente all'involucro.

L'integrato ISD1620 va montato su uno zoccolo dip a 8 piedini per lato, badando di rispettare l'orientamento previsto) il riferimento va rivolto al condensatore C2.

Completato il montaggio, per l'alimentazione conviene collegare al circuito una presa polarizzata (a strappo) per pile da 9 volt, alla quale collegherete un portastilo da tre posti, le cui pile daranno i 4,5 volt necessari a far funzionare l'insieme.

per il

MATERIALE

Il progetto descritto in queste pagine è disponibile in scatola di montaggio (il kit viene prodotto dalla società Velleman e distribuito da Futura Elettronica). Può essere ordinato (cod. MK174) al costo di 19,00 Euro, IVA compresa. La scatola di montaggio comprende tutti i componenti, la basetta forata e serigrafata, il Chipcorder, l'altoparlante e il portapile AA a tre posizioni.

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

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



Corso di programmazione modulo Tibbo EM1000

A CURA DI **SAVINO GIUSTO**

Corso di programmazione e utilizzo del modulo EM1000 di casa Tibbo, un potente Ethernet Server completamente programmabile con funzioni di Web Server, Mail Server e molto altro ancora. In questa puntata vediamo come sia possibile utilizzare la modalità mail server per monitorare lo stato logico di una o più linee di ingresso e provocare l'invio di una e-mail che ne segnali la variazione di stato. Settima puntata.

Al giorno d'oggi la posta elettronica rappresenta uno dei principali mezzi di comunicazione: le e-mail consentono lo scambio di informazioni eterogenee, in modo comodo e veloce che, a prescindere dalla posizione geografica del mittente e del destinatario, vengono recapitate nell'arco di pochi secondi o, al massimo, di pochi minuti. Partendo da tali considerazioni e sfruttando ancora una volta le potenzialità del modulo

Tibbo, vedremo come realizzare un'applicazione che effettui il monitoraggio di una o più linee di ingresso dell'EM1000 ed invii, ad ogni variazione dello stato logico di almeno una di esse, una e-mail di allarme ad un indirizzo impostabile.

La posta elettronica secondo EM1000

La ricezione e lo smistamento dei messaggi di posta elettronica si svolgono, oggi, prevalentemente ➤

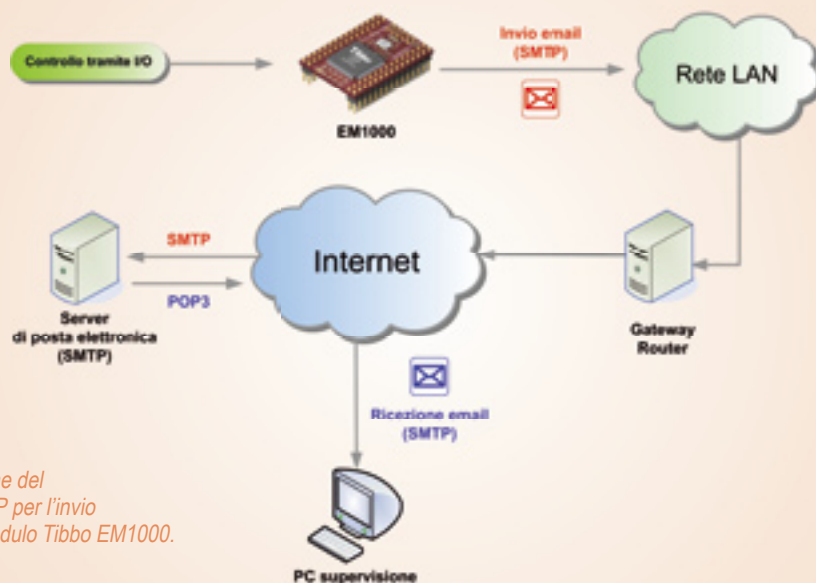


Fig. 1

Implementazione del protocollo SMTP per l'invio di e-mail col modulo Tibbo EM1000.

tramite il protocollo SMTP; durante l'accesso ai servizi di posta, invece, i protocolli più usati sono il POP3 (che scarica i vari messaggi sul computer dell'utente) e l'IMAP (che mantiene la posta sul server in modo che l'utente possa accedervi da diversi computer). Lo schema riportato in Fig. 1 riassume gli elementi che caratterizzano la nostra applicazione: un sensore generico con uscita di tipo 0/1 viene collegato al nostro EM1000 e, nel caso in cui l'ingresso dovesse cambiare di stato per via di una situazione di allarme, il modulo se ne accorgerebbe e farebbe immediatamente partire un'e-mail. Affinché ciò possa accadere, è necessaria la presenza di una connessione ad Internet, in modo che l'EM1000 possa interfacciarsi con il server di posta elettronica secondo il protocollo SMTP, passando attraverso un gateway oppure un router. In questo modo, non appena verrà ricevuto dal server di posta elettronica, il messaggio potrà essere notificato e scaricato sul client di posta installato su di un qualsiasi PC di supervisione.

Analisi del protocollo SMTP tramite Telnet

Prima di vedere nel dettaglio come si può realizzare un e-mail server con il modulo Tibbo, è indispensabile conoscere almeno i concetti base del protocollo SMTP, effettuando alcuni esercizi in pratica. Avendo a disposizione un PC è abbastanza facile verificare come funziona un server SMTP mediante un client Telnet: è sufficiente aprire una sessione di DOS, selezionare Start/Esegui e digitare CMD. A questo si punto digita la stringa

telnet smtp.tele2.it

nell'ipotesi che il proprio provider di posta sia

Listato 1

```
220 mailfe10.swip.net ESMTP 5.1.13
EHLO localhost
250-mailfe01.swip.net domain name should be qualified localhost
250-DSN
250-SIZE 20971520
250-AUTH LOGIN PLAIN
250-ETRN
250-TURN
250-ATRN
250-NO-SOLICITING
250-8BITMIME
250-HELP
250-PIPELINING
250 EHLO
AUTH LOGIN
334 VXN1cm5HBWU6
c2F2aW5vZ21lc3RvQHRlbgUyLm10
334 UGFzc3dvcmQ6
c2F2aW5v
235 cxu-8ye-6yn@tee12.it relaying authenticated
MAIL FROM: <sender@mydomain.com>
250 Ok
RCPT TO: <friend@example.com>
250 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Subject: messaggio di prova
From: sender@mydomain.com
To: friend@example.com

Ciao,
prova invio, by Elettronica In

250 Ok: queued as 12345
QUIT
221 Bye
```

tele2. Potete, altrimenti, selezionare un server SMTP dall'elenco disponibile all'indirizzo Internet [1] posto all'ultima pagina. In questo modo è possibile avviare la comunicazione con il server SMTP. Esso si aspetta di essere interrogato con particolari comandi, cui risponde con messaggi di errore oppure di conferma. Nel Listato 1 si riportano le stringhe ricevute dal server e quelle inviate dal client nel caso di invio di un ipotetico messaggio di posta elettronica.

Negli ultimi anni, quasi tutti i provider di posta elettronica hanno adottato dei protocolli di auten-

Il protocollo SMTP

Simple Mail Transfer Protocol (SMTP) è il protocollo standard per la trasmissione via Internet di e-mail. È un protocollo relativamente semplice, testuale, che permette di specificare uno o più destinatari di un messaggio, verificarne l'esistenza, e trasferire il messaggio. L'SMTP usa il protocollo di trasmissione TCP attraverso la porta 25. Per associare il server SMTP a un dato nome di dominio (DNS) si usa un record denominato MX (Mail Exchange).

L'SMTP iniziò a diffondersi nei primi anni '80: a quel tempo era un'alternativa all'UUCP, protocollo più adatto a gestire il trasferimento di e-mail fra computer, la cui connessione era intermittente. L'SMTP, d'altra parte, funziona meglio se i computer sono sempre collegati alla rete.

Sendmail fu uno dei primi (se non proprio il primo) *mail transfer agent* ad implementare il protocollo SMTP. Fino al 2001 sono stati scritti molti applicativi che implementano il protocollo SMTP come client (mittente dei messaggi) o server (destinatario del messaggio). Altri server molto diffusi sono Exim di Philip Hazel, Postfix di Wietse Venema, qmail di D. J. Bernstein, Courier di Sam Varshavchik e Microsoft Exchange Server.

Poiché SMTP è un protocollo testuale basato sulla codifica ASCII, non è permesso trasmettere direttamente testo composto con un diverso set di caratteri. Lo standard MIME permette di estendere il formato dei messaggi mantenendo la compatibilità col software esistente. Per esempio, al giorno d'oggi molti server SMTP supportano l'estensione 8BITMIME, che permette il trasferimento di un testo contenente caratteri accentati (non-ASCII) senza bisogno di codificarlo. Altri limiti di SMTP, quale la lunghezza massima di una riga, impediscono la spedizione di file binari senza codifica. L'SMTP è un protocollo che permette soltanto di inviare messaggi di posta, ma non di richiederli ad un server: per fare questo il client di posta deve usare altri protocolli, quali il POP3 (Post Office Protocol) e l'IMAP (Internet Message Access Protocol).

Per ulteriori approfondimenti fare riferimento alle RFC 2821 [2] e RFC 2554[4].

ticazione al fine di ridurre il problema dello spam (invio di grandi quantità di messaggi indesiderati, per lo più commerciali). Proprio in quest'ottica si è sviluppato il protocollo ESTMP (Extended SMTP), definito dalla specifica RFC-1869 [3] e incorporato (con SMTP) nella RFC-2821 [2]. La principale differenza rispetto all'SMTP, è che bisogna presentarsi con EHLO hostname, anziché HELO hostname. Se il server ricevente "parla" ESMTP, deve dichiarare le estensioni che supporta, una per riga, nella sua risposta all'EHLO (come nel caso del Listato 1). Nell'articolo sarà mostrato come autenticarsi mediante AUTH LOGIN PLAIN. Tale procedura prevede la codifica della propria password e nome utente in base 64. Nel corso dell'articolo si approfondirà anche questo argomento. Una sessione SMTP attraversa almeno sette fasi:

1. Il client SMTP contatta il server sulla porta TCP 25. Se questo è in ascolto e la connessione è accettata risponde con un messaggio 220 (Ready);
2. Il client chiede di stabilire la sessione SMTP inviando il comando EHLO seguito dal FQDN (Fully Qualified Domain Name). Se il server accetta, risponde con un messaggio 250 (Ok). Nella risposta del server sono contenute importanti informazioni relative ai protocolli supportati. In particolare, nel Listato 1 si noti la presenza del protocollo AUTH LOGIN PLAIN; questo significa che è necessario fornire l'autenticazione di password e nome utente codificate in base 64.
3. Il client inserisce il comando AUTH LOGIN e successivamente password e username codifica-

te. Sarà poi compito del server decodificare tali dati e visualizzare il codice 235 nel caso in cui risultassero corretti.

4. Il client indica il proprio indirizzo tramite il comando MAIL FROM: <indirizzo mittente>. Il server risponde con 250 (Ok) per ogni mittente accettato.
5. Successivamente il client indica al server i destinatari del messaggio tramite RCPT TO: <indirizzo destinatario> ed il server invia un codice 250 (Ok) per ogni destinatario accettato.
6. Il client comunica al server l'intenzione di scrivere il corpo del messaggio con DATA. Il server risponde con un codice 354 e indica come marcare il termine del messaggio. I campi come Date, Subject, To, Cc, From vanno inseriti tra i dati dell'e-mail.
7. Completato il messaggio da scrivere, il server memorizza l'e-mail. A questo punto è possibile, scrivere un nuovo messaggio oppure inviare il comando QUIT, dopo il quale il server invia ➤

Primo piano del modulo EM1000 equipaggiato con l'espansione WiFi WA1000. Nella puntata successiva (conclusiva del corso) verrà spiegato come utilizzare l'espansione, a prescindere dal tipo di applicazione realizzata.



i messaggi e risponde con un codice 221 (Closing) e la connessione TCP viene terminata. Quanto descritto non è altro che la normale procedura seguita dal nostro programma client per inviare un'e-mail; in altre parole questo è quanto eseguito quando, ad esempio, si preme il tasto Invia e-mail con Outlook Express oppure con qualsiasi altro programma client di posta elettronica. In questa puntata spiegheremo come programmare il modulo EM1000, affinché possa inviare e-mail in maniera completamente automatica. Questa funzionalità può essere sfruttata in modi differenti ma il più probabile è il monitoraggio di una o più linee del modulo e programmarlo in modo che invii una e-mail quando si verifica una variazione di stato di tale ingresso. La linea può essere collegata ad un sensore di qualsiasi tipo, di cui si vuole realizzare un monitoraggio event-driver (ossia, senza effettuare un'interrogazione ciclica del sensore).

Struttura del programma

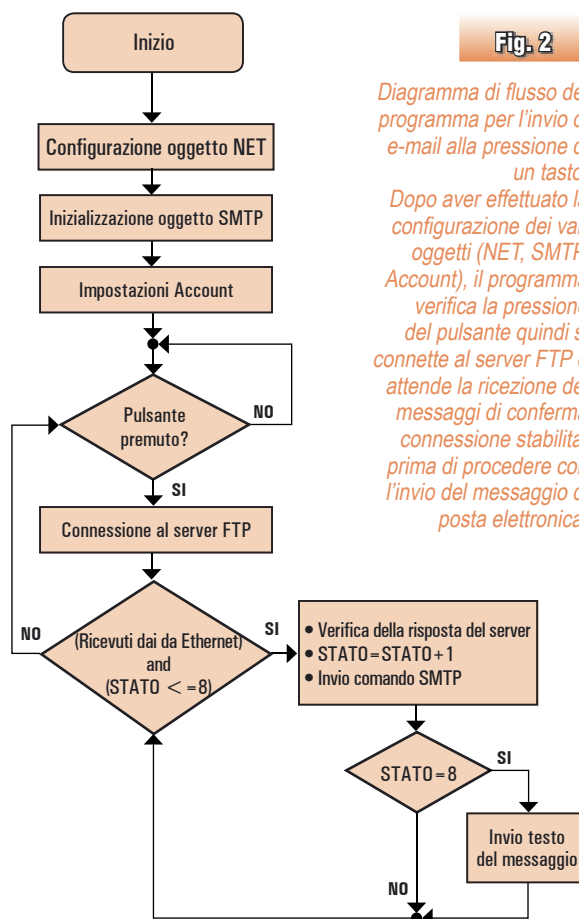
I file utilizzati nel progetto sono tre:

- programma principale (MAIN.tbs),
- funzioni per implementare il protocollo SMTP (SMTP.tbs),
- File di risorse per la codifica delle credenziali (BASE64Table.txt).

Vediamone le caratteristiche salienti.

Il programma principale

Il programma presentato in questo articolo consente di monitorare lo stato del pulsante presente sulla scheda di sviluppo del modulo Tibbo. Premendo tale pulsante, si indica al dispositivo di inviare una e-mail con i parametri specificati. Il diagramma di flusso è riportato in Fig. 2, mentre nel Listato 2 è riportato il codice completo del MAIN.tbs. All'interno della subroutine di inizializzazione (*on_sys_init*) è necessario specificare, come di solito accade, indirizzo IP e subnet mask del modulo Tibbo, tramite le proprietà dell'oggetto *Net*. Oltre a questo, è indispensabile indicare anche il *gateway* utilizzato (riga 5 del Listato 2): le applicazioni viste fino a questa puntata, infatti, non richiedevano di effettuare una connessione all'esterno della nostra rete LAN; ora, invece, è necessario accedere ad un server SMTP che si trova fuori dalla intranet dell'EM1000. Lo scopo principale del gateway è quello di veicolare i pacchetti dati all'esterno della rete locale (LAN). Da notare che "gateway" è un termine generico che indica il servizio di inoltrare i pacchetti verso l'esterno; il dispositivo hardware che porta a termine questo



Listato 2

```

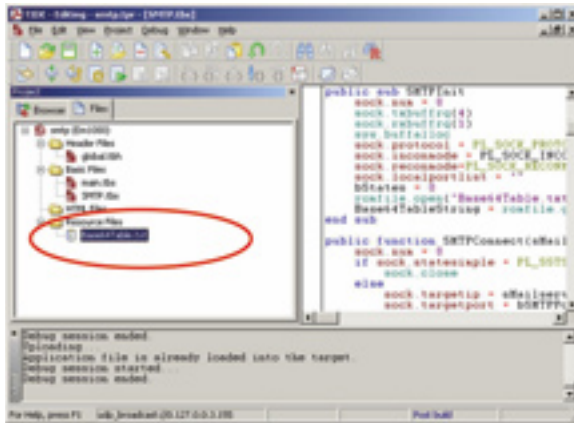
1. dim ConnectState as byte
2. include "global.tbh"
3. sub on_sys_init
4.   Impostazione indirizzo IP
5.   net.gatewayip = "192.168.0.1"
6.   net.netmask = "255.255.255.0"
7.   net.ip = "192.168.0.95"
8.   SMTPInit
9.   SetAccounts("username","password",
10.    "mittente@tele2.it",
11.    "destinatario@gmail.com")
12. end sub
13.
14. sub on_sock_data_arrival
15.   dim s as string
16.   s = sock.getdata(255)
17.   ConnectState = CheckIncomingPacket(s)
18.   if ConnectState = 8 then
19.     AddMessage("prova invio,
20.     by Elettronica In")
21.     AddMessage("")
22.   end if
23. end sub
24.
25. ' Modificare l'indirizzo IP con quello del
26. ' proprio server SMTP
27. sub on_button_pressed
28.   SMTPConnect("212.247.156.13", 25)
29. end sub

```

compito è tipicamente un *router*: l'indirizzo IP da specificare per questo campo, quindi, è quello del router tramite cui effettuiamo la connessione Internet. La funzione *SMTPInit* definisce le proprietà dell'oggetto *Sock*, come già descritto nelle precedenti puntate: il Listato 3 ne riporta il contenuto. Oltre ai comandi già noti, in questa routine sono

Fig. 3

Il Resource File inserito nel progetto serve per la codifica/decodifica delle credenziali di accesso al server SMTP.



Listato 3

```

1. public sub SMTPInit
2.     sock.num = 0
3.     sock.txbufreq(4)
4.     sock.rxbufreq(1)
5.     sys.bufalloc
6.     sock.protocol = PL_SOCK_PROTOCOL_TCP
7.     sock.inconmode =
8.         PL_SOCK_INCONMODE_ANY_IP_ANY_PORT
9.     sock.reconmode=PL_SOCK_RECONMODE_3
10.    sock.localportlist = ""
11.    bStates = 0
12.    romfile.open("Base64Table.txt")
13.    Base64TableString = romfile.getdata
    (romfile.size)
13. end sub

```

utilizzati due comandi nuovi relativi all'oggetto *Romfile*. Questo oggetto consente di accedere ai cosiddetti *File di risorse* preventivamente aggiunti al progetto. Essi sono riportati nella cartella "Resource File" nella finestra *Project* dell'ambiente di sviluppo, come mostrato nella Fig. 3. Tali file non vengono processati dal compilatore, bensì semplicemente aggiunti al file binario inserito nella memoria Flash dell'EM1000. Per accedere al file di risorse è necessario utilizzare il metodo *romfile.open(<nome file>)* con l'indicazione del file da aprire. A questo punto si possono estrarre i dati dal file mediante il metodo *romfile.getdata(<dimensione>)*. Volendo estrarre l'intero contenuto del file è necessario conoscerne la dimensione. Questa informazione può essere fornita in maniera statica in fase di compilazione oppure ricavata dalla proprietà *romfile.size* a run-time. Nel caso specifico riportato nel Listato 3 si carica l'intero contenuto di *Base64Table.txt* all'interno della variabile *Base64TableString*. Tale variabile sarà poi utilizzata per codificare la password e il nome utente da inviare al server. L'ultima funzione presente all'interno della procedura di inizializzazione è *SetAccount*. All'interno di tale funzione bisogna specificare

- il nome utente per l'accesso al server,
- la password per l'accesso al server,
- l'indirizzo e-mail del mittente,
- l'indirizzo e-mail del destinatario.

Oltre all'evento *on_sys_init*, il programma principale gestisce altri due eventi:

- *on_button_pressed*
- *on_sock_data_arrival*

Il primo serve per intercettare la pressione del tasto: subito dopo inizia la procedura per l'invio e-mail; in particolare è richiamata la funzione *SMTPConnect*. Come si nota dal Listato 4, si tratta di una funzione che ha come parametri di ingresso l'indirizzo IP e la porta riferiti al server SMTP con cui si desidera effettuare la connessione. Se l'oggetto *Sock* non risulta già connesso, viene avviata la connessione. I parametri SMTP possono essere recuperati all'indirizzo [1]. È importante una considerazione: la funzione *SMTPConnect* accetta come parametro l'indirizzo IP del server ma non il DNS. Nell'ipotesi di utilizzare tele2 come provider, il relativo SMTP sarà

smtp.tele2.it

cui verrà associato il relativo indirizzo IP:

SMTPConnect("212.247.156.13", 25)

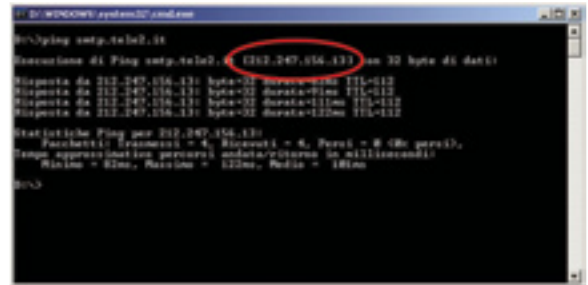


Fig. 4

Recupero dell'indirizzo IP del server SMTP tramite il comando ping.

Listato 4

```

1. public function SMTPConnect(sMailserverIP
   as string, bSMTPPort as byte) as byte
2.     sock.num = 0
3.     if sock.statesimple = PL_SSTS_EST then
4.         sock.close
5.     else
6.         sock.targetip = sMailserverIP
7.         sock.targetport = bSMTPPort
8.         sock.connect
9.     end if
10. end function

```

L'indirizzo IP del server può essere recuperato da Internet o molto più semplicemente aprendo una shell DOS e digitando

ping smtp.tele2.it

L'indirizzo da utilizzare nell'esempio è evidenziato con un cerchio rosso in Figura 4.

Generalmente, le connessioni ai server SMTP avvengono sulla porta 25: il secondo parametro da specificare in *SMTPConnect*, quindi, è il numero 25, salvo diversa indicazione del proprio provider.

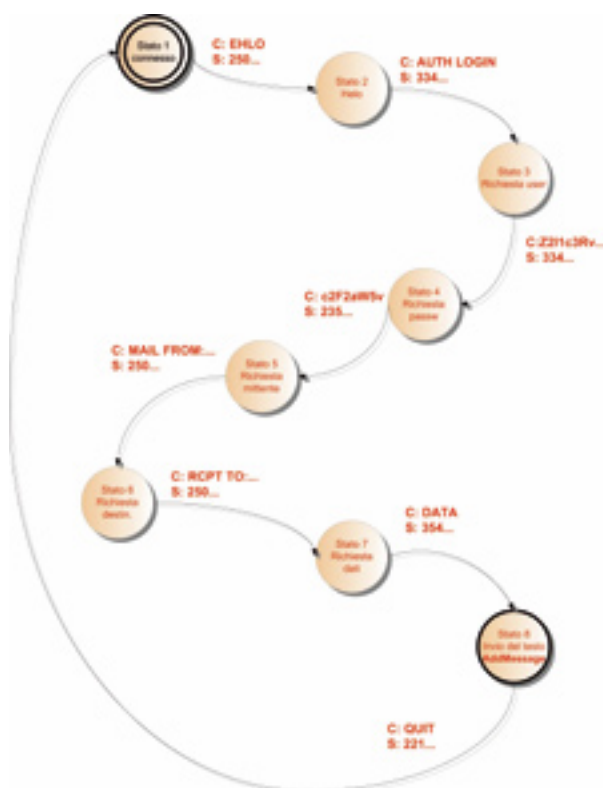
L'ultimo evento che costituisce il main è *on_sock_data_arrival*. Tale routine è richiamata ogni volta che il dispositivo EM1000 riceve dati sulla porta Ethernet. Come si nota anche dal flow-chart (Fig. 2), tale evento è richiamato dopo che è stato pre-

Listato 5

```
1. public function AddMessage
   (sMessage as string) as byte
2. sock.num = 0
3. if bStates = 8 AND len(sMessage) > 0 then
   sock.setdata(sMessage)
   sock.send
   AddMessage = len(sMessage)
4. else
   if bStates = 8 AND len(sMessage) = 0 then
   sock.setdata(chr(13) + chr(10) + "."
   + chr(13) + chr(10))
   sock.send
   end if
   AddMessage = 0
5. end if
6. end function
```

Fig. 5

Diagramma degli stati della funzione *CheckIncomingPacket*.



mutato il pulsante sul modulo e quindi si è instaurata la connessione con il server. Si intercettano le risposte del server tramite il metodo *sock.getdata(255)* e la relativa stringa è passata alla funzione *CheckIncomingPacket(<nome_string>)*, la quale

Listato 6

```
public function CheckIncomingPacket
(sIncoming as string) as byte
dim sSMTPCode as string
dim bPos as byte
dim i as byte
i = 2
bPos = 0
do while instr(1,sIncoming,chr
(13)+chr(10),i) > 0
bPos = instr(1,sIncoming,chr(13)+
chr(10),i-1)
i = i + 1
loop
if bPos > 0 then
sSMTPCode = mid(sIncoming,bPos+2,3)
else
sSMTPCode = left(sIncoming,3)
end if
select case sSMTPCode
case "220":
dim sHELO as string
sHELO = "EHLO " + sock.targetip
sock.setdata(sHELO)
sock.setdata(chr(13)+chr(10))
sock.send
bStates = 1
case "250":
select case bStates
case 1:
sock.num = 0
sock.setdata("AUTH LOGIN")
sock.setdata(chr(13)+chr(10))
sock.send
bStates = 2
case 5:
sock.num = 0
sock.setdata("RCPT TO:<" +
sRecpt + ">")
sock.setdata(chr(13)+chr(10))
sock.send
bStates = 6
case 6:
sock.setdata("DATA")
sock.setdata(chr(13)+chr(10))
sock.send
bStates = 7
case 8:
sock.num = 0
sock.setdata("quit")
sock.send
sock.close
bStates = 0
end select
case "334":
select case bStates
case 2:
sock.num = 0
sock.setdata(Base64Encode(sUserName))
sock.setdata(chr(13)+chr(10))
sock.send
bStates = 3
case 3:
sock.num = 0
sock.setdata(Base64Encode(sPassword))
sock.setdata(chr(13)+chr(10))
sock.send
bStates = 4
end select
case "235":
sock.num = 0
sock.setdata("MAIL FROM:<" +
sSender + ">")
sock.setdata(chr(13)+chr(10))
sock.send
bStates = 5
case "354":
bStates = 8
case else
sock.num = 0
sock.close
bStates = 0
end select
end select
CheckIncomingPacket = bStates
end function
```

effettua il parsing (lettura carattere per carattere) del codice restituito dal server (maggiori dettagli sono riportati nel paragrafo successivo). Tale funzione restituisce un valore intero che rappresenta lo stato di connessione: quando esso è pari ad 8,

Tabella 1 - Tabella di codifica base 64 utilizzata nel protocollo AUTH LOGIN

Valore	ASCII	Valore	ASCII	Valore	ASCII	Valore	ASCII
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

significa che è possibile immettere il testo dell'e-mail. Quest'ultima operazione è eseguita tramite la funzione *AddMessage*, il cui codice è riportato nel Listato 5. Essa ha una duplice funzione:

1. Inserire il testo dell'e-mail.
2. Terminare il testo dell'e-mail; in questo caso sarà sufficiente richiamare la funzione passando una stringa vuota: *AddMessage(“”)*.

La funzione *CheckIncomingPacket*

Possiamo considerarla come il fulcro di tutto il programma, poiché è proprio tramite tale funzione che si implementa effettivamente il protocollo SMTP. Essa è l'interfaccia verso il server, in grado di interpretare le sue risposte ed inviare gli opportuni comandi previsti dallo standard.

Per comprendere meglio il flusso delle operazioni eseguite in questa funzione si riporta il relativo diagramma degli stati in Fig. 5.

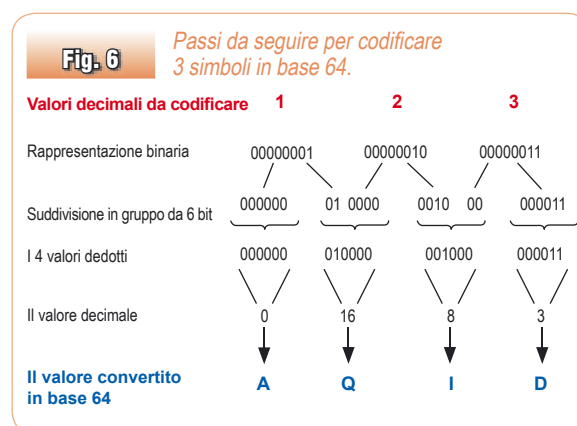
All'interno dei nodi sono riportati gli stati in cui il modulo EM1000 viene a trovarsi, mentre gli archi rappresentano le transizioni di stato. Sugli archi sono riportati i comandi inviati dal client (C) ed i codici ottenuti come risposta dal server (S).

Codifica base 64

Il Base64 è un sistema di numerazione posizionale (ossia a corrispondenza “simbolo per posizione”) che usa 64 simboli. Viene usato principalmente come codifica di dati binari nelle e-mail, per con-

vertire i dati nel formato ASCII. L'algoritmo che effettua la conversione suddivide la parola (nel nostro caso password e username) in gruppi di 6 bit, i quali possono, quindi, contenere i valori da 0 a 63. Ogni possibile valore viene convertito in un carattere ASCII secondo la Tabella 1.

Ecco spiegata l'utilità del file *Base64Table.txt* aggiunto al progetto. In questo modo si implementa una LUT (look-up-table) in cui a ciascuna posizione (da 0 a 63) corrisponde uno specifico carattere ASCII.



L'algoritmo causa un aumento delle dimensioni dei dati del 33%, poiché ogni gruppo di 3 byte viene convertito in 4 caratteri. Ad esempio, dati 3 byte di valore 1, 2, 3, partendo dalla loro rappresentazione binaria, applichiamo l'algoritmo come ➤

Listato 7

```
function Base64Encode(sSourceString as string)
as string
    dim sEncode as string
    dim sSubString as string
    dim s as string
    dim i as byte
    dim j as byte
    dim k as byte
    dim bTemp as byte
    for k = 1 to len(sSourceString)
        dim sBinString as string
        sSubString = mid(sSourceString,k,3)
        for i = 1 to len(sSubString)
            dim sTemp as string
            sTemp = mid(sSubString,i,1)
            sTemp = bin(asc(sTemp))
            sTemp = right(sTemp,len(sTemp)-2)
            for j = len(sTemp) to 7
                sTemp = "0" + sTemp
            next j
            sBinString = sBinString + sTemp
        next i
        for i = 1 to len(sBinString)
            dim sTemp as string
            sTemp = mid(sBinString,i,6)
            for j = len(sTemp) to 5
                sTemp = sTemp + "0"
            next j
            i = i + 5
            sTemp = "&b" + sTemp
            bTemp = val(sTemp)
            sTemp = mid(Base64TableString,
            bTemp + 1,1)
            sEncode = sEncode + sTemp
        next i
        j = fMod(len(sEncode), 4)
        for i = 1 to j
            sEncode = sEncode + "="
        next i
    next k
    Base64Encode = sEncode
end function
```

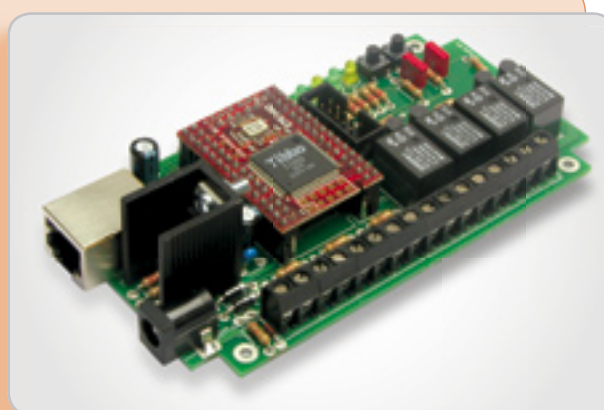
descritto in Fig. 6.

La sequenza di valori 123 viene quindi convertita nella sequenza AQID. Nel caso in cui i bit non siano multipli di 6, vengono inseriti alla fine tanti bit nulli (0) quanti ne occorrono e nella codifica viene aggiunto un simbolo '=' per ogni coppia di bit mancanti. L'algoritmo appena descritto è realizzato mediante il codice riportato nel Listato 7.

Mail Server: il progetto pratico

La puntata successiva concluderà il corso dedicato all'EM1000: essa sarà interamente dedicata all'Add-On WiFi WA1000, oggetto che può essere utilizzato esattamente come abbiamo fatto con l'EM1000 nelle esercitazioni pratiche. Nello stesso fascicolo, inoltre, pubblicheremo il primo di una serie di progetti pratici basati su questo potente Ethernet server: si tratta di un e-mail server con telecontrollo, in grado di inviare una serie di messaggi di posta elettronica ad indirizzi e-mail differenti quando uno qualsiasi dei suoi quattro ingressi cambia di stato. È, altresì, in grado di gestire quattro carichi distinti attraverso altrettanti relé di cui è dotato.

Sia la telegestione che la configurazione vengono effettuate attraverso un'interfaccia Web, implementata direttamente a bordo del dispositivo che, pertanto, oltre che da e-mail server, si comporta anche da web server. Come potete constatare, il progetto fa uso di buona parte delle nozioni acquisite durante il corso, ossia quelle relative alla gestione degli I/O (quarta puntata), del Web Server (sesta puntata) e dell'e-mail server (puntata corrente). Vi anticipiamo, inoltre, che sul fascicolo 130 vi mostreremo come realizzare la versione WiFi dello stesso progetto.



Conclusioni

L'applicazione descritta in questa puntata è molto utile per monitorare lo stato di uno o più sensori connessi al modulo in maniera event-driven, ossia senza la necessità che l'utente interroghi ciclicamente il dispositivo. Sarà il dispositivo stesso, infatti, ad avvertire l'utente che si è verificata una variazione nello stato logico della linea e a comunicarlo direttamente tramite e-mail. Nella prossima puntata, che concluderà il corso, vedremo come utilizzare l'ultimogenito di casa Tibbo, ossia il modulo WA1000. Si tratta di un piccolo add-on, da montare sull'EM1000, che garantisce al dispositivo le funzionalità di connessione wireless, molto richieste per realizzare, ad esempio, embedded web server per applicazioni portatili, senza però alterare le funzionalità analizzate fino ad ora. Appuntamento quindi al mese prossimo in edicola, oltre che per il corso, anche per la presentazione del primo progetto pratico, del quale trovate, in questa pagina, una breve anticipazione.

Link Utili

- [1] <http://www.sandrodiremigio.com/documenti/tutorials/pop3smtpnntp.htm>
- [2] RFC 2821 – Simple Mail Transfer Protocol: <http://www.ietf.org/rfc/rfc2821.txt>
- [3] RFC 1869 - SMTP Service Extensions: <http://www.faqs.org/rfcs/rfc1869.html>
- [4] RFC2554 – SMTP Service Extension for Authentification: <http://www.faqs.org/rfcs/rfc2554.html>
- [5] RFC3548 - The Base16, Base32, and Base64 Data Encodings: <http://rfc.net/rfc3548.html>